NTNU – Trondheim
Norwegian University of
Science and Technology

Behavior of Aluminum at Elevated Strain Rates and Temperatures

Eivind Semb

Civil and Environmental Engineering
Submission date: June 2013
Supervisor: Arild Holm Clausen, KT

Norwegian University of Science and Technology
Department of Structural Engineering

Department of Structural Engineering
Faculty of Engineering Science and Technology
NTNU- Norwegian University of Science and Technology

ACCESSIBILITY
OPEN

MASTER THESIS 2013

SUBJECT AREA:	DATE:	NO. OF PAGES:
Computational Mechanics	14 June 2013	11 + 117 + 108

TITLE:

Behavior of Aluminum at Elevated Strain Rates and Temperatures

Oppførsel av aluminium ved høye tøyningshastigheter og temperaturer

BY:

Eivind Semb

SUMMARY:

This thesis explores the thermomechanical behavior of aluminum. Experimental work has been conducted for a wide range of temperatures and strain rates for three AA6060 alloys in both quasi-static and split-Hopkinson tension bar test rigs. An induction heater system, pyrometer and high-speed camera was used to obtain elevated temperatures and information about the geometry in the necked section of the specimen. Some tests show slightly different material behavior between the alloys studied with respect to yield stress and strain hardening. However, no coherent difference can be established as the deviations are not seen from all tests, and are probably not significant. Three material models have been fitted with an available database containing material data for a similar alloy for a wide range of strain rates and temperatures. No adequate fit is obtained for the investigated models using the procedure described, but some significant differences between the models are seen. Numerical simulations of the split-Hopkinson tension bar experiments have been performed, but no good prediction for the material behavior until fracture was found. The reason for this is believed to be the material model parameters implemented. Numerical simulations with damage coupling have also been performed and show that fracture is predicted earlier.

RESPONSIBLE TEACHER: Professor Arild Holm Clausen

SUPERVISOR(S): PhD Candidate Vincent Vilamosa and Professor Arild Holm Clausen

CARRIED OUT AT: SIMLab, The Department of Structural Engineering, NTNU

NORGES TEKNISK-NATURVITENSKAPELIGE UNIVERSITET
INSTITUTT FOR KONSTRUKSJONSTEKNIKK

MASTER THESIS 2013

Eivind Semb

Behavior of Aluminum at Elevated Strain Rates and Temperatures

(Oppførsel av aluminium ved høye tøyningshastigheter og temperaturer)

Like most other materials, the strength of aluminum increases with increasing strain rate, while increasing temperature has the opposite effect. The ductility is also influenced by rate and temperature. It turns out, however, that the strain rate sensitivity increases with increasing temperature. This interaction is relevant in several applications, e.g. forming operations.

A procedure for tension tests on aluminum samples at a spectre of temperatures and strain rates has recently been established at SIMLab. A split-Hopkinson tension bar (SHTB) can apply strain rates between 200 and 1000 s^{-1} to the sample, while a conventional material test machine is employed for quasi-static tests with strain rates up to approx. 1 s^{-1}. An induction-based apparatus is used to heat the sample to temperatures up to approx. 500°C. The test rig is instrumented with a pyrometer and a high-speed camera. An important part of this thesis is to generate an experimental data base on three aluminium alloys AA6060 with slightly different chemical compositions. Another part of the thesis is concerned with numerical modeling. The candidate shall explore how existing models in LS-DYNA (or Abaqus) handle the coupling between temperature and strain rate. The coefficients involved in the models have to be determined with data from the material tests. The numerical model should be validated for instance by doing simulations of a SHTB test.

Some keywords for activities related to this master thesis project may include:

- Litterature: Behaviour of aluminium. Experimental techniques. Material models.
- Experimental tests: Tension tests at different temperatures and rates. Presentation of results.
- Calibration: Treatment of experimental data. Identify coefficients of the material models.
- Numerical modelling: Simulation of tests.

The candidate may agree with the supervisors to pay particular attention to specific parts of the investigation, or include other aspects than those already mentioned.

The thesis is to be organized as a research report, recognising the guidelines provided by Department of Structural Engineering.

Supervisors: Vincent Vilamosa and Arild Holm Clausen

The report is due at 14 June 2013.

NTNU, 18 January 2013

Arild Holm Clausen

Preface

This thesis was written during the spring of 2013 and submitted as a partial requirement for the degree of Masters of Science in Civil and Environmental Engineering with specialization in Computational Mechanics. The experimental work was funded and the problem statement was formulated by the Structural Impact Laboratory (SIMLab) at the Department of Structural Engineering at the Norwegian University of Science and Technology (NTNU).

Trondheim, June 14, 2013

Eivind Semb

Acknowledgements

Dr. Torodd Berstad provided the computer resources necessary and helped out with the installation of LS-DYNA. Mr. Trond Auestad has been irreplaceable for carrying out the experimental work for this thesis. His knowledge and experience regarding both split-Hopkinson bar and quasi-static experiments have been absolutely essential for conducting the experiments.

My colleagues that I have shared office with for the last year; Christoffer Rognseth, Johannes Aalberg, Paul Kopperud, Magnus Rogne Myklebost and Andreas Riseng, have contributed to create a social and stimulating work environment. Their personal characteristics and knowledge have been both motivating and helping me with relevant and irrelevant issues.

The support and help from my supervisor Professor Arild Holm Clausen is very much appreciated. Meetings in the beginning made sure I got the necessary follow-up to get started, while answering my emails day (and night!) has been a very good support.

Lastly, a special thanks go out to PhD candidate Vincent Vilamosa that have guided me through the work for this thesis. He has, together with Trond Auestad, been carrying out the experimental work and challenged me to gain as much insight into the topic of high strain rate experiments and modeling as possible. Even when having a tight schedule himself, in addition to becoming a father during the last month of this work, he was always available to answer my questions and help out with issues that I encountered.

Abstract

This thesis explores the thermomechanical behavior of aluminum. Experimental work has been conducted for a wide range of temperatures and strain rates for three AA6060 alloys in both quasi-static and split-Hopkinson tension bar test rigs. An induction heater system, pyrometer and high-speed camera was used to obtain elevated temperatures and information about the geometry in the necked section of the specimen. Some tests show slightly different material behavior between the alloys studied with respect to yield stress and strain hardening. However, no coherent difference can be established as the deviations are not seen from all tests, and are probably not significant. Three material models have been fitted with an available database containing material data for a similar alloy for a wide range of strain rates and temperatures. No adequate fit is obtained for the investigated models using the procedure described, but some significant differences between the models are seen. Numerical simulations of the split-Hopkinson tension bar experiments have been performed, but no good prediction for the material behavior until fracture was found. The reason for this is believed to be the material model parameters implemented. Numerical simulations with damage coupling have also been performed and show that fracture is predicted earlier.

Sammendrag

Denne oppgaven utforsker den termomekaniske oppførselen til aluminium. Eksperimentelle tester har blitt utført for et bredt spekter av temperaturer og tøyningshastigheter for tre AA6060-legeringer i både kvasi-statisk og split-Hopkinson tension bar testrigger. En induksjonsoppvarmer, pyrometer og høyhastighetskamera har blitt brukt for å oppnå forhøyede temperaturer og informasjon om geometrien i den innsnevrede delen av prøvestykket. Noen forsøk viser noe forskjellig materialoppførsel for de studerte legeringene med hensyn på flytespenning og fastning. Likevel kan ingen betydelig forskjell bli etablert ettersom den avvikende oppførselen ikke er sett for alle forsøk, og er antageligvis heller ikke signifikant. Tre materialmodeller har blitt tilpasset for en tilgjengelig database som inneholder materialdata for en liknende legering for et bredt spekter av tøyningshastigheter og temperaturer. Ingen tilfredsstillende tilpasning er funnet for de studerte modellene ved å bruke metoden som er beskrevet, men signifikante forskjeller mellom modellene kan sees. Numeriske simuleringer er blitt utført for split-Hopkinson tension bar forsøkene, men ingen god prediksjon av materialoppførselen fram til brudd er funnet. Grunnen til dette er antatt å være koblet til parametrene for materialmodellen som er implementert. Numeriske simuleringer koblet med skadeutvikling er og blitt utført og resulterer i at brudd blir predikert tidligere.

Contents

Preface i

Acknowledgements iii

Abstract v

Contents ix

1 Introduction 1

2 Theoretical Background 5
- 2.1 Mechanical Behavior of Metals . . . 5
  - 2.1.1 Elasticity and Plasticity Theory . . . 5
  - 2.1.2 Strain Measures . . . 5
  - 2.1.3 Necking and the Effect of Non-Uniformities of Stress at Neck 7
  - 2.1.4 Rheological Model . . . 9
- 2.2 Internal Structure of Metals . . . 11
  - 2.2.1 Bonding Between Atoms . . . 11
  - 2.2.2 Crystal Unit Cell Structures . . . 12
  - 2.2.3 Stacking Sequences . . . 14
  - 2.2.4 Dislocation Mechanisms . . . 15
  - 2.2.5 Ductile Fracture and Nucleation and Growth of Voids . . . 16
- 2.3 Aluminum . . . 17
  - 2.3.1 Alloy Designation . . . 17
  - 2.3.2 Temper Designation . . . 18
- 2.4 Constitutive Relations and Fracture Criteria . . . 19
  - 2.4.1 Johnson-Cook Constitutive Relation . . . 20
  - 2.4.2 Introduction to Microstructural Based (Semi-)Physical Models 21
  - 2.4.3 Microstructural Based Models for BCC and FCC Metals . . 23
  - 2.4.4 Zerilli-Armstrong Constitutive Relation . . . 26
  - 2.4.5 A Modified Zerilli-Armstrong Constitutive Relation . . . 26
  - 2.4.6 A Combined Constitutive Relation for both BCC and FCC Metals . . . 27
  - 2.4.7 Cockroft-Latham Fracture Criterion . . . 29

3 Experimental Work 31
- 3.1 The AA6060 Alloy . . . 31
- 3.2 Calculation of Response in Test Specimen from SHTB Tests . . . 32
- 3.3 Experimental Tests (SHTB) . . . 33
- 3.4 Experimental Tests (Quasi-Static) . . . 39

4 Experimental Results 41
- 4.1 Correction of Young's Modulus and Strains . . . 41
- 4.2 Post-Processing of Data from Quasi-Static Experiments . . . 42
- 4.3 Results from Quasi-Static Experiments . . . 46

4.3.1 Experimental Program 46
4.3.2 Strain Hardening 48
4.3.3 Yield Stress 55
4.3.4 Fracture Strain 59
4.4 Post-Processing of Data from SHTB Experiments 63
4.5 Results from SHTB Experiments 71
4.5.1 Experimental Program 71
4.5.2 Experimental Results 72

5 Calibration of Material Models **75**
5.1 Modified Johnson-Cook Model 75
5.2 Modified Zerilli-Armstrong Model 80
5.3 Combined BCC and FCC Model 85
5.4 Comparison of Material Models and Discussion 90

6 Numerical Analysis **93**
6.1 Introduction 93
6.2 Finite Element Model of SHTB Setup 93
6.3 Material Model 97
6.4 Results From Simulations 100

7 Concluding Remarks **109**

8 Further Work **113**

References **115**

A Historical Overview of SHTB Test Setups **A1**

B One-Dimensional Elastic Wave Theory **B3**

C Calculation of Response in Test Specimen from SHTB Tests **C5**

D Matlab Scripts **D9**
D.1 Post-Processing of Data from Quasi-Static Tests D9
D.2 Post-Processing of Data from SHTB Tests D26
D.3 Post-Processing of Data from Simulations in LS-DYNA D42
D.4 Calibration of Material Models D44

E Experimental Results from Quasi-Static Tests **E57**
E.1 AA6060-OLD E57
E.2 AA6060-L E62
E.3 AA6060-H E73

F Experimental Results from SHTB Tests **F85**
F.1 AA6060-L F85
F.2 AA6060-H F90

G Pictures of Specimens Post-Fracture from Quasi-Static Tests **G95**

G.1 AA6060-OLD G96
G.2 AA6060-L G97
G.3 AA6060-H G100

H Pictures of Specimens Post-Fracture from SHTB Tests H103
H.1 AA6060-L H103
H.2 AA6060-H H104

I LS-DYNA Keyword File I105

1 Introduction

Aluminum alloys are attractive for use in different applications owing to its low weight, high strength-to-weight ratio and good resistance to corrosion to name a few. Aluminum has very much substituted other established metals such as copper, steel and iron in a lot of fields. There has recently been a change in the automotive industry where aluminum is now substituting steel in various components due to the industry's everlasting endeavor for lighter components. In crash situations, automotive parts will be subjected to load cases where both high strain rates and increasing temperatures due to adiabatic heating may be present. Explosions are another example where metals may be subjected to such load cases. Moreover, materials are subjected to high temperatures and deformation rates in forming operations. Material properties obtained under quasi-static loading conditions cannot be directly applied to describe the material behavior during high rate loading conditions. When designing aluminum components, e.g. for the automotive industry, material properties obtained at the same loading rates that occur during crashes should be taken into account. In such situations, local strain rates can be of order 10^2 to 10^3 s^{-1}.

The split-Hopkinson tension bar is a widely used and recognized test procedure for conducting high strain rate experiments. Several methods for heating the test specimen to elevated temperatures exists, and for this thesis, an induction heater system is used for both quasi-static tests and split-Hopkinson tension bar tests to conduct experiments for a wide range of temperatures. Coupled with a pyrometer and a high-speed camera, it is possible to conduct experiments at elevated temperatures and under controlled conditions, and local measurement of the stress and strain state of the specimen can be obtained from the camera recordings.

The main scope for this thesis is to conduct experiments and study the behavior for two aluminum alloys. The aluminum alloys studied are referred to as AA6060-L and AA6060-H, where L and H denotes respectively "low" and "high" due to the content of alloying elements. Fig. 1.1 illustrates the magnesium (Mg) and silicon (Si) content of the two alloys. Another AA6060 alloy, denoted AA6060-OLD, has also similar content of alloying elements, but the specific magnesium and silicon content is not known, but is within the same limits as for the two other, ref. Fig. 1.1.

Unfortunately, the manufacturing of tensile test specimens from the AA6060-L and AA6060-H alloy have been delayed during the work for this thesis, such that only a limited number of test specimens were delivered. Quasi-static tests have been conducted for a complete range of temperatures for the AA6060-L and AA6060-H alloys, while for split-Hopkinson tension bar tests, the number of test specimens were not sufficient to conduct an experimental program for a complete range of temperatures and strain rates. A limited number of quasi-static tests have also been conducted for the AA6060-OLD alloy.

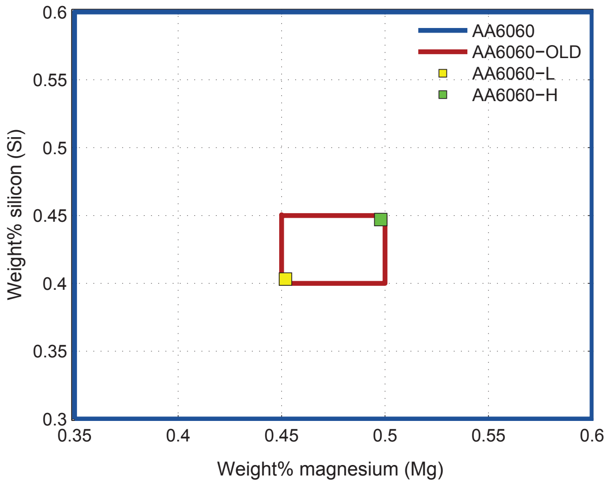


Figure 1.1: Overview of the silicon and magnesium content of the aluminum alloys studied

Section 2, Theoretical Background: Gives an introduction to the most relevant theory for the work on this thesis. The emphasize is on the mechanical behavior and the internal structure of metals. A comprehensive presentation of several constitutive relations and material models is also given.

Section 3, Experimental work: Describes in detail the execution of the experimental tests done during the work for this thesis. Both split-Hopkinson tension bar tests and quasi-static tests have been carried out.

Section 4, Experimental Results: Post-processing of experimental data from the experimental work is presented. Results from the post-processing are presented and discussed.

Section 5, Calibration of Material Models: Three material models have been calibrated and fitted for experimental data for a wide range of temperatures and strain rates. Experimental tests for a wide range of strain rates and temperatures was unfortunately not conducted during the work for this thesis, thus the experimental data used for the material model calibration have been obtained earlier, but for a similar alloy.

Section 6, Numerical Analysis: The finite element model of the split-Hopkinson tension bar setup is presented. Simulations with and without damage coupling have been run and the results are presented and discussed.

Section 7, Concluding Remarks: A short summary of the results obtained is presented and discussed.

Section 8, Further Work: Suggestions for further work related to the work done for this thesis are given.

2 Theoretical Background

2.1 Mechanical Behavior of Metals

2.1.1 Elasticity and Plasticity Theory

Elastic material behavior is both reversible and path independent. Reversibility means that there exists a unique dependence between the stress and strain, such that the strains are recovered and no permanent physical change is present after the material is unloaded. Path independence is meant by that the stored elastic energy does not depend on the strain path [25].

However, for an elastic-plastic material subjected to plastic deformations, the material behavior is both irreversible and path dependent. When a material is deformed into the elastic-plastic region, some of the strains will not be recovered after unloading. This is due to permanent, physical changes on atom level, and the irreversible strains are denoted plastic strains. The material behavior is path dependent because the behavior does depend on the straining history, such that there exists no unique relationship between the stress and strain [25].

2.1.2 Strain Measures

There exists several strain measures that are applicable to a variety of applications and analysis. For a linear analysis, a linear strain measure such as the engineering strain will express the strains adequately. For a nonlinear analysis, a finite strain measure is needed and must be able to represent local deformations for large deformations. In such analysis, a body may be subjected to both large deformations and large rigid body motions, thus the strain should vanish for arbitrary rigid body translations and rotations. In addition, the strain must reduce to the infinitesimal strains if it is linearized, i.e. when the nonlinear strain terms are neglected [31].

The Almansi strain, Green strain and true (logarithmic) strain are examples of finite strain measures. When having to decide which strain measure to adopt, it is essential that the measure is able to represent realistic finite strain values. For large strain deformation analysis, the strain value should tend to go to $-\infty$ for full compression and ∞ for infinite elongation. The different strain measures are expressed and illustrated in Table 2.1 and Fig. 2.1 where L_0 refers to the initial length and L is the current length. As seen, only the true (logarithmic) strain measure is able to express realistic values for large strain deformations.

Finite strain measure	Definition	Zero strains for arbitrary rigid body motions	Reduce to infinitesimal strains if it is linearized	$-\infty$ for full compression	∞ for infinite stretching
Engineering	$\varepsilon_e = \frac{L-L_0}{L_0}$	$\surd$	$\surd$	χ	$\surd$
Logarithmic	$\varepsilon_t = ln(\frac{L}{L_0})$	$\surd$	$\surd$	$\surd$	$\surd$
Green	$\varepsilon_g = \frac{L^2-L_0^2}{2L_0^2}$	$\surd$	$\surd$	χ	$\surd$
Almansi	$\varepsilon_a = \frac{L^2-L_0^2}{2L^2}$	$\surd$	$\surd$	$\surd$	χ

Table 2.1: Comparison of different strain measures

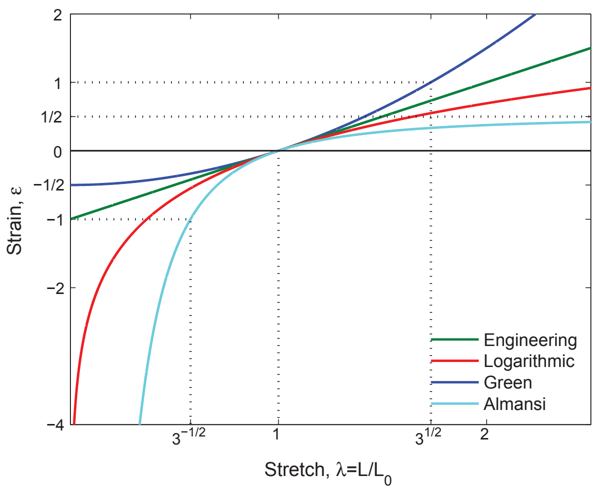


Figure 2.1: Comparison of different strain measures

The logarithmic strain and true stress can be expressed in terms of the engineering strain and stress. The following equations describe the necessary relations and the different properties refer to a typical tensile test where L_0 is the initial length, L is the current length, F is the axial tensile load, A_0 is the initial cross sectional area, A is the current cross sectional area and $u = L - L_0$ is the displacement [25].

The engineering strain ε_e, as defined in Table 2.1, is written as:

$$\varepsilon_e = \frac{L - L_0}{L_0} \tag{2.1}$$

The engineering (or nominal) stress σ_e is defined as the axial force divided by the initial area A_0:

$$\sigma_e = \frac{F}{A_0} \tag{2.2}$$

For large deformations it will be necessary to account for geometrical changes of the specimen, thus defining the strain increment with respect to the current length rather than the original, such that:

$$d\varepsilon_t(t) = \frac{du(t)}{L} \tag{2.3}$$

Integration of the strain increment gives an expression for the true (logarithmic) strain:

$$\varepsilon_t = \int_0^u \frac{du}{L} = \int_{L_0}^L \frac{dL}{L} = ln\left(\frac{L}{L_0}\right) = ln(1+\varepsilon_e) \tag{2.4}$$

As for the true strain measurement, the true stress σ_t takes geometrical changes into account, such that the axial force is divided by the current area rather than the initial area:

$$\sigma_t = \frac{F}{A} = \frac{F}{A_0}\frac{A_0}{A} = \sigma_e e^{\varepsilon_t} = \sigma_e(1+\varepsilon_e) \tag{2.5}$$

Eq. (2.5) have been derived assuming volume constancy, such that $A_0 L_0 = AL$.

2.1.3 Necking and the Effect of Non-Uniformities of Stress at Neck

The relations derived in Section 2.1.2 are based upon the assumption that the deformation is uniform throughout the whole length of the considered body. For a specimen stretched in tension, this assumption is only valid until the point of necking which implies a rapid localized deformation of the cross sectional area somewhere along the specimen with increased elongation. Necking is an instability phenomena, and at the onset of necking, the strains can no longer be assumed to be uniform within the considered body [15]. As the applied force reaches its maximum value, the neck will be initiated, hence the point of neck initiation can be found as the point of maximum stress state along the engineering stress-strain curve. This point can therefore be found by setting the derivative of $\sigma_e = F/A_0$ equal to zero [25].

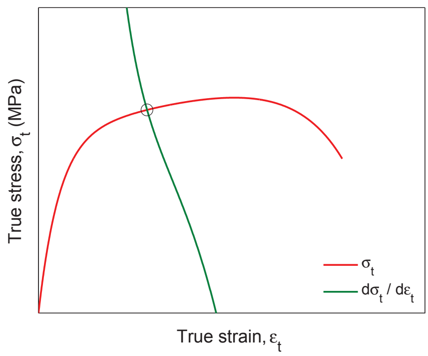
By using the definition of true stress in Eq. (2.5) and the chain rule, the incremental change of the engineering stress can be found:

$$d\sigma_e = d\sigma_t e^{-\varepsilon_t} - \sigma_t e^{-\varepsilon_t} d\varepsilon_t = (d\sigma_t - \sigma_t d\varepsilon_t)e^{-\varepsilon_t} \tag{2.6}$$

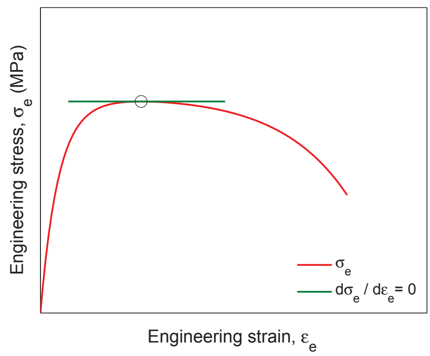
As the maximum value of the engineering stress is found when $d\sigma_e = 0$, the point of neck initiation can be found when

$$\frac{d\sigma_t}{d\varepsilon_t} = \sigma_t \tag{2.7}$$

Fig. 2.2 shows the definition for the initiation of necking for both a true stress-strain curve and an engineering stress-strain curve. It should be noted that Fig 2.2a illustrates a true stress-strain curve determined directly from an engineering stress-strain curve using Eqs. (2.4) and (2.5), such that only the values until the point of necking are valid.



(a)



(b)

Figure 2.2: (a) Definition for initiation of necking for a true stress-strain curve and (b) the same definition only for an engineering stress-strain curve

Considering a tensile specimen subjected to tensile loading, a complex tri-axial stress state arises in the necked area when subjected to continued straining [15]. The average true stress at the neck, defined as $\sigma_t = F/A_{min}$, where F is the axial tensile load and A_{min} is the minimum cross sectional area of the specimen at the neck, will be overestimated compared to the stress required to cause plastic flow when considering tension load only [15].

Bridgman [12] carried out a mathematical analysis of the total stress distribution at the neck in 1952 by taking into account transverse stresses and the geometry of the neck. The equivalent stress distribution $\sigma_{eq}(r)$ at the neck is obtained as [12]:

$$\sigma_{eq}(r) = \frac{1}{1 + ln(\frac{a^2+2aR-r^2}{2aR})}\sigma_t \tag{2.8}$$

where a is the minimum radius of the specimen at the neck, R is the radius of the curvature of the neck, r is the radial coordinate and σ_t is the average true stress at the neck.

An expression for the average equivalent stress at the neck was also obtained by Bridgman [12]:

$$\sigma_{eq} = \frac{1}{(1 + \frac{2R}{a})ln(1 + \frac{a}{2R})}\sigma_t \tag{2.9}$$

The latter expression is the one that will be used later in this thesis to correct the stress state in the smallest cross section at the neck. It should be noted that the mathematical analysis by Bridgman was based on several assumptions; the shape of the neck can be approximated by the arc of a circle, the cross section of the neck has a circular shape during the whole test, and the strain distribution over the minimum cross section is constant [15]. The applicability of the formulas by Bridgman rely on the possibility to be able to measure the radius of curvature and the minimum radius of the cross section at the neck.

2.1.4 Rheological Model

In order to describe material behavior for a wide range of strain values, it is necessary to know the material dependency of strain rate and temperature for both elastic and plastic straining. This can be illustrated by establishing a rheological model where springs, viscous dashpots and friction elements represent respectively elastic behavior, strain rate dependency and strain hardening. Young's modulus for aluminum is found to be independent of strain rate from experiments with strain rates ranging from quasi-static testing to dynamic testing where strain rates of approximately $10^6 s^{-1}$ was reached [32]. However, Young's modulus is found to be strongly dependent on temperature and the relationship can be represented by Eq. (2.10) and is illustrated in Fig. 2.3 [23]:

$$E = -3.9e^{0.0033T} + 79 \tag{2.10}$$

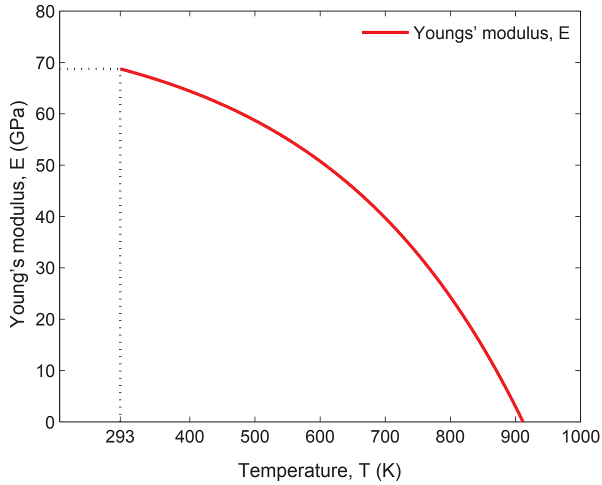


Figure 2.3: Young's modulus as function of temperature from Eq. (2.10)

For large strains, on the other hand, plastic flow will be highly dependent on both temperature and strain rate. Strain hardening, also referred to as work hardening, is also present and is illustrated by the friction element that will have higher resistance for increased plastic straining. The thermoelastic-thermoviscoplastic material behavior for aluminum can be represented with the rheological model shown in Fig. 2.4.

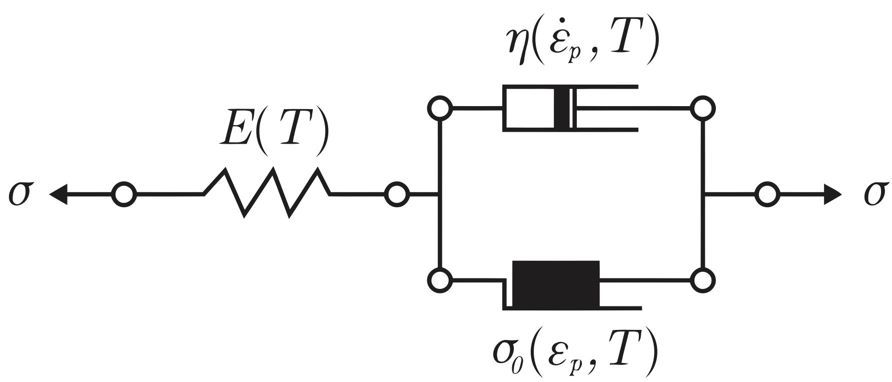


Figure 2.4: Thermoelastic-thermoviscoplastic rheological model [26]

2.2 Internal Structure of Metals

When examining the internal structure of metals, the structural composition can be subdivided into macrostructure, mesostructure, microstructure and atomic arrangement. Macrostructure refers to what can be seen with the naked eye, while an optical microscope is normally used for studying the mesostructure by a magnification of 50 to 1000 times. Using an optical microscope, heterogeneities from alloying elements or naturally occurring impurities can be seen (microstructure). The atomic structure describes how the atoms are arranged relative to each other. Fig. 2.5 illustrates the structural composition of aluminum at different magnifications for both cold worked and fully annealed samples.

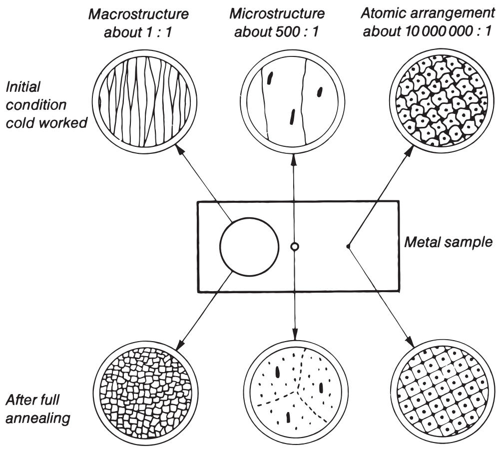


Figure 2.5: Internal structure of aluminum at different magnifications [6]

2.2.1 Bonding Between Atoms

Material properties of solid metals, such as Young's modulus and the yield stress, are very much determined by the bonds holding atoms together and the way in which atoms are packed together. The interatomic bonds are the forces that act as "springs" to link the different atoms together in solid state, while the atom packing defines the density of atoms and therefore also the "density of springs" in metals. Atoms can be bound together by primary bonds or secondary bonds. Primary bonds are the strongest, and these are either ionic, covalent or metallic. Secondary bonds are either Van der Waals or hydrogen bonds and are in comparison much weaker. Most metals, including aluminum, are held together by metallic bonds. For such materials, the highest energy electrons tend to free themselves from their original atoms, thus the atom becomes an ion. These free electrons will then wander freely having no special attachment to any of the ions, as illustrated in Fig. 2.6, and thus give rise to the interatomic forces. The free wandering of electrons is also the reason for the excellent electrical conductivity found in metals [10].

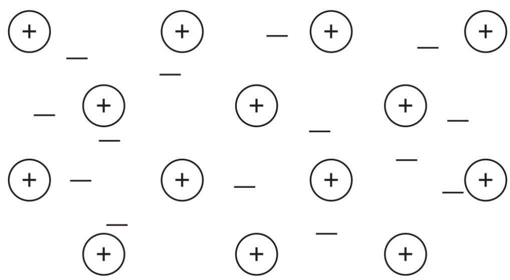
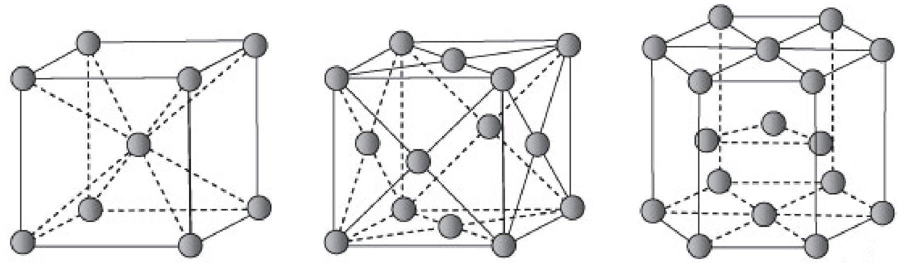


Figure 2.6: Illustration of the free wandering of electrons [10]

2.2.2 Crystal Unit Cell Structures

More than 90% of all solids, either naturally occurring or artificially prepared, have crystalline structure. This type of structure can be described as being a periodic and repeating structure; a specific arrangement is repeated. A crystal is made up of repetitions of unit cell structures, thus unit cells can be considered as the building blocks for a crystal. Material characteristics and physical properties are also associated with the properties of the unit cell structure. Unit cells are always made up of atoms at its corners, and may also have additional atoms at the center of the faces or in the middle of the cell itself [3, 13].

Even though there are 14 different types of crystal unit cell structures, most metals have unit cell structures described as either body-centered cubic (BCC), face-centered cubic (FCC) or hexagonal close packed (HCP). In general, BCC metals, e.g. iron (Fe), are usually less ductile but stronger. FCC metals, e.g. copper (Cu), gold (Au) or aluminum (Al), are often both soft and ductile, while HCP metals, e.g. Zinc (Zn), are usually brittle. These different material characteristics lead to various suitable applications and designs. For instance, soft and ductile FCC metals can more easily be bent and shaped, while HCP metals will be less suited for bending because of their brittle behavior. Examples of other characteristics and properties that rely on the type of crystal structure are material density, deformation processes and alloying behavior [3, 13]. Fig. 2.7 illustrates the different crystal unit cell structures with respect to the atomic arrangement.



(a) (b) (c)

Figure 2.7: (a) Body-centered cubic (BCC), (b) face-centered cubic (FCC) and (c) hexagonal close packed (HCP) unit cell structure [2]

Permanent displacement of metal atoms occurs during plastic deformation by four primary mechanisms: slip, twinning, grain boundary sliding and diffusional creep [35]. Slip is by far the most important deformation mode, and may be defined as the parallel movement of two adjacent crystal regions relative to each other across some plane (or planes) [35]. A slip system is the combination of a plane and a direction lying in the plane where slip occurs. Slip usually occurs on the most close packed planes, while the slip directions are always in the direction of the closest packing [35].

The body-centered cubic unit cell is made up of one atom in the middle and atoms at each corner, eight corners in total. Each of the corner atoms will also be the corner of another unit cell, such that eight unit cells share the same corner atoms. As a result of this, the net total of atoms is two in a BCC unit cell. Compared to the FCC and HCP unit cell structure, the BCC structure does not allow the atoms to pack together as closely. For this type of structure there are no close packed planes, only close packed directions. The {110} planes contain the highest atomic density, and for each six of these planes there are two (111) close packed directions, thus a total of 12 slip systems of {110} and (111) [3, 13].

The face-centered cubic unit cell is made up of atoms at the centers of all the faces and atoms at each corner. In similar way as for the BCC structure, eight corner atoms is shared among eight other unit cells, but the face centered atoms are also shared by an adjacent unit cell. The net total of atoms for this structure is therefore four. In comparison to the BCC structure, atoms pack more closely together in the FCC structure. This type of structure has four {111} close packed planes with three corresponding (110) close packed directions, thus a total of 12 slip systems of {111} and (110) [3, 13].

The hexagonal close packed unit cell is made up of three layers of atoms. At the top and bottom, six atoms are arranged in the shape of a hexagon in addition to one atom in the middle, while in the middle layer three atoms are placed in a triangular fashion. The net total of atoms for this structure is six, compared to two atoms for the BCC structure and four for the FCC structure. The HCP structure has three {001} close packed planes with only one corresponding (2TTO) close packed direction, such that there exists only three slip systems of {001} and (2TTO) [3, 13].

2.2.3 Stacking Sequences

An atomic plane where atoms are packed in a triangular fashion is called a close-packed plane, and a crystal is made up of several atomic planes with identical packing pattern lying on top of each other. The atoms take up the least volume when placed in between the depressions between neighboring atoms, thus this structure is referred to as a close-packed structure. The BCC structure does not have a stacking sequence as it does neither have close-packed planes. FCC structures, on the other hand, will stack in an ABCABC... sequence, where A, B and C corresponds to atom center sites relative to a close-packed layer. For this particular stacking sequence, the fourth atomic plane is therefore being placed directly above the first plane. HCP structures will stack in an ABAB... sequence, such that the third atomic plane is placed directly above the first plane. Fig. 2.8 illustrates how close-packed planes are stacked in ABCABC... and ABAB... sequences for FCC and HCP structures respectively [10, 13].

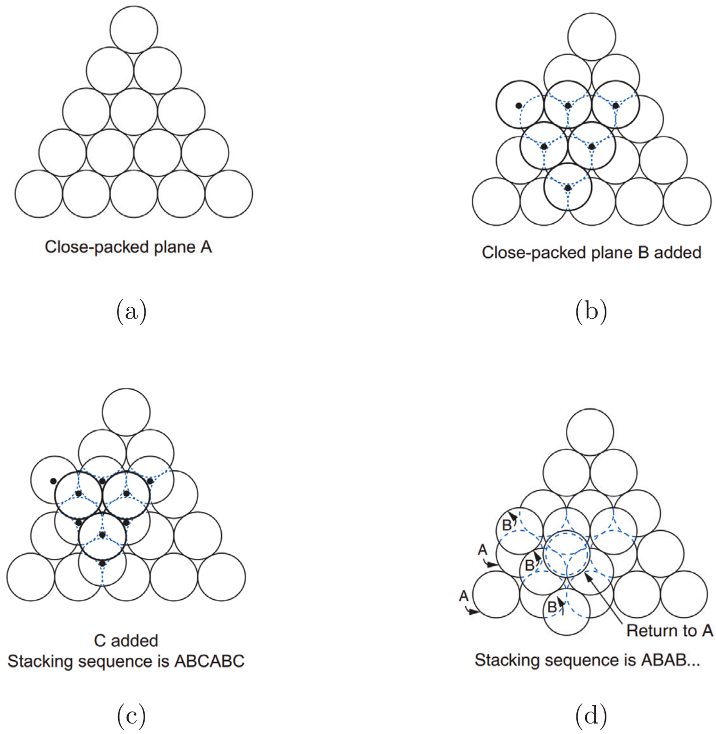


Figure 2.8: (a) Close-packed plane A, (b) close-packed plane B added, (c) ABCABC... stacking sequence and (d) ABAB... stacking sequence [10]

The atomic structure is decided by that arrangement that gives the least energy, such that the structure may in fact not be close packed, or even geometrically simple, but a repeating three-dimensional pattern is needed for it to be a crystal. The energy difference between various packing structures may be very small, such that by heating a metal, the atomic structure may change and give rise to altered material properties [10].

2.2.4 Dislocation Mechanisms

A pure metal will in general contain numerous defects in the crystal structure that can be classified as point defects, line defects or plane defects. Dislocations are the only line defect and the main reason for the strain hardening behavior seen in metals [35]. This Section is a short introduction to dislocation mechanisms.

Even though crystal structures are made up of atoms packed together in a regular and repeating pattern, they are in fact not perfect. Dislocations in crystals are defects in the structure that very much determines the yield stress and also the plastic deformation behavior for metals. Plastic straining is associated with permanent and physical changes in the atom structure, and is a direct result of dislocation motion and rearrangement of atoms within the structure. For dislocations to move, the shear stress that exerts the force on the dislocations must be large enough to overcome its resistance to movement, thus the force needed for yielding to take place will increase as the resistance to movement increases. There are two fundamental types of dislocations; edge dislocations and screw dislocations [35]. Fig. 2.9 shows the motion sequence from the introduction of an edge dislocation into a crystal on the left side and to its expulsion on the right side. As can be seen, the lower part of the crystal is displaced a distance b, the Burgers vector, relative to the upper part. Such locations are also referred to as line defects because the locus of defective points produced by the dislocation in the lattice lie along a line [35]. Screw dislocations are much more difficult to visualize geometrically, but it can be illustrated by that atom planes are converted into a helical surface, and the most significant difference relative to edge dislocations is that the Burgers vector is parallel to the dislocation line (perpendicular to the dislocation line for edge dislocations) [10, 35]. All dislocations in crystals are either edge dislocations, screw dislocations or a combination of the two [10].

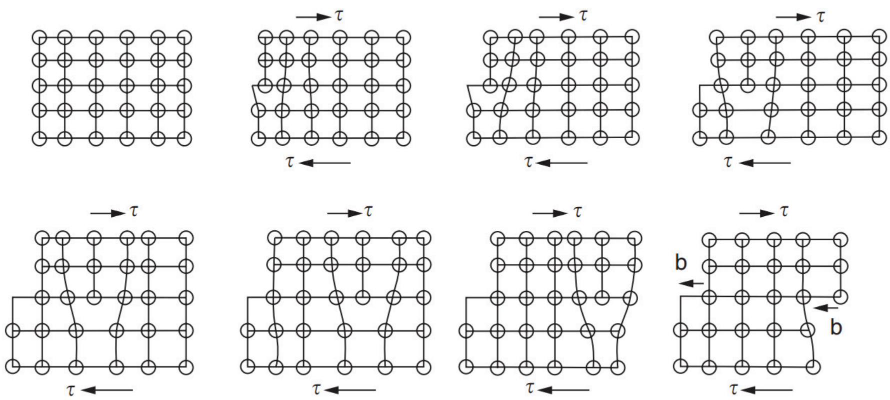


Figure 2.9: Complete motion sequence of an edge dislocation in a crystal [10]

2.2.5 Ductile Fracture and Nucleation and Growth of Voids

Nucleation, growth and the coalescence of microscopic voids (pores) that are initiated at inclusions and second-phase particles are usually the reason for ductile fracture in metals [9]. A tensile specimen of a very high purity material may neck down to a sharp point, such that extremely large plastic strains and close to 100% reduction of the cross sectional area are observed. However, materials containing impurities will experience fracture at lower strains due to the nucleation, growth and coalescence of voids. Fig. 2.10 illustrates the nucleation and growth of voids in a material subjected to loading. The theory of fracture mechanics has only been touched briefly upon in this Section to clarify some terms.

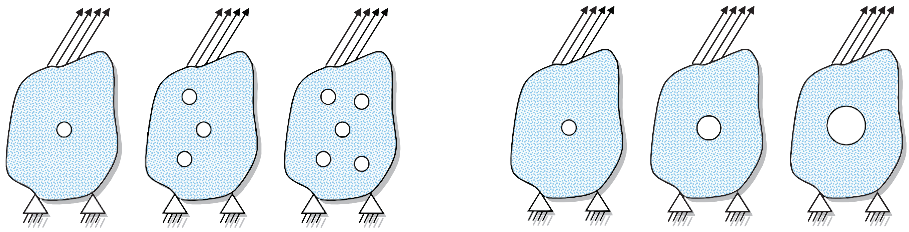


Figure 2.10: (a) Void nucleation and (b) void growth [24]

2.3 Aluminum

Aluminum is a versatile metal with unique characteristics, and has very much substituted other established metals such as copper, steel and iron in a lot of applications. Some of its characteristics are light weight, high strength-to-weight ratio, resistance to corrosion and reasonable cost. In its purest form, aluminum has a density of $2.7\frac{g}{cm^3}$, Young's modulus of $69 - 72GPa$ and a yield strength of $15 - 20MPa$ [21], but by adding alloying elements and by undergoing heat treatment, material properties and characteristics such as yield strength, ductility and workability can be significant altered to specific needs. Aluminum is also easy to form and can be produced in many different shapes, such as rolled plates, sheets, foils, castings and cables. As a result of this, aluminum is now being used in a variety of industries, ranging from automotive and aerospace manufacturing to building and highway structures [6].

2.3.1 Alloy Designation

Aluminum alloys can be divided into two major categories: casting alloys and wrought alloys. Casting alloys contain a greater amount of alloying elements than wrought alloys and are used for cast parts, while wrought alloys are suitable for forming processes such as rolling and extrusion [6]. There exists a widely used and recognized alloy designation system created and maintained by the Aluminum Association [1], and this system is in fact recognized by about 90% of the world's aluminum industry [30]. Four numerical digits are used as notation: first digit identifies the alloy group (major alloying elements), second digit defines modifications and impurity limits, while the two last digits are used to differentiate the alloys within the same series. A summary of the designation system together with alloy characteristics for wrought alloys is reproduced in Table 2.2.

Alloy	Main alloying element(s)	Basic behavioral and performance characteristics
1xxx	Mostly pure aluminum	Low strength, extremely high ductility and formability, exceptionally high electrical conductivity and corrosion resistance
2xxx	Copper [Cu]	Relatively high strength, lower ductility, less resistant to corrosion, not readily welded, heat treatable
3xxx	Manganese [Mn]	Modest strength increase, relatively high formability and ductility, very high resistance to corrosion, readily weldable
4xxx	Silicon [Si]	Low to medium strength, less resistant to corrosion, excellent flow and finishing characteristics, readily welded, some alloys are heat treatable
5xxx	Magnesium [Mg]	Very high strength, exceptionally tough, readily welded, excellent corrosion resistance
6xxx	Magnesium [Mg] and silicon [Si]	Very high strength, excellent corrosion resistance, easy to extrude, readily welded, heat treatable
7xxx	Zinc [Zn]	Can provide the highest strengths of any alloy, less resistant to corrosion, less tough and susceptible to cracking, heat treatable
8xxx	Other elements	Contains less frequently used alloying elements such as iron [Fe] and tin [Sn], characteristics depend on the major alloying element(s)
9xxx	Unassigned	-

Table 2.2: A summary of the wrought alloy designation system [30]

2.3.2 Temper Designation

A temper designation is usually presented right after the alloy designation, referring to what treatment the alloy has undergone during production, and is made up of a letter and one or more digits, e.g. 6060-T651. The letter represents the general class of treatment, and the digits are used to further categorize the basic tempers into subdivisions. A summary of the different treatment classes is reproduced in Table 2.3.

Class	Description
F - fabricated	Either wrought or cast products, no control over thermal conditions or strain-hardening processes to obtain specific material characteristics
O - annealed	Annealed wrought products to increase the workability but reduced strength, annealed cast products to improve ductility
H - strain hardened	Products with increased strength through strain hardening, may also be subjected to thermal treatments
W - solution heat treated	Alloys that age instantly after solution heat treating
T - thermally treated	Thermally treated to produce stable tempers, may also be subjected to strain hardening

Table 2.3: A summary of the temper designation treatment classes [30]

2.4 Constitutive Relations and Fracture Criteria

The material flow behavior for metals, e.g. in crash and explosion situations where both high strain rates and temperatures may be present, is rather complex and cannot easily be completely and accurately described. Strain, strain rate, temperature and microstructural development will influence on the hardening and softening mechanisms controlling the material flow behavior [28]. There has been proposed several constitutive relations that aim to describe the material flow stress, and these vary in both applicability and nature of origin. A constitutive relation describes the relation between the stress and strain tensor by taking into account one or more attributes. Depending on what potential applications the material model is designed for, it may include attributes such as strain hardening, strain rate effects, thermal effects and anisotropy or orthotropy, and may be applicable to specific materials such as composites, metals or biological materials [20].

In general, constitutive relations for describing material flow can be divided into models of phenomenological nature and physical models based on dislocation mechanics that may also take thermodynamics into account. Several models referred to as physical models are in fact semi-physical, hence they do not completely describe the physical atomic behavior and interaction. The material constants for a phenomenological model cannot be interpreted physically, while for a (semi-)physical model the constants may be defined and interpreted by microstructural parameters. The Johnson-Cook constitutive relation is a widely used and popular phenomenological material model. Several models based on microstructural dislocation mechanics have been proposed by Voyiadjis, Abed, Zerilli and Armstrong (e.g. Voyiadjis and Abed (2005) [38], Abed and Voyiadjis (2005) [4] and Zerilli and Armstrong (1987) [39]). Both the phenomenological model by Johnson and Cook and several semi-physical models will be presented in this Section and later evaluated for a wide range of strain rates and temperatures in Section 5.

For a certain type of applications, the complete material model must also represent fracture. This can be done by operating with two separate models, whereas one representing the plastic flow and the other fracture. These two models can either be coupled or uncoupled [18]. The Cockroft-Latham fracture criterion is a rather simple criterion that will be presented in this Section and later used in numerical simulations in Section 6.

2.4.1 Johnson-Cook Constitutive Relation

A widely used and popular constitutive model of phenomenological nature has been proposed by Johnson and Cook [27] that involves rather few parameters and has proven to be well-suited for numerical simulations of static and dynamic analysis. The main advantage of the model is that it can be calibrated rather easily with a minimum of experimental data, and it is able to predict the flow stress at different strain rates and temperatures. However, these two parameters are uncoupled which implies that the strain rate hardening will be independent of the temperature. For most metals this is in fact not the case, as it has been found that the strain rate sensitivity increases with increasing temperature and the resulting decrease of flow stress [38]. The Johnson-Cook model associates the effective von Mises flow stress with the equivalent plastic strain, strain rate and temperature, and is given on the original form as [27]:

$$\sigma = (A + B\varepsilon_p^n)(1 + Cln\dot{\varepsilon}_p^*)(1 - T^{*m}) \tag{2.11}$$

where ε_p is the equivalent plastic strain, $\dot{\varepsilon}_p^* = \dot{\varepsilon}_p/\dot{\varepsilon}_0$ is the dimensionless plastic strain rate where $\dot{\varepsilon}_0$ is a user-defined reference strain rate typically set to the strain rate from quasi-static tests, and $T^* = (T - T_r)/(T_m - T_r)$ is the dimensionless homologous temperature where T is the actual temperature, T_r is the reference temperature typically set to the ambient temperature in the laboratory, and T_m is the melting temperature of the material. A, B, n, C and m are material constants that needs to be determined. The individual expressions in the three sets of brackets represent respectively the strain hardening, strain rate hardening and thermal softening and can all be calibrated separately [18]. In the situation of very small strain rates, hence static conditions, the logarithmic function $ln\dot{\varepsilon}_p^*$ in Eq. (2.11) will approach $-\infty$ and thus result in numerical difficulties. To avoid this, a modified version of the Johnson-Cook constitutive relation can be written as [11]:

$$\sigma = (A + B\varepsilon_p^n)(1 + \dot{\varepsilon}_p^*)^C(1 - T^{*m}) \tag{2.12}$$

The same parameters and material constants are used in Eq. (2.12) as in Eq. (2.11), but the constant C will take on a different value due to the altered formulation. In Eq. (2.12), the strain hardening part is defined as $B\varepsilon_p^n$, namely the power (or Ludwig) law, but it may also be defined by Voce rule [25]. When replacing the power law with Voce rule involving two terms, Eq. (2.12) transforms into:

$$\sigma = (A + \sum_{i=1}^{2} Q_i(1 - e^{-C_i \varepsilon_p}))(1 + \dot{\varepsilon_p}^*)^C(1 - T^{*m}) \tag{2.13}$$

where Q_1, C_1, Q_2 and C_2 are material constants that needs to be determined.

2.4.2 Introduction to Microstructural Based (Semi-)Physical Models

Seen from a microstructural point of view, the inelastic behavior and the material flow stress for various strain rates and temperatures are very much closely linked to the dislocation mechanics of the material. As the material is loaded into the inelastic region, dislocations are generated, moved and stored within the crystal structure. As dislocations move through the crystal, plastic strains are generated and the material has as a result exceeded its elastic limit. Dislocations can be classified into statistically stored dislocations and geometrically necessary dislocations. The former type are dislocations stored and trapped in a random way, while the latter are dislocations that are stored in a specific geometric pattern to maintain the continuity of various components of the material [4, 38].

Two different types of obstacles will try to prevent any further movement through the lattice for a dislocation, namely long-range and short-range (Peierls) barriers. Long-range barriers arise as a result of the material structure and cannot be overcome by introducing thermal energy, while short-range barriers can so. Overcoming long-range barriers will therefore contribute to the total flow stress with a stress component that is not thermally activated, an athermal stress component, while overcoming short-range barriers will contribute with an thermal stress component. Thus, the material flow stress can be additively decomposed into [38]:

$$\sigma = \sigma_{ath} + \sigma_{th} \tag{2.14}$$

where σ_{ath} is the athermal component and σ_{th} is the thermal component. The assumption of this decomposition has been proven through experiments and is stated by several authors [38].

The equivalent plastic strain rate, $\dot{\varepsilon}_p = (2\dot{\varepsilon}^p_{ij}\dot{\varepsilon}^p_{ij}/3)^{0.5}$, can be related to the mobile dislocation density ρ_m, the dislocation speed v and the magnitude of Burgers vector b (ref. Section 2.2.4) through Orowan's equation [38]:

$$\dot{\varepsilon}_p = \tilde{m} b \rho_m v \tag{2.15}$$

where $\tilde{m}$ is the Schmidt orientation factor and is a material constant. According to Voyiadjis and Abed [38], the following equation has been suggested by Kubin and Estrin to describe the mobile dislocation density evolution:

$$\dot{\rho}_m = (\lambda_1/b^2 - \lambda_2\rho_m - \lambda_3\sqrt{\rho_f}/b)\dot{\varepsilon}_p \tag{2.16}$$

where λ_i are constants related to the multiplication of mobile dislocations (λ_1), their mutual annihilation and trapping (λ_2) and their immobilization through interaction with forest dislocations (λ_3), and ρ_f is the forest dislocation density. An equation for describing the evolution of the forest dislocation density was also presented by the same authors. According to Voyiadjis and Abed [38], an expression for the average dislocation velocity v has been suggested by Bammann and Aifantis:

$$v = v_0 exp(-G(\tau)/kT) \tag{2.17}$$

where $v_0 = d/t_w$ is the reference dislocation velocity, where t_w is the time period a dislocation waits at an obstacle and d is the average distance the dislocation moves between the obstacles, G is the Gibbs free energy of activation that is a function of shear stress, temperature and the internal structure, k is Boltzmann's constant, and T is the absolute temperature. A relation for Gibbs free energy of activation can be obtained by utilizing Eqs. (2.15)-(2.17) and the definition for the evolution of the statistically stored dislocation density $\dot{\rho}_{ss}$ and the plastic flow rate $\dot{\varepsilon}_p$ that is further discussed in a paper by Voyiadjis and Abed [38]:

$$G = (ln(\frac{\tilde{m}b\rho_m v_0}{\tilde{m}bl\lambda_2\rho_m + \tilde{m}l\lambda_3\sqrt{\rho_f} - \tilde{m}l\lambda_1/b + 1}) - ln\dot{\varepsilon}_p)kT \tag{2.18}$$

According to Voyiadjis and Abed [38], the Gibbs free energy of activation can also be related to the thermal flow stress σ_{th} as suggested by Kocks et al.:

$$G = G_0(1 - (\frac{\sigma_{th}}{\hat{\sigma}})^p)^q \tag{2.19}$$

where G_0 is the reference Gibbs energy at $T = 0K$, $\hat{\sigma}$ is the threshold stress, i.e. the stress state where dislocations may overcome barriers without thermal activation, and p and q are constants associated with the short-range barrier shape.

The temperature and strain rate dependency for the activation volume has been investigated by several researchers, for which no common conclusion was obtained. However, it seems to be a common assumption that the activation volume decreases for increased plastic straining for FCC metals, while it is being essentially constant and independent of plastic strains for BCC metals. As a result of this, the thermal stress relation will be different for materials having these two types of unit cell structures [38].

It has been found that for metals, the flow stress will be decreasing for increasing temperature until a critical temperature value is reached, for which no further decrease of flow stress is obtained. The flow stress at this point can be addressed as the athermal stress and is independent of the strain rate, but the critical temperature value will be stain rate dependent [4].

2.4.3 Microstructural Based Models for BCC and FCC Metals

Voyiadjis and Abed [38] have derived semi-physical based constitutive relations for both BCC and FCC metals based on the concept of thermal activation analysis. The derivation of these relations has been studied and will be presented shortly here, while a more thoroughly presentation can be found in the original paper by the authors [38].

Athermal component for BCC metals:

It is found that the plastic strain hardening is almost independent of strain rate and temperature for BCC metals, such that it contributes to the athermal part only. According to Voyiadjis and Abed [38], Nemat-Nasser and Guo studied BCC metals and indicated that the athermal resistance to dislocation movement is linked to the stress caused directly by dislocations, point defects, grain boundaries and other impurities found in the material. They suggested that the elastic strain could be used to define the mentioned reasons for the dislocation movement resistance since the plastic strain increases monotonically and the plastic strain rate is always positive. The athermal flow stress component can therefore be defined as [38]:

$$\sigma_{ath} = Y_a + B_1 \varepsilon_p^{n_1} \tag{2.20}$$

where Y_a is the athermal yield stress and B_1 and n_1 are athermal hardening parameters.

Athermal component for FCC metals:

The yield stress is found to be not affected by either temperature or strain rate for most FCC metals, such that the stress-strain curve will have the same starting point for different temperatures and strain rates when the material has not been subjected to previous plastic straining. The athermal component can be expressed by the initial athermal yield stress Y_a only [38]:

$$\sigma_{ath} = Y_a \tag{2.21}$$

Thermal component for BCC metals:

It is found that the plastic yield stress for BCC metals is strongly dependent on both temperature and strain rate, and the deformation mechanism is closely linked to the resistance of the dislocation motion by the short-range Peierls barriers which are responsible for the thermal activation analysis behavior. The expression for the thermal yield stress can be found by utilizing Eqs. (2.18) and (2.19) and solving for the thermal flow stress σ_{th} [38]:

$$\sigma_{th} = \widehat{Y}(1 - (\beta T)^{1/q})^{1/p} \tag{2.22}$$

where $\widehat{Y}$ is the threshold yield stress for dislocations to move through the Peierls barriers and β is defined as $\beta = \beta_1 - \beta_2 ln\dot{\varepsilon}_p$ where β_1 and β_2 are defined as:

$$\beta_1 = \frac{k}{G_0} ln(\frac{\tilde{m}b\rho_m v_0}{1 - \tilde{m}l\lambda_1/b + \tilde{m}bl\lambda_2\rho_m + \tilde{m}l\lambda_3\sqrt{\rho_f}}) \tag{2.23}$$

and

$$\beta_2 = \frac{k}{G_0} \tag{2.24}$$

The strong dependency on strain rate and temperature for the thermal yield stress for BCC metals can be addressed to the dislocation size and the corresponding concentration of Cottrell's atmosphere. As dislocations are moved through the crystal, their corresponding atmosphere of interstitial atoms will also be moved, such that a drag force arises within the lattice. This drag force will increase with increasing concentration mismatch between the Cottrell's atmosphere and the surrounding solute atoms. As the concentration of solute atoms is dependent on both strain rate and temperature, the yield strength caused by this drag force is too [38].

Thermal component for FCC metals:

For FCC metals, the thermal activation analysis behavior is controlled and dominated by the emergence and evolution of dislocations and the long-range intersections between dislocations, such that the thermal activation is strongly dependent on the plastic strain. From this it is seen that the activation volume, and therefore also the distance d between dislocation intersections, will attribute to the formulation of the thermal flow stress component. The thermal component will therefore be coupled with strain rate, temperature and the plastic strain and is found in the same way as for BCC metals. However, $\hat{\sigma}$ ($\widehat{Y}$ in Eq. (2.22)) is no longer interpreted as the threshold yield stress, but rather the flow stress that is related to both dislocation densities and the strain, such that [4, 38]:

$$\sigma_{th} = \hat{\sigma}(1 - (\beta T)^{1/q})^{1/p} \tag{2.25}$$

where

$$\hat{\sigma} \approx f(b/d) \approx \sigma_0 \varepsilon_p^n \tag{2.26}$$

Resulting constitutive relations

The resulting constitutive relation for BCC metals is found by substituting Eqs. (2.20) and (2.22) into Eq. (2.14):

$$\sigma = \widehat{Y}(1 - (\beta_1 T - \beta_2 T ln\dot{\varepsilon}_p)^{1/q})^{1/p} + B\varepsilon_p^n + Y_a \tag{2.27}$$

The resulting constitutive relation for FCC metals is found by substituting Eq. (2.26) into Eq. (2.25) and utilizing Eqs. (2.14) and (2.21):

$$\sigma = B\varepsilon_p^n(1 - (\beta_1 T - \beta_2 T ln\dot{\varepsilon}_p)^{1/q})^{1/p} + Y_a \tag{2.28}$$

where B (σ_0 in Eq. (2.26)) and n are hardening parameters. As mentioned earlier, and as can be seen from Eq. (2.28), the initial yield stress is independent of both strain rate and temperature for FCC metals. However, this is not always the behavior seen in FCC metals, and by altering the yield stress part to be slightly temperature and strain rate sensitive this problem is overcome [38].

2.4.4 Zerilli-Armstrong Constitutive Relation

A constitutive relation has been proposed by Zerilli and Armstrong [39] that is based on the dislocation mechanics concept, thus being a semi-physical model. Strain, strain rate and temperature are being coupled in the constitutive model, and physical mechanisms such as dislocation density, Burgers vector, dislocation velocity, thermal activation analysis and the influence by solute and grain size has been studied to describe the various stress components that together determine the material flow behavior. As these mechanisms depend on the type of atomic unit cell structure, two models were proposed for BCC and FCC metals respectively and are given on their original form as [39]:

$$\sigma = \Delta\sigma_G^{'} + c_1 exp(-c_3 T + c_4 T ln\dot{\varepsilon}_p) + c_5 \varepsilon_p^n + kl^{-1/2} \tag{2.29}$$

$$\sigma = \Delta\sigma_G^{'} + c_2 \varepsilon_p^{1/2} exp(-c_3 T + c_4 T ln\dot{\varepsilon}_p) + kl^{-1/2} \tag{2.30}$$

where $\Delta\sigma_G^{'}$ is an additional component of stress due to the influence of solute and dislocation density on the yield stress, T is temperature, $\dot{\varepsilon}_p$ is the plastic strain rate, k is the microstructure stress intensity, l is the inverse square root of the average grain diameter, and $c_{1,} c_{2,} c_{3,} c_{4,} c_5$ and n are material parameters. $kl^{-1/2}$ is an incremental stress component that relates to the stress needed for transmission of plastic flow between polycrystal grains and is present for both BCC and FCC unit cell structures. As can be seen from Eqs. (2.29) and (2.30), the component that relates to the dislocation activation area is constant for BCC metals but proportional to $\varepsilon_p^{1/2}$ for FCC metals. Hence, strain rate hardening and thermal softening are increased for increased plastic straining for FCC metals. The strain rate and thermal effects are uncoupled with strain hardening for BCC metals, leading to the addition of a separate plastic strain hardening contribution from the power law.

2.4.5 A Modified Zerilli-Armstrong Constitutive Relation

Voyiadjis and Abed [38] have suggested modified versions of the original Zerilli-Armstrong constitutive relations. A brief summary of their proposed models is presented here, and a more thoroughly derivation can be found in the original paper by the authors [38]. By taking into account that the activation volume is dependent on the temperature and applying the approximation $ln(1 + x) \simeq x$, Voyiadjis and Abed suggested a constitutive relation for BCC metals:

$$\sigma = Y_a + B\varepsilon_p^n + \widehat{Y} exp(-\beta_3 T + \beta_2 T ln\dot{\varepsilon}_p) \tag{2.31}$$

where $\beta_3 = \beta_1 + (1/T)ln\frac{V'}{V_0}$, V_0 is the activation volume independent of both temperature and strain rate, V' is the activation volume dependent on temperature but not strain rate, and β_1 and β_2 are similar to those given in Eqs. (2.23) and (2.24). This model is in fact quite similar to the model presented by Zerilli and Armstrong, but the physical parameter β_1 is interpreted differently. In the approximation $ln(1+x) \simeq x$, x is defined as $(k/G_0)Tln(\dot{\varepsilon}_p/\tilde{m}b\rho v_0)$, and this approximation will not be valid for low strain rates coupled with high temperatures. As a result, β_1, β_2 and β_3 can no longer be interpreted physically and these parameters will in fact be of phenomenological nature. By considering both the temperature and strain rate dependency for the activation volume, in addition to using the exact value of $ln(1+x)$, a modified constitutive relation for BCC metals is presented [38]:

$$\sigma = Y_a + B\varepsilon_p^n + \widehat{Y}(1 + \beta_0 T^m - \beta_1 T + \beta_2 T ln\dot{\varepsilon}_p) \tag{2.32}$$

where m is an exponent constant and $\beta_0 = f(\dot{\varepsilon}_p)$ is a function of the plastic strain rate.

