Characterization and Control of a Woven Biomimetic Actuator for Wearable Neurorehabilitative Devices

Vaughan K. Murphy, The University of Western Ontario

Supervisor: Trejos, Ana Luisa, The University of Western Ontario
A thesis submitted in partial fulfillment of the requirements for the Master of Engineering Science degree in Electrical and Computer Engineering
© Vaughan K. Murphy 2020

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Part of the Electrical and Computer Engineering Commons

Recommended Citation
https://ir.lib.uwo.ca/etd/7548

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlswadmin@uwo.ca.
Characterization and Control of a Woven Biomimetic Actuator for Wearable Neurorehabilitative Devices

Vaughan Murphy

Department of Electrical and Computer Engineering
Western University

Abstract

Strokes are a major cause of death and disability, with nearly 300,000 Canadians suffering from stroke-related disability. Many survivors suffer from upper limb paresis, which makes ordinary tasks difficult. Availability of care is an issue, with only 43.3% of patients in Ontario receiving proper treatment. Robot-assisted neurorehabilitation has been used to address limited clinician availability, and has been shown to improve patient strength and range of motion faster than or comparable to traditional methods. However, the size and weight of current devices limit their use to within therapists’ offices. Incorporating actuators into clothing would make the devices more comfortable, natural to use, and allow for at-home care, which has been shown to improve therapy compliance, frequency, and efficacy.

In this work, a woven fabric bi-pennate artificial muscle system is presented. Its contractive stroke, force, speed, and efficiency were characterized, and the transient stroke and force were modelled. The latter was used to create a composite feedforward-feedback controller, which was tested to determine its performance in controlling the actuator using solely force feedback. The maximum stroke, force, and controlled contraction period were found to be 12.23%, 11.28 N, and 2.60 s respectively, with the first two scaling linearly with the number of active actuators. Results indicate that the actuator is able to meet or exceed clinical performance requirements, albeit nonconcurrently, and that the designed controller was an effective force feedback control method, limiting overshoot to 2.6% and steady state error to 0.94%. Further work is suggested to refine the actuator’s design.

Index terms— Artificial muscles, smart textiles, wearable robotic systems.
Lay Summary

Strokes are one of the leading causes of death and disability in developed countries, with nearly 300,000 Canadians being disabled because of them. Many stroke survivors suffer from nervous system damage affecting their arms, limiting their ability to perform acts of daily living. It has been shown that therapy that uses robots to move the patient’s wrists and arms for them can help improve their strength and range of motion. However, currently available devices are very big, heavy, and rigid, making them uncomfortable and limiting their use to inside therapist offices. Putting the moving components of the therapy devices into clothing would make them more natural to use, more comfortable, and let people use them from the comfort of their own homes, which would let more people get the help they need.

In this work, a soft artificial muscle system woven into a fabric that could be used for such devices is presented. To determine its suitability, its maximum range of motion, strength, speed, and efficiency were found and compared to therapy requirements. These data were also used to find mathematical relationships to predict the force and distance during operation of the muscle system, based on how many muscles were being used at a time. Finally, two different methods of controlling the system were designed, tested, and compared, to see which was better if one were to be used for a therapy device. The results suggest that the designed artificial muscle system could be used to create soft, wearable robotic therapy devices that can easily be controlled, but work remains to be done to further improve the design.
Acknowledgements

I am extremely honoured and lucky to have had the opportunity to work under my supervisor, Dr. Ana Luisa Trejos. Not only has she provided the guidance and support I’ve needed to be able to pull together this thesis, but she’s been an upstanding human being throughout the process, directing and helping me with patience and compassion. Thank you for making me welcome in your lab, and thank you for giving me the opportunity to work under you. It’s been an incredibly enjoyable experience that I will always remember, and I hope one day I can repay the kindness you’ve shown me.

Sincere thanks go out as well to my friends at the Wearable Biomechatronics Laboratory, who have helped make this experience so memorable. I’m lucky to have worked alongside you guys, you’re an exceptionally talented and fun group of people. I hope to keep in touch, and I’m looking forward to future board game nights once this pandemic is finally over. In particular, I would like to thank Brandon Edmonds, the TCA guru whose advice and guidance have been instrumental in getting this thesis off the ground. Thank you for setting me down this path for my project, and helping me troubleshoot the (numerous) problems I had with my setup. Another person I would like to thank in particular is Eugen Porter, whose electrical wizardry and patience helped me achieve as much as I was able to.

I would also like to thank my parents, Claudia and Greg, and my siblings, Sibeal and Quinn, for their love, support, and patience. I know this thesis has been stressful for all of us at different times, so thank you for tolerating me and my inverted sleep schedule, and providing me with an outlet during my extended hermitage.

This research was funded by the following grants awarded to Dr. A.L. Trejos: the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant RGPIN-2014-03815,
the Canadian Foundation for Innovation (CFI), the Ontario Research Fund (ORF), and the Ontario Ministry of Economic Development, Trade and Employment and the Ontario Ministry of Research and Innovation through the Early Researcher Award.
# Contents

Certificate of Examination ii

Abstract iv

Lay Summary v

Acknowledgements vi

Table of Contents viii

List of Figures xiii

List of Tables xviii

Nomenclature and Acronyms xxv

1 Introduction 1
   1.1 Motivation ................................................. 2
   1.2 General Problem Statement ................................. 3
   1.3 Research Objectives and Scope ............................ 4
   1.4 Overview of the Thesis .................................... 4

2 Literature Review 6
   2.1 Wrist Neurorehabilitation ................................ 6
   2.2 Robot-Assisted Therapy .................................... 8
   2.3 Twisted Coiled Actuators .................................. 10
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4 Fabric-Embedded Artificial Muscles</td>
<td>11</td>
</tr>
<tr>
<td>2.5 Biomimetic Pennate Structure</td>
<td>13</td>
</tr>
<tr>
<td>2.6 Control Strategies</td>
<td>14</td>
</tr>
<tr>
<td>2.7 Developing System Requirements</td>
<td>15</td>
</tr>
<tr>
<td>2.8 Conclusions</td>
<td>20</td>
</tr>
<tr>
<td>3 TCA Development and Calibration</td>
<td>21</td>
</tr>
<tr>
<td>3.1 TCA Fabrication</td>
<td>21</td>
</tr>
<tr>
<td>3.2 TCA Training</td>
<td>24</td>
</tr>
<tr>
<td>3.3 Stroke Sensor Calibration</td>
<td>26</td>
</tr>
<tr>
<td>3.4 Force Sensor Calibration</td>
<td>28</td>
</tr>
<tr>
<td>3.5 Temperature Sensor Calibration</td>
<td>30</td>
</tr>
<tr>
<td>3.6 Power Sensor Calibration</td>
<td>32</td>
</tr>
<tr>
<td>3.7 Force Production Pilot Study</td>
<td>35</td>
</tr>
<tr>
<td>3.8 Stroke Production Pilot Study</td>
<td>37</td>
</tr>
<tr>
<td>3.9 Conclusions</td>
<td>41</td>
</tr>
<tr>
<td>4 Actuator Characterization</td>
<td>42</td>
</tr>
<tr>
<td>4.1 Stroke Characterization</td>
<td>42</td>
</tr>
<tr>
<td>4.1.1 Purpose</td>
<td>42</td>
</tr>
<tr>
<td>4.1.2 Experimental Methods</td>
<td>43</td>
</tr>
<tr>
<td>4.1.3 Analytical Methods</td>
<td>50</td>
</tr>
<tr>
<td>4.1.4 Results</td>
<td>52</td>
</tr>
<tr>
<td>4.1.5 Stroke Production Analysis</td>
<td>60</td>
</tr>
<tr>
<td>4.1.6 Stroke Production Regression</td>
<td>69</td>
</tr>
<tr>
<td>4.1.7 Efficiency Analysis</td>
<td>75</td>
</tr>
<tr>
<td>4.2 Force Characterization</td>
<td>77</td>
</tr>
<tr>
<td>4.2.1 Purpose</td>
<td>77</td>
</tr>
<tr>
<td>4.2.2 Experimental Methods</td>
<td>78</td>
</tr>
<tr>
<td>4.2.3 Analytical Methods</td>
<td>84</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.2.4 Results</td>
<td>84</td>
</tr>
<tr>
<td>4.2.5 Force Production Analysis</td>
<td>92</td>
</tr>
<tr>
<td>4.2.6 Force Production Regression</td>
<td>99</td>
</tr>
<tr>
<td>4.3 Comparing Stroke &amp; Force to Clinical Requirements</td>
<td>104</td>
</tr>
<tr>
<td>4.4 Contraction Period Characterization</td>
<td>111</td>
</tr>
<tr>
<td>4.4.1 Purpose</td>
<td>111</td>
</tr>
<tr>
<td>4.4.2 Time Characteristics Analytical Methods</td>
<td>112</td>
</tr>
<tr>
<td>4.4.3 Time Characteristics Analysis</td>
<td>113</td>
</tr>
<tr>
<td>4.4.4 Constant Period Experimental Methods</td>
<td>120</td>
</tr>
<tr>
<td>4.4.5 Constant Period Analytical Methods</td>
<td>124</td>
</tr>
<tr>
<td>4.4.6 Constant Period Results</td>
<td>125</td>
</tr>
<tr>
<td>4.4.7 Constant Period Analysis</td>
<td>128</td>
</tr>
<tr>
<td>4.4.8 Minimum Period Experimental Methods</td>
<td>132</td>
</tr>
<tr>
<td>4.4.9 Minimum Period Analytical Methods</td>
<td>135</td>
</tr>
<tr>
<td>4.4.10 Minimum Period Results</td>
<td>136</td>
</tr>
<tr>
<td>4.4.11 Minimum Period Analysis</td>
<td>140</td>
</tr>
<tr>
<td>4.4.12 Efficiency Analysis</td>
<td>147</td>
</tr>
<tr>
<td>4.5 Conclusions</td>
<td>149</td>
</tr>
<tr>
<td>5 Force Control Implementation</td>
<td>153</td>
</tr>
<tr>
<td>5.1 Purpose</td>
<td>153</td>
</tr>
<tr>
<td>5.2 Control Law Development</td>
<td>154</td>
</tr>
<tr>
<td>5.3 Sample Size Calculations</td>
<td>164</td>
</tr>
<tr>
<td>5.4 Experimental Methods</td>
<td>165</td>
</tr>
<tr>
<td>5.5 Analytical Methods</td>
<td>168</td>
</tr>
<tr>
<td>5.6 Results</td>
<td>170</td>
</tr>
<tr>
<td>5.7 Analysis</td>
<td>178</td>
</tr>
<tr>
<td>5.8 Conclusions</td>
<td>194</td>
</tr>
</tbody>
</table>
6 Concluding Remarks

6.1 Contributions ................................................. 200
6.2 Limitations & Future Work ............................... 201

References ......................................................... 205

Appendices ......................................................... 217

A Engineering Drawings ........................................... 217

A.1 Electrical Components ........................................ 218
  A.1.1 UNO Pinout ............................................... 218
  A.1.2 ESP32 Pinout ........................................... 219
  A.1.3 UNO Functional Schematic ............................ 220
  A.1.4 ESP32 Functional Schematic .......................... 221
  A.1.5 Power Control Board Schematic ....................... 222

A.2 3D Printed Components ....................................... 223
  A.2.1 IR Sensor Holder ........................................ 223
  A.2.2 PCB Spacer ............................................. 224
  A.2.3 Centre Tendon .......................................... 225
  A.2.4 Side Tendon ............................................. 226
  A.2.5 Vertical Tendon Holder ................................ 227
  A.2.6 Pin Caps ................................................ 228
  A.2.7 Load Cell Hook ......................................... 233
  A.2.8 Lateral Tendon Holders ............................... 234

A.3 Traditionally Fabricated Components ....................... 236
  A.3.1 Rigid Support Structure ................................ 236
  A.3.2 Load Cell Attach Point ................................ 239

B MATLAB Code .................................................. 240

B.1 Characterization Experiments Code ......................... 240
  B.1.1 Stroke Regression Code ............................... 240
CONTENTS

B.1.2 Force Regression Code ........................................ 244
B.2 Force Control Experiment Code ................................. 249
   B.2.1 Thermal Constant Calculation Code .................... 249
   B.2.2 System Identification Code ............................... 250
   B.2.3 Controller Performance Calculation Code .............. 256
   B.2.4 Simulink Performance Calculation Code ................ 259

C Arduino Code ....................................................... 263
   C.1 Sensor Calibration Code .................................. 263
      C.1.1 Stroke Sensor Calibration Code ...................... 263
      C.1.2 Force Sensor Calibration Code ...................... 264
      C.1.3 Temperature Sensor Calibration Code ............... 266
      C.1.4 Power Sensor Calibration Code ...................... 268
   C.2 Pilot Study Code ........................................... 270
      C.2.1 Force Production Pilot Study Code .................. 270
      C.2.2 Stroke Production Pilot Study Code ................. 273
   C.3 Characterization Experiment Code ......................... 276
      C.3.1 Temperature Reading and Communication Experiment Code .... 276
      C.3.2 Stroke Characterization Experiment Code ........... 279
      C.3.3 Force Characterization Experiment Code ............ 291
      C.3.4 Constant Frequency Characterization Experiment Code .... 305
      C.3.5 Maximum Frequency Characterization Experiment Code .... 317
   C.4 Force Control Experiment Code .............................. 330
      C.4.1 PID Control Experiment Code ......................... 330
      C.4.2 FF Control Experiment Code .......................... 341

Vita ................................................................. 354
List of Figures

2.1 Different wrist rehabilitation motions. .................................................. 7
2.2 Examples of pennate structures found in nature. ................................. 14

3.1 TCA fabrication setup. ................................................................. 22
3.2 TCA coiling stages. ................................................................. 23
3.3 Completed untrained TCA. .......................................................... 23
3.4 TCA training setup. ................................................................. 24
3.5 Visual comparison between an untrained and trained TCA. .................. 25
3.6 Setup used for stroke sensor calibration. ......................................... 27
3.7 Curve fitting for displacement sensor calibration values. ..................... 27
3.8 Setup used for force sensor calibration. .......................................... 29
3.9 Curve fitting for force sensor calibration values. .............................. 29
3.10 Setup used for temperature sensor calibration. ............................... 31
3.11 Curve fitting for temperature sensor calibration values. ..................... 32
3.12 Control board constructed for the various experiments. ..................... 33
3.13 Setup used for power sensor calibration. ....................................... 34
3.14 Curve fitting for power sensor calibration values. ............................ 34
3.15 Setup used for force production pilot experiment, with thermocouple removed. ... 36
3.16 Results from the force production pilot experiment. ......................... 36
3.17 Setup used for stroke production pilot experiment. .......................... 38
3.18 Results for stroke production pilot experiment. .............................. 39
LIST OF FIGURES

4.1 PCB with all external components added in preparation for the characterization experiments. .................................................. 44
4.2 All of the 3D printed components fabricated for the stroke characterization experiment. 45
4.3 Actuator with TCAs strung between tendons, prior to thermocouple attachment. . 46
4.4 Thermocouple attachment process ........................................ 47
4.5 Actuator weaving at various stages of completion, various views. ............... 48
4.6 Weights and IR sensor placement for experimentation. .......................... 50
4.7 Randomly selected examples of individual trials for the data collected from the stroke characterization experiment. .............................. 54
4.8 Experimental results of the stroke characterization experiment. ................ 55
4.9 Randomly selected examples of individual trials for the data collected from the contraction phase of the stroke characterization experiment. .......... 56
4.10 Randomly selected examples of individual trials for the data collected from the relaxation phase of the stroke characterization experiment. ............... 57
4.11 Experimental results of the contraction phase of the stroke characterization experiment. 58
4.12 Experimental results of the relaxation phase of the stroke characterization experiment. 59
4.13 Confidence intervals of the percent contraction compared to activation level, with an overlaid linear regression to show linearity of the results. ............... 61
4.14 A simplified free body representation of the woven bi-pennate actuator. ....... 64
4.15 Average RMSE per activation level for stroke regression in each phase, by degree polynomial. .................................................. 71
4.16 Average RMSE per activation level for stroke regression in each phase, by degree polynomial, on a different scale. ................................. 72
4.17 Regression of the relationship between temperature and stroke for each activation level of the actuator in contraction. ................................. 73
4.18 Regression of the relationship between temperature and stroke for each activation level of the actuator in relaxation. ................................. 74
4.19 Wooden structure constructed to house the force characterization experimental setup. 79
4.20 Components fabricated prior to actuator construction. .......................... 80
4.21 Thermocouples attached to the TCAs of the actuator. .......................... 81
4.22 Close-up view of the weave density of actuator. ............................... 82
4.23 Various views of the completed setup of the force characterization experiment. . 83
4.24 Examples of individual trial data collected for the force characterization experiment. 86
4.25 Experimental results of the force characterization experiment. .................. 87
4.26 Examples of individual trial data collected for the contraction phase of the force characterization experiment. ......................................................... 88
4.27 Examples of individual trial data collected for the relaxation phase of the force characterization experiment. ......................................................... 89
4.28 Experimental results of the contraction phase of the force characterization experiment. 90
4.29 Experimental results of the relaxation phase of the force characterization experiment. 91
4.30 Confidence intervals of the contractive force of the actuator compared to activation level, with an overlaid linear regression to show linearity of the results. .... 96
4.31 Average RMSE per active pair for each degree of polynomial regressed. ........ 100
4.32 Regression of force compared to temperature for the data collected when the actuator was in contraction. ......................................................... 102
4.33 Regression of force compared to temperature for the data collected when the actuator was in relaxation. ......................................................... 103
4.34 The two power supplies connected together in parallel via Marrettes. ........... 121
4.35 An actuator used in the constant period experiment after having the thermocouples attached to its TCAs. ......................................................... 122
4.36 An actuator used in the constant period experiment after being half-woven. .... 123
4.37 An example of a fully constructed actuator used in the constant period experiment. 124
4.38 Data collected for the constant period experiment. .................................. 126
4.39 Data collected for the contraction phase of the constant period experiment. ...... 127
4.40 Confidence intervals of the percent contraction compared to activation level for the constant period characterization experiment, with an overlaid linear regression to show linearity of the results. ........................................ 129
4.41 An actuator used in the minimum period experiment after having the thermocouples attached to its TCAs. .............................. 133
4.42 An actuator used in the minimum period experiment after being half-woven. ... 134
4.43 An example of an actuator used in the minimum period experiment. ............ 135
4.44 Total data collected for the minimum period experiment. ....................... 138
4.45 Total data collected for the contraction phase of the minimum period experiment. 139
4.46 Confidence intervals of the percent contraction compared to activation level for the minimum contraction period experiment, with an overlaid linear regression to show linearity of the results. .............................................................. 143

5.1 The three PID control loops implemented in Simulink, used to tune the coefficients $K_P$, $K_I$, and $K_D$ of the PID controllers for each activation level. ............... 160
5.2 The three FF control loops implemented in Simulink, used to tune the coefficients $K_E$ of the FF controllers for each activation level. ......................... 163
5.3 An example of an inverted subsystem block, used in the construction of the FF control loops. ...................................................... 163
5.4 Actuator in various stages of construction, with TCAs strung between the holsters clamped to the wooden support structure from the Force Characterization Experiment.166
5.5 Completed experimental setup presented in two angles, in addition to two views of a disassembled actuator illustrating its weave-density and flexibility. ........... 167
5.6 Simulated step response of each designed controller. ............................. 171
5.7 Experimental step response of each designed controller for the full duration of each trial, with each trial shown. ............................................. 172
5.8 Example experimental step responses of each designed controller for the full duration of a randomly selected trial. ........................................... 173
5.9 Experimental step responses of each designed controller for the transient phase of each trial, with each trial shown. ........................................... 174
5.10 Example experimental step responses of each designed controller for the transient phase of a randomly selected trial. ................................. 175
5.11 Experimental step responses of each designed controller for the oscillatory phase of each trial, with each trial shown. .............................................................. 176

5.12 Example experimental step responses of each designed controller for the oscillatory phase of a randomly selected trial. .............................................................. 177
List of Tables

2.1 Examples of neurorehabilitative robots found in the literature, with affected joint, actuator, and portability listed. ................................. 9
2.2 Design requirements found in the literature. .............................. 16
2.2 Design requirements found in the literature. .............................. 17
2.2 Design requirements found in the literature. .............................. 18
2.2 Design requirements found in the literature. .............................. 19
2.3 Minimum, maximum, and average design requirements based on clinical requirements. 19

4.1 Minimum discernible effect calculations by activation level, using produced stroke results from the stroke characterization experiment. ..................... 60
4.2 Repeated measures one-way ANOVA analysis results for stroke produced by the actuator in contraction. .................................................. 62
4.3 Stroke production values for each activation level of the actuator in contraction. 63
4.4 Repeated measures one-way ANOVA analysis results for stroke produced by the actuator in relaxation. .................................................. 67
4.5 Stroke production values for each activation level of the actuator in relaxation. 68
4.6 Vertical and diagonal contraction of the TCAs, expressed as a percent stroke value. 69
4.7 Vertical and diagonal relaxation of the TCAs, expressed as a percent stroke value. 69
4.8 Regressed equation coefficient values for relationship between temperature and stroke for each activation level of the actuator in contraction. .......... 72
4.9 Regressed equation coefficient values for relationship between temperature and stroke for each activation level of the actuator in relaxation. ........... 72
4.10 Repeated measures one-way ANOVA analysis results for the efficiency of the actuator in contraction. ................................. 76
4.11 Efficiency values for each activation level of the actuator in contraction. ...................... 77
4.12 Minimum discernible effect calculations by activation level, using produced force results from the force characterization experiment. ............................... 92
4.13 Repeated measures one-way ANOVA results for the force production of the actuator in contraction. .................................. 93
4.14 Repeated measures one-way ANOVA results for the force production of the actuator in relaxation. ................................ 94
4.15 Force production values for each activation level of the actuator in contraction. .. 97
4.16 Force production values for each activation level of the actuator in relaxation. .. 98
4.17 Calculated force and pressure production values of the actuator in contraction. .. 98
4.18 Horizontal and diagonal force produced per active TCA in contraction. ............... 99
4.19 Regressed equation coefficient values for relationship between temperature and force production for each activation level of the actuator in contraction. .............. 101
4.20 Regressed equation coefficient values for relationship between temperature and force production for each activation level of the actuator in relaxation. .............. 101
4.21 Clinical requirements calculated from values present in the literature, as found in Chapter 2. .................................................. 104
4.22 Minimum radial arm to convert the force production of the actuator to the desired torques for clinical applications. .................................. 105
4.23 Maximum radial arm to convert the stroke of the actuator in contraction to the desired revolute range of motion for clinical applications. ......................... 106
4.24 ANOVA results comparing the heating time of both experiments, grouped by activation level. ................................................ 114
4.25 ANOVA results comparing the cooling time of both experiments, grouped by activation level. ................................................ 115
4.26 Kruskal-Wallis analysis results for each activation level of the actuator in heating, grouped by actuator number. ......................... 116
4.27 Kruskal-Wallis analysis results for each activation level of the actuator in heating, grouped by orientation. ................................................................. 117
4.28 Kruskal-Wallis analysis results for each activation level of the actuator in cooling, grouped by actuator number. ................................................................. 117
4.29 Kruskal-Wallis analysis results for each activation level of the actuator in cooling, grouped by orientation. ................................................................. 117
4.30 Calculated 95% confidence interval values of the time required for the actuator to heat by activation level and orientation. .............................................. 118
4.31 Calculated 95% confidence interval values of the time required for the actuator to cool by activation level and orientation. .............................................. 118
4.32 Trial number calculations by activation level, using heating time results from both characterization experiments. ................................................................. 119
4.33 Trial number calculations by activation level, using produced stroke results from the stroke characterization experiment. .............................................. 120
4.34 Repeated measures one-way ANOVA analysis results for maximum stroke produced by the actuator operating at a constant contraction period of 2.0 s. 128
4.35 Repeated measures one-way ANOVA analysis results for maximum temperature achieved by the actuator operating at a constant contraction period of 2.0 s. 129
4.36 Constructed 95% confidence interval for maximum stroke production values for the actuator running at a constant contraction period. ......................... 130
4.37 Constructed 95% confidence interval for maximum achieved temperature values for the actuator running at a constant contraction period. ......................... 130
4.38 Stroke production of the actuator operating at a constant contraction period of 2.0 s, compared to the corresponding results from the stroke characterization experiment. 130
4.39 Minimum sample size calculations by activation level, using achieved temperature results from the constant period characterization experiment. ................. 131
4.40 Repeated measures one-way ANOVA analysis results for maximum stroke produced by the actuator operating at its minimum contraction period. ................... 141
4.41 Repeated measures one-way ANOVA analysis results for maximum temperature achieved by the actuator operating at its minimum contraction period. ........................................ 141
4.42 Repeated measures one-way ANOVA analysis results for the time required to achieve maximum temperature in heating by the actuator operating at its minimum contraction period. ..................................................... 141
4.43 Results of Kruskal-Wallis test performed on the heating time results of the actuator operating at its minimum possible contraction period. ............................................. 142
4.44 Constructed 95% confidence interval for maximum achieved temperature values for the actuator running at the minimum possible contraction period. ..................... 142
4.45 Production of the actuator operating at the minimum possible contraction period, compared to the corresponding results from the stroke characterization experiment. 143
4.46 Constructed 95% confidence interval for maximum stroke production values for the actuator running at the maximum possible speed. ................................. 144
4.47 Constructed 95% confidence interval for the time required to achieve maximum temperature in heating for the actuator running at the maximum possible speed. 144
4.48 Repeated measures one-way ANOVA analysis results for the efficiency of the actuator operating at a constant contraction period. ............................................... 147
4.49 Repeated measures one-way ANOVA analysis results for the efficiency of the actuator operating at its minimum possible contraction period. ......................... 148
4.50 Constructed 95% confidence interval for the efficiency of the actuator operating at a constant contraction period. ............................................................... 149
4.51 Constructed 95% confidence interval for the efficiency of the actuator operating at the minimum possible contraction period. ............................................. 149

