Droplet-Based Processing of Magnesium Alloys for the Production of High-Performance Bulk Materials

A Dissertation Presented

by

Hiroki Fukuda

to

The Department of Mechanical and Industrial Engineering

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in

Mechanical Engineering

in the field of

Material Science and Engineering

Northeastern University
Boston, Massachusetts

August, 2009
ABSTRACT

The magnesium consumption worldwide has been rapidly increasing in the recent years as magnesium alloys replace other materials in a variety of structural applications in industrial machinery, materials handling, commercial and aerospace equipment, and more recently in audio-video-computer-communication equipment and automobiles. Development of new high-performance alloys, however, is hampered in part by magnesium’s pyrophoricity, which poses a serious barrier to applying conventional droplet-based processing to magnesium alloys. This research addresses the processing of high-performance magnesium alloys with mono-size droplets produced by the capillary breakup-based uniform-droplet spray (UDS) process which permits stringent control of the thermal state of droplets while assuring safe spraying. Strengthening magnesium alloys for elevated temperature service is another key to increasing the magnesium utilization in manufacturing. The Mg-Zn-Y system, having various intermetallic phases such as the I-phase (Mg$_3$Zn$_6$Y$_1$), X-phase (Mg$_{12}$ZnY) and W-phase (Mg$_3$Zn$_3$Y$_2$), offers unique opportunities for the development of high-performance Mg alloys.

Three Mg-Zn-Y alloys, Mg$_{97}$Zn$_1$Y$_2$, Mg$_{88}$Zn$_{10}$Y$_2$ and Mg$_{76.5}$Zn$_{20}$Y$_{3.5}$, were chosen for this research to investigate the potentials for the droplet-based processing of high-performance alloys by the UDS process. Spray deposits produced with uniform (mono-size) droplets 700-1000 µm in diameter were characterized by full density and fine equiaxed grains, in contrast to the coarse dendritic structure of the original cast ingot. While the primary equiaxed grains were the HCP Mg-rich solid solution in the deposits of all three compositions, the Mg$_{97}$Zn$_1$Y$_2$ deposits also had X-phase, and the Mg$_{88}$Zn$_{10}$Y$_2$ and
Mg\textsubscript{76.5}Zn\textsubscript{20}Y\textsubscript{3.5} deposits had the \textit{I}- and \textit{W}-phases in the intergranular regions. The primary grains in the Mg\textsubscript{97}Zn\textsubscript{1}Y\textsubscript{2} deposits exhibited striations indicative of \textit{X}-phase precipitation on basal planes. The occurrence of \textit{I}-phase in the Mg\textsubscript{88}Zn\textsubscript{10}Y\textsubscript{2} and Mg\textsubscript{76.5}Zn\textsubscript{20}Y\textsubscript{3.5} alloys is in line with previous reports on the existence of a pseudo-binary system between Mg and \textit{I}-phase.

Subsequent thermomechanical processing of the spray deposits by rolling resulted in further microstructural refinement and improved room- and elevated-temperature strength, especially in the case of Mg\textsubscript{97}Zn\textsubscript{1}Y\textsubscript{2} which had a room-temperature tensile strength of 340 MPa and an elevated-temperature (473K) tensile strength of 271 MPa. The other two alloys, Mg\textsubscript{88}Zn\textsubscript{10}Y\textsubscript{2} and Mg\textsubscript{76.5}Zn\textsubscript{20}Y\textsubscript{3.5}, respectively, had 305 MPa and 271 MPa at room temperature, and 237 MPa and 216 MPa at 473 K. These strength values significantly exceed those of commercial wrought Mg alloys, which typically range 200 MPa to 300 MPa at room temperature and 150 MPa to 200 MPa at 473 K. Although the process parameters still need to be further optimized, UDS processing, combined with thermomechanical processing, enables safe and effective processing of high-performance magnesium alloys.
ACKNOWLEDGMENTS

To Professor Teiichi Ando goes my most sincere and heartfelt gratitude for giving me this wonderful opportunity of a lifetime. His constant words of wisdom and encouragement have helped me through difficult times in the course of this study and will continue to do so throughout the rest of my life. I would like to thank my thesis committee members Professor Peter Y. Wong of Tufts University and Professor Yung Joon Jung of Northeastern University for their time and valuable guidance. I appreciated Dr. Wong’s great assistance with the tensile tests. I would also like to thank Professors Hongwei Sun and Haris Doumanidis for their help throughout this project.

I would like to thank Fukuda Metal Foil & Powders Co., Ltd., for funding this research project through the first and last years of the project and providing this chance to further my education. I also wish to acknowledge the financial support of the National Science Foundation (DMII 0423228) in 2005 - 2008. Without these funding sources this research would not have been possible.

I am fortunate to be one of the proud members of the Advanced Materials Laboratory (AMPL) at Northeastern University, which has been the aboard of many talented individuals who all helped me and dealt me with friendship.

I would like to thank especially Rajesh Ranganathan for his consistent and great assistance, help, honesty and unconditional friendship throughout my 10 years of this graduate student life. Thanks to Jason Small for his friendship and contributions to my personal and professional development. I would like to thank Dr. Ping Wu for the opportunity to work with her and learn from her during my first year at AMPL. Thanks
to my collaborator Pengtao Wang of University of Massachusetts-Lowell. My thanks go to Suresh Kumar Pillai, Gokce Gulsoy, Sudesna Roy, Emre Gunduz, David Colanto, Dinc Erdeniz, Bora Islier, Faryar Tavakoli, Charles Tuffiìle, Younes Abesi, Hui Lu, Debin Zhang, Olga Beliavskaya, GuangKai Shi, Michael Beauregard, Yangwei Song, Selis Onel, Ceren Cura Eraslan, Laura Xu, Dr. Yoshinori Nishida and Dr. Kozo Nakazawa for all their assistance and for making the lab what it is.

I am indebted to the Mechanical and Industrial Engineering Department of Northeastern University for providing various forms of lab space, resources, and dynamic learning environment. I would like to specially mention and thank John Doughty for his assistance in the machine shop. My gratitude goes to other members of the MIE Department for all their professional assistance.

Finally, I would like to thank my family and friends for constantly encouraging me in various ways to remain focused on my goals.
# TABLE OF CONTENTS

ABSTRACT ........................................................................................................................................................................... ii
ACKNOWLEDGMENTS ........................................................................................................................................................... iv
TABLE OF CONTENTS ........................................................................................................................................................... vi
LIST OF FIGURES ................................................................................................................................................................. viii
LIST OF TABLES ................................................................................................................................................................. xvi
1. INTRODUCTION ................................................................................................................................................................. 1
   1.1 Magnesium Alloys and Applications .......................................................................................................................... 1
   1.2 Objective of Research .................................................................................................................................................... 4
2. BACKGROUND ......................................................................................................................................................................... 5
   2.1 Mg-Zn-Y Alloys .............................................................................................................................................................. 5
      2.1.1 Mg-Zn-Y alloy ternary phases ......................................................................................................................... 7
      2.1.2 Phase diagrams of Mg, Zn and Y and their compounds ............................................................................... 17
   2.2 Mg Alloys Processing .................................................................................................................................................... 22
      2.2.1 Conventional processing ...................................................................................................................................... 22
      2.2.2 Uniform-droplet spray process .......................................................................................................................... 24
      2.2.3 Preliminary UDS works with conventional magnesium alloys ........................................................................ 32
   2.3 Modeling of the Droplet Production Process ................................................................................................................ 35
      2.3.1 Prediction of droplet thermal state ..................................................................................................................... 35
3. EXPERIMENTAL SET-UP AND PROCEDURE .................................................................................................................. 38
   3.1 Materials ............................................................................................................................................................................ 38
   3.2 UDS Apparatus .............................................................................................................................................................. 40
      3.2.1 Heating system ....................................................................................................................................................... 43
      3.2.2 Crucible and insert design .................................................................................................................................... 48
      3.2.3 Vibration transmission system .......................................................................................................................... 55
      3.2.4 Droplet deposition system ................................................................................................................................... 58
   3.3 Experimental Procedure .................................................................................................................................................. 62
      3.3.1 Deposit production scheme .................................................................................................................................... 62
      3.3.2 Droplet-in-flight solidification model .................................................................................................................. 66
      3.3.3 Process-structure relationships .......................................................................................................................... 66
4. THERMAL/SOLIDIFICATION MODELING ....................................................................................................................... 68
   4.1 Mg-I Phase Pseudo-Binary Phase Diagram .................................................................................................................. 68
   4.2 In-Flight Solidification ................................................................................................................................................... 71
4.2.1 Droplet motion model ................................................................. 72
4.2.2 Droplet thermal model .............................................................. 76
4.2.3 Droplet solidification model ...................................................... 78

5. RESULTS AND DISCUSSION .......................................................... 83

5.1. Droplets and Deposits Production Experiments ............................... 83
  5.1.1 AZ91D ......................................................................................... 83
  5.1.2 Mg97Zn1Y2 ............................................................................... 85
  5.1.3 Mg88Zn10Y2 ............................................................................. 114
  5.1.4 Mg76.5Zn20Y3.5 ....................................................................... 126
  5.1.5 Splat collection ......................................................................... 137
          5.1.5.1 Overview of numerical simulation of droplet deformation .......... 151

5.2 Rolling Tests .................................................................................. 154

5.2 Tensile Tests .................................................................................. 172
  5.2.1 Tensile strength testing procedures ........................................... 172
  5.2.2 Tensile test results ..................................................................... 177

6. CONCLUSIONS ............................................................................... 198

7. APPENDIX ..................................................................................... 202

Appendix A: X-Y Table Control System .............................................. 202

  C source code for X-Y table motion control .................................. 207
  Example source code for an input file .......................................... 215
  Example source code for XSLT and CSS .................................... 218
  Example source code for XML schema ....................................... 222
  Java source code for X-Y table motion simulator applet ............... 224

REFERENCES ................................................................................. 260
LIST OF FIGURES

Figure 2-1: SEM image of I-phase quasicrystal found in Zn$_{50}$Mg$_{45}$Y$_5$ alloy by Niikura et al. [82]................................................................................................................................. 9

Figure 2-2: Schematic of (a) the Mackay type approximant in Mn-Al-Si system, (b) the Bergman type approximant in Mg-Al-Zn system and (c) the Tsai-type cluster in the i-Yb–Cd quasicrystal atomic structures [93]. ...................................................... 13

Figure 2-3: A structure model for the face centered icosahedral (fci) Mg$_{25}$Y$_{11}$Zn$_{64}$ alloy. White, grey and dark grey balls represent Mg, Zn and Y atoms, respectively. [111]........................................................................................................................................... 14

Figure 2-4: High-resolution image of the X-phase projected along the [100] orientation [69].................................................................................................................................................. 16

Figure 2-5: Mg-rich corner of the ternary Mg-Zn-Y phase diagram. (a) liquidus projection and (b) solidus projection [81, 122]........................................................................................................... 18

Figure 2-6: Ternary Mg-Zn-Y phase diagram at 573 K reported by Padezhnova et al. [118].................................................................................................................................................. 19

Figure 2-7: Equilibrium phase diagram of the Mg-Zn system [120]. ........................................ 19

Figure 2-8: Equilibrium phase diagram of the Mg-Y system [119]. ......................................... 20

Figure 2-9: Equilibrium phase diagram of the Y-Zn system [121]. ......................................... 20

Figure 2-10: Schematic of the UDS droplet generator. ............................................................... 26

Figure 2-11: Measured and predicted enthalpy of 185 µm Sn-5wt%Pb droplets [27]. .... 26

Figure 2-12: 285 µm mono-size Cu droplets showing the size uniformity and sphericity of droplets produced by the UDS process [39]................................................................. 29

Figure 2-13: Single-crystalline Sn-5 wt% Pb deposit produced on single-crystal Sn substrate [34]. ......................................................................................................................... 29

Figure 2-14: Sn-5wt%Pb spray deposit produced with 185µm droplets. .............................. 30

Figure 2-15: 1000 µm Mg alloy (AZ91D) droplets ................................................................. 31

Figure 2-16: 50 µm lead-free Sn-base solder alloy (Sn-Ag-Cu) droplets. ............................ 31

Figure 2-17: Uniform breakup of a molten AM60B jet [40].................................................. 33

Figure 2-18: 200 µm AZ91D droplets oil-quenched in molten (left) and mushy states (right) [40] ......................................................................................................................... 34

Figure 2-19: Fine α-Mg dendrites in 850 µm AM60B droplet [40]. ........................................ 34

Figure 2-20: Calculated dendritic growth rate for a Mg-6 wt.% Al alloy [40].................. 37

Figure 2-21: Calculated droplet temperature and fraction solid of 200, 800 and 2000 µm Mg-6Al droplets [40]. ........................................................................................................ 37
Figure 3-1: Mg-rich corner of the Mg-Zn-Y Gibbs triangle. The red region shown here encompasses alloys that form the X-phase as a typical secondary phase, while alloys in the blue region form the I-phase as a secondary phase. 39

Figure 3-2: UDS apparatus in AMPL lab. 41

Figure 3-3: A schematic of the side view of UDS apparatus. 42

Figure 3-4: A photograph showing the induction coil without assembly. 43

Figure 3-5: Schematic of the layout of induction unit. 44

Figure 3-6: Coil and crucible assembly. 46

Figure 3-7: Alumina insulation block used inside the upper part of the UDS chamber. 47

Figure 3-8: CAD drawing of the top and bottom part of the graphite crucible. 49

Figure 3-9: CAD drawing of the graphite insert. 49

Figure 3-10: CAD drawing of the stainless steel bottom plate with the insert. 51

Figure 3-11: The stainless steel bottom plate and the insert with orifice hole. 52

Figure 3-12: Schematic of the wire-pull method. 54

Figure 3-13: Crucible bottom part with the wire-pull system. 54

Figure 3-14: Schematic of the piezoelectric vibration assembly. 57

Figure 3-15: Schematic of the transducer tip. 57

Figure 3-16: Photograph of the substrate motion system placed in the UDS chamber. 59

Figure 3-17: Schematic of the substrate motion system. 60

Figure 3-18: Schematic of the current setting of deposition experiment. 60

Figure 3-19: Schematic illustration of the substrate system. 61

Figure 3-20: Schematic of the droplet deposition along a programmed path. 67

Figure 4-1: The Mg - I-phase pseudo binary phase diagram obtained by linear-fitting of literature values of liquidus and solidus by Zaselyan et al. [144]. The horizontal line was drawn at 713 K to intersect with the liquidus at the eutectic composition of Mg_{73.2}Zn_{23}Y_{3.8} [44]. 70

Figure 4-2: Schematic showing the forces acting on a droplet moving in gaseous medium. 74

Figure 4-3: Schematic diagram of the thermal and solutal profiles across the tip of a free dendrite. The dendrite tip is approximated by a paraboloid of revolution with radius R [133]. 79

Figure 4-4: Schematic of a non-linear binary phase diagram. 79

Figure 5-1: Break-up image captured during AZ91D UDS experiment. 81

Figure 5-2: AZ91D deposit produced with 700 µm in order to optimize motion parameters for the X-Y table. 83

Figure 5-3: Break-up image captured from the Mg_{97}Zn_{1}Y_{2} UDS experiment. 87
Figure 5-4: Temperature vs. flight distance of 1: 700 µm, 2: 1000 µm, and 3: 700 µm for $Mg_{97}Zn_{1}Y_{2}$ droplets calculated for assumed undercoolings $\Delta T=50K$, for melt temperatures of 973 K and 1073 K. ................................................................. 88

Figure 5-5: Cross section of the $Mg_{97}Zn_{1}Y_{2}$ ingot filled with coarse dendritic structures. 89

Figure 5-6: Optical micrographs of the bulk ingot supplied from FUKUDA. (a) Low magnification. (b) High magnification................................................................. 90

Figure 5-7: Optical micrographs showing the spray boundary in the spray deposit produced with the 700 µm $Mg_{97}Zn_{1}Y_{2}$ droplets (Experiment 1). .......... 91

Figure 5-8: Optical micrographs taken near the chill surface in the spray deposit produced with 700 µm $Mg_{97}Zn_{1}Y_{2}$ droplets (Experiment 1). (a) low magnification, (b) medium magnification, and (c) high magnification. ............................................. 92

Figure 5-9: XRD pattern of $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 700 µm droplets from Experiment 1. ................................................................. 95

Figure 5-10: Optical micrographs taken near the chill surface in the $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 1000 µm droplets (Experiment 2), (a) low magnification and (b) high magnification. ................................................................. 98

Figure 5-11: Optical micrographs taken near the top of the deposit in the $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 1000 µm droplets (Experiment 1), (a) low magnification and (b) high magnification. ................................................................. 99

Figure 5-12: Optical micrographs of $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 700 µm droplets (Experiment 3), (a) near top of the deposit and (b) near the chill surface. 102

Figure 5-13: Optical micrographs of $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 700 µm droplets at a spray boundary (Experiment 3), (a) low magnification and (b) high magnification. ................................................................. 103

Figure 5-14: Comparison of microstructures at the spray layer boundaries of the deposits from (a) Experiment 1 (700 µm droplets), (b) Experiment 2 (1000 µm droplets) and (c) Experiment 3 (700 µm droplets). ................................................................. 104

Figure 5-15: XRD pattern of $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 1000 µm droplets (Experiment 2)................................................................. 106

Figure 5-16: XRD pattern of $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 700 µm droplets (Experiment 3)................................................................. 106

Figure 5-17: EPMA analysis of $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 700 µm droplets (Experiment 3). (a) Back-scattered image indicating concentration of the heavier elements Zn and Y in the intergranular regions. (b)-(d) Color mappings of Mg, Zn and Y confirming the enrichment of Zn and Y in the
intergranular region and high Mg content in the equiaxed primary grains.

Figure 5-18: Back-scattered electron image of the region subjected to EDX analysis. The specimen is from the \( \text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2} \) deposit produced with 700 \( \mu \text{m} \) droplets (Experiment 3), a: dark region - the equiaxed grain, b: gray region - the intergranular region. The EDX results are listed in Table 5-3. .................. 110

Figure 5-19: Microstructure of Ti-Al alloy which has striated morphologies resembling that of the \( \text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2} \) deposit produced with 700 \( \mu \text{m} \) droplets. (a): Ti - 47.5 at\% Al. (b): Ti - 48.2 at\% Al [166]............................................................. 113

Figure 5-20: Optical micrographs of the bulk ingot of \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_{2} \) supplied from FUKUDA. (a) Low magnification. (b) High magnification.................. 115

Figure 5-21: XRDS pattern of \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_{2} \) as-cast ingot............................................................. 116

Figure 5-22: Optical micrographs of the bulk ingot of \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_{2} \). Region A, B and C represent \( \alpha\)-Mg, \( \alpha\)-Mg and I and/or W eutectic, and probably I-phase, respectively. .......................................................... 116

Figure 5-23: Break-up image captured from the \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_{2} \) UDS experiment. ............ 117

Figure 5-24: \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_{2} \) deposit produced with 700 \( \mu \text{m} \) droplets. ........................................ 118

Figure 5-25: Optical micrographs of the spray deposit of the \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_{2} \) alloy. (a) Low magnification. (b) High magnification. ............................. 119

Figure 5-26: XRDS patterns of the \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_{2} \): (a) as-sprayed deposit and (b) as-cast ingot. As-sprayed deposit has lower relative intensities of I-phase with respect to those of \( \alpha\)-Mg. ............................................................................. 122

Figure 5-27: EPMA analysis of the \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_{2} \) deposit produced with 700 \( \mu \text{m} \) droplets. (a) Back-scattered image indicating concentration of the heavier elements Zn and Y in the intergranular regions. (b)-(d) Color mappings of Mg, Zn and Y confirming the enrichment of Zn and Y in the intergranular region and high Mg content in the equiaxed primary grains. .......................................................... 123

Figure 5-28: Back-scattered electron image of the region subjected to EDX analysis. The specimen is from the \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_{2} \) deposit produced with 700 \( \mu \text{m} \) droplets. a: dark region - the equiaxed grain. b: white region - the intergranular region. c: lamellar region. The EDX results are listed in Table 5-5. .................. 125

Figure 5-29: Optical micrographs of the bulk ingot of \( \text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5} \) supplied from FUKUDA. (a) Low magnification. (b) High magnification. .................. 128

Figure 5-30: XRDS pattern of \( \text{Mg}_{88}\text{Zn}_{20}\text{Y}_{3.5} \) as-cast ingot. \( \alpha\)-Mg, \( W\)-phase, I-phase and \( \text{Mg}_{7}\text{Zn}_{3} \) are identified. The binary intermetallic compound of \( \text{Mg}_{7}\text{Zn}_{3} \) is thought to be a relevant phase for the formation of I-phase. .................. 129
Figure 5-31: Break-up image captured from the $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ UDS experiment. ... 130
Figure 5-32: $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposit produced with 700 $\mu$m droplets. The patterned tail which was sprayed next to the main deposit, was deposited to test the motion of the X-Y table. ... 131
Figure 5-33: Optical micrographs of the spray deposit of the $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ alloy. (a) Low magnification. (b) High magnification. ... 132
Figure 5-34: XRD pattern of $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposit, (a) Spray deposit, (b) As-cast ingot. ... 133
Figure 5-35: EPMA analysis of the $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposit produced with 700 $\mu$m droplets. (a) Back-scattered image indicating concentration of the heavier elements Zn and Y in the intergranular regions. (b)-(d) Color mappings of Mg, Zn and Y confirming the enrichment of Zn and Y in the intergranular region and high Mg content in the primary grains. ... 134
Figure 5-36: Back-scattered electron image of the region subjected to EDX analysis. The specimen is from the $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposit produced with 700 $\mu$m droplets. a: dark region – the dendritic region. b: white region - the interdendritic region. c: the lamellar eutectic region. The EDX results are listed in Table 5-7. ... 136
Figure 5-37: (a): Photograph of the droplet quencher. (b): Schematic of the droplet quenching apparatus attached to the X-Y table. ... 138
Figure 5-38: The $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ splat collected from the droplet quencher at a flight distance of 0.25 m. (a) The $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ splat deformed into a pancake shape. (b) Bottom view of the $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ splat. (Chill side). The splat is elongated due to shear caused by the quencher blade motion. (c) Top view of the $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ splat. ... 139
Figure 5-39: Longitudinal cross section of the $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ splat. ... 140
Figure 5-40: Longitudinal cross section of the $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ splat. (a): Edge region (b): Middle region. ... 141
Figure 5-41: (a): Illustration of scattered droplets that took flight paths that inclined from the main droplet stream and landed at different places on the substrate (b): $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ droplet that landed outside the target position. ... 144
Figure 5-42: The $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ splat in Figure 5-41. (a): Side view. (b): Top view. (c): Bottom view. ... 145
Figure 5-43: Cross sectional micrograph of the $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ splat in Figures 5-41 and 5-42. (a): Entire cross section of the splat. (b): Medium magnification image for
the entire section. (c): High magnification image taken from near the top region. (d): High magnification image taken at the near chill region.

Figure 5-44: Temperature vs. flight distance of “aligned” and “isolated” 700 μm $Mg_{97}Zn_{1}Y_{2}$ droplets.

Figure 5-45: Cross section of a $Mg_{97}Zn_{1}Y_{2}$ splat obtained near the spray deposit. (a): Low magnification. (b): High magnification.

Figure 5-46: Simulated velocity vectors and temperature contours of a 693 μm $Mg-Zn-Y$ droplet quenched on a substrate [169].

Figure 5-47: Simulated temperature contour and solid-liquid interface position during rapid solidification of a 693 μm $Mg-Zn-Y$ droplet quenched on a substrate [169].

Figure 5-48: $Mg_{97}Zn_{1}Y_{2}$ deposit rolled under the condition A listed in Table 5-8.

Figure 5-49: $Mg_{97}Zn_{1}Y_{2}$ deposit rolled under the condition B listed in Table 5-8. (a): Before rolling. (b): After rolling.

Figure 5-50: Optical micrographs of rolled $Mg_{97}Zn_{1}Y_{2}$ deposit preheated to 473 K for 20 min prior to rolling. (a): Low magnification. (b): Medium magnification. (c): High magnification.

Figure 5-51: Optical micrographs of the rolled $Mg_{97}Zn_{1}Y_{2}$ deposit preheated to 673 K for 20 min prior to rolling. (a): Low magnification. (b): High magnification.

Figure 5-52: Optical micrographs of rolled $Mg_{88}Zn_{10}Y_{2}$ deposit preheated to 673 K for 20 min prior to rolling. (a): Low magnification. (b): High magnification.

Figure 5-53: Optical micrographs of rolled $Mg_{76.5}Zn_{20}Y_{3.5}$ deposit preheated to 673 K for 20 min prior to rolling. (a): Low magnification. (b): High magnification.

Figure 5-54: XRD pattern of the $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 700 μm droplets. (a): Annealed specimen. (b): As-rolled specimen. (c): As-sprayed specimen.

Figure 5-55: XRD pattern of the $Mg_{88}Zn_{10}Y_{2}$ deposit produced with 700 μm droplets. (a): Annealed specimen. (b): As-rolled specimen. (c): As-sprayed specimen.

Figure 5-56: XRD pattern of the $Mg_{76.5}Zn_{20}Y_{3.5}$ deposit produced with 700 μm droplets. (a): Annealed specimen. (b): As-rolled specimen. (c): As-sprayed specimen.

Figure 5-57: Optical micrographs of the specimens used for Vickers hardness measurement. (a): As-sprayed $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 1000 μm droplets. (b): As-sprayed $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 700 μm droplets.

Figure 5-58: Vickers hardness values of the as-sprayed deposits of $Mg_{97}Zn_{1}Y_{2}$, $Mg_{88}Zn_{10}Y_{2}$ and $Mg_{76.5}Zn_{20}Y_{3.5}$ alloy produced with 700 μm droplets in comparison with
those of conventional Mg alloys. (Vickers hardness values of ASTM Mg alloys are estimated from Brinell hardness values.)

Figure 5-59: Drawing showing the dimensions of the tensile test specimen.

Figure 5-60: Photograph of tensile test specimens machined from $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ spray deposits.

Figure 5-61: Instron tensile machine for room temperature testing (INSTRON-3366).

Figure 5-62: Instron tensile machine for elevated temperature testing (INSTRON-5864).

Figure 5-63: The clamping grips of the tensile test machine.

Figure 5-64: Dimensions of the tensile test specimen for elevated temperature tensile tests.

Figure 5-65: Photograph showing the fractured tensile test specimens produced from the as-rolled $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ deposits (700 $\mu$m droplets).

Figure 5-66: Load vs. displacement curves at room temperature of the as-sprayed, rolled and annealed $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ specimens.

Figure 5-67: Load vs. displacement curves at the room temperature of the as-sprayed and as-rolled specimens of (a) $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_{2}$ deposit, and (b) $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposit.

Figure 5-68: Micrograph of the tensile specimen grip area, (a) showing deformation at the tensile specimen grip. (b) An approximately 70 $\mu$m slippage on each grip occurred due to the deformation.

Figure 5-69: Comparison of the room-temperature tensile properties of rolled $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$, $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_{2}$, and $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ with those of the commercial wrought and cast Mg alloys [3, 170]. (a): UTS. (b): 0.2% Proof stress. Rolled $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ spray deposit shows superior UTS value than those of wrought Mg alloys.

Figure 5-70: Micrographs of the fracture surface of the as-sprayed $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ deposit after room temperature tensile test, (a) Low magnification SEM image showing the edge of the specimen, (b) Optical micrograph of the longitudinal cross section.

Figure 5-71: Micrographs of the fracture surface of the as-rolled $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ deposit after room temperature tensile test, (a) Low magnification SEM image, (b) Optical micrograph of the longitudinal cross section.

Figure 5-72: Micrographs of the fracture surface of the rolled and annealed $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ deposit after room temperature tensile test, (a) Low magnification SEM image, (b) Optical micrograph of the longitudinal cross section.
Figure 5-73: Micrographs of the fracture surface of the as-rolled $Mg_{88}Zn_{10}Y_2$ deposit after room temperature tensile test, (a) Low magnification SEM image, (b) Optical micrograph of the longitudinal cross section showing cracks propagating along the intergranular regions. .............................................. 190

Figure 5-74: Micrographs of the fracture surface of the as-rolled $Mg_{76.5}Zn_{20}Y_{3.5}$ deposit after room temperature tensile test, (a) Low magnification SEM image, (b) Optical micrograph of the longitudinal cross section showing cracks forming in the intergranular regions. .............................................. 191

Figure 5-75: Load vs. displacement curve of rolled and annealed $Mg_{97}Zn_{1}Y_2$ deposit tested at 473 K. ..................................................................................................... 192

Figure 5-76: Load vs. displacement curve of rolled and annealed $Mg_{87}Zn_{10}Y_2$ deposit tested at 473 K. ..................................................................................................... 192

Figure 5-77: Load vs. displacement curve of rolled and annealed $Mg_{76.5}Zn_{20}Y_{3.5}$ deposit tested at 473 K ..................................................................................................... 193

Figure 5-78: Tensile specimen of the rolled and annealed $Mg_{97}Zn_{1}Y_2$ spray deposit after the tensile test at 473 K. ..................................................................................................... 194

Figure 5-79: Optical micrograph of the longitudinal cross section of the fracture area of the rolled and annealed $Mg_{97}Zn_{1}Y_2$ deposit after tensile testing at 473 K, showing the elongated grains. ........................................................................... 195

Figure 5-80: Optical micrograph of the longitudinal cross section of the fracture area of the rolled and annealed $Mg_{88}Zn_{10}Y_2$ deposit after elevated temperature tensile testing at 473 K. ................................................................................... 195

Figure 5-81: Optical micrograph of the longitudinal cross section of the fracture area of the rolled and annealed $Mg_{76.5}Zn_{20}Y_{3.5}$ deposit after tensile testing at 473 K, showing cracks forming in intergranular region ........................................... 196

Figure 5-82: Comparison of the elevated-temperature (473 K) tensile properties of rolled and annealed $Mg_{97}Zn_{1}Y_2$, $Mg_{88}Zn_{10}Y_2$, and $Mg_{76.5}Zn_{20}Y_{3.5}$ with those of the commercial Mg alloys [170-172]. (The UTS values of AZ91D, AE42, AS21, WE43-T6 and ZE41-T5 alloys are extrapolated values from a literature data available within a range between 293 K – 423 K (20 °C – 180 °C) [171].) Rolled and annealed $Mg_{97}Zn_{1}Y_2$, $Mg_{88}Zn_{10}Y_2$, and $Mg_{76.5}Zn_{20}Y_{3.5}$ spray deposits show superior UTS values than those of the commercial Mg alloys. ................................................................................... 197

Figure 7-1: Source codes used to control X-Y table ................................................................... 204

Figure 7-2: Error massages suggest possible problems for an input file. .............................. 205
Figure 7-3: X-Y table motion simulator demonstrates the Java applet running with a web browser.

LIST OF TABLES

Table 2-1: Classification of Quasicrystals [90-92]................................................................. 10
Table 2-2: Partial Compounds List in Zn-Mg-Y System. [83, 86, 117, 119-121, 123-131] .............................................................................................................................................................. 21
Table 3-1: Specifications of the Induction Unit........................................................................ 44
Table 5-1: Experimental Parameters....................................................................................... 84
Table 5-2: Experimental Parameters for the $Mg_{97}Zn_{1}Y_{2}$ Deposits................................. 87
Table 5-3: EDX Result for the $Mg_{97}Zn_{1}Y_{2}$ Deposit Produced with 700 µm Droplets  (Experiment 3) ........................................................................................................................................ 112
Table 5-4: Experimental Parameters for the $Mg_{88}Zn_{10}Y_{2}$ Deposit.................................. 117
Table 5-5: EDX Result for the $Mg_{88}Zn_{10}Y_{2}$ Deposit Produced with 700 µm Droplets... 125
Table 5-6: Experimental Parameters for the $Mg_{76.5}Zn_{20}Y_{3.5}$ Deposits.............................. 130
Table 5-7: EDX Result for the $Mg_{88}Zn_{10}Y_{2}$ Deposit Produced with 700 µm Droplets... 136
Table 5-8: Rolling Test Condition and Results....................................................................... 156
Table 5-9: Vickers Microhardness Results of the different Mg-Zn-Y Alloys......................... 169
Table 5-10: Mg-Zn-Y Spray Deposit Room Temperature Tensile Test Results..................... 184
Table 5-11: Elevated Temperature Tensile Test Result of Mg-Zn-Y Spray Deposits..... 194
1. INTRODUCTION

1.1 Magnesium Alloys and Applications

Conventional magnesium alloys (e.g., AS41, AM60, AZ91, AZ31, and ZK60) have moderate room-temperature strength, but their hcp crystal structure puts limits to room-temperature ductility and toughness [1]. Casting alloys, represented by Alloys AS41, AM60 and AZ91, have room-temperature tensile yield strengths ranging from about 100 to 200 MPa and elongations less than 10%. Wrought alloys, represented by Alloys AZ31 and ZK60, have higher room-temperature tensile yield strengths between 200 and 300 MPa. Their % elongations often exceed 10%, but only up to 15% or less. The generally low room temperature ductility of magnesium alloys is intrinsic to the hcp structure, which has restricted slip capacity. Commercial alloys in general also have low creep resistance due to their low Young’s moduli, low melting points, hcp structure [2] and precipitation of intermetallic compounds, which adversely affects the creep strength. For example, in die cast AZ91, the supersaturated $\alpha$-Mg decomposes into a lamellar structure by discontinuous precipitation of $\text{Mg}_{17}\text{Al}_{12}$ alloy microstructure at elevated temperature. This decomposition of the microstructure is considered to be a primary reason for the low creep resistance of die cast AZ91 alloy. Grain boundary migration and sliding, which is the predominant creep mechanism in magnesium, is promoted by the discontinuous precipitation reaction [2, 3].

In recent years, magnesium alloy utilization is increasing worldwide at a rate exceeding 20%, particularly in the automotive and electronics industries [4]. In the
electronics industry, semisolid injection molded magnesium parts are replacing engineering plastic parts because of the low density, high damping capacity, high thermal and electric conductivity, superior rigidity and strength, preferable metallic texture and recyclability of Mg alloys, the properties indispensably required in audio-video-computer-communication equipment. The automotive industry has an even more obvious motivation for the use of magnesium technology, the light-weighting of the vehicle [5, 6].

Further growth would follow in these and other industries, e.g., aeronautic and aerospace industries, if the properties and cost-performance of magnesium parts are improved by the development of new alloys and new processing routes. An important processing route for magnesium alloys that has not been sufficiently explored is rapid solidification processing (RSP). Extensive application of RSP to commercial magnesium alloys has not been realized primarily because of the metal’s pyrophoricity or susceptibility to violent oxidation, which is difficult to prevent in conventional RSP processes. For example, gas atomization of magnesium alloys necessarily requires costly equipment designed to prevent pyrophoric explosions [7-11]. While such equipment has been studied in R&D, high capital and operation costs have not permitted commercialization.

This research was intended to explore and characterize the droplet-based rapid solidification of Mg-Zn-Y alloys. The ultimate goal was to develop safety-proven uniform droplet spray (UDS)-based enabling technology for processing high-performance magnesium alloys. The knowledge gained could be extended to a wider class of magnesium and other light metal alloys, thus promising superior material properties of
interest to automotive, electronics and other industrial applications. The modeling scheme and control capabilities for the solidification of UDS droplets gained in this work will also lead to applications to manufacturing with higher-melting point alloys, including ferrous alloys. Beyond monolithic alloys, the thermo-structurally controlled UDS process is suitable for deposition of functionally graded materials (FGM) and metal matrix composites (MMC) [12-14] with a variety of reinforcement fibers and particulates. The in-flight and splat deposition heat transfer and solidification models are expected to be adaptable to a variety of other processes, including thermal spray, atomization and welding methods with droplet transfer.
1.2 Objective of Research

This research addresses an innovative enabling technology, which exploits molten metal sprays consisting only of mono-size droplets, large enough to prevent pyrophoric explosions but small enough to produce all rapid solidification effects in Mg alloys. This mono-size droplet generating process, termed the uniform-droplet spray (UDS) process [15, 16], operates on the concept of regulated capillary breakup of laminar jet (Section 2.2.2 and 4.2) and has been applied to various alloys, including tin, zinc, aluminum, copper and ferrous alloys [15-42]. Safety has been proven in a recent preliminary study with conventional magnesium alloys [40] (Section 2.2.2.1).

The objective of this research is to study, understand, and control the microstructure evolution in Mg-Zn-Y alloys during droplet-based rapid solidification, with the ultimate goal of developing a novel processing route for the production of high-performance magnesium alloys in the forms of bulk materials. Mg-Zn-Y alloys are chosen because these alloys, if processed to have an RSP structure strengthened by various secondary phases such as the stable quasicrystalline icosahedral phase ($Mg_3Zn_6Y$) [43-68] and/or X-phase ($Mg_{17.5}ZnY$) [58, 69-74], may possess drastically improved mechanical properties at room and elevated temperatures [58]. The UDS process, controlled via de-coupled process parameters, is ideally suited for performing systematic experiments to study the rapid solidification of these alloys while it is also highly viable for commercial applications through scalable multi-orifice spraying configurations.
2. BACKGROUND

2.1 \textit{Mg-Zn-Y} Alloys

Commercial magnesium alloys are available as casting or wrought alloys. Most of commercial alloys are based on the \textit{Mg-Al} or \textit{Mg-Zn} binary systems, with varying degrees of \textit{Al} and \textit{Zn} contents and additional alloying with \textit{Th}, \textit{Zr} and rare earth (RE) elements [1, 14]. The striking discovery of the quasicrystalline phases in the \textit{Mg-Zn-RE} systems may lead to the development of a totally new class of high-strength alloys [44-47]. For example the \textit{Mg-Zn-Y} system contains an icosahedral phase (\textit{I-phase}) with a stoichiometry of \textit{Mg}_{42}\textit{Zn}_{50}\textit{Y}_{8} [49, 50]. Other rare earths, such as \textit{Ho}, \textit{Er}, \textit{Dy}, \textit{Gd}, and \textit{Tb}, also produce similar quasicrystalline phases with \textit{Mg} and \textit{Zn}. The \textit{Mg-Zn-Y} system is of particular importance as its \textit{I-phase} is thermodynamically stable even at elevated temperatures [44, 47] and can thus be exploited to develop high-performance structural alloys. Other attractive properties of the \textit{I-phases} are their low electrical and thermal conductivity, good corrosion and oxidation resistance, and high hydrogen storage capacity [66-68, 75].

\textit{Mg-Zn-Y} alloys with compositions between \textit{Mg} and \textit{I-phase} are reported to exhibit a solidification microstructure consisting of a primary phase (either \textit{\alpha-Mg} solid solution or \textit{I-phase}) and a lamellar eutectic, which suggests existence of a pseudo binary eutectic system between the two compositions [44]. Among these alloys, hypoeutectic alloys are of industrial importance as they can be processed into \textit{\alpha-Mg} matrix alloys strengthened by stable \textit{I-phase} particles. Such composite microstructures are stable
because of the low energy of the $I$-phase/$\alpha$-Mg interphase boundary [47] and hence would lead to the development of advanced magnesium alloy products suitable for service at both low and elevated temperatures. However, conventionally cast and hot rolled $Mg-Zn-Y$ alloys with low $I$-phase contents, e.g., $Mg_{95}Zn_{4.3}Y_{0.7}$ and $Mg_{97.8}Zn_{2}Y_{0.2}$, have tensile yield strengths (150 to 220 MPa) no better than those typical of conventional alloys due to the low $I$-phase contents [45]. Conventionally cast alloys with higher $I$-phase content are very brittle and cannot be subjected to thermomechanical processing such as hot rolling and extrusion. Therefore, a new processing approach needs to be developed to take full advantage of the $I$-phase in $Mg-Zn-Y$ alloys.

The $X$-phase is another potential important ternary phase which may be important in producing high-strengthening alloy. $Mg-Zn-Y$ alloys of compositions near the $Mg$ rich corner of the $Mg-Zn-Y$ Gibbs triangle, when subjected to RSP and thermomechanical processing, have been shown to have improved yield strength, as elaborated in the following section (2.1.1). It is suggested that this strengthening effect is obtained by the $X$-phase and $Mg_{24}Y_{5}$ phase which are formed in the $Mg-Zn-Y$ alloy [58, 73, 76].

It is well understood that ternary or higher alloy systems are usually characterized by complex thermodynamic conditions that produce different phases in alloys that belong to these systems. This is certainly the case in the alloys of $Mg-Zn-Y$ ternary system. In fact, it was only recently that the phase evolution as well as other characteristics such as crystal structures in the ternary system became relatively well understood. The exact evolution of many of the complex stable and metastable ternary phases is still controversial. As investigations progress, even for binary intermetallic
compounds, e.g., \(\text{Mg}_{27}\text{Zn}_{25}\) which is conventionally called \(\text{MgZn}\), characteristics in crystallography are continuously updated [77].