Voyiadjis and Abed have studied how the activation volume correlates to plastic straining and the athermal component's independence on temperature and strain rate for FCC metals, and have suggested the following constitutive relation [38]:

$$\sigma = Y_a + (Y_0 + B\varepsilon_p^n)exp(-\beta_3 T + \beta_2 T ln\dot{\varepsilon}_p) \tag{2.33}$$

As for BCC metals, a physical interpretation of the material parameters β_i cannot be done accurately because of the applied approximation $ln(1+x) \simeq x$. However, if the exact value of $ln(1+x)$ is used, the parameters can be interpreted physically and the model is then defined as [38]:

$$\sigma = Y_a + (Y_0 + B\varepsilon_p^n)(1 + \beta_0 T^m - \beta_1 T + \beta_2 T ln\dot{\varepsilon}_p) \tag{2.34}$$

2.4.6 A Combined Constitutive Relation for both BCC and FCC Metals

As previously discussed, microstructural mechanisms that control the plastic flow behavior are different for BCC and FCC metals. Abed and Voyiadjis [4] have proposed a constitutive relation specifically for the AL-6XN stainless steel alloy taking into account mechanisms for both BCC and FCC metals. The motivation for studying this material model is due to the fact that the AA6060 alloys studied in this thesis also show material behavior typical for both BCC and FCC metals.

As for the other models proposed by Abed and Voyiadjis, the flow stress can be decomposed into a thermal σ_{th} and an athermal σ_{ath} stress component. The thermal part, which is linked to the short-range barriers, includes the Peierls stress, point defects (e.g. vacancies and self-interstitials) and dislocations intersection with slip planes. The athermal part is linked to the long-range barriers and includes the stress field of dislocation forests and grain boundaries and does not depend on either temperature or strain rate, such that there will always be a significant amount of stress in the material independent of temperature and strain rate. Dynamic strain aging effects such as diffusion and creep are not included in the proposed model as they will not be dominant in the considered interval of temperatures and strain rates, thus the flow stress is determined by considering the motion of dislocations only [4].

FCC metals will usually have an athermal stress component that is independent of strain, while BCC metals will have an additional strain dependent component. Zerilli and Armstrong [39] linked the initial dislocation density and the influence of the solute to the strain independent part of the athermal stress. For the model suggested by Abed and Voyiadjis, the athermal flow stress component is defined as [4]:

$$\sigma_{ath} = Y_a + B_1 \varepsilon_p^{n_1} \tag{2.35}$$

where Y_a is the athermal yield stress and B_1 and n_1 are athermal hardening parameters.

As discussed, movement of dislocations for BCC metals is linked to overcoming the short-range barriers. These barriers are overcome by the movement of the original (initial) dislocations, such that the dislocation movement for BCC metals will not be dependent on the accumulation of dislocations associated with increased plastic straining. On the contrary, for FCC metals are the cutting of dislocations forests the main mechanism. This is attributed to both the evolution and accumulation of dislocations, thus the thermal stress will be dependent on the plastic strain. The thermal flow stress component can therefore be decomposed into two parts [4]:

$$\sigma_{th} = Y_{th} + H_{th} \tag{2.36}$$

where Y_{th} and H_{th} are defined as:

$$Y_{th} = \hat{Y}(1 - (\beta_1^Y T - \beta_2^Y T ln \dot{\varepsilon}_p)^{1/q})^{1/p} \tag{2.37}$$

$$H_{th} = B_2 \varepsilon_p^{n_2}(1 - (\beta_1^H T - \beta_2^H T ln \dot{\varepsilon}_p)^{1/q})^{1/p} \tag{2.38}$$

where B_2 and n_2 are the thermal hardening parameters. The parameters β_1^i and β_2^i have the same definition as for those defined in Eqs. (2.23) and (2.24), but β_1^Y is related to the initial mobile dislocation density ρ_m, the initial forest dislocation density ρ_f and the initial dislocation distance l, while β_1^H is related to the average values of these microstructural parameters. It should be mentioned that the value of the activation energy G_0 differs for different mechanisms, such that β_2^Y and β_2^H will not have the same numerical value. As both the mobile and forest dislocation density will not be constant with increased plastic straining, defining β_1^H based on average values will in fact not be accurate. However, this parameter can be estimated by taking into account the plastic strain parameter by the following expression [4]:

$$\beta_1^H \approx \frac{k}{G_0} ln(\frac{v_0 \varepsilon_p}{1 - \tilde{m} l \lambda_1 / b + \lambda_2 \varepsilon_p + l \lambda_3 \lambda_4 \varepsilon_p^{0.5}}) \tag{2.39}$$

One major problem with this formulation is that the parameter β_1^H is no longer constant and independent of the plastic strain, such that the higher complexity of the resulting constitutive relation will make it more complicated to fit the model parameters numerically.

By utilizing Eqs. (2.35), (2.37) and (2.38), the resulting expression for the flow stress is defined as:

$$\sigma = Y_a + B_1 \varepsilon_p^{n_1} + \hat{Y}(1 - (\beta_1^Y T - \beta_2^Y T ln \dot{\varepsilon}_p)^{1/q})^{1/p} + B_2 \varepsilon_p^{n_2}(1 - (\beta_1^H T - \beta_2^H T ln \dot{\varepsilon}_p)^{1/q})^{1/p} \tag{2.40}$$

2.4.7 Cockroft-Latham Fracture Criterion

Ductile fracture in metals is usually caused by nucleation, growth and coalescence of voids in the material that are caused by second phase particles or other imperfections (ref. Section 2.2.5). When a material is subjected to plastic deformation, the number of voids will grow until the voids coalesce to initiate a crack. Cockroft and Latham have proposed an isotropic energy-based fracture criterion that have been adopted in similar studies [16]:

$$W = \int_0^{\varepsilon_p} max(\hat{\sigma}_1, 0) d\varepsilon_p \geq W_{cr} \tag{2.41}$$

where W is the Cockroft-Latham integral, $\hat{\sigma}_1$ is the maximum principal stress and W_{cr} is the critical fracture parameter. In numerical simulations, the element is eroded and all stresses are set to zero when the criterion is fulfilled. The Cockroft and Latham criterion can also be derived for anisotropic materials [15].

The critical fracture parameter can be obtained from:

$$W_{cr} = \int_0^{\varepsilon_f} \hat{\sigma}_1 d\varepsilon_p \tag{2.42}$$

where ε_f is the fracture strain. The fracture strain can be estimated by microscope measurements of the fractured area A_f in the gauge section of a specimen subjected to tension loading [16]:

$$\varepsilon_f = ln(\frac{A_s}{A_f}) \tag{2.43}$$

3 Experimental Work

3.1 The AA6060 Alloy

The aluminum alloys studied and subjected to the experimental tests for this thesis have been processed by Hydro Aluminium. The chemical components are tabulated in Table 3.1 and a microscope image can be seen in Fig. 3.1. According to the manufacturer, the alloy is developed to satisfy a yield strength of $70MPa$, an ultimate tensile strength of $150MPa$ and Brinell hardness of 43 for the T4 temper. All three alloys studied are of the T4 temper. Special properties include good formability, moderate machinability, suitable for all welding methods, good corrosion resistance, well suitable for all types of mechanical surface treatment and very good for anodizing [7]. See Section 1 for specific silicon (Si) and magnesium (Mg) concentrations for the three alloys studied.

As the microscope image (Fig. 3.1) reveals, the individual grains are of circular shape and of random pattern, such that the alloys should have isotropic material behavior. The grains were measured to have an average size of $94.052\mu m$ and $92.694\mu m$ in respectively the horizontal and vertical direction in Fig. 3.1, thus the volume is estimated to be $1.387 \cdot 10^{-7} \mu m^3$ assuming a spherical shape [36].

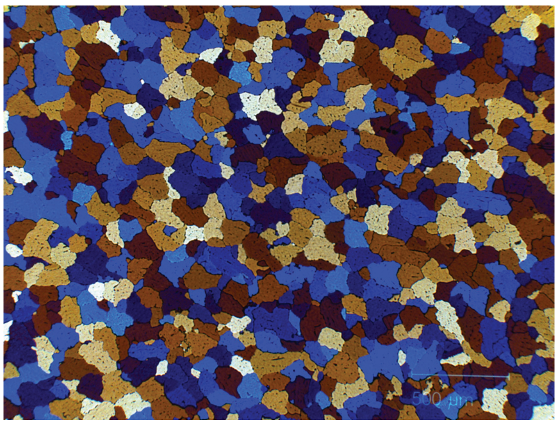


Figure 3.1: Microscope image of the AA6060-OLD alloy [36]

%	Si	Fe	Cu	Mn	Mg	Zn	Ti	Cr	Other elements	Al
Minimum	0.40	0.18	-	-	0.45	-	-	-	-	Balance
Maximum	0.45	0.22	0.02	0.03	0.50	0.02	0.02	0.02	0.10	

Table 3.1: Chemical composition of the AA6060-alloy [7]

3.2 Calculation of Response in Test Specimen from SHTB Tests

This Section aims to present the relations necessary to calculate the response in a test specimen subjected to a typical SHTB test. A thorough derivation of the relations presented here can be found in Appendix C. One-dimensional elastic wave theory is presented in Appendix B and is supplementary theory to better understand the derivations presented in Appendix C.

In order to determine the response, that is the stress and strain state, of the specimen, only two strain gauges mounted on the bars are necessary (strain gauge 2 and 3 in Fig. 3.2). The signals from the strain gauges are resistance changes of the metal filament that is converted to voltage changes through a connected Wheatstone bridge. Strain values are then calculated as:

$$\varepsilon = \frac{2\Delta V}{V_0 k} \cdot \frac{1}{f_a} \tag{3.1}$$

where ΔV is the voltage change measured from the strain gauge, V_0 is the battery output voltage and k is the gauge factor (resistance change/strain gauge elongation proportionality coefficient). The strains are also divided by a parameter f_a due to connected amplifiers to enhance the recorded signal.

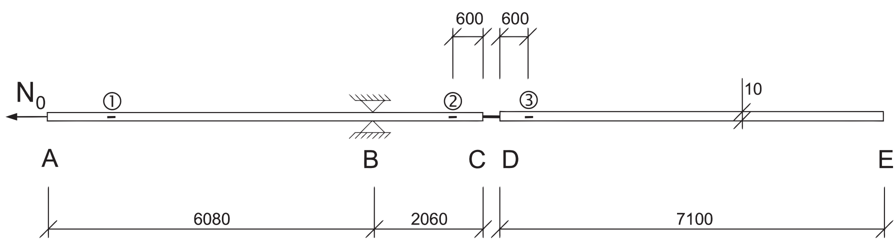


Figure 3.2: Principle overview of the split-Hopkinson tension bar test rig used in the experiments (modified figure from [17])

As the tensile stress wave propagates towards position C and the specimen (ref. Fig. 3.2), the strain in the incident bar will be measured by strain gauge 2 and is referred to as the incident strain, ε_I. When the stress wave reaches position C, it will partly be reflected back, and partly transmitted to the specimen. The reflected strain measured by strain gauge 2 is denoted the reflected strain, ε_R, while the transmitted strain measured by strain gauge 3 is denoted the transmitted strain, ε_T. Referring to Fig. 3.2, there will be no change in signal between strain gauge 2 and point C, and between point D and strain gauge 3, except for the time lag, as long as there are no dispersion present. Clausen and Auestad [17] have reported that there seems to be no dispersion present between strain gauge 2 and point C, and between point D and strain gauge 3, for the particular SHTB test setup used for the experimental work for this thesis, such that this possible source of error will not be devoted further investigation.

For the relations stated below, subscript s denotes specimen, subscript 0 refers to the bars, A is cross sectional area, E is Young's modulus, L_s is the length of the gauge section of the specimen, c_0 is the wave propagation velocity and t is time. The stress in the specimen is found by dividing the force at point D (or at point C) by the cross sectional area of the specimen:

$$\sigma_s = \frac{F_s}{A_s} = \frac{E_0 A_0}{A_s}\varepsilon_T \tag{3.2}$$

By assuming that all strains in the specimen take place in the gauge section (discussed in Section 4.1), the strain in the specimen is calculated as:

$$\varepsilon_s = \frac{c_0}{L_s}\int_0^t (-\varepsilon_T - (-\varepsilon_I + \varepsilon_R))d\tau = -2\frac{c_0}{L_s}\int_0^t \varepsilon_R d\tau \tag{3.3}$$

The corresponding strain rate is simply the time derivative of the strain state:

$$\dot{\varepsilon_s} = \frac{\partial \varepsilon_s}{\partial \tau} = -2\frac{c_0}{L_s}\varepsilon_R \tag{3.4}$$

3.3 Experimental Tests (SHTB)

The SHTB test rig used for the experimental tests is situated in the laboratories of the Department of Structural Engineering at the Norwegian University of Science and Technology (NTNU). Referring to Fig. 3.2, the test rig consists of an incident bar (A-C), transmission bar (D-E), a test specimen (C-D), a friction lock mechanism at position B and a loading mechanism at position A. The geometry of the test specimens is seen in Fig. 3.3. The bars are of steel quality Tibnor $52SiCrNi5$ with an approximate yield stress of $900MPa$, Young's modulus of $210GPa$ at room temperature and have circular cross sections with diameter of $10mm$, thus a cross sectional area of $78.5mm^2$ [17]. To ensure elastic behavior of the two bars, the tension force N_0 should not exceed $70kN$. The incident and transmission bar have respectively a length of $8140mm$ and $7100mm$.

The test rig is mounted to a rigid steel frame supported by the floor and with bearings between the bars and frame made of $60mm$ PVC tubes. The tubes are mounted 625mm apart from each other and ensure minimal friction and large electrical resistivity to not affect the experimental test results. The PVC tubes are the only physical connection between the bars and steel frame except from position A and B. The experimental test can be studied in two phases:

First phase: right hand part of incident bar is restrained by a friction clamp, while left hand part is stretched in tension.

Second phase: sudden release of the friction clamp such that a tensile stress wave will propagate towards the test specimen.

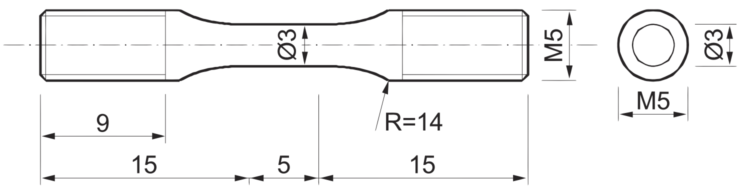
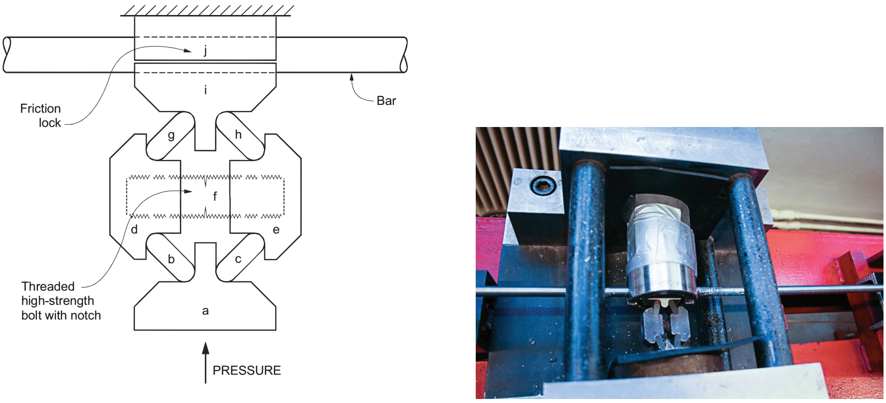


Figure 3.3: Geometry of test specimens [17]

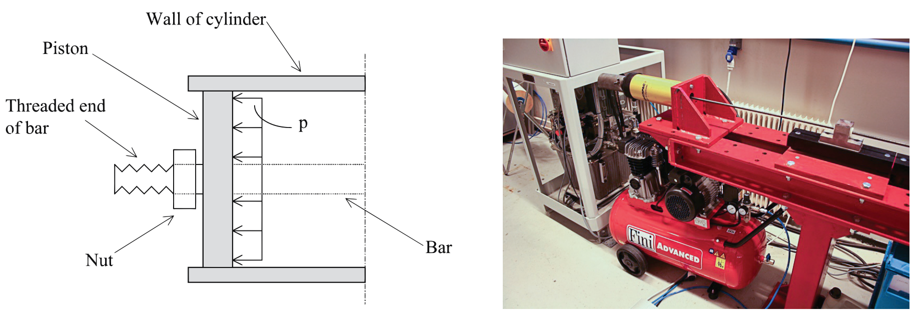
Part B-C of the incident bar is restrained against longitudinal movement by a friction lock mechanism in the first phase. As seen in Fig. 3.4a, the friction clamp (i)-(j) is held together by applying pressure to component (a) by use of a hydraulic jack. It can be seen that the force from the hydraulic jack is transferred to the clamp via components (b) to (h) as long as the bolt (f) is intact. The notched bolt is made of cold working tool steel with an approximate hardness of HRC 50 and with diameter of 12mm in the threaded part and 6mm in the notched part. The abrupt fracture of the bolt causes the friction forces in the clamp to be suddenly removed, such that a tensile stress wave will be propagating from A-B towards the test specimen. To ensure both a short rise time of the stress wave and controlled conditions during the experiment, the brittleness of the bolt is of crucial importance.



(a) (b)

Figure 3.4: (a) Principal overview of friction lock mechanism [17] and (b) picture of friction lock mechanism

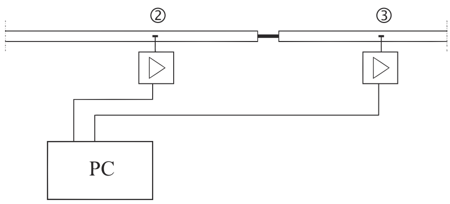
Part A-B of the incident bar is stretched in tension by use of a hydraulic jack at position A. The force from the hydraulic cylinder is transferred to the bar through a nut, and the applied force is monitored by strain gauge 1 (ref. Fig. 3.2). When the desired tension force is reached, the force in the friction lock is increased until the abrupt fracture of the bolt in the friction lock mechanism. Fig. 3.5 shows the loading mechanism.



(a) (b)

Figure 3.5: (a) Principal overview of loading mechanism [17] and (b) picture of loading mechanism

After the abrupt fracture of the notched bolt and the release of the propagating stress wave, the second phase of the test takes very short time. The stress wave will be propagating towards the test specimen with velocity $c_0 \approx 5100\frac{m}{s}$ and total duration for the entire stress wave to pass a point on the bar is $2.3\frac{m}{s}$, and by then the specimen will be stressed in tension and usually ruptured [17]. To ensure an adequate number of data points, the minimum logging frequency required is $1MHz$ [17]. Strain gauges are glued on the bars at three separate locations, and also on either side of the bars to cancel out any undesirable bending effects. As previously mentioned, only strain gauge 2 and 3 are used for measurements of the response of the test specimen. Due to weak signals from the strain gauges and presence of several sources of noise, an amplifier unit is connected to each pair of strain gauges, resulting in a total amplification of the output signal of 125 [14]. The amplifiers are connected to a PC that log the signals, see illustration in Fig. 3.6a.



(a)

(b)

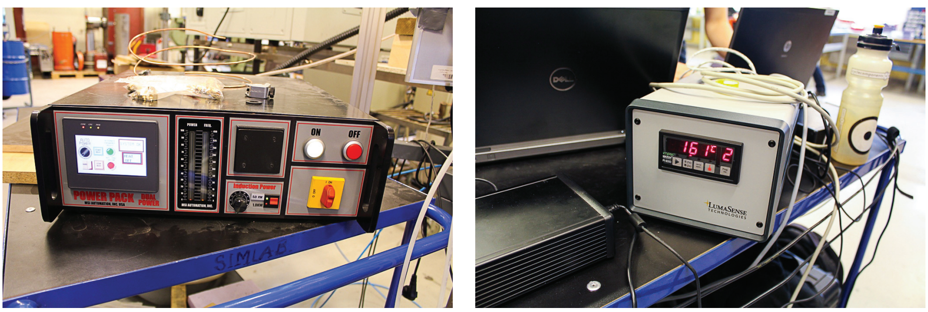
Figure 3.6: (a) Principal overview of data registration system [17] and (b) picture of test rig

Experimental tests have been conducted for a wide range of temperatures, ranging from room temperature ($293K$) to $633K$. The temperature rise of the test specimen is done by applying a water-cooled induction heater system coupled with a coil. Induction heating refers to the process where an electrically conducting object, e.g. a metal object, is heated electrically by electromagnetic induction. A high-frequency alternating current (AC) is passed through the induction heater, that consists of an electromagnet, to generate eddy currants (Foucault currents) within the object that generates heating due to its thermo-resistivity. The induction heater system used for the experiments can deliver a power of $5kW$ at $180kHz$, such that heating rates up to approximately $10Ks^{-1}$ can be reached on average. The induction heater system can be seen in Fig. 3.8a and the coil is seen in Fig. 3.7b. It should be noted that the power delivered by the heating system may lead to visible noise in the strain gauge measurements and may be overcome by using a low-pass filter.

(a) (b)

Figure 3.7: (a) Overview of test setup where two light sources, the pyrometer and the high-speed camera can be seen and (b) test specimen mounted in test rig together with coil and laser beam

A pyrometer (IP 140 MB12, Lumasense Technologies) was used to monitor the temperature of the test specimen, which is a device that is able to intercept and measure the thermal radiation (pyrometry) from an object by non-contact. The surface temperature of the object can be determined from the thermal radiation that is registered and the emissivity of the object. The test specimens were painted black using a thermal-resistant paint applicable for temperatures ranging up to $1123K$ and giving a constant emissivity of about 0.95, see Fig. 3.9. The pyrometer used for the experiments applies a laser beam onto the specimen to measure the thermal radiation, and has a sampling frequency of $666Hz$ that ensures adequate thermal control when heating the specimen to desired temperature. However, the sampling frequency is too low to measure the adiabatic heating of the specimen during high strain rate tests. The temperature measured was monitored from a temperature monitoring device coupled with the pyrometer and can be seen in Fig. 3.7a and 3.8b.



(a) (b)

Figure 3.8: (a) Power supply for induction heater system and (b) the temperature monitoring device

As discussed in Section 2.1.3, the true stress-plastic strain curve determined from global measurements, i.e. from strain gauge measurements in the SHTB experiments, is only valid until the onset of necking. A localized and complex tri-axial stress state will arise in the necked region, such that the stress and strain fields are no longer uniform within the gauge section and the strain rate will be increased inside the neck. Metals, e.g. aluminum, become relative more ductile and soft at elevated temperatures, such that the onset of necking will occur after very little plastic straining. This implies that the valid experimental data from conventional strain gauge measurements will be of very limited range. However, this problem can be overcome by obtaining information of the local deformation state at the necked section until fracture. Several methods have been applied and tested for previous SHTB experiments by other authors, such as laser-based techniques, frame-based techniques for detecting the edges of the sample and stereo digital image correlation (SDIC) [37].

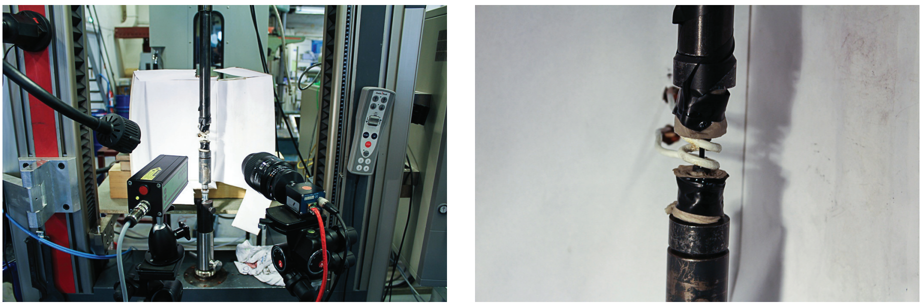
(a) (b)

Figure 3.9: Test specimens (a) before and (b) after painting

A solution using one high-speed digital camera to record images of the necked section was chosen, such that the edges of the specimen during the entire test are recorded, and, thus, the geometry of the neck can be calculated. The post-processing of the recorded images to obtain the local deformation state at the neck is further discussed in Section 4.4. The camera setup consists of a digital high-speed Photron SA1.1 camera equipped with a Sigma 105mm macro lens. The acquisition frequency used was between $50000 Hz$ and $90000 Hz$ depending on the desired strain rate of the tests. As only one camera was used, the cross sectional area of the specimen is assumed to remain circular during the whole deformation period, hence the material is assumed to be isotropic with respect to plastic flow. Two light sources and a white paper box were used together to minimize shadows and increase the contrast between the test specimen and the background for easier detection of the edges of the specimen. The recording of the images by the camera was triggered by the incoming stress wave at strain gauge 2, ref. Fig. 3.2.

3.4 Experimental Tests (Quasi-Static)

Experimental quasi-static tests have been conducted in the laboratories of the Department of Structural Engineering at the Norwegian University of Science and Technology (NTNU) for strain rates of $0.01s^{-1}$ and $1s^{-1}$ and for temperatures ranging between 293 and $633K$. The tests have been performed in a Zwick/Roell Z030 test rig that was calibrated the day prior to the experimental tests, such that the experimental results provided by the rig should be adequately accurate. It can be seen from Fig. 3.10a that the same pyrometer was used as for the SHTB tests. The same induction heater system and coil was also used, but there was no need for a high-speed camera during these tests due to the quasi-static strain rates. Instead, a non high-speed digital camera was used to record images of the necked section during the test. The camera used, together with a light source and a white paper box to minimize shadows and increase the contrast between the test specimen and the background, and the pyrometer, can be seen in Fig. 3.10a, while a close up picture of the specimen mounted in the test rig together with the coil can be seen in Fig. 3.10b.



(a) (b)

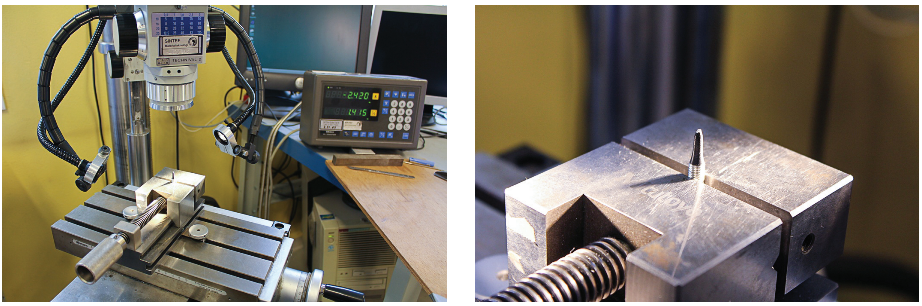
Figure 3.10: (a) Overview of test setup for quasi-static tests and (b) test specimen mounted in test rig together with the coil

Two registration systems were present during the experiments. Fig. 3.11a shows the registration system for the camera recordings, where the live feed from the camera can be seen on the left laptop, while Fig. 3.11b shows the registration system for recording of the force and displacement provided by the test rig itself. The camera recordings were initiated manually just prior to the start of testing as the acquisition frequency is very much relatively lower compared to the camera recordings from the SHTB experiments. Fig. 3.8 shows the induction heater system and the temperature monitoring device connected to the pyrometer.

(a) (b)

Figure 3.11: Overview of data registration system (a) for camera recordings and (b) from test rig

As discussed in Section 2.4.7, the fracture strain can be estimated from the fracture area using Eq. (2.43). The fracture area was measured for all specimens post-fracture from the quasi-static tests using a Carl Zeiss Jena Technival 2 microscope. Depending on the strain rate and temperature reached, different post-fracture shapes were observed from the tests (see Appendix G for a full overview). The accuracy of the area measurements must be addressed carefully as the shallow depth of field provided by the microscope resulted in difficulties determining the edges of the fracture area for several specimens.



(a) (b)

Figure 3.12: (a) Microscope equipment for measuring post-fracture area of test specimens and (b) test specimen mounted in microscope setup

4 Experimental Results

4.1 Correction of Young's Modulus and Strains

An important assumption taken into account in the derivation of Eq. (3.3) is that the entire straining of the specimen is located in the gauge section. All plastic strains are in fact believed to be localized in the gauge section, but during the elastic deformation there will most likely be some elastic straining in the transitional regions of the specimen where the diameter is reduced. As a result of this, Young's modulus calculated from the experimental data will be too low, see Fig. 4.1. Albertini and Montagnani [5] states that this error is due to the deformation taking place outside the gauge section of the specimen and have suggested a formula to correct the strain values:

$$\varepsilon_e = \varepsilon_m - \sigma_e \cdot \frac{E - E_m}{E \cdot E_m} \tag{4.1}$$

where ε_e is the corrected engineering strain, ε_m is the measured engineering strain, σ_e is the corresponding engineering stress, E_m is the measured Young's modulus and E is the correct Young's modulus. Young's modulus for aluminum is found to not vary from quasi-static to high strain rate experiments, but being dependent on temperature [23, 32], such that the correct value can be found in the literature and adjusted for the temperature using Eq. (2.10). Clausen and Auestad [17] have carried out experiments with specimens of different geometry equipped with strain gauges to investigate the correction method based on Eq. (4.1), and their results indicate that the correction formula provides valid strain values.

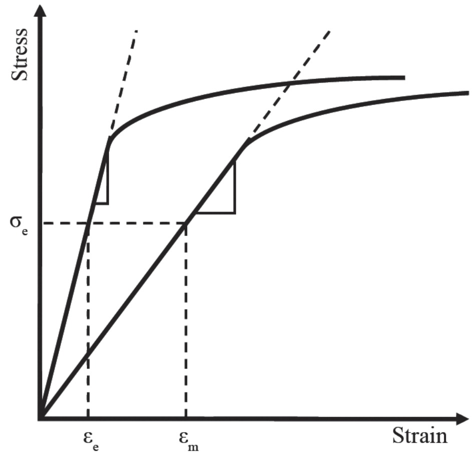


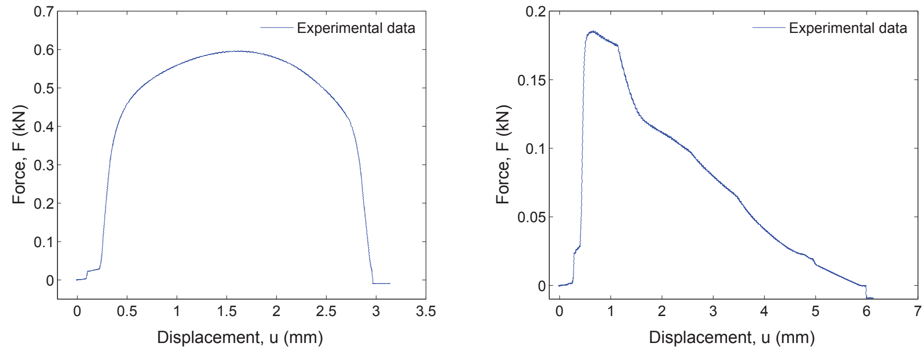
Figure 4.1: Correction of Young's modulus and strains

4.2 Post-Processing of Data from Quasi-Static Experiments

Quite some effort has been devoted to create Matlab scripts for efficient, accurate and user-friendly post-processing of experimental data from the quasi-static tests. This Section aims to describe how the data is processed and to discuss difficulties and uncertainties that have arisen for the post-processing of the experimental data.

The complete Matlab code for the post-processing can be found in Appendix D.1 and can readily be modified to any experimental data from quasi-static tests; the only input from files necessary are values for force and displacement. As a total of 47 quasi-static tests carried out needed to be post-processed (see Section 4.3.1), the scripts have been designed with emphasize on a minimum of necessary manual interference by the user. No local measurements from the camera recordings have been provided for the quasi-static experiments during the work for this thesis, thus only experimental data from the test rig have been post-processed.

Fig. 4.2a and b show two examples of (raw) experimental data in terms of force and displacement values from the quasi-static tests carried out, respectively at $523K$ and $633K$ and strain rate of $0.01s^{-1}$ for both tests. The two figures showcase two examples of typical behavior that is present; some initial stabilization problems and difficulties in terms of keeping a constant temperature throughout the whole test for $0.01s^{-1}$ strain rate tests. The initial stabilization problems can be seen as the force applied from the test rig has typically two constant plateaus in the beginning of the test and can be seen from both Fig. 4.2a and b, and this behavior is related to rigid body movements of the specimen in the test rig. As long as this behavior is present only in the beginning of a test, such that the measured Young's modulus can be calculated from the elastic region, it will not have any affection on the resulting true stress-plastic strain curve.



(a) (b)

Figure 4.2: Two examples of force vs. displacement plots from quasi-static tests for (a) $523K$ and $0.01s^{-1}$ and (b) $633K$ and $0.01s^{-1}$

The other typical behavior present is that there are more difficulties associated with keeping the test specimen at a constant temperature for higher temperatures, and this is noticeably more difficult for temperatures higher than $573K$. The fluctuating temperature can result in a force-displacement curve as illustrated in Fig. 4.2b. However, at such high temperatures, the specimen will start to neck after very little plastic straining, such that the true stress-plastic strain values until onset of necking will most likely be hardly influenced by temperature fluctuations. On the other hand, when applying true strain values from camera measurements to calculate the true stress-plastic strain beyond necking, these values will be directly influenced by the temperature fluctuations, thus the validity and applicability of these results must be carefully assessed. The temperature fluctuations are in fact only noticeable for tests at $0.01s^{-1}$ strain rate for the tests carried out due to the relatively longer total time period of testing.

As discussed in Section 4.1, it is necessary to correct the engineering strain values due to the initial straining located at the shoulders of the test specimen. In order to do this, both the measured Young's modulus and the correct Young's modulus are needed. The correct Young's modulus is a function of temperature and is calculated from Eq. (2.10), while the measured Young's modulus needs to be calculated from the uncorrected engineering stress-strain curve. The script lets the user choose a representative data interval in the elastic region by point-and-click and automatically calculates the measured Young's modulus, translates the stress and strain values to start from origo, and corrects the strain values using Eq. (4.1). As can be seen in Fig. 4.3, the script plots both the uncorrected strains, corrected strains and the measured Young's modulus for stress and strain values translated to start from origo. From this figure, the yield stress can be determined from the intersection point between the uncorrected engineering stress-strain curve and the measured Young's modulus curve. It should be noted that the strain range shown in Fig. 4.3 is not the total range for the particular test, but a zoomed in area determined by the user. As some tests show fluctuating values for both force and displacement, the engineering stress and strain have been smoothed using a moving average including 2 points on each side of the considered stress and strain point. Using this number of points proved to be adequate for easier determination of both the Young's modulus and yield stress.

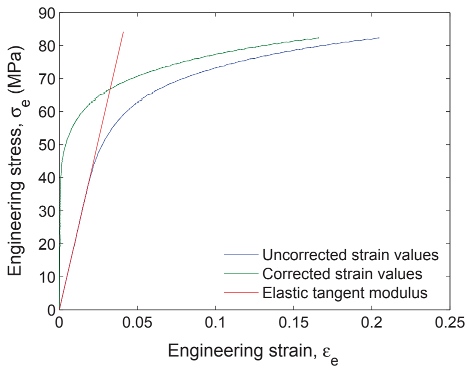
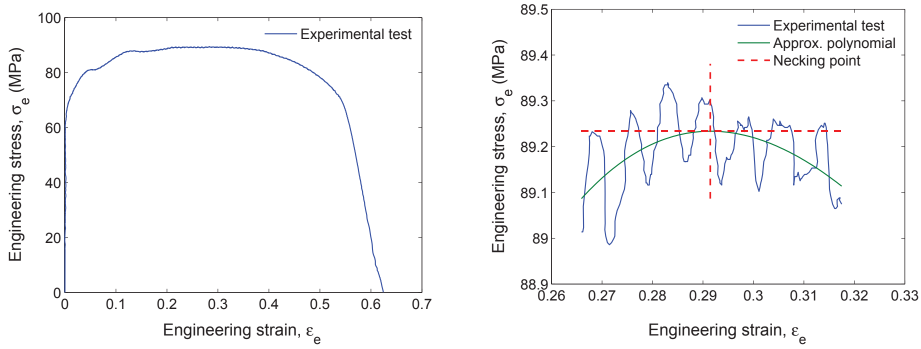


Figure 4.3: Plot showing the measured Young's modulus together with both uncorrected and corrected strains

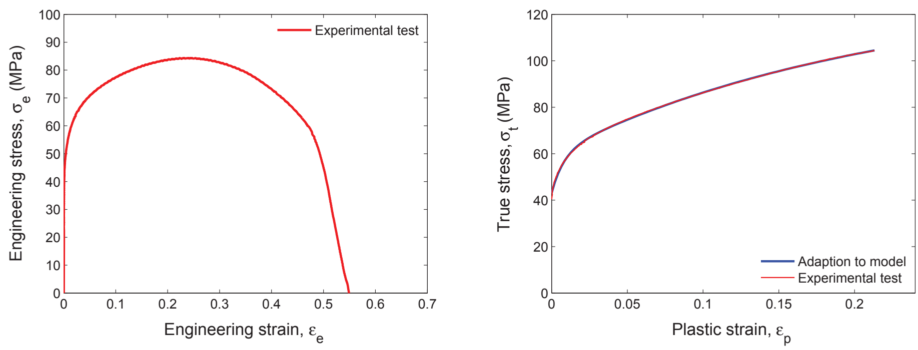
As discussed in Section 2.1.3, only the true stress-strain values until onset of necking are in fact representative for the stress and strain state within the gauge length considered from measurements from the test rig. As a result of this, the strain value for onset of necking has to be calculated. Fig. 4.4a shows an example of an engineering stress-strain curve where the strain value for onset of necking cannot be easily determined. To calculate the necking point, the script lets the user choose a strain range (the range must contain the strain value for onset of necking), and an approximated second degree polynomial for the stress is calculated for the defined range. From this approximation, the necking point is found from the strain value corresponding to the maximum approximated stress value. Fig. 4.4b illustrates the engineering stress-strain values in the determined range, the approximated polynomial and the resulting calculated necking point.



(a) (b)

Figure 4.4: (a) Engineering stress-strain curve from a test where the necking point is difficult to locate and (b) determination of necking point using an approximated polynomial

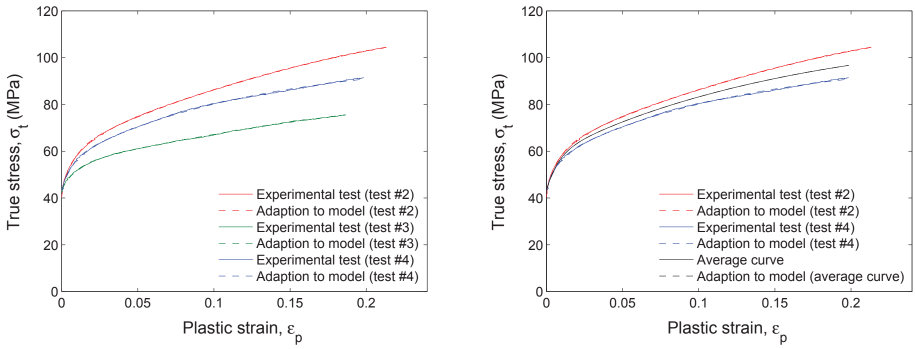
Plots of the resulting engineering stress-strain and true stress-plastic strain curves, as shown in Fig. 4.5a and b, are automatically created and saved. The true stress-plastic strain curves are fitted with Voce hardening parameters which takes on the form $\sigma_t = \sigma_Y + \sum_{i=1}^{2} Q_i(1 - e^{-C_i \varepsilon_p})$. The resulting plots, yield stress and Voce hardening parameters for all tests can be found in Appendix E.



(a) (b)

Figure 4.5: (a) Engineering stress-strain plot and (b) true stress-plastic strain plot from post-processing

Two, three or four tests have been carried out with the same boundary conditions (temperature and strain rate) for the major part of the tests. This is done in order to examine the validity of the results, as the test specimens are easily twisted when mounted in the test rig. For the case where two tests of same boundary conditions are believed to be valid and the specimen is assumed to not be twisted, the average curve for the two tests are calculated and fitted with Voce hardening parameters. For the case with three tests with same boundary conditions, all three resulting true stress-plastic strain curves are plotted together, see Fig. 4.6a, and an average curve is calculated from the two most coinciding curves, as seen in Fig. 4.6b.



(a) (b)

Figure 4.6: (a) Comparison of tests at same temperature and strain rate and (b) resulting two closest curves together with the average curve

4.3 Results from Quasi-Static Experiments

In this Section, results from the quasi-static experiments are presented. An overview of the experimental program is given in Section 4.3.1, while results for the strain hardening, yield stress and estimated fracture strain are presented in respectively Section 4.3.2, 4.3.3 and 4.3.4.

4.3.1 Experimental Program

An overview of the experimental program for the quasi-static tests is presented in Table 4.1. Tests have been performed for $0.01s^{-1}$ and $1s^{-1}$ strain rate and for temperatures ranging from $293K$ (room temperature) to $633K$. Some specimens are believed to be damaged while mounting into the test rig, and these tests are therefore not shown in the results presented.

AA6060-L				AA6060-H			
Test #	Strain rate (s^{-1})	Temp. ($^{\circ}K$)	Comment	Test #	Strain rate (s^{-1})	Temp. ($^{\circ}K$)	Comment
1	0.01	293	OK	1	0.01	523	OK
2	0.01	473	D.s.*	2	0.01	523	OK
3	0.01	473	OK	3	0.01	523	OK
4	0.01	473	OK	4	0.01	523	OK
5	0.01	523	D.s.*	5	0.01	573	OK
6	0.01	523	OK	6	0.01	573	OK
7	0.01	523	OK	7	0.01	293	OK
8	0.01	573	OK	8	0.01	473	OK
9	0.01	573	OK	9	0.01	473	OK
10	0.01	633	OK	10	0.01	633	OK
11	0.01	633	OK	11	0.01	633	OK
12	1	293	OK	12	1	293	OK
13	1	473	OK	13	1	473	OK
14	1	473	OK	14	1	473	OK
15	1	523	OK	15	1	473	D.s.*
16	1	523	OK	16	1	523	OK
17	1	573	OK	17	1	523	OK
18	1	573	OK	18	1	573	OK
19	1	633	OK	19	1	573	OK
				20	1	633	OK
				21	1	633	OK

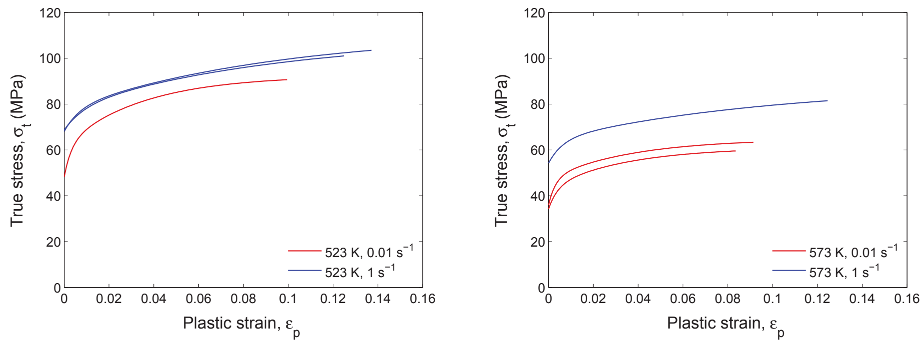
* D.s. = Damaged specimen (usually from mounting in the test rig)

AA6060-OLD			
Test #	Strain rate (s^{-1})	Temp. ($^{\circ}K$)	Comment
1	1	523	OK
2	1	523	OK
3	1	573	OK
4	0.01	523	OK
5	0.01	523	Aborted
6	0.01	573	OK
7	0.01	573	OK

Table 4.1: Overview of experimental program for quasi-static tests

4.3.2 Strain Hardening

Fig. 4.7 illustrates the results for the AA6060-OLD alloy for same temperature but various strain rates in the same plot. For tests at $573K$ it is difficult to detect any dependence on the strain rate for the strain hardening, while at $523K$ the test at $0.01s^{-1}$ hardens more than the test at $1s^{-1}$. It is also seen that the yield stress is noticeably lower for tests at $0.01s^{-1}$ compared to $1s^{-1}$ at both temperatures.

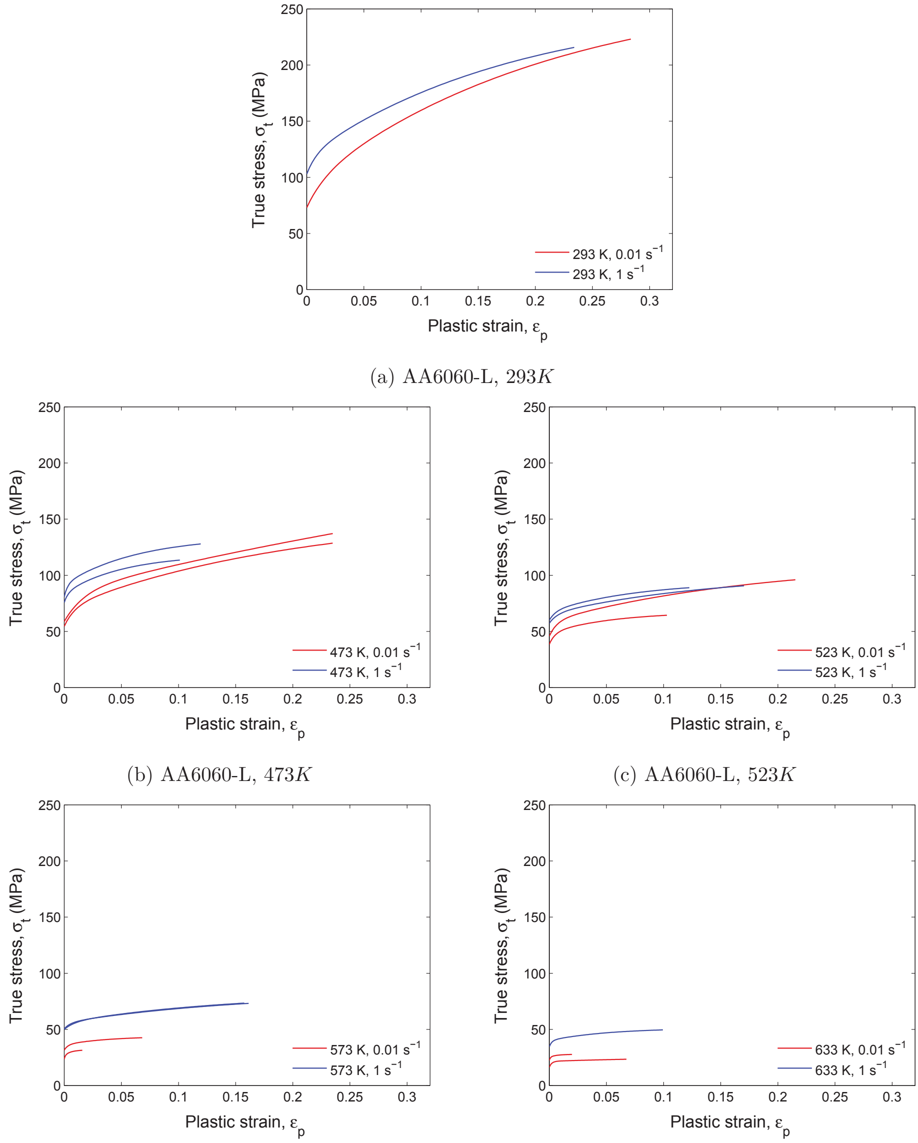


(a) AA6060-OLD, $523K$

(b) AA6060-OLD, $573K$

Figure 4.7: Plots (a)-(b) show the strain hardening for same temperature but various strain rates for AA6060-OLD

Fig. 4.8 illustrates the results for the AA6060-L alloy for same temperature but various strain rates in the same plot. It is seen that the strain hardening is dependent on the strain rate, but it seems to show somewhat different dependency for different temperatures. For temperatures between $293K$ and $523K$ the material strain hardens more for $0.01s^{-1}$ strain rate, while the opposite behavior is seen for temperatures between $573K$ and $633K$. Tests at same temperature and strain rate show to some extent different behavior, especially at $473K$ and $523K$, thus no specific correlation between strain hardening and strain rate can be established. It is also seen that the yield stress is noticeably lower for tests at $0.01s^{-1}$ compared to $1s^{-1}$ for all temperatures.



(a) AA6060-L, $293K$

(b) AA6060-L, $473K$

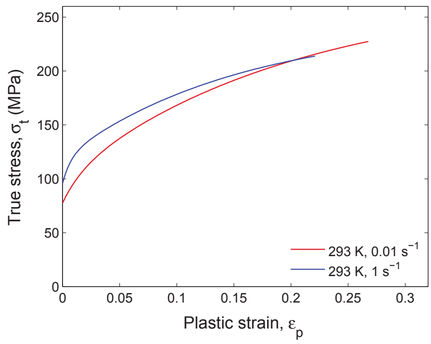
(c) AA6060-L, $523K$

(d) AA6060-L, $573K$

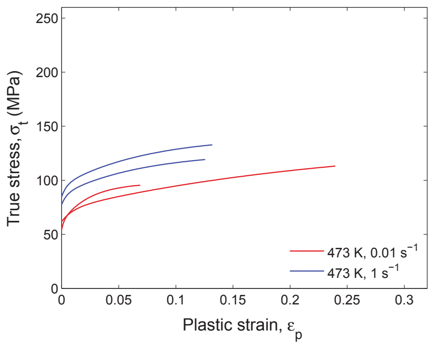
(e) AA6060-L, $633K$

Figure 4.8: Plots (a)-(e) show the strain hardening for same temperature but various strain rates for AA6060-L

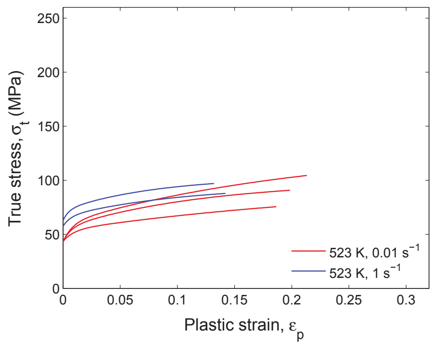
Fig. 4.9 illustrates the results for the AA6060-H alloy for same temperature but various strain rates in the same plot. As for the AA6060-L alloy, it is seen that the strain hardening is dependent on the strain rate, and it also seems to show somewhat different dependency for different temperatures. For temperature of $293K$ the material seems to strain harden more at $0.01s^{-1}$, while being essential non-dependent on the strain rate for temperatures between $473K$ and $573K$. At $633K$ the material shows noticeably more strain hardening at $1s^{-1}$. It is also seen that the yield stress is noticeably lower for tests at $0.01s^{-1}$ compared to $1s^{-1}$ for all temperatures.



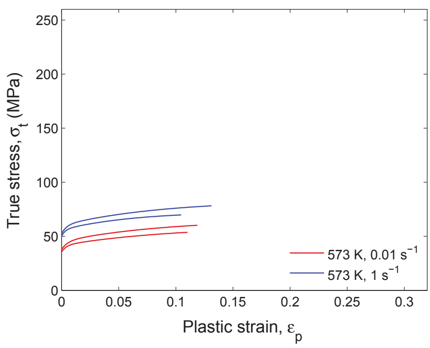
(a) AA6060-H, $293K$



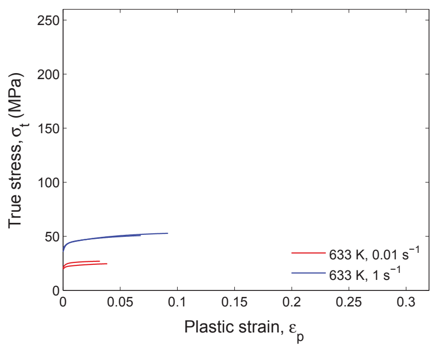
(b) AA6060-H, $473K$



(c) AA6060-H, $523K$



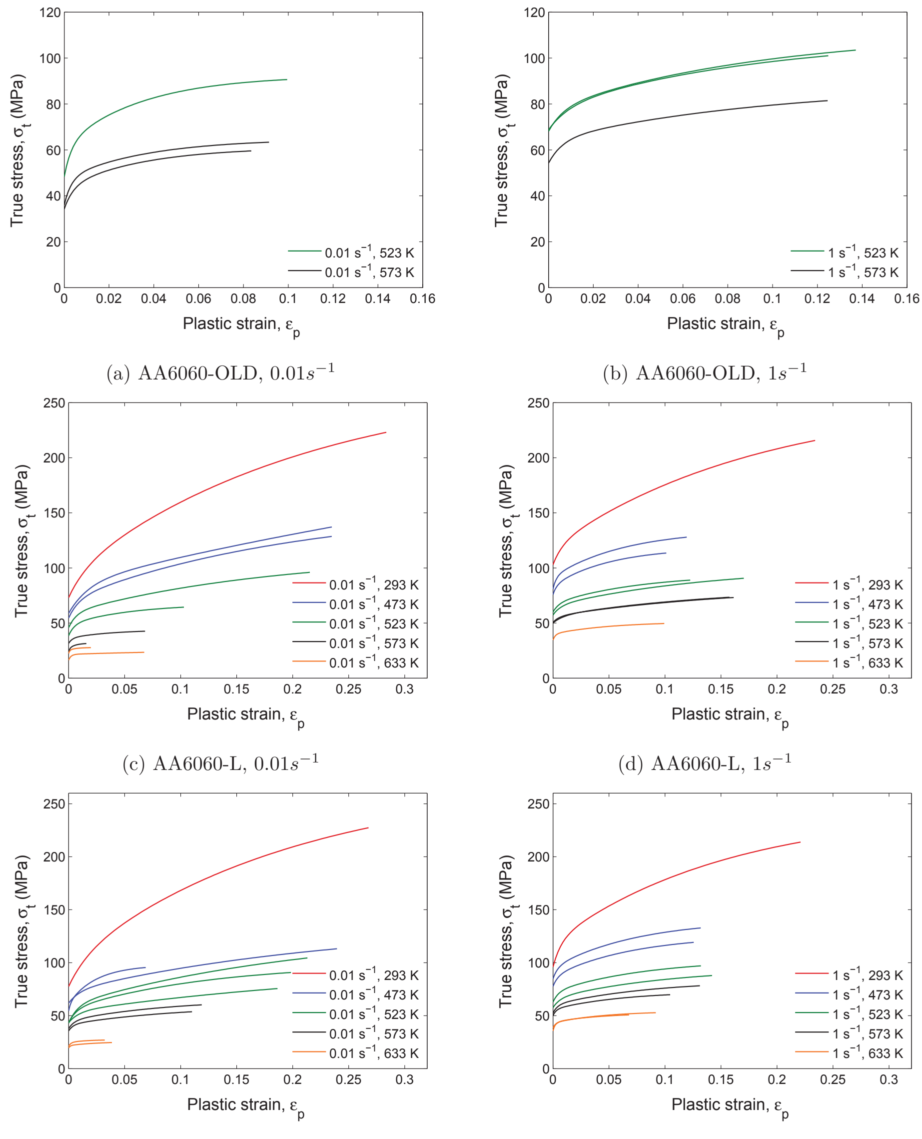
(d) AA6060-H, $573K$



(e) AA6060-H, $633K$

Figure 4.9: Plots (a)-(e) show the strain hardening for same temperature but various strain rates for AA6060-H

Fig. 4.10 illustrates the results for (a)-(b) the AA6060-OLD alloy, (c)-(d) the AA6060-L alloy and (e)-(f) the AA6060-H alloy for same strain rate but various temperatures in the same plot. It is clearly seen that the strain hardening shows strong dependency on the temperature for all materials, i.e. the materials show less strain hardening when the temperature is increased. It is not possible to detect any difference for the temperature dependency for the two strain rates.



(a) AA6060-OLD, $0.01s^{-1}$

(b) AA6060-OLD, $1s^{-1}$

(c) AA6060-L, $0.01s^{-1}$

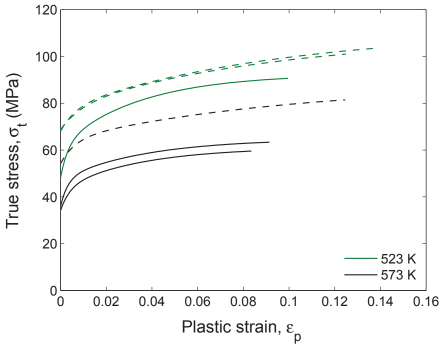
(d) AA6060-L, $1s^{-1}$

(e) AA6060-H, $0.01s^{-1}$

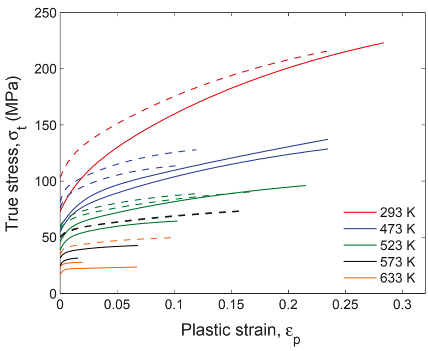
(f) AA6060-H, $1s^{-1}$

Figure 4.10: Plots show the strain hardening for same strain rate but various temperatures for respectively (a)-(b) AA6060-OLD, (c)-(d) AA6060-L and (e)-(f) AA6060-H

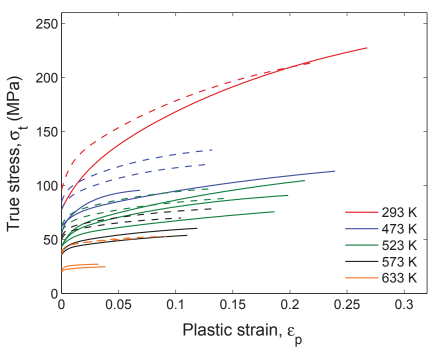
Fig. 4.11 illustrates the results for (a) the AA6060-OLD alloy, (b) the AA6060-L alloy and (c) the AA6060-H alloy for all tests in the same plot and sums up what can be seen in Fig. 4.7 to 4.10. Solid lines are $0.01s^{-1}$ strain rate and dashed lines are $1s^{-1}$ strain rate.



(a) AA6060-OLD



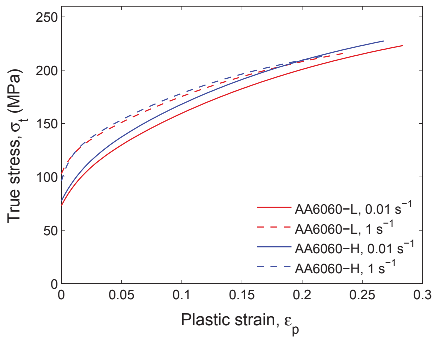
(b) AA6060-L



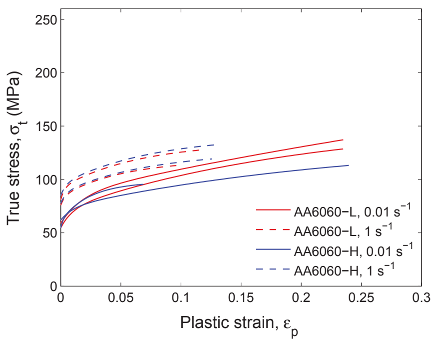
(c) AA6060-H

Figure 4.11: Plots show the strain hardening for all temperatures and strain rates for respectively (a) AA6060-OLD, (b) AA6060-L and (c) AA6060-H. Solid lines are $0.01s^{-1}$ strain rate and dashed lines are $1s^{-1}$ strain rate

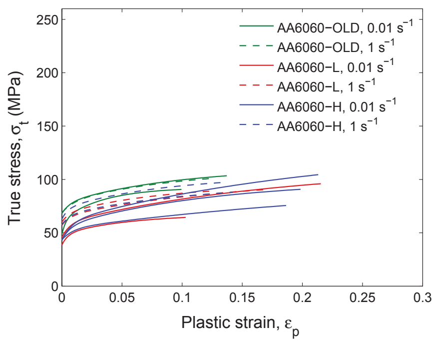
Fig. 4.12 illustrates the results for all alloys and for same temperature plotted together. Different behavior is observed with respect to the temperature dependency for the strain hardening for all materials and is most noticeable for $0.01s^{-1}$ strain rate. Except for tests at $473K$, the AA6060-H alloy seem to strain harden more than the AA6060-L alloy, but the difference is in fact not significant. This is as expected as the alloy contains more alloying elements than the AA6060-L alloy.



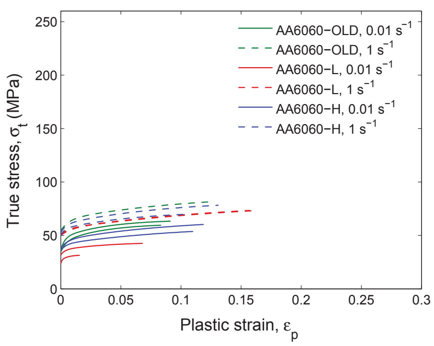
(a) $293K$



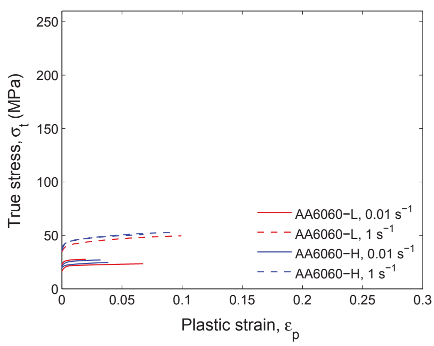
(b) $473K$



(c) $523K$



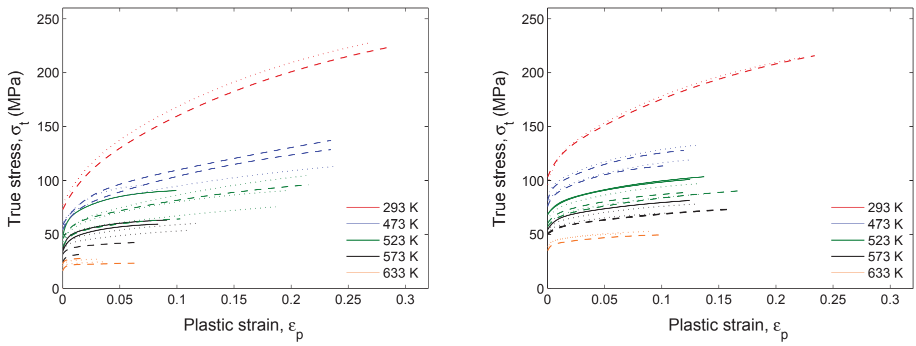
(d) $573K$



(e) $633K$

Figure 4.12: Plots (a)-(e) show the strain hardening for same temperature but various strain rates for AA6060-OLD, AA6060-L and AA6060-H

Fig. 4.13 illustrates the results for all alloys and for same strain rate plotted together. Solid lines are the AA6060-OLD alloy, dashed lines are the AA6060-L alloy and dotted lines are the AA6060-H alloy. Almost coinciding temperature dependence for the strain hardening is observed for all materials and for both strain rates. The AA6060-H alloy seems to strain harden slightly more than the AA6060-L alloy for all temperatures. It is also interesting to notice that the AA6060-OLD alloy seems to strain harden more than the two other.



(a) $0.01s^{-1}$

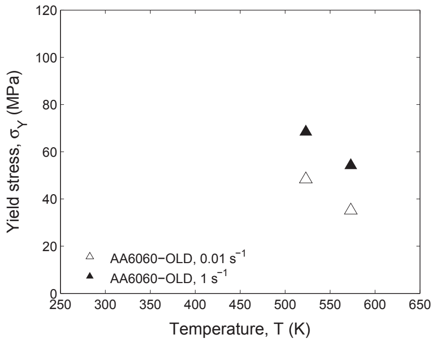
(b) $1s^{-1}$

Figure 4.13: Plots (a)-(b) show the strain hardening for same strain rate but various temperatures for AA6060-OLD (solid line), AA6060-L (dashed line), AA6060-H (dotted line)

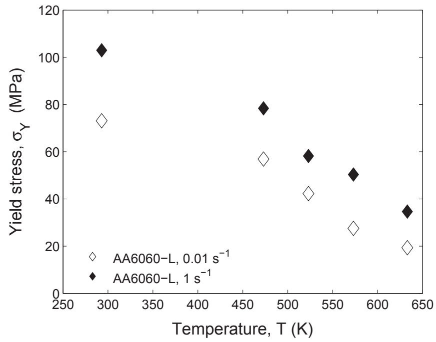
4.3.3 Yield Stress

The average yield stress has been plotted for tests with same strain rate and temperature for the plots presented in this Section.

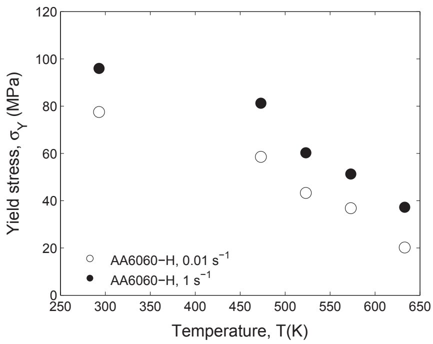
Fig. 4.14 illustrates the yield stress vs. temperature for all materials. It is seen that the yield stress is clearly dependent on both strain rate and temperature. The strain rate sensitivity seems to be constant within the whole temperature range considered, while the temperature sensitivity varies within the temperature range. An inverted s-shape is seen, and it looks like the increased temperature dependency is most present between temperatures of $450K$ and $550K$. The AA6060-H alloy seems to have a slightly higher yield stress than the AA6060-L alloy for both strain rates and at all temperatures. Notice also that the AA6060-OLD alloy seems to have a higher yield stress than the other two alloys.