5.1 Thermal resistances and time constants of the system for each activation level, calculated from experimental data. ................................................................. 156
5.2 Temperature–force transfer functions identified from the regressed time-domain relationships, by activation level. ................................................................. 157
5.3 Power–temperature transfer functions identified from the simulated values, by activation level. ........................................................................................................ 158
5.4 Tuned coefficients for the designed PID controller, by activation level. .......... 161
5.5 Calculated representations of the inverted system transfer functions, by activation
level. ........................................................................................................... 162
5.6 Tuned feedforward controller coefficient, by activation level. ................. 163
5.7 Sample size calculations based on the results of the force characterization experiment
in Section 4.2.4. ............................................................................................ 164
5.8 Target force setpoints used throughout experimentation and simulation, based on
the results found in Section 4.2.4. ................................................................. 168
5.9 Repeated measures one-way ANOVA analysis results for the first peak time of the
actuator when controlled by a PID controller. ............................................ 179
5.10 Repeated measures one-way ANOVA analysis results for the time constant of the
actuator when controlled by a PID controller. ............................................ 179
5.11 Repeated measures one-way ANOVA analysis results for the rise time of the actuator
when controlled by a PID controller. ............................................................ 180
5.12 Repeated measures one-way ANOVA analysis results for the percent overshoot of
the actuator when controlled by a PID controller. ........................................ 180
5.13 Repeated measures one-way ANOVA analysis results for the settling value of the
actuator when controlled by a PID controller. ............................................ 180
5.14 Repeated measures one-way ANOVA analysis results for the first peak time of the
actuator when controlled by a FF controller. ................................................. 181
5.15 Repeated measures one-way ANOVA analysis results for the time constant of the
actuator when controlled by a FF controller. ................................................. 181
5.16 Repeated measures one-way ANOVA analysis results for the rise time of the actuator
when controlled by a FF controller. ............................................................... 181
5.17 Repeated measures one-way ANOVA analysis results for the percent overshoot of
the actuator when controlled by a FF controller. ........................................... 182
5.18 Repeated measures one-way ANOVA analysis results for the settling value of the
actuator when controlled by a FF controller. ................................................. 182
5.19 Between-subject comparison results of the repeated measures one-way ANOVA tests performed for the PID controller, comparing different actuators. .................. 183
5.20 Between-subject comparison results of the repeated measures one-way ANOVA tests performed for the FF controller, comparing different actuators. .................. 183
5.21 Between-subject comparison results of the repeated measures one-way ANOVA test between controllers, comparing the two. ................................. 184
5.22 Results of Kruskal-Wallis testing performed on the first peak time results of the actuator when controlled by a PID controller, grouped by actuator number. .. 185
5.23 Results of Kruskal-Wallis testing performed on the time constant results of the actuator when controlled by a PID controller, grouped by actuator number. .. 185
5.24 Results of Kruskal-Wallis testing performed on the rise time results of the actuator when controlled by a PID controller, grouped by actuator number. .......... 185
5.25 Results of Kruskal-Wallis testing performed on the first peak time results of the actuator when controlled by a FF controller, grouped by actuator number. .. 185
5.26 Results of Kruskal-Wallis testing performed on the time constant results of the actuator when controlled by a FF controller, grouped by actuator number. .. 185
5.27 Results of Kruskal-Wallis testing performed on the rise time results of the actuator when controlled by a FF controller, grouped by actuator number. .......... 185
5.28 Results of Kruskal-Wallis testing performed on the first peak time results of the actuator, grouped by controller type. ................................. 186
5.29 Results of Kruskal-Wallis testing performed on the time constant results of the actuator, grouped by controller type. ................................. 186
5.30 Results of Kruskal-Wallis testing performed on the rise time results of the actuator, grouped by controller type. ................................. 186
5.31 Constructed 95% confidence intervals of the first peak time achieved by the actuator when controlled by a PID controller. ................................. 187
5.32 Constructed 95% confidence intervals of the time constant achieved by the actuator when controlled by a PID controller. ................................. 187
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.33</td>
<td>Constructed 95% confidence intervals of the rise time achieved by the actuator when controlled by a PID controller.</td>
<td>187</td>
</tr>
<tr>
<td>5.34</td>
<td>Constructed 95% confidence intervals of the first peak time achieved by the actuator when controlled by a FF controller.</td>
<td>188</td>
</tr>
<tr>
<td>5.35</td>
<td>Constructed 95% confidence intervals of the time constant achieved by the actuator when controlled by a FF controller.</td>
<td>188</td>
</tr>
<tr>
<td>5.36</td>
<td>Constructed 95% confidence intervals of the rise time achieved by the actuator when controlled by a FF controller.</td>
<td>188</td>
</tr>
<tr>
<td>5.37</td>
<td>Constructed 95% confidence intervals of the percent overshoot achieved by the actuator when controlled by a PID controller.</td>
<td>189</td>
</tr>
<tr>
<td>5.38</td>
<td>Constructed 95% confidence intervals of the percent overshoot achieved by the actuator when controlled by a FF controller.</td>
<td>189</td>
</tr>
<tr>
<td>5.39</td>
<td>Constructed 95% confidence intervals of the settling value achieved by the actuator when controlled by a PID controller.</td>
<td>190</td>
</tr>
<tr>
<td>5.40</td>
<td>Constructed 95% confidence intervals of the settling value achieved by the actuator when controlled by a FF controller.</td>
<td>190</td>
</tr>
<tr>
<td>5.41</td>
<td>Steady state error of the experimental settling values compared to their target setpoints, for both the PID and FF control schemes.</td>
<td>190</td>
</tr>
<tr>
<td>5.42</td>
<td>Comparison of the experimental first peak time values to those obtained via simulation, for both the PID and FF control schemes.</td>
<td>191</td>
</tr>
<tr>
<td>5.43</td>
<td>Comparison of the experimental time constant values to those obtained via simulation, for both the PID and FF control schemes.</td>
<td>192</td>
</tr>
<tr>
<td>5.44</td>
<td>Comparison of the experimental rise time values to those obtained via simulation, for both the PID and FF control schemes.</td>
<td>192</td>
</tr>
<tr>
<td>5.45</td>
<td>Comparison of the experimental percent overshoot values to those obtained via simulation, for both the PID and FF control schemes.</td>
<td>193</td>
</tr>
<tr>
<td>5.46</td>
<td>Comparison of the experimental settling values to those obtained via simulation, for both the PID and FF control schemes.</td>
<td>194</td>
</tr>
</tbody>
</table>
Nomenclature and Acronyms

Latin Letters

\( a_i \) \( i^{th} \) polynomial coefficient

\( A \) Polynomial coefficient

\( B \) Polynomial coefficient

\( C \) Polynomial coefficient

\( d \) Displacement

\( D \) Polynomial coefficient

\( e \) Euler’s constant

\( E \) Polynomial coefficient

\( f \) Test statistic for ANOVA comparisons

\( F \) Polynomial coefficient

\( F_{\text{corrected}} \) Corrected load cell force reading

\( F_g \) Gravitational force

\( F_{\text{linear}} \) Linearly produced force

\( F_{\text{prod}} \) Force produced by the actuator

\( F_s \) Spring force

\( F_{\text{settle}} \) Settling value of the force production of the actuator

\( F_{s,y} \) Spring force acting in the \( y \)-direction

\( F_{\text{target}} \) Target force setpoint
\[ G(s) \quad \text{General transfer function} \]
\[ G^{-1}(s) \quad \text{General inverse transfer function} \]
\[ G_{PT}(s) \quad \text{Transfer function between the power and temperature of the actuator} \]
\[ G_{PT}^{-1}(s) \quad \text{The inverse of } G_{PT}(s) \]
\[ G_{TF}(s) \quad \text{Transfer function between the temperature and force production of the actuator} \]
\[ G_{TF}^{-1}(s) \quad \text{The inverse of } G_{TF}(s) \]
\[ h(t) \quad \text{Time domain representation of } H(s) \]
\[ H(s) \quad \text{General polynomial transfer function} \]
\[ i \quad \text{Placeholder value denoting index} \]
\[ i_{\text{add}} \quad \text{Additional number of pairs of TCAs required to meet goal force production} \]
\[ I \quad \text{Infrared sensor analog reading value} \]
\[ K \quad \text{Stiffness value} \]
\[ K_D \quad \text{Derivative coefficient in a PID controller} \]
\[ K_E \quad \text{Control coefficient in a FF controller} \]
\[ K_I \quad \text{Integral coefficient in a PID controller} \]
\[ K_P \quad \text{Proportional coefficient in a PID controller} \]
\[ K_{\text{total}} \quad \text{Total stiffness of the actuator} \]
\[ K_x \quad \text{Stiffness in the } x\text{-direction} \]
\[ K_y \quad \text{Stiffness in the } y\text{-direction} \]
\[ L \quad \text{Load cell analog reading value} \]
\[ m_{\text{force}} \quad \text{Slope of the force per activated pair of TCAs regression} \]
\[ n \quad \text{Sample size} \]
\[ p \quad \text{Probability value} \]
\[ p_k \quad k^{th} \text{ pole of a transfer function } G(s) \]
\[ P_{\text{corrected}} \quad \text{Corrected power sensor reading} \]
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{\text{sensor}}$</td>
<td>Raw power sensor reading</td>
</tr>
<tr>
<td>$r$</td>
<td>Radial arm length</td>
</tr>
<tr>
<td>$R^2$</td>
<td>Coefficient of correlation</td>
</tr>
<tr>
<td>$R_T$</td>
<td>Thermal resistance</td>
</tr>
<tr>
<td>$s$</td>
<td>Frequency domain variable</td>
</tr>
<tr>
<td>$s_{\text{prod}}$</td>
<td>Stroke produced by the actuator</td>
</tr>
<tr>
<td>$t$</td>
<td>Time</td>
</tr>
<tr>
<td>$T$</td>
<td>Temperature of the actuator</td>
</tr>
<tr>
<td>$T_{\text{amb}}$</td>
<td>Ambient temperature</td>
</tr>
<tr>
<td>$T_{\text{corrected}}$</td>
<td>Corrected temperature value of the temperature sensor</td>
</tr>
<tr>
<td>$T_i$</td>
<td>Initial temperature of the actuator</td>
</tr>
<tr>
<td>$T_{\text{sensor}}$</td>
<td>Raw temperature reading of the temperature sensor</td>
</tr>
<tr>
<td>$y_{\text{corrected}}$</td>
<td>Corrected IR distance reading</td>
</tr>
<tr>
<td>$Z$</td>
<td>Z-Score associated with certainty</td>
</tr>
<tr>
<td>$Z_\alpha$</td>
<td>Z-Score of associated Type I error</td>
</tr>
<tr>
<td>$Z_\beta$</td>
<td>Z-Score of associated Type II error</td>
</tr>
<tr>
<td>$%_{\text{diff}}$</td>
<td>Percent difference</td>
</tr>
</tbody>
</table>

**Greek Letters**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>Critical test value of the null-hypothesis</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Arc length</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Difference between two values</td>
</tr>
<tr>
<td>$\delta(t)$</td>
<td>Dirac-Delta function</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Standard deviation</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Angle</td>
</tr>
</tbody>
</table>
\( \theta(t) \) Heaviside function
\( \mu \) Mean value
\( \sigma \) Standard deviation
\( \tau \) Torque
\( \tau_c \) Thermal time constant

**Acronyms**

3-D Three-Dimensional
AC Alternating Current
ANOVA Analysis of Variance
BIBO Bounded Input Bounded Output
DC Direct Current
DOF Degree of Freedom
EAP Electroactive Polymer
FF 2 DOF PID-Feedforward Controller
IR Infrared
I\(^2\)C Inter-Integrated Circuit Communication Protocol
MOSFET Metal-Oxide Semiconductor Field-Effect Transistor
MRF Magnetorheological Fluid
PCB Printed Circuit Board
PID Proportional-Integral-Derivative
PWM Pulse Width Modulation
RMSE Root Mean Square Error
SMA Shape Memory Alloy
TCA Twisted Coiled Actuator
Chapter 1

Introduction

Stroke is one of the leading causes of death and disability within developed nations [1], with around 300,000 Canadians living with injuries caused by stroke [2]. Many stroke survivors suffer from reduced motor capabilities, with arm paresis being common among such patients [3]. Arm paresis is a neurological condition that limits arm movement, reducing the ability of the patient to move all parts of their arm, including their wrists [3]. In order to rehabilitate such patients, repetitive task training is often used with favourable results. Helping the patient to repetitively flex and extend their wrist develops strength, flexibility, and balance; however, access to such therapy is often limited by the cost and availability of trained therapists [4]. Therefore, improvements should be made to increase the availability, convenience, and efficiency of this type of therapy so that as many paretic stroke patients as possible can benefit from treatment.

Wearable robotic manipulative therapy devices have been shown to be effective methods of implementing this therapy, speeding wrist motor control recovery after a stroke event. When used by a therapist, they reduce the number of contact hours required with patients, even going so far as to be usable by non-therapist operators after preliminary setup [5]. If wearable devices could be made convenient and portable enough, they could even be used at home. At home use increases patient adherence to exercises [6], removes accessibility barriers, and allows for continued therapy during service disruptions, such as during pandemics or other disasters. Conventional implementations of such devices have used electric motors or pneumatic actuators in the past to actuate the wrist; however, these actuation methods are bulky, heavy, and non-compliant,
1.1 Motivation

Nylon 6,6 twisted coiled actuators (TCAs) are one of the newer smart materials seeking to replace these conventional actuation methods, with higher power to weight output, larger force generation, larger strokes, and being more linear with less hysteresis when compared to other smart material actuators and biological muscle fibres [10]. They operate by contracting when a heat source is applied to them, applying a tensile force along the axis of motion. This heating can be facilitated by passing an electric current through a conductive heating element bound with the twisted nylon, making the actuators electrically controllable [11]. While they have many desirable characteristics, TCAs are not without their drawbacks. They have very low efficiency, each individual fibre does not produce a large gross force, their response time is significantly slower in expansion compared to contraction, and when operating with a load or large stroke their lifecycles can diminish severely [12]. These shortcomings are currently affecting their ability to be implemented into commercial devices, but there are promising ways of overcoming these pitfalls.

Currently in the literature, attempts have been made with other forms of smart material artificial muscles to incorporate them into an actuating mesh, weaving threads through the artificial muscles to create an active fabric. This helped bolster stress and strain production compared to other multi-fibre actuators [13], and due to the similarities in the actuation methods it is a reasonable avenue of inquiry to ask whether the same results could be obtained using TCAs. Employing such a method also has the possibility of increasing TCA mechanical efficiency via increased thermal insulation, reducing the amount of waste heat generated while heating the actuators during contraction. While the theoretical benefits of this implementation could improve actuator performance, this alone is not enough to make TCAs see widespread use.

In order to further improve the viability of TCAs, multi-fibre bi-pennate structures mimicking human muscle have been proposed. This structure is one that operates by having opposing pairs making them less than ideal for use in portable wearable devices [7]. New forms of actuation, such as shape memory alloys (SMAs) or electroactive polymers (EAPs) have sought to overcome these limitations, being lighter and flexible, but they often lack force or displacement production capabilities, and often suffer from nonlinearities and hysteresis [8][9].
of TCAs arranged along a central tendon at an acute angle, allowing the fibres to work in concert without interfering with one another [14]. This increases the maximum force production of such units as a whole [15]. The amount of force produced is controlled by selectively activating opposing pairs of TCAs, varying the stiffness of the unit to produce different forces at different levels of displacement. In addition to this, the pennate structure has an architectural-based velocity gearing, which innately damps aperiodic disturbances, providing a greater degree of safety to the user than normal compliant actuators [16]. If this bi-pennate structure could be used in conjunction with an active woven mesh, the performance of a TCA-based actuating module could see significant improvements over current compliant actuator designs. It would also open up new avenues of development in active textiles, and provide a concealable, portable, safe, and easy-to-use actuation system for wearable rehabilitative and supportive devices.

To improve the stress and strain production capabilities of a theoretical TCA-driven rehabilitative device to combat arm paresis, this project proposes using a bi-pennate TCA structure embedded in a woven mesh. It is predicted that this configuration would have the increased compliance and force production found in bi-pennate structures, the increased stress and strain production capabilities found in active meshes, and possibly increased efficiency due to insulating effects from being embedded in fabric. As well, having the actuators embedded in a woven mesh would facilitate the use of these rehabilitative devices in garments, making them more natural to use, less conspicuous, and provide avenues of extension to other possible applications. There is a demonstrated need to explore the performance of such a combination of technologies, and would help wearable mechatronic rehabilitative devices fulfil the impending requirements of Canada’s aging population.

1.2 General Problem Statement

As one of the largest disabled groups in Canada, paretic stroke patients need an innovative rehabilitation treatment due to the limited efficiency and limited availability of current therapist-based treatment. Robotic therapeutic devices look to be a viable solution to this problem, with proven results; however, the actuators they use are often large, bulky, and non-compliant, reducing their portability, safety, and ease-of-use. TCAs are desirable for use in wearable mechatronic
rehabilitative devices due to their high power to weight ratio, large force and displacement production abilities, their compliant nature, and their innate advantages over other smart materials. However, practical implementation of these actuators are limited by their low efficiency and low gross force production capabilities. In order to overcome these deficiencies, a novel woven pennate structure actuator comprised of TCA fibres is proposed, with aims to improve the stress and stroke production capabilities of the actuator while also increasing the efficiency.

1.3 Research Objectives and Scope

This thesis aims to develop the design of a woven bi-pennate artificial muscle actuator, characterize its performance, and determine its best use cases according to its capabilities.

The primary objectives of this thesis are as follows:

1. To determine the minimum and maximum force production of the actuator.
2. To determine the minimum and maximum displacement of the actuator.
3. To characterize the performance of the actuator for variable recruitment scenarios.
4. To calculate the mechanical efficiency of the actuator compared to its electrical power consumption.
5. To implement variable force control.
6. To assess the best therapeutic use cases of the actuator based on its performance.

1.4 Overview of the Thesis

The structure of this thesis is summarized in the outline below:

Chapter 1 Introduction: Explains the motivation and general problem statement that this thesis aims to address.
1.4 Overview of the Thesis

Chapter 2  Literature Review: Presents a review of wrist neurorehabilitative therapy, twisted coiled actuator modelling and characterization, control implementations of twisted and coiled actuators, wrist rehabilitation orthosis design, pennate structure and variable recruitment control of actuators, and flexible textile-embedded artificial muscles.

Chapter 3  Preliminary Work: Presents the methods used to fabricate the twisted coiled actuators, the methods used to characterize the various sensors, and the pilot experimentation done to establish the experimental parameters required to have statistically significant data.

Chapter 4  Actuator Characterization: Presents the testing and characterization of the stroke, force, and frequency production capabilities of the constructed actuator.

Chapter 5  Force Control Implementation: Presents the creation and testing of the force control scheme constructed using the results of the force characterization of the actuator.

Chapter 6  Conclusions and Future Work: Emphasizes the contributions made through this work and provides recommendations for future work.

Appendix A  Engineering Drawings: Provides detailed schematics of the physical components created for this work.

Appendix B  MATLAB Code: Describes the MATLAB code used for data segmentation, preparation, and analysis.

Appendix C  Arduino Code: Describes the Arduino code used to control the experimental sequence and data collection.
Chapter 2

Literature Review

This chapter presents a review of the literature examined upon which the need for this thesis and the principles of its operation were established. Post-stroke wrist rehabilitation, robot-assisted rehabilitation, twisted coiled actuators, fabric-embedded artificial muscles, biomimetic muscle configurations, and control systems were researched in order to develop design requirements for the proposed actuator. Academic literature was searched for using Google Scholar, which compiled resources from sources such as IEEExplore, PubMed, Scopus, IOP, and other relevant publishers and databases, between the time of May 2019 and June 2020. Keywords used in this search included post-stroke arm paresis, robotic arm rehabilitation, robotic wrist orthoses, artificial muscles, embedded artificial muscles, twisted and coiled actuators, biomimetic actuators, pennate actuators, compliant actuators, polymer actuator control systems, and combinations and variations of those key words. A total of 149 relevant papers were found and used in this thesis.

2.1 Wrist Neurorehabilitation

Many people who suffer from stroke have post-stroke symptoms which include weakness or partial paralysis on one side of their body, often experienced in the upper limb [17]. This is referred to as post-stroke arm paresis [3], and can make many acts of daily living difficult or unattainable for patients, limiting their ability to live independently. Thankfully, this is a treatable condition, with many patients able to regain motor function after a rehabilitative therapy regime [6, 18, 19], which also serves to improve range of motion, muscle strength, and coordination. These rehabilitation
2.1 Wrist Neurorehabilitation

Methods vary in implementation and duration based on the needs of the patient, but they generally follow the following procedure.

In the beginning of the therapy regime, a therapist will move an affected joint through its range of motion, revolving about one axis. The therapist will carefully apply a force to the limb to move it as far as is comfortable for the patient, monitoring the patient constantly for feedback. For the wrist, therapists will typically manipulate the hand through wrist flexion and extension, and ulnar–radial deviation, all of which can be seen in Figure 2.1. After this mode of therapy has run

(a) Radial deviation is rotation of the wrist towards the thumb.

(b) Ulnar deviation is rotation of the wrist towards the pinky.

(c) Wrist flexion is rotation of the wrist towards the palm.

(d) Wrist extension is rotation of the wrist towards the back of the hand.

Figure 2.1: Different wrist rehabilitation motions.
its course, the therapist will assist the patient while they attempt to actuate the joint, reducing assistance as the patient’s capabilities improve. Once the patient is fully capable of actuating their joint on their own, the therapist will apply pressure to provide some resistance to their efforts to further improve their strength and motor control, and begin to have them complete acts of daily living and other task-oriented exercises [20–22]. While this therapy regime is known to be effective, it is also known to be quite time-consuming, and its availability is limited by the number of therapists, the required hours of contact time between the patient and therapists, and the ability for patients to come in for regular, lengthy appointments [4]. Because of this limited availability, it has been shown that only 43.3% of stroke patients in Ontario receive care with specialists, doctors, or nurses specialized in stroke care [23]. There is a well-defined need to improve the availability of physiotherapy treatments, so other alternative methods of delivering this therapy must be considered.

2.2 Robot-Assisted Therapy

Robot-assisted rehabilitation has been used to great effect to mitigate these issues, providing similar or better clinical results in terms of range of motion, increased strength, and coordination [17, 18, 20, 24–26], while also improving the availability of therapy and reducing the time and effort demands on therapists [19]. Table 2.1 shows a list of currently available therapeutic robots. In order to limit the search to those pertaining to this thesis, only robots that were designed for one degree of freedom, flexion and extension based physical therapy of the upper limb are presented here.
Table 2.1: Examples of neurorehabilitative robots found in the literature, with affected joint, actuator, and portability listed.

<table>
<thead>
<tr>
<th>Primary Designer</th>
<th>Joint</th>
<th>Actuator Type Used</th>
<th>Stationary/Portable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cheng [27]</td>
<td>Elbow</td>
<td>DC motors</td>
<td>Stationary</td>
</tr>
<tr>
<td>Colombo [28]</td>
<td>Wrist</td>
<td>Servomotors</td>
<td>Stationary</td>
</tr>
<tr>
<td>Cozens [29]</td>
<td>Elbow</td>
<td>Servomotors</td>
<td>Stationary</td>
</tr>
<tr>
<td>Kiguchi [31]</td>
<td>Elbow</td>
<td>DC motors</td>
<td>Stationary</td>
</tr>
<tr>
<td>Martinez [32]</td>
<td>Wrist</td>
<td>DC motors</td>
<td>Stationary</td>
</tr>
<tr>
<td>Mavroidis [33]</td>
<td>Elbow</td>
<td>DC motors</td>
<td>Stationary</td>
</tr>
<tr>
<td>Oda [34]</td>
<td>Elbow</td>
<td>MRF Brakes</td>
<td>Stationary</td>
</tr>
<tr>
<td>Ogce [35]</td>
<td>Elbow</td>
<td>DC step motors</td>
<td>Portable</td>
</tr>
<tr>
<td>Pehlavin [36]</td>
<td>Wrist</td>
<td>DC motors</td>
<td>Stationary</td>
</tr>
<tr>
<td>Pylatiuk [37]</td>
<td>Elbow</td>
<td>Hydraulics</td>
<td>Portable</td>
</tr>
<tr>
<td>Song [38]</td>
<td>Elbow</td>
<td>AC servomotors</td>
<td>Stationary</td>
</tr>
<tr>
<td>Song [39]</td>
<td>Wrist</td>
<td>DC servomotors</td>
<td>Stationary</td>
</tr>
<tr>
<td>Stein [40]</td>
<td>Elbow</td>
<td>DC motors</td>
<td>Portable</td>
</tr>
<tr>
<td>Vanderniepen [41]</td>
<td>Elbow</td>
<td>Electric motors</td>
<td>Stationary</td>
</tr>
</tbody>
</table>

As can be seen from Table 2.1 these robotic therapy devices typically take the form of stationary units that interact with patients using electromagnetic motors, with a few non-stationary devices that interacted with patients’ elbows, rather than wrists. While they are effective, they are attached to a wall, table, or the ground due to their size and mass, and cannot be used for at-home rehabilitation. Smaller, lightweight, wearable devices are desired as they increase the possibility for mobile or at-home rehabilitation, which would increase therapy compliance and frequency, increasing efficacy of the treatment [6]. One hurdle to overcome with these devices in making them smaller and lighter weight is their use of electromagnetic motors as actuators. These are dense, heavy, and rigid actuators, which make them less than ideal for use in wearable devices [42].

Research has been conducted to find viable replacements for DC motors as actuators in robotic therapeutic devices. Popular choices of replacement actuators include pneumatic artificial muscles, shape-memory alloys (SMAs), and electroactive polymers (EAPs). These actuators are lightweight, capable of producing the required forces, and compliant; however, they have other considerations.
that limit their practical use in commercial therapeutic devices. Pneumatic artificial muscles, while the actuators themselves are light and compliant, require pressurized valves, tubing, and air tanks or compressors in order to work properly, which are bulky, heavy, and rigid, making them non-ideal for wearable devices [43]. As well, they are exceedingly loud, making them non-ideal for regular and at home use. SMAs are limited by their relatively small linear stroke (in the motion sense, not the cerebrovascular sense) when in contraction, and possess large hysteresis [44], making them non-suitable for this type of application. EAPs require immense voltages to operate properly, on the order of kilovolts, which is considered unsafe for medical and wearable devices [45, 46]. They also require additional electronic converters to operate properly [47] and have relatively small stroke lengths [48], further limiting their use. Therefore, a different form of artificial muscle is required that can overcome these limitations.

