# NEURO-FUZZY MODEL OF SUPERELASTIC SHAPE MEMORY ALLOYS WITH APPLICATION TO SEISMIC ENGINEERING

A Thesis

by

OSMAN ESER OZBULUT

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2007

Major Subject: Civil Engineering

# NEURO-FUZZY MODEL OF SUPERELASTIC SHAPE MEMORY ALLOYS WITH APPLICATION TO SEISMIC ENGINEERING

A Thesis

by

OSMAN ESER OZBULUT

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

| | |
|---|---|
| Chair of Committee, | Paul N. Roschke |
| Committee Members, | Jose Roesset |
| | Harry Hogan |
| Head of Department, | David Rosowsky |

August 2007

Major Subject: Civil Engineering

# ABSTRACT

Neuro-Fuzzy Model of Superelastic Shape Memory Alloys with Application to Seismic Engineering. (August 2007)

Osman Eser Ozbulut, B.S., Istanbul Technical University

Chair of Advisory Committee: Dr. Paul N. Roschke

Shape memory alloys (SMAs) have recently attracted much attention as a smart material that can be used in passive protection systems such as energy dissipating devices and base isolation systems. For the purpose of investigating the potential use of SMAs in seismic engineering applications a soft computing approach, namely a neuro-fuzzy technique is used to model dynamic behavior of CuAlBe shape memory alloy wires. Experimental data are collected from two test programs that have been performed at the University of Chile. First, in order to evaluate the effect of temperature changes on the behavior of superelastic SMA wires, a large number of cyclic, sinusoidal, tensile tests are conducted at various temperatures. Second, to assess dynamic effects of the material, a series of laboratory experiments are conducted on a scale model of a three story model of a building that is stiffened with SMA wires and given excitation by a shake table.

Two fuzzy inference systems (FISes) that can predict hysteretic behavior of CuAlBe wire have been created using these experimental data. Both fuzzy models employ a total of three input variables (strain, strain-rate, and temperature or prestress) and one output variable (predicted stress). Values of the initially assigned membership functions for each input are adjusted using a neural-fuzzy procedure to accurately predict the correct stress level in the wires. Results of the trained FISes are validated using test results from experimental records that had not been previously used in the training procedure.

Finally, numerical simulations are conducted to illustrate practical use of these wires in a civil engineering application. In particular, dynamic analysis of a single story frame and a three story benchmark building that are equipped with SMA damping elements are conducted. Then, an isolated bridge that utilizes a linear rubber bearing together with SMA elements is analyzed. Next, in order to show recentering ability of SMAs, nonlinear time history analysis of a chevron like braced frame is implemented. The results reveal the applicability for structural vibration control of CuAlBe wire whose highly nonlinear behavior is modeled by a simple, accurate, and computational efficient FIS.

# DEDICATION

To my beloved mother

# ACKNOWLEDGEMENTS

I would like to thank my advisor Prof. Paul N. Roschke for his invaluable guidance and suggestions. Without his support, I would have had many more difficulties during my studies. I also want to thank to the faculty of Structures Division of Zachry Department of Civil Engineering for teaching me during several classes that I took and individual meetings.

I want to specially thank Prof. Moroni and Prof. Sarrazine at the University of Chile for sharing with me their experimental test results on CuAlBe shape memory alloys and answering my questions.

I would like to also thank all of my friends in our research group, especially David Shook, and the others that we had the opportunity to meet.

Finally, I would like to send special thanks overseas to my family for their infinite support during my study here. I also want to thank my uncles, Mahmut Ozbulut and Erkan Bilhan, for their support and guidance.

# NOMENCLATURE

| | |
|---|---|
| ANFIS | Adaptive neuro-fuzzy inference system |
| Al | Aluminum |
| $A_f$ | Austenite finish temperature |
| $A_s$ | Austenite start temperature |
| Cr | Chromium |
| Cu | Copper |
| $E_D$ | Energy loss per cycle |
| $E_{so}$ | Maximum strain energy per cycle |
| Fe | Iron |
| FIS | Fuzzy inference system |
| MANSIDE | Memory alloys for new seismic isolation and energy dissipation devices |
| $M_d$ | Austenite stabilization temperature |
| $M_f$ | Martensite finish temperature |
| $M_s$ | Martensite start temperature |
| Mn | Manganese |
| Ni | Nickel |
| PGA | Peak ground acceleration |
| Si | Silicon |
| SMA | Shape memory alloy |
| Ti | Titanium |
| Zn | Zinc |

# TABLE OF CONTENTS

Page

Page

# LIST OF TABLES

Page

# LIST OF FIGURES

Page

Page

Page

Page

# 1. INTRODUCTION

## 1.1. General

Natural hazards such as earthquakes, hurricanes or tsunamis often cause significant damage to the built environment. For example, structural and non-structural damage can be extensive and loss of life is an accompanying tragedy. To better understand the consequences of these natural hazards, results of the Northridge (1994) and Izmit (1999) ground motions are illustrated in Fig. 1. The Northridge (1994) earthquake caused the death of sixty people and is estimated to have cost up to forty billion dollars in damages. Whereas only a few buildings collapsed, it is considered to be one of the worst earthquakes in the history of the United States with regard to the amount of property damage. By comparison, the Izmit (1999) earthquake that occurred in Turkey, was one of the longest ground motions ever recorded in the world, and it killed more than 17,000 people. This large loss of life was largely caused by the collapse of many buildings.

As a result of strong motion earthquakes the destruction of many structures in various countries throughout the world reveals the need for retrofitting existing structures that have been built using outdated seismic codes. Damage observed after the above-mentioned earthquakes and other recent disasters are leading civil engineers to investigate innovative materials and design methods for this purpose. One of the more promising techniques for anti-seismic resistance of structures involves the use of shape memory alloys (SMAs). This class of metals is known to exhibit several exceptional properties. These favorable properties and other attractive features have encouraged many researchers to consider use of SMAs for earthquake resistance. Their potential application in seismic engineering as passive dampers or retrofitting materials for both buildings and bridges is being carefully investigated.

---

This thesis follows the style of *Journal of Structural Engineering*.

**Fig. 1.** Damage to a bridge and residential housing after (a) Northridge (1994) and (b) Izmit (1999) earthquakes

SMAs owe their unique properties to solid-solid phase transformations that occur as a result of thermal or mechanical changes. SMA materials undergo transformations between two stable phases, austenite and martensite, and return to their original u.deformed position due to either a change of temperature, the shape memory effect, removal of stresses, or the superelastic effect. Since the material behavior is hysteretic and yet does not have any residual displacement, a substantial amount of energy dissipation capacity and recentering ability is offered.

Since this unique behavior of shape memory alloys was discovered in the 1960s, a wide variety of applications have been implemented including those in the medical, aerospace, and automotive industries. Although the application of SMA materials to civil engineering is relatively new, a number of doctoral and master's level studies have been completed (Auricchio 1995, Fugazza 2003, Penar 2005, and McCormick 2006). For example, Auricchio (1995) developed one and three dimensional thermomechanical constitutive models for SMA materials. Fugazza (2003) proposed a uniaxial constitutive model for superelastic SMAs that are candidates for installation into civil engineering

structures. Both of these research studies included numerical simulations to show the ability of their models and exemplify the dynamic behavior of SMAs. More recently, Penar (2005) studied NiTi shape memory material with the goal of developing a recentering beam-column connection for steel frames. Also, McCormick (2006) investigated cyclical properties of large diameter shape memory alloys for structural applications.

Despite the fact that recent studies have contributed significantly to comprehension of the potential for using shape memory alloys in civil engineering applications, there are still many questions to be answered before full-scale applications can be implemented. To this end, this thesis proposes a material model for superelastic SMAs using a soft computing technique, namely a neuro-fuzzy model, in order to represent behavior of the material during cyclical excitations and in different ambient temperatures. With linear and nonlinear numerical simulations, it is shown that earthquake response of a structure integrated with SMA elements can be significantly mitigated.

In the sections that follow, a number of aspects related to implementation of SMAs in civil engineering structures will be discussed. First, a general introduction to the material properties and behavior of SMAs is given and some of their applications outside of civil engineering are described. Next, mechanical properties and modeling of SMAs along with applications more specific to infrastructure and large civil engineering structures are discussed. Before proposing a soft computing approach to model CuAlBe SMA wires, experimental data obtained from cyclic tensile tests and shake table tests at the University of Chile are introduced. Then, a neuro-fuzzy technique for modeling superelastic SMAs is presented along with two different fuzzy models. The first model predicts behavior of material at various temperatures while the second considers dynamic effects on the response of SMAs. Finally, in order to investigate potential applications of SMAs in seismic engineering and to evaluate applicability of developed model to numerical simulations, time history analyses of several civil engineering structures are conducted.

# 2. OVERVIEW OF SHAPE MEMORY ALLOYS

## 2.1. General Characteristics of Shape Memory Alloys

Shape memory alloys (SMAs) are an extraordinary class of metals that exhibit several unique properties such as large deformation recoveries upon heating (shape memory effect) or upon unloading (superelasticity), high strength capacity, important fatigue resistance and high damping capacity. An SMA has two stable phases: martensite and austenite. Martensite also has two forms that are termed twinned and detwinned (Fig. 2). Typically, material in the martensite form is stable at low temperatures and high stresses while the austenite (also named parent phase) material is stable at high temperatures and low stresses. SMAs owe their peculiar characteristics to the solid-to-solid transformations between these two phases.

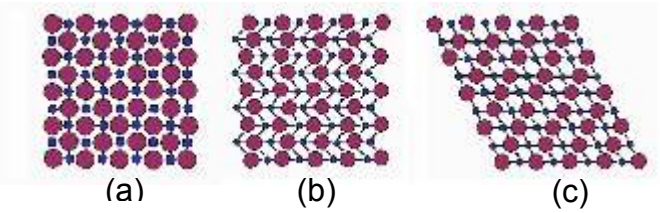



**Fig. 2.** SMA phases: (a) austenite; (b) twinned martensite; and (c) detwinned martensite (SMART Laboratory at Texas A&M University, 2006)

Phase transformations may be temperature-induced or stress-induced. An SMA that is in the austenite state transforms to the martensite state upon cooling. A reverse transformation, from martensite to austenite, takes place when the material is heated (Fig. 3). The four characteristic temperatures of an SMA are defined as follows: $M_s$ and $M_f$ are the start and finish temperatures of the transformation from austenite to martensite, while $A_s$ and $A_f$ are the temperatures at which a transformation from martensite into austenite starts and completes.

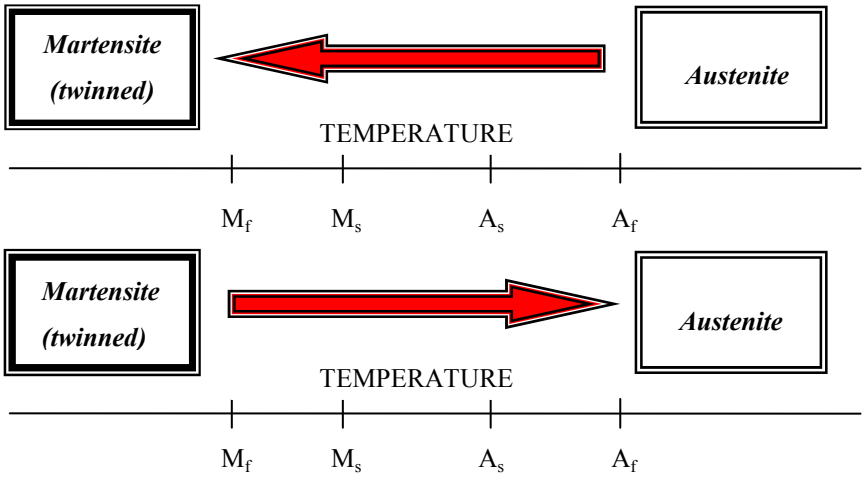



**Fig. 3**. Temperature-induced phase transformations

### 2.1.1. Shape Memory Effect

Two important characteristics associated with the phase transformations of SMAs are the *shape memory effect* and the *superelastic effect*. Due to the shear-like mechanism of martensetic transformations, mechanical loading helps to initiate the phase transformations (Otsuka and Ren 2005). When stress is applied to a material that is in a twinned martensite state, it is possible to detwin martensite when it is loaded above a threshold level of stress. Until the detwinning process is complete the stress level remains constant. Upon unloading the detwinned shape of the martensite material is maintained. However, heating the material to a temperature that is above $A_s$ recovers the large strains. This is known as the shape memory effect (Fig. 4).

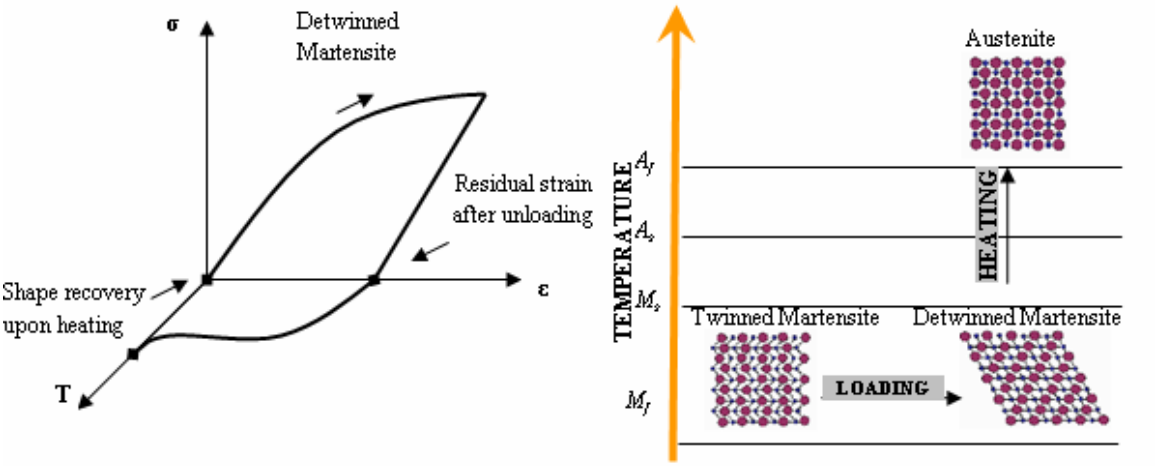



**Fig. 4**. Shape memory effect (SMART Laboratory at Texas A&M University, 2006)

### 2.1.2. Superelastic Effect

Another unique property of SMAs is superelasticity. Material in the austenite state starts to transform to the martensite state when it is loaded above a certain stress level (*forward transformation*). If loading continues, the material becomes completely martensite at some level of strain and, the stress again starts to increase in an elastic way. Upon unloading the martensite is no longer stable, and if the temperature is above $A_f$, a *reverse transformation* to the austenite state occurs that results in complete shape recovery and a substantial hysteretic loop. If the temperature exceeds $M_d$, which is the temperature at which the austenitic phase is stabilized, the martensite phase cannot be stress-induced. If the temperature is between $A_f$ and $A_s$, the material recovers part of the deformation, and there is no residual strain at the end of unloading. If the temperature is below $A_s$, the martensite remains deformed during unloading. Fig. 5 shows a schematic representation of superelasticity.

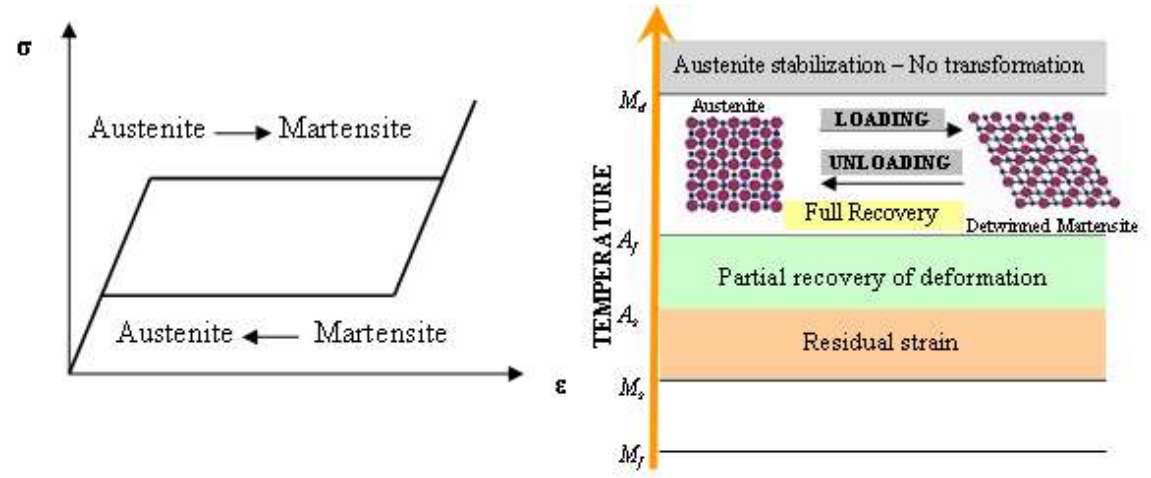



**Fig. 5**. Superelastic effect (SMART Laboratory at Texas A&M University, 2006)

## 2.2. Commonly Used Shape Memory Alloys

### 2.2.1. Shape Memory Materials

There are several families of commercially available SMAs including NiTi-based alloys, Cu-based alloys and Fe-based alloys. The nickel-titanium (NiTi) alloys are often represented by the name 'nitinol' which is an abbreviation for "**Ni**ckel-**Ti**tanium at **N**aval **O**rdinance **L**aboratory" after a superelastic effect was observed by researchers at the U.S. Naval Ordinance Laboratory for the first time in 1961 (Lin, 1996). This binary system is based on an almost equiatomic compound of nickel and titanium. The Ni:Ti ratio changes transformation temperatures from -50 °C to 95 °C. A small amount of chromium (Cr), copper (Cu) or iron (Fe) is added as a third metal to compose a ternary alloy that has improved properties for commercial use. For example, nitinol is the most widely investigated alloy for use in devices intended for seismic applications. Examples of beneficial nitinol properties include excellent superelasticity and high corrosion resistance. However, difficulties in processing the material and the need for special machining techniques increase the cost of the material substantially (Special Metals, 2006).

By comparison, Cu-based alloys are less expensive than nitinol, although the cost again increases due to techniques required for manufacturing the material into useful shapes. CuZnAl was the first copper-based alloy to be commercially exploited. Even at room temperature stabilization of the martensite phase is a major problem in using this alloy for some applications because of aging. The addition of the nickel or beryllium to CuAl binary extends the transformation range significantly (Copper Development Association Inc., 2006).

Some of the salient physical, mechanical, and superelastic properties of NiTi, CuAlBe, and CuZnAl are presented in Table 1 along with the properties of structural steel for comparison purposes.

**Table 1.** Properties of several SMAs and structural steel (Compiled from Special Metals Coorperation, 2006 and Nimesis Inteligent Materials, 2006)

| Material | CuZnAl | CuAlBe | NiTi | Steel |
|---|---|---|---|---|
| | | Physical properties | | |
| Composition | 15-30% Zn and 3-7% Al | | 55.8% Nickel | |
| Melting point | 950-1,020 ºC | 970-990 ºC | 1,300 ºC | |
| Density | 7.8-8.0 g/cm$^3$ | 7.3 g/cm$^3$ | 6.5 g/cm$^3$ | 7.85 g/cm$^3$ |
| Electrical resistivity | 0.7-0.12 Ω.m × $10^{-6}$ | 0.07-09 Ω.m × $10^{-6}$ | 0.5-1.1 Ω.m × $10^{-6}$ | 0.13-1.25 Ω.m × $10^{-6}$ |
| Coefficient of thermal expansion | 17 $10^{-6}$/K | | 6.6-10 $10^{-6}$/K | 11-16.6 $10^{-6}$/K |
| | | Mechanical properties | | |
| Modulus of elasticity | 90 | 70-100 | 41-75 GPa | 190-210 |
| Ultimate tensile strength | 800-900 MPa | 900-1,000 MPa | 1,068 MPa | 276-882 |
| Elongation to failure | 15 % | 15 % | 30% | 10-32% |
| Maximum strain recovery | 4-5% | 4-5% | 8% | |
| | | Superelastic properties | | |
| Loading plateau stress | | | 379 MPa | |
| Unloading plateau stress | | | 138 MPa | |
| Trans. temperature ($A_f$) | 10 ºC | 15 ºC | 5 to 18 ºC | |

Fe-based alloys have also attracted the interest of researchers in recent years. Iron-manganese-silicon based (Fe-Mn-Si-X) SMAs have been investigated as an alternative shape memory alloy (Janke *et al.,* 2005; Sawaguchi *et al.,* 2005; Sawaguchi *et al.,* 2006). It is found that these materials have a high damping capacity and, through appropriate thermomechanical treatment, superelastic characteristics of the alloy can be improved. Although cost of the material is relatively low, more research is required before these alloys can be used in seismic applications.

### 2.2.2. Applications of Shape Memory Alloys

There is a wide variety of applications of shape memory alloys including biomedical, aerospace, automotive and other industries. For example, shape memory alloys are being used in a large number of medical applications. Broken bones can be mended with SMA staples. The use of SMA accelerates the healing process by exerting a compressive force on the broken bone at the fracture point. A spinal vertebra spacer made with an SMA is another application in the medical field. These spacers permit relative motions of two vertebrae by changing their shapes in such a manner as to prevent failure of arthrodesis (Gil and Plannel, 1998). Another orthopedic application includes SMA gloves that help regain the activity of hand muscles (Fig. 6). Heating the gloves exploits a temperature-induced transformation, and shortens the SMA wires. The wires return to their original shapes upon cooling; this results in opening the hand.

**(a)** **(b)**

**Fig. 6**. SMA glove: (a) original (low temperature) shape; (b) shape after heating (adapted from AMT (@ Medical Technologies n.v.), 2002 )

Applications of shape memory alloys in heart surgery include artificial heart muscles, vane cave filters that stop blood clots, and stents that have a self-expanding ability to counteract a disease-induced localized flow constriction (Morgan, 2004). Eyeglasses frames that can be bent back and forth, orthodontic archwires that are used in dentistry, rehabilitation devices, and clinical instruments are among the other medical applications.

Unique properties of shape memory alloys have also attracted interest in the aerospace industry. SMA adaptive wings are used to adjust the shape of an air foil to the present flight conditions by exploiting temperature-induced transformations (Fig. 7). An adaptive winglet that keeps the wing in an optimal shape that is appropriate to incoming flow is another aerospace application.

**(a)** **(b)**

**Fig. 7**. Adaptive wings: (a) straight wing, (b) bent wing (adapted from Smart Material Products, 2006)

Finally, other industrial applications of shape memory alloys include temperature control systems with memory alloys that immediately shut down the system in the presence of a temperature increase, heat-activated fasteners, attitude control systems of stationary satellites, and cellular telephone antennas.

While this brief review of SMAs in industrial applications is not intended to be exhaustive, it serves as an introduction to the primary application of this work, namely as a control device that is installed in a civil engineering structure. In the following

section, first the mechanical properties of several SMAs and their modeling techniques are described, and then some applications of this material in seismic engineering are introduced.

# 3. SHAPE MEMORY ALLOYS IN SEISMIC ENGINEERING: MECHANICAL PROPERTIES, MODELING AND APPLICATIONS

## 3.1. Introduction

Recent earthquakes have revealed a lack of conventional seismic design techniques that can be easily applied to civil engineering structures in high risk zones. This shortfall does not correlate with the extensive efforts of researchers to apply modern techniques to improve seismic response of the structures in the recent past. Active, semi-active, hybrid and passive vibration control techniques are being developed to establish an acceptable level of seismic protection (Spencer and Nagarajaiah, 2003). Passive structural control techniques are being advocated that do not require an external power source for operation of a damping device. To date they are being widely implemented in practical civil engineering projects (Soong and Dargush, 1997). More recently, shape memory alloys have attracted a great deal of attention as a smart material that can be used in passive protection systems such as energy dissipating devices and base isolation systems (Song *et al.,* 2004, Wilson and Wesolowsky, 2005).

This section reviews the applications of shape memory alloys and especially superelastic SMAs in civil engineering structures. First, mechanical properties of SMAs are discussed, and then some of the material models, especially those that have been proposed for seismic applications, are presented. Finally, analytical and experimental studies on SMA-based devices for building and bridge applications are introduced.

## 3.2. Mechanical Properties of Shape Memory Alloys

Many experimental studies have been carried out to investigate mechanical characteristics of SMAs for applications in vibration control of structures. Although the superelastic effect of some SMAs that possess recentering ability and damping capacity have fascinated many civil engineers, a complete understanding of the mechanical

behavior of an SMA is required before applying it to a real structure. To this end, the effects of temperature, cyclic loading, strain rate, strain amplitude and thermomechanical processing are outlined in the following sections. The influence of each important behavioral parameter is briefly discussed.

### 3.2.1. Influence of Temperature

Since phase transformations of SMAs are not only dependent on mechanical loading but also on temperature, experimental studies are required to determine the degree to which superelastic properties of SMAs are affected by variations of temperature. Dolce and Cardone (2001) investigated the effects of temperature on superelastic behavior of NiTi wires that have a diameter of 1 to 2 mm. They report that the critical stress that initiates the phase transformation changes noticeably with temperature; more specifically an increase in temperature corresponds to a linear increase in transformation stress. The amount of dissipated energy is found to be almost insensitive to the temperature. In contrast, the equivalent viscous damping decreases linearly with an increase in the secant stiffness (Fig. 8).

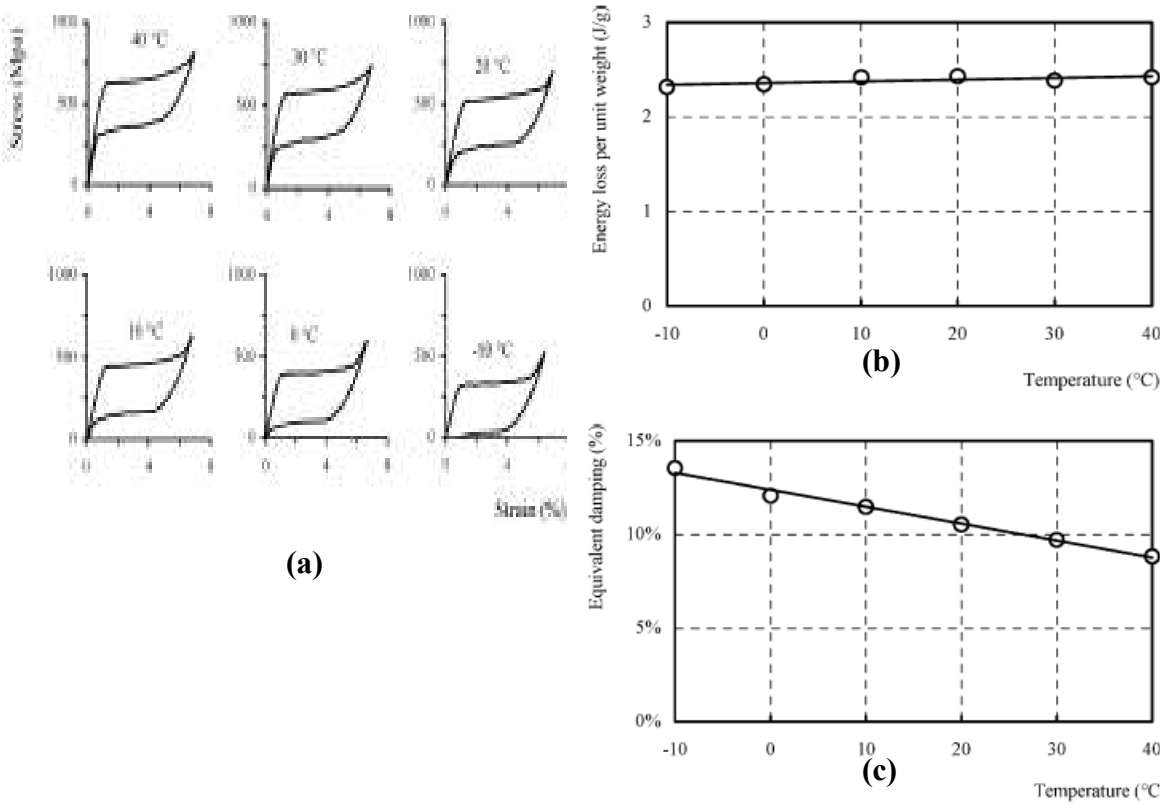



**Fig. 8.** (a) Strain-stress curve of nitinol wires at different temperatures; (b) energy loss per unit weight versus temperature; and (c) equivalent damping versus temperature (Dolce and Cardone, 2001)

Strandel *et al*. (1995) observed similar effects of temperature on material behavior as discussed above. Fig. 9 shows stress-strain curves of NiTi alloys for various temperatures. As expected when the ambient temperature of the metal is lower than $A_f$, residual strains occur.

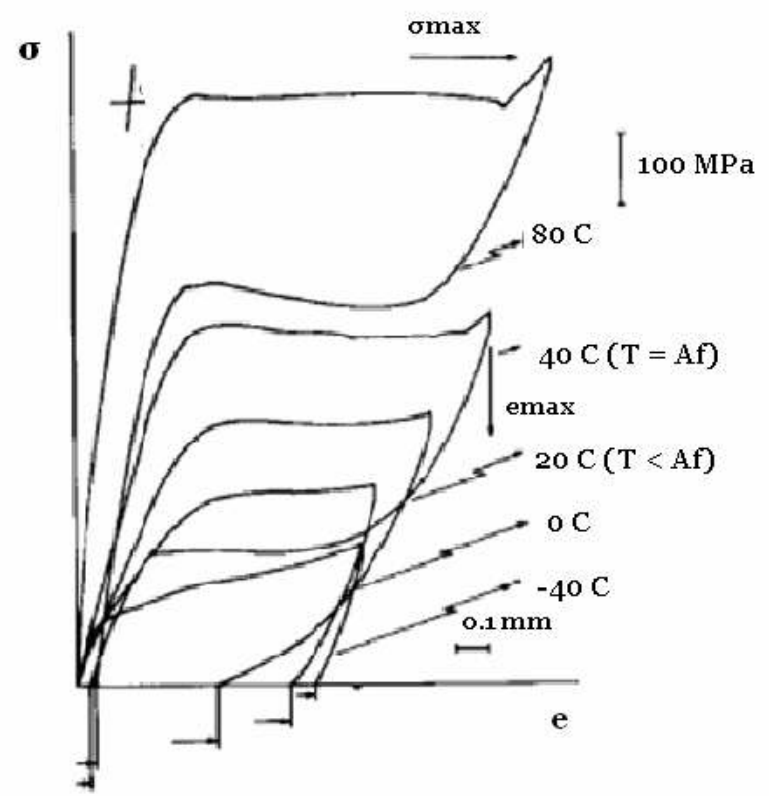



**Fig. 9**. Influence of temperature on superelastic behavior of NiTi alloys (Strandel *et al*., 1995)

### 3.2.2. Influence of Cyclic Loading

Knowledge of material behavior under repeated loading conditions is important for seismic applications of SMAs. DesRoches *et al*. (2004) tested wires having different diameters as well as bars of various sizes to evaluate cyclical properties of NiTi SMAs. Superelastic characteristics of both wires and bars were found to be excellent overall. Nevertheless, an increase in residual strain proportional to the number of loading cycles is mentioned. In another study Dolce and Cardone (2001) show that the hysteresis loops of a superelastic SMA stabilize after a certain number of loading cycles. Cyclical loading narrows the hysteresis loop and, as a result, decreases the dissipated energy. Wolons *et al*. (1998) also noticed the same effect. As shown in Fig. 10, the stress level

for the forward transformation decreases, while for the reverse transformation the stress increases slightly. As a consequence, the area of the hysteresis loop decreases.

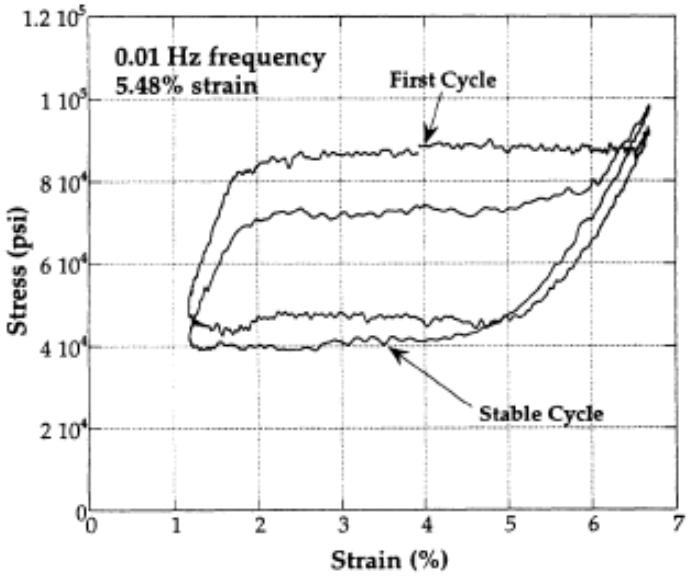



**Fig. 10**. Effect of cyclical loading (Wolons *et al*., 1998)

In a related study Tamai and Kitagawa (2002) performed a series of tests on 1.7 mm diameter NiTi wires. They point out that the increment of the residual strain and decrement of the dissipated energy per cycle progressively reduces during cyclical loading. These studies show that cyclic loading causes an increase in residual displacement and a reduction in hysteresis. Therefore, designers of SMA damping systems need to be aware of potential deterioration in superelastic properties of SMAs due to cyclical loads.

### 3.2.3. Influence of Strain Rate

Several research efforts have been carried out to clarify the influence of the strain rate on superelasticity of shape memory alloys. Experimental studies show that the

effect of the strain rate is due to self-heating (or cooling) of the SMA material during loading (or unloading). High loading frequencies do not allow the material to transfer latent heat to the environment. As a result, the temperature of the material changes and this, in turn, alters the shape of the hysteresis loops and the transformation stresses as described above (Wu *et al*., 1996). Dolce and Cardone (2001) determined that the low frequency range for seismic consideration (< 0.01 Hz) does not have a considerable effect on behavior of the wires, but that higher frequencies (0.2-4 Hz) cause a reduction in damping capacity of the wires. Similar results were observed by Wolons *et al*. (1998), Piedboeuf *et al*. (1998), and Sun and Rajapakse (2003). Tobushi *et al*. (1998) carried out an extensive investigation into the effects of strain on superelastic characteristics of TiNi wires. They also observed that superelastic behavior does not depend on small strain rates (< 2%/min). However, contrary to findings of other researchers, they report an increase in dissipated energy for increasing strain rates.

### 3.2.4. Influence of Strain Amplitude

It is well known that the damping capacity of superelastic SMAs increases with increasing strain amplitude. Dolce and Cardone (2001) performed tensile tests on a 1.84 mm diameter NiTi wire with different strain amplitudes (Fig. 11). They concluded that the equivalent viscous damping increases linearly until approximately a 5% strain amplitude is reached and that it starts to decrease thereafter.

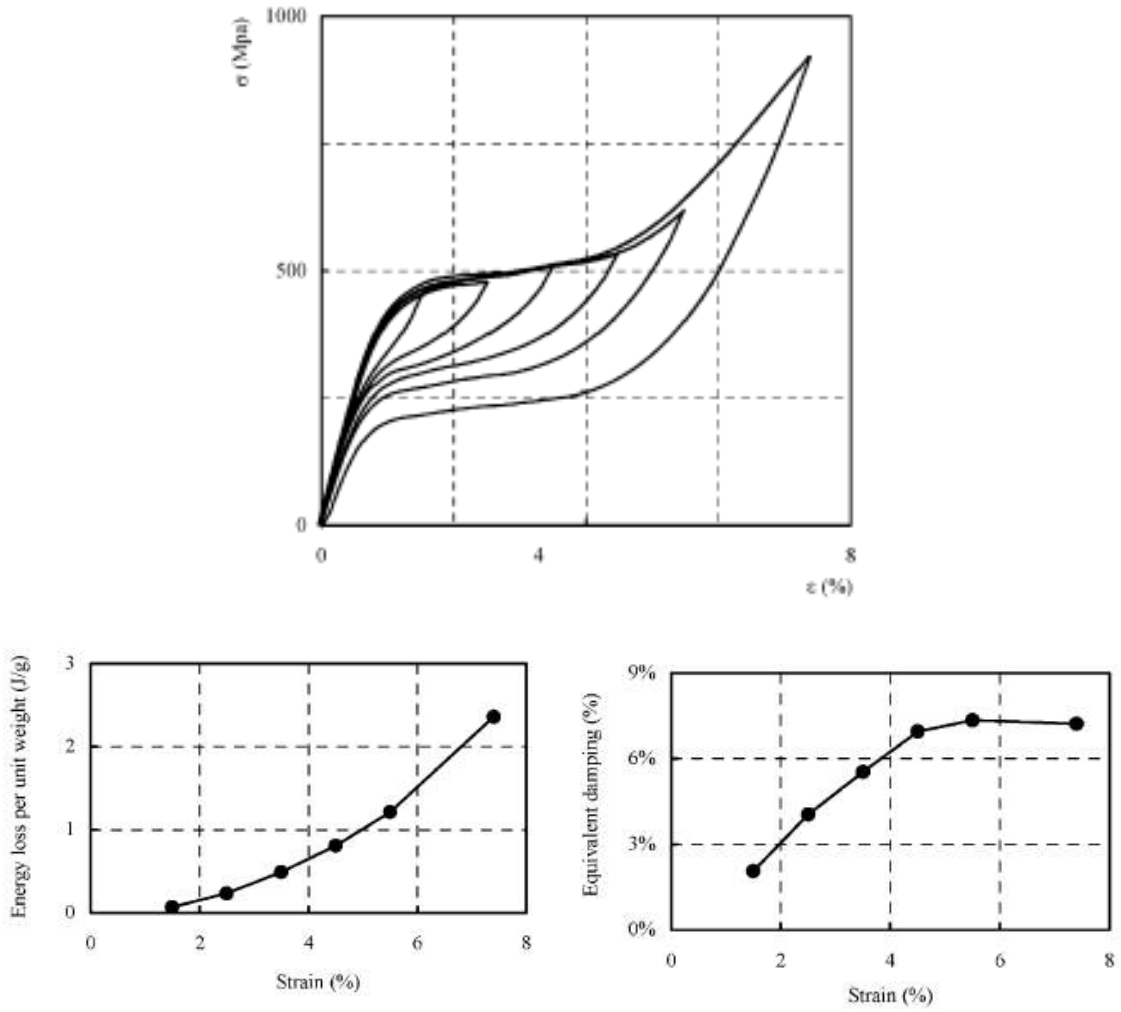



**Fig. 11**. Effect of strain amplitude on damping capacity of SMA wires (Dolce and Cardone 2001)

DesRoches *et al*. (2004) also observed a similar trend. They point out that both strain hardening at large deformations and a reduction in the forward yield plateau with large strains are the reasons for a decrease in equivalent damping at large strain amplitudes.

### 3.2.5. Influence of Thermomechanical Processing

In addition to the influence of strain amplitude, thermomechanical treatment greatly affects the mechanical behavior of SMAs. Nemat-Nasser and Guo (2006) investigated the effect of annealing temperature on 4.5 mm diameter NiTi wires. As shown from Fig. 12, the transformation stresses decrease as the annealing temperature increases. Also, the dissipated energy changes with the annealing temperature.

Otsuka and Ren (2005) examined the effect of cold working followed by annealing. Fig 13(a) shows test results from NiTi wires that are annealed at 673 K for one hour, while Fig. 13(b) illustrates the results from the specimens that are heavily cold-drawn and, after that annealed in the some way. While superelasticity is not observed for the first case, very good superelastic and shape memory characteristics are present for the latter case.

## 3.3. Modeling of Shape Memory Alloys

The highly favorable mechanical properties of superelastic SMAs discussed in the preceding sections have been recognized by the seismic engineering community over the past decades. Numerous researchers have reported studies on structural systems that are strengthened with SMA-based protection devices. Others have proposed material models for the purpose of estimating the modification of structural response due to inclusion of SMA devices. Before implementing an SMA-based device in a seismic application, accurate material models that describe highly nonlinear behavior of SMAs are needed.

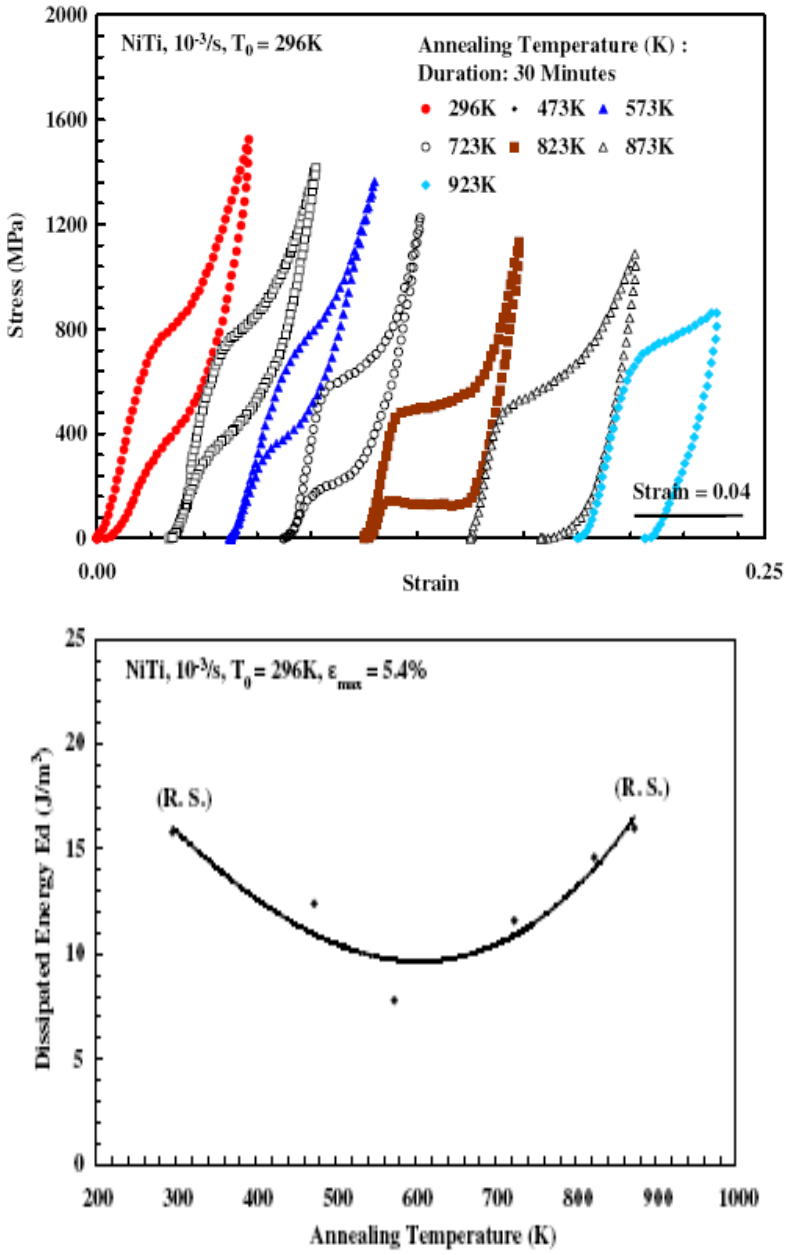



**Fig. 12.** Effect of annealing temperature on superelastic behavior: (a) strain-stress curves for different annealing temperatures; (b) change of dissipated energy with annealing temperature (Nemat-Nasser and Guo, 2006)

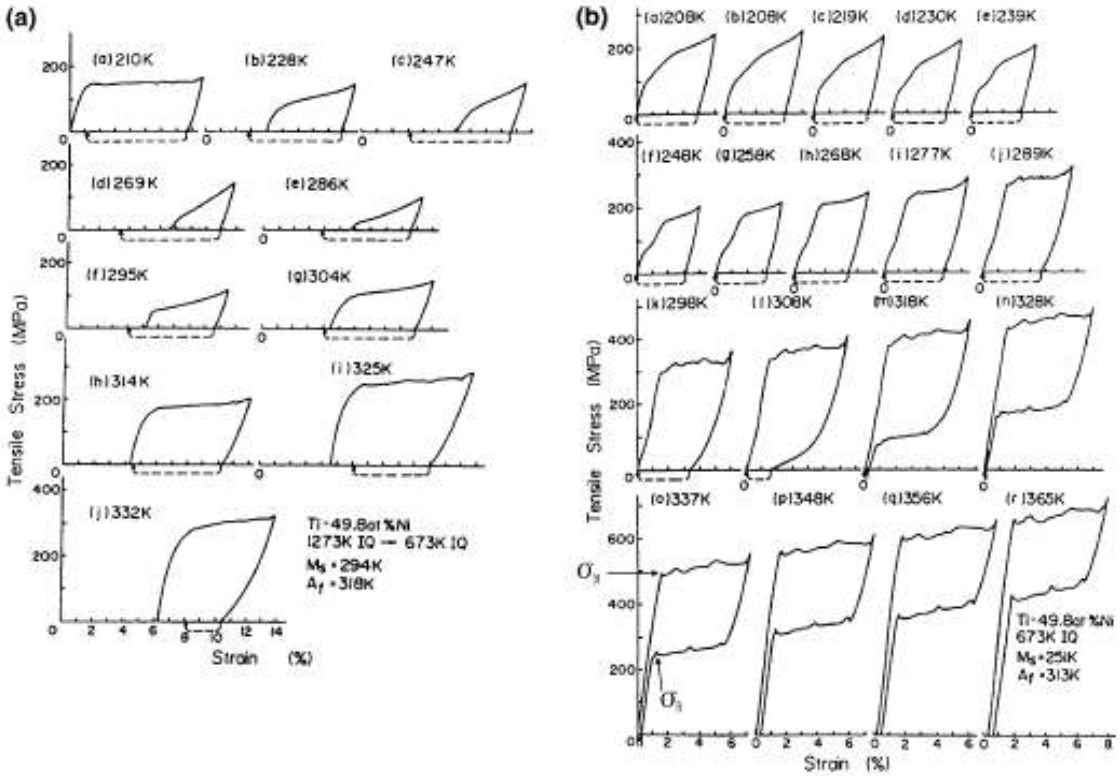



**Fig. 13**. Effect of heat treatment as a function of temperature: test results from specimen (a) without being cold-drawn; and (b) with being cold-drawn (Otsuka and Ren 2005)

The goal of a constitutive model is to represent the material performance as a function of one or more variables. Phase transformations are essential variables as are the unique properties of SMAs. These transformations are activated by thermal or mechanical changes. Therefore, stress, strain, temperature, and their associated rates are the main variables that characterize the behavior of SMAs (Prahlad and Chopra, 2001). SMA constitutive models are often based on thermomechanical and thermodynamical considerations as well as on experimental observations of material behavior. Tanaka's model (1986) is one of the first constitutive models developed for SMAs. Using the principles of thermodynamics and Helmholtz's free energy, Tanaka derived a constitutive equation to predict behavior of SMAs. Several researchers modified

Tanaka's constitutive equations to improve and simplify the model (Liang and Rogers, 1990; Tobushi *et al*., 1996; and Tamai and Kitagawa, 2002). Brinson (1993) modified the model of Liang and Rogers using a thermomechanical approach to predict performance of the material over a full range of temperatures. Boyd and Lagoudas (1998) developed a thermodynamical model that utilizes Gibbs free energy to derive constitutive relations.

In general, the complexity of the above constitutive models makes them impractical for seismic applications. Moreover, only quasi-static loading conditions are considered in most of these models (Prahlad and Chopra, 2001). To overcome these limitations, several phenomenological models that are relatively simple to implement in numerical analyses have been introduced. A macrostructural model that is proposed by Graesser and Cozzarelli (1991), as well as its extension that considers martensitic hardening at large strain amplitudes as proposed by Wilde *et al*. (2000), are among the models that have been proposed for seismic applications of SMAs. Another proposed model by Sun and Rajapakse (2003) that includes frequency dependent behavior of SMAs for earthquake engineering is a modified form of Brinson's model. Bilinear hysteresis models - with or without a trigger line - are also used mostly to simplify behavior of the material (Thomson *et al*., 1997; Saadat *et al.*, 2001; Masuda and Noori, 2002; Andrawes and DesRoches, 2005; Han *et al*., 2005; and Choi *et al*., 2005).

Even though phenomenological models satisfactorily predict essential features of SMAs, there are associated difficulties. For example, the presence of numerous material parameters, difficulties in establishing mathematically described parameters with experimental methods, complex effects of thermal changes, variable rates of loading and cyclical loading require more work to be done in order to both represent material performance more accurately and include the model in numerical procedures.

### 3.4. Seismic Applications of Shape Memory Alloys

Several studies have been performed to determine the potential applicability of SMAs in civil engineering. Some researchers have developed devices in which unique

properties of SMAs such as recentering and damping capacities can be exploited. After modeling of an SMA device has been developed, an experimental test is often conducted. One commonly used implementation involves installation of SMAs in the form of either base isolation devices or energy dissipating systems in a large structure. Also being proposed are bracing systems for buildings. In the transportation field a few studies have explored the possibility of using an SMA as a restrainer or as part of a base isolation system for control of seismic motion of bridges. In the following sections, each of these research efforts is briefly discussed.

### 3.4.1. SMA-based Devices

One of the most extensive studies on SMAs to date is the MANSIDE project, a European Union endeavor, in which material properties and device development have been thoroughly investigated. As part of this project, Dolce *et al*. (2000) developed an SMA-based passive control device that is intended to serve as a bracing system in buildings and for base isolation. Nitinol wires are placed to the device so that they are only strained in tension. The wires are divided into two groups that represent re-centering and energy dissipating capabilities of the SMAs as shown in Fig. 14. The desired performance of the device can be achieved by changing the number and characteristics of the two groups of wires. Experimental tests demonstrate the capability of the devices for both base isolation and bracing systems.

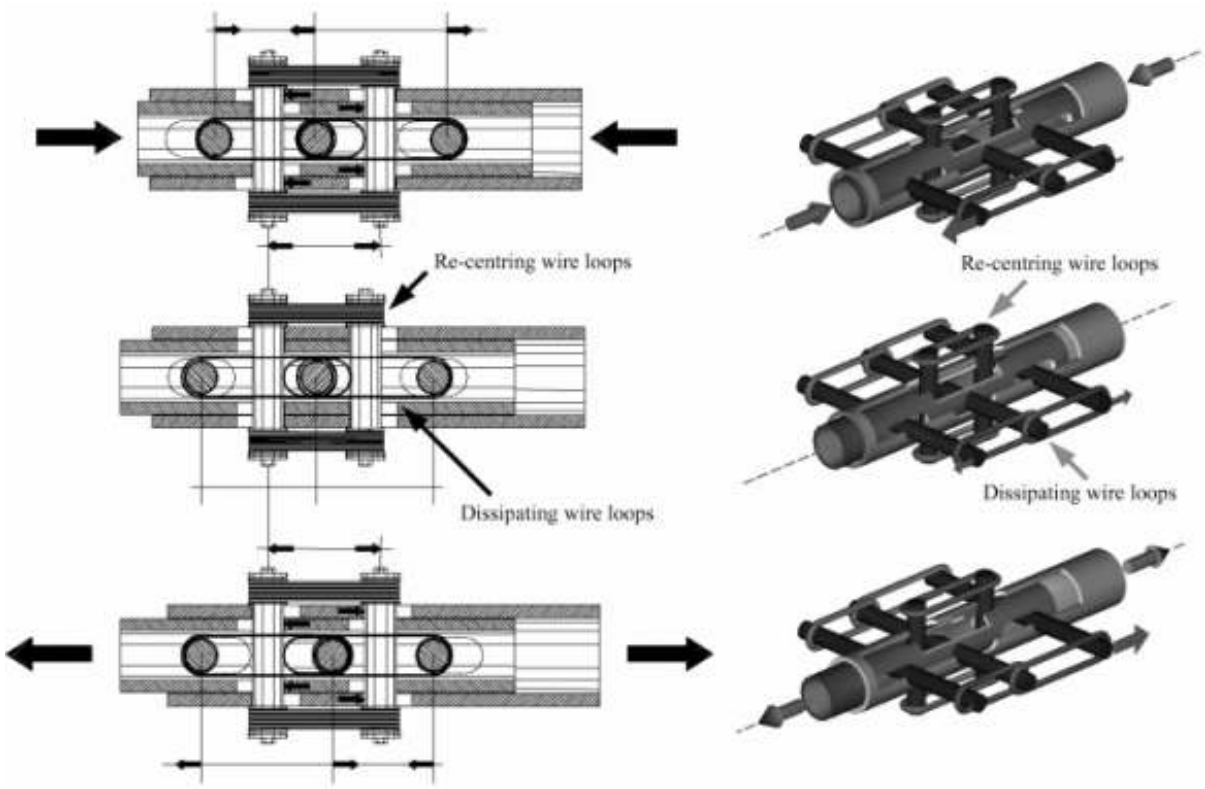



**Fig. 14.** Functioning of SMA-based control device developed by Dolce *et al*. (2000)

Han *et al.* (2005) developed an SMA damper that can simultaneously work in tension, compression and torsion (Fig. 15). NiTi wires that are subjected to tensile strains for all loading cases are used to create a damper. To verify effectiveness of the damper for tensile, compressive and torsional motion, analytical and experimental studies are carried out on three reduced-scale dampers.