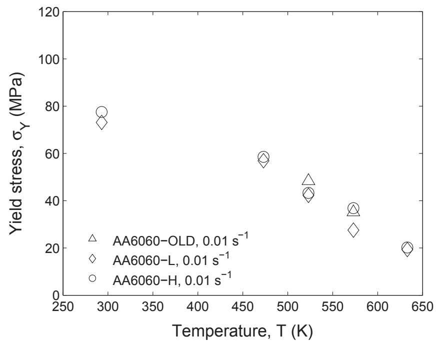
(a) AA6060-OLD



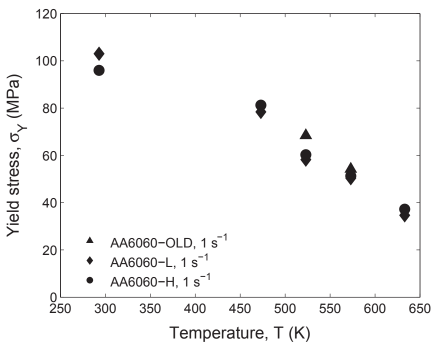
(b) AA6060-L



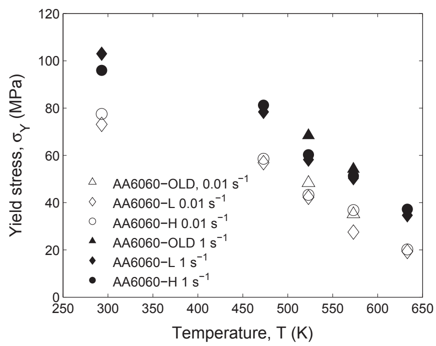
(c) AA6060-H



(d) $0.01s^{-1}$



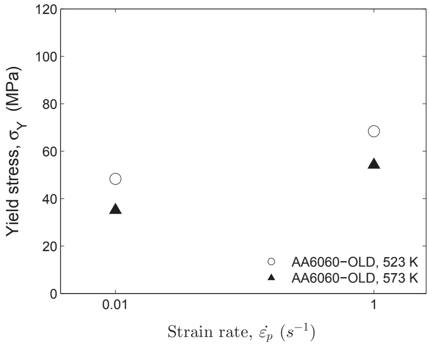
(e) $1s^{-1}$



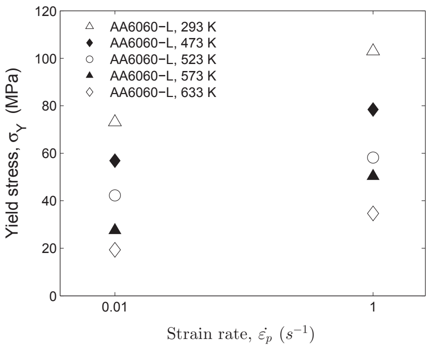
(f)

Figure 4.14: Plots (a)-(c) show the yield stress vs. temperature for various strain rates for respectively (a) AA6060-OLD, (b) AA6060-L and (c) AA6060-H. Plots (d)-(e) show the yield stress vs. temperature for (d) $0.01s^{-1}$ and (e) $1s^{-1}$ strain rate. Plot (f) shows plot (d) and (e) together.

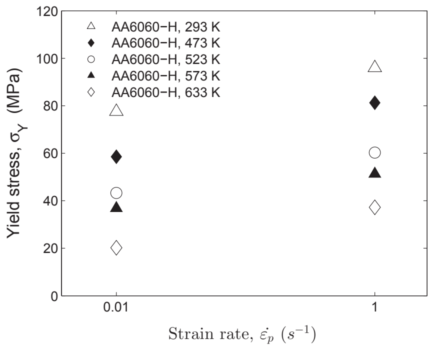
Fig. 4.15 illustrates the yield stress vs. strain rate for all materials. The strain rate sensitivity seems to be constant within the temperature range considered for all alloys.



(a) AA6060-OLD



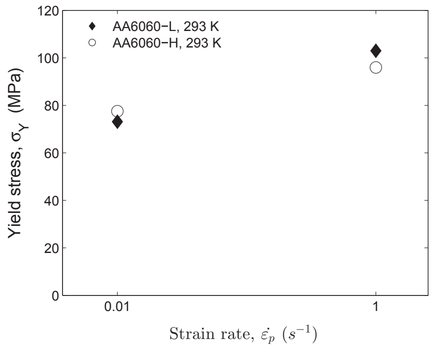
(b) AA6060-L



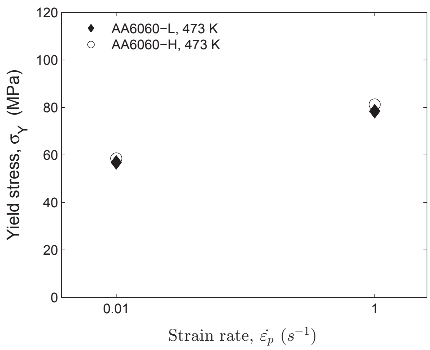
(c) AA6060-H

Figure 4.15: Plots (a)-(c) show the yield stress vs. strain rate for all temperatures for respectively (a) AA6060-OLD, (b) AA6060-L and (c) AA6060-H

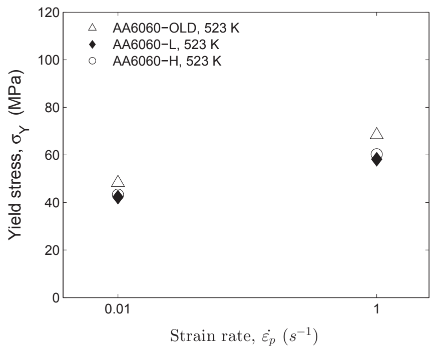
Fig. 4.16 illustrates the yield stress vs. strain rate for all alloys and same temperature in the same plot. No significant difference between the materials is seen with respect to the yield stress dependence on the strain rate. However, at $293K$ the AA6060-L and AA6060-H alloy seem to show slightly different strain rate dependency, but as this is not seen for the other temperatures, this may in fact not represent the real behavior.



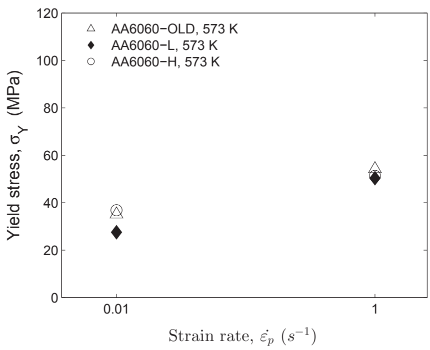
(a) $293K$



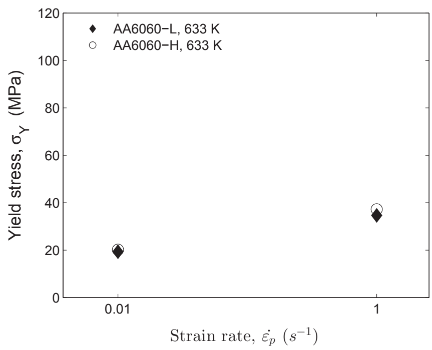
(b) $473K$



(c) $523K$



(d) $573K$



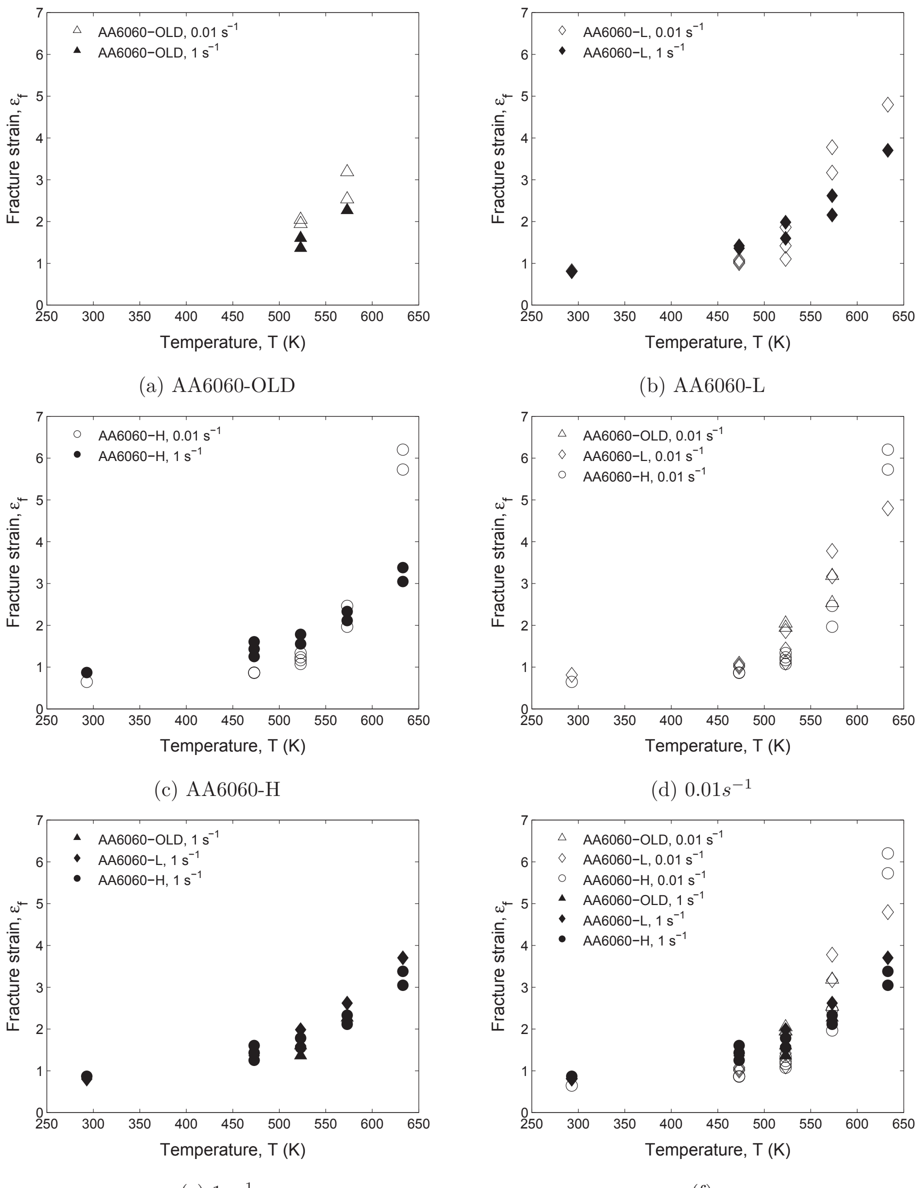
(e) $633K$

Figure 4.16: Plots (a)-(e) show the yield stress vs. strain rate for same temperature in the same plot

4.3.4 Fracture Strain

The fracture area has been measured using a microscope for all tests, and an estimate for the fracture strain has been calculated using Eq. (2.43). Values from all tests have been plotted because of the large deviation associated with the measurement from tests with same temperature and strain rate.

Fig. 4.17 illustrates the fracture strain vs. temperature for all tests. It is seen that the temperature dependency for the fracture strain is evidently strain rate sensitive, see Fig. 4.17b, c and f. The trend line for the temperature dependency for both strain rates have been plotted for the AA6060-L and AA6060-H alloy and can be seen in Fig. 4.18. Based on the microscope measurements, the fracture strain seems to show stronger dependence on the temperature for $0.01s^{-1}$ strain rate compared to $1s^{-1}$ strain rate for both materials.

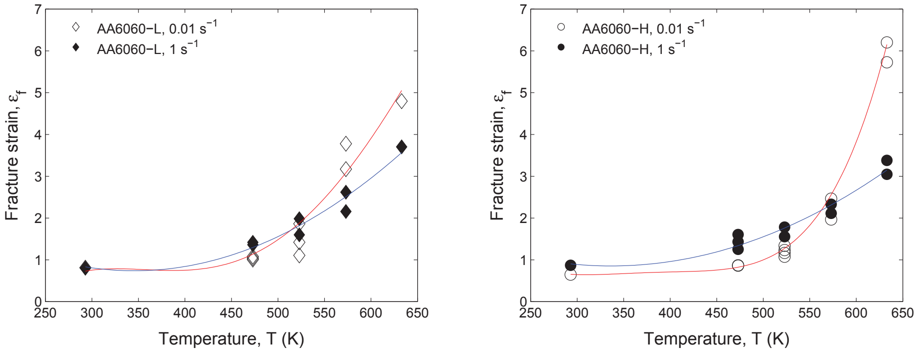


(a) AA6060-OLD (b) AA6060-L

(c) AA6060-H (d) $0.01s^{-1}$

(e) $1s^{-1}$ (f)

Figure 4.17: Plots (a)-(c) show fracture strain vs. temperature for $0.01s^{-1}$ and $1s^{-1}$ strain rate for respectively (a) AA6060-OLD, (b) AA6060-L and (c) AA6060-H. Plots (d)-(e) show fracture strain vs. temperature for respectively (d) $0.01s^{-1}$ and (e) $1s^{-1}$ strain rate. Plot (f) show plot (d) and (e) together

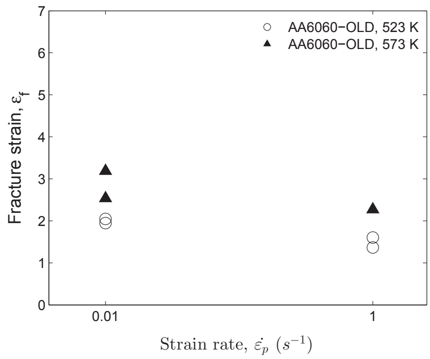


(a) AA6060-L

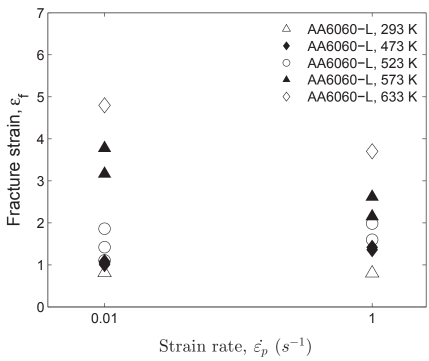
(b) AA6060-H

Figure 4.18: Plots show fracture strain vs. temperature for $0.01s^{-1}$ and $1s^{-1}$ strain rate for respectively (a) AA6060-L (same as Fig. 4.17b) and (b) AA6060-H (same as Fig. 4.17c) plotted together with trend lines

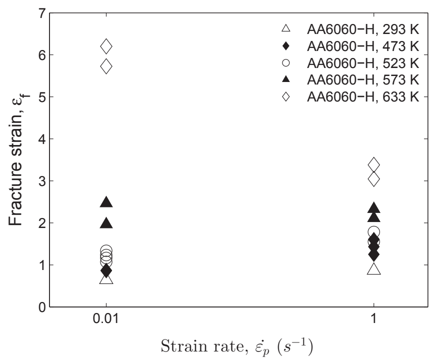
Fig. 4.19 illustrates the fracture strain vs. strain rate for all tests. Fig. 4.20 shows the results for the AA6060-L and AA6060-H alloy plotted together with trend lines. As can be seen, the strain rate dependency seems to be quite sensitive to the temperature. At room temperature, $293K$, there seems to be essentially no strain rate dependency. For temperatures of $473K$ and $523K$ the fracture strain is lower for $0.01s^{-1}$ strain rate compared to $1s^{-1}$ strain rate, while the inverse dependency is seen for temperatures of $573K$ and $633K$ as the fracture strain is significant higher for $0.01s^{-1}$ strain rate. However, there are very much uncertainties associated with the measuring of the fracture area using a microscope for small fracture areas. Thus, the validity of the largest estimated values for the fracture strain must be carefully addressed before any conclusions can be drawn.



(a) AA6060-OLD

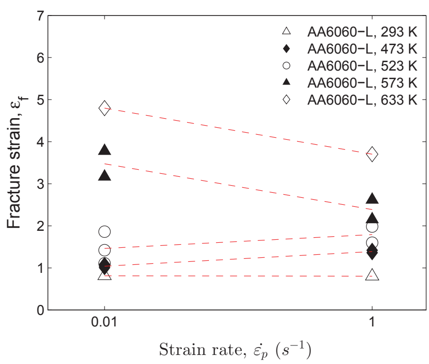


(b) AA6060-L

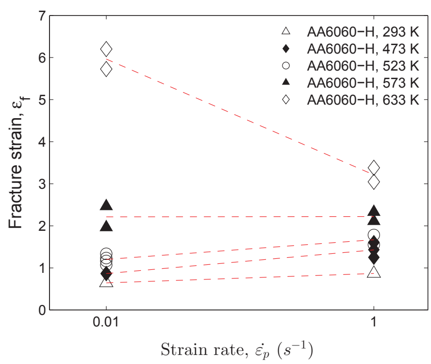


(c) AA6060-H

Figure 4.19: Plots (a)-(c) show fracture strain vs. strain rate for all temperatures for respectively (a) AA6060-OLD, (b) AA6060-L and (c) AA6060-H



(a) AA6060-L



(b) AA6060-H

Figure 4.20: Plots show fracture strain vs. strain rate for all temperatures for respectively (a) AA6060-L (same as Fig. 4.19b) and (b) AA6060-H (same as Fig. 4.19c) plotted together with trend lines

4.4 Post-Processing of Data from SHTB Experiments

Post-processing of experimental data from both strain gauges and local measurement from camera recordings have been done for the SHTB experiments. The procedure for post-processing the strain gauge data is quite similar as for the quasi-static tests as described in Section 4.2 and will therefore not be as thoroughly described here. As opposed to the quasi-static tests, where values for force and displacement are given in the output files, values for engineering stress and uncorrected engineering strain are given from the strain gauge measurements from the SHTB tests. The post-processing of data from local measurement from camera recordings is a much more tedious procedure, but provides essential data such as the local strain rate and potentially the true stress-plastic strain curve until fracture, and will be explained more detailed.

Measurements from strain gauges

When comparing the engineering stress-strain curve from quasi-static experiments with SHTB experiments, some different behavior is apparent. First of all, the stress-strain curve in the elastic region is rather slightly curved than linear for the entire region, thus making it more difficult to approximate both the measured Young's modulus and the yield stress. There is also a noticeably stress plateau after plastic strain of ≈ 0.01 for most tests. Both the slightly curved stress-strain curve in the elastic region and the stress plateau can be seen in Fig. 4.21. The reason for these two observed phenomena has not been further investigated.

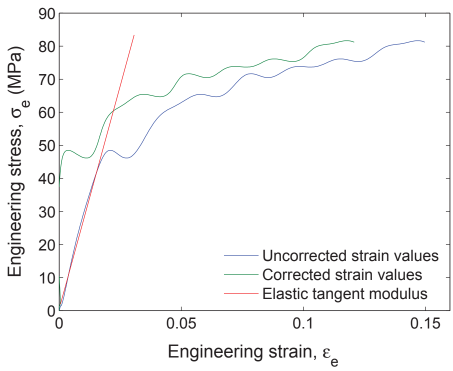
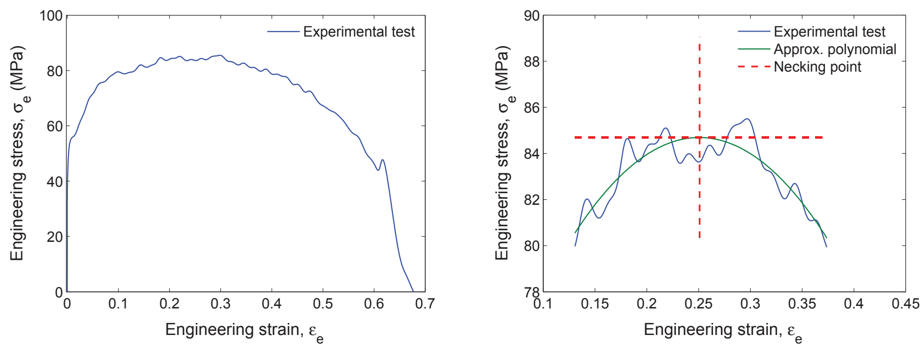


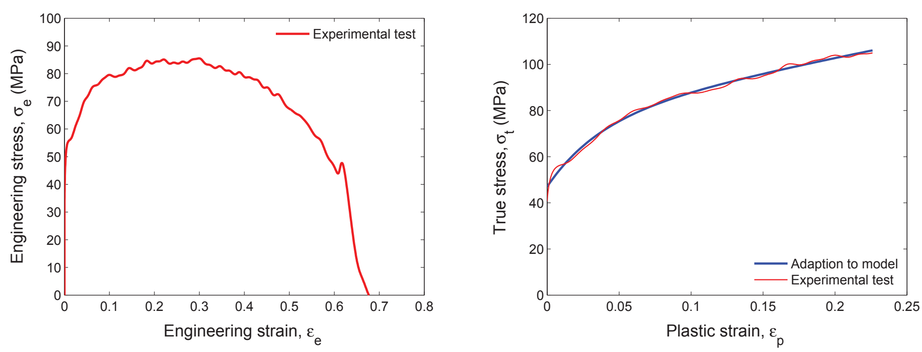
Figure 4.21: Plot showing the measured Young's modulus together with both uncorrected and corrected strains

The same procedure is used for calculating the necking point for SHTB experiments as for quasi-static experiments. As can be seen from Fig. 4.4a and Fig. 4.22a, there are noticeably more fluctuations in the stress-strain curve from the SHTB tests. As a result of this, it may be even more difficult to determine the necking point from SHTB tests compared with quasi-static tests. Fig. 4.22b illustrates how the necking point is calculated for a typical SHTB test. Plots of the resulting engineering stress-strain and true stress-plastic strain curves, as shown in Fig. 4.23a and b, are automatically created and saved. The true stress-plastic strain curves are fitted with Voce hardening parameters which takes on the form $\sigma_t = \sigma_Y + \sum_{i=1}^{2} Q_i(1 - e^{-C_i \varepsilon_p})$. The resulting plots, yield stress and Voce hardening parameters for all tests can be found in Appendix F.



(a) (b)

Figure 4.22: (a) Engineering stress-strain curve from a test where the necking point is difficult to locate and (b) determination of necking point using an approximated polynomial

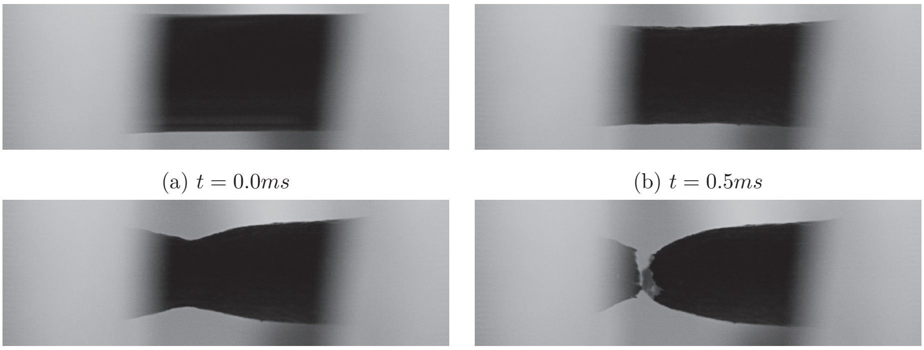


(a) (b)

Figure 4.23: (a) Engineering stress-strain plot and (b) true stress-plastic strain plot from post-processing

Local measurement from camera recordings

The motivation for using camera measurements has been discussed in Section 3.3. Fig. 4.24 shows a series of representative frames from the camera recordings during a SHTB test. Frame (a) is at the beginning of the test, frame (b) shows that the specimen diameter has been reduced, but the neck cannot easily be located, frame (c) shows clearly where the neck is located and its shape, while frame (d) is post-fracture and at the end of the test. The white object that can be seen on both sides is the coil.



(a) $t = 0.0ms$ (b) $t = 0.5ms$

(c) $t = 1.5ms$ (d) $t = 2.0ms$

Figure 4.24: (a)-(d) Representative frames from high-speed camera showing the different stages during an experimental test [37]

The frames from the camera recordings have been post-processed by PhD candidate Vincent Vilamosa at SIMLab at NTNU using Matlab. The frames from the camera are used to calculate the minimum diameter of the specimen at the neck during deformation until fracture. From this, both true strain ε_t and true stress σ_t can easily be calculated from the following equations:

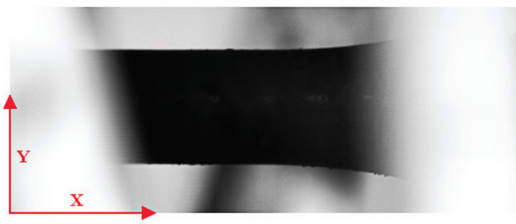
$$\varepsilon_t = ln(\frac{A_0}{A_s}) = 2ln(\frac{D_0}{D_s}) \tag{4.2}$$

$$\sigma_t = \sigma_e \frac{A_0}{A_s} = \sigma_e \frac{D_0^2}{D_s^2} \tag{4.3}$$

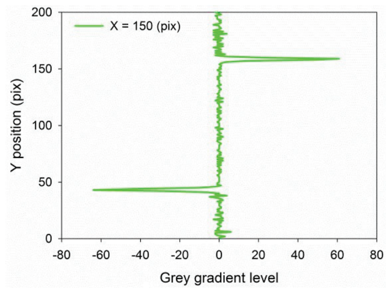
where A_0 is the initial area, A_s is the minimum area measured at the neck, D_0 is the initial diameter, D is the minimum diameter measured at the neck and σ_e is the engineering stress. The true stress can also be calculated from the loading force $F = \sigma_e A_0$:

$$\sigma_t = \frac{F}{A_s} = \frac{F}{\pi R_s^2} = \frac{F}{\frac{\pi}{4} D_s^2} \tag{4.4}$$

The pixel size in the frames is approximately $20x20\mu m^2$, thus approximately 150 pixels are used to represent the specimen diameter of $3mm$. To determine the edges of the specimen, both the gray level and the gray gradient level can be used. By using the latter, the accuracy is improved with a factor of 10 relative to using the former as the gray gradient level provides sub-pixel information [37]. Hence, measurement of displacements are narrowed down to $2\mu m$ from the camera recordings, while in comparison strain gauge measurements have a lower limit of $0.3\mu m$ [37]. The gray gradient level for a typical frame can be seen in Fig. 4.25b. The two peaks in Fig. 4.25b corresponds to the two edges of the sample in Fig. 4.25a. The minimum diameter of the specimen can be calculated by first calculating the difference in Y-axis position for both peaks for all positions along the X-axis. By applying this procedure for all frames, values for the real minimum diameter are then found for each frame taken during the whole test.



(a)



(b)

Figure 4.25: (a) Typical frame from high-speed camera and (b) gray gradient level along the Y axis for a given position X corresponding to (a) [37]

As significant necking was observed during the tests, especially at elevated temperatures, the stress state in the specimen is believed to be rather tri-axial than uni-axial. The Bridgman correction factor is discussed in Section 2.1.3 and has been applied to correct the stress values. Fig. 4.26 illustrates the principle for calculating the radius of curvature in the necked section. The shape of the edge is approximated with a 10^{th} degree Chebyshev polynomial. The area of interest for determining the circle, and also the radius of curvature, see the blue line in Fig. 4.26, is bounded by the points where the second derivative of the Chebyshev polynomial is equal to zero. A best fit for the circle is approximated using least-squares method (lsqnonlin function) in Matlab.

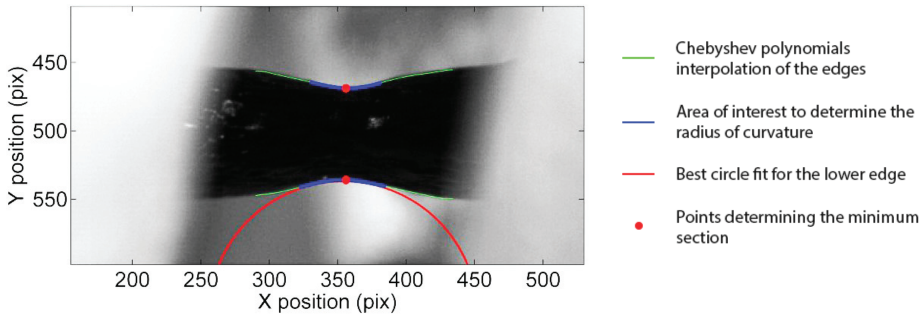
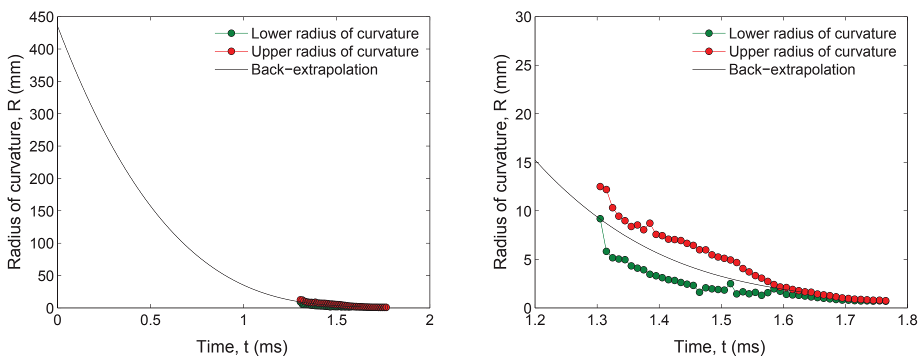


Figure 4.26: Determination of radius of curvature in the necked section [37]

For small deformations, i.e. large values for the radius of curvature, some part of the area of interest may be hidden behind the coil. This is solved by back-extrapolating the values for the radius of curvature to the beginning of the test. Fig. 4.27 shows an example of the back-extrapolation together with calculated values from both upper and lower part of the specimen. Note that the Bridgman correction should only be applied to the stress values after onset of necking, and the real value of the radius of curvature is in fact equal to ∞ until this point because no localized deformation will be present within the gauge section. Back-extrapolating to the beginning of the test and applying the correction formula to the stress for the whole plastic strain range may therefore give rise to non-conservative stress values until onset of necking, but, as can be seen from the resulting true stress-plastic strain curves in Appendix F, the correction of the stress values until necking is almost not detectable. This, will of course, depend on the slope of the curve for the back-extrapolated radius of curvature.



(a) (b)

Figure 4.27: (a)-(b) show the interpolation and back-extrapolation for determination of radius of curvature from a typical test

As mentioned, the applicability of the Bridgman correction rely on the possibility to measure the geometry of the neck during deformation. This is usually both time consuming and not easily achieved, and a purely empirical way to determine the geometry is of course desired. Le Roy et al. [33] states that the ratio of the minimum radius at the neck and the radius of curvature can be estimated from an empirical formula:

$$\frac{a}{R} = \begin{cases} 0 & \bar{\varepsilon} \leq \varepsilon_u \\ \kappa(\bar{\varepsilon} - \varepsilon_u) & \bar{\varepsilon} > \varepsilon_u \end{cases} \tag{4.5}$$

where a is the minimum radius at the neck, R is the radius of curvature, κ is a material constant, $\bar{\varepsilon}$ is the plastic strain and ε_u is the strain value at onset of necking. Le Roy et al. [33] have investigated this empirical formula for steel and have estimated the factor κ to 1.11. The ratio of the minimum radius at the neck and the radius of curvature and the Bridgman corrected stress using Eq. (4.5) and the estimated factor has been calculated for all tests and are compared with direct measurements from camera recordings. An example of the comparison is presented in Fig. 4.28, and the results from all tests can be found in Appendix F.

As seen in Fig. 4.28a, the shape of the ratio as function of the plastic strain seems to be somewhat coinciding with camera measurements. However, it cannot be seen directly from the figure what impact the deviation will have on the resulting corrected true stress-plastic strain curve. Fig. 4.28b shows the resulting true stress-plastic strain curve for this particular test. It is seen that the empirical formula provides non-conservative correction of the true stress, and this is also seen for all other tests from the experiments, ref. Appendix F. This indicates that the material parameter κ set equal to 1.11 cannot be readily applied to aluminum and the AA6060 alloy, but an adequately approximation may be achieved for an altered value for κ as the empirical formula predicts deviations for the stress within approximately the same range for all tests.

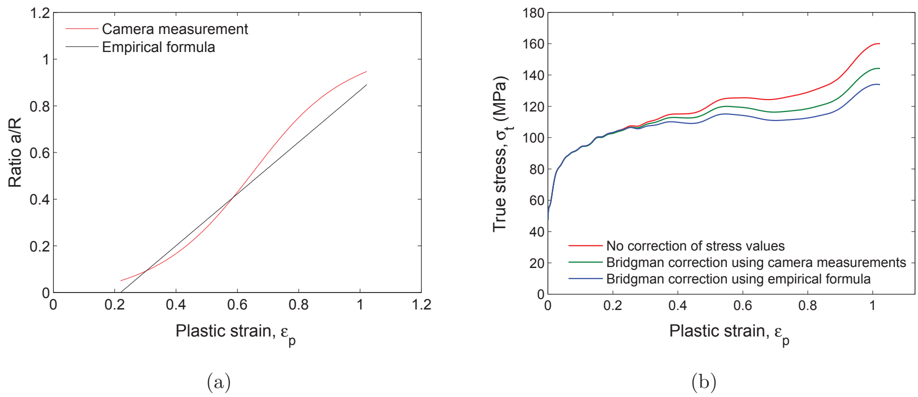
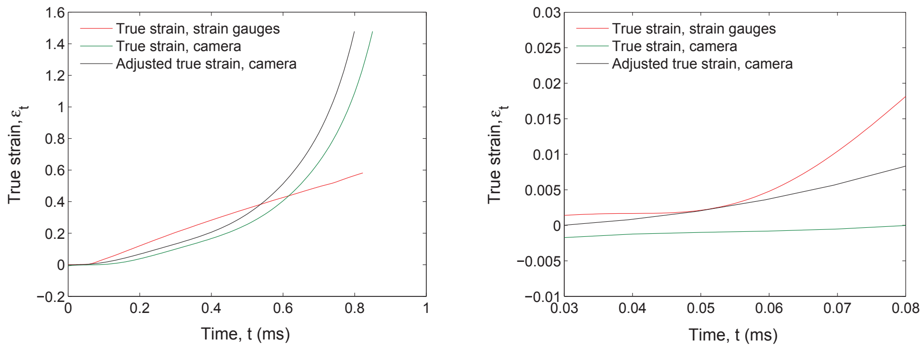


Figure 4.28: (a) show the minimum radius at the neck-radius of curvature ratio from both camera measurements and the empirical formula and (b) the resulting true stress-plastic strain plot using Bridgman correction for a typical test

Data for true strain, minimal diameter and time from all tests provided by Vincent Vilamosa are used to calculate the uncorrected true stress-plastic strain values until the plastic strain value corresponding to the maximum value of the true stress. Values for true strain and minimal diameter have to be synchronized in terms of time with respect to the experimental data from the strain gauges to match the force values, or engineering stress values, with the corresponding values for true strain and minimal diameter. Fig. 4.29 shows how this is done using Matlab. The values are believed to be synchronized when the two true strain vectors have the same values at the beginning of straining. The correct time delay value was found by trial and error until the two curves are coincident at the beginning of straining.

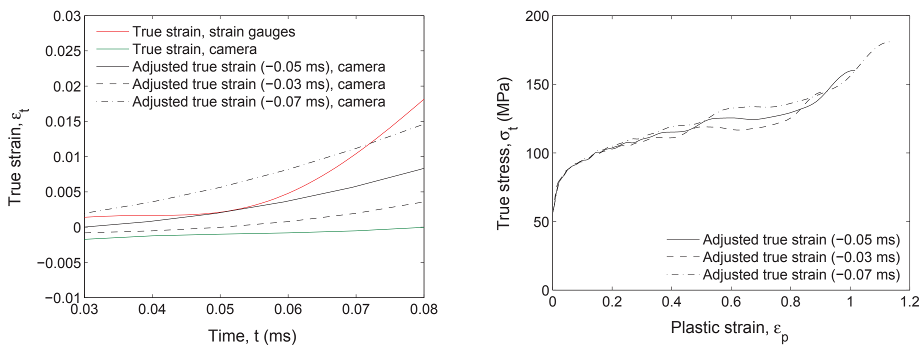
Note that the higher strain values from the strain gauge measurements are due to the straining taking place outside the gauge section as compared to the local strain values. It is also interesting to note that the strain curve from the local measurement is linear, thus having a constant strain rate, until a certain point where the curve suddenly becomes non-linear that implies an increase of strain rate. This point should coincide with the necking point calculated from the engineering stress-strain curve, as the increase of strain rate is due to the localized deformation taking place within the necked section.



(a) (b)

Figure 4.29: (a) Synchronization of time for strain values from camera measurement and (b) showing the same plot for a zoomed in area

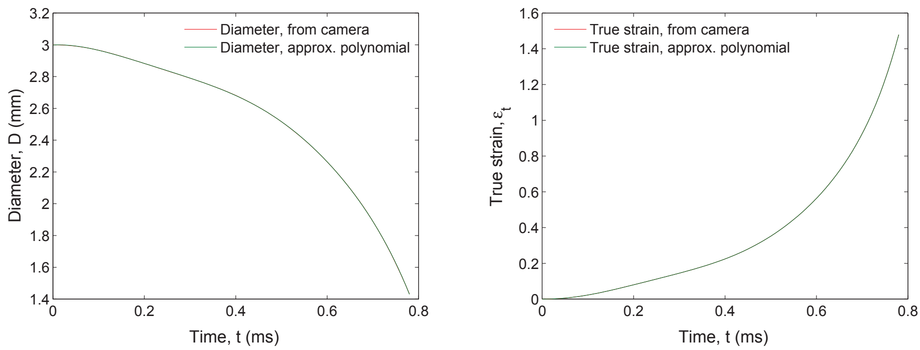
To study the influence of the synchronization of true strain and diameter values on the resulting true stress-plastic strain curve, synchronization has been done for three time delay values for a representative test: $0.05ms$ (which is believed to be the correct value), $0.03ms$ and $0.07ms$. Fig. 4.30a shows the resulting true strain curve with respect to time. It can be seen that the curves for $0.03ms$ and $0.07ms$ time delay are clearly lying on each side of the strain curve from the strain gauge measurements, such that the correct value for this particular test is believed to be bounded by these values, and probably close to $0.05ms$. Fig. 4.30b shows that the true stress-plastic strain curve is not highly dependent on the time delay value for small strains, but for larger strains the influence will be significant, hence it is necessary to account for this.



(a) (b)

Figure 4.30: (a) Synchronization of time using different values and (b) showing the resulting true stress-plastic strain curve

The true stress corresponding to the true strain measured by the camera is calculated using Eq. (4.4). Since the record frequency is much higher for strain gauge measurements compared with camera measurements, the diameter and strain values have to be fitted with a polynomial so that the engineering stress values can be coupled with correct values for diameter and strain in terms of time. An 8^{th} degree polynomial has been used for the approximation and it can be seen from Fig. 4.31 that the fit is adequate.



(a) (b)

Figure 4.31: Data from post-processing and approximated polynomial for (a) minimum diameter and (b) true strain

Plots of the resulting true stress-plastic strain curves are automatically created and saved using the script for post-processing. The true stress-plastic strain curves are fitted with Voce hardening parameters which takes on the form $\sigma_t = \sigma_Y + \sum_{i=1}^{2} Q_i(1 - e^{-C_i \varepsilon_p})$. The resulting plots, yield stress and Voce hardening parameters for all tests can be found in Appendix F.

4.5 Results from SHTB Experiments

In this Section, results from the SHTB experiments are presented. An overview of the experimental program is given in Section 4.5.1, while results for the strain hardening and yield stress are presented in Section 4.5.2.

4.5.1 Experimental Program

An overview of the experimental program for the SHTB tests is presented in Table 4.2. Tests have been performed for strain rates between $340s^{-1}$ and $800s^{-1}$ and for temperatures ranging from $523K$ to $673K$. Only four tests in total were successfully conducted, such that only the results from these tests are presented. The reason for the large number of unsuccessful tests have not been investigated, but it is believed that it might be that the bars were not cooled down prior to a new test, thus modifying the response registered from the strain gauges. The strain rates tabulated in Table 4.2 is measured by the strain gauges and is coherent with the strain rates measured from the camera recordings until necking.

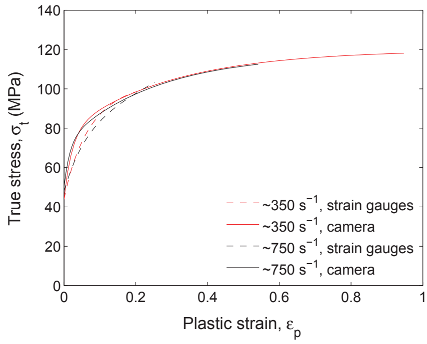
AA6060-L				AA6060-H			
Test #	Strain rate (s^{-1})	Temp. ($^\circ K$)	Comment	Test #	Strain rate (s^{-1})	Temp. ($^\circ K$)	Comment
7	340	523	N.s.*	1	350	523	N.s.*
8	372	523	N.s.*	2	376	523	OK
9	354	523	OK	3	365	613	N.s.*
10	460	523	N.s.*	4	388	673	N.s.*
11	781	523	OK	5	789	523	OK
		523	N.s.*	6	800	573	N.s.*

* N.s. = not successful experiment

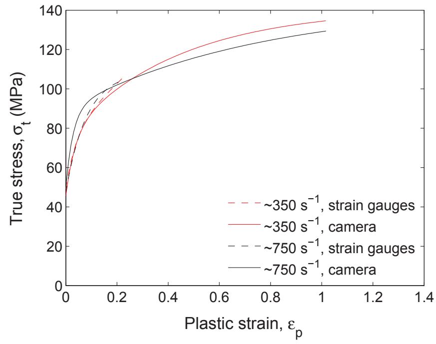
Table 4.2: Overview of experimental program for SHTB tests

4.5.2 Experimental Results

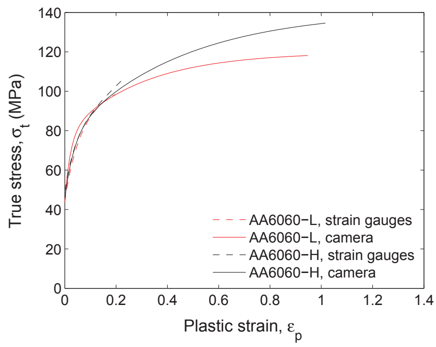
Fig. 4.32 shows the results from the SHTB experiments and both strain gauge measurements and camera measurements are shown. The strain hardening seems to be hardly influenced by the strain rate for both alloys, while it can be seen that the AA6060-H alloy seems to harden more than the AA6060-L alloy for $\sim 350s^{-1}$ strain rate. From Fig 4.32f it is seen that the yield stress measured from the strain gauges are $3-5\%$ lower compared to the local measurement, and may be explained by the reason that the strain was measured in two different ways. It is also seen that the yield stress for AA6060-H is about $2-5\%$ higher than for AA6060-L for both strain rates.



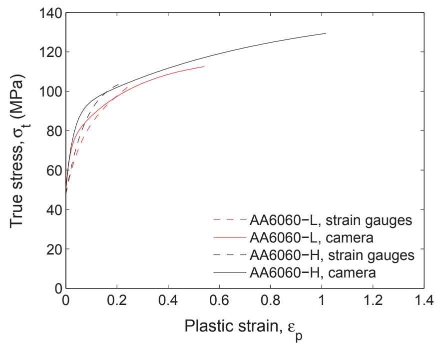
(a) AA6060-L



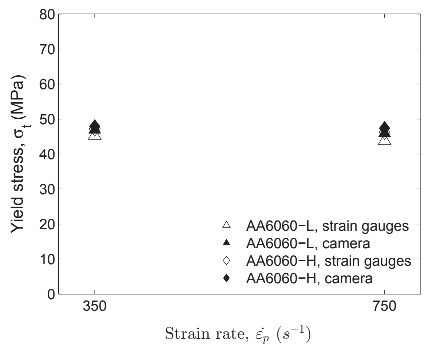
(b) AA6060-H



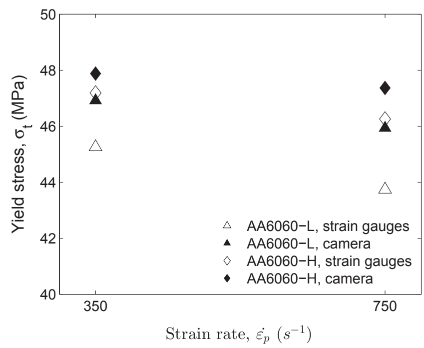
(c) $\sim 350s^{-1}$ strain rate



(d) $\sim 750s^{-1}$ strain rate



(e)



(f)

Figure 4.32: Plot (a)-(b) show the strain hardening for respectively the AA6060-L and AA6060-H alloy. Plot (c)-(d) show the strain hardening for both alloys together for respectively $\sim 350s^{-1}$ and $\sim 750s^{-1}$ strain rate. Plot (e)-(f) show the yield stress vs. strain rate

5 Calibration of Material Models

As not a sufficient number of SHTB experiments to obtain relevant material data for the two alloys studied for a complete temperature and strain rate range was conducted during the time period for this thesis, an old database with material data for a complete temperature and strain rate range for the AA6060-OLD alloy, both strain gauge measurements and camera measurements, has been used to calibrate three material models. This is done in order to study whether it is possible to get an adequate fit for a wide range of strain rates and temperatures using a material model containing only a limited number of parameters. The three material models that are calibrated are the modified Johnson-Cook model, the modified Zerilli-Armstrong model and a combined material model suitable for materials showing typical behavior from both BCC and FCC metals. The reason for choosing the two former material models is that they are widely used and easily implemented in non-linear finite element codes such as LS-DYNA, while the latter model is chosen as it is believed to predict accurate results because the AA6060-OLD alloy show material behavior seen in both BCC and FCC metals, i.e. strong dependence of both the yield stress and strain hardening on temperature and to some degree on strain rate. The three material models have been thoroughly presented in Section 2.4.

The material model calibration has been done for two temperature ranges for all three material models; first range covering all temperatures ($295K$ to $827K$), the other range covering temperatures from $450K$ to $827K$. The reason for choosing the latter temperature range is that the yield stress is decreasing at a higher rate with respect to temperature increase for temperatures exceeding approximately $450K$ (e.g. see Fig. 5.1), such that this range is believed to provide a better fit for the calibrated models. The material model calibration has been done using least-squares method (lsqnonlin function) in Matlab. The procedure for the calibration of each model is described in their respective Sections. The complete Matlab code for the calibration of the three mentioned material models can be found in Appendix D.4 and is readily suitable for other experimental data by small alterations of the code.

5.1 Modified Johnson-Cook Model

The modified Johnson-Cook model is presented in Section 2.4.1 and defined by Eq. (2.13) which is reviewed here:

$$\sigma = (A + \sum_{i=1}^{2} Q_i(1 - e^{-C_i \varepsilon_p}))(1 + \dot{\varepsilon_p}^*)^C(1 - T^{*m}) \tag{5.1}$$

The calibration is done in three steps (referring to Eq. (5.1)):

Step 1: Calibration of first bracket only, thus the hardening part, taking into account tests at room temperature and at quasi-static strain rates only.

Step 2: Calibration of second bracket, thus the strain rate sensitivity, taking into account tests at room temperature and at all strain rates.

Step 3: Calibration of third bracket, thus the thermal softening, taking into account all tests from the database.

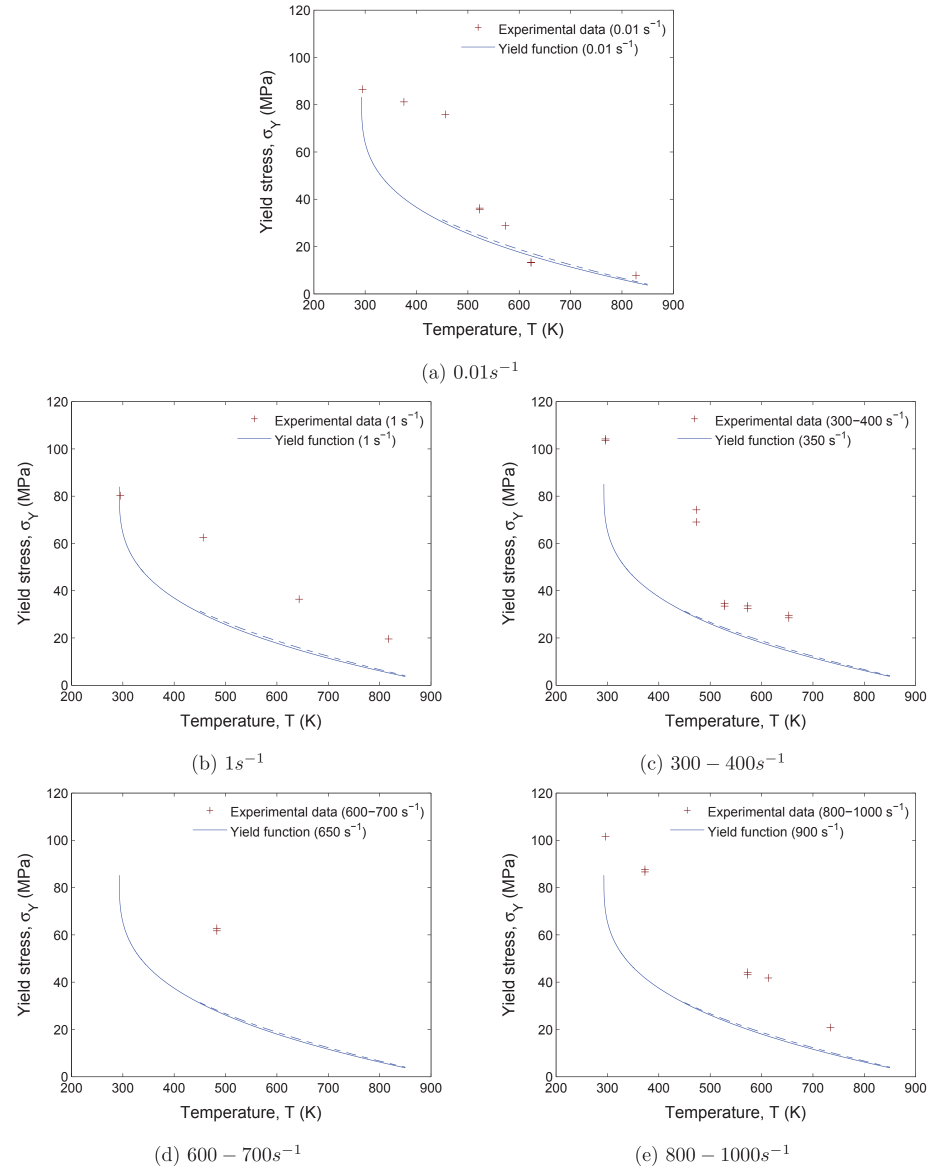
The resulting parameters are tabulated in Table 5.1 and the resulting yield stress and strain hardening plots can be found in Fig. 5.1-5.3.

It can be seen that the model does not catch the shape of the yield stress function when plotted vs. temperature, and the values are in general too low (Fig. 5.1). The experimental data for yield stress plotted vs. strain rate show no identifiable shape of the yield function, and the somewhat fluctuating values may to some degree be explained by that they are plotted within a $100K$ to $150K$ temperature range. It is clearly seen from Fig. 5.3 that no good fit was found for the model. For all temperature intervals, and for both set of parameters fitted for the two temperature ranges, the model predicts in general too low strain hardening. It is also interesting to notice that the fit for the narrower temperature range seems to predict almost the exact yield stress as the model taking into account all temperatures, but that the true stress-plastic strain curves deviate even more.

Material model parameters (modified Johnson-Cook model)

Model parameters	$T \in (293 - 850K)$	$T \in (450 - 850K)$
A (MPa)	83.16	67.74
Q_1 (MPa)	351.6	81.18
C_1	0.9510	4.167
Q_2 (MPa)	74.19	47.17
C_2	20.62	41.53
C	0.002202	2.220E-14
m	0.3236	0.4426

Table 5.1: Calibrated material model parameters for the modified Johnson-Cook model for two temperature ranges



(a) $0.01s^{-1}$

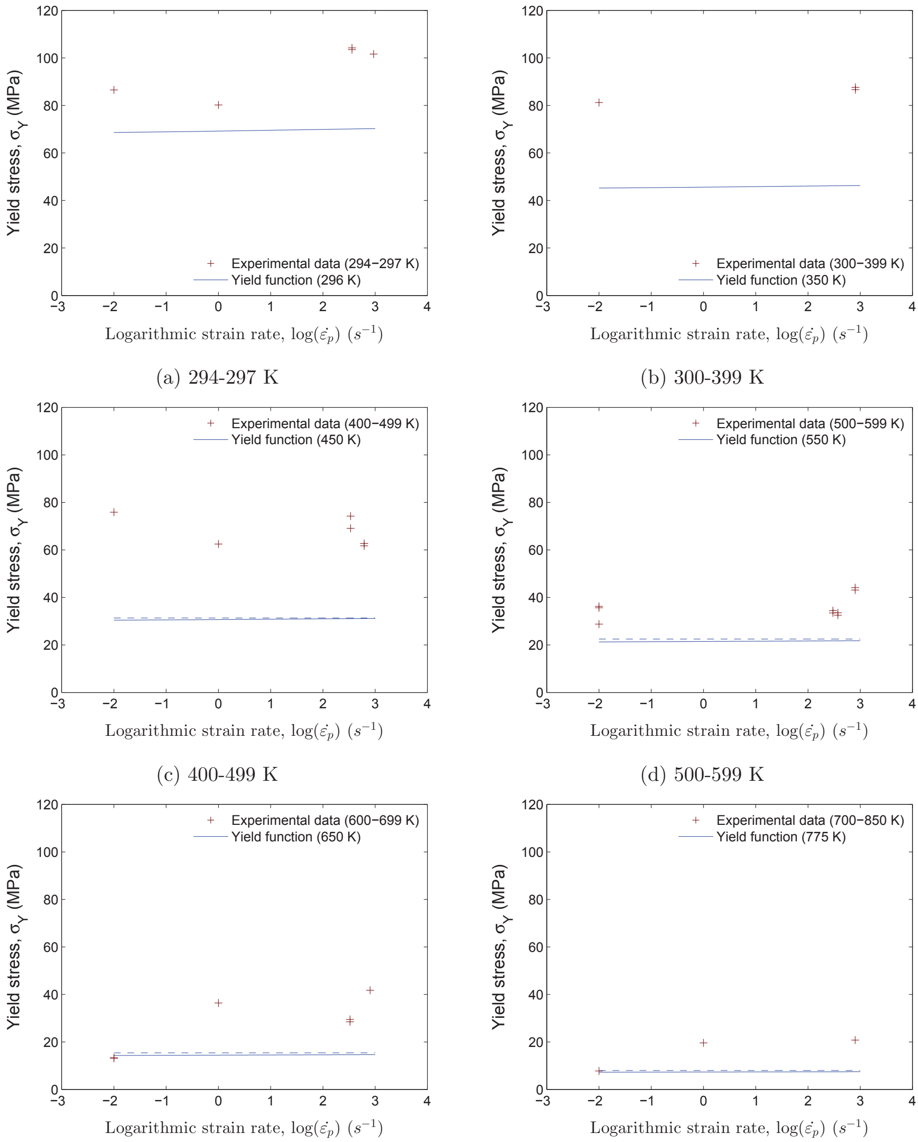
(b) $1s^{-1}$

(c) $300 - 400s^{-1}$

(d) $600 - 700s^{-1}$

(e) $800 - 1000s^{-1}$

Figure 5.1: Plot (a)-(e) show the yield stress function vs. temperature for the modified Johnson-Cook model together with experimental data. The solid line is for the parameters fitted for the entire temperature range, while the dashed line is for the narrower range of temperature.



(a) 294-297 K

(b) 300-399 K

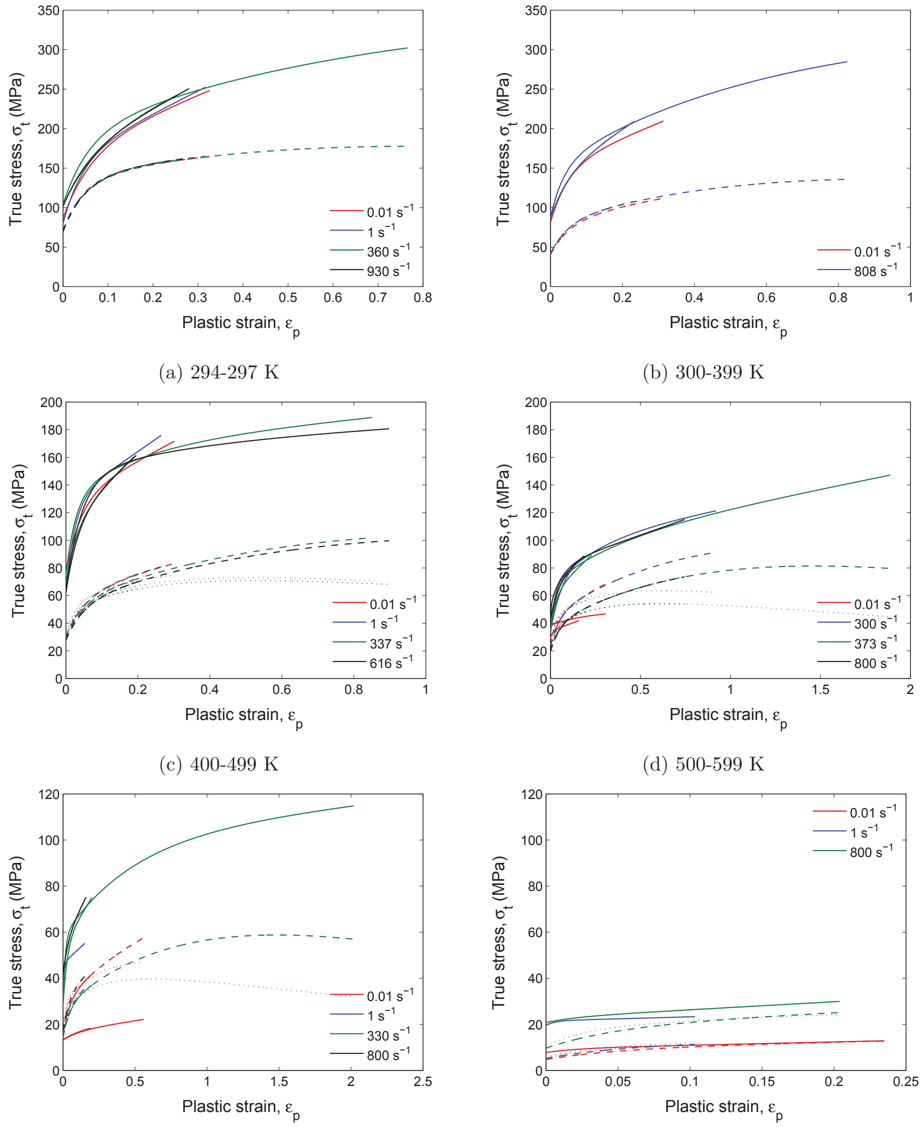
(c) 400-499 K

(d) 500-599 K

(e) 600-699 K

(f) 700-850 K

Figure 5.2: Plot (a)-(f) show the yield stress function vs. strain rate for the modified Johnson-Cook model together with experimental data. The solid line is for the parameters fitted for the entire temperature range, while the dashed line is for the narrower range of temperature.



(a) 294-297 K

(b) 300-399 K

(c) 400-499 K

(d) 500-599 K

(e) 600-699 K

(f) 700-850 K

Figure 5.3: Plot (a)-(f) show the true stress-plastic strain curve for the modified Johnson-Cook model together with experimental data. Solid lines are the experimental data, dashed lines are for the parameters fitted for the entire temperature range, while dotted lines are for the narrower range of temperatures and are therefore only plotted in (c)-(f)

5.2 Modified Zerilli-Armstrong Model

The original Zerilli-Armstrong model and a modified Zerilli-Armstrong model are presented in respectively Section 2.4.4 and Section 2.4.5. The modified model is defined by Eq. (2.33) and reviewed here:

$$\sigma = Y_a + (Y_0 + B\varepsilon_p^n)exp(-\beta_3 T + \beta_2 T ln\dot{\varepsilon}_p) \tag{5.2}$$

A slightly modified model of Eq. (5.2) is defined in LS-DYNA which the experimental data has been fitted for. The slightly modified model is defined as:

$$\sigma = Y_a + Y_0 exp(-(\beta_0 T - \beta_1 T ln\dot{\varepsilon}_p)) + B\varepsilon_p^n exp(-(\alpha_0 T - \alpha_1 T ln\dot{\varepsilon}_p)) \tag{5.3}$$

The calibration is done in two steps (referring to Eq. (5.3)):

Step 1: Calibration of yield stress, $\sigma_Y = Y_a + Y_0 exp(-(\beta_0 T - \beta_1 T ln\dot{\varepsilon}_p))$, taking into account all tests

Step 2: Calibration of strain hardening, $\sigma_H = B\varepsilon_p^n exp(-(\alpha_0 T - \alpha_1 T ln\dot{\varepsilon}_p))$, taking into account all tests

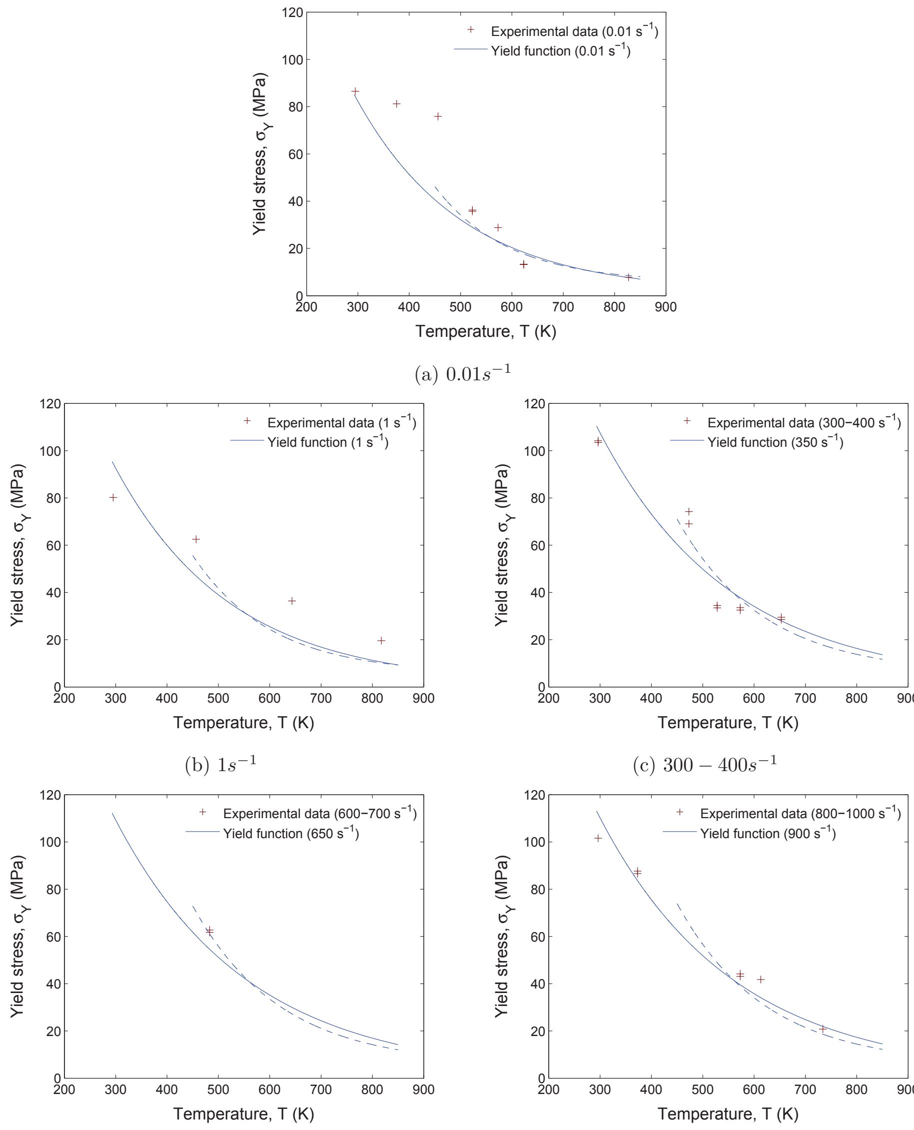
The resulting parameters are tabulated in Table 5.2 and the resulting yield stress and strain hardening plots can be found in Fig. 5.4-5.6.

The model predicts both too high and too low values for the yield stress within the whole range for temperatures and strain rates, but does not completely catch the shape of the yield function when plotted vs. temperature. As for the modified Johnson-Cook model, the modified Zerilli-Armstrong model also predicts in general too low strain hardening. When calibrated for a narrower temperature range, both the fit for yield stress and strain hardening seems to be slightly improved.