2.3 Twisted Coiled Actuators

Twisted coiled actuators (TCAs) are a class of thermally activated actuators currently being investigated for use as artificial muscle fibres. These actuators are comprised of various thread-like polymers that possess an anisotropic thermal expansion coefficient, meaning that when they heat, they expand in one direction while contracting in the other [49]. Of particular use in this domain are materials that expand radially and contract axially when heated, such as polyethylene, carbon nanotubes, and nylon 6, 6 [50]. These thread-like polymers are then supercoiled, forming a spring-like structure that is the actuator’s final form [10]. When this spring-like structure is heated, it contracts axially, causing the coils to shrink in on one another, giving it muscle-like actuation abilities [10]. This contractive behaviour on a macro-scale is similar to increasing the stiffness of a linear spring; at a constant level of loading, it will contract the muscle, while at a constrained length it will increase the axial force applied by the actuator [51, 52]. Their stroke range is typically limited by the amount they are capable of contracting until their coils begin to touch [53], which can be extended by coiling the TCAs about a mandrel, albeit at the expense of force production capabilities [54]. Of particular interest are the nylon 6, 6 variants of these actuators, as they are the least expensive per unit weight while still outperforming human muscle fibres on a per-mass basis. These were the TCAs investigated in this thesis, and are what will be
referred to henceforth when referring to TCAs without any further qualifiers.

Non-mandrel-coiled versions of nylon TCAs are enticing prospects as they are able to produce strains of up to 34% while being lightweight, having a small footprint, being flexible, and producing stresses of up to 35 MPa [10, 55–58]. These actuators have a maximum safe operating temperature of 120°C, after which they are prone to breaking [51, 58]. Below their glass transition temperature of 40–50°C, they tend to give lessened response per degree temperature change [51], giving them a normal operating temperature range of between 40°C and 120°C. To facilitate electrothermal control of these actuators and remove the need for additional heating components, they are often coated in a conductive silver paint, which allows electrical power to be delivered to them. The innate resistance of the actuator dissipates the electrical power delivered to it as heat, a process known as Joule heating, in turn causing the muscle to contract [11]. While this makes implementing these actuators much more straightforward, the silver paint is prone to chip and flake during large contractions. Large strokes tend to limit the cycle-life of the actuator to a few cycles, while strokes limited to around 10% see much higher cycle-life [11]. This still results in actuation capabilities exceeding other smart material actuators [9], which makes this limitation acceptable.

While these actuators are promising, they are not without other limitations that make their widespread implementation difficult. While their stress production is quite large when normalized across their filament diameter, their individual gross force production can be less than 1 N [12, 48, 59–61] per actuator, which is undesirable. As well, their efficiency in converting electrical power delivered to mechanical power produced is quite low, achieving 1–2% efficiency [10, 11, 62] during contraction. Finally, their actuation frequency has also been noted as limiting their adoption [50, 62, 63], as the time required to heat and cool them is often too slow for some purposes. These limitations must be overcome if these actuators are to be used for the purpose of driving wearable therapeutic devices.

**2.4 Fabric-Embedded Artificial Muscles**

Fabric-embedded soft actuators are desirable in the wearable therapeutic device field, and wearable robotics as a whole, due to their lightweight nature, ability to assume complex shapes, intuitive usage, easy incorporation into garments that would already be worn, collapsibility, versatility,
and additional safety due to their flexibility [45, 47, 48, 64–67]. Examples found in the literature have focused on the use of soft actuators such as pneumatic actuators, SMAs, EAPs, and TCAs [45, 47, 48, 65–67], embedding them into fabrics via sewing, embroidery, bundling, polymer-based sheathing [68–70], and weaving. No examples were found of TCAs being embedded into fabrics, so the principles observed from embedding other similar actuators must be extrapolated.

Sewing, knitting, and braiding were found to have been investigated using pneumatic actuators [71–76] and SMAs [77–79], while polymer sheathing and bundling examples were found for TCAs [68, 69, 80]. Each study mentioned noted that the embedding of the actuators in their respective manners were able to create flexible actuating textiles with little to no drawbacks, although some reported a constrained range of motion [81], and the pneumatic actuator implementations were noted as being non-ideal due to the bulk of their required support components such as air tanks and compressors [73, 75] and swelling from some geometries [71, 72], while the SMAs were noted as having limited stroke production [66, 77, 82], which is also non-ideal. Bundled TCAs were found to increase their efficiency [80] compared to non-bundled versions, indicating that there are possible advantages from embedding them into fabrics. TCAs were also noted as being better choices for textile-based actuators due to their ability to be electrically controlled [43], making them more compact than pneumatic actuators, their aforementioned advantages in controllability and strain production compared to SMAs [66], their low cost, and their light weight [42].

While these are viable methods of creating flexible actuating textiles, woven textile actuators were found to be more desirable than the methods listed above [48, 83, 84]. This is because investigations using EAPs [13, 59] and SMAs [78, 82] have seen them able to produce an amplified stroke [13, 82] and stress [48, 59, 78] compared to non-embedded actuators, making them appealing for achieving larger ranges of motion, which are particularly desirable in wearable devices [42]. As well, weaving allows for the embedding of sensors and other electronics into the fabric matrix [83]. These results indicate that weaving is a superior method of embedding actuators into textiles; however, there was a noted lack of examples found that implemented this method with TCAs. If the observed increases in stroke and force production could be replicated using TCAs, which have been shown to outperform other smart material actuators, while also seeing the increased efficiency from embedding in fabrics mentioned before, woven TCA matrices could be the actuator
required to progress the field of wearable devices, therapeutic or otherwise.

In addition to finding the need to investigate woven TCA actuating textiles, strategies towards their implementation were explored. From studies involving pneumatic actuators [76, 85] that investigated different yarn materials, it was recommended that the stiffness of the material comprising the fabric matrix should be less than or equal to that of the actuator used so as to not limit stroke production. This was confirmed to hold true as well with SMAs [77], indicating its applicability to electrothermally controlled actuators. In regards to weaving around nylon-based TCAs, this means that the yarn used should be less stiff or of similar stiffness to nylon. Studies and surveys [47, 84, 86], in addition to common sense, recommend that the fabric matrix should be thermally and electrically insulating, so as to shield the user from the hot, conductive actuating fibres. Pure nylon, being thermally and electrically insulating and matching the stiffness of the nylon actuators, therefore, is a good choice for use as the fabric material. It was also found that having increased biomimicry would benefit actuating textiles [43, 70, 87] increasing mechanical stability, compliance, and modularity, all of which would improve the performance and design of wearable robotic devices.

2.5 Biomimetic Pennate Structure

A particular instance of biomimicry that is interesting is the arrangement of actuating muscle fibres, like TCAs, in a pennate structure similar to those found in biological muscle structures [88–90], examples of which can be seen in Figure 2.2. This structure uses individual muscle fibres arranged in fascicles flaring out of a central tendon at an acute angle. This results in a dynamic gearing ratio between the fibre contraction and the contraction of the muscle as a whole, changing the force and velocity of the muscle as it nears the limits of its contractions to lessen the chance of hyperextending or otherwise damaging the muscle and surrounding tissues from its efforts [88]. In addition, this structure also allows for multiple fibres to work in concert with one another without interference [89], which is greatly desired as heavier loads require greater force outputs, which multiple TCAs can provide [91, 92].

Research has been done attempting to implement these kinds of structures with pneumatic artificial muscle actuating fibres [15, 16, 93], which saw success in improving the gross force
2.6 Control Strategies

Control systems are typically implemented to alter the overshoot, oscillatory behaviour, and response time of a system, allowing designers to meet the requirements of their specific design. In the context of TCAs, they are often implemented to affect either the force or stroke produced by the actuators in contraction. Limiting factors that may affect their control without an appropriate scheme are their small amounts of hysteresis [94], non-linear behaviour [95], compliance [96], and slow cooling times [97]. While the final item of that list cannot be accounted for without production of the actuator [93] and similar gearing behaviour to that of biological muscles [16]. It was also observed that pennate-based pneumatic muscles were able to match or exceed the performance of simple parallel-bundled actuators [15], indicating that loss of performance due to angulation should not occur. One example of a pennate structure could be found that incorporated TCAs [14], which did note that increasing the number of active TCAs was able to linearly change the amount of force and stroke produced, but did not comment on efficiency or on appropriate control systems that should be employed to control the actuation system. Further investigation is required to determine if the same effects can be observed from pennate orientation of TCAs as in pneumatic muscles, whether these same benefits can be observed when inserted into a woven fabric matrix, and if the implementation of a control system can be made beneficial to their use.

![Feathers and muscles are both examples of pennate structures found in nature.](a) "feather11093f" by Kel and Val, use licensed under CC BY 2.0. (b) "Magnified view of a Tendon" by Manu5, use licensed under CC BY-SA 4.0. ](figures)

Figure 2.2: Feathers and muscles are both examples of pennate structures found in nature.
additional components such as hydraulic [98] or mechanical [97, 99] forced cooling, the others can be addressed with various compensation schemes. Numerous previous efforts have seen success incorporating a feedforward mechanism [8, 12, 95, 100–104], with feedback elements, which predicts the effects of a disturbance and accounts for them before they have time to impact the system. With respect to TCAs, this would be accounting for the effects of delivering power to the TCA before it has time to heat, giving a prediction of its behaviour as a consequence of this power delivery. Using control schemes with these predictive components, the aforementioned implementations were able to achieve high degrees of accuracy in following a reference signal [95], decreased overshoot [100], decreased recovery time [8], and fewer oscillations about a target force setpoint [102]. Both as complements to feedforward components, and as their own controllers, proportional-integral-derivative (PID) controllers are often used [8, 12, 95, 97, 100, 102, 104–107]. These operate by having a multiplicative (proportional), integrating (integral), and differential (derivative) component applied to the system, changing the characteristics of its response signal [108]. These can be used in conjunction with feedforward mechanisms for improved performance, as examples in the literature have shown [8, 12, 95, 100, 102, 104].

While these control schemes have been shown to have great effect on single-TCA [12, 94–101, 103–107, 109] or antagonistic dual-TCA systems [8, 102], no implementations could be found for fabric-embedded TCAs, and the examples for multi-fibre TCA control did not use similar control schemes [92, 110]. There is an identifiable gap in the literature with respect to the use of feedforward-PID composite controllers with fabric-embedded multi-fibre TCA-based actuation systems, which should be rectified to both check the performance of such implementations and see the performance gains compared to implementations in systems with fewer TCAs.

2.7 Developing System Requirements

After establishing a need to investigate the actuation capabilities of a woven bi-pennate TCA actuation module, the exact physiological and therapeutic design requirements of the module had to be obtained and considered. It was decided that only therapies involved with wrist flexion and extension would be considered when developing actuator requirements, in order to limit the scope of the project and allow for actionable design parameters to be collected. In particular, the
force production, stroke production, and contraction period requirements were developed through research. Because the wrist behaves similarly to a hinge joint with a single axis of rotation when acting in flexion and extension, the linear force and stroke production requirements of the actuator must be translated from their rotating equivalents, torque and angled range-of-motion, respectively. To determine the viability of the actuator, the torque and range of motion requirements were collected from examples in the literature, and then the required lever arm to achieve them with the produced force and stroke of the actuator was calculated after completion of the characterization experiments, to determine if it was feasible to achieve these values. The collected requirements are presented in Table 2.2.

Table 2.2: Design requirements found in the literature. All values deviating from stated units have parenthetical notes clarifying why they are different. N/A values signify that the information was not present in that piece of literature.

<table>
<thead>
<tr>
<th>Paper</th>
<th>Authors</th>
<th>Torque (Nm)</th>
<th>Range of Motion (Degrees)</th>
<th>Contraction Period (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Robotic Forearm Orthosis Using Soft Fabric-Based Helical Actuators</td>
<td>J. Realmuto and T. Sanger</td>
<td>1.70</td>
<td>90</td>
<td>4.17</td>
</tr>
<tr>
<td>A Soft Robotic Orthosis for Wrist Rehabilitation [112]</td>
<td>N. Bartlett, V. Lyau, W. Raiford, et al.</td>
<td>N/A</td>
<td>91</td>
<td>N/A</td>
</tr>
<tr>
<td>Bioinspired 10 DOF Wearable Powered Arm Exoskeleton for Rehabilitation [113]</td>
<td>S. Manna and S. Bhaumik</td>
<td>0.43</td>
<td>160</td>
<td>N/A</td>
</tr>
<tr>
<td>Design of a Cable-Driven Arm Exoskeleton (CAREX) for Neural Rehabilitation [114]</td>
<td>Y. Mao and S. Agrawal</td>
<td>2.70</td>
<td>180</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Table 2.2: Design requirements found in the literature. All values deviating from stated units have parenthetical notes clarifying why they are different. N/A values signify that the information was not present in that piece of literature.

<table>
<thead>
<tr>
<th>Paper</th>
<th>Authors</th>
<th>Torque (Nm)</th>
<th>Range of Motion (Degrees)</th>
<th>Contraction Period (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design of Wrist Gimbal: a Forearm and Wrist Exoskeleton for Stroke Rehabilitation [32]</td>
<td>J. Martinez, P. Ng, S. Lu, et al.</td>
<td>1.76</td>
<td>180</td>
<td>N/A</td>
</tr>
<tr>
<td>Developing a Whole-Arm Exoskeleton Robot with Hand Opening and Closing Mechanism for Upper Limb Stroke Rehabilitation [115]</td>
<td>Y. Ren, H. Park, and L. Zhang</td>
<td>1.25</td>
<td>100</td>
<td>N/A</td>
</tr>
<tr>
<td>Development and Control of a ‘Soft-Actuated’ Exoskeleton for Use in Physiotherapy and Training [4]</td>
<td>N. Tsagarakis and D. Caldwell</td>
<td>4.00</td>
<td>140</td>
<td>N/A</td>
</tr>
<tr>
<td>Development of a Hand Motion Assist Robot for Rehabilitation Therapy by Patient Self-Motion Control [116]</td>
<td>H. Kawasaki, S. Ito, Y. Ishigure et al.</td>
<td>1.30</td>
<td>160</td>
<td>N/A</td>
</tr>
<tr>
<td>Development of an Active Upper-Limb Orthosis [117]</td>
<td>A. Alutei, A. Vaida, D. Madru, et al.</td>
<td>3.062 N (force required to lift hand)</td>
<td>N/A</td>
<td>17.5 °/s (dependent on range of motion)</td>
</tr>
<tr>
<td>Development of Wrist Rehabilitation Equipment Using Pneumatic Parallel Manipulator [118]</td>
<td>M. Takaiwa and T. Noritsugu</td>
<td>0.40</td>
<td>160</td>
<td>5.0</td>
</tr>
</tbody>
</table>
Table 2.2: Design requirements found in the literature. All values deviating from stated units have parenthetical notes clarifying why they are different. N/A values signify that the information was not present in that piece of literature.

<table>
<thead>
<tr>
<th>Paper</th>
<th>Authors</th>
<th>Torque (Nm)</th>
<th>Range of Motion (Degrees)</th>
<th>Contraction Period (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamics of Wrist Rotations [119]</td>
<td>S. Charles and N. Hogan</td>
<td>0.40</td>
<td>N/A</td>
<td>70 °/s (dependent on range of motion)</td>
</tr>
<tr>
<td>Hyperspasticity for Ergonomie Design of a Wrist Exoskeleton [120]</td>
<td>M. Esmaeili, N. Jarassé, W. Dailey, et al.</td>
<td>0.71</td>
<td>140</td>
<td>N/A</td>
</tr>
<tr>
<td>Powered Orthosis and Attachable Power-Assist Device with Hydraulic Bilateral Servo System [121]</td>
<td>K. Ohnishi, Y. Saito, T. Oshima, et al.</td>
<td>0.80</td>
<td>90</td>
<td>N/A</td>
</tr>
<tr>
<td>Psychophysical Study of Six Hand Movements [122]</td>
<td>V. Ciriello, S. Snook, B. Webster, et al.</td>
<td>1.30</td>
<td>N/A</td>
<td>4.0</td>
</tr>
</tbody>
</table>
Table 2.2: Design requirements found in the literature. All values deviating from stated units have parenthetical notes clarifying why they are different. N/A values signify that the information was not present in that piece of literature.

<table>
<thead>
<tr>
<th>Paper</th>
<th>Authors</th>
<th>Torque (Nm)</th>
<th>Range of Motion (Degrees)</th>
<th>Contraction Period (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMA Based Wrist Exoskeleton for Rehabilitation Therapy [127]</td>
<td>D. Serrano, D. Copaci, L. Moreno, et al.</td>
<td>0.50</td>
<td>175</td>
<td>4.0</td>
</tr>
</tbody>
</table>

From Table 2.2, the minimum, maximum, and average torque, range of motion, and contraction period requirements can be found. These are presented in Table 2.3. These show that there is a fairly large range of design requirements for wrist neurorehabilitation in flexion and extension due to the presence of a few outliers. While outliers were kept in mind when comparing the performance of the designed actuator, the average requirements were given more importance due to being more representative of the body of literature.

Table 2.3: Minimum, maximum, and average design requirements. Period values, where relevant, were calculated from given angular velocity values and range of motion values.
2.8 Conclusions

In this chapter, the motivation, methods, and requirements for wrist rehabilitation for post-stroke patients have been reviewed. This review included a survey of literature regarding wrist rehabilitation, the state of the art of robot-assisted wrist rehabilitation, TCAs, fabric-embedded artificial muscles, biomimetic pennate structures, control strategies, and design requirements for actuators for soft robotic therapeutic devices. A need was established to replace conventional actuators in therapeutic devices with TCAs to make them lighter and flexible, the TCAs’ operation in contraction was explored, the principles of their implementation in a biomimetic fabric matrix was investigated, control strategies were reviewed, and requirements by which the designed actuator would be compared were found. With this knowledge at hand, preliminary experimentation could begin on the designed actuator.
Chapter 3

TCA Development and Calibration

This chapter details the methods used to create the TCAs from raw materials and train them to make them into usable artificial muscles for implementation in the woven bi-pennate actuating module, which was then characterized and tested in experiments detailed in further chapters. As well, it explains the sensor calibration methods used to increase the validity of the results obtained.

3.1 TCA Fabrication

In order to construct the TCAs for use in the experimental evaluation of the actuator, a 2.5 metre length of silver coated nylon thread (Shieldex 4 ply coated nylon thread part no. 260151023534) was attached at one end to a DC motor. The other end was attached to 175 g calibration weights, such that the weights kept the thread taught while the motor spun to twist the thread. This was achieved using the setup seen in Figure 3.1 developed by Brandon Edmonds [62] for the purpose of fabricating TCAs.
3.1 TCA Fabrication

Figure 3.1: TCA fabrication setup.

As seen in Figure 3.1, the motor was kept horizontal in a weighted housing (to prevent slippage) while the weighted end of the thread was attached to a wheeled runner, allowing the thread and weight to move as the thread coiled. The weights were also attached to the runner and draped over a pulley over the edge of the table, to facilitate the tautening of the thread. The DC motor was run such that it twisted the thread until it first became coiled, then supercoiled, both of which can be seen in Figure 3.2. This supercoiling occurred when the thread completely coiled upon itself twice, significantly changing the width and shortening the thread structure. The DC motor housing was adjusted as it wound the thread so the weights were always hanging over the edge of the table.
The supercoiled thread was then clamped with non-insulated quick-disconnect terminals (Electerm Part no. BB2-MQ1) in intervals such that the length of thread between terminal bases was 100 mm apart. Care was taken to select stretches of supercoiled thread free of snags, runs, or irregularities. Excess material was then removed, such that the clamped terminals were the endpoints of a 100 mm length of supercoiled nylon thread. This completed structure is the untrained TCA, which can be seen in Figure 3.3.
3.2 TCA Training

In order for the TCAs to contract when heated, a 'training' procedure is required in order to change their structure to support such operations. To do this, a procedure and materials developed by Brandon Edmonds [62] was followed, as it had proven results. To begin this procedure, an untrained TCA constructed in the manner described in Section 3.1 was attached to a metal hook attached to a rigid wooden structure seen in Figure 3.4 at one end and a free-hanging 50 g calibration weight at the other to keep it taught. The TCA was attached at both ends using 3D printed hooks, held in place by sliding a stainless steel coiled spring pin (1.19 mm in diameter, 10 mm in length) through a hole in the printed hooks that lined up with the hole in the terminals at the end of the TCA. One of these hooks was then attached to an attach point in the setup, while the other was attached to the 50 g weight via a string. The weight was then draped over the edge of the table to facilitate tautening the TCA, with the string resting on a pulley to allow the weight to move.

Figure 3.4: TCA training setup. Left: a view of the training setup in its entirety. Centre: a TCA stretched between the structure and calibration weights, with attached alligator clips. Right: a view of the tautening weight suspension.
3.2 TCA Training

on the TCA when contracting or relaxing. Wired alligator clips were then attached to these steel pins and connected to a DC power supply (BK Precision 1671A). With the TCA installed, 0.8 A of current were applied through the TCA at 3.6 V, giving a total initial power delivery value of 2.88 W, less any possible losses from the use of the alligator clips. This heated the TCA due to the internal electrical resistance of the actuator. The temperature of the TCA was measured using a thermocouple (Agilent U1272A Multi-metre with thermocouple attachment). When the TCA was heated to 120°C Celsius, the DC power supply was turned off to prevent overheating and possible damage to the actuator. The TCA was allowed to cool to room temperature, verified via readings taken using the thermocouple. This total heating and cooling process was repeated for a total of five cycles per TCA. After this was completed, the TCA was considered trained and able to contract with the application of Joule heating. Its appearance can be seen in Figure 3.5.

![TCA appearance before training.](image1)
![TCA appearance after training.](image2)

Figure 3.5: Visual comparison between an untrained and trained TCA.
3.3 Stroke Sensor Calibration

Once the TCAs were fabricated and trained, a calibration process was required to account for intrinsic biases present in the sensors that would affect the results of the evaluation of the actuator. The SHARP GP2Y0A51SK0F Infrared Distance Sensor (IR sensor), the chosen stroke sensor, was placed on a lab bench, and connected to an ESP32 microcontroller in order to log analog readings from the sensor. Analog readings are unitless integer representations of voltages produced by sensors and other phenomena, and are used by microcontrollers because they operate in digital rather than analog scales, and need to discretize analog signals before they can be used in operations. In this instance the analog readings represented the voltage produced by the IR sensor every time it took a reading, which represented the distance between its IR emitter and an object in its line of sight. A double layer of masking tape was applied uniformly to a sheet of cardboard to provide a controlled reflectivity across its surface, and then placed on the lab bench such that it was exactly 147 mm (just shy of the maximum range of the sensor, measured via Vernier calliper) from the IR emitter of the sensor, perpendicular to the IR emitter and collector. This can be seen in Figure 3.6. One thousand analog reading samples were taken from the sensor and averaged using the microcontroller to get the average analog reading the sensor returned at that distance. This process was repeated, decreasing the distance between the emitter and cardboard sheet by 7 mm each set of readings until readings at a 2 cm distance (the minimum range of the sensor) had been taken. Using these averaged values, a mathematical function was then fitted to accurately predict the true distance of the sheet from the sensor from its obtained analog readings. This was done using Microsoft Excel’s trend line functionality. Fitting options were tuned until the choice with the highest coefficient of correlation was found. The curve and its trend line can be seen in Figure 3.7.
3.3 Stroke Sensor Calibration

(a) Top view of setup.  
(b) A straight edge was used to align components.

Figure 3.6: Setup used for stroke sensor calibration.

Figure 3.7: Curve fitting for displacement sensor calibration values.
3.4 Force Sensor Calibration

From that trend line fitting procedure, it was found that the equation that best expressed the relationship between the IR sensor’s readings and the actual distance from the IR emitter was the following:

\[ y_{Corrected} = 61763I^{-1.031}. \]

Where \( y_{Corrected} \) is the distance of the sheet from the IR emitter in mm and \( I \) is the digital representation of the analog value returned by the IR sensor to the microcontroller. This equation was evaluated for fit based on its coefficient of correlation, or \( R^2 \) value, which shows the degree to which the regressed formula fits the input data on a scale from one to zero. It was able to achieve an \( R^2 \) value of 0.995, indicating that this equation captured the relationship with minimal error.

3.4 Force Sensor Calibration

Similar to the IR sensor, the force sensor, selected to be the Bolson Tech 5 kg load cell (load cell) also had to be calibrated to remove intrinsic biases that could negatively affect the accuracy of the results of the actuator evaluation. To do this, the load cell was clamped to a levelled lab bench using a vice clamp, such that it overhung the edge of the bench and could have free-hanging weights affixed to its end. It was connected to a XFW-HX711 load cell amplifier, which was in turn connected to an ESP32 microcontroller in order to log readings from the load cell amplifier. This setup can be seen in Figure 3.8. While the cell was unloaded, one thousand samples of the load cell readings were taken and averaged in order to ascertain the zero offset of the sensor. These raw readings took the form of integer values returned by the load cell amplifier that represented the amplified voltage produced by the strain gauges attached to the load cell. These operate similarly to the analog readings described prior; however, because of the use of an amplifier to sense more minute changes in voltage than typically possible with an ananlog-digital converter, these integer values were required to be communicated over I\(^2\)C to retrieve the data from the amplifier. A 100 g calibration weight was then hung from the end of the load cell, and one thousand amplifier readings were taken and averaged. This was repeated, adding 100 g each repetition until the total loading on the cell was 2 kg. After all of these averaged readings had been taken, the zero offset was subtracted to ’zero’ the readings. Using these zeroed average values obtained from each
repetition, a polynomial function was fitted in order to accurately predict the true loading on the cell from the values returned by the attached amplifier. This was done in a similar manner to the IR sensor using Microsoft Excel’s trend line fitting feature. The curve and its trend line can be seen in Figure 3.9.

Figure 3.8: Setup used for force sensor calibration.