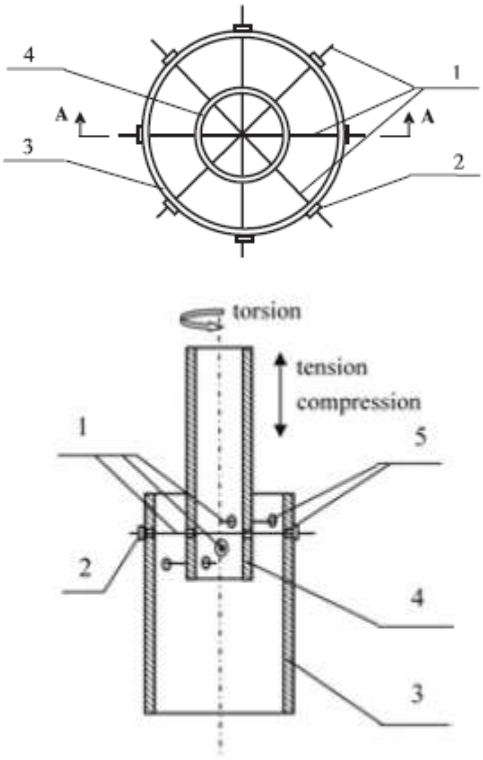
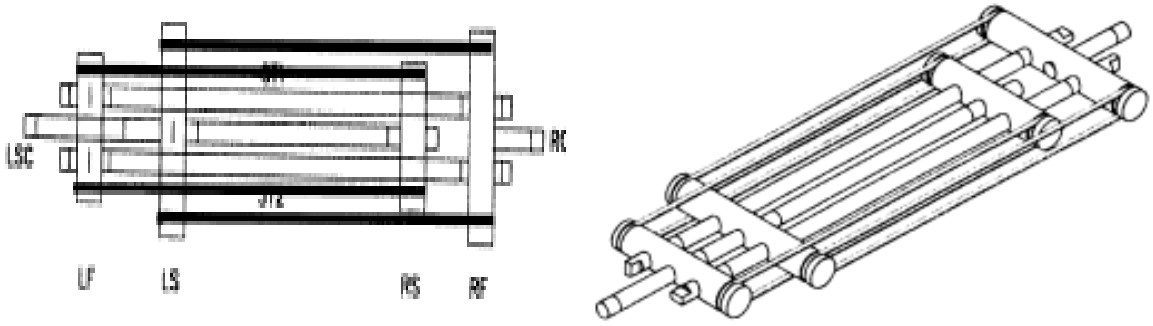



**Fig. 15**. SMA damper developed by Han *et al*. (2005)

In another study Krumme *et al.* (1995) developed a passive damping device that is composed of two opposing pairs of SMA wires (Fig. 16). The device can provide four different hysteretic shapes depending on the position and pre-stressing level of the wires. Experiments conducted on the device show that its performance is independent of the ambient temperature and loading frequency. Using DRAIN-2DX, a nonlinear analysis of a reinforced concrete frame retrofitted with SMA dampers is implemented to demonstrate its potential. The interstory drifts are effectively reduced to desirable values.



(a) (b)

**Fig. 16**. SMA center-tapped devices: (a) plan view; (b) device with SMA wires (Krumme *et al*., 1995)

Clark *et al.* (1995) designed and tested two different types of dampers using nitinol wires. The configuration of their devices consists of multiple loops of superelastic wire wrapped around cylindrical support posts. Experimental results of the first design type that utilizes a single layer of 100 loops show that the loading frequency and cyclical loading decrease the damping capacity of the device. On the other hand, a double-sided device that uses 70 loops of pre-tensioned wire in three layers exhibits relatively stable behavior under different loading frequencies and during numerous loading cycles. Strain-stress behavior of the double-sided device is almost insensitive to changes in temperature; by contrast, higher temperatures reduce the damped energy in the single-sided devices. A preliminary analysis of a six story steel frame equipped with a double-sided damping device demonstrates that a decrease of up to 50% in structural response is achievable. Aizawa *et al*. (1998) further investigated the performance of the SMA damper developed by Clark *et al.* (1995) under earthquake excitations. They performed a shake table test on the same frame that was studied before. Energy absorbed by the frame significantly decreased when the SMA device is installed in the structure.

### 3.4.2. Applications for Seismic Control of Buildings

Bruno and Valente (2002) compared conventional damping devices and a SMA-based device that was developed in the MANSIDE project for both isolation systems and energy dissipating braces through numerical simulations and fragility analyses. A six-story reinforced concrete frame was selected for investigation. Results show that regardless of the protection strategy, base isolation systems are more reliable than energy dissipating braces. The SMA-based device demonstrated superior performance in comparison with rubber isolators, while having similar structural responses in comparison with steel braces. Nevertheless, the recentering ability of the SMA device was underlined as an advantage over installation of steel braces. In another study, Dolce *et al*. (2005) tested a three-story, one bay reduced-scale reinforced concrete frame with installed SMA-based braces in a shake table. In order to compare results they performed the same shake table test for a plane frame, a frame with an infilled masonry panel, and a frame with a steel-based dissipating device. Both the SMA braces and the steel braces showed significant and similar performance in protecting the structure against damage.

Han *et al*. (2003) suggested use of an SMA damper that consists of two steel wires with a diameter of 7 mm and one SMA wire with a diameter of 0.75 mm. They placed eight of these dampers in a two-story steel frame (Fig. 17). An initial displacement was given to the frame until the restraint was quickly released. Vibration decay of the frame was recorded during free vibration motion. Results are correlated with a finite element model of the frame that models SMA wires as linear springs. Numerical simulations illustrate the efficiency of the SMA dampers in vibration control.

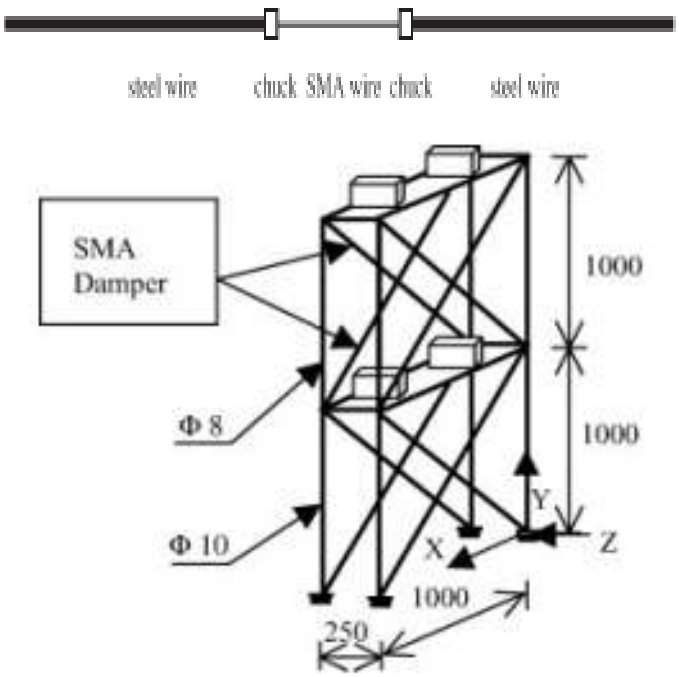



**Fig. 17**. SMA damper proposed by Han *et al*. (2003), and two-story experimental frame

Salichs *et al*. (2001) investigated the feasibility of using SMA wires as diagonal braces for vibration control of buildings. Experimental and numerical studies on a one-story building show that the SMA bracing wires effectively damp out undesirable vibrations of the frame. Seelecke *et al*. (2002) conducted an analytical study on a single-degree-of-freedom system that is equipped with SMA wires under seismic excitations. Their results show that hysteretic behavior of SMAs has a significant role in reduction of structural response. Also, they point out that an optimal diameter of the SMA wire might be defined for maximum vibration control. Bartera and Giacchetti (2004) studied the response of a single story reinforced concrete frame that had been upgraded by different types of steel dissipating braces (Fig. 18). A high damping rubber pad (HDRD) and an SMA device (SMAD) were utilized as supplemental energy dissipation devices in series with steel braces. The SMA device consisted of 20 groups of 10 nitinol wires that

have a diameter of 1 mm and a length of 427 mm. Free vibration and forced vibration tests were carried out to evaluate dynamic response of braced frames. Both dissipating bracing systems suppress vibration of the frame by adding a significant amount of damping.

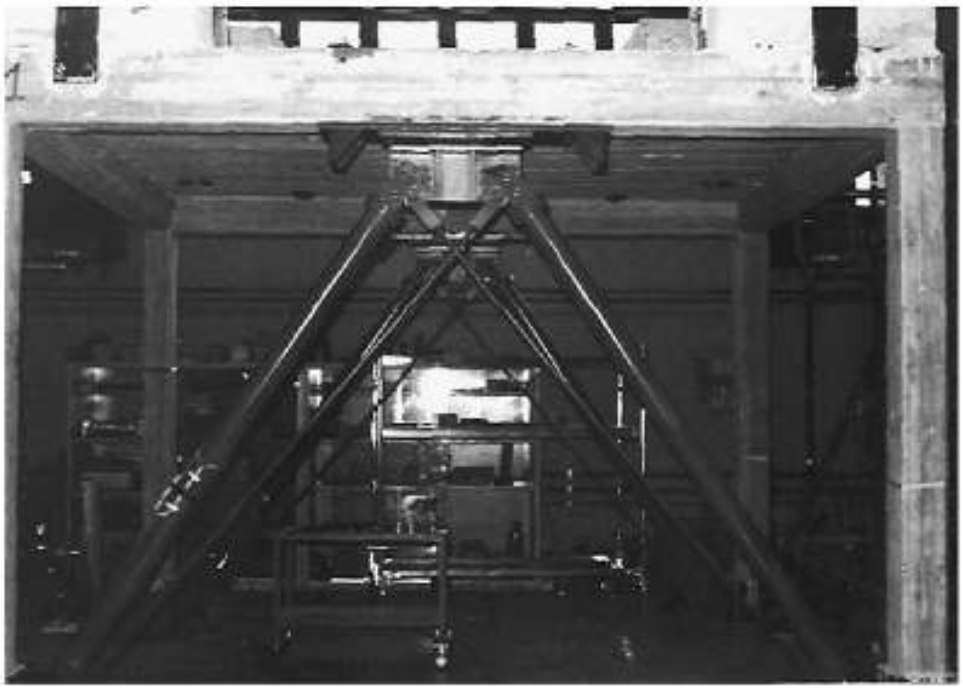


**Fig. 18**. Reinforced concrete frame with SMAD bracing system (Bartera and Giacchetti, 2004)

### 3.4.3. Applications for Seismic Control of Bridges

A few researchers have also explored the potential of using SMAs for seismic design of bridges. Wilde *et al*. (2000) proposed a base isolation system that is composed of a laminated rubber bearing and an SMA device. They placed a set of two SMA bars that are assumed to have the same response in tension and compression between a pier and its superstructure. Using the amount of absorbed energy as a metric, they compared the performance of the proposed isolation system with a laminated rubber bearing that

has a lead core isolation system. The SMA isolation system achieves good control against small, medium and large size of earthquakes, but the input energy to the system is greater than that of the rubber bearing system.

Andrawes and DesRoches (2005) performed a nonlinear dynamic analysis of a reinforced concrete bridge with multiple frames that utilizes SMA bars as restrainers. Superelastic behavior of the SMA bars is modeled by using two link elements and a truss element in Drain-2DX. Also, a conventional steel cable restrainer is modeled to evaluate the capabilities of the SMA restrainer. Results show a substantial decrease in relative hinge displacements in the bridge when superelastic SMA bars are used. Choi *et al*. (2005) developed a new isolation system for seismic protection of bridges using elastomeric bearings and SMA wires. Analytical studies on a multispan steel bridge illustrate that the combination of an SMA-rubber bearing effectively decreases relative displacement between deck and pier when compared with a conventional lead-rubber bearing. Nevertheless, load demands on the piers increased more for the SMA-rubber bearing system.

As mentioned earlier shape memory alloys are metals that exhibit unusual properties such as superelasticity and the shape memory effect. Superelastic SMAs that remember their original shapes and show hysteretic behavior have a significant potential as passive vibration control devices in seismic engineering. To this end there are significant research efforts that intend to facilitate use of SMAs in civil engineering structures. Nevertheless, there are still some difficulties that must be overcome before they can be employed in real applications with their full potential. The sections follow describe a neuro-fuzzy model that is developed to model dynamic behavior of a superelastic SMA. It also considers environmental effects which are especially important in seismic engineering. Collection of experimental data for fuzzy modeling is explained first in the next section.

# 4. EXPERIMENTAL TESTS AND DATA COLLECTION

## 4.1. Introduction

In order to create a fuzzy model of the dynamic behavior of a superelastic CuAlBe shape memory alloy, experimental data are required. In this study, all data used for modeling SMA wires are obtained from experiments performed at the University of Chile. First, a description is given about the material used for the tests and the heat treatment of the wire. Next, a brief discussion is made on a series of experimental tests that were conducted.

## 4.2. Material Characterization

CuAlBe shape memory alloy wires that have a composition of 87.7% Cu, 11.8% Al, and 0.5% Be are selected for this study. The diameter of the wire is 0.5 mm. Transformation temperatures reported by the manufacturer of the wire material are presented in Fig. 19. Since ambient temperatures for civil engineering structures are usually greater than the $A_f$ transition temperature of 2 °C, the material is expected to operate entirely within the superelastic range.

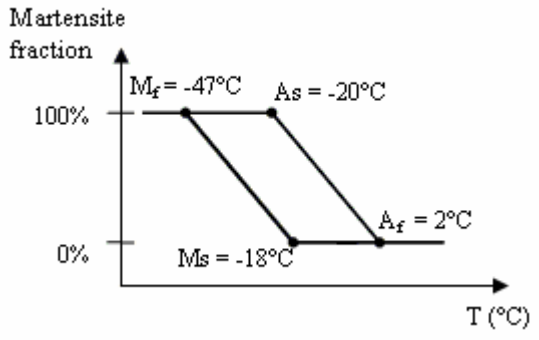



**Fig. 19**. Transformation temperatures

A thermo-mechanical treatment is applied to the wires in order to eliminate possible dislocations, and to have a uniform and fine grain size that provides acceptable mechanical properties. Fig. 20 shows the microstructure of the material before and after heat treatment. Three grain sizes that are obtained from different durations of heat treatment are 60, 100 and 200 μm. The process of heat treatment eliminates dislocations and results in a homogenous and larger grain size. However, to avoid intergranular brittle fracture (Funakubo 1987), a small grain size (60 μm) is selected for the experimental tests.

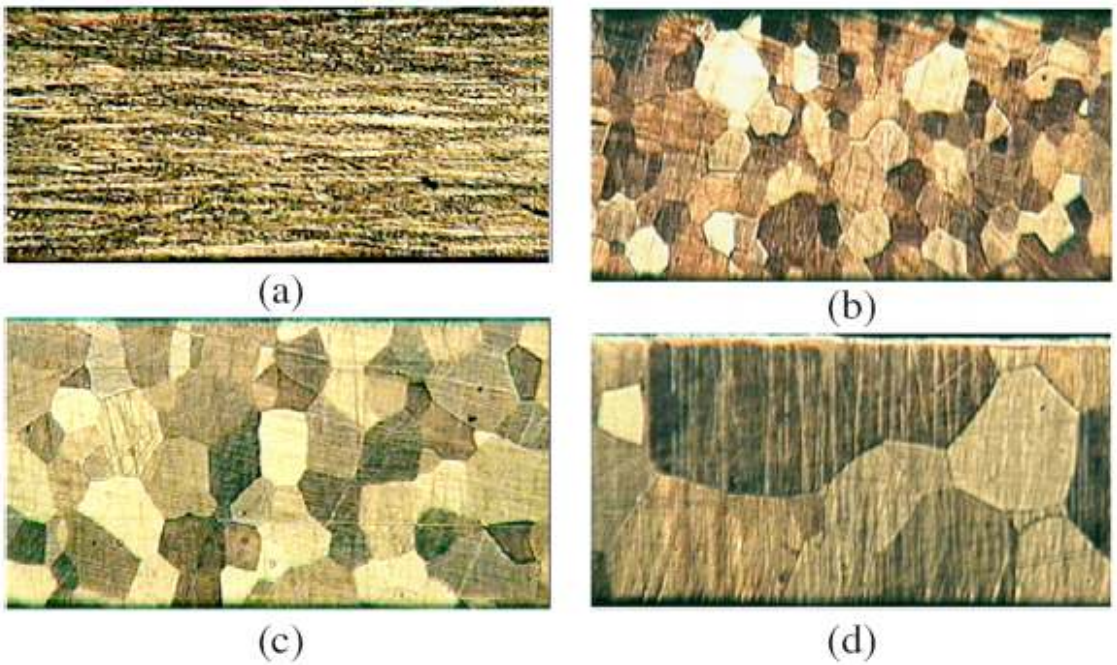


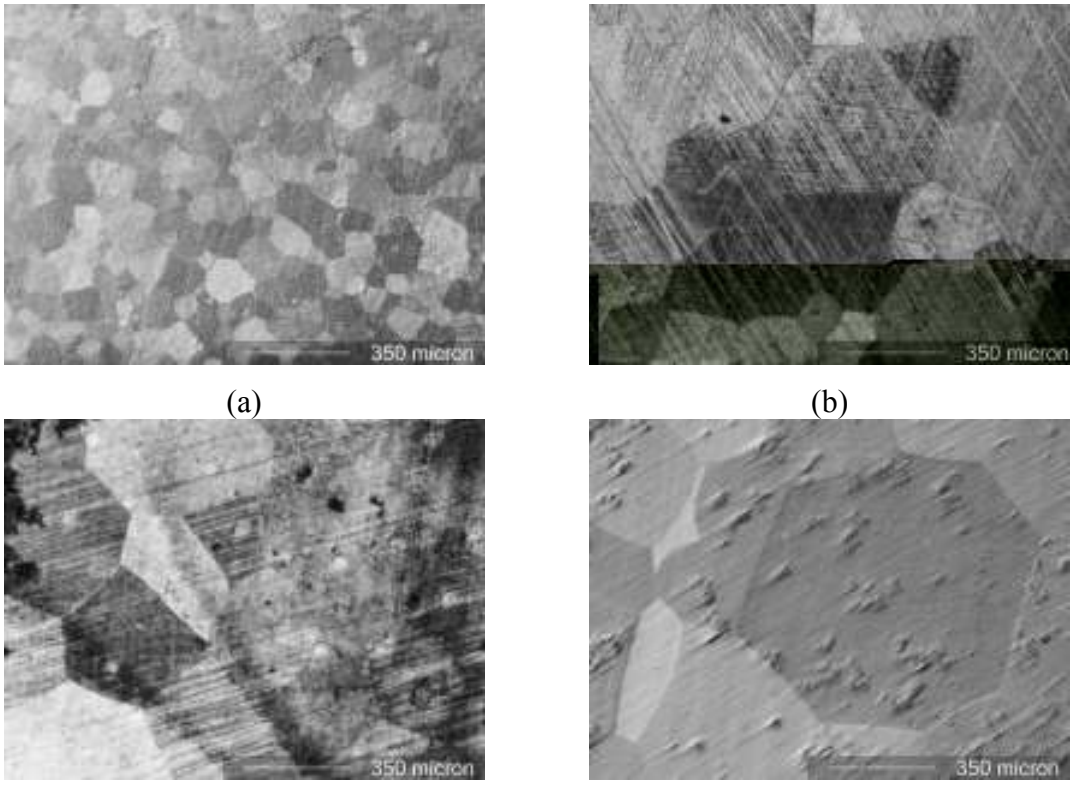

**Fig. 20**. Microstructure of material: (a) as received; and heat-treated for (b) 20 s; (c) 30 s; and (d) 180 s

To exemplify the heat treatment procedure, heat treatment of 2 mm diameter CuAlBe wires conducted in the Department of Mechanical Engineering at Texas A&M University is explained. Four small samples were taken from 2 mm diameter CuAlBe

wire supplied by a French producer of SMA materials called Nimesis. Three samples were kept in the oven at 850 °C for 90, 120 and 150 sec, respectively. After removal from the oven the samples were quenched in a water bath and then returned to the oven for additional annealing at 100 °C for 24 hours. In order to grip the samples during polishing, they were placed in small cylindrical forms, and the forms were filled with epoxy. Next, the samples were polished with 200, 600, 1000, 1300, and 2400 grit papers. Finally, aluminum powder was used to refine the material surface by means of polishing paper (Fig. 21).

**Fig. 21**. CuAlBe samples and polishing

The samples were etched with aqua 15% $HNO_3$ for 1~3 min before optical microscopy images were taken. As shown in Fig. 22, grain size increases with increasing duration of heat treatment. Note that grains are visible for the as-received sample in Fig. 22, whereas there is no visible grain in the virgin material in Fig. 20. This is the case because the material had already been heat-treated by the manufacturer when it was received for the latter case.




(a) (b)

(c) (d)

**Fig. 22.** Microstructure of 2 mm diameter of CuAlBe: (a) as received; and heat-treated for (b) 90 s; (c) 120 s; and (d) 150 s

### 4.3. Experimental Tests

In order to investigate applicability of CuAlBe wires for seismic applications, two experimental data sets that are obtained from different test programs at the University of Chile are used for modeling. The first data set comes from cyclic tensile and unloading tests that were conducted on CuAlBe wire at various temperatures. The second data set consist of the results of shake table tests on a scale model of a three story structure that is stiffened with SMA wires. Each experiment is explained in the subsections that follow.

### 4.3.1. Tensile Tests

Superelastic properties of shape memory alloys are not only strongly dependent on material composition and thermomechanical treatment, but also they are very sensitive to temperature changes. Fig. 23 shows hysteretic behavior of an SMA wire as a function of ambient temperature. A significant influence of temperature on material characteristics is clear. Although a linear relation between temperature and the threshold stress level that starts the phase transformation is observed for most cases, the explanation and modeling of thermal effects is complex and requires knowledge at the atomic level as underlined in a recent study by Kafka and Vokoun (2006).

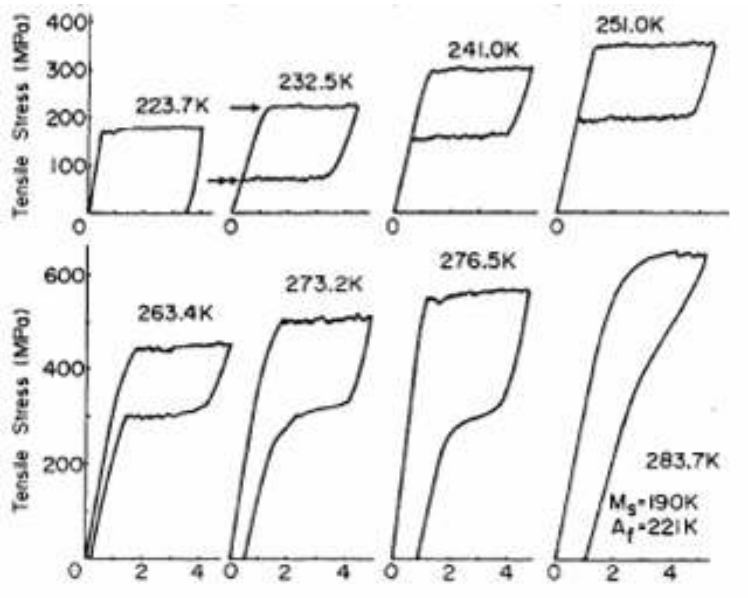



**Fig. 23**. Stress-strain curves of NiTi SMAs at various ambient temperatures (Otsuka and Ren 2005)

In order to collect data that are necessary to obtain an SMA fuzzy model that correctly predicts behavior of CuAlBe material at different ambient temperatures, a set of tests are carried out on CuAlBe wires with the testing system shown in Fig. 24. The

wire samples that have a length of 12 cm are tested at 0 °C, 25 °C, and 50 °C. Cyclical tensile tests are conducted by applying a sinusoidal force and a strain-controlled rate of loading. To simulate pseudo-dynamic loading, the frequency of loading is set to 1 Hz. Three maximum strain amplitudes are selected as 0.8 %, 1.5 %, and 2.2 %. An extensometer with a 25 mm gage length is used to measure axial strain. Each test consists of 20 cycles at the defined maximum strain. Results of numerical modeling of this series of tests are presented in the next section.

**Fig. 24**. Tensile testing machine

### 4.3.2. Shake Table Tests

In addition to the controlled temperature tests with sinusoidal loading as described above, a series of shake table tests are performed on a three story model of a frame building in order to collect data for modeling dynamic behavior of the material (Cerda *et al.* 2006). Height, length, and width dimensions of the frame used in the testing program are 120 cm, 70 cm and 42 cm, respectively; the steel columns (30 × 4 mm) are bolted to the beams that are welded to the floor slab (see Fig. 25). Motion of

the model is intended to be only in the longitudinal direction. Each floor weighs 180 N for a total weight of 540 N.

Four SMA braces (dampers) are installed at each level. Each brace consists of a steel channel shape (15 × 15 × 1.5 mm) that is 45 cm long as well as a CuAlBe wire with a diameter of 0.5 mm and length of 40 cm. At one end of the steel channel a longitudinal bolt (9.19 mm in diameter) serves to tension the CuAlBe wire by means of a nut. The wire is fixed to the steel channel and to the floor slab through a transverse bolt (1.53 mm in diameter). Also, the CuAlBe wires are pre-stressed with a tensile force of approximately 30 N so that they remain in tension during the dynamic motion of the frame.

**Fig. 25**. Three story structure with SMA wires and sensors

Twelve transducers are installed on each floor to measure longitudinal and transverse accelerations; also, eight load-cells and eight potentiometers measure forces

and axial deformations in the dampers that are located below the first and second floors. Shake table tests are performed using 5 min of white noise motion and amplitude-scaled records from Sylmar 1994 (0.36 g Peak Ground Acceleration - PGA), Kobe 1995 (0.23 g PGA), Taft 1952 (0.36 g PGA), El Centro 1940 (0.21 g PGA) and Llolleo 1985 (0.24 g PGA) earthquakes.

Data that were obtained from the tensile and shake table tests performed at the University of Chile are used to generate fuzzy models of the CuAlBe SMAs as discussed in the next section.

# 5. NEURO-FUZZY MODEL OF SUPERELASTIC SHAPE MEMORY ALLOYS

## 5.1. Introduction

Analytical and numerical modeling of the complex stress-strain behavior of SMAs is another topic receiving widespread attention by researchers in structural engineering. While there are a large number of studies that propose analytical models that range from relatively simple to very complex, only a subset of these models is considered to be suitable and effective for application in seismic analysis of engineered structures.

The inherent complexity of superelastic behavior of SMAs makes modeling behavior of the material challenging. Therefore, available analytical models of SMAs involve a large number of material parameters as described in Section 3 that can cause difficulties while implementing computer simulations. Furthermore, as pointed out by other researchers (e.g. Wolons *et al.* 1998, Prahlad and Chopra 2001), most of the current constitutive models do not deal with dynamic response of SMAs. Clearly, in seismic applications it is important to be able to represent behavior of SMAs under cyclic loading.

In this section, a soft computing approach, namely a neuro-fuzzy technique, is proposed to model the superelastic behavior of a CuAlBe shape memory alloy. Two fuzzy inference systems (FISes) have been developed. First, a FIS that predicts SMA behavior at various temperatures is generated. Then, a FIS that considers dynamic effects on material response is developed. Three major steps in developing a fuzzy inference system, namely, data selection, training and validation, are clearly explained for both fuzzy models.

## 5.2. Fuzzy Modeling of SMA Wires

A Sugeno-type fuzzy inference system (Sugeno, 1985) is employed to model superelastic shape memory alloys. A FIS maps a set of input data through membership

functions and rules to a single-valued output. Here, the input and output variables selected for use with the two FISes developed in this study are shown in Fig. 26.

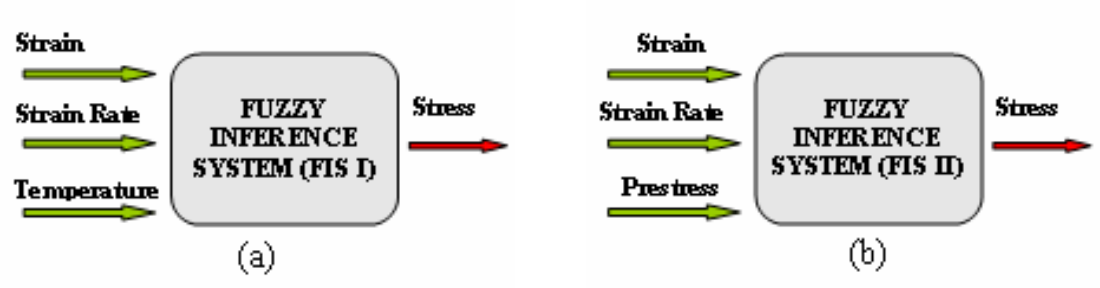



**Fig. 26**. Fuzzy inputs and output for (a) FIS I; and (b) FIS II

Inputs of the first inference system, named FIS I, are strain, strain rate and temperature, whereas the second inference system, named FIS II, employs strain, strain rate and prestress. Both of the FISes aim to predict stress in the CuAlBe wires. Adaptive Neuro-Fuzzy Inference System (ANFIS), a part of Matlab's Fuzzy Logic Toolbox (2007), is used to adjust parameters of the membership functions of both FISes by using a learning technique that combines a back-propagation algorithm with a least squares method.

A typical rule in a Sugeno fuzzy model has the form:

$$\text{Rule } i\text{: IF } X_1 \text{ is } \lambda_1 \text{ and } X_2 \text{ is } \lambda_2 \ldots \text{and } X_n \text{ is } \lambda_n$$

$$\text{THEN } Y = \mu_0 + \mu_1 X_1 + \mu_2 X_2 + \ldots + \mu_n X_n$$

where $X_1$, $X_2$, …, $X_n$ are antecedent variables; Y is the consequent variable; $\lambda_1$, $\lambda_2$, …, and $\lambda_n$ are fuzzy sets defined over the domains of the respective antecedents; and $\mu_0$, $\mu_1$, …, $\mu_n$ are constant coefficients that characterize the linear relationship of the $i^{th}$ rule in the rule set, ($i = 1, 2, \ldots, r$).

An outline of the three main steps used in this study to develop a fuzzy model of an SMA is as follows:

1. Set up data sets for training, checking and validation from experimental tests.
2. By employing ANFIS, generate a FIS that predicts stress in a wire at specified increments of time.
3. Validate the new model by comparing its predicted results to experimental results.

In the subsections that follow, each of these steps is briefly described for both fuzzy inference systems.

## 5.3. Fuzzy Model I

### 5.3.1. Data Selection

Training, checking and validation data sets for FIS I consist of data from cyclical tension tests that are described in the previous subsection. Two test sequences with a maximum strain of 0.8 %, 1.5 %, and 2.2 % at three different temperatures (0°C, 25°C, and 50°C ) comprise a total of 18 test records that are used by ANFIS to create a fuzzy model of the SMA wires. Each record consists of 11,000 data points. Sixteen of the records that contain a total of 176,000 data points are concatenated to obtain training and checking data sets (see Fig. 27). Odd numbered data values are used for training while even numbered data points are used for checking. A fourth-order central difference method is used to calculate strain rate data from measured strain data. A second-order Butterworth low-pass filter is applied to the strain data before taking a numerical derivative in order to eliminate high frequency content. Results from the remaining two experiments comprise the validation data set which has a total of 22,000 data points.

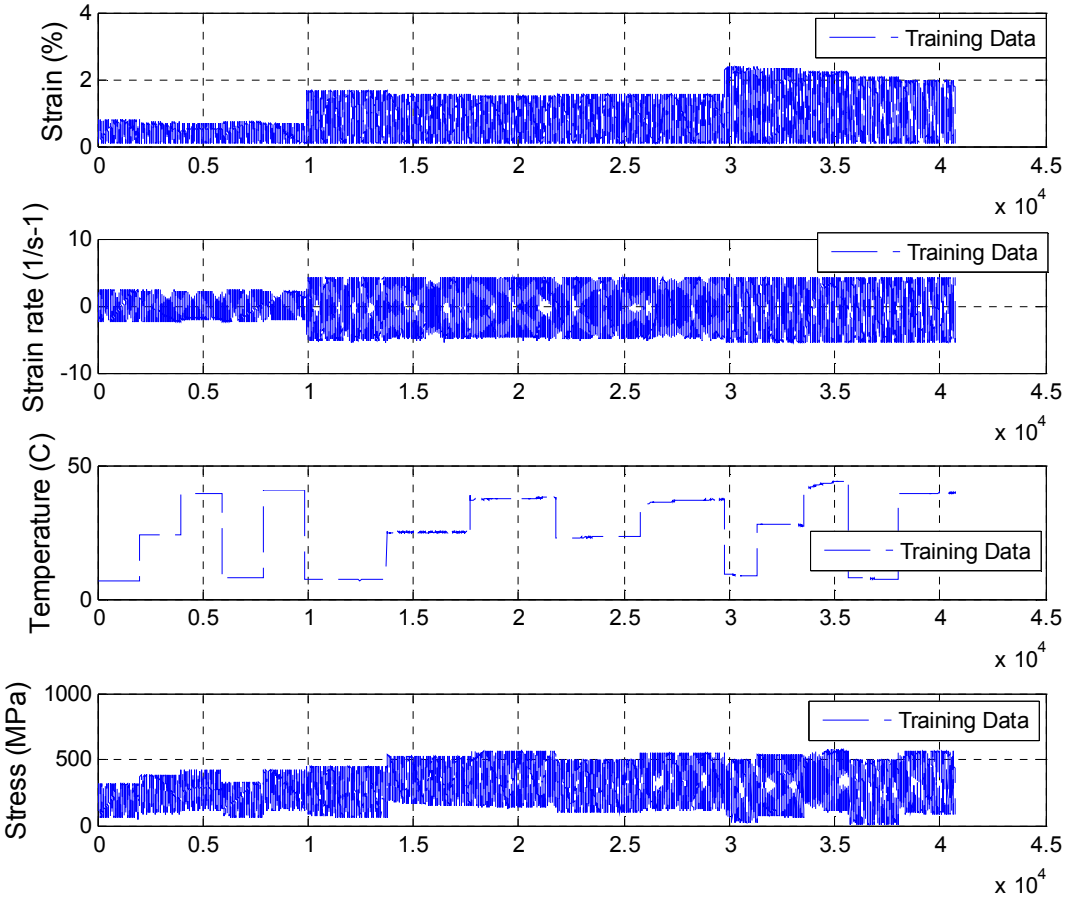



**Fig. 27**. Training data for FIS II: (a) experimental strain; (b) experimental strain rate; (c) experimental temperature; and (d) experimental stress

### 5.3.2. Training of Neuro-Fuzzy Model

An initial FIS is required before training and checking can proceed with ANFIS. Since, a priori, a FIS has no information about target behavior before training with ANFIS, the inference system is initially assigned membership functions that have random parameters for its variables. As mentioned above, the inference system has strain, strain rate, and temperature as inputs and stress as the single output.

After an iterative trial and error process to determine the type and number of membership functions, three Gaussian membership functions are selected for strain, and two membership functions are used for strain rate and for temperature in order to satisfactorily model behavior of the SMA wires. A total of 12 if-then rules are employed to map input membership functions to output characteristics. Also a total of 200 epochs serve to adjust parameters of the membership functions in ANFIS. Fig. 28 shows the membership functions before and after training, and Fig. 29 gives two of the surfaces of that relate the predicted stress to strain, strain rate and temperature for FIS I.

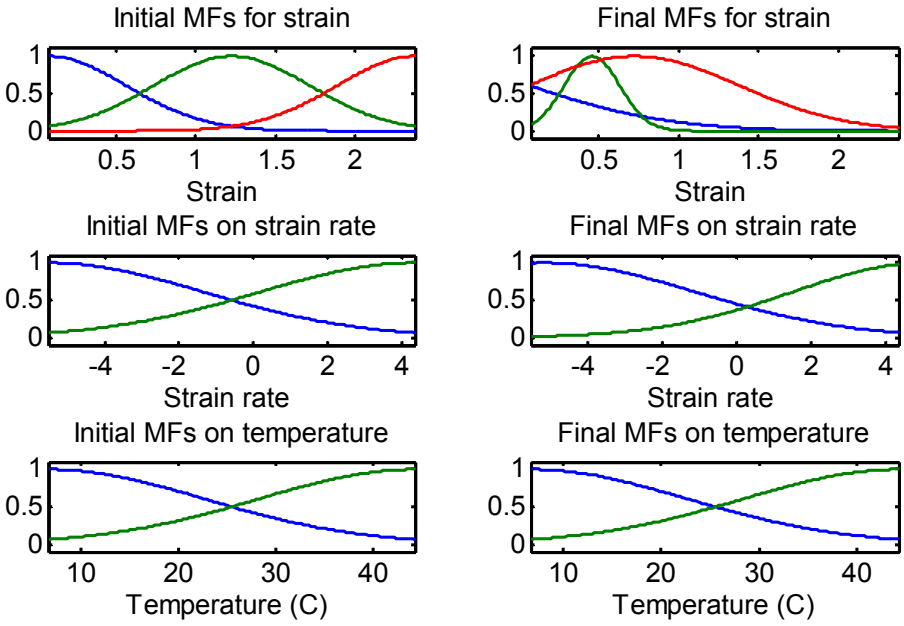



**Fig. 28**. Initial and final input membership functions for FIS I

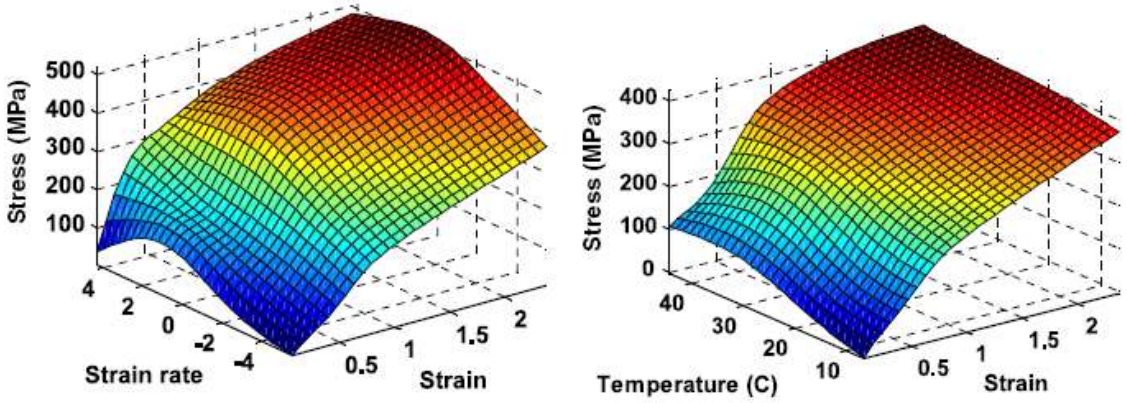



**Fig. 29.** Surface of stress from FIS I versus (a) strain and strain rate, (b) temperature and strain

In order to assess the ability of the neuro-fuzzy model to capture superelastic behavior of the SMA wires due to sinusoidal excitations at different temperatures, Fig. 30 shows experimental stress along with the fuzzy prediction of stress. For each case of temperature and strain amplitude, stress prediction by the developed model is deemed to be satisfactorily close to the experimental data.

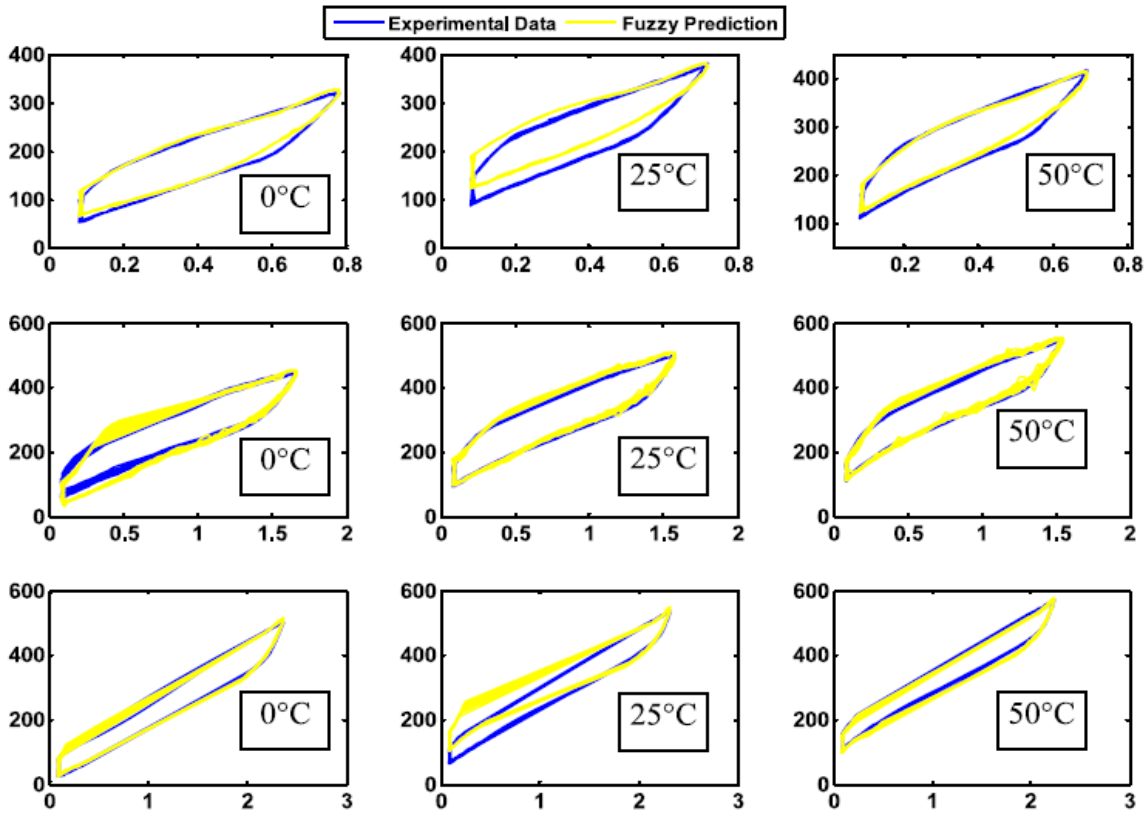



**Fig. 30**. Stress-strain diagrams of SMA wires at different temperatures for experimental data and fuzzy prediction of stress

Table 2 shows the maximum tangent modulus of elasticity and the effective stiffness for the first series of tests, and Fig. 31 shows phase transformation stresses for 1.5 % strain amplitude as an aid in evaluation of the effect of temperature on superelastic behavior. Clearly an increase in modulus of elasticity and stiffness is evident with increasing temperature. Phase transformation stresses (see Fig. 31) also increase with temperature. Fig. 32 shows the relationship of stress and temperature for CuAlBe material. Note that the slope of the lines gives an average value of 1.6 ºC/MPa which is considerably less than for NiTi shape memory alloys which have values of 7-8 ºC/MPa (Dolce and Cardone 2001, Torra *et al.* 2004). That is, the CuAlBe is found to be less dependent to the temperature than nitinol.

**Table 2**. Modulus of elasticity and stiffness as a function of temperature and strain amplitude

| T ºC | Maximum strain 0.8% | | Maximum strain 1.5% | | Maximum strain 2.2% | |
|---|---|---|---|---|---|---|
| | E (GPa) | k (GPa) | E (GPa) | k (GPa) | E (GPa) | k (GPa) |
| 0 | 90.3 | 36.0 | 72.1 | 23.2 | 45.5 | 19.6 |
| 25 | 113.6 | 44.2 | 89.9 | 24.3 | 68.4 | 20.0 |
| 50 | 145.9 | 51.0 | 110.0 | 29.8 | 84.3 | 21.0 |

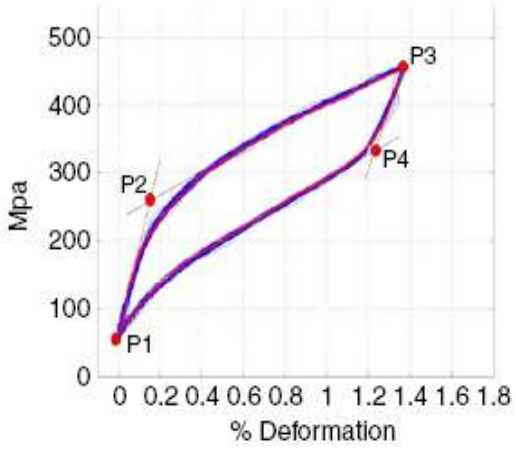



**Fig. 31**. Phase transformation stresses

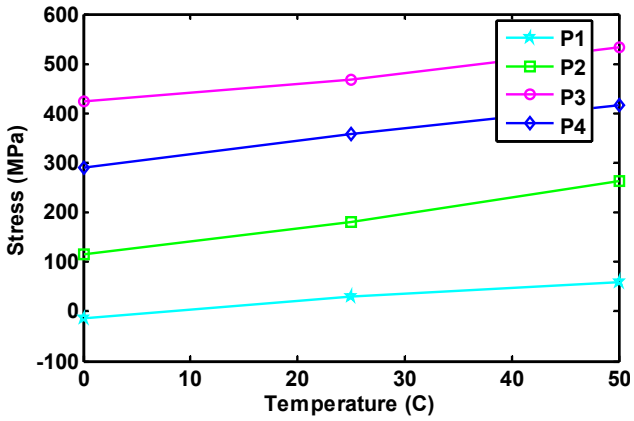



**Fig. 32**. Start and finish transformation stresses for 1.5% strain amplitude

Equivalent viscous damping for each experiment is calculated as follows (Chopra, 2001):

$$\xi_{eq} = \frac{E_D}{4\pi E_{so}} \qquad (1)$$

where $E_D$ stands for energy loss per cycle and $E_{so}$ is the maximum strain energy per cycle. Results for various strain amplitudes and temperatures are presented in Table 3. For each test series it is clear that a decrease in equivalent viscous damping occurs as the temperature increases.

**Table 3**. Damping capacity of SMAs at different temperatures and strain amplitudes

| Series | Temperature ºC | Equivalent viscous damping ratios | | |
|---|---|---|---|---|
| | | Maximum strain 0.8% | Maximum strain 1.5% | Maximum strain 2.2% |
| 1 | 0 | 3.2 | 3.4 | 2.9 |
| | 25 | 2.9 | 2.8 | 2.1 |
| | 50 | 2.6 | 2.6 | 2.0 |
| 2 | 0 | 3.0 | 3.3 | 3.0 |
| | 25 | 2.9 | 3.1 | 2.9 |
| | 50 | 2.0 | 2.5 | 2.2 |

### 5.3.3. Model Validation

After training, it is important to validate the resultant FIS by employing a data set that has not been used during training. Here, test results at 0 ºC with 1.5 % strain amplitude, and with strain amplitudes of 2.2 % at 25 ºC are reserved for validation (see Fig. 33). Stress-strain diagrams are plotted for both cases in Fig. 34. An adequate to good stress prediction ability of the fuzzy model is apparent from the hysteresis figures.

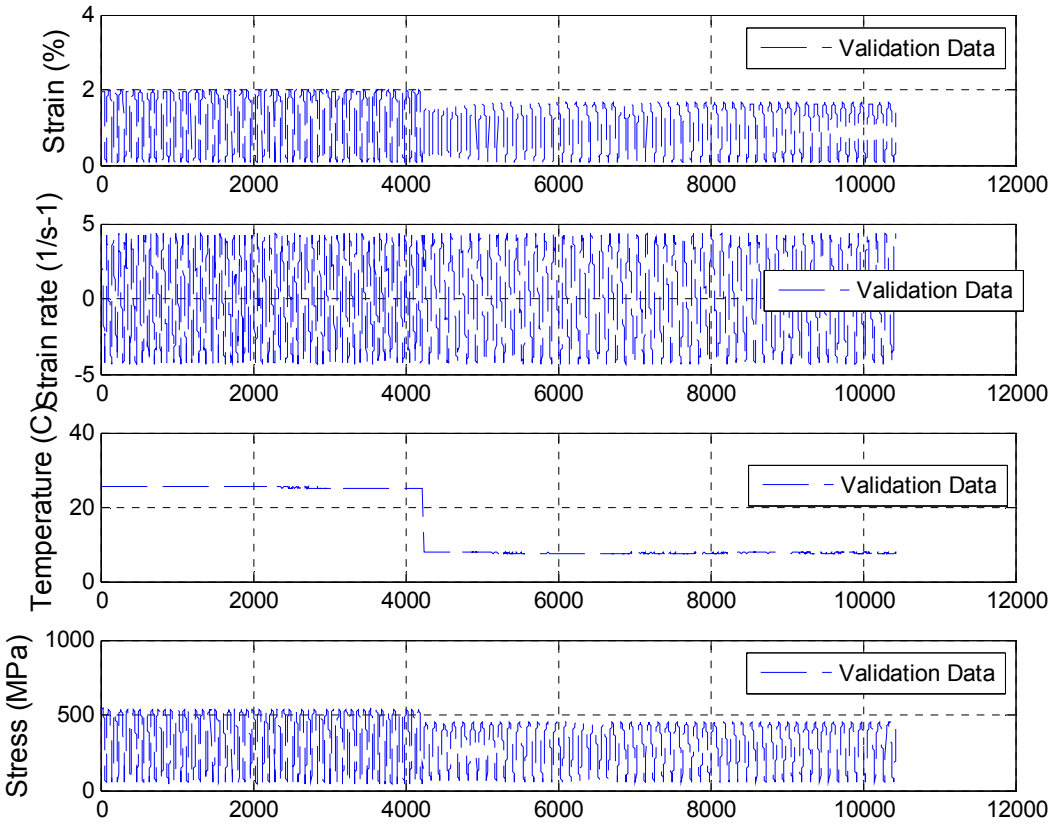



**Fig. 33.** Validating data: (a) strain; (b) strain rate; (c) temperature; and (d) stress

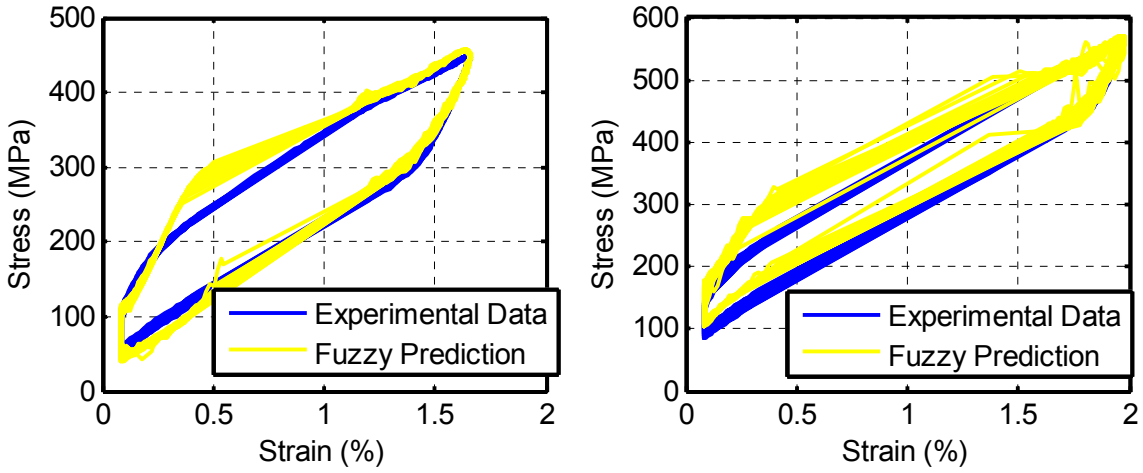



**Fig. 34**. Validation of FIS I

## 5.4. Fuzzy Model II

### 5.4.1. Data Selection

Sets of dynamic data for FIS II are obtained from the shake table tests described earlier. Experimental results from the eight wires that are mounted in the first and second floors are used. Training and checking data sets are obtained from four different earthquake excitations (Sylmar, Kobe, Taft, and Llolleo) that are imposed on the frame by the shake table. Prestressing forces in the wires are recorded before each test. Prestress levels of the wires for each experiment vary between 101.9 MPa and 172.9 MPa.