Material model parameters (modified Zerilli-Armstrong model)

Model parameters		$T \in (293 - 850K)$	$T \in (450 - 850K)$
Y_a	(MPa)	1.288	5.611
Y_0	(MPa)	343.6	946.4
β_0	(K^{-1})	0.004419	0.006533
β_1	(K^{-1})	0.00008647	0.0001018
B	(MPa)	1572	4284
n		0.5087	0.4114
α_0	(K^{-1})	0.006125	0.008581
α_1	(K^{-1})	0.0002146	0.0003635

Table 5.2: Calibrated material model parameters for the modified Zerilli-Armstrong model for two temperature ranges



(a) $0.01s^{-1}$

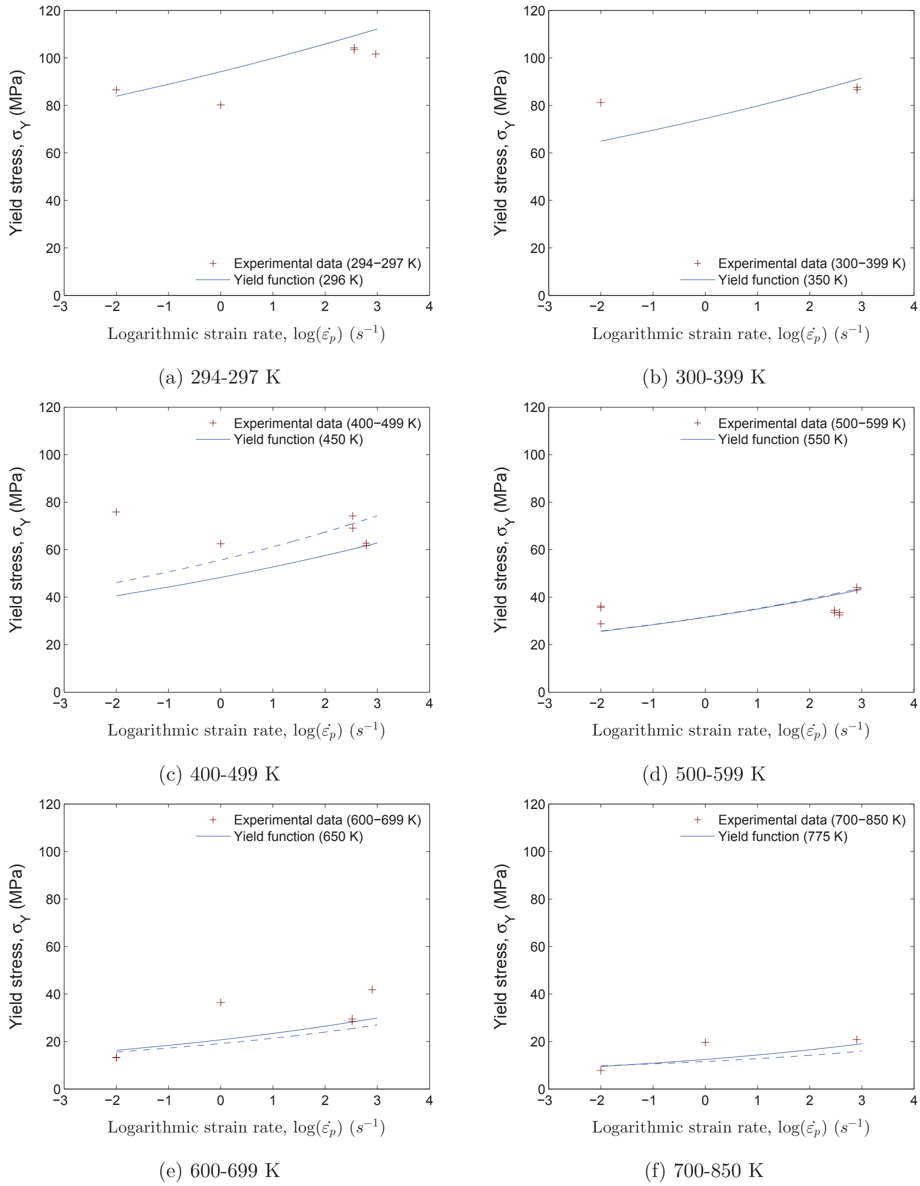
(b) $1s^{-1}$

(c) $300 - 400s^{-1}$

(d) $600 - 700s^{-1}$

(e) $800 - 1000s^{-1}$

Figure 5.4: Plot (a)-(e) show the yield stress function vs. temperature for the modified Zerilli-Armstrong model together with experimental data. The solid line is for the parameters fitted for the entire temperature range, while the dashed line is for the narrower range of temperature.



(a) 294-297 K

(b) 300-399 K

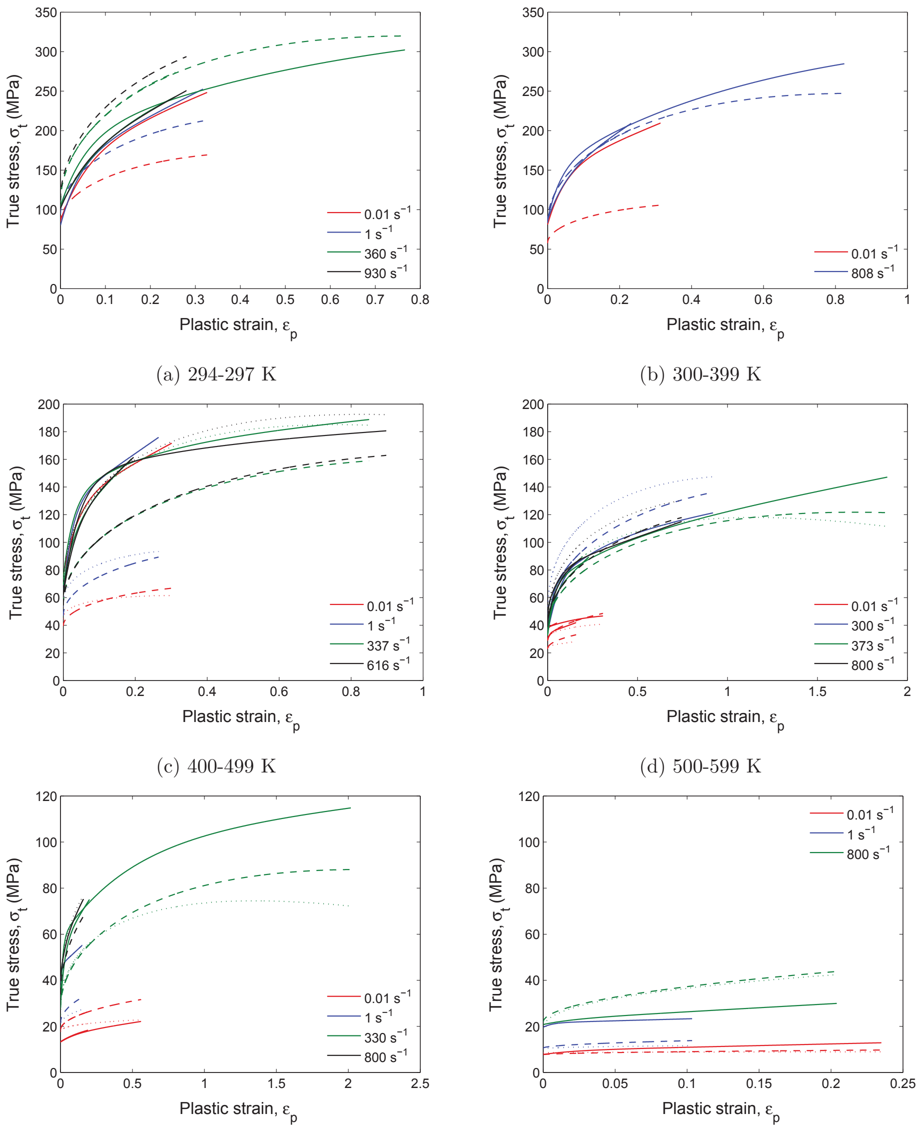
(c) 400-499 K

(d) 500-599 K

(e) 600-699 K

(f) 700-850 K

Figure 5.5: Plot (a)-(f) show the yield stress function vs. strain rate for the modified Zerilli-Armstrong model together with experimental data. The solid line is for the parameters fitted for the entire temperature range, while the dashed line is for the narrower range of temperature.



(a) 294-297 K

(b) 300-399 K

(c) 400-499 K

(d) 500-599 K

(e) 600-699 K

(f) 700-850 K

Figure 5.6: Plot (a)-(f) show the true stress-plastic strain curve for the modified Zerilli-Armstrong model together with experimental data. Solid lines are the experimental data, dashed lines are for the parameters fitted for the entire temperature range, while dotted lines are for the narrower range of temperatures and are therefore only plotted in (c)-(f)

5.3 Combined BCC and FCC Model

The combined material model suitable for materials showing typical behavior from both BCC and FCC metals is presented in Section 2.4.6 and defined by Eq. (2.40) which is reviewed here:

$$\sigma = Y_a + B_1\varepsilon_p^{n_1} + \hat{Y}(1-(\beta_1^Y T - \beta_2^Y T ln\dot{\varepsilon}_p)^{1/q})^{1/p} + B_2\varepsilon_p^{n_2}(1-(\beta_1^H T - \beta_2^H T ln\dot{\varepsilon}_p)^{1/q})^{1/p} \tag{5.4}$$

The calibration is done in two steps (referring to Eq. (5.4)):

Step 1: Calibration of yield stress, $\sigma_Y = Y_a + \hat{Y}(1 - (\beta_1^Y T - \beta_2^Y T ln\dot{\varepsilon}_p)^{1/q})^{1/p}$, taking into account all tests

Step 2: Calibration of strain hardening, $\sigma_H = B_1\varepsilon_p^{n_1} + B_2\varepsilon_p^{n_2}(1 - (\beta_1^H T - \beta_2^H T ln\dot{\varepsilon}_p)^{1/q})^{1/p}$, taking into account all tests

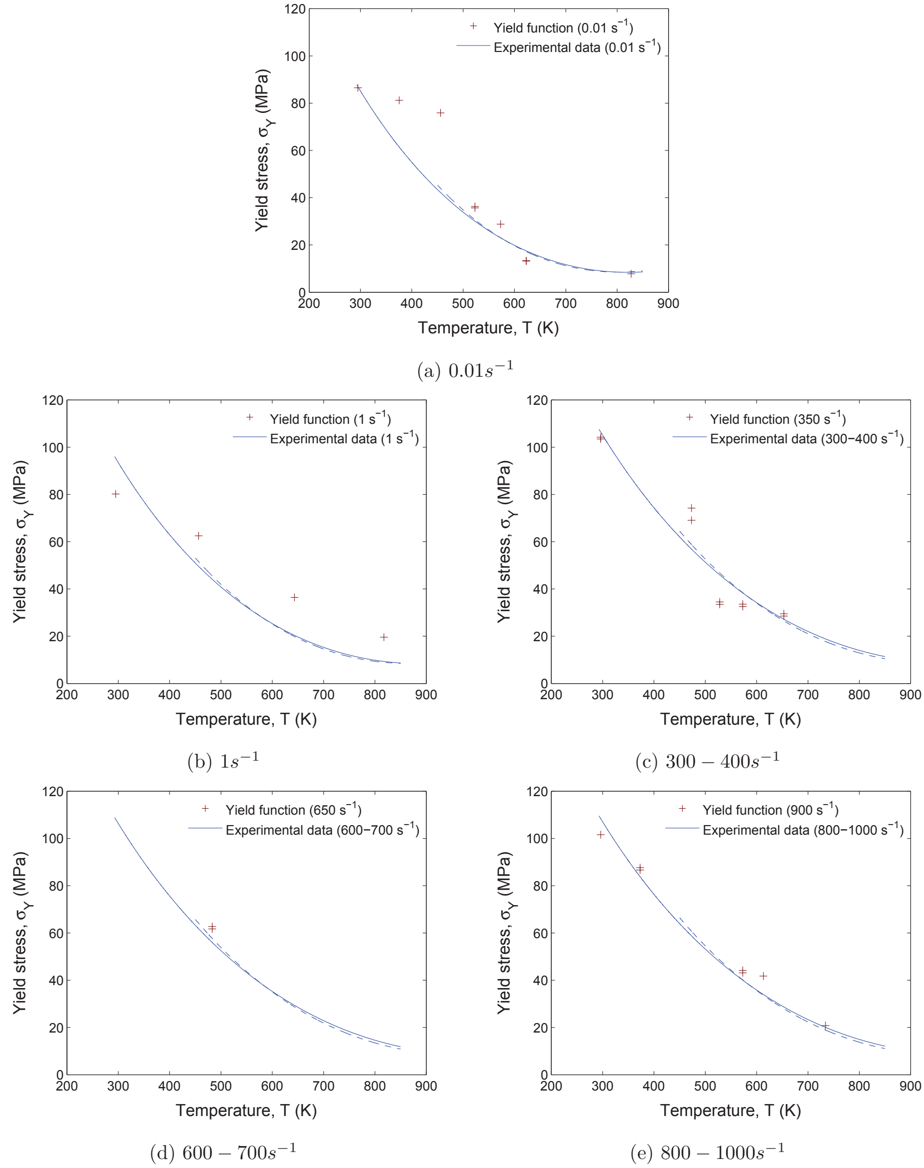
The resulting parameters are tabulated in Table 5.3 and the resulting yield stress and strain hardening plots can be found in Fig. 5.7-5.9.

It can be seen that the model predicts a yield stress function very similar to the modified Zerilli-Armstrong model, thus both too high and too low values within the whole range for temperatures and strain rates. The predicted true stress-plastic strain curves are also very much coinciding with the Zerilli-Armstrong model, but it seems to predict a slightly poorer fit. The same behavior is seen for the parameters fitted for the narrower temperature range as for the Zerilli-Armstrong model. Notice from Fig. 5.9 that the model predicts a rather unncorrect, to say the least, strain hardening curve for plastic strains exceeding approximately 1. This shape of the curve is of course not what any aluminum alloy would have shown from any experimental test.

Material model parameters (combined BCC and FCC model)

Model parameters	$T \in (293 - 850K)$	$T \in (450 - 850K)$
Y_a (MPa)	8.419	8.497
$\widehat{Y}$ (MPa)	321.1	360.1
B_1 (MPa)	4.999	6.063
B_2 (MPa)	1157	1187
β_1^Y (K^{-1})	0.001127	0.001159
β_2^Y (K^{-1})	0.00001981	0.00001898
β_1^H (K^{-1})	0.001231	0.001242
β_2^H (K^{-1})	0.00003125	0.00003053
n_1	3.450	3.363
n_2	0.4671	0.4198
p	0.5 (constant)	0.5 (constant)
q	1.5 (constant)	1.5 (constant)

Table 5.3: Material model parameters for the combined BCC and FCC model for two temperature ranges



(a) $0.01s^{-1}$

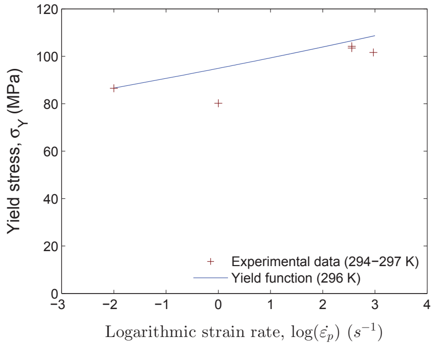
(b) $1s^{-1}$

(c) $300 - 400s^{-1}$

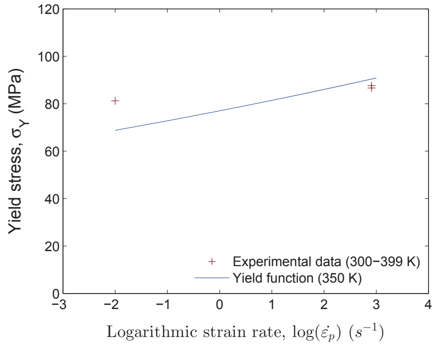
(d) $600 - 700s^{-1}$

(e) $800 - 1000s^{-1}$

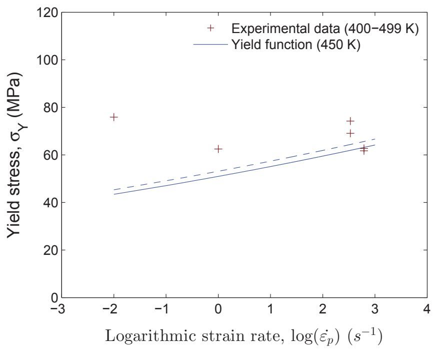
Figure 5.7: Plot (a)-(e) show the yield stress function vs. temperature for the combined BCC and FCC model together with experimental data. The solid line is for the parameters fitted for the entire temperature range, while the dashed line is for the narrower range of temperature.



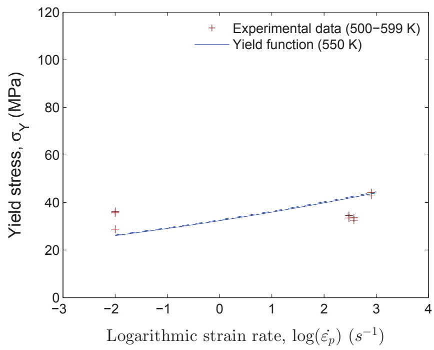
(a) 294-297 K



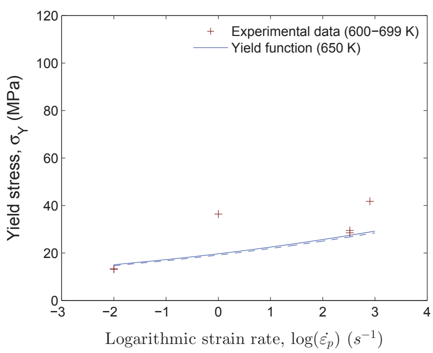
(b) 300-399 K



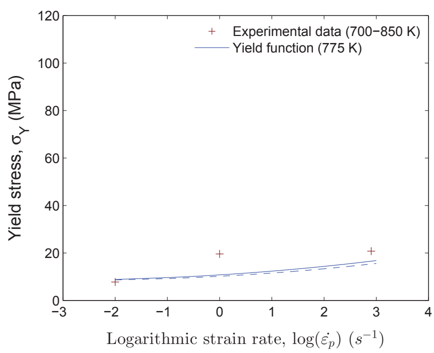
(c) 400-499 K



(d) 500-599 K

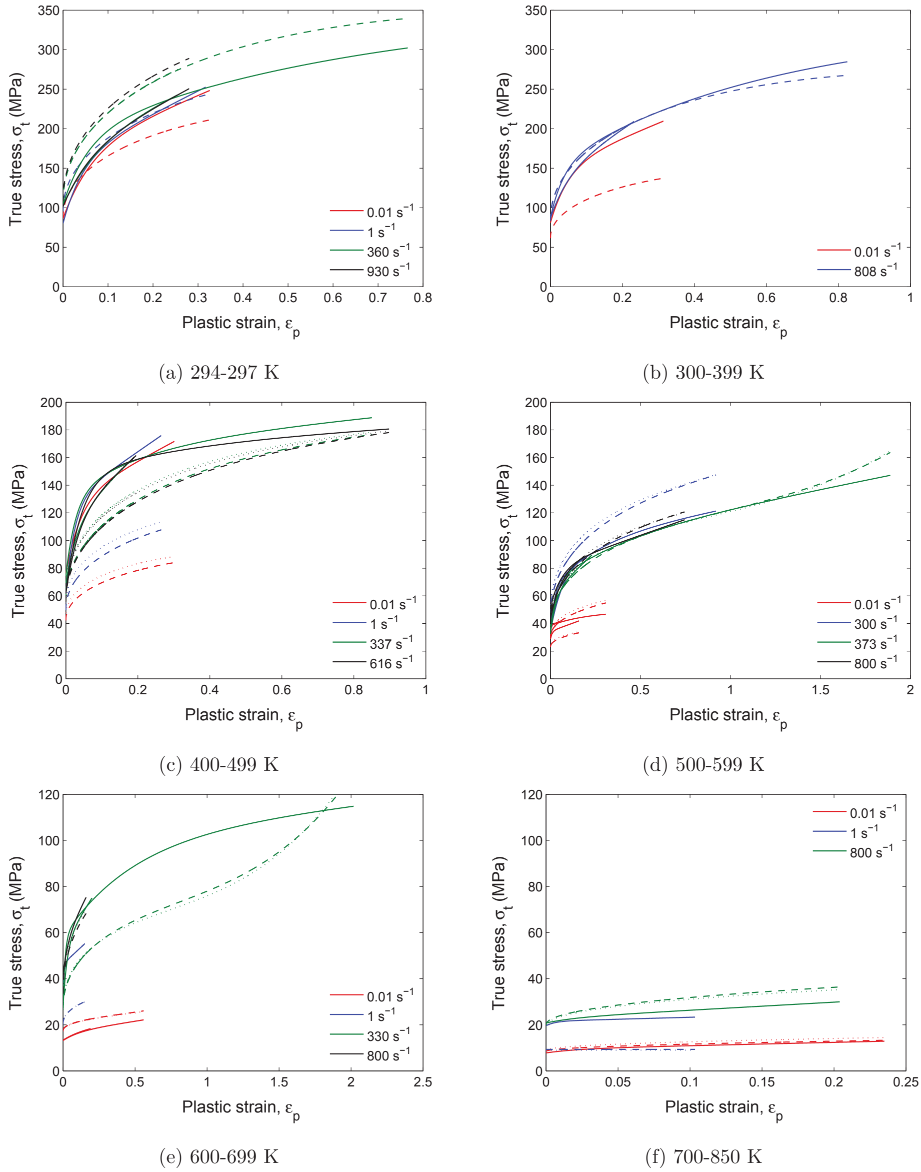


(e) 600-699 K



(f) 700-850 K

Figure 5.8: Plot (a)-(f) show the yield stress function vs. strain rate for the combined BCC and FCC model together with experimental data. The solid line is for the parameters fitted for the entire temperature range, while the dashed line is for the narrower range of temperature.



(a) 294-297 K

(b) 300-399 K

(c) 400-499 K

(d) 500-599 K

(e) 600-699 K

(f) 700-850 K

Figure 5.9: Plot (a)-(f) show the true stress-plastic strain curve for the combined BCC and FCC model together with experimental data. Solid lines are the experimental data, dashed lines are for the parameters fitted for the entire temperature range, while dotted lines are for the narrower range of temperatures and are therefore only plotted in (c)-(f)

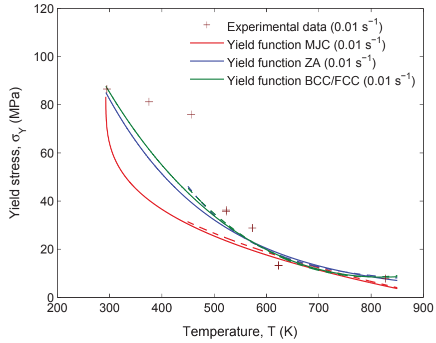
5.4 Comparison of Material Models and Discussion

In this Section, a comparison between the three calibrated material models and a short discussion of the results is presented. Fig. 5.10 and 5.11 show the yield stress vs. respectively temperature and strain rate for all models together.

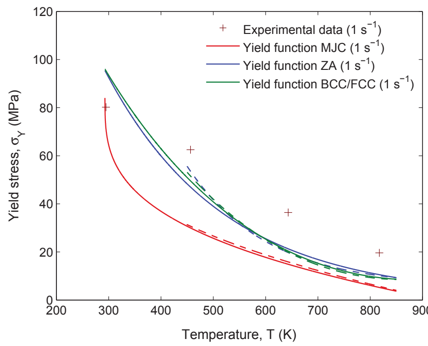
As discussed, the modified Johnson-Cook model does not catch the shape of the yield function when plotted vs. temperature. A much better fit is seen from both the modified Zerilli-Armstrong and combined BCC and FCC model. It is seen from Fig 5.10a and c that the yield stress sensitivity on the temperature is in fact varying within the temperature range considered. An increased sensitivity between approximately $450K$ and $600K$ is seen, and is most noticeable from tests with strain rate of $0.01s^{-1}$ and $300-400s^{-1}$. However, this inverted s-shape is not seen for all strain rate intervals, which may be resulting from both the execution of the experimental work and the post-processing of data, or may in fact be due to the different strain rates. Several tests, especially for strain rates between $500s^{-1}$ and $1000s^{-1}$, are needed to better determine the correct shape of the yield function. As the yield stress show stronger dependence on temperature than strain rate, it is not possible to determine the shape of the yield function when plotted vs. strain rate for temperature intervals of $100K$ or more. Even for tests carried out at room temperature only, see Fig. 5.11a, no correlation between yield stress and strain rate can be obtained. In order to so, several tests need to be carried out under very well-controlled conditions and accurate post-processing of the experimental data.

One important aspect that needs to be addressed regarding the calibration is that experimental data from both strain gauges and camera measurements are used. A better fit is believed to be achieved if the calibration was done for a complete temperature and strain rate range from camera measurements only, since several of the strain gauge measurements from the database are very limited in terms of the plastic strain range. The modified Johnson-Cook model is also a purely phenomenological model, and quite simple, as apposed to the Zerilli-Armstrong and combined BCC and FCC model which are semi-physical and more complex of nature. The latter model has been specifically designed for a metal showing behavior seen from both BCC and FCC metals, which also the AA6060 alloy does. Abed and Voyiadjis [4] have explained a much more thoroughly procedure for determination of the material model parameters for the combined BCC and FCC model than what has been done for this thesis, such that a better fit is believed to be achieved if the suggested procedure is used as apposed to the simple two step procedure shown here.

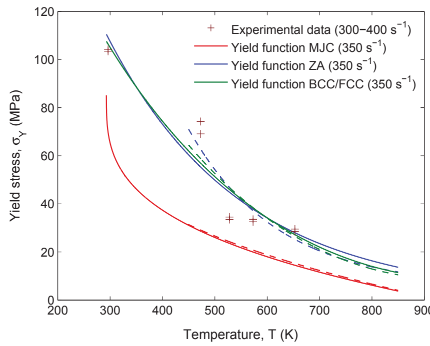
With the previous results and discussion in mind, it is without doubt that an adequate material model for a wide range of temperatures and strain rates can not be achieved easily for the AA6060 alloy. The material models chosen, the procedure for determining the material model parameters and the validity of the experimental data used for the fitting are all aspects that need to be taken carefully into account when establishing such a material model.



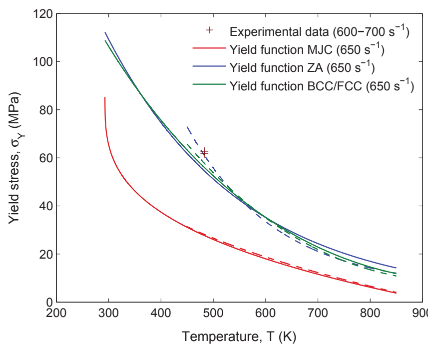
(a) $0.01s^{-1}$



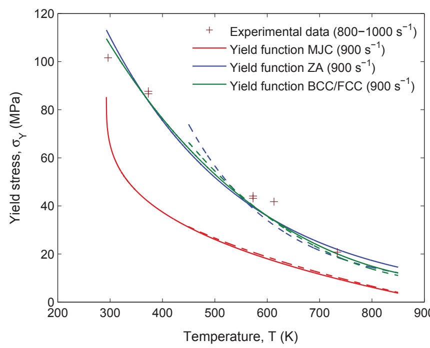
(b) $1s^{-1}$



(c) $300 - 400s^{-1}$

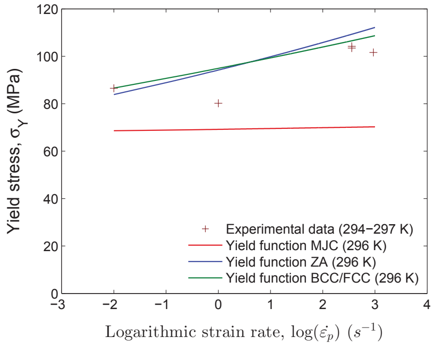


(d) $600 - 700s^{-1}$

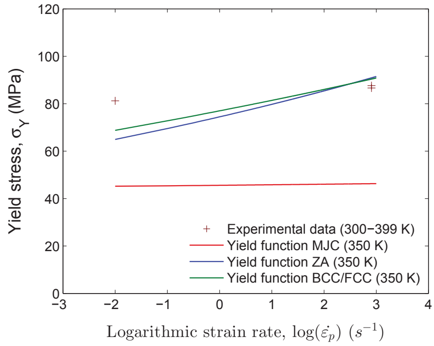


(e) $800 - 1000s^{-1}$

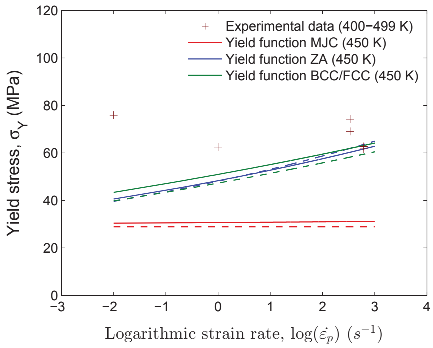
Figure 5.10: Plot (a)-(e) show the yield stress function vs. temperature for all models together with experimental data. The solid line is for the parameters fitted for the entire temperature range, while the dashed line is for the narrower range of temperature.



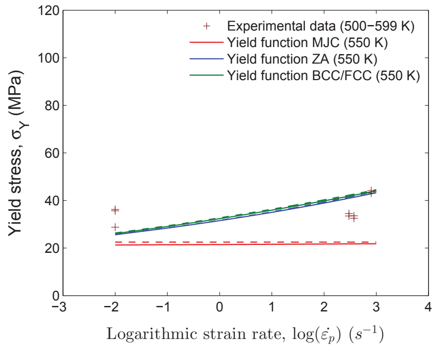
(a) 294-297 K



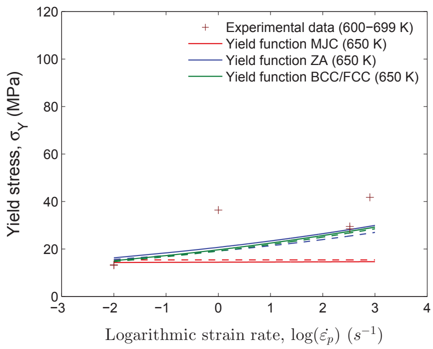
(b) 300-399 K



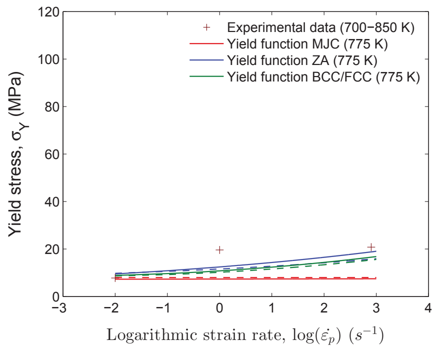
(c) 400-499 K



(d) 500-599 K



(e) 600-699 K



(f) 700-850 K

Figure 5.11: Plot (a)-(f) show the yield stress function vs. temperature for all models together with experimental data. The solid line is for the parameters fitted for the entire temperature range, while the dashed line is for the narrower range of temperature.

6 Numerical Analysis

6.1 Introduction

A finite element analysis of a given structure or component can provide highly accurate results and predictions for response and material behavior. Compared to real full-scale testing in a laboratory, numerical simulations using the finite element method is considerably cheaper and less time consuming, and the model can easily be adjusted to a variety of situations and conditions in short time. In order to predict accurate results, such numerical models do have to take into account a representative material model that will adequately describe the material behavior depending on temperature, strain rate or other variables. Such material models can be quite complex when non-linearities are introduced, thus all parts of the model must be investigated carefully to ensure that it will represent realistic and correct material behavior. As it would be too physical challenging to validate the simulations from a complex structure directly, numerical simulations of a material sample subjected to testing in a laboratory can be performed. In the case of simulating SHTB tests, simulations of the specimen only or simulations of the entire test setup can be performed. In this Section, numerical modelling of the SHTB tests described in Section 3.3 including both bars are presented. As only a total of four tests were successfully conducted, simulations have been performed for these tests only.

6.2 Finite Element Model of SHTB Setup

Finite element simulations of the SHTB tests have been performed with the non-linear finite element code LS-DYNA to evaluate the test setup and calibration of material models. To provide most realistic numerical simulations, the entire SHTB setup discussed in Section 3.3 has been modeled in real dimensions. By doing so, it is also possible to extract data in the same way as done in the experiments in the laboratory, such that these data can be compared to data extracted directly from the specimen for validation of the setup. A principal overview of the SHTB setup and the geometry of the test specimen were presented in Section 3.3 (respectively Fig. 3.2 and 3.3).

The geometry and mesh for the model has been generated in Abaqus CAE and imported to LS-DYNA by manually editing the element and node data generated by Abaqus CAE. The SHTB setup is modeled as an axis-symmetric volume weighted model with shell elements. An axis-symmetric area weighted shell formulation is not chosen as it is preferable for high explosive applications, while an axis-symmetric volume weighted shell formulation is best situated for structural applications [19]. This type of model is also a lot more cost-effective in terms of computational time compared to a 3D-model using solid elements, and has also been successfully adopted for similar simulations in other studies [37]. The nodes connecting the specimen with the bars, a total of 20 nodes that can be seen in Fig. 6.1, have been merged to best represent the contact condition between the specimen and bars. Fig. 6.1 and 6.2 show the finite element mesh for the test specimen respectively with and without part of bars.

Figure 6.1: Finite element mesh for test specimen and part of bars

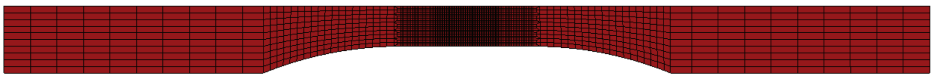


Figure 6.2: Finite element mesh for test specimen only

Reduced integration is chosen over full integration as it is considerably more cost-effective in terms of computational time and storage requirements due to the reduced number of integration points. However, reduced integration does come with some aspects that need to be carefully taken into account. When using this technique, it may produce what is called zero-energy deformation modes, such that for a deformation there are no straining at the integration points. This can result in a phenomenon called "hourglassing" that can lead to propagated deformations throughout the mesh that will provide inaccurate solutions. LS-DYNA can account for this by adding an artificial stiffness to the elements in case of these zero-energy deformation modes [34], and a Flanagan-Belytschko stiffness form is chosen as the hourglass control type. When using reduced integration, it is therefore important to check the contribution of the artificial strain energy to the total energy, and this shall not exceed approximately 10% [8].

The numerical simulations in LS-DYNA can be considered in two steps, which is the same as described in Section 3.3. In the first step, the top nodes at the end of the incident bar, ref. position A in Fig. 3.2, are stretched in tension until a desired displacement is reached. The desired displacement is reached after $0.1ms$ and then kept constant throughout the simulation as seen in Fig. 6.3a. At the same time as position A is stretched in tension, the nodes at position B are restrained against any longitudinal movement, thus creating a tension force in part A-B of the incident bar, while the rest of the setup remains stress free. In the second step, the restraining of the nodes at position B is then suddenly terminated after $0.3ms$, thus creating a tensile stress wave that will propagate towards position C and the specimen, see Fig. 6.3b. The first step is run as an implicit analysis, while the second step is run as an explicit analysis. The curves that can be seen in Fig. 6.3 (a)-(c) are implemented in LS-DYNA to respectively control the elongation at position A, the restraining of nodes at position B and to switch from an implicit to an explicit analysis. In Fig. 6.3c, the value 1 is associated with an implicit analysis, while the value 0 is associated with an explicit analysis. It should be noted that the curves are defined for a total time period of $5ms$, while the total simulation time is set to $t_{end} = 2.5ms$. The total simulation time needed to ensure fracture during the simulation will depend on the stretching of the incident bar and the temperature of the specimen, but a total simulation time of $2.5ms$ is adequate to ensure fracture in all simulations.

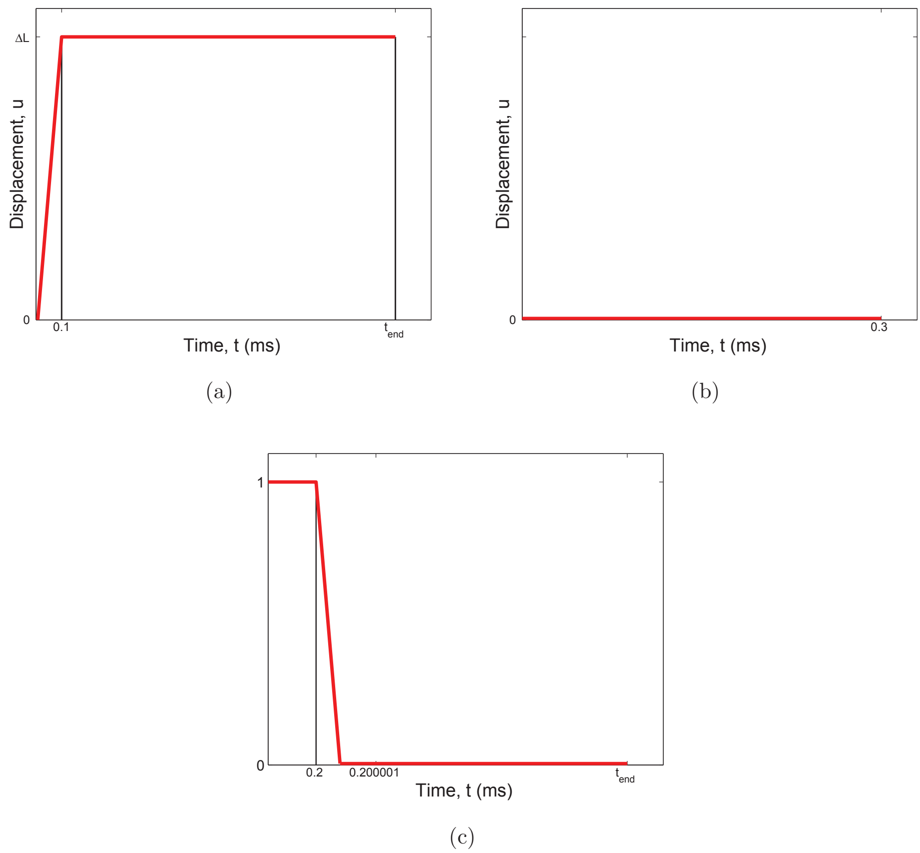


Figure 6.3: Curves implemented in LS-DYNA used for simulations of SHTB tests: (a) stretching of top nodes at position A, (b) clamping at position B and (c) implicit/explicit switch (ref. Fig. 3.2)

By assigning different values to the displacement of the nodes at position A, different strain rates can be achieved. The desired displacement ΔL can be estimated from [15]:

$$\Delta L = \varepsilon_{AB} L_{AB} = \frac{N}{E_b A_b} L_{AB} = \frac{N L_{AB}}{E_b A_b} \tag{6.1}$$

where ε_{AB} is the strain in bar A-B from the experiment, L_{AB} is the length of the incident bar A-B, E_b is the Young's modulus of the bar, A_b is the cross sectional area of the bar and N is the applied force at position A. The applied force N can be calculated from:

$$N = 2A_bE_b\varepsilon_{I,plateau} \tag{6.2}$$

where $\varepsilon_{I,plateau}$ is the incoming strain plateau in the incident bar measured from strain gauge 2 during the experiments. Inserting Eq. (6.2) into Eq. (6.1) yields an explicit expression for the desired displacement of nodes as function of the incoming strain plateau and the length of the incident bar from position A to B:

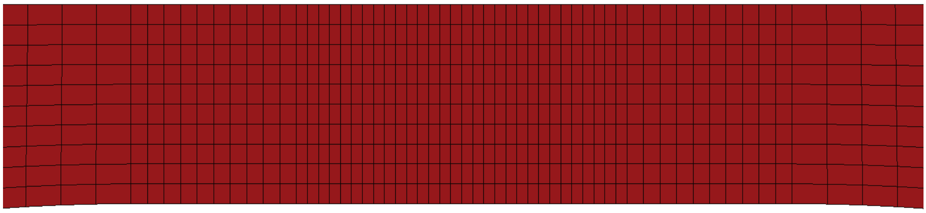
$$\Delta L = 2L_{ab}\varepsilon_{I,plateau} \tag{6.3}$$

Eq. (6.3) estimated the incoming strain wave with approximately 1% error for simulations compared to the experimental data, such that only a minor adjustment to the elongation was needed to get the correct value for the incoming strain wave.

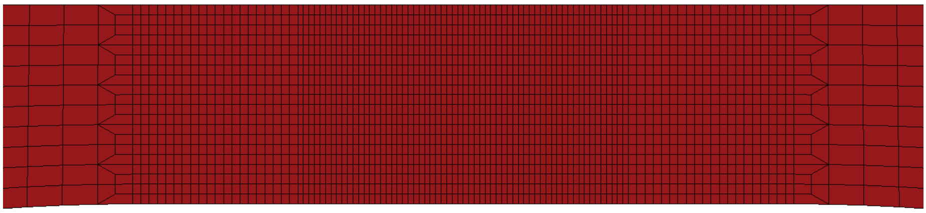
Results from initial simulations showed that the comparison of the local measurement taken directly from the specimen were almost coincident with the experimental tests until the maximum value of the true stress. However, some distinct deviations for the measured strain wave in the strain gauges were noticed. Therefore, a refined mesh was created for the gauge section of the specimen that was later used in all simulations. Table 6.1 and Fig. 6.4 summarizes and illustrates the initial and refined mesh.

	Initial mesh	**Refined mesh**
Specimen	Radial direction: 10 elements and element size between 0.15mm and 0.25mm. Element size vary between 0.08mm and 1.5mm along the longitudinal axis. 1100 elements in total.	Radial direction: 10 elements at shoulders, 20 elements in gauge section and element size between 0.08mm and 0.25mm. Element size vary between 0.04mm and 1.5mm along the longitudinal axis. 2640 elements in total.
Bars	Radial direction: 10 elements and element size of 0.5mm. Element size vary between 1mm and 15mm along the longitudinal axis. 16110 elements in total.	Unchanged
Total	17210 elements in total	18750 elements in total

Table 6.1: Overview of number of elements and element size for numerical model



(a)



(b)

Figure 6.4: Comparison of (a) initial and (b) refined mesh for gauge section of specimen

6.3 Material Model

It is of uttermost importance to establish a representative material model for the numerical simulations to get correct material behavior. There are more than 150 material models in the material library in LS-DYNA to choose from which are applicable for simulations for a large variety of experiments and situations [20]. Both the incident bar and transmission bar have been modeled with an elastic material model (MAT_001), using standard values of $E = 210000 MPa$, $\nu = 0.3$ and $\rho = 7850 \frac{kg}{m^3}$ for steel. The material model chosen for the specimen is the modified Johnson-Cook model (MAT_107) that also incorporates the Cockroft-Latham fracture criterion. The Johnson-Cook constitutive relation and the Cockroft-Latham fracture criterion are thoroughly presented in respectively Section 2.4.1 and 2.4.7. The parameters for the constitutive relation have been found using least-squares method (lsqnonlin function) in Matlab, and one fit for each material has been found. Table 6.2 summarizes the parameters implemented.

Alloy	*Test #*	A	Q_1	C_1	Q_2	C_2	C	m
AA6060-L	9, 11	48.03	45.31	3.49	27.35	52.27	2.22E-14	5.00
AA6060-H	2, 5	47.93	58.82	1.83	35.56	29.51	2.22E-14	5.00

Table 6.2: Parameters for the modified Johnson-Cook constitutive relation used in numerical simulations

A yield criterion is needed to define the transition between elastic and plastic straining, i.e. the yield surface represents the limitations of the elastic region in the stress space. A phenomenological yield function for isotropic materials proposed by several authors, e.g. Hershey and Hosford, has been adopted [15]:

$$\sigma_{eq} = \left\{ \frac{1}{2} \left(|\sigma_1 - \sigma_2|^m + |\sigma_2 - \sigma_3|^m + |\sigma_3 - \sigma_1|^m \right) \right\}^m \tag{6.4}$$

where σ_1, σ_2 and σ_3 are principal stresses and m is a material constant. In this study, a value of $m = 2$ is used and Eq. (6.4) is therefore reduced to the well known von Mises yield function. If, however, a value of $m \to \infty$ is assigned, Eq. (6.4) would reduce to the Tresca yield function [15].

Fracture and element erosion are initiated when one of the following criteria are fulfilled [20]:

1. Damage is greater than the critical value:

$$\widetilde{D} \geq D_C \tag{6.5}$$

2. Temperature is greater than the critical value:

$$T \geq T_C \tag{6.6}$$

The Cockroft-Latham damage evolution is defined as:

$$\dot{\widetilde{D}} = \frac{D_C}{W_C} max(\sigma_1, 0)\dot{\varepsilon}_p \tag{6.7}$$

where $D_C \leq 1$ is the critical damage, W_C is the critical fracture parameter defined in Eq. (2.42), σ_1 is the principal stress and $\dot{\varepsilon}_p$ is the plastic strain rate.

Nucleation and growth of voids are shortly discussed in Section 2.2.5 and will reduce the effective cross sectional area of a specimen due to the damage evolution, thus resulting in an effective damage-equivalent stress. Numerical simulations have been performed with and without damage coupling with the stress parameter. The damage-equivalent stress $\widetilde{\sigma}_{eq}$ implemented in the numerical model is defined by:

$$\widetilde{\sigma}_{eq} = \frac{1}{1 - \beta(\frac{W}{W_C})^D} \sigma_{eq} \tag{6.8}$$

where β is the coupling parameter, W is the plastic work and D is the damage coefficient. When $\beta = 0$, there are no damage coupling and $\widetilde{\sigma}_{eq} = \sigma_{eq}$.

As the specimen is subjected to large plastic strains and high strain rates, adiabatic heating conditions are also taken into account in the material model by the temperature rate $\dot{T}$ [20]:

$$\dot{T} = \chi \frac{\widetilde{\sigma}_{eq}\dot{\varepsilon}_p}{\rho C_p} \tag{6.9}$$

where χ is the Taylor-Quinney empirical parameter that defines the amount of energy due to plastic work that is converted to heat, ρ is the material density and C_p is the specific heat capacity. The temperature rate and the plastic strain rate can be expressed by respectively $\dot{T} = \Delta T/\Delta t$ and $\dot{\varepsilon}_p = \Delta\varepsilon_p/\Delta t$, such that an explicit expression for the actual temperature can be obtained:

$$T_{n+1} = T_n + \Delta T = T_n + \Delta t\dot{T} = T_n + \Delta t\chi \frac{\widetilde{\sigma}_{eq}\dot{\varepsilon}_p}{\rho C_p} = T_n + \chi \frac{\widetilde{\sigma}_{eq}\Delta\varepsilon_p}{\rho C_p} \tag{6.10}$$

All parameters implemented in the model can be found in Table 6.3. Young's modulus is assumed to be $57000MPa$ at $523K$ from Eq. (2.10). The Taylor-Quinney parameter χ is set to 0.9 as suggested by Børvik et al. [11]. However, Kapoor and Nemat-Nasser [29] have reported that close to 100% of the plastic work done during high strain rate deformation is converted to heat, thus the correct value of χ may be essentially set equal to one. The critical fracture parameter W_C is set equal to $260MPa$ which is the same value as adopted in similar studies for a similar alloy [37].

It should be noted that LS-DYNA did not take into account the initial temperature parameter T_0 in the material model, such that a user-defined material model was implemented to solve this issue.

Parameter	Unit	Value	Comment
E	MPa	57000	Young's modulus
ν		0.33	Poisson's ratio
ρ	kg/m^3	2700	Material density
T_m	K	933	Melting temperature
T_r	K	293	Room temperature
T_0	K	523	Initial temperature
χ		0.9	Taylor-Quinney parameter
C_p	J/KgK	9.6	Specific heat capacity
α	K^{-1}	0.0001	Thermal expansion coefficient
$\dot{p}_0$	s^{-1}	0.01	Reference strain rate
D_C		1	Critical damage parameter
W_C	J	260	Critical plastic work parameter
T_C	K	933	Critical temperature parameter

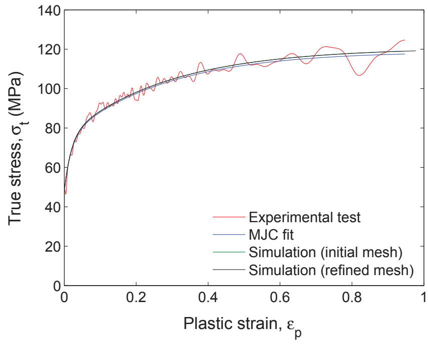
Table 6.3: Overview of parameters implemented in the modified Johnson-Cook material model

6.4 Results From Simulations

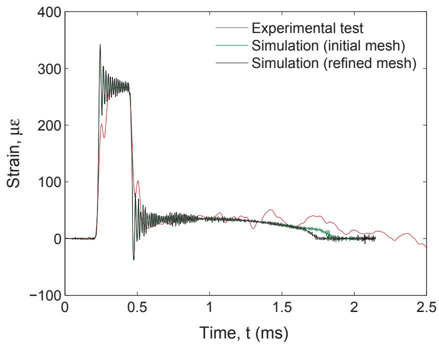
As mentioned in Section 6.2, results from initial simulations showed that there were some distinct deviations for the measured strain wave in the strain gauges. It was believed that the reason for this could be of two possible sources: too coarse mesh or the material model implemented. For large plastic straining and until fracture where the cross sectional area is significantly reduced, the number of elements in the necked section will be of uttermost importance to predict correct behavior. As the parameters for the constitutive relation have been fitted for a limited range of stress-strain values, the model can not be expected to be accurate within the whole range for strain values until fracture. A refined mesh was created to check whether the mesh might be the source of error, and the results are presented in Fig. 6.5 and are shown for test 9, ref. Table 4.2.

As can be seen, the true stress-plastic strain curve until the maximum value for plastic strain which the model was fitted for was not affected by the mesh refinement. However, it can be seen that the strain wave in strain gauge 2 and 3 was affected noticeably, but that the new mesh shows an even more distinct deviation from the experimental test. On the other hand, it is seen that the shape of the strain curve from strain gauge 3, see Fig. 6.5c, matches the experimental test better for the refined mesh. Thus it is believed that the primary source of error for this deviation is due to the parameters for the constitutive relation found in

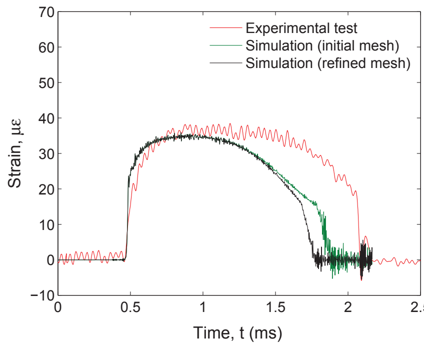
Table 6.2, taking into account the fact that a very similar numerical model has predicted very good results in similar studies [37].



(a)



(b)



(c)

Figure 6.5: Plots show comparison from simulation for initial and refined mesh for respectively (a) the true stress-plastic strain curve, (b) strain measurement from strain gauge 2 and (c) strain measurement from strain gauge 3

As mentioned in Section 6.2, the contribution from the introduced artificial energy to the total energy must be checked and shall not exceed approximately 10% [8]. As seen from Fig. 6.6, the artificial energy accounts for approximately 0.05% of the total energy, thus "hourglassing" is believed to not be of any concern.

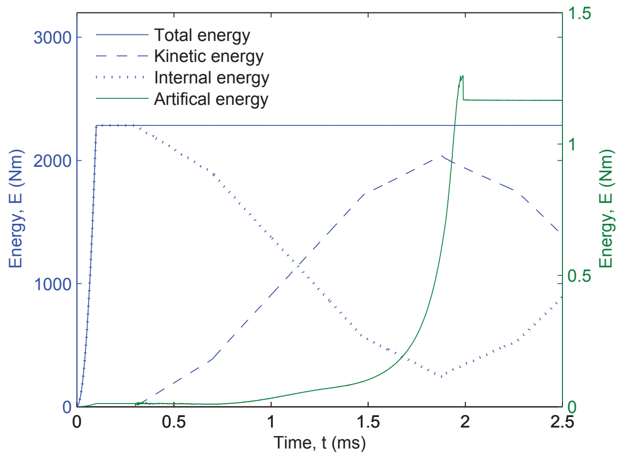


Figure 6.6: Energy plot from simulation showing the kinetic, internal, artificial and total energy

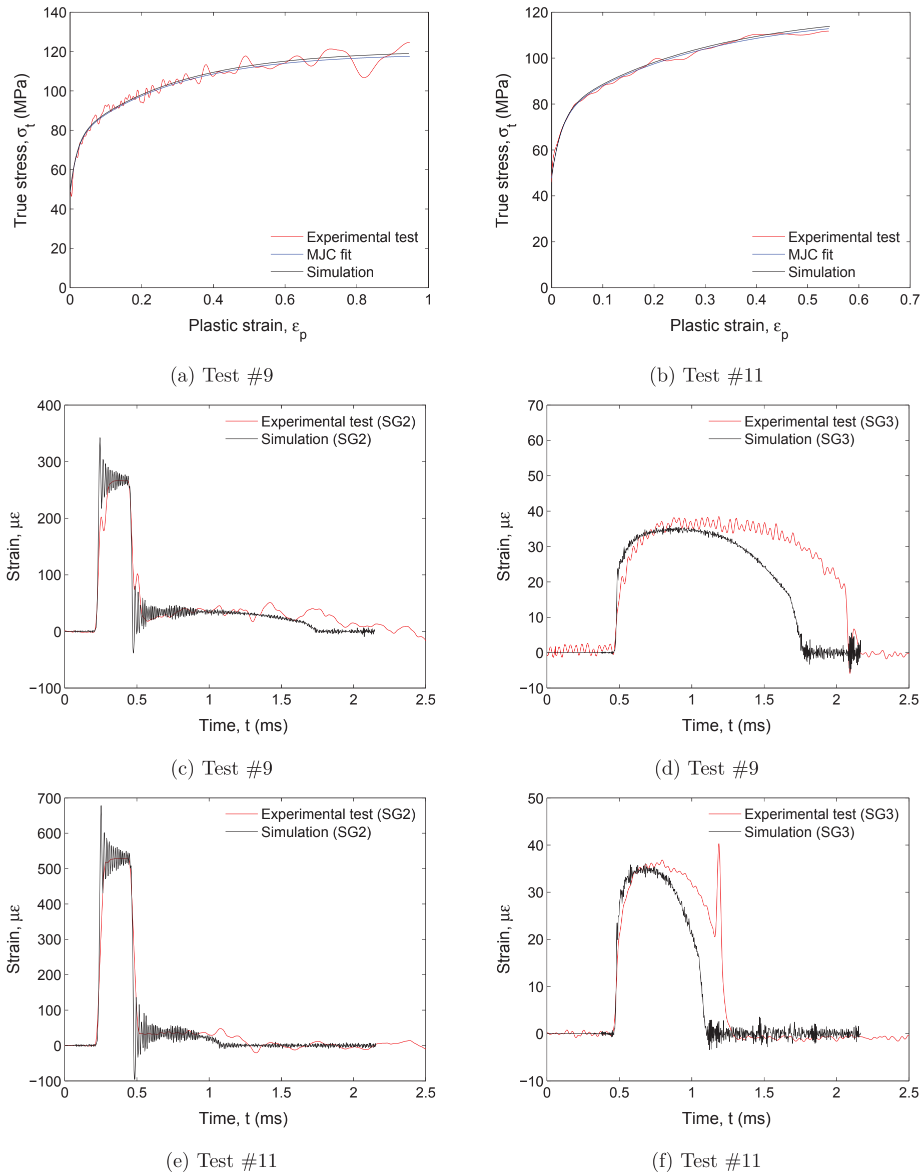
Simulations without damage coupling

Results from the simulations without damage coupling are shown in Fig. 6.7 and 6.8. The true stress-plastic strain curve has been plotted for the experimental data, modified Johnson-Cook constitutive relation fitted parameters and simulations. Strain gauge measurements from the experiments and simulations are also shown. As can be seen, the true stress-plastic strain curve from simulations is catching the correct behavior within the strain range the parameters for the constitutive relation are fitted for for all tests.

The strain gauge measurements from simulations show some distinct deviations from the experimental data for all tests, and the reason for this, as discussed, is believed to be the parameters implemented for the modified Johnson-Cock constitutive relation. Notice also that the incoming strain wave measured by strain gauge 2 is in fact very much coinciding with the experimental data when synchronized in terms of time, and deviations measured from strain gauge 2 and 3 are not noticeable until the incoming strain wave has reached the specimen.

It would be of interest to study how the results from simulations are affected when altering the parameters for the constitutive relation, e.g. adjusting the parameters to get increased strain hardening for large strain values, which may have explained the deviations seen from the simulations compared to the experimental tests. When studying the results from strain gauge 3 in Fig. 6.7 and 6.8, it is seen that the simulation of test 2 is by far most coinciding with the experimental tests. By looking at the true stress-strain curves it is also seen that the fitted parameters for this particular test ensures more strain hardening when compared to test

9 and 11, and it is also better fitted with the experimental data when compared to test 5. Significant fluctuations is seen from the experimental data for test 5 and may be the reason for a poorer fit of the material model, which, in turn, might be the reason for the bigger deviations seen relative to test 2 for the simulations. Unfortunately there was no time to investigate this further during the work for this thesis.

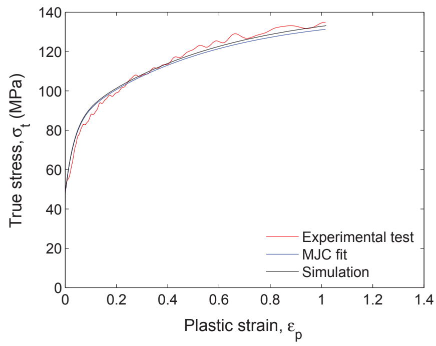


(a) Test #9 (b) Test #11

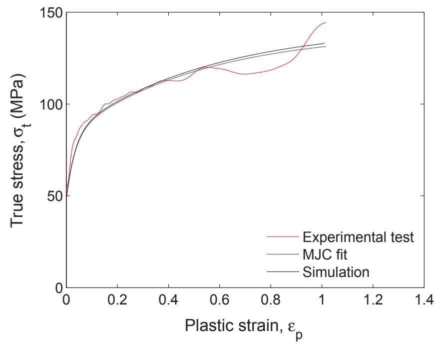
(c) Test #9 (d) Test #9

(e) Test #11 (f) Test #11

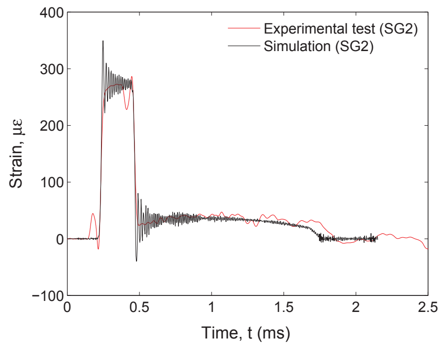
Figure 6.7: Plots (a)-(b) show the true stress-plastic strain curve from experiments and simulations, plots (c)-(f) show the strains from strain gauges from experiments and simulations



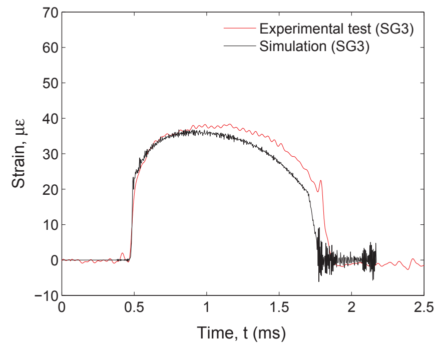
(a) Test #2



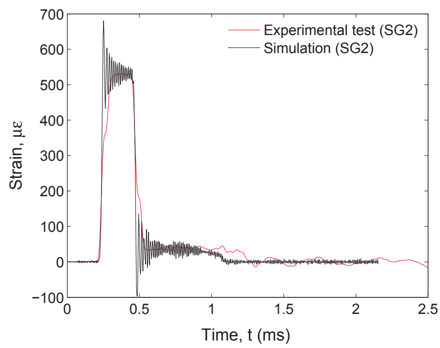
(b) Test #5



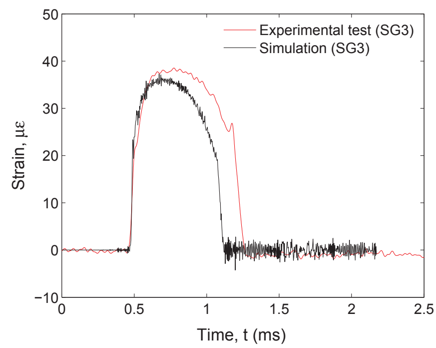
(c) Test #2



(d) Test #2



(e) Test #5



(f) Test #5

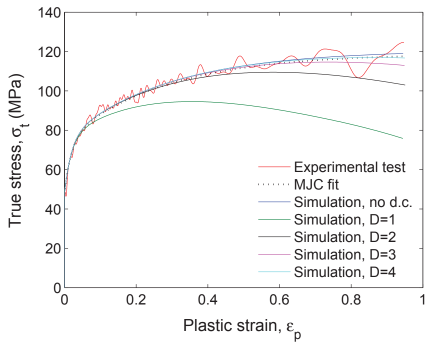
Figure 6.8: Plots (a)-(b) show the true stress-plastic strain curve from experiments and simulations, plots (c)-(f) show the strains from strain gauges from experiments and simulations

Simulations with damage coupling

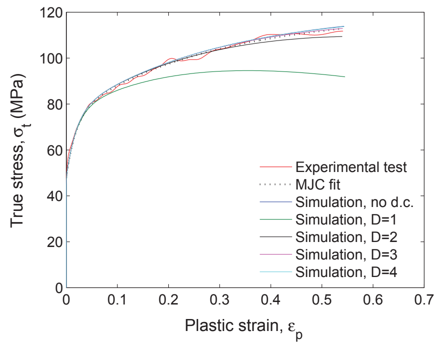
Simulations with damage coupling have also been performed. Eq. (6.8) has been implemented in the material model to account for the development of nucleation, growth and coalescence of voids in the necked section. However, as the strain gauge measurements without damage coupling in fact predicts fracture too early compared to experimental results, the introduction of damage coupling will not improve the results. It will, in fact, predict fracture and reduction of force even earlier. The simulations have been run only to study how the implementation of the coupled damage equation affects the results.

Fig. 6.9 and 6.10 show the results from the simulations with the damage coefficient D set equal to $1, 2, 3$ and 4 together with the experimental results and simulations with no damage coupling. It is seen that simulations with damage coupling and the damage coefficient set equal to 4 seem to predict best results for the true stress-plastic strain curve for test 9 and 11, while too much damage is predicted for test 2 and 5 for the same value. A value between 5 and 6 is believed to predict better results for the latter tests. It is clearly seen that that a value of 1 predicts rather inaccurate results for all simulations from the true stress-plastic strain curves.

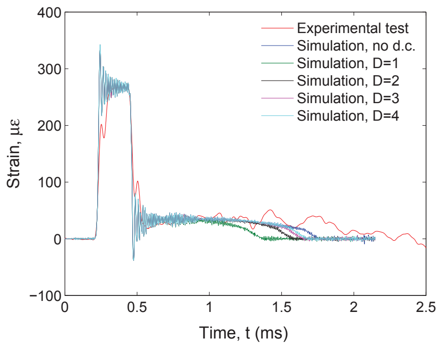
It is seen from the strain gauge measurements that fracture is predicted significant earlier when damage coupling is introduced. If a better prediction was obtained from the simulations, the damage coefficient D could be calibrated rather easily to predict fracture at the correct time to fit with the experimental results.



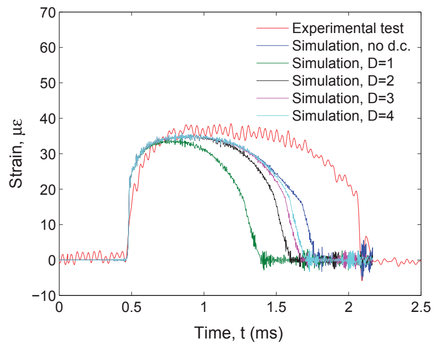
(a) Test #9



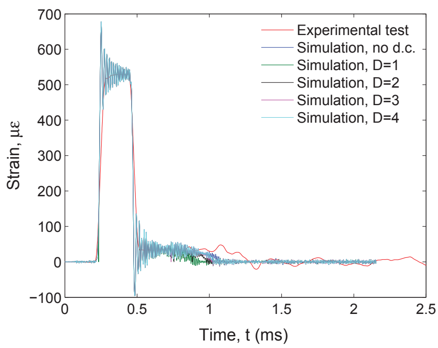
(b) Test #11



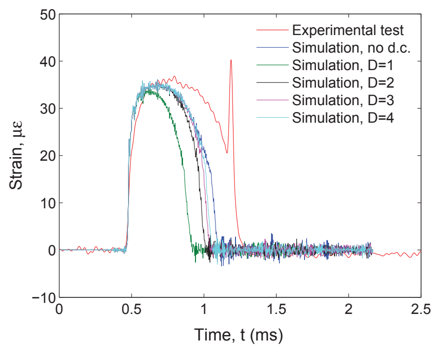
(c) Test #9



(d) Test #9

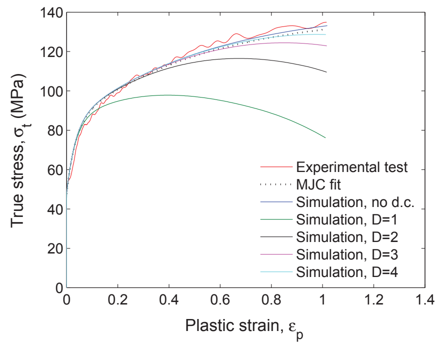


(e) Test #11

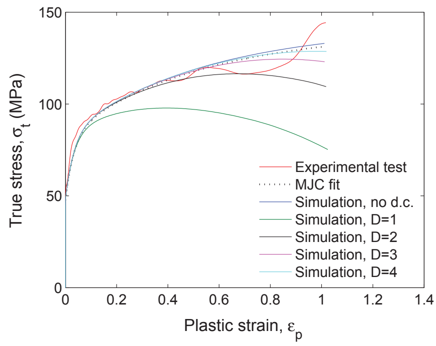


(f) Test #11

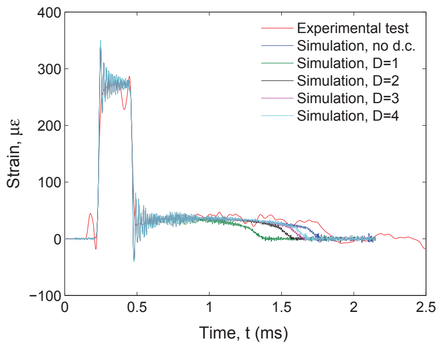
Figure 6.9: Plots (a)-(b) show the true stress-plastic strain curve from experiments and simulations with and without damage coupling, plots (c)-(f) show the strains from strain gauges from experiments and simulations with and without damage coupling



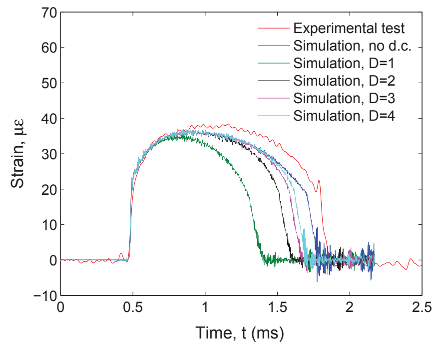
(a) Test #2



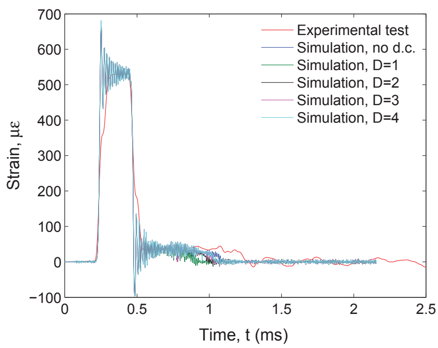
(b) Test #5



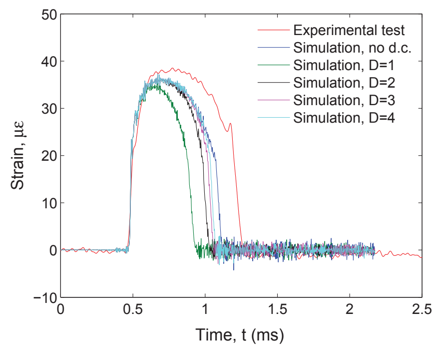
(c) Test #2



(d) Test #2



(e) Test #5



(f) Test #5

Figure 6.10: Plots (a)-(b) show the true stress-plastic strain curve from experiments and simulations with and without damage coupling, plots (c)-(f) show the strains from strain gauges from experiments and simulations with and without damage coupling

7 Concluding Remarks

The work for this thesis can be divided into four parts, corresponding to Section 3-6, and will each be summarized in this Section.