Figure 3.9: Curve fitting for force sensor calibration values.
From the procedure used to fit the trend line, it was found that the equation that best expressed the relationship between the zeroed load cell readings and the actual loading experienced by the load cell was the following:

\[ F_{\text{Corrected}} = 0.0027L. \]

Where \( F_{\text{Corrected}} \) is the force acting on the load cell in grams equivalent and \( L \) is the zeroed load cell reading. This equation had a coefficient of correlation of 1, indicating that this equation captured the relationship nearly flawlessly.

\[ \text{3.5 Temperature Sensor Calibration} \]

In order to account for non-linearities and other possible sources of error that could negatively affect the accurate reporting of data during experimentation, the temperature sensors, chosen to be K-Type thermocouples, needed to be calibrated in a similar manner to the stroke and force sensors. In order to do this, a trained TCA was put into the setup described in Section 3.2 such that it was held taught by a 50 g calibration weight, and connected to a DC power supply (BK Precision 1671A) in order to apply electrical power for the purpose of heating the TCA. A K-Type thermocouple, attached to a MAX6675 amplification module, was tied to the TCA using nylon sewing thread such that the weld bead was in direct contact with the TCA. Thermal grease (Halzniye 510) was applied liberally to ensure proper contact between the thermocouple and the TCA, to provide electrical insulation, and to remove any insulating air between the two components. A control thermocouple (Agilent U1272A Multi-metre with thermocouple attachment) was similarly affixed to the surface of the TCA. The MAX6675 module was then connected to an Arduino Uno microcontroller for the purpose of logging its readings. Ten samples were taken and averaged by the microcontroller to determine the average current temperature reading of the K-Type thermocouple. This was done in a free running loop, logging this averaged reading as often as possible. The room temperature of the TCA was noted in this way using the K-Type thermocouple, as well as read using the control thermocouple. The temperature of the TCA was then increased using Joule heating at 2.88 W, provided by the DC power supply, until the temperature was confirmed to be greater than 30\(^\circ\)C on the control thermocouple. The maximum temperature reading was noted on both of the
3.5 Temperature Sensor Calibration

thermocouples. The TCA was allowed to cool to its initial room temperature, before it was heated again, increasing the threshold temperature by $10^\circ$ C. This was repeated until the threshold temperature was $120^\circ$ C. This entire process was performed six times, replacing the used TCA with a new TCA between each of these iterations. The average at each temperature level was then taken across these six trials, giving a set of averaged control values and a set of averaged K-Type thermocouple values. These two sets were then used to fit a polynomial function in order to predict the actual temperature of the surface of the TCA from the values returned by the attached thermocouple and amplifier. This was done using the trend line feature in Microsoft Excel, where the degree and parameters of the trend line were tuned until the correlation coefficient was maximized. The curve and its trend line can be seen in Figure 3.11. From fitting the trend line, the equation that best represented the relationship between the K-Type thermocouple’s temperature reading and that of the control thermocouple’s was as follows:

$$T_{Corrected} = 0.9278T_{Sensor} + 0.1862$$
Figure 3.11: Curve fitting for temperature sensor calibration values.

where \( T \) is the temperature in \(^\circ\)C, the subscript \( \text{Corrected} \) denotes the corrected value, and the subscript \( \text{Sensor} \) denotes the read sensor temperature. With an \( R^2 \) value of 0.95, this equation was shown to have minimal error in capturing the relationship between the two sets of values.

3.6 Power Sensor Calibration

Much like the other sensors, the power sensors also needed calibration in order to remove non-linearities and other possible sources of error, which could have affected the results of the actuator evaluation. After completely constructing the printed circuit board (PCB) described in Appendices A.1.1–A.1.4, seen in Figure 3.12, TCAs were connected to the relevant TCA\(^+\) and TCA\(^-\) ports such that they would be controlled by four transistors, and their power draw would be measured by two of the INA219B power modules. All other ports were left open, such that all power going through the board was flowing through these TCAs. The circuit board was then connected to a programmable DC power supply (Rigol DP811) such that it would supply power to the TCA board, and also was connected to an ESP32 microcontroller in order to record the power values output by the power sensor.
3.6 Power Sensor Calibration

Figure 3.12: Control board constructed for the various experiments.

The ESP32 was programmed such that it turned the transistors on, allowing power to course through the TCAs, after which it took 1000 samples of the power consumption as reported by the INA219Bs. The microcontroller then averaged the samples, output them, and turned the transistor off to break the circuit and disallow any further power consumption. The power supply was set to supply 0.5 A with unbounded voltage, so that current was the limiting factor, and the program was run. The power supplied by the power supply was recorded and compared to the averaged sensor reading. This was repeated, increasing the current by 0.5 A at each repetition until 5 A were reached. A polynomial function was then found in order to predict the power consumption values of the system from the values returned by the power sensors. This again was done using Microsoft Excel’s trend line function, with the trend line parameters being tuned until the correlation coefficient was maximized. The curve and its fitted trend line can be seen in Figure 3.14.
3.6 Power Sensor Calibration

Figure 3.13: Setup used for power sensor calibration.

Figure 3.14: Curve fitting for power sensor calibration values.
From that trend line fitting procedure, the equation found that best represented the relationship between the INA219B power sensor consumption readings and the stated power supply was:

\[ P_{\text{Corrected}} = 0.2841P^2_{\text{Sensor}} + 0.6191P_{\text{Sensor}} + 0.625. \]

Where \( P \) denotes power consumed, the subscript \( \text{Corrected} \) denotes the corrected value after calibration, and the \( \text{Sensor} \) subscript denotes the raw reported sensor value. With a coefficient of correlation 0.997, this equation had minimal error in capturing the relationship between the two power values.

### 3.7 Force Production Pilot Study

In order to determine the number of trials required to find statistically significant data with an acceptable level of variance in regards to the force production capabilities of the actuator, a pilot study was performed. A set of TCAs was tested numerous times to analyze their mean force produced and their variance, and to extrapolate the number of samples required for the results to be usable in the construction of a confidence interval with a variance of 5% of the mean force produced, and a 95% confidence level. In order to do this, a rigid wooden testing structure was constructed, such that TCAs inserted into the structure would be strung between the load cell described above and a rigid steel attach hook. The structure was adjusted and clamped with vice clamps such that the TCAs were kept level, and stretched until they just became taught. The TCAs were attached to the load cell and structure using the same 3D printed hooks used for the training procedure, and secured in place with the same stainless steel coiled spring pins and alligator clips wired to a DC power supply. After the TCA was secured in the setup, a K-Type thermocouple was tied to the TCA using nylon sewing thread, with thermal paste applied to remove air gaps between the two components and ensure thermal conduction and electrical insulation. The thermocouple and load cell were both connected to an ESP32 microcontroller to log their respective data values. The load cell was tared to record the additional force produced, rather than the existing tensile force keeping the TCA taught. The TCAs were then heated using Joule heating, at a rate of 2.88 W (3.6 V at 0.8 A) supplied by the power supply. The temperature
3.7 Force Production Pilot Study

Figure 3.15: Setup used for force production pilot experiment, with thermocouple removed.

and force values were monitored until the TCA heated to a temperature over 80° C, at which point the power supply was turned off and the TCA allowed to cool to room temperature. This was repeated five times per TCA for 12 TCAs, giving a total of 60 trials of this force production pilot. Each TCA was fabricated and trained using the methods described, then discarded. Using their maximum force production values, the mean, median, and standard deviation of the data set were recorded, as shown in Figure 3.16.

Figure 3.16: Results from the force production pilot experiment. The force is reported in grams force equivalent, as that is the unit reported by the load cell. This can be converted to N by multiplying the value divided by 1000 by the gravitational constant 9.81 m/s².
The standard deviation value (represented by $\sigma$) and mean force obtained were then used to calculate the number of trials required to achieve statistically significant results. This was done by selecting the allowable risk of false positive results (Type I error, represented by the z-score $Z_\alpha$), false negative results (Type II error, represented by the z-score $Z_\beta$), and minimum discernible effect size (represented by $\Delta$), and applying these values to the standard sample size calculation to determine the minimum number of samples $n$ required to achieve the desired levels of certainty:

$$n = \frac{2\sigma^2 (Z_\alpha + Z_\beta)^2}{\Delta^2}$$  \hspace{1cm} (3.1)

$Z_\alpha$ was selected to be 1.96, representing a Type I error probability of 0.05, $Z_\beta$ was selected to be 0.80, representing a Type II error probability of 0.20, and $\Delta$ was selected to be $0.05 \times \mu$, where $\mu$ is the mean value. This $\Delta$ value was chosen to express the calculation in terms of a percentage, so that the sample size could remain pertinent to experiments with differing levels and means of production. This also necessitated dividing the standard deviation by the mean value produced, to also express it in terms of a percent. These values were substituted into Equation 3.1, and the number of samples required ($n$) was calculated:

$$n = \frac{2(19.85)^2 (1.96 + 0.84)^2}{(147.60 \times 0.05)^2}$$

$$n = 113.4$$

As seen, after rounding to the nearest integer, 114 samples are required at minimum to be able to conclude with the desired certainty that the variance of the force production of one actuating set of TCAs is 5% that of the mean. This would indicate that, for every pair of TCAs used in the bi-pennate woven actuator, there should be at minimum 114 trials to achieve that desired level of accuracy when characterizing the force production of the module.

3.8 Stroke Production Pilot Study

Similar to the force production characterization of the actuator, a pilot study was performed to determine the minimum number of trials required to find statistically significant data with an
acceptable level of variance for the displacement production of the actuator. A set of TCAs was tested numerous times to analyze their mean displacement produced, their variance, and extrapolate the number of samples required in order for the results to be usable in constructing a confidence interval of the desired variance of 5% of the mean value and a 95% confidence level. In order to do this, TCAs were hung from a levelled pole held rigidly by a vice clamp with a 150 g calibration weight attached such that the TCAs were held taught by their applied weight. The TCAs were attached using the same 3D printed hooks used for their training and the force production pilot study, and secured in place with the same stainless steel coiled spring pins and alligator clips wired to a DC power supply. A custom 3D printed IR sensor holder, as seen in Appendix A.2.1, was fit over the calibration weights such that the IR distance sensor was pointed at the ground below the TCAs and weights. A cardboard surface with a double layer of masking tape was placed under the IR emitter to ensure the same reflectivity as that in the IR sensor calibration. The sensor and printed holder had a cumulative mass of 30 g, giving a total mass of 180 g hanging from each TCA. The TCAs were then heated via Joule heating at a rate of 2.88 W.

Figure 3.17: Setup used for stroke production pilot experiment. Paper background included where required to increase visibility.
(3.6 V at 0.8 A) supplied by the DC power supply. Using the K-type thermocouple described above, affixed to the TCAs in a similar manner to that described in the temperature sensor calibration and force production pilot sections using nylon thread and thermal paste, the surface temperature of the TCAs were recorded using an ESP32 microcontroller. The distance from the cardboard surface was also logged, with the initial value being subtracted to give the TCA contraction displacement. These values were recorded until the TCAs heated to a temperature over 80°C, at which point the power supply was turned off and the TCAs were allowed to cool to room temperature. This was repeated on individual TCAs five times per TCA for 12 TCAs, giving a total of 60 trials of this displacement production pilot. Each TCA was constructed and trained for the purpose of this pilot study, then discarded. Using their maximum displacement production values, the mean, median, and standard deviation of the data set were recorded, shown in Figure 3.18. The obtained standard deviation and mean force values were then used to calculate the number of trials required to achieve statistically significant results in a similar fashion to that of the force production pilot experiment. The same z-scores of 1.96 for $Z_\alpha$ and 0.84 for $Z_\beta$ were used to keep the level of certainty consistent between the stroke and force production experiments, and again the minimum discernible effect $\Delta$ was expressed as a percent of the mean stroke produced to allow for the sample
3.8 Stroke Production Pilot Study

size calculation to remain relevant despite changing production values as the number of active TCAs increased throughout the experiment. These values were input into Equation 3.1, giving the following calculations:

\[
n = \frac{2 \times (2.05)^2 \times (1.96 + 0.84)^2}{(12.28 \times 0.05)^2} = 174.1
\]

After rounding this result up to the nearest integer, it was determined that 175 samples are required at minimum to be able to conclude with the desired certainty that the variance of the stroke production of one actuating set of TCAs is 5% that of the mean with the standard deviation and mean stroke produced values from this pilot study. While the methods and calculations from these pilot studies were determined to be sound, when looking at this and the 114 required sample size value from the force production pilot, at surface level both calculated values seemed very high. Looking in the literature, sample sizes used in characterization and validation experiments for soft actuators were as low as 5 [9], 3 [55], 3 [62], 3 [57], and 20 [104], which indicated that these calculated values were maybe higher than required.

Because of this possibility, it was determined that a measured, conservative approach to experimentation sample sizing should be taken. Rather than possibly repeating experiments superfluously, it was decided that 50 trials would be done for both stroke and force production, and the statistical significance of the results would be re-evaluated. The minimum discernible effect \( \Delta \) would be calculated based on the results up to that point, as a percentage of the mean value produced, by rearranging Equation 3.1 and dividing by the mean \( \mu \) as follows:

\[
\Delta = \left( \frac{1}{\mu} \right) \sqrt{\frac{2 \sigma^2 (Z\alpha + Z\beta)^2}{n}}
\]  

(3.2)

where each symbol refers to the same value as that in Equation 3.1, and using the experimentally found standard deviation and mean produced values found up to that point. If the \( \Delta \) value was found to be less than or equal to 0.05, it would indicate that the experimental results had met the desired level of certainty, and the experiment could be halted. Otherwise, the experiment would be continued, and another segment of 50 trials would be done for both stroke and force production,
with the process repeating until the minimum discernible effect became less than that threshold for the desired levels of certainty. It was decided that these trials would be split across ten actuators, such that each actuator was tested five times at each of the six activation levels for both modes of production. This would allow for analysis on whether there were significant differences between different instances of fabricated actuators. These same ten actuators would be used if another 50 trials would be required, with the same number of trials split between them.

3.9 Conclusions

In this chapter, the methods used to fabricate and train the TCAs used in the construction of the woven bi-pennate actuators was described. As well, the methods, results, and regressed equations obtained from the sensor calibration procedures were presented. These were important to complete, as these helped remove non-linearities and biases present in the sensors due to their innate characteristics and manufacturing, increasing their accuracy in reporting results, which were then used to characterize the actuators. However, a proper assessment process was not performed, which would have served to verify calibration results. This could have improved accuracy and reliability further, and should have been done to prevent this possible source of error. Such a program is recommended for future work. The force and stroke production pilot experiments were also presented. These gave a baseline for the mean production capabilities of the actuator, as well as the expected variance of the force and stroke of the actuators. These values were then used to determine the number of trials required for each activation pair of the actuator to have statistically significant results, and a methodical experimentation plan was decided on to maintain experimental validity. Having completed these steps, the characterization experiments could move forward unimpeded with maximized accuracy and statistical validity.
Chapter 4

Actuator Characterization

This chapter details the experiments used to characterize the stroke production, force production, and contraction period capabilities of the woven bi-pennate actuator. These include the results of the experiments, the methods used to analyze them, numerical regressions where appropriate, and the conclusions that can be drawn from these experiments.

4.1 Stroke Characterization

4.1.1 Purpose

The purpose of this experiment was to characterize the contractive capabilities of the woven bi-pennate actuator in an isotonic configuration, supporting a dead load vertically and contracting its constituent TCAs to raise the dead load. This was done to find the maximum contractive capability at each activation level after having hit a maximum average temperature threshold of 80°C, which was then analyzed for linearity to see to what degree the addition or removal of pairs of TCAs would affect stroke production, and how predictable the degree of the effect would be. This serves to help determine the applicability of the designed actuator for use in a wearable rehabilitative device, as it helps determine at what point the number of TCA pairs would be sufficient to provide adequate range of motion for rehabilitative exercises, as found in Chapter 2. The maximum contraction of the actuator and the energy it consumed in reaching that state were then to be used to determine the efficiency of the actuator in converting electrical power input into
4.1 Stroke Characterization

mechanical power output. By comparing this to other forms of soft, linear actuation presented in
the literature, one can determine the viability of the actuator in being applied in future designs,
as well as how it compares to current robotic rehabilitation solutions. Transient data were used
to determine a numerical relationship between the temperature and stroke produced per number
of active TCA pairs using a polynomial regression, which would then be further generalized to
create a relationship between temperature, number of active pairs of TCAs, and stroke produced.
This in turn would provide a way of gauging the maximum temperature required to actuate a
desired range of motion, and provide a basis for future endeavours in designing control systems for
position control. In total, all of these aspects of the experiment served to assess the capability of
the actuator and quantify its performance, for determining its viable operation range and enabling
the prediction of its contractive range of motion performance based on the number of active TCA
pairs and temperature achieved.

4.1.2 Experimental Methods

This experiment started with the fabrication of the necessary components. The control PCB shown
in Figure 4.1 from Section 3.6 was used once more, with all of the relevant components, including
the thermocouples, ESP32 microcontroller, Arduino Uno, and Rigol DP811 programmable DC
power supply attached. The TCA and objective sensor ports were attached to quick-release cable
modules (Cat6) to allow for quick connecting and disconnecting, with all of the TCA$^+$ ports
connecting to one module, TCA$^-$ ports corresponding to the left side of the actuator connecting
to another, and TCA$^-$ ports corresponding to the right side of the actuator connecting to another.
The UNO and ESP32 were connected such that they would communicate via SPI, the code for
which can be seen in Appendix C.3.1. PCB spacers were used to balance the PCB on the table,
seen in Appendix A.2.2 The tendons and holsters for the actuator were 3D printed according to
the drawings presented in Appendices A.2.3–A.2.5 from a Markforged Onyx Pro 3D printer from
Onyx Nylon filament. Stainless steel coiled spring pins (1.19 mm in diameter, 10 mm in length)
were then inserted into the holes of the same diameter in the tendons and epoxied to affix them
there. Pin caps were then 3D printed according to the drawings presented in Appendix A.2.6 using
a Stratysis Objet 260 Connex3 printer with VeroMagenta filament, with six caps being printed
4.1 Stroke Characterization

each for the left, right, and central designs. These components can be seen in Figure 4.2. Twenty-four gauge insulated copper wire was stripped, cut into 2.5 cm lengths, and inserted through the similarly sized holes in the pin caps. These were epoxied in place, and had non-insulated quick-disconnect terminals (Electerm part no. BB2-MQ1) soldered to their ends to provide probe points for debugging purposes. Excess wiring was trimmed, and 24 gauge insulated copper wire was soldered to the same terminals and connected to a quick release cable module to allow for quick disconnecting and connecting to the control PCB. Oversized pin caps were also created and printed as necessary, as some of the thermocouples were frayed in their insulating layers and could not fit through the normal-sized thermocouple guides. The thermocouples were then fed through the thermocouple guides in the pin caps and epoxied so that their weld bead was precisely 2 cm from the end of the thermocouple guide.

![Picture of PCB with components]

(a) Front view.  
(b) Back view.

Figure 4.1: PCB with all external components added in preparation for the characterization experiments.
4.1 Stroke Characterization

(a) Central tendon, front view. (b) Central tendon, side view. (c) Pin caps for each tendon.

(d) Vertical tendon holder, front view. (e) Vertical tendon holder, side view. (f) Vertical tendon holder, with side tendon.

(g) Side tendon with pins, front view. (h) Side tendon with pins, side view. (i) Spacer used to prop up and balance the PCB.

Figure 4.2: All of the 3D printed components fabricated for this experiment. The IR holder from Section 3.7 was used as well.
Having fabricated those components, 120 TCAs were then made and trained according to the procedures outlined in Sections 3.1 and 3.2, respectively. This number was chosen because ten actuators were desired, as determined from the results of the stroke production pilot experiment. Twelve TCAs per actuator were chosen because this would give six possible activation levels of the actuator, which was comparable to or greater than the number of active pairs tested in variable recruitment experiments performed on other artificial muscle actuators in the literature, where examples such as five levels [93], three levels [92], and five levels [80] were found. After preparing the materials necessary for constructing the woven bi-pennate actuators, the actuator itself was constructed. The upright support holsters were clamped to a levelled lab bench 141 mm apart, measured with a digital Vernier caliper, to create a 45° initial pennation angle for the actuator. The TCAs were then slid over the steel pins on either side and held in place by sliding the relevant pin caps through the hole in the steel pins, which created a friction fit. The TCAs were then attached to the central tendon in opposing pairs in a similar fashion, as seen in Figure 4.3.

After securing all of the TCAs to the tendons, 600 g of calibration weights were hung from the hook on the central tendon, with the custom built IR sensor holder used in the stroke pilot study slid over the bottom to record the distance from a piece of cardboard with two layers of masking

Figure 4.3: Actuator with TCAs strung between tendons, prior to thermocouple attachment.
tape (to maintain the reflectivity from the calibration process) during the experiment. The weights were to ensure that the actuator was taught during the weaving process. Thermocouples were then attached by tying them to their designated TCA using nylon sewing thread as shown in Figure 4.4a, ensuring that the numbered thermocouple went to the same numbered TCA. After they were attached, thermal paste (Halzniye 510) was applied liberally to eliminate air pockets and ensure good thermal conductivity and electrical insulation between the TCA and the thermocouple. This can be seen in Figure 4.4b.

After the thermal paste was applied, the actuator was woven using black 100% nylon 18 weight yarn (Part No. 1005019 from anniescatalog.com). This was done using a standard over-under manual weaving technique using the TCAs as the warp, compressing the weft with every pass to make the weave as tight as possible, seen in Figure 4.5a. One side was done at a time, with the weaving continuing until the thermocouples had been covered by a minimum of three loops on each

(a) Thermocouples after being tied to TCAs. (b) After thermal paste has been applied.

Figure 4.4: Thermocouple attachment process
4.1 Stroke Characterization

Side. Stages of the weaving process can be seen in Figure 4.5. Once both sides of the actuator had been woven, the mass applied to it was increased to 1.2 kg (plus the mass of the holder and sensor) to give it its experimental load of 1.23 kg, as seen in Figure 4.6. This mass was chosen to make the loading comparable on a per-TCA basis to the pilot study. The upright holsters were then adjusted to ensure that the $45^\circ$ pennation angle was still maintained, after which the actuator was considered fully fabricated and ready for experimentation.

Figure 4.5: Actuator weaving at various stages of completion, various views.
For the experiment itself, the number of pairs of TCAs being heated, or levels of activation of the actuator, were iteratively tested to determine its stroke production capacity. This was done by choosing a pair of TCAs randomly (to remove selection bias) to designate it as ‘activated’. This pair was allowed to heat, being supplied a maximum of 30 W (10 V at 3 A) of power by the power supply. Controlled by the ESP32 microcontroller, the transistors were kept open to allow the power to course through the activated pair while closing off the other TCAs. The activated TCAs were heated until they both reached a temperature of 80°C as read by the thermocouples. If one TCA heated to that temperature before the other, its power was cut off by closing its transistor until its temperature went below 80°C, when its transistor was opened again until it went over 80°C again, when it would be closed once more. This oscillation was maintained until both active TCAs reached the 80°C threshold. Once they achieved this temperature, the power was cut off to both TCAs by closing their transistors to allow them to cool. They were kept closed until all TCAs reached a temperature below 30°C Celsius. After having cooled down, a random activated pair would be chosen again, and the process would begin anew. This would be repeated for a total of five times, giving a total of five trials for that actuator at that activation level, in accordance with results of the analysis from the stroke production pilot experiment. While these trials were running, the temperature, power, distance, and time were logged by the ESP32 in a free running loop, which was found after preliminary testing to give a sampling rate of 0.11 seconds.

After this set of trials was complete, the power made available to the actuator was increased by 10 W by increasing the current available by 1 A. The number of activated pairs of TCAs was also increased to two active pairs. These were again randomly selected, with care being taken to ensure duplicate pairs would not be selected for any trial. A set of five heating and cooling trials were performed following the same procedure as above, only that two sets of TCAs were active at a time. This was repeated, increasing the number of activated pairs by one each repetition, until the number of activated pairs reached six, the total number of pairs in the actuator. Each time the number of activated pairs increased, the power available to the actuator was increased by 10 W by increasing the current available by 1 A, until eventually 80 W were made available at 10 V and 8 A. In total, five trials were completed per activation level, for a total of 30 trials per actuator. After these were complete, the actuator was disassembled and a new actuator was constructed using a
new set of twelve TCAs, following the construction steps outlined above. The experimentation process was then repeated on this actuator, yielding a further 30 trials worth of data. This was repeated until ten actuators had been tested, giving a total of 50 trials per activation level. The microcontroller code used for this experiment can be found in Appendix C.3.2.

4.1.3 Analytical Methods

In addition to employing the methods in Sections 3.7 and 3.8 to determine if the number of trials was sufficient for statistical relevancy, further analysis was employed to determine if the results of each activation level were significantly different, as well as to construct confidence intervals of their possible values. The method by which the significance of their difference was determined was through a repeated measures one-way analysis of variance (ANOVA) test of the results at each activation level, comparing each level to each other level to determine if they differed to a statistically significant degree. One-way ANOVA tests compare the means between groups, testing the null-hypothesis for each by determining their variance from a common mean value [130]. They do this by calculating a value within an $f$-distribution based on the mean values of the data and group characteristics. These correspond to $p$ values, which give the likelihood that the groups reject the null-hypothesis, meaning they are statistically significantly different. If the $p$ value is
less than a critical threshold, known as the $\alpha$ value, there is a $(100 - \alpha\%)$ chance that the groups are statistically significantly different without there being a false positive. Typically, the $\alpha$ is chosen to be 0.05, meaning that if the $p$ value corresponding to the calculated $f$ test statistic is less than 0.05 there is a 95% level of certainty that the groups are statistically significantly different without a false positive. The repeated measures ANOVA differs slightly, in that it assumes that the groups are in some way related [130]. This incorporates within-group subject and between-subject variability into its $f$ statistic calculation, but otherwise behaves very similarly, being linked to a corresponding $p$ value which is compared to a critical value to determine if the null hypothesis can be rejected. In addition to giving a within-subject significance test, it also gives a between-subject significance test, and also a set of trend comparisons to tell if the within-subject factor changes significantly per level when taking into account between-subject variance. The order of this trend changes with each comparison, going from one to the number of within-subject factors less one. Therefore, each repeated measures ANOVA will give three results: a set of pairwise comparisons, seeing if each within-subject comparison is significantly different or not, a singular result to see if between-subject variations are statistically significantly different, and a set of trend comparisons, distinguishing if there are any trends of any order between the within- and between-subject factors.