Table 4 shows the ranges of the strain and strain rate data from the experiments on the shake table. Strain rate data are obtained from a numerical derivative of the strain data. Here, strain and strain rate are 'relative' values; that is, the initial pre-strain is subtracted from the measured strain for each wire. Experimental data points that are outside of the specified range of the FIS are eliminated, and the remaining data points from a number of earthquake tests are concatenated sequentially. Training and checking data sets comprise a file that contains a total of 144,000 data points. Again, odd numbered data values are used for training while even numbered data sets are used for checking. Fig. 35 shows the concatenated data set used for training and checking of the FIS. Shake table test results from an excitation using the El Centro record comprise the validation data set that contains 152,000 data points.

**Table 4**. Ranges of strain and strain rate

| Earthquake | Maximum strain (%) | Minimum strain (%) | Maximum strain rate ($sec^{-1}$) | Minimum strain rate ($sec^{-1}$) |
|---|---|---|---|---|
| Sylmar | 2.34 | -1,90 | 43.25 | -51.28 |
| Kobe | 1.21 | -1.19 | 40.70 | -39.75 |
| Taft | 0.49 | -0.43 | 14.88 | -13.65 |
| Llolleo 1 | 1.11 | -0.92 | 41.52 | -39.94 |
| Llolleo 2 | 1.23 | -1.01 | 39.45 | -42.15 |
| El Centro | 1.10 | -0.93 | 33.48 | -30.66 |

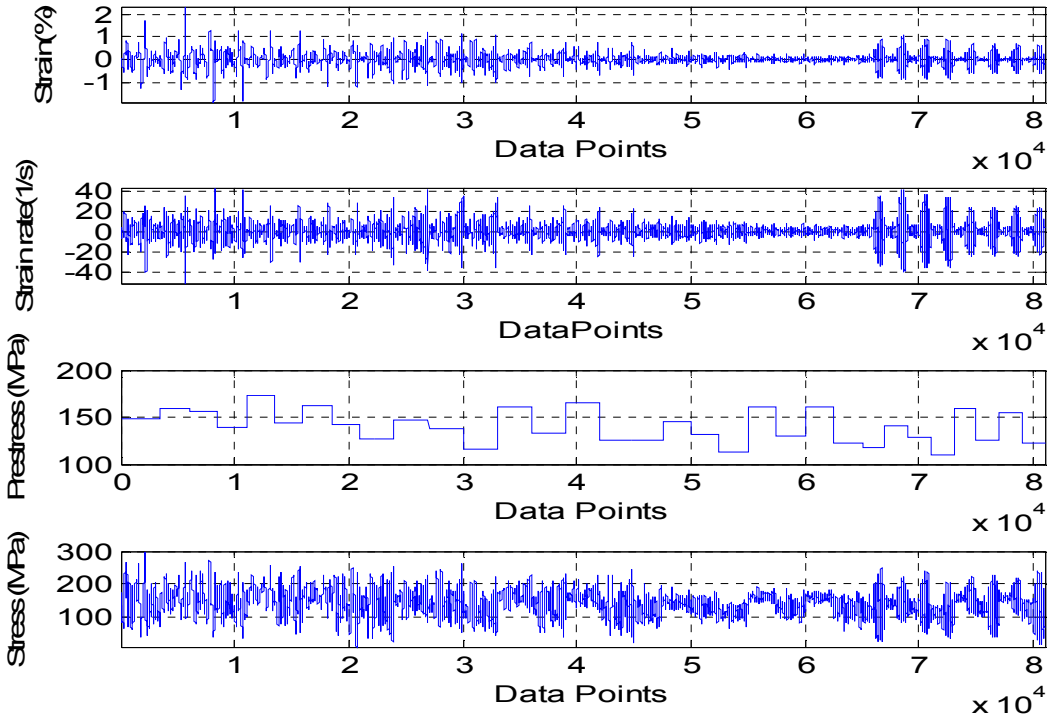



**Fig. 35**. Training data for FIS II: (a) experimental strain; (b) experimental strain rate; (c) experimental prestress; and (d) experimental stress

### 5.4.2. Training of Neuro-Fuzzy Model

After several trials, the optimum number of membership functions for each input variable is determined to be two. Gaussian membership functions are employed for the inputs and a total of 8 if-then rules are used. The initial step size, step size increment, and step size decrement are chosen to be 0.13, 1.20 and 0.80, respectively. Membership functions are tuned with a total of 200 epochs in ANFIS.

Initial and final membership functions for the input variables are shown in Fig. 36. Initial random parameters of Gaussian membership functions that are assigned to the strain input are adjusted considerably by ANFIS, whereas parameters for strain rate and prestress experience little change.

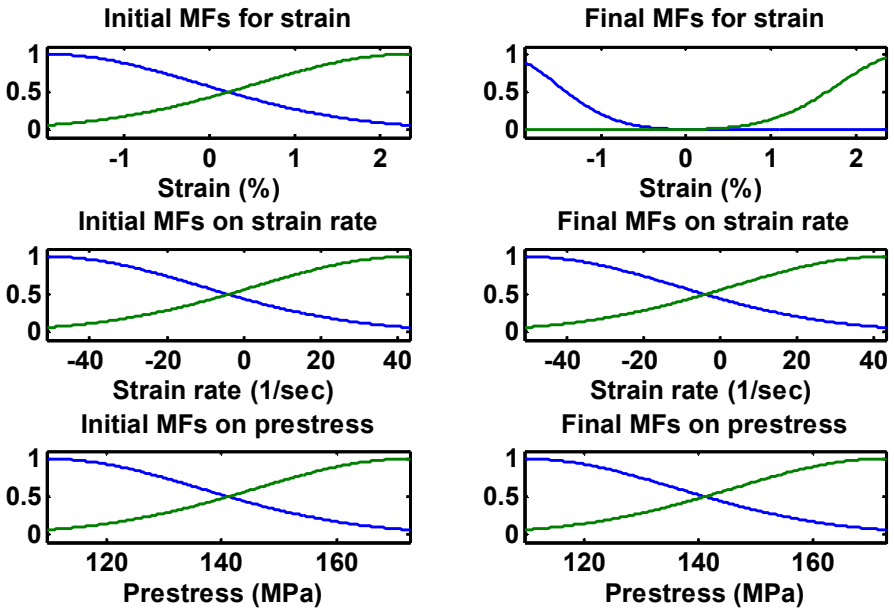



**Fig. 36**. Initial and final input membership functions for FIS II

After training, the fuzzy inference system predicts normal stress with a maximum and root mean square error of 75.1 and 11.7 MPa, respectively. Total computation time for training a neuro-fuzzy model of the SMA with a Pentium 4 (3.05 GHz) machine is approximately 17 min.

The two subplots of Fig. 37 each show two of the three input variables as well as the fuzzy prediction of stress in the CuAlBe SMA wire. These surfaces show that stress changes nonlinearly with a variation in either strain or strain rate; however, variation of stress with strain rate is even more nonlinear.

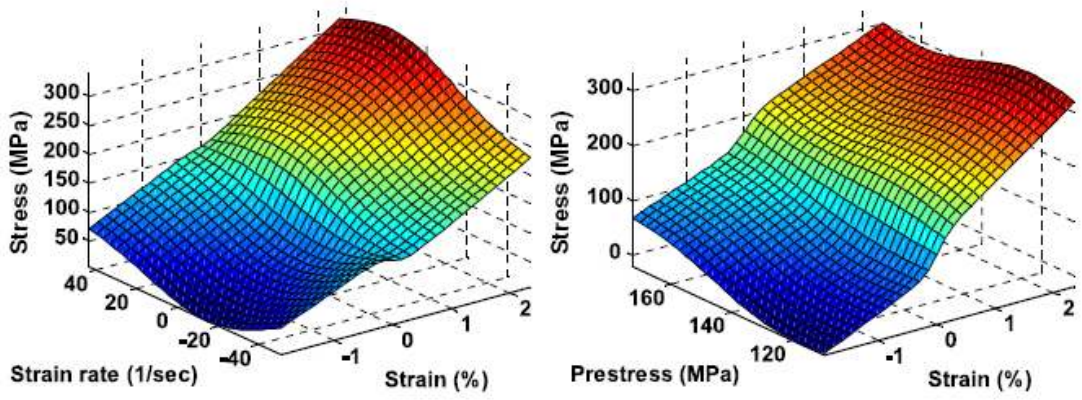



**Fig. 37**. Surface of stress from FIS II versus (a) strain and strain rate; (b) prestress and strain

### 5.4.3. Validating Data

Test results from the El Centro earthquake record are reserved for validation (see Fig. 38. Fuzzy prediction of stress and the corresponding experimental stress are presented in Fig. 39. As shown from the figure, the developed fuzzy model predicts stress in the wire very well. The magnitude of the error is acceptable for most engineering applications.

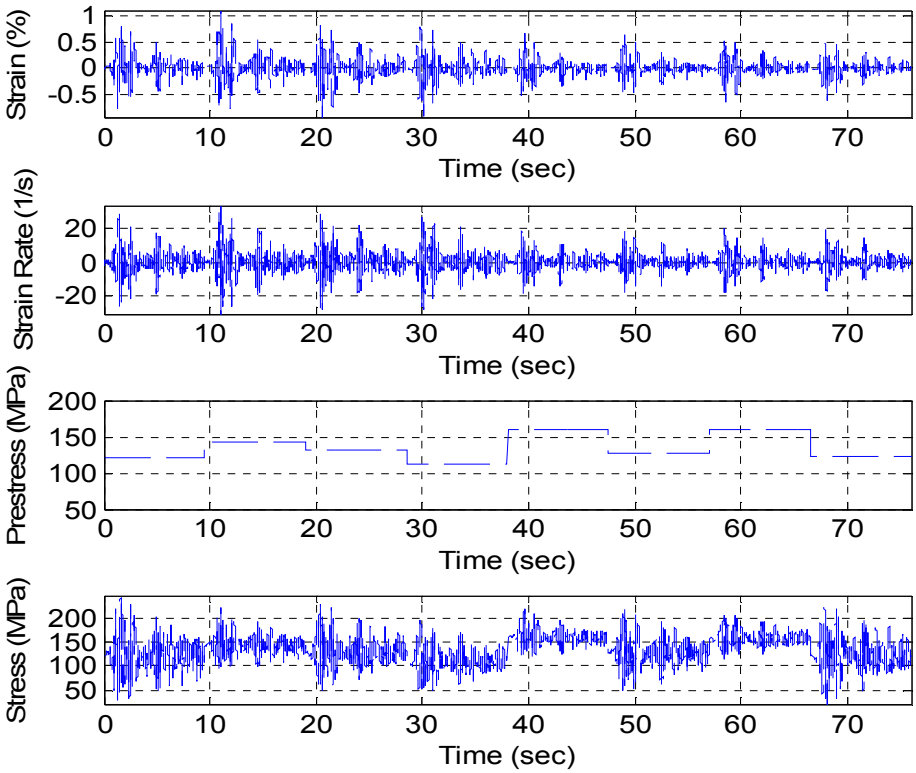



**Fig. 38.** Validating data: (a) strain; (b) strain rate; (c) prestress; and (d) stress

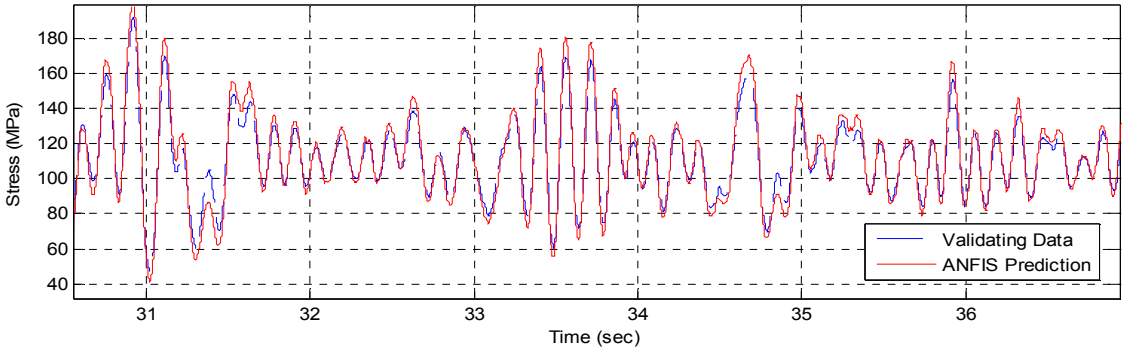



**Fig. 39**. Validation of fuzzy model: experimental stress and fuzzy prediction

### 5.5. Conclusion

Because of modeling difficulties arising from dependence of SMA behavior on many parameters such as strain amplitude, loading frequency, number of loading cycles, and temperature, a fuzzy modeling approach has been chosen in this study to describe the nonlinear behavior of SMAs rather than an analytical model. In particular, a Sugeno-type fuzzy inference system (FIS) that can represent material performance at different ambient temperatures and another FIS that can represent dynamic behavior of SMAs are developed and applied. Validation of both FISes shows that neuro-fuzzy models reproduce actual behavior of superelastic CuAlBe shape memory alloys very closely.

In the next section, applicability of developed fuzzy models to numerical simulations is discussed, and linear and nonlinear simulation results are presented.

# 6. NUMERICAL SIMULATIONS

## 6.1. Introduction

In this section, in order to investigate the usefulness of CuAlBe SMA material in seismic applications and to demonstrate the ability of computer implementation of the FIS model developed in the preceding section, numerical simulations of single and multi degree of freedom systems equipped with SMA bracing elements are carried out for several loading conditions.

First, linear analyses on a single degree of freedom (SDOF) system are conducted. Free vibration, sinusoidal excitation and earthquake excitation cases are simulated using Matlab (2007). Simulations on a bare (uncontrolled) frame, an SMA-braced frame, and a steel braced frame are implemented for comparison purposes. Then, numerical simulations with this methodology are extended to a three-story benchmark building. Maximum displacement, acceleration and interstory drift are analyzed by means of a parametric study. As another example, an isolated bridge is modeled as a two degree of freedom system in order to compare two different isolation strategies: a high damping rubber bearing and a natural rubber bearing with added SMA damping elements. Finally, nonlinear simulations of a SDOF system are introduced to investigate the recentering ability of SMAs.

## 6.2. Dynamic Analysis of a Single Degree of Freedom System

A single story frame is selected as a full-scale example for numerical simulations (Fig. 40). The story height and width are each 3 m. The weight of each story is 6,000 kg and the mass of each column is neglected in comparison with that of the story above it. A uniform rigid slab is assumed at the first floor level. The columns are made of steel, and have a section of I150×150×7×10. For steel, the modulus of elasticity is 210 GPa, and the yield stress is 250 MPa. Damping of the bare frame structure is taken to be 5% of its critical damping value. The second moment of cross sectional area of each

column is 2,919.2 $cm^3$. The stiffness of one column and the frame are calculated as follows:

$$k_c = \frac{12EI}{h^3} = 2724.6 \text{ kN/m} \quad (2)$$

and

$$k = 4k_c = 10898 \text{ kN/m} \quad (3)$$

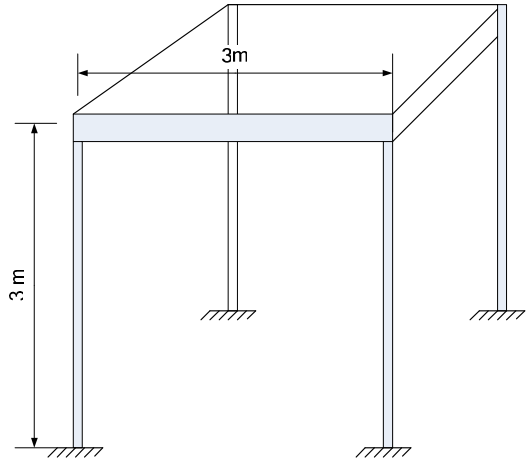



**Fig. 40**. Single story frame

After system parameters are determined as outlined above, numerical simulations of a bare (uncontrolled) frame, SMA-braced frame, and steel braced frame are performed. The SMA-braced and steel-braced frames are shown in Fig. 41. Each SMA bracing element consists of a bundled group of wires that are attached as a diagonal member. The number and length of the wires vary and are given separately for each simulation case. Rigid segments are assumed to be installed between the corners of the frame and the SMA wires; that is, the SMA wires are placed in the middle of each diagonal. Note that the SMA wires are prestressed to an initial value of 140 MPa which is the average value of the shake table tests conducted in Chile on a three story frame braced with SMA wires. Stiffness against lateral translation of the steel-braced frame is

selected to be the same as for the SMA-braced frame in order to make a valid comparison between the effectiveness of each strategy.

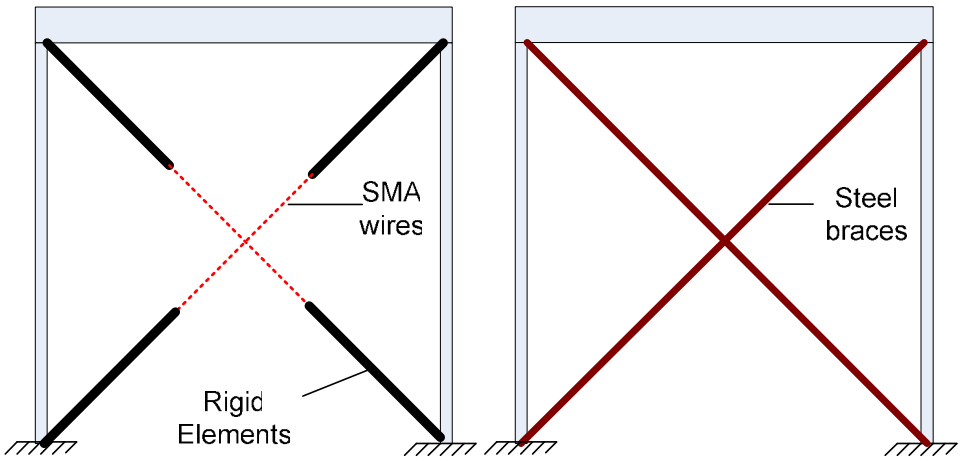



**Fig. 41.** One story frames with SMA and steel bracing elements

The equation governing dynamic response of the system is given by (Chopra, 2001):

$$m\ddot{u} + c\dot{u} + ku + f_{brace} = f_{ext} \tag{4}$$

where $m$, $c$, and $k$ denote mass, damping, and stiffness, respectively; $u$, $\dot{u}$, and $\ddot{u}$ are the dynamic responses of the frame (displacement, velocity, and acceleration) relative to the base; and $f_{ext}$ is the external loading. $f_{brace}$ represents the horizontal component of the force exerted by either the SMA or steel damping elements. The fuzzy model developed in the previous section is used to predict forces imposed on the structure by the SMA damping elements. Equation (1) is rewritten in state-space format to enable the following representation:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{5}$$

where **x** is the state vector of the system, and **y** is the desired output vector. **u** is the input vector of the structure or, in the case, it is the excitation force from a seismic event. **A** and **B** are matrices that define characteristics of the structure, while **C** and **D** describe the output vector that is preferred. After a state-space formulation of the system is realized, solution of the dynamic system is obtained through a numerical simulation using the Dormand-Prince Runge Kutta integration method and Simulink (2007). A fixed time step of 0.01 sec is chosen.

### 6.2.1. Free Vibrations

In this section, responses of the steel-braced and SMA-braced frames to sudden release from an initial displacement (that is, there are no external forces applied) are studied. The magnitude of the initial displacement is chosen such that the SMA bracing elements experience strains close to their maximum superelastic strain limits in the model. Total cross-sectional area and length for each group of CuAlBe SMA wires are 0.59 $cm^2$ and 50 cm, respectively. Simulations are conducted for 1% and 5% viscous damping ratio. Results for the 1% damping case are given in Figs. 42 and 43. Similarly, results for the 5% damping are shown in Figs. 44 and 45. Displacement and acceleration time histories of the first floor of the structure, stress-strain diagrams of the response of the SMA wires, and strain-strain rate phase diagrams are presented.

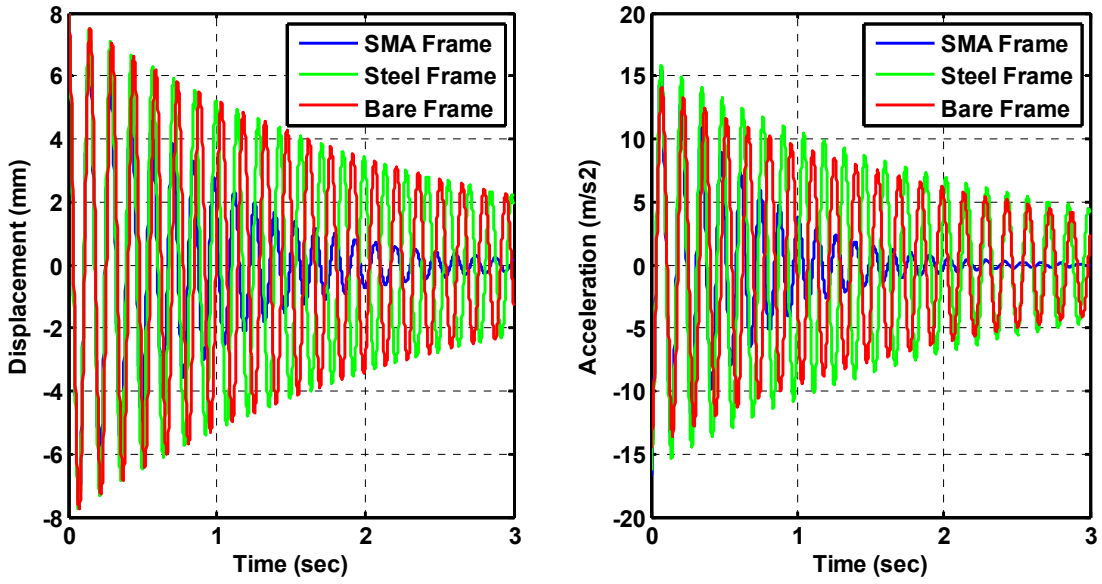



**Fig. 42.** Free vibrations: (a) displacement-time history; (b) acceleration-time history (ξ = 1%)

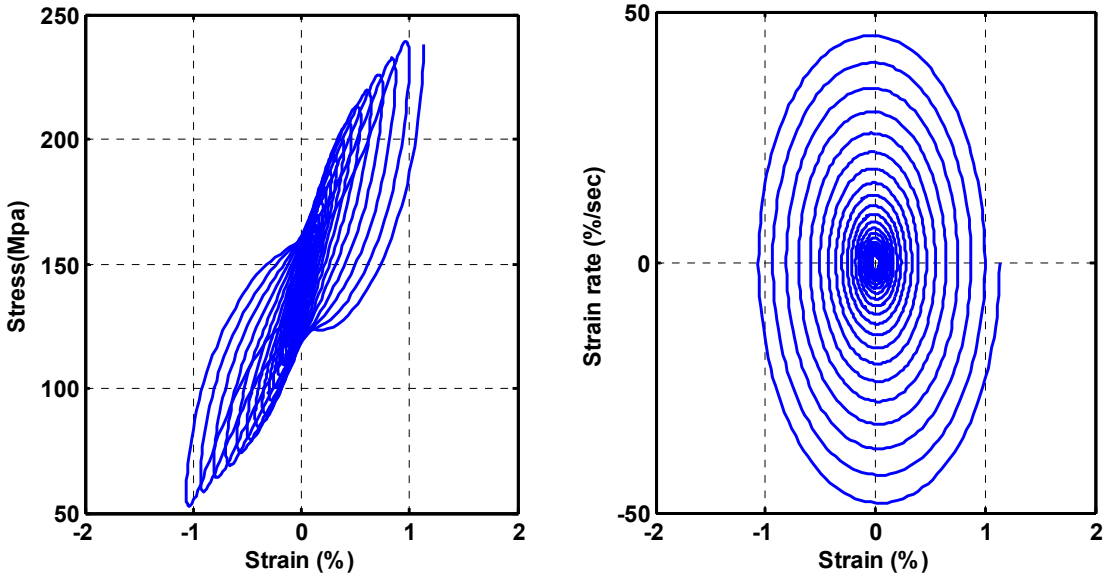



**Fig. 43.** Free vibrations: (a) stress-strain diagram of SMA wires; (b) phase diagram of strain and strain rate. (ξ = 1%)

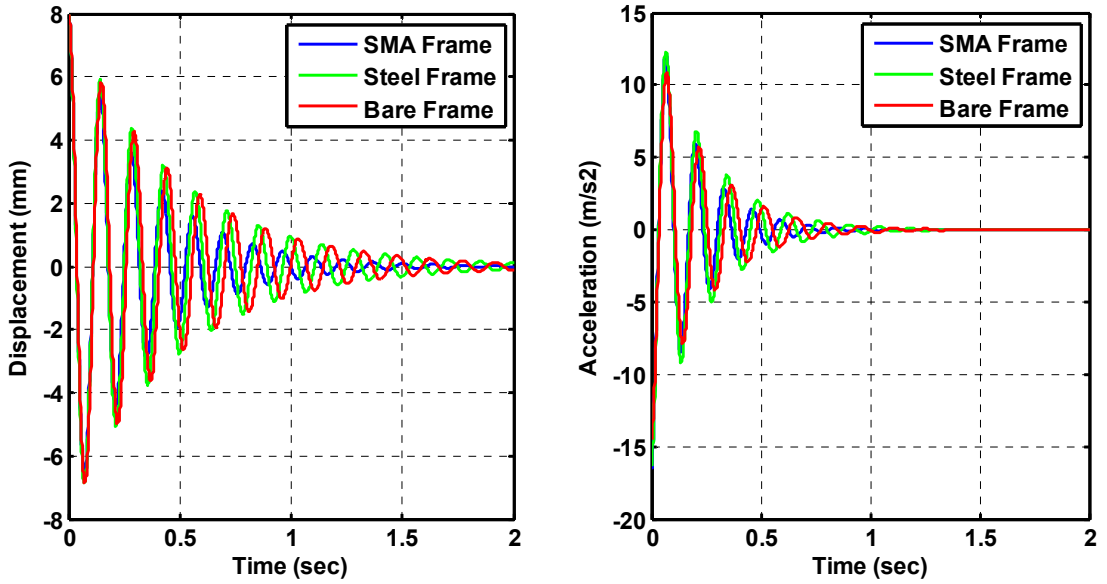



**Fig. 44.** Free vibrations: (a) displacement-time history; (b) acceleration-time history ($\xi$ = 5%)

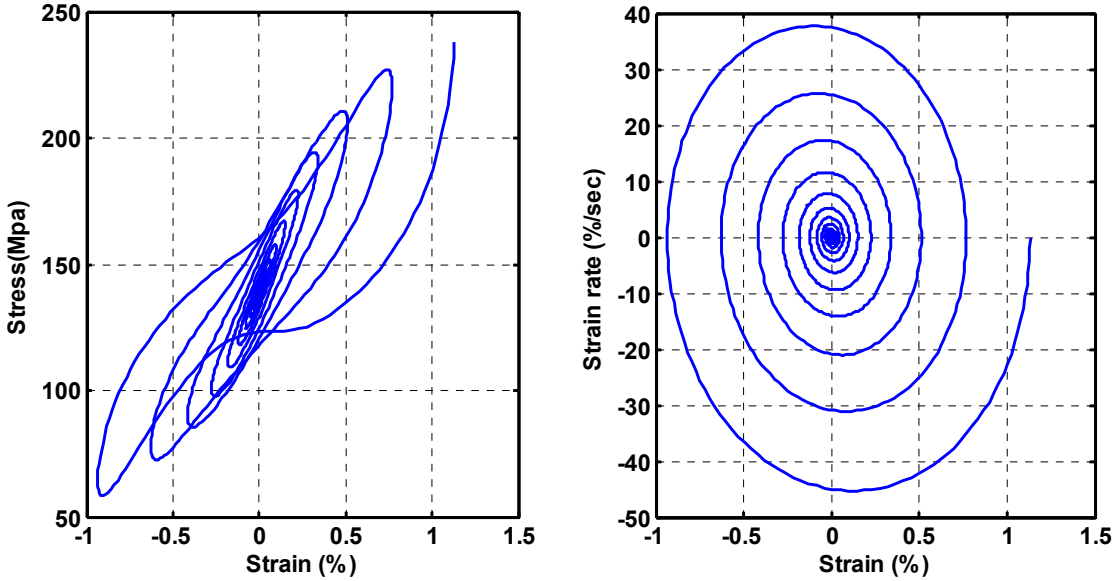



**Fig. 45.** Free vibrations: (a) stress-strain diagram of SMA wires; (b) phase diagram of strain and strain rate. ($\xi$ = 5%)

Based on the results of Figs. 42-44, it can be concluded that the frame braced with SMA wires damps out the motion more rapidly than either the bare frame or the steel-braced frame for both damping ratios. As expected, the response of the frame in all cases dies out in proportion to increasing values of the damping. Note that response modification is not significant for the steel-braced frame. This is also expected because the stiffness of steel braces are equalized with the stiffness of the SMA wires for purposes of comparison; this results in a relatively small area of steel since the modulus of elasticity of steel is considerably higher than that of the SMA wires. Therefore, the steel-braced frame experiences oscillations that have a shorter period than the SMA-braced frame due to the modest increase of lateral stiffness. Finally, since steel remains linear and elastic during these oscillations, there is no additional dissipation of energy by the steel which results in a similar response to that for the bare frame.

### 6.2.2. Harmonic Excitations

Next, the dynamic response of several SDOF frames that are equipped with SMA and steel braces (see Fig. 41) is studied. A forcing function that has a harmonic excitation is applied to the first floor as follows:

$$F = F_{ext} = P_0 \sin(\omega t) \tag{6}$$

where $P_0$ is the amplitude of the force, and $\omega$ is the forcing frequency. For the SMA-braced frame, each brace has SMA wires that have a total area of 0.98 $cm^2$ and length of 50 cm. These values are selected to enable the SMA wires to operate in the superelastic range for the given excitation. $P_0$ is chosen such that maximum deformation of the SMA wires does not exceed the strain and strain rate limits of the model. Furthermore, in order to study the effect of excitation frequency, frequency-response curves for both displacement and acceleration are created using Matlab with 1% and 5% viscous damping ratios of the frame.

Figs. 46 and 47, show frequency-response curves of displacement and acceleration for two different damping ratios with semi-log plots. The excitation frequency on the horizontal axis is normalized with the natural frequency of the frame.

Note that the maximum response for the bare frame occurs when the excitation frequency is equal to the natural frequency of the frame ($\omega/\omega_n = 1$). However, for SMA and steel braced frames, the maximum response shifts to the right slightly due to the increase in stiffness which also increases the natural frequency of the frame.

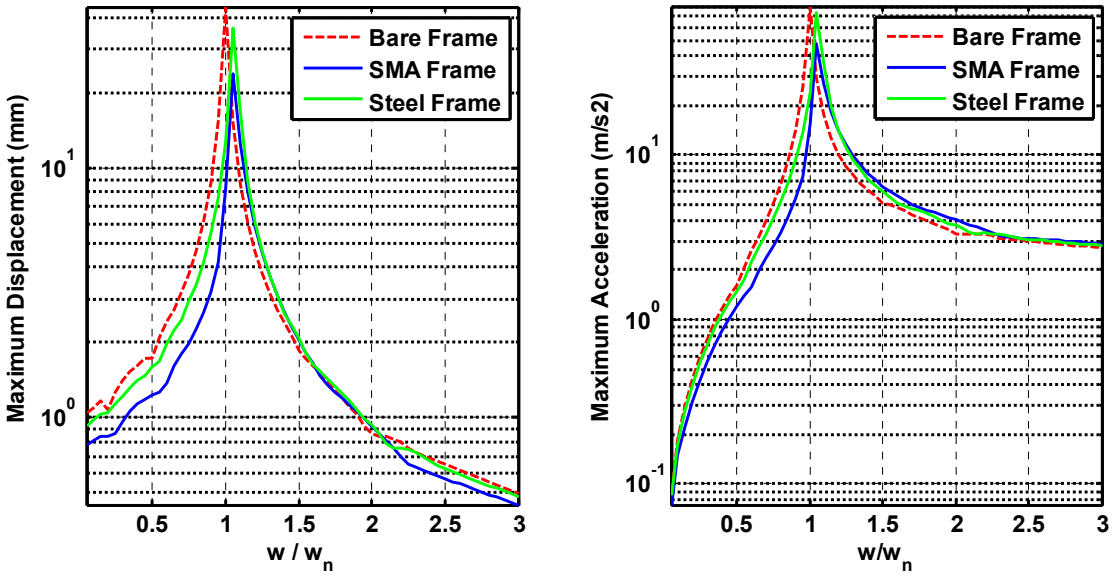



**Fig. 46.** Harmonic excitations: frequency-response curve of (a) displacement; (b) acceleration ($\xi = 1\%$)

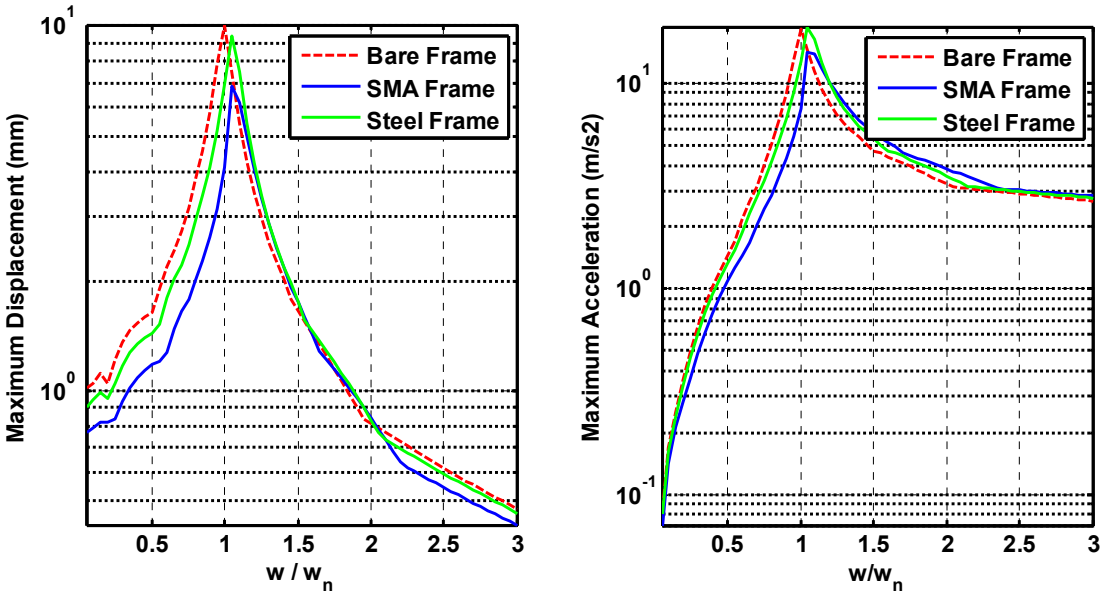



**Fig. 47.** Harmonic excitations: frequency-response curve of (a) displacement; (b) acceleration ($\xi$ = 5%)

For both damping ratios, the maximum displacement of the frame experiences a decrease of between 29% and 46% when SMA wires are attached to the frame. However, the relative effectiveness of the superelastic damping decreases with increasing values of viscous damping of the structure itself. The same conclusion can be drawn also for the maximum acceleration of the frame. For this case, decreases of 40 % and 25 % correspond to damping ratios of 1 and 5%, respectively.

Although it is not as significant as for the SMA-braced frame, there is a reduction of between 25 % and 14 % in the maximum displacement for the steel-braced frame. also, maximum acceleration of the frame decreases 10 % for both 1 and 5% damping ratios.

In general, it is observed that an SMA-braced frame significantly attenuates the response of the frame especially at low frequencies; however, viscous damping is more effective at high frequencies. This conclusion is in agreement with the study of Fosdick

and Ketema (1998) that reports on passive vibration characteristics of a one degree of freedom oscillator.

### 6.2.3. Earthquake Excitation

In this section, response of a SDOF structure that is subjected to ground shaking caused by an earthquake is examined. In particular, numerical simulations are conducted for a bare frame, and steel- and SMA-braced frames as they respond to the 1995 Kobe earthquake record (Fig. 48). The external force given in equation (4) can be expressed as $F_{ext} = -m\ddot{u}_g$, where $m$ is the mass of the frame and is $\ddot{u}_g$ the ground acceleration. Peak acceleration of the Kobe record is scaled to five different amplitudes in order to compare responses for a variety of intensities of the same ground motion.

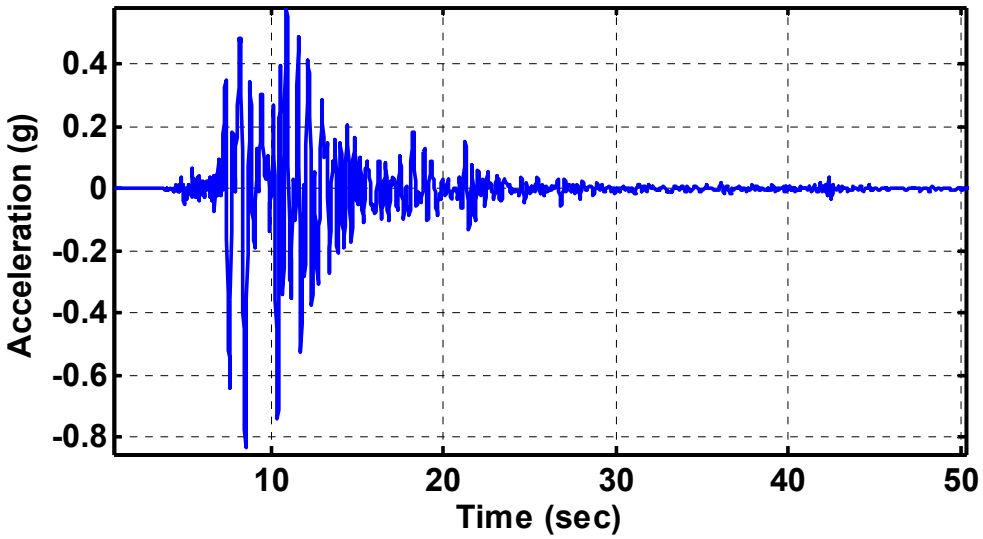



**Fig. 48**. Acceleration record of 1995 Kobe earthquake

Several plots and a table are presented below to enable analysis of the results. Fig. 49 shows the displacement-time history while Fig. 50 gives the acceleration-time

history of the frame for three configuration cases. A stress-strain diagram for the SMA wires is plotted in Fig. 51. In all of these plots, the excitation used is the Kobe record that has been scaled to a peak ground acceleration (PGA) of 1.5 g. Results for the other selected levels of peak ground acceleration are summarized in Table 5 and Fig. 52.

The SMA-braced frame attenuates deformation and acceleration responses of the frame considerably (see Table 5). When the different amplitudes of excitation are considered, an average decrease of 32% and 35% in maximum displacement and acceleration, respectively, is recorded for the SMA-braced frame. Also, for the range of PGA levels chosen, it is observed that the percent reduction in displacement response decreases with increasing PGAs while the trend is vice versa for maximum acceleration as shown in Fig. 52.

In the case of a steel-braced frame, the response reduction is much more modest. That is, there is a constant decrease of 14 % and 10 % in displacement and acceleration, respectively, for all amplitudes of ground motion.

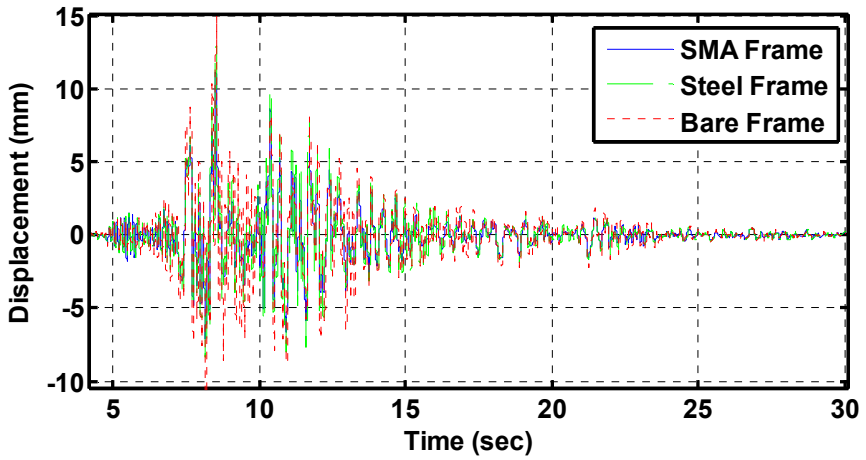



**Fig. 49.** Earthquake excitation: displacement-time history

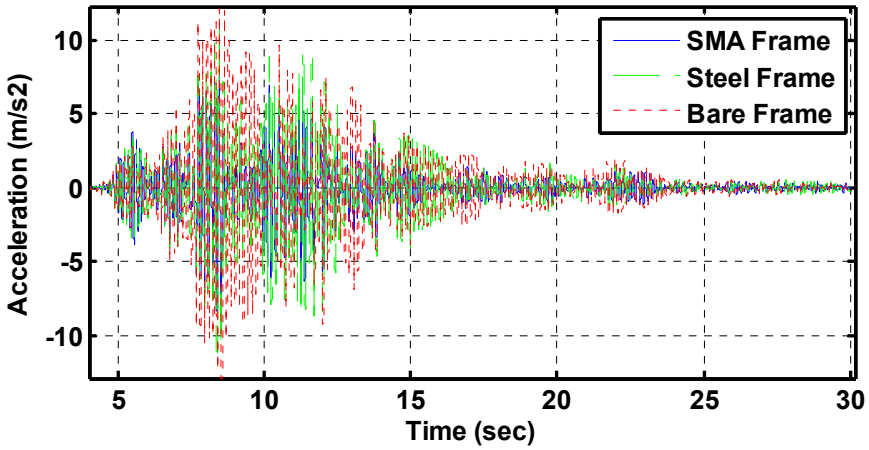



**Fig. 50.** Earthquake excitation: acceleration-time history

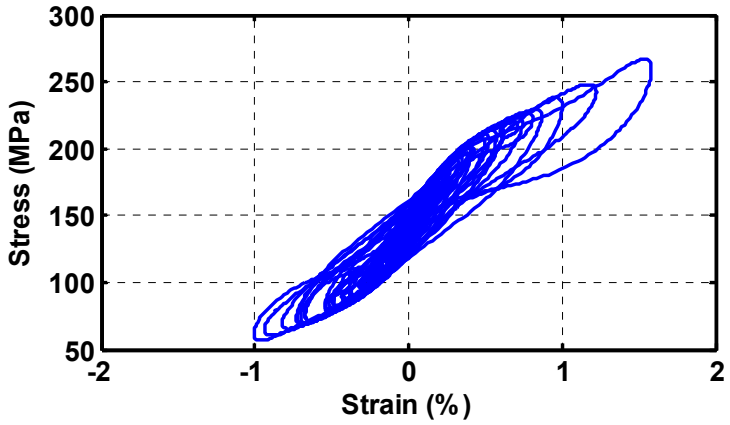



**Fig. 51.** Earthquake excitation: stress-strain diagram

**Table 5.** Maximum response of the frames to the scaled Kobe earthquake record

| PGA (g) | Maximum Displacement (mm) | | | Maximum Acceleration ($m/s^2$) | | |
|---|---|---|---|---|---|---|
| | Bare Frame | SMA Frame | Steel Frame | Bare Frame | SMA Frame | Steel Frame |
| 0.4 | 4.03 | 2.41 | 3.45 | 3.23 | 2.26 | 2.89 |
| 0.8 | 8.06 | 5.33 | 6.89 | 6.46 | 4.30 | 5.78 |
| 1.0 | 10.07 | 6.94 | 8.61 | 8.07 | 5.22 | 7.23 |
| 1.2 | 12.08 | 8.60 | 10.34 | 9.68 | 6.10 | 8.67 |
| 1.5 | 15.10 | 11.19 | 12.92 | 12.10 | 7.61 | 10.84 |

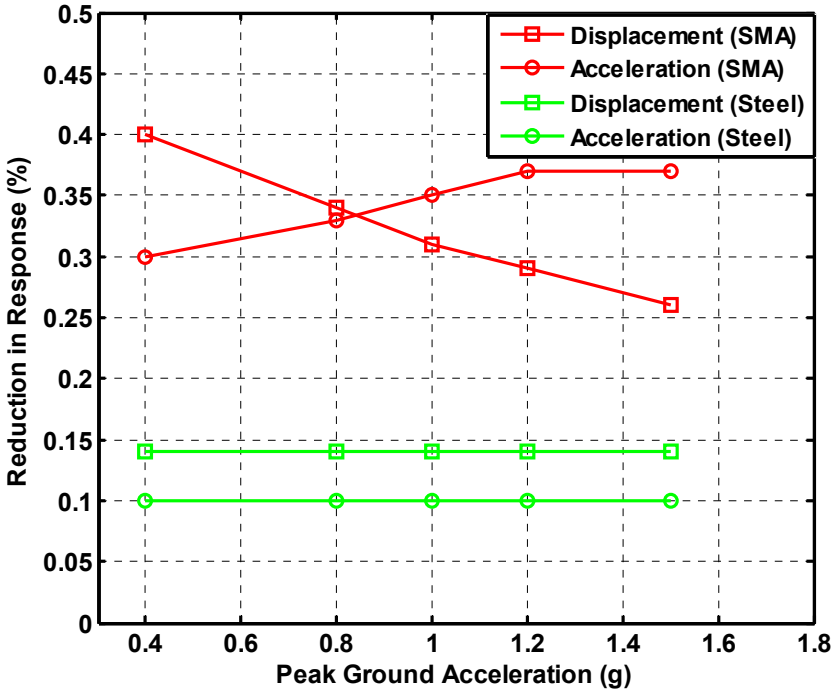



**Fig. 52**. Percent reduction of maximum displacement and acceleration of SMA- and steel-braced frames

### 6.3. Dynamic Analysis of a Three Story Benchmark Building

In this section, time-history analyses of a three story benchmark building that has SMA damping elements installed (see Fig. 53. are conducted. For comparison purposes, simulations of the uncontrolled and steel-braced frames are also implemented. The steel braces are designed to have the same initial stiffness as the SMA braces. Also linear and elastic behavior is assumed for steel members. Mass, stiffness and the damping coefficient of each story are summarized in Table 6

**Table 6.** Dynamic characteristic of benchmark building

| Floor | Mass (kg) | Stiffness (kN/m) | Damping Coefficient (kN-s/m) |
|---|---|---|---|
| 1 | 6,000 | 1,595 | 5,388 |
| 2 | 6,000 | 1,038 | 8,055 |
| 3 | 6,000 | 2,488 | 6,041 |

Rayleigh damping of 5 % is assumed in order to calculate the damping matrix (Chopra 2001). The area and the length of the SMA bracing elements for each floor are optimized using a genetic algorithm technique that is explained in more detail in the subsection that follows.

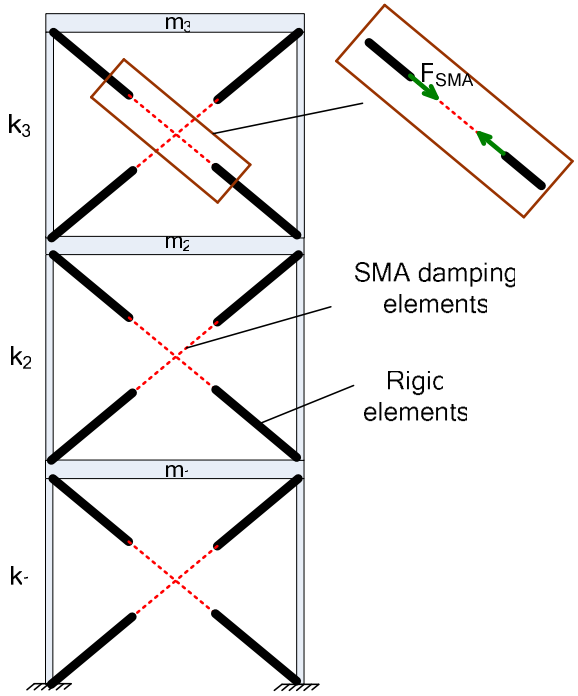



**Fig. 53.** Three story benchmark building with SMA braces

### 6.3.1. Optimization of SMA Bracing Elements

In order to take advantage of hysteretic behavior of the SMA material, the strain levels experienced by the wire during an earthquake should be within their superelastic range. An SMA behaves similar to linear-elastic material at small strains. However, strain hardening is observed after the phase transformations are completed. Here, since the wires are prestressed, the initial linear-elastic behavior is avoided. Yet, it is still necessary to ensure that the displacement response of the braces remains in the superelastic strain range, which requires that the SMA bracing elements need to be designed carefully. In addition, the equivalent viscous damping of the SMA wires changes nonlinearly with increasing strain amplitude (Dolce and Cardone, 2001). Therefore, it is not practical to decide salient parameters of the SMA braces by a trial and error procedure. For example, the length and the area of the SMA wires directly affect the strain amplitude that the wires undergo in a given earthquake.

Therefore, to overcome these problems a non-dominated multi-objective algorithm (NSGA-II) is employed for optimization (Deb and Goel, 2001; Kim and Roschke, 2006) of these characteristics of the SMA wire for each floor. A total of six variables are adjusted in each chromosome of the population: the length and area of each SMA brace in each floor. Four objective functions (peak and RMS drift and acceleration) are measured for the near-fault artificial earthquake and organized into a set of Pareto fronts.

Near-fault ground motions, which are often characterized by intense velocity, high amplitude and short duration impulses, are accorded special consideration in seismic engineering. Substantial increases in interstory drift and base shear of a structure have been observed for this kind of earthquake (Akkar et al. 2005, Mavroeidis *et al.* 2004). Therefore, an artificial earthquake that represents near-fault ground motions is selected for numerical optimization in order to investigate the effectiveness of

SMA damping elements against near-fault excitations. The length and area of the SMA wires that are determined by GA optimization are listed in Table 7.

**Table 7.** Length and area of SMA damping elements

| Floor | $L_{SMA}$ (m) | $A_{SMA}$ ($cm^2$) |
|---|---|---|
| 1 | 1.14 | 0.69 |
| 2 | 1.20 | 0.83 |
| 3 | 0.89 | 0.36 |

### 6.3.2. Time History Analyses and Results

The equation governing the dynamic response of the system is given by:

$$M\ddot{u} + C\dot{u} + Ku + F_{brace} = -M\ddot{u}_g \tag{7}$$

$$M = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}$$

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} \tag{8}$$

$$C = \alpha M + \beta K$$

$$\alpha = 2\frac{\zeta\omega_1\omega_3}{\omega_1 + \omega_3},\ \beta = 2\frac{\zeta}{\omega_1 + \omega_3}$$

where $M$, $C$, and $K$ denote mass, damping, and stiffness matrices, respectively; $u$, $\dot{u}$, and $\ddot{u}$ are the dynamic response of the frame (displacement, velocity, and acceleration) relative to the base; and $\ddot{u}_g$ is the acceleration record of the ground motion. $F_{brace}$ is the vector of forces exerted by SMA damping elements or steel braces. $\zeta$ is the critical damping ratio, and $\omega_i$ are the fundamental frequencies of the structure. $\alpha$ and $\beta$ are coefficients that are used to define a Rayleigh damping matrix from the first and third natural frequencies. Forces from SMA damping elements are computed using the fuzzy model described above. SMA wires are each prestressed to a value of 140 MPa.

Equation (7) is rewritten in state-space format, and Runge-Kutta integration is used to solve the first order differential equations.