Experimental work: The execution of experimental tests at both quasi-static strain rates and at dynamic strain rates in a split-Hopkinson tension bar test setup has been a major part of the work for this thesis. Several observations have been acquired during the work:

- The induction heater system coupled with a coil is a simple, yet effective device to increase the temperature in a test specimen. However, there were significant difficulties associated with keeping a constant temperature during the entire test for strain rates of $0.01s^{-1}$. The resulting true stress-plastic strain curve is directly affected by this, and it is especially noticeable for temperatures of $573K$ and higher, such that an improved control system for keeping a constant temperature during the whole test is believed to improve the validity of the results.
- A pyrometer was used to measure the temperature during the tests and proved to be an effective solution. The accuracy of the measured temperature has not been validated within the scope of this thesis.
- Local measurement of the geometry of the necked section was obtained using a high-speed camera, such that the response of the test specimen could be calculated beyond the onset of necking. However, the post-processing of the camera recordings is a time-consuming procedure and the validity of the results obtained has not been investigated. The stress values after onset of necking have been corrected using Bridgman's formula, taking into account the geometry of the necked section from camera recordings. There are uncertainties associated with the validity of the measured geometry that may lead to either conservative or non-conservative values of the corrected stress.

Experimental results: Experimental data have been post-processed from both quasi-static tests and SHTB tests, and results for true stress-plastic strain curve, yield stress and estimated fracture strain have been presented.

- It is found that both the yield stress and strain hardening for all alloys are very much dependent on temperature, but no noticeable dependence on the strain rate can be found.
- Some tests show slightly different material behavior between the alloys studied. However, the results do not differ significantly from each other, and the deviations are not seen from all tests, such that no distinctive difference with respect to material behavior can be established.

Calibration of material models: Three material models have been fitted to the available database containing material data for a wide range of strain rates and temperatures.

- The modified Johnson-Cook constitutive relation predicts in general too low yield stress for tests within the complete range of temperatures and strain rates. The fit for the strain hardening is far from adequate.
- The modified Zerilli-Armstrong model predicts much better results for the yield stress, and the fit for the strain hardening is also improved.
- The combined BCC and FCC material model predicts both yield stress and strain hardening very similar to the Zerilli-Armstrong model.
- The material model parameters were calibrated for two temperature ranges of $293K - 850K$ and $450K - 850K$, but the narrower temperature range did not provide considerably better fit.
- It seems that the investigated models cannot be calibrated easily to an adequately fit for a wide range of temperatures and strain rates with the procedure used. A better fit could possible be found from camera measurements for the entire range of temperatures and strain rates. However, the main reason for the poor fit is believed to be that the material models studied are too simple to predict correct stress within such a wide range of temperatures and strain rates.

Numerical analysis: Numerical simulations of the SHTB experiments have not been a major part of the work for this thesis. Still, some interesting results were observed:

- The geometry of the neck, especially for ductile fracture and for large strain values, is rather complex and require a high mesh density to be represented adequately. Two meshes were created for the gauge section of the specimen, and the refined mesh proved to predict different material behavior, especially for large strain values and until fracture. The mesh density is believed to be of uttermost importance for SHTB test simulations and in particular for models incorporating fracture.
- The material model parameters implemented in LS-DYNA are also believed to be of crucial importance to predict correct material behavior. The results from the simulations did not coincide well with the experimental tests, but unfortunately there was no time to investigate this further. It is believed that the main reason for the deviations seen is the predicted strain hardening from the material model.
- Simulations with and without damage coupling was run. Results from the simulations with damage coupling show that fracture is predicted earlier, and the shape of the strain wave measured by strain gauge 3 seems to be

more coinciding with the experimental tests. However, it was not possible to calibrate the damage coupling due to the large deviations seen between simulations and experiments.

8 Further Work

There are a lot of uncertainties associated with conducting experimental tests at both elevated strain rates and temperatures. Many potential sources of error may influence on the validity of the results obtained from such experiments and has not been investigated thoroughly for this thesis. Heating of test specimens, the temperature measuring, local measurement from camera recordings and the correction of stress values after onset of necking may all be sources of error that will lead to non-valid results obtained. A study on the validity of the results obtained from such experiments would be of great interest. In particular, it might be worthwhile to have a closer look to the part of the test setup which is related to temperature. Such an investigation could involve the accuracy of the temperature measurement and the homogeneity of the temperature field in the test specimen.

The main scope for this thesis has been to conduct experimental tests for a wide range of temperatures and strain rates for the AA6060-L and AA6060-H alloy and was unfortunately not obtained due to delayed manufacturing of test specimens. A natural suggestion for further work will be to continue the work which was started during this thesis. A complete database for quasi-static loading conditions and for temperatures ranging from $293K$ to $633K$ exists now, but several experiments in the split-Hopkinson tension bar is needed to also include a complete range of strain rates.

Numerical modeling of the SHTB experiments was done for this thesis, but not a major part of the work was devoted to this. Good results were obtained for the strain hardening until the maximum value of strain the material model was fitted for, but fracture was predicted too early. The believed reasons for this is discussed in Section 6.4. Further work on this part would be of great interest to identify the problems associated with the numerical simulations.

References

[1] The aluminum association. URL `http://www.aluminum.org`. Accessed: 4.2.2013.

[2] Crystalline structure subject guide | information resources | asm international. URL `http://www.asminternational.org/portal/site/www/SubjectGuideItem/?vgnextoid=ad7cdc8cc359d210VgnVCM100000621e010aRCRD`. Accessed: 14.03.2013.

[3] Solid state structure. URL `http://www.ndt-ed.org/EducationResources/CommunityCollege/Materials/Structure/solidstate.htm`. Accessed: 12.04.2013.

[4] F. H. Abed and G. Z. Voyiadjis. Plastic deformation modeling of al-6xn stainless steel at low and high strain rates and temperatures using a combination of bcc and fcc mechanisms of metals. *International Journal of Plasticity*, 21 (8):1618–1639, 2005.

[5] C. Albertini and M. Montagnani. Dynamic material properties of several steels for fast breeder reactor safety analysis. *EUR 5787, Commission of the European Communities, Boite Postale 1003, Luxembourg.*

[6] D. Altenpohl. *Aluminum Viewed from Within.* Aluminum-Verlag, 1982.

[7] Hydro Aluminium. Extrusion Ingot. Technical Datasheet. AlMgSi alloy 606035, 2001.

[8] M. E. Andersen. Personal communication, 2012.

[9] T. L. Anderson. *Fracture Mechanics. Fundamentals and Applications.* CRC Press. Taylor and Francis Group, third edition, 2005.

[10] M. F. Ashby and D. R. H. Jones. *Engineering Materials 1 - An Introduction to Properties, Applications, and Design.* Elsevier, 2012.

[11] T. Børvik, O.S. Hopperstad, T. Berstad, and M. Langseth. A computational model of viscoplasticity and ductile damage for impact and penetration. *European Journal of Mechanics - A/Solids*, 20(5):685–712, 2001.

[12] P. W. Bridgman. *Studies in Large Plastic Flow and Fracture.* Harvard University Press, Cambridge, Massachusetts, 1964.

[13] R. Brown. *Lecture Notes CHE333 Materials Engineering.* The University of Rhode Island, Kingston, USA, 2012.

[14] W. Chen and B. Song. *Split Hopkinson (Kolsky) Bar.* Springer, 2011.

[15] Y. Chen. *Modelling of Dynamic Material Behaviour and Fracture of Aluminium Alloys for Structural Applications.* Tapir Uttrykk, Trondheim, Norway, 2009.

[16] Y. Chen, A. H. Clausen, O. S. Hopperstad, and M. Langseth. Application of a split-hopkinson tension bar in a mutual assessment of experimental tests and numerical predictions. *International Journal of Impact Engineering*, 38 (10):824–836, 2011.

[17] A. H. Clausen and T. Auestad. Split-hopkinson bar, experimental set-up and theoretical considerations. Technical Report R-16-02, NTNU, 2002.

[18] A. H. Clausen, T. Børvik, O. S. Hopperstad, and A. Benallal. Flow and fracture characteristics of aluminium alloy aa5083-h116 as function of strain rate, temperature and triaxiality. *Materials Science and Engineering: A*, 364: 260–272, 2004.

[19] Livermore Software Technology Corporation. *LS-DYNA Keyword User's Manual Volume I*. 2012.

[20] Livermore Software Technology Corporation. *LS-DYNA Keyword User's Manual Volume II Material Models*. 2012.

[21] J. R. Davis. *Aluminum and Aluminum Alloys*. ASM International, 1993.

[22] J.E. Field, S.M. Walley, W.G. Proud, H.T. Goldrein, and C.R. Siviour. Review of experimental techniques for high rate deformation and shock studies. *International Journal of Impact Engineering*, 30(7):725–775, 2004.

[23] Z. L. Greer. Temperature, frequency, and young's modulus of an aluminum tuning fork. *ISB Journal of Physics*, 2011.

[24] Y. Hammi and M.F. Horstemeyer. A physically motivated anisotropic tensorial representation of damage with separate functions for void nucleation, growth, and coalescence. *International Journal of Plasticity*, 23(10-11):1641–1678, 2007.

[25] O.S. Hopperstad and T. Børvik. *Lecture Notes TKT4135 Mechanics of Materials*. The Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2012.

[26] J. Johnsen and J. K. Holmen. *Effects of Heat Treatment on the Ballistic Properties of AA6070 Aluminium Plates*. Master's Thesis, The Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2012.

[27] G. R. Johnson and W. H. Cook. A constitutive model and data for metals subjected to large strains, high strain rates and high temperatures. In *Proceedings of the 7th International Symposium on Ballistics*, volume 21, pages 541–547. The Hague, Netherlands: International Ballistics Committee, 1983.

[28] J. Kajberg and K.-G. Sundin. Material characterisation using high-temperature split hopkinson pressure bar. *Journal of Materials Processing Technology*, 213(4):522–531, 2013.

[29] R. Kapoor and S. Nemat-Nasser. Determination of temperature rise during high strain rate deformation. *Mechanics of Materials*, 27(1):1 – 12, 1998.

[30] J. G. Kaufman. *Introduction to Aluminum Alloys and Tempers.* ASM International, 2000.

[31] K. M. Mathisen. *Lecture Notes TKT4197 Nonlinear Finite Element Analysis.* The Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2012.

[32] P. J. Rae, C. Trujillo, and M. Lovato. The young's modulus of 1018 steel and 67061-t6 aluminum measured from quasi-static to elastic precursor strain-rates. *Conference: APS Meeting on the Shock Compression of Condensed Matter; June 28, 2009; Nashville, TN*, 2009.

[33] G. Le. Roy, J.D. Embury, G. Edwards, and M.F. Ashby. A model of ductile fracture based on the nucleation and growth of voids.

[34] Simulia. *Abaqus Theory Manual.* Dassault Systémes, 2011.

[35] J. D. Verhoeven. *Fundamentals of Physical Metallurgy.* John Wiley and Sons, 1975.

[36] V. Vilamosa. Personal communication, 2013.

[37] V. Vilamosa, A. H. Clausen, E. Fagerholt, O.S. Hopperstad, and T. Børvik. Extended measurement of stress-strain behavior of ductile materials at elevated temperatures in a split-hopkinson tension bar system. 2013.

[38] G. Z. Voyiadjis and F. H. Abed. Microstructural based models for bcc and fcc metals with temperature and strain rate dependency. *Mechanics of Materials*, 37(2-3):355–378, 2005.

[39] F. J. Zerilli and R. W. Armstrong. Dislocation-mechanics-based constitutive relations for material dynamics calculations. *Journal of Applied Physics*, 61 (5):1816–1825, 1987.

A Historical Overview of SHTB Test Setups

In crash situations, automotive parts may be subjected to local strain rates of order $10^2 s^{-1}$ to $10^3 s^{-1}$. It is necessary to have knowledge about the mechanical properties of the materials when subjected to such load cases to be able to design and analyze these structures. Most servo-hydraulic test machines cannot impose strain rates higher than $1 s^{-1}$, and it is therefore necessary to apply different techniques to obtain such loading conditions and elevated strain rates. This Appendix briefly summarizes the historical development of test setups for high strain rate experiments and is taken from the book by Chen and Song [14].

Field et al. [22] reviewed several techniques for elevated strain rate experiments, such as use of dropweights, the split-Hopkinson bar and the Taylor impact test. The split-Hopkinson bar can be used for both tension, compression, torsion and combined torsion and axial loading, and this seems to be the most adopted technique for obtaining crash relevant strain rates [14].

The first versions of split-Hopkinson bars for tension loading emerged in the 1960's. Harding et al. (1960) designed a test setup where the input bar was made of a hollow tube with the test specimen assembled inside the tube. The specimen was then stressed in tension by use of a mechanical joint that transferred the compression pulse into a tension pulse. A modified version was designed by Harding and Welsh (1983) that was very similar to a design by Hauser (1966). All these test designs transferred the external impact into axial tension loading, such that loading devices from compression bar systems could be used directly. The most evident weakness of such systems is that the entire tension setup is inside a solid tube, thus making it difficult to mount instrumentation devices and visual observation is very limited [14].

Another approach suggested by Lindholm and Yeakley (1968) was to mount a "top hat" specimen between the incident bar and a hollow transmission tube. The gauge section of the test specimen was loaded in tension when the compression stress wave in the incident bar strikes the inside of the specimen geometry [14].

Nicholas (1981) proposed a design where the initial compression stress wave was reflected back as a tensile wave after traveling to the free end of the transmission bar and thus propagating back towards the specimen [14].

However, loading by direct tension is the most commonly used method. Direct tension can be acquired in two different ways. One method is to store elastic energy in the incident bar by stretching in tension, and thus releasing a tensile stress wave when the elastic energy is abruptly released. The other method is to generate kinetic energy to strike a flange at the end of the incident bar.

B One-Dimensional Elastic Wave Theory

It is beneficial to have a minimum insight into longitudinal stress wave theory to understand the derivations for the calculation of the response in the specimen during a SHTB test. This Appendix serves this purpose.

Elastic wave propagation is a three-dimensional problem. However, the bars in a typical SHTB test rig have a small diameter-to-length ratio such that all waves but the longitudinal waves can be neglected [17]. Lateral inertia effects are also neglected due to the same reasons. It is further assumed that both bars have elastic material behavior and constant cross sectional area. With these assumptions taken into account, the differential equation of the one-dimensional wave problem is stated as:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \tag{B.1}$$

where u is the longitudinal displacement, x is the longitudinal coordinate along the bar, t is time and c is the wave propagation velocity defined by:

$$c = \sqrt{\frac{E}{\rho}} \tag{B.2}$$

where E is Young's modulus and ρ is the material density. Eq. (B.1) is a partial differential equation that has solution on the form:

$$u(x,t) = f(x - ct) + g(x + ct) \tag{B.3}$$

It can easily be verified that Eq. (B.3) satisfies Eq. (B.1) by substitution. The strain in the bar is found by:

$$\varepsilon(x,t) = \frac{\partial u}{\partial x} = f'(x - ct) + g'(x + ct) \tag{B.4}$$

By the definition of Hooke's law, the stress in the bar can be found as:

$$\sigma(x,t) = E\varepsilon(x,t) = E(f'(x - ct) + g'(x + ct)) \tag{B.5}$$

From Eqs. (B.3), (B.4) and (B.5) it is evident that both the displacement, strain and stress state, in addition to the particle velocity, defined as:

$$v(x,t) = \frac{\partial u}{\partial t} = c(-f'(x-ct) + g'(x+ct)) \tag{B.6}$$

will move along the bar. It can be shown that $f(x-ct)$ and $g(x+ct)$ are functions that represents a wave moving respectively in positive and negative x-direction with respect to time. For a wave traveling in positive x-direction, Eq. (B.5) will reduce to:

$$\sigma(x,t) = Ef'(x-ct) = -\frac{E}{c}v(x,t) = -\rho c v(x,t) \tag{B.7}$$

by substituting Eqs. (B.2) and (B.6) into Eq. (B.5). It is seen from Eq. (B.7) that the particle velocity is negative for a stress wave moving in the positive x-direction. Referring to Section 3.3, it is now shown that for the incident bar stretched in tension, the wave will propagate towards the specimen while at the same time particles will be moving in the opposite direction, thus the specimen will be subjected to a tension load.

C Calculation of Response in Test Specimen from SHTB Tests

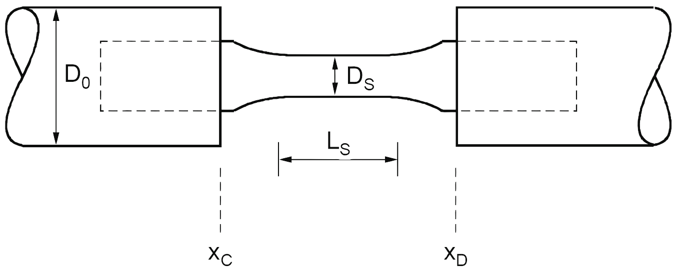


Figure C.1: Transition between bars and specimen [17]

Referring to Fig. C.1 and the definition of the incident, reflected and transmitted strain from Section 3.2, in addition to the derivations in Appendix B, the strain at position C can be defined as:

$$\varepsilon(x_c, t) = f'(x_C - c_0 t) + g'(x_C + c_0 t) = \varepsilon_I + \varepsilon_R \tag{C.1}$$

since f' represents the incoming strain (ε_I) and g' represents the reflected strain (ε_R). From Eqs. (B.6) and (C.1), the particle velocity at position C can be calculated:

$$v(x_C, t) = c_0(-f'(x_C - c_0 t) + g'(x_C + c_0 t)) = c_0(-\varepsilon_I + \varepsilon_R) \tag{C.2}$$

The displacement at position C cannot be calculated analytically, but by numerical integration of the velocity the displacement is found:

$$u(x_c, t) = \int_0^t v(x_C, \tau) d\tau = c_0 \int_0^t (-\varepsilon_I + \varepsilon_R) d\tau \tag{C.3}$$

When the strain in the bar at position C is known, the corresponding force can be found:

$$F(x_C, t) = E_0 A_0 \varepsilon(x_C, t) = E_0 A_0(\varepsilon_I + \varepsilon_R) \tag{C.4}$$

The particle velocity, displacement and force at position D can be calculated in the same way as for position C. It is assumed that the specimen will fracture before any waves will be reflected at the end of the transmission bar, thus strain gauge 3 will only measure the incoming transmitted stress wave ε_T. Using the same considerations as for position C, and setting $\varepsilon_I = \varepsilon_T$ and $\varepsilon_R = 0$, the particle velocity, displacement and force at position D are found:

$$v(x_D, t) = -c_0 \varepsilon_T \tag{C.5}$$

$$u(x_D, t) = -c_0 \int_0^t \varepsilon_T d\tau \tag{C.6}$$

$$F(x_D, t) = E_0 A_0 \varepsilon_T \tag{C.7}$$

Neglecting any inertia forces, equilibrium of the specimen yields that the force at position C must equal the force at position D. Utilizing Eqs. (C.4) and (C.7) yields:

$$\varepsilon_I + \varepsilon_R = \varepsilon_T \tag{C.8}$$

This equilibrium relation can be used to ensure the accuracy of the measured signals from a SHTB test. The stress in the specimen is found by dividing the force at position D (or at position C) by the cross sectional area of the specimen:

$$\sigma_s = \frac{F_s}{A_s} = \frac{F(x_D, t)}{A_s} = \frac{E_0 A_0}{A_s} \varepsilon_T \tag{C.9}$$

By assuming that all strains in the specimen take place in the gauge section (discussed in Section 4.1), and using the relation in Eq. (C.8), the strain in the specimen can be calculated:

$$\varepsilon_s = \frac{u(x_D, t) - u(x_c, t)}{L_s} = \frac{c_0}{L_s} \int_0^t (-\varepsilon_T - (-\varepsilon_I + \varepsilon_R)) d\tau = -2 \frac{c_0}{L_s} \int_0^t \varepsilon_R d\tau \tag{C.10}$$

The corresponding strain rate is simply the time derivative of the strain state:

$$\dot{\varepsilon_s} = \frac{\partial \varepsilon_s}{\partial \tau} = -2\frac{c_0}{L_s}\varepsilon_R \tag{C.11}$$

D Matlab Scripts

D.1 Post-Processing of Data from Quasi-Static Tests

Script name	Script/function	Explanation
import_new_single.m	Script	Post-processing of exp. data from one test only
import_new_plot.m	Script	Post-processing of exp. data from two or three tests together for comparison
import_new_double.m	Script	Post-processing of exp. data from two or three tests
engs_func_single.m	Function	Calculation of approx. engineering stress around the neck for one test only
engs_func_plot.m	Function	Calculation of approx. engineering stress around the neck for two or three tests
voce_test_single.m	Function	Calculation of the resulting approx. true stress parameters by Voce hardening parameters for one test only
voce_test_plot.m	Function	Calculation of the resulting approx. true stress parameters by Voce hardening parameters for two or three tests
voce_test_double.m	Function	Calculation of the resulting approx. true stress parameters by Voce hardening parameters for the average curve

Table D.1: Overview of Matlab scripts for post-processing of data from quasi-static tests

import_new_single.m:

```
%% Post-processing of experimental data (for 1 data set only)
% Input from files needed: force and displacement
clearvars -except parameters results
global eng_strain_calc eng_stress_calc plastic_strain ...
    true_plastic_stress

%% Manual input
% REMEMBER TO CLEAR PARAMETERS AND RESULTS VARIABLES BEFORE NEW SERIES
test='Filename';
test_n=1;
test_tot=10;
temp=293;
path='path\exp_data';
path_save='path\save';
cd(path)
gauge_length=5;
d=3;
A0=pi*(d/2)^2;
```

```

%% Importing data
f1=[path test '.txt'];
fid = fopen(f1);
fseek(fid, 0, 'eof');
endpos = ftell(fid);
fseek(fid, 0, 'bof');
z=1;

while  ftell(fid) ~= endpos;
    tline = fgetl(fid);
    nn=size(tline);
    if z>2
        data = sscanf(tline,'%f%f%f');
        state(z-2,1)=data(1,1);
        state(z-2,2)=data(2,1);
        state(z-2,3)=data(3,1);
    end
    z=z+1;
end

fclose(fid);
time(:,1)=state(:,1);
force(:,1)=state(:,2)-state(1,2);
displacement(:,1)=state(:,3)-state(1,3);

%% Defining relevant data interval (beginning to end of test)
figure
plot(displacement,force)
title('Select data range to be used')
xlabel('Displacement (mm)')
ylabel('Force (kN)')
legend('Experimental data (raw)','Location','NorthEast')

pause on
[x,y]=ginput(2);
x5=x(1); x6=x(2);
close

pos1=find(min(abs(displacement-x5))==abs(displacement-x5),1);
pos2=find(min(abs(displacement-x6))==abs(displacement-x6),1);
eng_strain=displacement(pos1:pos2)/gauge_length;
eng_stress=force(pos1:pos2)*1000/A0;

%% Smoothing of eng. stress-strain for data interval (using 5 points)
eng_strain_smooth(1)=sum(eng_strain(1:2))/2;
eng_strain_smooth(2)=sum(eng_strain(1:3))/3;
eng_strain_smooth(length(eng_strain))=sum(eng_strain(end-1:end))/2;
eng_strain_smooth(length(eng_strain)-1)=sum(eng_strain(end-2:end))/3;
for i = 3:length(eng_strain)-2
    eng_strain_smooth(i)=sum(eng_strain(i-2:i+2))/5;
end
eng_strain=eng_strain_smooth';

eng_stress_smooth(1)=sum(eng_stress(1:2))/2;
eng_stress_smooth(2)=sum(eng_stress(1:3))/3;
eng_stress_smooth(length(eng_stress))=sum(eng_stress(end-1:end))/2;
eng_stress_smooth(length(eng_stress)-1)=sum(eng_stress(end-2:end))/3;
for i = 3:length(eng_stress)-2
```

```
    eng_stress_smooth(i)=sum(eng_stress(i-2:i+2))/5;
end
eng_stress=eng_stress_smooth';

%% Determing the elastic tangent modulus
figure
plot(eng_strain,eng_stress)
xlabel('Engineering strain')
ylabel('Engineering stress (MPa)')
title('Select data range for determining E-modulus + yield stress')
legend('Experimental data','Location','NorthEast')
axis([-0.1 max(eng_strain)*1.1 -10 max(eng_stress)*1.1])

pause on
[x,y]=ginput(2);
x5=x(1); x6=x(2);
close

pos1=find(min(abs(eng_strain-x5))==abs(eng_strain-x5),1);
pos2=find(min(abs(eng_strain-x6))==abs(eng_strain-x6),1);
pos1_r=pos1;
pos2_r=pos2;

figure
plot(eng_strain(pos1:pos2),eng_stress(pos1:pos2))
xlabel('Engineering strain')
ylabel('Engineering stress (MPa)')
title('Select data range for E-modulus')
legend('Experimental data','Location','NorthEast')
axis([0 eng_strain(pos2) 0 max(eng_stress(pos1:pos2))])

pause on
[x,y]=ginput(2);
x5=x(1); x6=x(2);

pos1=find(min(abs(eng_strain-x5))==abs(eng_strain-x5),1);
pos2=find(min(abs(eng_strain-x6))==abs(eng_strain-x6),1);
E_x=eng_strain(pos1:pos2);
E_y=eng_stress(pos1:pos2);
P=polyfit(E_x,E_y,1);
E_calc_x_min=min(eng_strain);
E_calc_x_max=eng_strain(find(eng_stress==max(eng_stress),1));
E_calc_x=linspace(E_calc_x_min,E_calc_x_max,100);
E_calc_y=P(2)+P(1).*E_calc_x;
E_calc_x_end=find(min(abs(E_calc_y-max(eng_stress)))==...
    abs(E_calc_y-max(eng_stress)));
E_calc_x_start=find(abs(E_calc_y)==min(abs(E_calc_y)));
E_meas=(E_calc_y(end)-E_calc_y(1))/(E_calc_x(end)-E_calc_x(1));
hold on
plot(E_calc_x(E_calc_x_start:E_calc_x_end),E_calc_y...
    (E_calc_x_start:E_calc_x_end),'r')
legend('Experimental data','Elastic tangent ...
    modulus','Location','NorthEast')

%% Translating start of straining to origo
eps_var=eng_strain(pos1)-eng_stress(pos1)/E_meas;
eng_strain_corr(1)=0;
eng_strain_corr(2)=eng_stress(pos1)/E_meas;
eng_strain_corr(3:length(eng_strain(pos1:end))+2)=...
```

```
    eng_strain(pos1:end)-eps_var;
eng_stress_corr(1)=0;
eng_stress_corr(2)=eng_stress(pos1);
eng_stress_corr(3:length(eng_stress(pos1:end))+2)=eng_stress(pos1:end);
eng_strain=eng_strain_corr;
eng_stress=eng_stress_corr;

%% Correction of Young's modulus
E_corr=(-3.9*exp(0.0033*temp)+79)*1000;
eng_strain_corr=eng_strain-eng_stress.*((E_corr-E_meas)/...
    (E_corr*E_meas));

%% Determing the yield point
pos2_r_new=pos2_r-pos1+2;
figure
hold on
plot(eng_strain(1:pos2_r_new),eng_stress(1:pos2_r_new),'b')
plot(E_calc_x(E_calc_x_start:E_calc_x_end)-eps_var,E_calc_y...
    (E_calc_x_start:E_calc_x_end),'r')
plot(eng_strain_corr(1:pos2_r_new),eng_stress(1:pos2_r_new),...
    'Color',[0 0.5 0])
title('Determine the yield point (for the uncorrected curve)')
xlabel('Engineering strain')
ylabel('Engineering stress (MPa)')
legend('Uncorrected strain values','Elastic tangent ...
    modulus','Corrected strain values','Location','SouthEast')

pause on
[x,y]=ginput(1);
x5=x(1);
close all

ypos=find(min(abs(eng_strain-x5))==abs(eng_strain-x5),1);
eng_strain=eng_strain_corr;
true_strain=log(1+eng_strain);
true_stress=eng_stress.*(1+eng_strain);

%% Curve fitting of engineering stress-strain curve (for ...
    determination of necking point)
figure
plot(eng_strain,eng_stress)
xlabel('Engineering strain')
ylabel('Engineering stress (MPa)')
title('Define data range for defining necking point')
legend('Experimental data','Location','NorthEast')

pause on
[x,y]=ginput(2);
x5=x(1); x6=x(2);
close

pos1=find(min(abs(eng_strain-x5))==abs(eng_strain-x5),1);
pos2=find(min(abs(eng_strain-x6))==abs(eng_strain-x6),1);
eng_strain_calc=eng_strain(pos1:pos2);
eng_stress_calc=eng_stress(pos1:pos2);

A0=5; B0=5; C0=5; D0=5;
eng_strain_0=[A0 B0 C0 D0];
lowerb=[-1000 -1000 -1000 -1000];
```

```
upperb=[1000 1000 1000 1000];

options = optimset('TolFun', 1e-1000, 'TolX', 1e-1000, ...
    'MaxFunEvals',100000,'MaxIter',100000, 'PlotFcns', ...
    @optimplotresnorm);
[eng_strain_cal(1:4),eng_strain_cal(5)] = ...
    lsqnonlin(@engs_func_single,eng_strain_0,lowerb,upperb,options);
close

eng_strain_lin=linspace(min(eng_strain_calc),max(eng_strain_calc),100);
eng_stress_calc_new=eng_strain_cal(1)+eng_strain_cal(2).*...
    eng_strain_lin+eng_strain_cal(3).*eng_strain_lin.^2+...
    eng_strain_cal(4).*eng_strain_lin.^3;
np=find(eng_stress_calc_new==max(eng_stress_calc_new),1);
np_strain=eng_strain_lin(np);
np_x_h=eng_strain_lin;
np_y_h=ones(length(eng_strain_lin))*max(eng_stress_calc_new);
np_x_v=[eng_strain_lin(np) eng_strain_lin(np)];
np_y_v=[min(eng_stress_calc_new) ...
    min(eng_stress_calc_new)+(max(eng_stress_calc_new)-...
    min(eng_stress_calc_new))*2];

figure
plot(eng_strain_calc,eng_stress_calc,eng_strain_lin,...
    eng_stress_calc_new,np_x_h,np_y_h,'r--',np_x_v,np_y_v,'r--')
legend('Experimental data', 'Calculated curve','Necking point')
xlabel('Engineering strain')
ylabel('Engineering stress (MPa)')

%% Engineering stress-strain from start --> onset of necking
np_eng=find(min(abs(eng_strain-np_strain))==abs(eng_strain-...
    np_strain),1); %position of necking point in eng_strain vector
eng_strain_np=eng_strain(1:np_eng);
eng_stress_np=eng_stress(1:np_eng);

%% True stress-strain from start --> onset of necking
np_strain_true=log(1+np_strain);
np_true=find(min(abs(true_strain-np_strain_true))==...
    abs(true_strain-np_strain_true),1);
true_strain_np=true_strain(1:np_true);
true_stress_np=true_stress(1:np_true);

%% Plotting of eng. stress-strain and true stress-strain until ...
    onset of necking
figure
plot(eng_strain_np,eng_stress_np,true_strain_np,true_stress_np)
legend('Eng. stress-strain until necking', 'True stress-strain ...
    until necking', 'Location', 'SouthEast')

%% True stress-plastic strain from yield to necking
plastic_strain=true_strain(ypos:np_true)-true_strain(ypos);
true_plastic_stress=true_stress(ypos:np_true);

%% Calculating Voce-rule parameters
sigY_0=50; Q1_0=50; C1_0=10; Q2_0=20; C2_0=40;
voce_0=[sigY_0 Q1_0 C1_0 Q2_0 C2_0];
lowerb=[-1000 -1000 -1000 -1000 -1000 -1000];
upperb=[1000 1000 1000 1000 1000 1000];

```

```
[voce(1:5),voce(6)] = ...
    lsqnonlin(@voce_test_single,voce_0,lowerb,upperb,options);
close all

parameters(1:test_tot,1)=1:test_tot;
parameters(test_n,2)=true_stress(ypos);
parameters(test_n,3:7)=voce(1:5);
parameters(test_n,8)=max(plastic_strain);

%% Plotting the resulting true stress-plastic strain curve from ...
    Voce parameters
plastic_strain_new=linspace(0,max(plastic_strain),100);
voce_stress=voce(1)+voce(2).*(1-exp(-voce(3).*plastic_strain_new))+...
    voce(4).*(1-exp(-voce(5).*plastic_strain_new));
hFig=figure;
hAxes=axes;
hold on
plot(plastic_strain_new,voce_stress,'b','LineWidth',2)
plot(plastic_strain,true_plastic_stress,'r','LineWidth',1)
legend('Adaption to model', 'Experimental ...
    test','Location','SouthEast')
xlabel('Plastic strain, \epsilon_p')
ylabel('True stress, \sigma_t (MPa)')
axis([0 ceil(max(plastic_strain_new)*1.1/0.01)*0.01 0 ...
    ceil(max(voce_stress)*1.1/10)*10])

%% Saving last figure to directory
cd(path_save)
filename=sprintf('Test-0%g-(true_stress).eps',test_n);
save_figure_small(filename,hFig,hAxes)

%% Plotting the corrected eng. stress-strain curve
hFig=figure;
hAxes=axes;
plot(eng_strain,eng_stress,'r','LineWidth',2)
legend('Experimental test','Location','NorthEast')
xlabel('Engineering strain, \epsilon_e')
ylabel('Engineering stress, \sigma_e (MPa)')
axis([0 ceil(max(eng_strain)*1.1/0.1)*0.1 0 ...
    ceil(max(eng_stress)*1.1/10)*10])

%% Saving last figure to directory
filename=sprintf('Test-0%g-(eng_stress).eps',test_n);
save_figure_small(filename,hFig,hAxes)

%% Saving stress-strains in "results" array
results{test_n}(1:length(eng_strain),1)=eng_strain;
results{test_n}(1:length(eng_stress),2)=eng_stress;
results{test_n}(1:length(true_strain_np),3)=true_strain_np;
results{test_n}(1:length(true_stress_np),4)=true_stress_np;
results{test_n}(1:length(plastic_strain),5)=plastic_strain;
results{test_n}(1:length(true_plastic_stress),6)=true_plastic_stress;
results{test_n}(1:length(plastic_strain_new),7)=plastic_strain_new;
results{test_n}(1:length(voce_stress),8)=voce_stress;

%% Saving of calculations to .xlsx and .mat files
if test_n(end)==test_tot
    parameters_xlsx_string=arrayfun(@num2str, parameters, 'unif', 0);

```

```
    parameters_xlsx = {'Parameters for yield stress + Voce ...
        hardening and max plastic strain at onset of necking', '', ...
        '', '','','','',''; ...
        'Test number' '\sigma_Y (directly)', '\sigma_Y', 'Q_1', ...
            'C_1', 'Q_2', 'C_2', 'Max. plastic strain'};
    parameters_xlsx(3:size(parameters,1)+2,1:8)=parameters_xlsx_string;
    xlswrite('parameters.xlsx', parameters_xlsx);

    save('parameters', 'parameters')
    save('results', 'results')
end
cd('C:\Users\Eivind\Documents\MATLAB\Post-process QS')
```

import_new_plot.m:

```
%% Post-processing of experimental data (for 2 or 3 data sets)
% Input from files needed: force and displacement
clearvars -except parameters results
global eng_strain_calc_plot eng_stress_calc_plot ...
    plastic_strain_plot true_plastic_stress_plot i

%% Manual input
% REMEMBER TO CLEAR PARAMETERS AND RESULTS VARIABLES BEFORE NEW SERIES
test(1,:)=strcat('Filename1');
test(2,:)=strcat('Filename2');
test(3,:)=strcat('Filename3');
test_n=[1 2 3];
test_tot=10;
temp=293;
path='path\exp_data';
path_save='path\save\';
gauge_length=5;
d=3;
A0=pi*(d/2)^2;

%% Importing data
for i = 1:length(test_n)
    clear state
    f1=[path test(i,:) '.txt'];
    fid = fopen(f1);
    fseek(fid, 0, 'eof');
    endpos = ftell(fid);
    fseek(fid, 0, 'bof');
    z=1;

    while  ftell(fid) ~= endpos;
        tline = fgetl(fid);
        nn=size(tline);
        if z>2
            data = sscanf(tline,'%f%f%f');
            state(z-2,1)=data(1,1);
            state(z-2,2)=data(2,1);
            state(z-2,3)=data(3,1);
        end
        z=z+1;
    end
```

```

    fclose(fid);
    time{i}=state(:,1);
    force{i}=state(:,2)-state(1,2);
    displacement{i}=state(:,3)-state(1,3);
end

%% Defining relevant data interval (beginning to end of test)
for i = 1:length(test_n)
    hFig=figure;
    hAxes=axes;
    plot(displacement{i},force{i})
    str = sprintf('Select data range to be used (test ...
        #%g)',test_n(i));
    title(str);
    xlabel('Displacement, u (mm)')
    ylabel('Force, F (kN)')
    legend('Experimental data','Location','NorthEast')

    pause on
    [x,y]=ginput(2);
    x5(i)=x(1); x6(i)=x(2);
    close

    pos1(i)=find(min(abs(displacement{i}-x5(i)))==...
        abs(displacement{i}-x5(i)),1);
    pos2(i)=find(min(abs(displacement{i}-x6(i)))==...
        abs(displacement{i}-x6(i)),1);
    eng_strain{i}=displacement{i}(pos1(i):pos2(i))/gauge_length;
    eng_stress{i}=force{i}(pos1(i):pos2(i))*1000/A0;
end

%% Smoothing of eng stress-strain for data interval (using 5 points)
for i = 1:length(test_n)
    eng_strain_smooth{i}(1)=sum(eng_strain{i}(1:2))/2;
    eng_strain_smooth{i}(2)=sum(eng_strain{i}(1:3))/3;
    eng_strain_smooth{i}(length(eng_strain{i}))=...
        sum(eng_strain{i}(end-1:end))/2;
    eng_strain_smooth{i}(length(eng_strain{i})-1)=...
        sum(eng_strain{i}(end-2:end))/3;
    for j = 3:length(eng_strain{i})-2
        eng_strain_smooth{i}(j)=sum(eng_strain{i}(j-2:j+2))/5;
    end
    eng_strain{i}=eng_strain_smooth{i}';

    eng_stress_smooth{i}(1)=sum(eng_stress{i}(1:2))/2;
    eng_stress_smooth{i}(2)=sum(eng_stress{i}(1:3))/3;
    eng_stress_smooth{i}(length(eng_stress{i}))=...
        sum(eng_stress{i}(end-1:end))/2;
    eng_stress_smooth{i}(length(eng_stress{i})-1)=...
    sum(eng_stress{i}(end-2:end))/3;
    for j = 3:length(eng_stress{i})-2
        eng_stress_smooth{i}(j)=sum(eng_stress{i}(j-2:j+2))/5;
    end
    eng_stress{i}=eng_stress_smooth{i}';
end

%% Determing the elastic tangent modulus
for i = 1:length(test_n)
```

```
    figure
    plot(eng_strain{i},eng_stress{i})
    xlabel('Engineering strain')
    ylabel('Engineering stress (MPa)')
    title('Select data range for determining E-modulus + yield ...
        stress')
    legend('Experimental data','Location','NorthEast')
    axis([-0.1 max(eng_strain{i})*1.1 -10 max(eng_stress{i})*1.1])

    pause on
    [x,y]=ginput(2);
    x5=x(1); x6=x(2);
    close

    pos1_int(i)=find(min(abs(eng_strain{i}-x5))==...
        abs(eng_strain{i}-x5),1);
    pos2_int(i)=find(min(abs(eng_strain{i}-x6))==...
        abs(eng_strain{i}-x6),1);
end

for i = 1:length(test_n)
    figure
    plot(eng_strain{i}(pos1_int(i):pos2_int(i)),...
        eng_stress{i}(pos1_int(i):pos2_int(i)))
    xlabel('Engineering strain')
    ylabel('Engineering stress (MPa)')
    str = sprintf('Select data range for E-modulus (test ...
        #%g)',test_n(i));
    title(str);
    legend('Experimental data','Location','NorthEast')
    axis([eng_strain{i}(pos1_int(i)) eng_strain{i}(pos2_int(i)) 0 ...
        max(eng_stress{i}(pos1_int(i):pos2_int(i)))])

    pause on
    [x,y]=ginput(2);
    x5(i)=x(1); x6(i)=x(2);

    pos1=find(min(abs(eng_strain{i}-x5(i)))==...
        abs(eng_strain{i}-x5(i)),1);
    pos2=find(min(abs(eng_strain{i}-x6(i)))==...
        abs(eng_strain{i}-x6(i)),1);
    E_x{i}=eng_strain{i}(pos1:pos2);
    E_y{i}=eng_stress{i}(pos1:pos2);
    P(:,i)=polyfit(E_x{i},E_y{i},1);
    E_calc_x_min(i)=min(eng_strain{i});
    E_calc_x_max(i)=eng_strain{i}(find(eng_stress{i}==...
        max(eng_stress{i}),1));
    E_calc_x{i}=linspace(E_calc_x_min(i),E_calc_x_max(i),100);
    E_calc_y{i}=P(2,i)+P(1,i).*E_calc_x{i};
    E_calc_x_end(i)=find(min(abs(E_calc_y{i}-max(eng_stress{i})))...
        ==abs(E_calc_y{i}-max(eng_stress{i})));
    E_calc_x_start(i)=find(abs(E_calc_y{i})==min(abs(E_calc_y{i})));
    E_meas(i)=(E_calc_y{i}(end)-E_calc_y{i}(1))/(E_calc_x{i}(end)-...
        E_calc_x{i}(1));
    hold on
    plot(E_calc_x{i}(E_calc_x_start(i):E_calc_x_end(i)),E_calc_y{i}...
        (E_calc_x_start(i):E_calc_x_end(i)),'r')
    legend('Experimental data','Elastic tangent ...
        modulus','Location','NorthEast')
```

```

    %% Translating start of straining to origo
    eps_var(i)=eng_strain{i}(pos1)−eng_stress{i}(pos1)/E_meas(i);
    eng_strain_corr{i}(1)=0;
    eng_strain_corr{i}(2)=eng_stress{i}(pos1)/E_meas(i);
    eng_strain_corr{i}(3:length(eng_strain{i}(pos1:end))+2)=...
        eng_strain{i}(pos1:end)−eps_var(i);
    eng_stress_corr{i}(1)=0;
    eng_stress_corr{i}(2)=eng_stress{i}(pos1);
    eng_stress_corr{i}(3:length(eng_stress{i}(pos1:end))+2)=...
        eng_stress{i}(pos1:end);
    eng_strain{i}=eng_strain_corr{i};
    eng_stress{i}=eng_stress_corr{i};

    %% Correction of Young's modulus
    E_corr=(−3.9*exp(0.0033*temp)+79)*1000;
    eng_strain_corr{i}=eng_strain{i}−eng_stress{i}.*...
        ((E_corr−E_meas(i))/(E_corr*E_meas(i)));

    %% Determing the yield point
    pos2_int_new(i)=pos2_int(i)−pos1+2;
    figure
    hold on
    plot(eng_strain{i}(1:pos2_int_new(i)),eng_stress{i}...
        (1:pos2_int_new(i)))
    plot(eng_strain_corr{i}(1:pos2_int_new(i)),eng_stress{i}...
        (1:pos2_int_new(i)),'Color',[0 0.5 0])
    plot(E_calc_x{i}(E_calc_x_start(i):E_calc_x_end(i))−eps_var(i),...
        E_calc_y{i}(E_calc_x_start(i):E_calc_x_end(i)),'r')
    str = sprintf('Define yield point (for the uncorrected curve) ...
        (test #%g)',test_n(i));
    title(str);
    xlabel('Engineering strain, \epsilon_e')
    ylabel('Engineering stress, \sigma_e (MPa)')
    legend('Uncorrected strain values','Corrected strain ...
        values','Elastic tangent modulus','Location','SouthEast')

    pause on
    [x,y]=ginput(1); x5(i)=x(1);
    close all

    ypos(i)=find(min(abs(eng_strain{i}−x5(i)))==...
        abs(eng_strain{i}−x5(i)),1);
    eng_strain{i}=eng_strain_corr{i};
    true_strain{i}=log(1+eng_strain{i});
    true_stress{i}=eng_stress{i}.*(1+eng_strain{i});
end

%% Curve fitting of engineering stress−strain curve (for ...
    determination of necking point)
clear eng_strain_calc
clear eng_stress_calc
clear eng_calc
for i = 1:length(test_n)
    figure
    plot(eng_strain{i},eng_stress{i},'LineWidth',1.5)
    xlabel('Engineering strain, \epsilon_e')
    ylabel('Engineering stress, \sigma_e (MPa)')
```

```
    str = sprintf('Define data range for defining necking point ...
        (test #%g)',test_n(i));
    title(str);
    legend('Experimental test','Location','NorthEast')

    pause on
    [x,y]=ginput(2);
    x5(i)=x(1); x6(i)=x(2);
    close

    pos1(i)=find(min(abs(eng_strain{i}-x5(i)))==...
        abs(eng_strain{i}-x5(i)),1);
    pos2(i)=find(min(abs(eng_strain{i}-x6(i)))==...
        abs(eng_strain{i}-x6(i)),1);
    eng_strain_calc_plot{i}=eng_strain{i}(pos1(i):pos2(i));
    eng_stress_calc_plot{i}=eng_stress{i}(pos1(i):pos2(i));

    A0=5; B0=5; C0=5; D0=5;
    eng_strain_plot_0=[A0 B0 C0 D0];
    lowerb=[-1000 -1000 -1000 -1000];
    upperb=[1000 1000 1000 1000];

    options = optimset('TolFun', 1e-1000, 'TolX', 1e-1000, ...
        'MaxFunEvals',100000,'MaxIter',100000, 'PlotFcns', ...
        @optimplotresnorm);
    [eng_strain_cal_plot(1:4,i),eng_strain_cal_plot(5,i)] = ...
        lsqnonlin(@engs_func_plot,eng_strain_plot_0,lowerb,upperb,...
        options);
    close

    eng_strain_lin{i}=linspace(min(eng_strain_calc_plot{i}),...
        max(eng_strain_calc_plot{i}),100);
    eng_stress_calc_new{i}=eng_strain_cal_plot(1,i)+...
        eng_strain_cal_plot(2,i).*eng_strain_lin{i}+...
        eng_strain_cal_plot(3,i).*eng_strain_lin{i}.^2+...
        eng_strain_cal_plot(4,i).*eng_strain_lin{i}.^3;
    np=find(eng_stress_calc_new{i}==max(eng_stress_calc_new{i}),1);
    np_strain(i)=eng_strain_lin{i}(np);
    np_x_h=eng_strain_lin{i};
    np_y_h=ones(length(eng_strain_lin{i}))*max(eng_stress_calc_new{i});
    np_x_v=[eng_strain_lin{i}(np) eng_strain_lin{i}(np)];
    np_y_v=[min(eng_stress_calc_new{i}) ...
        min(eng_stress_calc_new{i})+(max(eng_stress_calc_new{i})-...
        min(eng_stress_calc_new{i}))*2];

    figure
    hold on
    plot(eng_strain_calc_plot{i},eng_stress_calc_plot{i},'LineWidth',1)
    plot(eng_strain_lin{i},eng_stress_calc_new{i},'Color',[0 0.5 ...
        0],'LineWidth',1)
    plot(np_x_h,np_y_h,'r--','LineWidth',1.5)
    plot(np_x_v,np_y_v,'r--','LineWidth',1.5)
    legend('Experimental test', 'Approx. polynomial', 'Necking point')
    xlabel('Engineering strain, \epsilon_e')
    ylabel('Engineering stress, \sigma_e (MPa)')
end

%% Engineering stress-strain from start --> onset of necking
for i = 1:length(test_n)
```

```
    np_eng(i)=find(min(abs(eng_strain{i}−np_strain(i)))==...
        abs(eng_strain{i}−np_strain(i)),1); %position of necking ...
            point in eng_strain vector
    eng_strain_np{i}=eng_strain{i}(1:np_eng(i));
    eng_stress_np{i}=eng_stress{i}(1:np_eng(i));

    %% True stress−strain from start −−> onset of necking
    np_strain_true(i)=log(1+np_strain(i));
    np_true(i)=find(min(abs(true_strain{i}−np_strain_true(i)))==...
        abs(true_strain{i}−np_strain_true(i)),1);
    true_strain_np{i}=true_strain{i}(1:np_true(i));
    true_stress_np{i}=true_stress{i}(1:np_true(i));

    %% Plotting of eng. stress−strain and true stress−strain until ...
        onset of necking
    figure
    plot(eng_strain_np{i},eng_stress_np{i},true_strain_np{i},...
        true_stress_np{i})
    legend('Eng. stress−strain until necking', 'True stress−strain ...
        until necking', 'Location', 'SouthEast')
    str = sprintf('Stress−strain curves (test #%g)',test_n(i));
    title(str);

    %% True stress−plastic strain from yield to necking
    plastic_strain_plot{i}=true_strain{i}(ypos(i):np_true(i))−...
        true_strain{i}(ypos(i));
    true_plastic_stress_plot{i}=true_stress{i}(ypos(i):np_true(i));
end

%% Calculating Voce−rule parameters
sigY_0=50; Q1_0=50; C1_0=10; Q2_0=20; C2_0=40;

voce_plot_0=[sigY_0 Q1_0 C1_0 Q2_0 C2_0];
lowerb=[−1000 −1000 −1000 −1000 −1000 −1000];
upperb=[1000 1000 1000 1000 1000 1000];

for i = 1:length(test_n)
    [voce_plot(1:5,i),voce_plot(6,i)] = ...
        lsqnonlin(@voce_test_plot,voce_plot_0,lowerb,upperb,options);
    close all
    %% Saving of Voce parameters etc
    parameters(1:test_tot,1)=1:test_tot;
    parameters(test_n(i),2)=true_plastic_stress_plot{i}(1);
    parameters(test_n(i),3:7)=voce_plot(1:5,i);
    parameters(test_n(i),8)=plastic_strain_plot{i}(end);
end

cd(path_save)
%% Plotting the resulting true stress−plastic strain curves for ...
    all tests from Voce parameters
hFig=figure;
hAxes=axes;
for i = 1:length(test_n)
    plastic_strain_new{i}=linspace(0,max(plastic_strain_plot{i}),100);
    voce_stress{i}=voce_plot(1,i)+voce_plot(2,i).*...
        (1−exp(−voce_plot(3,i).*plastic_strain_new{i}))+...
        voce_plot(4,i).*(1−exp(−voce_plot(5,i).*plastic_strain_new{i}))
    hold on
    max_plastic_strain(i)=max(plastic_strain_new{i});
```

```
    max_voce_stress(i)=max(voce_stress{i});
    if i==1
        plot(plastic_strain_plot{i},true_plastic_stress_plot{i},'r',...
            plastic_strain_new{i},voce_stress{i},'--r')
    elseif i==2
        plot(plastic_strain_plot{i},true_plastic_stress_plot{i},...
            'Color',[0 0.5 0],'LineStyle','-')
        plot(plastic_strain_new{i},voce_stress{i},'Color',[0 0.5 ...
            0],'LineStyle','--')
    else
        plot(plastic_strain_plot{i},true_plastic_stress_plot{i},'b',...
            plastic_strain_new{i},voce_stress{i},'--b')
    end
    str1(i,:) = sprintf('Experimental test (test #%g)',test_n(i));
    str2(i,:) = sprintf('Adaption to model (test #%g)',test_n(i));
end
xlabel('Plastic strain, \epsilon_p')
ylabel('True stress, \sigma_t (MPa)')
axis([0 ceil(max(max_plastic_strain)*1.1/0.01)*0.01 0 ...
    ceil(max(max_voce_stress)*1.1/10)*10])
if length(test_n)==2
    legend(str1(1,:), str2(1,:),str1(2,:), ...
        str2(2,:),'Location','SouthEast');
elseif length(test_n)==3
    legend(str1(1,:), str2(1,:),str1(2,:), str2(2,:), str1(3,:), ...
        str2(3,:), 'Location','SouthEast');
    for i = 1:2
        if i==1
            %% Saving last figure to directory
            filename=sprintf('Test-0%g-0%g-0%g-(1).eps',test_n(1), ...
                test_n(2), test_n(3));
            save_figure_small(filename,hFig,hAxes)
        else
            axis auto
            xlim([0 ceil(max(max_plastic_strain)*1.1/0.01)*0.01])
            %% Saving last figure to directory
            filename=sprintf('Test-0%g-0%g-0%g-(2).eps',test_n(1), ...
                test_n(2), test_n(3));
            save_figure_small(filename,hFig,hAxes)
        end
    end
else disp('Code is not valid!')
end

%%
for i = 1:length(test_n)
    %% Plotting the resulting true stress-plastic strain curve ...
        from Voce parameters
    hFig=figure;
    hAxes=axes;
    hold on
    plot(plastic_strain_new{i},voce_stress{i},'b','LineWidth',2)
    plot(plastic_strain_plot{i},true_plastic_stress_plot{i},'r',...
        'LineWidth',1)
    legend('Adaption to model','Experimental ...
        test','Location','SouthEast')
    xlabel('Plastic strain, \epsilon_p')
    ylabel('True stress, \sigma_t (MPa)')
```

```
    axis([0 ceil(max(plastic_strain_new{i})*1.1/0.01)*0.01 0 ...
        ceil(max(voce_stress{i})*1.1/10)*10])

    %% Saving last figure to directory
    filename=sprintf('Test-0%g-(true_stress).eps',test_n(i));
    save_figure_small(filename,hFig,hAxes)

    %% Plotting the corrected eng. stress-strain curve
    hFig=figure;
    hAxes=axes;
    plot(eng_strain{i},eng_stress{i},'r','LineWidth',2)
    legend('Experimental test','Location','NorthEast')
    xlabel('Engineering strain, \epsilon_e')
    ylabel('Engineering stress, \sigma_e (MPa)')
    axis([0 ceil(max(eng_strain{i})*1.1/0.1)*0.1 0 ...
        ceil(max(eng_stress{i})*1.1/10)*10])

    %% Saving last figure to directory
    filename=sprintf('Test-0%g-(eng_stress).eps',test_n(i));
    save_figure_small(filename,hFig,hAxes)
end

%% Saving stress-strains in "results" array
for i = 1:length(test_n)
    results{test_n(i)}(1:length(eng_strain_np{i}),1)=eng_strain_np{i};
    results{test_n(i)}(1:length(eng_stress_np{i}),2)=eng_stress_np{i};
    results{test_n(i)}(1:length(true_strain_np{i}),3)=true_strain_np{i}
    results{test_n(i)}(1:length(true_stress_np{i}),4)=true_stress_np{i}
    results{test_n(i)}(1:length(plastic_strain_plot{i}),5)=...
        plastic_strain_plot{i};
    results{test_n(i)}(1:length(true_plastic_stress_plot{i}),6)=...
        true_plastic_stress_plot{i};
    results{test_n(i)}(1:length(plastic_strain_new{i}),7)=...
        plastic_strain_new{i};
    results{test_n(i)}(1:length(voce_stress{i}),8)=voce_stress{i};
end
cd('path\')
```

import_new_double.m:

```
%% Post-processing of experimental data (for 2 or 3 data sets)
global plastic_strain_new2 voce_stress_new

%% Manual input
% REMEMBER TO CLEAR PARAMETERS AND RESULTS VARIABLES BEFORE NEW SERIES
test_n_fit=[1 2];

%% Calculations
test_pos1=find(test_n_fit(1)==test_n);
test_pos2=find(test_n_fit(2)==test_n);
eps_p_max1=max(plastic_strain_plot{test_pos1});
eps_p_max2=max(plastic_strain_plot{test_pos2});
eps_p_max_min=min([eps_p_max1 eps_p_max2]);
eps_p_max_max=max([eps_p_max1 eps_p_max2]);
plastic_strain_new2=linspace(0,eps_p_max_min,100);
voce_stress_db(:,1)=voce_plot(1,test_pos1)+voce_plot(2,test_pos1).*...
```

```
    (1-exp(-voce_plot(3,test_pos1).*plastic_strain_new2))+...
    voce_plot(4,test_pos1).*(1-exp(-voce_plot(5,test_pos1).*...
    plastic_strain_new2));
voce_stress_db(:,2)=voce_plot(1,test_pos2)+voce_plot(2,test_pos2).*...
(1-exp(-voce_plot(3,test_pos2).*plastic_strain_new2))+...
voce_plot(4,test_pos2).*(1-exp(-voce_plot(5,test_pos2).*...
plastic_strain_new2));
voce_stress_db(:,3)=(voce_stress_db(:,1)+voce_stress_db(:,2))/2;
voce_stress_new=voce_stress_db(:,3);

options = optimset('TolFun', 1e-1000, 'TolX', 1e-1000, ...
    'MaxFunEvals',100000,'MaxIter',100000, 'PlotFcns', ...
    @optimplotresnorm);
[voce_new(1:5),voce_new(6)] = ...
    lsqnonlin(@voce_test_double,voce_plot_0,lowerb,upperb,options);

voce_stress_db(:,4)=voce_new(1)+voce_new(2).*(1-exp(-voce_new(3).*...
    plastic_strain_new2))+voce_new(4).*(1-exp(-voce_new(5).*...
    plastic_strain_new2));

%% Plotting of true stress-plastic strain
cd(path_save)
for i = 1:2
    hFig=figure;
    hAxes=axes;
    hold on
    if find(test_n==test_n_fit(1))==1
       plot(plastic_strain_plot{test_pos1},true_plastic_stress_plot...
            {test_pos1},'-r')
       plot(plastic_strain_new{test_pos1},voce_stress{test_pos1},'--r')
    end
    if find(test_n==test_n_fit(1))==2
       plot(plastic_strain_plot{test_pos1},true_plastic_stress_plot...
            {test_pos1},'Color',[0 0.5 0],'LineStyle','-')
       plot(plastic_strain_new{test_pos1},voce_stress{test_pos1},...
            'Color',[0 0.5 0],'LineStyle','--')
    end
    if find(test_n==test_n_fit(2))==2
       plot(plastic_strain_plot{test_pos2},true_plastic_stress_plot...
            {test_pos2},'Color',[0 0.5 0],'LineStyle','-')
       plot(plastic_strain_new{test_pos2},voce_stress{test_pos2},...
            'Color',[0 0.5 0],'LineStyle','--')
    end
    if find(test_n==test_n_fit(2))==3
       plot(plastic_strain_plot{test_pos2},true_plastic_stress_plot...
            {test_pos2},'-b')
       plot(plastic_strain_new{test_pos2},voce_stress{test_pos2},'--b')
    end
    plot(plastic_strain_new2,voce_stress_db(:,3),'k')
    plot(plastic_strain_new2,voce_stress_db(:,4),'--k')
    str1 = sprintf('Experimental test (test #%g)',test_n_fit(1));
    str2 = sprintf('Adaption to model (test #%g)',test_n_fit(1));
    str3 = sprintf('Experimental test (test #%g)',test_n_fit(2));
    str4 = sprintf('Adaption to model (test #%g)',test_n_fit(2));
    str5 = sprintf('Average curve');
    str6 = sprintf('Adaption to model (average curve)');
    legend(str1, str2 ,str3, str4, str5, str6,'Location','SouthEast');
    xlabel('Plastic strain, \epsilon_p')
    ylabel('True stress, \sigma_t (MPa)')
```

```
    if i==1
        axis([0 ceil(eps_p_max_max*1.1/0.01)*0.01 0 ...
            ceil(max([voce_stress{test_pos1} ...
            voce_stress{test_pos2}])*1.1/10)*10])
        %% Saving last figure to directory
        filename=sprintf('Test-0%g-0%g-(1).eps',test_n_fit(1), ...
            test_n_fit(2));
        save_figure_small(filename,hFig,hAxes)
    else
        xlim([0 ceil(eps_p_max_max*1.1/0.01)*0.01])
        %% Saving last figure to directory
        filename=sprintf('Test-0%g-0%g-(2).eps',test_n_fit(1), ...
            test_n_fit(2));
        save_figure_small(filename,hFig,hAxes)
    end
end

%% Saving of Voce parameters etc (for average curve)
if size(parameters,1)==test_tot
    row_n=test_tot+2;
else
    row_n=size(parameters,1)+1;
end
parameters(row_n,1)=test_n(1);
parameters(row_n,2)=(true_plastic_stress_plot{test_pos1}(1)+...
    true_plastic_stress_plot{test_pos2}(1))/2;
parameters(row_n,3:7)=voce_new(1:5);
parameters(row_n,8)=eps_p_max_min;

%% Saving of calculations to .xlsx and .mat files
if test_n(end)==test_tot
    parameters_xlsx_string=arrayfun(@num2str, parameters, 'unif', 0);

    parameters_xlsx = {'Parameters for yield stress + Voce ...
        hardening and max plastic strain at onset of necking', '', ...
        '', '','','','','',''; ...
        'Test number' '\sigma_Y (directly)', '\sigma_Y', 'Q_1', ...
            'C_1', 'Q_2', 'C_2', 'Max. plastic strain'};
    parameters_xlsx(3:size(parameters,1)+2,1:8)=parameters_xlsx_string;
    xlswrite('parameters.xlsx', parameters_xlsx);

    save('parameters', 'parameters')
    save('results', 'results')
end
cd('C:\Users\Eivind\Documents\MATLAB\Post-process QS')
```

engs_func_single.m:

```
function eng_res = engs_func_single(eng)
global eng_strain_calc eng_stress_calc

eng_calc=eng(1)+eng(2).*eng_strain_calc+eng(3).*eng_strain_calc.^2+...
    eng(4).*eng_strain_calc.^3;
eng_res=abs(eng_calc-eng_stress_calc);
```

engs_func_plot.m:

```
function eng_res_plot = engs_func_plot(eng_plot)
global i eng_strain_calc_plot eng_stress_calc_plot

eng_calc_plot=eng_plot(1)+eng_plot(2).*eng_strain_calc_plot{i}+...
    eng_plot(3).*eng_strain_calc_plot{i}.^2+eng_plot(4).*...
    eng_strain_calc_plot{i}.^3;
eng_res_plot=abs(eng_calc_plot-eng_stress_calc_plot{i});
```

voce_test_single.m:

```
function voce_res = voce_test_single(voce)
global plastic_strain true_plastic_stress

voce_calc=voce(1)+voce(2).*(1-exp(-voce(3).*plastic_strain))+...
    voce(4).*(1-exp(-voce(5).*plastic_strain));
voce_res=abs(voce_calc-true_plastic_stress);
```

voce_test_plot.m:

```
function voce_res_plot = voce_test_plot(voce_plot)
global plastic_strain_plot true_plastic_stress_plot i

voce_calc_plot=voce_plot(1)+voce_plot(2).*(1-exp(-voce_plot(3).*...
    plastic_strain_plot{i}))+voce_plot(4).*...
    (1-exp(-voce_plot(5).*plastic_strain_plot{i}));
voce_res_plot=abs(voce_calc_plot-true_plastic_stress_plot{i});
```

voce_test_double.m:

```
function voce_res = voce_test_double(voce)
global plastic_strain_new2 voce_stress_new

voce_calc=(voce(1)+voce(2).*(1-exp(-voce(3).*plastic_strain_new2))+...
    voce(4).*(1-exp(-voce(5).*plastic_strain_new2)))';
voce_res=abs(voce_calc-voce_stress_new);
```