Because of the use of repeated multiple comparisons, one runs into the multiple comparison problem in statistics [131]. As the number of comparisons increases, it becomes more likely that the groups being compared will appear to differ by chance due to compounding chance of error. Take, for example, the use of the common critical value threshold of 0.05 mentioned prior, which signifies a 5% chance of incorrectly rejecting the null-hypothesis. If 100 independent comparisons are conducted, the expected value for these errors would be 5, falsely indicating that five attributes had statistically significantly differences. Therefore, a correction factor, or Bonferroni correction, must be applied, in order to change the comparison criteria for the calculated $p$ value to account for this. Typically, this is done by either multiplying the $p$ value by the number of comparisons performed, or dividing the critical threshold $\alpha$ value by the number of comparisons to minimize the possibility of false null hypothesis rejection.

In addition to determining if each activation level was statistically different in its stroke production, it was also desired to construct confidence intervals of their maximum contraction
values. These give a range of values centred around the mean that incorporate the variance and sample size of the experimental results, giving a range of expected values to a certain level of certainty [132]. These are calculated as follows:

\[ CI = \mu \pm Z \frac{\sigma}{\sqrt{n}} \]

where \( \mu \) is the mean, \( \sigma \) is the standard deviation, and \( n \) is the number of samples of the data. \( Z \) is the \( z \)-score associated with the desired certainty level. For a certainty of 95%, the \( Z \) value is set to 1.96, as in the population size equations explored in Chapter 3.

With regards to these methods as they related to the experiment, a repeated measures one-way ANOVA was used to determine the statistical significance of the difference between activation levels when the actuator was contracting, or heating, and relaxing, or cooling. The within-subject factor was the activation level being used for each trial, as that was the condition being changed for each subject, and the between-subjects factor was the actuator being used, as multiple were being tested. As well, confidence intervals were constructed for the stroke values in both scenarios. Both of these were generated using the statistical processing software SPSS, using experimental results.

4.1.4 Results

After performing all of the trials for each activation level of each actuator, the data collected were processed and made usable. The temperature of each active TCA at each time point was averaged to find the average active temperature of the actuator, and the initial distance to the floor recorded by the IR sensor was subtracted from each of the subsequent recordings to get the stroke of the actuator at that time point. Examples of the results of individual trials can be seen in Figure 4.7, and the total results of every trial for each of the activation levels can be seen in Figure 4.8. Once this was done, each trial was split into two phases: the actuator in heating, and the actuator in cooling. This was done by finding the time point at which the average temperature of the active TCAs began to decrease while the total power consumption of the actuator was zero. The purpose of splitting each trial into these two phases was to capture the relationship between temperature and stroke in contraction, or heating, and in relaxation, or cooling. Examples of these results can
be seen in Figure 4.9 and Figure 4.10 for contraction and relaxation respectively, as well as the total results for each of the activation levels in Figure 4.11 and Figure 4.12 for contraction and relaxation respectively. Once this separation was complete, the maximum stroke was found for each trial in contraction and the minimum stroke was found for each trial in relaxation, giving the range of motion for the actuator in each phase. These values were then averaged, and the standard deviation found, in order to gauge whether further trials were required or if this number could be considered statistically relevant.
4.1 Stroke Characterization

(a) One pair of TCAs active.

(b) Two pairs of TCAs active.

(c) Three pairs of TCAs active.

(d) Four pairs of TCAs active.

(e) Five pairs of TCAs active.

(f) Six pairs of TCAs active.

Figure 4.7: Randomly selected examples of individual trials for the data collected from the stroke characterization experiment. Different time scales have been used for clarity purposes.
4.1 Stroke Characterization

(a) One pair of TCAs active.  
(b) Two pairs of TCAs active.  
(c) Three pairs of TCAs active.  
(d) Four pairs of TCAs active.  
(e) Five pairs of TCAs active.  
(f) Six pairs of TCAs active.

Figure 4.8: Experimental results of the stroke characterization experiment. All trials are overlaid on top of one another. Different time scales have been used for clarity purposes.
4.1 Stroke Characterization

(a) One pair of TCAs active.
(b) Two pairs of TCAs active.
(c) Three pairs of TCAs active.
(d) Four pairs of TCAs active.
(e) Five pairs of TCAs active.
(f) Six pairs of TCAs active.

Figure 4.9: Randomly selected examples of individual trials for the data collected from the contraction phase of the stroke characterization experiment. Different time scales have been used for clarity purposes.
4.1 Stroke Characterization

Figure 4.10: Randomly selected examples of individual trials for the data collected from the relaxation phase of the stroke characterization experiment. Different time scales have been used for clarity purposes.
4.1 Stroke Characterization

(a) One pair of TCAs active.
(b) Two pairs of TCAs active.
(c) Three pairs of TCAs active.
(d) Four pairs of TCAs active.
(e) Five pairs of TCAs active.
(f) Six pairs of TCAs active.

Figure 4.11: Experimental results of the contraction phase of the stroke characterization experiment. All trials are overlaid on top of one another. Different time scales have been used for clarity purposes.
4.1 Stroke Characterization

(a) One pair of TCAs active.

(b) Two pairs of TCAs active.

(c) Three pairs of TCAs active.

(d) Four pairs of TCAs active.

(e) Five pairs of TCAs active.

(f) Six pairs of TCAs active.

Figure 4.12: Experimental results of the relaxation phase of the stroke characterization experiment. All trials are overlaid on top of one another. Different time scales have been used for clarity purposes.
4.1.5 Stroke Production Analysis

The first piece of analysis required was to decide if further trials were required, and for that purpose the minimum discernible effect for the experiment after 50 trials for each activation level was calculated. As noted in Sections 3.7 and 3.8, while the initial required sample size had been calculated based on the results from preliminary experimentation, examples in the literature were much smaller than those calculated. It was decided that a large number of trials (50) would be used, and repeated if the results did not perform in accordance with the desired certainty for a specified minimum effect size. To do this, Equation 3.2 was used, with $\Delta$ representing the minimum discernible effect size, $Z_\alpha$ being set to 1.96 to represent a 95% certainty of a lack of a Type I error, $Z_\beta$ being set to 0.84 to represent a power of 80%, and $\sigma$ being set as the standard deviation for the trials performed at each activation level, expressed as a percentage of the mean force produced at that level. Table 4.1 shows the calculated $\Delta$s.

As can be seen from these results, the largest, least desirable minimum discernible effect size was calculated to be 0.038, or 3.8% of the mean force produced at any activation level. This is far below the 5% threshold set in Sections 3.7 and 3.8, meaning that 50 trials were indeed enough at each activation level to achieve the desired level of statistical certainty. Because of this, no further trials were required in order to improve the validity of the testing results for the stroke characterization experiment. Confirming this, further analysis could be conducted knowing that the statistical validity of the sample size was assured.

Table 4.1: Minimum discernible effect calculations by activation level, using produced stroke results from the stroke characterization experiment.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean Maximum Contraction (mm)</th>
<th>Std. Deviation (mm)</th>
<th>Calculated $\Delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.74</td>
<td>0.15</td>
<td>0.031</td>
</tr>
<tr>
<td>2</td>
<td>4.34</td>
<td>0.29</td>
<td>0.038</td>
</tr>
<tr>
<td>3</td>
<td>6.33</td>
<td>0.35</td>
<td>0.031</td>
</tr>
<tr>
<td>4</td>
<td>8.55</td>
<td>0.13</td>
<td>0.008</td>
</tr>
<tr>
<td>5</td>
<td>10.82</td>
<td>0.11</td>
<td>0.005</td>
</tr>
<tr>
<td>6</td>
<td>12.23</td>
<td>0.12</td>
<td>0.006</td>
</tr>
</tbody>
</table>
From the results of the repeated measures one-way ANOVA analysis performed for within-subject factors, as seen in Table 4.2, it is clear that the null hypothesis can be rejected for each of the activation levels in contraction, with \( p \) values of \(< 0.01\) for each activation level. With each \( p \) value below the threshold of 0.05, each of the stroke production values for each activation level were statistically significantly different from one another. This indicates that the addition of activated actuating pairs leads to statistically significant differences in stroke production in contraction for the actuator when moving a static load. This was not the case for the between-subject comparisons, which found there was no statistical difference between fabricated actuators in pair-wise comparisons \((p = 0.68, 0.06, 0.38, 0.79, 0.63\) for each trend order comparison from 1–5, respectively, comparing the difference between activation levels across different actuators, and \( p = 0.36 \) for strictly between-subject comparisons\). This means that the actuator operates similarly between different fabricated instances. After calculating the confidence intervals of the contraction of the actuator, as seen in Table 4.3, the mean values had a linear regression applied to determine to what degree they could be said to be linearly scaling. This was done using Microsoft Excel, the results of which can be seen in Figure 4.13.

![Figure 4.13: Confidence intervals of the percent contraction compared to activation level, with an overlaid linear regression to show linearity of the results. The formula for the linear regression and its \( R^2 \) value are presented as well.](image-url)
Table 4.2: Repeated measures one-way ANOVA analysis results for stroke produced by the actuator in contraction. All values had a Bonferroni correction applied, with a threshold of 0.05 signifying significance.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference (mm)</th>
<th>Std. Error (mm)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-1.60</td>
<td>0.047</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-3.60</td>
<td>0.050</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-5.81</td>
<td>0.030</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-8.08</td>
<td>0.027</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-9.49</td>
<td>0.027</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1.60</td>
<td>0.047</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-1.99</td>
<td>0.068</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-4.21</td>
<td>0.042</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-6.48</td>
<td>0.041</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-7.89</td>
<td>0.043</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3.59</td>
<td>0.050</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.99</td>
<td>0.068</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-2.22</td>
<td>0.056</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-4.49</td>
<td>0.053</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-5.90</td>
<td>0.051</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5.81</td>
<td>0.030</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4.21</td>
<td>0.042</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>2.22</td>
<td>0.056</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-2.27</td>
<td>0.021</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-3.68</td>
<td>0.028</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>8.08</td>
<td>0.027</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>6.48</td>
<td>0.041</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4.49</td>
<td>0.053</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2.27</td>
<td>0.021</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-1.41</td>
<td>0.023</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>9.49</td>
<td>0.027</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>7.89</td>
<td>0.043</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>5.90</td>
<td>0.051</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.68</td>
<td>0.028</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.41</td>
<td>0.023</td>
<td>&lt; 0.01</td>
</tr>
</tbody>
</table>
Table 4.3: Stroke production values for each activation level of the actuator in contraction. All values in mm.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>2.74</td>
<td>0.15</td>
<td>0.022</td>
<td>2.70</td>
</tr>
<tr>
<td>2</td>
<td>4.34</td>
<td>0.29</td>
<td>0.041</td>
<td>4.26</td>
</tr>
<tr>
<td>3</td>
<td>6.33</td>
<td>0.35</td>
<td>0.050</td>
<td>6.23</td>
</tr>
<tr>
<td>4</td>
<td>8.55</td>
<td>0.13</td>
<td>0.018</td>
<td>8.51</td>
</tr>
<tr>
<td>5</td>
<td>10.82</td>
<td>0.11</td>
<td>0.015</td>
<td>10.79</td>
</tr>
<tr>
<td>6</td>
<td>12.23</td>
<td>0.12</td>
<td>0.017</td>
<td>12.20</td>
</tr>
</tbody>
</table>

From the $R^2$ value, it is apparent that the contraction of the actuator scales very linearly with the number of active pairs of TCAs when acting on a static load. This slope of the line, or increase in stroke per additional active pair, is approximately 1.97 mm, or 1.97%. This linearity is in accordance with the literature, which states that heating TCAs is similar to changing the stiffness of a spring [52] in terms of its mode of action. Because the actuator puts equivalent TCAs in parallel, equivalent changes in stiffness in pairs of TCAs would result in linear changes in total actuator stiffness, which would proportionally affect its stroke production. The free body diagram in Figure 4.14, a simplified representation of the woven bi-pennate actuator, and the following process explain this concept further.

In a state of equilibrium, the spring force exerted by the actuator and the gravitational force exerted on the system by the hanging mass would be equivalent. From Hooke’s law [133], the spring force acting to balance the gravitational force exerted by the mass would be the following, where $K$ represents a spring constant for a pair of TCAs and $d$ the displacement of the mass, or stroke, of the system:

$$F_s = K_{total}d$$

$$F_s = (K_{y,1} + K_{y,2} + K_{y,3} + K_{y,4} + K_{y,5} + K_{y,6})d$$
4.1 Stroke Characterization

Figure 4.14: A simplified free body representation of the woven bi-pennate actuator. The force of gravity exerted by the mass hanging from the actuator, denoted by $F_g$, will be equal to the vertical component of the spring force $F_{s,y}$ exerted by the TCAs as the two are in equilibrium.

This could then be rearranged to solve for the stroke of the actuator, as follows:

$$-F_g = (K_{y,1} + K_{y,2} + K_{y,3} + K_{y,4} + K_{y,5} + K_{y,6}) d$$

$$d = -\frac{F_g}{(K_{y,1} + K_{y,2} + K_{y,3} + K_{y,4} + K_{y,5} + K_{y,6})}.$$

Because all of the TCAs are assumed to be identical, each pair will have the same stiffness $K$, making the equivalency instead the following:

$$d = -\frac{F_g}{(K + K + K + K + K)}.$$
4.1 Stroke Characterization

\[ d = \frac{-F_g}{6K}. \]

By heating just one pair of TCAs, their stiffness is changed to some unknown value \( xK \), an unknown multiple of \( K \). This changes the equivalency to the following:

\[ d = \frac{-F_g}{(xK + K + K + K + K + K)}, \]

\[ d = \frac{-F_g}{(5 + x)K}. \]

Doing this again to two TCAs, the two heated TCAs should have the same altered stiffness as they have been constructed the same and are being heated to the same temperature, making the equivalency the following:

\[ d = \frac{-F_g}{(xK + xK + K + K + K + K)}, \]

\[ d = \frac{-F_g}{(4 + 2x)K}. \]

This can be repeated until eventually all six TCAs are being heated, giving the final equivalency of the following:

\[ d = \frac{-F_g}{(0 + 6x)K}. \]

As can be seen, with each iteration the total stiffness of the system is equal to \((6 - i) + (i)x)K\), which is a linear function based on the number of active pairs of TCAs \( i \). Thus, the linearity of the results is in accordance with the literature and scientific principles, as the actuator should behave as an array of parallel springs whose stiffness changes as each pair of TCAs is heated.

This linearity is desirable as it allows for users and designers to predict how devices incorporating these actuators will behave based on limited data and how they will scale with the addition of more pairs of active TCAs. There is a maximum to this scaling effect as the literature has shown that TCAs have a defined contracting range within a temperature span [51], but this knowledge does help in designing within that operational range. While this is beneficial for predicting the behaviour of the actuator in contraction, analysis on the displacement produced in the relaxation phase revealed that these same predictive benefits could not be seen when the actuator was in relaxation.
4.1 Stroke Characterization

From the results of the repeated measures one-way ANOVA analysis performed on the results from the actuator in relaxation, as seen in Table 4.4, it is observed that for the most part the differences between the within-subject factors are once again statistically significantly different in the amount of slack that was produced with \( p \) values of between 0.00 and 0.02 for most pairwise comparisons; however, there was no statistically significant difference in the amount of slack produced in relaxation between four active pairs and six with a \( p \) value of 1.00 for the comparison between those two pairs. There is no easily discernible reason for why this is, but there are some possibilities that could be explored in future investigations. It could be that once the TCAs relax past a certain point, the amount of slack they produce is confined to a range. This could mean that it does not matter how many actuators are relaxing at once past a certain point, as there is only so far that each can relax. It could also be that relaxing has a higher variability than the contraction of the TCAs, making the total slack produced not as predictable with the addition of more TCAs. In either case, the relaxation of both of the activation levels can still be characterized and their minimum relaxation values can still guide future work by providing expected values for the actuator at these activation levels, which can be seen in Table 4.5. However, it does serve to pique interest into the possible reasons the relaxation became statistically insignificantly different past a certain activation level threshold. The lack of a significant difference between activation levels from the ANOVA analysis indicates that the actuator cannot behave linearly in relaxation. As mentioned prior this requires further investigation to determine its cause, but this less-predictable amount of slack production does necessitate the use of a workaround in order to overcome it in the use of wearable therapeutic devices. Comparing the between-subject factors, the different fabricated actuators were not statistically significantly different (\( p = 0.86, 0.48, 0.86, 0.62, 0.54 \) for each trend order comparison from 1–5, respectively, comparing between different activation levels across actuators, and \( p = 0.38 \) for strictly between-subject comparisons), matching the results from the heating phase which found that the designed actuator operates similarly between different fabricated instances.
Table 4.4: Repeated measures one-way ANOVA analysis results for stroke produced by the actuator in relaxation. All values had a Bonferroni correction applied, with a threshold of 0.05 signifying significance.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference (mm)</th>
<th>Std. Error (mm)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.68</td>
<td>0.053</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.07</td>
<td>0.064</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1.40</td>
<td>0.061</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2.19</td>
<td>0.084</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1.44</td>
<td>0.088</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-0.68</td>
<td>0.053</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.39</td>
<td>0.075</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.72</td>
<td>0.065</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.51</td>
<td>0.098</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.76</td>
<td>0.084</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-1.07</td>
<td>0.064</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.39</td>
<td>0.075</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.33</td>
<td>0.073</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.11</td>
<td>0.10</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.37</td>
<td>0.11</td>
<td>0.02</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>-1.40</td>
<td>0.061</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.72</td>
<td>0.065</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.33</td>
<td>0.076</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.79</td>
<td>0.094</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.038</td>
<td>0.087</td>
<td>1.00</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>-2.19</td>
<td>0.084</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-1.51</td>
<td>0.098</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-1.11</td>
<td>0.10</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-0.79</td>
<td>0.094</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-0.75</td>
<td>0.087</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>-1.44</td>
<td>0.088</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.76</td>
<td>0.084</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.37</td>
<td>0.11</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-0.038</td>
<td>0.087</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.75</td>
<td>0.12</td>
<td>&lt; 0.01</td>
</tr>
</tbody>
</table>
Table 4.5: Stroke production values for each activation level of the actuator in relaxation. All values in mm.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>-1.23</td>
<td>0.19</td>
<td>0.027</td>
<td>-1.29</td>
</tr>
<tr>
<td>2</td>
<td>-1.91</td>
<td>0.30</td>
<td>0.043</td>
<td>-2.00</td>
</tr>
<tr>
<td>3</td>
<td>-2.31</td>
<td>0.42</td>
<td>0.060</td>
<td>-2.43</td>
</tr>
<tr>
<td>4</td>
<td>-2.63</td>
<td>0.39</td>
<td>0.055</td>
<td>-2.75</td>
</tr>
<tr>
<td>5</td>
<td>-3.42</td>
<td>0.57</td>
<td>0.080</td>
<td>-3.58</td>
</tr>
<tr>
<td>6</td>
<td>-2.67</td>
<td>0.58</td>
<td>0.082</td>
<td>-2.84</td>
</tr>
</tbody>
</table>

Dividing the stroke in millimetres by the original length of 100 mm, the stroke in percent contraction and relaxation can be seen, found in Table 4.6 and Table 4.7 respectively. It should be noted that this stroke occurred in a vertical direction, while the TCAs were arrayed about the central rib with a pennation angle of 45°, making their contraction longer than that of the stroke by a factor of $1/\cos(45°)$, which can also be seen in Tables 4.6 and Table 4.7. Figure 4.14 shows the relationship between the vertical stroke and the angular contraction of the TCAs. In terms of gross stroke production, as would be experienced by the dead load, this puts the actuating module in line with other artificial muscle actuators [43, 51, 64, 85], but able to exceed those same actuators in the diagonal direction. This is in line with the literature, which observed that when formed into a composite fabric matrix, artificial muscles were able to generate a greater stroke than otherwise capable [13]. It should be noted that not-insignificant slack was also produced in relaxation, which tended to increase as the number of activated pairs increased. This would need to be overcome in any device implementing these actuating modules, as this would lead to drift in the range of motion of these devices, which is particularly undesirable for therapeutic devices trying to move the user through a constrained range of motion repeatedly. A system for tightening and re-positioning the TCAs within the module after use could work, but thought would have to be put in to ensure that they did not get over-tightened or snap.
Table 4.6: Vertical displacement of the end effector of the actuator acting in contraction expressed as a percent stroke of the initial length of the actuator, as well as the percent contraction the TCAs along their initial angle within the actuator. All values presented as decimal percentage values.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Vertical Mean</th>
<th>95% Conf. Interval</th>
<th>Angled Mean</th>
<th>95% Conf. Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>L. Bound</td>
<td>U. Bound</td>
<td>L. Bound</td>
</tr>
<tr>
<td>1</td>
<td>0.027</td>
<td>0.027</td>
<td>0.028</td>
<td>0.039</td>
</tr>
<tr>
<td>2</td>
<td>0.043</td>
<td>0.043</td>
<td>0.044</td>
<td>0.061</td>
</tr>
<tr>
<td>3</td>
<td>0.063</td>
<td>0.062</td>
<td>0.064</td>
<td>0.090</td>
</tr>
<tr>
<td>4</td>
<td>0.086</td>
<td>0.085</td>
<td>0.086</td>
<td>0.12</td>
</tr>
<tr>
<td>5</td>
<td>0.11</td>
<td>0.11</td>
<td>0.11</td>
<td>0.15</td>
</tr>
<tr>
<td>6</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
<td>0.17</td>
</tr>
</tbody>
</table>

Table 4.7: Vertical displacement of the end effector of the actuator acting in relaxation expressed as a percent stroke of the initial length of the actuator, as well as the percent contraction the TCAs along their initial angle within the actuator. All values presented as decimal percentage values.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Vertical Mean</th>
<th>95% Conf. Interval</th>
<th>Angled Mean</th>
<th>95% Conf. Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>L. Bound</td>
<td>U. Bound</td>
<td>L. Bound</td>
</tr>
<tr>
<td>1</td>
<td>-0.012</td>
<td>-0.013</td>
<td>-0.012</td>
<td>-0.017</td>
</tr>
<tr>
<td>2</td>
<td>-0.019</td>
<td>-0.020</td>
<td>-0.018</td>
<td>-0.027</td>
</tr>
<tr>
<td>3</td>
<td>-0.023</td>
<td>-0.024</td>
<td>-0.022</td>
<td>-0.033</td>
</tr>
<tr>
<td>4</td>
<td>-0.026</td>
<td>-0.027</td>
<td>-0.025</td>
<td>-0.037</td>
</tr>
<tr>
<td>5</td>
<td>-0.034</td>
<td>-0.036</td>
<td>-0.033</td>
<td>-0.048</td>
</tr>
<tr>
<td>6</td>
<td>-0.027</td>
<td>-0.028</td>
<td>-0.025</td>
<td>-0.038</td>
</tr>
</tbody>
</table>

4.1.6 Stroke Production Regression

Having found that the results were statistically significant, and having found the bounds of the stroke production for each activation level, the contraction and relaxation results were used to perform a univariate regression between temperature and stroke production for each phase. This was done by separating the temperature and stroke values from the other data for each trial, and constructing a set of all data points for each activation level for each phase. With the temperature and stroke data for each of the 50 trials for each activation level separated, a regression was performed with temperature being the independent variable and stroke being the dependent
variable using the `polyfit` function in the scripting language MATLAB. The degree of the regressed polynomial was tested iteratively, starting from a degree of one and continuing until a degree of ten was tested.

Finding the optimal degree polynomial was done by applying the `polyfit` function across the entirety of the data set in order to fit that degree of regressed function to the data. Then, the temperature values were input back into the regressed function, and the output compared to the actual stroke values associated with each of the input data points. The differences between the two were taken, squared, and summed across the dataset, giving a sum of the squared errors for that degree of regressed function on the data set for that activation level. The sum of squared errors was then divided by the number of data points to give the average squared error per input data point for each sum of squared errors. The square root of that was then taken to get the root mean square error (RMSE) for each degree polynomial regressed for each activation level. Because it was desirable to determine the degree polynomial that best represented the actuator’s stroke production capability, and not just an individual activated pair of TCAs, the RMSEs for each degree polynomial were then averaged across each activation level, giving ten averaged RMSE values denoting the performance of each degree polynomial in representing the stroke production of the actuator. These average RMSE values are presented in Figure 4.15, and the code used to execute this regression and testing process can be found in Appendix B.1.1.

The best performing degree of polynomial was then selected. Contrary to the instinct of choosing the degree that minimized the average RMSE, when attempting to fit a regression function, it is best to select the function that minimizes the error metric while the error metric is not behaving asymptotically, as that asymptotic behaviour is likely only decreasing in error by fitting to the noise present in the system, rather than extracting information from the dataset [134]. Looking at Figure 4.15, at first glance there is a definite curve for the contraction curve which follows this rule; however, looking at the actual values, the average RMSE only decreases from 0.352 mm to 0.349 mm after testing each polynomial, or 0.003 mm. Looking at the average RMSE values on a different scale seen in Figure 4.16, one can see how flat the curve actually is. This indicates that a linear scale, or degree of one, is acceptable for modelling the data, as higher degree polynomials are likely just capturing the noise in the system. The same happens with the
relaxation curve, which only decreases from an average RMSE of 0.415 mm to 0.414 mm, a change of 0.001 mm, after testing each degree polynomial. Therefore, a degree of one can also be used to model the relationship between stroke and temperature in relaxation, as it differs even less than the contraction regression. Both of the average RMSE values are low, indicating a high degree of fit across each dataset for each activation level. These give the functions seen in Tables 4.8 and 4.9 for the stroke production of the actuator in contraction and relaxation, respectively. \( s_{\text{prod}} \) represents the produced stroke, \( T \) represents the temperature in degrees Celsius, the subscript \( H \) denotes contraction, and the subscript \( C \) denotes relaxation. The general form of the polynomials are shown prior to the tables, whose coefficients were found in the regression.