Five earthquake records that are considered along with an artificial record in numerical simulations are the 1986 North Palm Spring (California), 1994 Northridge (California), 1995 Kobe (Japan), 1999 Chi-chi (Taiwan), and 1999 Bolu (Turkey) earthquakes. Maximum accelerations of all records are individually scaled to a PGA level (Fig. 54) so that the superelastic range of the SMA wires is not exceeded for the given characteristics of the building and earthquake record.

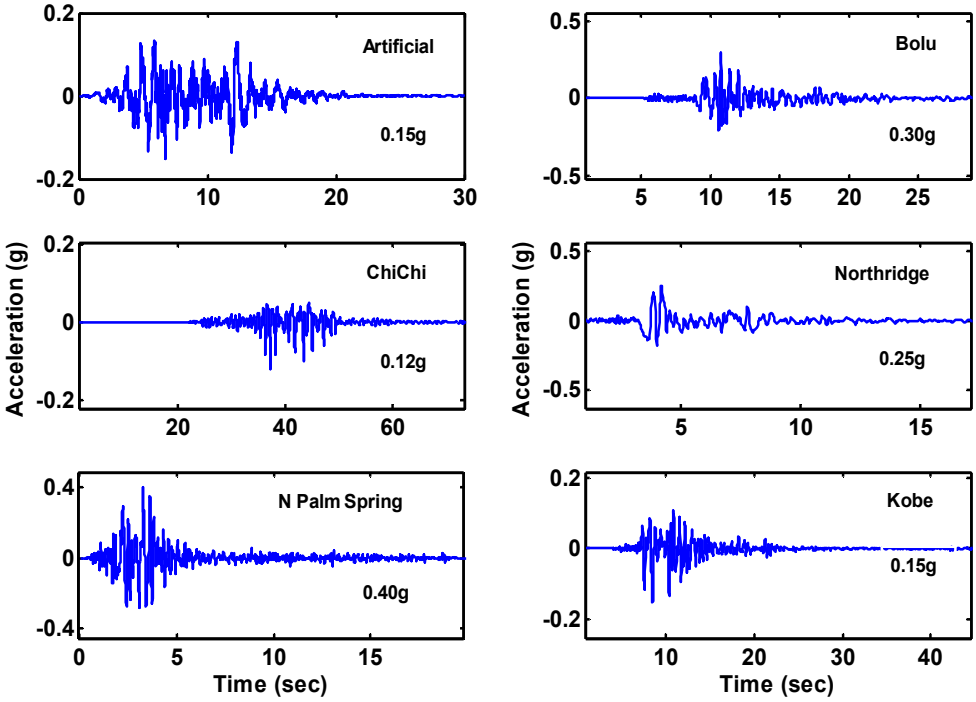



**Fig. 54.** Adjusted acceleration records of earthquakes that are used in simulations

The results for the Bolu ground motion are presented in detail below while the results for the other excitations are summarized with four performance indices that are

defined later. Fig. 55 shows profiles of maximum story displacement, maximum story acceleration and maximum interstory drift. Displacement and acceleration time-histories of the first, second, and third floor that result from the Bolu temblor are plotted in Figs. 56 and 57. As shown in both figures, the story displacements relative to the ground experience a considerable decrease when the SMA dissipating braces are present rather than the steel braces. First, second and third floor drift reductions of displacements for the SMA frame are 43 %, 40 %, and 33 %, whereas for the steel-braced structure, there is a 5 % and 33 % decrease for the first and second story drifts, but a 204 % increase in third floor drift. Maximum absolute acceleration of each floor experiences an average of a 20 % decrease for the SMA frame, with a maximum value of 4.65 m/sec$^2$. The steel-braced frame has a peak acceleration of 5.78 m/s$^2$ which is almost the same as that of the third floor for the bare frame, although it has a 41 % and 14 % reduction for the first and second floors. Also, structural responses of the system damp out more rapidly for the SMA-braced frame.

Fig. 58 shows stress-strain curves that are obtained from the developed fuzzy model for SMA wires installed below the first, second and third floor. Note that the initial pre-strain is not included with the strain values shown.

In order to quantitatively evaluate the results of numerical simulations for each excitation, four performance indices are defined and calculated as follows:

$$J_1 = \max\left\{\max_j \left|\frac{u_{j,cont.}}{u_{j,unc.}}\right|\right\},\ J_2 = \max\left\{\max_j \left|\frac{\ddot{u}_{j,cont.}}{\ddot{u}_{j,unc.}}\right|\right\}$$

$$J_3 = \max\left\{rms_j \left|\frac{u_{j,cont.}}{u_{j,unc.}}\right|\right\},\ J_4 = \max\left\{rms_j \left|\frac{\ddot{u}_{j,cont.}}{\ddot{u}_{j,unc.}}\right|\right\} \tag{9}$$

where u and ü denote relative story displacement and absolute story acceleration, respectively, and *j* represents the story that is considered. For the controlled case either SMA braces or steel braces are present in the building. All four indices are calculated

for both cases. The first two indices are based on the peak relative displacement and absolute acceleration of each floor, while the other two evaluation criteria consider the entire duration of the motion and compute the root-mean-square (RMS) of the peak relative displacement and absolute acceleration for each floor.

Fig. 59 shows the computed performance indices for the SMA- and steel-braced frames. Also, as a reference $J_i = 1$ for the uncontrolled frame. As shown in the figure, there is a substantial decrease in peak interstory drift and RMS interstory drift while SMA bracing elements are present. The reduction in peak displacement is between 56 % for the North Palm Spring ground motion and 14 % for the Chi-Chi earthquake. A minimum of 35 % decrease exists for the RMS relative displacement for all earthquakes that are simulated. SMA braces are not only effective for reducing displacement response but also modestly decrease the peak acceleration that the building experiences. Moreover, the RMS acceleration, which is important for comfort of the inhabitants during an earthquake, decreases at least 18 % except for the Chi-Chi earthquake which has only a 3 % reduction. Examination of results for the steel-braced frame shows that there is a moderate decrease in peak interstory displacement for most of the ground motions. However, this reduction usually comes at the expense of an increase in RMS interstory drift or RMS acceleration.

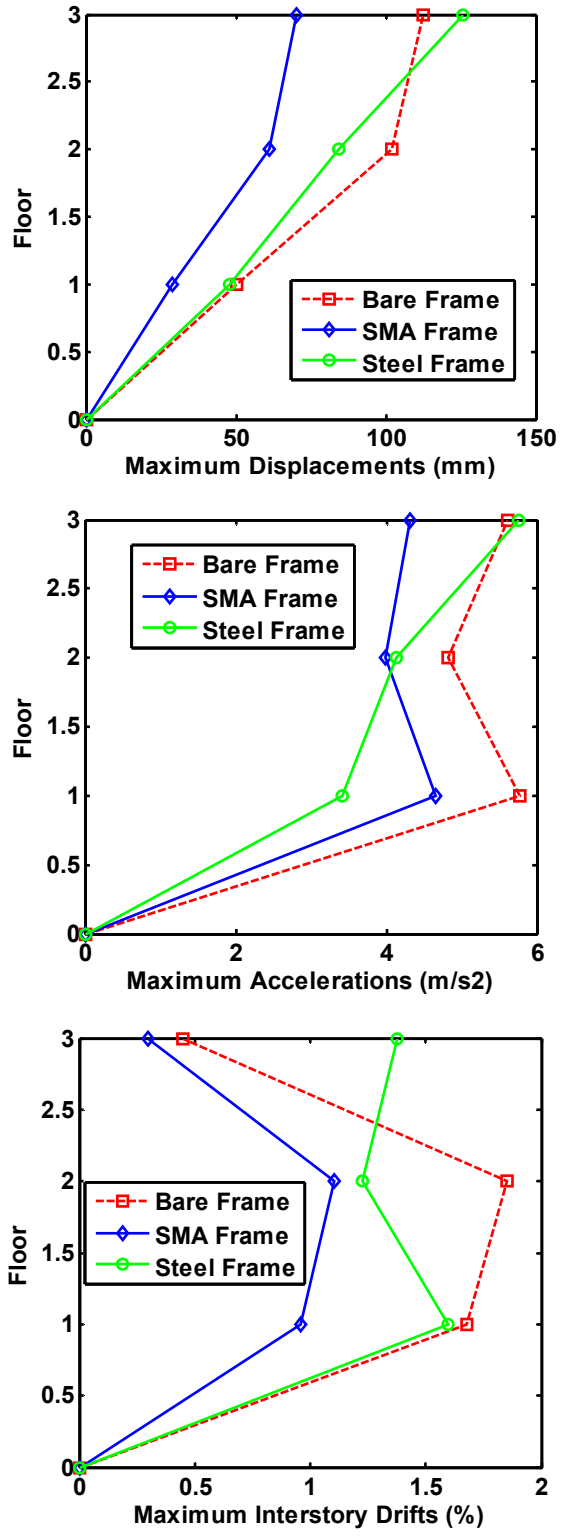
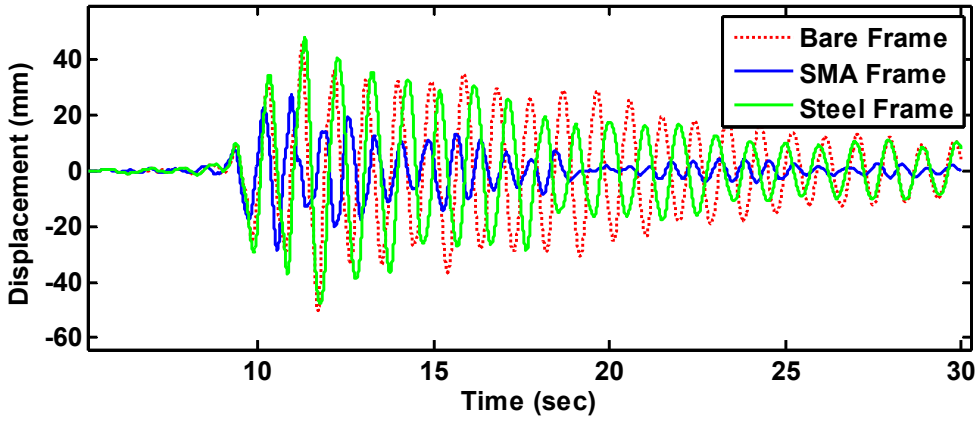
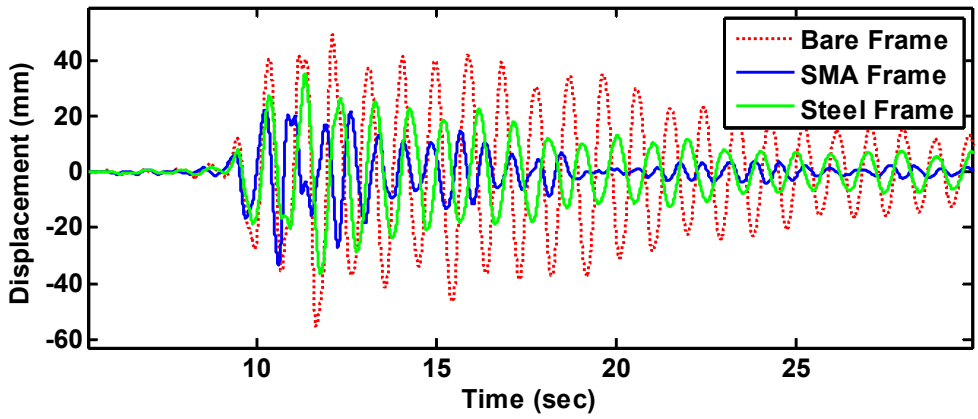


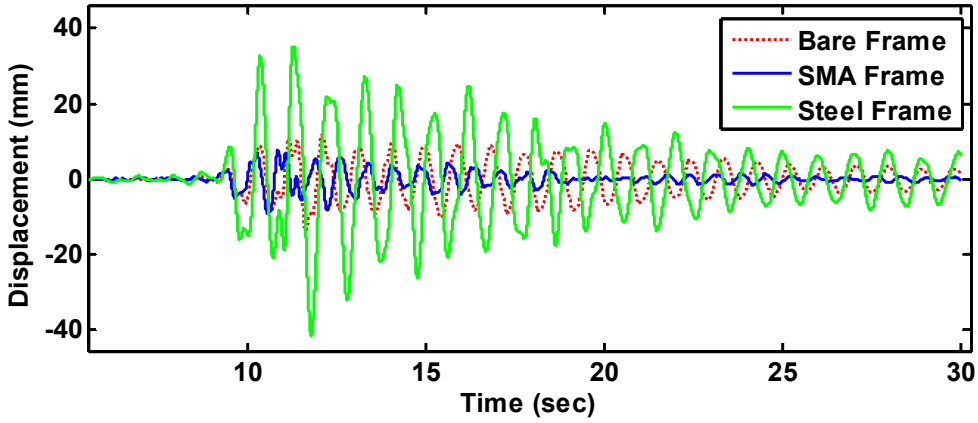

**Fig. 55**. Profile of floor peak displacement, accelerations and interstory drifts




(a)

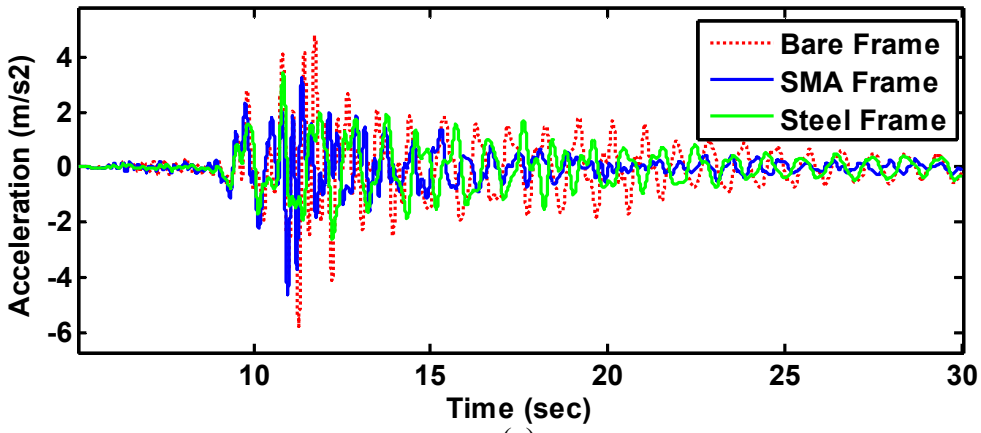



(b)

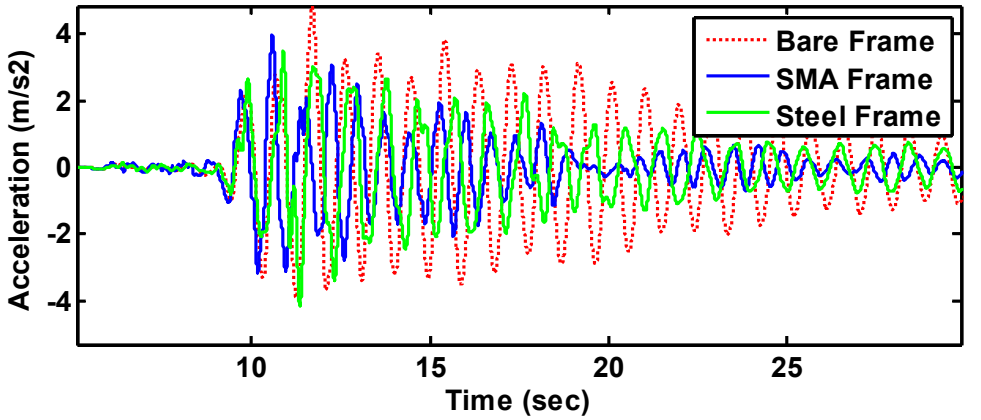



(c)

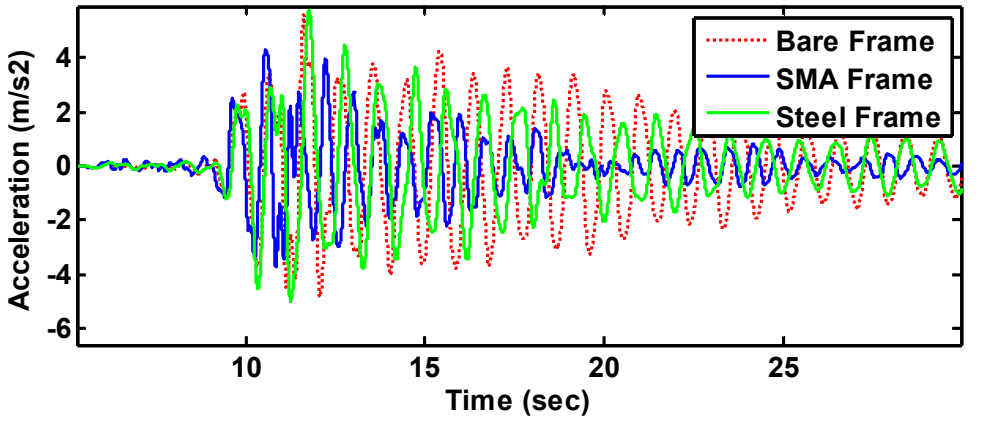
**Fig. 56.** Relative displacement-time histories for (a) first; (b) second; and (c) third floor




(a)

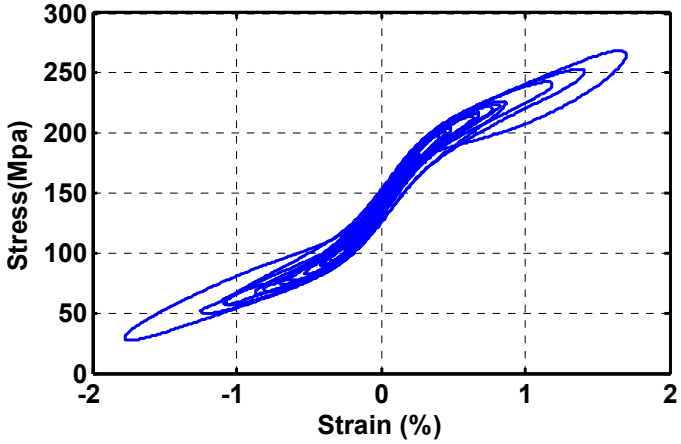



(b)

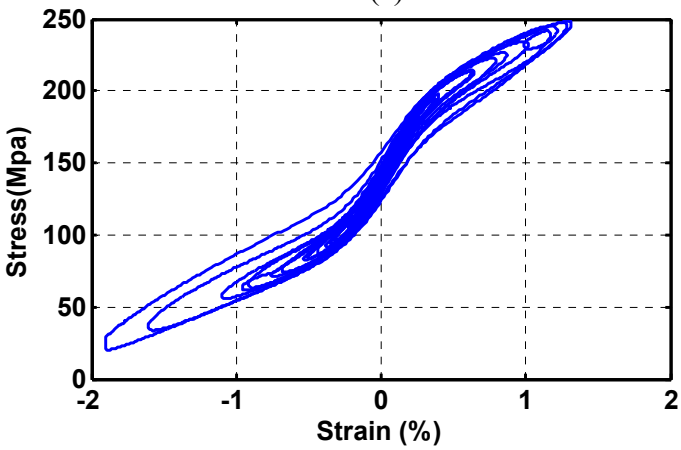



(c)

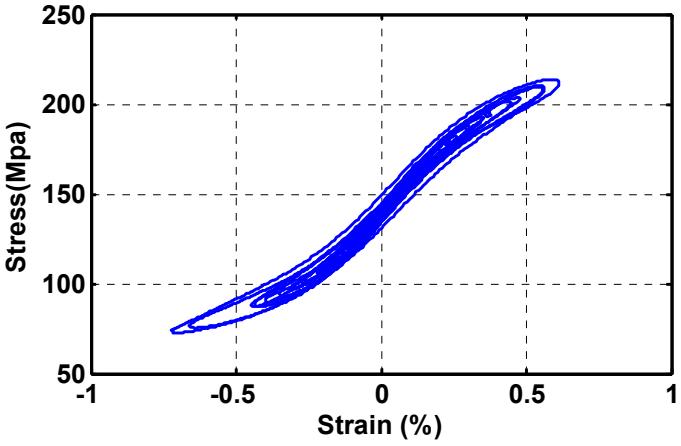
**Fig. 57.** Absolute acceleration-time histories for (a) first; (b) second; and (c) third floor




(a)




(b)




(c)

**Fig. 58**. SMA wires strain-stress curves for (a) first; (b) second; and (c) third floor

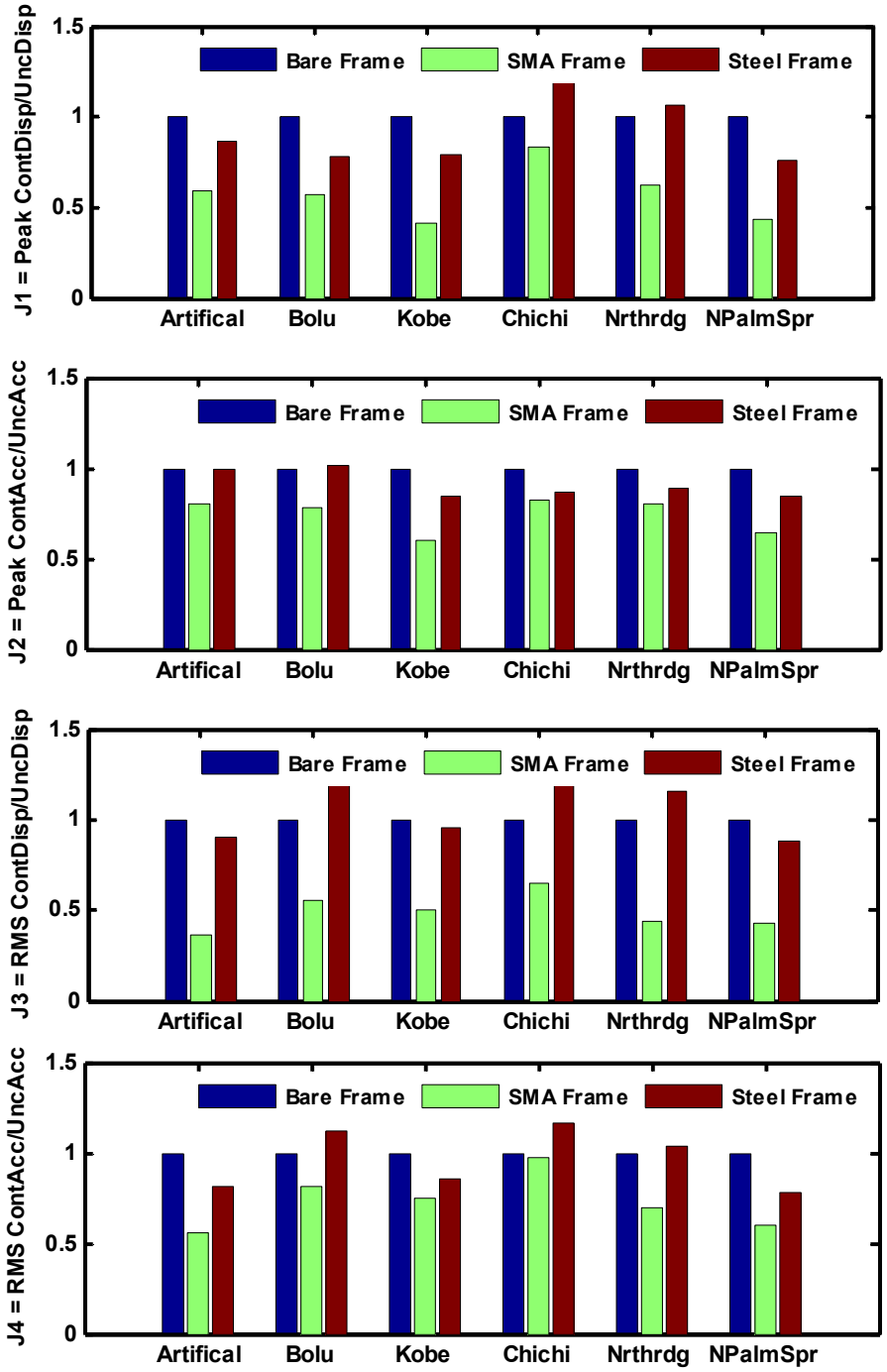



**Fig. 59.** Performance indices for different excitations

### 6.4. Dynamic Analysis of an Isolated Bridge

In this section, an isolated five-span continuous bridge (Lee and Kawashima, 2006) is studied to compare the performance of two isolation systems: a high-damping rubber bearing isolator and a natural rubber bearing that is augmented with prestressed SMA damping elements. One of the interior spans of the bridge is used for the analysis that follows. Values of the effective mass of the deck ($m_d$) and piers ($m_p$), the stiffness of piers ($k_p$), the stiffness of a natural rubber bearing ($k_b$), the elastic ($k_1$) and post-yielding ($k_2$) stiffness of the high-damping rubber bearing and its yielding displacement ($u_y$) are summarized in Table 8. 2% viscous damping is assumed for the pier and natural rubber bearing. Models of the bridge that have two degrees of freedom for both isolation systems are shown in Fig. 60.

**Table 8.** System parameters

| | |
|---|---|
| $m_d$ | 600 ton |
| $m_p$ | 243.5 ton |
| $k_p$ | 112.70 MN/m |
| $k_b$ | 7.93 MN/m |
| $k_1$ | 47.60 MN/m |
| $k_2$ | 9.04 MN/m |
| $u_y$ | 0.016 m |

The fuzzy model described earlier is used to predict the instantaneous force for the prestressed SMA damping elements. The high-damping rubber bearings are modeled using a Bouc-Wen model (Wen, 1976). The hysteretic force of the elastomeric bearing is given by

$$F_{HDR} = \alpha \frac{Q_y}{u_y} x + (1-\alpha) Q_y z \tag{10}$$

where $\alpha$ is the ratio of the post yielding to the elastic stiffness, $Q_y$ is the yield strength, $x$

= $u_2$-$u_1$ is the relative displacement, and $z$ is a hysteretic dimensionless quantity governed by the following differential equation:

$$u_y \dot{z} + \gamma |\dot{x}||z|^{n-1} z + \beta \dot{s} |z|^n - \dot{x} = 0 \tag{11}$$

where β, γ, and $n$ are dimensionless parameters that control the shape of the hysteretic curve; they are specified to be 0.5, 0.5, and 1, respectively, for the simulations that follow.

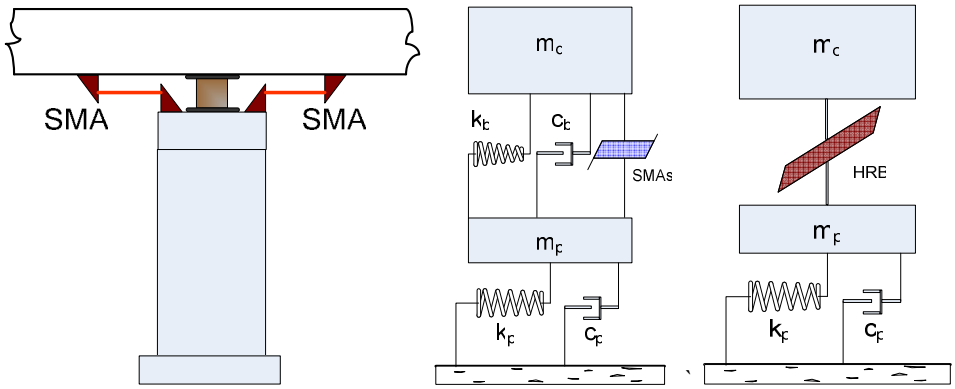



**Fig. 60.** Physical models of the isolated bridge

Equations of motion for the bridge pier and the deck are as follows:

$$\begin{aligned} m_p \ddot{u}_1 + c_p \dot{u}_1 + k_p u_1 - R_{iso} &= -m_p \ddot{u}_g \\ m_d \ddot{u}_2 + R_{iso} &= -m_d \ddot{u}_g \end{aligned} \tag{12}$$

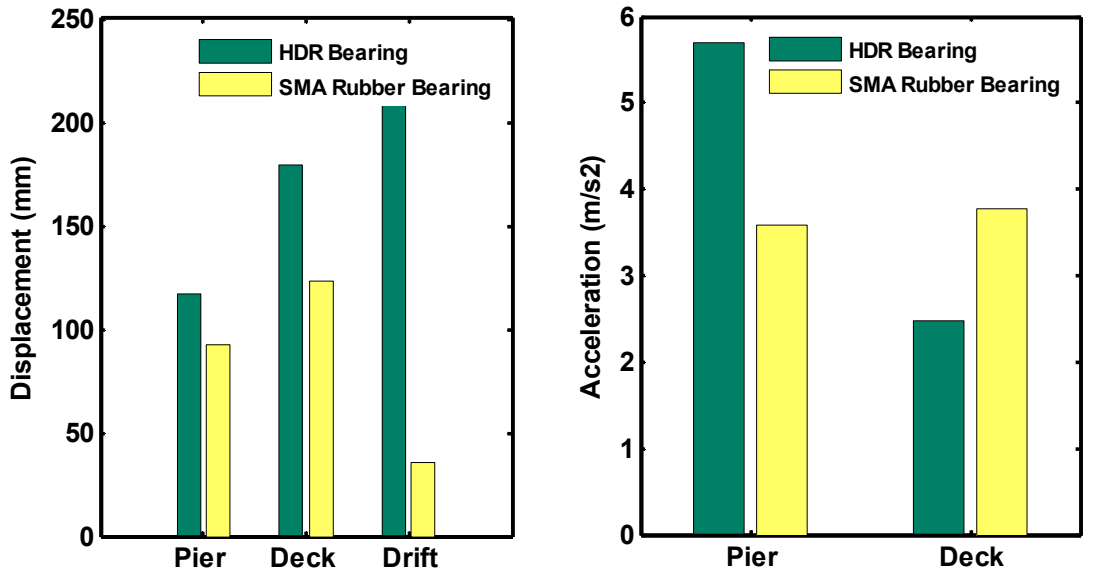
where $R_{iso} = F_{HDR}$ or $R_{iso} = F_{SMA} + c_b(\dot{u}_2 - \dot{u}_1) + k_b(u_2 - u_1)$ represent the restoring forces of the high damping rubber bearing or the SMA-augmented natural rubber bearing.

An artificial earthquake with a PGA of 0.25 g (see Fig. 54) is used for excitation. The length and the area of the SMA damping elements are selected as 2.0 m and 0.0372 $m^2$, respectively. These values are chosen by a trial and error process so that the maximum strain of the SMA remains in the superelastic range. After the equations of motion have been cast in a state-space format, the solution is obtained by using Runga-Kutta integration with a variable time step in Matlab.

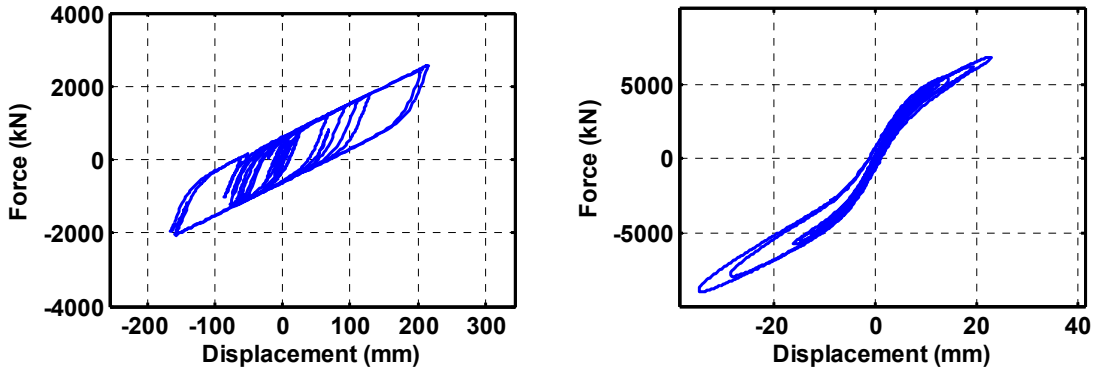
The displacement and acceleration responses of the isolated bridge are summarized in Fig. 61. Also, force-displacement curves of the high-damping rubber bearing and SMA damping elements are given. It can be seen that when the SMA damping elements are used with a low-damping rubber bearing not only does the maximum displacement response of the pier and deck decrease, but also the peak displacement of the bearing experiences a significant reduction. Nevertheless, the acceleration response of the deck increases by 53 % as an expense.

### 6.5. Nonlinear Dynamic Analysis of SDOF Frame

Shape memory alloys that can restore a structure to its original position even after considerable deformation can effectively limit post-earthquake displacement response of a system against a near-fault ground motion.




(a)




(b)

**Fig. 61.** (a) Peak displacement and acceleration response of the isolated bridge; (b) force-displacement curves of high-damping rubber bearing and SMA damping elements

Although the recentering ability of SMAs have been discussed thoroughly in the relevant literature, experimental and analytical studies that actually display this capability are rare. In this section, a nonlinear time-history analysis of a single degree of freedom (SDOF) system is conducted to explore the recentering characteristics of SMAs.

A chevron-like braced-steel frame that has been studied by Dicleli and Mehta (2006) is selected for the current analysis (see Fig. 62). The braced frame has a period of 0.24 sec, and a viscous damping ratio of 5 %. Sizes of the beam, columns, and steel bracing elements are also given in Fig. 8. The height and width of the frame are 3.6 m and 6 m, respectively. The columns and braces are modeled to numerically simulate bilinear material behavior. In addition, the steel braces are assumed to be designed to carry compressive loads without buckling.

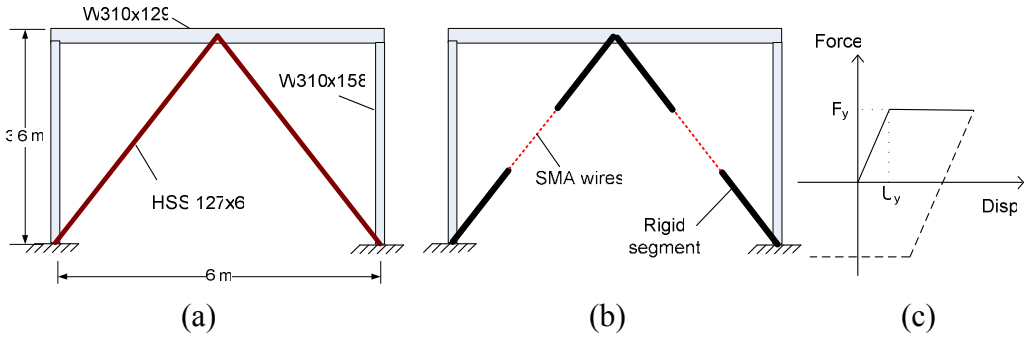



**Fig. 62.** SDOF frames: (a) steel braced frame; (b) SMA frame; and (c) bilinear model

The same frame is also designed with SMA damping elements. For simulation of dynamic behavior of the SMAs, the fuzzy model described earlier is used again. Modulus of elasticity of steel and the SMA are taken as 210,000 MPa and 23,000 MPa, respectively. The yield stress of steel is chosen to be 345 MPa for both columns and braces.

In order to compare performance of the two different bracing systems, the initial axial stiffness of the braces and the maximum base shear for the frame are required to be same. To carry out this determination, the lateral stiffness and yield strength of the steel-braced frame are computed for given element sizes and geometry. Then, the area of the SMA wires is calculated as follows:

$$A_{SMA} = \frac{k_{STEEL}}{E_{SMA}} L_{SMA} \tag{13}$$

where $A_{SMA}$, $L_{SMA}$, $E_{SMA}$ is the area, length and elastic modulus of the SMA wires, and $k_{STEEL}$ is the initial axial stiffness of the steel braces. The length of SMA wires in equation () is selected so that the maximum lateral force that is exerted on the frame by the SMA wires is the same as for a steel-braced frame, and the maximum strain of the wire is within the superelastic range of strain. Therefore, a length of 2.20 m and a corresponding area of 0.0118 $m^2$ are selected for the SMA damping elements. Also, the yield strength and lateral displacement at the top of the frame that corresponds to yield of the steel braces are computed to be 997.5 kN and 8.0 mm, respectively.

The nonlinear equation of the motion is given by

$$m\ddot{u} + c\dot{u} + f_s(u,\dot{u}) + F_{brace} = -m\ddot{u}_g \tag{14}$$

where $m$, $c$, and $f_s(u,\dot{u})$ denote mass, damping, and resisting force, respectively. $F_{brace}$ represents the total lateral force exerted by the SMA or steel braces. Newmark's integration method with $\delta = 1/2$ and $\alpha = 1/4$ is used to solve the equation of motion for each time step. The artificial earthquake described above is scaled to a PGA of 0.9 g and used for the simulations.

Displacement time-histories for steel- and SMA-braced frames are shown in Fig. 63. Also, the total lateral load that is exerted on the frame by the steel or SMA braces is indicated. Maximum displacement of the steel-braced frame is found to be 39 mm and residual deformation of the frame is 6 mm. By contrast, the frame braced with SMA wires returns to its original position, and experiences a maximum displacement of 25 mm. The maximum acceleration for steel and SMA-braced frames is 10.9 and 11.0 $m/sec^2$, respectively. These latter values show that the SMA wires do not substantially increase the peak acceleration of the frame, while the peak lateral and residual displacements are greatly ameliorated.

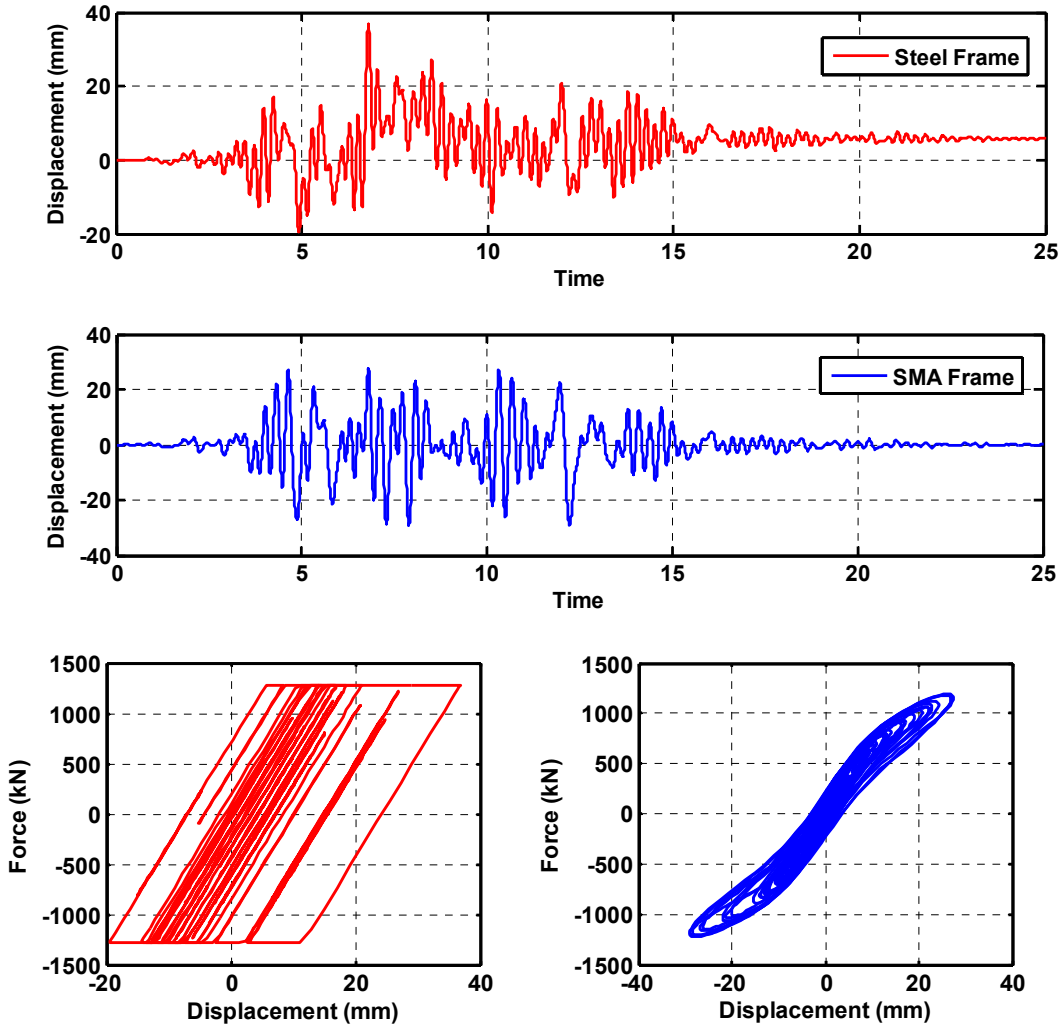



**Fig. 63.** Displacement-time histories and force-displacement diagrams for steel- and SMA-braced frame

### 6.6. Conclusion

A comprehensive numerical study that investigates response modification of single and multiple degree of freedom systems braced with SMA dampers and an isolated bridge that utilizes SMA damping elements is presented. Results of numerical simulation show that, with a few exceptions, SMA damping elements effectively

decrease the displacement response of a structure to strong earthquake motion without increasing the accelearation response. Moreover, a framed structure equipped with SMA damping elements is predicted to return to its original position after the excitation has ceased.

# 7. CONCLUSIONS

In this study, a soft computing approach, namely a fuzzy model, is developed to represent highly nonlinear behavior of superelastic CuAlBe shape memory alloys. Through numerical simulations, potential applications of SMAs to seismic engineering are investigated. The goal is to mitigate the response of structures such as buildings and bridges to strong external excitations.

First, a fuzzy inference system (FIS) that can predict behavior of superelastic SMA wires at different temperatures is developed. Then, in order to consider dynamic effects on response of SMAs, another fuzzy model is created. Data that are used to develop both fuzzy systems are obtained from experimental tests conducted at the University of Chile. Data for the first FIS are obtained from tensile tests on SMA wire conducted by employing cyclic sinusoidal excitations with a frequency of 1 Hz at various temperatures. Experimental results from shake table tests on a three story frame that are braced with SMA wires provide data sets for the second FIS.

A set of numerical simulations are implemented to demonstrate the beneficial use of SMAs in earthquake engineering and to illustrate the applicability of developed fuzzy model of SMA behavior for numerical simulations. In order to illustrate the response modification of SMA damping elements on a simple structural system, a single story frame braced with SMA elements is simulated first. Response of the frame to free vibration, harmonic excitations and earthquake excitations are reported. Next, simulations are expanded to a three story benchmark building, and time-history analyses for six different ground motion records are performed and analyzed. An NSGA-II genetic algorithm is employed to optimize the physical characteristics (length and area) of SMA damping elements at each floor. The same frame is also braced with steel in order to provide the response of a comparable structure in these above simulations.

Next, numerical simulations of a multi-span continuous bridge that is isolated with rubber bearings that are, in turn, augmented with SMA elements is presented.

Results are compared with the isolated bridge that is equipped with high-damping rubber bearings.

Finally, to show the recentering ability of an SMA, nonlinear simulations on a single story building that is braced with SMA elements is discussed. Results for a seismic acceleration time-history are compared with those of a building that has steel braces. The columns of the frame and steel braces are modeled to have elasto-plastic behavior.

Shape memory alloys have unique properties such as superelasticity that enables a device that is carefully calibrated to cause a structure to return to its undeformed position upon unloading. Also SMA dampers provide substantial energy dissipation as a result of hysteretic behavior. If used judiciously they can be considered as an energy dissipation and recentering element in civil engineering structures such as buildings and bridges. However, the complex behavior of SMAs should be understood better before full-scale applications are widely deployed. This study aims to answer some of the questions that may arise while designing structures with SMA elements via modeling the behavior of the material and implementing numerical simulations.

Dynamic analyses of single story and multi story buildings showed that when considered as an innovative bracing system, shape memory alloys can effectively improve performance of structures against seismic excitations. Specifically, displacement response of a building braced with SMA elements experiences at least a 20% more decrease than its conventional steel counterpart in all simulations. Yet, SMA braced buildings are not subjected to an increase in peak acceleration response whereas that increase usually appear to be present for steel braced buildings as an expense of modest decrease in displacement response. Not only do SMA bracing elements alleviate story drifts of a building but they also recenter the structure after cessation of ground motion. Simulations on a single story steel frame that is modeled to show nonlinear behavior for large deformations reveal beneficial properties of SMAs. Although more elaborated study is required, early results from the bridge simulation also show that SMAs can be considered as a candidate to be used in seismic isolation systems.

# REFERENCES

Aizawa, S., Kakizawa, T., and Higasino, M. (1998). "Case studies of smart materials for civil structures." *Smart Materials and Structures*, 7, 617-626.

Akkar, S., Yazgan, U., and Gulkan, P. (2005). "Drift estimates in frame buildings subjected to near-fault ground motions." *Journal of Structural Engineering*, 131 (7), 1014-1024.

AMT (@ Medical Technologies n.v.). (2002). Retrieved from <http://www.amtbe.com>

Andrawes, B., and DesRoches, R. (2005). "Unseating prevention for multiple frame bridges using superelastic devices." *Smart Materials and Structures,* 14, 60-67.

Auricchio, F. (1995). "Shape memory alloys: applications, micromechanics, macromodeling and numerical simulations." PhD Dissertation, Department of Civil and Environmental Engineering, The University of California at Berkeley.

Bartera, F., and Giacchetti, R. (2004). "Steel dissipating braces for upgrading existing building frames." *Journal of Constructional Steel Research*, 60, 751-769.

Boyd, J. G. and Lagoudas, D. C. (1998). "A thermodynamic constitutive model for the shape memory materials part I. The monolithic shape memory alloys." *International Journal of Plasticity*, 6, 805–842.

Brinson, L. C. (1993). "One-dimensional constitutive behavior of shape memory alloys: thermomechanical derivation with non-constant material functions and redefined martensite internal variable." *Journal of Intelligent Material Systems and Structures,* 4, 229–241.

Bruno, S., Valente, C. (2002). "Comparative response analysis of conventional and innovative seismic protection strategies." *Earthquake Engineering and Structural Dynamics*, 31, 1067-1092.

Cerda, M., Boroschek, R., Farias, G., Moroni, O., and Sarrazin, M. (2006). "Shaking table test of a reduced-scale structure with copper-based SMA energy dissipation devices." *Proc. of the 8th U.S. National Conference on Earthquake Engineering*, San Francisco, CA, USA, Paper Nº 1003.

Choi, E., Nam, T. H., and Cho, B. S. (2005). "A new concept of isolation bearings for highway steel bridges using shape memory alloys." *Canadian Journal of Civil Engineering*, 32, 957-967.

Chopra, A. K. (2001). *Dynamics of structures.* Prentice Hall, SaddleRiver, NJ.

Clark, P. W., Aiken, I. D., Kelly J. M., Higashino, M., and Krumme, R. C. (1995). "Experimental and analytical studies of shape memory alloy damper for structural control." *Proc. of SPIE Smart Structures and Materials: Passive Damping*, San Diego, CA, 2445, 241-251.

Copper Development Association Inc. (2006). Retrieved from <http://www.copper.org>

Deb, K., and Goel, T. (2001). "Controlled-elitist non-dominated sorting genetic algorithms for better convergence." *Proc. of the First International Conference on Evolutionary Multi-Criterion Optimization*, Zurich, Switzerland, 67-81.

DesRoches, R., McCormick C., and Delemont M. (2004). "Cyclic properties of superelastic shape memory alloy wires and bars." *Journal of Structural Engineering*, 130(1), 38-46.

Dicleli, M., and Mehta, A. (2006). "Effect of near-fault ground motion and damper characteristics on the seismic performance of chevron braced steel frames." *Earthquake Engineering and Structural Dynamics*, 36, 927-948.

Dolce M., Cardone D., and Marnetto R. (2000). "Implementation and testing of passive control devices based on shape memory alloys." *Earthquake Engineering and Structural Dynamics,* 29, 945-968.

Dolce M., and Cardone, D. (2001). "Mechanical behavior of shape memory alloys for seismic applications 2. Austenite NiTi bars subjected to tension." *International Journal of Mechanical Sciences*, 43, 2657-2677.

Dolce, M., Cardone D., Ponzo F. C., and Valente C. (2005). "Shaking table tests on reinforced concrete frames without and with passive control systems." *Earthquake Engineering and Structural Dynamics*, 34, 1687-1717.

Fosdick, R., and Ketema, Y. (1998). "Shape memory alloys for passive vibration damping." *Journal of Intelligent Material Systems and Structures*, 9, 854-870.

Fugazza, D. (2003). "Shape-memory alloy devices for earthquake engineering: mechanical properties, constitutive modeling and numerical simulations." Masters' Thesis, European School for Advanced Studies in Reduction of Seismic Risk, Pavia, Italy.

Funakubo, H. (1987). *Shape memory alloys,* ed. Kennedy, J. B., Amsterdam: Gordon and Breach Science, p. 141.

Gil, F. J., and Planell, J. A. (1998). "Shape memory alloys for medical applications." *Proc Inst Mech Eng [H,]* 212(6), 473-88.

Graesser, E. J., and Cozzarelli, F. A. (1991) "Shape-memory alloys as new materials for aseismic isolation." *Journal of Engineering Mechanics,* 117, 2590-2608.

Han, Y. H., Li, Q.S., Li, A.Q., Leung, A. Y. T., Lin, P.H. (2003). "Structural vibration control by shape memory alloy damper." *Earthquake Engineering and Structural Dynamics,* 32, 483-494.

Han, Y. L., Xing, D. J., Xiao E. T., and Li, A. Q. (2005). "NiTi-wire shape memory alloy dampers to simultaneously damp tension, compression, and torsion." *Journal of Vibration and Control*, 11(8), 1067-1084.

Janke, L., Czaderski, M., Motavalli, M., and Ruth, J. (2005). "Applications of shape memory alloys in civil engineering structures – Overview, limits and new ideas." *Materials and Structures*, 38, 578-592.

Kafka, V., and Vokoun, D. (2006). "Background of two characteristic features of shape memory phenomena." *Journal of Intelligent Material Systems and Structures*, 17, 511-520.

Kim, H. S., Roschke, P. N. (2006). "Design of fuzzy logic controller for smart base isolation system using genetic algorithms." *Engineering Structures*, 28(1), 84-96.

Krumme, R., Hayes, J., and Sweeney, S. (1995). "Structural damping with shape-memory alloys: one class of devices." *Proc. of SPIE Smart Structures and Materials: Passive Damping*, San Diego, CA, 2445, 225-240.

Lee, T. Y., and Kawashima, K. (2006). "Displacement response control of seismic-excited nonlinear isolated bridges." *Proc. of the 8th U.S. National Conference on Earthquake Engineering*, San Francisco, USA, Paper No. 419.

Liang, C. and Rogers, C. A. (1990). "One-dimensional thermo mechanical constitutive relations for shape memory material." *Journal of Intelligent Material Systems and Structures*, 1: 207–234.

Lin, R. (1996). *Shape memory alloys and their applications*. Retrieved from <http://www.stanford.edu/~richlin1/sma/sma.html>

Masuda, A., and Noori, M. (2002). "Optimization of hysteretic characteristics of damping devices based on pseudoelastic shape memory alloys." *International Journal of Non-Linear Mechanics,* 37, 1375–1386.

Math Works, Inc. (2007). *Fuzzy logic toolbox users' guide*., Nantick, MA.

Math Works, Inc. _(2007)_ *SIMULINK users' guide*., Nantick, MA.

Mavroeidis, G., Dong, G., and Papageorgiou, A. (2004). "Near-fault ground motions, and the response of elastic and inelastic single-degree-of-freedom (SDOF) systems." *Earthquake Engineering and Structural Dynamics*, 33 (9), 1023-1049.

McCormick, J. P. (2006). "Cyclic behavior of shape memory alloys: material characterization and optimization." Ph.D. Dissertation. School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, GA.

Morgan, N. B. (2004). "Medical shape memory alloy applications—the market and its products." *Material Science and Engineering*, A 378, 16-23.

Nemat-Nasser S., Guo, W. G. (2006). "Superelastic and cyclic response of NiTi SMA at various strain rates and temperatures." *Mechanics of Materials*, 38, 463-474.

Nimesis Inteligent Materials. (2006). Retrieved from <http://www.nimesis.com>

Otsuka, K., and Ren, X. (2005). "Physical metallurgy of Ti-Ni based shape memory alloys." *Progress in Material Science,* 50, 511-678.

Penar, B. W. (2005). "Recentering beam-column connections using shape memory alloys." Masters Thesis. School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, GA.

Piedboeuf, M. C., Gauvin, R. and Thomas, M. (1998). "Damping behavior of shape memory alloys: Strain amplitude, frequency, and temperature effects." *Journal of Sound and Vibration*. 214(5), 885-901.