D.2 Post-Processing of Data from SHTB Tests

Script name	Script/function	Explanation
import_txt_strain_gauge.m	Script	Post-processing of exp. data from strain gauges
import_cam_measurements.m	Script	Post-processing of exp. data from camera
engs_func_single_shtb.m	Function	Calculation of approx. engineering stress around the neck
voce_test_single_shtb.m	Function	Calculation of the resulting approx. true stress parameters by Voce hardening parameters
voce_test_single_shtb_corr.m	Function	Calculation of the resulting approx. true stress parameters for Bridgman correctin by Voce hardening parameters
true_strain_cal_func.m	Function	Calculation of approx. polynomial for true strain values
diameter_cal_func.m	Function	Calculation of approx. polynomial for min. diameter values
rad_cal_func.m	Function	Calculation of approx. polynomial for radius of curvature

Table D.2: Overview of Matlab scripts for post-processing of data from SHTB tests

import_txt_strain_gauge.m:

```
%% Script for post-processing of data from SHTB tests (strain ...
    gauge measurements)
clearvars -except parameters results
global eng_strain_calc eng_stress_calc plastic_strain ...
    true_plastic_stress

%% Manual input
filename = 'Filename';
test_n=1;
test_id=1;
test_tot=10;
temp=293;
path='path\experimental_data';
cd(path)
delimiter = ',';
startRow = 19; % CHECK THIS!

%% Format string for each line of text:
formatSpec = '%s%s%s%s%s%s%s%s%s%s%[^\n\r]';

%% Open the text file.
fileID = fopen(filename,'r');
path='path\';
```

```
cd(path)

%% Read columns of data according to format string.
dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter, ...
    'HeaderLines' ,startRow-1, 'ReturnOnError', false);

%% Close the text file.
fclose(fileID);

%% Convert the contents of columns containing numeric strings to ...
    numbers.
% Replace non-numeric strings with NaN.
raw = [dataArray{:,1:end-1}];
numericData = NaN(size(dataArray{1},1),size(dataArray,2));

for col=[1,2,3,4,5,6,7,8,9,10]
    % Converts strings in the input cell array to numbers. ...
        Replaced non-numeric
    % strings with NaN.
    rawData = dataArray{col};
    for row=1:size(rawData, 1);
        % Create a regular expression to detect and remove ...
            non-numeric prefixes and
        % suffixes.
        regexstr = ...
            '(?<prefix>.*?)(?<numbers>([-]*(\d+[\,]*)+[\.]{0,1}\d*...
        [eEdD]{0,1}[-+]*\d*[i]{0,1})|([-]*(\d+[\,]*)*[\.]{1,1}\d+...
            [eEdD]{0,1}[-+]*\d*[i]{0,1}))(?<suffix>.*)';
        try
            result = regexp(rawData{row}, regexstr, 'names');
            numbers = result.numbers;

            % Detected commas in non-thousand locations.
            invalidThousandsSeparator = false;
            if any(numbers==',');
                thousandsRegExp = '^\d+?(\,\d{3})*\.{0,1}\d*$';
                if isempty(regexp(thousandsRegExp, ',', 'once'));
                    numbers = NaN;
                    invalidThousandsSeparator = true;
                end
            end
            % Convert numeric strings to numbers.
            if ~invalidThousandsSeparator;
                numbers = textscan(strrep(numbers, ',', ''), '%f');
                numericData(row, col) = numbers{1};
                raw{row, col} = numbers{1};
            end
        catch me
        end
    end
end

%% Replace non-numeric cells with 0.0
R = cellfun(@(x) (~isnumeric(x) && ~islogical(x)) || ...
    isnan(x),raw); % Find non-numeric cells
raw(R) = {0.0}; % Replace non-numeric cells

%% Allocate imported array to column variable names
test_time = cell2mat(raw(:, 1)); %time
```

```
eng_stress = cell2mat(raw(:, 2)); %engineering stress
eng_strain = cell2mat(raw(:, 3)); %engineering strain (not correct ...
    Young's modulus)
test_strain_rate = cell2mat(raw(:, 4)); %engineering strain rate
test_ing_strain = cell2mat(raw(:, 5)); %engineering strain ...
    (correct Young's modulus)
test_true_stress = cell2mat(raw(:, 6)); %true stress
test_true_strain = cell2mat(raw(:, 7)); %true strain
test_true_strain_rate = cell2mat(raw(:, 8)); %true strain rate
test_plastic_strain = cell2mat(raw(:, 9)); %plastic strain
test_plastic_work = cell2mat(raw(:, 10)); %plastic work

%% Clear temporary variables
clearvars filename delimiter startRow formatSpec fileID dataArray ...
    ans raw numericData col rawData row regexstr result numbers ...
    invalidThousandsSeparator thousandsRegExp me R;

%% Determining Young's modulus
figure
plot(eng_strain,eng_stress)
xlabel('Engineering strain')
ylabel('Engineering stress (MPa)')
title('Select data range for determining E-modulus + yield stress')
legend('Experimental data','Location','NorthEast')
axis([-0.1 max(eng_strain)*1.1 -10 max(eng_stress)*1.1])

pause on
[x,y]=ginput(2);
x5=x(1); x6=x(2);
close

pos1=find(min(abs(eng_strain-x5))==abs(eng_strain-x5),1);
pos2=find(min(abs(eng_strain-x6))==abs(eng_strain-x6),1);
pos1_r=pos1;
pos2_r=pos2;

figure
plot(eng_strain(pos1:pos2),eng_stress(pos1:pos2))
xlabel('Engineering strain')
ylabel('Engineering stress (MPa)')
title('Select data range for E-modulus')
legend('Experimental data','Location','NorthEast')
axis([0 eng_strain(pos2) 0 max(eng_stress(pos1:pos2))])

pause on
[x,y]=ginput(2);
x5=x(1); x6=x(2);

pos1=find(min(abs(eng_strain-x5))==abs(eng_strain-x5),1);
pos2=find(min(abs(eng_strain-x6))==abs(eng_strain-x6),1);
E_x=eng_strain(pos1:pos2);
E_y=eng_stress(pos1:pos2);
P=polyfit(E_x,E_y,1);
E_calc_x_min=min(eng_strain);
E_calc_x_max=eng_strain(find(eng_stress==max(eng_stress),1));
E_calc_x=linspace(E_calc_x_min,E_calc_x_max,100);
E_calc_y=P(2)+P(1).*E_calc_x;
E_calc_x_end=find(min(abs(E_calc_y-max(eng_stress)))==...
    abs(E_calc_y-max(eng_stress)));
```

```
E_calc_x_start=find(abs(E_calc_y)==min(abs(E_calc_y)));
E_meas=(E_calc_y(end)-E_calc_y(1))/(E_calc_x(end)-E_calc_x(1));
hold on
plot(E_calc_x(E_calc_x_start:E_calc_x_end),E_calc_y...
    (E_calc_x_start:E_calc_x_end),'r')
legend('Experimental data','Elastic tangent ...
    modulus','Location','NorthEast')

%% Translating start of straining to origo
eps_var=eng_strain(pos1)-eng_stress(pos1)/E_meas;
eng_strain_corr(1)=0;
eng_strain_corr(2)=eng_stress(pos1)/E_meas;
eng_strain_corr(3:length(eng_strain(pos1:end))+2)=...
    eng_strain(pos1:end)-eps_var;
eng_stress_corr(1)=0;
eng_stress_corr(2)=eng_stress(pos1);
eng_stress_corr(3:length(eng_stress(pos1:end))+2)=eng_stress(pos1:end);
eng_strain=eng_strain_corr;
eng_stress=eng_stress_corr;

%% Correction of Young's modulus
E_corr=(-3.9*exp(0.0033*temp)+79)*1000;
eng_strain_corr=eng_strain-eng_stress.*...
    ((E_corr-E_meas)/(E_corr*E_meas));

%% Determing the yield point
pos2_r_new=pos2_r-pos1+2;
figure
hold on
plot(eng_strain(1:pos2_r_new),eng_stress(1:pos2_r_new),'b')
plot(eng_strain_corr(1:pos2_r_new),eng_stress(1:pos2_r_new),'Color',...
    [0 0.5 0])
plot(E_calc_x(E_calc_x_start:E_calc_x_end)-eps_var,E_calc_y...
    (E_calc_x_start:E_calc_x_end),'r')
title('Determine the yield point (for the uncorrected curve)')
xlabel('Engineering strain, \epsilon_e')
ylabel('Engineering stress, \sigma_e (MPa)')
legend('Uncorrected strain values','Corrected strain ...
    values','Elastic tangent modulus','Location','SouthEast')

pause on
[x,y]=ginput(1);
x5=x(1);
close all

ypos=find(min(abs(eng_strain-x5))==abs(eng_strain-x5),1);
eng_strain=eng_strain_corr;
true_strain=log(1+eng_strain);
true_stress=eng_stress.*(1+eng_strain);

%% Curve fitting of engineering stress-strain curve (for ...
    determination of necking point)
figure;
plot(eng_strain,eng_stress,'LineWidth',1)
xlabel('Engineering strain, \epsilon_e')
ylabel('Engineering stress, \sigma_e (MPa)')
title('Define data range for defining necking point')
legend('Experimental test','Location','NorthEast')
axis([-0.002 0.7 0 100])
```

```

pause on
[x,y]=ginput(2);
x5=x(1); x6=x(2);
close

pos1=find(min(abs(eng_strain-x5))==abs(eng_strain-x5),1);
pos2=find(min(abs(eng_strain-x6))==abs(eng_strain-x6),1);
eng_strain_calc=eng_strain(pos1:pos2);
eng_stress_calc=eng_stress(pos1:pos2);

A0=5; B0=5; C0=5;
eng_strain_0=[A0 B0 C0];
lowerb=[-1000 -1000 -1000];
upperb=[1000 1000 1000];

options = optimset('TolFun', 1e-1000, 'TolX', 1e-1000, ...
    'MaxFunEvals',100000,'MaxIter',100000, 'PlotFcns', ...
    @optimplotresnorm);
[eng_strain_cal(1:3),eng_strain_cal(4)] = ...
    lsqnonlin(@engs_func_single_shtb,eng_strain_0,lowerb,upperb,...
    options)
close

eng_strain_lin=linspace(min(eng_strain_calc),max(eng_strain_calc),100);
eng_stress_calc_new=eng_strain_cal(1)+eng_strain_cal(2).*...
    eng_strain_lin+eng_strain_cal(3).*eng_strain_lin.^2;
np=find(eng_stress_calc_new==max(eng_stress_calc_new),1);
np_strain=eng_strain_lin(np);
np_x_h=eng_strain_lin;
np_y_h=ones(length(eng_strain_lin))*max(eng_stress_calc_new);
np_x_v=[eng_strain_lin(np) eng_strain_lin(np)];
np_y_v=[min(eng_stress_calc_new) ...
    min(eng_stress_calc_new)+(max(eng_stress_calc_new)-...
    min(eng_stress_calc_new))*2];

figure
hold on
plot(eng_strain_calc,eng_stress_calc,'b','LineWidth',1)
plot(eng_strain_lin,eng_stress_calc_new,'Color',[0 0.5 ...
    0],'LineWidth',1)
plot(np_x_h,np_y_h,'r--','LineWidth',1.5)
plot(np_x_v,np_y_v,'r--','LineWidth',1.5)
legend('Experimental test', 'Approx. polynomial','Necking point')
xlabel('Engineering strain, \epsilon_e')
ylabel('Engineering stress, \sigma_e (MPa)')

%% Engineering stress-strain from start --> onset of necking
np_eng=find(min(abs(eng_strain-np_strain))==...
    abs(eng_strain-np_strain),1); %position of necking point in ...
        eng_strain vector
eng_strain_np=eng_strain(1:np_eng);
eng_stress_np=eng_stress(1:np_eng);

%% True stress-strain from start --> onset of necking
np_strain_true=log(1+np_strain);
np_true=find(min(abs(true_strain-np_strain_true))==...
    abs(true_strain-np_strain_true),1);
true_strain_np=true_strain(1:np_true);
```

```
true_stress_np=true_stress(1:np_true);

%% Plotting of eng. stress−strain and true stress−strain until ...
    onset of necking
figure
plot(eng_strain_np,eng_stress_np,true_strain_np,true_stress_np)
legend('Eng. stress−strain until necking', 'True stress−strain ...
    until necking', 'Location', 'SouthEast')

%% True stress−plastic strain from yield to necking
plastic_strain=true_strain(ypos:np_true)−true_strain(ypos);
true_plastic_stress=true_stress(ypos:np_true);

%% Calculating Voce−rule parameters
sigY_0=50; Q1_0=50; C1_0=10; Q2_0=20; C2_0=40;
voce_0=[sigY_0 Q1_0 C1_0 Q2_0 C2_0];
lowerb=[−1000 −1000 −1000 −1000 −1000 −1000];
upperb=[1000 1000 1000 1000 1000 1000];

[voce(1:5),voce(6)] = ...
    lsqnonlin(@voce_test_single_shtb,voce_0,lowerb,upperb,options);
close all

parameters(test_n,1)=test_id;
parameters(test_n,2)=true_stress(ypos);
parameters(test_n,3:7)=voce(1:5);
parameters(test_n,8)=max(plastic_strain);

%% Plotting the resulting true stress−plastic strain curve from ...
    Voce parameters
plastic_strain_new=linspace(0,max(plastic_strain),100);
voce_stress=voce(1)+voce(2).*(1−exp(−voce(3).*plastic_strain_new))+...
    voce(4).*(1−exp(−voce(5).*plastic_strain_new));
hFig=figure;
hAxes=axes;
hold on
plot(plastic_strain_new,voce_stress,'b','LineWidth',2)
plot(plastic_strain,true_plastic_stress,'r','LineWidth',1)
legend('Adaption to model', 'Experimental ...
    test','Location','SouthEast')
xlabel('Plastic strain, \epsilon_p')
ylabel('True stress, \sigma_t (MPa)')
axis([0 ceil(max(plastic_strain_new)*1.1/0.01)*0.01 0 ...
    ceil(max(voce_stress)*1.1/10)*10])

%% Saving last figure to directory
path='path\plots_strain_gauges';
cd(path)
filename=sprintf('Test−0%g−f−Data−(true_stress).eps',test_id);
save_figure_small(filename,hFig,hAxes)

%% Plotting the corrected eng. stress−strain curve
hFig=figure;
hAxes=axes;
plot(eng_strain,eng_stress,'r','LineWidth',2)
legend('Experimental test','Location','NorthEast')
xlabel('Engineering strain, \epsilon_e')
ylabel('Engineering stress, \sigma_e (MPa)')
axis([0 ceil(max(eng_strain)*1.1/0.1)*0.1 0 ...
```

```
    ceil(max(eng_stress)*1.1/10)*10])

%% Saving last figure to directory
filename=sprintf('Test-0%g-f-Data-(eng_stress).eps',test_id);
save_figure_small(filename,hFig,hAxes)

%% Saving stress-strains in "results" array
results{test_n}(1:length(eng_strain),1)=eng_strain;
results{test_n}(1:length(eng_stress),2)=eng_stress;
results{test_n}(1:length(true_strain_np),3)=true_strain_np;
results{test_n}(1:length(true_stress_np),4)=true_stress_np;
results{test_n}(1:length(plastic_strain),5)=plastic_strain;
results{test_n}(1:length(true_plastic_stress),6)=true_plastic_stress;
results{test_n}(1:length(plastic_strain_new),7)=plastic_strain_new;
results{test_n}(1:length(voce_stress),8)=voce_stress;

%% Saving of calculations to .xlsx and .mat files
if test_n==test_tot
    parameters_xlsx_string=arrayfun(@num2str, parameters, 'unif', 0);

    parameters_xlsx = {'Parameters for yield stress + Voce ...
        hardening and max plastic strain at onset of necking', '', ...
        '', '','','','',''; ...
        'Test number' '\sigma_Y (directly)', '\sigma_Y', 'Q_1', ...
            'C_1', 'Q_2', 'C_2', 'Max. plastic strain'};
    parameters_xlsx(3:size(parameters,1)+2,1:8)=parameters_xlsx_string;
    xlswrite('parameters.xlsx', parameters_xlsx);

    save('parameters', 'parameters')
    save('results', 'results')
end
cd('path\')
```

import_cam_measurements.m:

```
%% Script for post-processing of data from SHTB tests (camera ...
    measurements)
clearvars -except parameters results
global diameter_s timecam true_strain true_plastic_stress ...
    plastic_strain rad_curv_data timecam_rad true_plastic_stress_corr

%% Manual input
path='path\exp_data\';
cd(path)
load ('Filename') % camera file
area_initial=pi*(3/2)^2;
filename = 'Filename'; % strain gauge measurement
temp=523;
E_meas_corr=2000;
t_delay=-0.05;
test_n=1;
test_id=1;
test_tot=10;
delimiter = ',';
startRow = 19; % CHECK THIS!

```

```
%% Format string for each line of text:
formatSpec = '%s%s%s%s%s%s%s%s%s%s%[^\n\r]';

%% Open the text file.
fileID = fopen(filename,'r');
path='C:\Users\Eivind\Documents\MATLAB\Post-process SHTB';
cd(path)

%% Read columns of data according to format string.
dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter, ...
    'HeaderLines' ,startRow-1, 'ReturnOnError', false);

%% Close the text file.
fclose(fileID);

%% Convert the contents of columns containing numeric strings to ...
    numbers.
% Replace non-numeric strings with NaN.
raw = [dataArray{:,1:end-1}];
numericData = NaN(size(dataArray{1},1),size(dataArray,2));

for col=[1,2,3,4,5,6,7,8,9,10]
    % Converts strings in the input cell array to numbers. ...
        Replaced non-numeric
    % strings with NaN.
    rawData = dataArray{col};
    for row=1:size(rawData, 1);
        % Create a regular expression to detect and remove ...
            non-numeric prefixes and
        % suffixes.
        regexstr = ...
            '(?<prefix>.*?)(?<numbers>([-]*(\d+[\,]*)+[\.]{0,1}\d*...
        [eEdD]{0,1}[-+]*\d*[i]{0,1})|([-]*(\d+[\,]*)*[\.]{1,1}\d+...
            [eEdD]{0,1}[-+]*\d*[i]{0,1}))(?<suffix>.*)';
        try
            result = regexp(rawData{row}, regexstr, 'names');
            numbers = result.numbers;

            % Detected commas in non-thousand locations.
            invalidThousandsSeparator = false;
            if any(numbers==',');
                thousandsRegExp = '^\d+?(\,\d{3})*\.{0,1}\d*$';
                if isempty(regexp(thousandsRegExp, ',', 'once'));
                    numbers = NaN;
                    invalidThousandsSeparator = true;
                end
            end
            % Convert numeric strings to numbers.
            if ~invalidThousandsSeparator;
                numbers = textscan(strrep(numbers, ',', ''), '%f');
                numericData(row, col) = numbers{1};
                raw{row, col} = numbers{1};
            end
        catch me
        end
    end
end

%% Replace non-numeric cells with 0.0
```

```
R = cellfun(@(x) (~isnumeric(x) && ~islogical(x)) || ...
     isnan(x),raw); % Find non-numeric cells
raw(R) = {0.0}; % Replace non-numeric cells

%% Allocate imported array to column variable names
test_time = cell2mat(raw(:, 1));
eng_stress = cell2mat(raw(:, 2)); %engineering stress
eng_strain = cell2mat(raw(:, 3)); %engineering strain (not correct ...
     Young's modulus)
test_strain_rate = cell2mat(raw(:, 4)); %engineering strain rate
test_ing_strain = cell2mat(raw(:, 5)); %engineering strain ...
     (correct Young's modulus)
test_true_stress = cell2mat(raw(:, 6)); %true stress
test_true_strain = cell2mat(raw(:, 7)); %true strain
test_true_strain_rate = cell2mat(raw(:, 8)); %true strain rate
test_plastic_strain = cell2mat(raw(:, 9)); %plastic strain

%% Synchronization of diameter and strains with respect to time
timecam=timecam+t_delay;
pos_d=find(abs(timecam)==min(abs(timecam)));
timecam=timecam(pos_d:end);
diameter_s=diameter_s(pos_d:end);
true_strain=true_strain(pos_d:end);
rinf=rinf(pos_d:end);
rsup=rsup(pos_d:end);

%% Determining the elastic tangent modulus
figure
plot(eng_strain,eng_stress)
xlabel('Engineering strain')
ylabel('Engineering stress (MPa)')
title('Select data range for determining E-modulus + yield stress')
legend('Experimental data','Location','NorthEast')
axis([-0.1 max(eng_strain)*1.1 -10 max(eng_stress)*1.1])

pause on
[x,y]=ginput(2);
x5=x(1);
x6=x(2);
close

pos1=find(min(abs(eng_strain-x5))==abs(eng_strain-x5),1);
pos2=find(min(abs(eng_strain-x6))==abs(eng_strain-x6),1);
figure
plot(eng_strain(pos1:pos2),eng_stress(pos1:pos2))
xlabel('Engineering strain')
ylabel('Engineering stress (MPa)')
title('Select data range for E-modulus')
legend('Experimental data','Location','NorthEast')
axis([0 eng_strain(pos2) 0 max(eng_stress(pos1:pos2))])

pause on
[x,y]=ginput(2);
x5=x(1);
x6=x(2);

pos1=find(min(abs(eng_strain-x5))==abs(eng_strain-x5),1);
pos2=find(min(abs(eng_strain-x6))==abs(eng_strain-x6),1);
E_x=eng_strain(pos1:pos2);
```

```
E_y=eng_stress(pos1:pos2);
P=polyfit(E_x,E_y,1);
E_calc_x_min=min(eng_strain);
E_calc_x_max=eng_strain(find(eng_stress==max(eng_stress),1));
E_calc_x=linspace(E_calc_x_min,E_calc_x_max,100);
E_calc_y=P(2)+P(1).*E_calc_x;
E_calc_x_end=find(min(abs(E_calc_y-max(eng_stress)))==...
    abs(E_calc_y-max(eng_stress)));
E_calc_x_start=find(abs(E_calc_y)==min(abs(E_calc_y)));
% E_meas=(E_calc_y(end)-E_calc_y(1))/(E_calc_x(end)-E_calc_x(1));
E_meas=E_meas_corr;

%% Correction of Young's modulus
E_corr=(-3.9*exp(0.0033*temp)+79)*1000;
eng_strain_corr=eng_strain-eng_stress.*...
    ((E_corr-E_meas)/(E_corr*E_meas))

%% Plotting true strain for strain gauge measurement + camera
test_true_strain_new=log(1+test_ing_strain);
test_true_strain_formula=log(1+eng_strain_corr);
force=eng_stress*area_initial; %force

figure
hold on
plot(test_time,test_true_strain_formula,'r')
plot(timecam,true_strain,'Color',[0 0.5 0])
legend('True strain, strain gauges','True strain, ...
    camera','Location','NorthWest')
xlabel('Time, t (ms)')
ylabel('True strain, \epsilon_t')

figure
plot(test_time,eng_strain,test_time,test_ing_strain,test_time,...
    eng_strain_corr,test_time,test_true_strain,test_time,...
    test_true_strain_new,test_time,test_true_strain_formula)
legend('Eng. strain (E-modulus not corrected)','Eng. strain ...
    (E-modulus corrected)','Eng. strain (E-modulus corrected ...
    (formula)','True strain',...
    'True strain ln(1+eng. strain)','True strain ...
        (formula)','Location','SouthEast')
xlabel('Time, t (s)')
ylabel('Strain, \epsilon')
title('From strain gauges')

%% Curvefitting of diameter and strains
options = optimset('TolFun', 1e-1000, 'TolX', 1e-1000, ...
    'MaxFunEvals',100000,'MaxIter',100000, 'PlotFcns', ...
    @optimplotresnorm);
A0=5; B0=5; C0=5; D0=5; E0=5; F0=5; G0=5; H0=5;
diameter_cal_0=[A0 B0 C0 D0 E0 F0 G0 H0];
true_strain_cal_0=[A0 B0 C0 D0 E0 F0 G0 H0];
lowerb=[];
upperb=[];

[diameter_cal(1:8),diameter_cal(9)] = ...
    lsqnonlin(@diameter_cal_func,diameter_cal_0,lowerb,upperb,options);
close

if test_time(end)>timecam(end)
```

```
    pos_max=find(min(abs(test_time-timecam(end)))==...
        abs(test_time-timecam(end)),1);
else
    pos_max=length(test_time);
end

timecam_new=test_time(1:pos_max);
diameter_new=diameter_cal(1).*(timecam_new.^7)+diameter_cal(2).*...
    (timecam_new.^6)+diameter_cal(3).*(timecam_new.^5)+...
    diameter_cal(4).*(timecam_new.^4)+...
    diameter_cal(5).*(timecam_new.^3)+diameter_cal(6).*...
    (timecam_new.^2)+diameter_cal(7).*timecam_new+diameter_cal(8);

[true_strain_cal(1:8),true_strain_cal(9)] = ...
    lsqnonlin(@true_strain_cal_func,true_strain_cal_0,...
    lowerb,upperb,options);
close

true_strain_new=true_strain_cal(1).*(timecam_new.^7)+...
    true_strain_cal(2).*(timecam_new.^6)+true_strain_cal(3).*...
    (timecam_new.^5)+true_strain_cal(4).*(timecam_new.^4)+...
    true_strain_cal(5).*(timecam_new.^3)+true_strain_cal(6).*...
    (timecam_new.^2)+true_strain_cal(7).*timecam_new+true_strain_cal(8)

%% Curvefitting and back-extrapolation of curvature of radius
if test_n ~= 4
    rad_curv_data=(rinf+rsup)./2;
else
    rad_curv_data=rsup;
end
pos_cam=find(rad_curv_data>3,1);
timecam_rad=timecam(pos_cam:end);
rad_curv_data=rad_curv_data(pos_cam:end);

A0=5; B0=5; C0=5; D0=5;
rad_cal_0=[A0 B0 C0 D0];
lowerb=[];
upperb=[];
[rad_cal(1:4),rad_cal(5)] = ...
    lsqnonlin(@rad_cal_func,rad_cal_0,lowerb,upperb,options);
rad_curv=rad_cal(1).*(timecam_new.^3)+rad_cal(2).*(timecam_new.^2)+...
    rad_cal(3).*timecam_new+rad_cal(4);

figure
hold on
plot(timecam(pos_cam:end),rinf(pos_cam:end),'Color',[0 0.5 ...
    0],'Marker','o','MarkerFaceColor',[0 0.5 0],'MarkerEdgeColor','k')
plot(timecam(pos_cam:end),rsup(pos_cam:end),'COlor','r','Marker',...
    'o','MarkerFaceColor','r','MarkerEdgeColor','k')
plot(timecam_new,rad_curv,'k')
legend('Lower radius of curvature','Upper radius of ...
    curvature','Back-extrapolation')
xlabel('Time, t (ms)')
ylabel('Radius of curvature, R (mm)')

%% Determing the yield point
true_stress=force(1:pos_max)./(pi.*(diameter_new.^2)./4);
figure
plot(true_strain_new,true_stress)
```

```
title('Determine the yield point')
xlim([-0.05 0.3])
ylim([0 100])
legend('From camera')
xlabel('True strain, \epsilon_t')
ylabel('True stress, \sigma_t')

pause on
[x,y]=ginput(1);
x5=x(1);
close
ypos=find(min(abs(true_strain_new-x5))==abs(true_strain_new-x5),1);

%% True stress-plastic strain from yield
plastic_strain=true_strain_new(ypos:end)-true_strain_new(ypos);
true_plastic_stress=true_stress(ypos:end);

%% Bridgman correction
true_plastic_stress_corr=1./((1+2.*rad_curv(ypos:end)./...
    (diameter_new(ypos:end)./2)).*log(1+(diameter_new(ypos:end)./2)...
    ./2./rad_curv(ypos:end))).*true_plastic_stress;
figure
plot(plastic_strain,true_plastic_stress,plastic_strain,...
    true_plastic_stress_corr)

%% Bridgman correction (Empirical formula)
kappa=1.11;
eps_uts=[0.2248 0.2194 0.1994 0.2534]; %true strain at onset of ...
    necking (test 2, 5, 9, 11)
a_R_cam=diameter_new(ypos:end)./rad_curv(ypos:end);
a_R_approx=kappa.*(plastic_strain-eps_uts(test_n));
true_plastic_stress_corr_approx=1./((1+2./a_R_approx).*...
    log(1+a_R_approx./2)).*true_plastic_stress;

%% Determining the maximum plastic strain value
figure
plot(plastic_strain,true_plastic_stress,...
    plastic_strain,true_plastic_stress_corr)
legend('Uncorrectec', 'Bridgman corrected')
xlabel('True strain, \epsilon_t')
ylabel('True stress, \sigma_t')
title('Determine the maximum plastic strain value for fitting')

pause on
[x,y]=ginput(1);
x5=x(1);
close
p_max=find(min(abs(plastic_strain-x5))==abs(plastic_strain-x5),1);
plastic_strain=plastic_strain(1:p_max);
true_plastic_stress=true_plastic_stress(1:p_max);
true_plastic_stress_corr=true_plastic_stress_corr(1:p_max);
true_plastic_stress_corr_approx=...
    true_plastic_stress_corr_approx(1:p_max);
a_R_cam=a_R_cam(1:p_max);
a_R_approx=a_R_approx(1:p_max);

%%
np_pos=find(min(abs(plastic_strain-eps_uts(test_n)))==...
    abs(plastic_strain-eps_uts(test_n)));
```

```
true_plastic_stress_corr_approx(1:np_pos-1)=...
    true_plastic_stress(1:np_pos-1);
figure;
hold on
plot(plastic_strain(np_pos:end),a_R_cam(np_pos:end),'r')
plot(plastic_strain(np_pos:end),a_R_approx(np_pos:end),'k')
legend('Camera measurement','Empirical ...
    formula','Location','NorthWest')
xlabel('Plastic strain, \epsilon_p')
ylabel('Ratio a/R')

%% Calculating Voce-rule parameters
sigY_0=50; Q1_0=50; C1_0=10; Q2_0=20; C2_0=40;
voce_0=[sigY_0 Q1_0 C1_0 Q2_0 C2_0];
lowerb=[-1000 -1000 -1000 -1000 -1000 -1000];
upperb=[1000 1000 1000 1000 1000 1000];
path='path\';
cd(path)

for i = 1:2
    if i==1
        [voce(1:5),voce(6)] = ...
            lsqnonlin(@voce_test_single_shtb,voce_0,lowerb,...
            upperb,options);
        close
    else
        [voce_corr(1:5),voce_corr(6)] = ...
            lsqnonlin(@voce_test_single_shtb_corr,voce_0,lowerb,...
            upperb,options);
        close
    end
end

parameters(test_n,1)=test_id;
parameters(test_n,2)=true_stress(ypos);
parameters(test_n,3:7)=voce(1:5);
parameters(test_n,8)=max(plastic_strain);
parameters(test_n,9:13)=voce_corr(1:5);
results{test_n}(:,1)=plastic_strain;
results{test_n}(:,2)=true_plastic_stress_corr;

%% Plotting the resulting true stress-plastic strain curve for ...
    uncorrected and corrected stress (camera+approx.)
hFig=figure;
hAxes=axes;
hold on
plot(plastic_strain,true_plastic_stress,'r','LineWidth',1)
plot(plastic_strain,true_plastic_stress_corr,'Color',[0 0.5 ...
    0],'LineWidth',1)
plot(plastic_strain,true_plastic_stress_corr_approx,'b','LineWidth',1)
legend('No correction of stress values','Bridgman correction using ...
    camera measurements','Bridgman correction using empirical ...
    formula','Location','SouthEast')
xlabel('Plastic strain, \epsilon_p')
ylabel('True stress, \sigma_t (MPa)')
axis([0 ceil(max(plastic_strain)*1.1/0.01)*0.01 0 ...
    ceil(max(true_plastic_stress)*1.1/10)*10])

%% Saving last figure to directory
```

```
path='C:\Users\Eivind\Documents\MATLAB\Post-process ...
     SHTB\plots_camera';
cd(path)
filename=sprintf('Test-0%g-(true_stress-plastic_strain_ALL).eps',...
    test_id);
save_figure_small(filename,hFig,hAxes)

%% Plotting the resulting true stress-plastic strain curve from ...
     Voce parameters for uncorrected and corrected stress
plastic_strain_new=linspace(0,max(plastic_strain),100);
voce_stress=voce(1)+voce(2).*(1-exp(-voce(3).*plastic_strain_new))+...
    voce(4).*(1-exp(-voce(5).*plastic_strain_new));
voce_stress_corr=voce_corr(1)+voce_corr(2).*(1-exp(-voce_corr(3).*...
    plastic_strain_new))+voce_corr(4).*(1-exp(-voce_corr(5).*...
    plastic_strain_new));
hFig=figure;
hAxes=axes;
hold on
plot(plastic_strain_new,voce_stress_corr,'b','LineWidth',2)
plot(plastic_strain,true_plastic_stress_corr,'r','LineWidth',1)
plot(plastic_strain_new,voce_stress,'--b','LineWidth',2)
plot(plastic_strain,true_plastic_stress,'--r','LineWidth',1)
legend('Adaption to model (Bridgman corrected)', 'Experimental ...
     test (Bridgman corrected)','Adaption to model (not ...
     corrected)', 'Experimental test (not ...
     corrected)','Location','SouthEast')
xlabel('Plastic strain, \epsilon_p')
ylabel('True stress, \sigma_t (MPa)')
axis([0 ceil(max(plastic_strain_new)*1.1/0.01)*0.01 0 ...
     ceil(max(voce_stress)*1.1/10)*10])

%% Saving last figure to directory
path='C:\Users\Eivind\Documents\MATLAB\Post-process ...
     SHTB\plots_camera';
cd(path)
filename=sprintf('Test-0%g-(true_stress,camera,uncorr_and_corr).eps',...
    test_id);
save_figure_small(filename,hFig,hAxes)

%% Plotting the resulting true-stress-plastic strain curve for ...
     corrected stress
hFig=figure;
hAxes=axes;
hold on
plot(plastic_strain_new,voce_stress_corr,'b','LineWidth',2)
plot(plastic_strain,true_plastic_stress_corr,'r','LineWidth',1)
legend('Adaption to model', 'Experimental ...
     test','Location','SouthEast')
xlabel('Plastic strain, \epsilon_p')
% ylabel('True stress, \sigma_t (MPa)')
axis([0 ceil(max(plastic_strain_new)*1.1/0.01)*0.01 0 ...
     ceil(max(voce_stress)*1.1/10)*10])

%% Saving last figure to directory
path='C:\Users\Eivind\Documents\MATLAB\Post-process ...
     SHTB\plots_camera';
cd(path)
filename=sprintf('Test-0%g-(true_stress,camera,corr).eps',test_id);
save_figure_small(filename,hFig,hAxes)
```

```

%% Saving parameters
if test_n==test_tot
    parameters_xlsx_string=arrayfun(@num2str, parameters, 'unif', 0);

    parameters_xlsx = {'Parameters for yield stress + Voce ...
        hardening and max plastic strain at onset of necking', '', ...
        '', '','','','','','','','','','',''; ...
        'Test number' '\sigma_Y (directly)', '\sigma_Y', 'Q_1', ...
            'C_1', 'Q_2', 'C_2', 'Max. plastic strain','\sigma_Y', ...
            'Q_1', 'C_1', 'Q_2', 'C_2'};
    parameters_xlsx(3:size(parameters,1)+2,1:13)=parameters_xlsx_string
    xlswrite('parameters.xlsx', parameters_xlsx);

    save('parameters_cam', 'parameters')
    save('results_cam','results')
end
cd('C:\Users\Eivind\Documents\MATLAB\Post-process SHTB')
```

engs_func_single_shtb.m:

```
function eng_res = engs_func_single_shtb(eng)
global eng_strain_calc eng_stress_calc

eng_calc=eng(1)+eng(2).*eng_strain_calc+eng(3).*eng_strain_calc.^2;
eng_res=abs(eng_calc-eng_stress_calc);
```

voce_test_single_shtb.m:

```
function voce_res = voce_test_single_shtb(voce)
global plastic_strain true_plastic_stress

voce_calc=voce(1)+voce(2).*(1-exp(-voce(3).*plastic_strain))+...
    voce(4).*(1-exp(-voce(5).*plastic_strain));
voce_res=abs(voce_calc-true_plastic_stress);
```

voce_test_single_shtb_corr.m:

```
function voce_res = voce_test_single_shtb_corr(voce)
global plastic_strain true_plastic_stress_corr

voce_calc=voce(1)+voce(2).*(1-exp(-voce(3).*plastic_strain))+...
    voce(4).*(1-exp(-voce(5).*plastic_strain));
voce_res=abs(voce_calc-true_plastic_stress_corr);
```

true_strain_cal_func.m:

```
function true_strain_cal_res = true_strain_cal_func(true_strain_cal)
global true_strain timecam

true_strain_calc=true_strain_cal(1).*(timecam.^7)+...
    true_strain_cal(2).*(timecam.^6)+true_strain_cal(3)...
    .*(timecam.^5)+true_strain_cal(4).*(timecam.^4)+...
    true_strain_cal(5).*(timecam.^3)+true_strain_cal(6)...
    .*(timecam.^2)+true_strain_cal(7).*timecam+true_strain_cal(8);
true_strain_cal_res=abs(true_strain-true_strain_calc);
```

diameter_cal_func.m:

```
function diameter_cal_res = diameter_cal_func(diameter_cal)
global diameter_s timecam

diameter_calc=diameter_cal(1).*(timecam.^7)+diameter_cal(2).*...
    (timecam.^6)+diameter_cal(3).*(timecam.^5)+diameter_cal(4)...
    .*(timecam.^4)+...
    diameter_cal(5).*(timecam.^3)+diameter_cal(6).*(timecam.^2)...
    +diameter_cal(7).*timecam+diameter_cal(8);
diameter_cal_res=abs(diameter_s-diameter_calc);
```

rad_cal_func.m:

```
function rad_cal_res = rad_cal_func(rad_cal)
global rad_curv_data timecam_rad

rad_calc=rad_cal(1).*(timecam_rad.^3)+rad_cal(2).*(timecam_rad.^2)+...
    rad_cal(3).*timecam_rad+rad_cal(4);
rad_cal_res=abs(rad_curv_data-rad_calc);
```

D.3 Post-Processing of Data from Simulations in LS-DYNA

Script name	Script/function	Explanation
elout_nodout.m	Script	Importing data from LS-DYNA simulations

Table D.3: Overview of Matlab scripts for post-processing of data from LS-DYNA simulations

elout_nodout.m:

```
%% Elout
% Script imports elout file from LS—DYNA simulations and saves the ...
    data to variables

%% Calculations
fid = fopen('elout');
fseek(fid, 0, 'eof');
endpos = ftell(fid);
fseek(fid, 0, 'bof');
z=1;
k=0; % k defines time step k

while  ftell(fid) ~= endpos;
  if z==0
  break
  else
    tline = fgetl(fid);
    nn=size(tline);
    if nn(1,2) >= 17
        if strcmp(tline(1:16),' e l e m e n t  ')==1
            infot = sscanf(tline,' e l e m e n t   s t r e s s   c ...
                a l c u l a t i o n s   f o r   t i m e  s t e p  ...
                %g  ( at time  %g )');
            k=k+1;
            t(k)=infot(2);
            elnum=1;
        elseif strcmp(tline(1:16),'   1— 15 elastic')==1
            infosig = sscanf(tline,'   1— 15 elastic   %g  %g  %g  ...
                %g  %g  %g    %g');
            sig(k,:,elnum)=infosig;
            elnum=elnum+1;
        elseif strcmp(tline(1:10),' lower ipt')==1
            infoeps = sscanf(tline,' lower ipt          %g  %g  %g  ...
                %g  %g  %g');
            eps(k,:,elnum)=infoeps;
        elseif strcmp(tline(1:10),' upper ipt')==1
            infoepsU = sscanf(tline,' upper ipt          %g  %g  %g ...
                 %g  %g  %g');
            epsU(k,:,elnum)=infoepsU;
            elnum=elnum+1;
        elseif strcmp(tline(1:17),' strains (global)')==1
            elnum=1;
```

```
        else
        end
    end
   end
end
fclose(fid);
save('elout_nodout')
```

D.4 Calibration of Material Models

Script name	Script/function	Explanation
matmod_cal_MJC_hard.m	Script	Calibration of hardening part for MJC model
matmod_cal_MJC_sr.m	Script	Calibration of strain rate part for MJC model
matmod_cal_MJC_temp.m	Script	Calibration of temperature part for MJC model
matmod_cal_ZA_yield.m	Script	Calibration of yield function for ZA model
matmod_cal_ZA_hard.m	Script	Calibration of hardening function for ZA model
matmod_cal_comb_yield.m	Script	Calibration of yield function for comb. model
matmod_cal_comb_hard.m	Script	Calibration of hardening function for comb. model
matmod_cal_comb_initial.m	Script	Calculation of parameters used for calibration
MJC_hard.m	Function	Calculation of hardening parameters for MJC model
MJC_sr.m	Function	Calculation of strain rate parameter for MJC model
MJC_temp.m	Function	Calculation of temperature parameter for MJC model
ZA_yield.m	Function	Calculation of yield function parameters for ZA model
ZA_hard.m	Function	Calculation of hardening function parameters for ZA model
comb_yield.m	Function	Calculation of yield function parameters for comb. model
comb_hard.m	Function	Calculation of hardening function parameters for comb. model

Table D.4: Overview of Matlab scripts for calibration of material models

matmod_cal_MJC_hard.m:

```
clear all
%% Loads the database
load('Database')
global r strain_r strain_r_c n_m n_m_c sig_db sig_db_c eps_0_dot ...
    test_n n_tests test_id n_strain

%% Manual input
eps_0_dot=0.01; %reference strain rate
xsi=0.9;
rho=2.7E-9;
Cp=9.6E+8;
Tr=293;
Tm=933;
test_n=[13 16]; %tests to fit for
n_strain=1000; %number of strain values
lol=0;

%% Calling script for initial calculations
matmod_cal_initial

%% Starting guess MJC parameters
A_MJC_0=103.669998; Q1_MJC_0=80.78500; C1_MJC_0=11.1749780; ...
    Q2_MJC_0=100.277969; C2_MJC_0=15.462350;
MJC_0=[A_MJC_0, Q1_MJC_0, C1_MJC_0 Q2_MJC_0 C2_MJC_0];
lowerb_MJC=[0 0 0 0 0];
upperb_MJC=[1000 1000 1000 1000 1000];
```

```

%% Fitting of MJC material model using lsqnonlin
options = optimset('TolFun', 1e-1000, 'TolX', 1e-1000, ...
    'MaxFunEvals',100000,'MaxIter',100000, 'PlotFcns', ...
    @optimplotresnorm);
[MJC_cal(1:5),MJC_cal(6)] = ...
    lsqnonlin(@MJC_hard,MJC_0,lowerb_MJC,upperb_MJC,options);

%% Calculation stress values for resulting calibrated material ...
    models (MJC)
for i = 1:size(r,2)
    sigy_MJC(:,i)=MJC_cal(1)+MJC_cal(2).*(1-exp(-MJC_cal(3).*...
        r(:,i)))+MJC_cal(4).*(1-exp(-MJC_cal(5).*r(:,i)));
end

%% Plotting of resulting material models (MJC) comparison
for i = 1:length(test_n);
    test_id=test_n(i);
    figure
    plot(r(:,test_id),sig_db(:,test_id),r(:,test_id),...
        sigy_MJC(:,test_id))
    ylim([0 400])
    xlabel('Strain')
    ylabel('Stress (MPa)')
    legend('Stress-strain from database', 'Stress-strain MJC ...
        (calibrated)', 'Location', 'NorthEast')
    legend BOXOFF
    str = sprintf('Model %g',test_id);
    title(str);
end
```

matmod_cal_MJC_sr.m:

```
clear all
%% Loads the database
load('Database')
global r strain_r strain_r_c n_m n_m_c sig_db sig_db_c eps_0_dot ...
    test_n n_tests test_id n_strain A_MJC_0 Q1_MJC_0 C1_MJC_0 ...
    Q2_MJC_0 C2_MJC_0

%% Manual input
eps_0_dot=0.01; %reference strain rate
xsi=0.9;
rho=2.7E-9;
Cp=9.6E+8;
Tr=293;
Tm=933;
test_n=[1 2 13 16 23];
n_strain=1000; %number of strain values
lol=0;

%% Pre-calibrated parameters
A_MJC_0=83.1606; Q1_MJC_0=351.5559; C1_MJC_0=0.951; ...
    Q2_MJC_0=74.1947; C2_MJC_0=20.6163;

%% Calling script for initial calculations
```

```
matmod_cal_initial

%% Starting guess MJC parameters
C_MJC_0=0.0500;
MJC_0=[C_MJC_0];
lowerb_MJC=[0];
upperb_MJC=[10];

%% Fitting of MJC material model using lsqnonlin
options = optimset('TolFun', 1e-1000, 'TolX', 1e-1000, ...
    'MaxFunEvals',100000,'MaxIter',100000, 'PlotFcns', ...
    @optimplotresnorm);
[MJC_cal(1),MJC_cal(2)] = ...
    lsqnonlin(@MJC_sr,MJC_0,lowerb_MJC,upperb_MJC,options);

%% Calculation stress values for resulting calibrated material ...
    models (MJC)
for i = 1:size(r,2)
    sigy_MJC(:,i)=(A_MJC_0+Q1_MJC_0.*(1-exp(-C1_MJC_0.*r(:,i)))+...
        Q2_MJC_0.*(1-exp(-C2_MJC_0.*r(:,i)))).*...
        (1+strain_r(i)/eps_0_dot).^MJC_cal(1);
end

%% Plotting of resulting material models (MJC) comparison
for i = 1:length(test_n);
    test_id=test_n(i);
    figure
    plot(r(:,test_id),sig_db(:,test_id),r(:,test_id),...
        sigy_MJC(:,test_id))
    ylim([0 400])
    xlabel('Strain')
    ylabel('Stress (MPa)')
    legend('Stress-strain from database', 'Stress-strain MJC ...
        (calibrated)', 'Location', 'NorthEast')
    legend BOXOFF
    str = sprintf('Model %g',test_id);
    title(str);
end
```

matmod_cal_MJC_temp.m:

```
clear all
%% Loads the database
load('Database')
global r strain_r strain_r_c T n_m n_m_c sig_db sig_db_c eps_0_dot ...
    T_homo test_n n_tests test_id n_strain A_MJC_0 Q1_MJC_0 ...
    C1_MJC_0 Q2_MJC_0 C2_MJC_0 C_MJC_0

%% manual input
eps_0_dot=0.01; %reference strain rate
xsi=0.9;
rho=2.7E-9;
Cp=9.6E+8;
Tr=293;
Tm=933;
test_n=[1:20];
```

```
n_strain=1000; %number of strain values
lol=0;

%% Pre-calibrated parameters
A_MJC_0=83.1606; Q1_MJC_0=351.5559; C1_MJC_0=0.951; ...
    Q2_MJC_0=74.1947; C2_MJC_0=20.6163; C_MJC_0=0.0022;

%% Calling script for initial calculations
matmod_cal_initial

%% Starting guess MJC parameters
m_MJC_0=0.900;
MJC_0=[m_MJC_0];
lowerb_MJC=[0];
upperb_MJC=[10];

%% Fitting of MJC material model using lsqnonlin
options = optimset('TolFun', 1e-1000, 'TolX', 1e-1000, ...
    'MaxFunEvals',100000,'MaxIter',100000, 'PlotFcns', ...
    @optimplotresnorm);
[MJC_cal(1),MJC_cal(2)] = ...
    lsqnonlin(@MJC_temp,MJC_0,lowerb_MJC,upperb_MJC,options);

%% Calculation stress values for resulting calibrated material ...
    models (MJC)
for i = 1:size(r,2)
    sigy_MJC(:,i)=(A_MJC_0+Q1_MJC_0.*(1-exp(-C1_MJC_0.*r(:,i)))+...
        Q2_MJC_0.*(1-exp(-C2_MJC_0.*r(:,i)))).*...
        ((1+strain_r(i)/eps_0_dot).^C_MJC_0).*...
        (1-T_homo(:,i).^MJC_cal(1));
end

%% Plotting of resulting material models (MJC) comparison
for i = 1:length(test_n);
    test_id=test_n(i);
    figure
    plot(r(:,test_id),sig_db(:,test_id),r(:,test_id),...
        sigy_MJC(:,test_id))
    ylim([0 400])
    xlabel('Strain')
    ylabel('Stress (MPa)')
    legend('Stress-strain from database', 'Stress-strain MJC ...
        (calibrated)', 'Location', 'NorthEast')
    legend BOXOFF
    str = sprintf('Model %g',test_id);
    title(str);
end
```

matmod_cal_ZA_yield.m:

```
clear all
%% Loads the database
format long
load('Database')
global r strain_r strain_r_c T n_m n_m_c sig_db sig_db_c eps_0_dot ...
    test_n n_tests n_strain y_test
```

```
%% Manual input
eps_0_dot=0.01; %reference strain rate
xsi=0.9;
rho=2.7E-9;
Cp=9.6E+8;
Tr=293;
Tm=933;
test_n=[1:32];
n_strain=1000; %number of strain values
lol=0;

%% Calling script for initial calculations
matmod_cal_initial

%% Starting guess ZA parameters
siga_ZA_0=8; B_ZA_0=100; beta0_ZA_0=0.00012134; beta1_ZA_0=.000006434;
ZA_0=[siga_ZA_0 B_ZA_0 beta0_ZA_0 beta1_ZA_0];
lowerb_ZA=[0 0 0 0 ];
upperb_ZA=[50 1000 1 1];

%% Fitting of ZA material model using lsqnonlin
options = optimset('TolFun', 1e-1000, 'TolX', 1e-1000, ...
    'MaxFunEvals',100000,'MaxIter',100000, 'PlotFcns', ...
    @optimplotresnorm);
[ZA_cal(1:4),ZA_cal(5)] = ...
    lsqnonlin(@ZA_yield,ZA_0,lowerb_ZA,upperb_ZA,options);

%% Calculation yield stress values for resulting calibrated ...
    material models (ZA)
for i = 1:size(r,2)
    sigy_ZA_yield(i)=ZA_cal(1)+ZA_cal(2)*exp(-(ZA_cal(3)-...
        ZA_cal(4)*log(strain_r(i)))*T(1,i));
end

%% Plotting of calculated yield stress compared to tests
figure
for i = 1:length(test_n)
    hold on
    plot([test_n(i) test_n(i)],[sigy_ZA_yield(test_n(i)) ...
        y_test(test_n(i))])
end
hold on
for i = 1:length(test_n)
    scatter(test_n(i),sigy_ZA_yield(test_n(i)),50,[0.5 0 0],'+')
    scatter(test_n(i),y_test(test_n(i)),50,[0 .5 0],'+')
end
title('Calibrated yield stress \sigma_Y for ZA model');
xlabel('Test number (32 total)');
ylabel('Yield stress, \sigma_Y  (MPa)');
```

matmod_cal_ZA_hard.m:

```
clear all
%% Loads the database
load('Database')
```

```
global r strain_r strain_r_c T n_m n_m_c sig_db sig_db_c eps_0_dot ...
    test_n n_tests test_id n_strain y_test siga_ZA_0 B_ZA_0 ...
    beta0_ZA_0 beta1_ZA_0

%% manual input
eps_0_dot=0.01; %reference strain rate
xsi=0.9;
rho=2.7E-9;
Cp=9.6E+8;
Tr=293;
Tm=933;
test_n=[1:32];
n_strain=1000; %number of strain values
lol=0;

%% Calling script for initial calculations
matmod_cal_initial

%% Pre-calibrated parameters
siga_ZA_0=1.287971; B_ZA_0=343.597; beta0_ZA_0=0.004419; ...
    beta1_ZA_0=0.0000865;

%% Starting guess ZA parameters
A_ZA_0=150; n_ZA_0=0.5; alpha0_ZA_0=0.001; alpha1_ZA_0=0.00001;
ZA_0=[A_ZA_0 n_ZA_0 alpha0_ZA_0 alpha1_ZA_0];
lowerb_ZA=[0 0 0 0 ];
upperb_ZA=[50000 10 1 1];

%% Fitting of ZA material model using lsqnonlin
options = optimset('TolFun', 1e-1000, 'TolX', 1e-1000, ...
    'MaxFunEvals',100000,'MaxIter',100000, 'PlotFcns', ...
    @optimplotresnorm);
[ZA_cal(1:4),ZA_cal(5)] = ...
    lsqnonlin(@ZA_hard,ZA_0,lowerb_ZA,upperb_ZA,options);

%% Calculation stress values for resulting calibrated material ...
    models (ZA)
for i = 1:size(r,2)
    sigy_ZA(:,i)=siga_ZA_0+B_ZA_0.*exp(-(beta0_ZA_0-beta1_ZA_0.*...
        log(strain_r(i))).*T(1,i))+ZA_cal(1).*(r(:,i).^...
        ZA_cal(2)).*exp(-(ZA_cal(3)-ZA_cal(4).*...
        log(strain_r(i))).*T(:,i));
end

%% Plotting of resulting material models (ZA) comparison
for i = 1:length(test_n);
    test_id=test_n(i);
    figure
    plot(r(:,test_id),sig_db(:,test_id),...
        r(:,test_id),sigy_ZA(:,test_id))
    ylim([0 400])
    xlabel('Strain')
    ylabel('Stress (MPa)')
    legend('Stress-strain from database', 'Stress-strain ZA ...
        (calibrated)', 'Location', 'NorthEast')
    legend BOXOFF
    str = sprintf('Model %g',test_id);
    title(str);
end
```

matmod_cal_comb_yield.m:

```
clear all
%% Loads the database
load('Database')
global r strain_r strain_r_c T n_m n_m_c sig_db sig_db_c test_n ...
    n_tests n_strain y_test p q

%% Manual input
xsi=0.9;
rho=2.7E-9;
Cp=9.6E+8;
Tr=293;
Tm=933;
test_n=[1:32];
n_strain=1000; %number of strain values
lol=0;

%% Calling script for initial calculations
matmod_cal_initial

%% Starting guess comb. parameters
Y_a_0=139; Y_threshold_0=1100; beta_1_Y_0=0.00121; ...
    beta_2_Y_0=0.0000618;
q=1.5; p=0.5;
comb_0=[Y_a_0 Y_threshold_0 beta_1_Y_0 beta_2_Y_0];
lowerb_comb=[0 0 0 0];
upperb_comb=[1000 10000 1 1];

%% Fitting of comb. material model using lsqnonlin
options = optimset('TolFun', 1e-1000, 'TolX', 1e-1000, ...
    'MaxFunEvals',100000,'MaxIter',100000, 'PlotFcns', ...
    @optimplotresnorm);
[comb_cal(1:4),comb_cal(5)] = ...
    lsqnonlin(@comb_yield,comb_0,lowerb_comb,upperb_comb,options);

%% Calculation yield stress values for resulting calibrated ...
    material models (comb)
for i = 1:size(r,2)
    sigy_comb_yield(i)=comb_cal(1)+comb_cal(2)*(1-(comb_cal(3)*...
        T(1,i)-comb_cal(4)*T(1,i)*log(strain_r(i)))^(1/q))^(1/p);
end

%% Plotting of calculated yield stress compared to tests
figure
for i = 1:length(test_n)
    hold on
    plot([test_n(i) test_n(i)],[sigy_comb_yield(test_n(i)) ...
        y_test(test_n(i))])
end
hold on
for i = 1:length(test_n)
    scatter(test_n(i),sigy_comb_yield(test_n(i)),50,[0.5 0 0],'+')
    scatter(test_n(i),y_test(test_n(i)),50,[0 .5 0],'+')
end
title('Calibrated yield stress \sigma_Y for comb. model');
xlabel('Test number (32 total)');
ylabel('Yield stress, \sigma_Y  (MPa)');
```

matmod_cal_comb_hard.m:

```
clear all
%% Loads the database
load('Database')
global r strain_r strain_r_c T n_m n_m_c sig_db sig_db_c test_n ...
    n_tests n_strain y_test p q Y_a Y_threshold beta_1_Y beta_2_Y

%% Manual input
xsi=0.9;
rho=2.7E-9;
Cp=9.6E+8;
Tr=293;
Tm=933;
test_n=[1:32];
n_strain=1000; %number of strain values
lol=0;

%% Calling script for initial calculations
matmod_cal_initial

%% Pre-calibrated parameters
Y_a=8.4194; Y_threshold=321.1257; beta_1_Y=0.0011269; ...
    beta_2_Y=0.000019808;

%% Starting guess comb. parameters
B1_0=800; n1_0=0.45; B2_0=2190; n2_0=0.71; beta_1_H_0=0.00113; ...
    beta_2_H_0=0.000051;
q=1.5; p=0.5;
comb_0=[B1_0 n1_0 B2_0 n2_0 beta_1_H_0 beta_2_H_0];
lowerb_comb=[0 0 0 0 0 0];
upperb_comb=[5000 10 20000 10 1 1];

%% Fitting of comb. material model using lsqnonlin
options = optimset('TolFun', 1e-1000, 'TolX', 1e-1000, ...
    'MaxFunEvals',100000,'MaxIter',100000, 'PlotFcns', ...
    @optimplotresnorm);
[comb_cal(1:6),comb_cal(7)] = ...
    lsqnonlin(@comb_hard,comb_0,lowerb_comb,upperb_comb,options);

%% Calculation of stress values for resulting calibrated material ...
    models (comb)
for i = 1:size(r,2)
    sigy_comb(:,i)=Y_a+Y_threshold*(1-(beta_1_Y*T(1,i)-beta_2_Y*...
        T(1,i)*log(strain_r(i)))^(1/q))^(1/p)+...
        comb_cal(1).*r(:,i).^comb_cal(2)+comb_cal(3).*(r(:,i).^...
        comb_cal(4)).*(1-(comb_cal(5).*T(:,i)-comb_cal(6).*T(:,i).*...
        log(strain_r(i))).^(1/q)).^(1/p);
end

%% Plotting of resulting material models (comb.) comparison
for i = 1:length(test_n);
    test_id=test_n(i);
    figure
    plot(r(:,test_id),sig_db(:,test_id),r(:,test_id),...
        sigy_comb(:,test_id))
    ylim([0 400])
    xlabel('Strain')
```

```
    ylabel('Stress (MPa)')
    legend('Stress–strain from database', 'Stress–strain comb. ...
        (calibrated)', 'Location', 'NorthEast')
    legend BOXOFF
    str = sprintf('Model %g',test_id);
    title(str);
end
```

matmod_cal_initial.m:

```
%% Matmod_cal_initial
% Database = db_table2013_03_21
if lol==1
test_n=1;
else
end

%% Various definitions
n_m=size(db_table2013_03_21,1); %number of tests (measured with ...
    strain gauges)
n_m_c=size(db_table2013_03_21C,1); %number of tests (measured with ...
    camera)
n_tests=length(test_n);

%% Assigne values to variables (from strain gauges)
test_n_db=db_table2013_03_21(:,1);
strain_r=db_table2013_03_21(:,2);
temp=db_table2013_03_21(:,3);
Ya=db_table2013_03_21(:,4);
Q1=db_table2013_03_21(:,5);
C1=db_table2013_03_21(:,6);
H=db_table2013_03_21(:,7);
eps_true_max=db_table2013_03_21(:,8);
eps_fracture=db_table2013_03_21(:,9);

%% assigne values to variables (from camera)
test_n_db_c=db_table2013_03_21C(:,1);
strain_r_c=db_table2013_03_21C(:,2);
temp_c=db_table2013_03_21C(:,3);
Ya_c=db_table2013_03_21C(:,4);
Q1_c=db_table2013_03_21C(:,5);
C1_c=db_table2013_03_21C(:,6);
Q2_c=db_table2013_03_21C(:,8);
C2_c=db_table2013_03_21C(:,9);
H_c=db_table2013_03_21C(:,7);
eps_true_max_c=db_table2013_03_21C(:,10);
eps_fracture_c=db_table2013_03_21C(:,11);

%% assigning strain rates
y_test=Ya;
y_test(n_m+1:n_m+n_m_c)=Ya_c;
strain_r(n_m+1:n_m+n_m_c)=strain_r_c;

%% calculation of strain matrix
for i = 1:n_m
    r(:,i)=linspace(0,eps_true_max(i),n_strain);
```

```
end
for i = 1:n_m_c
    r_c(:,i)=linspace(0,eps_true_max_c(i),n_strain);
end
r(:,n_m+1:n_m+n_m_c)=r_c;

%% calculation of stress values from database material models
for i = 1:n_m
    sig_db(:,i)=Ya(i)+Q1(i)*(1-exp(-C1(i)*r(:,i))) + H(i)*r(:,i);
end
for i = 1:n_m_c
    sig_db_c(:,i)=Ya_c(i)+Q1_c(i)*(1-exp(-C1_c(i)*r_c(:,i)))+...
        Q2_c(i)*(1-exp(-C2_c(i)*r_c(:,i))) + H_c(i)*r_c(:,i);
end
sig_db(:,n_m+1:n_m+n_m_c)=sig_db_c;

%% calculation of temperature and homologous temperature
T(1,:)=temp;
T(1,n_m+1:n_m+n_m_c)=temp_c;

for i = 2:n_strain
    T(i,:)=T(i-1,:)+xsi.*sig_db(i,:).*(r(i,:)-r(i-1,:))./rho./Cp;
end

T_homo=(T-Tr)/(Tm-Tr);
```

MJC_hard.m:

```
function MJC_res = MJC_hard(MJC_p)
global r sig_db test_n n_tests

for k = 1:n_tests
    sigy_MJC_calc(:,k)=MJC_p(1)+MJC_p(2).*(1-exp(-MJC_p(3).*...
        r(:,test_n(k))))+MJC_p(4).*(1-exp(-MJC_p(5).*...
        r(:,test_n(k))));
    MJC_res(:,k)=abs((sig_db(:,test_n(k))-sigy_MJC_calc(:,k))).*...
        100./sig_db(:,test_n(k));
end
```

MJC_sr.m:

```
function MJC_res = MJC_sr(MJC_p)
global r sig_db test_n n_tests A_MJC_0 Q1_MJC_0 C1_MJC_0 Q2_MJC_0 ...
    C2_MJC_0 eps_0_dot strain_r

for k = 1:n_tests
    sigy_MJC_calc(:,k)=(A_MJC_0+Q1_MJC_0.*(1-exp(-C1_MJC_0.*...
        r(:,test_n(k))))+Q2_MJC_0.*(1-exp(-C2_MJC_0.*...
        r(:,test_n(k))))).*(1+strain_r(test_n(k))/...
        eps_0_dot).^MJC_p;
    MJC_res(:,k)=abs((sig_db(:,test_n(k))-sigy_MJC_calc(:,k))).*...
        100./sig_db(:,test_n(k));
end
```

MJC_temp.m:

```
function MJC_res = MJC_temp(MJC_p)
global r sig_db test_n n_tests A_MJC_0 Q1_MJC_0 C1_MJC_0 Q2_MJC_0 ...
    C2_MJC_0 eps_0_dot strain_r T_homo C_MJC_0

for k = 1:n_tests
    sigy_MJC_calc(:,k)=(A_MJC_0+Q1_MJC_0.*(1-exp(-C1_MJC_0.*...
        r(:,test_n(k))))+Q2_MJC_0.*(1-exp(-C2_MJC_0.*...
        r(:,test_n(k))))).*((1+strain_r(test_n(k))/...
        eps_0_dot).^C_MJC_0).*(1-T_homo(:,test_n(k))...
        .^MJC_p);
    MJC_res(:,k)=abs((sig_db(:,test_n(k))-sigy_MJC_calc(:,k))).*...
        100./sig_db(:,test_n(k));
end
```

ZA_yield.m:

```
function ZA_res = ZA_yield(ZA_p)
global test_n n_tests strain_r T y_test

for k = 1:n_tests
    sigy_ZA_calc(k)=ZA_p(1)+ZA_p(2)*exp(-(ZA_p(3)-ZA_p(4)*...
        log(strain_r(test_n(k))))*T(1,test_n(k)));
    ZA_res(k)=abs((y_test(test_n(k))-sigy_ZA_calc(k)))*...
        100/y_test(test_n(k));
end
```

ZA_hard.m:

```
function ZA_res = ZA_hard(ZA_p)
global test_n n_tests strain_r T siga_ZA_0 B_ZA_0 beta0_ZA_0 ...
    beta1_ZA_0 r sig_db

for k = 1:n_tests
    sigy_ZA_calc(:,k)=siga_ZA_0+B_ZA_0.*exp(-(beta0_ZA_0-beta1_ZA_0...
        .*log(strain_r(test_n(k)))).*T(1,test_n(k)))+ZA_p(1).*...
        (r(:,test_n(k)).^ZA_p(2)).*exp(-(ZA_p(3)-ZA_p(4).*...
        log(strain_r(test_n(k)))).*T(:,test_n(k)));
    ZA_res(:,k)=abs((sig_db(:,test_n(k))-sigy_ZA_calc(:,k))).*...
        100./sig_db(:,test_n(k));
end
```

comb_yield.m:

```
function comb_res = comb_yield(comb_p)
global test_n n_tests strain_r T y_test q p
```

```

for k = 1:n_tests
    sigy_comb_calc(k)=comb_p(1)+comb_p(2)*(1-(comb_p(3)*...
        T(1,test_n(k))-comb_p(4)*T(1,test_n(k))*...
        log(strain_r(test_n(k))))^(1/q))^(1/p);
    comb_res(k)=abs((y_test(test_n(k))-sigy_comb_calc(k)))*...
        100/y_test(test_n(k));
end
```

comb_hard.m:

```
function comb_res = comb_hard(comb_p)
global test_n n_tests strain_r T q p Y_a Y_threshold beta_1_Y ...
    beta_2_Y r sig_db

for k = 1:n_tests
    sigy_comb_calc(:,k)=Y_a+Y_threshold*(1-(beta_1_Y*...
        T(1,test_n(k))-beta_2_Y*T(1,test_n(k))*...
        log(strain_r(test_n(k))))^(1/q))^(1/p)+...
        comb_p(1).*r(:,test_n(k)).^comb_p(2)+comb_p(3).*...
        (r(:,test_n(k)).^comb_p(4)).*(1-(comb_p(5).*...
        T(:,test_n(k))-comb_p(6).*T(:,test_n(k)).*...
        log(strain_r(test_n(k)))).^(1/q)).^(1/p);
comb_res(:,k)=abs((sig_db(:,test_n(k))-sigy_comb_calc(:,k))).*...
    100./sig_db(:,test_n(k));
end
```

E Experimental Results from Quasi-Static Tests

E.1 AA6060-OLD

Table E.1 and E.2 show respectively an overview of the experimental quasi-static tests and the resulting true stress-plastic strain parameters and estimated fracture strain. The true stress-plastic strain parameters are fitted with Voce rule, $\sigma = \sigma_Y + \sum_{i=1}^{2} Q_i(1 - e^{-C_i \varepsilon_p})$. In Table E.2, the measured yield stress from experimental data is denoted $\sigma_{Y,experiment}$, while the yield stress fitted with Voce rule is denoted σ_Y. The plastic strain value at onset of necking is denoted $\varepsilon_{p,max}$ and the estimated fracture strain is denoted ε_f.