Early research on this ternary system started in the late 60s and continued through the early 80s [78-81]. Padezhnova et al. [78] led the early efforts, which then stimulated interest in the research of the \(\text{Mg-Zn-Y}\) system [78-80], particularly for further understandings of the ternary phases that occur in the magnesium-zinc side and/or magnesium rich corner of the Gibbs triangle and exploiting these phases in alloy development.

Initially, the three phases having the compositions of \(\text{Mg}_3\text{Zn}_6\text{Y}_1\), \(\text{Mg}_{12}\text{YZn}_6\) and \(\text{Mg}_3\text{Y}_2\text{Zn}_3\), discovered by Drits and Padezhnova et al. in their early research [78-80], were denoted \(Z\) phase, \(X\) phase and \(W\) phase, respectively. Later, Luo et al. [55] reported on a stable icosahedral ternary phase within this ternary system and Niikura and Tsai et al. [75, 82] reported that this phase was actually identical to the \(Z\) phase which was initially reported as an unknown phase for which no crystallographic data existed [78, 79]. Later the name \(Z\) was assigned to a hexagonal phase, which forms at the composition of \(\text{Mg}_{28}\text{Zn}_{65}\text{Y}_7\) [83, 84], while \(\text{Mg}_3\text{Zn}_6\text{Y}_1\) is now called \(I\)-phase.

### 2.1.1 \(\text{Mg-Zn-Y}\) alloy ternary phases

Up to now, various ternary compounds are found in the \(\text{Mg-Zn-Y}\) system as listed in Table 2-2. Generally, the ternary phases that form in the \(\text{Mg-Zn-Y}\) system are known to be harder than \(\alpha\)-\(\text{Mg}\) [79, 85]. Among them, the \(I\)-phase and the \(X\)-phase are of particular importance as they may contribute to the strengthening of \(\text{Mg-Zn-Y}\)-based alloys [43-74].
The \( I \)-phase is a compound represented by the composition \( Mg_3Zn_6Y_1 \), which was formerly referred to as \( Z \) phase [78]. This phase is now widely called \( I \)-phase after the identification of this phase as an icosahedral quasicrystalline phase, but some confusion still remains as to the denotations of this phase due to this historical reason. Presently, this phase is referred to as \( Mg_3Zn_6Y_1 \), \( I \)-phase or \( \tau_2 \) phase (or even \( Z \) phase using the old convention) depending on researchers [86]. In the present study, the phase is called the \( I \)-phase. Figure 2-1 shows a pentagonal dodecahedral growth morphology of the \( I \)-phase which was most commonly observed in both Mackay and Bergman-type phases [77, 97] explained below.

This icosahedral quasicrystalline phase has a narrow range of composition near that of \( Mg_3Zn_6Y_1 \) [55]. The \( I \)-phase is considered to be a face centered icosahedral (fci) \( Mg-Zn-Y \) phase which is categorized as a Bergman type icosahedral phase. Initially this phase was discovered with no crystallographic data, but it was soon identified to be a stable quasicrystalline phase [55, 75, 82] after the discovery of quasicrystalline phases as new states of matter by Shechtman et al. [87, 88].

A quasicrystal is regarded as an intermediate structure between crystalline and amorphous states and has a non-periodical (quasi-periodical) ordered atomic arrangements and rotational symmetries forbidden to classical crystallography which allows only 2, 3, 4, and 6-fold rotational symmetries, e.g., 5-fold rotational symmetry [89]. As listed in Table 2-1, various quasicrystalline phases have been confirmed to exist with two or three dimensional atomic arrangements both as stable and metastable phases, and a Fibonacci sequence in Mathematics is regarded as a one dimensional quasicrystal due to its quasi-periodicity (or aperiodicity).
Figure 2-1: SEM image of $I$-phase quasicrystal found in $Zn_{50}Mg_{45}Y_5$ alloy by Niikura et al. [82]
<table>
<thead>
<tr>
<th>Space</th>
<th>Type</th>
<th>Alloy *</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-D QCs</td>
<td>Icosahedral phase (20 sides polyhedron)</td>
<td>[Al-TM Class] Al-TM (TM=V, Cr, Mn, Ru, Re...), Al-(Mn, Cr, Fe)-(Si, Ge), Al-(Cu, Pd)-TM (TM=Cr, Mn, Fe, Mo, Ru, Re, Os), Al$<em>{60}$Cu$</em>{25}$TM$<em>{15}$, Al$</em>{65}$Cu$<em>{20}$TM$</em>{15}$, Al$<em>{60}$Pd$</em>{20}$TM$<em>{10}$ (TM= Mn, Re), Al$</em>{72}$Pd$<em>{13}$X$</em>{10}$ X=(V$<em>{5}$Co$</em>{5}$, Cr$<em>{5}$Fe$</em>{5}$, Mo$<em>{5}$Ru$</em>{5}$, W$<em>{5}$Os$</em>{5}$), Ga-Pd-Mn, Al$<em>{17}$Z$</em>{20}$Si$<em>{8}$, Cu$</em>{10}$Ge$_{2}$, etc.</td>
</tr>
<tr>
<td></td>
<td>MI type (Mackay icosahedron)</td>
<td>Al$<em>{63}$Cu$</em>{25}$TM$<em>{12}$, Al$</em>{65}$Cu$<em>{20}$TM$</em>{15}$, Al$<em>{60}$Pd$</em>{20}$TM$<em>{10}$ (TM= Mn, Re), Al$</em>{72}$Pd$<em>{13}$X$</em>{10}$ X=(V$<em>{5}$Co$</em>{5}$, Cr$<em>{5}$Fe$</em>{5}$, Mo$<em>{5}$Ru$</em>{5}$, W$<em>{5}$Os$</em>{5}$), Ga-Pd-Mn, Al$<em>{17}$Z$</em>{20}$Si$<em>{8}$, Cu$</em>{10}$Ge$_{2}$, etc.</td>
</tr>
<tr>
<td></td>
<td>[Mg-Al-Zn Class] Bergman type (RT type, FK type)</td>
<td>Mg-Al-(Zn,Cu,An,Pd), Mg$<em>{45}$Al$</em>{14}$Pd$<em>{14}$, Mg$</em>{47}$Al$<em>{38}$Pd$</em>{15}$, Al-Mg-(Cu, Ag), Ga-Mg-Zn, Ga$<em>{10}$Mg$</em>{10}$Zn$<em>{21}$, Ga$</em>{20}$Mg$<em>{37}$Zn$</em>{43}$, Al-Li-(Cu, Au, Zn, Mg), Al$<em>{3}$Li$</em>{1}$Cu$<em>{14}$, Zn-Mg-RE (RE=Y, Nd, Sm, Dy, Gd, Er, Ho, Tb), Ti$</em>{10}$Zr$<em>{8}$Ni$</em>{17}$, Zn$<em>{60}$Mg$</em>{30}$RE$<em>{10}$, Zn$</em>{60}$Mg$<em>{30}$RE$</em>{8}$, (RE=Y, Dy, Gd, Ho, Tb, Er), etc.</td>
</tr>
<tr>
<td></td>
<td>Tsai type</td>
<td>Cu$<em>{28}$Ga$</em>{14}$Mg$<em>{5}$Sc$</em>{15}$, Zn$<em>{6}$Mg$</em>{5}$Sc$<em>{15}$, Zn-(Fe,Co,Ni,Pd,Cu,Ag,Au)-Sc$</em>{16}$, Zn$<em>{2}$Fe$</em>{5}$Sc$<em>{16}$ (Ho,Er,Tm)$</em>{16}$, Zn$<em>{70}$Mg$</em>{10}$Yb$<em>{14}$, Ag$</em>{22}$In$<em>{24}$ (Ca,Yb)$</em>{16}$, Cd$<em>{60}$Mg$</em>{50}$ (Ca,RE)$<em>{15}$ (RE=Y, Gd, Tb, Dy, Ho, Er, Tm, Yb, Lu), Cd$</em>{60}$Cu$<em>{15}$, Cd$</em>{60}$Yb$_{16}$, etc.</td>
</tr>
<tr>
<td>2-D QCs</td>
<td>Decagonal phase (10 sides)</td>
<td>[Al-TM class] Al-TM (TM=Mn, Co, Fe, Pd), Al-(Cu,Ni,Pd)-TM (TM=Fe, Ru,Re,Co,Rh,Ir), Al$<em>{3}$Ni$</em>{2}$Co$<em>{10}$x (x=10–20), Al$</em>{60}$Cu$<em>{15}$Co$</em>{20}$, Al$<em>{65}$Cu$</em>{20}$Co$<em>{15}$, Al$</em>{72}$Pd$<em>{13}$TM$</em>{10}$ (TM=Fe,Ru,Os), Al$<em>{72}$Pd$</em>{13}$Mn$_{17}$, etc.</td>
</tr>
<tr>
<td></td>
<td>[Mg-Al-Zn class]</td>
<td>Fe$<em>{60}$Nd$</em>{40}$, Zn-Mg-RE (RE=Y, Dy, Ho, Lu, Tb, Gd), Zn$<em>{60}$Mg$</em>{40}$RE$_{2}$ (RE=Y, Dy, Ho, Lu, Tb, Gd), etc.</td>
</tr>
<tr>
<td></td>
<td>Octagonal phase (8 sides)</td>
<td>Cr$<em>{71}$Ni$</em>{29}$, V$<em>{15}$Ni$</em>{10}$Si, etc.</td>
</tr>
<tr>
<td></td>
<td>Dodecagonal phase (12 sides)</td>
<td>Cr$<em>{5}$Ni$</em>{3}$Si$<em>{2}$, V$</em>{15}$Ni$_{10}$Si, etc.</td>
</tr>
</tbody>
</table>

* These underlined alloys represent stable phases.
The so-called icosahedral phases were initially categorized into two groups, (a): the Al-transition metal (Al-TM) class and (b): the Al-Mg-Zn class. Later a new group of the stable binary icosahedral phases, called Tsai-type occasionally, was found in the Cd-Yb and Cd-Ca systems. These three groups can be distinguished by their building units (atomic clusters) that build the quasicrystalline structure as illustrated in Figure 2-2 [93]. The Al-TM class, the Al-Mg-Zn class and the binary stable quasicrystals, respectively, have the building units related to the Mackay icosahedral (MI) cluster which is formed in $\alpha$-Mn(Al,Si) crystalline phase [94-96], the Bergman cluster which is formed in $T$-Mg-(Al,Zn) crystalline phase [97, 98], and the Tsai-type cluster which formed in Cd-Yb system [99], respectively. According to the differences in atomic arrangement, these icosahedral phases are also grouped into the MI type, the Bergman-type and the Tsai-type. The Bergman-type quasicrystals are sometimes referred to under the names of Rhombic Triacontrahedron (RT) type or Frank-Kasper (FK) type [100-102] from their related crystal structures or atomic arrangements.

The quasicrystalline phases of the Mg-Zn-Y system belong to the Al–Mg–Zn class having the Bergman type cluster [93]. Crystalline phases involving the above mentioned clusters are so-called approximants which are compounds whose compositions and atomic arrangements are close to those of quasicrystals, but follow a classical crystallography [99, 103]. Structural investigations on quasicrystalline systems are hindered due to the lack of periodicity in these phases. Thus, a crystalline approximant, which is related to a quasicrystalline phase, has great importance to bring a route for a detailed structure analysis [104]. In the Zn-Mg-Y system, no crystalline phase has been clearly defined as an approximant phase for this $I$-phase [105] until recently, but the crystal structure of $Mg_{51}Zn_{20}$ [106], which is conventionally called $Mg_{7}Zn_{3}$, has been
suggested as the 1/1 approximant phase by Luo et al. [50, 106-108]. Brühne et al. [109-111] described a local structure model for the fci - Mg$_{25}$Y$_{11}$Zn$_{64}$ alloy, Figure 2-3. Although the crystallography of these quasicrystalline phases, including the Mg-Zn-Y I-phase, is still not clearly established due to the reason mentioned above, it has been reported that quasicrystal-reinforced Mg-Zn-Y alloys bring enhanced mechanical properties such as high strength and ductility, and good formability [47, 108].
Figure 2-2: Schematic of (a) the Mackay type approximant in Mn-Al-Si system, (b) the Bergman type approximant in Mg-Al-Zn system and (c) the Tsai-type cluster in the i-Yb–Cd quasicrystal atomic structures [93].
Figure 2-3: A structure model for the face centered icosahedral (fci) $Mg_{22}Y_{11}Zn_{64}$ alloy. White, grey and dark grey balls represent $Mg$, $Zn$ and $Y$ atoms, respectively. [111].
Another important ternary phase $Mg_{12}YZn$, denoted as $X$-phase [78-80], attracts attention as a strengthening phase for $Mg-Zn-Y$ alloys. Kawamura et al. [58, 73, 76] reported that a $Mg_{97}Zn_{1}Y_{2}$ alloy, produced by hot extrusion of gas-atomized powder, showed yield strengths over 600 MPa. It is considered that the superior mechanical properties of this alloy was achieved by both refined microstructure of $\alpha$-$Mg$ matrix and $X$-phase with a 18R-type\textsuperscript{[1]} long-period stacking ordered (LPSO or LPO) structure [73, 74]. The $X$-phase, which is known to occur at the $Mg$ rich region in the $Mg-Zn-Y$ system, was also initially reported as an unknown phase [78, 79]. Later, Luo et al. [69] reported that the phase has a 18R-type LPSO structure, Figure 2-4. The LPSO structures are reported to form in various stacking sequences such as 18R, 14H, and 6H. The alloy containing the LPSO structure is expected to have improved mechanical properties due to its high thermal stability [73]. Also, the LPSO structure prohibits the deformation twinning of the 2H-$Mg$ matrix [70]. The LPSO structure has been reported to form both in cast ingots and rapidly solidified alloys [73], although the strengthening effect in the as-cast $Mg_{97}Zn_{1}Y_{2}$ alloys is sacrificed due to coarse grains in their microstructure. Thus, in order to realize strengthening with the LPSO structure, a rapid solidification process is required as it results in a refined microstructure [74].

\textsuperscript{1} This expression is base on the Ramsdell notation. [112-115]
Figure 2-4: High-resolution image of the X-phase projected along the [100] orientation [69].
2.1.2 Phase diagrams of Mg, Zn and Y and their compounds

As explained in previous section 2.1.1, early investigations of the magnesium rich corner of the Mg-Zn-Y ternary system date back to the late 60’s [81]. Zaselyan et al. [81] investigated the Mg-Zn-Y ternary system and presented partial projections of the liquidus and the solidus for the concentration range of 80 to 100 wt% Mg, Figure 2-5, followed by Drits and Padezhnova et al. who reported the existence of three ternary phases [78-80]. Those previous studies were summarized by Rayner [116], and more recently, by Tsai et al.[83], the Material Science International Team (MSIT) [86] and others [117]. Figure 2-6 shows the ternary Mg-Zn-Y phase diagram at 573 K reported by Padezhnova et al. [118] where the intermetallic phases $\text{Mg}_{12}\text{YZn}$, $\text{Mg}_{3}\text{YZn}_{4}$ are identified. Figures 2-7 – 2-9 show the equilibrium binary Mg-Zn, Mg-Y and Zn-Y systems [119-121]. Table 2-2 lists the possible compounds that are formed in the system.
Figure 2-5: Mg-rich corner of the ternary Mg-Zn-Y phase diagram. (a) liquidus projection and (b) solidus projection [81, 122].
Figure 2-6: Ternary Mg-Zn-Y phase diagram at 573 K reported by Padezhnova et al. [118].

Figure 2-7: Equilibrium phase diagram of the Mg-Zn system [120].
Figure 2-8: Equilibrium phase diagram of the \textit{Mg-Y} system [119].

Figure 2-9: Equilibrium phase diagram of the \textit{Y-Zn} system [121].
Table 2-2: Partial Compounds List in Zn-Mg-Y System. [83, 86, 117, 119-121, 123-131]

<table>
<thead>
<tr>
<th>Phase</th>
<th>Pearson symbol</th>
<th>Space group</th>
<th>Lattice type / Structure</th>
<th>Lattice parameters (nm)</th>
<th>References and Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Mg)</td>
<td>hP2</td>
<td>P6_3/mmc</td>
<td>A3 / hcp / Mg</td>
<td>a=0.321, c=0.521, (\gamma=120)</td>
<td>[86, 120, 123-127]</td>
</tr>
<tr>
<td>(βY)</td>
<td>c12</td>
<td>I (\overline{3}m)</td>
<td>A2/ bcc / W</td>
<td>a=0.407</td>
<td>[86, 127]</td>
</tr>
<tr>
<td>(αY)</td>
<td>hP2</td>
<td>P6_3/mmc</td>
<td>A3 / hcp / Mg</td>
<td>a=0.3651, c=0.5739, (\gamma=120)</td>
<td>[86, 125]</td>
</tr>
<tr>
<td>(Zn)</td>
<td>hP2</td>
<td>P6_3/mmc</td>
<td>A3 / hcp / Mg</td>
<td>a=0.2666, c=0.4949, (\gamma=120)</td>
<td>[86, 125, 126]</td>
</tr>
<tr>
<td>Mg-Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mg(_{2})Y(_3) (c)</td>
<td>c158</td>
<td>(\overline{4}3m)</td>
<td>A12 / cub / (\alpha)Mn, (\beta)Y, Mg(_{2})Mg(_3)Y(<em>6), Mg(</em>{2})Y(_6)</td>
<td>a=1.1251</td>
<td>[86, 125, 127, 128]</td>
</tr>
<tr>
<td>Mg(_2)Y</td>
<td>hP12</td>
<td>P6_3/mmc</td>
<td>C14/ hex / MgZn(_2)</td>
<td>a=0.6037, c=0.9752, (\gamma=120)</td>
<td>[86, 125, 127, 128]</td>
</tr>
<tr>
<td>MgY</td>
<td>cP2</td>
<td>P(\overline{3}m)</td>
<td>B2 / cub / C(_{6})Cl</td>
<td>a=0.3810</td>
<td>[125, 127, 128]</td>
</tr>
<tr>
<td>Mg-Zn</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mg(<em>{2})Zn(</em>{10}) (Mg(<em>{2})Zn(</em>{2}))</td>
<td>a1158</td>
<td>I(\overline{3}m)</td>
<td>- / orth / Mg(<em>{2})Zn(</em>{10})</td>
<td>a=1.4025, b=1.4083, c=1.4486</td>
<td>[83, 86, 125, 1194, D7(<em>5)/Ta(</em>{2})B(_{2}) in [126]</td>
</tr>
<tr>
<td>Mg(<em>2)Zn(</em>{10}) (Mg(<em>{2})Zn(</em>{2}))</td>
<td>hB76</td>
<td>R_3_c</td>
<td>- / - / Zr(<em>{2})Re(</em>{2})</td>
<td>a=2.5758, c=0.87624, (\gamma=120)</td>
<td>[120]</td>
</tr>
<tr>
<td>Mg(<em>{2})Zn(</em>{10}) (Mg(<em>{2})Zn(</em>{2}))</td>
<td>mC110</td>
<td>R(_{3})</td>
<td>- / trigl / Mg(<em>{2})Zn(</em>{2})</td>
<td>a=2.596, b=0.524, c=1.428, (\beta=102.5)</td>
<td>[125]</td>
</tr>
<tr>
<td>Mg(<em>2)Zn(</em>{10})</td>
<td>hP12</td>
<td>P6_3/mmc</td>
<td>C14/ hex / MgZn(_{2})</td>
<td>a=0.5253, c=0.8566, (\gamma=120)</td>
<td>[83, 86, 117, 125, 126, 129]</td>
</tr>
<tr>
<td>Mg(<em>2)Zn(</em>{11})</td>
<td>cP39</td>
<td>Pm(_3)</td>
<td>D(<em>8) / cub / Mg(</em>{2})Zn(_{11})</td>
<td>a=0.8552</td>
<td>[83, 117, 125, 126, 129]</td>
</tr>
<tr>
<td>Zn-Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YZn</td>
<td>cP2</td>
<td>P(\overline{3}m)</td>
<td>B2 / cub / C(_{6})Cl</td>
<td>a=0.3577</td>
<td>[117, 125, 131]</td>
</tr>
<tr>
<td>aYZn(_{2})</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>[125]</td>
</tr>
<tr>
<td>oYZn(_{2})</td>
<td>o112</td>
<td>Im(\overline{3}m)</td>
<td>- / orth / C(_{6})Cu(_2)</td>
<td>a=0.4504, b=0.7143, c=0.7664</td>
<td>[117, 121, 131]</td>
</tr>
<tr>
<td>YZn(_{3})</td>
<td>oP(_{16})</td>
<td>P(_{4})</td>
<td>- / orth / Zn(_{3})Y</td>
<td>a=0.5990, b=0.4405, c=1.0111</td>
<td>[125]</td>
</tr>
<tr>
<td>Y(<em>{2})Zn(</em>{11})</td>
<td>o128</td>
<td>I(\overline{3}m)</td>
<td>- / orth / La(<em>{2})Al(</em>{11})</td>
<td>a=0.4388, b=1.2929, c=0.8871</td>
<td>[117, 125, 131]</td>
</tr>
<tr>
<td>Y(<em>{3})Zn(</em>{10})</td>
<td>hP146</td>
<td>P6_3/mmc</td>
<td>- / hex / Zn(<em>{3})(</em>{2})Zn(_{10})</td>
<td>a=1.428, c=1.40, (\gamma=120)</td>
<td>[83, 117, 125]</td>
</tr>
<tr>
<td>Y(<em>{2})Zn(</em>{3})</td>
<td>hP36</td>
<td>P6_3/mmc</td>
<td>- / hex / Zn(_{2})Er</td>
<td>a=0.8885, c=0.9207, (\gamma=120)</td>
<td>[83, 117, 125, 131]</td>
</tr>
<tr>
<td>Y(<em>{2})Zn(</em>{3})</td>
<td>hP38</td>
<td>P6_3/mmc</td>
<td>- / hex / Th(<em>{2})Ni(</em>{17})</td>
<td>a=0.899, c=0.8764, (\gamma=120)</td>
<td>[125, 131]</td>
</tr>
<tr>
<td>Y(<em>{2})Zn(</em>{3})</td>
<td>hR57</td>
<td>R(_{3})</td>
<td>- / hex / Zn(<em>{2})Th(</em>{2})</td>
<td>a=0.89719, c=1.31414, (\gamma=120)</td>
<td>[83, 86, 125, 131]</td>
</tr>
<tr>
<td>Y(<em>{2})Zn(</em>{3})</td>
<td>hI26</td>
<td>I(\overline{4}3)</td>
<td>D(<em>{2d}) / tet / Th(</em>{2})Ni(_{17})</td>
<td>a=0.8861, c=0.5205, (\gamma=120)</td>
<td>[83, 125, 132]</td>
</tr>
<tr>
<td>Zn-Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W - Mg(<em>{2})Zn(</em>{1})</td>
<td>cF16</td>
<td>Fm(_{3})</td>
<td>L(<em>{2}) / cub / MnCu(</em>{2})Al</td>
<td>a=0.6848</td>
<td>[83, 86, 117, 125, 130]</td>
</tr>
<tr>
<td>Z - Mg(<em>{2})Zn(</em>{10})Y(_{2})</td>
<td>h**</td>
<td>P6_3/mmc</td>
<td>- / hex / -</td>
<td>a=1.4579, c=0.8687, (\gamma=120)</td>
<td>[83, 86, 117]</td>
</tr>
<tr>
<td>I - Mg(<em>{2})Zn(</em>{2})</td>
<td>-</td>
<td>Fm53</td>
<td>- / icosahedral / Quasicrystal</td>
<td>a=0.519</td>
<td>[83, 86, 117]</td>
</tr>
<tr>
<td>D - Mg(<em>{2})Zn(</em>{10})Y(_{2})</td>
<td>-</td>
<td>P100</td>
<td>- / decagonal / Quasicrystal</td>
<td>a=0.227, c=0.255</td>
<td>[83, 86]</td>
</tr>
<tr>
<td>H - Mg(<em>{2})Zn(</em>{10})Y(_{5})</td>
<td>h**</td>
<td>P6_3/mmc</td>
<td>- / hex / -</td>
<td>a=0.776, c=0.92, (\gamma=120)</td>
<td>[83, 86, 117]</td>
</tr>
<tr>
<td>X - Mg(<em>{2})Zn(</em>{10})</td>
<td>-</td>
<td>-</td>
<td>- / hex / -</td>
<td>a=0.3224, c=4.6985, (\gamma=120)</td>
<td>[86, 117]</td>
</tr>
</tbody>
</table>

\(\tau_5\) in [86]
2.2  

Mg Alloys Processing

2.2.1  Conventional processing

Both casting and working are used to process magnesium alloys. Of the conventional casting processes, die-casting and squeeze casting have been the most widely used mass production processes [4]. Squeeze casting, adapted in the automotive industries in the US, Japan and Europe over the past 20 years, differs from conventional die-casting in that the melt is supplied into the mold cavity through a much wider sprue than in die-casting. While squeeze casting can produce full-density products at high production rates, macro-segregation is inherently severe in this process because the eutectic liquid between primary dendrites is squeezed into thinner mold sections where it solidifies into an undesirable eutectic structure. This pressure-enhanced alloy segregation, unique to squeeze casting, occurs over much larger distance than in conventional casting, giving rise to product deficiency and design constraints. Other drawbacks of die and squeeze casting include inherently high capital cost and environmental concerns caused by the use of high-temperature alloy melt and die-casting parting agents.

Semi-solid injection molding permits segregation-free, net-shape mass production of thin-walled, complex components at competitive costs in some product areas [4]. Exploiting the thixotropic behavior of semi-solids manifested at relatively low temperatures, semi-solid injection molding eliminates much of the environmental hazards
associated with conventional metal die/squeeze-casting, while adopting the well-established basic scheme of plastic injection molding. The current semi-solid magnesium injection molding process, however, uses costly chip pellets [4] cut from cast ingots which cause high product rejection rates because of their irregular shapes and chemical segregation inherited from the mother ingot. Alternative low-cost pellets with no macro-segregation are highly desired in this process.

Rapid solidification processing (RSP) is considered the most effective processing route to developing high performance magnesium alloys. In fact, as briefly described in section 2.1.1.1, Mg-Zn-Y alloys, when produced by the extrusion of gas-atomized powder, show yield strengths up to 600 MPa [58]. However, commercialization by conventional RSP has not been possible because of cost impediments and the danger of pyrophoric explosion inherent to most existing RSP methods, such as gas atomization and melts spinning. This research addresses an explosion-free, mono-size droplet-based rapid solidification process that can produce desirable microstructures in hypo-eutectic alloys. The product area addressed in this study is the production of high-performance bulk Mg-Zn-Y alloys produced by droplet deposition. Another area that this work may lead to is the production of advanced mono-size particulate materials with controlled RSP microstructure suitable for innovative applications, e.g., pellets for semi-solid injection molding.
2.2.2 Uniform-droplet spray process

The uniform-droplet spray (UDS) process, used in this research, is a new RSP process that generates only mono-size droplets of desired diameter. This ability to control the droplet diameter assures safe processing of pyrophoric alloys without sacrificing RSP effects. The initial form of the UDS process, developed in the Laboratory for Manufacturing and Productivity at the Massachusetts Institute of Technology [15, 16], was applicable only to low-melting-point alloys.

As schematically illustrated in Figure 2-10, an alloy melted in the crucible is gas pressure-ejected from an orifice as a laminar jet. Regulated vibrations are imposed on the exiting jet to cause capillary breakup of the jet into mono-size molten droplets. The diameter of the mono-size droplets can be adjusted at any value chosen between about 50 and 2000 µm through adjusting the orifice diameter, the initial jet velocity and the vibration frequency, all of which can be varied independently from each other. The mono-size droplets can be electrically charged if required as they break from the jet to prevent in-flight droplet merging. The droplets cool and solidify in an essentially identical manner while they travel down the inert-atmosphere chamber. The thermal state of the droplets in a uniform-droplet spray (UDS) therefore depends only on the flight distance, and can thus be predicted using a simulation model, like the one developed by DiVenuti et al. [133], which is outlined in Section 2.3.
The essentially identical solidification paths of the droplets in a UDS are verified in Figure 2-11, which shows calorimetrically determined droplet enthalpy values together with model predictions as a function of flight distance for 185 μm Sn-5wt% Pb droplets [27, 32]. While both the experimental and theoretical values of droplet enthalpy decrease monotonically up to about 0.46 m from the orifice, in line with the second law of thermodynamics, the experimental values fall short of the theoretical predictions by about 0.0003 J/droplet. This discrepancy was caused by the formation of metastable Sn solid solution supersaturated with Pb in the droplets quenched in the calorimeter fluid before nucleation set in [27]. The experiment and simulation agree above 0.46 m where the droplets were quenched after nucleation and recalescence, which left little supersaturation in the Sn solid solution of the solidified droplets. The sharp increase in experimental values at 0.46 m provides evidence that nucleation occurred over a very narrow distance of 0.46 m in the majority of the droplets, producing nearly identical solidification paths for all the droplets.
Figure 2-10: Schematic of the UDS droplet generator.

Figure 2-11: Measured and predicted enthalpy of 185 μm Sn-5wt%Pb droplets [27].
The UDS process applies to two important areas of materials processing; 
*particulate production and droplet deposition*. With the nearly identical solidification 
paths assured for all droplets in a spray, the UDS process permits ultimate levels of control 
over the product microstructure in both particulate production and droplet deposition. In 
particulate production, the droplets (50 - 2000 µm in diameter) may be solidified 
individually during flight or quenched in a fluid or on a substrate at a flight distance that 
corresponds to a desired droplet thermal state. Particulate products so produced differ 
from their conventional counterparts in that the particles in a batch all have the same rapid 
solidification microstructure. The alloys that have been studied at Northeastern 
University, MIT, Oak Ridge National Laboratory, UC Irvine, and industrial laboratories 
include Sn alloys [15, 19, 22-34, 37, 38], Al alloys [21, 30, 36], Cu and Cu alloys [17, 20, 
35, 39, 42], Mg-Al alloys [40] and Fe-B, Fe-B-C and Fe-B-Cr alloys [18, 41].

Figure 2-12 shows 285 µm mono-size Cu droplets arranged in the 
(111)$_{fccc}$ array for demonstration of size uniformity and sphericity [39]. In-flight solidified 
mono-size particles produced by the UDS process have found applications in electronics 
packaging [24]. Other potential applications include micro-bearings, advanced P/M 
processing, and pellets for semi-solid injection molding.

In droplet deposition with UDS, mono-size droplets in a desired thermal state are 
deposited onto a substrate to form a dense deposit with novel rapid solidification 
microstructures that cannot be produced by conventional spray deposition. Unlike 
conventional droplet or spray deposition processes, the UDS process, because all droplets 
have an identical fraction liquid, can produce deposits with virtually full density [19, 23, 26, 
34]. Figure 2-13 shows a cross section of a single-crystalline Sn-5 wt% Pb spray deposit 
produced by incremental epitaxial growth on a single-crystalline Sn substrate [34, 38].
Figure 2-14 shows a Sn - 5 wt% Pb spray deposit produced with 185 µm droplets. Figures 2-15 and 2-16 show 1 mm Mg alloy (AZ91D) droplets and 50 µm lead-free Sn-base solder alloy (Sn-Ag-Cu) droplets, respectively.
Figure 2-12: 285 μm mono-size Cu droplets showing the size uniformity and sphericity of droplets produced by the UDS process [39].

Figure 2-13: Single-crystalline Sn-5 wt% Pb deposit produced on single-crystal Sn substrate [34].
Figure 2-14: $Sn$–5wt$\%Pb$ spray deposit produced with 185µm droplets.
Figure 2-15: 1000 µm Mg alloy (AZ91D) droplets.

Figure 2-16: 50 µm lead-free Sn-base solder alloy (Sn-Ag-Cu) droplets.
2.2.3 Preliminary UDS works with conventional magnesium alloys

A preliminary study with conventional casting alloys $AM60B$ and $AZ91D$ has shown that stable breakup of $Mg$ alloy jets indeed requires the oxygen content be kept below about 1 ppm [40] because the oxidation of the jet caused by residual oxygen in the chamber atmosphere increases the resistance to capillary jet breakup [134]. Figure 2-17 shows such stable breakup of a molten $AM60B$ jet. Figure 2-18 shows 200 µm droplets of an $AZ91D$ alloy created along different solidification paths. Strong effects of solidification path on the dendrite arm spacing are evident. UDS processing also produces other rapid solidification effects. For example, the 850 µm $AM60B$ droplet shown in Figure 2-19, despite its relatively large size, exhibits a very fine microstructure that consists predominantly of supersaturated $\alpha$-$Mg$ dendrites and only a small amount of interdendritic intermetallics. Such extended solid solubility, commonly observed in rapid solidification, can be fully exploited in UDS processing. The safety of UDS magnesium processing was also verified in the preliminary study [40].
Figure 2-17: Uniform breakup of a molten *AM60B* jet [40].
Figure 2-18: 200 µm AZ91D droplets oil-quenched in molten (left) and mushy states (right) [40].

Figure 2-19: Fine α-Mg dendrites in 850 µm AM60B droplet [40].
2.3  Modeling of the Droplet Production Process

Predictive microstructure modeling of a droplet deposition process requires full knowledge of the thermal (and solidification) state of the droplets during their flight to the substrate as well as a thermo-microstructural deposit solidification model that can describe the interdependence of material microstructure with the thermal conditions generated during droplet deposition stage. To achieve this in the present study of the UDS processing of \textit{Mg-Zn-Y} alloys, the basic modeling schemes developed in the prior works [29, 133, 135], outlined below, were adopted in the microstructural modeling of both in-flight droplet solidification and droplet as deposition described in Chapter 4.

2.3.1  Prediction of droplet thermal state

To model the splat solidification in droplet deposition, one must first know the state of the droplets just before they hit the substrate or the prior splat surface. For the UDS process, this is readily achieved by modifying the droplet in-flight solidification simulation model previously developed by DiVenuti \textit{et al.} for the solidification of \textit{Sn-Pb} alloy droplet [133]. The in-flight solidification simulation model incorporates the free dendritic growth model [136] initially developed after Boettinger \textit{et al.} [137], which accounts for curved liquidus and solidus and high Peclet number conditions that are relevant to the crystal growth in a highly supercooled alloy melt. The model predicts the dendrite growth velocity, the interfacial compositions, the non-equilibrium partition
coefficient and the tip radius of the growing dendrites for a given value of supercooling. Figure 2-20 shows the dendritic growth rate calculated for a Mg – 6 wt.% Al binary alloy as a function of melt temperature.

The droplet in-flight solidification model is based on a thermal balance equation coupled with a droplet motion equation [15, 137], which is numerically solved for droplet velocity, droplet temperature, droplet enthalpy, fraction solidified, dendrite tip radius and the composition of the solid as a function of the flight distance. At every computation step, these solidification parameters are calculated from the free dendritic growth model, which then permits the computation of the solidification rate at the step via \( \frac{df_s}{dt} = \frac{vA}{V_d} \) where \( t \) is time, \( f_s \) is the fraction solid, \( v \) is the growth rate, \( A \) is the area of the solid-liquid interface and \( V_d \) is the droplet volume. The rate of solidification relates to the rate of latent heat generation, which, together with the rate of heat removal by convection and radiation, gives the droplet temperature for the next computation step. Figure 2-21 shows an example of such simulation performed for droplets of a Mg – 6 wt.% Al binary alloy generated by the UDS process. The droplet in-flight solidification model is described in Chapter 4 in more detail.
Figure 2-20: Calculated dendritic growth rate for a Mg-6 wt.% Al alloy [40].

Figure 2-21: Calculated droplet temperature and fraction solid of 200, 800 and 2000 μm Mg-6Al droplets [40].
3. EXPERIMENTAL SET-UP AND PROCEDURE

3.1 Materials

The Mg-Zn-Y alloys, supplied from FUKUDA Metal Foil & Powder Co., Ltd., investigated in this research had nominal compositions $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$, $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_{2}$ and $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ in at. %. These alloys are indicated by the solid circles in Figure 3-1. In addition to these alloys, AZ91D was used to roughly optimize UDS parameters for the experiments with the Mg-Zn-Y alloys, as well as to provide a microstructural reference. Alloys with a composition that falls on the line between $\alpha$-Mg and $I$-phase have microstructures with varying ratios of $I$-phase and $\alpha$-Mg. As briefly discussed in Section 2.1, compositions near the $I$-phase ($\text{Mg}_{30}\text{Zn}_{60}\text{Y}_{10}$) give microstructures consisting primarily of the non-ductile $I$-phase, and thus are not of interest. The alloy located in a red region in Figure 3-1 forms the $X$-phase as a typical secondary phase [58, 69-74], which is expected to bring an excellent mechanical property. The composition of $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$, which is overlapped with the blue region in Figure 3-1, has been reported to produce both $I$ and $X$-phases in the as-cast alloys [138, 139]. Alloys with compositions falling in the blue region containing the straight line in Figure 3-1, if refined sufficiently, would have much higher strength while maintaining sufficient toughness and ductility, at both room and elevated temperatures.
Figure 3-1: Mg-rich corner of the Mg-Zn-Y Gibbs triangle. The red region shown here encompasses alloys that form the $X$-phase as a typical secondary phase, while alloys in the blue region form the $I$-phase as a secondary phase.
3.2 UDS Apparatus

The UDS apparatus currently used in AMPL to produce uniform droplet sprays was developed and modified over a decade by former graduate students, Abessi [21], Tuffile [22], Fortner [20] and others [42] during their graduate studies at Northeastern University. The UDS apparatus is equipped with a heating system, a crucible, a vibration transmission system, a gas circulation/pressurization system, a droplet charging system, and a stroboscopic visualization/recording system, etc [20, 21]. More recently, an induction heating system replaced the previous clamshell type resistance heater to extend the processing capacity of the UDS apparatus, particularly for high-melting point alloys. The modifications included redesigning the crucibles by graduate students, Ranganathan [17], Pillai [18] and the author.

Figure 3-2 and 3-3 show the actual UDS apparatus and a side view of the UDS chamber. The following sections describe the individual components in more detail.
Figure 3-2: UDS apparatus in AMPL lab.
Figure 3-3: A schematic of the side view of UDS apparatus.
3.2.1 Heating system

The induction-powered unit, manufactured by Nissin-Giken, Iruma, Saitama, Japan, includes a matching box, an induction coil system and a power control unit as shown in Figure 3-4. Table 3-1 lists the specifications of the induction unit. A diagram of the overall induction unit setup is shown in Figure 3-5.

Figure 3-4: A photograph showing the induction coil without assembly.
Table 3-1: Specifications of the Induction Unit.

<table>
<thead>
<tr>
<th>Company</th>
<th>Nissin-Giken</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>NET-10I</td>
</tr>
<tr>
<td>Power supply</td>
<td>200V 50/60Hz</td>
</tr>
<tr>
<td>Nominal power</td>
<td>10 kW</td>
</tr>
<tr>
<td>Oscillating frequency</td>
<td>15 to 25 kHz</td>
</tr>
<tr>
<td>Absorbed power</td>
<td>12 kVA</td>
</tr>
<tr>
<td>Cooling system pressure</td>
<td>0.2 MPa</td>
</tr>
<tr>
<td>Cooling water flow rate</td>
<td>30 liters/min</td>
</tr>
</tbody>
</table>

Figure 3-5: Schematic of the layout of induction unit.
The power supply provides alternating current through the coil, generating a magnetic field which induces eddy currents mainly in the crucible placed in the center of the coil, generating precise amounts of heat to the crucible.

Figure 3-6 shows a schematic of the coil and the crucible system. The temperature of the induction coil, made from copper tubing with 8 mm O.D., as well as that of the entire induction system including the matching box and the power unit, are maintained low with cooling water circulated from a chiller (NESLAB: Refrigerated recirculator CFT-300). The coil and the crucible are isolated by a quartz tube to avoid direct contact between them. One layer of graphite (carbon) felt is placed inside the quartz crucible to minimize radiation losses.

High density alumina insulation materials (Fiberfrax® Duraboard® LD), which can be placed inside of the furnace chamber to retain heat (Figure 3-7), depending on required temperature ranges, permit heating up to 1,773~1,873 K.
Figure 3-6: Coil and crucible assembly.
Figure 3-7: Alumina insulation block used inside the upper part of the UDS chamber.
3.2.2 Crucible and insert design

Two different types of crucibles were used in UDS experiments. One was made of graphite and the other was made of stainless steel. Initial experiments employed the graphite crucible, but later experiments shifted to the stainless steel crucible, which provided more room to control experimental parameters, such as use of higher ejection pressure. The stainless steel crucibles are more reusable than graphite crucibles and require less time to prepare for an experiment as well.

The graphite crucible was designed as a two-piece screw-type crucible to reduce the heat conduction through the crucible to the other parts. The bottom part of the crucible measured 63.5 mm in outer diameter (OD) x 44.5 mm in inner diameter (ID) x 86.5 mm in length (L) and the top part of the crucible measured 63.5 mm - OD x 54.5 mm - ID x 58.5 mm - L, Figure 3-8. The bottom part of the crucible has a ‘seat’ machined in its inside at the bottom to place a graphite insert with a hole machined for seating an orifice. The insert, Figure 3-9, is cemented to the bottom part of the crucible with a high temperature cement (Aremco Products, Inc: Ceramabond 503) to avoid leakage of molten metal around the connection part of the insert.
Figure 3-8: CAD drawing of the top and bottom part of the graphite crucible.