Prahlad, H., and Chopra, I. (2001). "Comparative evaluation of shape memory alloy constitutive models with experimental data." *Journal of Intelligent Material Systems and Structures,* 12, 383-395.

Saadat, S., Noori, M., Davoodi, H., Hou, Z., Suziki, Y., and Masuda, A. (2001). "Using NiTi SMA tendons for vibration control of coastal structures*." Smart Material and Structures*, 10, 695-704.

Salich, J., Hou. Z., and Noori M. (2001). "Vibration suppression of structures using passive shape memory alloy energy dissipation devices." *Journal of Intelligent Material Systems and Structures,* 12, 671-680.

Sawaguchi, T., Kikuchi, T., and Kajiwara, S. (2005). "The pseudoelastic behavior of Fe-Mn-Si based shape memory alloys containing Nb and C." *Smart Materials and Structures*, 14, 317-322.

Sawaguchi, T., Kikuchi, T., Ogawa, K., Yin F. X., Kajiwara, S., Kushibe, A., and Ogawa, T. (2006). "Internal friction of Fe-Mn-Si-based shape memory alloys containing Nb and C and their application as a seismic damping material." *Key Engineering Materials*, 319, 53-57.

Seelecke, S., Heintze, O., and Masuda, A. (2002). "Simulation of earthquake-induced structural vibrations in systems with SMA damping elements." *Proc. of SPIE Smart Structures and Materials*, 4697, 238-245.

SMART Laboratory at Texas A&M University. (2006). Retrieved from <http://smart.tamu.edu>

Smart Material Products. (2006). Retrieved from <http://www.smarterial.com>

Song, G., Ma, N., and Li, H. N. (2004). "Review of applications of shape memory alloys in civil structures." *Earth and Space,* 559-566.

Soong, T. T., and Dargush G. F. (1997). *Passive energy dissipation systems in structural engineering*. John Wiley & Sons, New York, NY.

Special Metals Corporation. (2006). Retrieved from <http://www.specialmetals.com>

Special Metals – Shape Memory Alloy Division. (2006). Retrieved from <http://www.shape-memory-alloys.com>

Spencer, B. F., and Nagarajaiah, S. (2003). "State of the art of structural control." *Journal of Structural Engineering*, 129(7), 845-856.

Strandel, B., Ohashi, S., Ohtsuka, H., Ishihara, T., and Miyazaki S. (1995). "Cyclic stress-strain characteristics of Ti-Ni and Ti-Ni-Cu shape memory alloys." *Material Science and Engineering*, A202, 148-156.

Sugeno, M. (1985). *Industrial applications of fuzzy control.* Elsevier Science Pub. Co., Amsterdam.

Sun, S., and Rajapakse R. K. N. D. (2003). "Simulation of pseudoelastic behavior of SMA under cyclic loading." *Computational Material Science*, 28, 663-674.

Tamai, H., and Kitagawa, Y. (2002). "Pseudoelastic behavior of shape memory alloy wire and its application to seismic resistance member for building." *Computational Material Science*. 25, 218-227.

Tanaka, K. (1986). "A thermomechanical sketch of shape memory effect: one-dimensional tensile behavior." *Res. Mechanica*, 18, 251–263.

Thomson, P., Balas, G. J., and Leo, P. H. (1997). "Analysis of trigger line models for shape memory hysteresis based on dynamic testing." *Journal of Intelligent Material Systems and Structures*, 8,193-201.

Tobushi, H., Yamada, S., Hachisuka, T., Ikai, A. and Tanaka, K. (1996). "Thermomechanical properties due to martensitic and R-phase transformations of TiNi shape memory alloy subjected to cyclic loadings." *Journal of Smart Materials and Structures*, 5: 788–795.

Tobushi, H., Shimeno, Y., Hachisuka, T., and Tanaka, K. (1998). "Influence of strain rate on superelastic properties of TiNi shape memory alloy." *Mechanics of Materials*, 30(2), 141–150.

Torra, V., Isalgue, A., Lovey, F. C., Martorell, F., Molina, F. J., Sade, M., and Tachorie, H. (2004). "Shape memory alloys: from the physical properties of metastable phase

transitions to dampers for civil engineering applications." *Journal of Physics IV,* 113, 85-90.

Wen, Y. K. (1976). "Method for random vibration of hysteretic system." *Journal of Engineering Mechanics Division*, *ASCE,* 102(EM2), 249-263.

Wilde, K., Gardoni, P., and Fujino, Y. (2000). "Base isolation system with shape memory alloy device for elevated highway bridges." *Engineering Structures*, 22, 222-229.

Wilson, J. C., and Wesolowsky, M. J. (2005) "Shape memory alloys for seismic response modification: A state-of-the art review." *Earthquake Spectra,* 21(2), 569-601.

Wolons, D., Gandhi, F., and Malovrah, B. (1998). "Experimental investigation of the pseudoelastic hysteresis damping characteristics of shape memory alloy wires." *Journal of Intelligent Material Systems and Structures*, 9, 116-126.

Wu., K., Yang, F., Pu, Z., and Shi, J. (1996). "The effect of strain rate on detwinning and superelastic behavior of NiTi shape memory alloys." *Journal of Intelligent Material Systems and Structures,* 7, 138-144.

# APPENDIX A

# MATLAB SCRIPT FILES

This appendix presents the Simulink programs used for all numerical simulations. Main programs (systems) are named as S#, while subsystems are identified with S#.#, etc.

The list of the programs is given below.

**Section 5:**

S1. sim_sma_valid_temp

S2. sim_sma_valid_vel

S3. SDOFfree_all

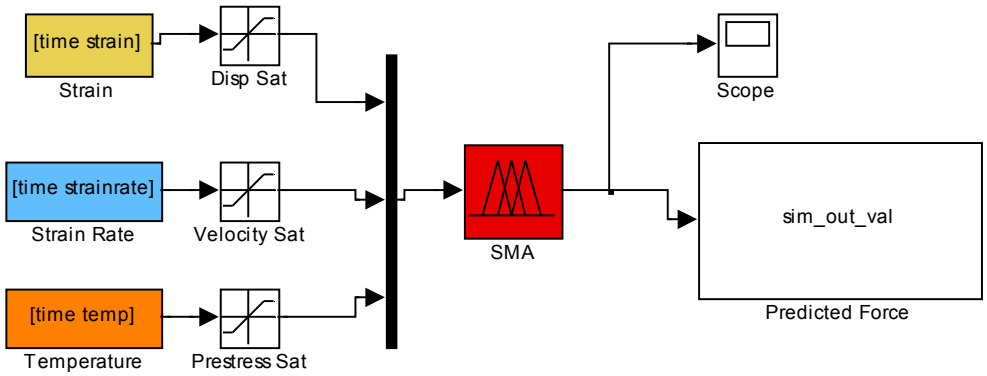
S4. SDOFsine_all

S5. SDOFearth_all

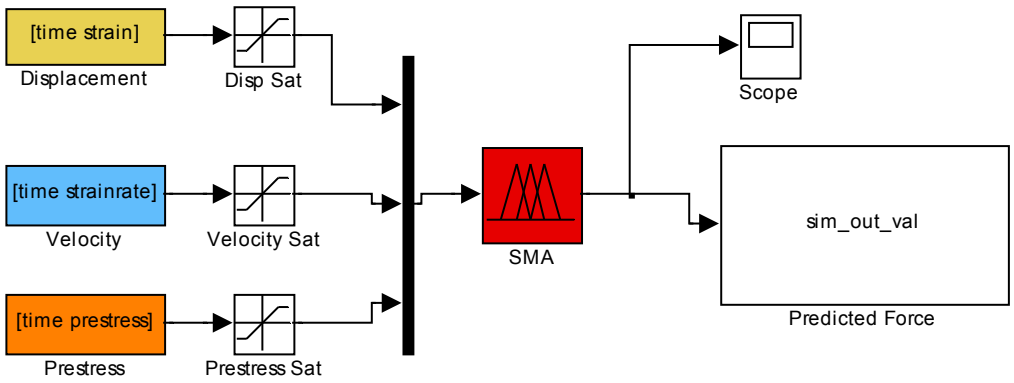
S6. sma3storeyEQsim2

S7. sma2DOFbridge_final

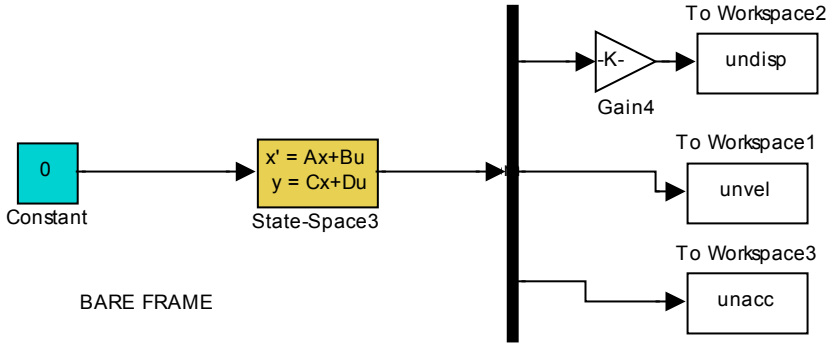
## S1

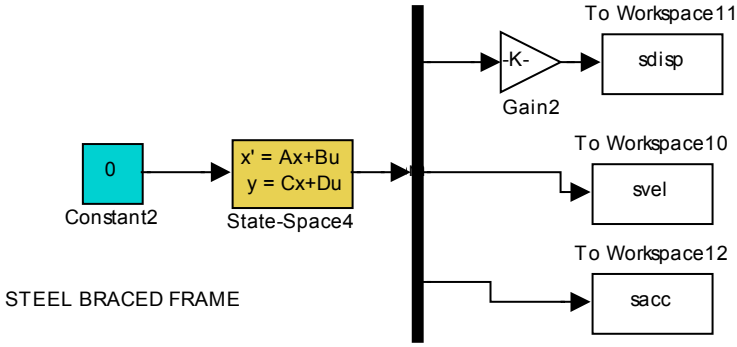

[time strain]
Strain
Disp Sat
[time strainrate]
Strain Rate
Velocity Sat
[time temp]
Temperature
Prestress Sat
SMA
Scope
sim_out_val
Predicted Force

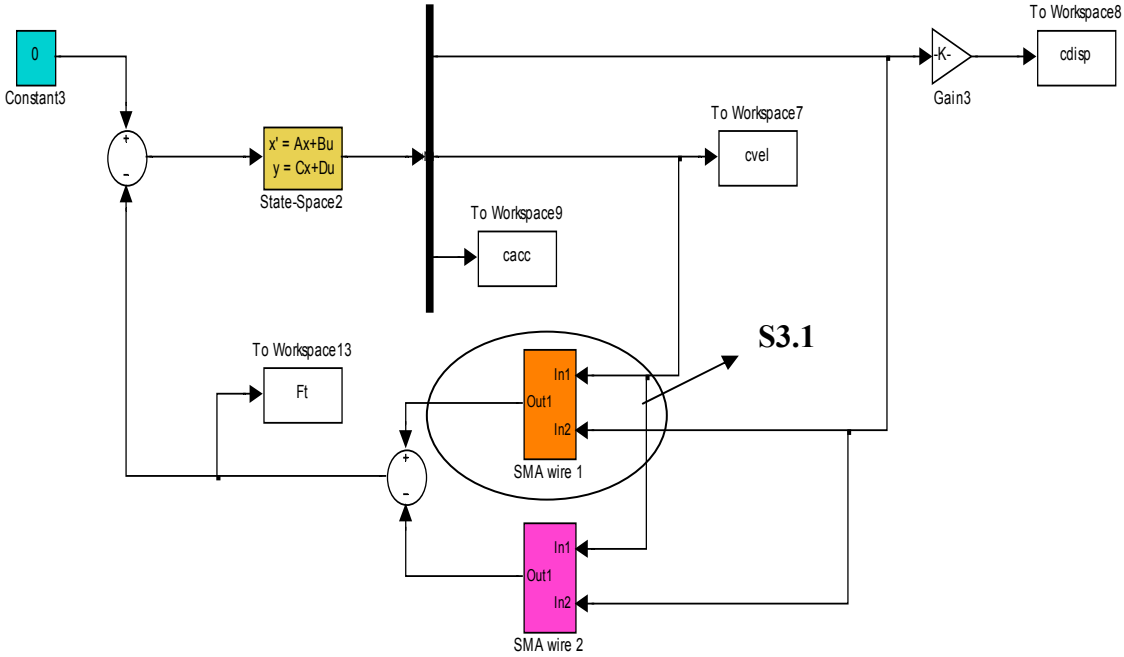
## S2


[time strain]
Displacement
Disp Sat
[time strainrate]
Velocity
Velocity Sat
[time prestress]
Prestress
Prestress Sat
SMA
Scope
sim_out_val
Predicted Force

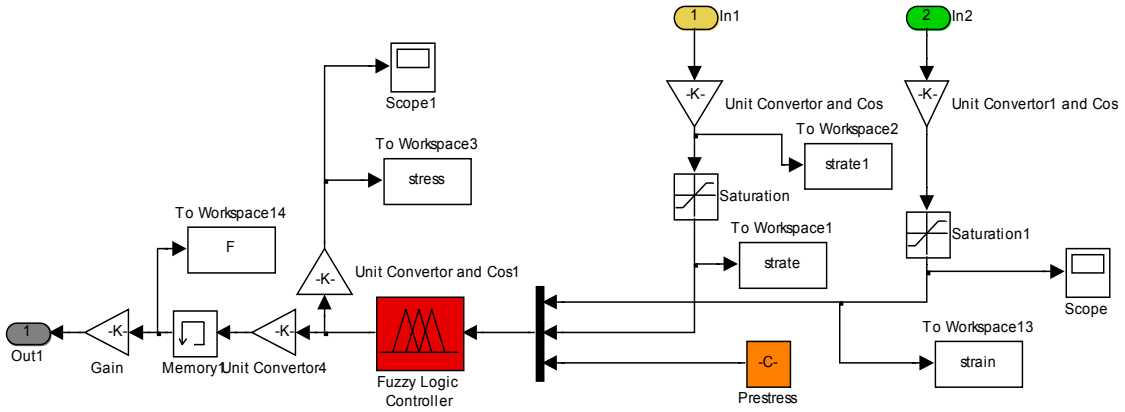
# S3


To Workspace2
-K-
undisp
Gain4
0
Constant
x' = Ax+Bu
y = Cx+Du
State-Space3
To Workspace1
unvel
To Workspace3
unacc
BARE FRAME

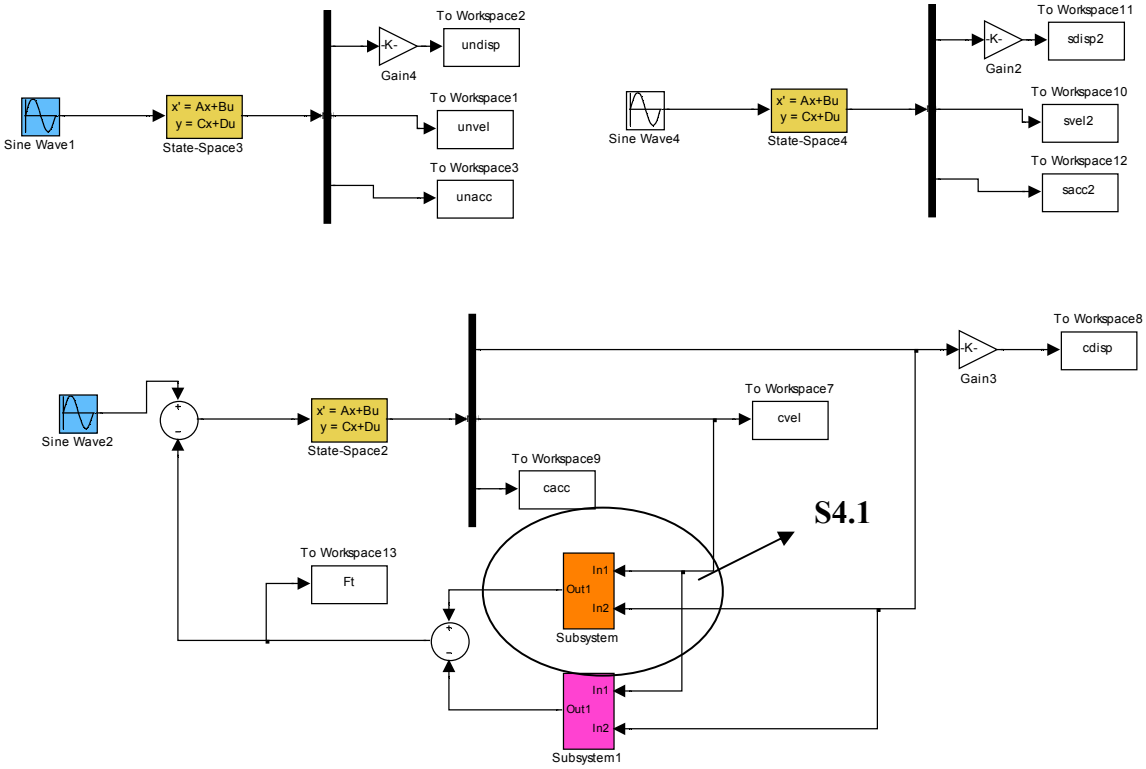

To Workspace11
-K-
sdisp
Gain2
0
Constant2
x' = Ax+Bu
y = Cx+Du
State-Space4
To Workspace10
svel
To Workspace12
sacc
STEEL BRACED FRAME


0
Constant3
To Workspace8
-K-
cdisp
Gain3
To Workspace7
cvel
x' = Ax+Bu
y = Cx+Du
State-Space2
To Workspace9
cacc
S3.1
To Workspace13
Ft
In1
Out1
In2
SMA wire 1
In1
Out1
In2
SMA wire 2

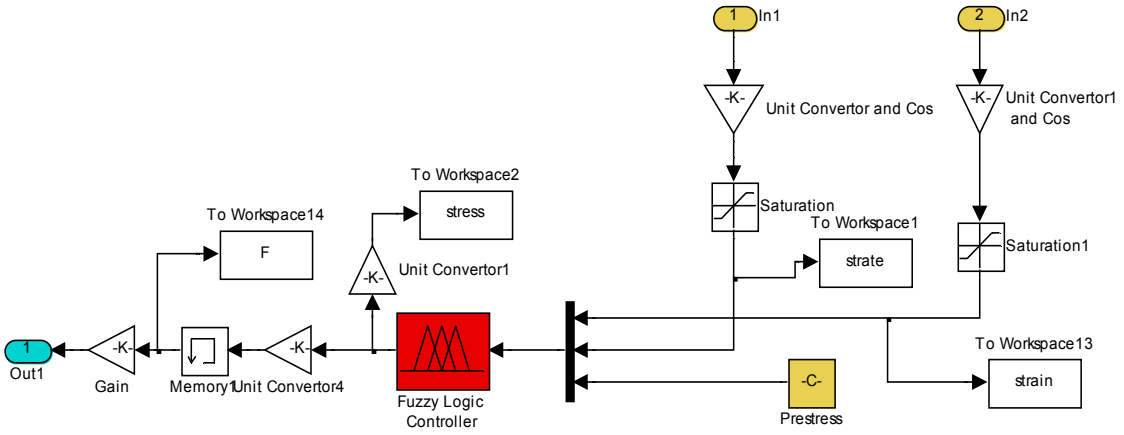
# S3.1.


1
In1
2
In2
Scope1
-K-
Unit Convertor and Cos
-K-
Unit Convertor1 and Cos
To Workspace2
strate1
To Workspace3
stress
Saturation
To Workspace14
F
To Workspace1
strate
Saturation1
-K-
Unit Convertor and Cos1
Scope
1
Out1
-K-
Gain
Memory1
-K-
Unit Convertor4
Fuzzy Logic
Controller
To Workspace13
strain
-C-
Prestress

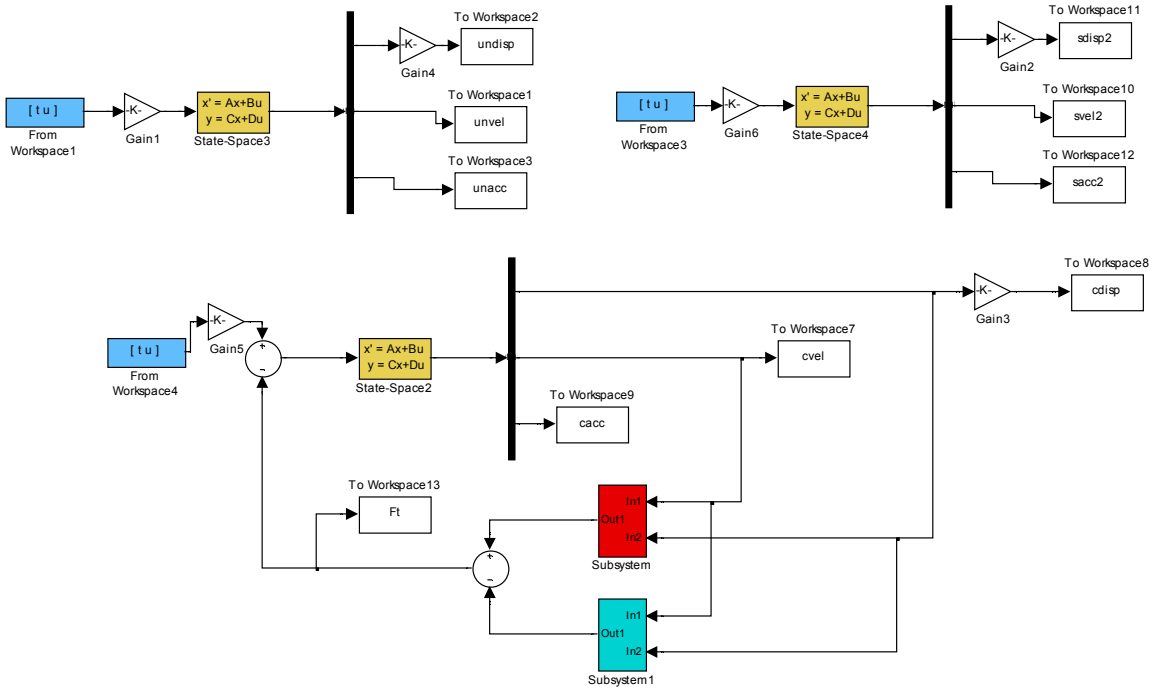
# S4


To Workspace2
-K-
undisp
Gain4
Sine Wave1
x' = Ax+Bu
y = Cx+Du
State-Space3
To Workspace1
unvel
To Workspace3
unacc
To Workspace11
-K-
sdisp2
Gain2
Sine Wave4
x' = Ax+Bu
y = Cx+Du
State-Space4
To Workspace10
svel2
To Workspace12
sacc2
To Workspace8
-K-
cdisp
Gain3
Sine Wave2
x' = Ax+Bu
y = Cx+Du
State-Space2
To Workspace7
cvel
To Workspace9
cacc
S4.1
To Workspace13
Ft
In1
Out1
In2
Subsystem
In1
Out1
In2
Subsystem1

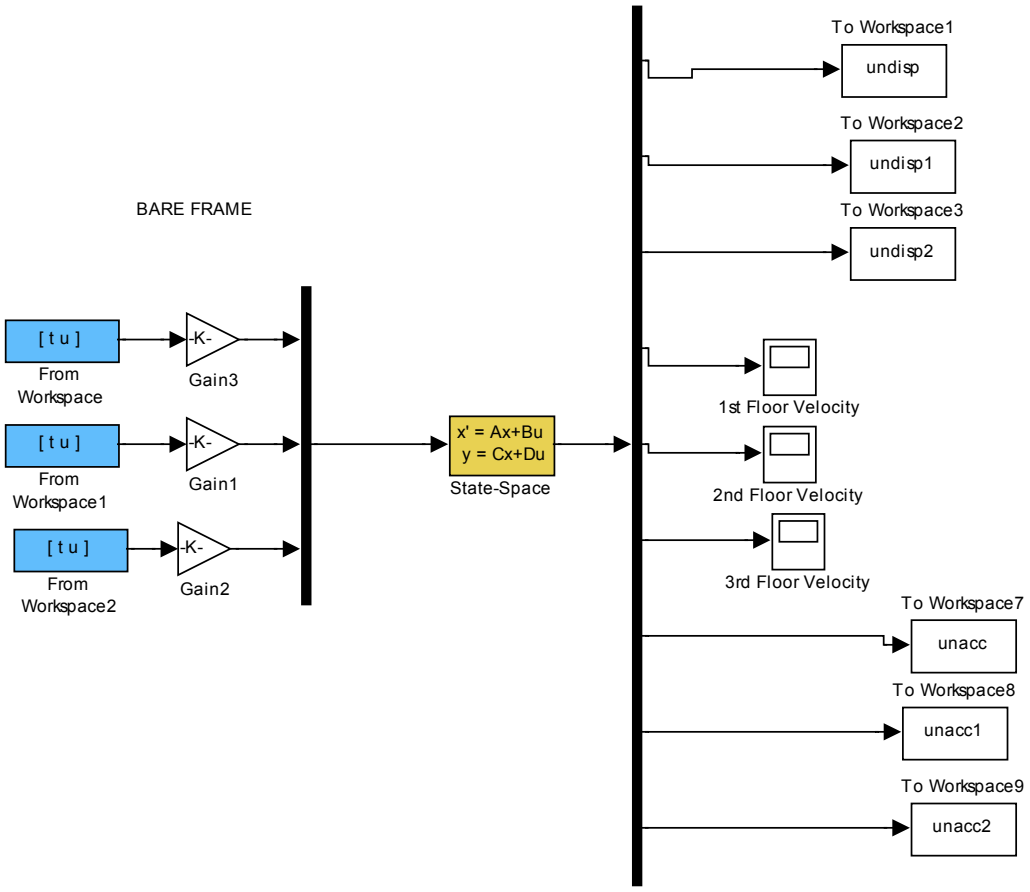
# S4.1

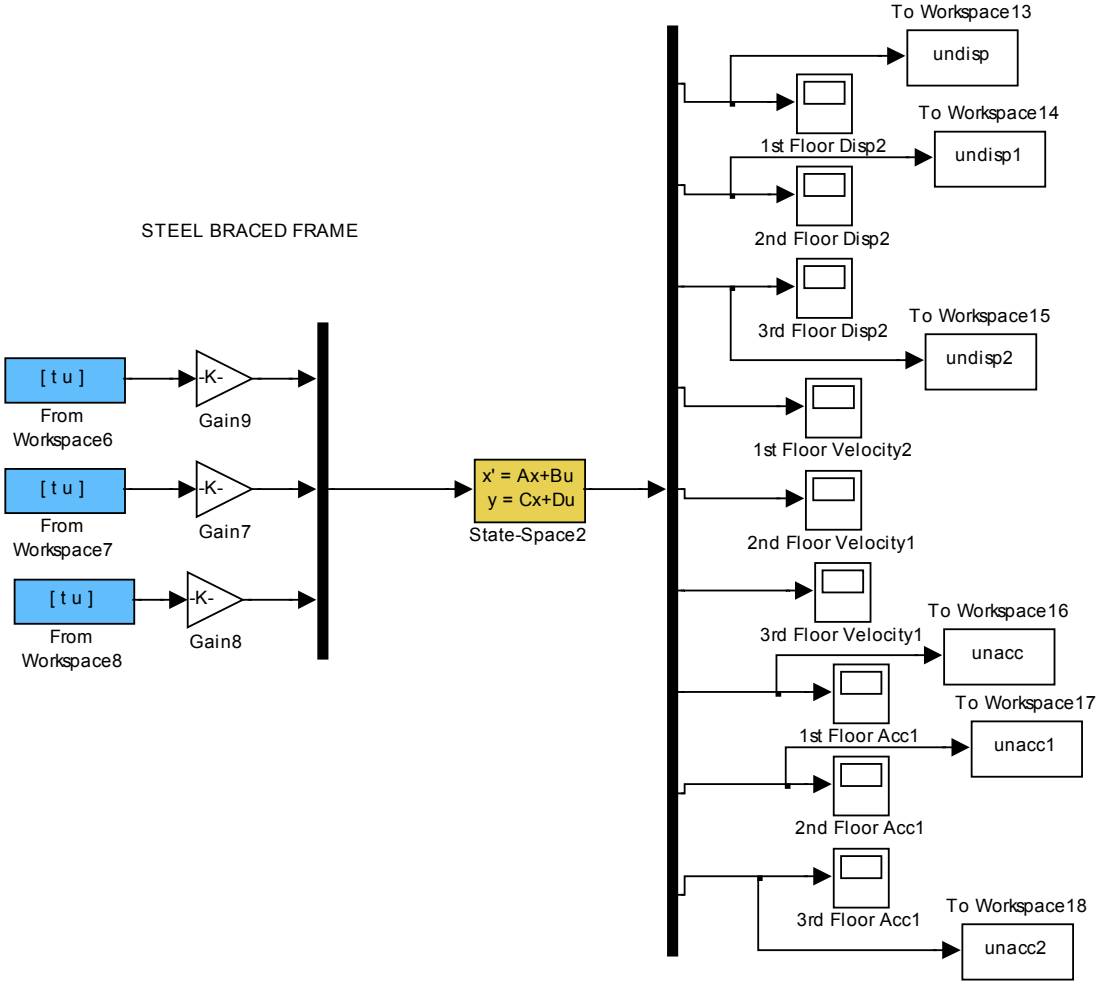

1
In1
2
In2
-K-
Unit Convertor and Cos
-K-
Unit Convertor1
and Cos
To Workspace2
stress
Saturation
To Workspace1
strate
Saturation1
To Workspace14
F
-K-
Unit Convertor1
1
Out1
-K-
Gain
Memory1
-K-
Unit Convertor4
Fuzzy Logic
Controller
-C-
Prestress
To Workspace13
strain

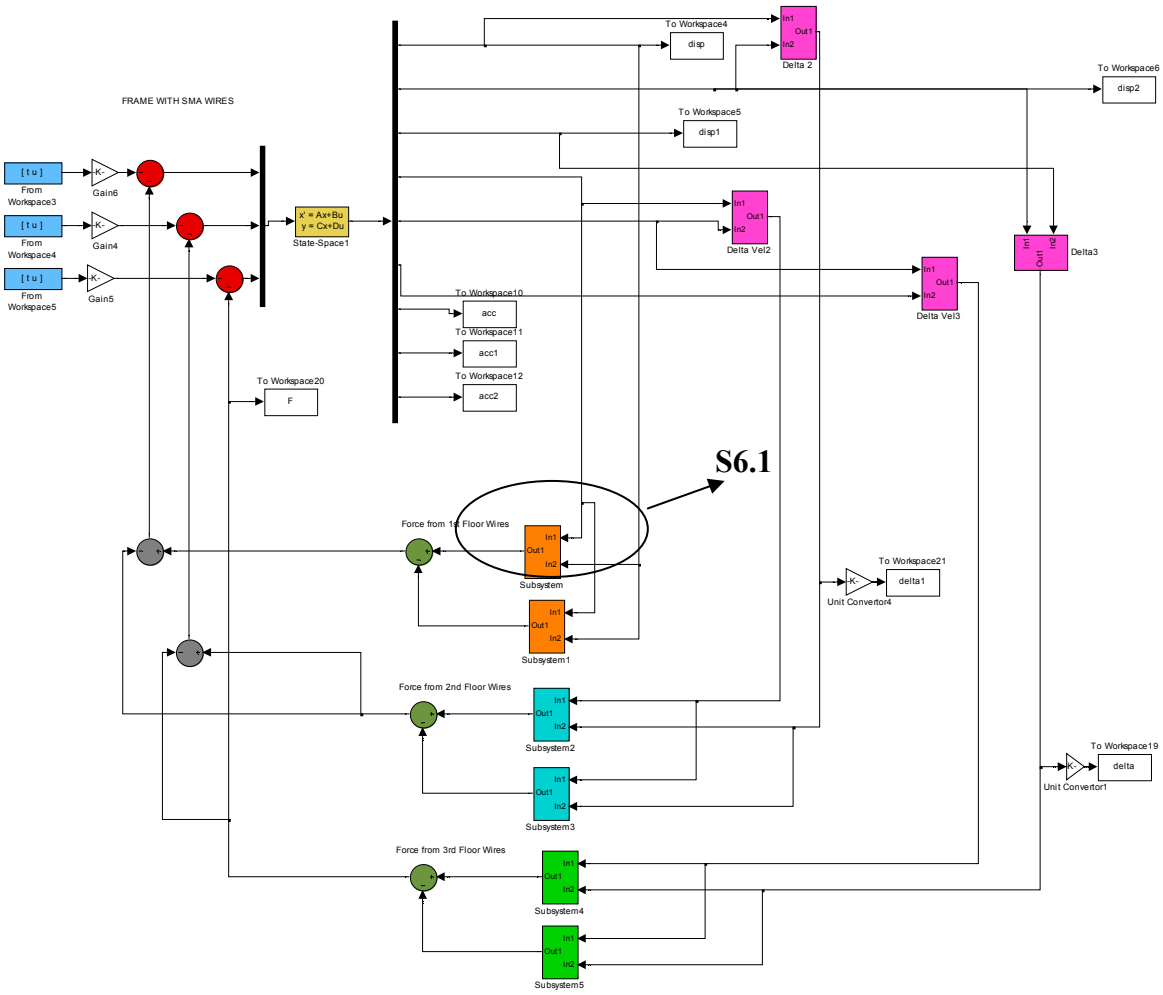
# S5


[ t u ]
From
Workspace1
-K-
Gain1
x' = Ax+Bu
y = Cx+Du
State-Space3
-K-
Gain4
To Workspace2
undisp
To Workspace1
unvel
To Workspace3
unacc
[ t u ]
From
Workspace3
-K-
Gain6
x' = Ax+Bu
y = Cx+Du
State-Space4
-K-
Gain2
To Workspace11
sdisp2
To Workspace10
svel2
To Workspace12
sacc2
[ t u ]
From
Workspace4
-K-
Gain5
x' = Ax+Bu
y = Cx+Du
State-Space2
-K-
Gain3
To Workspace8
cdisp
To Workspace7
cvel
To Workspace9
cacc
To Workspace13
Ft
In1
Out1
In2
Subsystem
In1
Out1
In2
Subsystem1

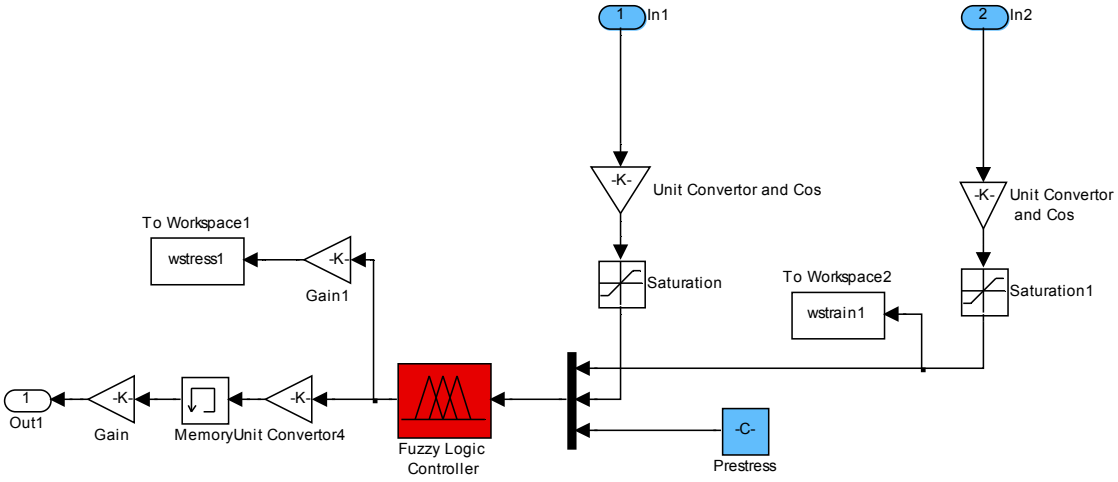
# S6


BARE FRAME
[ t u ]
From Workspace
-K-
Gain3
[ t u ]
From Workspace1
-K-
Gain1
[ t u ]
From Workspace2
-K-
Gain2
x' = Ax+Bu
y = Cx+Du
State-Space
To Workspace1
undisp
To Workspace2
undisp1
To Workspace3
undisp2
1st Floor Velocity
2nd Floor Velocity
3rd Floor Velocity
To Workspace7
unacc
To Workspace8
unacc1
To Workspace9
unacc2

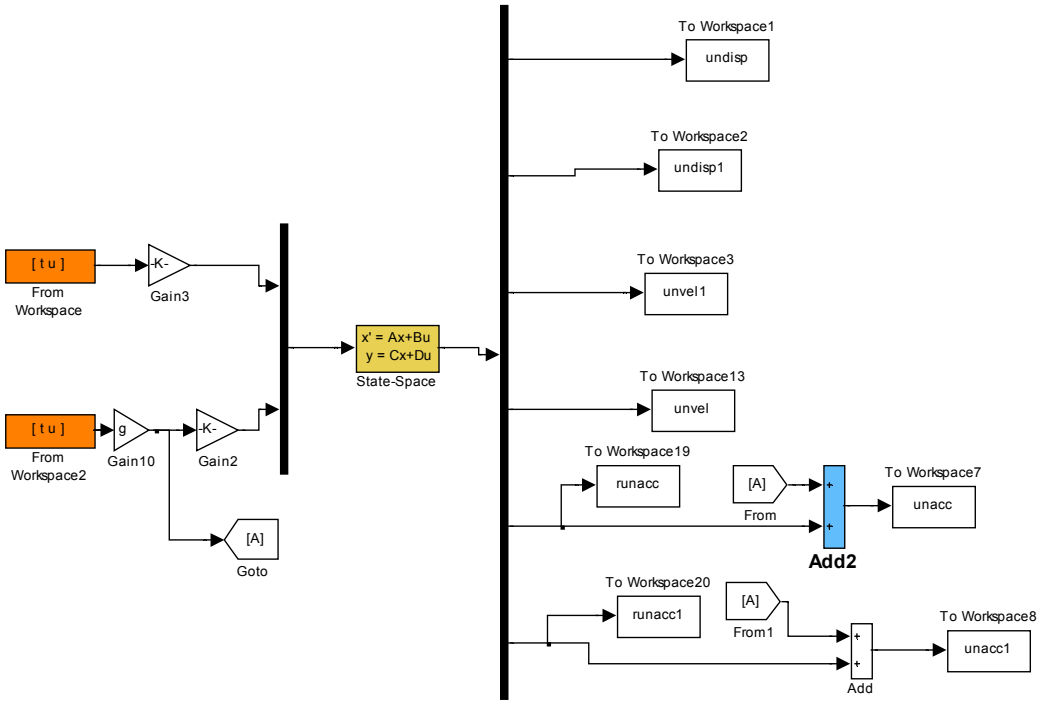

STEEL BRACED FRAME
[ t u ]
From Workspace6
-K-
Gain9
[ t u ]
From Workspace7
-K-
Gain7
[ t u ]
From Workspace8
-K-
Gain8
x' = Ax+Bu
y = Cx+Du
State-Space2
To Workspace13
undisp
1st Floor Disp2
To Workspace14
undisp1
2nd Floor Disp2
3rd Floor Disp2
To Workspace15
undisp2
1st Floor Velocity2
2nd Floor Velocity1
3rd Floor Velocity1
To Workspace16
unacc
1st Floor Acc1
To Workspace17
unacc1
2nd Floor Acc1
3rd Floor Acc1
To Workspace18
unacc2

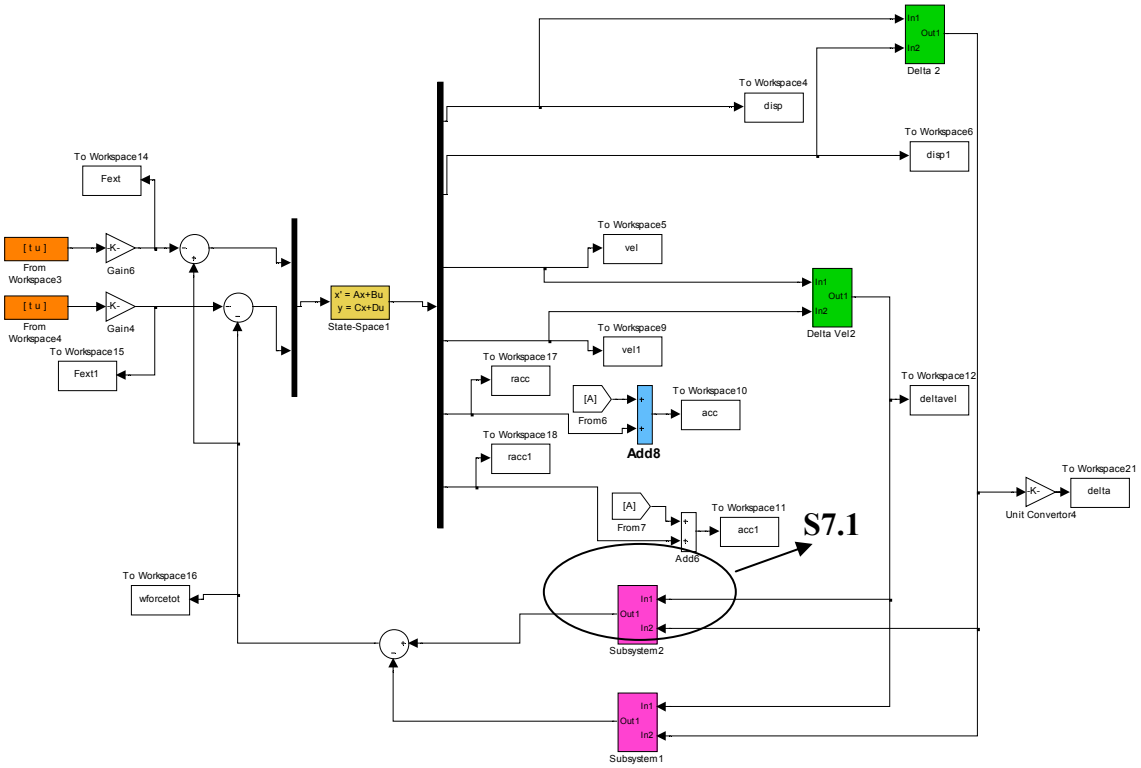

FRAME WITH SMA WIRES
[ t u ]
From Workspace3
[ t u ]
From Workspace4
[ t u ]
From Workspace5
K-
Gain6
K-
Gain4
K-
Gain5
x' = Ax+Bu
y = Cx+Du
State-Space1
To Workspace4
disp
In1
Out1
In2
Delta 2
To Workspace6
disp2
To Workspace5
disp1
In1
Out1
In2
Delta Vel2
In1
Out1
In2
Delta Vel3
In1
Out1
In2
Delta3
To Workspace10
acc
To Workspace11
acc1
To Workspace12
acc2
To Workspace20
F
S6.1
Force from 1st Floor Wires
Subsystem
Subsystem1
To Workspace21
delta1
Unit Convertor4
Force from 2nd Floor Wires
Subsystem2
Subsystem3
To Workspace19
delta
Unit Convertor1
Force from 3rd Floor Wires
Subsystem4
Subsystem5

**S6.1**

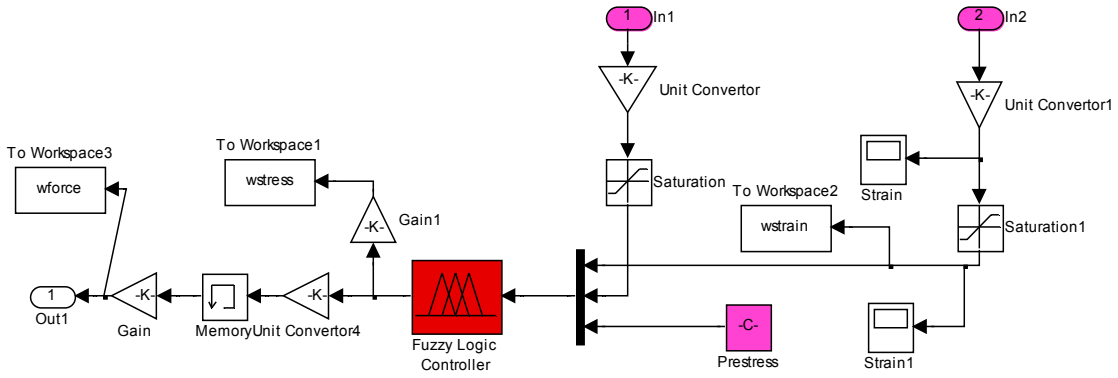

1
In1
2
In2
-K-
Unit Convertor and Cos
-K-
Unit Convertor and Cos
To Workspace1
wstress1
-K-
Gain1
Saturation
To Workspace2
wstrain1
Saturation1
1
Out1
-K-
Gain
MemoryUnit Convertor4
-K-
Fuzzy Logic Controller
-C-
Prestress

**S7**


To Workspace1
undisp
To Workspace2
undisp1
To Workspace3
unvel1
To Workspace13
unvel
[ t u ]
From Workspace
-K-
Gain3
x' = Ax+Bu
y = Cx+Du
State-Space
[ t u ]
From Workspace2
g
Gain10
-K-
Gain2
[A]
Goto
To Workspace19
runacc
[A]
From
To Workspace7
unacc
Add2
To Workspace20
runacc1
[A]
From1
To Workspace8
unacc1
Add


In1
Out1
In2
Delta 2
To Workspace4
disp
To Workspace6
disp1
To Workspace14
Fext
[ t u ]
From Workspace3
K-
Gain6
[ t u ]
From Workspace4
K-
Gain4
To Workspace15
Fext1
x' = Ax+Bu
y = Cx+Du
State-Space1
To Workspace5
vel
In1
Out1
In2
Delta Vel2
To Workspace9
vel1
To Workspace17
racc
[A]
From6
To Workspace10
acc
Add8
To Workspace18
racc1
To Workspace12
deltavel
To Workspace21
K-
Unit Convertor4
delta
[A]
From7
To Workspace11
acc1
Add6
S7.1
To Workspace16
wforcetot
In1
Out1
In2
Subsystem2
In1
Out1
In2
Subsystem1

**S7.1**


1
In1
-K-
Unit Convertor
2
In2
-K-
Unit Convertor1
To Workspace3
wforce
To Workspace1
wstress
-K-
Gain1
Saturation
Strain
To Workspace2
wstrain
Saturation1
1
Out1
-K-
Gain
MemoryUnit Convertor4
-K-
Fuzzy Logic Controller
-C-
Prestress
Strain1

# APPENDIX B

# MATLAB SCRIPT FILES

This appendix presents the Matlab script files that are used for creating neuro-fuzzy model of CuAlBe shape memory alloys in section 5, and for implementing time history analyses of civil engineering structures with SMA elements in section 6.

The contents of this appendix are summarized below.