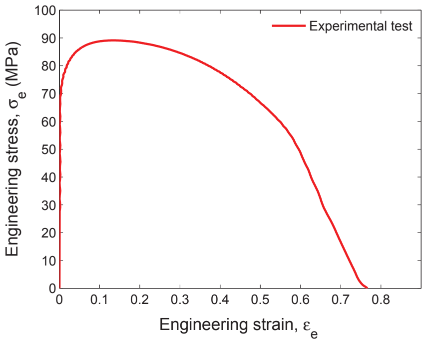
AA6060-OLD

Test #	Strain rate (s^{-1})	Temp. ($^\circ K$)	Diameter (mm)	Gauge length (mm)	Comment
1	1	523	3.00	5	OK
2	1	523	3.02	5	OK
3	1	573	3.02	5	OK
4	0.01	523	3.03	10	OK
5	0.01	523	3.03	10	Aborted
6	0.01	573	3.01	10	OK
7	0.01	573	3.00	10	OK

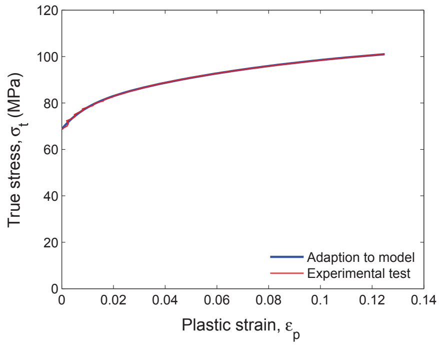
Table E.1: Overview of experimental quasi-static tests for AA6060-OLD

AA6060-OLD								
$Test\ \#$	$\sigma_{Y,experiment}$ (MPa)	σ_Y (MPa)	Q_1 (MPa)	C_1	Q_2 (MPa)	C_2	$\varepsilon_{p,max}$	ε_f
1	68.77	68.76	31.30	10.76	9.11	112.51	0.1249	1.604
2	67.71	68.08	33.74	10.63	9.48	152.31	0.1371	1.365
3	54.31	54.23	24.08	10.07	10.01	158.94	0.1246	2.271
4	44.31	48.35	30.37	28.39	13.71	306.48	0.0995	1.946
5	-	-	-	-	-	-	-	-
6	35.17	36.03	17.34	26.97	11.47	316.29	0.0900	2.537
7	32.62	34.24	17.62	28.81	9.28	251.13	0.0835	3.186
1-2	68.24	68.43	32.50	10.80	9.17	132.63	0.1249	-
6-7	33.89	35.15	17.50	28.02	10.32	286.27	0.0835	-

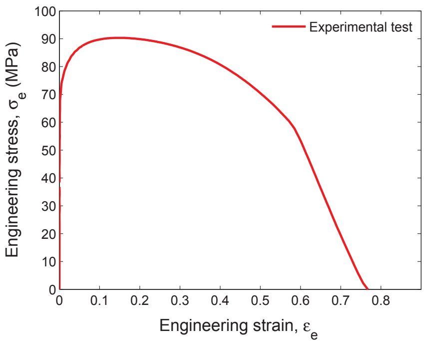
Table E.2: Resulting true stress-plastic strain parameters and estimated fracture strain for AA6060-OLD from quasi-static experiments



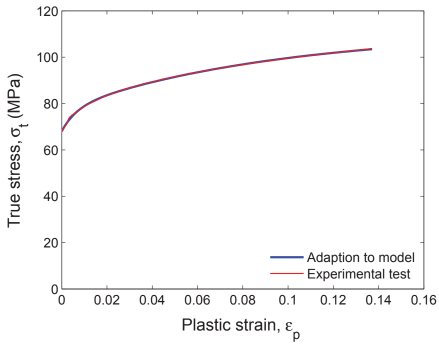
(a) Test #1



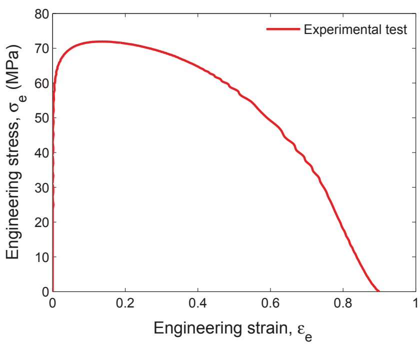
(b) Test #1



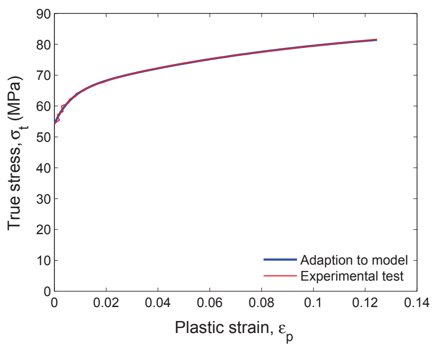
(c) Test #2



(d) Test #2

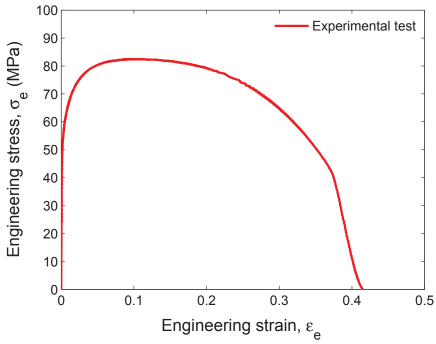


(e) Test #3

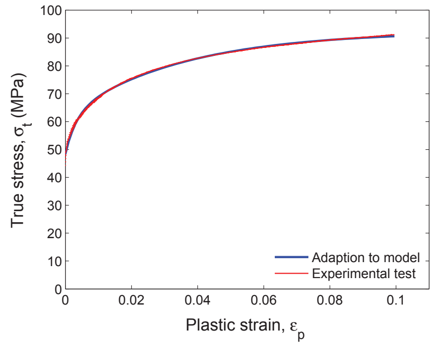


(f) Test #3

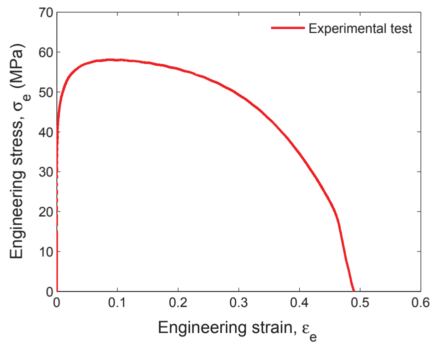
Figure E.1: Continues...



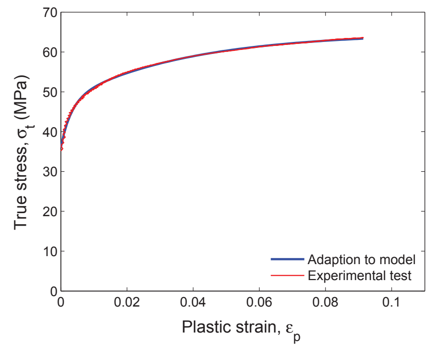
(a) Test #4



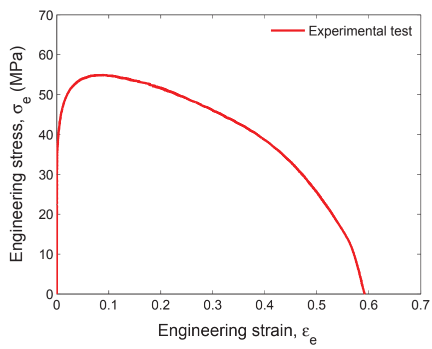
(b) Test #4



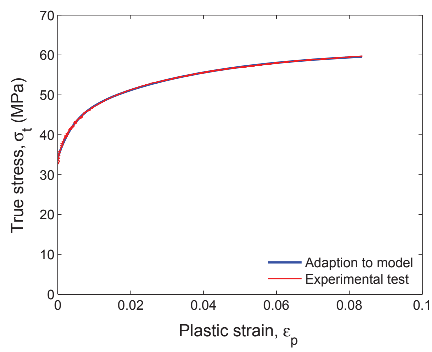
(c) Test #6



(d) Test #6



(e) Test #7



(f) Test #7

Figure E.1: Plots (a)-(f) show the engineering stress-strain curve and true stress-plastic strain curve from quasi-static experiments for AA6060-OLD

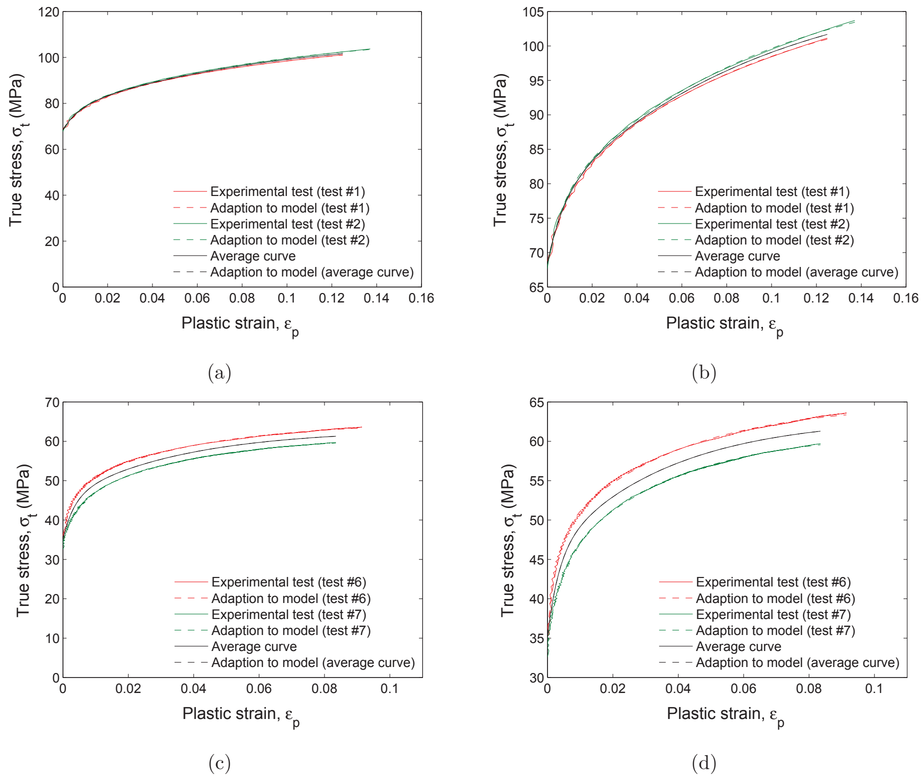


Figure E.2: Plots (a)-(d) show the true stress-plastic strain curve from quasi-static experiments for AA6060-OLD for same boundary conditions together with the average curve. Plots in right column is equal to plots in left column, but for a narrower range of values on the ordinate axis.

E.2 AA6060-L

Table E.3 and E.4 show respectively an overview of the experimental quasi-static tests and the resulting true stress-plastic strain parameters and estimated fracture strain. The true stress-plastic strain parameters are fitted with Voce rule, $\sigma = \sigma_Y + \sum_{i=1}^{2} Q_i(1 - e^{-C_i\varepsilon_p})$. In Table E.4, the measured yield stress from experimental data is denoted $\sigma_{Y,experiment}$, while the yield stress fitted with Voce rule is denoted σ_Y. The plastic strain value at onset of necking is denoted $\varepsilon_{p,max}$ and the estimated fracture strain is denoted ε_f.

AA6060-L

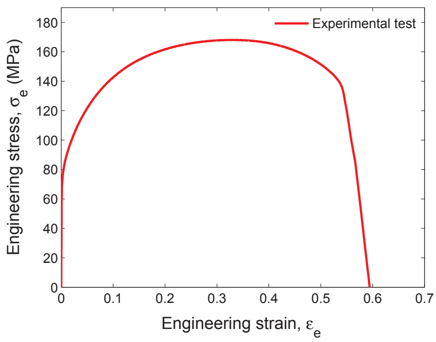
Test #	Strain rate (s^{-1})	Temp. ($^\circ K$)	Diameter (mm)	Gauge length (mm)	Comment
1	0.01	293	not meas.	5	OK
2	0.01	470	not meas.	5	D.s.*
3	0.01	470	not meas.	5	OK
4	0.01	470	not meas.	5	OK
5	0.01	523	not meas.	5	D.s.*
6	0.01	523	not meas.	5	OK
7	0.01	523	not meas.	5	OK
8	0.01	573	not meas.	5	OK
9	0.01	573	not meas.	5	OK
10	0.01	630	not meas.	5	OK
11	0.01	630	not meas.	5	OK
12	1	293	not meas.	5	OK
13	1	473	not meas.	5	OK
14	1	473	not meas.	5	OK
15	1	523	not meas.	5	OK
16	1	523	not meas.	5	OK
17	1	573	not meas.	5	OK
18	1	573	not meas.	5	OK
19	1	630	not meas.	5	OK

* D.s. = Damaged specimen (usually from mounting in the test rig)

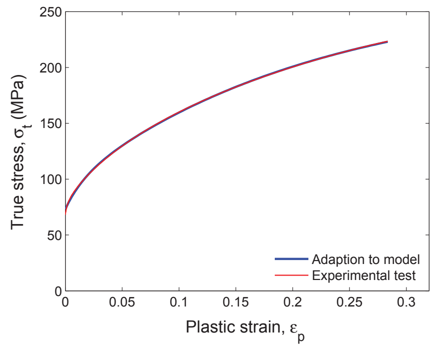
Table E.3: Overview of experimental quasi-static tests for AA6060-L

AA6060-L								
$Test\ \#$	$\sigma_{Y,experiment}$ (MPa)	σ_Y (MPa)	Q_1 (MPa)	C_1	Q_2 (MPa)	C_2	$\varepsilon_{p,max}$	ε_f
1	68.04	73.10	175.72	4.66	21.16	64.12	0.2836	0.812
2	-	-	-	-	-	-	-	-
3	55.14	58.84	182.05	1.42	26.80	56.76	0.2350	1.079
4	51.59	54.58	83.77	4.94	16.52	106.98	0.2349	1.040
5	-	-	-	-	-	-	-	-
6	38.31	38.58	18.71	17.86	10.10	197.46	0.1029	1.862
7	43.36	45.89	53.02	5.53	13.22	144.55	0.2153	1.107
8	30.85	31.55	7.61	25.74	4.76	309.35	0.0682	3.169
9	23.18	23.50	4.96	167.23	3.31	1000.00	0.0158	3.778
10	22.45	22.37	2.45	149.57	3.01	1000.00	0.0199	0.803
11	16.60	16.30	4.27	8.44	5.29	470.89	0.0675	4.797
12	100.64	102.98	134.14	5.43	16.23	117.10	0.2340	0.803
13	75.22	75.84	34.45	19.37	8.11	282.31	0.1012	1.418
14	80.26	80.95	42.73	15.44	11.01	326.47	0.1194	1.360
15	58.88	60.07	26.91	12.93	7.51	183.47	0.1225	1.599
16	55.63	57.26	35.66	7.14	8.28	142.73	0.1703	1.984
17	49.04	50.61	27.02	6.27	5.39	174.42	0.1613	2.619
18	48.58	50.05	26.12	6.87	6.10	111.08	0.1575	2.155
19	34.11	34.67	11.14	19.26	5.41	446.75	0.0994	3.702
3-4	53.36	56.90	100.52	3.36	21.08	72.91	0.2349	-
6-7	40.83	42.27	29.77	10.04	11.80	163.05	0.1029	-
8-9	27.02	27.54	5.48	137.35	2.69	747.74	0.0158	-
10-11	19.52	19.36	2.04	140.63	3.69	682.87	0.0199	-
13-14	77.74	78.40	38.37	17.23	9.57	305.81	0.1012	-
15-16	57.11	58.19	29.36	10.32	8.21	164.56	0.1225	-
17-18	48.81	50.35	26.40	6.68	5.64	138.51	0.1575	-

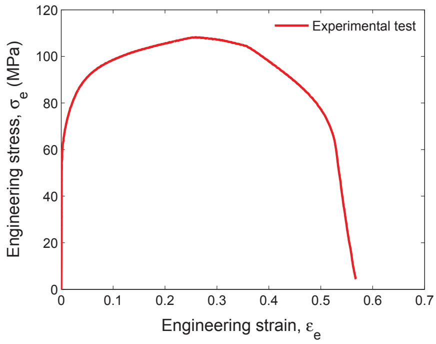
Table E.4: Resulting true stress-plastic strain parameters and estimated fracture strain for AA6060-L from quasi-static experiments



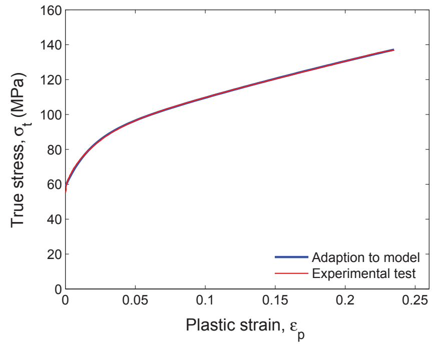
(a) Test #1



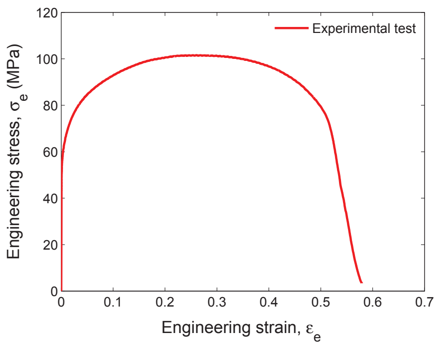
(b) Test #1



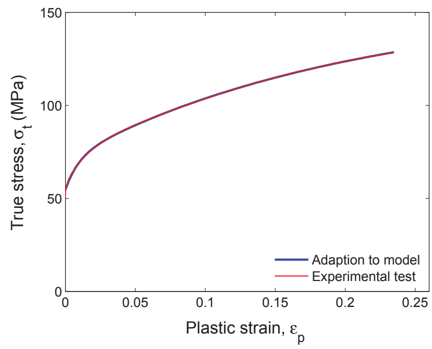
(c) Test #3



(d) Test #3

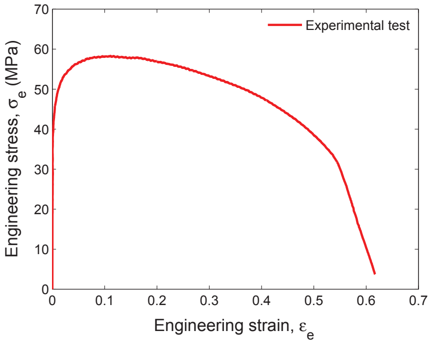


(e) Test #4

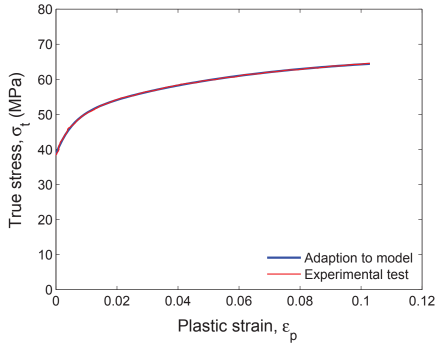


(f) Test #4

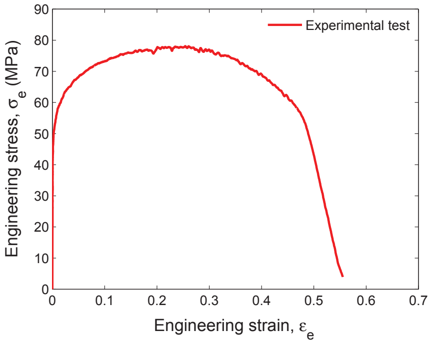
Figure E.3: Continues...



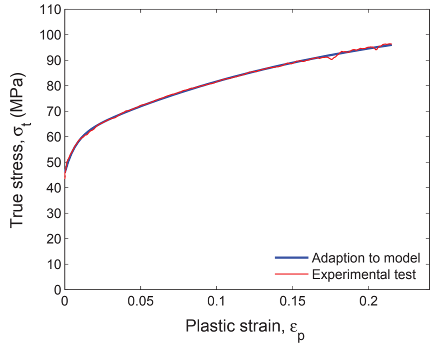
(g) Test #6



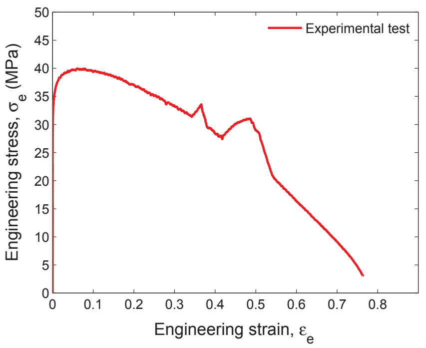
(h) Test #6



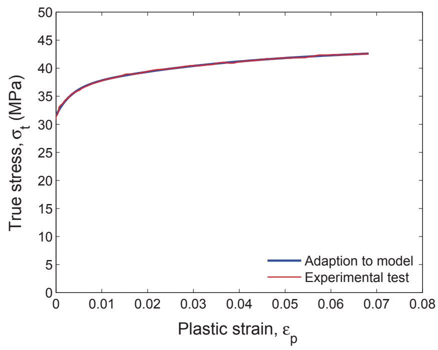
(i) Test #7



(j) Test #7

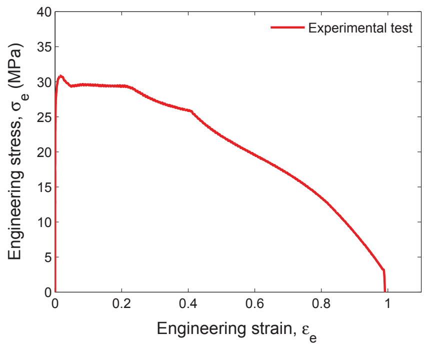


(k) Test #8

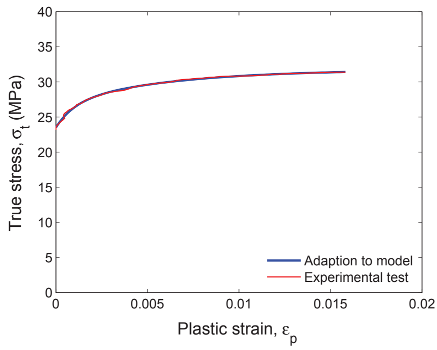


(l) Test #8

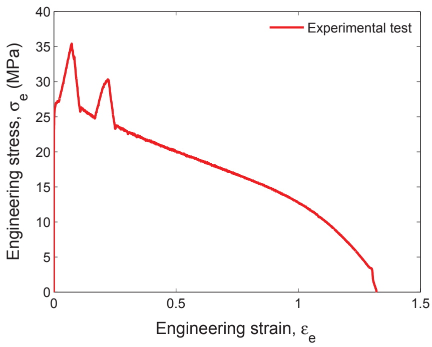
Figure E.3: Continues...



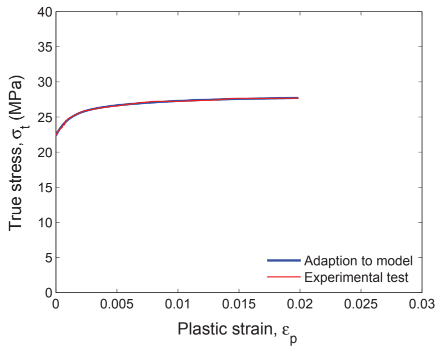
(m) Test #9



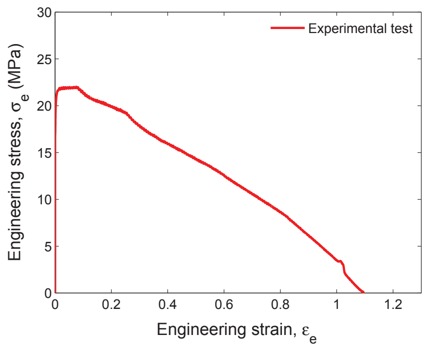
(n) Test #9



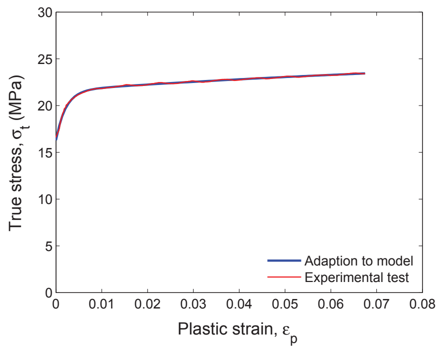
(o) Test #10



(p) Test #10

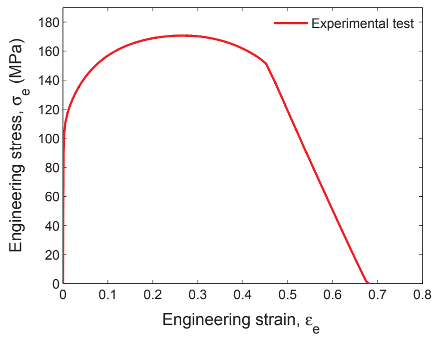


(q) Test #11

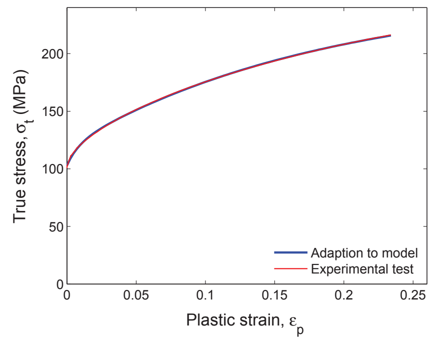


(r) Test #11

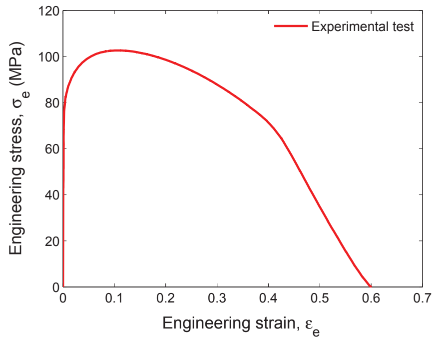
Figure E.3: Continues...



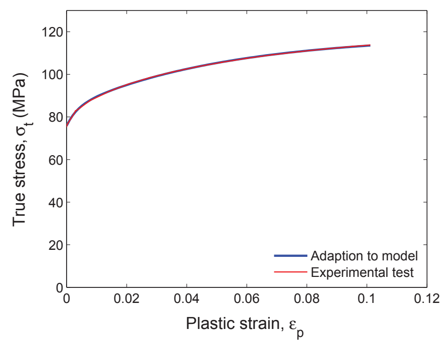
(s) Test #12



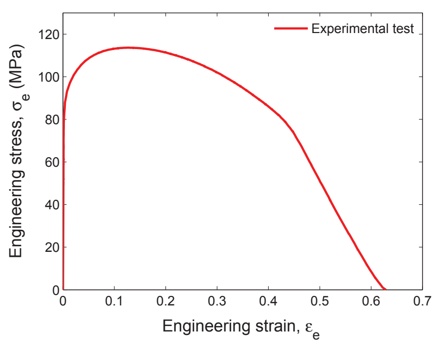
(t) Test #12



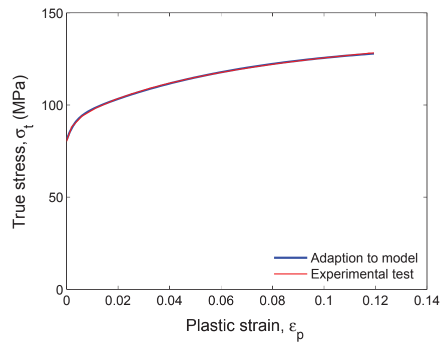
(u) Test #13



(v) Test #13

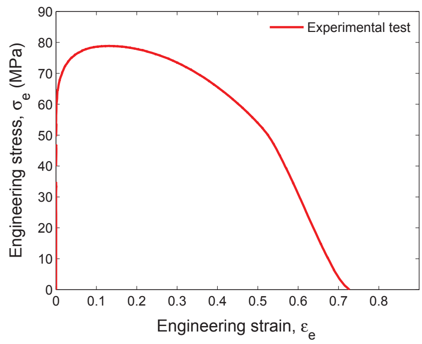


(w) Test #14

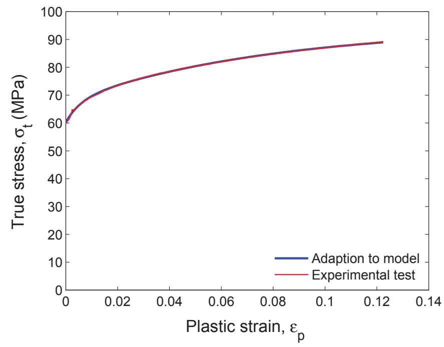


(x) Test #14

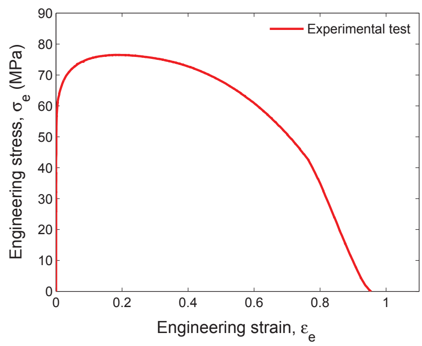
Figure E.3: Continues...



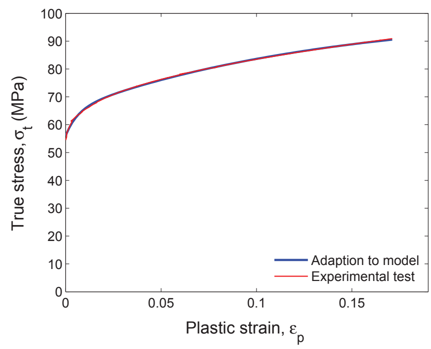
(y) Test #15



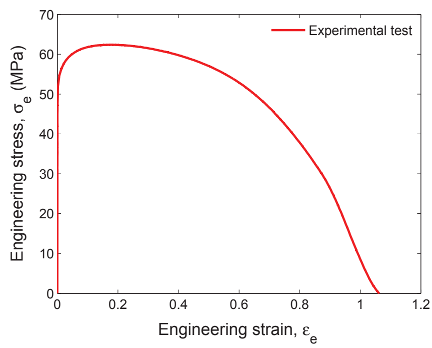
(z) Test #15



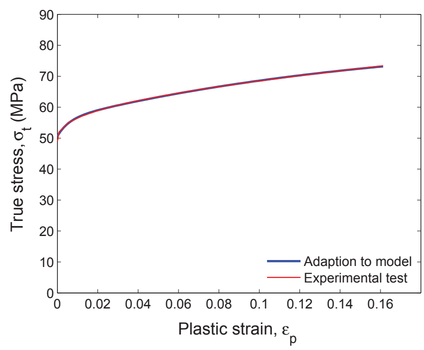
(aa) Test #16



(bb) Test #16

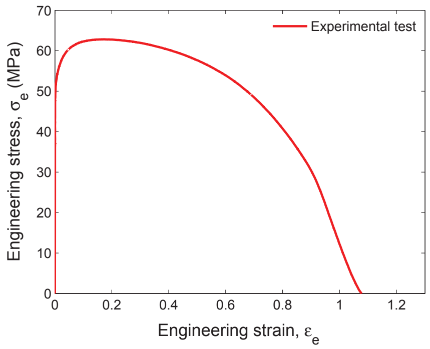


(cc) Test #17

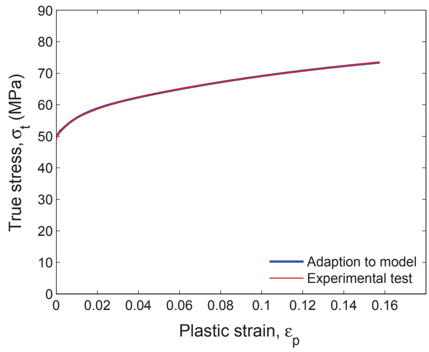


(dd) Test #17

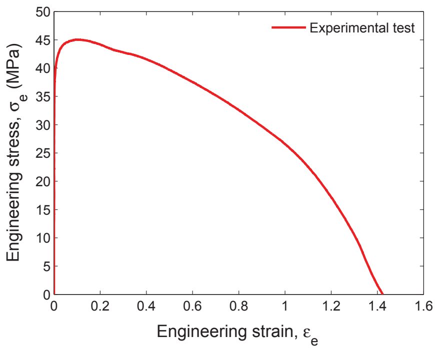
Figure E.3: Continues...



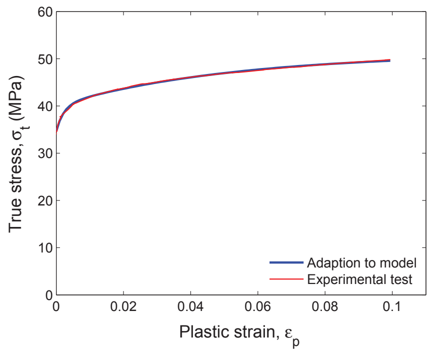
(ee) Test #18



(ff) Test #18

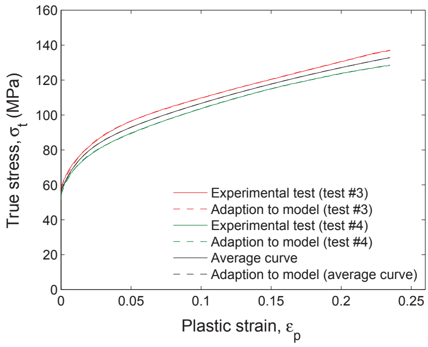


(gg) Test #19

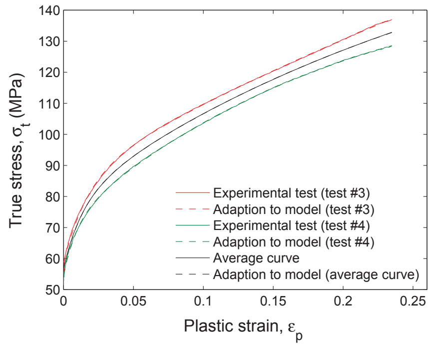


(hh) Test #19

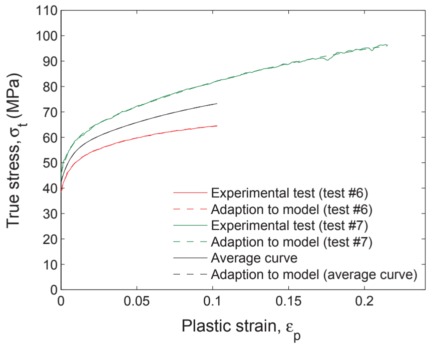
Figure E.3: Plots (a)-(hh) show the engineering stress-strain curve and true stress-plastic strain curve from quasi static experiments for AA6060-L



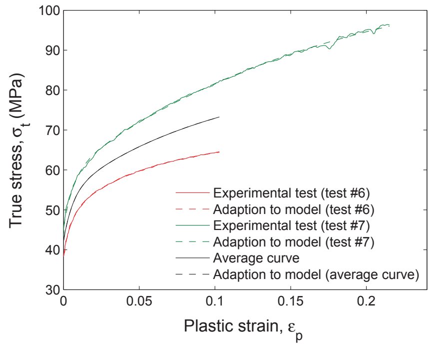
(a)



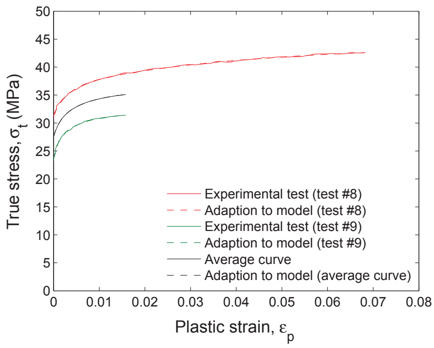
(b)



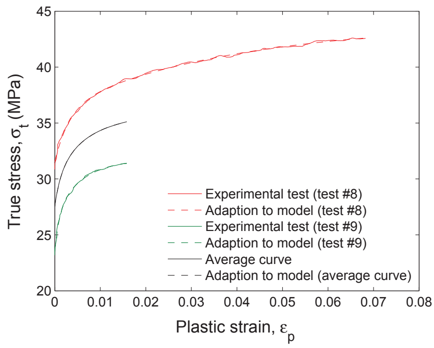
(c)



(d)

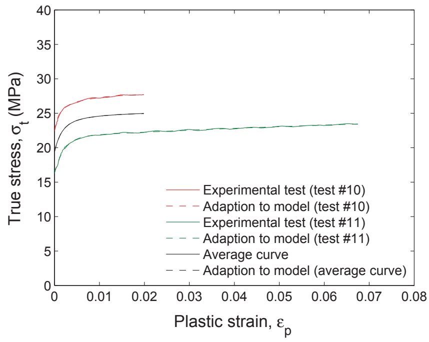


(e)

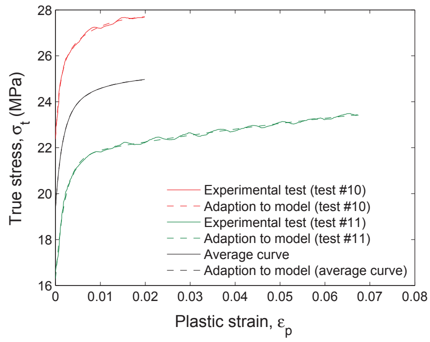


(f)

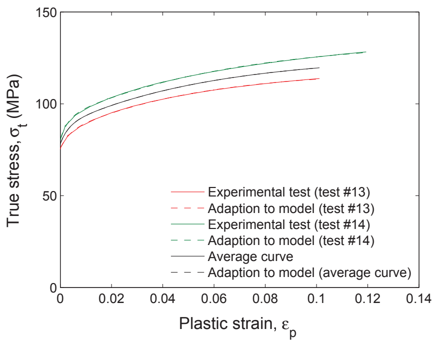
Figure E.4: Continues...



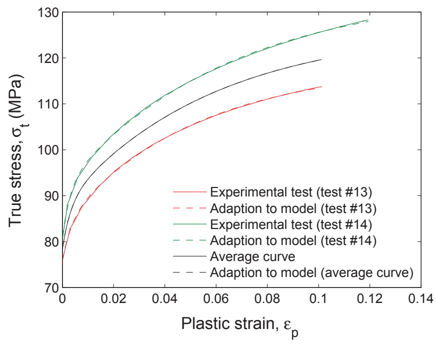
(g)



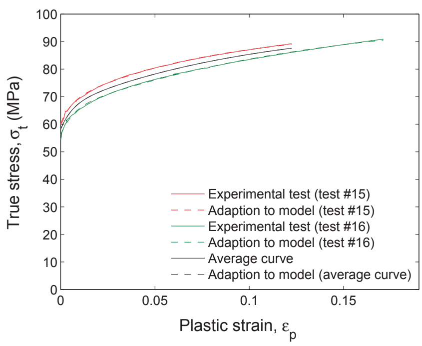
(h)



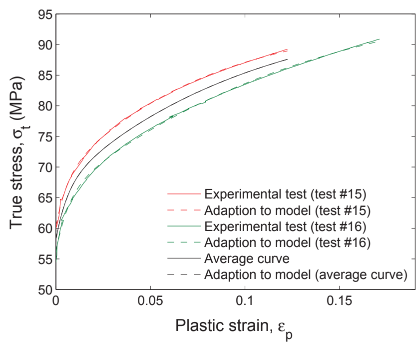
(i)



(j)

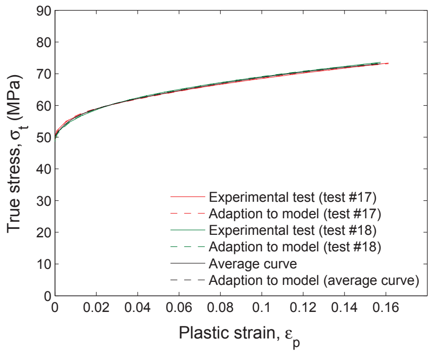


(k)

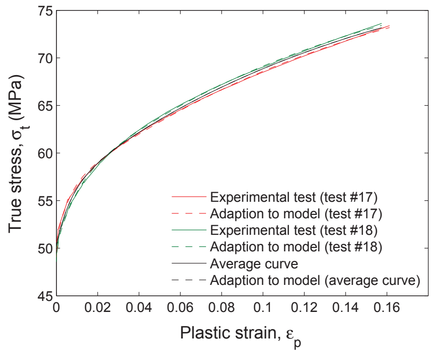


(l)

Figure E.4: Continues...



(m)



(n)

Figure E.4: Plots (a)-(n) show the true stress-plastic strain curve from quasi-static experiments for AA6060-L for same boundary conditions together with the average curve. Plots in right column is equal to plots in left column, but for a narrower range of values on the ordinate axis.

E.3 AA6060-H

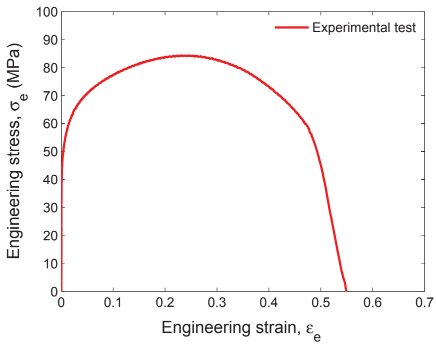
Table E.5 and E.6 show respectively an overview of the experimental quasi-static tests and the resulting true stress-plastic strain parameters and estimated fracture strain. The true stress-plastic strain parameters are fitted with Voce rule, $\sigma = \sigma_Y + \sum_{i=1}^{2} Q_i(1 - e^{-C_i \varepsilon_p})$. In Table E.6, the measured yield stress from experimental data is denoted $\sigma_{Y,experiment}$, while the yield stress fitted with Voce rule is denoted σ_Y. The plastic strain value at onset of necking is denoted $\varepsilon_{p,max}$ and the estimated fracture strain is denoted ε_f.

Test #	Strain rate (s^{-1})	Temp. ($^{\circ}K$)	Diameter (mm)	Gauge length (mm)	Comment
1	0.01	523	3.00	5	OK
2	0.01	523	3.02	5	OK
3	0.01	523	2.99	5	OK
4	0.01	523	2.99	5	OK
5	0.01	573	2.99	5	OK
6	0.01	573	2.99	5	OK
7	0.01	293	3.01	5	OK
8	0.01	473	3.01	5	OK
9	0.01	473	3.01	5	OK
10	0.01	633	3.01	5	OK
11	0.01	633	not meas.	5	OK
12	1	293	not meas.	5	OK
13	1	470	not meas.	5	OK
14	1	470	not meas.	5	OK
15	1	470	not meas.	5	D.s.*
16	1	523	not meas.	5	OK
17	1	523	not meas.	5	OK
18	1	573	not meas.	5	OK
19	1	573	not meas.	5	OK
20	1	633	not meas.	5	OK
21	1	633	not meas.	5	OK

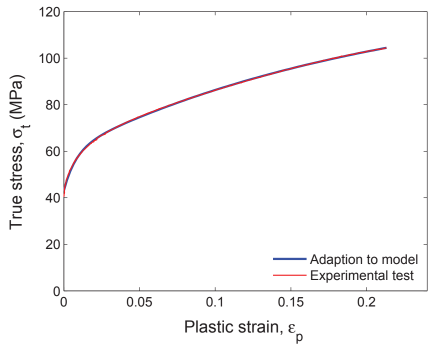
Table E.5: Overview of experimental quasi-static tests for AA6060-H

AA6060-H								
$Test\ \#$	$\sigma_{Y,experiment}$ (MPa)	σ_Y (MPa)	Q_1 (MPa)	C_1	Q_2 (MPa)	C_2	$\varepsilon_{p,max}$	ε_f
1	-	-	-	-	-	-	-	-
2	40.37	43.00	70.41	4.70	16.90	124.60	0.2130	1.166
3	42.41	43.52	45.84	3.49	10.16	108.59	0.1902	1.234
4	43.12	43.55	43.58	8.11	12.37	140.86	0.1974	1.335
5	34.83	35.70	18.02	10.09	5.95	241.41	0.1111	2.464
6	37.44	37.94	20.44	11.15	7.30	189.46	0.1198	1.964
7	73.96	77.54	171.65	5.02	22.98	59.82	0.2667	0.645
8	60.98	61.92	63.74	4.10	11.37	98.80	0.2418	0.869
9	53.13	54.55	36.05	40.35	7.11	486.41	0.0705	0.86
10	19.90	20.93	2.72	64.53	3.70	532.19	0.0315	6.202
11	19.30	19.50	3.75	35.72	2.37	660.47	0.0389	5.726
12	93.37	95.96	124.73	6.43	23.23	137.13	0.2200	0.868
13	83.44	84.87	45.39	13.34	10.37	228.41	0.1330	1.251
14	77.35	77.63	38.83	13.96	9.45	214.56	0.1253	1.432
15	-	-	-	-	-	-	-	-
16	61.10	62.89	30.86	12.11	9.52	196.88	0.1328	1.555
17	56.36	57.61	28.95	9.68	8.65	154.12	0.1423	1.783
18	51.48	52.59	23.36	11.33	7.60	231.02	0.1311	2.112
19	49.61	49.98	17.12	15.92	5.97	342.21	0.1049	2.329
20	38.42	38.72	8.65	23.51	5.12	343.58	0.0679	3.379
21	35.02	35.70	11.25	22.26	7.34	533.79	0.0918	3.047
2-3	41.74	43.29	54.81	6.16	14.72	129.99	0.1984	-
5-6	36.13	36.83	19.18	10.73	6.59	211.65	0.1101	-
8-9	57.06	58.54	30.35	29.11	7.25	230.46	0.0688	-
10-11	19.60	20.22	3.11	48.87	3.04	577.51	0.0322	-
13-14	80.40	81.25	42.10	13.63	9.91	221.69	0.1256	-
16-17	58.73	60.26	29.71	11.01	9.05	175.36	0.1321	-
18-19	50.55	51.31	19.93	13.50	6.73	274.11	0.1046	-
20-21	36.72	37.23	9.93	23.27	6.14	447.58	0.0680	-

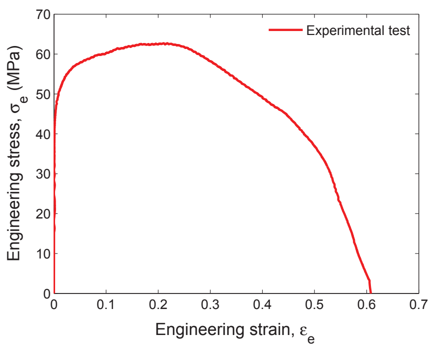
Table E.6: Resulting true stress-plastic strain parameters and estimated fracture strain for AA6060-H from quasi-static experiments



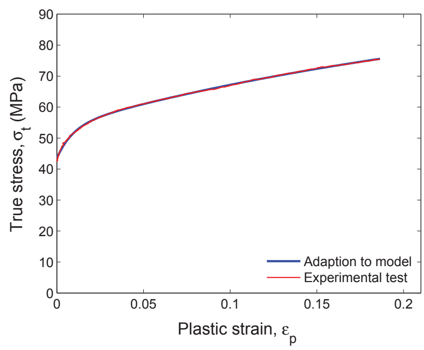
(a) Test #2



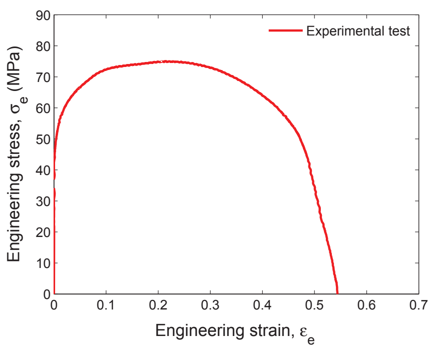
(b) Test #2



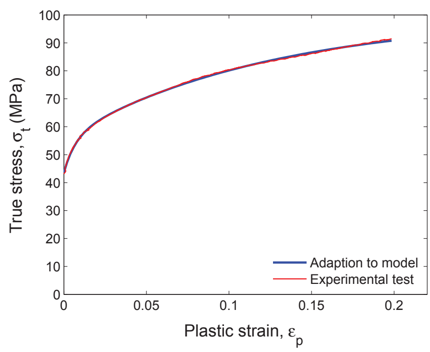
(c) Test #3



(d) Test #3

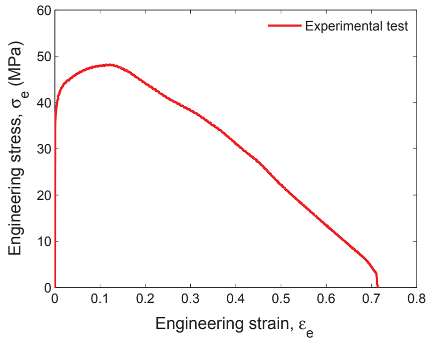


(e) Test #4

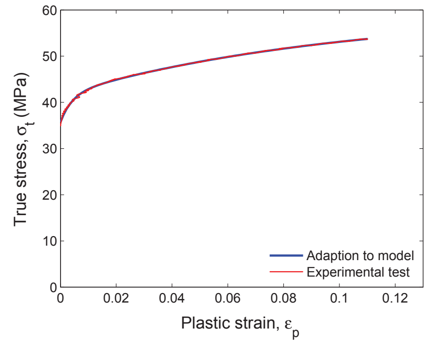


(f) Test #4

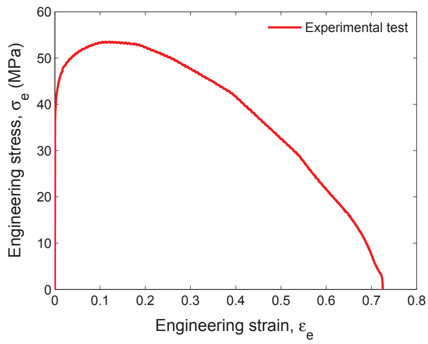
Figure E.5: Continues...



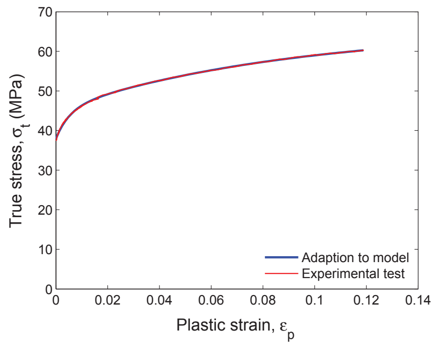
(g) Test #5



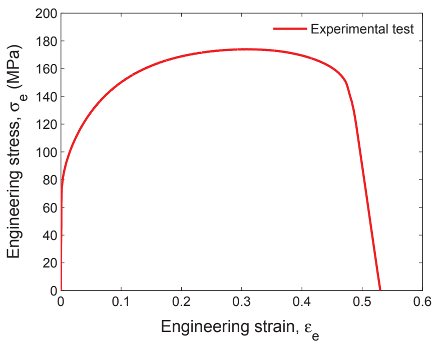
(h) Test #5



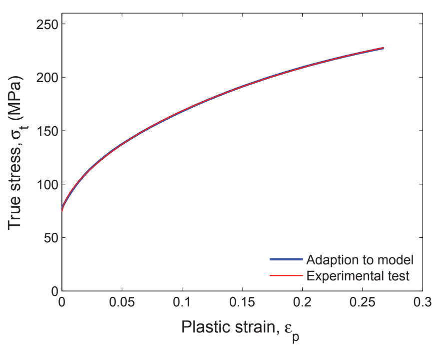
(i) Test #6



(j) Test #6

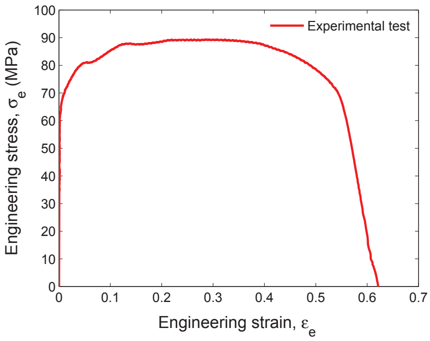


(k) Test #7

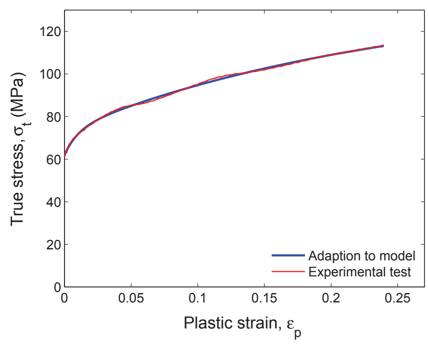


(l) Test #7

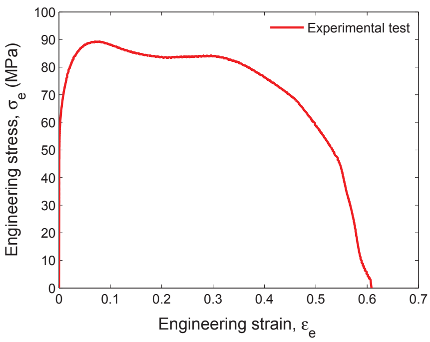
Figure E.5: Continues...



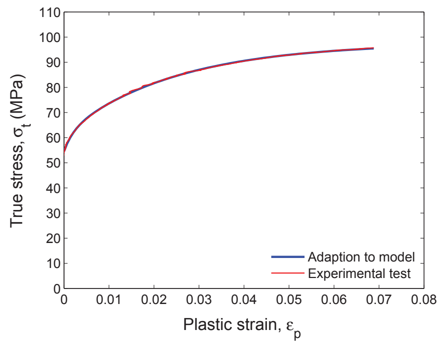
(m) Test #8



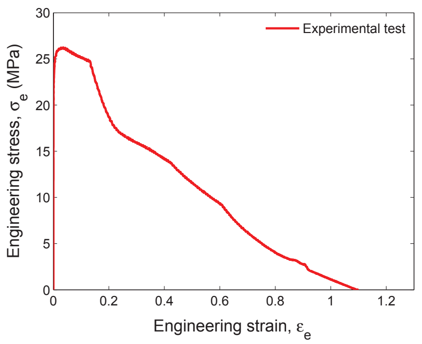
(n) Test #8



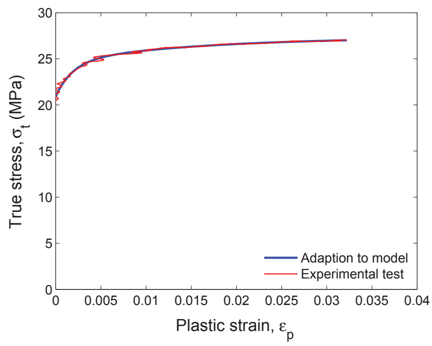
(o) Test #9



(p) Test #9

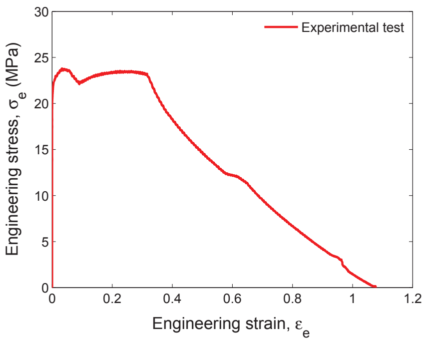


(q) Test #10

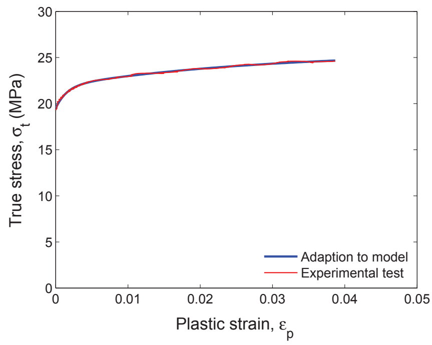


(r) Test #10

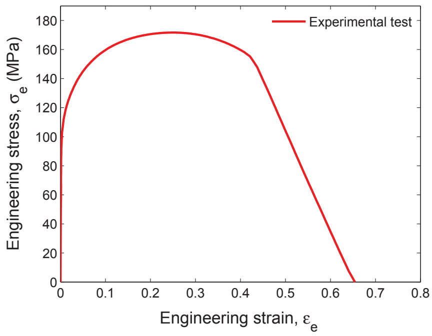
Figure E.5: Continues...



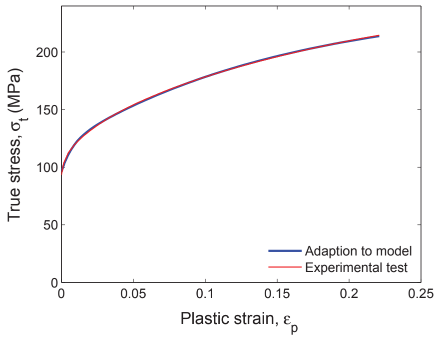
(s) Test #11



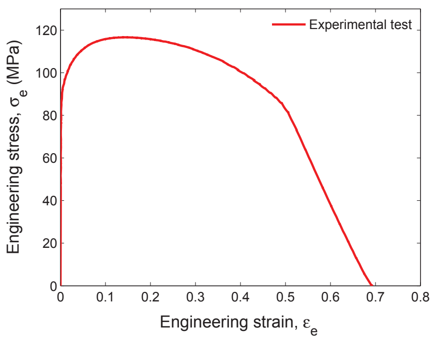
(t) Test #11



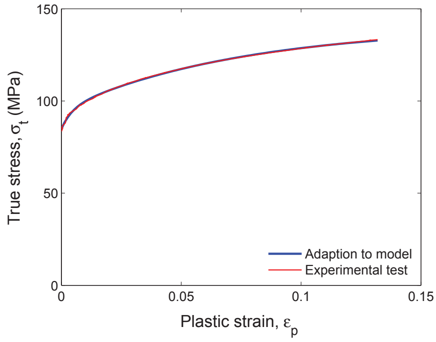
(u) Test #12



(v) Test #12

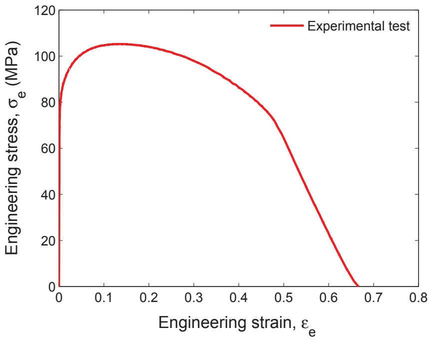


(w) Test #13

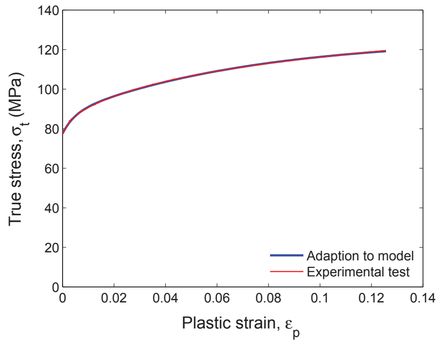


(x) Test #13

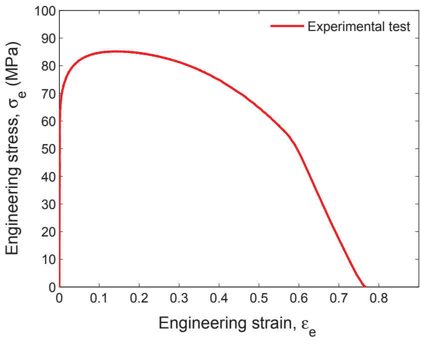
Figure E.5: Continues...



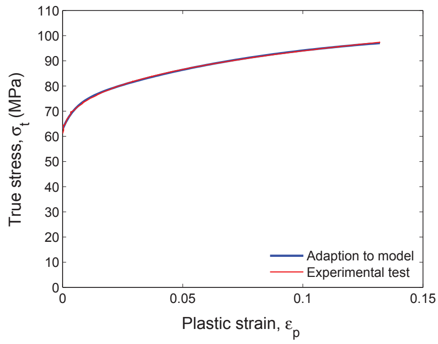
(y) Test #14



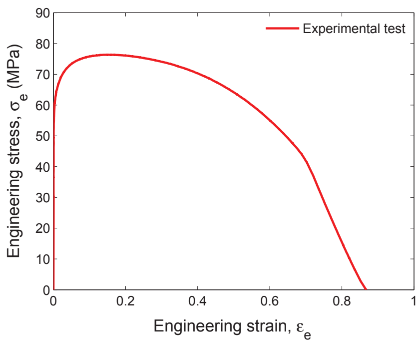
(z) Test #14



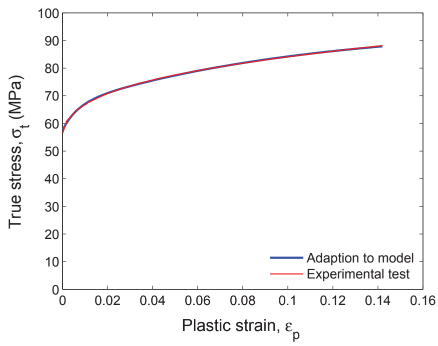
(aa) Test #16



(bb) Test #16

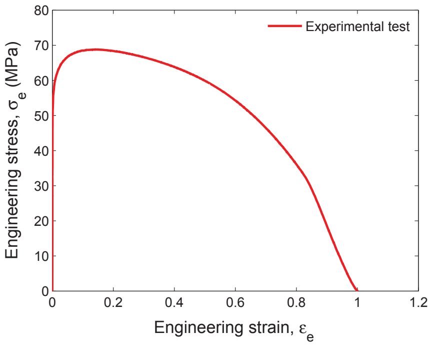


(cc) Test #17

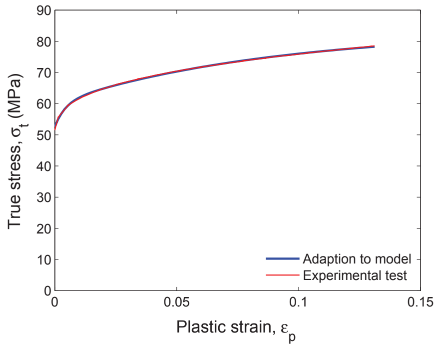


(dd) Test #17

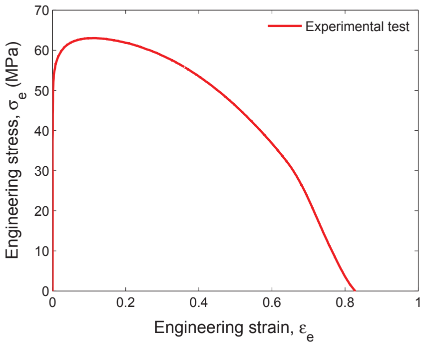
Figure E.5: Continues...