Figure 3-9: CAD drawing of the graphite insert.
The stainless steel crucible, made of 304 stainless steel, measured 40.6 mm - ID x 59.2 mm - OD x 152.4 mm - L and consists of a cylindrical crucible wall and a bottom plate with an orifice hole directly machined into the plate. The bottom plate is fixed to the main crucible with eight stainless steel bolts, and a graphite gasket is inserted in between them to avoid leakage of molten metal. The bottom plate holds an insert which has an orifice hole directly machined in its center, Figure 3-10. The molten metal is gas pressure-ejected through this orifice into the gas chamber. A photograph of the bottom plate is shown in Figure 3-11.
Figure 3-10: CAD drawing of the stainless steel bottom plate with the insert.
Figure 3-11: The stainless steel bottom plate and the insert with orifice hole.
In UDS experiments with Mg alloys, the chance of orifice clogging increases due to the oxidation of the molten metal. In order to avoid such oxidation, the UDS apparatus had to be further modified. Although there may be various methods to prevent clogging or freezing in the orifice, the present work employed a wire-pull method developed through this work. Figure 3-12 schematically illustrates the wire-pull method, while Figure 3-13 gives an actual picture of the wire on the swing arm. Both the swing arm and the wire were made of type 304 stainless steel. The wire used to plug the 348 µm orifice had a diameter of 305 µm.
Figure 3-12: Schematic of the wire-pull method.

Figure 3-13: Crucible bottom part with the wire-pull system.
3.2.3 Vibration transmission system

The vibration transmission system produces regulated perturbations on the laminar jet, which breaks the jet into mono-size droplets [15, 16]. The system is assembled with a function generator, a voltage amplifier, a piezoelectric vibration assembly, a transducer rod and a dual-trace oscilloscope. Figure 3-14 shows a schematic of the piezoelectric vibration assembly. The piezoelectric vibration assembly is used to convert the electrical waves, generated by the function generator and amplified by the voltage amplifier, to mechanical vibrations. It is constructed with a stack of 5 piezoelectric transducer disks (*EDO Electro-Ceramic: EC-65*) separated by brass spacers. The stack is screw-fastened to a stainless steel base plate by screws and an aluminum plate placed on top of the disks, while an o-ring is placed between the bottom of the stack and the base plate to avoid unrestrained vibrations of the piezoelectric disks. The assembly is mounted on the top plate of the heating furnace, which is about 20 cm from the molten material and is far enough from the heat of molten material to avoid losing their piezoelectric behavior [15] due to heating in excess of the piezo’s Curie temperature during operation. The piezoelectric vibration assembly is driven by the amplifier connected to the function generator that can selectively provide a range of frequency required in UDS experiments. The generated vibrations are transmitted through the vibration rod to the melt in the orifice region, and to the existing jet. A dual-trace oscilloscope (*LEADER: LBO-514*) is connected to the piezoelectric vibration assembly to monitor the frequency and the amplitude of the amplified current. The 2 MHz digital display function generator (*BK PRECISION: 3011B*) is used to supply the electrical wave
for the vibration assembly as well as the strobe light *(Strobotac: Stroboslave Type 1539-A)*. The piezoelectric assembly can produce 1-μm perturbation amplitude for each disk at 526 VAC peak to peak [21], which cannot be provided directly by the function generator. The 20-watt AC/Mobile PA amplifier *(Radio Shack: MPA-31)* is installed in order to supply a necessary voltage to the piezoelectric assembly. A ceramic thermocouple protection tube from OMEGA Engineering Inc. is chosen as the transducer rod and a K type thermocouple is placed into the tube to measure the temperature of the crucible. The bottom end of the transducer rod holds a transducer tip made from a graphite or the stainless steel as illustrated in Figure 3-15. The tip should preferably be conically shaped for the ease for removing the tip from the solidified block of metal when the orifice clogs during UDS spraying. The flat surface of the tip needs to be properly oversized to hover over the area above the orifice in the crucible. The distance between the graphite transducer tip and the orifice is set to be around 2.5 mm, which is a typical distance obtained empirically.
Figure 3-14: Schematic of the piezoelectric vibration assembly.

Figure 3-15: Schematic of the transducer tip.
3.2.4 Droplet deposition system

The substrate motion is computer-controlled in the X-Y directions via a X-Y table (DCI Design Components: EB6-120). The z-direction position is manually set with a scissor jack prior to deposition. Figures 3-16 and 3-17 show an actual appearance of X-Y table inside of the UDS chamber and a schematic of the substrate motion system, respectively. The X-Y table was driven by two stepping motors (Clifton: 23-SHAT- 14DB/ H102), a motion control PC board (Technology 80: 5000), a micro stepping drive (Intelligent Motion Systems: IM483), a dc power supply (Condor: E48-4.0), and a PC (CECI 486/33, CPU: DX486/33). The substrate was moved along various pre-programmed paths required for a given set of experimental parameters. Figure 3-18 represents a side view of the substrate motion system.
Figure 3-16: Photograph of the substrate motion system placed in the UDS chamber.
Figure 3-17: Schematic of the substrate motion system.

Figure 3-18: Schematic of the current setting of deposition experiment.
The substrate is set on top of the X-Y table as illustrated in Figure 3-18. The substrate system consisted of a square 304 stainless steel plate with 3.2 mm thickness and 152 mm length, a square aluminum plate with 20 mm thickness and 180 mm length, a rectangular aluminum plate with 20 mm thickness, 330 mm length and 190 mm width, and 254 mm thick insulation board (*Fiberfrax*® *Duraboard® LD*) as shown in Figure 3-19.

![Figure 3-19: Schematic illustration of the substrate system.](image-url)
3.3 Experimental Procedure

3.3.1 Deposit production scheme

The UDS experiment is generally performed in the following steps.

1. Material Preparation: The desired material, e.g., Mg alloy, is prepared by cutting into pieces small enough to fit in the crucible followed by grinding, polishing and ultrasonic cleaning.

2. Equipment Assembly: The material is charged into the crucible which is then loaded to a top plate and then fastened to the UDS chamber.

3. Adjusting Chamber Conditions: The chamber conditions, e.g., gas type, pressure, oxygen level, etc., are adjusted to the required process conditions.

4. Heating Process: The material charged in the crucible is heated and melted using an induction furnace.

5. Spray and Deposition Process: The molten material is ejected through the orifice by applying pressure to the crucible. The molten jet is broken up into uniform droplets by the vibrations imposed with the piezoelectric transducer at a desired frequency. The droplets are deposited on to the substrate at the required thermal state.

Material Preparation

For the material preparation step, a cast ingot of the magnesium alloy is cut into pieces of rectangular bar that fit the crucible. When cutting the alloy a running coolant
must be used to prevent the alloy from catching fire. The alloy bars were ground for shape adjustment and surface cleaning. Additional grinding and ultrasonic cleaning is performed right before chamber assembly to remove oxides and impurities that may cause problems such as orifice clogging. A charge between 50 and 100 g is generally used depending on experimental parameters such as orifice size.

**Equipment Assembly**

The crucible is mounted to the top plate. Prior to loading the alloy into the crucible, the transducer rod is set to the piezoelectric vibration assembly mounted on the top plate. The rod is fixed in order to maintain a proper clearance of 0.1” between the transducer tip and the orifice inlet. All the electrical wires are checked to confirm that there is no short circuit. This process is carefully examined in order to avoid any malfunction of the heating unit during the operation. Then, all the contact areas of the UDS chamber are cleaned using acetone to remove any dust, oil and grease to maintain good sealing during evacuating and backfilling with an inert gas. The cleaned alloy bars are then carefully placed into the crucible. The crucible bottom plate is fixed together with a graphite gasket. The plate is tightly fastened with eight bolts to prevent metal leakage from the side portion of the crucible. The top plate with the crucible is loaded to the furnace section. Again, the electric wires, which include the thermocouples and those on the vibration system, are verified to check under the actual condition. Then, the top plate is fixed carefully to avoid any gas leakage. Other settings such as the X-Y table position are also set at this stage. The main window of the UDS chamber is closed.
Adjusting Chamber Conditions

The UDS chamber is evacuated by the vacuum pump down to about 300 mTorr (40 Pa) and then back-filled with ultra high-purity (UHP) argon to a pressure of 5 psi \( (3.45 \times 10^4 \text{ Pa}) \). This evacuation and back-filling process is repeated multiple times, until a sufficiently low vacuum level is reached, e.g., 200 mTorr. Then, the chamber is filled with UHP argon to 5 psi and the oxygen content is monitored by the oxygen analyzer. The oxygen content in the UDS chamber is reduced to about 1 ppm with the repetition process of purging and refilling of UHP Ar gas and/or with a gas circulation through a gettering furnace.

Heating Process

The circulation of cooling water is started to maintain the temperature on the UDS system such as a wall of the UDS chamber, the induction coil and the induction power unit and the matching box, Figure 3-5. The induction heating system is started under an automatic mode or a manual mode for the output control on heating. For example, in the automatic mode, a target temperature is set under the automatic-constant mode. Since an output gain dial works to set an upper limit for the power output in the automatic mode, a proper output limit is set for safety purpose. A certain time range needs to be waited such that the alloy is completely melted after it reached the set temperature on the crucible.

Spray and Deposition Process
On the starting stage on the droplets production, the conditions are inspected to confirm that the oxygen content, preset pressure on the regulated for a gas cylinder to the crucible. The function generator, the frequency divider, the strobe, the oscilloscope and the monitor are all turned on, prior to start producing the droplets. When the oxygen content is less than 1 ppm and the alloy is completely melted, the conditions for uniform droplets generation are met and spraying may be started. The stirring system is turned on if it required. The function generator, the frequency divider, the strobe, the oscilloscope and the monitor are all turned on. A 2.4 kHz sinusoidal wave is selected from the function generator for a 348 µm orifice experiment. The camera is focused on the orifice location to visualize the jet break-up. The voltage amplifier is turned on. The wave input form is monitored by the oscilloscope. The tape recorder is turned on to record the spraying. The gas circulation is turned off. Then the valves are switched so as to pressurize the melt to 2 psi higher than the chamber pressure to eject the melt through the office as a laminar jet. The imposed perturbations break the jet into uniform droplets. The droplets travel down in the chamber gas and are deposited on to the substrate. The frequency is tuned slightly and the crucible gas supply is adjusted properly to maintain the breakup stable and uniform till the end of spray. After all the melt is discharged, all the instruments are turned off. It is important that the oxygen analyzer is turned off first to prevent it from exposure to air. The gas circulation system is restored to its original state; the gettering furnace is isolated by the valves. The cooling fan of the gettering furnace continues to run till the cooling of furnace is completed. At the end of the experiment, the chamber is allowed to cool down gradually to room temperature with a slightly positive argon atmosphere kept inside the chamber.
3.3.2 Droplet-in-flight solidification model

The droplet thermal state can be thoroughly characterized as a function of the distance from the orifice by performing controlled experiments and further cross-verifying the experimental observations with model predictions. This section describes the experimental flow. The modeling of the droplet solidification path is described in Section 4.2.

3.3.3 Process-structure relationships

For a given alloy, the droplet thermal state is described by the diameter, the velocity, the temperature, the supercooling, the enthalpy, the S/L interfacial compositions, and the solid fraction/morphology of the droplets. The most important spray variable is the droplet diameter. It determines the droplet cooling rate for a given chamber atmosphere. The droplet cooling rate determines the supercooling of the droplet prior to nucleation for a given droplet size and nucleation potency.

Once the droplet state is fully characterized as a function of the position in space, the droplets can be deposited on a substrate to produce bulk deposits. The substrate position (height) is chosen so as to deposit mono-size $Mg-Zn-Y$ droplets in fully molten or mushy states, with regulated superheating, supercooling and solid fraction to produce fully dense bulk deposits with different novel rapid-solidification microstructures. In the present research, systematic UDS experiments were performed in the diameter range of
700 µm to 1000 µm to determine the correlations between the droplet diameter and the resultant microstructure.

The temperature and the motion of the substrate also affect the deposit microstructure and are therefore important variables in the droplet deposition experiments. In the present study the temperature of the substrate is kept constant at 298 K. For all the experiments, the substrate is moved on an X-Y table along programmed multiple-path scheme as shown schematically in Figure 3-20.

Figure 3-20: Schematic of the droplet deposition along a programmed path.
4. THERMAL/SOLIDIFICATION MODELING

An important goal of this research is to develop a microstructure-predictive process control model that addresses all the rapid solidification effects in droplet production and deposition. This section describes the modeling schemes for the non-equilibrium droplet processes.

4.1. Mg-I Phase Pseudo-Binary Phase Diagram

In order to provide the phase boundary information required in the simulation of the solidification of Mg-Zn-Y droplets during their flight, a pseudo-binary phase diagram was obtained for the Mg-I phase vertical section of the Mg-Zn-Y ternary phase diagram. This treatment was necessary because the current free dendritic growth models are all developed for binary alloys. The binary alloy approximation, used in the droplet simulation in this work, was justified on the basis that the phase boundary information needed in the simulation model lies in the Mg-I phase vertical section at low concentrations of yttrium, only as much as 10 at.%, and that the portion of the pseudo-binary phase boundary that is used in the simulation is the $\alpha$-Mg liquidus for which yttrium concentrations are even lower and less than 3.5 at.%

Since no exact thermodynamic data are available for the Mg-I phase pseudo-binary system, the phase diagram was approximated with linear phase boundaries. This was justified because most current free dendritic growth models are developed on linear liquidus and solidus [137, 140, 141].
The $Mg-I$ phase pseudo-binary system was identified as a eutectic system by Yi et al. [44] who suggested that the eutectic composition and temperature of the pseudo-binary system were $Mg_{73.2}Zn_{23.8}$ and 713 K, respectively [45, 47, 142]. Later Ok et al. [143] verified the occurrence of the eutectic reaction metallographically by observing a full eutectic microstructure in a cast alloy of composition $Mg_{72}Zn_{23.5}Y_{3.5}$. The eutectic temperature of 713 K was determined by differential thermal analysis (DTA) of several $Mg$-rich alloys, such as $Mg_{95}Zn_{4.3}Y_{0.7}$, etc [45, 47, 142].

Figure 4-1 shows the pseudo-binary $Mg-I$ phase eutectic phase diagram approximated by linear liquidus and solidus together with the experimental values of liquidus and solidus (diamonds) reported by Zaselyan et al. [144]. The linear solidus was drawn by fitting the experimental data. The linear liquidus, drawn simply between the melting point of pure Mg and the point representing the eutectic compositions and temperature, i.e., $Mg_{73.2}Zn_{23.8}$ and 713 K, shows excellent agreement with the experimental data points, indicating that the approximate pseudo-binary phase diagram bears reasonable accuracy.
Figure 4-1: The Mg - I-phase pseudo binary phase diagram obtained by linear-fitting of literature values of liquidus and solidus by Zaselyan et al. [144]. The horizontal line was drawn at 713 K to intersect with the liquidus at the eutectic composition of Mg$_{73.2}$Zn$_{23}$Y$_{3.8}$ [44].

As seen in Figure 4-1, the liquidus and solidus slopes are simply given as:

$$T_L = T_{m.p.} - 547.83 \, X_I, \quad T_S = T_L \, k, \quad k = S/E = 0.0373$$

(4.1)

where $T_L$ is $\alpha$-Mg liquidus temperature, $T_{m.p.}$ is the melting point of pure Mg, $T_S$ is $\alpha$-Mg solidus temperature, $k$ is the partition coefficient, $E$ is the eutectic composition, $S$ is the solubility limit at the eutectic temperature. $X_I$ is the atomic fraction of I-phase. The obtained pseudo-binary Mg-I phase is utilized in the droplet solidification model, which is described in Section 4.2.3, to acquire important parameters on the droplets solidification such as the solid/liquid interfacial velocity.
4.2. In-Flight Solidification

The thermal state of the droplets just before they hit the substrate or the prior splat was predicted by using the droplet in-flight solidification simulation model previously developed for Sn-Pb alloy droplet [133, 135], which is outlined in Section 2.3. The in-flight solidification simulation model incorporates a free dendritic growth model [136, 137] which accounts for curved liquidus and solidus and high Peclet number conditions and predicts the parameters required in the in-flight droplet solidification simulation, namely the dendrite growth velocity, the compositions at the solid-liquid interface, the non-equilibrium partition coefficient and the tip radius of the growing dendrites for a given value of droplet temperature. The dendritic growth model requires knowledge of the positions of the metastable phase boundaries in the relevant phase diagram. The metastable phase boundaries was determined for the pseudo-binary Mg - I-phase eutectic system by assuming linear liquidus and solidus as explained in Section 4.1 and incorporated in the growth model to predict the driving force for the growth of the primary phase, i.e., α-Mg solid solution, in Mg-Zn-Y alloys of relevant compositions. The droplet in-flight solidification model, which is based on a thermal balance equation coupled with a droplet motion equation [15, 133], uses the values of growth rate, tip radius, interfacial compositions from the dendritic growth model and predicts the droplet temperature, droplet enthalpy, fraction solidified, dendrite tip radius, as well as the droplet velocity and spray mass density as a function of flight distance for a given set of UDS parameters, i.e., orifice diameter, ejection pressure, vibration frequency, droplet charging voltage and chamber atmosphere gas.
4.2.1 Droplet motion model

The production of mono-size droplets in the UDS process is due to the instability of a laminar jet, which was first addressed by Rayleigh in 1878 [145]. A laminar jet breaks up into droplets because the surface energy of a liquid sphere is lower than that of a liquid cylinder of the same volume. Rayleigh assumed that the droplet breakup behavior can be induced into a controlled manner by imposing certain disturbances at the orifice far larger than those occurring naturally due to factors such as surface roughness. Rayleigh determined that the natural perturbations which grow most rapidly have a wavelength, $\lambda_m$, of:

$$\lambda_m = 4.51d_j$$

(4.2)

where $d_j$ is the jet diameter, by assuming that oscillations caused by disturbances grow exponentially as they travel from their point of inception within the orifice to farther downstream. Rayleigh also suggested that the application of a periodic oscillation in the exiting laminar jet would break the jet into mono-size spheres [145]. The breakup of a laminar jet into mono-size spheres can occur over a range of applied oscillation wavelengths. However, the applied oscillation frequency must result in a wavelength, $\lambda_d$, which is significantly larger than the wavelength of naturally occurring disturbances, $\lambda_m$. Rayleigh determined that applied oscillations must have wavelengths greater than the circumference of the laminar jet to cause instability in the jet. The frequency of droplet formation, which is equal to the frequency of the imposed disturbance, $f_d$, is given by:

$$f_d = \frac{\bar{v}_j}{\lambda_d}$$

(4.3)
where \( \vec{v}_j \) is the jet velocity. Later researchers, Schneieder and Hendricks [146], observed that a laminar jet can be broken into mono-sized or uniform droplets with applied oscillation wavelengths in the range of 3.5 to 7 times the jet diameter. These conditions have been used to produce uniform droplets of water [145-148], molten sodium nitrite [149], ink for inkjet printing technology [150, 151], and liquid metal [15-41, 133-136, 152-154].

The jet velocity at the orifice is given by:

\[
\vec{v}_j = k_f \sqrt{\Delta P_j} \quad (4.4)
\]

where \( k_f \) is an empirically determined orifice friction coefficient and \( \Delta P_j \) is the pressure difference in the crucible and the UDS chamber.

The droplet motion model adopted in the present work was originally presented by Passow [15] and Abel [135]. The fundamental droplet motion is governed by Newton’s second law and the droplets are subjected to the forces due to the gravitational field, \( \vec{F}_g \), aerodynamic drag, \( \vec{F}_d \), and the electrostatic repulsion among charged droplets, \( \vec{F}_{e_i} \):

\[
m_d \frac{d\vec{v}_d}{dt} = \vec{F}_g + \vec{F}_d + \sum \vec{F}_{e_i} \quad (4.5)
\]

where, \( \vec{v}_d \) is the droplet velocity, and \( m_d \) is the mass of an individual droplet.
Figure 4-2 illustrates a force balance on a traveling droplet. The gravitational force, $\vec{F}_g$, is given by:

$$\vec{F}_g = m_d \ddot{g}$$  \hspace{1cm} (4.6)

where, $\ddot{g}$ is the acceleration of the droplet due to gravity.

![Diagram of droplet forces](image)

Figure 4-2: Schematic showing the forces acting on a droplet moving in gaseous medium.

The droplets experience an aerodynamic drag force as they fly through the chamber atmosphere. The governing equation for the aerodynamic drag force on a droplet is given by [155, 156]:

$$\vec{F}_d = \frac{1}{8} C_d \pi \rho_d d_d^2 |\vec{v}_d| \vec{v}_d$$  \hspace{1cm} (4.6)
where, \( C_d \) is the drag coefficient for a single droplet, \( \rho_d \) is the droplet density and \( d_d \) is the droplet diameter.

The droplets are electrically charged with an equal magnitude and polarity so that they repel each other as they travel through the atmosphere and hence avoid merging. The total electrostatic force acting on a droplet \( i \) due to its \( n \) nearest neighboring droplets is given by [15]:

\[
\sum \vec{F}_{ij} = \frac{1}{4\pi\varepsilon_o} \sum_{j=0}^{n} \frac{q_d^2}{r_{ij}}
\]  

(4.7)

where, \( q_d \) is the charge on the droplet, \( \varepsilon_o \) is the permittivity constant of free space, and \( \vec{r}_{ij} \) is the distance vector between droplets \( i \) and \( j \) [15, 133].

In cases involving high temperature melts no charging is applied. Also depending on the alloy system, at elevated temperature the possible formation of metallic vapor or fume can lead to discharging. Thus, Equation (4.4) can be simplified to:

\[
m_a \frac{d\vec{V}_a}{dt} = \vec{F}_g + \vec{F}_d
\]  

(4.8)

Equation (4.8), however assumes no merging of droplets, which may not hold in the absence of charging.
4.2.2 Droplet thermal model

A droplet thermal model developed by DiVenuti [133] assumed Newtonian heat transfer by convection between the droplets and the chamber gas, while neglecting the radiational heat transfer, which becomes significant at high temperatures. The rate of enthalpy loss, which is presented by Gutierrez-Miravete [156] and by Mathur [155, 157], accounts for both effects, is given as:

$$m_d \frac{dh_d}{dt} = h_{dg} A_s^d (T_d - T_g) + \sigma \varepsilon A_s^d (T_d^4 - T_g^4)$$  (4.9)

where, $h_d$ is the droplet enthalpy per unit mass, $t$ is the time, $h_{dg}$ is the convective heat transfer coefficient between the droplets and the UDS chamber gas, $T_d$ is the droplet temperature, $T_g$ is the chamber gas temperature, $A_d^s$ is the surface area of the droplet, $\sigma$ is the Stefan-Boltzman constant, and $\varepsilon$ is the emissivity of the droplet.

The heat transfer coefficient, $h_{dg}$, is calculated by the Ranz and Marshall equation [158] for the convective heat transfer between a spherical droplet and the surrounding gas as:

$$h_{dg} = \frac{K_g}{d} \left(2.0 + 0.6 \text{Re}^{1/2} \text{Pr}^{1/3} \left(\frac{c_{p(g, avg)}}{c_{p(g)}}\right)^{0.26}\right)$$  (4.10)

where, $K_g$ is the thermal conductivity of the chamber gas, $c_{p(g)}$ is the specific heat of the chamber gas, $c_{p(g, avg)}$ is the specific heat of the gas evaluated at the average of the gas and the droplet temperatures, and $\text{Pr}$ is the Prandtl number, which is given by:
\[ \text{Pr} = \frac{\mu_g c_{p(g)}}{K_g} \] (4.11)

and \( R_e \) is the droplet Reynolds number given by:

\[ R_e = \frac{\overrightarrow{V_d} d_d \rho_g}{\mu_g} \] (4.12)

where \( \mu_g \) is the viscosity of the UDS chamber gas, \( \overrightarrow{V_d} \) is the absolute velocity of the droplet and \( \rho_g \) is the density of the chamber gas.

The rate of temperature change of a droplet in flight is calculated from:

\[
\frac{dT_d}{dt} = \frac{\Delta H_f}{c_{p(d)}} \frac{df_s}{dt} - \frac{h_{dg}}{\rho_d c_{p(d)}} \left( \frac{6}{d_d} \right) \left( T_d - T_g \right) - \frac{\sigma \varepsilon}{\rho_d c_{p(d)}} \left( \frac{6}{d_d} \right) \left( T_d^4 - T_g^4 \right)
\] (4.13)

where \( \Delta H_f \) is the latent heat of fusion of the metal or alloy, of the droplet per unit mass, \( f_s \) is the droplet solid fraction, and \( c_{p(d)} \) is the specific heat of the droplet [133].

The rate of solidification of a spherical droplet is defined as:

\[
\frac{df_s}{dt} = \frac{V_i A_i^s}{V_d}
\] (4.14)

where \( A_i^s \) is the area of the solid-liquid interface, \( V_d \) is the droplet volume, and \( V_i \) is the interface velocity given by an appropriate non-equilibrium solidification model such as BCT model developed by Boettinger, Coriell and Trivedi (BCT) [137] or the one by DiVenuti and Ando (DA) [133, 136]. The relevant details of Equations (4.9) – (4.14) are given in References [133] and [15].
4.2.3 Droplet solidification model

This section describes the relevant portions of the DA free dendritic growth model [133, 136] which provides the solid/liquid interfacial velocity, and presents predicted free-dendritic growth kinetics for the solidification of Mg-Zn-Y alloy droplets. The DA model is equivalent to the BCT model [137] for the case where the liquidus and solidus are linear [22, 133].

The “free dendritic growth” refers to the growth of a dendrite into a supercooled liquid. Figure 4-3 illustrates a schematic of the thermal and solutal profiles across the tip in free dendritic growth. The solid-liquid interface velocity $V_i$, far-field temperature $T_\infty$, and the far-field liquid solute concentration $X_0$ are treated as constants for the steady-state growth conditions. In rapid solidification, solute trapping arises due to incomplete atomic rearrangement at the solid-liquid interface. Under such conditions, the solid and liquid solute concentrations at the interface, $X^*_S$ and $X^*_L$, deviate from the equilibrium solid and liquid concentrations, $X^*_{eq}$ and $X^*_{eq}$, given by the solidus and liquidus, at the interface temperature, $T_i$, Figure 4-4.
The solid-liquid interface growth velocity depends on the total supercooling $\Delta T$ which is defined as the difference between the liquidus temperature of the alloy and the far-field liquid temperature in the droplet. The total supercooling, $\Delta T$, consists of four components [133, 137, 159, 160] and is given by,

$$\Delta T = \Delta T_c + \Delta T_r + \Delta T_c + \Delta T_k$$

where $\Delta T_c$ is the curvature supercooling arising from the curvature associated with the
solid/liquid interface, $\Delta T_i$ is the thermal supercooling originating from the difference between the interface temperature and the far-field temperature and $\Delta T_c$ is the solutal or constitutional supercooling resulting from the presence of a solute-rich boundary layer in the liquid ahead of the interface and $\Delta T_k$ the kinetic undercooling as illustrated in a schematic binary phase diagram accounted for interface curvature effects, Figure 4-4. The equilibrium solidus and liquidus solute concentrations can be expressed as functions of temperature, $f_S(T)$ and $f_L(T)$. $f_S^{-1}$ and $f_L^{-1}$ are thus the inverse functions of $f_S(T)$ and $f_L(T)$, respectively. As illustrated in Figure 4-4, the four components of total supercooling are given by [133, 137, 141, 160, 161]:

\[
\Delta T_r = \frac{2\Gamma}{r} \quad (4.16)
\]

\[
\Delta T_c = f_L^{-1}(X_o) - f_L^{-1}(X_L^*) \quad (4.17)
\]

\[
\Delta T_k = f_L^{-1}(X_L^*) - T_i \quad (4.18)
\]

\[
\Delta T_i = T_i - T_\infty = \frac{\Delta H_f IV(P_i)}{C_p} \quad (4.19)
\]

where $\Gamma$ is the Gibbs-Thompson coefficient, $r$ is the dendrite tip radius, $\Delta H_f$ is the heat of fusion, $C_p$ is the specific heat of the alloy and $IV(P_i)$ is the Ivantsov function [159, 162].
Figure 4-4: Schematic of a non-linear binary phase diagram.
The solid/liquid interface velocity is given by [133]:

\[
V_i = -V_o \left( \frac{\Delta G'}{RT_i} \right) = -V_o \left[ X_L^* (1 - k) + f_s(T_i + \Delta T_s) - f_L(T_i + \Delta T_s) + kX_L^* \ln \frac{k}{k_0} \right]
\]  

(4.20)

where \( \Delta G' \) is the driving force corrected for interface curvature, \( R \) is the gas constant and \( k_0^* \) is the equilibrium partition coefficient corrected for interface curvature. The non-equilibrium partition coefficient \( k \) is obtained from the Aziz equation [163] and can be rewritten as [133, 137]:

\[
k = \frac{k_0 + \beta_o V_i}{1 + \beta_o V_i} = \frac{f_s(T_i + \Delta T_s)/f_L(T_i + \Delta T_s) + 2\beta_o P_t \alpha / R}{1 + 2\beta_o P_t \alpha / R}
\]  

(4.21)

where \( \beta_o \) is the solute-trapping parameter, \( \alpha \) is the thermal diffusivity of the alloy, \( P_t \) is the thermal Peclet number, and \( R \) is the dendritic tip radius. The dendritic tip radius is given by [133]:

\[
R = \frac{k^*}{\sigma^*} \left( \frac{P_t \Delta H_f}{c_p} \right)^{\frac{1}{2}} \left( \frac{2P_t \alpha X_o (1-k)}{D_p(T_i) [1-(1-k) \ln (P_c)] \frac{df_L^{-1}}{dX |_{X=X_L^*}} \xi^*} \right)
\]  

(4.22)

where \( \sigma^* \) is the stability constant equal to \( 1/4\pi^2 \), \( P_c \) is the solutal Peclet number, \( \xi_i \) and \( \xi_c \) are the thermal and constitutional stability functions, respectively [161], and \( df_L^{-1}/dX |_{X=X_L^*} \) is the value of the slope of the liquidus at the interfacial solute concentration \( X_L^* \).
5. RESULTS AND DISCUSSION

5.1. Droplets and Deposits Production Experiments

5.1.1 AZ91D

Prior to performing the experiments with the Mg-Zn-Y alloys, preliminary deposition experiments were conducted with AZ91D to optimize the motion patterns of the X-Y table. These parameters were used as references for the deposition experiments with the Mg-Zn-Y alloys. Figure 5-1 shows a break-up image captured from one of the experiments with AZ91D alloy. Figure 5-2 shows the deposits produced with 700 µm droplets under the experimental conditions shown in Table 5-1. The other procedures of the UDS experiments are described in detail in Chapter 3.

Figure 5-1: Break-up image captured during AZ91D UDS experiment.
Table 5-1: Experimental Parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orifice Diameter [µm]:</td>
<td>348</td>
</tr>
<tr>
<td>Pressure Differential [psi]:</td>
<td>5</td>
</tr>
<tr>
<td>Perturbation Frequency [kHz]:</td>
<td>2.39</td>
</tr>
<tr>
<td>Melt Temp [K]:</td>
<td>973</td>
</tr>
<tr>
<td>Alloy:</td>
<td>AZ91D</td>
</tr>
<tr>
<td>Alloy Melt Temp [K]:</td>
<td>~868</td>
</tr>
<tr>
<td>Chamber Atmosphere:</td>
<td>Argon</td>
</tr>
<tr>
<td>Flight Distance [m]:</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Figure 5-2: AZ91D deposit produced with 700 µm in order to optimize motion parameters for the X-Y table.
5.1.2 \( \text{Mg}_{97}\text{Zn}_{1}\text{Y}_2 \)

Spray deposits of \( \text{Mg}_{97}\text{Zn}_{1}\text{Y}_2 \) were produced with 700 \( \mu \text{m} \) and 1000 \( \mu \text{m} \) droplets under the conditions listed in Table 5-2. The UDS experimental procedures are detailed in Chapter 3. Figure 5-3 shows a break-up image captured from the \( \text{Mg}_{97}\text{Zn}_{1}\text{Y}_2 \) UDS experiment. Simulation of the droplet temperature for Experiments 1, 2 and 3 (700, 1000 and 700 \( \mu \text{m} \) droplets), Figure 5-4, indicates that the droplets were still superheated in these experiments by, respectively, 109 K, 63 K and 62 K, above the liquidus (903 K) when they reached the substrate. Superheated droplet deposition conditions allow the liquid droplets to fill in droplet interstices and produce \( \text{Mg}_{97}\text{Zn}_{1}\text{Y}_2 \) deposits with practically full density [22]. In the deposition experiments, the melt temperature was set at approximately 1023 K in the Experiment 1, while it was lowered to 973 K to avoid deleterious effects of oxidation in Experiments 2 and 3.

Figures 5-5 and 5-6 show the microstructures of the original \( \text{Mg}_{97}\text{Zn}_{1}\text{Y}_2 \) ingot. The ingot exhibits a coarse non-orthogonal dendritic structure that is characteristic of a cast magnesium alloy. The dendritic grains in the ingot are 100 \( \mu \text{m} \) - 2 mm in size and have a secondary dendrite arm spacing (SDAS) of about 20 \( \mu \text{m} \), which is expected for the slow solidification rates associated with conventional casting.

Figures 5-7 and 5-8 show the microstructures of the spray deposit produced with 700 \( \mu \text{m} \) droplets, taken at a spray boundary and near the chill surface, respectively. Figure 5-7 indicates that many of the grains formed in the as-sprayed deposit grew epitaxially through the spray boundaries.
Figure 5-8 shows that the as-sprayed deposit has an equiaxed structure where the grains exhibit a non-dendritic morphology characterized by precipitates formed on a habit plane and serrated grain boundaries. From Figures 5-6 to 5-8, it is evident that the equiaxed grains in the as-sprayed deposit are much smaller (40 - 200 µm) than the dendritic grains (~2 mm) in the as-cast microstructure, reflecting the rapid solidification conditions during the droplet deposition process. Formation of an equiaxed microstructure directly from the molten state is an important characteristic of a spray deposition process as reported in many previous studies of conventional spray forming [164]. Another important observation is that the spray deposit contains essentially no porosity as opposed to conventionally spray formed materials in which a finite amount of porosity is always found [164]. Such full-density spray forming is possible only with a thermal spray that consists only of molten droplets such as UDS [22].
Figure 5-3: Break-up image captured from the $Mg_{97}Zn_{1}Y_{2}$ UDS experiment.

<table>
<thead>
<tr>
<th>Experiment No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orifice Diameter [μm]:</td>
<td>348</td>
<td>500</td>
<td>348</td>
</tr>
<tr>
<td>Pressure Differential [psi]:</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Perturbation Frequency [kHz]:</td>
<td>2.39</td>
<td>1.03</td>
<td>2.39</td>
</tr>
<tr>
<td>Crucible Temp [K]:</td>
<td>1023</td>
<td>973</td>
<td>973</td>
</tr>
<tr>
<td>Chamber Atmosphere</td>
<td>Argon</td>
<td>Argon</td>
<td>Argon</td>
</tr>
<tr>
<td>O$_2$ contents [ppm]:</td>
<td>1.08</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Flight Distance to Substrate: [m]</td>
<td>0.30</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>Droplet Diameter [μm]:</td>
<td>700</td>
<td>1000</td>
<td>700</td>
</tr>
<tr>
<td>Droplet temp. at substrate [K]</td>
<td>1012</td>
<td>967</td>
<td>965</td>
</tr>
<tr>
<td>Experimental Condition</td>
<td>Unstable stream</td>
<td>Unstable stream</td>
<td>Stable</td>
</tr>
</tbody>
</table>
Figure 5-4: Temperature vs. flight distance of 1: 700 μm, 2: 1000 μm, and 3: 700 μm for Mg₉Zn₁Y₂ droplets calculated for assumed undercoolings ΔT = 50K, for melt temperatures of 973 K and 1073 K.
Figure 5-5: Cross section of the Mg$_{97}$Zn$_1$Y$_2$ ingot filled with coarse dendritic structures.
Figure 5-6: Optical micrographs of the bulk ingot supplied from FUKUDA. (a) Low magnification. (b) High magnification.
Figure 5-7: Optical micrographs showing the spray boundary in the spray deposit produced with the 700 µm $Mg_9Zn_1Y_2$ droplets (Experiment 1).
Figure 5-8: Optical micrographs taken near the chill surface in the spray deposit produced with 700 μm $Mg_9Zn_7Y_2$ droplets (Experiment 1). (a) low magnification, (b) medium magnification, and (c) high magnification.
Figure 5-8: Continued.
The serrated matrix grains in the deposit produced with 700 µm droplets (Figure 5-8) indicate that their growth occurred in a certain preferential crystallographic direction. The latter is also seen from the occurrence of striations in the matrix grains, which is indicative of second-phase precipitation on a habit plane [165]. Also, the matrix grains touch directly each other leaving no room for other structures, such as eutectic constituents, to form between them. This suggests that in the region near the bottom chill surface of this deposit, there was little solute concentration in the liquid remaining toward the end of solidification, which permitted the matrix grains to grow until they impinge on each other, completing the solidification. Such serrated boundaries may have important effects on the mechanical properties of the material. Similar grain boundary serrations have been observed in Ti-Al alloys [166] in which the serrated boundaries improve creep resistance.

Figure 5-9 shows an X-ray diffraction (XRD) pattern of the same spray deposit produced with 700 µm droplets. The major XRD peaks are reflections of the hcp magnesium matrix. The X-phase (Mg12ZnY) [69, 70] is also formed as the second phase in this spray deposit, while there is no clear indication of other phases such as the compound Mg24Y5 and/or I phase with similar compositions [44-47, 58, 72]. Thus, the XRD results strongly suggest that the precipitates that formed the striations are the X-phase.
Figure 5-9: XRD pattern of $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 700 µm droplets from Experiment 1.

Figures 5-10 and 5-11 show micrographs of a spray deposit produced with 1000 µm droplets, i.e., Experiment 2, at the chill zone and the top of the deposit, respectively. Both locations show equiaxed microstructures consisting of matrix grains with striations similar to those in the deposit produced with 700 µm droplets, Figure 5-8 (c), but the matrix grains apparently did not grow until they collided with one another, leaving a layer of liquid between them that solidified by itself, replacing the growth of the primary grains at the end of solidification. This is indicative of a higher growth front temperature, which would promote solute partitioning that concentrates solutes in the remaining liquid. The liquid is considered to solidify into a eutectic-like constituent.
The deposit from Experiment 1, produced with 700 µm droplets, too, had a similar microstructure in regions distant from the chilled bottom surface, Figure 5-7 (a), indicating that the cooling rates in these regions of the deposit were lower than those in the chilled region.

Comparison of Figures 5-8 (c) and 5-10 (a) shows that the matrix grains in the chill zone of the deposit produced with 700 µm droplets are coarser (40 - 200 µm) than the grains in the chill zone of the deposit produced by 1000 µm droplets (10 - 60 µm). This indicates that the solidification of the sprayed layers in the latter deposit involved more nuclei and lower growth rates of the matrix grains than in the deposit produced with 700 µm droplets.

Figure 5-11 also shows that the grains in the region just above the layer deposited in the previous spray cycle are much smaller than those in the prior sprayed region. This is because the prior sprayed layer was largely solidified when the droplets in the next cycle were deposited, acting as a quenching substrate. A close look at the spray boundary region suggests that the growth of the grains at the spray boundaries was largely epitaxial through the spray boundary. Such epitaxial growth would favor a very rate of heat transfer at the spray boundary which caused the observed grain size reduction in the region above the spray boundary.

Another interesting observation made in Figure 5-11 is that the striations that are caused by the precipitation on a habit plane are more clearly revealed in the region near the edges of the matrix grains. This indicates that the interior of the primary grains is associated with a lower degree of precipitation, caused by even lower interfacial temperatures during the initial stages of the growth of the grains. As the grains grew,
the interface temperature increased, allowing for more precipitation toward the peripheries of the grains.

The striations in the equiaxed matrix grains in the region near the chill surface, Figure 5-10, are more uniform and go across the grains. Thus, the quench rate in this region was not as high, despite the location of near the ‘chill surface’. This apparent discrepancy may be explained on the basis of heat transfer, which was not so high as the deposit-substrate interface as that at a spray boundary where epitaxial growth permitted high rates of heat transfer.
Figure 5-10: Optical micrographs taken near the chill surface in the $Mg_{97}Zn_Y Y_2$ deposit produced with 1000 µm droplets (Experiment 2), (a) low magnification and (b) high magnification.
Figure 5-11: Optical micrographs taken near the top of the deposit in the $Mg_{97}Zn_1Y_2$ deposit produced with 1000 µm droplets (Experiment 1), (a) low magnification and (b) high magnification.
Figure 5-12 shows micrographs of another spray deposit produced with 700 µm droplets near the chill and top surface of the deposit (Experiment 3). This deposit, however, was produced at a lower melt temperature of 973 K, instead of 1023 K that was used in Experiment 1. The lower melt temperature was applied to minimize oxidation, which may occur at the orifice clogging the orifice partially or even completely. Any partial orifice clogging makes the metal flow unstable and thus causes non-uniform deposition conditions. In fact, the metal flow in Experiment 1 was observed to be somewhat unstable, whereas that in Experiment 3, done with the same orifice size, was very stable. This is clearly reflected in the micrographs of the deposit shown in Figure 5-12, where equiaxed grains uniformly fill the entire volume of the deposit. The grain size increases gradually from the chill zone (10 ~ 40 µm) to the top region (20 ~ 60 µm), reflecting the gradual decrease in cooling rate as the deposit grew.