![Figure 4.15](image.png)

(a) Average RMSE in contraction.  
(b) Average RMSE in relaxation.

Figure 4.15: Average RMSE per activation level for stroke regression in each phase, by degree polynomial.
4.1 Stroke Characterization

Figure 4.16: Average RMSE per activation level for stroke regression in each phase, by degree polynomial, on a different scale.

\[ s_{\text{prod},H} = A_H T + B_H \]  

(4.1)

Table 4.8: Regressed equation coefficient values for relationship between temperature and stroke for each activation level of the actuator in contraction.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>(A_H)</th>
<th>(B_H)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.039</td>
<td>-0.97</td>
</tr>
<tr>
<td>2</td>
<td>0.067</td>
<td>-1.89</td>
</tr>
<tr>
<td>3</td>
<td>0.104</td>
<td>-2.95</td>
</tr>
<tr>
<td>4</td>
<td>0.143</td>
<td>-3.96</td>
</tr>
<tr>
<td>5</td>
<td>0.192</td>
<td>-5.85</td>
</tr>
<tr>
<td>6</td>
<td>0.213</td>
<td>-6.25</td>
</tr>
</tbody>
</table>

\[ s_{\text{prod},C} = A_C T + B_C \]  

(4.2)

Table 4.9: Regressed equation coefficient values for relationship between temperature and stroke for each activation level of the actuator in relaxation.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>(A_C)</th>
<th>(B_C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.06</td>
<td>-2.46</td>
</tr>
<tr>
<td>2</td>
<td>0.10</td>
<td>-4.43</td>
</tr>
<tr>
<td>3</td>
<td>0.14</td>
<td>-6.24</td>
</tr>
<tr>
<td>4</td>
<td>0.19</td>
<td>-7.93</td>
</tr>
<tr>
<td>5</td>
<td>0.24</td>
<td>-10.26</td>
</tr>
<tr>
<td>6</td>
<td>0.26</td>
<td>-9.91</td>
</tr>
</tbody>
</table>
The fit of these regressions can be seen in Figures 4.17 and 4.18, for the actuator in contraction and relaxation respectively.

(a) Regression for one pair of active TCAs.

(b) Regression for two pairs of active TCAs.

(c) Regression for three pairs of active TCAs.

(d) Regression for four pairs of active TCAs.

(e) Regression for five pairs of active TCAs.

(f) Regression for six pairs of active TCAs.

Figure 4.17: Regression of the relationship between temperature and stroke for each activation level of the actuator in contraction.
4.1 Stroke Characterization

Figure 4.18: Regression of the relationship between temperature and stroke for each activation level of the actuator in relaxation.
4.1 Stroke Characterization

4.1.7 Efficiency Analysis

As well as characterizing the relationship between stroke production and temperature for the actuator at each activation level, one of the objectives in this experiment was to investigate the efficiency of the actuator converting electrical power into mechanical work. This can be calculated by finding the ratio between the electrical power consumed and the mechanical work output; however, due to the power draw of the actuator changing over time due to internal resistances of the TCAs changing as they heated [57], as well as fluctuation due to temperature control (seen in Figures 4.7 and 4.9), a more involved approach had to be taken to determine the total electrical energy consumed by the system during the heating phase before being contrasted to the total mechanical energy output by the system. The electrical energy consumed for each trial was calculated by multiplying the power consumption at each time interval by the duration of the time interval, which was the sampling rate of 0.11 seconds. This was done over the course of the entirety of the heating phase for each trial, giving the total energy consumption of the heating in Joules. The mechanical energy output by the system in the same time period was found by multiplying the gravitational force exerted by the 1.23 kg total mass on the system (a combination of the weights, sensor, and its holster) by the maximum stroke of that trial in heating. This gave the peak mechanical work output of the system, which, when divided by the total electrical energy required to drive it, gave the efficiency of that particular trial. This was repeated for every trial for every activation level.

After obtaining the efficiency for each trial, SPSS was used to perform a repeated measures one-way ANOVA on the results, which can be seen in Table 4.10. As before, the within-subject factors were chosen to be the different activation levels, and the within subject factors were chosen to be the different actuators made. A Bonferroni correction was applied to prevent errors from repeated independent comparisons. As can be seen from the results in Table 4.10, it is clear that the null hypothesis can be rejected for each number of activated pairs \( p < 0.01 \) compared to each other activated pairs, meaning that there is a statistically significant difference in efficiency in the actuator based on the number of pairs active in contraction. The between-subjects factor testing found that the different actuators behaved similarly at each activation level \( p = 0.96, 0.53, 0.52, 0.90, 0.66 \) for each ascending trend comparison order from 1–5, comparing
differences between activation levels across actuators, and $p = 0.41$ for strictly between-subject comparisons), indicating consistency between different fabricated actuators in efficiency in addition to stroke, as found previously.

Table 4.10: Repeated measures one-way ANOVA analysis results for the efficiency of the actuator in contraction. All mean difference and std. error values presented as decimal percent values. All significance values had a Bonferroni correction applied, with a threshold of 0.05 signifying significance

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference</th>
<th>Std. Error</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.004</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.005</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.006</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.007</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.007</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-0.004</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.001</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.002</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.003</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.003</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-0.005</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.001</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.001</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.002</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.003</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>-0.006</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.002</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.001</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.001</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.002</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>-0.007</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.003</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.002</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-0.001</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.001</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>-0.007</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.003</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.003</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-0.002</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-0.001</td>
<td>0.000</td>
<td>&lt; 0.01</td>
</tr>
</tbody>
</table>
As well, confidence intervals were constructed for each of the efficiencies at each of the levels, as seen in Table 4.11. These show the possible range of efficiency values for the actuator in contraction, ranging from 1.10% to 0.36% for one pair and six pairs of TCAs active in contraction, respectively.

### Table 4.11: Efficiency values for each activation level of the actuator in contraction. All values presented as decimal percent values.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>0.0110</td>
<td>0.000923</td>
<td>0.000</td>
<td>0.0101</td>
</tr>
<tr>
<td>2</td>
<td>0.0071</td>
<td>0.000718</td>
<td>0.000</td>
<td>0.0064</td>
</tr>
<tr>
<td>3</td>
<td>0.0062</td>
<td>0.000541</td>
<td>0.000</td>
<td>0.0057</td>
</tr>
<tr>
<td>4</td>
<td>0.0054</td>
<td>0.000232</td>
<td>0.000</td>
<td>0.0052</td>
</tr>
<tr>
<td>5</td>
<td>0.0044</td>
<td>0.000230</td>
<td>0.000</td>
<td>0.0042</td>
</tr>
<tr>
<td>6</td>
<td>0.0036</td>
<td>0.000169</td>
<td>0.000</td>
<td>0.0034</td>
</tr>
</tbody>
</table>

### 4.2 Force Characterization

This section discusses the experimental methods, results, analysis, and conclusions that could be drawn from the force characterization experiment. In particular, it explores the statistical analysis of the maximum force production capabilities of the actuator, the creation of a relationship between temperature and force, and a comparison between the force and stroke production capabilities of the actuator.

#### 4.2.1 Purpose

The purpose of this experiment was to characterize the contractive capabilities of the woven bipennate actuator in an isometric configuration, attached horizontally to a load cell and contracting to apply a tensile force to the cell. This was done to find the maximum force production capability at each activation level after having reached a chosen maximum temperature threshold, which was then analyzed for linearity to see how predictable the addition or removal of pairs of TCAs would be in regards to force production. This helps determine if the designed actuator is suitable for use in a wearable rehabilitative device, as it helps determine at what point the number of TCA pairs...
would be sufficient to provide the required torque for rehabilitative exercises, found from values in the literature and presented in Chapter 2. By comparing this to other similar actuators presented in the literature, one can determine the viability of the actuator when applied in commercial and medical designs, as well as how it compares to current robotic rehabilitation solutions. Transient data were used to determine a numerical relationship between the temperature and force produced per number of active TCA pairs using a polynomial regression, which would then be further generalized to create a relationship between temperature, number of active pairs of TCAs, and force produced. In Chapter 5 this would provide the basis for designing control systems for force control. In total, all of these aspects of the experiment served to assess the capability of the actuator and quantify its performance, determining its viable operation range and enabling the prediction of force production capacity based on the number of active TCA pairs and temperature achieved.

4.2.2 Experimental Methods

The design of the experimental setup was to suspend the actuator horizontally, strung between a rigid structure and a load cell so that, on contraction, its isometric force would be measured by the cell. To create the rigid structures required, the wooden structures presented in Appendix A.3 were constructed. Using the measurements in those drawings, the rigid attach structures were created using 1” × 3” pine planks, cut to size, and attached together using 2” wood screws. They were then clamped to a levelled lab bench, ensuring a steady fixture for attaching both the actuator and load cell. A Bolson Tech 5 kg load cell was attached to the smaller of the two structures, which was clamped between the arms of the other structure. These rigid wooden structures can be seen in Figure 4.19.

In addition to these wooden components, the ring presented in Appendix A.2.7 and the lateral holsters from Appendix A.2.8 were 3D printed on a Markforged Onyx Pro 3D printer from Onyx Nylon filament, with the ring being bolted to the load cell using M5 bolts and nuts with a nylon washer used as a spacer between the cell and ring. These components can be seen in Figure 4.20. The lateral holsters were clamped to the elevated horizontal planks of the rigid structure and
adjusted on top of them until their inside edges were 141 mm apart, measured with a digital Vernier caliper, and such that their leading pins (facing the load cell and hook) were 141 mm from the leading pin of the central tendon when attached to the load cell ring. These distances were marked on the wooden structure to speed alignment in future trials. The assembled side tendons used in the stroke characterization experiment were then slid into these lateral holsters, and the construction of the actuator began.

The same TCAs used in the stroke characterization experiment in the same bundles of 12 were used to fabricate the actuators. A bundle of 12 was slid over the pins of the side tendons and affixed using the pin caps from the previous experiment. The central tendon was attached by the hook to the load cell ring and held horizontal while the free ends of the TCAs were attached to the pins of the central tendon with the pin caps from before. These pin caps were again wired to the PCB using the quick release cable modules, containing the same thermocouples in their guides which were also wired to the PCB by the quick release cable modules from before. The load cell structure and lateral holsters were then adjusted until there was a slight amount of tension in the TCAs to remove any slack that could interfere with the weaving process, while maintaining the pennation angle of 45° that the initial setup gave. In a similar manner to the previous
4.2 Force Characterization

Figure 4.20: Components fabricated prior to actuator construction.

experiment, the thermocouples attached to the pin caps were tied to their designated TCAs using nylon sewing thread. After doing so, thermal paste (Halzniye 510) was applied liberally to electrically insulate and improve thermal conductivity to each thermocouple. Pictures of the grease-covered thermocouples before and after having been woven over can be seen in Figure 4.21.
4.2 Force Characterization

After the thermocouples were attached, the actuator was woven using the same 100% nylon yarn as before (Part No. 1005019 from anniescatalog.com). The actuator was woven using the same standard over-under manual weaving technique, using the TCAs as the warp and compressing the weft with every pass to make the weave as tight as possible, the results of which can be seen in Figure 4.22. One side was done at a time, with the weaving continuing until the thermocouples had been covered by a minimum of three loops on each side. Once the actuator had been fully woven, the wooden load cell supportive structure was adjusted to ensure that a 45° pennation angle for the TCAs was still maintained, after which the actuator was considered fully fabricated. Pictures of the actuator in various stages of its construction can be seen in Figure 4.23.

In this experiment, the number of pairs of TCAs being heated, or levels of activation of the actuator, were iteratively tested to determine its isometric force production capacity. This was done with similar methods to those outlined in the stroke characterization methods section. A pair of TCAs were chosen randomly and heated at 30 W (10 V at 3 A) supplied by a Rigol DP811 programmable DC power supply until they both reached a temperature of 80° C. If one reached this temperature before the other, its temperature was maintained at 80° by alternating it on and off using its corresponding transistor until the other TCA reached 80°, at which point both were allowed to cool until they both reached 30° Celsius. This process was controlled by the same microcontrollers (ESP32 and Arduino UNO) used in the previous experiment, mounted on the same PCB. During the trial, the temperature, power, force in grams equivalent, and time were
logged by the EPS32 in a free running loop, which was found to give a sampling rate of 0.11 seconds after preliminary testing.

After this was completed, this was repeated with another random activated pair, repeating for a total of five times. After five trials had been run, the number of activated pairs of TCAs was increased, and the process done again with an additional 10 W of power (increase of supply by 1 A) for another five trials, save with four rather than two TCAs being heated. Every five trials the activation level was increased by one, and the power supplied was increased by 10 W (increase of supply by 1 A) until eventually all six pairs of TCAs were heated at 80 W (10 V at 8 A). This gave a total of 30 trials per actuator. After these were complete, the actuator was disassembled and a new actuator was constructed using a new set of 12 TCAs, following the construction steps above. The experimentation process was then repeated on the new actuator, giving an additional 30 trials worth of data. This was repeated until 10 actuators had been tested, giving a total of 50 trials per activation level, the number selected based on the results of the stroke and force production pilot experiments. The microcontroller code used to control this experiment can be found in Appendix C.3.3.
4.2 Force Characterization

(a) View of unwoven actuator attached to wooden structure.
(b) Actuator in process of being woven.
(c) Frontal view of completed woven actuator.
(d) Oblique angle view of woven actuator.
(e) Secondary angled view of woven actuator.
(f) Close view of woven actuator with caps removed for visibility.

Figure 4.23: Various views of the completed setup of the force characterization experiment.
4.2.3 Analytical Methods

As in the stroke characterization experiment, analysis was performed to determine if the results of each activation level were significantly different from one another. For this purpose, a repeated measures one-way ANOVA test was conducted comparing the results of each activation level to one another implementing a Bonferroni correction to correct for repeated independent comparisons, setting the within-subject factor to the activation level and the between-subject factor to the actuator number again. The method for doing this was identical to that detailed in Section 4.1.3, and was implemented again in SPSS. SPSS was also used to generate confidence intervals for the force production capabilities of each activation level using the methods given in Section 4.1.3. The force values are given by the load cell in grams-force-equivalent, and are reported as such, but there was also a need to convert these forces into SI units in order to compare the production of the actuator to examples in the literature. The force in Newtons was calculated by multiplying the gram value by the gravitational acceleration constant 9.81 m/s², and the pressure value was calculated by dividing this force value by the contact area between the ring on the load cell and the hook of the central tendon, which was found to be 6.30 mm² from their Solidworks CAD files. These force values were then divided by the number of active TCAs to give the force production per TCA, and the same $1/\cos(45^\circ)$ correction was applied to these values as in Section 4.1.3 to find the force being exerted by the TCAs in their direction of contraction.

4.2.4 Results

After performing all of the trials for each activation level on each actuator, the results were processed as in Section 4.1.3. This was done by finding the average temperature of the activated pairs at each time point, in order to get the average temperature of the TCAs involved in the actuation of the module, and subtracting the initial force applied by the actuator on the load cell from tension from each of the subsequent force values to get the force produced by the actuator at each of those time points. Examples of the results can be seen in Figure 4.24, as well as the total results for each of the activation levels in Figure 4.25. Once this processing was done, in a similar manner to that of the stroke characterization experiment the results of each trial were split into a heating and cooling phase in order to properly characterize the actuator’s performance.
in contraction (heating) and relaxation (cooling), and find the time required to heat the actuator at each activation level. This again was done by finding the time point at which the average temperature of the active TCAs began to decrease while the total power consumption of the actuating module was zero. Examples of these results can be seen in Figures 4.26 and 4.27 for contraction and relaxation respectively, and the total results for each of the activation levels can be seen in Figures 4.28 and 4.29 for contraction and relaxation respectively. After this separation, the maximum force produced was found for each trial in heating, as well as the minimum force for each trial in cooling, giving the limits of force production for the actuator in each phase. These values were then averaged, and the standard deviation found, in order to gauge whether further trials were required or if the data collected could be considered statistically relevant.
4.2 Force Characterization

(a) One pair of TCAs active.

(b) Two pairs of TCAs active.

(c) Three pairs of TCAs active.

(d) Four pairs of TCAs active.

(e) Five pairs of TCAs active.

(f) Six pairs of TCAs active.

Figure 4.24: Examples of individual trial data collected for the force characterization experiment. Different time scales have been used for clarity purposes.
4.2 Force Characterization

(a) One pairs of TCAs active.
(b) Two pairs of TCAs active.
(c) Three pairs of TCAs active.
(d) Four pairs of TCAs active.
(e) Five pairs of TCAs active.
(f) Six pairs of TCAs active.

Figure 4.25: Experimental results of the force characterization experiment with each trial overlaid on top of one another. Different time scales have been used for clarity purposes.
4.2 Force Characterization

(a) One pair of TCAs active.

(b) Two pairs of TCAs active.

(c) Three pairs of TCAs active.

(d) Four pairs of TCAs active.

(e) Five pairs of TCAs active.

(f) Six pairs of TCAs active.

Figure 4.26: Examples of individual trial data collected for the contraction phase of the force characterization experiment. Different time scales have been used for clarity purposes.
4.2 Force Characterization

Figure 4.27: Examples of individual trial data collected for the relaxation phase of the force characterization experiment. Different time scales have been used for clarity purposes.
4.2 Force Characterization

(a) One pair of TCAs active.
(b) Two pairs of TCAs active.
(c) Three pairs of TCAs active.
(d) Four pairs of TCAs active.
(e) Five pairs of TCAs active.
(f) Six pairs of TCAs active.

Figure 4.28: Experimental results of the contraction phase of the force characterization experiment. All trials are overlaid on top of one another. Different time scales have been used for clarity purposes.
4.2 Force Characterization

(a) One pair of TCAs active.

(b) Two pairs of TCAs active.

(c) Three pairs of TCAs active.

(d) Four pairs of TCAs active.

(e) Five pairs of TCAs active.

(f) Six pairs of TCAs active.

Figure 4.29: Experimental results of the relaxation phase of the force characterization experiment. All trials are overlaid on top of one another. Different time scales have been used for clarity purposes.
4.2.5 Force Production Analysis

After the results for these trials had been collected, the minimum discernible effect for the experiment after 50 trials for each activation level was calculated. As discussed in Sections 3.7 and 3.8, this was required because of the large disparity between the calculated required sample sizes and examples found in the literature. To do this, Equation 3.2 was used, with $\Delta$ representing the minimum discernible effect size, $Z_{\alpha}$ being set to 1.96 to represent a 95% certainty of a lack of a Type I error, $Z_{\beta}$ being set to 0.84 to represent a power of 80%, $\sigma$ being set as the standard deviation for the trials performed at each activation level, and $\mu$ being set as the mean force produced at that level. Table 4.12 shows the results of these calculations. As can be seen from these results, the largest minimum discernible effect size calculated was 0.020, or 2.0% of the mean force produced at any activation level. This is far below the 0.05 threshold set in Sections 3.7 and 3.8, meaning that 50 trials were enough at each activation level to achieve the desired level of certainty. Because of this, no further trials were required in order to improve the validity of the testing results.

Having determined the statistical validity of the results, the ANOVAs were performed. From the repeated measures one-way ANOVA results, available in Table 4.13, it is clear that the null hypothesis can be rejected for each of the activation levels’ force production while the actuator is in contraction. This means that each of the force production values for each activation level are statistically significantly different, which indicates that the addition of activated actuating pairs leads to statistically significant differences in force production for the actuator in isometric contraction. The same cannot be said for the actuator in relaxation. From the results from the same statistics test performed on the actuator results while in relaxation, seen in Table 4.14 it is

Table 4.12: Minimum discernible effect calculations by activation level, using produced force results from the force characterization experiment.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Calculated $\Delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>203.06</td>
<td>6.94</td>
<td>0.019</td>
</tr>
<tr>
<td>2</td>
<td>412.72</td>
<td>12.54</td>
<td>0.017</td>
</tr>
<tr>
<td>3</td>
<td>572.20</td>
<td>17.84</td>
<td>0.018</td>
</tr>
<tr>
<td>4</td>
<td>823.42</td>
<td>29.18</td>
<td>0.020</td>
</tr>
<tr>
<td>5</td>
<td>954.50</td>
<td>25.35</td>
<td>0.015</td>
</tr>
<tr>
<td>6</td>
<td>1150.02</td>
<td>36.48</td>
<td>0.018</td>
</tr>
</tbody>
</table>
Table 4.13: Repeated measures one-way ANOVA results for the force production of the actuator in contraction. All values had a Bonferroni correction applied, with a threshold of 0.05 signifying significance.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference (g)</th>
<th>Std. Error (g)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-209.66</td>
<td>1.97</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-369.14</td>
<td>2.78</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-620.36</td>
<td>4.26</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-751.44</td>
<td>3.59</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-947.14</td>
<td>5.04</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>209.66</td>
<td>1.97</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-159.48</td>
<td>2.92</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-410.70</td>
<td>4.48</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-541.78</td>
<td>4.15</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-737.48</td>
<td>5.12</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>369.14</td>
<td>2.783</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>159.48</td>
<td>2.92</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-251.22</td>
<td>4.64</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-382.30</td>
<td>4.35</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-578.00</td>
<td>5.64</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>620.36</td>
<td>4.257</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>410.700</td>
<td>4.475</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>251.220</td>
<td>4.643</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-131.080</td>
<td>5.427</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-326.780</td>
<td>6.882</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>751.44</td>
<td>3.592</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>541.78</td>
<td>4.15</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>382.30</td>
<td>4.35</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>131.08</td>
<td>5.48</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-195.70</td>
<td>6.23</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>947.14</td>
<td>5.04</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>737.48</td>
<td>5.12</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>578.00</td>
<td>5.64</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>326.78</td>
<td>6.88</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>195.70</td>
<td>6.23</td>
<td>&lt; 0.01</td>
</tr>
</tbody>
</table>
4.2 Force Characterization

Table 4.14: Repeated measures one-way ANOVA results for the force production of the actuator in relaxation. All values had a Bonferroni correction applied, with a threshold of 0.05 signifying significance.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference (g)</th>
<th>Std. Error (g)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-17.06</td>
<td>1.98</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>9.940</td>
<td>2.810</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>27.980</td>
<td>4.208</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>20.540</td>
<td>4.125</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>25.480</td>
<td>5.020</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>17.06</td>
<td>1.98</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>27.00</td>
<td>2.93</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>45.04</td>
<td>4.53</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>37.60</td>
<td>4.48</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>42.54</td>
<td>5.21</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-9.94</td>
<td>2.81</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-27.00</td>
<td>2.93</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>18.04</td>
<td>4.57</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>10.60</td>
<td>4.46</td>
<td>0.32</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>15.54</td>
<td>5.68</td>
<td>0.13</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>-27.98</td>
<td>4.21</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-45.04</td>
<td>4.53</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-18.04</td>
<td>4.57</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-7.44</td>
<td>5.77</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-2.50</td>
<td>6.66</td>
<td>1.00</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>-20.54</td>
<td>4.13</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-37.60</td>
<td>4.48</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-10.60</td>
<td>4.46</td>
<td>0.32</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>7.44</td>
<td>5.77</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>4.94</td>
<td>6.33</td>
<td>1.00</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>-25.48</td>
<td>5.02</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-42.54</td>
<td>5.21</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-15.54</td>
<td>5.68</td>
<td>0.13</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2.50</td>
<td>6.66</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-4.94</td>
<td>6.33</td>
<td>1.00</td>
</tr>
</tbody>
</table>

observed that the differences are for the most part not statistically significantly different, exceeding the 0.05 significance threshold. Only activation levels one and two are completely significantly different from the other activation levels, with level three not being significantly different from
levels five and six, level four not being significantly different from level five and six, and levels five and six not being significantly different from one another. This is similar to the results of the stroke characterization experiment, which also found that higher activation levels did not have statistically significant different results in relaxation. Also similar to that experiment, there is no easily discernible reason for why this is, but again there are some possible explanations that could be explored in future investigations. It could be that once TCAs relax past a certain point the amount they relax is confined to a range, regardless of how many actuators are relaxing at once. It could also be that relaxation is less predictable than contraction, giving rise to greater variability, which resulted in the above statistics. While these results were not expected for the two experiments, it is promising that this result was seen for the actuator in relaxation for both stroke and force production, as it indicates that it is a repeatable phenomena that can be explored in future works. Also similar to the stroke characterization experiment, the between-subjects factor testing found no statistical difference between different fabricated actuators ($p = 0.14, 0.63, 0.14, 0.55, 0.88$ and $p = 0.19, 0.38, 0.42, 0.56, 0.94$ for contraction and relaxation, respectively, at each trend order comparison from 1–5 in ascending order, and $p = 0.51$ and $p = 0.67$ for strictly between-subject comparisons for contraction and relaxation, respectively), indicating that the force production is similar between different fabricated actuators.

Similar to what was done for the stroke experiment, after calculating the confidence intervals (seen in Tables 4.15 and 4.16 for contraction and relaxation respectively) of the contraction of the actuator the linearity of the mean force production values were found via linear regression. This was again done using Microsoft Excel, the results of which can be seen in Figure 4.30. From the $R^2$ value, which again dictates better fitting the closer it is to one, it is apparent that the contraction of the actuator scales very linearly with the number of active pairs of TCAs when acting on a static load, increasing by 188.9 g per additional pair of TCAs. Converting this to a force value by multiplying it by the gravitational constant 9.81 m/s$^2$, this is 1.85 N per additional pair. This linearity is in accordance with both the literature [52], which states that heating TCAs is similar to changing the stiffness of a spring, and the results shown in the previous experiment.
Figure 4.30: Confidence intervals of the force applied by the actuator in grams equivalent compared to activation level, with an overlaid linear regression to show linearity of the results. The formula for the linear regression and its $R^2$ value are presented as well.

As explored in that section (4.1.5) of this thesis, increasing the pairs of TCAs being heated should linearly change the total stiffness of the actuator accordingly, which can be presented by an array of parallel springs. The same principle can be applied to force production, because the stiffness of a spring, the force it produces, and its elongation are all related by Hooke’s law [133]:

$$F_s = dK$$

In the previous experiment, the spring force $F_s$ was kept constant by applying constant loading to the actuator, and it was shown that linearly changing the total stiffness of the actuator would linearly change its elongation. In this experiment, the elongation was kept constant, so, using the same process as in Section 4.1.5, it can be shown that the force should scale linearly with a linearly changing total stiffness of the system.