**Section 5:**

M1. BuildTrainCheckValidTemperaturePequena.m

M2. smaTrainTemp.m

M3. smaValidTemp.m

M4. BuildTrainCheckValidDatamanyWires.m

M5. smaTrainVel2.m

M6. smaValidVel2.m

**Section 6:.**

M7. SDOFfree_allscript.m

M8. SDOF_sine_all_Vfreqscript.m

M9. SDOFearth_allscript.m

M10. sma3storeyEQscript2st.m

M11. bridge2DOF_ILL.m

M12. solver_bridge2DOF_ILL.m

M13. sma2DOFbridgescript_Final.m

M14. nonlinear.m

M15. CAA.m

M16. nonlinearSMA.m

M17. CAAsma.m

M18. nonlinearSteel2B.m

M19. CAAsteel2B.m

M20. runopt_gauss.m

# SECTION 5

## M1

```
%BuildTrainCheckValidTemperaturePequena.m
clear all
close all

addpath 'C:\Documents and Settings\oeo2452\My Documents\Shape Memory
Alloys\Mir\Registros\Registros Ciclicos\0_8def'
addpath 'C:\Documents and Settings\oeo2452\My Documents\Shape Memory
Alloys\Mir\Registros\Registros Ciclicos\1_5def'
addpath 'C:\Documents and Settings\oeo2452\My Documents\Shape Memory
Alloys\Mir\Registros\Registros Ciclicos\2_2def'


% MEDIO - SEGUNDA
% %0.8 strain
load P17X
load P17R
load P17U
% %1.5 strain
load P18Q
load P18L
load P18O
% %2.2 strain
load P19P
load P19L
load P19N

% MEDIO - PRIMERA
% %0.8 strain
load P17O
load P17J
load P17L
% %1.5 strain
load P18J
load P18E
load P18H
% %2.2 strain
load P19J
load P19E
load P19H

% Specify 'dt' for the data time step (units are sec.)
dt = 0.005;
tc_or_valid = 0;

% Set up coefficients to calculate strain, strain rate and stress
wirelength = 25.4 ; %mm
wirearea = 1.963*10^-7; %m2

coef_s = 10^-2 ;
coef_sr = 10^-2 ;
coef_stress = 10^6/9.806 ;
```

```
length_strain = wirelength*coef_s ;
length_strate = wirelength*coef_sr ;
area = wirearea*coef_stress;

[b,a] = butter(2,10*dt*2);

% Define starting and stop data point
ss = 1:11040;
ss1 = 1:5040;
%%%%%0.8%%%%%
%MEDIO - SEGUNDO
% Set up displacement array
disp1 = [P17O(ss1,2)*0.711];
force1 = [P17O(ss1,1)*62.208];
temp1 = [P17O(ss1,3)*100];

disp1 = filter(b,a,disp1);
force1 = filter(b,a,force1);
temp1 = filter(b,a,temp1);
% Calculate velocity
vel1 = vel_calc(disp1, dt, tc_or_valid);

for n=1:20
    disp1(n)=[];
    vel1(n) =[];
    force1(n)=[];
    temp1(n)=[];
end
for n=5020:-1:5001
    disp1(n)=[];
    vel1(n)=[];
    force1(n)=[];
    temp1(n)=[];
end

disp2 = [P17J(ss1,2)*0.711];
force2 = [P17J(ss1,1)*62.208];
temp2 = [P17J(ss1,3)*100];

disp2 = filter(b,a,disp2);
force2 = filter(b,a,force2);
temp2 = filter(b,a,temp2);
% Calculate velocity
vel2 = vel_calc(disp2, dt, tc_or_valid);

for n=1:20
    disp2(n)=[];
    vel2(n) =[];
    force2(n)=[];
    temp2(n)=[];
```

```
end
for n=5020:-1:5001
    disp2(n)=[];
    vel2(n)=[];
    force2(n)=[];
    temp2(n)=[];
end

disp3 = [P17L(ss1,2)*0.711];
force3 = [P17L(ss1,1)*62.208];
temp3 = [P17L(ss1,3)*100];

disp3 = filter(b,a,disp3);
force3 = filter(b,a,force3);
temp3 = filter(b,a,temp3);
% Calculate velocity
vel3 = vel_calc(disp3, dt, tc_or_valid);
for n=1:20
    disp3(n)=[];
    vel3(n) =[];
    force3(n)=[];
    temp3(n)=[];
end
for n=5020:-1:5001
    disp3(n)=[];
    vel3(n)=[];
    force3(n)=[];
    temp3(n)=[];
end

% MEDIO - PRIMERA
disp4 = [P17X(ss1,2)*0.711];
force4 = [P17X(ss1,1)*62.208];
temp4 = [P17X(ss1,3)*100];

disp4 = filter(b,a,disp4);
force4 = filter(b,a,force4);
temp4 = filter(b,a,temp4);
% Calculate velocity
vel4 = vel_calc(disp4, dt, tc_or_valid);
for n=1:20
    disp4(n)=[];
    vel4(n) =[];
    force4(n)=[];
    temp4(n)=[];
end
for n=5020:-1:5001
    disp4(n)=[];
    vel4(n)=[];
    force4(n)=[];
    temp4(n)=[];
end
```

```
% disp5 = [P17R(ss1,2)*0.711];
% force5 = [P17R(ss1,1)*62.208];
% temp5 = [(P17R(ss1,3)*100-25)];
%
%
% disp5 = filter(b,a,disp5);
% force5 = filter(b,a,force5);
% temp5 = filter(b,a,temp5);
% % Calculate velocity
% vel5 = vel_calc(disp5, dt, tc_or_valid);
% for n=1:20
%     disp5(n)=[];
%     vel5(n) =[];
%     force5(n)=[];
%     temp5(n)=[];
% end
% for n=5020:-1:5001
%     disp5(n)=[];
%     vel5(n)=[];
%     force5(n)=[];
%     temp5(n)=[];
% end

disp6 = [P17U(ss1,2)*0.711];
force6 = [P17U(ss1,1)*62.208];
temp6 = [P17U(ss1,3)*100];

disp6 = filter(b,a,disp6);
force6 = filter(b,a,force6);
temp6 = filter(b,a,temp6);
% Calculate velocity
vel6 = vel_calc(disp6, dt, tc_or_valid);
for n=1:20
    disp6(n)=[];
    vel6(n) =[];
    force6(n)=[];
    temp6(n)=[];
end
for n=5020:-1:5001
    disp6(n)=[];
    vel6(n)=[];
    force6(n)=[];
    temp6(n)=[];
end


%%%%1.5%%%%%%
%MEDIO - SEGUNDO
disp7 = [P18J(ss,2)*0.711];
force7 = [P18J(ss,1)*62.208];
temp7 = [P18J(ss,3)*100];
```

```
disp7 = filter(b,a,disp7);
force7 = filter(b,a,force7);
temp7 = filter(b,a,temp7);
%Calculate velocity
vel7 = vel_calc(disp7, dt, tc_or_valid);
for n=1:20
    disp7(n)=[];
    vel7(n) =[];
    force7(n)=[];
    temp7(n)=[];
end
for n=11020:-1:11001
    disp7(n)=[];
    vel7(n)=[];
    force7(n)=[];
    temp7(n)=[];
end

disp8 = [P18E(ss,2)*0.711];
force8 = [P18E(ss,1)*62.208];
temp8 = [P18E(ss,3)*100];

disp8 = filter(b,a,disp8);
force8 = filter(b,a,force8);
temp8 = filter(b,a,temp8);
%Calculate velocity
vel8 = vel_calc(disp8, dt, tc_or_valid);
for n=1:20
    disp8(n)=[];
    vel8(n) =[];
    force8(n)=[];
    temp8(n)=[];
end
for n=11020:-1:11001
    disp8(n)=[];
    vel8(n)=[];
    force8(n)=[];
    temp8(n)=[];
end

disp9 = [P18H(ss,2)*0.711];
force9 = [P18H(ss,1)*62.208];
temp9 = [P18H(ss,3)*100];


disp9 = filter(b,a,disp9);
force9 = filter(b,a,force9);
temp9 = filter(b,a,temp9);
%Calculate velocity
vel9 = vel_calc(disp9, dt, tc_or_valid);
for n=1:20
```

```
    disp9(n)=[];
    vel9(n) =[];
    force9(n)=[];
    temp9(n)=[];
end
for n=11020:-1:11001
    disp9(n)=[];
    vel9(n)=[];
    force9(n)=[];
    temp9(n)=[];
end

% % MEDIO - PRIMERA
% disp10 = [P18Q(ss,2)*0.711];
% force10 = [P18Q(ss,1)*62.208];
% temp10 = [P18Q(ss,3)*100];
%
% disp10 = filter(b,a,disp10);
% force10 = filter(b,a,force10);
% temp10 = filter(b,a,temp10);
% % Calculate velocity
% vel10 = vel_calc(disp10, dt, tc_or_valid);
% for n=1:20
%     disp10(n)=[];
%     vel10(n) =[];
%     force10(n)=[];
%     temp10(n)=[];
% end
% for n=11020:-1:11001
%     disp10(n)=[];
%     vel10(n)=[];
%     force10(n)=[];
%     temp10(n)=[];
% end


disp11 = [P18L(ss,2)*0.711];
force11 = [P18L(ss,1)*62.208];
temp11 = [P18L(ss,3)*100];

disp11 = filter(b,a,disp11);
force11 = filter(b,a,force11);
temp11 = filter(b,a,temp11);
% Calculate velocity
vel11 = vel_calc(disp11, dt, tc_or_valid);
for n=1:20
    disp11(n)=[];
    vel11(n) =[];
    force11(n)=[];
    temp11(n)=[];
end
for n=11020:-1:11001
```

```
    disp11(n)=[];
    vel11(n)=[];
    force11(n)=[];
    temp11(n)=[];
end

disp12 = [P18O(ss,2)*0.711];
force12 = [P18O(ss,1)*62.208];
temp12 = [P18O(ss,3)*100];


disp12 = filter(b,a,disp12);
force12 = filter(b,a,force12);
temp12 = filter(b,a,temp12);
%Calculate velocity
vel12 = vel_calc(disp12, dt, tc_or_valid);
for n=1:20
    disp12(n)=[];
    vel12(n) =[];
    force12(n)=[];
    temp12(n)=[];
end
for n=11020:-1:11001
    disp12(n)=[];
    vel12(n)=[];
    force12(n)=[];
    temp12(n)=[];
end

%%%%%2.2%%%%%%
% MEDIO - SEGUNDO
disp13 = [P19J(ss,2)*0.711];
force13 = [P19J(ss,1)*62.208];
temp13 = [P19J(ss,3)*100];


disp13 = filter(b,a,disp13);
force13 = filter(b,a,force13);
temp13 = filter(b,a,temp13);
% Calculate velocity
vel13 = vel_calc(disp13, dt, tc_or_valid);
for n=1:20
    disp13(n)=[];
    vel13(n) =[];
    force13(n)=[];
    temp13(n)=[];
end
for n=11020:-1:9001
    disp13(n)=[];
    vel13(n)=[];
    force13(n)=[];
    temp13(n)=[];
```

```
end

disp14 = [P19E(ss,2)*0.711];
force14 = [P19E(ss,1)*62.208];
temp14 = [P19E(ss,3)*100];


disp14 = filter(b,a,disp14);
force14 = filter(b,a,force14);
temp14 = filter(b,a,temp14);
%Calculate velocity
vel14 = vel_calc(disp14, dt, tc_or_valid);
for n=1:20
    disp14(n)=[];
    vel14(n) =[];
    force14(n)=[];
    temp14(n)=[];
end
for n=11020:-1:11001
    disp14(n)=[];
    vel14(n)=[];
    force14(n)=[];
    temp14(n)=[];
end

disp15 = [P19H(ss,2)*0.711];
force15 = [P19H(ss,1)*62.208];
temp15 = [P19H(ss,3)*100];


disp15 = filter(b,a,disp15);
force15 = filter(b,a,force15);
temp15 = filter(b,a,temp15);
%Calculate velocity
vel15 = vel_calc(disp15, dt, tc_or_valid);
for n=1:20
    disp15(n)=[];
    vel15(n) =[];
    force15(n)=[];
    temp15(n)=[];
end
for n=11020:-1:11001
    disp15(n)=[];
    vel15(n)=[];
    force15(n)=[];
    temp15(n)=[];
end

%MEDIO - PRIMERA
disp16 = [P19P(ss,2)*0.711];
force16 = [P19P(ss,1)*62.208];
```

```
temp16 = [P19P(ss,3)*100];

disp16 = filter(b,a,disp16);
force16 = filter(b,a,force16);
temp16 = filter(b,a,temp16);
% Calculate velocity
vel16 = vel_calc(disp16, dt, tc_or_valid);
for n=1:20
    disp16(n)=[];
    vel16(n) =[];
    force16(n)=[];
    temp16(n)=[];
end
for n=11020:-1:11001
    disp16(n)=[];
    vel16(n)=[];
    force16(n)=[];
    temp16(n)=[];
end

% disp17 = [P19L(ss,2)*0.711];
% force17 = [P19L(ss,1)*62.208];
% temp17 = [P19L(ss,3)*100];
%
% disp17 = filter(b,a,disp17);
% force17 = filter(b,a,force17);
% temp17 = filter(b,a,temp17);
% % Calculate velocity
% vel17 = vel_calc(disp17, dt, tc_or_valid);
% for n=1:20
%     disp17(n)=[];
%     vel17(n) =[];
%     force17(n)=[];
%     temp17(n)=[];
% end
% for n=11020:-1:11001
%     disp17(n)=[];
%     vel17(n)=[];
%     force17(n)=[];
%     temp17(n)=[];
% end

disp18 = [P19N(ss,2)*0.711];
force18 = [P19N(ss,1)*62.208];
temp18 = [P19N(ss,3)*100];

disp18 = filter(b,a,disp18);
force18 = filter(b,a,force18);
temp18 = filter(b,a,temp18);
% Calculate velocity
vel18 = vel_calc(disp18, dt, tc_or_valid);
for n=1:1000
```

```
    disp18(n)=[];
    vel18(n) =[];
    force18(n)=[];
    temp18(n)=[];
end
for n=10020:-1:10001
    disp18(n)=[];
    vel18(n)=[];
    force18(n)=[];
    temp18(n)=[];
end

disp = [];
disp =
[disp1;disp2;disp3;disp4;disp6;disp7;disp8;disp9;disp11;disp12;disp13;d
isp14;disp15;disp16;disp18];

velocity = [];
velocity =
[vel1;vel2;vel3;vel4;vel6;vel7;vel8;vel9;vel11;vel12;vel13;vel14;vel15;
vel16;vel18];

temp = [];
temp =
[temp1;temp2;temp3;temp4;temp6;temp7;temp8;temp9;temp11;temp12;temp13;t
emp14;temp15;temp16;temp18];

force = [];
force =
[force1;force2;force3;force4;force6;force7;force8;force9;force11;force1
2;force13;force14;force15;force16;force18];

% disp = [];
% disp = [disp1;disp2;disp3;disp4;disp6;disp7;disp8;disp9;disp12];
%
% velocity = [];
% velocity = [vel1;vel2;vel3;vel4;vel6;vel7;vel8;vel9;vel12];
%
% temp = [];
% temp = [temp1;temp2;temp3;temp4;temp6;temp7;temp8;temp9;temp12];
%
% force = [];
% force =
[force1;force2;force3;force4;force6;force7;force8;force9;force12];

big = [disp,velocity,temp,force];

figure (110)
plot(disp);

figure (120)
```

```
plot(velocity)

n = [];
nsave1 = [];
count = 0;
 for n=1:length(big)
     if big(n,1) <= 0.02;
     count = count + 1;
     nsave1(count) = n;
     big(n,:) = 0;
     end
 end

n = [];
nsave2 = [];
count = 0;
 for n=1:length(big)
    if big(n,4) <= 0.2;
     count = count + 1;
    nsave2(count) = n;
     big(n,:) = 0;
    end
  end

n = [];
nsave5 = [];
count = 0;
for n = 1:length(big)
 if big(n,2) >= 1.1;
    count = count + 1;
    nsave5(count) = n;
    big(n,:) = 0;
    end
end
%big(nsave,:) = [];

n = [];
nsave6 = [];
count = 0;
for n =1:length(big)
    if big(n,2)<= -1.4;
    count = count + 1;
    nsave6(count) = n;
    big(n,:) = 0;
    end
end

%big(nsave,:) = [];
%
% n = [];
% nsave7 = [];
% count = 0;
```

```
% for n = 110000:size(big)
%     if  abs(big(n,2)) >= 1;
%     count = count + 1;
%     nsave7(count) = n;
%     big(n,:) = 0;
%     end
% end
% %big(nsave,:) = [];

%ndel =
[nsave1,nsave2,nsave3,nsave4,nsave5,nsave6,nsave7,nsave8,nsave9,nsave10
,nsave11,nsave12,nsave13];
ndel = [nsave1,nsave2,nsave5,nsave6];
%ndel = [nsave1];
big(ndel,:) = [];

disp = big(:,1);
velocity = big(:,2);
temp = big(:,3);
force = big(:,4);

figure(210)
plot(big(:,1))
title('After elemination')
figure(220)
plot(big(:,2))
title('After elemination')

% Define initial empty matrix for validation data
disp_val = [];
force_val = [];
temp_val = [];
% Define validation data

disp_val1 = [P19L(ss,2)*0.711];
force_val1 = [P19L(ss,1)*62.208];
temp_val1 = [P19L(ss,3)*100];

% Filter the data
disp_val1 = filter(b,a,disp_val1);
force_val1 = filter(b,a,force_val1);
temp_val1 = filter(b,a,temp_val1);
% Calculate velocity
vel_val1 = vel_calc(disp_val1, dt, tc_or_valid);

% Trim the first and last part of the filtered data
for n=1:20
    disp_val1(n)=[];
    vel_val1(n) =[];
    force_val1(n)=[];
    temp_val1(n)=[];
```

```
end
for n=11020:-1:11001
    disp_val1(n)=[];
    vel_val1(n)=[];
    force_val1(n)=[];
    temp_val1(n)=[];
end

disp_val2 = [P18Q(ss,2)*0.711];
force_val2 = [P18Q(ss,1)*62.208];
temp_val2 = [P18Q(ss,3)*100];

disp_val2 = filter(b,a,disp_val2);
force_val2 = filter(b,a,force_val2);
temp_val2 = filter(b,a,temp_val2);
% Calculate velocity
vel_val2 = vel_calc(disp_val2, dt, tc_or_valid);
for n=1:20
    disp_val2(n)=[];
    vel_val2(n) =[];
    force_val2(n)=[];
    temp_val2(n)=[];
end
for n=11020:-1:11001
    disp_val2(n)=[];
    vel_val2(n)=[];
    force_val2(n)=[];
    temp_val2(n)=[];
end


disp_val = [disp_val1;disp_val2];
force_val = [force_val1;force_val2];
temp_val = [temp_val1;temp_val2];
velocity_val = [vel_val1;vel_val2];

big_val = [disp_val,velocity_val,temp_val,force_val];

n = [];
nsave1v = [];
count = 0;
for n=1:length(big_val)
     if big_val(n,1) <= 0.02;
     count = count + 1;
     nsave1v(count) = n;
     big_val(n,:) = 0;
     end
end
n = [];
nsave2v = [];
count = 0;
for n=1:length(big_val)
```

```
     if big_val(n,4) <= 0.2;
     count = count + 1;
    nsave2v(count) = n;
     big_val(n,:) = 0;
     end
end
n = [];
nsave3v = [];
count = 0;
for n=1:length(big_val)
     if abs(big_val(n,2)) >= 1.1;
     count = count + 1;
     nsave3v(count) = n;
     big_val(n,:) = 0;
     end
 end
% big(nsave,:) = [];

ndelv = [nsave1v,nsave2v,nsave3v];
big_val(ndelv,:) = [];

figure (125)
title ('Validation Vel')
plot(velocity_val);


disp_val = big_val(:,1);
velocity_val = big_val(:,2);
temp_val = big_val(:,3);
force_val = big_val(:,4);

%Convert to strain, strain rate, stress
strain_all = disp/length_strain;
strainrate_all = velocity/length_strate;
stress_all = force/area;
temp_all = temp;

%Convert validation data to strain, strain rate, stress
strain_val = disp_val/length_strain;
strainrate_val = velocity_val/length_strate;
stress_val = force_val/area;
temp_val = temp_val;

% Load training and checking data
sma_anfis = [ strain_all, strainrate_all, temp_all, stress_all ];
save sma_anfis sma_anfis;

% Load validation data
sma_anfis_valid = [ strain_val, strainrate_val, temp_val, stress_val ];
save sma_anfis_valid sma_anfis_valid;
```

```
[n_rows,n_cols] = size(sma_anfis);
% Use every other data points for training
strain = sma_anfis(1:2:n_rows,1) ;
strainrate = sma_anfis(1:2:n_rows,2) ;
temperature = sma_anfis(1:2:n_rows,3);
stress = sma_anfis(1:2:n_rows,4);

% Similiarly use every other data points for checking
strain_chk = sma_anfis(2:2:n_rows,1) ;
strainrate_chk = sma_anfis(2:2:n_rows,2) ;
temperature_chk = sma_anfis(2:2:n_rows,3);
stress_chk = sma_anfis(2:2:n_rows,4);

% Define training and checking data
smaData_train = [strain,     strainrate,     temperature,      stress
] ;
smaData_chk   = [strain_chk, strainrate_chk, temperature_chk,
stress_chk] ;
% Define validation data
smaData_valid = sma_anfis_valid;


% Plot the training data
figure (15000)
%
subplot(4,1,1)
plot(smaData_train(:,1), 'b--');
legend('Training Data');
grid;
ylabel('Strain (%)', 'FontSize', 12);
%
subplot(4,1,2)
plot(smaData_train(:,2), 'b--');
legend('Training Data');
grid;
ylabel('Strain rate (1/s-1)', 'FontSize', 12);
%
subplot(4,1,3)
plot(smaData_train(:,3), 'b--');
legend('Training Data');
grid;
ylabel('Temperature (C)', 'FontSize', 12);
%
subplot(4,1,4)
plot(smaData_train(:,4), 'b--');
legend('Training Data');
grid;
ylabel('Stress (MPa)', 'FontSize', 12);

% Plot the validation data
figure (16000)
%
```

```
subplot(4,1,1)
plot(smaData_valid(:,1), 'b--');
legend('Validation Data');
grid;
ylabel('Strain (%)', 'FontSize', 12);
%
subplot(4,1,2)
plot(smaData_valid(:,2), 'b--');
legend('Validation Data');
grid;
ylabel('Strain rate (1/s-1)', 'FontSize', 12);
%
subplot(4,1,3)
plot(smaData_valid(:,3), 'b--');
legend('Validation Data');
grid;
ylabel('Temperature (C)', 'FontSize', 12);
%
subplot(4,1,4)
plot(smaData_valid(:,4), 'b--');
legend('Validation Data');
grid;
ylabel('Stress (MPa)', 'FontSize', 12);
```

## M2

```
%smaTrainTemp.m

clear all
close all

%  Override the default settings for graphs in Matlab
set(0, 'DefaultAxesFontSize', 13)
set(0, 'DefaultAxesFontWeight', 'bold')
set(0, 'DefaultAxesLineWidth', 2)
set(0, 'DefaultLineLineWidth', 1.6)

% Load training, checking and validation data
load sma_anfis
load sma_anfis_val

[n_rows,n_cols] = size(sma_anfis);
% Use every other data points for training
strain = sma_anfis(1:2:n_rows,1) ;
strainrate = sma_anfis(1:2:n_rows,2) ;
temperature = sma_anfis(1:2:n_rows,3);
stress = sma_anfis(1:2:n_rows,4);

% Similiarly use every other data points for checking
strain_chk = sma_anfis(2:2:n_rows,1) ;
strainrate_chk = sma_anfis(2:2:n_rows,2) ;
temperature_chk = sma_anfis(2:2:n_rows,3);
```

```
stress_chk = sma_anfis(2:2:n_rows,4);

% Define training and checking data
smaData_train = [strain,     strainrate,     temperature,      stress
] ;
smaData_chk   = [strain_chk, strainrate_chk, temperature_chk,
stress_chk] ;
% Define validation data
smaData_valid = sma_anfis_val;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set parameters for training with ANFIS
mf_strain = 3 ;
mf_strate = 2 ;
mf_temp = 2 ;
% Define the number of membership functions are to be used for each
input
mf_n   = [mf_strain  mf_strate  mf_temp ];

%  Specify the step size for the ANFIS iteration
ss = 0.12;
ss_dec_rate = 0.8;
ss_inc_rate = 1.2;

%  Specify the number of epochs to train with
epoch_n = 200;

% Define the type of membership function
%mf_type = 'gbellmf';
mf_type = 'gaussmf';
%mf_type = 'trapmf';
%mf_type = 'gauss2mf';
%mf_type = 'trimf';

% Generate FIS structure
disp ( ' Before genfis1')
in_fismat = genfis1(smaData_train, mf_n, mf_type);

% Start the clock to track time for training
tic
%
disp ( ' Before training starts')
% Start training data with ANFIS
[trn_out_fismat trn_error step_size chk_out_fismat chk_error] = ...
    anfis(smaData_train, in_fismat, [epoch_n nan ss ss_dec_rate
ss_inc_rate], ...
    [1,1,1,1], smaData_chk);

% End the clock that tracks training
train_time = toc ;
```

```
train_time = train_time / 60.0;      % Convert execution time to
minutes

% Get filename if all plots are needed to be saved
FIS_filename = input('Enter filename to save FIS : ','s');
FIS_filename

disp(['Total training time = ',           num2str(train_time)]);
disp(['Number of data points = ',
num2str(size(smaData_train,1))]);
disp(['Minimum RMSE of training data = ', num2str(min(trn_error))]);
disp(['Minimum RMSE of checking data = ', num2str(min(chk_error))]);
disp(' ');

% Set label and show FIS surface
% FIS structure of original data
in_fismat.input(1).name = 'Strain  ';
in_fismat.input(2).name = 'Strain rate ';
in_fismat.input(3).name = 'Temperature (C) ';
in_fismat.output(1).name = 'Stress (MPa) ';
% FIS structure of training data
trn_out_fismat.input(1).name = 'Strain ';
trn_out_fismat.input(2).name = 'Strain rate ';
trn_out_fismat.input(3).name = 'Temperature (C)';
trn_out_fismat.output(1).name = 'Stress (Mpa) ';
% FIS structure of checking data
chk_out_fismat.input(1).name = 'Strain ';
chk_out_fismat.input(2).name = 'Strain rate';
chk_out_fismat.input(3).name = 'Temperature (C) ';
chk_out_fismat.output(1).name = 'Stress (MPa)';

trn_out_surf = figure;
% Generate a surface plot of the output versus two of the inputs.
gensurf(trn_out_fismat,[1 3],1,[30 30],[nan -5 nan]);
figure
gensurf(trn_out_fismat,[1 2],1,[30 30]);
figure
gensurf(trn_out_fismat,[1 2],1,[30 30],[nan nan 10]);
figure
gensurf(trn_out_fismat,[2 1],1,[30 30],[nan nan 10]);
title({'FIS Structure from Training Data',...
      ['(',FIS_filename,'.fig)']},'Interpreter','none');

% chk_out_surf = figure;
% gensurf(chk_out_fismat,[1 3],1);
% gensurf(chk_out_fismat,[1 2],1);
% title('FIS Structure from Checking Data');

% Plot MFs before and after training
MF_Fig = figure;
orient portrait;
set(gcf, 'PaperPosition', [0.25 0.25 8.00 10.50]);
```

```
subplot(3,2,1)
plotmf(in_fismat, 'input', 1);
subplot(3,2,3)
plotmf(in_fismat, 'input', 2);
subplot(3,2,5)
plotmf(in_fismat, 'input', 3);
subplot(3,2,2)
plotmf(trn_out_fismat, 'input', 1);
subplot(3,2,4)
plotmf(trn_out_fismat, 'input', 2);
subplot(3,2,6)
plotmf(trn_out_fismat, 'input', 3);
delete(findobj(gcf, 'type', 'text'));
subplot(321);
title('Initial MFs for strain');
subplot(322);
title('Final MFs for strain');
subplot(323);
title('Initial MFs on strain rate');
subplot(324);
title('Final MFs on strain rate');
subplot(325);
title('Initial MFs on temperature');
subplot(326);
title('Final MFs on temperature');

% Plot RMSE of Training and Checking Data
% Compute error of FIS on training and checking data
trn_out_anfis = evalfis(smaData_train(:,1:3), trn_out_fismat);
figure
plot(trn_out_anfis);
title('Show stress from ANFIS', 'FontSize', 12);

% Error from prediction
start_ind = 1;
end_ind   = size(smaData_train,1);
ind_mat   = [1:end_ind]';
% Compute the difference between the force of the experiment and the
force
% of the FIS.
smaData_train_error = abs(smaData_train(:,4)-trn_out_anfis);

RMSE_Fig = figure;
orient landscape;
set(gcf, 'PaperPosition', [0.25 0.25 10.50 8.00]);
subplot(4,1,1)
plot(step_size);
grid
xlabel('Epochs', 'FontSize', 12);
ylabel('Step Size', 'FontSize', 12);
title({'Results of Training with ANFIS', ['(ss=',num2str(ss),',
ss_dec_rate=',...
```

```
       num2str(ss_dec_rate),', ss_inc_rate
=',num2str(ss_inc_rate),')']},'Interpreter',...
       'None', 'FontSize', 12);
subplot(4,1,2)
plot(trn_error);
grid
xlabel('Epoch', 'FontSize', 12);
ylabel('RMSE', 'FontSize', 12);
subplot(4,1,3)
plot(ind_mat, smaData_train(:,4), 'b--', ind_mat, trn_out_anfis, 'r');
%axis([0 ind_mat(end) 0 0.4]);
legend('Training Data', 'ANFIS Prediction');
grid;
%title('Plot of Time History and ANFIS Prediction of Training Data');
%xlabel('Time (sec)');
ylabel('Stress (MPa)', 'FontSize', 12);
subplot(4,1,4)
plot(ind_mat,smaData_train_error);
%axis([0 ind_mat(end) 0 0.2]);
grid;
xlabel('Data Points', 'FontSize', 12);
ylabel('Error (MPa)', 'FontSize', 12);

inputs_out = figure(6);
subplot (4,1,1);
plot (ind_mat, smaData_train(:,1));
grid;
xlabel('Data Points','FontSize',12);
ylabel('Strain(%)', 'FontSize', 12);
subplot(4,1,2);
plot (ind_mat, smaData_train(:,2));
grid;
xlabel('DataPoints', 'FontSize',12);
ylabel('Strain rate', 'FontSize',12);
subplot(4,1,3);
plot(ind_mat, smaData_train(:,3));
grid;
xlabel('Data Points', 'FontSize', 12);
ylabel('Temperature (C)', 'FontSize',12);
subplot(4,1,4);
plot(ind_mat, smaData_train(:,4), 'b--', ind_mat, trn_out_anfis, 'r');
%axis([0 ind_mat(end) 0 0.4]);
legend('Training Data', 'ANFIS Prediction');
grid;
title('Plot of Time History and ANFIS Prediction of Training Data');
%xlabel('Time (sec)');
ylabel('Stress (MPa)', 'FontSize', 12);

figure
subplot(2,2,1)
n1 = 1000;
n2 = 1500;
```

```
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
legend('Experimental Data', 'Fuzzy Prediction')
subplot(2,2,2)
n1 = 2100;
n2 = 3500;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')
subplot(2,2,3)
n1 = 4200;
n2 = 5500;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')
subplot(2,2,4)
n1 = 6200;
n2 = 7500;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')

figure
subplot(2,2,1)
n1 = 8200;
n2 = 9500;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
legend('Experimental Data', 'Fuzzy Prediction')
subplot(2,2,2)
n1 = 10000;
n2 = 13000;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')
subplot(2,2,3)
n1 = 14000;
n2 = 17000;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')
subplot(2,2,4)
n1 = 18000;
n2 = 21000;
```

```
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')
%
%
figure
subplot(2,2,1)
n1 = 22000;
n2 = 25000;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
legend('Experimental Data', 'Fuzzy Prediction')
subplot(2,2,2)
n1 = 26000;
n2 = 29000;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')
subplot(2,2,3)
n1 = 30000;
n2 = 31000;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')
subplot(2,2,4)
n1 = 32000;
n2 = 33000;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')

%
figure
subplot(2,2,1)
n1 = 34000;
n2 = 35000;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
legend('Experimental Data', 'Fuzzy Prediction')
subplot(2,2,2)
n1 = 36000;
n2 = 38000;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')
subplot(2,2,3)
```

```
n1 = 39000;
n2 = 40000;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')
% subplot(2,2,4)
% n1 = 79000;
% n2 = 81000;
% plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
% %legend('Experimental Data', 'Fuzzy Prediction')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%\
figure
subplot(3,3,1)
n1 = 1000;
n2 = 1500;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
legend('Experimental Data', 'Fuzzy Prediction')
subplot(3,3,2)
n1 = 2100;
n2 = 3500;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')
subplot(3,3,3)
n1 = 4200;
n2 = 5500;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')
subplot(3,3,4)
n1 = 10000;
n2 = 13000;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')
subplot(3,3,5)
n1 = 22000;
n2 = 25000;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
```

```
%legend('Experimental Data', 'Fuzzy Prediction')
subplot(3,3,6)
n1 = 26000;
n2 = 29000;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')
subplot(3,3,7)
n1 = 30000;
n2 = 31000;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')
subplot(3,3,8)
n1 = 32000;
n2 = 33000;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')
subplot(3,3,9)
n1 = 34000;
n2 = 35000;
plot
(smaData_train(n1:n2,1),smaData_train(n1:n2,4),smaData_train(n1:n2,1),t
rn_out_anfis(n1:n2),'y');
%legend('Experimental Data', 'Fuzzy Prediction')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

max_trn_error = max(smaData_train_error);
min_trn_error = min(smaData_train_error);
disp(['Maximum Error = ', num2str(max_trn_error),' N']);
disp(['Minimum Error = ', num2str(min_trn_error),' N']);

% Check if data are to be saved
if ~isempty(FIS_filename)
    cd 'ANFIS Data';
    writefis (trn_out_fismat, FIS_filename);
    hgsave (MF_Fig,   [FIS_filename,'_MF.fig']);
    hgsave (RMSE_Fig, [FIS_filename,'_trn_RMSE.fig']);
    hgsave (inputs_out, [FIS_filename,'_inputs_out.fig']);
    cd ..
    disp(' ');
    disp(['FIS has been saved to ".\ANFIS Data\',FIS_filename,'.fis']);
    clear MF_Fig
end

% Save some training parameters for future reference
```

```
save(['.\ANFIS Data\',FIS_filename,'_info.mat'], 'epoch_n', 'ss',
'ss_dec_rate',...
      'ss_inc_rate', 'trn_error', 'step_size', 'chk_error',
'train_time',...
      'max_trn_error', 'min_trn_error');
```

**M3**

```
%-------------------------------------------------------------------
% Filename : smaValidTemp.m
% MATLAB M-file to verify FIS for Univ. of Chile shape memory alloy.
%   trained from smaTrain.m
%
% Original code written by  Prof. Paul N. Roschke
%
%-------------------------------------------------------------------
%  Load the validation data that was saved by
%  BuildTrainCheckValidData_simple.m and grafdiag_mod.m
load sma_anfis_valid sma_anfis_valid

%  Use selected data points for verifying (i.e. not necessarily all)
%interval = 8;

% Determine the number of data points in the validation data set
nDataPoints = size(sma_anfis_valid,1)


%time  = mrdata_val(:,5);
%dt_valid = time(2) - time(1) ;
% Temporarily assign a time step size in order to run Simulink
dt_valid = 0.005;
% Make up a time vector:
time_max = (nDataPoints-1)*dt_valid ;
time = [0:dt_valid:time_max]';

%%Offset the second input (previous displacement) by n_offset time
% increments as the second input variable.
%n_offset = 50;
%n_offset = 40;

% Set up training arrays.  Use every other data point.
% Load the first column of the input dataset into the displacement
strain = [sma_anfis_valid(1:1:size(sma_anfis_valid,1), 1)] ;
strainrate = sma_anfis_valid(1:1:size(sma_anfis_valid,1), 2);
temp = sma_anfis_valid(1:1:size(sma_anfis_valid,1), 3) ;
stress = sma_anfis_valid(1:1:size(sma_anfis_valid,1), 4) ;

% Set up input and output matrix for verification
Exp_input = [strain strainrate temp ];
%Exp_output = [force];

%  Load the fuzzy MR file into the workspace
FISname = input('Enter FIS to be loaded : ', 's');
```

```
%SMAfis = readfis ( ['.\ANFIS Data\', FISname, '.fis'] );
SMAfis = readfis ( ['ANFIS Data\', FISname, '.fis'] );
% Get limits of saturation for fuzzy model of the MR damper
% These are equivalent to the extreme limits of the universes of
discourse
%    for the three input variables, displacement (cm), velocity
(cm/sec), and voltage (V).
sat_limits          = getfis(SMAfis, 'inrange');
disp_sat_min        = sat_limits(1, 1);
disp_sat_max        = sat_limits(1, 2);
velocity_sat_min    = sat_limits(2, 1);
velocity_sat_max    = sat_limits(2, 2);
prestress_sat_min   = sat_limits(3, 1);
prestress_sat_max   = sat_limits(3, 2);
% Replace the saturation of prestress to be between 0 and 10 kg-f
%prestress_sat_min = 2.0;
%prestress_sat_max = 4.0;

% Reset label of input and output
SMAfis.input(1).name = 'Strain';
SMAfis.input(2).name = 'Strain rate';
SMAfis.input(3).name = 'Temperature';
SMAfis.output(1).name = 'Stress';

%  Define starting time for Simulink simulation
%time_start = sma_data_val(1,5);
%  Define ending time for Simulink simulation
%time_end = mrdata_val(end,5);
% Temporarily guess starting and ending times for the validation run in
% Simulink.  Units (sec.)
time_start = 0
time_end   = time_max


%  Start running SIMULINK to evaluate the force of validating data
disp(' ---> SIMULATION: sim_sma_valid_temp.mdl');
OPTIONS = simset('solver', 'ode5', 'FixedStep', dt_valid);
sim('sim_sma_valid_temp', [time_start time_end], OPTIONS, [])

% Plot RMSE of Validation Data
% Compute error of FIS on validation data
val_out_fis = sim_out_val;
clear sim_out_val;
%start_time = 0;                          % Starting time of
verification in sec.
%end_time   = time(end);             % Stopping time of verification in
sec.
start_time = time_start;
end_time = time_end;
%  Set up starting and ending indices (number of the data point)
start_ind  = start_time/dt_valid + 1;
end_ind    = round(end_time/dt_valid) + 1;
```

```
time_ind = [start_time:dt_valid:end_time];
plot_ind = [start_ind:end_ind];
% Error from prediction
smadata_val_error = abs(stress - val_out_fis);
clear start_time start_ind end_time end_ind;

% Plot validation data
Val_Fig = figure;
orient landscape;
set(gcf, 'PaperPosition', [0.25 0.25 10.50 8.00]);

subplot(4,1,1)
plot(time_ind, strain, 'b--');
axis([time_start, time_end+1.0, -1, 1]);
%axis([200, 205, -5000, 5000]);
legend('Validating Strain Data');
grid;
title({'Plot of Time Strain History ',...
       ['(', FISname, '_Val.fig)']}, 'FontSize', 12, 'Interpreter',
'none');
xlabel('Time (sec)');
ylabel('Strain (%)');

subplot(4,1,2)
plot(time_ind, strainrate, 'b--');
axis([time_start, time_end+1.0, -20, 20]);
axis tight
legend('Validating Strain Rate Data');
grid;
title({'Plot of Strain Rate Time History ',...
       ['(', FISname, '_Val.fig)']}, 'FontSize', 12, 'Interpreter',
'none');
xlabel('Time (sec)');
ylabel('Strain Rate (1/s)');

subplot(4,1,3)
plot(time_ind, stress, 'b--', time_ind, val_out_fis(plot_ind), 'r');
%axis([time_start, time_end+1.0, 0,0.5 ]);
axis tight
legend('Validating Data', 'ANFIS Prediction');
grid;
title({'Plot of Time History and ANFIS Prediction of Validating
Data',...
       ['(', FISname, '_Val.fig)']}, 'FontSize', 12, 'Interpreter',
'none');
xlabel('Time (sec)');
ylabel('Stress (MPa)');

subplot(4,1,4)
plot(time_ind, smadata_val_error);
%axis([time_start, time_end+1.0, 0, 0.3]);
axis tight
```

```
grid;
xlabel('Time (sec)', 'FontSize', 12);
ylabel('Error (MPa)', 'FontSize', 12);

figure;
n1 = 1;
n2 = 4000;
subplot(2,1,2)
plot(strain(n1:n2),stress(n1:n2));
xlabel ('Strain (%)');
ylabel ('Stress (MPa)');
title ('Experimental Results');
%axis ( [-1 1.5 0 0.30]);
grid
hold on
plot(strain(n1:n2), val_out_fis(n1:n2),'-y');
title ('FIS Prediction');
%axis ( [-1 1.5 0 0.30]);
subplot(2,1,1)
n1 = 5000;
n2 = 10000;
plot(strain(n1:n2),stress(n1:n2));
xlabel ('Strain (%)');
ylabel ('Stress (MPa)');
title ('Experimental Results');
%axis ( [-1 1.5 0 0.30]);
grid
hold on
plot(strain(n1:n2), val_out_fis(n1:n2),'-y');
title ('FIS Prediction');
%axis ( [-1 1.5 0 0.30]);

% Save validating figure to "Valid Data" folder
hgsave(Val_Fig, ['.\Valid Data\', FISname,'_Val.fig']);
clear Val_Fig

% Show statistical data
disp(' ');
input('Press Enter to proceed statistical analysis : ');
val_max_error = max(smadata_val_error);
val_min_error = min(smadata_val_error);

F_exp = stress;
F_fuz = val_out_fis;
totaltime = time (end) ;

eta_t = sqrt(sum((F_exp - F_fuz).^2));
eta_strate = sqrt(sum((F_exp - F_fuz).^2.*abs(strainrate)));
eta_ps = sqrt(sum((F_exp - F_fuz).^2.*abs(temp)));

mu_F = mean(F_fuz);
```

```
sigma_F = sqrt(sum((F_exp - mu_F).^2));

% Error measured between the fuzzy predicted force and the validation
data
E_t = eta_t/sigma_F;    % as a function of time
E_strate = eta_strate/sigma_F;  % as a function of displacement
E_ps = eta_ps/sigma_F;  % as a function of prestress force
Errors = [E_t E_strate E_ps];
%Errors = [E_t E_strate];
%Errors = [E_t E_x];

save(['.\Valid Data\', FISname,'_Stats.mat'], 'val_max_error',
'val_min_error',...
     'Exp_input', 'F_exp', 'F_fuz', 'Errors', 'mu_F', 'sigma_F');
clc;
disp(['Statistical data for validation of ', FISname]);
disp(' ');
disp(['Maximum Error = ', num2str(val_max_error),' kN']);
disp(['Minimum Error = ', num2str(val_min_error),' kN']);
disp(['E_t      = ', num2str(E_t)]);
disp(['E_strate      = ', num2str(E_strate)]);
disp(['E_ps      = ', num2str(E_ps)]);
disp(['mu_F     = ', num2str(mu_F)]);
disp(['sigma_F  = ', num2str(sigma_F)]);
disp(' ');
```

## M4

```
% BuildTrainCheckValidDatamanyWires.m
%
% Generate velocity checking sets for neural fuzzy
% training of SMA data with ANFIS.  Train a fis for one individual
wire.
%  Modified  26 April 2005   Prof. Roschke
clear all
close all

addpath 'F:\Gloria\Routines'
addpath 'F:\Gloria\ensayos mecánicos de alambres SMA - CuAlBe'
addpath 'F:\Gloria\registros de diagonales SMAs'
addpath 'F:\Gloria\RESULTADOS ENSAYO ESTRUCTURA CON DIAGONALES SMA'
% Bring in data sets from another directory from an earthquake loading
on
% the experimental building.  Gloria gave me these files.
% Created by Prof. Roschke
% Modified:  14 April 2005
% Load Sylmar earthquake response data from experiment
load r697_alambres ;
% Load Kobe   earthquake response data from experiment
load r691_alambres ;
% Load Taft   earthquake response data from experiment
```

```
load r684_alambres ;
% Load El Centro earthquake response data from experiment
load r682_alambres ;
% Load Llolleo1 earthquake response data from experiment
load r620_alambres ;
% Load Llolleo2 earthquake response data from experiment
load r620_alambres2 ;

% Delete previous files used for earlier runs.
delete sma_anfis
delete sma_anfis_valid

% Specify 'dt' rate at which data were taken in lab.
dt = 0.0005 ;


% Note all data were taken at a rate of 2000 Hz (0.0005 sec).
% The fuzzy inference system is being designed to have 3 inputs and one
% output.

% Set up an array defining the starting and ending data set locations
% for training/checking data and the initial prestressing force that
was
% measured in the laboratory.  Use this to eliminate beginning and
% ending non-dynamic portions of the data from the data that are to
% be used during the fuzzy inference system training.
% These values are determined by
% looking at a plot of the reduced data and arbitrarily setting
boundaries.
% The prestressed forces 'PS' are determined from a table of values
that
% Gloria prepared.
% There are 8 wires and several earthquakes are used in the training.
The
% input below is in groups of eight.  Earthquake order is Sylmar, Kobe,
% Taft, El Centro, Llolleo1, and Llolleo2.
% Gloria's thesis (toward the end).
% First earthquake is Sylmar.
startStopPSdisp = [ 15000  22000 2.977 6.38 ;...
                15000  20000 3.183 8.61;...
                15000  20000 3.133 9.01;...
                15000  20000 2.780 8.51;...
                15000  20000 3.459 7.77;...
                15000  20000 2.891 5.15;...
                15000  20000 3.252 8.25;...
                15000  20000 2.840 5.14;...
                % Kobe
                12000  18000 2.548 5.29;...
                12000  18000 2.951 7.73;...
                12000  18000 2.753 6.10;...
                12000  18000 2.333 5.60;...
                12000  18000 3.222 7.05;...
```

```
12000  18000 2.676 4.38;...
12000  18000 3.314 6.88;...
12000  18000 2.508 3.50;...
% Taft
13000  18000 2.508 3.52;...
13000  18000 2.913 5.42;...
13000  18000 2.629 4.33;...
13000  18000 2.279 4.03;...
13000  18000 3.224 5.77;...
13000  18000 2.612 3.71;...
13000  18000 3.232 5.81;...
13000  18000 2.448 3.03;...
% El Centro
12000  31000 2.419 4.03;...
12000  31000 2.857 6.45;...
12000  31000 2.619 5.48;...
12000  31000 2.25 5.005;...
12000  31000 3.196 6.24;...
12000  31000 2.541 3.70;...
12000  31000 3.195 6.54;...
12000  31000 2.439 3.19;...
% Llolleo1
26000  30000 2.370 6.19;...
26000  30000 2.822 8.38;...
26000  30000 2.586 8.16;...
26000  30000 2.193 7.71;...
26000  30000 3.184 7.35;...
26000  30000 2.519 4.53;...
26000  30000 3.106 8.35;...
26000  30000 2.456 5.31;...
% Llolleo2
14000  25000 2.229 0;...
14000  25000 2.658 0;...
14000  25000 2.317 0;...
14000  25000 2.040 0;...
14000  25000 3.011 0;...
14000  25000 2.349 0;...
14000  25000 3.049 0;...
14000  25000 2.193 0];

% Set up initially empty matrices to hold the training/check and
validation
% datasets.
sma_anfis = [];
sma_anfis_valid = [];

% For training and checking, set an index:
tc_or_valid = 0 ;

 %Run Gloria's file to reduce the 'raw' experimental data for Sylmar
eq_num = 697 ;
```

```
[dos1,dos2,dps1,dps2,don1,don2,dpn1,dpn2,cos1,cos2,cps1,cps2,con1,con2,
cpn1,cpn2] ...
       =   grafdiag_mod_manywires(r697_alambres,   0,   startStopPSdisp,
sma_anfis, sma_anfis_valid, eq_num, tc_or_valid, dt) ;

% Load files for ANFIS training/checking
load sma_anfis sma_anfis

% Run Gloria's file to reduce the 'raw' experimental data for Kobe
eq_num = 691 ;
[dos1,dos2,dps1,dps2,don1,don2,dpn1,dpn2,cos1,cos2,cps1,cps2,con1,con2,
cpn1,cpn2] ...
       =   grafdiag_mod_manywires(r691_alambres,   0,   startStopPSdisp,
sma_anfis, sma_anfis_valid, eq_num, tc_or_valid, dt) ;

% Load files for ANFIS training/checking
load sma_anfis sma_anfis
 %Run Gloria's file to reduce the 'raw' experimental data for Taft
eq_num = 684 ;
[dos1,dos2,dps1,dps2,don1,don2,dpn1,dpn2,cos1,cos2,cps1,cps2,con1,con2,
cpn1,cpn2] ...
      =   grafdiag_mod_manywires(r684_alambres,   0,   startStopPSdisp,
sma_anfis, sma_anfis_valid, eq_num, tc_or_valid, dt) ;

%Load files for ANFIS training/checking
load sma_anfis sma_anfis

% Run Gloria's file to reduce the 'raw' experimental data for El Centro
%eq_num = 682 ;
%tc_or_valid = 0 ;
%[dos1,dos2,dps1,dps2,don1,don2,dpn1,dpn2,cos1,cos2,cps1,cps2,con1,con2
,cpn1,cpn2] ...
 %         = grafdiag_mod_manywires(r682_alambres,  0,  startStopPSdisp,
sma_anfis, sma_anfis_valid, eq_num, tc_or_valid, dt) ;

% Load files for ANFIS training/checking
%load sma_anfis sma_anfis

%Run Gloria's file to reduce the 'raw' experimental data for Llolleo1
eq_num = 620 ;
[dos1,dos2,dps1,dps2,don1,don2,dpn1,dpn2,cos1,cos2,cps1,cps2,con1,con2,
cpn1,cpn2] ...
      =   grafdiag_mod_manywires(r620_alambres,   0,   startStopPSdisp,
sma_anfis, sma_anfis_valid, eq_num, tc_or_valid, dt) ;

% Load files for ANFIS training/checking
load sma_anfis sma_anfis

% For VALIDATION:
% Run Gloria's file to reduce the 'raw' experimental data for El Centro
```

```
%eq_num = 620 ;     % Don't forget to change the number in subroutine
grafdiag.. at below
eq_num = 682 ;
%eq_num = 684 ;
%eq_num = 697 ;
tc_or_valid = 1 ;
[dos1,dos2,dps1,dps2,don1,don2,dpn1,dpn2,cos1,cos2,cps1,cps2,con1,con2,
cpn1,cpn2] ...
       =   grafdiag_mod_manywires(r682_alambres,   0,   startStopPSdisp,
sma_anfis, sma_anfis_valid, eq_num, tc_or_valid, dt) ;

% Load files for ANFIS validation
load sma_anfis_valid sma_anfis_valid
%    ++++++++++++++++++++++++++        Plotting    Training/Checking
++++++++++++++++++++
%
% Offset the second input (previous displacement) by n_offset time
% increments.
%n_offset = 50;
% Load files for ANFIS training/checking
load sma_anfis sma_anfis

% Set up training arrays
% Determine the size of the sma_anfis array
[n_rows,n_cols] = size(sma_anfis) ;
% Calculate the velocity for every displacement data point.
disp_all = [sma_anfis(1:1:n_rows,1)] ;
% Load the first column of the input dataset into the displacement
% Use every other data point for training.  Save others for checking.
disp = [sma_anfis(1:2:n_rows,1)] ;
% Calculate the velocity
velocity = sma_anfis(1:2:n_rows,2) ;
%tc_or_valid = 0 ;
%[velo] = vel_calc (disp_all, dt, tc_or_valid) ;
% Load the prestress vector from the 3rd column in the input dataset.
prestress = sma_anfis(1:2:n_rows,3);

% Begin the 'prev_disp' with zeros and follow with the values from the
% input dataset; put in column format
%%prev_disp = [zeros(n_offset,1); disp(1:1:size(disp)-n_offset, 1)] ;
% Load the force vector from the 4th column in the input dataset.
force = sma_anfis(1:2:n_rows,4) ;

% Set up checking arrays in a similar manner.  Use every other data
point
disp_chk = [sma_anfis(2:2:n_rows,1)] ;
%prev_disp_chk                  =                  [zeros(n_offset,1);
disp_chk(n_offset:1:size(disp_chk)-1, 1)] ;
velocity_chk = sma_anfis(2:2:n_rows,2) ;
%velocity_chk = [velo(2:2:n_rows)] ;
prestress_chk = sma_anfis(2:2:n_rows,3) ;
force_chk = sma_anfis(2:2:n_rows,4) ;
```

```
% Define training and checking data
%smaData_train = [disp,     prev_disp,     prestress,     force    ] ;
%smaData_chk   = [disp_chk, prev_disp_chk, prestress_chk, force_chk] ;
smaData_train = [disp,     velocity,     prestress,     force    ] ;
smaData_chk   = [disp_chk, velocity_chk, prestress_chk, force_chk] ;

% Plot the training data
figure (eq_num+15000)
%
subplot(4,1,1)
plot(smaData_train(:,1), 'b--');
legend('Training Data');
grid;
ylabel('Displacement (mm)', 'FontSize', 12);
%
subplot(4,1,2)
plot(smaData_train(:,2), 'b--');
legend('Training Data');
grid;
ylabel('Velocity (mm/s)', 'FontSize', 12);
%
subplot(4,1,3)
plot(smaData_train(:,3), 'b--');
legend('Training Data');
grid;
ylabel('Prestress (kg-f)', 'FontSize', 12);
%
subplot(4,1,4)
plot(smaData_train(:,4), 'b--');
legend('Training Data');
grid;
ylabel('Force (kg-f)', 'FontSize', 12);

%      ++++++++++++++++++++++++++++            Plotting      Validation
++++++++++++++++++++
%
% Offset the second input (previous displacement) by n_offset time
% increments
%n_offset = 50;
% Load files for ANFIS validation
load sma_anfis_valid sma_anfis_valid

% Set up validation arrays
% Determine the size of the sma_anfis array
[n_rows,n_cols] = size(sma_anfis_valid) ;
% Calculate the velocity for every displacement data point.
%disp_valid_all = [sma_anfis_valid(1:1:n_rows,1)] ;
% Calculate the velocity for validation
%tc_or_valid = 1 ;
%[velo_valid] = vel_calc (disp_valid_all, dt, tc_or_valid) ;
```

```
% Load the first column of the input dataset into the displacement
disp_valid = [sma_anfis_valid(1:1:n_rows,1)] ;
%% Begin the 'prev_disp' with zeros and follow with the values from the
%% input dataset; put in column format
% Load the velocity vector
velocity_valid = sma_anfis_valid(1:1:n_rows,2) ;
prestress_valid = sma_anfis_valid(1:1:n_rows,3) ;
% Load the force vector from the 2nd column in the input dataset.
force_valid = sma_anfis_valid(1:1:n_rows,4) ;

% Define validation data
smaData_valid    =    [disp_valid,    velocity_valid,    prestress_valid,
force_valid ] ;

% Plot the validation data
figure (eq_num+16000)
%
subplot(4,1,1)
plot(smaData_valid(:,1), 'b--');
legend('Validation Data');
grid;
ylabel('Displacement (mm)', 'FontSize', 12);
%
subplot(4,1,2)
plot(smaData_valid(:,2), 'b--');
legend('Validation Data');
grid;
ylabel('Velocity (mm/s)', 'FontSize', 12);
%
subplot(4,1,3)
plot(smaData_valid(:,3), 'b--');
legend('Validation Data');
grid;
ylabel('Prestress (kg-f)', 'FontSize', 12);
%
subplot(4,1,4)
plot(smaData_valid(:,4), 'b--');
legend('Validation Data');
grid;
ylabel('Force (kg-f)', 'FontSize', 12);
```