The stable metal flow (deposition rate) also had an effect on the microstructure at spray boundaries. As seen in Figure 5-13, the spray boundary region in this deposit also exhibit largely epitaxial deposit growth through a spray boundary. However, the spray boundaries were less discernible than those in the deposit from Experiment 1, indicating that the prior deposit surface, when the next spray cycle began, was not as much oxidized as that in Experiment 1. This indicates that the metal flow remained steady, and sufficiently high, throughout the experiment. As a result, the primary grains did not grow to collide one another even in the chill zone, Figure 5-12 (b), leaving intergranular liquid to solidify at the end, Figures 5-12 and 5-13. The presence of liquid kept the prior deposit surface near the liquidus temperature when the next spray cycle began. Consequently, the droplets solidifying in the hot prior deposit surface were not
quenched severely, and no appreciable change in grain size resulted at the spray boundaries in this deposit.

Figure 5-14 compares the spray boundary regions of the deposits produced in Experiments 1, 2 and 3. In all experiments, epitaxial growth through spray boundaries is evident, although the spray boundaries in the deposit in Experiment 1 (produced with 700 µm droplets) are clearly discernible due to the oxidation of prior deposit surface (which was less severe in Experiment 3, done with 700 µm droplets, but at a higher pouring temperature of 1023 K, and absent in Experiment 2 done with 1000 µm droplets). The occurrence of epitaxial growth though spray boundaries in Experiments 1 and 3 suggest that the oxidation film was not fully continuous, or otherwise was porous, permitting the prior grains to grow epitaxially through it.

The apparent absence of oxide film formation in Experiment 2 done with 1000 µm droplets is indicative of a high prior deposit surface temperature kept around the liquidus temperature and a low rate of heat conduction into the prior deposit. Such conditions would occur in this experiment done with 1000 µm droplets which kept the temperature gradients lower than those in Experiments 1 and 3 done with 700 µm droplets. The low temperature gradients in the material deposited toward the end of the spray cycle translate into low rates of solidification in this region of the material. Therefore, the resultant grains are coarse in these regions. The material deposited at the beginning of a spray cycle, however, is solidified at a high rate as the temperature gradient in the prior deposited surface has regained during the spray intermission. This produced a discontinuous decrease in grains size as seen in Figure 5-14 (b).
Figure 5-12: Optical micrographs of $Mg_{97}Zn_1Y_2$ deposit produced with 700 $\mu$m droplets (Experiment 3), (a) near top of the deposit and (b) near the chill surface.
Figure 5-13: Optical micrographs of $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 700 µm droplets at a spray boundary (Experiment 3), (a) low magnification and (b) high magnification.
Figure 5-14: Comparison of microstructures at the spray layer boundaries of the deposits from (a) Experiment 1 (700 µm droplets), (b) Experiment 2 (1000 µm droplets) and (c) Experiment 3 (700 µm droplets).
Figures 5-15 and 5-16 show X-ray diffraction (XRD) patterns of the spray deposits from Experiment 2 (1000 μm droplets) and Experiment 3 (700 μm droplets), respectively. The XRD peaks match those of the α-Mg and the X-phase ($Mg_{12}ZnY$). Thus, the XRD peaks of these deposits are essentially identical to those of the $Mg_{97}Zn_{1}Y_{2}$ deposit produced in Experiment 1 with 700 μm droplets, Figure 5-9. This indicates that there is no significant phase change or evolution within the range of the experimental parameters such as the crucible temperature (973 – 1023 K) and the droplet diameter (700 – 1000 μm) used in Experiments 1 - 3.
Figure 5-15: XRD pattern of $\text{Mg}_{97}\text{Zn},\text{Y}_2$ deposit produced with 1000 µm droplets (Experiment 2).

Figure 5-16: XRD pattern of $\text{Mg}_{97}\text{Zn},\text{Y}_2$ deposit produced with 700 µm droplets (Experiment 3).
Figure 5-17 shows results from electron probe micro analysis (EPMA), of the $Mg_{97}Zn_1Y_2$ deposit produced with 700 µm droplets (Experiment 3), conducted with assistance of FUKUDA Metal Foil & Powder Co., Ltd. The back-scattered electron image in Figure 5-17 (a) indicates that the primary grains (the dark regions) are associated with a light element, which in $Mg_{97}Zn_1Y_2$ must be $Mg$. The intergranular regions (the lighter regions) are clearly enriched with the heavier elements of $Zn$ and $Y$. Figures 5-17 (b), (c) and (d) are elemental maps of the same region of the specimen showing the distributions of $Mg$, $Zn$ and $Y$, respectively. As expected from the back-scattered electron image, the primary grains are rich in $Mg$, whereas the intergranular regions are enriched with both $Zn$ and $Y$.

The above partitioning of the elements is quantitatively verified by the energy dispersive X-ray fluorescence (EDX) analysis of a primary grain (a) and an intergranular region (b) of the same deposit shown in Figure 5-18. The EDX results of the two spots, summarized in Table 5-3, clearly show that the primary grain in fact has a composition of 98.6 at.% of $Mg$, 0.4 at.% of $Zn$ and 1.0 at.% of $Y$, and that the composition of the intergranular region is 91.6 at.% of $Mg$, 3.1 at.% of $Zn$ and 5.3 % of $Y$. Although these values of concentrations may be affected by possible overlapping of the signals from the primary grain and the intergranular region because of the beam size used of about 10 µm x 10µm vs. the size of the features in the microstructure, they certainly attest that the primary grains consist essentially of $\alpha-Mg$ solution and that the intergranular regions are associated with a $Mg-Zn-Y$ compound, which according to the XRD result in Figure 5-16, should be the $X$-phase ($Mg_{12}ZnY$). Taking the $Zn$ and $Y$ concentrations of 3.1 at.% and 5.3 at. % determined for the intergranular region (Table 5-3) to be of the same level (justified on the basis of possible errors due to signal overlapping), the $Mg$ concentration,
expected for the case where the intergranular region is entirely made of $X$-phase, would be about 50 at.%. Since this is lower than the actual EDX value of 91.6 at.%, the intergranular regions are considered to have some $\alpha$-Mg as well as the $X$-phase, indicating presence of a eutectic structure in the intergranular region.

The EDX data in Table 5-3 do not decisively show whether the primary grains are associated with precipitation within them as indicated metallographically by the striations in these grains (Figure 5-14). However, the average $Y$ concentration of the primary grains, which is only about 1.0 at.% as shown in Table 5-3, seems to indicate that there may not be much precipitation in the grain. Thus, intergranular precipitation is not suggested by the EPMA results. However, the presence of intergranular constituents around the grain ‘a’ suggests that alloy partitioning was complete in this region, thus allowing little precipitation inside this grains.

The striations in primary grains suggest that the intragranular precipitation appear to have occurred on a habit plane in the $\alpha$-Mg grains. Since each grain shows striations (i.e., trace of the habit planes) only in one direction, it may be deduced that the habit planes are the basal planes.

The observations of serrated boundaries of the primary grains in the chilled region of the deposit from Experiment 1 indicates that the precipitation on the habit planes may have occurred in-situ as the grains grew to impinge upon each other. Such in-situ precipitation during the growth of primary grains would provide a mechanism for alloy partitioning within the growing solid which does not cause solute accumulation in the liquid at the solidification front. Such a mode of growth may therefore be regarded as coupled growth of two phases. Under such conditions, growing primary grains can impinge on each other without forming solute-enriched interdendritic regions, forming
serrated grain boundaries, as seen in the chilled zones in the deposit from Experiment 1. Apparently, in other deposits produced under different conditions, solute partitioning did occur, producing intergranular liquid which solidified between the primary grains as evident in Figures 5-12 and 5-13. Thus, the occurrence of coupled growth depends on the local solidification conditions, and, is probably favored by high rates of heat removal which may produce undercooled solidification front. However, further work is necessary to fully understand the mechanism of such a solidification behavior.

The serrated grain boundaries may be of industrial importance as they may improve the resistance to creep through increased resistance to grain boundary sliding. In fact, serrated grain boundaries in Ti-Al alloys increase creep resistance. Figure 5-19 show such serrated boundaries in a Ti - 48 at.% Al alloy that was heat treated in a \( \gamma - \alpha_2 \) two phase region to form alternating lamellae of \( \gamma \) (L1\(_1\)) and \( \alpha_2 \) (D0\(_{19}\)) phases [165]. The serrated boundaries are formed as the \( \gamma \) and \( \alpha_2 \) lamellae from two primary \( \gamma \) grains impinged. Although the latter transformation occurs in the solid state, it resembles the striated growth seen in the \( Mg_{97}Zn_1Y_2 \) deposits in that simultaneous formation of two phase permit growth without solute accumulation in the transformation front.
Figure 5-17: EPMA analysis of $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 700 µm droplets (Experiment 3). (a) Back-scattered image indicating concentration of the heavier elements Zn and Y in the intergranular regions. (b)-(d) Color mappings of Mg, Zn and Y confirming the enrichment of Zn and Y in the intergranular region and high Mg content in the equiaxed primary grains.
Figure 5-17: Continued.
Figure 5-18: Back-scattered electron image of the region subjected to EDX analysis. The specimen is from the $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ deposit produced with 700 $\mu$m droplets (Experiment 3), a: dark region - the equiaxed grain, b: gray region - the intergranular region. The EDX results are listed in Table 5-3.

Table 5-3: EDX Result for the $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ Deposit Produced with 700 $\mu$m Droplets (Experiment 3).

<table>
<thead>
<tr>
<th></th>
<th>Mg (at.%)</th>
<th>Zn (at.%)</th>
<th>Y (at.%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a (ave.)</td>
<td>98.6</td>
<td>0.4</td>
<td>1.0</td>
</tr>
<tr>
<td>a-1</td>
<td>98.87 ± 0.26</td>
<td>0.41 ± 2.48</td>
<td>0.72 ± 2.6</td>
</tr>
<tr>
<td>a-2</td>
<td>98.43 ± 0.23</td>
<td>0.45 ± 2.24</td>
<td>1.12 ± 2.33</td>
</tr>
<tr>
<td>a-3</td>
<td>98.35 ± 0.23</td>
<td>0.46 ± 2.18</td>
<td>1.19 ± 2.26</td>
</tr>
<tr>
<td>b (ave.)</td>
<td>91.6</td>
<td>3.10</td>
<td>5.30</td>
</tr>
<tr>
<td>b – 1</td>
<td>93.50 ± 0.22</td>
<td>2.18 ± 1.63</td>
<td>4.32 ± 1.61</td>
</tr>
<tr>
<td>b – 2</td>
<td>89.75 ± 0.18</td>
<td>4.07 ± 1.11</td>
<td>6.18 ± 1.07</td>
</tr>
<tr>
<td>b – 3</td>
<td>91.39 ± 0.2</td>
<td>3.06 ± 1.33</td>
<td>5.55 ± 1.3</td>
</tr>
</tbody>
</table>

* 3 points are tested for each target area.
Figure 5-19: Microstructure of Ti-Al alloy which has striated morphologies resembling that of the Mg97Zn1Y2 deposit produced with 700 µm droplets. (a): Ti - 47.5 at.% Al. (b): Ti - 48.2 at.% Al [166].
5.1.3  \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_2 \)

Spray deposits of \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_2 \) alloy were produced with 700 \( \mu \text{m} \) droplets under the conditions listed in Table 5-4. The original ingot of \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_2 \) had a coarse dendritic microstructure, Figure 5-20, which consists of dendrites of \( \alpha\)-\( \text{Mg} \) solid solution and interdendritic eutectic of \( \text{Mg}_3\text{Y}_2\text{Zn}_3 \) (or \( I \)-phase) and \( \alpha\)-\( \text{Mg} \) solid solution. Figure 5-21 shows a XRD pattern of the ingot which confirms the presence of three phases, \( \alpha\)-\( \text{Mg} \), \( I \)-phase and \( W \)-phase (\( \text{Mg}_3\text{Y}_2\text{Zn}_3 \)). The peaks indicated with solid squares are from the \( \alpha\)-\( \text{Mg} \) matrix. The other two phases, which are thought to be associated primarily with the interdendritic eutectic, Figure 5-20 (b), are the \( I \)-phase (triangles) and \( W \)-phase (diamonds). Of the two intermetallic phases, the \( I \)-phase is the more predominant phase. The XRD result confirms the occurrence of \( I \)-phase previously reported in the alloys that belong to the \( \text{Mg}-I \) phase pseudo binary system. Unlike the \( \text{Mg}_{97}\text{Zn}_1\text{Y}_2 \) alloy, the \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_2 \) alloy has no \( X \)-phase. Figure 5-22 shows an optical micrograph of the three different regions (A, B and C) present in the microstructure. Region A corresponds to the \( \alpha\)-\( \text{Mg} \) grains that cover major portions of the microstructure. Region B is the interdendritic eutectic consisting of \( \alpha\)-\( \text{Mg} \) solid solution and \( I \) and/or \( W \) phases. Region C refers to a pocket of probably \( I \)-phase [167].

Figure 5-23 shows a break-up image captured from the \( \text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5} \) experiment. Figure 5-24 shows an actual view of the \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_2 \) deposit produced. The as-sprayed deposit, exhibits a fine, equiaxed grains 5-20 \( \mu \text{m} \) in diameter, Figure 5-25, which sharply contrasts the much coarser, dendritic structure in the cast ingot. Unlike the primary grains in the spray deposit of \( \text{Mg}_{97}\text{Zn}_1\text{Y}_2 \), no striations indicative of precipitation within \( \alpha\)-\( \text{Mg} \) grains are noted in the \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_2 \) deposit.
Figure 5-20: Optical micrographs of the bulk ingot of $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$
 supplied from FUKUDA. (a) Low magnification. (b) High magnification.
Figure 5-21: XRD pattern of $Mg_{88}Zn_{10}Y_2$ as-cast ingot.

Figure 5-22: Optical micrographs of the bulk ingot of $Mg_{88}Zn_{10}Y_2$. Region A, B and C represent $\alpha$-Mg, $\alpha$-Mg and I and/or $W$ eutectic, and probably I- phase, respectively.
Figure 5-23: Break-up image captured from the $Mg_{88}Zn_{10}Y_2$ UDS experiment.

Table 5-4: Experimental Parameters for the $Mg_{88}Zn_{10}Y_2$ Deposit.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orifice Diameter [µm]</td>
<td>348</td>
</tr>
<tr>
<td>Pressure Differential [psi]</td>
<td>3</td>
</tr>
<tr>
<td>Perturbation Frequency [kHz]</td>
<td>2.39</td>
</tr>
<tr>
<td>Crucible Temp [K]</td>
<td>923</td>
</tr>
<tr>
<td>Alloy [at%]: $Mg_{88}Zn_{10}Y_2$</td>
<td></td>
</tr>
<tr>
<td>Chamber Atmosphere:</td>
<td>Argon</td>
</tr>
<tr>
<td>$O_2$ contents [ppm]:</td>
<td>&lt; 1.0</td>
</tr>
<tr>
<td>Flight Distance [m]:</td>
<td>0.25</td>
</tr>
<tr>
<td>Droplet Diameter [µm]:</td>
<td>700</td>
</tr>
<tr>
<td>Droplet temp. at substrate [K]</td>
<td>917</td>
</tr>
</tbody>
</table>
Figure 5-24: $Mg_{88}Zn_{10}Y_2$ deposit produced with 700 μm droplets.
Figure 5-25: Optical micrographs of the spray deposit of the $Mg_{88}Zn_{10}Y_2$ alloy. (a) Low magnification. (b) High magnification.
Figure 5-26 (a) shows an XRD pattern of the $Mg_{88}Zn_{10}Y_2$ spray deposit, shown together with that of the as-cast ingot for composition. As in the cast alloy, the same three phases, $\alpha$-Mg, I and W phases were identified in the spray deposit. However, the XRD pattern of the as-sprayed $Mg_{88}Zn_{10}Y_2$ deposit shows lower intensities of I-phase peaks relative to those of $\alpha$-Mg than those of the XRD pattern of the as-cast ingot. This implies that the rapid solidification effects of the spray deposition process are reflected not only in microstructural refinement, but also in suppressing the precipitation of secondary phases, which is mainly the I-phase in this case. As explained in Section 2, the I-phase is known to be a hard and brittle phase. Thus it is preferred to minimize the formation of coarse particles of I-phase.

Figure 5-27 shows elemental EPMA mapping results of the as-sprayed $Mg_{88}Zn_{10}Y_2$ deposit. Clearly, Mg is associated mainly with the equiaxed primary grains, confirming that these grains are $\alpha$-Mg solid solution. Zn and Y are mostly in the intergranular regions, indicating that the I and W phases form in the intergranular regions.

Figure 5-28 and Table 5-5 show the results of EDX performed on a primary grain (a), an intergranular precipitate (b) and intergranular eutectic (c). The primary grain (a) contains approximately 97.4 at.% of Mg, 2.5 at.% of Zn and 0.1 at.% of Y. The intergranular precipitate (b) has an average composition of 88.5 at.% of Mg, 9.8 at.% of Zn and 1.7 % of Y, whereas the intergranular eutectic (c) has an average composition of 73.3 at.% of Mg, 23.4 at.% of Zn and 3.3 % of Y. Although the latter compositions, particularly those of regions (b) and (c), may not be accurate due to possible signal overlaps in the EDX analysis, the Zn/Y ratio of region (b) is close to 6/1, the one expected for I-phase. Thus the precipitate (b) may indeed be the I-phase. The Zn/Y ratio of Region (c) is about 8/1. Since this ratio is the average of the phases present in the
eutectic, it is not possible to identify the phases with clarity, although presence of $I$-phase is likely. Also, presence of $W$-phase identified by XRD, in the eutectic is a possibility. Further work is thus necessary to fully characterize the phase evolution in the spray deposit.
Figure 5-26: XRD patterns of the $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$: (a) as-sprayed deposit and (b) as-cast ingot. As-sprayed deposit has lower relative intensities of $I$-phase with respect to those of $\alpha$-Mg.
Figure 5-27: EPMA analysis of the $Mg_{88}Zn_{10}Y_2$ deposit produced with 700 µm droplets. (a) Back-scattered image indicating concentration of the heavier elements Zn and Y in the intergranular regions. (b)-(d) Color mappings of Mg, Zn and Y confirming the enrichment of Zn and Y in the intergranular region and high Mg content in the equiaxed primary grains.
Figure 5-27: Continued.
Figure 5-28: Back-scattered electron image of the region subjected to EDX analysis. The specimen is from the $Mg_{88}Zn_{10}Y_2$ deposit produced with 700 µm droplets. a: dark region - the equiaxed grain. b: white region - the intergranular region. c: lamellar region. The EDX results are listed in Table 5-5.

Table 5-5: EDX Result for the $Mg_{88}Zn_{10}Y_2$ Deposit Produced with 700 µm Droplets.

<table>
<thead>
<tr>
<th>*</th>
<th>Mg  (at%)</th>
<th>Zn  (at%)</th>
<th>Y   (at%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a (ave.)</td>
<td>97.4</td>
<td>2.5</td>
<td>0.1</td>
</tr>
<tr>
<td>a-1</td>
<td>97.47 ± 0.19</td>
<td>2.47 ± 1.54</td>
<td>0.06 ± 1.64</td>
</tr>
<tr>
<td>a-2</td>
<td>97.28 ± 0.18</td>
<td>2.51 ± 1.44</td>
<td>0.21 ± 1.53</td>
</tr>
<tr>
<td>a-3</td>
<td>97.40 ± 0.19</td>
<td>2.52 ± 1.49</td>
<td>0.08 ± 1.60</td>
</tr>
<tr>
<td>b (ave.)</td>
<td>88.5</td>
<td>9.8</td>
<td>1.7</td>
</tr>
<tr>
<td>b – 1</td>
<td>91.08 ± 0.15</td>
<td>7.99 ± 0.81</td>
<td>0.92 ± 0.86</td>
</tr>
<tr>
<td>b – 2</td>
<td>88.46 ± 0.16</td>
<td>8.18 ± 0.82</td>
<td>3.36 ± 0.85</td>
</tr>
<tr>
<td>b – 3</td>
<td>85.85 ± 0.15</td>
<td>13.39 ± 0.59</td>
<td>0.76 ± 0.64</td>
</tr>
<tr>
<td>c (ave.)</td>
<td>73.3</td>
<td>23.4</td>
<td>3.3</td>
</tr>
<tr>
<td>c-1</td>
<td>72.34 ± 0.15</td>
<td>23.99 ± 0.46</td>
<td>3.67 ± 0.49</td>
</tr>
<tr>
<td>c-2</td>
<td>71.13 ± 0.17</td>
<td>28.87 ± 0.45</td>
<td>&lt; 0</td>
</tr>
<tr>
<td>c-3</td>
<td>76.46 ± 0.12</td>
<td>17.21 ± 0.44</td>
<td>6.34 ± 0.45</td>
</tr>
</tbody>
</table>

* 3 points are tested for each target area.
5.1.4 $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$

Spray deposits of $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ alloy were produced with 700 µm droplets. The original ingot of $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ used for this spray deposition experiment, had a coarse dendritic microstructure, as seen in Figure 5-29, that consisted of $\alpha$-$\text{Mg}$ dendrites and interdendritic eutectic of $W$-phase (or $I$-phase) and $\alpha$-$\text{Mg}$ solid solution, which is similar to the case of $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ ingot, but with slightly finer dendrites. Also, the $\text{Mg}_{51}\text{Zn}_{20}$ binary intermetallic compound, which was previously called $\text{Mg}_7\text{Zn}_3$ and thought to be related to the $I$-phase [50, 106-108], was detected by XRD, Figure 5-30.

Figure 5-31 shows a jet breakup image captured from a video of $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ UDS experiment which was performed under the condition listed in Table 5-6.

Figure 5-32 shows an actual view of the $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposit which was produced together with a patterned tail which was deposited to test the motion of the X-Y table. Figure 5-33 shows optical micrographs of the $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ spray deposit. The as-sprayed deposit has an equiaxed structure consisting of refined grains (20 ~ 50 µm) which is contrasted by the coarse dendritic microstructure of the ingot, Figure 5-29 (a). The primary grains still partially preserve their original dendritic morphology although they are randomly oriented as a result of breakup while forming during the deformation of droplets upon landing. The intergranular regions show a eutectic structure consisting at least of two phases. Figure 5-34 (a) shows an XRD pattern of the $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ spray deposit. Similar to the $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ deposit, the XRD peaks are indexed as those of $\alpha$-$\text{Mg}$ matrix (solid squares), $I$-phase ($\text{Mg}_3\text{Zn}_6\text{Y}$, gray triangles) and $W$-phase ($\text{Mg}_3\text{Y}_2\text{Zn}_3$, open diamonds). Of the two $\text{Mg}$-$\text{Zn}$-$\text{Y}$ intermetallics, the $I$-phase is clearly more
predominant than the $W$-phase. Figure 5-34 also suggests that the as-sprayed deposit does not contain $\text{Mg}_7\text{Zn}_3$, in appreciable amounts, as opposed to the cast ingot that had a significant amount of the latter phase. It may be deduced that an increased amount of $I$-phase formed in the deposit at the expense of $\text{Mg}_7\text{Zn}_3$.

The EPMA and EDX, Figures 5-35 and 5-36, conducted on the $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ spray deposit, actually show that a relatively high amount of $\text{Zn}$ is dissolved in $\text{Mg}$-rich primary grains. Since the maximum solubility limit of $\text{Zn}$ in $\text{Mg}$ is approximately 2.4 at. % [119, 168], it is likely that the $\alpha$-$\text{Mg}$ matrix is supersaturated with $\text{Zn}$. In fact, the EDX analysis of the primary grain (a) in Figure 5-36 yielded a composition of 96.5 at.% of $\text{Mg}$, 3.4 at.% of $\text{Zn}$ and 0.1 at.% of $\text{Y}$, confirming the $\text{Zn}$ supersaturation. Both EPMA and EDX indicate that the intergranular regions are enriched with $\text{Zn}$ and $\text{Y}$. As in the $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ deposit, the $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposit has both monolithic intergranular regions (b) and eutectic intergranular region (c), Figure 5-36. The monolithic region (b) has a composition of 35.3 at.% of $\text{Mg}$, 54.7 at.% of $\text{Zn}$ and 10.0 at.% of $\text{Y}$ which is close to that of the $I$-phase ($\text{Mg}_3\text{Zn}_6\text{Y}$). Thus, the $I$-phase formed intergranularly. The eutectic region (c) has a composition of 76.7 at.% of $\text{Mg}$, 18.4 at.% of $\text{Zn}$ and 4.9 at.% of $\text{Y}$. This region is considered to have $\alpha$-$\text{Mg}$ matrix, $I$-phase and $W$-phase ($\text{Mg}_3\text{Y}_2\text{Zn}_3$).

The supersaturation of $\text{Zn}$ in the $\alpha$-$\text{Mg}$ primary grains may facilitate precipitation of secondary phases upon secondary thermal processing. The phase(s) that may actually precipitate are not known.
Figure 5-29: Optical micrographs of the bulk ingot of $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ supplied from FUKUDA. (a) Low magnification. (b) High magnification.
Figure 5-30: XRD pattern of \(\text{Mg}_{88}\text{Zn}_{20}\text{Y}_{3.5}\) as-cast ingot. \(\alpha\)-Mg, \(W\)-phase, \(I\)-phase and \(\text{Mg}_7\text{Zn}_3\) are identified. The binary inter metallic compound of \(\text{Mg}_7\text{Zn}_3\) is thought to be a relevant phase for the formation of \(I\)-phase.
Figure 5-31: Break-up image captured from the $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ UDS experiment.

Table 5-6: Experimental Parameters for the $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ Deposits.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orifice Diameter [$\mu$m]</td>
<td>348</td>
</tr>
<tr>
<td>Pressure Differential [psi]</td>
<td>3</td>
</tr>
<tr>
<td>Perturbation Frequency [kHz]</td>
<td>2.39</td>
</tr>
<tr>
<td>Crucible Temp [K]</td>
<td>848</td>
</tr>
<tr>
<td>Alloy [at%]:</td>
<td>$\text{Mg}<em>{76.5}\text{Zn}</em>{20}\text{Y}_{3.5}$</td>
</tr>
<tr>
<td>Chamber Atmosphere:</td>
<td>Argon</td>
</tr>
<tr>
<td>O$_2$ Content [ppm]:</td>
<td>&lt; 1.0</td>
</tr>
<tr>
<td>Flight Distance [m]:</td>
<td>0.25 (to substrate)</td>
</tr>
<tr>
<td>Droplet Diameter [$\mu$m]</td>
<td>700</td>
</tr>
<tr>
<td>Droplet Temp. at Substrate [K]</td>
<td>844</td>
</tr>
</tbody>
</table>
Figure 5-32: $Mg_{76.5}Zn_{20}Y_{3.5}$ deposit produced with 700 μm droplets. The patterned tail which was sprayed next to the main deposit, was deposited to test the motion of the X-Y table.
Figure 5-33: Optical micrographs of the spray deposit of the $Mg_{76.5}Zn_{20}Y_{3.5}$ alloy. (a) Low magnification. (b) High magnification.
Figure 5-34: XRD pattern of $Mg_{76.5}Zn_{20}Y_{3.5}$ deposit, (a) Spray deposit, (b) As-cast ingot.
Figure 5-35: EPMA analysis of the $Mg_{76.5}Zn_{20}Y_{3.5}$ deposit produced with 700 µm droplets. (a) Back-scattered image indicating concentration of the heavier elements Zn and Y in the intergranular regions. (b)-(d) Color mappings of Mg, Zn and Y confirming the enrichment of Zn and Y in the intergranular region and high Mg content in the primary grains.
Figure 5-35: Continued.
Figure 5-36: Back-scattered electron image of the region subjected to EDX analysis. The specimen is from the \( \text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5} \) deposit produced with 700 \( \mu \text{m} \) droplets. a: dark region – the dendritic region. b: white region - the interdendritic region. c: the lamellar eutectic region. The EDX results are listed in Table 5-7.

Table 5-7: EDX Result for the \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_{2} \) Deposit Produced with 700 \( \mu \text{m} \) Droplets.

<table>
<thead>
<tr>
<th></th>
<th>( \text{Mg} ) (at%)</th>
<th>( \text{Zn} ) (at%)</th>
<th>( \text{Y} ) (at%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a (ave.)</td>
<td>96.5</td>
<td>3.4</td>
<td>0.1</td>
</tr>
<tr>
<td>a-1</td>
<td>96.43 ± 0.17</td>
<td>3.54 ± 1.21</td>
<td>0.03 ± 1.30</td>
</tr>
<tr>
<td>a-2</td>
<td>96.39 ± 0.16</td>
<td>3.48 ± 1.14</td>
<td>0.13 ± 1.22</td>
</tr>
<tr>
<td>a-3</td>
<td>96.70 ± 0.16</td>
<td>3.16 ± 1.19</td>
<td>0.14 ± 1.27</td>
</tr>
<tr>
<td>b (ave.)</td>
<td>35.3</td>
<td>54.7</td>
<td>10.0</td>
</tr>
<tr>
<td>b – 1</td>
<td>35.70 ± 0.20</td>
<td>54.39 ± 0.42</td>
<td>9.91 ± 0.43</td>
</tr>
<tr>
<td>b – 2</td>
<td>35.79 ± 0.20</td>
<td>54.22 ± 0.41</td>
<td>9.99 ± 0.43</td>
</tr>
<tr>
<td>b – 3</td>
<td>34.29 ± 0.20</td>
<td>55.50 ± 0.41</td>
<td>10.21 ± 0.43</td>
</tr>
<tr>
<td>c (ave.)</td>
<td>76.7</td>
<td>18.4</td>
<td>4.9</td>
</tr>
<tr>
<td>c-1</td>
<td>76.69 ± 0.14</td>
<td>20.88 ± 0.45</td>
<td>2.43 ± 0.48</td>
</tr>
<tr>
<td>c-2</td>
<td>78.30 ± 0.12</td>
<td>15.41 ± 0.44</td>
<td>6.29 ± 0.44</td>
</tr>
<tr>
<td>c-3</td>
<td>75.12 ± 0.12</td>
<td>18.92 ± 0.40</td>
<td>5.97 ± 0.40</td>
</tr>
</tbody>
</table>

* 3 points are tested for each target area.
5.1.5 Splat collection

Uniform droplets were collected at various flight distances to determine the droplet thermal state and verify simulation results with the experimental results. A droplet quencher, Figure 5-37 (a), equipped with adjustable blade heights and variable rotation speed [41] was employed to collect the droplets at desired flight distances. The droplet quencher was fixed adjacent to the deposition substrate as illustrated schematically in Figure 5-37 (b). The quencher uses small substrate boxes (or blades) to collect droplets at various flight distances.

After adjusting the UDS parameters for stable uniform breakup, the quencher on the X-Y table was brought to the droplet stream to collect splats in the substrate boxes while rotating the quencher at a speed adjusted to prevent splat overlapping.

Figure 5-38 shows a $Mg_7Zn_1Y_2$ splat of a 700 µm droplet collected at a flight distance of 0.25 m in an experiment performed under the same conditions as those used for Experiment 3, Table 5-2. Figures 5-39 and 5-40 show micrographs of the splat in the longitudinal cross section. The cross section shows a fine dendritic microstructure with the dendritic growth direction being largely normal to the chill surface, indicating that the splat was formed by the deposition of a molten droplet on the substrate and that the splat solidified after deformation was completed, by nucleation and dendritic growth normal to the chill surface. The dendrites show very small primary arm spacings of about 1 µm. The above observations confirm the simulated molten states of the depositing droplets at the flight distance of 0.25 m, Figure 5-44.
Figure 5-37: (a): Photograph of the droplet quencher. (b): Schematic of the droplet quenching apparatus attached on the X-Y table.
Figure 5-38: The $Mg_{97}Zn_{1}Y_{2}$ splat collected from the droplet quencher at a flight distance of 0.25 m. (a) The $Mg_{97}Zn_{1}Y_{2}$ splat deformed into a pancake shape. (b) Bottom view of the $Mg_{97}Zn_{1}Y_{2}$ splat. (Chill side). The splat is elongated due to shear caused by the quencher blade motion. (c) Top view of the $Mg_{97}Zn_{1}Y_{2}$ splat.
Figure 5-39: Longitudinal cross section of the $Mg_{97}Zn_{1}Y_{2}$ splat.
Figure 5-40: Longitudinal cross section of the $\text{Mg}_{97}\text{Zn}_1\text{Y}_2$ splat. (a): Edge region (b): Middle region.
Another splat morphology was noted for droplets that did not stay in the vertical droplet stream. They took flight paths that were inclined from the main droplet stream and landed at different places on the substrate, as illustrated in Figure 5-41 (a). Such droplets may cool as “isolated” droplets that are not aerodynamically shielded by their leading droplets during the flight. Thus those isolated droplets were subjected to higher cooling rates than the droplets that are “aligned” in the droplet stream due to the lack of aerodynamic shielding effects. Figures 5-41 (b) and 5-42 show the appearances of one such splat. The splat consists of a nearly semispherical dome and a flange-like periphery, as opposed to the fully flattened pancake-shaped morphology of the splat produced with the droplets in the droplet stream. Figure 5-43 shows micrographs of a cross section of such a splat obtained with an isolated droplet. The cross section exhibits a rapidly solidified microstructure with no indication of crystallization prior to landing on the substrate, Figure 5-43 (a). Close examination reveals fine dendrites in the chill region that grew largely normal to the chill surface, Figure 5-43 (d). In regions away from the chill surface, Figure 5-43 (b), dendrites are more randomly oriented showing little directionality. This indicates that the dendrites, as they grew, were fragmented by the shear caused by the deformation of the droplet. Toward the top region, coarser, equiaxed dendrites are formed, Figure 5-43 (b), which reflects the lower rates of solidification in this region, caused by the rise in solidification front temperature. (See the simulation result in the next section).

Despite the absence of prior solidification in the landing droplet, the splat produced did not flatten as much as the droplets in the main stream (Figure 5-38). This
suggests that the rate of solidification (i.e., nucleation and growth) in this splat was higher than that in the splat of a main-stream droplet. This is supported by the observation that the dendrites in this splat are largely equiaxed, except in the chill zone. This would occur when solidification proceeds while the droplet is still deforming. In contrast, the splat in Figure 5-38, produced with a main stream droplet, is fully flattened and has directional dendrites that are largely normal to chill surface. This would occur when dendrites grew in an already deformed (flattened) droplet.

The different solidification rates considered for the fat and flat splats reflect the different temperatures of the isolated and main stream droplets. Figure 5-45 shows simulated cooling curves for 700 µm droplets in isolated and aligned conditions. Although both cooling conditions would give a superheated droplet, the isolated droplet is at a much lower temperature at the substrate. This would make the solidification rate more comparable to the deformation rate.

Figure 5-44 shows a cross section of another splat produced with an isolated droplet that has a similar dome morphology although the microstructure of this splat is similar to the one in Figure 5-43, coarser dendrites are formed in a small region, indicating that this particular droplet had nucleated, probably just before landing on the substrate. Such a condition might occur in a smaller droplet, but no information is available on the size of the droplet that produced this splat.
Figure 5-41: (a): Illustration of scattered droplets that took flight paths that inclined from the main droplet stream and landed at different places on the substrate (b): $Mg_{97}Zn_1Y_2$ droplet that landed outside the target position.
Figure 5-42: The $Mg_{97}Zn_1Y_2$ splat in Figure 5-41. (a): Side view. (b): Top view. (c): Bottom view.
Figure 5-43: Cross sectional micrograph of the $Mg_{97}Zn_Y$ splat in Figures 5-41 and 5-42. (a): Entire cross section of the splat. (b): Medium magnification image for the entire section. (c): High magnification image taken from near the top region. (d): High magnification image taken at the near chill region.
Figure 5-43: Continued.
Figure 5-43: Continued.
Figure 5-44: Temperature vs. flight distance of “aligned” and “isolated” 700 μm $Mg_7Zn_3Y_2$ droplets.
Figure 5-45: Cross section of a $Mg_{97}Zn_{1}Y_{2}$ splat obtained near the spray deposit. (a): Low magnification. (b): High magnification.
5.1.5.1 Overview of numerical simulation of droplet deformation

The splat deformation of solidification has been modeled by Wang and Sun, project collaborators at the University of Massachusetts Lowell [169]. The simulation model is built on a Level Set Method to track the splat boundary, while adopting a free dendritic growth model to address the attendant rapid solidification in the deforming droplet. Figures 5-46 and 5-47 show the instantaneous splat shape with velocity and temperature contours and solidification front, calculated at t = 0.03 ms and t = 0.10 ms for a 693 µm Mg-Zn-Y droplet impinging on the substrate under conditions similar to those for the splat in Figure 5-43. The velocity contours strongly suggest that the dendrites in the deformed splat would be sheared into smaller pieces which then would grow as equiaxed dendrites as seen in Figure 5-43.
Figure 5-46: Simulated velocity vectors and temperature contours of a 693 µm \textit{Mg-Zn-Y} droplet quenched on a substrate [169].
Figure 5-47: Simulated temperature contour and solid-liquid interface position during rapid solidification of a 693 µm Mg-Zn-Y droplet quenched on a substrate [169].
5.2 Rolling Tests

The Mg-Zn-Y spray deposits were subjected to thermomechanical processing by rolling in order to further refine the microstructure. To minimize the temperature drop upon contact with the rolls, the specimens were sandwiched between two 1 mm-thick stainless steel sheets and preheated to the rolling temperature. Rolling conditions are listed in Table 5-8. A specimen 20.8 mm by 8.38 mm by 5.2 mm, cut from the \( \text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2} \) deposit, was rolled under Condition A, for a cumulative thickness reduction from 5.2 mm to 3.15 mm, or about 40\%, in four passes (Figure 5-48). The specimen temperature might have slightly dropped, perhaps by a few degrees, as it passed through the roll gap, but no measurements were made. Despite the relatively small thickness reductions, edge cracking initiated in the third pass and fracture resulted in the final pass, making further reductions impractical under Condition A. Condition B was applied to \( \text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}, \text{Mg}_{88}\text{Zn}_{10}\text{Y}_{2} \) and \( \text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5} \) to roll them without cracking. Figure 5-49 shows a \( \text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2} \) specimen rolled under Condition B for a cumulative reduction of 33 \% in 11-passes. Improved rollability is apparent. The \( \text{Mg}_{88}\text{Zn}_{10}\text{Y}_{2} \) and \( \text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5} \) specimens also rolled with no cracking up to reductions of 25 \% and 15 \%, respectively.

The difference in rollability between Conditions A and B is understood in terms of the deformation mechanisms involved. Figure 5-50 shows the microstructure of the rolled \( \text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2} \) specimen at different magnifications. It is seen that the striations in the grains, which were initially straight, now exhibit zig-zag patterns. This happened as the grains deformed mainly by mechanical twinning. At the low rolling temperature (473 K), slip is restricted to the basal plane, making extensive deformation entirely by
slip difficult. In addition, the precipitates of $X$-phase, presumed to have formed in the basal plane, would make basal slip difficult. At the higher rolling temperature (~673 K) of Condition B, non-basal slip was probably activated allowing the material to deform primarily by slip. Figure 5-51 shows the microstructure of the rolled $Mg_{97}Zn_{1}Y_{2}$ specimen. The striations are more wavy than zigzagged, indicating the occurrence of slip in multiple slip systems. Some of the grains in Figure 5-51 show evidence of mechanical twinning, although it is considered secondary to slip in this specimen rolled under Condition B.
Table 5-8: Rolling Test Condition and Results.

<table>
<thead>
<tr>
<th>Heat treatment Condition</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Temperature</strong></td>
</tr>
</tbody>
</table>
| A | 473 K | 20 min | 10 % | \(Mg_{97}Zn_{1}Y_{2}\) spray deposit: Edge cracking initiated at 30 %. Completely fractured at 40 %.
| B | 673 K | 20 min | 3 % | \(Mg_{97}Zn_{1}Y_{2}\) spray deposit: Achieved reduction ratio: 33 %
|   |       |       |     | \(Mg_{88}Zn_{10}Y_{2}\) spray deposit: Achieved reduction ratio: 25 %
|   |       |       |     | \(Mg_{76.5}Zn_{20}Y_{3.5}\) spray deposit: Achieved reduction ratio: 15 % |
Figure 5-48: $Mg_{97}Zn_1Y_2$ deposit rolled under the condition A listed in Table 5-8.