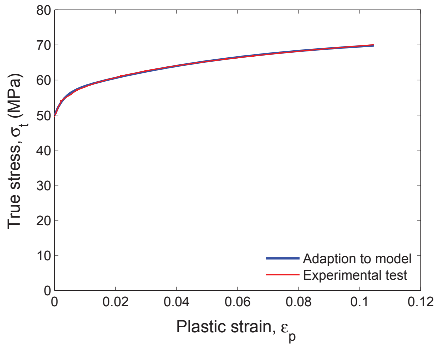
(ee) Test #18



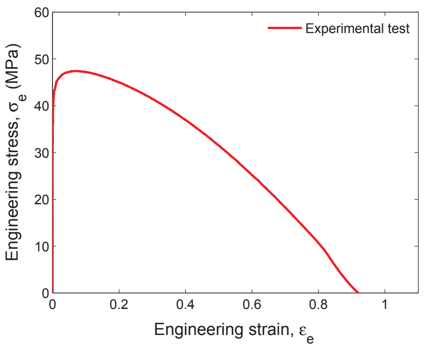
(ff) Test #18



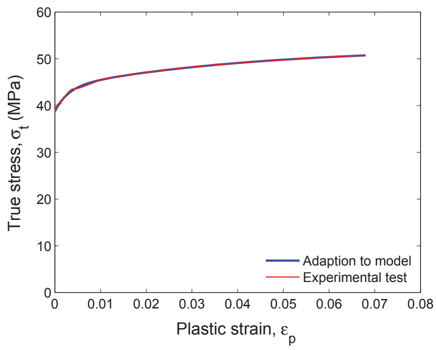
(gg) Test #19



(hh) Test #19

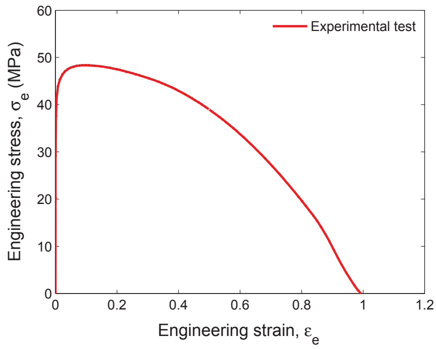


(ii) Test #20

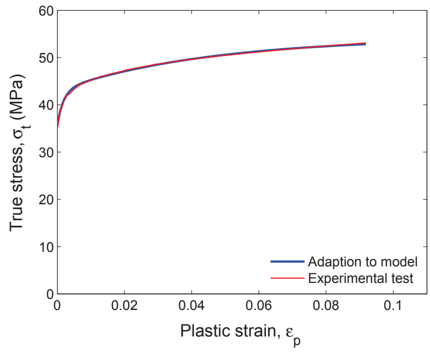


(jj) Test #20

Figure E.5: Continues...

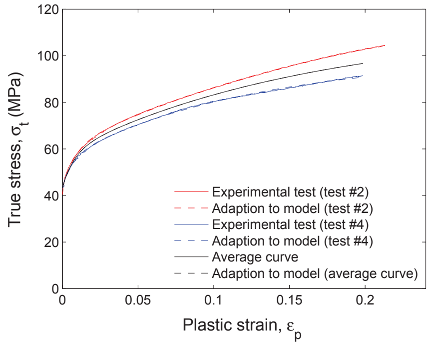


(kk) Test #21

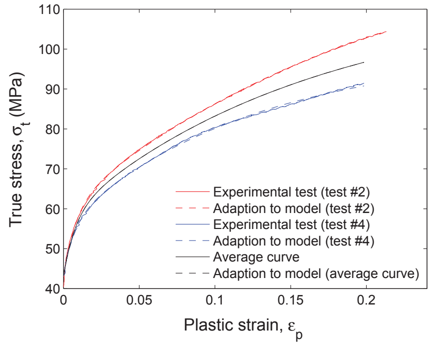


(ll) Test #21

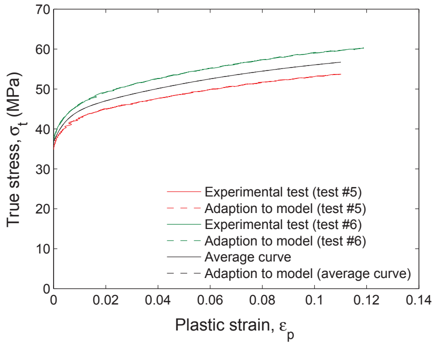
Figure E.5: Plots (a)-(ll) show the engineering stress-strain curve and true stress-plastic strain curve from quasi static experiments for AA6060-H



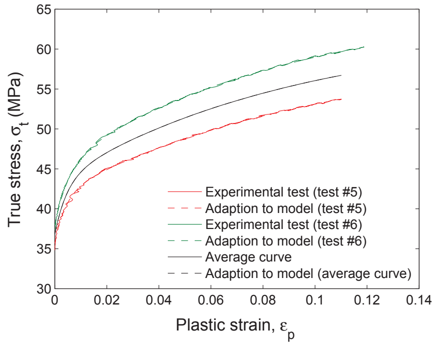
(a)



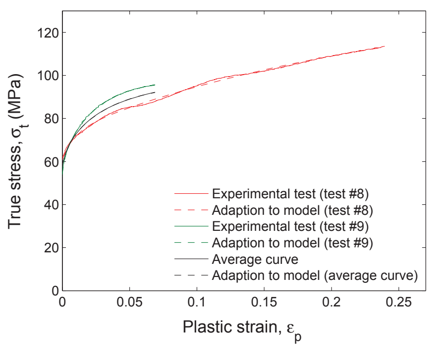
(b)



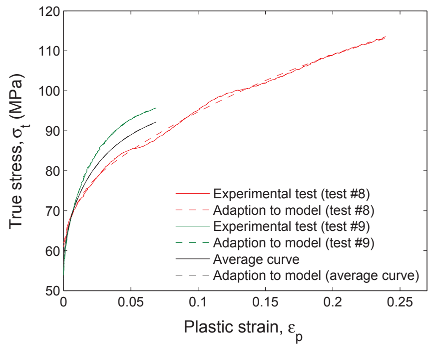
(c)



(d)



(e)



(f)

Figure E.6: Continues...

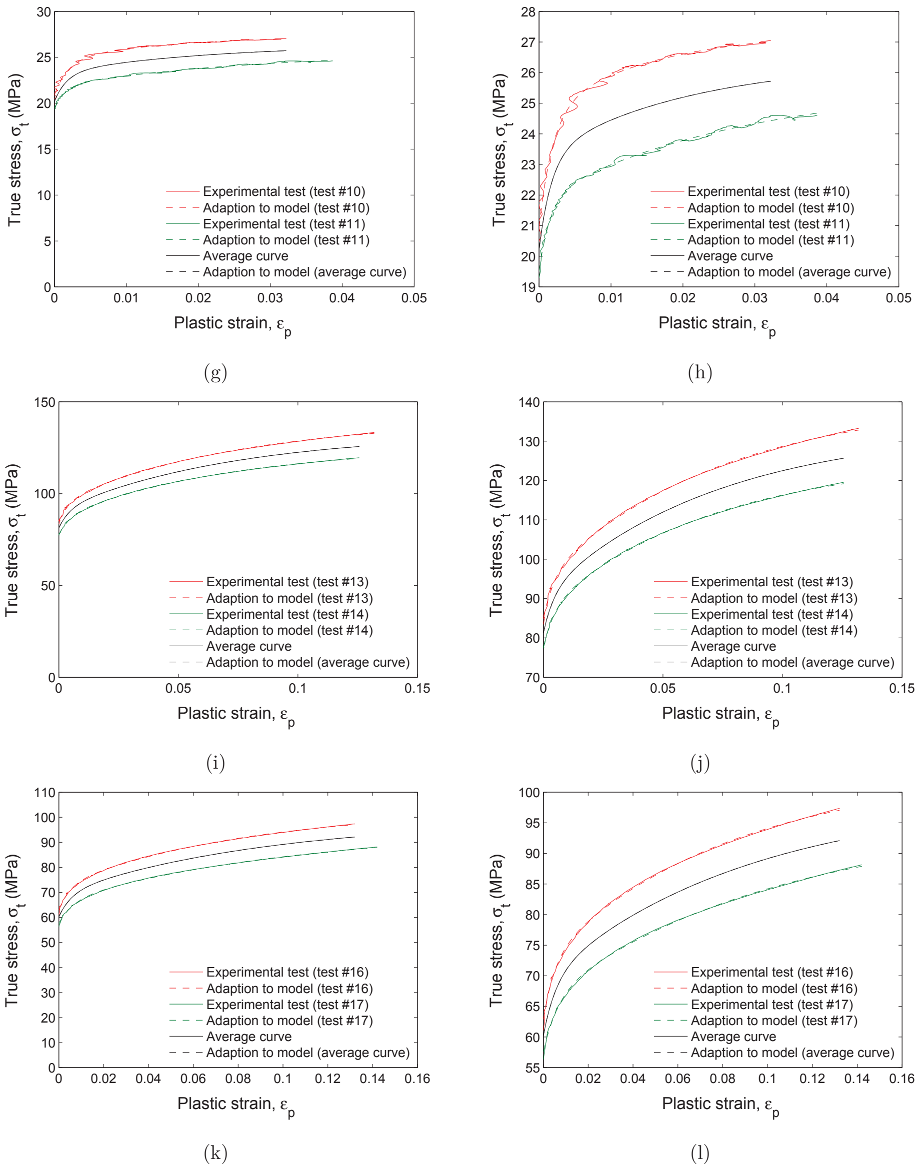
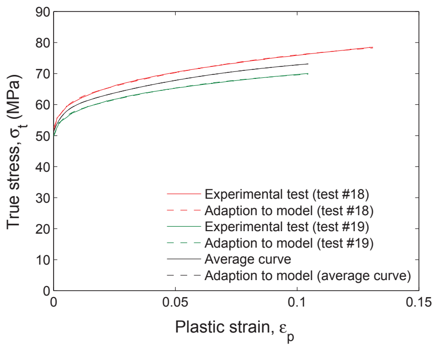
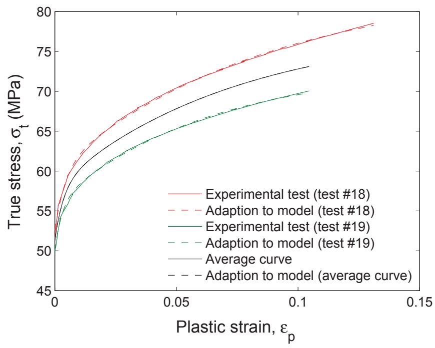


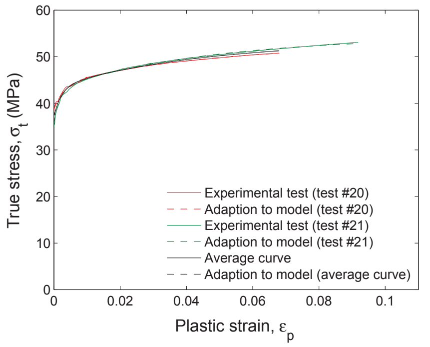
Figure E.6: Continues...



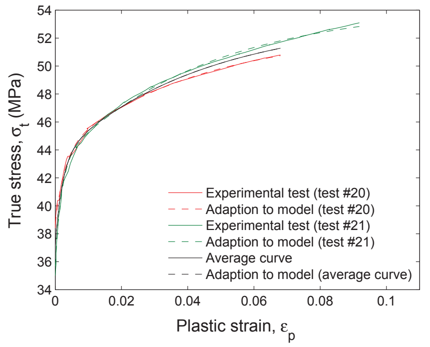
(m)



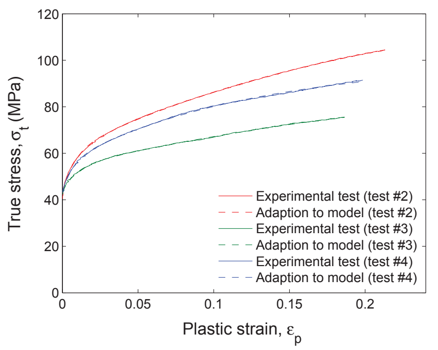
(n)



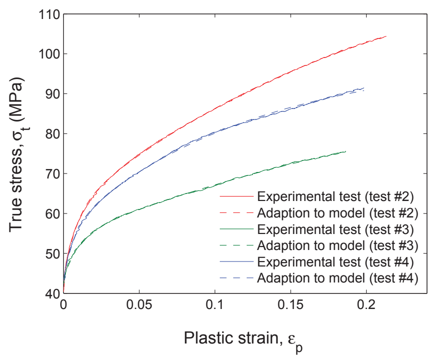
(o)



(p)



(q)



(r)

Figure E.6: Plots (a)-(p) show the true stress-plastic strain curve from quasi-static experiments for AA6060-H for the same boundary conditions together with the average curve. Plots in right column is equal to plots in left column, but for a narrower range of values on the ordinate axis. Plots (q) and (r) show the true stress-plastic strain curve for test 2, 3 and 4 with same boundary conditions.

F Experimental Results from SHTB Tests

F.1 AA6060-L

Table F.1 and F.2 show respectively an overview of the experimental quasi-static tests and the resulting true stress-plastic strain parameters and estimated fracture strain. The true stress-plastic strain parameters are fitted with Voce rule, $\sigma = \sigma_Y + \sum_{i=1}^{2} Q_i(1 - e^{-C_i\varepsilon_p})$. In Table F.2, the measured yield stress from experimental data is denoted $\sigma_{Y,experiment}$, while the yield stress fitted with Voce rule is denoted σ_Y. The plastic strain value at onset of necking is denoted $\varepsilon_{p,max}$ and the estimated fracture strain is denoted ε_f.

AA6060-L

Test #	Strain rate (s^{-1})	Temp. $(^\circ K)$	Diameter (mm)	Gauge length (mm)	Comment
7	340	523	not meas.	5	N.s.*
8	372	523	not meas.	5	N.s.*
9	354	523	not meas.	5	OK
10	460	523	not meas.	5	N.s.*
11	781	523	not meas.	5	OK

* N.s. = not successful experiment

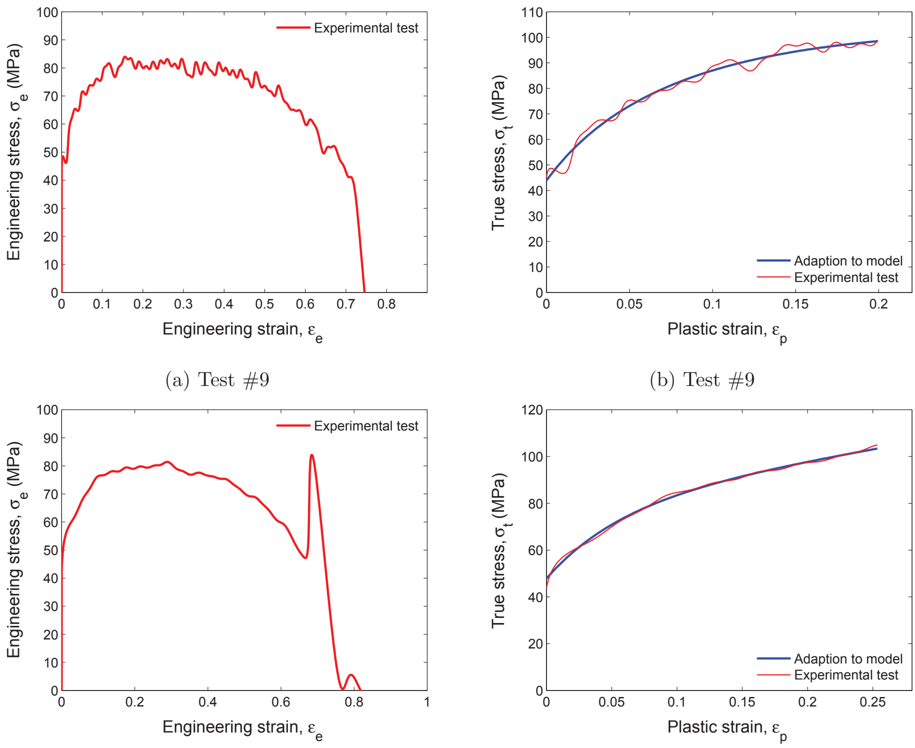
Table F.1: Overview of experimental SHTB tests for AA6060-L

AA6060-L

$Test\ \#$	$\sigma_{Y,experiment}$ (MPa)	σ_Y (MPa)	Q_1 (MPa)	C_1	Q_2 (MPa)	C_2	$\varepsilon_{p,max}$	ε_f
9 (s.g.)	45.26	43.83	53.48	11.72	6.34	42.22	0.1994	
9 (camera)	45.93	44.02	42.35	3.41	33.46	44.80	0.9476	2.195
9 (camera*)	45.93	44.37	52.83	2.24	35.25	41.36	0.9476	
11 (s.g.)	43.74	47.84	1000.00	0.10	31.63	17.10	0.2534	
11 (camera)	43.79	48.87	44.85	4.17	23.17	67.42	0.5666	2.088
11 (camera*)	43.79	49.50	58.58	2.27	26.18	53.81	0.5666	

* not corrected using Bridgman's formula

Table F.2: Resulting true stress-plastic strain parameters for AA6060-L from SHTB experiments



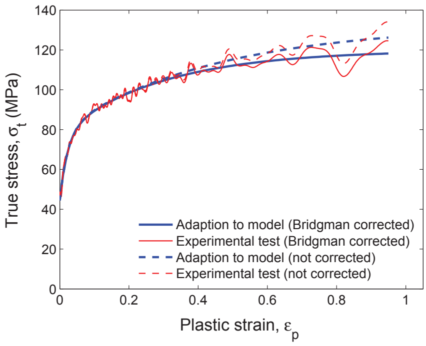
(a) Test #9

(b) Test #9

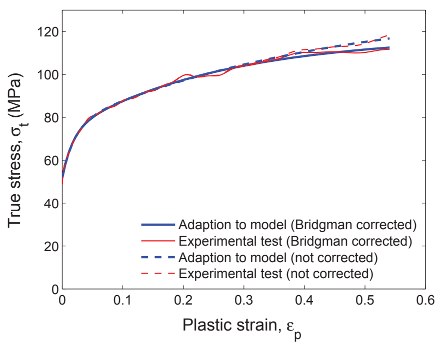
(c) Test #11

(d) Test #11

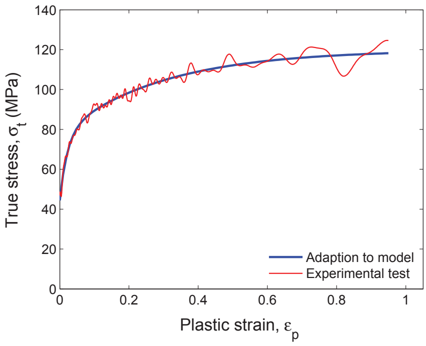
Figure F.1: Plots show the engineering stress-strain curve and true stress-plastic strain curve from SHTB experiments for AA6060-L (strain gauge measurements)



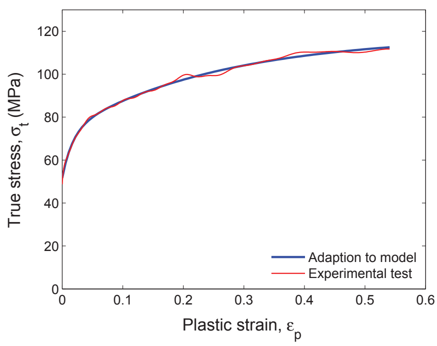
(a) Test #9



(b) Test #11

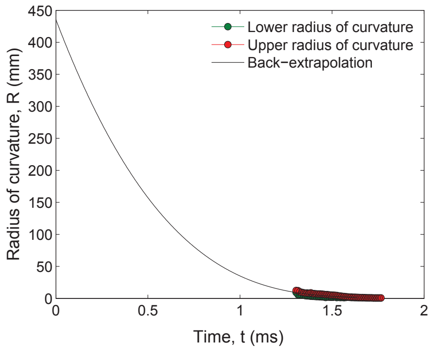


(c) Test #9

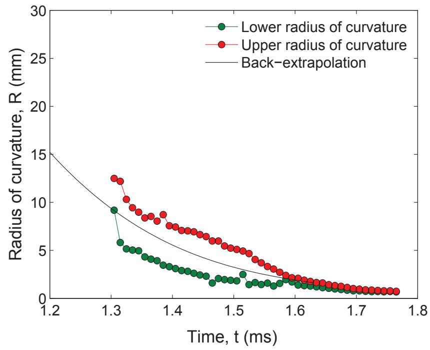


(d) Test #11

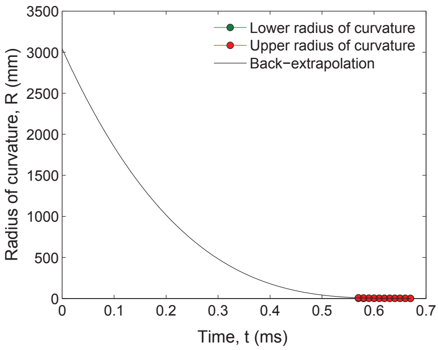
Table F.3: Plots (a)-(b) show both uncorrected and Bridgman corrected true stress-plastic strain curves from camera measurements for AA6060-L. Plots (c)-(d) show only the Bridgman corrected true stress-plastic strain curves.



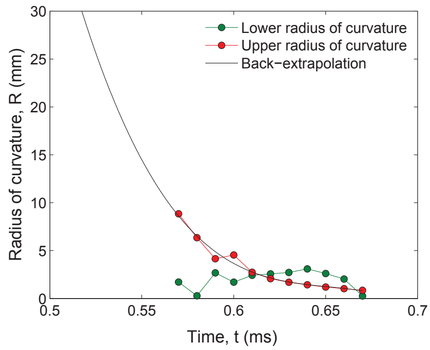
(a) Test #9



(b) Test #9

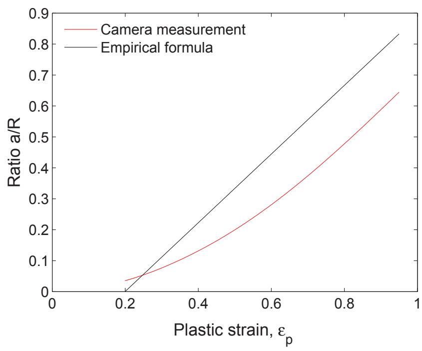


(c) Test #11

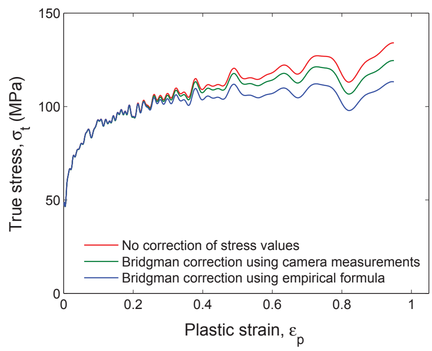


(d) Test #11

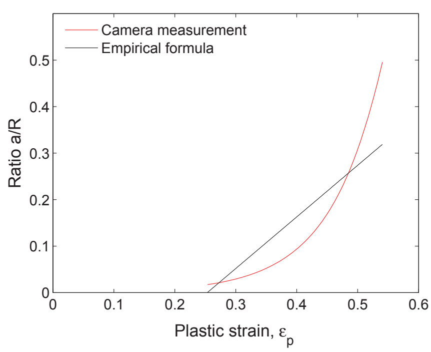
Table F.4: Plots (a)-(d) show the back-extrapolation of radius of curvature for AA6060-L. Plots (b) and (d) show the same as (a) and (c) but for a narrower range for both the ordinate and abscissa axis



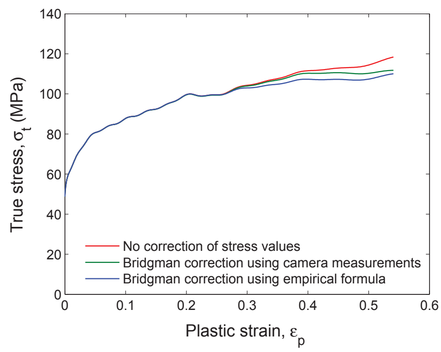
(a) Test #9



(b) Test #9



(c) Test #11



(d) Test #11

Figure F.2: Plots (a)-(d) show the minimum radius at the neck-radius of curvature ratio from both camera measurements and the empirical formula and the resulting true stress-plastic strain plot using Bridgman correction for AA6060-L

F.2 AA6060-H

Table F.5 and F.6 show respectively an overview of the experimental quasi-static tests and the resulting true stress-plastic strain parameters and estimated fracture strain. The true stress-plastic strain parameters are fitted with Voce rule, $\sigma = \sigma_Y + \sum_{i=1}^{2} Q_i(1 - e^{-C_i \varepsilon_p})$. In Table F.6, the measured yield stress from experimental data is denoted $\sigma_{Y,experiment}$, while the yield stress fitted with Voce rule is denoted σ_Y. The plastic strain value at onset of necking is denoted $\varepsilon_{p,max}$ and the estimated fracture strain is denoted ε_f.

AA6060-H

Test #	Strain rate (s^{-1})	Temp. ($°K$)	Diameter (mm)	Gauge length (mm)	Comment
1	350	523	not meas.	5	N.s.*
2	376	523	not meas.	5	OK
3	365	613	not meas.	5	N.s.*
4	388	673	not meas.	5	N.s.*
5	789	523	not meas.	5	OK
6	800	573	not meas.	5	N.s.*

* N.s. = not successful experiment

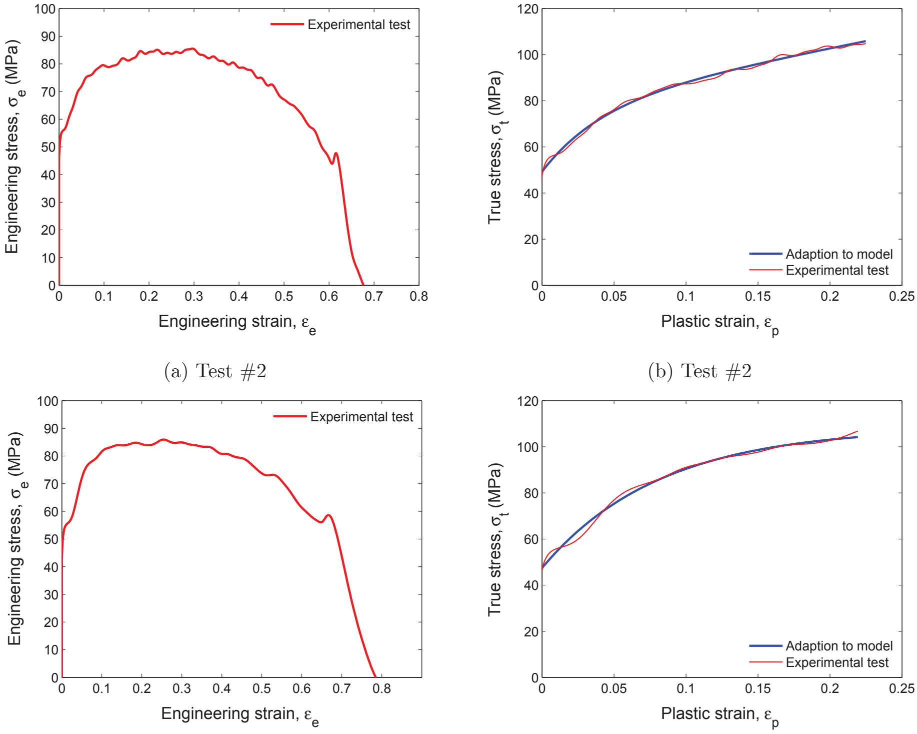
Table F.5: Overview of experimental SHTB tests for AA6060-H

AA6060-H

Test #	$\sigma_{Y,experiment}$ (MPa)	σ_Y (MPa)	Q_1 (MPa)	C_1	Q_2 (MPa)	C_2	$\varepsilon_{p,max}$	ε_f
2 (s.g.)	47.19	49.26	1000.00	0.13	28.74	23.95	0.2248	
2 (camera)	49.92	49.94	110.25	1.08	30.84	32.93	1.4709	1.856
2 (camera*)	49.92	49.36	176.04	0.72	30.21	37.01	1.4709	
5 (s.g.)	46.26	47.45	45.00	12.21	15.95	12.21	0.2194	
5 (camera)	51.47	51.94	59.01	1.10	38.54	32.85	1.0211	2.013
5 (camera*)	51.47	52.23	1000.00	0.06	39.35	31.47	1.0211	

* not corrected using Bridgman's formula

Table F.6: Resulting true stress-plastic strain parameters for AA6060-H from SHTB experiments



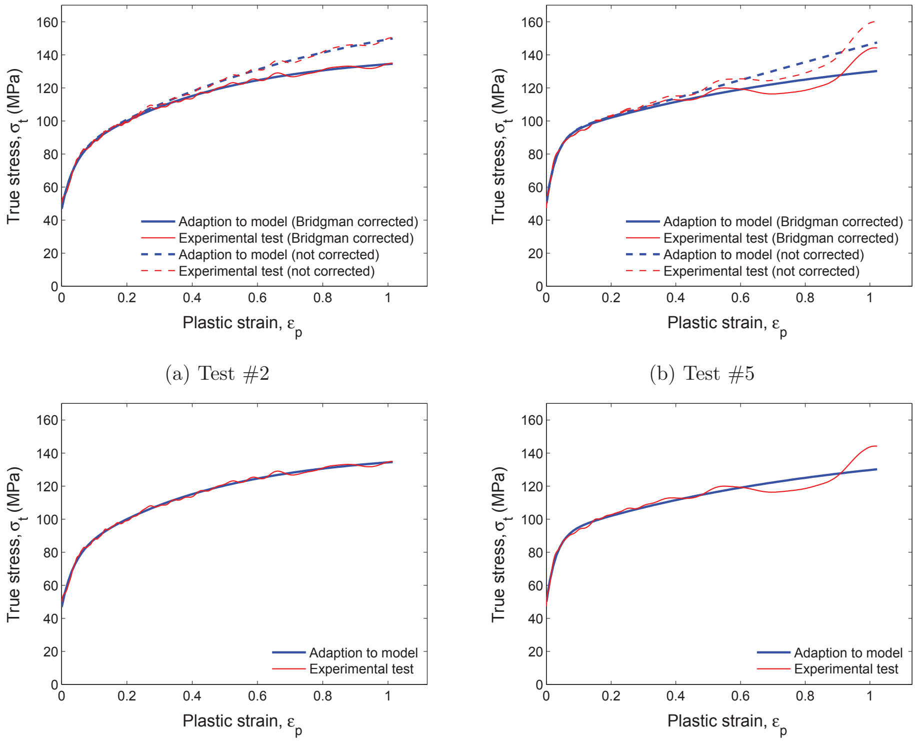
(a) Test #2

(b) Test #2

(c) Test #5

(d) Test #5

Figure F.3: Plots show the engineering stress-strain curve and true stress-plastic strain curve from SHTB experiments for the AA6060-H alloy (strain gauge measurements)



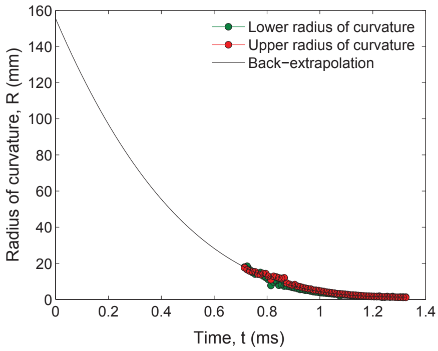
(a) Test #2

(b) Test #5

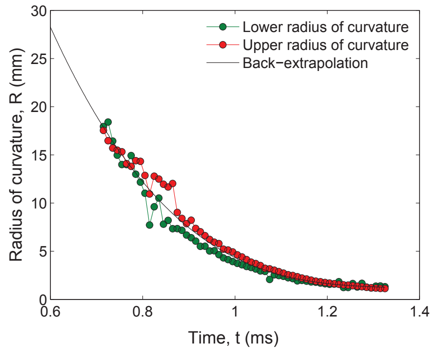
(c) Test #2

(d) Test #5

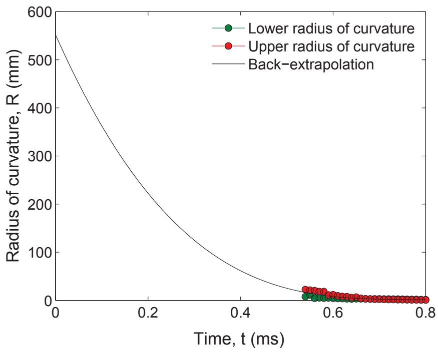
Figure F.4: Plots (a)-(b) show both uncorrected and Bridgman corrected true stress-plastic strain curves from camera measurements for AA6060-H. Plots (c)-(d) show only the Bridgman corrected true stress-plastic strain curves.



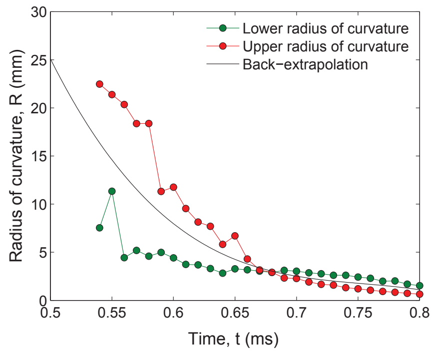
(a) Test #2



(b) Test #2

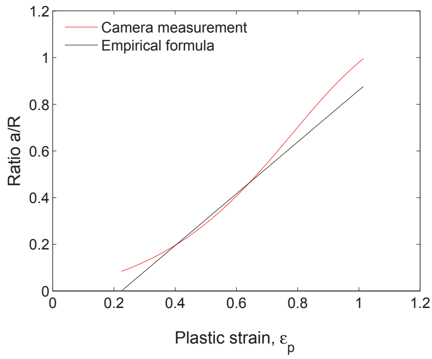


(c) Test #5

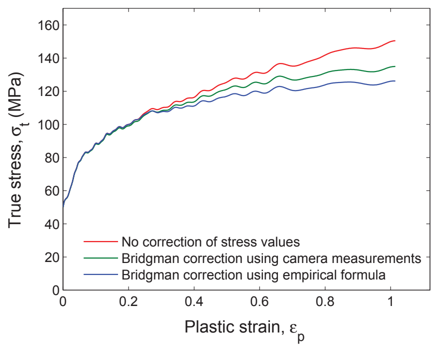


(d) Test #5

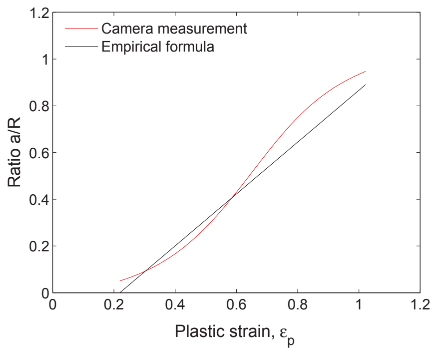
Table F.7: Plots (a)-(d) show the back-extrapolation of radius of curvature for AA6060-H. Plots (b) and (d) show the same as (a) and (c) but for a narrower range for both the ordinate and abscissa axis



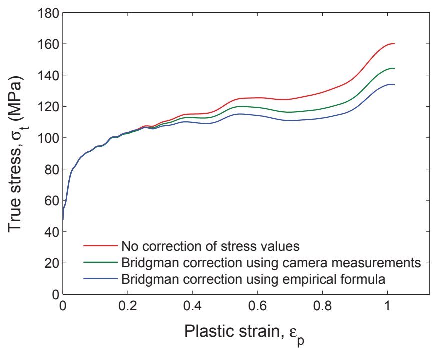
(a) Test #2



(b) Test #2



(c) Test #5



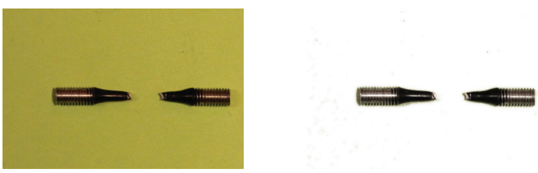
(d) Test #5

Figure F.5: Plots (a)-(d) show the minimum radius at the neck-radius of curvature ratio from both camera measurements and the empirical formula and the resulting true stress-plastic strain plot using Bridgman correction for AA6060-H

G Pictures of Specimens Post-Fracture from Quasi-Static Tests

Pictures have been taken of specimens post-fracture from all quasi-static and SHTB tests. This has been done using a digital SLR camera, a tripod and a light source to light up the shadows best possible. The specimens were placed on a sheet of paper with a contrast color (a light green color was used, even though it looks from Fig. G.1 the color is yellow) to be able to remove the background using photo editing software. Two pieces of strings attached to the table and paper sheet were used to ensure that the specimens were placed at the same exact spot to get the same proportions of the specimens in the resulting pictures. In order to do this effectively, an algorithm was created for Adobe Photoshop to execute the same image processing routine for all pictures taken. Fig. G.2a and G.2b show respectively an example of a specimen before and after applying the image processing routine. Some remaining background noise were present in all pictures and have been removed manually.

Figure G.1: Setup for taking pictures of tensile specimens post-fracture



(a) (b)

Figure G.2: (a)-(b): Example of picture of tensile specimen respectively before and after subjected to image processing routine

G.1 AA6060-OLD

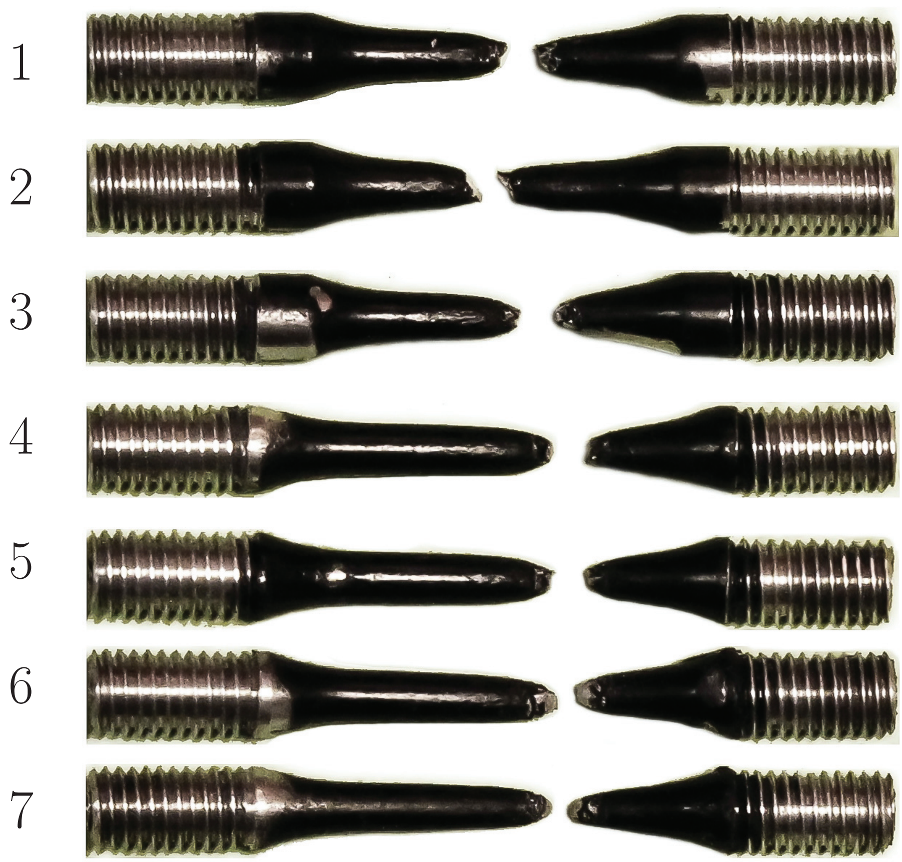


Figure G.3: Tensile specimens post-fracture for AA6060-OLD from all quasi-static tests

G.2 AA6060-L

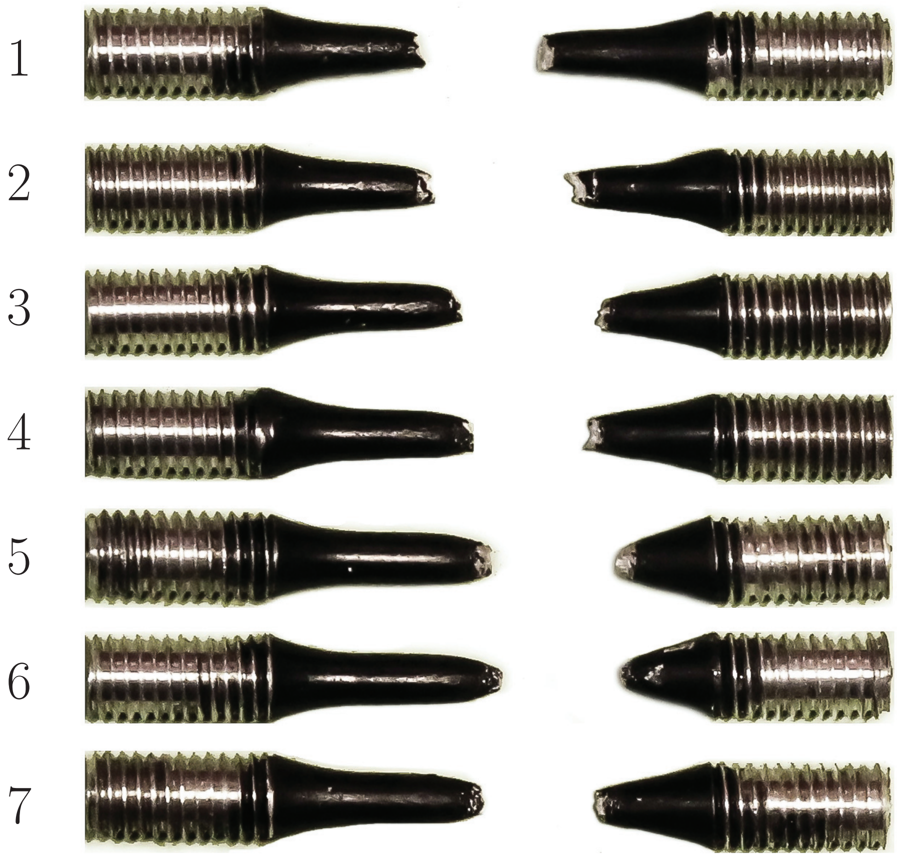


Figure G.4: Continues...

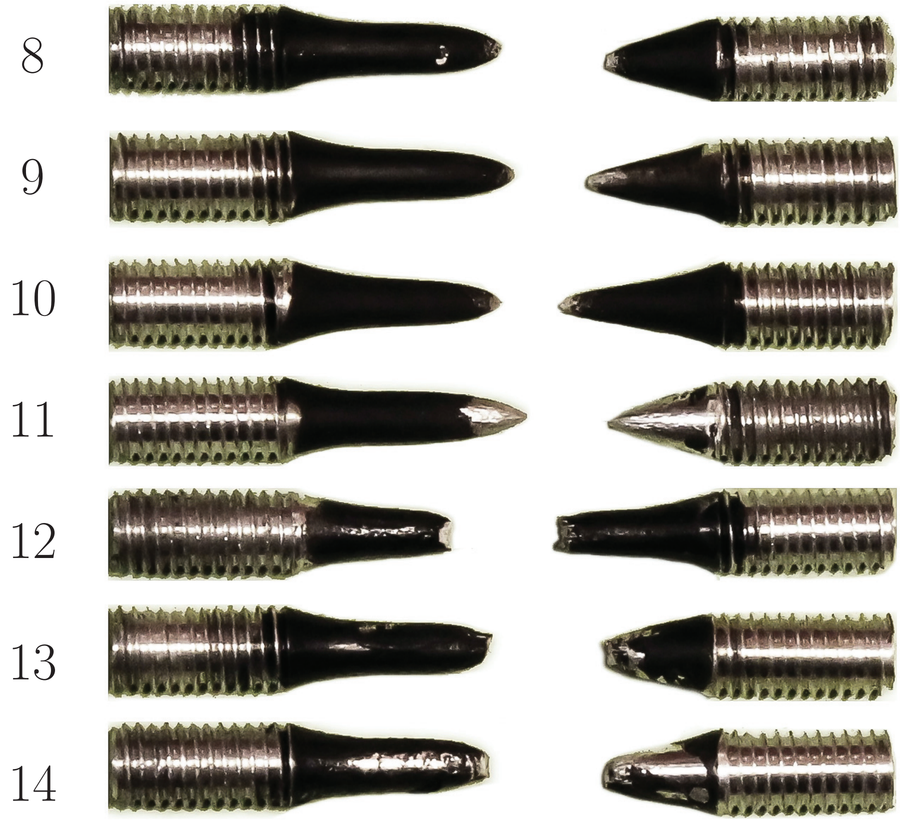


Figure G.4: Continues...

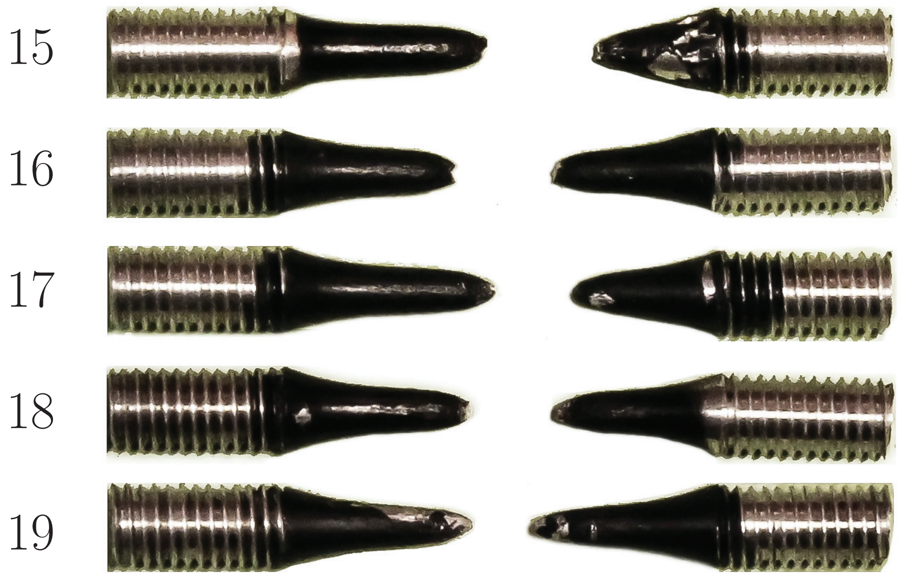


Figure G.4: Tensile specimens post-fracture for AA6060-L from all quasi-static tests

G.3 AA6060-H

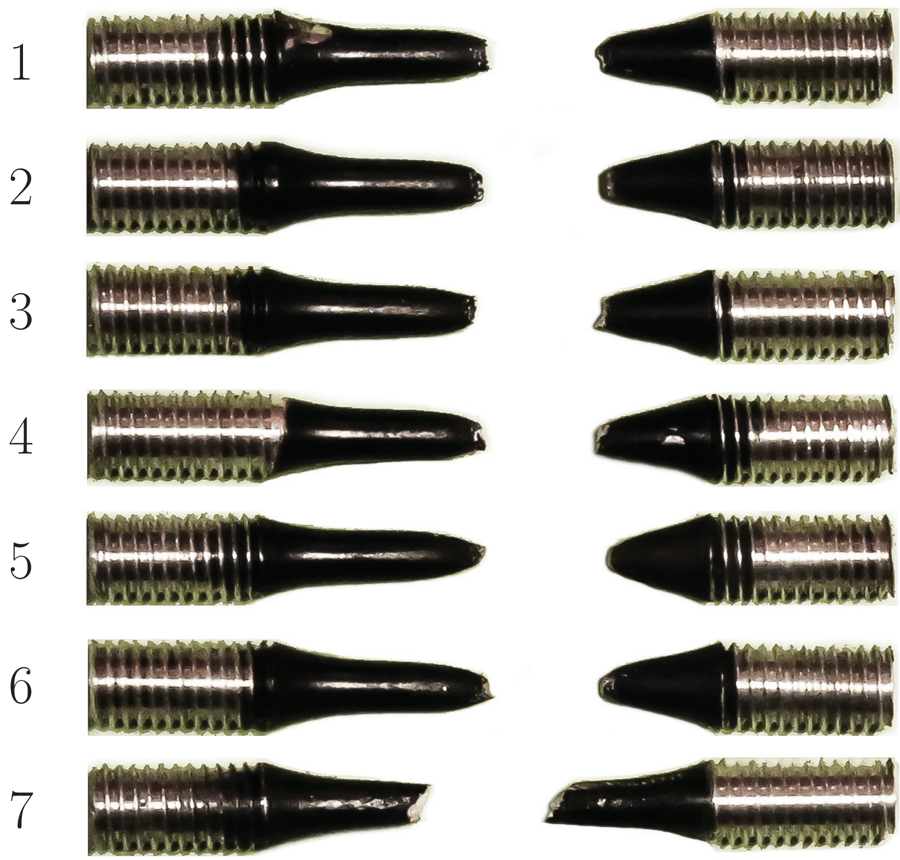


Figure G.5: Continues...

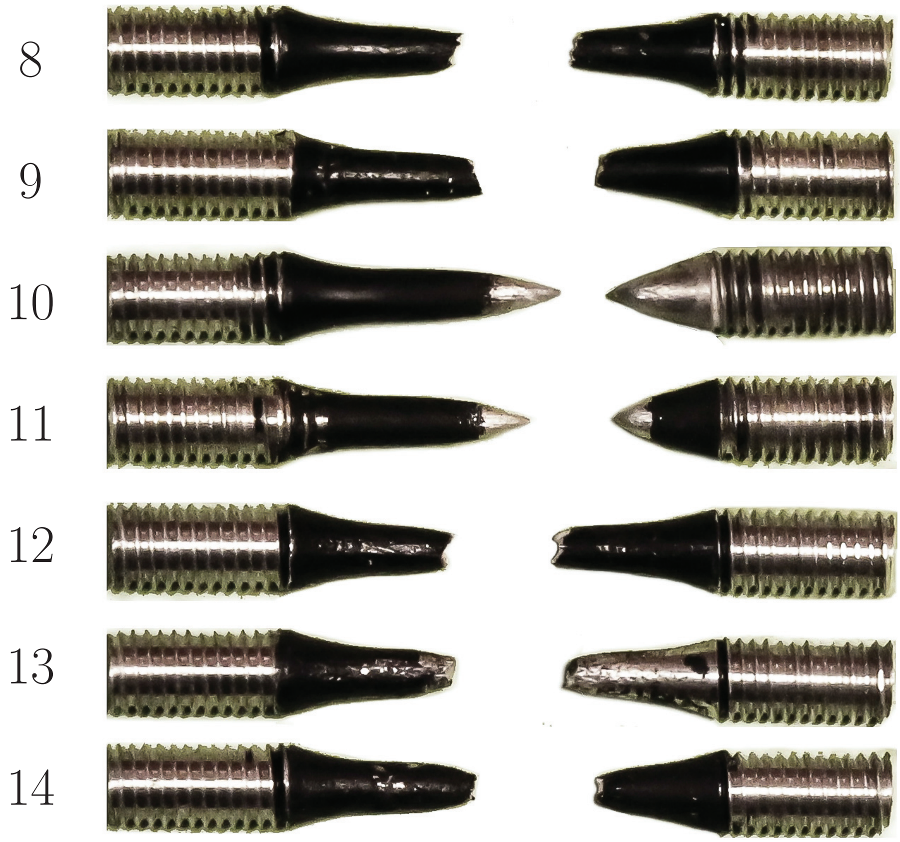


Figure G.5: Continues...

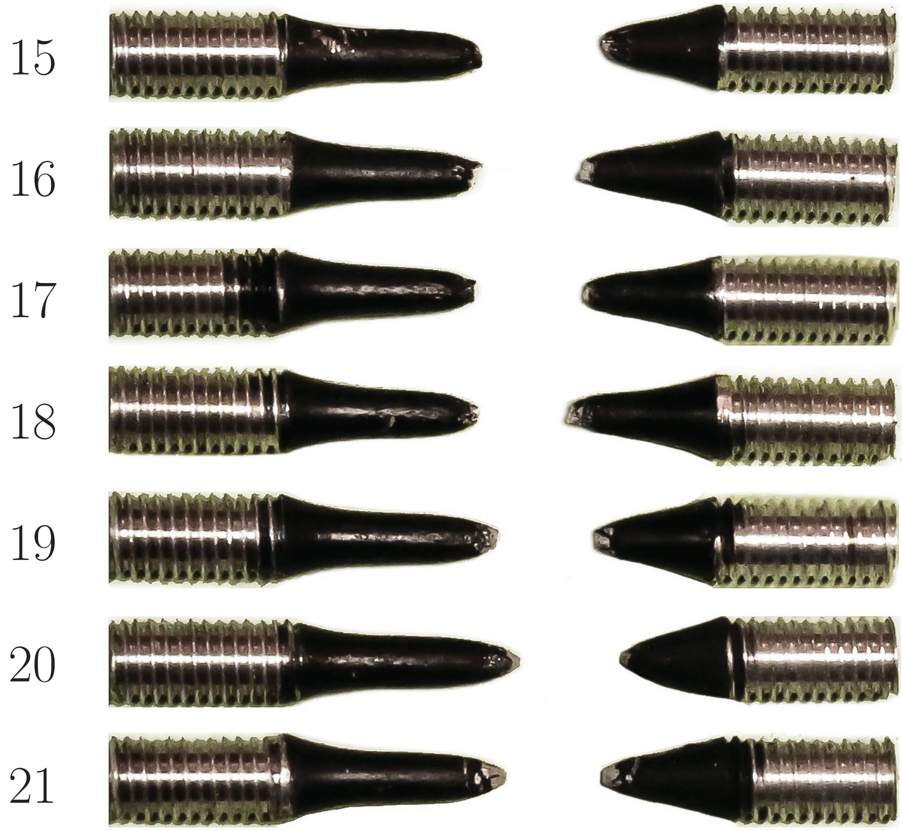


Figure G.5: Tensile specimens post-fracture for AA6060-H from all quasi-static tests

H Pictures of Specimens Post-Fracture from SHTB Tests

H.1 AA6060-L

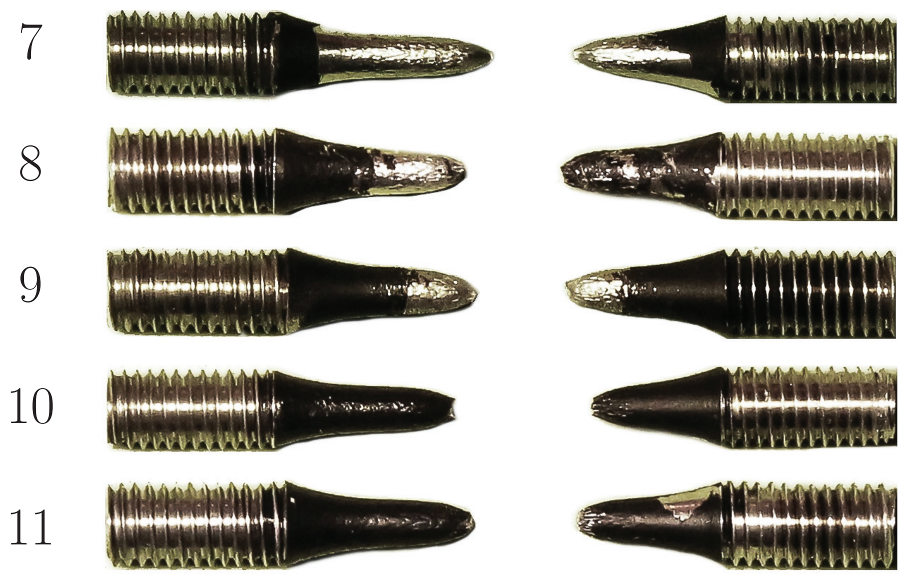


Figure H.1: Tensile specimens post-fracture for AA6060-L from all SHTB tests

H.2 AA6060-H

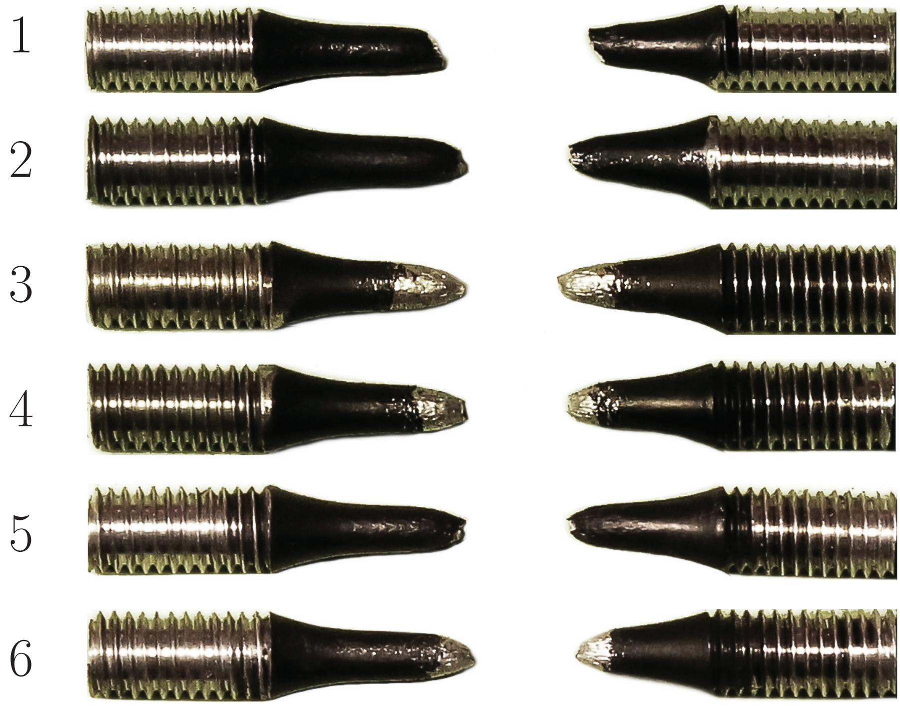


Figure H.2: Tensile specimens post-fracture for AA6060-H from all SHTB tests

I LS-DYNA Keyword File

```
$# LS—DYNA Keyword file created by LS—PrePost 4.1 (Beta) — ...
    22Apr2013(19:00)
$# Created on Jun—07—2013 (15:09:42)
*KEYWORD
*PARAMETER
R ya       48.026798
R qe1      45.312500
R ce1       3.487500
R qe2      27.351999
R ce2      52.274300
R p0        0.010000
R t0       523.00000
R tm       933.00000
R cp       9.7200E+8
R pd       260.00000
R a1        1.000000
R b1        1.000000
R c1        1.000000
R f1        1.000000
R g1        1.000000
R h1        1.000000
R m1        2.000000
R beta1        0.000
R nd1          0.000
*TITLE
$# title
LS—DYNA keyword deck by LS—PrePost
*BOUNDARY_PRESCRIBED_MOTION_SET_ID
$#      id ...
                                                                     ...
    heading
         0Stretching in A
$#    nsid       dof       vad      lcid        sf       vid     ...
    death     birth
         1         2         2         5  1.000000          ...
              01.0000E+28     0.000
$#      id ...
                                                                     ...
    heading
         0Clamping in B
$#    nsid       dof       vad      lcid        sf       vid     ...
    death     birth
         2         2         2         4  1.000000         0 ...
              3.0000E—4     0.000
*PART
$# title
Specimen
$#     pid     secid       mid     eosid      hgid      grav    ...
    adpopt      tmid
         1         1         1         0         1         0 ...
                     0         0
*SECTION_SHELL_TITLE
Specimen
```

```
$#   secid    elform      shrf       nip     propt   qr/irid     ...
     icomp     setyp
         1        15  1.000000         2         1         0 ...
                   0         1
$#      t1        t2        t3        t4      nloc     marea      ...
     idof    edgset
     0.000     0.000     0.000     0.000     0.000     0.000     ...
         0.000         0
*MAT_USER_DEFINED_MATERIAL_MODELS_TITLE
user defined vv
$#     mid        ro        mt       lmc       nhv    iortho     ...
     ibulk        ig
         1 2.5000E-9        41        32        21         0 ...
                  29        30
$#   ivect     ifail    itherm    ihyper      ieos      lmca    ...
     unused    unused
         0         1         0         0         0         0
$# p1             p2        p3        p4        p5        p6 ...
            p7        p8
6.900E+04 3.330E-01 &Ya       &qe1      &ce1      &qe2      &ce2 ...
         1.000e-004
$# p1             p2        p3        p4        p5        p6 ...
            p7        p8
2.22E-14  &p0        5   293.0 &tm       &Cp       2.700e    -0090.9
$# p1             p2        p3        p4        p5        p6 ...
            p7        p8
&pd       &t0       &tm       &a1       &b1       &c1       &f1 ...
          &g1
$# p1             p2        p3        p4        p5        p6 ...
            p7        p8
&h1       &m1       &BETA1    &ND1      6.900E+4  2.300E+4
*HOURGLASS
$#    hgid       ihq        qm       ibq        q1        q2    ...
     qb/vdc        qw
         1         4     0.000         0     0.000     0.000     ...
             0.000     0.000
*PART
$# title
Bars
$#     pid     secid       mid     eosid      hgid      grav    ...
     adpopt      tmid
         2         2         2         0         1         0 ...
                   0         0
*SECTION_SHELL_TITLE
Bars
$#   secid    elform      shrf       nip     propt   qr/irid     ...
     icomp     setyp
         2        15  1.000000         2         1         0 ...
                   0         1
$#      t1        t2        t3        t4      nloc     marea      ...
     idof    edgset
     0.000     0.000     0.000     0.000     0.000     0.000     ...
         0.000         0
*MAT_ELASTIC_TITLE
Bars
$#     mid        ro         e        pr        da        db  not used
         2 7.8500E-9 2.1000E+5  0.300000     0.000     0.000 ...
                   0
*DATABASE_DCFAIL
```

```
$#        dt    binary      lcur     ioopt
 1.0000E-6         0         0         1
*DATABASE_ELOUT
$#        dt    binary      lcur     ioopt   option1   option2   ...
     option3   option4
 1.0000E-6         0         0         1         0         0 ...
             0         0
*DATABASE_GLSTAT
$#        dt    binary      lcur     ioopt
 1.0000E-6         0         0         1
*DATABASE_MATSUM
$#        dt    binary      lcur     ioopt
 1.0000E-6         0         0         1
*DATABASE_NODOUT
$#        dt    binary      lcur     ioopt   option1   option2
 1.0000E-6         0         0         1     0.000         0
*DATABASE_RBDOUT
$#        dt    binary      lcur     ioopt
 1.0000E-6         0         0         1
*DATABASE_RCFORC
$#        dt    binary      lcur     ioopt
 1.0000E-6         0         0         1
*DATABASE_SECFORC
$#        dt    binary      lcur     ioopt
 1.0000E-6         0         0         1
*DATABASE_BINARY_D3PLOT
$#        dt      lcdt      beam     npltc    psetid
 3.0000E-6         0         0         0         0
$#   ioopt
         0
*DATABASE_BINARY_D3THDT
$#        dt      lcdt      beam     npltc    psetid
 1.5000E-6         0         0         0         0
*DATABASE_EXTENT_BINARY
$#   neiph     neips    maxint    strflg    sigflg    epsflg    ...
     rltflg    engflg
        20        20         3         1         1         1 ...
                     1         1
$#  cmpflg    ieverp    beamip     dcomp      shge     stssz    ...
     n3thdt   ialemat
         0         0         0         1         1         1 ...
                     2         0
$# nintsld   pkp_sen      sclp    unused     msscl     therm    ...
     intout    nodout
         0         0  1.000000                   0         0STRESS
$#    dtdt    resplt
         0         0
*DATABASE_HISTORY_NODE_SET
$#     id1       id2       id3       id4       id5       id6       ...
     id7       id8
         4         0         0         0         0         0 ...
                     0         0
*DATABASE_HISTORY_SHELL_SET
$#     id1       id2       id3       id4       id5       id6       ...
     id7       id8
         2         3         4         0         0         0 ...
                     0         0
*CONTROL_ENERGY
$#    hgen      rwen    slnten     rylen
```

```
         2         2         1         1
*CONTROL_IMPLICIT_AUTO
$#   iauto    iteopt    itewin     dtmin     dtmax     dtexp
         0        11         5     0.000     0.000     0.000
*CONTROL_IMPLICIT_GENERAL
$#  imflag       dt0    imform      nsbs       igs     cnstn      ...
    form    zero_v
        -3 1.0000E-5         2         1         2         0 ...
              0         1
*CONTROL_IMPLICIT_SOLUTION
$#  nsolvr    ilimit    maxref     dctol     ectol     rctol      ...
    lstol    abstol
         2        11        15  0.001000  0.0100001.0000E+10  ...
          0.9000001.0000E-10
$#   dnorm    diverg     istif   nlprint    nlnorm   d3itctl     cpchk
         2         1         1         0         2         0 ...
               0
$#  arcctl    arcdir    arclen    arcmth    arcdmp
         0         0     0.000         1         2
$#   lsmtd     lsdir      irad      srad      awgt      sred
         1         2     0.000     0.000     0.000     0.000
*CONTROL_SHELL
$#  wrpang     esort     irnxx    istupd    theory       bwc      ...
    miter      proj
     0.000         0         0         1        15         2 ...
            1         0
$# rotascl    intgrd    lamsht    cstyp6    tshell
     0.000         0         0         0         0
$# psstupd   sidt4tu     cntco    itsflg    irquad
         0         0         0         0         2
$#  nfail1    nfail4   psnfail    keepcs     delfr   drcpsid    drcprm
         0         0         0         0         0         0  1.000000
*CONTROL_SOLUTION
$#    soln       nlq     isnan     lcint
         0         0         0       100
*CONTROL_TERMINATION
$#  endtim    endcyc     dtmin    endeng    endmas
  0.002500         0     0.000     0.000     0.000
*CONTROL_TIMESTEP
$#  dtinit    tssfac      isdo    tslimt     dt2ms      lctm      ...
    erode     ms1st
     0.000  0.400000         0     0.000     0.000         0 ...
            0         0
$#  dt2msf   dt2mslc     imscl    unused    unused     rmscl
     0.000         0         0                         0.000
```