## M5

```
%-------------------------------------------------------------------------
-
% Filename : smaTrainVel2.m
% MATLAB M-file to train laboratory data of shape memory alloy.
%
% For training, the following parameters are defined
%   Input  : Displacement (mm), Velocity (mm/s), Prestress (kg-f)
%
%   Output : Force (kg-f)
```

```
%
% Original code written by Prof. Paul N. Roschke
%
%-------------------------------------------------------------------------
-

close all;
clear all;
clc;

% Change current directory
cd 'C:\Documents and Settings\oeo2452\My Documents\Shape Memory
Alloys\SMA Fuzzy Model'

% Specify 'dt' for the data time step (units are sec.)
dt = 0.0005 ;

% Load the training and checking data
load sma_anfis ;

% Offset the second input (previous displacement) by n_offset time
% increments as the second input variable.
%n_offset = 50;

% Determine the size of the sma_anfis array
[n_rows,n_cols] = size(sma_anfis) ;

% Calculate the velocity for every displacement data point.
disp      = sma_anfis(1:2:n_rows,1) ;
velocity  = sma_anfis(1:2:n_rows,2) ;
preforce = sma_anfis(1:2:n_rows,3) ;
force     = sma_anfis(1:2:n_rows,4) ;

disp_chk      = sma_anfis(2:2:n_rows,1) ;
velocity_chk  = sma_anfis(2:2:n_rows,2) ;
preforce_chk = sma_anfis(2:2:n_rows,3) ;
force_chk     = sma_anfis(2:2:n_rows,4) ;

% Convert to strain,strain rate,stress
wirelength = 400 ; %mm
wirearea = 1.963*10^-7 ; %m^2

coef_s = 10^-2 ;
coef_sr = 10^-2 ;
coef_stress = 10^6/9.806 ;

length_strain = wirelength*coef_s ;
length_strate = wirelength*coef_sr ;
area = wirearea*coef_stress ;
```

```
strain = disp/length_strain ;
strainrate = velocity/length_strate ;
prestress = preforce/area ;
stress = force/area ;

strain_chk = disp_chk/length_strain ;
strainrate_chk = velocity_chk/length_strate ;
prestress_chk = preforce_chk/area ;
stress_chk = force_chk/area ;

% Set up training arrays.  Use every other data point.
% Load the first column of the input dataset into the displacement
%disp = [sma_anfis(1:2:size(sma_anfis,1),1)] ;
% Begin the 'prev_disp' with zeros and follow with the values from the
% input dataset; put in column format
%
% Load the velocity vector
%velocity = velo(1:2:n_rows)

%prev_disp = [zeros(n_offset,1); disp(1:1:size(disp)-n_offset, 1)] ;
% Load the force vector from the 2nd column in the input dataset.
%force = sma_anfis(1:2:size(sma_anfis,1),2) ;
% Insert the prestress force as the third input variable.
%prestress = [sma_anfis(1:2:size(sma_anfis,1),3)]


% Set up checking arrays in a similar manner.  Use every other data
point
%disp_chk = [sma_anfis(2:2:size(sma_anfis,1),1)] ;
%velocity_chk = [velo(2:2:n_rows)] ;
%prev_disp_chk = [zeros(n_offset,1);
disp_chk(n_offset:1:size(disp_chk)-1, 1)] ;
%force_chk = sma_anfis(2:2:size(sma_anfis,1),2) ;
%prestress_chk = [sma_anfis(2:2:size(sma_anfis,1),3)]

% Define training and checking data
%smaData_train = [disp,     velocity,     prestress,      force    ] ;
%smaData_chk   = [disp_chk, velocity_chk, prestress_chk,  force_chk] ;

% Define training and checking data
smaData_train = [strain,     strainrate,     prestress,      stress
] ;
smaData_chk   = [strain_chk, strainrate_chk, prestress_chk,
stress_chk] ;

% Set parameters for training with ANFIS
mf_strain = 2 ;
mf_strate = 2 ;
mf_prestress = 2 ;
% Define the number of membership functions are to be used for each
input
```

```
%mf_n   = [mf_disp  mf_prev_disp  mf_prestress];
%mf_n   = [mf_disp  mf_prev_disp  mf_prestress];
mf_n   = [mf_strain  mf_strate  mf_prestress ];

%  Specify the step size for the ANFIS iteration
ss = 0.13;
ss_dec_rate = 0.8;
ss_inc_rate = 1.2;

%  Specify the number of epochs to train with
epoch_n = 200 ;

% Define the type of membership function
%mf_type = 'gbellmf';
mf_type = 'gaussmf';
%mf_type = 'trapmf';
%mf_type = 'gauss2mf';
%mf_type = 'trimf';

% Generate FIS structure
disp ( ' Before genfis1')
in_fismat = genfis1(smaData_train, mf_n, mf_type);

% Start the clock to track time for training
tic
%
disp ( ' Before training starts')
% Start training data with ANFIS
[trn_out_fismat trn_error step_size chk_out_fismat chk_error] = ...
    anfis(smaData_train, in_fismat, [epoch_n nan ss ss_dec_rate
ss_inc_rate], ...
    [1,1,1,1], smaData_chk);

% End the clock that tracks training
train_time = toc ;
train_time = train_time / 60.0;      % Convert execution time to
minutes

% Get filename if all plots are needed to be saved
FIS_filename = input('Enter filename to save FIS : ','s');
FIS_filename

disp(['Total training time = ',           num2str(train_time)]);
disp(['Number of data points = ',
num2str(size(smaData_train,1))]);
disp(['Minimum RMSE of training data = ', num2str(min(trn_error))]);
disp(['Minimum RMSE of checking data = ', num2str(min(chk_error))]);
disp(' ');

% Set label and show FIS surface
% FIS structure of original data
```

```
in_fismat.input(1).name = 'Strain (%)  ';
in_fismat.input(2).name = 'Strain rate (1/sec) ';
in_fismat.input(3).name = 'Prestress (MPa) ';
in_fismat.output(1).name = 'Stress (MPa) ';
% FIS structure of training data
trn_out_fismat.input(1).name = 'Strain (%) ';
trn_out_fismat.input(2).name = 'Strain rate (1/sec) ';
trn_out_fismat.input(3).name = 'Prestress (MPa)';
trn_out_fismat.output(1).name = 'Stress (MPa) ';
% FIS structure of checking data
chk_out_fismat.input(1).name = 'Strain ';
chk_out_fismat.input(2).name = 'Strain rate';
chk_out_fismat.input(3).name = 'Prestress (MPa) ';
chk_out_fismat.output(1).name = 'Stress (MPa)';

trn_out_surf = figure;
% Generate a surface plot of the output versus two of the inputs.
gensurf(trn_out_fismat,[2 3],1);
gensurf(trn_out_fismat,[1 2],1);
title({'FIS Structure from Training Data',...
      ['(',FIS_filename,'.fig)']},'Interpreter','none');

%trn_out_surf_disp = figure;
%gensurf(trn_out_fismat,[1 3],1);
%gensurf(trn_out_fismat,[1 3],1);
%title({'FIS Structure from Training Data with Displacement',...
%      ['(',FIS_filename,'.fig)']},'Interpreter','none');

chk_out_surf = figure;
gensurf(chk_out_fismat,[2 3],1);
gensurf(chk_out_fismat,[1 2],1);
title('FIS Structure from Checking Data');

% Plot MFs before and after training
MF_Fig = figure;
orient portrait;
set(gcf, 'PaperPosition', [0.25 0.25 8.00 10.50]);
subplot(3,2,1)
plotmf(in_fismat, 'input', 1);
subplot(3,2,3)
plotmf(in_fismat, 'input', 2);
subplot(3,2,5)
plotmf(in_fismat, 'input', 3);
subplot(3,2,2)
plotmf(trn_out_fismat, 'input', 1);
subplot(3,2,4)
plotmf(trn_out_fismat, 'input', 2);
subplot(3,2,6)
plotmf(trn_out_fismat, 'input', 3);
delete(findobj(gcf, 'type', 'text'));
subplot(321);
title('Initial MFs for strain');
```

```
subplot(322);
title('Final MFs for strain');
subplot(323);
title('Initial MFs on strain rate');
subplot(324);
title('Final MFs on strain rate');
subplot(325);
title('Initial MFs on prestress');
subplot(326);
title('Final MFs on prestress');

% Plot RMSE of Training and Checking Data
% Compute error of FIS on training and checking data
trn_out_anfis = evalfis(smaData_train(:,1:3), trn_out_fismat);
%trn_out_anfis = evalfis(smaData_train(:,1:2), trn_out_fismat);
figure
plot(trn_out_anfis);
title('Show stress from ANFIS', 'FontSize', 12);

% Error from prediction
start_ind = 1;
end_ind   = size(smaData_train,1);
ind_mat   = [1:end_ind]';
% Compute the difference between the force of the experiment and the
force
% of the FIS.
%smaData_train_error = abs(smaData_train(:,3))-abs(trn_out_anfis);
smaData_train_error = abs(smaData_train(:,4)-trn_out_anfis);
%clear start_time start_ind end_time end_ind;

RMSE_Fig = figure;
orient landscape;
set(gcf, 'PaperPosition', [0.25 0.25 10.50 8.00]);
subplot(4,1,1)
plot(step_size);
grid
xlabel('Epochs', 'FontSize', 12);
ylabel('Step Size', 'FontSize', 12);
title({'Results of Training with ANFIS', ['(ss=',num2str(ss),',
ss_dec_rate=',...
       num2str(ss_dec_rate),', ss_inc_rate
=',num2str(ss_inc_rate),')']},'Interpreter',...
       'None', 'FontSize', 12);
subplot(4,1,2)
plot(trn_error);
grid
xlabel('Epoch', 'FontSize', 12);
ylabel('RMSE', 'FontSize', 12);
subplot(4,1,3)
%plot(time_ind, mrdata_train(plot_ind,4), 'r', time_ind,
trn_out_anfis(plot_ind), 'g');
```

```
%%%plot(ind_mat, mrdata_train(:,4), 'b--', ind_mat, trn_out_anfis,
'r');
%plot(ind_mat, smaData_train(:,3), 'b--', ind_mat, trn_out_anfis, 'r');
plot(ind_mat, smaData_train(:,4), 'b--', ind_mat, trn_out_anfis, 'r');
%axis([0 ind_mat(end) 0 0.4]);
legend('Training Data', 'ANFIS Prediction');
grid;
%title('Plot of Time History and ANFIS Prediction of Training Data');
%xlabel('Time (sec)');
ylabel('Stress (MPa)', 'FontSize', 12);
subplot(4,1,4)
plot(ind_mat,smaData_train_error);
%axis([0 ind_mat(end) 0 0.2]);
grid;
xlabel('Data Points', 'FontSize', 12);
ylabel('Error (MPa)', 'FontSize', 12);

inputs_out = figure(6);
subplot (4,1,1);
plot (ind_mat, smaData_train(:,1));
grid;
xlabel('Data Points','FontSize',12);
ylabel('Strain(%)', 'FontSize', 12);
subplot(4,1,2);
plot (ind_mat, smaData_train(:,2));
grid;
xlabel('DataPoints', 'FontSize',12);
ylabel('Strain rate', 'FontSize',12);
subplot(4,1,3);
plot(ind_mat, smaData_train(:,3));
grid;
xlabel('Data Points', 'FontSize', 12);
ylabel('Prestress (MPa)', 'FontSize',12);
subplot(4,1,4);
plot(ind_mat, smaData_train(:,4), 'b--', ind_mat, trn_out_anfis, 'r');
%axis([0 ind_mat(end) 0 0.4]);
legend('Training Data', 'ANFIS Prediction');
grid;
title('Plot of Time History and ANFIS Prediction of Training Data');
%xlabel('Time (sec)');
ylabel('Stress (MPa)', 'FontSize', 12);

inputs_out = figure(7);
subplot (4,1,1);
plot (ind_mat, smaData_train(:,1));
grid;
xlabel('Data Points','FontSize',12);
ylabel('Strain(%)', 'FontSize', 12);
subplot(4,1,2);
plot (ind_mat, smaData_train(:,2));
grid;
xlabel('DataPoints', 'FontSize',12);
```

```
ylabel('Strain rate', 'FontSize',12);
subplot(4,1,3);
plot(ind_mat, smaData_train(:,3));
grid;
xlabel('Data Points', 'FontSize', 12);
ylabel('Prestress (MPa)', 'FontSize',12);
subplot(4,1,4)
plot(ind_mat, smaData_train(:,3));
grid;
xlabel('Data Points', 'FontSize', 12);
ylabel('Prestress (MPa)', 'FontSize',12);



max_trn_error = max(smaData_train_error);
min_trn_error = min(smaData_train_error);
disp(['Maximum Error = ', num2str(max_trn_error),' N']);
disp(['Minimum Error = ', num2str(min_trn_error),' N']);

% Check if data are to be saved
if ~isempty(FIS_filename)
    cd 'ANFIS Data';
    writefis (trn_out_fismat, FIS_filename);
    hgsave (MF_Fig,   [FIS_filename,'_MF.fig']);
    hgsave (RMSE_Fig, [FIS_filename,'_trn_RMSE.fig']);
    hgsave (inputs_out, [FIS_filename,'_inputs_out.fig']);
    cd ..
    disp(' ');
    disp(['FIS has been saved to ".\ANFIS Data\',FIS_filename,'.fis']);
    clear MF_Fig
end

% Save some training parameters for future reference
save(['.\ANFIS Data\',FIS_filename,'_info.mat'], 'epoch_n', 'ss',
'ss_dec_rate',...
      'ss_inc_rate', 'trn_error', 'step_size', 'chk_error',
'train_time',...
      'max_trn_error', 'min_trn_error');
```

## M6

```
%-------------------------------------------------------------------------
-
% Filename : smaValidVel2.m
% MATLAB M-file to verify FIS for Univ. of Chile shape memory alloy.
%   trained from smaTrain.m
%
% Original code written by  Prof. Paul N. Roschke

%-------------------------------------------------------------------------
-
```

```
% Starting and ending points (see below) are in seconds, not the number
of the data point.
% Purpose:  concatenate portions of one or more data files.

%  Data in the input file from Univ. of Chile is in columns in the
following order:
%   Col. 1  Displacement  (mm)
%   Col. 2  Force (kg-f)  -  measured by the MTS actuator load cell
%%%%%%%   Col. 3  Prestress force (kg-f) - from Gloria's thesis
%  Specify the increment of time between data points
%dt = 0.0005 ;

% close all;
%clear;
%clc;
% Load information of dt from the file saved from exData.m
% load('.\ANFIS Data\step_train.mat');
%  Temporary:

%  Define an empty matrix that will contain the validation data
%smaData_valid = [ ] ;

%  Load the validation data that was saved by
%  BuildTrainCheckValidData_simple.m and grafdiag_mod.m
load sma_anfis_valid sma_anfis_valid

% FISname = try1gauss222coef226ss0_13_e200

%  Use selected data points for verifying (i.e. not necessarily all)
%interval = 8;

% Determine the number of data points in the validation data set
nDataPoints = size(sma_anfis_valid,1);


%time  = mrdata_val(:,5);
%dt_valid = time(2) - time(1) ;
% Temporarily assign a time step size in order to run Simulink
dt_valid = 0.0005;
% Make up a time vector:
time_max = (nDataPoints-1)*dt_valid ;
time = [0:dt_valid:time_max]';

%%Offset the second input (previous displacement) by n_offset time
% increments as the second input variable.
%n_offset = 50;
%n_offset = 40;

% Set up training arrays.  Use every other data point.
% Load the first column of the input dataset into the displacement
disp = [sma_anfis_valid(1:1:size(sma_anfis_valid,1), 1)] ;
```

```
% Calculate the velocity
tc_or_valid = 1 ;
%[velocity] = vel_calc (disp, dt_valid, tc_or_valid)
velocity = sma_anfis_valid(1:1:size(sma_anfis_valid,1), 2)

% Convert to strain-strain rate
wirelength = 400 ; %mm
wirearea = 1.963*10^-7 ; %m^2

coef_s = 10^-2 ;
coef_sr = 10^-2 ;
coef_stress = 10^6/9.806 ;

length_strain = wirelength*coef_s ;
length_strate = wirelength*coef_sr ;
area = wirearea*coef_stress ;

strain = disp/length_strain ;
strainrate = velocity/length_strate ;

% Begin the 'prev_disp' with zeros and follow with the values from the
% input dataset; put in column format
%prev_disp = [zeros(n_offset,1); disp(1:1:size(disp)-n_offset, 1)] ;
% Load the force vector from the 2nd column in the input dataset.
% Load the prestress force vector from the 3rd column in the input
dataset.
preforce = sma_anfis_valid(1:1:size(sma_anfis_valid,1), 3) ;
force = sma_anfis_valid(1:1:size(sma_anfis_valid,1), 4) ;
%Convert to stress
prestress = preforce/area ;
stress = force/area ;

% prestress = prestress + 0.2;

% Set up input and output matrix for verification
Exp_input = [strain strainrate prestress ];
%Exp_output = [force];

%  Load the fuzzy MR file into the workspace
FISname = input('Enter FIS to be loaded : ', 's');
%SMAfis = readfis ( ['.\ANFIS Data\', FISname, '.fis'] );
SMAfis = readfis ( ['ANFIS Data\', FISname, '.fis'] );
% Get limits of saturation for fuzzy model of the MR damper
% These are equivalent to the extreme limits of the universes of
discourse
%    for the three input variables, displacement (cm), velocity
(cm/sec), and voltage (V).
sat_limits          = getfis(SMAfis, 'inrange');
disp_sat_min        = sat_limits(1, 1);
disp_sat_max        = sat_limits(1, 2);
velocity_sat_min    = sat_limits(2, 1);
```

```
velocity_sat_max    = sat_limits(2, 2);
prestress_sat_min   = sat_limits(3, 1);
prestress_sat_max   = sat_limits(3, 2);
% Replace the saturation of prestress to be between 0 and 10 kg-f
%prestress_sat_min = 2.0;
%prestress_sat_max = 4.0;

% Reset label of input and output
SMAfis.input(1).name = 'Strain';
SMAfis.input(2).name = 'Strain rate';
SMAfis.input(3).name = 'Prestress';
SMAfis.output(1).name = 'Stress';

%  Define starting time for Simulink simulation
%time_start = sma_data_val(1,5);
%  Define ending time for Simulink simulation
%time_end = mrdata_val(end,5);
% Temporarily guess starting and ending times for the validation run in
% Simulink.  Units (sec.)
time_start = 0
time_end   = time_max


%  Start running SIMULINK to evaluate the force of validating data
disp(' ---> SIMULATION: sim_sma_valid_vel2.mdl');
OPTIONS = simset('solver', 'ode5', 'FixedStep', dt_valid);
sim('sim_sma_valid_vel2', [time_start time_end], OPTIONS, [])

% Plot RMSE of Validation Data
% Compute error of FIS on validation data
val_out_fis = sim_out_val;
clear sim_out_val;
%start_time = 0;                          % Starting time of
verification in sec.
%end_time   = time(end);             % Stopping time of verification in
sec.
start_time = time_start;
end_time = time_end;
%  Set up starting and ending indices (number of the data point)
start_ind  = start_time/dt_valid + 1;
end_ind    = round(end_time/dt_valid) + 1;
time_ind = [start_time:dt_valid:end_time];
plot_ind = [start_ind:end_ind];
% Error from prediction
smadata_val_error = abs(stress - val_out_fis);
clear start_time start_ind end_ind;

% Plot validation data
Val_Fig = figure;
orient landscape;
%set(gcf, 'PaperPosition', [0.25 0.25 10.50 8.00]);
```

```
subplot(4,1,1)
plot(time_ind, strain, 'b--');
axis([time_start, time_end+1.0, -1, 1]);
%axis([200, 205, -5000, 5000]);
legend('Validating Strain Data');
grid;
title({'Plot of Time Strain History ',...
       ['(', FISname, '_Val.fig)']}, 'FontSize', 12, 'Interpreter',
'none');
xlabel('Time (sec)');
ylabel('Strain (%)');

subplot(4,1,2)
plot(time_ind, strainrate, 'b--');
axis([time_start, time_end+1.0, -20, 20]);
%axis([200, 205, -5000, 5000]);
legend('Validating Strain Rate Data');
grid;
title({'Plot of Strain Rate Time History ',...
       ['(', FISname, '_Val.fig)']}, 'FontSize', 12, 'Interpreter',
'none');
xlabel('Time (sec)');
ylabel('Strain Rate (1/s)');

subplot(4,1,3)
plot(time_ind, stress, 'b--', time_ind, val_out_fis(plot_ind), 'r');
axis tight
%axis([time_start, time_end+1.0, 0,0.5 ]);
%axis([200, 205, -5000, 5000]);
legend('Validating Data', 'ANFIS Prediction');
grid;
title({'Plot of Time History and ANFIS Prediction of Validating
Data',...
       ['(', FISname, '_Val.fig)']}, 'FontSize', 12, 'Interpreter',
'none');
xlabel('Time (sec)');
ylabel('Stress (MPa)');

subplot(4,1,4)
plot(time_ind, smadata_val_error);
%axis([time_start, time_end+1.0, 0, 0.3]);
%axis([200, 205, -5000, 5000]);
grid;
xlabel('Time (sec)', 'FontSize', 12);
ylabel('Error (MPa)', 'FontSize', 12);

figure
plot(time_ind, stress, 'b--', time_ind, val_out_fis(plot_ind), 'r');
axis tight
%axis([time_start, time_end+1.0, 0,0.5 ]);
%axis([200, 205, -5000, 5000]);
```

```
legend('Validating Data', 'ANFIS Prediction');
grid;
%title({'Plot of Time History and ANFIS Prediction of Validating
Data',...
      % ['(', FISname, '_Val.fig)']}, 'FontSize', 12, 'Interpreter',
'none');
xlabel('Time (sec)');
ylabel('Stress (MPa)');

% PRINT VALIDATION DATA
% Val_data  = figure
% subplot(4,1,1)
% plot(time_ind, strain, 'b--');
% axis tight
% %axis([time_start, time_end+1.0, -1, 1]);
% %axis([200, 205, -5000, 5000]);
% %legend('Validation Data');
% grid;
% xlabel('Time (sec)');
% ylabel('Strain (%)');
%
% subplot(4,1,2)
% plot(time_ind, strainrate, 'b--');
% axis tight
% %axis([time_start, time_end+1.0, -20, 20]);
% %axis([200, 205, -5000, 5000]);
% %legend('Validation Data');
% grid;
% xlabel('Time (sec)');
% ylabel('Strain Rate (1/s)');
%
% subplot(4,1,3)
% plot(time_ind, prestress, 'b--');
% axis([ 0 end_time 50 200])
% %axis([time_start, time_end+1.0, 0,0.5 ]);
% %axis([200, 205, -5000, 5000]);
% %legend('Validation Data');
% grid;
% xlabel('Time (sec)');
% ylabel('Prestress (MPa)');
%
% subplot(4,1,4)
% plot(time_ind, stress, 'b--');
% axis tight
% %axis([time_start, time_end+1.0, 0,0.5 ]);
% %axis([200, 205, -5000, 5000]);
% %legend('Validation Data');
% grid;
% xlabel('Time (sec)');
% ylabel('Stress (MPa)');
```

```
m = 80000
n =90000
figure;
plot(strain(m:n),stress(m:n));
xlabel ('Strain (%)');
ylabel ('Stress (MPa)');
title ('Experimental Results');
%axis ( [-1 1.5 0 0.30]);
grid
hold on
plot(strain(m:n), val_out_fis(m:n),'--m');
xlabel ('Strain (%)');
ylabel ('Stress (MPa)');
title ('FIS Prediction');
%axis ( [-1 1.5 0 0.30]);
grid

figure;
plot(strainrate(m:n),stress(m:n));
xlabel ('Strain (%)');
ylabel ('Stress (MPa)');
title ('Experimental Results');
%axis ( [-1 1.5 0 0.30]);
grid
hold on
plot(strainrate(m:n), val_out_fis(m:n),'--m');
xlabel ('Strain (%)');
ylabel ('Stress (MPa)');
title ('FIS Prediction');
%axis ( [-1 1.5 0 0.30]);
grid

% Save validating figure to "Valid Data" folder
hgsave(Val_Fig, ['.\Valid Data\', FISname,'_Val.fig']);
%clear Val_Fig

% Show statistical data
disp(' ');
input('Press Enter to proceed statistical analysis : ');
val_max_error = max(smadata_val_error);
val_min_error = min(smadata_val_error);

F_exp = stress;
F_fuz = val_out_fis;
totaltime = time (end) ;

eta_t = sqrt(sum((F_exp - F_fuz).^2));
eta_strate = sqrt(sum((F_exp - F_fuz).^2.*abs(strainrate)));
eta_ps = sqrt(sum((F_exp - F_fuz).^2.*abs(prestress)));
```

```
mu_F = mean(F_fuz);

sigma_F = sqrt(sum((F_exp - mu_F).^2));

% Error measured between the fuzzy predicted force and the validation
data
E_t = eta_t/sigma_F;    % as a function of time
E_strate = eta_strate/sigma_F;  % as a function of displacement
E_ps = eta_ps/sigma_F;  % as a function of prestress force
Errors = [E_t E_strate E_ps];
%Errors = [E_t E_strate];
%Errors = [E_t E_x];

save(['.\Valid Data\', FISname,'_Stats.mat'], 'val_max_error',
'val_min_error',...
     'Exp_input', 'F_exp', 'F_fuz', 'Errors', 'mu_F', 'sigma_F');
clc;
disp(['Statistical data for validation of ', FISname]);
disp(' ');
disp(['Maximum Error = ', num2str(val_max_error),' kN']);
disp(['Minimum Error = ', num2str(val_min_error),' kN']);
disp(['E_t      = ', num2str(E_t)]);
disp(['E_strate      = ', num2str(E_strate)]);
disp(['E_ps      = ', num2str(E_ps)]);
disp(['mu_F     = ', num2str(mu_F)]);
disp(['sigma_F  = ', num2str(sigma_F)]);
disp(' ');
```

## SECTION 6

## M7

```
%%%%%%%%%%%%%%%%%%%%%%
% SDOFfree_allscript.m
%%%%%%%%%%%%%%%%%%%%%%

clear all
close all

addpath 'C:\Documents and Settings\oeo2452\My Documents\Shape Memory
Alloys\SMA Fuzzy Model\Special Pro\Earthquakes'

% System Parameters
%k = 1783;  %kN/m
%c = 5588.9e-003; %kN-s/m
k = 10898 % kN/m
c = 25.5711*2
m = 6; %kN-s^2/m
```

```
% Wire Characteristics
cos_teta = cos(pi/4);
wirearea = 1.963*10^-7 ; %m^2
Esma = 1.366e7 %kN/m2
coef_s = 10^2 ;
coef_sr = 10^2 ;
coef_stress = 10^2;

intdis = 0.008 %m
num = 500;
len = 0.5;

% Define stiffness of steel as same as SMA brace stiffness
ksteel = Esma*wirearea/len*num % =ksma
kadd = ksteel*cos(pi/4)^2;

ks = k + kadd   ;  %kN/m

wn = (k/m)^0.5
zeta = c / (2*m*wn)

wns = (ks/m)^0.5
zetas = c / (2*m*wns)

% ABCD
A= [0,1;-k/m,-c/m];
A2 = [0,1;-ks/m,-c/m];

B= [0;1/m];
C= [eye(2);-k/m,-c/m];
C2= [eye(2);-ks/m,-c/m];
D= [0; 0; 1/m];

% 1st Floor Wires
prestress = 0.1412; %GPa
num_wire = num;
kbar = num_wire*cos_teta;

wirelength = len; %m

length_strain = coef_s/wirelength ;
length_strate = coef_sr/wirelength ;
force_coef = wirearea*10^9*10^-3;

% SMA Fuzzy Prediction
FISname = 'try1gauss222ss0_08coef229'
SMAfis = readfis ( [ FISname, '.fis'] )

sat_limits          = getfis(SMAfis, 'inrange');
strain_sat_min      = sat_limits(1, 1);
```

```
strain_sat_max      = sat_limits(1, 2);
strate_sat_min      = sat_limits(2, 1);
strate_sat_max      = sat_limits(2, 2);
prestress_sat_min   = sat_limits(3, 1);
prestress_sat_max   = sat_limits(3, 2);


% Simulation Options
dt_valid = 0.001;
time_start = 0;
time_end = 2;
time = [time_start:dt_valid:time_end];


OPTIONS = simset('solver', 'ode5', 'FixedStep', dt_valid);
sim('SDOFfree_all', [time_start time_end], OPTIONS, [])

%  Override the default settings for graphs in Matlab
set(0, 'DefaultAxesFontSize', 12)
set(0, 'DefaultAxesFontWeight', 'bold')
set(0, 'DefaultAxesLineWidth', 2)
set(0, 'DefaultLineLineWidth', 2)

figure
plot(time,cdisp);
ylabel ('Displacement (mm)')
xlabel ('Time (sec)')
grid on
%title ('Displacement-Time History')
hold on
plot(time,sdisp2,'-g');
hold on
plot(time,undisp,'-r');
legend ('SMA Frame', 'Steel Frame', 'Bare Frame')

figure
plot(time,cacc);
ylabel ('Acceleration (m/s2)')
xlabel ('Time (sec)')
grid on
%title ('Acceleration-Time History')
hold on
plot(time, sacc2,'-g');
hold on
plot(time,unacc,'-r')
legend ('SMA Frame','Steel Frame', 'Bare Frame')

wdisp = len*strain*10;
Fall = F*num_wire;

figure
```

```
plot(strain,stress);
%title ('Stress-Strain Diagram')
grid on
xlabel('Strain (%)');
ylabel('Stress(Mpa)');

figure
plot(cdisp,cvel);
%title ('Phase Diagram')
grid on
xlabel('Displacement');
ylabel('Velocity');

figure
plot(strain,strate);
%title ('Phase Diagram')
grid on
xlabel('Strain (%)');
ylabel('Strain rate (%/sec)');

% figure
% plot(cdisp,Ft);
% title ('Force-Displacement Diagram')
% grid on
% ylabel('Force (kN)');
% xlabel('Displacement (mm))');
```

## M8

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SDOF_sine_all_Vfreqscript.m
% Harmonic Excitation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all

addpath 'C:\Documents and Settings\oeo2452\My Documents\Shape Memory
Alloys\SMA Fuzzy Model\Special Pro\Earthquakes'

% System Parameters
% k = 1783;  %kN/m
% c = 5588.9e-003; %kN-s/m
k = 10898 % kN/m
c = 25.5711*0.2
m = 6; %kN-s^2/m

% Wire Characteristics
cos_teta = cos(pi/4);
wirearea = 1.963*10^-7 ; %m^2
Esma = 1.366e7 %kN/m2
coef_s = 10^2 ;
```

```
coef_sr = 10^2 ;
coef_stress = 10^2;

num = 500;
len = 0.5;

ksteel = Esma*wirearea/len*num % =ksma
kadd = ksteel*cos(pi/4)^2;

ks = k + kadd   ;  %kN/m

p0= 11 %kN
omega_n = (k/m)^0.5
zeta = c/(2*m*omega_n)
Tn = 2*pi/omega_n % sec

% Rd = 1 / (((1-n^2)^2+(2*zeta*n)^2))^0.5

% ABCD
A= [0,1;-k/m,-c/m];
A2 = [0,1;-ks/m,-c/m];

B= [0;1/m];
C= [eye(2);-k/m,-c/m];
C2= [eye(2);-ks/m,-c/m];
D= [0; 0; 1/m];


% 1st Floor Wires
prestress = 0.1412; %GPa
num_wire = num;
kbar = num_wire*cos_teta;

wirelength = len; %m

length_strain = coef_s/wirelength ;
length_strate = coef_sr/wirelength ;
force_coef = wirearea*10^9*10^-3;

% SMA Fuzzy Prediction
FISname = 'try1gauss222ss0_08coef229'
SMAfis = readfis ( [ FISname, '.fis'] )

sat_limits          = getfis(SMAfis, 'inrange');
strain_sat_min      = sat_limits(1, 1);
strain_sat_max      = sat_limits(1, 2);
strate_sat_min      = sat_limits(2, 1);
strate_sat_max      = sat_limits(2, 2);
prestress_sat_min   = sat_limits(3, 1);
prestress_sat_max   = sat_limits(3, 2);
```

```
% Simulation Options
dt_valid = 0.001;
time_start = 0;
time_end = 5;
time = [time_start:dt_valid:time_end];

number = 60;
omega_array = zeros(1,number);
uncdisp_max = zeros(1,number);
cdisp_max = zeros(1,number);
sdisp2_max = zeros(1,number);


Rd = zeros(1,number);
for n=1:number;
m = 3;
omega = m/number*n*omega_n
f= m/number*n;
Rd(1,n) = 1 / (((1-f^2)^2+(2*zeta*f)^2))^0.5;
OPTIONS = simset('solver', 'ode5', 'FixedStep', dt_valid);
sim('SDOF_sine_all', [time_start time_end], OPTIONS, [])
uncdisp_max(1,n) = max(abs(undisp));
cdisp_max(1,n) = max(abs(cdisp));
sdisp2_max(1,n) = max(abs(sdisp2));
%accel
uncacc_max(1,n) = max(abs(unacc));
cacc_max(1,n) = max(abs(cacc));
sacc2_max(1,n) = max(abs(sacc2));
omega_array(1,n) = omega/omega_n;

if n == 10
omega
figure
plot(time,cdisp);
ylabel ('Displacement (mm)')
xlabel ('Time (sec)')
title ('Displacement-Time History')
hold on
plot(time,sdisp2,'-g');
hold on
plot(time,undisp,'-r');
legend ('SMA Frame', 'Steel Frame', 'Bare Frame')

figure
plot(time,cacc);
ylabel ('Acceleration (m/s2)')
xlabel ('Time (sec)')
title ('Acceleration-Time History')
hold on
```

```
plot(time, sacc2,'-r');
hold on
plot(time,unacc,'-g')
legend ('SMA Frame','Steel Frame', 'Bare Frame')

wdisp = len*strain*10;
Fall = F*num_wire;

figure
plot(strain,stress);
title ('Stress-Strain Diagram')
grid on
xlabel('Strain (%)');
ylabel('Stress(Mpa)');

figure
plot(strain,strate);
title ('Phase Diagram')
grid on
xlabel('Displacement');
ylabel('Velocity');

end

if n == 11
omega11 = omega

figure
plot(strain,stress);
title ('Stress-Strain Diagram')
grid on
xlabel('Strain (%)');
ylabel('Stress(Mpa)');

figure
plot(strain,strate);
title ('Phase Diagram')
grid on
xlabel('Displacement');
ylabel('Velocity');

end
end

ust = p0/k*1000;

%  Override the default settings for graphs in Matlab
set(0, 'DefaultAxesFontSize', 12)
set(0, 'DefaultAxesFontWeight', 'bold')
set(0, 'DefaultAxesLineWidth', 2)
set(0, 'DefaultLineLineWidth', 2)
```

```
figure
plot (omega_array, uncdisp_max);
%title ('Frequency-Response Curve')
xlabel ('w / w_n')
ylabel ('Maximum Displacement')
grid on
hold on
plot (omega_array, cdisp_max, 'r--')
hold on
plot (omega_array, Rd*ust,'y')
hold on
plot(omega_array, sdisp2_max, 'k')

figure
semilogy (omega_array, uncdisp_max,'r--');
%title ('Frequency-Response Curve')
xlabel ('w / w_n')
ylabel ('Maximum Displacement (mm)')
grid on
hold on
semilogy (omega_array, cdisp_max )
hold on
semilogy(omega_array, sdisp2_max, 'g')
axis tight
legend ('Bare Frame','SMA Frame','Steel Frame')

figure
semilogy (omega_array*omega_n, uncdisp_max, 'g--')
hold on
semilogy (omega_array*omega_n, cdisp_max,'b')
hold on
semilogy(omega_array*omega_n, sdisp2_max, 'r')
axis tight
%title ('Frequency-Response Curve')
xlabel ('w')
ylabel ('Maximum Displacement')
legend ('Bare Frame','SMA Frame','Steel Frame')
grid on


figure
semilogy (omega_array, uncacc_max, 'r--')
hold on
semilogy (omega_array, cacc_max,'b')
hold on
semilogy(omega_array, sacc2_max, 'g')
axis tight
%title ('Frequency-Response Curve')
xlabel ('w/w_n')
ylabel ('Maximum Acceleration (m/s2)')
legend ('Bare Frame','SMA Frame','Steel Frame')
```

```
grid on

max_disp = [max(uncdisp_max); max(cdisp_max);max(sdisp2_max)]
max_acc = [max(uncacc_max);max(cacc_max);max(sacc2_max)]
```

## M9

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SDOFearth_allscript.m
% Earthquake Excitation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all

addpath 'C:\Documents and Settings\oeo2452\My Documents\OSMAN\SMAs\SMA
Fuzzy Model\Special Pro\Earthquakes'
load elcentro

% System Parameters
% k = 1783;  %kN/m
% c = 5588.9e-003; %kN-s/m
k = 10898 % kN/m
c = 25.5711*0.4
m = 6; %kN-s^2/m

% Wire Characteristics
cos_teta = cos(pi/4);
wirearea = 1.963*10^-7 ; %m^2
Esma = 1.366e7 %kN/m2
coef_s = 10^2 ;
coef_sr = 10^2 ;
coef_stress = 10^2;

% Number and length of SMA wires
num = 300;
len = 0.5;

ksteel = Esma*wirearea/len*num % =ksma
kadd = ksteel*cos(pi/4)^2;

ks = k + kadd   ;  %kN/m

wn = (k/m)^0.5
zeta = c / (2*m*wn)
Tn = 2*pi/wn

wns = (ks/m)^0.5
zetas = c / (2*m*wns)

% ABCD
A= [0,1;-k/m,-c/m];
```

```
A2 = [0,1;-ks/m,-c/m];

B= [0;1/m];
C= [eye(2);-k/m,-c/m];
C2= [eye(2);-ks/m,-c/m];
D= [0; 0; 1/m];

% 1st Floor Wires
prestress = 0.1412; %GPa
num_wire = num;
kbar = num_wire*cos_teta;

wirelength = len; %m

length_strain = coef_s/wirelength ;
length_strate = coef_sr/wirelength ;
force_coef = wirearea*10^9*10^-3;

g = 9.81;
% Eathquake Data
load kobe
%load northrdg
t = k(1,:)';
u = k(2,:)'/g;
u = u/abs(min(u))*1.5;

% load AccData2.mat;
% t = AccData(:,1);
% u = AccData(:,2)*2;
figure
plot(t,u, 'b-')
grid off
xlabel('Time (sec)')
ylabel('Acceleration (g)')
axis tight
grid on
%set(gca, 'XTick', [1 2 3 4 5 6 7 8 9 10 11.2]);
%title ('Earthquake Time-History')


% SMA Fuzzy Prediction
FISname = 'try1gauss222ss0_08coef229'
SMAfis = readfis ( [ FISname, '.fis'] )

sat_limits          = getfis(SMAfis, 'inrange');
strain_sat_min      = sat_limits(1, 1);
strain_sat_max      = sat_limits(1, 2);
strate_sat_min      = sat_limits(2, 1);
strate_sat_max      = sat_limits(2, 2);
prestress_sat_min   = sat_limits(3, 1);
prestress_sat_max   = sat_limits(3, 2);
```

```
% Simulation Options
dt_valid = 0.001;
time_start = 0;
time_end = 30;
time = [time_start:dt_valid:time_end];


OPTIONS = simset('solver', 'ode5', 'FixedStep', dt_valid);
sim('SDOFearth_all', [time_start time_end], OPTIONS, [])

%  Override the default settings for graphs in Matlab
set(0, 'DefaultAxesFontSize', 12)
set(0, 'DefaultAxesFontWeight', 'bold')
set(0, 'DefaultAxesLineWidth', 1.5)
set(0, 'DefaultLineLineWidth', 2)

figure
plot(time,cdisp,'b');
ylabel ('Displacement (mm)')
xlabel ('Time (sec)')
%title ('Displacement-Time History')
hold on
plot(time,sdisp2,'--g');
hold on
plot(time,undisp,':r');
legend ('SMA Frame','Steel Frame', 'Bare Frame')
axis tight
grid on

figure
plot(time,cacc);
hold on
ylabel ('Acceleration (m/s2)')
xlabel ('Time (sec)')
%title ('Acceleration-Time History')
hold on
plot(time, sacc2,'--g');
hold on
plot(time,unacc,':r')
legend ('SMA Frame','Steel Frame', 'Bare Frame')
axis tight
grid on

%stress = F/wirearea*1e-6; %MPa
wdisp = len*strain*10;
F2 = F*num_wire;


figure
```

```
plot(strain,stress);
%title ('Stress-Strain Diagram')
grid on
xlabel('Strain (%)');
ylabel('Stress (MPa)');
grid on


max_disp = [max(undisp); max(cdisp);max(sdisp2)]
max_acc = [max(unacc);max(cacc);max(sacc2)]

dis_rate = [0.40    0.34    0.31    0.29    0.26];
acc_rate = [0.30    0.33    0.35    0.37    0.37];
sdis_rate = [0.14 0.14 0.14 0.14 0.14];
sacc_rate = [0.10 0.10 0.10 0.10 0.10];
peak_acc = [0.4 0.8 1 1.2 1.5];

figure
plot(peak_acc, dis_rate,'-rs')
hold on
plot(peak_acc, acc_rate,'-ro')
hold on
plot(peak_acc, sdis_rate,'-gs')
hold on
plot(peak_acc, sacc_rate,'-go')
xlabel('Peak Ground Acceleration (g)')
ylabel('Reduction in Response (%)')
legend('Displacement (SMA)','Acceleration (SMA)','Displacement
(Steel)','Acceleration (Steel)')
axis([0.3 1.8 0 0.5])
grid on
```

## M10

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% sma3storeyEQscript2st.m
% Three Story Benchmark Building
% SMA Braced Frame vs. Steel Braced Frame
% Created by Osman E. Ozbulut
% January - May 2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
%% Define excitation


addpath 'C:\Documents and Settings\oeo2452\My Documents\OSMAN\SMAs\SMA
Fuzzy Model\Special Pro\Earthquakes'
```

```
addpath 'C:\Documents and Settings\oeo2452\My Documents\OSMAN\SMAs\SMA
Fuzzy Model\Special Pro\Earthquakes\New Folder'
%load AccData2
g = 9.806;
% %%Eathquake Data
% El Centro
%load elcentro
% t = e(1,:)';
% u = 0.43065*e(2,:)'/9.81; % 0.15g

% % % Northridge
% load northrdg
% t = n(1,:)';
% u = n(2,:)'/g;
% u = (n(2,:)'/g)/max(abs(u))*0.25;

% % % Bolu - EW
load bolu
t = b(:,1);
u = b(:,2);
u = b(:,2)/max(abs(u))*0.30;

% % % Chichi
% load Chichi084
% t = Chi(:,1);
% u = Chi(:,3);
% u = Chi(:,3)/max(abs(u))*0.12;

% % % N Palm Spring -
% load Npalmspr
% t = Npalm(:,1);
% u = Npalm(:,2);
% u = Npalm(:,2)/max(abs(u))*0.40;

% % % Hachinhe
%load hachinhe
% t = h(1,:)';
% u = h(2,:)'/g;
% u = (h(2,:)'/g)/max(abs(u))*0.15;

% % %Kobe
% load kobe2
% t = kobe(1,:)';
% u = kobe(2,:)'/g;
% u = u/max(abs(u))*0.15;

% % % Artifical
% load arteq
% t = arteq(1,:)';
% u = arteq(2,:)'/g';
% u = u/max(abs(u))*0.15;
```

```
plot(t,u, 'b-')
grid off
xlabel('Time (sec)')
ylabel('Acceleration (g)')
%set(gca, 'XTick', [1 2 3 4 5 6 7 8 9 10 11.2]);
%title ('Earthquake Time-History')

%% System Parameters

k1 = 1594.6; %kN/m
k2 = 1037.5; %kN/m
k3 = 2488;   %kN/m
c1 = 5387.6e-003; %kN-s/m
c2 = 8054.9e-003; %kN-s/m
c3 = 6041.1e-003; %kN-s/m
m1 = 6; %kN-s^2/m
m2 = 6; %kN-s^2/m
m3 = 6; %kN-s^2/m

num1 = 349;
num2 = 425;
num3 = 183;
len1 = 1.14;
len2 = 1.20;
len3 = 0.88;

wirearea = 1.963*10^-7 ; %m^2
area1 = num1*wirearea*10^4
area2 = num2*wirearea*10^4
area3 = num3*wirearea*10^4

% ksteel = 2e8*1.963*1e-7/len
% k = ksteel*num
ksma1 = 1.366e7*1.963*1e-7/len1;
ks1 = ksma1*num1
ksma2 = 1.366e7*1.963*1e-7/len2;
ks2 = ksma2*num2
ksma3 = 1.366e7*1.963*1e-7/len3;
ks3 = ksma3*num3

kadd1 = ks1*cos(pi/4)^2
kadd2 = ks2*cos(pi/4)^2
kadd3 = ks3*cos(pi/4)^2

k4 = ks1 + kadd1;  %kN/m
k5 = ks2 + kadd2; %kN/m
k6 = ks3 + kadd3;  %kN/m


% ABCD
```

```
MassVec = [m1,m2,m3];
M = diag(MassVec);

K = [k1+k2 -k2     0;
        -k2     k2+k3 -k3;
         0     -k3     k3];

K2 = [k4+k5 -k5     0;
        -k5     k5+k6 -k6;
         0     -k6     k6];

     C = [c1+c2 -c2      0;
        -c2     c2+c3 -c3;
         0      -c3      c3];
nDOF = 3;

%  Build state space matrices for the building structure.
A = [zeros(nDOF) eye(nDOF); -inv(M)*K -inv(M)*C];
A2 = [zeros(nDOF) eye(nDOF); -inv(M)*K2 -inv(M)*C];
B = [zeros(nDOF); inv(M)];
C1 = [eye(6);-inv(M)*K -inv(M)*C];
C2 = [eye(6);-inv(M)*K2 -inv(M)*C];
D = [zeros(6,3); inv(M)];


% Wire Characteristics
cos_teta = cos(pi/4);
coef_s = 10^2 ;
coef_sr = 10^2 ;
coef_stress = 10^2;
prestress = 0.1412 %GPa

% 1st Floor Wires
prestress1 = prestress; % kg-f/m^2
num_wire1 = num1;
kbar1 = num_wire1*cos_teta;

wirelength1 = len1; %m
length_strain1 = coef_s/wirelength1 ;
length_strate1 = coef_sr/wirelength1 ;
force_coef1 = wirearea*10^9*10^-3;

% 2nd Floor Wires
prestress2 = prestress; % kg-f/m^2
num_wire2 = num2;
kbar2 = num_wire2*cos_teta;

wirelength2 = len2; %m
length_strain2 = coef_s/wirelength2 ;
length_strate2 = coef_sr/wirelength2 ;
force_coef2 = wirearea*10^9*10^-3;
```

```
% 3rd Floor Wires
prestress3 = prestress; % kg-f/m^2
num_wire3 = num3;
kbar3 = num_wire3*cos_teta;

wirelength3 = len3; %m
length_strain3 = coef_s/wirelength3 ;
length_strate3 = coef_sr/wirelength3 ;
force_coef3 = wirearea*10^9*10^-3;

% SMA Fuzzy Prediction
FISname = 'try1gauss222ss0_08coef229'
SMAfis = readfis ( [ FISname, '.fis'] )

sat_limits          = getfis(SMAfis, 'inrange');
strain_sat_min      = sat_limits(1, 1);
strain_sat_max      = sat_limits(1, 2);
strate_sat_min      = sat_limits(2, 1);
strate_sat_max      = sat_limits(2, 2);
prestress_sat_min   = sat_limits(3, 1);
prestress_sat_max   = sat_limits(3, 2);
%% Start Simulation

% Simulation Options
dt_valid = 0.001;
time_start = 0;
time_end = 30;
time = [time_start:dt_valid:time_end];

tic

OPTIONS = simset('solver', 'ode5', 'FixedStep', dt_valid);
sim('sma3storeyEQsim2absacc', [time_start time_end], OPTIONS, [])
toc

te = toc/60
%% Plots
% Set up for plots
%  Override the default settings for graphs in Matlab
set(0, 'DefaultAxesFontSize', 13)
set(0, 'DefaultAxesFontWeight', 'bold')
set(0, 'DefaultAxesLineWidth', 2)
%%%%%%%%DISPLACEMENTS%%%%%%%%%%%%%%%%
disp_1 = disp1*1000;      %mm
disp_2 = disp2*1000;
disp_3 = disp3*1000;

undisp_1 = undisp1*1000;
undisp_2 = undisp2*1000;
undisp_3 = undisp3*1000;
```

```
undisp_4 = undisp4*1000;
undisp_5 = undisp5*1000;
undisp_6 = undisp6*1000;

max_disp1 = max(abs(min(disp_1)),max(disp_1));
max_disp2 = max(abs(min(disp_2)),max(disp_2));
max_disp3 = max(abs(min(disp_3)),max(disp_3));

max_undisp1 = max(abs(min(undisp_1)),max(undisp_1));
max_undisp2 = max(abs(min(undisp_2)),max(undisp_2));
max_undisp3 = max(abs(min(undisp_3)),max(undisp_3));

max_undisp4 = max(abs(min(undisp_4)),max(undisp_4));
max_undisp5 = max(abs(min(undisp_5)),max(undisp_5));
max_undisp6 = max(abs(min(undisp_6)),max(undisp_6));

max_disp = [0,max_disp1,max_disp2,max_disp3];
max_undisp = [0,max_undisp1,max_undisp2,max_undisp3];
max_sdisp = [0,max_undisp4,max_undisp5,max_undisp6];

storey = [0,1,2,3];

%%%%%%%%%%%%ACCELERATIONS%%%%%%%%%%%%%%%%%%%%%%
unacc_1 = unacc1;
unacc_2 = unacc2;
unacc_3 = unacc3;

unacc_4 = unacc4;
unacc_5 = unacc5;
unacc_6 = unacc6;

acc_1 = acc1;
acc_2 = acc2;
acc_3 = acc3;

max_acc1 =  max(abs(min(acc_1)),max(acc_1));
max_acc2 =  max(abs(min(acc_2)),max(acc_2));
max_acc3 =  max(abs(min(acc_3)),max(acc_3));

max_unacc1 =  max(abs(min(unacc_1)),max(unacc_1));
max_unacc2 =  max(abs(min(unacc_2)),max(unacc_2));
max_unacc3 =  max(abs(min(unacc_3)),max(unacc_3));

max_unacc4 =  max(abs(min(unacc_4)),max(unacc_4));
max_unacc5 =  max(abs(min(unacc_5)),max(unacc_5));
max_unacc6 =  max(abs(min(unacc_6)),max(unacc_6));

max_acc = [0,max_acc1, max_acc2, max_acc3];
max_unacc = [0,max_unacc1, max_unacc2, max_unacc3];
```

```
max_sacc = [0,max_unacc4, max_unacc5, max_unacc6];

J2sma = max(max_acc)/max(max_unacc);
J2st = max(max_sacc)/max(max_unacc);

%%%%%%%%%%%%%%%% DRIFTS %%%%%%%%%%%%%%%%%%%%
delta_2 = delta2;
delta_3 = delta3;

undelta_3 = undisp_3 - undisp_2;
undelta_2 = undisp_2 - undisp_1;

undelta_5 = undisp_5 - undisp_4;
undelta_6 = undisp_6 - undisp_5;

max_delta2 = max(abs(min(delta_2)), max(delta_2));
max_delta3 = max(abs(min(delta_3)), max(delta_3));

max_undelta2 = max(abs(min(undelta_2)), max(undelta_2));
max_undelta3 = max(abs(min(undelta_3)), max(undelta_3));

max_undelta5 = max(abs(min(undelta_5)), max(undelta_5));
max_undelta6 = max(abs(min(undelta_6)), max(undelta_6));

max_delta = [0,max_disp1, max_delta2, max_delta3];
max_undelta = [0,max_undisp1, max_undelta2, max_undelta3];
max_sdelta = [0,max_undisp4, max_undelta5, max_undelta6];

J1sma = max(max_delta)/max(max_undelta);
J1st = max(max_sdelta)/max(max_undelta);

%%%%%%%%%%%%%%%%%%% RMS_DISP %%%%%%%%%%%%%%%%
rmsdisp1 = rms(disp1);
rmsdisp2 = rms(disp2);
rmsdisp3 = rms(disp3);
max_rms_disp = max(rmsdisp1,max(rmsdisp2,rmsdisp3));

rmsundisp1 = rms(undisp1);
rmsundisp2 = rms(undisp2);
rmsundisp3 = rms(undisp3);
max_rms_undisp = max(rmsundisp1,max(rmsundisp2,rmsundisp3));

rmsundisp4 = rms(undisp4);
rmsundisp5 = rms(undisp5);
rmsundisp6 = rms(undisp6);
max_rms_undisp_st = max(rmsundisp4,max(rmsundisp5,rmsundisp6));

max_rms_disp_all = [max_rms_undisp, max_rms_disp, max_rms_undisp_st];

J3sma = max_rms_disp/max_rms_undisp;
```

```
J3st = max_rms_undisp_st/max_rms_undisp;
%%%%%%%%%%%%%%%%%%%%%%%%RMS_ACC%%%%%%%%%%%%%%%%%%%%%%
rmsacc1 = rms(acc1);
rmsacc2 = rms(acc2);
rmsacc3 = rms(acc3);
max_rms_acc = max(rmsacc1,max(rmsacc2,rmsacc3));

rmsunacc1 = rms(unacc1);
rmsunacc2 = rms(unacc2);
rmsunacc3 = rms(unacc3);
max_rms_unacc = max(rmsunacc1,max(rmsunacc2,rmsunacc3));

rmsunacc4 = rms(unacc4);
rmsunacc5 = rms(unacc5);
rmsunacc6 = rms(unacc6);
max_rms_unacc_st = max(rmsunacc4,max(rmsunacc5,rmsunacc6));

max_rms_acc_all = [max_rms_unacc, max_rms_acc, max_rms_unacc_st];