Figure 5-49: $Mg_{97}Zn_1Y_2$ deposit rolled under the condition B listed in Table 5-8. (a): Before rolling (b): After rolling.
Figure 5-50: Optical micrographs of rolled $Mg_{97}Zn_{1}Y_{2}$ deposit preheated to 473 K for 20 min prior to rolling. (a): Low magnification. (b): Medium magnification. (c): High magnification.
Figure 5-51: Optical micrographs of the rolled $Mg_{97}Zn_1Y_2$ deposit preheated to 673 K for 20 min prior to rolling. (a): Low magnification. (b): High magnification.
Figure 5-52 shows the optical micrographs of a longitudinal cross section of the rolled $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ deposit. The equiaxed grains produced by UDS deposition, Figure 5-25, are well preserved although they are somewhat elongated in the rolling direction. The network structure of intergranular constituents, identified by EPMA and EDX as either monolithic $I$-phase particles or a eutectic involving $\alpha$-$\text{Mg}$, and compounds, probably of $I$- and $W$-phases, is also well preserved. The $I$- and $W$-phase particles are, however, broken up into finer, more spheroidal particles, although these particles largely remained in the network structure.

Figure 5-53 shows optical micrographs obtained of a longitudinal cross section of the rolled $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposit. The equiaxed dendrites produced during the UDS deposition, Figure 5-33, are broken up into more rounded grains, although, they are still more irregularly shaped than the equiaxed grains in the rolled $\text{Mg}_{97}\text{Zn}_1\text{Y}_2$ deposit and $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ deposit which are well-rounded.

The irregular grain morphology is caused in part by the presence of particles of $I$- and $W$-phase particles that had formed in the intergranular liquid, Figure 5-33. These particles were also broken up into smaller pieces, but still remain largely in the prior intergranular regions.
Figure 5-52: Optical micrographs of rolled $Mg_{88}Zn_{10}Y_2$ deposit preheated to 673 K for 20 min prior to rolling. (a): Low magnification. (b): High magnification.
Figure 5-53: Optical micrographs of rolled $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposit preheated to 673 K for 20 min prior to rolling. (a): Low magnification. (b): High magnification.
The occurrence of $X$-phase in $Mg_{97}Zn_{1}Y_2$ deposit and $I$- and $W$-phases in the $Mg_{88}Zn_{10}Y_2$ and $Mg_{76.5}Zn_{20}Y_{3.5}$ deposits is verified in Figures 5-54 ~ 5-56, which show XRD patterns of the three deposits in the as-sprayed, rolled and rolled and annealed states. The annealing after rolling was done at 673 K for 30 minutes to investigate the possible structural changes by recovery.

In all three alloys, the phases that make up these structures, i.e., $\alpha$-Mg and $X$-phase in $Mg_{97}Zn_{1}Y_2$ and $\alpha$-Mg, $I$-phase and $W$-phase in both $Mg_{88}Zn_{10}Y_2$ and $Mg_{76.5}Zn_{20}Y_{3.5}$, remained unchanged. For $Mg_{97}Zn_{1}Y_2$, the intensities of the $X$-peaks, Figure 5-54, appear to have decreased in the rolled and rolled and annealed specimens. For both $Mg_{88}Zn_{10}Y_2$ and $Mg_{76.5}Zn_{20}Y_{3.5}$, the rolled and rolled and annealed specimens have lower peak intensities for $W$-phase relative to those of $I$-phase. This implies that some of the $W$-phase formed during spray deposition transformed to $I$-phase.
Figure 5-54: XRD pattern of the $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 700 µm droplets. (a) Annealed specimen. (b) As-rolled specimen. (c) As-sprayed specimen.
Figure 5-55: XRD pattern of the $Mg_{88}Zn_{10}Y_2$ deposit produced with 700 µm droplets. (a): Annealed specimen. (b): As-rolled specimen. (c): As-sprayed specimen.
Figure 5-56: XRD pattern of the $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposit produced with 700 µm droplets. (a): Annealed specimen. (b): As-rolled specimen. (c): As-sprayed specimen.
The hardness of the as-sprayed, rolled and rolled and annealed specimens of the $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ deposits were measured to investigate the effect of thermo-mechanical processing. The hardness measurements were performed with a Vickers micro hardness tester ($\text{SHIMADZU MICRO HARDNESS TESTER: HMV-2T}$) with 98.07 mN (0.01 kgf) loading. The results of the hardness test are summarized in Table 5-9. Each hardness value represents the average of 10 measurements. All of the as-sprayed specimens of the $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ deposits have higher hardness values than those of as-cast ingot, due probably to the finer grain sizes in the deposits. The low solute concentrations in the primary grains, Table 5-3, suggest that other RSP effects, such as increased solid solution strengthening, did not play major roles. The as-sprayed $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ deposit produced with 1000 µm droplets (Experiment 2) has a slightly higher hardness than the one produced with 700 µm droplets (Experiment 1), while the deposit produced with 700 µm droplets (Experiment 3) provides the highest hardness value (110 HV). The highest value of hardness is considered to have resulted due primarily to the grain refinement achieved in $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ deposit produced in Experiment 3 which was performed at a lower initial temperature of 973 K, Figure 5-12. Figure 5-57 shows the indentations made on the as-sprayed specimens of $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ from Experiments 2 and 3.

The rolled $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ deposit from Experiment 3 has even higher hardness (117 HV), probably owing to strain hardening. The rolled $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ deposit, when further annealed at 673 K for 30 minutes, had only slightly reduced hardness of 114 HV. Thus, the increase in hardness upon rolling was not entirely lost by the annealing process. This implies that recovery is sluggish in this alloy and/or possible age hardening offsets
softening caused by recovery. The retention of high hardness after annealing may lead to improved strength in elevated-temperature service.

The hardness values were also measured for the $Mg_{88}Zn_{10}Y_2$ and $Mg_{76.5}Zn_{20}Y_{3.5}$ deposits in the as-deposited and rolled states. These deposits have higher hardness than the $Mg_{97}Zn_1Y_2$ deposits, due primarily to their higher concentration of Zn and Y that produce larger amounts of brittle compounds ($I$ and $W$ phases) in the microstructure. The hardness values of the as-cast ingots of these two alloys are scattered widely, reflecting coarse dendritic grains. Low hardness values are measured for the large $\alpha$-$Mg$ dendrites, while the secondary phases give high hardness values. The as-sprayed specimen and the rolled specimens of both $Mg_{88}Zn_{10}Y_2$ and the $Mg_{76.5}Zn_{20}Y_{3.5}$ have higher and less scattered hardness values than those of the ingots, reflecting the homogenizing effects, as well as the effects of strain hardening and possible dynamic age hardening, of the rolled operations.

Figure 5-58 compares the Vickers hardness values of the as-sprayed $Mg_{97}Zn_1Y_2$, $Mg_{88}Zn_{10}Y_2$ and $Mg_{76.5}Zn_{20}Y_{3.5}$ deposits produced with 700 $\mu$m droplets to those of conventional $Mg$ alloys. The as-sprayed deposits clearly have higher hardness. Although most of the hardness values of conventional alloys are determined on ingots (which intrinsically have larger grains than those in spray deposits and hence are softer), it is also noted that the $Mg$-$Zn$-$Y$ deposits have higher hardness than those of the wrought (extruded, sheet) alloys. The higher hardness of the $Mg$-$Zn$-$Y$ deposits arises from the yttrium in these alloys that form useful strengthening phases with $Mg$ and $Zn$, as well as the grain refinement effects of UDS deposition.
Table 5-9: Vickers Microhardness Results of the different Mg-Zn-Y Alloys.

<table>
<thead>
<tr>
<th>Alloy</th>
<th>Condition</th>
<th>Droplet Dia. [µm]</th>
<th>Hardness [HV 0.01] (ave.)</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_{97}Zn_{1}Y_2$</td>
<td>As-Cast</td>
<td>n/a</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td></td>
<td>As-Sprayed</td>
<td>700</td>
<td>85</td>
<td>Exp. #1</td>
</tr>
<tr>
<td></td>
<td>As-Rolled</td>
<td>700</td>
<td>109</td>
<td>Exp. #1</td>
</tr>
<tr>
<td></td>
<td>As-Sprayed</td>
<td>1000</td>
<td>86</td>
<td>Exp. #2</td>
</tr>
<tr>
<td></td>
<td>As-Sprayed</td>
<td>700</td>
<td>110</td>
<td>Exp. #3</td>
</tr>
<tr>
<td></td>
<td>As-Rolled</td>
<td>700</td>
<td>117</td>
<td>Exp. #3</td>
</tr>
<tr>
<td></td>
<td>Rolled and Annealed</td>
<td>700</td>
<td>114</td>
<td>Exp. #3</td>
</tr>
<tr>
<td>$M_{88}Zn_{10}Y_2$</td>
<td>As-Cast</td>
<td>n/a</td>
<td>(70-280)*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>As-Sprayed</td>
<td>700</td>
<td>123</td>
<td></td>
</tr>
<tr>
<td></td>
<td>As-Rolled</td>
<td>700</td>
<td>125</td>
<td></td>
</tr>
<tr>
<td>$M_{76.5}Zn_{20}Y_{3.5}$</td>
<td>As-Cast</td>
<td>n/a</td>
<td>(70-280)*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>As-Sprayed</td>
<td>700</td>
<td>147</td>
<td></td>
</tr>
<tr>
<td></td>
<td>As-Rolled</td>
<td>700</td>
<td>150</td>
<td></td>
</tr>
</tbody>
</table>

* Since these as-cast ingots contain coarse microstructures, the obtained values are scattered.
Figure 5-57: Optical micrographs of the specimens used for Vickers hardness measurement. (a): As-sprayed $M_{97}Zn_1Y_2$ deposit produced with 1000 µm droplets. (b): As-sprayed $M_{97}Zn_1Y_2$ deposit produced with 700 µm droplets.
Figure 5-58: Vickers hardness values of the as-sprayed deposits of $Mg_{97}Zn_{1}Y_{2}$, $Mg_{88}Zn_{10}Y_{2}$ and $Mg_{76.5}Zn_{20}Y_{3.5}$ alloy produced with 700 µm droplets in comparison with those of conventional $Mg$ alloys. (Vickers hardness values of ASTM $Mg$ alloys are estimated from Brinell hardness values.)
5.2 Tensile Tests

5.2.1 Tensile strength testing procedures

The rolled Mg-Zn-Y deposits were machined into flat tensile specimens 1.1 mm in thickness, 3.2 mm in width and 12.7 mm in gauge length, as illustrated in Figure 5-59, for a room temperature tensile test. Figure 5-60 gives the appearance of the machined tensile specimens. The tensile tests were performed on an Instron machine (Instron-3366), Figure 5-61, with a constant crosshead speed of 0.08 mm/min which translates into an initial strain rate of $10^{-4}$ s$^{-1}$.

Elevated-temperature tensile tests were conducted at FUKUDA Metal Foil & Powder Co., Ltd., Kyoto, Japan with their assistance. An Instron-5864 testing machine equipped with a heating furnace, Figure 5-62, was used for the elevated temperature tensile test. Figure 5-63 shows the clamping grips inside of the heating furnace. Tensile specimens for the elevated temperature tests were prepared by initially hot rolling the spray deposits at 673 K followed by annealing at 673 K for 30 minutes. The rolled and annealed specimens were then machined into simple rectangular plate with dimensions of 1.0 mm in thickness, 5.0 mm in width and 5.0 mm in gauge length, Figure 5-64. The surfaces of the specimens were final-polished with 1500 grit emery paper in order to remove any surface defect. Prior to testing, the furnace was heated to 473 K over 20 minutes with additional 5 minute holding time. The tensile tests were performed at 473 K with a constant crosshead speed of 0.3 mm/min, which corresponds to the initial strain rate of $10^{-3}$ s$^{-1}$.  

172
Figure 5-59: Drawing showing the dimensions of the tensile test specimen.

Thickness: 1.1 mm

Figure 5-60: Photograph of tensile test specimens machined from $Mg_9Zn_1Y_2$ spray deposits.
Figure 5-61: Instron tensile machine for room temperature testing (*INSTRON-3366*).
Figure 5-62: Instron tensile machine for elevated temperature testing (*INSTRON-5864*).
Figure 5-63: The clamping grips of the tensile test machine.

Thickness: 1 mm

Figure 5-64: Dimensions of the tensile test specimen for elevated temperature tensile tests.
5.2.2  Tensile test results

The room-temperature tensile tests showed that the $Mg_{97}Zn_{1}Y_{2}$ spray deposit, produced with 700 $\mu$m droplets (Experiment 3) and subjected to thermo-mechanical processing, had superior properties compared to those of typical commercial wrought $Mg$ alloys. Figure 5-65 shows the tensile specimens of the $Mg_{97}Zn_{1}Y_{2}$ spray deposits, after the room temperature tensile tests. Figure 5-66 shows the load-displacement curves of the $Mg_{97}Zn_{1}Y_{2}$ deposit produced with 700 $\mu$m droplets (Experiment 3) tested in the as-sprayed, as-rolled and rolled and annealed states. The 0.2 % proof yield strength and the ultimate tensile strength calculated from the results reached 320 MPa and 340 MPa, respectively for the as-rolled $Mg_{97}Zn_{1}Y_{2}$ spray deposit.

Figure 5-67 shows the load-displacement curves of the as-rolled specimens of the $Mg_{88}Zn_{10}Y_{2}$ and $Mg_{76.5}Zn_{20}Y_{3.5}$ spray deposits. The load-displacement curves indicate that both of the specimens fractured brittlely with no significant plastic deformation. In both cases the measured UTS (defined at fracture point) was significantly lower than that of the $Mg_{97}Zn_{1}Y_{2}$ specimens, which is attributed to the higher $Y$ and $Zn$ concentrations in the two alloys that produced a larger amounts of brittle phases, i.e., $I$ and $W$.

The values of Young’s modulus, calculated from the load displacement curves of the room-temperature tensile tests are found to be lower than the values expected for magnesium alloys. This is most likely caused by the local deformation of the specimen at the grips that occurred during the tensile tests as seen in Figure 5-68. The local deformation at the grips causes larger extension/strain values than the actual value. By measuring the amount of extension caused by the local deformation at the grips to be
approximately 70 µm, the corrected Young’s modulus can be calculated to be 45 GPa, which is more in line with the values of most commercial Mg alloys, including Mg$_{97}$Zn$_1$Y$_2$[76].

Table 5-10, summarizing the tensile data, indicates that as the Y and Zn concentration increases the elastic modulus slightly increases, while the UTS and the yield strength decrease. Both the Mg$_{88}$Zn$_{10}$Y$_2$ and Mg$_{76.5}$Zn$_{20}$Y$_{3.5}$ spray deposits exhibit brittle fracture, Figure 5-67, while Mg$_{97}$Zn$_1$Y$_2$ is comparatively more ductile. Thus, the secondary phases, i.e., I and W, found in the Mg$_{88}$Zn$_{10}$Y$_2$ and Mg$_{76.5}$Zn$_{20}$Y$_{3.5}$ deposits were not very effective in strengthening the spray-deposited alloys, owing to their relatively coarse particle sizes and intergranular locations, as seen in the micrographs in Figures 5-25 and 5-33.

Figure 5-69 compares the tensile test data of the rolled Mg-Zn-Y deposits produced with 700 µm droplets and with those of commercial Mg alloys. Despite the presence of intergranular constituents, the Mg$_{97}$Zn$_1$Y$_2$, Mg$_{88}$Zn$_{10}$Y$_2$ and Mg$_{76.5}$Zn$_{20}$Y$_{3.5}$ deposits, after rolling, have room-temperature UTS and yield strength values superior to those of any of the commercial wrought alloys. They are also stronger than high-performance casting alloys such as ZK51-T5 and ZK61-T5 that contain Zn or Y as the primary alloying element. Even severe work hardening achieved by extrusion and forging, as in AZ61-F and AZ80-T6, does not produce room-temperature strength as high as that of the UDS processed and rolled Mg-Zn-Y alloys.

Figures 5-70 to 5-72 show the fracture surface and the longitudinal cross sections near the fracture surface of the of the Mg$_{97}$Zn$_1$Y$_2$ tensile specimens tested at room-temperature in the as-sprayed, as-rolled and rolled and annealed conditions, respectively. The fracture surface of the as-sprayed specimen basically follows the
intergranular region delineated by the $X$-phase network. The fracture surface also shows some evidence of plastic shear, reflecting the larger elongation observed on the as-sprayed specimen. Both the as-rolled and rolled and annealed specimens also exhibit a fracture path that essentially matches that of the intergranular $X$-phase network, indicating that rolling did not effect major alterations of the network morphology of the intergranular phases. However, since rolling did increase the hardness of the $\alpha$-$Mg$ grains by strain hardening, and subsequent anneal might have caused age hardening of the $\alpha$-$Mg$ grains, the rolled specimens, with or without subsequent annealing, showed higher strength, but failed without exhibiting appreciable elongations.

Figures 5-73 and 5-74 show the fracture surfaces and the longitudinal cross sections of the as-rolled deposits, $Mg_{88}\text{Zn}_{10}\text{Y}_2$ and $Mg_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$, both of which were produced with 700 $\mu$m droplets. The fracture surfaces of these specimens also largely followed the intergranular constituent network which persisted even after the rolling and annealing.

Figures 5-75 to 5-77 show the load vs. displacement curves obtained from the tensile test conducted with the rolled and annealed $Mg_{97}\text{Zn}_{1}\text{Y}_2$, $Mg_{87}\text{Zn}_{10}\text{Y}_2$ and $Mg_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposits. The results of the elevated temperature tensile tests are listed in Table 5-11.

The UTS and 0.2 % offset yield strength of the $Mg_{97}\text{Zn}_{1}\text{Y}_2$ at 473 K determined on the load-displacement curve, Figure 5-75, are about 271 MPa and 195 MPa, respectively. These values significantly exceed the literature values of as-cast and hot-pressed $Mg_{97}\text{Zn}_{1}\text{Y}_2$ which are typically 107 – 220 MPa (UTS) and 75-171 MPa (0.2 proof) [170, 171]. While having the high strength, the $Mg_{97}\text{Zn}_{1}\text{Y}_2$ specimen was ductile, as seen in Figure 5-78. This ductility arises from the $\alpha$-$Mg$-$X$ phase microstructure.
where both phases are ductile at 473 K, (see Figure 5-79). The rolled and annealed specimen of $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ fractured with no elongation at 237 MPa, and the rolled and annealed $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ specimen also fractured brittlely at 216 MPa, Figures 5-76 and 5-77, and Table 5-11.

The lack of ductility noted for the latter two specimens, tested at 473 K, is considered to be caused by the $I$ and $W$ phases present in the intergranular regions. Figures 5-80 and 5-81 show that the fracture in these specimens indeed occurred along the intergranular regions. Unlike the $X$-phase in the $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_2$ specimen, the $I$ and $W$ particles in the $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ and $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ specimens did not flow at all.

Figure 5-82 compares the UTS of the rolled and annealed $\text{Mg-Zn-Y}$ deposits produced with 700 $\mu$m droplets with those of commercial $\text{Mg}$ alloys [170, 171]. The $\text{Mg}_{97}\text{Zn}_1\text{Y}_2$, $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ and $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposits do have superior elevated-temperature tensile strength to those of the commercial alloys. The brittle failure of the $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ and $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ alloys may be reduced, or even eliminated, by improving conditions for both spray deposition and subsequent thermomechanical processing. The $\text{Mg}_{97}\text{Zn}_1\text{Y}_2$ alloy, produced in the manner shown in the present work, applies to elevated-temperature service much more readily. Further optimization of process conditions would make such elevated-temperature applications of this alloy even more practical.
Figure 5-65: Photograph showing the fractured tensile test specimens produced from the as-rolled $Mg_{97}Zn_1Y_2$ deposits (700 µm droplets).

Figure 5-66: Load vs. displacement curves at room temperature of the as-sprayed, rolled and annealed $Mg_{97}Zn_1Y_2$ specimens.
Figure 5-67: Load vs. displacement curves at the room temperature of the as-sprayed and as-rolled specimens of (a) $Mg_{88}Zn_{10}Y_2$ deposit, and (b) $Mg_{76.5}Zn_{20}Y_{3.5}$ deposit.
Figure 5-68: Micrograph of the tensile specimen grip area, (a) showing deformation at the tensile specimen grip. (b) An approximately 70 μm slippage on each grip occurred due to the deformation.
Table 5-10: Mg-Zn-Y Spray Deposit Room Temperature Tensile Test Results.

<table>
<thead>
<tr>
<th>Alloy (^*1)</th>
<th>Condition</th>
<th>Young’s Modulus (GPa) (^*2)</th>
<th>Yield Strength, (\sigma_{0.2}) [MPa]</th>
<th>UTS (MPa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Mg_{97}Zn_{1}Y_{2})</td>
<td>As-Sprayed</td>
<td>45</td>
<td>144</td>
<td>233</td>
</tr>
<tr>
<td></td>
<td>As-Rolled</td>
<td>45</td>
<td>320</td>
<td>340</td>
</tr>
<tr>
<td></td>
<td>Rolled and annealed</td>
<td>45</td>
<td>261</td>
<td>301</td>
</tr>
<tr>
<td>(Mg_{88}Zn_{10}Y_{2})</td>
<td>As-Rolled</td>
<td>47</td>
<td>298</td>
<td>305</td>
</tr>
<tr>
<td>(Mg_{76.5}Zn_{20}Y_{3.5})</td>
<td>As-Rolled</td>
<td>51</td>
<td>256</td>
<td>271</td>
</tr>
</tbody>
</table>

\(^*1\): All deposits tested for tensile tests were produced with 700 μm droplets. \(Mg_{97}Zn_{1}Y_{2}\) deposits were produced under the condition of Experiment 3.

\(^*2\): These Young’s moduli were obtained after correction to accommodate slippage.
Figure 5-69: Comparison of the room-temperature tensile properties of rolled $\text{Mg}_{97}\text{Zn}_1\text{Y}_2$, $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$, and $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ with those of the commercial wrought and cast Mg alloys [3, 170]. (a): UTS. (b): 0.2 % Proof stress. Rolled $\text{Mg}_{97}\text{Zn}_1\text{Y}_2$ spray deposit shows superior UTS value than those of wrought Mg alloys.
Figure 5-69: Continued.
Figure 5-70: Micrographs of the fracture surface of the as-sprayed $Mg_{97}Zn_{1}Y_{2}$ deposit after room temperature tensile test, (a) Low magnification SEM image showing the edge of the specimen, (b) Optical micrograph of the longitudinal cross section.
Figure 5-71: Micrographs of the fracture surface of the as-rolled $\text{Mg}_{97}\text{Zn}_{1}\text{Y}_{2}$ deposit after room temperature tensile test, (a) Low magnification SEM image, (b) Optical micrograph of the longitudinal cross section.
Figure 5-72: Micrographs of the fracture surface of the rolled and annealed $Mg_{97}Zn_1Y_2$ deposit after room temperature tensile test, (a) Low magnification SEM image, (b) Optical micrograph of the longitudinal cross section.
Figure 5-73: Micrographs of the fracture surface of the as-rolled $Mg_{88}Zn_{10}Y_2$ deposit after room temperature tensile test, (a) Low magnification SEM image, (b) Optical micrograph of the longitudinal cross section showing cracks propagating along the intergranular regions.
Figure 5-74: Micrographs of the fracture surface of the as-rolled $Mg_{76.5}Zn_{20}Y_{3.5}$ deposit after room temperature tensile test, (a) Low magnification SEM image, (b) Optical micrograph of the longitudinal cross section showing cracks forming in the intergranular regions.
Figure 5-75: Load vs. displacement curve of rolled and annealed \(Mg_{97}Zn_{1}Y_2\) deposit tested at 473 K.

Figure 5-76: Load vs. displacement curve of rolled and annealed \(Mg_{87}Zn_{10}Y_2\) deposit tested at 473 K.
Figure 5-77: Load vs. displacement curve of rolled and annealed $Mg_{76.5}Zn_{20}Y_{3.5}$ deposit tested at 473 K
Table 5-11: Elevated Temperature Tensile Test Result of \textit{Mg-Zn-Y} Spray Deposits

<table>
<thead>
<tr>
<th>Alloy</th>
<th>Condition</th>
<th>Yield Strength, $\sigma_{0.2}$ (MPa)</th>
<th>UTS (MPa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mg$_{97}$Zn$_1$Y$_2$</td>
<td>Rolled and annealed</td>
<td>195</td>
<td>271</td>
</tr>
<tr>
<td>Mg$<em>{88}$Zn$</em>{10}$Y$_2$</td>
<td>Rolled and annealed</td>
<td>152</td>
<td>237</td>
</tr>
<tr>
<td>Mg$<em>{76.5}$Zn$</em>{20}$Y$_{3.5}$</td>
<td>Rolled and annealed</td>
<td>211</td>
<td>216</td>
</tr>
</tbody>
</table>

The Young’s moduli are not listed in this table since slippages on the tensile specimen grip caused significant amount of errors to obtain proper strain values.

Figure 5-78: Tensile specimen of the rolled and annealed Mg$_{97}$Zn$_1$Y$_2$ spray deposit after the tensile test at 473 K.
Figure 5-79: Optical micrograph of the longitudinal cross section of the fracture area of the rolled and annealed $Mg_{97}Zn_{1}Y_2$ deposit after tensile testing at 473 K, showing the elongated grains.

Figure 5-80: Optical micrograph of the longitudinal cross section of the fracture area of the rolled and annealed $Mg_{88}Zn_{10}Y_2$ deposit after elevated temperature tensile testing at 473 K.
Figure 5-81: Optical micrograph of the longitudinal cross section of the fracture area of the rolled and annealed $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposit after tensile testing at 473 K, showing cracks forming in intergranular region.
Figure 5-82: Comparison of the elevated-temperature (473 K) tensile properties of rolled and annealed $Mg_{97}Zn_{1}Y_{2}$, $Mg_{88}Zn_{10}Y_{2}$, and $Mg_{76.5}Zn_{20}Y_{3.5}$ with those of the commercial Mg alloys [170-172]. (The UTS values of $AZ91D$, $AE42$, $AS21$, $WE43-T6$ and $ZE41-T5$ alloys are extrapolated values from a literature data available within a range between 293 K – 423 K (20 °C – 180 °C) [171].) Rolled and annealed $Mg_{97}Zn_{1}Y_{2}$, $Mg_{88}Zn_{10}Y_{2}$, and $Mg_{76.5}Zn_{20}Y_{3.5}$ spray deposits show superior UTS values than those of the commercial Mg alloys.
6. CONCLUSIONS

- A safety-proven rapid solidification technique based on mono-size droplet deposition of magnesium alloys was established in this study and applied to $\text{Mg}_{97}\text{Zn}_1\text{Y}_2$, $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ and $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ alloys. Mono-size droplets permit droplet deposition without involving fine, pyrophoric droplets, yet with sufficient rapid solidification effects. The use of mono-size droplet sprays also permits full-density spray deposition, in contrast to conventional spray forming in which porosity inherently forms in the deposited material.

- Full-density sprayed deposits of $\text{Mg}_{97}\text{Zn}_1\text{Y}_2$, $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ and $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ were produced with superheated (molten) 700 $\mu$m or 1000 $\mu$m droplets. Deposition of a uniform spray, consisting only of molten droplets with identical thermal history, produces identical local deposition conditions for all droplets to deform into fully flattened splats that stack up and fill space, while rapidly solidifying into a fully dense, metallurgically integrated spray deposit.

- Process parameters were adjusted to obtain deposition conditions with molten droplets by running simulations using a droplet in-flight solidification model that was developed over the course of previous studies and further modified in the present study for the $\alpha$-$\text{Mg}$ - $I$ phase pseudo-binary system.

- The sprayed deposits of $\text{Mg}_{97}\text{Zn}_1\text{Y}_2$, $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ and $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ exhibited an equiaxed microstructure that consisted of primary $\alpha$-$\text{Mg}$ grains and secondary phases formed in the intergranular regions and/or within the $\alpha$-$\text{Mg}$ grains. The
primary grains 10 - 200 µm in size, depending on the droplet size and initial superheat used, local deposition conditions and spray stability, are well rounded in the Mg$_{97}$Zn$_1$Y$_2$ and Mg$_{88}$Zn$_{10}$Y$_2$ deposits, but the Mg$_{76.5}$Zn$_{20}$Y$_{3.5}$ deposit exhibited primary grains that preserved some traits of dendritic morphology. The secondary phases were identified by XRD as the X-phase (Mg$_{12}$ZnY) in the Mg$_{97}$Zn$_1$Y$_2$ deposit, whereas the Mg$_{88}$Zn$_{10}$Y$_2$ and Mg$_{76.5}$Zn$_{20}$Y$_{3.5}$ deposits had I-phase (Mg$_3$Zn$_6$Y$_1$) and W-phase (Mg$_5$Zn$_3$Y$_2$). EPMA further verified that these secondary phases are associated mainly with the intergranular regions (which correspond to the interdendritic regions in conventionally cast alloys.) Mg$_7$Zn$_3$, which was identified in a cast Mg$_{76.5}$Zn$_{20}$Y$_{3.5}$ ingot, was absent in the Mg$_{76.5}$Zn$_{20}$Y$_{3.5}$ deposit, indicating that an increased amount of I-phase formed in the spray deposit at the expense of Mg$_7$Zn$_3$.

- The Mg$_{97}$Zn$_1$Y$_2$ deposit produced with 700 µm droplets had precipitates (X-phase) also on a habit plane within the primary grains, seen as striations in the primary grains. The striae extended across the grain diameter but only in one direction in each grain, indicating that the ‘habit plane’ was the basal plane. The striae, under the condition that the primary grains grew and impinged upon each other, created serrated grain boundaries, which would suppress grain boundary sliding and hence improve creep resistance.

- The direct impingement of striated grains upon each other implies the occurrence of a couple-zone growth that alleviates the need for solute accumulation at the solidification front. The striae indicative of X-phase precipitation were noted throughout the Mg$_{97}$Zn$_1$Y$_2$ deposits, although direct impingement of striated grains
and the resultant serrated grain boundaries were seen only in the chill region. Other locations in the deposits exhibited intergranular constituents consisting of $\alpha$-Mg and X-phase and no serrated grain boundaries. The $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ and $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposits had no primary grains that exhibited striations or impinge directly upon each other.

- Rolling the $\text{Mg}_{97}\text{Zn}_1\text{Y}_2$ deposit after preheating at 473 K resulted in edge cracking as cumulative thickness reductions approached 30%. This was caused by the occurrence of mechanical twinning which was the primary deformation mechanism at 473 K in this deposit. At 673 K, the $\text{Mg}_{97}\text{Zn}_1\text{Y}_2$ deposit rolled without major defects in excess of 30%, indicating the occurrence of non-basal slip. Further reductions appeared possible though not confirmed. Rolling $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ and $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposits at 673 K, after preheating at 673 K, was less forgiving; the deposits rolled to 25% and 15%, respectively, due primarily to increased amounts of intergranular intermetallic constituents.

- As-rolled deposits and rolled deposits subjected to annealing at 673 K had the same phases as those that were found in the as-sprayed state, i.e., $\alpha$-Mg and X in $\text{Mg}_{97}\text{Zn}_1\text{Y}_2$ and $\alpha$-Mg, I and W in $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ and $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$, reflecting the stability of these phases, particularly that of the quasicrystalline I-phase.

- The rolled $\text{Mg}_{97}\text{Zn}_1\text{Y}_2$, $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ and $\text{Mg}_{76.5}\text{Zn}_{20}\text{Y}_{3.5}$ deposits, exhibiting further refined intergranular secondary phases, have good tensile properties at both room temperature and an elevated temperature of 473 K. This is especially true for $\text{Mg}_{97}\text{Zn}_1\text{Y}_2$ which showed tensile strength of 340 MPa and 271 MPa at room temperature and 473 K, respectively. The other alloys, $\text{Mg}_{88}\text{Zn}_{10}\text{Y}_2$ and
$Mg_{76.5}Zn_{20}Y_{3.5}$, respectively, had 305 MPa and 271 MPa at room temperature and 237 MPa and 216 MPa at 473 K. The achieved strength values significantly exceed those of commercial wrought alloys which typically range 200 MPa to 300 MPa at room temperatures and 150 MPa to 200 MPa at 473 K. The $Mg_{97}Zn_{1}Y_2$, $Mg_{88}Zn_{10}Y_2$ and $Mg_{76.5}Zn_{20}Y_{3.5}$ deposits, however, had limited ductility owing primarily to the intergranular intermetallic constituents along which the tensile specimens fractured.

- Although further optimization is clearly required particularly for improving the ductility, the present work has shown that UDS processing can lead to new droplet-based processing of high-performance $Mg$-$Zn$-$Y$ and other $Mg$ alloys.
7. **APPENDIX**

Appendix A: X-Y Table Control System

The substrate motion is computer-controlled in the X-Y direction via a X-Y table (DCI Design Components: EB6-120) as describe in Chap 3. Control software is integrated with a control program and an input file (data file of parameters for the X-Y table motion). The control program coded in C language contains a function to move the X-Y table and read the input file. The input file is coded along with XML\(^2\) format. This file is linked to an XSLT\(^3\) file and an XML Schema\(^4\) file to check its XML format and values. The file can be verified using a simulation program coded with Java language as Java Applet\(^5\). Figure 7-1 shows a schematic of the software relations.

Generally, if a program is required to read some data files, numerous lines are wasted in the programming source code to verify its file format validity and values from the input data files to prevent program errors or bugs, which is not a main part in most of programs. Since XML schema can define the format of targeted XML file, by utilizing this property, external software called “XML validator” or “XML parser” can take over

\(^2\) **XML**: eXtensive Markup Language - a W3C-recommended general-purpose markup language for creating special-purpose markup languages.

\(^3\) **XSLT**: eXtensible Stylesheet Language Transformations - an XML-based language used for the transformation of XML documents. XSLT is most often used to convert data between different XML schemas or to convert XML data into web pages or PDF documents.

\(^4\) **XML Schema**: a description of a type of XML document.

\(^5\) **Java Applet**: A Java applet is an applet written in the Java programming language. Applet is a software component that runs in the context of another program, like a web browser.
the function to validate the data file. Or it can be quickly verified under a web browser with XSLT file. This means that some of these lines can be eliminated from the code. This allows the code to be further simplified and improved on its readability to prevent miss codlings of the main program.

In general if a user is not familiar with XML technology, utilizing XML validator or other software is complicated due to the difficulty in identifying the correct software. But instead of using complicated softwares, a Web browser, which is installed in most current computers, provides a simple method to utilize this technique. In this situation, XSLT can work as a programming language to check data in XML file, and to re-form into different data format. By applying this function, the data can be inspected more intuitively under a PC monitor. Figure 7-2 demonstrates how error messages are given on the screen with a XSLT file. A CSS\textsuperscript{6} file is applied together with the web browser for an aid of user interface on the screen.

The Java Applet is linked with each input file to check graphically the actual motion in the chamber according to the input parameters. Figure 7-3 demonstrates the actual Java applet (X-Y table motion simulator) image on a web browser. Left side image represents a view from top of the X-Y table, and right side image represents a 3D image inside of chamber to get a feeling of the X-Y table motion more intuitively. Current program allows to check only main sequence of parameters. The source codes used to control the X-Y table are listed in following pages.

\textsuperscript{6} CSS: Cascading Style Sheets is a style sheet language used to describe the presentation of a document written in a markup language.
Figure 7-1: Source codes used to control X-Y table.
Figure 7-2: Error messages suggest possible problems for an input file.
Figure 7-3: X-Y table motion simulator demonstrates the Java applet running with a web browser.
C source code for X-Y table motion control

XY table motion control program

2004 06 25 Version 2.0 sheet1.c Revised to deal with a input file and different motion patterns.
2004 06 28 Version 2.2 sheet2.c Revised for a different motion pattern.
2004 07 01 Version 2.5 sheet3.c Revised to optimize this source file.
2004 07 13 Version 2.6 sheet4.c Revised completely from Chuck's source. Input file is written in XML.
2004 11 07 Version 2.7 sheet5.c Revised to add additional controls for single steps with key strokes as well as multiple steps.
2004 11 08 Version 2.75 sheet55.c Revised to add additional control.
2005 10 10 Version 2.8 sheet6.c Revised to add additional control. (for OptionalControl part)
2005 10 14 Version 2.86 sheet66.c Revised to optimize this source file. Input file format is changed slightly.
2005 10 25 Version 2.87 sheet67.c Revised to optimize this source file. Input file format is changed slightly. [Some replacement from array into structure to improve readability]
2005 10 25 Version 2.87 sheet67.c [Added irregular motion part]

Hiroki Fukuda @ Northeastern University

Note: If Borland C++ is used to build this program, make a project file for DOS mode and add a node for TES5000L.LIB.

#define TEST 1 /* set 1 during debugging */
#define TMP 100 /* define the number of temp array */
#define VERSION "2.87" /* define the version number of this program */
#define P_MAX 15 /* define the maximum number of parameter sets */
#define SQ_MAX 30 /* define the maximum number of steps */

#include "TE5000.H" /* Header file provided from Technology 80, Inc. */
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>

*--------------------------------------------------------------------------------------------------------------------*

    Note: If Borland C++ is used to build this program, make a project file for DOS mode and add a node for TES5000L.LIB.

 XY table motion control program

2004 06 25 Version 2.0 sheet1.c Revised to deal with a input file and different motion patterns.
2004 06 28 Version 2.2 sheet2.c Revised for a different motion pattern.
2004 07 01 Version 2.5 sheet3.c Revised to optimize this source file.
2004 07 13 Version 2.6 sheet4.c Revised completely from Chuck's source. Input file is written in XML.
2004 11 07 Version 2.7 sheet5.c Revised to add additional controls for single steps with key strokes as well as multiple steps.
2004 11 08 Version 2.75 sheet55.c Revised to add additional control.
2005 10 10 Version 2.8 sheet6.c Revised to add additional control. (for OptionalControl part)
2005 10 14 Version 2.86 sheet66.c Revised to optimize this source file. Input file format is changed slightly.
2005 10 25 Version 2.87 sheet67.c Revised to optimize this source file. Input file format is changed slightly. [Some replacement from array into structure to improve readability]
2005 10 25 Version 2.87 sheet67.c [Added irregular motion part]

Hiroki Fukuda @ Northeastern University

Note: If Borland C++ is used to build this program, make a project file for DOS mode and add a node for TES5000L.LIB.