For similar reasons as explored with the stroke characterization experiment, the linearity of the system is desirable as it allows for users and device designers to easily predict the behaviour of similar devices from limited data. Unlike with stroke though, there is no reason that this should
be limited as stroke scaling is by the maximum contraction range of the TCAs comprising the actuator [110]. This implies that any number of TCAs can be added to a woven actuator in order to produce the required force for the desired task, and that this range of force production can be controlled in a discretized fashion by changing the number of active TCAs being heated, with each additional pair adding 1.85 N in force production to the actuator as per the linear regression. Future avenues of inquiry could determine the factors affecting this interval, but if they could be made small enough it could lead to highly accurate discrete control of such actuators. Unfortunately, the actuator does not behave so in relaxation, with only activation levels one and two being statistically significantly different from all other activation levels. Past those activation levels, there is a high level of variability that prevents the accurate prediction of the force production of the actuator in relaxation based on linear principles. This again requires further investigation to determine its cause, but this unpredictability does necessitate the use of a workaround in order to properly control wearable therapeutic devices incorporating these actuators when they are in relaxation.

After finding the linearity of the results, using the methods outlined in Section 4.2.3, the force and pressure exerted by the actuator on the load cell in contraction were calculated, which can be seen in Table 4.17. These values were then divided by the number of active TCAs required to generate them, the results of which can be seen in Table 4.18. From the per TCA force production values, it can be seen that the average force produced per activated TCA in contraction is between 1.32 and 1.43 N when corrected for angle, or between 0.94 and 1.01 N when not.

Table 4.15: Force production values for each activation level of the actuator in contraction. All values in grams equivalent.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error</th>
<th>95% Confidence Interval</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>203.06</td>
<td>6.94</td>
<td>0.98</td>
<td>201.09</td>
<td>205.03</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>412.72</td>
<td>12.54</td>
<td>1.77</td>
<td>409.16</td>
<td>416.28</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>572.20</td>
<td>17.84</td>
<td>2.52</td>
<td>567.13</td>
<td>577.27</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>823.42</td>
<td>29.18</td>
<td>4.13</td>
<td>815.13</td>
<td>831.71</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>954.50</td>
<td>25.35</td>
<td>3.59</td>
<td>947.30</td>
<td>961.71</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1150.02</td>
<td>36.48</td>
<td>5.16</td>
<td>1139.83</td>
<td>1160.57</td>
<td></td>
</tr>
</tbody>
</table>
Table 4.16: Force production values for each activation level of the actuator in relaxation. All values in force grams equivalent.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error</th>
<th>95% Confidence Interval Lower Bound</th>
<th>95% Confidence Interval Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-45.04</td>
<td>8.11</td>
<td>1.15</td>
<td>-47.35</td>
<td>-42.73</td>
</tr>
<tr>
<td>2</td>
<td>-27.98</td>
<td>12.88</td>
<td>1.82</td>
<td>-31.64</td>
<td>-24.32</td>
</tr>
<tr>
<td>3</td>
<td>-54.98</td>
<td>17.55</td>
<td>2.48</td>
<td>-59.97</td>
<td>-49.99</td>
</tr>
<tr>
<td>4</td>
<td>-73.02</td>
<td>27.73</td>
<td>3.92</td>
<td>-80.90</td>
<td>-65.14</td>
</tr>
<tr>
<td>5</td>
<td>-65.58</td>
<td>27.33</td>
<td>3.87</td>
<td>-73.35</td>
<td>-57.81</td>
</tr>
<tr>
<td>6</td>
<td>-70.52</td>
<td>36.02</td>
<td>5.09</td>
<td>-80.76</td>
<td>-60.28</td>
</tr>
</tbody>
</table>

These non-corrected force production ranges are either comparable [9, 79, 80] to or better than [12, 48, 59] examples of actuating fibres and fabrics found in the literature on a per soft actuator basis, indicating that this actuator is capable of outperforming current iterations of actuating fabrics when operating in accordance with its intended purpose. Once the corrected force production is taken into account, however, it is clear that the actuator is able to bolster the force production capabilities of each individual actuating fibre, opening up the possibility for even further performance increases dependent on initial pennation angle. In addition, the additional force produced per TCA is fairly consistent, again reinforcing the linearity of the design, which as stated is desirable when attempting to scale it larger or smaller to fit different use cases.

Table 4.17: Calculated force and pressure production values of the actuator in contraction.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean Force (g)</th>
<th>Mean Force (N)</th>
<th>Mean Pressure (Pa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>203.06</td>
<td>1.99</td>
<td>3.16×10^5</td>
</tr>
<tr>
<td>2</td>
<td>412.72</td>
<td>4.05</td>
<td>6.43×10^5</td>
</tr>
<tr>
<td>3</td>
<td>572.20</td>
<td>5.61</td>
<td>8.91×10^5</td>
</tr>
<tr>
<td>4</td>
<td>823.42</td>
<td>8.08</td>
<td>1.28×10^6</td>
</tr>
<tr>
<td>5</td>
<td>954.50</td>
<td>9.36</td>
<td>1.49×10^6</td>
</tr>
<tr>
<td>6</td>
<td>1150.02</td>
<td>11.28</td>
<td>1.80×10^6</td>
</tr>
</tbody>
</table>
Table 4.18: Calculated force production values of the actuator in contraction, expressed in terms of the horizontal and angled force produced at the end effector of the actuator force and by each individual TCAs.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Horizontal Force per TCA (g)</th>
<th>Horizontal Force per TCA (N)</th>
<th>Angled Force per TCA (g)</th>
<th>Angled Force per TCA (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>101.53</td>
<td>1.00</td>
<td>143.59</td>
<td>1.41</td>
</tr>
<tr>
<td>2</td>
<td>103.18</td>
<td>1.01</td>
<td>145.92</td>
<td>1.43</td>
</tr>
<tr>
<td>3</td>
<td>95.37</td>
<td>0.94</td>
<td>134.87</td>
<td>1.32</td>
</tr>
<tr>
<td>4</td>
<td>102.93</td>
<td>1.01</td>
<td>145.56</td>
<td>1.43</td>
</tr>
<tr>
<td>5</td>
<td>95.45</td>
<td>0.94</td>
<td>134.99</td>
<td>1.32</td>
</tr>
<tr>
<td>6</td>
<td>95.83</td>
<td>0.94</td>
<td>135.53</td>
<td>1.33</td>
</tr>
</tbody>
</table>

### 4.2.6 Force Production Regression

Knowing that each activation level was statistically significantly different from one another, the results of the experiment were used in a regression and optimization procedure similar to that in Section 4.1.6 to create a univariate regression between temperature and force production for each activation level in each phase. The temperature and force production data were separated by activation level and phase, and a regression was performed with temperature as the independent variable and force as the dependent variable. The \texttt{polyfit} function from MATLAB was used, with the degree being optimized in order to determine the degree polynomial that would best represent each activation level’s force production by temperature in the actuator. The optimization criteria was chosen as the average RMSE per activation level once more, and was compared across polynomials of degree one to ten in a manner identical to that found in Section 4.1.6. The code used for carrying out this process can be found in Appendix B.1.2. Using that method, the average RMSE across all six activation levels was found for each degree polynomial in both contraction and relaxation, and are presented in Figure 4.31.
4.2 Force Characterization

(a) RMSE optimization curve for regression of the contraction phase.  
(b) RMSE optimization curve for regression of the relaxation phase.

Figure 4.31: Average RMSE per active pair for each degree of polynomial regressed.

The best performing degree polynomial was then selected. Again, the degree was chosen so as to minimize the average RMSE per activation level while not being within the asymptotic zone of the optimization curves. This was done to avoid over fitting of the data, that is, fitting the regression curves to the noise of the data rather than extracting information from the dataset [134]. Therefore, looking at Figure 4.31, it is clear that the regressed polynomials of degree five should be chosen for the actuator in heating, or contraction, while regressed polynomials of degree two should be chosen to represent the actuator in cooling, or relaxation. These give the following equations seen in Tables 4.19 and 4.20 for the force production of the actuator in contraction and relaxation, respectively. How these regressed equations fit the data can be seen in Figure 4.32 for the actuator in contraction and Figure 4.33 for the actuator in relaxation. \( F_{\text{prod}} \) represents the produced force in grams equivalent, \( T \) represents the temperature in degrees Celsius, the subscript \( H \) denotes while in contraction, and the subscript \( C \) denotes when in relaxation. The general form of the polynomials are shown prior to the tables, whose coefficients were found in the regression.
\[ F_{\text{prod},H} = A_H T^5 + B_H T^4 + C_H T^3 + D_H T^2 + E_H T + F_H \]

Table 4.19: Regressed equation coefficient values for relationship between temperature and force production for each activation level of the actuator in contraction.

<table>
<thead>
<tr>
<th>Pair</th>
<th>( A_H )</th>
<th>( B_H )</th>
<th>( C_H )</th>
<th>( D_H )</th>
<th>( E_H )</th>
<th>( F_H )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-9.11 \times 10^{-7}</td>
<td>2.39 \times 10^{-4}</td>
<td>-0.024</td>
<td>1.21</td>
<td>-26.96</td>
<td>245.17</td>
</tr>
<tr>
<td>2</td>
<td>-9.43 \times 10^{-6}</td>
<td>2.51 \times 10^{-3}</td>
<td>-0.26</td>
<td>12.29</td>
<td>-272.94</td>
<td>2266.01</td>
</tr>
<tr>
<td>3</td>
<td>-1.19 \times 10^{-5}</td>
<td>3.29 \times 10^{-3}</td>
<td>-0.35</td>
<td>17.60</td>
<td>-410.85</td>
<td>3584.51</td>
</tr>
<tr>
<td>4</td>
<td>-9.49 \times 10^{-6}</td>
<td>2.68 \times 10^{-3}</td>
<td>-0.29</td>
<td>14.32</td>
<td>-318.80</td>
<td>2548.62</td>
</tr>
<tr>
<td>5</td>
<td>-7.29 \times 10^{-6}</td>
<td>1.86 \times 10^{-3}</td>
<td>-0.17</td>
<td>7.07</td>
<td>-89.32</td>
<td>-253.39</td>
</tr>
<tr>
<td>6</td>
<td>-1.37 \times 10^{-5}</td>
<td>3.67 \times 10^{-3}</td>
<td>-0.37</td>
<td>17.38</td>
<td>-345.13</td>
<td>2183.26</td>
</tr>
</tbody>
</table>

\[ F_{\text{prod},C} = A_C T^2 + B_C T + C_C \]

Table 4.20: Regressed equation coefficient values for relationship between temperature and force production for each activation level of the actuator in relaxation.

<table>
<thead>
<tr>
<th>Pair</th>
<th>( A_C )</th>
<th>( B_C )</th>
<th>( C_C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.054</td>
<td>-2.62</td>
<td>-7.81</td>
</tr>
<tr>
<td>2</td>
<td>0.11</td>
<td>-6.13</td>
<td>65.09</td>
</tr>
<tr>
<td>3</td>
<td>0.16</td>
<td>-9.36</td>
<td>89.67</td>
</tr>
<tr>
<td>4</td>
<td>0.24</td>
<td>-12.83</td>
<td>111.13</td>
</tr>
<tr>
<td>5</td>
<td>0.25</td>
<td>-11.39</td>
<td>69.71</td>
</tr>
<tr>
<td>6</td>
<td>0.30</td>
<td>-14.01</td>
<td>97.67</td>
</tr>
</tbody>
</table>
4.2 Force Characterization

(a) Regression of force compared to temperature for data with one pair active.

(b) Regression of force compared to temperature for data with two pairs active.

(c) Regression of force compared to temperature for data with three pairs active.

(d) Regression of force compared to temperature for data with four pairs active.

(e) Regression of force compared to temperature for data with five pairs active.

(f) Regression of force compared to temperature for data with six pairs active.

Figure 4.32: Regression of force compared to temperature for the data collected when the actuator was in contraction.
4.2 Force Characterization

(a) Regression of force compared to temperature for data with one pair active.

(b) Regression of force compared to temperature for data with two pairs active.

(c) Regression of force compared to temperature for data with three pairs active.

(d) Regression of force compared to temperature for data with four pairs active.

(e) Regression of force compared to temperature for data with five pairs active.

(f) Regression of force compared to temperature for data with six pairs active.

Figure 4.33: Regression of force compared to temperature for the data collected when the actuator was in relaxation.
4.3 Comparing Stroke & Force to Clinical Requirements

After having completed the characterization experiments, the results were compiled and compared to the clinical requirements found in Chapter 2. The comparison characteristics were the range of motion, torque, and contraction period requirements for robot-assisted wrist neurorehabilitation, whose minimum, maximum, and average values found in the literature can be seen in Table 4.21.

Manipulative wrist rehabilitation can be idealized as a motion about a revolute joint; however, the designed actuator operates in a linear fashion, so the values must be transformed to properly compare them to the clinical requirements. This can be done by calculating the radial arm required to convert the force to a torque and stroke to a range of motion, and comparing between the two to see if it is feasible to meet both requirements at once, while still remaining small enough to be worn as a wearable therapeutic device. For converting force to torque, one multiplies it by the distance of a radial arm, as follows:

\[ \tau = F_{\text{linear}} r, \]

where \( \tau \) is the torque, \( F_{\text{linear}} \) is the linear force, and \( r \) is the distance of the radial arm on which the force is acting. Dividing the torque requirement by the average force, one can get the minimum required radial arm to provide that amount of torque, as follows:

\[ r_{\text{min}} = \frac{\tau}{F_{\text{linear}}}. \]

The maximum average force produced by the actuator was 11.28 N at six active pairs of TCAs, so dividing the values in Table 4.21 by that, one finds the calculated minimum radial arm distances. These are presented in Table 4.22.

Table 4.21: Clinical requirements calculated from values present in the literature, as found in Chapter 2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Torque (Nm)</td>
<td>0.146</td>
<td>4.00</td>
<td>1.30</td>
</tr>
<tr>
<td>Range of Motion (Degrees)</td>
<td>75</td>
<td>180</td>
<td>126.06</td>
</tr>
<tr>
<td>Contraction Period (s)</td>
<td>1.08</td>
<td>10.31</td>
<td>3.57</td>
</tr>
</tbody>
</table>
4.3 Comparing Stroke & Force to Clinical Requirements

Table 4.22: Minimum radial arm to convert the force production of the actuator to the desired torques for clinical applications.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Torque (Nm)</th>
<th>Force (N)</th>
<th>Calculated radial arm (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.146</td>
<td>11.28</td>
<td>0.013</td>
</tr>
<tr>
<td>Maximum</td>
<td>4.00</td>
<td>11.28</td>
<td>0.355</td>
</tr>
<tr>
<td>Average</td>
<td>1.30</td>
<td>11.28</td>
<td>0.115</td>
</tr>
</tbody>
</table>

While the maximum torque requirement yields a fairly large radial arm, the minimum torque requirement found in the literature requires a 1.29 cm long radial arm to achieve its required torque, and the average torque requirement found in the literature requires an 11.5 cm long radial arm. These cannot be compared or analyzed without considering the radial arm required for the range of motion, so it too must be calculated.

A similar process can be used to do this, translating from a linear frame of reference to a rotational one. The definition of a radian is as follows:

$$\theta = \frac{\gamma}{r},$$

where $\theta$ is the angle in radians, $\gamma$ is the arc length, and $r$ is the radial arm length. Letting $\gamma$ be equivalent to the linear stroke $s$, one can rearrange the relationship to get the following:

$$r_{max} = \frac{\gamma}{\theta},$$

this will mean that the larger the desired range of motion $\theta$ is, the smaller the radial arm $r$ will be. If the size of the radial arm for range of motion is smaller than the required radial arm for torque requirements, this means that both cannot be satisfied by the same radial arm. The maximum average stroke produced by the actuator was 0.01223 m at six pairs of active TCAs, so dividing that by the values in Table 4.21, one finds the calculated values in Table 4.23.
As can be seen, the required radial arms are quite small, with the minimum range of motion requiring a 0.93 cm long radial arm, and the average range of motion requiring a 0.56 cm long radial arm. All of the calculated radial arms for the maximum range of motion were found to be shorter than the minimum torque radial arm requirement of 1.3 cm, indicating that more force would be required to shorten the torque radial arms to make the actuator able to achieve both torque and range of motion clinical requirements. As found in Section 4.2.5, the relationship between force output and number of pairs of active TCAs was exceedingly linear, increasing by 188.9 grams force equivalent, or 1.85 N, per additional active pair. By finding the amount of force required to achieve the minimum torque requirement of 0.146 Nm at a radial arm of 0.93 cm, then finding the number of additional pairs of TCAs required to achieve that force, one can find the number of additional pairs of TCAs required to achieve both the minimum range of motion and minimum torque requirements, as follows:

\[
F_{\text{linear,req}} = \frac{\tau_{\text{min}}}{r_{\text{min}}},
\]

\[
F_{\text{linear,req}} = \frac{0.146}{0.0093},
\]

\[
F_{\text{linear,req}} = 15.70 \text{ N},
\]

\[
\delta F = F_{\text{linear,req}} - F_{\text{prod}},
\]

\[
\delta F = 15.70 - 11.28,
\]

\[
\delta F = 4.42 \text{ N},
\]

\[
i_{\text{add}} = \frac{\delta F}{m_{\text{force}}},
\]
4.3 Comparing Stroke & Force to Clinical Requirements

\[ i_{add} = \frac{4.42}{1.85}, \]

\[ i_{add} = 2.39, \]

where \( F_{\text{linear,req}} \) is the required linear force, \( \tau_{\text{min}} \) is the minimum therapeutic torque requirement, \( r_{\text{min}} \) is the radial arm required by the actuator to produce the minimum range of motion, \( \delta F \) is the difference in force between the required linear force and the force the actuator is capable of producing, \( i_{add} \) is the number of additional pairs of TCAs required, and \( m_{\text{force}} \) is the amount that the force changes per additional TCA, found from the linear regression in Section 4.2.5 to be 1.85 N per active pair. From this calculation, three additional pairs of TCAs are required to achieve both the minimum torque requirement and minimum range of motion requirement. Applying the same process, but with the radial arm required to achieve the average range of motion instead of the minimum, one gets the following:

\[ F_{\text{linear,req}} = \frac{\tau_{\text{min}}}{r_{\text{avg}}}, \]

\[ F_{\text{linear,req}} = \frac{0.146}{0.0056}, \]

\[ F_{\text{linear,req}} = 26.07 \text{ N}, \]

\[ \delta F = F_{\text{linear,req}} - F_{\text{prod}}, \]

\[ \delta F = 26.07 - 11.28, \]

\[ \delta F = 14.79 \text{ N}, \]

\[ i_{add} = \frac{\delta F}{m_{\text{force}}}, \]

\[ i_{add} = \frac{14.79}{1.85}, \]

\[ i_{add} = 7.99, \]

where \( r_{\text{avg}} \) is the radial arm required to achieve the average therapeutic range of motion. From the results of this calculation, eight additional pairs of TCAs are required to achieve the average
4.3 Comparing Stroke & Force to Clinical Requirements

required therapeutic range of motion while achieving the minimum torque requirement as well. Both this and the previous additional TCA requirements are easily met, with the constituent components of thread and yarn being easily sourced and inexpensive, meaning that operating at this torque and with this range of motion is quite feasible with a radial arm of 0.56 cm. This radial arm was also found to be an order of magnitude smaller than the amount that devices protruded from the wearer’s arm when looking at examples in the literature, with a cable-driven design protruding 7.4 cm [114], an SMA-based design protruding 8.8 cm or 6.8 cm, depending on targeted exercise [127], and a pneumatic design protruding 5.5 cm, plus transmission components [4]. However, applying the same process with the average torque requirement, rather than the minimum, one finds the following:

\[ F_{\text{linear,req}} = \frac{\tau_{\text{avg}}}{r_{\text{avg}}}, \]

\[ F_{\text{linear,req}} = \frac{1.30}{0.0056}, \]

\[ F_{\text{linear,req}} = 232.14 \text{ N}, \]

\[ \delta F = F_{\text{linear,req}} - F_{\text{prod}}, \]

\[ \delta F = 232.14 - 11.28, \]

\[ \delta F = 220.86 \text{ N}, \]

\[ i_{\text{add}} = \frac{\delta F}{m_{\text{force}}}, \]

\[ i_{\text{add}} = \frac{220.86}{1.85}, \]

\[ i_{\text{add}} = 119.38, \]

where \( \tau_{\text{avg}} \) is the average therapeutic torque requirement found in the literature. From the calculations, it can be seen that 120 additional pairs of TCAs are required to achieve both the average torque and range of motion requirements. This is not feasibly met, as even with their small footprint and light weight adding an additional 120 pairs of TCAs would be too large and cumbersome to make a wearable therapeutic device. In order to achieve both the average torque
and range of motion requirements, additional transmission components, such as gears, would be needed to increase the torque output of the actuator while keeping the number of additional TCAs to a manageable size. For example, a 1:12 gear ratio would mean that the actuator would have to produce 12 times less torque, giving the following required torque output of the actuator:

\[
\tau_{\text{gear}} = \frac{\tau_{\text{avg}}}{12},
\]

\[
\tau_{\text{gear}} = 0.108 \text{ N m},
\]

\[
F_{\text{linear,req}} = \frac{\tau_{\text{gear}}}{r_{\text{avg}}},
\]

\[
F_{\text{linear,req}} = \frac{0.108}{0.0056},
\]

\[
F_{\text{linear,req}} = 19.29 \text{ N},
\]

\[
\delta F = F_{\text{linear,req}} - F_{\text{prod}},
\]

\[
\delta F = 19.29 - 11.28,
\]

\[
\delta F = 8.01 \text{ N},
\]

\[
i_{\text{add}} = \frac{\delta F}{m_{\text{force}}},
\]

\[
i_{\text{add}} = \frac{8.01}{1.85},
\]

\[
i_{\text{add}} = 4.33,
\]

where \(\tau_{\text{gear}}\) is the required torque to achieve the average torque requirements when aided by a geared torque amplifying transmission system with a reduction ratio of 1:12. This lowers the increased force output required to meet the average therapeutic torque requirement, reducing the number of additional pairs of TCAs needed to five. By implementing both a transmission system and increasing the number of TCAs, the actuator can be made to scale to meet the clinical requirements for torque and range of motion; however, this would come at the cost of increased actuator size and contraction period, as gear reductions increase torque at the expense
4.3 Comparing Stroke & Force to Clinical Requirements

of actuation speed [135]. Further investigation into the contraction period, performed in Section 4.4, was required to determine to what degree the torque could be increased while still being able to meet the therapeutic contraction period requirements.

Without the use of torque amplifying transmission components such as gears, the actuator is able to achieve either the average torque or average range of motion through the use of a radial arm; however, it cannot meet both at the same time. With a radial arm of 0.56 cm, the actuator is able to meet the minimum torque requirements and average range of motion requirements while being significantly smaller than the examples found in the literature, while the linearly scaling force production of the actuator allows it to scale to easily meet or exceed the sizes of the actuators found in the literature, with only seven additional pairs of TCAs required to match the smallest protrusion size of 5.5 cm found using the same process as before:

\[
F_{\text{linear}, \text{req}} = \frac{\tau_{\text{avg}}}{r_{\text{small}}},
\]

\[
F_{\text{linear}, \text{req}} = \frac{1.30}{0.055},
\]

\[
F_{\text{linear}, \text{req}} = 23.64 \text{ N},
\]

\[
\delta F = F_{\text{linear}, \text{req}} - F_{\text{prod}},
\]

\[
\delta F = 23.64 - 11.28,
\]

\[
\delta F = 12.36 \text{ N},
\]

\[
i_{\text{add}} = \frac{\delta F}{m_{\text{force}}},
\]

\[
i_{\text{add}} = \frac{12.36}{1.85},
\]

\[
i_{\text{add}} = 6.68,
\]

where \(r_{\text{small}}\) is the smallest protrusion size found in the literature. Because of the incompatibility of the two radius arms required to achieve the two average requirements, and the inability to reasonably scale the force production of the actuator enough to make them compatible without
a torque amplifying transmission system, the actuator would have to be used in more specialized ways. As mentioned in Chapter 2, there are both manipulative and resistive components to neurorehabilitation, with the former guiding the patient’s wrist through its natural range of motion and the latter increasing strength. With a very small radial arm, the actuator could be used solely for manipulative purposes, being able to achieve range of motion requirements at the minimum required torque, while with a larger radial arm (5.5 cm, for instance) it could be used solely for resistive exercises, being able to meet the required torque but not the required range of motion. The actuator is in separate scenarios able to fill both roles, but requires changes in the actuator between different use cases. This could be achieved by making two different actuation devices, or by incorporating a user-adjustable radial arm, to allow the user to modify exercise range of motion and torque application with the guidance of a therapist. With a torque amplifying transmission system, this could be avoided, with the actuator being able to meet both average torque and average range of motion requirements; however, this could come at the cost of being unable to reach required actuation speeds without requiring significantly more power. Regardless of the implementation, additional components are required for the actuator to meet more than two of three of the average therapeutic torque, range of motion, and contraction period requirements calculated from the literature in Chapter 2.

4.4 Contraction Period Characterization

This section discusses the reasoning behind and analysis affecting the design of the contraction period characterization experiment, and the results, analysis, and conclusions that could be drawn from it. In particular, it explores the statistical analysis of the maximum temperature and stroke achieved by the actuator at a heightened contraction speeds, an exploration of the minimum achievable contraction period of the actuator, and a statistical analysis of the efficiency of the actuator in converting electrical to mechanical power at greater contraction speeds.

4.4.1 Purpose

The purpose of the contraction period characterization experiment was to investigate the ability of the designed actuator to achieve contractive speeds greater than the requirements found by
reviewing the literature. This was done in an isotonic configuration, supporting a static load vertically and contracting to raise the static load. This was done for two scenarios: when operated at a constant contraction period of 2.0 s, and when operating at the minimum possible contraction period given the power limitations imposed by the power control circuitry. In both cases, the maximum stroke for each activation level was measured and compared to the results of the stroke characterization experiment, to see to what degree increased actuation speeds affected contractive performance. As well, the efficiency of the system was calculated, as prior results indicated that faster heating time increased system efficiency, and it was desired to see to what degree this effect continued.