J4sma = max_rms_acc/max_rms_unacc;
J4st = max_rms_unacc_st/max_rms_unacc;

%%%%%%%%%%%%%%%%%%%% J %%%%%%%%%%%%%%%%%%%%
Jsma = [J1sma, J2sma, J3sma, J4sma]
Jst = [J1st, J2st, J3st, J4st]
%%%%%%%%%%%%%%%%%%% DRATIOS %%%%%%%%%%%%%%%

h = 30; %3000mm*100 (%)

dratio1 = max_disp1/h;
dratio2 = max_delta2/h;
dratio3 = max_delta3/h;

un_dratio1 = max_undisp1/h;
un_dratio2 = max_undelta2/h;
un_dratio3 = max_undelta3/h;

un_dratio4 = max_undisp4/h;
un_dratio5 = max_undelta5/h;
un_dratio6 = max_undelta6/h;

dratio = [0, dratio1, dratio2, dratio3];
un_dratio = [0, un_dratio1, un_dratio2, un_dratio3];
sdratio = [0, un_dratio4, un_dratio5, un_dratio6];

%%%%%%%%%%%%%%%%%%%%%%WIRE DISPLACEMENTS %%%%%%%%%%%%%%%%%%%
w_disp1 = disp_1*cos_teta;
w_disp2 = delta_2*cos_teta;
w_disp3 = delta_3*cos_teta;
```

```
F3p_out = F3p*1000;    %N
F3n_out = F3n*1000;    %N

stress3p_out = F3p_out/wirearea*1e-6; %MPa
stress3n_out = F3n_out/wirearea*1e-6;

strain3p = w_disp3/(wirelength3*1000)*100;


%%%%%%%%% PLOTS FOR PAPER %%%%%%%%%%%%%%%%%%%%
figure
plot(max_undisp,storey,'--rs',max_disp,storey,'-d',max_sdisp,storey,'-
og')
legend ('Bare Frame', 'SMA Frame','Steel Frame')
xlabel ('Maximum Displacements (mm)')
ylabel ('Floor')

figure
plot(max_unacc,storey,'--rs',max_acc,storey,'-d',max_sacc,storey,'-og')
legend ('Bare Frame', 'SMA Frame','Steel Frame')
xlabel ('Maximum Accelerations (m/s2)')
ylabel ('Floor')

figure
plot(un_dratio,storey,'--rs',dratio,storey,'-d',sdratio,storey,'-og')
legend ('Bare Frame', 'SMA Frame','Steel Frame')
xlabel ('Maximum Interstory Drifts (%)')
ylabel ('Floor')


figure
plot(time,undelta_3,':r');
ylabel ('Displacement (mm)')
xlabel ('Time (sec)')
%title ('Third Floor Drift Time History')
hold on
plot(time,delta_3,'-')
hold on
plot(time,undelta_6,'-g')
legend ('Bare Frame','SMA Frame','Steel Frame')


figure
plot(time,undelta_2,':r');
ylabel ('Displacement (mm)')
xlabel ('Time (sec)')
%title ('Second Floor Displacement-Time History')
hold on
plot(time,delta_2,'-')
hold on
plot(time,undelta_5,'-g')
```

```
legend ('Bare Frame','SMA Frame','Steel Frame')

figure
plot(time,undisp_1,':r');
ylabel ('Displacement (mm)')
xlabel ('Time (sec)')
%title ('First Floor Displacement-Time History')
hold on
plot(time,disp_1,'-')
hold on
plot(time,undisp_4,'-g')
legend ('Bare Frame','SMA Frame','Steel Frame')

figure
plot(time,unacc_3,':r');
ylabel ('Acceleration (m/s2)')
xlabel ('Time (sec)')
%title ('Third Floor Acceleration-Time History')
hold on
plot(time,acc_3,'b')
hold on
plot(time,unacc_6,'g')
legend ('Bare Frame','SMA Frame','Steel Frame')

figure
plot(time,unacc_2,':r');
ylabel ('Acceleration (m/s2)')
xlabel ('Time (sec)')
%title ('Second Floor Acceleration-Time History')
hold on
plot(time,acc_2,'b')
hold on
plot(time,unacc_5,'g')
legend ('Bare Frame','SMA Frame','Steel Frame')


figure
plot(time,unacc_1,':r');
ylabel ('Acceleration (m/s2)')
xlabel ('Time (sec)')
%title ('First Floor Acceleration-Time History')
hold on
plot(time,acc_1,'b')
hold on
plot(time,unacc_4,'g')
legend ('Bare Frame','SMA Frame','Steel Frame')


% figure
% plot(strain3p,stress3p_out);
% %title ('Third Story Wires Stress-Strain Diagram')
```

```
% xlabel('Strain (%)');
% ylabel('Stress(Mpa)');
% grid on

figure
plot(wstrain3,wstress3);
%title ('Third Story Wires Stress-Strain Diagram')
xlabel('Strain (%)');
ylabel('Stress(Mpa)');
grid on

figure
plot(wstrain2,wstress2);
%title ('Second Story Wires Stress-Strain Diagram')
xlabel('Strain (%)');
ylabel('Stress(Mpa)');
grid on

figure
plot(wstrain1,wstress1);
%title ('First Story Wires Stress-Strain Diagram')
xlabel('Strain (%)');
ylabel('Stress(Mpa)');
grid on
```

## M11

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% bridge2DOF_ILL.m
% Bridge Isolated with High Damping
%   Rubber Bearings
% Created by Osman E. Ozbulut
% April-May 2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
addpath 'F:\Earthquakes'
addpath 'F:\Earthquakes\New Folder'

%% Define System Parameters
m1 = 2385.3;
m2 = 5886;
k1 = 112700;

k2 = 47600;  %kN/m
ky = 0.19*k2;
alpha = 1 - ky/k2;

uy = 0.016;
Qy = k2*uy;
```

```
c1 = 2*0.02*(k1*m1)^0.5;
c2 = 2*0.01*(k2*(m1+m2))^0.5;

beta = 0.5;
gama = 0.5;
n = 1;
%% Define excitation function
g = 9.806;
% load la15
% Tt = t';
% F1 = acc'/980.6; % in g's
% F  = F1*9.806*0.8;

% load elcentro
% Tt = e(1,:)';
% F1 = e(2,:)/9.806;
% F = F1/max(abs(F1))*9.806*0.34;

%%% El Centro
% load elcentro
% Tt = e(1,:)';
% F = e(2,:)'/g;
% F = F/max(abs(F))*0.34*g;

% % % Northridge
% load northrdg
% Tt = n(1,:)';
% F = n(2,:)'/g;
% F = F/max(abs(F))*0.5*g;

% Bolu - EW
% load bolu
% Tt = b(:,1);
% F = b(:,2);
% F = F/max(abs(F))*0.30*g;

% % % % Chichi
% load Chichi084
% t = Chi(:,1);
% u = Chi(:,3);
% %u = Chi(:,3)/max(abs(u)*g;

% % % N Palm Spring -
% load Npalmspr
% Tt = Npalm(:,1);
% F = Npalm(:,2);
% F = F/max(abs(F))*0.40*g;

% % % Hachinhe
%load hachinhe
```

```
% t = h(1,:)';
% u = h(2,:)'/g;
% u = u/max(abs(u))*0.15*g;

% % %Kobe
% load kobe2
% Tt = kobe(1,:)';
% F = kobe(2,:)';
% F = F/max(abs(F))*0.5*g;

% Artificial
load arteq
Tt = arteq(1,:);
F = arteq(2,:)/9.806;
F = F/max(abs(F))*0.25*9.806;



figure
plot(Tt,F)
grid on
xlabel('Time')
ylabel('Acceleration (m/s2)')

%% Solve governing equation
[time,u] = ode45(@solver_bridge2DOF_ILL,[0
30],[0,0,0,0,0],[],F,Tt,m1,m2,Qy,ky,k2,k1,c1,c2,n,gama,beta,alpha);

%% Set for plots
disp1 = u(:,1)*1000;  %mm
disp2 = u(:,2)*1000;  %mm
force = u(:,5)*alpha*Qy +  ky*(u(:,2)-u(:,1));   %kN

exc = interp1(Tt,F,time);
acc1 = -1/m1*(m1*exc + (k1+ky)*u(:,1) - ky*u(:,2) + (c1+c2)*u(:,3) -
c2*u(:,4) - alpha*Qy*u(:,5)) ;
acc2 = -1/m2*(m2*exc - ky*u(:,1) + ky*u(:,2) - c2*u(:,3) + c2*u(:,4) +
alpha*Qy*u(:,5)) ;

drift = disp2 - disp1;

max_disp1 = max(abs(disp1));
max_disp2 = max(abs(disp2));
max_delta2 = max(abs(drift))
max_acc1 = max(abs(acc1));
max_acc2 = max(abs(acc2));
resultHRB = [max_disp1; max_disp2; max_delta2; max_acc1; max_acc2]

%  Override the default settings for graphs in Matlab
set(0, 'DefaultAxesFontSize', 12)
set(0, 'DefaultAxesFontWeight', 'bold')
```

```
set(0, 'DefaultAxesLineWidth', 2)
set(0, 'DefaultLineLineWidth', 2)

figure
plot(time,disp1)
grid on
xlabel('Time (t)')
ylabel('Displacement (mm)')
title ('Pier Displacement-Time History')

figure
plot(time,disp2)
grid on
xlabel('Time (t)')
ylabel('Displacement (mm)')
title ('Deck Displacement-Time History')

figure
plot(time,drift,'r')
grid on
xlabel('Time (t)')
ylabel('Displacement (mm)')
%title ('Drift Time History')
legend('HDR Bearing')

figure
plot(time,acc1)
grid on
xlabel('Time (t)')
ylabel('Acceleration (m/s2)')
title ('Pier Acceleration-Time History')

figure
plot(time,acc2)
grid on
xlabel('Time (t)')
ylabel('Acceleration (m/s2)')
title ('Deck Acceleration-Time History')

figure
plot(time,force)
grid on
xlabel('Time (t)')
ylabel('Force (kN)')

figure
plot(drift,force)
grid on
ylabel('Force (kN)')
xlabel('Displacement (mm)')
```

```
figure
plot(drift,u(:,5))
grid on
ylabel('z')
xlabel('Displacement (mm)')
```

## M12

```
% solver_bridge2DOF_ILL.m
% ODE Solver
function du =
solver_bridge2DOF_ILL(time,u,F,Tt,m1,m2,Qy,ky,k2,k1,c1,c2,n,gama,beta,a
lpha)
exc = interp1(Tt,F,time);
du = [ u(3);
       u(4);
       -1/m1*(m1*exc + (k1+ky)*u(1) - ky*u(2) + (c1+c2)*u(3) - c2*u(4)
- alpha*Qy*u(5));
       -1/m2*(m2*exc - ky*u(1) + ky*u(2) - c2*u(3) + c2*u(4) +
alpha*Qy*u(5));
       k2/Qy*((u(4)-u(3)) -beta*(u(4)-u(3))*abs(u(5))- gama*abs(u(4)-
u(3))*u(5)) ];
```

## M13

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% sma2DOFbridgescript_Final.m
% Bridge Isolated with Low Damping
%   Rubber Bearings and SMA wires
% Created by Osman E. Ozbulut
% April-May 2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all

addpath 'E:\Earthquakes'
addpath 'E:\Earthquakes\New Folder'

g = 9.806;
% %%Eathquake Data
%%%% DEFINE as in g's %%%%
%% El Centro
load elcentro
t = e(1,:)';
u = e(2,:)'/g;
u = u/max(abs(u))*0.34;

% % % Northridge
```

```
% load northrdg
% t = n(1,:)';
% u = n(2,:)'/g;
% u = (n(2,:)'/g)/max(abs(u))*0.5;

% % % Bolu - EW
% load bolu
% t = b(:,1);
% u = b(:,2);
% u = b(:,2)/max(abs(u))*0.30;

% % % Chichi
% load Chichi084
% t = Chi(:,1);
% u = Chi(:,3);
%u = Chi(:,3)/max(abs(u);

% % % N Palm Spring -
% load Npalmspr
% t = Npalm(:,1);
% u = Npalm(:,2);
% u = Npalm(:,2)/max(abs(u))*0.40;

% % % Hachinhe
%load hachinhe
% t = h(1,:)';
% u = h(2,:)'/g;
% u = (h(2,:)'/g)/max(abs(u))*0.15;

% % %Kobe
% load kobe2
% t = kobe(1,:)';
% u = kobe(2,:)'/g;
% u = (kobe(2,:)'/g)/max(abs(u))*0.2;

% Artificial
% load arteq
% t = arteq(1,:)';
% u = arteq(2,:)'/9.806;
% u = u/max(abs(u))*0.25;

plot(t,u, 'b-')
grid off
xlabel('Time (sec)')
ylabel('Acceleration (g)')
%set(gca, 'XTick', [1 2 3 4 5 6 7 8 9 10 11.2]);
title ('Earthquake Time-History')

%%

m1 = 2385.3;
```

```
m2 = 5886;

k1 = 112700 ;
kb = 7933; %kN/m

c1 = 2*0.02*(k1*m1)^0.5;
cb = 2*0.01*(kb*(m1+m2))^0.5;

% ABCD
MassVec = [m1,m2];
M = diag(MassVec);

K = [k1+kb -kb;
      -kb   kb ];

C1 = [c1+cb -cb;
       -cb  cb] ;

nDOF = 2;

%  Build state space matrices for the building structure.
A = [zeros(nDOF) eye(nDOF); -inv(M)*K -inv(M)*C1];
B = [zeros(nDOF); inv(M)];
C = [eye(2*nDOF);-inv(M)*K -inv(M)*C1];
D = [zeros(2*nDOF,nDOF); inv(M)];


% Wire Characteristics
%cos_teta = cos(pi/4);
wirearea = 1.963*10^-7 ; %m^2
coef_s = 10^2 ;
coef_sr = 10^2 ;
coef_stress = 10^2;
prestress = 0.1412 %GPa

num = 220000;
len = 4.5;
wirelength1 = len; %m
AREA = wirearea*num*1e4
ksma = 1.366*10^7*AREA*10^-4/len;

length_strain = coef_s/wirelength1 ;
length_strate = coef_sr/wirelength1 ;
force_coef = wirearea*10^9*10^-3;

% SMA Fuzzy Prediction
FISname = 'try1gauss222ss0_08coef229'
SMAfis = readfis ( [ FISname, '.fis'] );

sat_limits          = getfis(SMAfis, 'inrange');
```

```
strain_sat_min      = sat_limits(1, 1);
strain_sat_max      = sat_limits(1, 2);
strate_sat_min      = sat_limits(2, 1);
strate_sat_max      = sat_limits(2, 2);
prestress_sat_min   = sat_limits(3, 1);
prestress_sat_max   = sat_limits(3, 2);

% Simulation Options
dt_valid = 0.001;
time_start = 0;
time_end = 50;
time = [time_start:dt_valid:time_end];

tic

OPTIONS = simset('solver', 'ode5', 'FixedStep', dt_valid);
sim('sma2DOFbridge_Final', [time_start time_end], OPTIONS, []);
toc

te = toc/60
%%

% Set up for plots
disp_1 = disp1*1000;      %mm
disp_2 = disp2*1000;

undisp_1 = undisp1*1000;
undisp_2 = undisp2*1000;

max_undisp1 = max(abs(undisp_1));
max_undisp2 = max(abs(undisp_2));


max_acc1 =  max(abs(acc1));
max_acc2 =  max(abs(acc2));

max_unacc1 =  max(abs(unacc1));
max_unacc2 =  max(abs(unacc1));

max_disp1 = max(abs(disp_1));
max_disp2 = max(abs(disp_2));
max_delta2 = max(abs(delta2))
max_racc1 = max(abs(racc1));
max_racc2 = max(abs(racc2));

undelta_2 = undisp_2 - undisp_1;
max_undelta2 = max(abs(undelta_2));

resultSMA = [max_disp1; max_disp2; max_delta2; max_racc1; max_racc2]
```

```
%% Plots
%  Override the default settings for graphs in Matlab
set(0, 'DefaultAxesFontSize', 12)
set(0, 'DefaultAxesFontWeight', 'bold')
set(0, 'DefaultAxesLineWidth', 2)
set(0, 'DefaultLineLineWidth', 2)

%%%%%%%%% PLOTS FOR PAPER %%%%%%%%%%%%%%%%%%%%

figure
plot(time,undisp_1,':r');
ylabel ('Displacement (mm)')
xlabel ('Time (sec)')
title ('Pier Displacement-Time History')
hold on
plot(time,disp_1,'-')
legend ('Uncontrolled','SMA')

figure
plot(time,undisp_2,':r');
ylabel ('Displacement (mm)')
xlabel ('Time (sec)')
title ('Deck Displacement-Time History')
hold on
plot(time,disp_2,'-')
legend ('Uncontrolled','SMA')

figure
plot(time,delta2);
ylabel ('Displacement (mm)')
xlabel ('Time (sec)')
%title ('Drift Time History')
legend('SMA Rubber Bearing')
% hold on
% plot(time,undelta2,'-')


figure
plot(time,runacc1,'r');
ylabel ('Acceleration (m/s2)')
xlabel ('Time (sec)')
title ('Pier Acceleration-Time History')
hold on
plot(time,racc1,'b')
legend ('Uncontrolled','SMA')

figure
plot(time,runacc2,'r');
ylabel ('Acceleration (m/s2)')
xlabel ('Time (sec)')
title ('Deck Acceleration-Time History')
```

```
hold on
plot(time,racc2,'b')
legend ('Uncontrolled','SMA')

figure
plot(wstrain,wstress);
%title ('First Story Wires Stress-Strain Diagram')
xlabel('Strain (%)');
ylabel('Stress(Mpa)');
grid on

figure
plot(delta2,wforcetot);
%title ('First Story Wires Stress-Strain Diagram')
xlabel('Displacement');
ylabel('Force (kN)');
grid on

rub_force = kb*delta2/1000 + cb*deltavel;


figure
plot(delta2,rub_force);
%title ('First Story Wires Stress-Strain Diagram')
xlabel('Rubber Displacement (mm)');
ylabel('Force (kN)');
grid on

subplot(3,1,1)
plot(wforcetot)
subplot(3,1,2)
plot(Fext1)
subplot(3,1,3)
plot(Fext2)
```

## M14

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% nonlinear.m

% Created by Osman E. Ozbulut
% January - May 20007
% Nonlinear-Time History Analysis of a SDOF bare frame using
%      Central Average Acceleration (CAA) Method
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
tic
% Define material properties
```

```
m = 167.9;
k = 23494 ;  %kN/m
c = 2*0.05*(k*m)^0.5;

% Define yield force
FY = 753.5;
UY = FY/k;


OMN = (k/m)^0.5
OM2 = OMN*OMN;
BETA = c /(2*(k*m)^0.5)
BOM = BETA*OMN;

%%  Define Excitation Function
% load elcentro
% DT = 0.001 ;
% t = 0:DT:max(e(1,:));
% A1 = interp1(e(1,:),e(2,:),t)/9.81;
% A = A1/max(abs(A1))*0.8*9.81;

% load northrdg
% DT = 0.005 ;
% t = 0:DT:30;
% A = interp1(n(1,:),n(2,:),t)/9.81;
% A = A/max(abs(A))*0.9*9.81;

% % Bolu - EW
% load bolu
% DT = 0.005;
% t = (0:DT:max(b(:,1)));
% A = interp1(b(:,1),b(:,2),t)/9.81;
% % t = b(:,1);
% % A = b(:,2);
% A = (A/max(abs(A))*0.9*9.81);


% % % Chichi
% load Chichi084
% DT = 0.005
% t = Chi(:,1);
% A = Chi(:,3);
% A = Chi(:,3)/max(abs(A))*0.90*9.81;

% % % N Palm Spring -
% load Npalmspr
% DT = 0.005 ;
% t = 0:DT:20;
% A = interp1(Npalm(:,1),Npalm(:,2),t)/9.81;
% A = (A/max(abs(A))*0.9*9.81);
```

```
% % % Hachinhe
% load hachinhe
% DT = 0.005 ;
% t = 0:DT:max(h(1,:));
% A = interp1(h(1,:),h(2,:),t)/9.81;
% A = A/max(abs(A))*1.0*9.81;

% % %Kobe
% load kobe2
% DT = 0.005 ;
% %t = 0:DT:max(kobe(1,:));
% t = 0:DT:40;
% A = interp1(kobe(1,:),kobe(2,:),t)/9.81;
% A = A/max(abs(A))*0.9*9.81;

DT = 0.005 ;
load arteq
t = 0:DT:max(arteq(1,:));
A = interp1(arteq(1,:),arteq(2,:),t)/9.806;
A = A/max(abs(A))*0.5*9.806;

figure
plot(t,A)
grid on
xlabel('Time')
ylabel('Acceleration (m/s2)')

N = length(t);
%% Solve differential equation using CAA method

[acc,vel,disp,force] = CAA(FY,UY,N,DT,BOM,OM2,A);

%  Override the default settings for graphs in Matlab
set(0, 'DefaultAxesFontSize', 12)
set(0, 'DefaultAxesFontWeight', 'bold')
set(0, 'DefaultAxesLineWidth', 2)
set(0, 'DefaultLineLineWidth', 2)

%% Set for plots

disp_mm = disp*1000;

figure
plot(t,disp_mm)
grid on
xlabel('Time')
ylabel('Displacement (mm)')

figure
plot(t,acc)
grid on
```

```
xlabel('Time')
ylabel('Acceleration')

figure
plot(disp_mm,force)
grid on
xlabel('Displacement (mm)')
ylabel('Force (kN)')

% x1 = 1;
% x2 = 200;
% figure
% plot(disp_mm(x1:x2),force(x1:x2))
% grid on
% xlabel('Displacement (mm)')
% ylabel('Force (kN)')
% figure
% plot(t(x1:x2),A(x1:x2))

figure
plot(t,force)
grid on
xlabel('Time (sec)')
ylabel('Force (kN)')

rmsdisp = rms(disp_mm);
rmsacc = rms(acc);
max_acc = max(abs(acc));
max_disp = max(abs(disp_mm));
res_disp = disp_mm(N) ;
max_force = max(force)*2;

eva_un = [max_disp, rmsdisp, res_disp, max_acc, rmsacc, max_force]

te = toc
ts = te/60
```

## M15

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CAA.m

% Created by Osman E. Ozbulut
% January - May 20007
% Central Average Acceleration (CAA) Method  (Bare Frame)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [YS,YP,Y,FOR] = CAA(FY,UY,N,DT,BOM,OM2,A);
F1(1) = 0;
F2(1) = 0;
FOR(1) = F1;
```

```
AK = FY/UY;
AKR(1) = 1;
N1 = N-1;

Y = zeros(1,length(N1));
YP = zeros(1,length(N1));
YS = zeros(1,length(N1));
R = zeros(1,length(N1));

DT2 = DT*DT;
DTH = DT/2;
DT4 = DT2/4;

for n = 1:N1;
    DET(n) = 1 + BOM*DT + OM2*DT4*AKR(n);
    P(n) = YP(n) + DTH*YS(n);
    S(n) = DT*YP(n) + DT4*YS(n);
    %  Calculate acc, vel, and disp
    YSS(n) = -(A(n+1) + 2*BOM*P(n) + OM2*S(n)*AKR(n) +
OM2*F2(n)/AK)/DET(n);
    YP(n+1) = P(n) + DTH*YSS(n);
    Y(n+1) = Y(n) + S(n) + DT4*YSS(n);

    YS(n+1) = YSS(n);
    R(n+1) = Y(n+1);
    F1(n) = F2(n);
    [F,akt] = force(R(n),R(n+1),F1(n),F2(n),FY,UY);
    FOR(n+1) = F;
    F2(n+1) = FOR(n+1);
    AKR(n+1) = akt/AK;
end

function [F2,AKT] = force(U1,U2,F1,F2,FY,UY);
AK = FY/UY;
FYN = -FY;
AKT=AK;
DU=U2-U1;

F2=F1+AK*DU;

if F2 > FY
    F2=FY;
end

if F2 < FYN
    F2=FYN;
end

if abs(F2) == FY
    AKT=0;
end
```

```
%return
```

## M16

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% nonlinearSMA.m

% Created by Osman E. Ozbulut
% January - May 20007
% Nonlinear-Time History Analysis of a SDOF SMA braced frame using
%      Central Average Acceleration (CAA) Method
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
%%
addpath 'E:\Earthquakes'
addpath 'E:\Earthquakes\New Folder'

% Define material properties
%m = 99.65;
%m = 167.9;
m = 190;
k = 23494 ;  %kN/m
c = 2*0.05*(k*m)^0.5;

% Define yield force
FY = 753.5;
%FY = 546;
UY = FY/k;

% Define brace angle
cos_teta = cos(0.279*pi);
wirearea = 1.963*10^-7;

len = 2.20
%num = 46322*len;
num = 27510*len;

% len = 0.25;
% num = 2.31e-3/wirearea;
AREA = num*wirearea*1e4;

OMN = (k/m)^0.5;
OM2 = OMN*OMN;
BETA = c /(2*(k*m)^0.5);
BOM = BETA*OMN;

%% Define excitation
% load elcentro
```

```
% DT = 0.001 ;
% t = 0:DT:max(e(1,:));
% A1 = interp1(e(1,:),e(2,:),t)/9.81;
% A = A1/max(abs(A1))*0.8*9.81;

% load northrdg
% DT = 0.005 ;
% t = 0:DT:30;
% A = interp1(n(1,:),n(2,:),t)/9.81;
% A = A/max(abs(A))*0.55*9.81;

% % Bolu - EW
% load bolu
% DT = 0.005;
% t = (0:DT:max(b(:,1)));
% A = interp1(b(:,1),b(:,2),t)/9.81;
% t = b(1:3500,1);
% A = b(1:3500,2);
% A = (A/max(abs(A))*0.9*9.81);



% % Chichi
% load Chichi084
% DT = 0.005
% t = Chi(:,1);
% A = Chi(:,3);
% A = Chi(:,3)/max(abs(A))*0.90*9.81;

% % % N Palm Spring -
% load Npalmspr
% DT = 0.005 ;
% t = 0:DT:20;
% A = interp1(Npalm(:,1),Npalm(:,2),t)/9.81;
% A = (A/max(abs(A))*0.7*9.81);

% % % Hachinhe
% load hachinhe
% DT = 0.005 ;
% t = 0:DT:max(h(1,:));
% A = interp1(h(1,:),h(2,:),t)/9.81;
% A = A/max(abs(A))*1.0*9.81;

% %Kobe
% load kobe2
% DT = 0.005 ;
% %t = 0:DT:max(kobe(1,:));
% t = 0:DT:30;
% A = interp1(kobe(1,:),kobe(2,:),t)/9.81;
% A = A/max(abs(A))*0.9*9.81;

DT = 0.005 ;
```

```
load arteq
t = 0:DT:25;
A = interp1(arteq(1,:),arteq(2,:),t)/9.806;
A = A/max(abs(A))*0.99*9.806;

figure
plot(t,A)
grid on
xlabel('Time')
ylabel('Acceleration (m/s2)')

N = length(t);
%%  Solve differential equation using CAA method
tic
[acc,vel,disp,force,Fsma1,Fsma2,Fsma,strain1] =
CAAsma2(FY,UY,N,DT,BOM,OM2,A,t,cos_teta,len,num);
te = toc
ts = te/60
%% Set for plots

%  Override the default settings for graphs in Matlab
set(0, 'DefaultAxesFontSize', 12)
set(0, 'DefaultAxesFontWeight', 'bold')
set(0, 'DefaultAxesLineWidth', 2)
set(0, 'DefaultLineLineWidth', 2)

disp_mm = disp*1000;
rmsdisp = rms(disp_mm);
rmsacc = rms(acc);

figure
plot(t,disp_mm)
grid on
xlabel('Time')
ylabel('Displacement (mm)')
legend('SMA Frame')

figure
plot(t,acc)
grid on
xlabel('Time')
ylabel('Acceleration (m/s2)')
legend('SMA Frame')

figure
plot(disp_mm,force)
grid on
xlabel('Displacement (mm)')
ylabel('Force (kN)')
legend('SMA Frame')
```

```
figure
plot(t,force)
grid on
xlabel('Time (sec)')
ylabel('Force (kN)')

smadisp = disp_mm*cos_teta;
%strain1 = smadisp/len;

stress1 = Fsma1/(AREA*1e-4)*10^-3;

figure
plot(strain1,stress1)
grid on
xlabel('Strain (%)')
ylabel('Stress (MPa)')

figure
plot(smadisp,Fsma1)
grid on
xlabel('Displacement (mm)')
ylabel('Force (kN)')

figure
plot(disp_mm, Fsma)
grid on
xlabel('Displacement (mm)')
ylabel('Force (kN)')

max(abs(Fsma))

% figure
% plot(t,Fsma1)
% grid on
% xlabel('Time (sec)')
% ylabel('SMA Force (kN)')

% figure
% plot(t,Fsma2)
% grid on
% xlabel('Displacement (mm)')
% ylabel('Force (kN)')
%
% figure
% plot(t,Fsma)
% grid on
% xlabel('Displacement (mm)')
% ylabel('Force (kN)')


% x1 = 300;
```

```
% x2 = 1000;
% figure
% plot(strain1(x1:x2)*len*10,Fsma1(x1:x2))
% grid on
% xlabel('Displacement (mm)')
% ylabel('Force (kN)')
%
% x1 = 300;
% x2 = 400;
% figure
% plot(strain1(x1:x2),stress1(x1:x2))
% grid on
% xlabel('Strain (%)')
% ylabel('Stress (MPa)')


max_acc = max(abs(acc));
max_disp = max(abs(disp_mm));
res_disp = disp_mm(N) ;
max_force = max(abs(Fsma));

eva = [max_disp, rmsdisp, res_disp, max_acc, rmsacc, max_force]
```

## M17

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CAAsma.m

% Created by Osman E. Ozbulut
% January - May 20007
% Central Average Acceleration (CAA) Method  (SMA Frame)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [YSS,YP,Y,FOR] = CAAsma(FY,UY,N,DT,BOM,OM2,A);
F1(1) = 0;
F2(1) = 0;
FOR(1) = F1;

AK = FY/UY;
AKR(1) = 1;
N1 = N-1;
cos_teta = cos(pi/4);

Y = zeros(1,length(N1));
YP = zeros(1,length(N1));
YS = zeros(1,length(N1));
R = zeros(1,length(N1));
Fsma1 = zeros(1,length(N1));
Fsma2 = zeros(1,length(N1));
Fsma = zeros(1,length(N1));
```

```
DT2 = DT*DT;
DTH = DT/2;
DT4 = DT2/4;

for n = 1:N1;
    DET(n) = 1 + BOM*DT + OM2*DT4*AKR(n);
    P(n) = YP(n) + DTH*YS(n);
    S(n) = DT*YP(n) + DT4*YS(n);
    %  Calculate acc, vel, and disp
    YSS(n) = -(A(n+1) + 2*BOM*P(n) + OM2*S(n)*AKR(n) +
OM2/AK*(F2(n)+Fsma(n)))/DET(n);
    YP(n+1) = P(n) + DTH*YSS(n);
    Y(n+1) = Y(n) + S(n) + DT4*YSS(n);
    % Compute Fs
    YS(n+1) = YSS(n);
    R(n+1) = Y(n+1);
    F1(n) = F2(n);
    [F,akt] = force(R(n),R(n+1),F1(n),F2(n),FY,UY);
    FOR(n+1) = F;
    F2(n+1) = FOR(n+1);
    AKR(n+1) = akt/AK;
    % Compute SMA forces for next step
    Fsma1(n+1) = SMAforce1(R(n+1),YP(n+1),cos_teta)
    Fsma2(n+1) = SMAforce2(R(n+1),YP(n+1),cos_teta)
    Fsma(n+1) = (Fsma1(n+1) - Fsma2(n+1))*cos_teta
end

%%%%%%%%%%%%%%%% Fs %%%%%%%%%%%%%%%%

function [F2,AKT] = force(U1,U2,F1,F2,FY,UY);
AK = FY/UY;
FYN = -FY;
AKT=AK;
DU=U2-U1;

F2=F1+AK*DU;

if F2 > FY
    F2=FY;
end

if F2 < FYN
    F2=FYN;
end

if abs(F2) == FY
    AKT=0;
end
%return

%%%%%%%%%%%%% Fsma1 %%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function Fsma1 = SMAforce1(U2,V2,cos_teta)
% Load 3D matrix
load SMA
% Define strain, strain rate and prestress
length = 0.5;
area = 1.963*10^-7;
strain = U2*cos_teta / length;
strate = V2*cos_teta / length;
prestress = 120;

% Get FIS input range
FISname = 'try1gauss222coef226ss0_13e200';
SMAfis = readfis ( [ FISname, '.fis'] );
sat_limits          = getfis(SMAfis, 'inrange');
strain_min = sat_limits(1, 1);
strain_max = sat_limits(1, 2);
strate_min = sat_limits(2, 1);
strate_max = sat_limits(2, 2);
prestress_min = sat_limits(3, 1);
prestress_max = sat_limits(3, 2);

% Define intervals
dim = 10;
inc1 = (strain_max - strain_min)/dim;
inc2 = (strate_max - strate_min)/dim;
inc3 = (prestress_max - prestress_min)/dim;

% Check max/min strain
if strain > strain_max
    strain = strain_max
end

if strain < strain_min
    strain = strain_min
end

% Check max/min strate
if strate > strate_max
    strate = strate_max
end

if strate < strate_min
    strate = strate_min
end

for x=1:dim
    if (strain >= (strain_min + inc1*(x-1)) & strain <= (strain_min +
inc1*x))
        for y=1:dim
```

```
            if (strate >= (strate_min + inc2*(y-1)) & strate <=
(strate_min + inc2*y))
                for z=1:dim
                   if(prestress >= (prestress_min + inc3*(z-1)) &
prestress <= (prestress_min + inc3*z))
                         stress1 = SMA(x,y,z)
                   end
                end
            end
        end
    end
end

Fsma1 = stress1*area;

%%%%%%%%%%%%%%%%%%% Fsma2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Fsma2 = SMAforce2(U2,V2,cos_teta)
% Load 3D matrix
load SMA
% Define strain, strain rate and prestress
length = 0.5;
area = 1.963*10^-7;
strain = -U2*cos_teta / length
strate = -V2*cos_teta / length;
prestress = 120;

% Get FIS input range
FISname = 'try1gauss222coef226ss0_13e200';
SMAfis = readfis ( [ FISname, '.fis'] );
sat_limits          = getfis(SMAfis, 'inrange');
strain_min = sat_limits(1, 1);
strain_max = sat_limits(1, 2);
strate_min = sat_limits(2, 1);
strate_max = sat_limits(2, 2);
prestress_min = sat_limits(3, 1);
prestress_max = sat_limits(3, 2);

% Define intervals
dim = 10;
inc1 = (strain_max - strain_min)/dim;
inc2 = (strate_max - strate_min)/dim;
inc3 = (prestress_max - prestress_min)/dim;

% Check max/min strain
if strain > strain_max
    strain = strain_max
end

if strain < strain_min
    strain = strain_min
```

```
end

% Check max/min strate
if strate > strate_max
    strate = strate_max
end

if strate < strate_min
    strate = strate_min
end

for x=1:dim
    if (strain >= (strain_min + inc1*(x-1)) & strain <= (strain_min +
inc1*x))
        for y=1:dim
             if (strate >= (strate_min + inc2*(y-1)) & strate <=
(strate_min + inc2*y))
                 for z=1:dim
                    if(prestress >= (prestress_min + inc3*(z-1)) &
prestress <= (prestress_min + inc3*z))
                        stress2 = SMA(x,y,z)
                    end
                end
            end
        end
    end
end

Fsma2 = stress2*area;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## M18

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% nonlinearSteel2B.m

% Created by Osman E. Ozbulut
% January - May 20007
% Nonlinear-Time History Analysis of a SDOF steel braced frame
%      (no buckling in compression) using
%      Central Average Acceleration (CAA) Method
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all

addpath 'E:\Earthquakes'
addpath 'E:\Earthquakes\New Folder'
```

```
tic
% Define material properties
%m = 99.65;
%m = 167.9;
m = 190;
K = 23494 ;  %kN/m
c = 2*0.05*(K*m)^0.5;
KB = 124209 ;
%KB = 178657;

% Define yield force
FY = 753.5;
%FY = 546;
UY = FY/K;
%FYB = 693.5;
FYB = 997.5;
UYB = FYB/KB;

cos_teta = cos(0.279*pi);

OMN = (K/m)^0.5;
OM2 = OMN*OMN;
BETA = c /(2*(K*m)^0.5);
BOM = BETA*OMN;
OMNB = (KB/m)^0.5;
OMB2 = OMNB*OMNB;

%% Define excitation

% load elcentro
% DT = 0.005 ;
% t = 0:DT:max(e(1,:));
% A1 = interp1(e(1,:),e(2,:),t)/9.81;
% A = A1/max(abs(A1))*0.9*9.81;

%%% Northridge -26 mm
% load northrdg
% DT = 0.005 ;
% t = 0:DT:30;
% A = interp1(n(1,:),n(2,:),t)/9.81;
% A = A/max(abs(A))*0.50*9.81;

% % Bolu - EW
% load bolu
% DT = 0.005;
% % t = (0:DT:max(b(:,1)));
% % A = interp1(b(:,1),b(:,2),t)/9.81;
% t = b(1:3500,1);
% A = b(1:3500,2);
% A = (A/max(abs(A))*0.9*9.81);
```

```
% % Chichi
% load Chichi084
% DT = 0.005
% t = Chi(:,1);
% A = Chi(:,3);
% A = Chi(:,3)/max(abs(A))*0.90*9.81;

% % % N Palm Spring -
% load Npalmspr
% DT = 0.005 ;
% t = 0:DT:20;
% A = interp1(Npalm(:,1),Npalm(:,2),t)/9.81;
% A = (A/max(abs(A))*0.7*9.81);

% % % Hachinhe
% load hachinhe
% DT = 0.005 ;
% t = 0:DT:max(h(1,:));
% A = interp1(h(1,:),h(2,:),t)/9.81;
% A = A/max(abs(A))*0.9*9.81;

%Kobe
% load kobe2
% DT = 0.005 ;
% t = 0:DT:30;
% A = interp1(kobe(1,:),kobe(2,:),t)/9.81;
% A = A/max(abs(A))*0.9*9.81;

% Artificial
DT = 0.005 ;
load arteq
t = 0:DT:25;
A = interp1(arteq(1,:),arteq(2,:),t)/9.806;
A = A/max(abs(A))*0.99*9.806;

figure
plot(t,A)
grid on
xlabel('Time')
ylabel('Acceleration (m/s2)')

%% Solve differential equation using CAA method

N = length(t);

[acc,vel,disp,force,Fr2,Fl2,Fbrace,disp1,disp2] =
CAAsteel2B_C(FY,UY,FYB,UYB,N,DT,BOM,OM2,OMB2,A,cos_teta);

te = toc
ts = te/60
```

```
%% Plots

%  Override the default settings for graphs in Matlab
set(0, 'DefaultAxesFontSize', 12)
set(0, 'DefaultAxesFontWeight', 'bold')
set(0, 'DefaultAxesLineWidth', 2)
set(0, 'DefaultLineLineWidth', 2)

disp_mm = disp*1000;

figure
plot(t,disp_mm,'r')
grid on
xlabel('Time')
ylabel('Displacement (mm)')
legend('Steel Frame')

figure
plot(t,acc,'r')
grid on
xlabel('Time')
ylabel('Acceleration (m/s2)')
legend('Steel Frame')

figure
plot(disp_mm,force)
grid on
xlabel('Displacement (mm)')
ylabel('Force (kN)')

figure
plot(t,force)
grid on
hold on
plot(t,Fbrace,'y')
xlabel('Time (sec)')
ylabel('Force (kN)')

figure
plot(disp_mm,Fbrace*cos_teta,'r')
grid on
xlabel('Displacement (mm)')
ylabel('Force (kN)')

figure
plot(t,Fr2)
grid on
xlabel('Time (sec)')
ylabel('Force (kN)')

figure
```

```
plot(disp2*1000,Fl2)
grid on
xlabel('Displacement (mm)')
ylabel('Force (kN)')

% Compute response quantities
rmsdisp = rms(disp_mm);
rmsacc = rms(acc);
max_acc = max(abs(acc));
max_disp = max(abs(disp_mm));
res_disp = disp_mm(N) ;
max_force = max(Fbrace*cos_teta);

eva_s = [max_disp, rmsdisp, res_disp, max_acc, rmsacc, max_force]
```

## M19

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CAAsteel2B.m

% Created by Osman E. Ozbulut
% January - May 20007
% Central Average Acceleration (CAA) Method  (Steel Frame)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [YS,YP,Y,FOR,Fr2,Fl2,Fbrace,disp1,disp2] =
CAAsteel2B(FY,UY,FYB,UYB,N,DT,BOM,OM2,OMB2,A,cos_teta);
F1(1) = 0;
F2(1) = 0;
FOR(1) = F1;

AK = FY/UY;
AKB = FYB/UYB;
AKR(1) = 1;
AKBR(1) = 1;
N1 = N-1;

Y = zeros(1,length(N1));
YP = zeros(1,length(N1));
YS = zeros(1,length(N1));
R = zeros(1,length(N1));
Fr1 = zeros(1,length(N1));
Fr2 = zeros(1,length(N1));
Fl1 = zeros(1,length(N1));
Fl2 = zeros(1,length(N1));
Fbrace = zeros(1,length(N1));
disp1 = zeros(1,length(N1));
disp2 = zeros(1,length(N1));

DT2 = DT*DT;
DTH = DT/2;
DT4 = DT2/4;
```

```
for n = 1:N1;
    DET(n) = 1 + BOM*DT + OM2*DT4*AKR(n)+ OMB2*DT4*AKBR(n);
    P(n) = YP(n) + DTH*YS(n);
    S(n) = DT*YP(n) + DT4*YS(n);
    %  Calculate acc, vel, and disp
    YSS(n) = -(A(n+1) + 2*BOM*P(n) + OM2*S(n)*AKR(n) + OM2/AK*F2(n)+
OMB2*S(n)*AKBR(n) + OMB2/AKB*Fbrace(n))/DET(n);
    YP(n+1) = P(n) + DTH*YSS(n);
    Y(n+1) = Y(n) + S(n) + DT4*YSS(n);
    % Compute Fs
    YS(n+1) = YSS(n);
    R(n+1) = Y(n+1);
    F1(n) = F2(n);
    Fr1(n) = Fr2(n);
    Fl1(n) = Fl2(n);
    [F,akt] = force(R(n),R(n+1),F1(n),F2(n),FY,UY);
    FOR(n+1) = F;
    F2(n+1) = FOR(n+1);
    AKR(n+1) = akt/AK;
    % Compute brace forces for next step
    [Fr2(n+1),delta1(n),akbt1] =
Fbrace1(R(n),R(n+1),cos_teta,Fr1(n),Fr2(n),FYB,UYB);
    [Fl2(n+1),delta2(n),akbt2] =
Fbrace2(R(n),R(n+1),cos_teta,Fl1(n),Fl2(n),FYB,UYB);
    disp1(n+1) = disp1(n) + delta1(n);
    disp2(n+1) = disp2(n) + delta2(n);
    Fbrace(n+1) = (Fr2(n+1)-Fl2(n+1))*cos_teta;
    AKBR(n+1) = akbt1/AKB;
end

%%%%%%%%%%%%%%%% Fs %%%%%%%%%%%%%%%%

function [F2,AKT] = force(U1,U2,F1,F2,FY,UY);
AK = FY/UY;
FYN = -FY;
AKT=AK;
DU=U2-U1;

F2=F1+AK*DU;

if F2 > FY
    F2=FY;
end

if F2 < FYN
    F2=FYN;
end

if abs(F2) == FY
    AKT=0;
```

```
end
%return

%%%%%%%%%%%%% Fbrace1 %%%%%%%%%%%%%%%%%%%%%%%%%%

function [Fr2,DU,AKBT1] = Fbrace1(U1,U2,cos_teta,Fr1,Fr2,FYB,UYB)
AKB = FYB/UYB;
FYBN = -FYB;
AKBT1 = AKB;
DU = (U2-U1)*cos_teta;
Fr2 = Fr1+AKB*DU ;

if Fr2 > FYB
    Fr2=FYB;
end

if Fr2 < FYBN
    Fr2=FYBN;
end

if abs(Fr2) == FYB
    AKBT1=0;
end
%AKBT1

%%%%%%%%%%%%%%%%%%% Fbrace2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Fl2,DU,AKBT2] = Fbrace2(U1,U2,cos_teta,Fl1,Fl2,FYB,UYB)
AKB = FYB/UYB;
FYBN = -FYB;
AKBT2 = AKB;
DU = -(U2-U1)*cos_teta;
Fl2 = Fl1+AKB*DU ;

if Fl2 > FYB
    Fl2=FYB;
end

if Fl2 < FYBN
    Fl2=FYBN;
end

if abs(Fl2) == FYB
    AKBT2=0
end
%AKBT2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

**M19**

```
% runopt_gauss.m

% Created by : Hyun-Su Kim
%              09-06-2004
% Modified by: PR    18 December 2004
% Modified by: PR    11 Feb 2005
% Modified b : PR    31 May 2005
% Modified b : PR    4  June 2005
% Modified by: DS    8  August 2005
%              DS   18  May 2006
%              OEO      September 2006
glbvar;

% Define the degrees of freedom of the structure
nDOF = 3 ;

% Initialize the number of the iteration to zero.
nIter = 0;

%
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% Define the earthquake name and time history (see Simulink)
% The goal is to get the earthquake into a file named eq.mat.
% Here we used lab data collected at NCREE benchmark tests from January
% 2006
% load ELC100_UNC
% eq = [data(1,:);data(2,:)];
% eq(2,:) = eq(2,:)*9.81;
% % Extract the time vector from the input file.
% t = eq(1,:);
% save eq eq

% % % El Centro
% load elcentro
% t = e(1,:);
% u = e(2,:)/9.806;
% %u = 0.43065*e(2,:); % 0.15g
% u = u/max(abs(u))*0.15*9.806;
% eq = [t;u];

load arteq
t = arteq(1,:);
u = arteq(2,:)/9.806;
u = u/max(abs(u))*0.15*9.806;
eq = [t;u];
save eq eq
% Prepare time increment and total simulation time for Simulink
dt = t(2) - t(1) ;
dt_valid = 0.001;
Duration = max(t(:)) ;
```

```
% ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% Enter the number of the run from the keyboard.  There are only two
% possibilities:  1 = first time run, 2 = any restart run.
numRun = input('Input number of run [Default = 1]: ');
if isempty(numRun)
  numRun = 1;
end
if numRun ~= 1
    disp('If you loaded previous results, strike any key to continue.
Otherwise, press [Ctrl+C], load previous results and try again!');
  pause
  initPop = endPop;
  bResume = 1;
else
  initPop = [];
  bResume = 0;
end

fitnessFunc = 'fitfunc1_gauss';
% Declare the number of variables which are cross sectional area and
% lengthof SMA wire
nVariableNum = 6 ;

% Declare the number of input variables: 'x' rules times variables
numVar = nVariableNum ;

% Set the total population size.  This is somewhat arbitrary.
popSize = 50 ;

%Number of generations for GA optimization
genNum = 200; % for one running

%Number of muation generations for GA optimization
mutGenNum = 50; %finalGenNum-genNum*(numRun-1);

% Set the range for input of the accelerometers instruments themselves.
% (Note:  this is not the same as the maximum/minimum acceleration that
% is used for training purposes).  The instruments are assumed to have
a
% greater range than the training range using for the controller.
% These values are used in the Simulink saturation blocks.
% Also see glbvar.m for transferring these values to other routines.
% Units here are m/s^2.
sensor_min_acc = -20 ;
sensor_max_acc =  20 ;

% Load Fuzzy model of SMA
SMAfis = readfis('try1gauss222ss0_08coef229.fis');
sat_limits          = getfis(SMAfis, 'inrange');
strain_sat_min      = sat_limits(1, 1);
```

```
strain_sat_max      = sat_limits(1, 2);
strate_sat_min      = sat_limits(2, 1);
strate_sat_max      = sat_limits(2, 2);
prestress_sat_min   = sat_limits(3, 1);
prestress_sat_max   = sat_limits(3, 2);

% Here define search space for GA
% Here define max ans min possible area and length
max_num1 = 500;
min_num1 = 100;
max_num2 = 500;
min_num2 = 100;
max_num3 = 500;
min_num3 = 100;
max_length1 = 1.3;
min_length1 = 0.8;
max_length2 = 1.3;
min_length2 = 0.8;
max_length3 = 1.3;
min_length3 = 0.8;
num_range1   = [min_num1 max_num1] ;
num_range2   = [min_num2 max_num2] ;
num_range3   = [min_num3 max_num3] ;
length_range1  = [min_length1 max_length1] ;
length_range2  = [min_length2 max_length2] ;
length_range3  = [min_length3 max_length3] ;

% Define a matrix of the slope and center point of each membership
function
% for the input and output variables.
initRange = [ ...
   num_range1; num_range2; num_range3;
length_range1;length_range2;length_range3;...
   ] ;

% The range of each input variable is given below.  The last two rows
give
% the range of the possible floor locations for the MR dampers.
% Inputs are roof acceleration, roof velocity and outputs are command
% voltages to each MR damper.
% They are arranged in sets of eight.  The order is as follows:
% 1. Distribution of the first input variable (acceleration; m/s^2)
% 2. Center of the first input variable (acceleration; m/s^2)
% 3. Distribution of the second input variable (velocity; m/s)
% 4. Center of the second input variable (velocity; m/s)
% 5. Distribution of the first output variable (voltage; V)
% 6. Center of the first output variable (voltage; V)
% 7. Distribution of the second output variable (voltage; V)
% 8. Center of the second output variable (voltage; V)

% Check to see that the number of rules times the number of variables
per
```

```
% rule agrees with the dimensions of initRange
if numVar ~= size(initRange,1)
    disp ('Check the size of the initRange and the number of rules,
etc.');
    stop
end

% Set the mutation rate for child individuals that are created by the
mutation operator.
% Some guidance was obtained from the Matlab GA user manual.
mutRate = 0.2;
% Compute the number of population members in the next generation that
are obtained from mutation.
mutNum =  mutRate*popSize;
% Set the rate of child individuals that are created by the crossover
% operator.  Again, some guidance was obtained from the Matlab GA user
% manual.
xoverRate = 1;
% Calculate the number of population members for the next generation
that are obtained from
% rounding down the crossover rate times the total population and
dividing
% by 2.  Division by 2 is done because the population is not large
enough
% to handle a large crossover rate.
xoverNum =  floor(xoverRate*popSize/2.);
% Set a tolerance rate.
tolerance = 1e-6;
% Set a switch for displaying values to the screen during calculations.
dispOnOff = 1;
% The objectives to be minimized are:
%    1. Maximum peak acceleration for all floors (m/s^2) normalized by
%    uncontrolled maximum peak acceleration
%    2. Maximum peak displacement for all floors (m) normalized by
%    uncontrolled maximum peak displacement
%    3. Maximum RMS acceleration for all floors (m/s^2) normalized by
%    uncontrolled maximum RMS acceleration
%    4. Maximum RMS displacement for all floors (m) normalized by
%    uncontrolled maximum RMS displacement
% Define the number of multiobjective functions
%   Peak Accel; Peak Displacement; RMS Accel; RMS Displacement
numObj = 4;

%=== Start GA function
=====================================================
[x,endPop,bPop,traceInfo] = NSGA2(initRange, fitnessFunc,[],initPop,...
    [popSize tolerance dispOnOff numObj
bResume],'maxGenTerm',genNum,...
  'tournSelectNSGA2',[2],['simpleXoverNSGA2'],[xoverNum
0],'nonUnifMutateNSGA2',[mutNum mutGenNum 3]);
```

# VITA

Osman Eser Ozbulut received his Bachelor of Science degree in civil engineering from Technical University of Istanbul in 2005. He enrolled in the structural engineering program at the Zachry Department of Civil Engineering, Texas A&M University, in August 2005, and received his Masters of Science degree in 2007. His research interests include structural vibration control techniques, smart materials and structures, and earthquake engineering.

Osman Ozbulut may be reached at 1000 Val Verde Drive, College Station, TX 77845. His e-mail address is ozbulute@yahoo.com.