#define TEST 1 /* set 1 during debugging */
#define TMP 100 /* define the number of temp array */
#define VERSION "2.87" /* define the version number of this program */
#define P_MAX 15 /* define the maximum number of parameter sets */
#define SQ_MAX 30 /* define the maximum number of steps */

#include "TE5000.H" /* Header file provided from Technology 80, Inc. */
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>

*--------------------------------------------------------------------------------------------------------------------*
while ( counter < op.lnum || !op.lnum ){                   /* break if (counter < LayerNumber) or
   
   /* Main loop */
   
   printf("\n--- To stop the cycle from repeating press <k> <Enter> ----\n");
   }
   
   else {  
        fflush(stdin);                                                  /* better not to use fflush() due to processor dependency */
        //getc(stdin);                                                  /* remove remained character(\n) if typed from command line */
        
        while(isspace(check = getchar()) && check != EOF)               /* check yes(AnyKey) or no(N)                   */
        
        ConfirmData( main_sq, add_sq, irr_sq_s, irr_sq_e, p, op);            /* print out input parameters                   */
        
        X=0;
        OPTION      op;                                                     /* Optional settings                            */
                irr_sq_e[SQ_MAX];                                       /* Sequence set for irregular motion part       */
                irr_sq_s[SQ_MAX],                                       /* Sequence set for irregular motion part       */
        SEQUENCE    main_sq[SQ_MAX],                                        /* Sequence set for main part                   */
        counter=0,                                                      /* Loop counter for sprayed layer               */
        c=0,                                                            /* Character by key stroke                      */
        check,                                                          /* Character by key stroke for main function    */
        
        int i,                                                              /* Loop counter                                 */
        unsigned short wBoardAddr = 0x300;
        int main(void)
        
        /*-----------------------------------------------------------------------------------------------------------------*/
        
        *-----------------------------------------------------------------------------------------------------------------*/
        
        /*-----------------------------------------------------------------------------------------------------------------*/
        int     LowerC(int);
        void    initOPTION(OPTION *, int );
if ( !counter || !op.opt || ( op.lnum && op.opt ) ) {
    printf("Unhit any key and/or <enter> ----------\n");
    // if(main_sq[0].axis == -1){
    //   if(kbhit()){if( ( c=LowerC(getch()) )=='k'){break;}}
    //     else continue;
    // }else{
    if((c=LowerC(getchar()))=='k'){break;}              /* break this loop if <k> is typed */
    // }
}

/* Irregular Motion Loop (Starting Point)*/
if(op.irr && !counter){
    printf("--- Starting Motion  ----------\n");
    for (i=1; i<=irr_sq_s[0].order; i++) {
        printf("[\%2d]\n",i);
        move(irr_sq_s[i],p[irr_sq_s[i].pn],0,i);
    }\n    X=0;
    Y=0;
}

/* Irregular Motion Loop (Ending Point)*/
if(op.irr)
    printf("--- Ending Motion  ----------\n");
    for (i=1; i<=irr_sq_e[0].order; i++) {
        printf("[\%2d]\n",i);
        move(irr_sq_e[i],p[irr_sq_e[i].pn],0,i);
    }

/* Additional Motion Loop */
if(op.add && counter) {
    printf("--- Additional Loop  ----------\n");
    for (i=1; i<=add_sq[0].order; i++) {
        printf("--- Hit any key and/or <enter> ----------\n");
        while(isspace(check = getchar()) && check != EOF) ;
        fflush(stdin);                                   /* try not to use fflush()                      */
        if( LowerC(getchar()) == 'k'){break;}            /* break this loop if <k> is typed */
    }
    //    if(!op.cnt){                                    /* if not Continuous.                           */
    //        printf("--- Hit any key and/or <enter> ----------\n");
    //        while(isspace(check = getchar()) && check != EOF) ;
    //        fflush(stdin);                                   /* try not to use fflush()                      */
    //        if( LowerC(getchar()) == 'k'){break;}            /* break this loop if <k> is typed */
    //    }
    printf("\nPosition X= %d, Position Y=%d\n",X,Y);
}

/* Continuous Motion Loop */
if(op.cnt)
    printf("--- Continuous Loop  ----------\n");
    for (i=1; i<=main_sq[0].order; i++) {
        printf("[\%2d]\n",i);
        if (move(main_sq[i],p[main_sq[i].pn],c,i)) AssemblyTitle("\n"; flag=1;)
    }

if( !flag ){ printf("Type a valid key. <a> <d> <x> <w> or <k>\n"); }
    counter++;

/* Regular Motion Loop */
if(main_sq[0].irr && !counter){
    printf("--- Hit any key and/or <enter> ----------\n");
    for (i=1; i<=main_sq[0].order; i++) {
        printf("[\%2d]\n",i);
        move(main_sq[i],p[main_sq[i].pn],0,i);
    }\n    X=0;
    Y=0;
}

printf("--- Move Finished ---\n");
}

if(getchar() != 'z');
printf("----- Terminating Program -----\n");
return 0;

*/

/* Syntax */
```
        if(!axis){ /* if A axis */
          printf("Axis=%d,Dis=%d,L_Vel=%d,Vel=%d,Acc=%d,Dec=%d,RDP=%d,Mul=%d,",axis,p.dis,p.lvel,p.vel,p.acc,p.dec,p.rdp,p.mul);
          #endif
          while(IsBusy(axis));                                                /* wait for move to be completed               */
          StartStop(axis, START1_MOVE);                                       /* start move                                  */
          StartStop(axis, RESET_MOVE);                                        /* reset move before starting up               */
          ModeSelect(axis, direction?POSMODE_DOWN:POSMODE_UP);                /* select preset mode and the move direction   */
          Multiplier(  axis, p.mul);
          DownPoint(   axis, p.rdp );
          Acceleration(axis, p.acc );
          Velocity1(   axis, p.vel );
          LowVelocity( axis, p.lvel);
          Distance(    axis, p.dis );                                         /* load the parameters into registers          */
        }
        else{ return 0;}
    }
    else{ delay( sq.pn );
      printf("Wait %6.3f seconds
[  ",((double)sq.pn )/1000 );
      //gotc(stdin )
      //
      fflush(stdin);
    if( !sq.pn )
      else if( sq.dir == -3){
    }
    else if((LowerC(c)=='w') && ln==2){ direction = 1;    axis=1;}      /* direction => -2,  1,  0  DOWN (B)   */
    else if((LowerC(c)=='x') && ln==2){ direction = 0;    axis=1;}      /* direction => -2,  0,  1  UP (B)     */
    else if((LowerC(c)=='d') && ln==1){ direction = 1;    axis=0;}      /* direction =>  1, -2, -2  RIGHT (A)  */
    else if(sq.dir == -2){
    }
    else if((LowerC(c)=='w') && ln!=1){ direction = 3-ln; axis=3-ln;}   /* direction => -2,  1,  0  DOWN (B->A)*/
    if(     (LowerC(c)=='a') && ln==1){ direction = 0;    axis=0;}      /* direction =>  0, -2, -2  LEFT (A)   */
    if(sq.dir == -1){
      printf("Wait %6.3f seconds
[  ",((double)sq.pn )/1000 );
      delay( sq.pn );
    }
    else return 0;
        }
        else if(sq.dir == -2){
          if( (LowerC(c)=='a') && ln==0) { direction = 0;    axis=0; } /* direction =>  0, -2, -2  LEFT (A)   */
          else if(LowerC(c)=='d') && ln==0) { direction = 1;    axis=0; } /* direction =>  1, -2, -2  RIGHT (A)  */
          else if((LowerC(c)=='w') && ln==0) { direction = -2,  0,  1  UP (B)     */
          else if((LowerC(c)=='w') && ln==0) { direction = -3,  1,  0  DOWN (B->A)*/
            else if( (LowerC(c)=='a') && ln==0) { direction = 0;    axis=0; } /* direction =>  0, -2, -2  LEFT (A)   */
            else return 0;
        }
        else if(sq.dir == -3){
          if( (LowerC(c)=='a') && ln==0) { direction = 0;    axis=0; } /* direction =>  0, -2, -2  LEFT (A)   */
          else if(LowerC(c)=='d') && ln==0) { direction = 1;    axis=0; } /* direction =>  1, -2, -2  RIGHT (A)  */
          else if((LowerC(c)=='w') && ln==0) { direction = -2,  0,  1  UP (B)     */
          else if((LowerC(c)=='w') && ln==0) { direction = -3,  1,  0  DOWN (B->A)*/
            else if( (LowerC(c)=='a') && ln==0) { direction = 0;    axis=0; } /* direction =>  0, -2, -2  LEFT (A)   */
            else return 0;
        }
        else if(sq.dir == -2){
          if( (LowerC(c)=='a') && ln==0) { direction = 0;    axis=0; } /* direction =>  0, -2, -2  LEFT (A)   */
          else if(LowerC(c)=='d') && ln==0) { direction = 1;    axis=0; } /* direction =>  1, -2, -2  RIGHT (A)  */
          else if((LowerC(c)=='w') && ln==0) { direction = -2,  0,  1  UP (B)     */
          else if((LowerC(c)=='w') && ln==0) { direction = -3,  1,  0  DOWN (B->A)*/
            else if( (LowerC(c)=='a') && ln==0) { direction = 0;    axis=0; } /* direction =>  0, -2, -2  LEFT (A)   */
            else return 0;
        }
        else if(sq.dir == -3){
          if( (LowerC(c)=='a') && ln==0) { direction = 0;    axis=0; } /* direction =>  0, -2, -2  LEFT (A)   */
          else if(LowerC(c)=='d') && ln==0) { direction = 1;    axis=0; } /* direction =>  1, -2, -2  RIGHT (A)  */
          else if((LowerC(c)=='w') && ln==0) { direction = -2,  0,  1  UP (B)     */
          else if((LowerC(c)=='w') && ln==0) { direction = -3,  1,  0  DOWN (B->A)*/
            else if( (LowerC(c)=='a') && ln==0) { direction = 0;    axis=0; } /* direction =>  0, -2, -2  LEFT (A)   */
            else return 0;
        }
        else if(sq.dir == -2){
          if( (LowerC(c)=='a') && ln==0) { direction = 0;    axis=0; } /* direction =>  0, -2, -2  LEFT (A)   */
          else if(LowerC(c)=='d') && ln==0) { direction = 1;    axis=0; } /* direction =>  1, -2, -2  RIGHT (A)  */
          else if((LowerC(c)=='w') && ln==0) { direction = -2,  0,  1  UP (B)     */
          else if((LowerC(c)=='w') && ln==0) { direction = -3,  1,  0  DOWN (B->A)*/
            else if( (LowerC(c)=='a') && ln==0) { direction = 0;    axis=0; } /* direction =>  0, -2, -2  LEFT (A)   */
            else return 0;
        }
        else if(sq.dir == -3){
          if( (LowerC(c)=='a') && ln==0) { direction = 0;    axis=0; } /* direction =>  0, -2, -2  LEFT (A)   */
          else if(LowerC(c)=='d') && ln==0) { direction = 1;    axis=0; } /* direction =>  1, -2, -2  RIGHT (A)  */
          else if((LowerC(c)=='w') && ln==0) { direction = -2,  0,  1  UP (B)     */
          else if((LowerC(c)=='w') && ln==0) { direction = -3,  1,  0  DOWN (B->A)*/
            else if( (LowerC(c)=='a') && ln==0) { direction = 0;    axis=0; } /* direction =>  0, -2, -2  LEFT (A)   */
            else return 0;
```
else
  y = direction?(p.dis):(-p.dis);
endif

if TEST
  return 1;
endif

delay(100);

return 0;

open ramifications

+ Syntax
+ void ConfirmData( SEQUENCE array_of_main_sequence[],
+   SEQUENCE array_of_additional_sequence[],
+   SEQUENCE array_of_irregular_sequence_start[],
+   SEQUENCE array_of_irregular_sequence_end[],
+   PARAMETER array_of_parameters[],
+   OPTION option)
+ {
+   int i=0;
+
+   printf("YnStep SequenceMn:");
+   for(i=1;i<main_sq[0].order;i++)
+     printf("[\%02d]  Axis=%d  Direction=%d  ParameterNo=%2d\n", i, main_sq[i].axis, main_sq[i].dir, main_sq[i].on);
+   }
+
+   printf("YnParametersMn:");
+   for(i=1;i<op[0].pn;i++)
+     printf("[\%02d]  Dis=%6d  L_Vel=%6d  Vel=%6d  Acc=%3d  Dec=%3d  RDP=%3d  Mul=%3d\n",
+       i, p[i].dis, p[i].lvel, p[i].vel, p[i].acc, p[i].dec, p[i].rdp, p[i].mul);
+
+   printf("YnOptionalControlsMn: LayerNumber=%3d  AdditionalMotionMn, IrregularMn, Irregular%3dMn", op.opt?"YES":"NO", op.lnum, op.add?"YES":"NO", op.irr?"YES":"NO");
+
+   printf("YnStep Sequence for Additional MotionMn");
+   for(i=1;i<add_sq[0].order;i++)
+     printf("[\%02d]  Axis=%d  Direction=%d  ParameterNo=%2d\n", i, add_sq[i].axis, add_sq[i].dir, add_sq[i].pn);
+
+   printf("YnStep Sequence for Irregular(startingPoint)Mn");
+   for(i=1;i<irr_sq_s[0].order;i++)
+     printf("[\%02d]  Axis=%d  Direction=%d  ParameterNo=%2d\n", i, irr_sq_s[i].axis, irr_sq_s[i].dir, irr_sq_s[i].pn);
+
+   printf("YnStep Sequence for Irregular(EndingPoint)Mn");
+   for(i=1;i<irr_sq_e[0].order;i++)
+     printf("[\%02d]  Axis=%d  Direction=%d  ParameterNo=%2d\n", i, irr_sq_e[i].axis, irr_sq_e[i].dir, irr_sq_e[i].pn);
+
+   printf("YnProgram Ver.=\"VERSION\"");
+
+   int GetInputData( SEQUENCE array_of_main_sequence[],
+   SEQUENCE array_of_additional_sequence[],
+   SEQUENCE array_of_irregular_sequence_start[],
+   SEQUENCE array_of_irregular_sequence_end[],
+   PARAMETER array_of_parameters[],
+   OPTION *option)
+ {
+   With a call to GetInputData, All X-Y table setting parameters are set into arrays declared for
+   these parameters in main function.
**Syntax**

```c
int SetSequence(FILE *fp, SEQUENCE sq[]){
    return 0;
}
```

**Description**

With a call to SetSequence, sequence settings are set into arrays declared for these parameters in main function.

**Return Value**

- On success, SetSequence returns 0.
- On error, SetSequence returns 1.

```c
int GetInputData(SEQUENCE sq[], SEQUENCE add_sq[], SEQUENCE irr_sq_s[], SEQUENCE irr_sq_e[], PARAMETER p[], OPTION *op)
```

```c
char tmp[250];
int chkflg=0;
FILE *fp;
```

**Return Value**

- On error, GetInputData prints error message and returns 1.
- On success, GetInputData returns 0.

```c
int GetInputData(SEQUENCE sq[], SEQUENCE add_sq[], SEQUENCE irr_sq_s[], SEQUENCE irr_sq_e[], PARAMETER p[], OPTION *op)
```
int SetOption(FILE *fp, SEQUENCE add_sq[], SEQUENCE irr_sq_s[], SEQUENCE irr_sq_e[], OPTION *op)
{
    return 0;
}

int SetOption(FILE *fp, PARAMETER p[])
{
    return 0;
}

char tmp[250];

if( GetElementValue( fp, "<ParameterSet", &p[0].pn ) )
{
    for( i=1; i<op[0].pn; i++ )
    {
        if( GetElementValue( fp, "Parameter", &p[i].pn ) )
            return 1;
        if( GetElementValue( fp, "Distance", &p[i].dis ) )
            return 1;
        if( GetElementValue( fp, "Low_Velocity", &p[i].lvel ) )
            return 1;
        if( GetElementValue( fp, "Velocity", &p[i].vel ) )
            return 1;
        if( GetElementValue( fp, "Deceleration", &p[i].decl ) )
            return 1;
        if( GetElementValue( fp, "Acceleration", &p[i].acc ) )
            return 1;
        if( GetElementValue( fp, "Ramp_Down_Point", &p[i].rd ) )
            return 1;
        if( GetElementValue( fp, "Multiplier", &p[i].mul ) )
            return 1;
        fscanf( fp, "%s", tmp );
            /* process to next line. (</Parameter>) */
        fscanf( fp, "%s", tmp );
            /* process to next line. (</ParameterSet>) */
    }
    return 0;
}

/* Syntax */

/* Description */

/* With a call to SetOption, Optional setting parameters are set into arrays declared for */
/* these parameters in main function. */

/* Return Value */

/* On success, SetOption returns 0. */
/* On error, SetOption returns 1. */

int SetOption(FILE *stream, SEQUENCE array_of_additional_sequence[],
              SEQUENCE array_of_irregular_sequence_start[],
              SEQUENCE array_of_irregular_sequence_end[],
              OPTION *op )
{
    return 0;
}

/* int SetSequence(FILE *stream, SEQUENCE start[], SEQUENCE end[] )
{
    return 0;
}

/* Description */

/* With a call to SetSequence, X-Y table setting parameters are set into arrays declared for */
/* these parameters in main function. */

/* Return Value */

/* On success, SetSequence returns 0. */
/* On error, SetSequence returns 1. */

int SetSequence(FILE *fp, SEQUENCE add_sq[], SEQUENCE irr_sq_s[], SEQUENCE irr_sq_e[])
{
    return 0;
}

char tmp[250];

if( GetElementValue( fp, "<ParameterSet", &p[0].pn ) )
{
    for( i=1; i<op[0].pn; i++ )
    {
        if( GetElementValue( fp, "Parameter", &p[i].pn ) )
            return 1;
        if( GetElementValue( fp, "Distance", &p[i].dis ) )
            return 1;
        if( GetElementValue( fp, "Low_Velocity", &p[i].lvel ) )
            return 1;
        if( GetElementValue( fp, "Velocity", &p[i].vel ) )
            return 1;
        if( GetElementValue( fp, "Deceleration", &p[i].decl ) )
            return 1;
        if( GetElementValue( fp, "Acceleration", &p[i].acc ) )
            return 1;
        if( GetElementValue( fp, "Ramp_Down_Point", &p[i].rd ) )
            return 1;
        if( GetElementValue( fp, "Multiplier", &p[i].mul ) )
            return 1;
        fscanf( fp, "%s", tmp );
            /* process to next line. (</Parameter>) */
        fscanf( fp, "%s", tmp );
            /* process to next line. (</ParameterSet>) */
    }
    return 0;
}

/* Syntax */

/* Description */

/* With a call to SetSequence, Optional setting parameters are set into arrays declared for */
/* these parameters in main function. */

/* Return Value */

/* On success, SetSequence returns 0. */
/* On error, SetSequence returns 1. */

/* Read irregular step sequences - starting point */

fscanf( fp, "%s", tmp );
    /* process to next line. (<StartingPoint>) */
if( SetSequence( fp, irr_sq_s[] ) )
{
    fscanf( fp, "%s", tmp );
        /* process to next line. (<StartingPoint>) */
    return 1;
}

/* Read irregular step sequences - ending point */

fscanf( fp, "%s", tmp );
    /* process to next line. (<EndingPoint>) */
if( SetSequence( fp, irr_sq_e[] ) )
{
    fscanf( fp, "%s", tmp );
        /* process to next line. (<EndingPoint>) */
    return 1;
}

/* Read additional step sequences */

fscanf( fp, "%s", tmp );
    /* process to next line. (<Irregular>) */
if( SetSequence( fp, add_sq[] ) )
{
    fscanf( fp, "%s", tmp );
        /* process to next line. (<Irregular>) */
    return 1;
}

/* Read additional step sequences */

fscanf( fp, "%s", tmp );
    /* process to next line. (<AdditionalMotion>) */
if( SetSequence( fp, add_sq[] ) )
{
    fscanf( fp, "%s", tmp );
        /* process to next line. (<AdditionalMotion>) */
    return 1;
}

return 0;

initOPTION( op, 0 );
initSEQUENCE( &add_sq[0], 0 );
initSEQUENCE( &irr_sq_s[0], 0 );
initSEQUENCE( &irr_sq_e[0], 0 );

if( GetElementValue( fp, "OptionalControl", &p[i].opt ) )
    return 1;
    /* set OptionalControl value */
if( GetElementValue( fp, "LayerNumber", &p[i].lnum ) )
    return 1;
    /* set LayerNumber value */
if( GetElementValue( fp, "Irregular", &p[i].irr ) )
    return 1;
    /* set Irregular Motion value */
```c
int LowerC(int c) {
    s->pn   = i;
    s->dir  = i;
    s->order= i;
}

void initSEQUENCE( SEQUENCE *s, int i )
    o->irr  = i;
    o->add  = i;
    o->lnum = i;
    o->opt  = i;
    fscanf(fp, "%a", line);
    return 1;
}

int GetElementValue(FILE *fp, char *element, int *n) {
    char line[TMP], temp[TMP];
    fscanf(fp, "%a", line);
    if( sscanf(line, "%[^>]%[>]%d%[^\n]", temp, temp, n, temp) != 4 ) { return 1; }
    if( sscanf(line, "%[^""]%["]%d%[^\n]", temp, temp, n, temp) != 4 ) { return 1; }
}
```

```c
/* Syntax */
/* Description */
/* With a call to GetElementValue, a number for the attribute is set to attribute_value. */
/* Return Value */
/* On success, GetElementValue returns 0. */
/* On error, GetElementValue returns 1. */
```
Example source code for an input file

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="xydata.xsl"?>

<!-- test5.xml -->
<!-- USAGE for Ver.2.87 -->
<!-- Axis       0=A  1=B -1=Key                                -->
<!-- Direction  0=UP 1=DOWN -1=MESH -2=CROSS -3 =delay(ms)      -->
<!-- Direction -2=CROSS                                        -->
<!--      StepNumber1 is assigned RIGHT(d)/LEFT (a) motion      -->
<!--      StepNumber2 is assigned UP(w)/DOWN(x)  motion         -->
<!--      StepNumber2&3 are assigned UP(w)/DOWN(x) motion      -->
<!--      Direction -3=delay                                        -->
<!--      ParameterNumber0 stop till typed                    -->
<!--      ParameterNumber=XXX  stop XXX msecond s               -->
<!-- LayerNumber 0=moved by key strokes. number=layered number -->

<XY-Table-Motion-Input-Data-File xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="xydata.xsd">
    <StepSequence Value="20">
        <Step Number="1">
            <Axis>1</Axis>
            <Direction>1</Direction>
            <ParameterNumber>1</ParameterNumber>
        </Step>
        <Step Number="2">
            <Axis>0</Axis>
            <Direction>1</Direction>
            <ParameterNumber>2</ParameterNumber>
        </Step>
        <Step Number="3">
            <Axis>1</Axis>
            <Direction>0</Direction>
            <ParameterNumber>3</ParameterNumber>
        </Step>
        <Step Number="4">
            <Axis>0</Axis>
            <Direction>1</Direction>
            <ParameterNumber>4</ParameterNumber>
        </Step>
        <Step Number="5">
            <Axis>1</Axis>
            <Direction>1</Direction>
            <ParameterNumber>5</ParameterNumber>
        </Step>
        <Step Number="6">
            <Axis>0</Axis>
            <Direction>1</Direction>
            <ParameterNumber>2</ParameterNumber>
        </Step>
        <Step Number="7">
            <Axis>1</Axis>
            <Direction>0</Direction>
            <ParameterNumber>3</ParameterNumber>
        </Step>
        <Step Number="8">
            <Axis>0</Axis>
            <Direction>1</Direction>
            <ParameterNumber>4</ParameterNumber>
        </Step>
    </StepSequence>
</XY-Table-Motion-Input-Data-File>
```
<Step Number="11">
<Axis>0</Axis>
<Direction>0</Direction>
<ParameterNumber>3</ParameterNumber>
</Step>

<Step Number="12">
<Axis>0</Axis>
<Direction>0</Direction>
<ParameterNumber>4</ParameterNumber>
</Step>

<Step Number="13">
<Axis>1</Axis>
<Direction>1</Direction>
<ParameterNumber>5</ParameterNumber>
</Step>

<Step Number="14">
<Axis>0</Axis>
<Direction>1</Direction>
<ParameterNumber>2</ParameterNumber>
</Step>

<Step Number="15">
<Axis>1</Axis>
<Direction>0</Direction>
<ParameterNumber>3</ParameterNumber>
</Step>

<Step Number="16">
<Axis>0</Axis>
<Direction>0</Direction>
<ParameterNumber>4</ParameterNumber>
</Step>

<Step Number="17">
<Axis>1</Axis>
<Direction>1</Direction>
<ParameterNumber>5</ParameterNumber>
</Step>

<Step Number="18">
<Axis>0</Axis>
<Direction>1</Direction>
<ParameterNumber>2</ParameterNumber>
</Step>

<Step Number="19">
<Axis>1</Axis>
<Direction>0</Direction>
<ParameterNumber>3</ParameterNumber>
</Step>

<Step Number="20">
<Axis>0</Axis>
<Direction>0</Direction>
<ParameterNumber>6</ParameterNumber>
</Step>

</StepSequence>

<ParameterSet Value="8">
<Parameter Number="1">
<Distance>1260</Distance>
<Low_Velocity>5</Low_Velocity>
<Velocity>5994</Velocity>
<Acceleration>100</Acceleration>
<Deceleration>100</Deceleration>
<Ramp_Down_Point>360</Ramp_Down_Point>
<Multiplier>610</Multiplier>
</Parameter>

<Parameter Number="2">
<Distance>4000</Distance>
<Low_Velocity>5</Low_Velocity>
<Velocity>6150</Velocity>
<Acceleration>81</Acceleration>
<Deceleration>81</Deceleration>
<Ramp_Down_Point>307</Ramp_Down_Point>
<Multiplier>610</Multiplier>
</Parameter>

<Parameter Number="3">
<Distance>700</Distance>
<Low_Velocity>5</Low_Velocity>
<Velocity>4114</Velocity>
<Acceleration>61</Acceleration>
<Deceleration>61</Deceleration>
<Ramp_Down_Point>104</Ramp_Down_Point>
<Multiplier>610</Multiplier>
</Parameter>

<Parameter Number="4">
<Distance>3880</Distance>
<Low_Velocity>5</Low_Velocity>
<Velocity>6429</Velocity>
<Acceleration>78</Acceleration>
<Deceleration>78</Deceleration>
<Ramp_Down_Point>323</Ramp_Down_Point>
<Multiplier>610</Multiplier>
</Parameter>

<Parameter Number="5">
<Distance>566</Distance>
<Low_Velocity>5</Low_Velocity>
<table>
<thead>
<tr>
<th>Parameter Number</th>
<th>Distance</th>
<th>Low_Velocity</th>
<th>Velocity</th>
<th>Acceleration</th>
<th>Deceleration</th>
<th>Ramp_Down_Point</th>
<th>Multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>4560</td>
<td>5</td>
<td>6510</td>
<td>77</td>
<td>77</td>
<td>327</td>
<td>610</td>
</tr>
<tr>
<td>7</td>
<td>18500</td>
<td>5</td>
<td>7704</td>
<td>65</td>
<td>65</td>
<td>386</td>
<td>610</td>
</tr>
<tr>
<td>8</td>
<td>3000</td>
<td>5</td>
<td>4283</td>
<td>117</td>
<td>117</td>
<td>215</td>
<td>610</td>
</tr>
</tbody>
</table>

OptionalControl Value="1"

LayerNumber=0
Continuous=0
Irregular=1

StartingPoint

StepSequence Value="2"
Step Number="1"
Axis=0
Direction=1
ParameterNumber=8

Step
Axis=1
Direction=1
ParameterNumber=7

StepSequence

EndingPoint

StepSequence Value="2"
Step Number="1"
Axis=0
Direction=1
ParameterNumber=7

Step
Axis=1
Direction=1
ParameterNumber=8

StepSequence

Irregular

AdditionalMotion Value="1"

StepSequence Value="2"
Step Number="1"
Axis=0
Direction=1
ParameterNumber=7

Step
Axis=1
Direction=1
ParameterNumber=8

StepSequence

AdditionalMotion

OptionalControl
Example source code for XSLT and CSS

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
  <xsl:output method="html" encoding="UTF-8"/>
  <!-- (c) Hiroki Fukuda @ Northeastern University 2004-2005 -->
  <!-- doesn't work if <style> is commented out as you do in HTML -->
  <xsl:template match="/">
    <html>
      <header>
        <link rel="stylesheet" type="text/css" href="xydata.css"/>
      </header>
      <body>
        <table>
          <tr>
            <th>
              <table>
                <tr>
                  <th>
                    Main Step Sequence
                  </th>
                  <th>
                    Irregular Step Sequence
                    <!-- Starting Point -->
                    <xsl:apply-templates select="//OptionalControl/Irregular/StartingPoint/StepSequence"/>
                    <!-- Ending Point -->
                    <xsl:apply-templates select="//OptionalControl/Irregular/EndingPoint/StepSequence"/>
                    Additional Step Sequence
                    <xsl:apply-templates select="//OptionalControl/AdditionalMotion/StepSequence"/>
                  </th>
                </tr>
              </table>
            </th>
          </tr>
          <tr>
            <th>
              Parameter Sets
              <xsl:apply-templates select="//ParameterSet"/>
            </th>
          </tr>
        </table>
        <a href="XYTEmu.html" target="inner">XY table motion simulator</a>
        <iframe src="blank.html" width="100%" height="600" name="inner" frameborder="0">
        </iframe>
      </body>
    </html>
</xsl:template>
<xsl:template match="XY-Table-Motion-Input-Data-File/StepSequence" mode="main">
  <xsl:call-template name="StepErr">
    <xsl:with-param name="PElem">
      <xsl:value-of select="/XY-Table-Motion-Input-Data-File/StepSequence/@Value"/>
    </xsl:with-param>
    <xsl:with-param name="CElem">
      <xsl:value-of select="count(/XY-Table-Motion-Input-Data-File/StepSequence/Step)"/>
    </xsl:with-param>
    <xsl:with-param name="Ptxt">StepSequence</xsl:with-param>
    <xsl:with-param name="Ctxt">Step</xsl:with-param>
  </xsl:call-template>
  <table>
    <tr>
      <xsl:call-template name="StepTitle"/>
    </tr>
    <xsl:for-each select="Step">
      <tr onmouseover="document.layers[para].para.style.backgroundColor='blue'" onmouseout="para.style.backgroundColor='white'">
        <td>
          <xsl:call-template name="StepNum">
            <xsl:with-param name="link">
              <xsl:value-of select="/XY-Table-Motion-Input-Data-File/StepSequence/@Value"/>
            </xsl:with-param>
          </xsl:call-template>
        </td>
      </tr>
    </xsl:for-each>
  </table>
</xsl:template>
```

### Parameters Table

<table>
<thead>
<tr>
<th>No.</th>
<th>Distance</th>
<th>Low Velocity</th>
<th>Velocity</th>
<th>Acceleration</th>
<th>Deceleration</th>
<th>Ramp Down Point</th>
<th>Multiplier</th>
</tr>
</thead>
</table>

### Step Sequence Table

<table>
<thead>
<tr>
<th>Step</th>
<th>Distance</th>
<th>Low Velocity</th>
<th>Velocity</th>
<th>Acceleration</th>
<th>Deceleration</th>
<th>Ramp Down Point</th>
<th>Multiplier</th>
</tr>
</thead>
</table>

#### Error Handling

- **PElem**: Parameter Set @Value
- **CElem**: Count of Parameters
- **Ptxt**: Parameter Set
- **Ctxt**: Parameter

#### Step Title

- **PElem**: Optional Control/Additional Motion/Step Sequence
- **CElem**: Count of Steps
- **Ptxt**: Step Sequence
- **Ctxt**: Step

---

[219]
<table>
  <tr>
    <td>
      <xsl:call-template name="StepTitle"/>
    </td>
  </tr>
</table>

<xsl:template match="//OptionalControl/Irregular/StartingPoint/StepSequence">
  <xsl:call-template name="StepErr">
    <xsl:with-param name="PElem">
      <xsl:value-of select="//OptionalControl/Irregular/StartingPoint/StepSequence/@Value"/>
    </xsl:with-param>
    <xsl:with-param name="CElem">
      <xsl:value-of select="count(//OptionalControl/Irregular/StartingPoint/StepSequence/Step)"/>
    </xsl:with-param>
    <xsl:with-param name="Ptxt">StepSequence</xsl:with-param>
    <xsl:with-param name="Ctxt">Step</xsl:with-param>
  </xsl:call-template>

  <table>
    <tr>
      <xsl:call-template name="StepNum">
        <xsl:with-param name="link">
          <xsl:value-of select="//OptionalControl/Irregular/StartingPoint/StepSequence/@Value"/>
        </xsl:with-param>
      </xsl:call-template>
      <xsl:call-template name="StepAxis"/>
      <xsl:call-template name="StepDir"/>
      <xsl:call-template name="StepParam"/>
    </tr>
  </table>
</xsl:template>

<xsl:template match="//OptionalControl/Irregular/EndingPoint/StepSequence">
  <xsl:call-template name="StepErr">
    <xsl:with-param name="PElem">
      <xsl:value-of select="//OptionalControl/Irregular/EndingPoint/StepSequence/@Value"/>
    </xsl:with-param>
    <xsl:with-param name="CElem">
      <xsl:value-of select="count(//OptionalControl/Irregular/EndingPoint/StepSequence/Step)"/>
    </xsl:with-param>
    <xsl:with-param name="Ptxt">StepSequence</xsl:with-param>
    <xsl:with-param name="Ctxt">Step</xsl:with-param>
  </xsl:call-template>

  <table>
    <tr>
      <xsl:call-template name="StepNum">
        <xsl:with-param name="link">
          <xsl:value-of select="//OptionalControl/Irregular/EndingPoint/StepSequence/@Value"/>
        </xsl:with-param>
      </xsl:call-template>
      <xsl:call-template name="StepAxis"/>
      <xsl:call-template name="StepDir"/>
      <xsl:call-template name="StepParam"/>
    </tr>
  </table>
</xsl:template>
Example source code for XML schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="XY-Table-Motion-Input-Data-File" type="setting"/>
  <xsd:complexType name="setting">
    <xsd:sequence>
      <xsd:element ref="StepSequence"/>
      <xsd:element ref="ParameterSet"/>
      <xsd:element ref="OptionalControl"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="StepSequence">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Step" maxOccurs="30">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element ref="Axis"/>
              <xsd:element ref="Direction"/>
              <xsd:element ref="ParameterNumber"/>
            </xsd:sequence>
            <xsd:attribute name="Number" type="xsd:int" use="required"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="ParameterSet">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Parameter" maxOccurs="20">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element ref="Distance"/>
              <xsd:element ref="Low_Velocity"/>
              <xsd:element ref="Velocity"/>
              <xsd:element ref="Acceleration"/>
              <xsd:element ref="Deceleration"/>
              <xsd:element ref="Ramp_Down_Point"/>
              <xsd:element ref="Multiplier"/>
            </xsd:sequence>
            <xsd:attribute name="Number" type="xsd:int" use="required"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="OptionalControl">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Parameter" maxOccurs="20">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element ref="Distance"/>
              <xsd:element ref="Low_Velocity"/>
              <xsd:element ref="Velocity"/>
              <xsd:element ref="Acceleration"/>
              <xsd:element ref="Deceleration"/>
              <xsd:element ref="Ramp_Down_Point"/>
              <xsd:element ref="Multiplier"/>
            </xsd:sequence>
            <xsd:attribute name="Number" type="xsd:int" use="required"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```
<xs:complexType>
  <xs:sequence>
    <xs:element ref="LayerNumber"/>
    <xs:element ref="Continuous"/>
    <xs:element ref="Irregular"/>
    <xs:element ref="AdditionalMotion"/>
  </xs:sequence>
  <xs:attribute name="Value" type="xsd:int" use="required"/>
</xs:complexType>
</xsd:element>

<xs:element name="Irregular">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="StartingPoint"/>
      <xs:element ref="EndingPoint"/>
    </xs:sequence>
    <xs:attribute name="Value" type="xsd:int" use="required"/>
  </xs:complexType>
</xsd:element>

<xs:element name="StartingPoint">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="StepSequence"/>
    </xs:sequence>
  </xs:complexType>
</xsd:element>

<xs:element name="EndingPoint">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="StepSequence"/>
    </xs:sequence>
  </xs:complexType>
</xsd:element>

<xs:element name="AdditionalMotion">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="StepSequence"/>
    </xs:sequence>
    <xs:attribute name="Value" type="xsd:int" use="required"/>
  </xs:complexType>
</xsd:element>

<xs:element name="LayerNumber">
  <xs:simpleType>
    <xs:restriction base="xsd:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="20"/>
    </xs:restriction>
  </xs:simpleType>
</xsd:element>

<xs:element name="Continuous">
  <xs:simpleType>
    <xs:restriction base="xsd:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xsd:element>

<xs:element name="Axis">
  <xs:simpleType>
    <xs:restriction base="xsd:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xsd:element>

<xs:element name="Direction">
  <xs:simpleType>
    <xs:restriction base="xsd:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xsd:element>

<xs:element name="ParameterNumber">
  <xs:simpleType>
    <xs:restriction base="xsd:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="20"/>
    </xs:restriction>
  </xs:simpleType>
</xsd:element>
Java source code for X-Y table motion simulator applet

```java
package xytable;

import java.applet.*;
import java.io.*;
import java.net.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;

/**
 * Coordination in 3D space.
 * All unit are followed with SI unit.
 */
```
might be uncomfortable on this axis configuration.
But this is coming from a situation that XY surface is assigned
to fit the monitor surface typically.

Coordination in the monitor. <p>
3D object file format : wavefront.obj <p>
References:
- Jdk***\demo\applets\WireFrame****.java.<br>
  - (http://codezine.jp/a/article.aspx?id=30).<br>
  - Joseph O'Neil. "Teach Yourself Java(独習 Java)". Japanese edition,<br>
  - ISBN 4-88135-748-4 C3058. SHUEISHA CO. Ltd. 1999.<br>
  - Yoshito Yamamoto, "Introduction to graphics processing by Java(Javaによる図形処理入門)",<br>
  - ISBN 4-7692-0399-3 C3058, Kougakutosho Ltd. Japan, 1998.<br>
  - to do: 1:other settings like "irre", "add" and "keystrokes"....

public class XYTEmu
extends Applet implements ActionListener, MouseListener, MouseMotionListener, ItemListener,
AdjustmentListener, Runnable {

/** Thread for the animation of XY table motion. */
private volatile Thread thread = null;
/** Boolean value for thread.resume and suspend. */
private volatile boolean threadSuspended = true;
/** Boolean value to check xy table motion. */
private boolean motionStarted = false;
/** XY table controller. */
private XY_XYtableControler XYtable;
/** Duration time of simulation. [second] */
private double time_d = 0.0;
/** Right side image width. [pixel] */
private int image_3D_width = 400;
/** Right side image height. [pixel] */
private int image_3D_height = 250;
/** Left side image width. [pixel] */
private int image_2D_width = 300;
/** Left side image height. [pixel] */
private int image_2D_height = 250;
/** Conversion for [degree] to [radian]. */
private final double RAD = Math.PI / 180.0;
/** Conversion for [inch] to [meter]. (actually, [step] to [meter]) */
private final double itom = 0.0254 / 2000.0;
/** Rotation angle along with X axis in 3D model. [rad] */
private double Rotated_phi = 0.0;
/** Rotation angle along with Y axis in 3D model. [rad] */
private double Rotated_theta = 0.0;
/** Magnification of rotation from a mouse motion. [#] */
private double Rotation_mag = 0.01;
/** Interval time for sleep. [ms] */
private int FPS = 33;
/** Number of points in 3D space (vertex). [#] */
private int point_num;
/** Number of polygons in 3D space. [#] */
private int surface_num;
/** Initial magnification for image. [#] */
private int i_zoom = 300;
/** Magnification of image. [#] */
private int zoom = i_zoom;
/** Central point of right side image area (for 3D). [pixel] */
private View_Point2D center3D;
/** Point where mouse has pressed. [pixel] */
private View_Point2D pressed;
/** Initial position of drawing. [pixel] */
private View_Point2D init;
/** Array which retains positions for xy table motion pattern. [step] */
private View_Point3D[] pattern_m;
/** Array which retains positions for deposit. [m] */
/** Values (x and z) are inverted from motion pattern against the origin, */
/** and converted from [step] to [m]. */
private View_Point3D[] pattern_s;
/** Array which retains positions for deposit after some processes. [#] */
private View_Point3D[] pattern_v;
/** Array which retains positions of 3D objects in 3D space. [m] */
private View_Point3D[] Real_Point3D;
/** Array which retains positions of 3D objects in 3D space after some processes. [m] */
private View_Point3D[] Virtual_Point3D;
/** Array which retains <code>View_Triangle</code>s for triangular surface in 3D space. */
private View_Triangle[] Real_Surface;
/** Array which retains <code>View_Triangle</code>s for triangular surface */
/** in 3D space after some processes. */
private View_Triangle[] Virtual_surface;
/** Point for the substrate center in 3D space. [m] */
private View_Point3D markerRPoint3D;
/** Point for the substrate center after some processes. [m] */
private View_Point3D markerVPoint3D;
/** Central point with the substrate surface height in 3D space. [p] */
private View_Point3D originRPoint3D;
/** Central point with the substrate surface height after some processes. [m] */
private View_Point3D originVPoint3D;
/** Off-screen image. (for double buffer) */
private Image dblb_3D_image = null;
/** Off-screen image. (for double buffer) */
private Image dblb_2D_image = null;
/** Main graphics. */
private Graphics main_g = null;
/** Double buffer for 2D. (left side) */
private Graphics dblb_3D_g = null;
/** Double buffer for 3D. (right side) */
private Graphics dblb_2D_g = null;
/** Distance from viewpoint to the origin. [m] */
private double proj_d = 3.0;
/** Distance from viewpoint to projected plane. [m] */
private double proj_s = 2.5;
/** Background color. */
private Color bkgColor = Color.white;
/** String color. */
private Color stringColor = Color.black;
/** Spray stream color. */
private Color streamColor = Color.red;
/** Deposit color. */
private Color depositColor = Color.magenta;
/** Button for XY table motion. */
/** Check to change drawing property. */
private Scrollbar view_ZOOM, view_X, view_Y, view_FPS,
        view_AH, view_AS, view_Aa, view_BH, view_BS, view_Ba,
        view_CH, view_CS, view_Ca, view_SH, view_SS, view_Sa;
/** Checkbox to change perspective mode. */
private Checkbox view_perspective;
/** Checkbox to show sprays. */
private Checkbox view_spraypattern;
/** Boolean value to show the motion pattern. */
private boolean view_motionpattern = true;
/** Boolean value to show the deposition. */
private boolean view_deposition = true;
/** Boolean value to show wire-frames of 3D objects. */
private boolean view_wireframe = false;
/** Boolean value to show polyons of 3D objects. */
private boolean view_polygon = true;
/** Instance for the label. */
private Label label_axisX, label_axisY, label_zoom, label_fps,
        label_AH, label_AS, label_Aa, label_BH, label_BS, label_Ba,
        label_CH, label_CS, label_Ca, label_SH, label_SS, label_Sa;
/** String for the input file name of XY table motion. */
private String motion_FileName = null;
/** String for the label. */
private String s_fps, s_axisX, s_axisY, s_zoom, s_motion_start, s_motion_reset;
/** Text area on the applet. */
private TextArea txt;
/** Instance to print out messages. */
private View_Print message;
/** Information set for 3D objects. */
private View_3DObjectInfoSet info;
/** Color set */
/** A axis group */
private View_ColorSet colorA = new View_ColorSet( 0.3f, 0.6f, 0.5f, 150 );
/** B axis group */
private View_ColorSet colorB = new View_ColorSet( 0.2f, 0.6f, 0.5f, 255 );
/** Chamber group */
private View_ColorSet colorC = new View_ColorSet( 0.5f, 0.15f, 0.5f, 255 );
/** Substrate group */
private View_ColorSet colorS = new View_ColorSet( 0.4f, 0.6f, 0.5f, 100 );
/** Multiplier of color value for scrollbar adjustment */
private double cMul = 100;
public void init() {
  center3D = new View_Point2D( image_3D_width / 2.0, image_3D_height / 2.0 );
  pressed = new View_Point2D( center3D.x, center3D.y );
  init = new View_Point2D( center3D.x, center3D.y );
  s_fps = "<INTERVAL[ms]>";
  s_axisX = "<AXIS_X>";
  s_axisY = "<AXIS_Y>";
  s_zoom = "<ZOOM>";
  s_motion_start = "Start/Suspend";
  s_motion_reset = "Reset/ReCalc";
  markerVPoint3D = new View_Point3D();
  originRPoint3D = new View_Point3D();
  originVPoint3D = new View_Point3D();
  /** Set 3D object information. */
  /** Set into the array, and set color properties. Data themselves are set in getObjectData(). */
  set3DObjectInfo();
  /** Set layout of the applet panel. */
  setStyle();
  /** message is better to be newed after setStyle(), because txt is set in that method. */
  message = new View_Print( txt );
  try {
    if ( motion_FileName != null || motion_FileName != "" ) {
      getObjectData();
      XYtable = new XY_XYtableController( info, txt );
      XYtable.getMotionData( getDocumentBase(), motion_FileName );
    }
int size = 1 + XYtable.getMainSequenceSize();
/**if the size is 20, then*/
pattern_m = new View_Point3D[size];
pattern_s = new View_Point3D[size];
pattern_v = new View_Point3D[size];
for ( int i = 0; i < size; i++ ) {
  pattern_m[i] = new View_Point3D();
  pattern_s[i] = new View_Point3D( 0, info.subst.getCurrentHeight( Real_Point3D ), 0 );
  pattern_v[i] = new View_Point3D();
  /* each value of .x and .z will be set in XYtable.calcTiming */
}
XYtable.calcTiming( pattern_m, pattern_s );
XYtable.confirmData( pattern_m, pattern_s, pattern_v );
XYtable.init( pattern_s, Real_Point3D, markerRPoint3D );
}
}
catch ( Exception e ) {
  message.printTextBoxN( "@init:" + e + " File:[" + motion_FileName + "]" );
}
adjustPosition(); /* set 3D objects into proper positions.*/
/**
 * Sets information of 3D objects such as 3D object name,
 * file name, color and so on.
 */
private void set3DObjectInfo() {
  info = new View_3DObjectInfoSet();
  set3DObjectColor();
}
/**
 * set3DObjectColor
 */
private void set3DObjectColor() {
  info.chamber.color.setHSBandAlpha( colorC );
  info.axisA.color.setHSBandAlpha( colorA );
  info.gearA.color.setHSBandAlpha( colorA );
  info.motorA.color.setHSBandAlpha( colorA );
  info.axisB.color.setHSBandAlpha( colorB );
  info.gearB.color.setHSBandAlpha( colorB );
  info.motorB.color.setHSBandAlpha( colorB );
  info.subst.color.setHSBandAlpha( colorS );
}
/**
 * Sets layout of Java applet and read parameters from HTML file.
 */
private void setStyle() {
  view_perspective = new Checkbox( "Perspective", true );
  view_spraypattern = new Checkbox( "SprayPattern", true );
  /* Get parameters from <param> tag. */
  info.chamber.filename = getParameter( "chamber_FileName" );
  info.axisA.filename = getParameter( "axis_Filename" );
  info.axisB.filename = getParameter( "axisFileName" );
  info.subst.filename = getParameter( "subtable_FileName" );
  info.gearA.filename = getParameter( "gearbox_FileName" );
}
info.gearB.filename = getParameter( "gearbox_fileName" );
info.motorA.filename = getParameter( "motor_fileName" );
info.motorB.filename = getParameter( "motor_fileName" );
motion_fileName = getParameter( "motion_fileName" );
Rotated_phi = Double.parseDouble( getParameter( "rotation_x_axis" ) ) * RAD; //along X axis
Rotated_theta = Double.parseDouble( getParameter( "rotation_y_axis" ) ) * RAD; //along Y axis
view_perspective.setState( "ON" .equals( getParameter( "Perspective" ) ) ? true : false );
view_perspective..setState( "ON" .equals( getParameter( "Perspective" ) ) ? true : false );
setLayout( new BorderLayout() );
Panel viewMain = new Panel( new GridLayout( 3, 4, 0, 0 ) );
Panel viewSub = new Panel( new BorderLayout() );
Panel viewColor = new Panel( new GridLayout( 24, 1, 0, 0 ) );
viewMain.add( view_perspective );
viewMain.add( view_spraypattern );
viewMain.add( motion_start = new Button( s_motion_start ) );
viewMain.add( motion_stop  = new Button(" stop");
viewMain.add( motion_reset = new Button( s_ motion_reset ) );
int OriLbl = Label.CENTER; /* orientation for label */
viewMain.add( label_fps = new Label( s_fps,  OriLbl ) );
viewMain.add( label_axisX = new Label( s_axisX + ( int ) init.x, OriLbl ) );
viewMain.add( label_axisY = new Label( s_axisY + ( int ) init.y, OriLbl ) );
viewMain.add( label_zoom = new Label( s_zoom + ( int ) i_zoom, OriLbl ) );
int OriScr = Scrollbar.HORIZONTAL; /* orientation for scrollbar */
viewMain.add( view_FPS = new Scrollbar( OriScr, FPS, 10, ( int ) ( 1000 / 59.0 ), 1000 + 1 ) ) ;
viewMain.add( view_X = new Scrollbar( OriScr, ( int ) init.x, 10, 0, 700 ) );
viewMain.add( view_Y = new Scrollbar( OriScr, ( int ) init.y, 0, 700 ) );
viewMain.add( view_ZOOM = new Scrollbar( OriScr, i_zoom, 10, ( i_zoom / 10 ), ( i_zoom * 10 ) ) );
viewSub.add( viewMain, "Center" );
viewSub.add( txt = new TextArea( 12, 20 ), "South" );
int cMinH = 0; /* Maximum value of H */
int cMaxH = ( int ) ( 0.999999 * cMul ); /* Minimum value of H */
int cMinS = 0; /* Maximum value of S */
int cMaxS = ( int ) ( 1 * cMul ); /* Minimum value of S */
int cMinAlpha = 0;
int cMaxAlpha = 255;
viewColor.add( label_AH = new Label( "A H", OriLbl ) );
viewColor.add( label_AS = new Label( "A S", OriLbl ) );
viewColor.add( label_Aa = new Label( "A a", OriLbl ) );
viewColor.add( label_BH = new Label( "B H", OriLbl ) );
viewColor.add( label_BS = new Label( "B S", OriLbl ) );
viewColor.add( label_Ba = new Label( "B a", OriLbl ) );
viewColor.add( label_CH = new Label( "C H", OriLbl ) );
viewColor.add( label_CS = new Label( "C S", OriLbl ) );
viewColor.add( label_Ca = new Label( "C a", OriLbl ) );
viewColor.add( label_SH = new Label( "S H", OriLbl ) );
viewColor.add( label_SS = new Label( "S S", OriLbl ) );
viewColor.add( label_Sa = new Label( "S a", OriLbl ) );
resetColorScrValueSet();
add( viewColor, "East" );
add( viewSub, "South" );
viewColor.setVisible( true );
view_FPS.addAdjustmentListener( this );
view_X.addAdjustmentListener( this );
view_Y.addAdjustmentListener( this );
view_ZOOM.addAdjustmentListener( this );
motion_start.addActionListener( this );
// motion_stop.addActionListener(this);
motion_reset.addActionListener( this );
view_AH.addAdjustmentListener( this );
view_AS.addAdjustmentListener( this );
view_Aa.addAdjustmentListener( this );
view_AH.addAdjustmentListener( this );
view_BH.addAdjustmentListener( this );
view_SS.addAdjustmentListener( this );
view_Ba.addAdjustmentListener( this );
view_CH.addAdjustmentListener( this );
view_CS.addAdjustmentListener( this );
view_Ca.addAdjustmentListener( this );
view_SH.addAdjustmentListener( this );
view_SS.addAdjustmentListener( this );
view_Sa.addAdjustmentListener( this );
addMouseListener( this );
addMouseMotionListener( this );
setBackground( Color.white );
// Applet resize();
// resize(image_width, image_height);
public void start() {
  if ( thread == null ) {
    thread = new Thread( this );
    thread.start();
  }
}
public void run() {
  Thread thisThread = Thread.currentThread();
  while ( thread == thisThread ) {
    try {
      thread.currentThread().sleep( FPS );
      // Rotated_theta += 0.01;
      if ( motionStarted && motion_FileName != null ) {
        if ( XYtable.getObjPosition( time_d ) ) {
          threadSuspended = true;
          motionStarted = false;
          time_d = 0.0;
          XYtable.reset();
        } else {
          time_d += FPS * speed;
        }
      }
      repaint();
      synchronized ( this ) {
        while ( threadSuspended && thread == thisThread ) {
          wait();
        }
      }
      catch ( InterruptedException e ) {
        break;
      }
    }
    catch ( InterruptedException e ) {
      break;
    }
  }
  public synchronized void stop() {
    thread = null;
    notify();
  }
  public void mousePressed( MouseEvent e ) {
    pressed.x = e.getX();
    pressed.y = e.getY();
  }
  public void mouseDragged( MouseEvent e ) {
    int dx = e.getX() - ( int ) pressed.x;
    int dy = e.getY() - ( int ) pressed.y;
    Rotated_theta += dx * Rotation_mag;
    Rotated_phi += dy * Rotation_mag;
    init.y = view_Y.getValue();
    if ( motion_FileName != null ) {
      adjustPosition();
    }
  }
  label_axisX.setText( s_axisX + view_X.getValue() );
  label_axisY.setText( s_axisY + view_Y.getValue() );
  if ( motion_started && motion_FileName != null ) {
    if ( Rotated_phi > 90 * RAD ) {
      Rotated_phi = 90 * RAD;
    } else if ( Rotated_phi < 5 * RAD ) {
      Rotated_phi = 5 * RAD;
    } else if ( Rotated_phi < 0 ) {
      Rotated_phi = 0.0;
    }
    label_axisX.setText( s_axisX + view_X.getValue() );
    label_axisY.setText( s_axisY + view_Y.getValue() );
  }
if(Rotated_theta < -90*RAD) {Rotated_theta = -90*RAD;}
if(Rotated_theta > 90*RAD) {Rotated_theta = 90*RAD;}
update(main_g);
drawObject();
pressted.x = e.getX();
pressted.y = e.getY();
}

class NumericApp extends Applet {
  private static final long serialVersionUID = 1L;
  private static final double RAD = Math.PI / 180;

  private synchronized void actionPerformed(ActionEvent e) {
    if (e.getSource() == motion_start) {
      threadSuspended = !threadSuspended;
      notify();
      motionStarted = true;
    } else if (e.getSource() == motion_stop) {
      threadSuspended = true;
    } else if (e.getSource() == motion_reset) {
      time_d = 0.0; /* reset the time. */
      XYtable.reset(); /* reset positions for XY table bodies. */
      XYtable.setSeqCount(0);
      reset();
      update(main_g);
      threadSuspended = true;
    }
    message.printTextBoxN("\[" + e.getActionCommand() + "]");
  }

  public void adjustmentValueChanged(AdjustmentEvent e) {
    FPS = view_FPS.getValue();
    zoom = view_ZOOM.getValue();
    if (e.getSource() == view_ZOOM) {
      adjustPosition();
    }
    init.x = view_X.getValue();
    init.y = view_Y.getValue();
    label_fps.setText(s_fps + view_FPS.getValue());
    label_axisX.setText(s_axisX + view_X.getValue());
    label_axisY.setText(s_axisY + view_Y.getValue());
    label_zoom.setText(s_zoom + view_ZOOM.getValue());
    if (e.getSource() == view_AH) {
      info.axisA.color.H = info.gearA.color.H = info.motorA.color.H = (float) (view_AH.getValue() / cMul);
    } else if (e.getSource() == view_AS) {
      info.axisA.color.S = info.gearA.color.S = info.motorA.color.S = (float) (view_AS.getValue() / cMul);
    } else if (e.getSource() == view_Aa) {
      info.axisA.color.alpha = info.gearA.color.alpha = info.motorA.color.alpha = view_Aa.getValue();
    } else if (e.getSource() == view_BH) {
      float h = (float) (view_BH.getValue() / cMul);
      info.axisB.color.H = info.gearB.color.H = info.motorB.color.H = (float) (view_BH.getValue() / cMul);
    } else if (e.getSource() == view_BS) {
      info.axisB.color.S = info.gearB.color.S = info.motorB.color.S = (float) (view_BS.getValue() / cMul);
    } else if (e.getSource() == view_Ba) {
      info.axisB.color.alpha = info.gearB.color.alpha = info.motorB.color.alpha = view_Ba.getValue();
    } else if (e.getSource() == view_CH) {
      info.chamber.color.H = (float) (view_CH.getValue() / cMul);
    } else if (e.getSource() == view_CS) {
      info.chamber.color.S = (float) (view_CS.getValue() / cMul);
    } else if (e.getSource() == view_Ca) {
S 232S
663|      info.chamber.color.alpha = view_Ca.getValue();
664|    } else if ( e.getSource() == view_SH ) {
665|      info.subst.color.H = ( float ) ( view_SH.getValue() / cMul );
666|    } else if ( e.getSource() == view_SS ) {
667|      info.subst.color.S = ( float ) ( view_SS.getValue() / cMul );
668|    } else if ( e.getSource() == view_Sa ) {
669|      info.subst.color.alpha = view_Sa.getValue();
670|    }
671|    // repaint();
672|    drawObject();
673|  }
674|
675|  public void update( Graphics g ) {
676|    paint( g );
677|  }
678|
679|  public void paint( Graphics g ) {
680|    if ( dblb_3D_image == null ) {
681|      dblb_3D_image = createImage( image_3D_width, image_3D_height );
682|      dblb_3D_g = dblb_3D_image.getGraphics();
683|      dblb_2D_image = createImage( image_2D_width, image_2D_height );
684|      dblb_2D_g = dblb_2D_image.getGraphics();
685|      main_g = getGraphics();
686|    }
687|    ClearScreen();
688|    // drawObject();
689|    if ( motion_FileName != null ) {
690|      drawObject();
691|    } else {
692|      g.setColor( stringColor );
693|      g.drawString( "No Input File Parameter", 10, 80 );
694|    }
695|  }
696|
697|  private void reset() {
698|    Rotated_phi = Double.parseDouble( getParameter( "rotation_x_axis" ) ) * RAD; /* along X axis */
699|    Rotated_theta = Double.parseDouble( getParameter( "rotation_y_axis" ) ) * RAD; /* along Y axis */
700|    zoom = i_zoom; /* set to initial zoom number */
701|    view_ZOOM.setValue( i_zoom );
702|    label_zoom.setText( "ZOOM: " + ( int ) i_zoom );
703|    /* set to initial drawing position */
704|    adjustPosition();
705|    label_axisX.setText( "AXIS-X:" + ( int ) init.x );
706|    label_axisY.setText( "AXIS-Y:" + ( int ) init.y );
707|    motionStarted = false;
708|    set3DObjectColor();
709|    resetColorScrValueSet(); /* reset positions of scrollbars */
710|  }
711|
712|  private void resetColorScrValueSet() {
713|    resetColorScrValue( view_AH, view_AS, view_Aa, colorA ); /* For group A-axis */
714|    resetColorScrValue( view_BH, view_BS, view_Ba, colorB ); /* For group B-axis */
715|    resetColorScrValue( view_CH, view_CS, view_Ca, colorC ); /* For group Chamber */
716|    resetColorScrValue( view_SH, view_SS, view_Sa, colorS ); /* For group Substrate */
717|  }
718|
719|  private void resetColorScrValue( Scrollbar H, Scrollbar S, Scrollbar a, View_ColorSet c ) {
720|    H.setValue( ( int ) ( c.H * cMul ) );
721|    S.setValue( ( int ) ( c.S * cMul ) );
722|    a.setValue( c.alpha );
723|  }
724|
725|  // resetColorScrValue
726|  /* adjustObjectPosition sets 3D objects into proper positions. */
727|  private void adjustObjectPosition() {
728|    /* set position for Axis B */
729|    info.axisB.setOffset( Real_Point3D, 0, 0, 0, -info.axisB.width / 2.0, 0.01, -info.axisB.length / 2.0 );
730|    /* set position for Gear Box */
731|    info.gearB.setOffset( Real_Point3D, 0, 0, 0, -info.gearB.width / 2.0, 0.01, -info.gearB.height / 2.0 );
732|  }
733|
734|  /* adjustObjectPosition sets 3D objects into proper positions. */
735|  private void adjustObjectPosition() {
736|    /* set position for Axis B */
737|    info.axisB.setOffset( Real_Point3D, 0, 0, 0, -info.axisB.width / 2.0, 0.01, -info.axisB.length / 2.0 );
738|    /* set position for Gear Box */
739|    info.gearB.setOffset( Real_Point3D, 0, 0, 0, -info.gearB.width / 2.0, 0.01, -info.gearB.height / 2.0 );
740|  }
741|
742|  /* adjustObjectPosition sets 3D objects into proper positions. */
743|  private void adjustObjectPosition() {
744|    /* set position for Axis B */
745|    info.axisB.setOffset( Real_Point3D, 0, 0, 0, -info.axisB.width / 2.0, 0.01, -info.axisB.length / 2.0 );
746|    /* set position for Gear Box */
747|    info.gearB.setOffset( Real_Point3D, 0, 0, 0, -info.gearB.width / 2.0, 0.01, -info.gearB.height / 2.0 );
748|  }
/* set position for Gear Box B */
info.motorB.setOffset( Real_Point3D, 90 * RAD, 0, 0, 0,
                    info.axisB.height / 2.0 + 0.01,
                    info.axisB.length / 2.0 + info.gearB.length );

/* set position for Axis A */
info.axisA.setOffset( Real_Point3D, 0, 90 * RAD, 0, -info.axisA.length / 2.0,
                   info.axisB.getCurrentHeight( Real_Point3D ) + 0.02,
                   info.axisA.width / 2.0 );

/* set position for Gear Box A */
info.gearA.setOffset( Real_Point3D, 0, -90 * RAD, 0, -info.axisA.length / 2.0,
                     info.axisB.getCurrentHeight( Real_Point3D ) + info.axisA.height - info.gearA.height / 2.0 + 0.02,
                     -info.gearA.width / 2.0 );

/* set position for Stepping Motor A */
info.motorA.setOffset( Real_Point3D, 0, 0, 90 * RAD,
                    -info.axisA.length / 2.0 - info.gearA.length,
                    info.axisB.getCurrentHeight( Real_Point3D ) + info.axisA.height / 2.0 + 0.02, 0 );

/* set position for substrate table */
info.subst.setOffset( Real_Point3D, 0, 0, 0, -info.subst.width / 2.0,
                   info.axisA.getCurrentHeight( Real_Point3D ) + 0.06,
                   -info.subst.length / 2.0 );

/**
 * adjustPosition adjusts the drawing position to make the position,
 * where the stream hits to the substrate, into the center of the image area.
 * 
 * private void adjustPosition()
 * {
 * markerVPoint3D.rotateRef( Rotated_phi, Rotated_theta, markerRPoint3D );
 * if ( view_perspective.getState() ) {
 *   markerVPoint3D.changePerspective( proj_s, proj_d );
 * }
 * view_Y.setValue( ( int ) ( center3D.y + markerVPoint3D.y * zoom ) );
 * view_X.setValue( ( int ) center3D.x );
 * init.x = view_X.getValue();
 * init.y = view_Y.getValue();
 * }
 * 
 * ClearScreen fills out the images with the background color.
 * 
 * private void ClearScreen() {
 *   dblb_3D_g.setColor( bkgColor );
 *   dblb_3D_g.fillRect( 0, 0, image_3D_width, image_3D_height );
 *   dblb_2D_g.setColor( bkgColor );
 *   dblb_2D_g.fillRect( 0, 0, image_2D_width, image_2D_height );
 * }
 * 
 * drawObject draws XY table objects into image areas for 2D and 3D.
 * 
 * private void drawObject() {
 *   int[] x = new int[3];
 *   int[] y = new int[3];
 *   ClearScreen();
 *   rotate( Rotated_phi, Rotated_theta, Real_Point3D, Virtual_Point3D );
 *   if ( view_perspective.getState() ) {
 *     changePerspective( proj_s, proj_d, Virtual_Point3D );
 *   }
 *   int[] frontSurface = getFrontSurface();
 *   sortByDepth( frontSurface );
 *   sortWithObjects( frontSurface );
 *   setColor( c, frontSurface );
 * }
/* Axis lines along X and Y. */
dblb_3D_g.setColor( Color.black );
dblb_3D_g.drawLine( ( int ) init.x, ( int ) init.y, ( int ) ( 100 + init.x ), ( int ) init.y );
dblb_3D_g.drawLine( ( int ) init.x, ( int ) init.y, ( int ) init.x, ( int ) ( -100 + init.y ) );
for ( int i = 0; i < frontSurface.length; i++ ) {
    dblb_3D_g.setColor( c[i] );
    for ( int j = 0; j < 3; j++ ) {
        x[j] = ( int ) Math.rint( Virtual_surface[frontSurface[i]].tri[j].x * zoom + init.x );
        y[j] = ( int ) Math.rint( -Virtual_surface[frontSurface[i]].tri[j].y * zoom + init.y );
    }
    if ( view_polygon ) {
        dblb_3D_g.fillPolygon( x, y, 3 );
    }
    if ( view_wireframe ) {
        dblb_3D_g.setColor( Color.black );
        dblb_3D_g.drawPolygon( x, y, 3 );
    }
    if ( view_deposition ) {
        drawDeposition3D();
    }
    main_g.drawImage( dblb_3D_image, 300, 0, this ); /* Draw 3D image into main graphic area. */
    drawPosition(); /* 2D image from top of xy table */
}
/**
 * sortByDepth
 *
 * @param f int[]
 */
private void sortByDepth( int[] f ) {
    int temp;
    int lim = f.length - 1;
    do {
        int m = 0;
        for ( int n = 0; n < lim; n++ ) {
            if ( Virtual_surface[f[n]].getDepth() > Virtual_surface[f[n + 1]].getDepth() ) {
                temp = f[n];
                f[n] = f[n + 1];
                f[n + 1] = temp;
                m = n;
            }
        }
        lim = m;
    } while ( lim != 0 );
/**
 * getFrontSurface
 *
 * @return int[]
 */
private int[] getFrontSurface() {
    int n = 0;
    int[] surface = new int[surface_num];
    double a1, a2, a3, b1, b2, b3;
    for ( int i = 0; i < surface.length; i++ ) {
        a1 = Virtual_surface[i].tri[1].x - Virtual_surface[i].tri[0].x;
        a2 = Virtual_surface[i].tri[1].y - Virtual_surface[i].tri[0].y;
        a3 = Virtual_surface[i].tri[1].z - Virtual_surface[i].tri[0].z;
        b1 = Virtual_surface[i].tri[2].x - Virtual_surface[i].tri[0].x;
        b2 = Virtual_surface[i].tri[2].y - Virtual_surface[i].tri[0].y;
        b3 = Virtual_surface[i].tri[2].z - Virtual_surface[i].tri[0].z;
        Virtual_surface[i].nz = a1 * b2 - a2 * b1;
        if ( Virtual_surface[i].nz > 0 ) {
Virtual_surface[i].nx = a2 * b3 - a3 * b2;
Virtual_surface[i].ny = a3 * b1 - a1 * b3;
surface[n] = i;

n++;
int[] frontSurface = new int[n];
for (int i = 0; i < n; i++) {
    frontSurface[i] = surface[i];
}
return frontSurface;

/**
 * sortByObject
 * Right now this is depending on View_3DObjectInfoSet
 * @param frontSurface int[]
 */
private void sortByObject(int[] frontSurface) {
    int[][] tmp;
    tmp = info.getGroupSize();
    int[] count = new int[tmp.length];
    for (int i = 0; i < count.length; i++) {
        count[i] = 0;
    }
    for (int i = 0; i < frontSurface.length; i++) {
        int faceIndex = frontSurface[i];
        for (int groupIndex = 0; groupIndex < count.length; groupIndex++) {
            if (info.withinGroup(groupIndex, faceIndex)) {
                tmp[groupIndex][count[groupIndex]] = faceIndex;
            }
        }
    }
    for (int maxArrayIndex = 0; maxArrayIndex < count.length; maxArrayIndex++) {
        int totalcount = 0;
        for (int arrayIndex = 0; arrayIndex < maxArrayIndex; arrayIndex++) {
            totalcount += count[arrayIndex];
        }
        for (int i = 0; i < count[maxArrayIndex]; i++) {
            frontSurface[totalcount + i] = tmp[maxArrayIndex][i];
        }
    }
}

/**
 * setColor
 * @param c Color[]
 * @param frontSurface int[]
 */
private void setColor(Color[] c, int[] frontSurface) {
    for (int i = 0; i < frontSurface.length; i++) {
        int n = frontSurface[i];
        double l = Math.sqrt(Virtual_surface[n].nx * Virtual_surface[n].nx +
                              Virtual_surface[n].ny * Virtual_surface[n].ny +
                              Virtual_surface[n].nz * Virtual_surface[n].nz);
        float B = (float) (Virtual_surface[n].nz / l);
        if (B < 0) {
            B = 0;
        } else if (B > 1) {
            B = 1;
        }
        for (int j = 0; j < info.getIndexSize(); j++) {
            if (info.index(j).withinF(n)) {
                if (info.index(j).color.alpha < 255) {
                } else {
                    c[i] = Color.getHSBColor(info.index(j).color.H, info.index(j).color.S, B);
                }
            }
        }
    }
}
private void drawDeposition3D() {
    int seq_count = XYtable.getSeqCount();
    if (seq_count < 1) {
        seq_count = 1;
    }
    rotate(Rotated_phi, Rotated_theta, pattern_s, pattern_v);
    originVPoint3D.rotateRef(Rotated_phi, Rotated_theta, originRPoint3D);
    if (view_perspective.getState()) {
        changePerspective(proj_s, proj_d, pattern_v);
        originVPoint3D.changePerspective(proj_s, proj_d);
    }
    dblb_3D_g.setColor(depositColor);
    for (int i = 1; i < seq_count; i++) {
        dblb_3D_g.drawLine((int) Math.rint(pattern_v[i - 1].x * zoom + init.x),
                          (int) Math.rint(-pattern_v[i - 1].y * zoom + init.y),
                          (int) Math.rint(pattern_v[i].x * zoom + init.x),
                          (int) Math.rint(-pattern_v[i].y * zoom + init.y));
    }
    dblb_3D_g.drawLine((int) Math.rint(pattern_v[seq_count - 1].x * zoom + init.x),
                        (int) Math.rint(-pattern_v[seq_count - 1].y * zoom + init.y),
                        (int) Math.rint(originVPoint3D.x * zoom + init.x),
                        (int) Math.rint(-originVPoint3D.y * zoom + init.y));
    if (motionStarted) {
        dblb_3D_g.setColor(streamColor);
        dblb_3D_g.drawLine((int) init.x, 0,
                          (int) Math.rint(originVPoint3D.x * zoom + init.x),
                          (int) Math.rint(-originVPoint3D.y * zoom + init.y));
    }
}

private void drawPosition() {
    int[] B_x = new int[4]; int[] B_z = new int[4];
    int[] A_x = new int[4]; int[] A_z = new int[4];
    int[] S_x = new int[4]; int[] S_z = new int[4];
    double x, z;
    x = image_2D_width / 2.0;
    z = image_2D_height / 2.0;
    for (int i = 0; i < 4; i++) {
        B_x[i] = (int) Math.rint(info.axisB.fourCorner[i].x * zoom + x);
        B_z[i] = (int) Math.rint(info.axisB.fourCorner[i].z * zoom + z);
        A_x[i] = (int) Math.rint(info.axisA.fourCorner[i].x * zoom + x);
        A_z[i] = (int) Math.rint(info.axisA.fourCorner[i].z * zoom + z);
        S_x[i] = (int) Math.rint(info.subst.fourCorner[i].x * zoom + x);
        S_z[i] = (int) Math.rint(info.subst.fourCorner[i].z * zoom + z);
    }
    dblb_2D_g.setColor(Color.white);
    dblb_2D_g.fillPolygon(B_x, B_z, 4);
    dblb_2D_g.setColor(Color.black);
    dblb_2D_g.drawPolygon(B_x, B_z, 4);
    dblb_2D_g.setColor(Color.white);
    dblb_2D_g.fillPolygon(A_x, A_z, 4);
    dblb_2D_g.setColor(Color.black);
    dblb_2D_g.drawPolygon(A_x, A_z, 4);
    dblb_2D_g.setColor(Color.white);
    dblb_2D_g.fillPolygon(S_x, S_z, 4);
    dblb_2D_g.setColor(Color.black);
    dblb_2D_g.drawPolygon(S_x, S_z, 4);
    double phi = 50;
    double theta = 50;
    phi = phi / 180 * Math.PI;
    theta = theta / 180 * Math.PI;
    double x = Math.cos(theta) * Math.sin(phi);
    double y = Math.cos(phi);
    double z = Math.sin(theta) * Math.sin(phi);
    x = x / 1000;
    y = y / 1000;
    z = z / 1000;
    dblb_2D_g.setColor(Color.red);
    gdb2D_g.drawLine((int) x, (int) y, (int) z);
    dblb_2D_g.setColor(Color.green);
    gdb2D_g.drawLine((int) x, (int) y, (int) z);
}
/** Marker */
dblb_2D_g.setColor( Color.black );
int rad = (int) (0.002 * zoom);
dblb_2D_g.fillOval(int markerRPoint3D.x * zoom + x - rad,
(int) (markerRPoint3D.z * zoom + z - rad), 2 * rad, 2 * rad);

/** Motion pattern */
if (view_motionpattern) {
dblb_2D_g.setColor(Color.red);
for (int i = 1; i < pattern_m.length; i++) {
dlb_2D_g.drawLine(int pattern_m[i - 1].x * c + x,
(int) (pattern_m[i - 1].z * c + z),
(int) (pattern_m[i].x * c + x), (int) (pattern_m[i].z * c + z));
}
}

/** Deposition pattern */
if (view_spraypattern.getState()) {
dblb_2D_g.setColor(Color.orange);
for (int i = 1; i < pattern_m.length; i++) {
dlb_2D_g.drawLine(int -pattern_m[i - 1].x * c + x,
(int) (-pattern_m[i - 1].z * c + z),
(int) (-pattern_m[i].x * c + x), (int) (-pattern_m[i].z * c + z));
}
}

/** deposit */
if (view_deposition) {
int seq_count = XYtable.getSeqCount();
dlb_2D_g.setColor(Color.magenta);
if (seq_count < 1) {
seq_count = 1;
}

for (int i = 1; i < seq_count; i++) {
dlb_2D_g.drawLine(int pattern_s[i - 1].x * zoom + x,
(int) pattern_s[i - 1].z * zoom + z,
(int) pattern_s[i].x * zoom + x,
(int) pattern_s[i].z * zoom + z);
}

dlb_2D_g.drawString("ON PROCESSING", 80, 20);
main_g.drawImage(dlb_2D_image, 0, 0, this);
}

/** getWaveFrontData reads 3D object data (WaveFront.obj type) and set some information to <code>3DObjectInfo</code>. */
private void getWaveFrontData(Vector v_main, Vector f_main, View_3DObjectInfo obj3D) {
Vector v_vec = new Vector();
Vector f_vec = new Vector();
try {
InputStream is = new URL(getDocumentBase(), obj3D.filename).openStream();
Reader r = new BufferedReader(new InputStreamReader(is));
StreamTokenizer st = new StreamTokenizer(r);
int token;
double[] min = new double[3];
double[] max = new double[3];
boolean c = true;
}
while ( ( token = st.nextToken() ) != StreamTokenizer.TT_EOF ) {
    if ( token == StreamTokenizer.TT_WORD ) {
        if ( st.sval.equals( "v" ) ) {
            double[] v = new double[3];
            st.nextToken(); v[0] = (double) st.nval;
            st.nextToken(); v[1] = (double) st.nval;
            st.nextToken(); v[2] = (double) st.nval;
            v_vec.addElement( v );
            v_main.addElement( v );
            if ( c ) {
                for ( int i = 0; i < 3; i++ ) {
                    min[i] = v[i]; max[i] = v[i];
                }
                c = false;
            }
            for ( int i = 0; i < 3; i++ ) {
                if ( min[i] >= v[i] ) {
                    min[i] = v[i];
                }
                if ( max[i] <= v[i] ) {
                    max[i] = v[i];
                }
            }
            obj3D.width = max[0] - min[0];
            obj3D.height = max[1] - min[1];
            obj3D.length = max[2] - min[2];
        } else if ( st.sval.equals( "f" ) ) {
            int[] f = new int[3];
            st.nextToken(); f[0] = (int) st.nval + (v_main.size() - v_vec.size());
            st.nextToken(); f[1] = (int) st.nval + (v_main.size() - v_vec.size());
            st.nextToken(); f[2] = (int) st.nval + (v_main.size() - v_vec.size());
            f_vec.addElement( f );
            f_main.addElement( f );
            obj3D.v_start = v_main.size() - v_vec.size();
            obj3D.v_end = v_main.size() - 1;
            obj3D.face_start = f_main.size() - f_vec.size();
            obj3D.face_end = f_main.size() - 1;
        }
    }
    else if ( st.sval.equals( "n" ) ) {
        //...
double[] v = (double[]) v_vec.elementAt(i);

Real_Point3D[i] = new View_Point3D(v[0], v[1], v[2]);
Virtual_Point3D[i] = new View_Point3D(v[0], v[1], v[2]);

info.axisB.setFourCorners(Real_Point3D);
info.axisA.setFourCorners(Real_Point3D);
info.subst.setFourCorners(Real_Point3D);

info.setGroup();

/**
 * to use "infoSubst.getCurrentHeight()",
 * adjustObjectPosition() is required to call previously
 */

markerRPoint3D = new View_Point3D(0, info.subst.getCurrentHeight(Real_Point3D), 0);

for (int i = 0; i < surface_num; i++) {
    int[] f = (int[]) f_vec.elementAt(i);
    /**
     * index number of array will be subtract one,
     * because in Wave Front format, index number of vertex is counted from 1~, not from 0.
     */
    Real_Surface[i] = new View_Triangle(Real_Point3D[f[0] - 1], Real_Point3D[f[1] - 1], Real_Point3D[f[2] - 1]);
    Virtual_surface[i] = new View_Triangle(Virtual_Point3D[f[0] - 1], Virtual_Point3D[f[1] - 1], Virtual_Point3D[f[2] - 1]);
}

message.printTextBoxN("[" + point_num + "]" + surface_num + "]");

for (int i = 0; i < info.getIndexSize(); i++) {
    message.printTextBoxN(info.index(i).label + "[" + info.index(i).vsize() + ""]");
}

/**
 * rotate makes 3D objects rotation with angle of <code>phi</code> and <code>theta</code>.
 * @param phi double. Rotation angle around X axis. [radian]
 * @param theta double. Rotation angle around Y axis. [radian]
 * @param r View_Point3D[]. Array of original 3D point data.
 * @param v View_Point3D[]. Array of rotated 3D point data.
 */
private void rotate(double phi, double theta, View_Point3D[] r, View_Point3D[] v) {
    for (int i = 0; i < r.length; i++) {
        v[i].rotateRef(phi, theta, r[i]);
    }
}

/**
 * changePerspective gives a transformation to give perspective image
 * of 3D object.
 * @param s double. Distance from viewpoint to projected plane. [m]
 * @param d double. Distance from viewpoint to the origin. [m]
 * @param v View_Point3D[]. Array of transformed 3D point data.
 */
private void changePerspective(double s, double d, View_Point3D[] v) {
    for (int i = 0; i < v.length; i++) {
        v[i].changePerspective(s, d);
    }
}
private View_3DObjectInfo[][] group;

/** View_3DObjectInfoSet */
public View_3DObjectInfoSet()
{
  index = new View_3DObjectInfo[8];

  index[0] = chamber = new View_3DObjectInfo( "Chamber", "chamber.obj" );
  index[1] = axisA = new View_3DObjectInfo( "A Axis", "axis.obj" );
  index[2] = axisB = new View_3DObjectInfo( "B Axis", "axis.obj" );
  index[3] = subst = new View_3DObjectInfo( "Substrate", "subtable.obj" );
  index[4] = gearA = new View_3DObjectInfo( "Gear Box A", "gearbox.obj" );
  index[5] = gearB = new View_3DObjectInfo( "Gear Box B", "gearbox.obj" );
  index[6] = motorA = new View_3DObjectInfo( "Step Motor A", "motor.obj" );
  index[7] = motorB = new View_3DObjectInfo( "Step Motor B", "motor.obj" );

  public void setGroup()
  {
    int[] A = new int[axisA.fsize() + gearA.fsize() + motorA.fsize()];
    int[] B = new int[axisB.fsize() + gearB.fsize() + motorB.fsize()];
    int[] S = new int[subst.fsize()];
    int[] C = new int[chamber.fsize()];
    int[][] groupSize = new int[4][];
    groupSize[0] = C;
    groupSize[1] = B;
    groupSize[2] = A;
    groupSize[3] = S;

    group = new View_3DObjectInfo[4][];
    View_3DObjectInfo[] gtC = {chamber};
    View_3DObjectInfo[] gtB = {axisB, gearB, motorB};
    View_3DObjectInfo[] gtA = {axisA, gearA, motorA};
    View_3DObjectInfo[] gtS = {subst};
    group[0] = gtC;
    group[1] = gtB;
    group[2] = gtA;
    group[3] = gtS;
  }

  public View_3DObjectInfo index( int i )
  {
    return this.index[i];
  }

  public int getIndexSize()
  {
    return this.index.length;
  }

  public int[][] getGroupSize()
  {
    return this.groupSize;
  }

  public boolean withinGroup( int groupIndex, int faceIndex )
  {
    for ( int i = 0; i < this.group.length; i++ )
    {
      if ( groupIndex == i )
      {
        for ( int j = 0; j < this.group[i].length; j++ )
        {
          if ( this.group[i][j].withinF( faceIndex ) )
            return true;
        }
      }
    }
    return false;
  }

  /** Index number of starting point of the array */
  public int v_start;
  /** Index number of ending point of the array */
  public int v_end;

  /** 3D object name. */
  public String label;
  /** File name. */
  public String filename;

  /** Index number of starting point of the array */
  public int face_start;
  /** Index number of ending point of the array */
  public int face_end;

  public String label;
  public String filename;
/** Condition of visibility. */
public boolean visible;

/** Body Color for 3D object */
public View_ColorSet color;

/** Offset value into proper position. [m] */
private double offsetX, offsetY, offsetZ;

/** Offset value into proper position. [radian] */
private double offsetPhi, offsetTheta, offsetGamma;

/** Width along X-axis. [m] */
public double width;

/** Width along Z-axis. [m] */
public double length;

/** Height along Y-axis. [m] */
public double height;

// public double halfWidth, halfLength, halfHeight;
// public View_Point3D centerR, centerV;

/**
 * Positions along with X-Z surface.
 * <li> fourCorner[0]: top right </li>
 * <li> fourCorner[1]: top left </li>
 * <li> fourCorner[2]: bottom left </li>
 * <li> fourCorner[3]: bottom right </li>
 */
public View_Point3D[] fourCorner;

public View_3DObjectInfo( String label, String filename, int v_start, int v_end,
                        int face_start, int face_end, boolean visible ) {
  this.label = label;
  this.filename = filename;
  this.v_start = v_start;
  this.v_end = v_end;
  this.face_start = face_start;
  this.face_end = face_end;
  this.visible = visible;
  this.color = new View_ColorSet();
}

public View_3DObjectInfo( String label, String filename ) {
  this( label, filename, 0, 0, 0, 0, true );
}

public View_3DObjectInfo() {
  this( "no-name", null, 0, 0, 0, true );
}

/**
 * Returns a size of the array for the part of this object.
 * @return int
 */
public int fsize() {
  return ( this.face_end - this.face_start + 1 );
}

/**
 * Returns a size of the array for the part of this object.
 * @return int
 */
public int vsize() {
  return ( this.v_end - this.v_start + 1 );
}

/**
 * Sets offset values and moves 3D object along with these values.
 * @param v View_Point3D[]. Array of 3D object point.
 * @param offsetPhi double. Offset angle around X axis. [radian]
 * @param offsetTheta double. Offset angle around Y axis. [radian]
 * @param offsetGamma double. Offset angle around Z axis. [radian]
 * @param offsetX double. Offset value along X axis. [m]
 * @param offsetY double. Offset value along Y axis. [m]
 * @param offsetZ double. Offset value along Z-axis. [m]
 */
public void setOffset( View_Point3D[] v,
                      double offsetPhi, double offsetTheta, double offsetGamma,
                      double offsetX, double offsetY, double offsetZ ) {
  for ( int i = this.v_start; i <= this.v_end; i++ ) {
    v[i].rotate( this.offsetPhi, this.offsetTheta, this.offsetGamma );
  }
}
v[i].moveBy( this.offsetX, this.offsetY, this.offsetZ );

/**
 * Returns highest point of the 3D object.
 * @param v View_Point3D[]: Array of 3D object point.
 * @return double: Height. (Max. value in Y axis.)
 */
public double getCurrentHeight( View_Point3D[] v ) {
  double max = v[this.v_start].y;
  for ( int i = this.v_start; i <= this.v_end; i++ ) {
    if ( max <= v[i].y ) {
      max = v[i].y;
    }
  }
  return max;
}

/**
 * setFourCorners sets points of 4 corner from the 3D object.
 *<p>
 * This routine works only for square shape object.
 * @param v View_Point3D[]: Array of 3D object point.
 */
public void setFourCorners( View_Point3D[] v ) {
  fourCorner = new View_Point3D[4];
  for ( int i = 0; i < 4; i++ ) {
    fourCorner[i] = new View_Point3D();
  }
  View_Point3D min, max;
  max = tmp[0];
  min = tmp[0];
  for ( int i = 0; i < tmp.length; i++ ) {
    if ( max.x <= tmp[i].x ) {
      max = tmp[i];
    }
    if ( min.x >= tmp[i].x ) {
      min = tmp[i];
    }
  }
  max = tmp[0];
  min = tmp[0];
  for ( int i = 0; i < tmp.length; i++ ) {
    if ( max.z <= tmp[i].z ) {
      max = tmp[i];
    }
    if ( min.z >= tmp[i].z ) {
      min = tmp[i];
    }
  }
  View_Point3D min, max:
  max = tmp[0];
  min = tmp[0];
  for ( int i = 0; i < tmp.length; i++ ) {
    if ( max.x <= tmp[i].x ) {
      fourCorner[3] = v[i + this.v_start];
    }
    if ( min.x >= tmp[i].x ) {
      fourCorner[1] = v[i + this.v_start];
    }
  }
  max = tmp[0];
  min = tmp[0];
  for ( int i = 0; i < tmp.length; i++ ) {
    if ( max.z <= tmp[i].z ) {
      fourCorner[2] = v[i + this.v_start];
    }
    if ( min.z >= tmp[i].z ) {
      fourCorner[0] = v[i + this.v_start];
    }
  }
}

/**
 * Returns a boolean value to check if the index number represents a part of this 3D object.
 *<li><code>true</code> if the index number is a part of this 3D object.</li>
 *<li><code>false</code> if the index number is not a part of this 3D object.</li>
 */
public boolean withinF(int index) {
    if (this.face_start <= index && index <= this.face_end) {
        return true;
    } else {
        return false;
    }
}

/*--View_Triangle---------------------------------------------------------------*/
class View_Triangle {

    public View_Point3D tri[] = new View_Point3D[3];
    public double nx, ny, nz, depth;
    public View_Triangle(View_Point3D tri0, View_Point3D tri1, View_Point3D tri2) {
        /* Here, you just give reference values of View_Point3D to make a link to this instance,
        * so don’t need to new these instances.
        */
        tri[0] = tri0;
        tri[1] = tri1;
        tri[2] = tri2;
        depth = 0;
        nx = 0;
        ny = 0;
        nz = 0;
    }
    public View_Triangle() {
        for (int i = 0; i < 3; i++) {
            tri[i] = new View_Point3D();
        }
        depth = 0;
        nx = 0;
        ny = 0;
        nz = 0;
    }
    public double getDepth() {
        return (tri[0].z + tri[1].z + tri[2].z);
        // return( Math.pow(tri[0].z,2) + Math.pow(tri[0].y,2) +
        //         Math.pow(tri[1].z,2) + Math.pow(tri[1].y,2) +
        //         Math.pow(tri[2].z,2) + Math.pow(tri[2].y,2) );
    }
    public void copy(View_Triangle t) {
        this().copy(t);
    }
}

/* Here, View_Point3D is required to new */

public View_Triangle() {
    for (int i = 0; i < 3; i++) {
        tri[i] = new View_Point3D();
    }
    depth = 0;
    nx = 0;
    ny = 0;
    nz = 0;
}

public double getDepth() {
    return (tri[0].z + tri[1].z + tri[2].z);
    // return( Math.pow(tri[0].z,2) + Math.pow(tri[0].y,2) +
    //         Math.pow(tri[1].z,2) + Math.pow(tri[1].y,2) +
    //         Math.pow(tri[2].z,2) + Math.pow(tri[2].y,2) );
}

public void copy(View_Triangle t) {
    this().copy(t.tri[0]);
    this.tri[1].copy(t.tri[1]);
    this.tri[2].copy(t.tri[2]);
    this.depth = t.depth;
    this nx = t nx;
    this.ny = t.ny;
    this.nz = t.nz;
}

/*--View_Point3D---------------------------------------------------------------*/
class View_Point3D {

    public double x, y, z;
    public View_Point3D(double x, double y, double z) {
        this.x = x;
        this.y = y;
        this.z = z;
    }
    public View_Point3D() {
        this(0.0, 0.0);
    }
    public View_Point3D(View_Point3D p) {
        this(p.x, p.y, p.z);
    }
    public void copy(View_Point3D p) {
        // View_Point3D v = new View_Point3D()
        this.x = p.x;
    }
}
public void moveTo( double x, double y, double z ) {
    this.x = x;
    this.y = y;
    this.z = z;
}

public void moveBy( double x, double y, double z ) {
    this.x += x;
    this.y += y;
    this.z += z;
}

public void rotate( double phi, double theta ) {
    this.rotateRef( phi, theta, this );
}

public void rotate( double phi, double theta, double gamma ) {
    this.rotateRef( phi, theta, gamma, this );
}

public void rotateRef( double phi, double theta, View_Point3D r ) {
    double x1, y1, z1;
    double sinT = Math.sin( theta );
    double cosT = Math.cos( theta );
    double sinP = Math.sin( phi );
    double cosP = Math.cos( phi );
    x1 = r.x * cosT + r.z * sinT;
    z1 = -r.x * sinT + r.z * cosT;
    this.x = x1;
    this.y = r.y;
    this.z = z1;
    y1 = this.y * cosP - this.z * sinP;
    z1 = this.y * sinP + this.z * cosP;
    this.y = y1;
    this.z = z1;
}

public void rotateRef( double phi, double theta, double gamma, View_Point3D r ) {
    double x2, y2, z1;
    double sinG = Math.sin( gamma );
    double cosG = Math.cos( gamma );
    this.rotateRef( phi, theta, r );
    x2 = this.x * cosG - this.y * sinG;
    y2 = this.x * sinG + this.y * cosG;
    this.x = x2;
    this.y = y2;
    this.z = this.z;
}

public void changePerspective( double s, double d ) {
    this.x = s / ( d - this.z ) * this.x;
    this.y = s / ( d - this.z ) * this.y;
}

/**--View_ColorSet-------------------------------------------------------------------------*/

class View_ColorSet {
  /** Hue.  [0 <= H < 1] */
  public float H;
  /** Saturation. [0 <= S <= 1] */
  public float S;
  /** Brightness. [0 <= B <= 1] */
  public float B;
  /** Alpha. [0 <= alpha <= 255] */
  public int alpha;

  public View_ColorSet() {
    setHSBandAlpha( 0, 0, 0, 255 ); /* black*/
  }

  /** View set HS Band Alpha / */
  public void setHSBandAlpha( int H, int S, int B, int Alpha ) {
    this.H = (float) H / 100.0;
    this.S = (float) S / 100.0;
    this.B = (float) B / 100.0;
    this.alpha = Alpha;
  }
}

*/--View_ColorSet--*
public View_ColorSet( float H, float S, float B, int alpha ) {
  setHSBandAlpha( H, S, B, alpha );
}

/**
 * Sets values of <code>H</code>, <code>S</code> and <code>B</code> and transparency.
 *
 * @param H float. Hue. [0 <= H < 1]
 * @param S float. Saturation. [0 <= S <= 1]
 * @param B float. Brightness. [0 <= B <= 1]
 * @param alpha int. Transparency. [0 <= alpha <= 255]
 */
public void setHSBandAlpha( View_ColorSet c ) {
  setHSBandAlpha( c.H, c.S, c.B, c.alpha );
}

/**
 * setHSBvalue sets values of <code>H</code>, <code>S</code> and <code>B</code>.
 *
 * @param H float. Hue. [0 <= H < 1]
 * @param S float. Saturation. [0 <= S <= 1]
 * @param B float. Brightness. [0 <= B <= 1]
 */
public void setHSBvalue( float H, float S, float B ) {
  float[] f = checkHSB( H, S, B );
  this.H = f[0];
  this.S = f[1];
  this.B = f[2];
}

/**
 * checkHSB
 *
 * @param H float
 * @param S float
 * @param B float
 * @return float[]
 */
public static float[] checkHSB( float H, float S, float B ) {
  float[] f = new float[3];
  if ( H < 0.0f ) {
    H = 0.0f;
  } else if ( H > 1f ) {
    H = 0.999999999f;
  }
  if ( S < 0.0f ) {
    S = 0.0f;
  } else if ( S > 1.0f ) {
    S = 1.0f;
  }
  if ( B < 0.0f ) {
    B = 0.0f;
  } else if ( B > 1.0f ) {
    B = 1.0f;
  }
  return f;
}

/**
 * Sets value of transparency as integer.
 *
 * @param alpha int. Transparency. [0 <= alpha <= 255]
 */
public void setAlpha( int alpha ) {
  this.alpha = checkAlpha( alpha );
}
public static int checkAlpha( int alpha ) {
    if ( alpha > 255 ) {
        alpha = 255;
    }
    if ( alpha < 0 ) {
        alpha = 0;
    }
    return alpha;
}

public static Color getHSBColorPlusAlpha( float H, float S, float B, int alpha ) {
    // Like below, if only Color class is used.
    // Color is required to new twice to get the final color.
    // (Inside of getHSBColor, Color is made new once.) */
    // c = Color.getHSBColor( H, S, B );
    // c = new Color( c.getRed(), c.getGreen(), c.getBlue(), alpha );
    int rgb = Color.HSBtoRGB( H, S, B );
    int r = ( rgb - (rgb & 0xFF00FFFF)) >> 16;
    int g = ( rgb - (rgb & 0xFFFF00FF)) >> 8;
    int b = rgb - (rgb & 0xFFFFFF00);
    Color c = new Color( r, g, b, alpha );
    return c;
}

class View_Point2D {
    public double x, y;
    public View_Point2D( double x, double y ) {
        this.x = x;
        this.y = y;
    }
}

class XY_Sequence {
    /** Step(Order) number. */
    private int order;
    /** Axis. */
    private int axis;
    /** Direction. */
    private int dir;
    /** Parameter Number. */
    private int pn;
    /** Total time [second] */
    private double t_t;
    /** Down time [second] */
    private double t_d;
}
private double t_d;
/** Up time [second] */
private double t_u;
/** Up displacement [# of steps] */
private double dis_u;
/** Down displacement [# of steps] */
private double dis_d;
/** Total displacement [# of steps] */
private double dis_t;

public XY_Sequence( int order, int axis, int dir, int pn, XY_Parameter p, double ctrlClock ) {
    this.order = order;
    this.axis = axis;
    this.dir = dir;
    this.pn = pn;
    getTiming( p, ctrlClock );
}

/**
   * getTiming calculates times and distances
   * for acceleration, deceleration and slew.
   *
   * <pre>
   *          |___________________
   *          |  /|                   | \
   *          | / |                   |  \
   *          |/  |                   |   \
   *          --------------------------------->[s]
   *          <---> up time (t_u)     |    |
   *          <---- down time (t_d) -->    |
   *          <---- total time (t_t)- ----->
   * </pre>
   *   In current (@2005) XY table config., [2000 steps] represents [one inch].
   *   (sequence.length) == (sequence[0].order + 1)
   *   @param p XY_Parameter
   *   @param ctrlClock double
   */
private void getTiming( XY_Parameter p, double ctrlClock ) {
    if ( pn != 0 ) {
        double dis = p.getDistance();
        double vel = p.getVelocity();
        double lvel = p.getLowVelocity();
        double acc = p.getAcceleration();
        double dec = p.getDeceleration();
        
        this.t_u = ( vel - lvel ) * acc / ctrlClock;
        this.t_d = ( vel - lvel ) * dec / ctrlClock;
        this.t_t = ( ( 2.0 * dis + lvel * ( this.t_u + this.t_d ) ) / vel + ( this.t_u + this.t_d ) ) / 2.0;
        this.t_d = this.t_t - this.t_d;
        this.dis_u = lvel * this.t_u + ( vel - lvel ) * this.t_u / 2.0;
        this.dis_d = this.dis_u + vel * ( this.t_d - this.t_u );
        this.dis_t = this.dis_d + vel * ( ( this.t_t - this.t_d ) + ( vel - vel ) * ( this.t_t - this.t_d ) ) / 2.0;
        // p.dis = (int)Math.rint(this.dis_t); // ?
    } else {
        this.t_t = 0.0;
        this.t_d = 0.0;
        this.t_u = 0.0;
        this.dis_u = 0.0;
        this.dis_d = 0.0;
        this.dis_t = 0.0;
    }
}

public int getOrder() {
    return this.order;
}

public int getAxis() {
    return this.axis;
}

public void getTiming( XY_Parameter p, double ctrlClock ) {
    if ( pn != 0 ) {
        double dis = p.getDistance();
        double vel = p.getVelocity();
        double lvel = p.getLowVelocity();
        double acc = p.getAcceleration();
        double dec = p.getDeceleration();
        
        this.t_u = ( vel - lvel ) * acc / ctrlClock;
        this.t_d = ( vel - lvel ) * dec / ctrlClock;
        this.t_t = ( ( 2.0 * dis + lvel * ( this.t_u + this.t_d ) ) / vel + ( this.t_u + this.t_d ) ) / 2.0;
        this.t_d = this.t_t - this.t_d;
        this.dis_u = lvel * this.t_u + ( vel - lvel ) * this.t_u / 2.0;
        this.dis_d = this.dis_u + vel * ( this.t_d - this.t_u );
        this.dis_t = this.dis_d + vel * ( ( this.t_t - this.t_d ) + ( vel - vel ) * ( this.t_t - this.t_d ) ) / 2.0;
        // p.dis = (int)Math.rint(this.dis_t); // ?
    } else {
        this.t_t = 0.0;
        this.t_d = 0.0;
        this.t_u = 0.0;
        this.dis_u = 0.0;
        this.dis_d = 0.0;
        this.dis_t = 0.0;
    }
}
public int getDirection() {
    return this.dir;
}

public int getParameterNumber() {
    return this.pn;
}

public double getTotalTime() {
    return this.t_t;
}

public double getUpTime() {
    return this.t_u;
}

public double getDownTime() {
    return this.t_d;
}

public double getTotalDisplacement() {
    return this.dis_t;
}

public double getUpDisplacement() {
    return this.dis_u;
}

public double getDownDisplacement() {
    return this.dis_d;
}

/*--XY_Option--------------------------------------------------------------------------*/
class XY_Option {
  /** Optional Control. <li> 0 = NO </li> <li> 1 = YES </li> */
  private boolean opt;

  /** Layer Number.<p>
   * <li> 0 = processes by key stroke </li>
   * <li> N >= 1, then processes continuously N times. </li>
   * */
  private int lnum;

  /** Additional Motion. <li> 0=NO </li> <li> 1=YES </li> */
  private boolean add;

  /** Irregular Motion. <li> 0 = don't have Irregular Motion </li>
   * <li> 1 = has Irregular Motion </li>
   * */
  private boolean irr;

  /** Continuous. <li> 0 = Not Continuous Motion </li>
   * <li> 1 = Continuous Motion </li>
   * */
  private boolean cnt;

  public XY_Option( boolean opt, int lnum, boolean add, boolean irr, boolean cnt ) {
    this.opt = opt;
    this.lnum = lnum;
    this.add = add;
    this.irr = irr;
    this.cnt = cnt;
  }

  public boolean getOptionalControl() {
    return this.opt;
  }

  public int getLayerNumber() {
    return this.lnum;
  }

  public boolean getAdditionalMotion() {
    return this.add;
  }

  public boolean getIrregularMotion() {
    return this.irr;
  }

  public boolean getContinuous() {
    return this.cnt;
  }
}
class XY_Parameter {
    /** Parameter number. */
    private int pn;

    /** Distance. [step number] */
    private double dis;

    /** LowVelocity. */
    private double lvel;

    /** Velocity. */
    private double vel;

    /** Acceleration. */
    private double acc;

    /** Deceleration. */
    private double dec;

    /** DownPoint. */
    private double rdp;

    /** Multiplier. */
    private double mul;

    public XY_Parameter( int pn, double dis, double lvel, double vel, double acc, double dec,
                        double rdp, double mul ) {
        this.pn = pn;
        this.dis = dis;
        this.lvel = lvel;
        this.vel = vel;
        this.acc = acc;
        this.dec = dec;
        this.rdp = rdp;
        this.mul = mul;
    }

    public int getParameterNumber() {
        return this.pn;
    }

    public double getDistance() {
        return this.dis;
    }

    public double getLowVelocity() {
        return this.lvel;
    }

    public double getVelocity() {
        return this.vel;
    }

    public double getAcceleration() {
        return this.acc;
    }

    public double getDeceleration() {
        return this.dec;
    }

    public double getDownPoint() {
        return this.rdp;
    }

    public double getMultiplier() {
        return this.mul;
    }
}

class View_Print {
    /** message text area */
    TextArea txt;

    public View_Print( TextArea txt ) {
        this.txt = txt;
    }

    /**
     * printTextBox prints out message
     * into the text area which was set in the constructor.
     *
     * @param message String
     */
    public void printTextBox( String message ) {
        txt.insert( message, txt.getText().length() );
    }
}

/* Print --------------------------------------------------------------------------*/
// printTextBoxN prints out message with a line break
// into the text area which was set in the constructor.
*
/**
 * printTextBoxN prints out message with a line break
 * into the text area which was set in the constructor.
 *
 * @param message String
 */
public void printTextBoxN( String message ) {
    txt.insert( message + "\n", txt.getText().length() );
}

/* XY_AxisBody --------------------------------------------------------------------*/
class XY_AxisBody {
    /**
     * Point to retain base position of this 3d object.
     */
    private View_Point3D Body;
    /**
     * Array to store 3D object data points.
     */
    public View_Point3D[] Point3D;
    /**
     * Index number of starting point for the array
     */
    public int s;
    /**
     * Index number of ending point for the array
     */
    public int e;

    public XY_AxisBody( double x, double z, double s, double e ) {
        this.s = s;
        this.e = e;
        Body = new View_Point3D( x, 0, z );
        this.s = s;
        this.e = e;
    }

    public XY_AxisBody( double x, double z, int s, int e, View_Point3D[] P ) {
        this( x, z, s, e );
        this.Point3D = P;
    }

    public void setAxisPosition( double x, double y, double z ) {
        Body.x = x;
        Body.y = y;
        Body.z = z;
    }

    public void moveBy( double x, double y, double z ) {
        for ( int i = s; i <= e; i++ ) {
            Point3D[i].moveBy( x, y, z );
        }
    }

    public void moveTo( double x, double y, double z ) {
        this.moveBy( x - Body.x, y - Body.y, z - Body.z );
        this.setAxisPosition( x, y, z );
    }

    public void moveTo( double x ) {
        this.moveTo( x, Body.y, Body.z );
    }

    public View_Point3D getAxisPosition() {
        return this.Body;
    }

    // public void setStagePosition() {} // public boolean isBusy() {return true;}
}

/*-XY_XYtableControler------------------------------------------------------------------------*/
class XY_XYtableControler {
    /**
     * Instance to express XY table object body.
     */
    private XY_AxisBody A, B, Stage, Deposit, GearBoxA, MotorA;
    /**
     * Point to indicate current position.
     */
    private View_Point3D currentPosition = new View_Point3D();
    /**
     * Point to indicate previous position.
     */
    private View_Point3D previousPosition = new View_Point3D();
    /**
     * Index value to indicate at where XY table is processing.
     */
    private int seq_count;
}
private View_Point3D markerPoint3D;
/** Point to indicates the center of substrate. */
private XY_Sequence[] main_sq;
/** Array of Sequence set for main part. */
private XY_Sequence[] add_sq;
/** Array of Sequence set for additional motion part. */
private XY_Sequence[] irr_sq_s;
/** Array of Sequence set for irregular motion part. */
private XY_Sequence[] irr_sq_e;
/** Array of Parameter set. */
private XY_Parameter[] para;
/** Optional setting. */
private XY_Option op;
/** Version of C program */
private String VERSION_C = "2.78";
/** Conversion for [inch] (2000 steps) to [meter] */
private final double itom = (0.02540 / 2000.0);
/** Clock speed. [5MHz] is a clock speed [Hz] of XY table cntl. board */
private final double ctrlClock = 5000000.0;
private View_3DObjectInfoSet info;
private final boolean debug = true;
public XY_XYtableController(View_3DObjectInfoSet info, TextArea txt) {
    position.x = 0.0;
    position.z = 0.0;
    pre_position.x = 0.0;
    pre_position.z = 0.0;
    message = new View_Print(txt);
    this.info = info;
}
/**
 * Sets all objects into instance of XY_AxisBody.
 */
public void init(View_Point3D[] SprayPattern, View_Point3D[] Real_Point3D, View_Point3D markerPoint3D) {
/* this require preset information of points */
B = new XY_AxisBody(0, 0, info.axisB.v_start, info.axisB.v_end, Real_Point3D);
A = new XY_AxisBody(0, 0, info.axisA.v_start, info.axisA.v_end, Real_Point3D);
GearBoxA = new XY_AxisBody(0, 0, info.gearA.v_start, info.gearA.v_end, Real_Point3D);
MotorA = new XY_AxisBody(0, 0, info.motorA.v_start, info.motorA.v_end, Real_Point3D);
/* for the substrate */
Stage = new XY_AxisBody(0, 0, info.subst.v_start, info.subst.v_end, Real_Point3D);
/* this is to move deposition pattern with substrate motion */
Deposit = new XY_AxisBody(0, 0, SprayPattern.length - 1, SprayPattern);
this.markerPoint3D = markerPoint3D;
}
/**
 * Sets all of XY table objects into initial position.
 */
public void reset() {
/* Reset XY_AxisBody objects */
A.moveTo(0.0, 0.0, 0.0);
GearBoxA.moveTo(0.0, 0.0, 0.0);
MotorA.moveTo(0.0, 0.0, 0.0);
B.moveTo(0.0, 0.0, 0.0);
Stage.moveTo(0.0, 0.0, 0.0);
Deposit.moveTo(0.0, 0.0, 0.0);
markerPoint3D.moveTo(0.0, markerPoint3D.y, 0.0);
position.x = 0.0;
position.z = 0.0;
pre_position.x = 0.0;
pre_position.z = 0.0;

/** If set_count is set to zero in this function,
 * then the line of the deposit will be removed
 * from the image area after finishing the sequences. */

/**
 * @param time_d double. Duration time. [second]
 */

/**
 * getObjPosition calculates current position,
 * then moves XY table objects into proper position.
 */

*/

/**
 * getMotionData reads the motion data from a input file.
 */

*/

/**
 * setSeqCount( int seq_count )
 * this.seq_count = seq_count;
 */

/*
 * setSeqCount( int seq_count )
 */

/**
 * getObjPosition calculates current position,
 * then moves XY table objects into proper position.
 */

/**
 * setSeqCount( int seq_count )
 */

/**
 * getObjPosition calculates current position,
 * then moves XY table objects into proper position.
 */

/**
 * getObjPosition calculates current position,
 * then moves XY table objects into proper position.
 */
For XY_Sequence constructor, new XY_Parameter instance prior to new XY_Sequence instances.

if (!chkerr) {
    para = new XY_Parameter[v_para.size()]; /* Parameter set */
    for (int i = 0; i < para.length; i++) {
        int s[] = (int[]) v_para.elementAt(i);
        para[i] = new XY_Parameter(s[0], s[1], s[2], s[3], s[4], s[5], s[6], s[7]);
    }
}
main_sq = new XY_Sequence[v_main.size()]; /* Sequence set for main part */
for (int i = 0; i < main_sq.length; i++) {
    int s[] = (int[]) v_main.elementAt(i);
    main_sq[i] = new XY_Sequence(s[0], s[1], s[2], s[3], para[s[3]], ctrlClock);
}
if (!chkerr) {
    Vector v_add_sq = new Vector();
    Vector v_irr_sq_s = new Vector();
    Vector v_irr_sq_e = new Vector();
    int[] op_tmp = new int[5];
    chkerr = setOption(r, v_add_sq, v_irr_sq_s, v_irr_sq_e, op_tmp);
    op = new XY_Option((op_tmp[0] == 1) ? true : false,
                         op_tmp[1],
                         (op_tmp[2] == 1) ? true : false,
                         (op_tmp[3] == 1) ? true : false,
                         (op_tmp[4] == 1) ? true : false);
    add_sq = new XY_Sequence[v_add_sq.size()]; /* Sequence set for additional motion part */
    irr_sq_s = new XY_Sequence[v_irr_sq_s.size()]; /* Sequence set for irregular motion part */
    irr_sq_e = new XY_Sequence[v_irr_sq_e.size()]; /* Sequence set for irregular motion part */
    for (int i = 0; i < add_sq.length; i++) {
        int s[] = (int[]) v_add_sq.elementAt(i);
        add_sq[i] = new XY_Sequence(s[0], s[1], s[2], s[3], para[s[3]], ctrlClock);
    }
    for (int i = 0; i < irr_sq_s.length; i++) {
        int s[] = (int[]) v_irr_sq_s.elementAt(i);
        irr_sq_s[i] = new XY_Sequence(s[0], s[1], s[2], s[3], para[s[3]], ctrlClock);
    }
    for (int i = 0; i < irr_sq_e.length; i++) {
        int s[] = (int[]) v_irr_sq_e.elementAt(i);
        irr_sq_e[i] = new XY_Sequence(s[0], s[1], s[2], s[3], para[s[3]], ctrlClock);
    }
}
break;
}
r.close();
}
catch (Exception e) {
    message.printTextBoxN("@getMotionData:");
    message.printTextBoxN(e);
}
if (!chkflg) {
    string += 
        "Wrong File Format or Parameter.
        -----------------------------
        Check File Format.
        <XY-Table-Motion-Input-Data-File is required as Root Element.
        -----------------------------
    " + string;
    message.printTextBox("" + string);
}
int j = 0;

String string = "";

for ( j = 1; j < main_sq.length; j++ ) {
  string += j + " ]
    + main_sq[j].getAxis() + 
    + main_sq[j].getDirection() + 
    + main_sq[j].getParameterNumber() + "\n";
}

String string = "\nStep Sequence
";

for ( j = 1; j < para.length; j++ ) {
  string += j + " ]
    + para[j].getDistance() + 
    + para[j].getLowVelocity() + 
    + para[j].getVelocity() + 
    + para[j].getAcceleration() + 
    + para[j].getDownPoint() + "\n";
}

String string = "\nOptionalControl=" + ( op.getOptionalControl() ? "YES" : "NO" ) + 
    + " LayerNumber=" + op.getLayerNumber();

String string = "\nStep Sequence for Additional MotionVn
";

for ( j = 1; j < add_sq.length; j++ ) {
  string += j + " ]
    + add_sq[j].getAxis() + 
    + add_sq[j].getDirection() + 
    + add_sq[j].getParameterNumber() + "\n";
}

String string = "\nStep Sequence for Irregular(StartingPoint)Vn
";

for ( j = 1; j < irr_sq_s.length; j++ ) {
  string += j + " ]
    + irr_sq_s[j].getAxis() + 
    + irr_sq_s[j].getDirection() + 
    + irr_sq_s[j].getParameterNumber() + "\n";
}

String string = "\nStep Sequence for Irregular(EndingPoint)Vn
";

for ( j = 1; j < irr_sq_e.length; j++ ) {
  string += j + " ]
    + irr_sq_e[j].getAxis() + 
    + irr_sq_e[j].getDirection() + 
    + irr_sq_e[j].getParameterNumber() + "\n";
}

String string = "\nC Program Ver.=" + VERSION_C + 
    + "\n";

message.printTextBoxN( string );

if ( debug ) {
  XY_Sequence[][] sq = new XY_Sequence[4][];
  String[] title = {"Main", "Additional", "Irregular(StartingPoint)", "Irregular(EndingPoint)"};
  for ( int i = 0; i < sq.length; i++ ) {
    message.printTextBoxN( "Calc.Test + title[i] + Sequence" );
    for ( j = 0; j < sq[i].length; j++ ) {
      message.printTextBox( "[ + j + ]" + 
        + sq[i][j].getUpTime() + 
        + sq[i][j].getDownTime() + 
        + sq[i][j].getTotalTime() + "\n" );
    }
  }

  message.printTextBoxN( "Calc.Position of Deposit" );
  for ( j = 0; j < p.m.length; j++ ) {
    message.printTextBox( "[ + j + ]" + p.m[j].x + "[ + j + ]" + p.m[j].z + "\n" );
  }
}

/**
 * getMainSequenceSize
 * Returns a number of steps for this sequence.<p>
 * not a number of length of this array.<p>
 * It returns ( array.length - 1 )
 */
public int getMainSequenceSize() {
    return main_sq.length - 1; /* length of array[0~array.length] - array[0] */
}

/**
 * @return int
 */
public int getMainSequenceSize() {
    return main_sq.length - 1; /* length of array[0~array.length] - array[0] */
}

/**
 * @return boolean
 */
public boolean calcTiming(View_Point3D[] p_m, View_Point3D[] p_s) {
    if (op.getIrregularMotion()) {
        //getPattern( irr_sq_s, p_m, p_s);
        //getPattern( irr_sq_e, p_m, p_s);
    }
    // main part
    getPattern(main_sq, p_m, p_s);
    if (op.getAdditionalMotion()) {
        //getPattern( add_sq, p_m, p_s);
    }
    return false;
}

/**
 * @param s XY_Sequence[]
 * @param p_m View_Point3D[]
 * @param p_s View_Point3D[]
 */
private void getPattern(XY_Sequence[] s, View_Point3D[] p_m, View_Point3D[] p_s) {
    for (int i = 1; i < s.length; i++) {
        int pn = s[i].getParameterNumber();
        int dir = s[i].getDirection();
        int axis = s[i].getAxis();
        double dis = para[pn].getDistance();
        if (axis == 0) {
            p_m[i].x = p_m[i - 1].x + (dir == 0 ? -dis : dis);
            p_m[i].z = p_m[i - 1].z;
            p_s[i].x = -p_m[i].x * itom;
            p_s[i].z = -p_m[i].z * itom;
        } else if (axis == 1) {
            p_m[i].x = p_m[i - 1].x + (dir == 0 ? -dis : dis);
            p_m[i].z = p_m[i - 1].z + (dir == 0 ? -dis : dis);
            p_s[i].x = -p_m[i].x * itom;
            p_s[i].z = -p_m[i].z * itom;
        }
    }
}

/**
 * @param position View_Point3D
 * @param sq XY_Sequence[]
 * @param time_d double. Duration time of simulation. [second]
 * @return boolean
 */
private boolean getMotion(View_Point3D position, XY_Sequence[] sq, double time_d) {
    */
    **/
    * getPattern
    * @param s XY_Sequence[]
    * @param p_m View_Point3D[]
    * @param p_s View_Point3D[]
    */
    private void getPattern(XY_Sequence[] s, View_Point3D[] p_m, View_Point3D[] p_s) {
        for (int i = 1; i < s.length; i++) {
            int pn = s[i].getParameterNumber();
            int dir = s[i].getDirection();
            int axis = s[i].getAxis();
            double dis = para[pn].getDistance();
            if (axis == 0) {
                p_m[i].x = p_m[i - 1].x + (dir == 0 ? -dis : dis);
                p_m[i].z = p_m[i - 1].z;
                p_s[i].x = -p_m[i].x * itom;
                p_s[i].z = -p_m[i].z * itom;
            } else if (axis == 1) {
                p_m[i].x = p_m[i - 1].x + (dir == 0 ? -dis : dis);
                p_m[i].z = p_m[i - 1].z + (dir == 0 ? -dis : dis);
                p_s[i].x = -p_m[i].x * itom;
                p_s[i].z = -p_m[i].z * itom;
            }
        }
    }

    */
    * getMotion calculates current position.
    * @param position View_Point3D
    * @param sq XY_Sequence[]
    * @param time_d double. Duration time of simulation. [second]
    * @return boolean
    */
    private boolean getMotion(View_Point3D position, XY_Sequence[] sq, double time_d) {
        */
    */
double incTime; /* Incremental time. [second]. The time passed till previous step. */
double t; /* Internal time within one sequence [second] */
int sqn; /* Sequence number */
double displacement; /* Displacement */
displacement = 0;
t = 0;
sqn = 0;
incTime = 0; // = sq[sq[i].getTotalTime()];
while((time_d > (incTime + sq[sqn].getTotalTime()) * 1000.0) {  
  incTime += sq[sqn].getTotalTime();
  sqn++;
  if (sqn >= sq.length) {  
    /** 
     * if i>sq[0], order (i >= sq.length), it means the sequence is finished. 
     * then, just stop this function. 
     * */
    return true;
  }  
  t = time_d / 1000.0 - incTime; /* total time - time passed till previous steps. */
  pre_position.x = 0; /* Reset the position. */
  pre_position.z = 0;
  for (int i = 1; i < sqn; i++) {  
    int pn = sq[i].getParameterNumber();
    int dir = sq[i].getDirection();
    int axis = sq[i].getAxis();
    double dis = para[pn].getDistance();
    if (axis == 0) {
      pre_position.x += (dir == 0) ? (-dis) : (dis) * itom;
    } else if (axis == 1) {
      pre_position.z += (dir == 0) ? (dis) : (-dis) * itom;
    }
  }
  int pn = sq[sqn].getParameterNumber();
  if (sqn != 0) {
    double v = para[pn].getVelocity();
    double lv = para[pn].getLowVelocity();
    double tU = sq[sqn].getUpTime();
    double tD = sq[sqn].getDownTime();
    double tT = sq[sqn].getTotalTime();
    double disU = sq[sqn].getDownDisplacement();
    double disD = sq[sqn].getDownDisplacement();
    if (t <= tU) {  
      /* In acc. displacement is still in [step]. */
      displacement = lv * t + (v - lv) * t * t / (2.0 * tU);
    } else if (tU < t && t <= tD) {  
      /* In slew. displacement is still in [step]. */
      displacement = disU + v * (t - tU);
    } else if (tD < t) {  
      /* In dec. displacement is still in [step]. */
      displacement = disD + v * (t - tD) + (lv - v) * Math.pow(t - tD, 2) / (2.0 * (tT - tD));
    }
  }
  /* unit conversion. [inch(2000step)] -> [meter] */
displacement *= itom;
  if (sq[sqn].getAxis() == 0) {  
    /* A-axis. Only x is changed. */
    position.x = pre_position.x + (dir == 0) ? (-displacement) : (displacement);
    position.z = pre_position.z;
  } else if (sq[sqn].getAxis() == 1) {  
    /* B-axis. Only z is changed. */
    position.x = pre_position.x;
    position.z = pre_position.z + (dir == 0) ? (displacement) : (-displacement);
  }
  seq_count = sqn;
  return false;
}}
private boolean setSequence(BufferedReader r, Vector v) {
```java
int i = 1;
int[] s0 = new int[4];
String[] tag = {"<Step">", "<Axis>", "<Direction>", "<ParameterNumber>"};

try {
    /* set Total step numbers <StepSequence Value=""> */
    if (getElementValue(r, "<StepSequence" , s0, 0) ) {
        return true;
    }
    int total = s0[0];
    v.addElement( s0 );
    for ( i = 1; i <= total; i++ ) {
        int[] s = new int[4];
        for ( int j = 0; j < s.length; j++ ) {
            if (getElementValue( r, tag[j], s, j ) ) {
                return true;
            }
        }
        v.addElement( s );
        r.readLine(); /* Process to next line. </Step>
    }
    r.readLine(); /* process to next line. </StepSequence>
}
catch ( Exception e ) {
    message.printTextBoxN( "@setSequence:loop=" + i + e );
}
return false;
}

private boolean setParameter( BufferedReader r, Vector p ) {
    int i = 1;
    int[] pp0 = new int[8];
    String[] tag = {"<Parameter">", "<Distance>", "<Low_Velocity>", "<Velocity>", "<Acceleration>",
                   "<Deceleration>", "<Ramp_Down_Point>", "<Multiplier>"};
    try {
        /* set Total parameter numbers <ParameterSet Value=""> */
        if (getElementValue( r, "<ParameterSet" , pp0, 0) ) {
            return true;
        }
        int total = pp0[0];
        p.addElement( pp0 );
        /* note: i starts with 1, j starts with 0 */
        for ( i = 1; i <= total; i++ ) {
            int[] pp = new int[8]; /* this array needs to be defined inside of this loop */
            for ( int j = 0; j < pp.length; j++ ) {
                if (getElementValue( r, tag[j], pp, j ) ) {
                    return true;
                }
            }
            p.addElement( pp );
            r.readLine(); /* process to next line. </Parameter>
        }
        r.readLine(); /* process to next line. </ParameterSet>
    }
    catch ( Exception e ) {
        message.printTextBoxN( "@SetParameter:loop=" + i + e );
    }
    return false;
}
```

```
3055|
3056|
3057|
3058|
3059|
3060|
3061|
3062|
3063|
3064|
3065|
3066|
3067|
3068|
3069|
3070|
3071|
3072|
3073|
3074|
3075|
3076|
3077|
3078|
3079|
3080|
3081|
3082|
3083|
3084|
3085|
3086|
3087|
3088|
3089|
3090|
3091|
3092|
3093|
3094|
3095|
3096|
3097|
3098|
3099|
3100|
3101|
3102|
3103|
3104|
3105|
3106|
3107|
3108|
3109|
3110|
3111|
3112|
3113|
3114|
3115|
3116|
3117|
3118|
3119|
3120|
3121|
3122|
3123|
3124|
3125|
3126|
3127|
3128|
3129|
3130|
3131|
3132|
3133|
3134|
3135|
3136|
3137|
3138|
3139|
3140|
3141|
3142|
3143|
3144|
3145|
3146|

/* Read irregular step sequences - Starting point */
r.readLine(); /* Process to next line. <StartingPoint>\n */
if ( setSequence( r, irr_sq_s ) ) {
return true;
}
r.readLine(); /* process to next line. </StartingPoint>\n */
/* Read irregular step sequences - Ending point */
r.readLine(); /* Process to next line. <EndingPoint>\n */
if ( setSequence( r, irr_sq_e ) ) {
return true;
}
r.readLine(); /* process to next line. </EndingPoint>\n */
r.readLine(); /* process to next line. </Irregular>\n */
/**
* Read additional step sequences
* Set Additional Motion value
*/
if ( getElementValue( r, tag[4], o, 4 ) ) {
return true;
}
if ( setSequence( r, add_sq ) ) {
return true;
}
}
catch ( Exception e ) {
message.printTextBoxN( "@setOption:" + e );
}
return false;
}
/**
* getElementValue
*
* @param r BufferedReader
* @param element String
* @param value int[]
* @param n int
* @return boolean
*/
private boolean getElementValue( BufferedReader r, String element, int[] value, int n ) {
String line;
try {
/* Skip empty lines */
while ( ( line = r.readLine().trim() ).length() == 0 ) {
;
}
/**
* Compare two strings( line & element ) if these are same,
* within a length defined by element.length().
*/
if ( line.substring( 0, element.length() ).equals( element ) ) {
/** Compare a last charater of the string is same as ">" */
if ( !element.endsWith( ">" ) ) {
/**
* <Tag Attr="value(int)">\n
* This format is assumed.
*/
StringTokenizer st = new StringTokenizer( line, "\"" );
if ( st.countTokens() == 3 ) {
st.nextToken();
value[n] = Integer.parseInt( st.nextToken() );
return false;
}
return true;
} else {
/**
* <Tag>value(int)</Tag>\n
* This format is assumed.
*/
StringTokenizer st = new StringTokenizer( line, "<>" );
if ( st.countTokens() == 3 ) { /* =3 means that a line has 3 elements. */
st.nextToken();
value[n] = Integer.parseInt( st.nextToken() );
return false;
}
return false;
}
}
}
catch ( Exception e ) {

258


message.printTextBox('#getElementValue: ' + e);
    return true;
  }
  return true;
}
REFERENCES


41. Bialiauskaya, V.V., "The Nucleation Kinetics of Traveling Fe-17at.%B Droplets Produced by the Uniform-Droplet Spray Process", in The Department of Mechanical and Industrial Engineering, 2007, Ph.D. Thesis, Northeastern University, Boston, MA.


FL, United States, Minerals, Metals and Materials Society, Warrendale, PA 15086, United States.