4.4.2 Time Characteristics Analytical Methods

Prior to being able to explore the maximum possible contraction speeds of the actuator, analysis was required to determine how the actuator performed in the previous two experiments. After performing all of the trials for each activation level on each actuator in both the stroke and force characterization experiments (found in Sections 4.1 and 4.2 respectively), the time required to heat the actuator at each activation level to an average of $80^\circ$ C for each trial was found, as well as the time required for the actuator to cool back down below $30^\circ$. This was done by finding the time elapsed for each trial in its heating phase, whose results can be found in Figures 4.11 and 4.28 for the heating time, and the time elapsed for each trial in its cooling phase, whose results can be found in Figures 4.12 and 4.29 in the previous experiments. After compiling these heating and cooling times, 100 in total for each phase at each activation level, they were further divided by orientation, with the stroke experiment taking place vertically and the force experiment taking place horizontally, to be able to differentiate between results obtained at different orientations. Confidence intervals were then calculated according to the method explained in Section 4.1.3 using SPSS. SPSS was also used to perform repeated measures one-way ANOVAs on the data according to the procedure detailed in Section 4.1.3 to establish whether the null-hypothesis could be rejected for the the heating time in contraction or the cooling time in relaxation for each activation level. The activation level was set as the within-subjects factor to be compared, and both orientation and actuation number were set as between-subject factors to be compared. Bonferroni corrections, as
in Section 4.1.3, were again applied to protect against errors stemming from repeated independent comparisons.

That having been completed, it should be noted that ANOVA comparisons are less adept at drawing conclusions from data when the data is known to be non-normal [136]. Because the data being interpreted is time data, it cannot be assumed to be normal; therefore, a different method is required to statistically analyze the results. To this end, a Kruskal-Wallis one-way analysis of variance was employed, as it is an analogue of one-way ANOVA between-subject comparisons but does not require a normal distribution of data [137]. Instead of calculating a test statistic based on variance about a common mean, instead the test statistic is calculated based on relative ranking of results, and how these ranks are dispersed between subjects. After it has been calculated, it is applied to the chi-square distribution with a corresponding degree of freedom, giving a $p$ value between 0 and 1. Just like the ANOVA test, if this $p$ value is lower than a critical significance threshold, the null-hypothesis can be rejected and the groupings can be considered statistically significantly different to a certainty of that threshold. SPSS was used to apply this Kruskal-Wallis analysis, using the different actuators at each activation level as one grouping parameter (analogous to a between-subject factor for an ANOVA test) for comparison and the orientation of the trials as another grouping parameter for comparison.

### 4.4.3 Time Characteristics Analysis

From the ANOVA analysis performed on the time data from the previous experiments, it can be seen in Table 4.24 that when the actuator is in heating each activation level has a statistically significantly different heating duration, with each having a significance value of less than 0.01, far below the 0.05 significance threshold. From Table 4.25, it can be seen that the cooling times largely also follow the same rule, save for pairs three and four which are over the threshold with a value of 0.25, indicating that there is not a statistically significant difference in their cooling times. For the between-subject factors comparisons, it was found there was no statistically significant difference between the actuator operating horizontally or vertically in either heating or cooling, with significance values of 0.34 and 0.90 respectively for strictly between-subject comparisons, and $p$ values of 0.38, 0.13, 0.052, 0.49, and 0.25, and 0.45, 0.93, 0.42, 0.53, and 0.30 for each order.
trend comparison in ascending order in heating and cooling, respectively, indicating the null-

Table 4.24: ANOVA results comparing the heating time of both experiments, grouped by activation level.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared to</th>
<th>Mean Difference (s)</th>
<th>Std. Error (s)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-8.66</td>
<td>0.11</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-19.01</td>
<td>0.13</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-30.55</td>
<td>0.16</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-44.07</td>
<td>0.27</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-57.50</td>
<td>0.31</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>8.66</td>
<td>0.11</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-10.35</td>
<td>0.16</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-21.89</td>
<td>0.17</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-35.42</td>
<td>0.27</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-48.85</td>
<td>0.32</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>19.01</td>
<td>0.13</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>10.35</td>
<td>0.16</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-11.54</td>
<td>0.19</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-25.07</td>
<td>0.28</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-38.49</td>
<td>0.33</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>30.55</td>
<td>0.16</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>21.89</td>
<td>0.17</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>11.54</td>
<td>0.19</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-13.53</td>
<td>0.31</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-26.96</td>
<td>0.32</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>44.07</td>
<td>0.27</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>35.42</td>
<td>0.27</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>25.07</td>
<td>0.28</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>13.53</td>
<td>0.31</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-13.43</td>
<td>0.41</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>57.50</td>
<td>0.31</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>48.85</td>
<td>0.32</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>38.49</td>
<td>0.33</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>26.96</td>
<td>0.32</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>13.43</td>
<td>0.41</td>
<td>&lt; 0.01</td>
</tr>
</tbody>
</table>

hypothesis cannot be rejected for any order trend comparing heating and cooling times between activation levels across orientation. For comparison between different fabricated actuators, it was found that different actuators had statistically insignificant impact on the heating and cooling
4.4 Contraction Period Characterization

Table 4.25: ANOVA results comparing the cooling time of both experiments, grouped by activation level.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared to</th>
<th>Mean Difference (s)</th>
<th>Std. Error (s)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-28.34</td>
<td>0.41</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-52.51</td>
<td>0.52</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-53.95</td>
<td>0.52</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-66.46</td>
<td>0.57</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-70.83</td>
<td>0.55</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>28.34</td>
<td>0.41</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-24.17</td>
<td>0.59</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-25.61</td>
<td>0.59</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-38.13</td>
<td>0.57</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-42.49</td>
<td>0.60</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>52.51</td>
<td>0.52</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>24.17</td>
<td>0.59</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-1.44</td>
<td>0.59</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-13.96</td>
<td>0.64</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-18.32</td>
<td>0.70</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>53.95</td>
<td>0.52</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>25.61</td>
<td>0.59</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.44</td>
<td>0.59</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-12.51</td>
<td>0.72</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-16.88</td>
<td>0.71</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>66.46</td>
<td>0.57</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>38.13</td>
<td>0.57</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>13.96</td>
<td>0.64</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>12.51</td>
<td>0.72</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>-4.37</td>
<td>0.70</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>70.83</td>
<td>0.55</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-28.34</td>
<td>0.41</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-52.51</td>
<td>0.52</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-53.95</td>
<td>0.52</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>-66.46</td>
<td>0.57</td>
<td>&lt; 0.01</td>
</tr>
</tbody>
</table>

times, with $p$ values of 0.11, 0.09, 0.18, 0.65, and 0.61, and 0.20, 0.67, 0.19, 0.81, and 0.91 for each order trend comparison in ascending order in heating and cooling, respectively, comparing the heating and cooling times at each activation level across actuators, and $p = 0.17$ and $p = 0.91$ for strictly between-subject comparisons for heating and cooling, respectively. These results indicate
that, for these woven bi-pennate actuators, adding additional activated pairs without increasing the power proportionally increases the heating time a statistically significant amount, and by and large also increases the cooling time a statistically significant amount. As well, having no statistically significant difference between the orientations or between different actuators indicates that changing both the orientation of the actuator and the specific fabricated actuator has no appreciable impact on the heating or cooling time, meaning that its performance will not change based on its configuration, such as when a user reorients themselves when wearing a device incorporating these actuators.

From the Kruskal-Wallis analysis, presented in Tables 4.26–4.29 for the heating results grouped by actuator number and orientation, and cooling results grouped by actuator number and orientation, respectively, it can be seen that for heating and cooling there are no statistically significant differences between actuators within the same activation level, nor are there statistically significant differences between orientations, with all significance values being greater than 0.05. This is beneficial, as it indicates that there is a non-appreciable difference in the heating and cooling performance of different actuators at the same activation level, regardless of orientation, which means that the actuators have predictable heating and cooling performance at each level, regardless of individual actuator or how the user is wearing it. This, in conjunction with the stroke and force production statistical analyses performed in the experiments discussed in Sections 4.1 and 4.2, indicate that the actuators have predictable performance, even between different instances of manufacturing. Despite these promising results, however, the heating times and cooling times found in the confidence interval calculations in Tables 4.30 and 4.31, and the ANOVA comparisons performed in Tables 4.24 and 4.25 need to be addressed. Having large heating times that increase significantly per activation level is non-ideal for wearable therapeutic devices, as the requirement for increased stroke or force production could limit their usability due to a slow actuation frequency.

Table 4.26: Kruskal-Wallis analysis results for each activation level of the actuator in heating, grouped by actuator number.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significance</td>
<td>0.98</td>
<td>0.09</td>
<td>0.15</td>
<td>0.71</td>
<td>0.60</td>
<td>0.18</td>
</tr>
</tbody>
</table>
4.4 Contraction Period Characterization

Table 4.27: Kruskal-Wallis analysis results for each activation level of the actuator in heating, grouped by orientation.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significance</td>
<td>0.94</td>
<td>0.07</td>
<td>0.86</td>
<td>0.92</td>
<td>0.42</td>
<td>0.078</td>
</tr>
</tbody>
</table>

Table 4.28: Kruskal-Wallis analysis results for each activation level of the actuator in cooling, grouped by actuator number.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significance</td>
<td>0.63</td>
<td>0.64</td>
<td>0.07</td>
<td>0.83</td>
<td>0.37</td>
<td>0.31</td>
</tr>
</tbody>
</table>

Table 4.29: Kruskal-Wallis analysis results for each activation level of the actuator in cooling, grouped by orientation.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significance</td>
<td>0.21</td>
<td>0.93</td>
<td>0.66</td>
<td>0.34</td>
<td>0.53</td>
<td>0.62</td>
</tr>
</tbody>
</table>

From the literature review performed in Section 2.7, it was determined that the average therapeutic device required a contraction period of 3.57 s, although some sources indicated that periods of 2.0 s [125] or even lower could be required in order to provide sufficient speed for repetitive flexion/extension exercises. In either case, this meant that the heating time of the designed actuator would have to be improved significantly in order to achieve speeds at which the actuator could be made clinically viable. Therefore, there was a need for further investigation into the minimum contraction period of the actuator, and what kind of performance it would have at elevated speeds. It was decided to first test the actuator operating at maximum power at a constant contraction period of 2.0 s, to see what level of performance could be obtained by doing so, and if required investigate the minimum contraction period the actuator could operate at while still retaining its complete range of contractive motion.

Once the need for further investigation into the contraction period of the actuator in heating had been confirmed, it was decided to pursue an experiment similar to that of the stroke characterization experiment in Section 4.1 so that the stroke production, efficiency, and heating times at faster actuation speeds could be explored. The first step towards designing this experiment was finding the minimum sample size required to have a statistically significant effect. This was done in a similar manner to that of the pilot experiments, by finding the mean and standard deviation of
Table 4.30: Calculated 95% confidence interval values of the time required for the actuator to heat by activation level and orientation. All time values in seconds. An orientation of zero represents being horizontal, while one represents being vertical.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Orientation</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error</th>
<th>95% Confidence Interval Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>13.04</td>
<td>0.59</td>
<td>0.082</td>
<td>12.88</td>
<td>13.20</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>13.031</td>
<td>0.57</td>
<td>0.082</td>
<td>12.87</td>
<td>13.14</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>21.53</td>
<td>0.93</td>
<td>0.13</td>
<td>21.27</td>
<td>21.80</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>21.85</td>
<td>0.93</td>
<td>0.13</td>
<td>21.59</td>
<td>22.11</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>32.07</td>
<td>1.04</td>
<td>0.16</td>
<td>31.76</td>
<td>32.39</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>32.017</td>
<td>1.19</td>
<td>0.16</td>
<td>31.70</td>
<td>32.33</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>43.56</td>
<td>1.52</td>
<td>0.22</td>
<td>43.13</td>
<td>43.99</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>43.60</td>
<td>1.55</td>
<td>0.22</td>
<td>43.17</td>
<td>44.03</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>57.33</td>
<td>2.72</td>
<td>0.37</td>
<td>56.60</td>
<td>58.07</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>56.89</td>
<td>2.50</td>
<td>0.37</td>
<td>56.15</td>
<td>57.62</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>70.02</td>
<td>2.98</td>
<td>0.43</td>
<td>69.17</td>
<td>70.87</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>71.06</td>
<td>3.08</td>
<td>0.43</td>
<td>70.21</td>
<td>71.91</td>
</tr>
</tbody>
</table>

Table 4.31: Calculated 95% confidence interval values of the time required for the actuator to cool by activation level and orientation. All time values in seconds. An orientation of zero represents being horizontal, while one represents being vertical.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Orientation</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error</th>
<th>95% Confidence Interval Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>52.16</td>
<td>2.23</td>
<td>0.34</td>
<td>51.49</td>
<td>52.83</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>51.61</td>
<td>2.54</td>
<td>0.34</td>
<td>50.94</td>
<td>52.28</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>80.14</td>
<td>3.52</td>
<td>0.48</td>
<td>79.19</td>
<td>81.09</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>80.31</td>
<td>3.25</td>
<td>0.48</td>
<td>79.36</td>
<td>81.26</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>104.16</td>
<td>4.44</td>
<td>0.64</td>
<td>102.90</td>
<td>105.42</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>104.63</td>
<td>4.55</td>
<td>0.64</td>
<td>103.37</td>
<td>105.89</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>106.29</td>
<td>4.64</td>
<td>0.62</td>
<td>105.06</td>
<td>107.53</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>105.38</td>
<td>4.16</td>
<td>0.62</td>
<td>104.15</td>
<td>106.62</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>118.07</td>
<td>4.84</td>
<td>0.71</td>
<td>116.67</td>
<td>119.47</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>118.63</td>
<td>5.13</td>
<td>0.71</td>
<td>117.23</td>
<td>120.03</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>122.47</td>
<td>4.53</td>
<td>0.71</td>
<td>121.06</td>
<td>123.88</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>122.96</td>
<td>5.48</td>
<td>0.71</td>
<td>121.55</td>
<td>124.37</td>
</tr>
</tbody>
</table>

the stroke and time values from the collected data in the heating phase and using those values with Equation 3.1 to determine the number of trials \( n \) required for the results to have a 5% chance of
a Type I error, 20% chance of a Type II error, and a minimum discernible effect size of 5% of the average stroke produced. The mean and standard deviation values, as well as the required number of trials to achieve the specified confidence levels, are shown in Tables 4.32 and 4.33 for the time and stroke values in heating, respectively. Equation 3.1 was used, with $Z_\alpha$ being set to 1.96 to reflect a 95% confidence in a lack of Type I error, $Z_\beta$ set to 0.84 to reflect a power of 80%, $\Delta$ set to 0.05 time the mean value, $\sigma$ set to the standard deviation, and $n$ being solved for to determine the required number of trials. The results can be seen in Tables 4.32 and 4.33.

From Tables 4.32 and 4.33, it can be seen that the largest number of trials required for an activated pair to achieve the specified level of variance and confidence was 28.20, which would be rounded up to 29 required samples. While this value was only calculated for when two pairs of TCAs were active, because it was the highest number found it was used as the representative minimum threshold for every active pair tested in the contraction period experiment, meaning that every active level should be tested at this minimum trial number threshold. To make the number of trials more easily divisible between actuators, this number of trials was increased to 30 per activation level. Based on the linearity of the results obtained in the previous experiments, it was decided that only three activation levels were required to be tested, as three levels is the minimum number required to affirm a linear relationship. If the results were found to be non-linear, additional activation levels would be tested as required. Therefore, it was decided that 30 trials per activation level for three activation levels should be performed in the contraction period characterization experiment, giving a total of 90 trials. With this knowledge, the experiment could then be executed with the knowledge it would give statistically significant results.

Table 4.32: Trial number calculations by activation level, using heating time results from both characterization experiments.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean (s)</th>
<th>Std. Deviation (s)</th>
<th>Calculated n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13.04</td>
<td>0.58</td>
<td>12.27</td>
</tr>
<tr>
<td>2</td>
<td>21.69</td>
<td>0.94</td>
<td>11.66</td>
</tr>
<tr>
<td>3</td>
<td>32.05</td>
<td>1.11</td>
<td>7.49</td>
</tr>
<tr>
<td>4</td>
<td>43.58</td>
<td>1.53</td>
<td>7.74</td>
</tr>
<tr>
<td>5</td>
<td>57.11</td>
<td>2.61</td>
<td>13.06</td>
</tr>
<tr>
<td>6</td>
<td>70.54</td>
<td>3.06</td>
<td>11.80</td>
</tr>
</tbody>
</table>
Table 4.33: Trial number calculations by activation level, using produced stroke results from the stroke characterization experiment.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean (s)</th>
<th>Std. Deviation (s)</th>
<th>Calculated n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.74</td>
<td>0.15</td>
<td>19.56</td>
</tr>
<tr>
<td>2</td>
<td>4.34</td>
<td>0.29</td>
<td>28.20</td>
</tr>
<tr>
<td>3</td>
<td>6.33</td>
<td>0.35</td>
<td>19.27</td>
</tr>
<tr>
<td>4</td>
<td>8.55</td>
<td>0.13</td>
<td>1.38</td>
</tr>
<tr>
<td>5</td>
<td>10.82</td>
<td>0.11</td>
<td>0.59</td>
</tr>
<tr>
<td>6</td>
<td>12.23</td>
<td>0.12</td>
<td>0.60</td>
</tr>
</tbody>
</table>

4.4.4 Constant Period Experimental Methods

This experiment aimed to assess the stroke production capacity of the actuator operating at maximum power at a constant 2.0 s contraction period. The MOSFET transistors used in the construction of the PCB for the stroke and force characterization experiment could withstand sustained electrical signals up to 20 V and 2 A each, so that placed a hard cap on the maximum power that could be used per TCA. The programmable DC power supply (Rigol DP811) was able to supply up to 10 A at 20 V, meaning for any activation level past two active pairs of TCAs additional power supplies would be required. To facilitate this, an Agilent E3620A Dual Output power supply capable of 1 A at 20 V in each output was connected in parallel to the original supply. As the Agilent was a dual supply model both of its outputs we connected in parallel, allowing the system to have a maximum achievable power of 12 A at 20 V, or 240 W, which would be enough to supply the desired three activated pairs at the maximum power. This can be seen in Figure 4.34. Thirty six TCAs were then fabricated and trained according to the procedures outlined in Sections 3.1 and 3.2, respectively. This number was chosen because six actuators composed of six TCAs were desired. These numbers were chosen so as to have five trials per actuator to achieve the required 30 trial number found from the sample size calculations, and the actuators were reduced to only have three pairs of TCAs because previous results found in Section 4.1 indicate that stroke production per activated pair is linear, and three data points is the minimum number required to reaffirm this linearity. Otherwise, the components used were the same as the ones used in the stroke characterization experiment. After preparing the materials necessary for fabricating
the woven bi-pennate actuator, the actuator was constructed. The upright support holsters were clamped to a levelled lab bench 141 mm apart, measured with a digital Vernier caliper, to create an initial 45° pennation angle for the actuator, and the side tendons were slid into the holsters. The TCAs were then slid over the steel pins embedded in the side tendons on either side and held in place by sliding their relevant pin caps through the hole in the pins, holding them in place via friction. The TCAs were also attached to the central tendon in a similar fashion. This can be seen in Figure 4.35. After attaching the TCAs to the tendons, 300 g of calibration weights were hung from the hook on the central tendon in order to keep the actuator taught during weaving. The IR sensor holder used in the stroke pilot and characterizations experiments was slid over the bottom of the calibration weights and aligned with a piece of cardboard with two layers of masking tape (to control reflectivity) to record the stroke of the actuator during the experiment. The thermocouples were then attached, with each numbered thermocouple attached to its respective TCA. This was done in a similar manner to the previous experiments, by tying them to their TCA using nylon sewing thread, and then applying thermal paste (Halzniye 510) liberally to eliminate air pockets, ensure good thermal conductivity, and provide electrical insulation between the two components.
4.4 Contraction Period Characterization

Figure 4.35: An actuator used in the constant period experiment after having the thermocouples attached to its TCAs.

After the thermocouples were attached, the actuator was woven using black 100% nylon 18 weight yarn (Part No. 1005019 from anniescatalog.com). This was done using a standard over-under manual weaving technique using the TCAs as the warp, compressing the weft with every pass to make the woven matrix as tight as possible. One side was done at a time, with the weaving continuing until the thermocouples had been covered by a minimum of three loops on each side. A picture of an actuator halfway through the weaving process can be seen in Figure 4.36. Once both sides of the actuator had been woven, the mass applied to it was increased to 600 g, to give it its experimental loading amount. This mass was chosen because it was half of that used in the stroke characterization experiment, as this experiment only used half of the number of activated pairs. The upright holsters were adjusted to ensure the $45^\circ$ pennation angle was still maintained, after which the actuator was considered fully fabricated and ready for experimentation. A completed actuator used in this experiment can be seen in Figure 4.37.
4.4 Contraction Period Characterization

Figure 4.36: An actuator used in the constant period experiment after being half-woven.

In this experiment, the number of pairs of TCAs being heated, or levels of activation of the actuator, were iteratively tested to determine the heating rate, stroke production capacity, and efficiency of the actuator operated at a contraction period of 2.0 s under the maximum power that could be supplied by its controlling circuitry. 2.0 s was chosen as a target as it was a value higher than the average calculated period requirement found in the literature. This was done by randomly choosing a pair of TCAs (to prevent selection bias) and heating them at 40 W per TCA (20 V and 2 A per TCA, making it 4 A for one pair) supplied by the parallel DC power supplies). This was allowed to continue for two seconds, before the MOSFETS were turned off, cutting off the power to the TCAs. The TCAs were then allowed to cool until the both reached temperatures of less than or equal to 30° C. Throughout the heating and cooling phases of the trial, the temperature, distance, and power consumed were recorded using the attached ESP32 microcontroller, which also controlled the heating and cooling process. After preliminary testing, the sampling rate was found to be 0.11 s. After the TCAs were cooled, this was repeated with another random activated
4.4 Contraction Period Characterization

pair, repeating for a total of five trials. After these five trials had been run, the number of activated pairs of TCAs was increased by one, and the process done again with an additional 40 W of power per TCA (increase of supply by 2 A per TCA) for an additional five trials. After five trials with two activated pairs, three activated pairs were tested with a further increase of 40 W per TCA, or a total power supply of 240 W (20 V at 12 A) giving an additional five trials and a total of 15 trials per actuator. After these were complete, the actuator was disassembled and a new actuator was constructed using a new set of six TCAs, following the same construction steps as above. The experimentation process was then repeated on the new actuator, giving an additional 15 trials worth of data. This was repeated until six actuators had been tested, giving a total of 30 trials per activation level. The microcontroller code used to control this experiment is presented in Appendix C.3.4.

4.4.5 Constant Period Analytical Methods

In order to analyze the results of the constant period experiment, SPSS was used to perform one-way repeated measures ANOVA tests as in Section 4.1.3, comparing the stroke produced and
maximum temperature achieved between activation levels. The within-subject factor was chosen to be the activation level for both tests, and the between-subject factor was again chosen to be the actuator number for both tests. SPSS was also used to construct confidence intervals for the likely stroke production and maximum temperature values, as in Section 4.1.3.

4.4.6 Constant Period Results

Similar to the previous experiments, after the data for the trials had been collected the data were processed and made usable for analysis. The temperature collected for each active TCA was averaged at each time point in each trial to get the average active temperature of the actuator, and the initial distance recorded by the IR sensor was subtracted from subsequent values in each trial to get the stroke of the actuator. The total processed data can be seen in Figure 4.38, which also gives examples of individual trials. Each trial was then separated to get the heating phase by finding the point at which the temperature began to decrease while the power consumed by the actuator was zero, at which point it was considered to be cooling. The heating phase data can be seen in Figure 4.39. After separating the data, the maximum temperature and stroke in the heating phase of each trial were found and recorded.
4.4 Contraction Period Characterization

(a) Total data for the trials with one pair of TCAs active.

(b) Total data for the trials with two pairs of TCAs active.

(c) Total data for the trials with three pairs of TCAs active.

(d) Example of data for one trial with one pair of TCAs active.

(e) Example of data for one trial with two pairs of TCAs active.

(f) Example of data for one trial with three pairs of TCAs active.

Figure 4.38: Data collected for the constant period experiment. Different time scales were used for clarity purposes.
4.4 Contraction Period Characterization

(a) Total data for the trials with one pair of TCAs active.

(b) Total data for the trials with two pairs of TCAs active.

(c) Total data for the trials with three pairs of TCAs active.

(d) Example of data for one trial with one pair of TCAs active.

(e) Example of data for one trial with two pairs of TCAs active.

(f) Example of data for one trial with three pairs of TCAs active.

Figure 4.39: Data collected for the contraction phase of the constant period experiment.
4.4.7 Constant Period Analysis

From the results of the repeated measures one-way ANOVA performed on the stroke production results, seen in Table 4.34, it is clear that the null hypothesis can be rejected \( (p < 0.01) \) for the stroke measured for each of the activation levels in contraction, signifying that each activation level changes the contraction of the actuator a significant amount when moving a static load. This is in line with the results from the stroke characterization experiment in Section 4.1.4, and was expected, giving further confirmation to the results from that experiment. Similarly according to expectations, from the results presented in Table 4.35 it was found that there was no statistically significant difference between the temperatures reached by the actuator in contraction. This was expected because the power and heating duration were kept constant for each trial, meaning the same amount of thermal energy was applied to each TCA in each trial which should raise the temperature to the same level. Both analyses also found no statistically significant differences between different fabricated actuators from the between-subjects factors testing \( (p = 0.93, 0.39 \) and \( p = 0.57, 0.73 \) for stroke and maximum temperature respectively, with the first value referring to the linear trend and the second value referring to the quadratic trend between values at each activation level and actuator number, and \( p = 0.93 \) and \( p = 0.91 \) for strictly between-subjects comparisons for stroke and maximum temperature, respectively), which was also expected due to similar results from the stroke and force characterization experiments.

In addition to the statistical significance of each trial being as expected, the linearity of the produced stroke was also quite favourable. In a similar manner to the previous characterization

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference</th>
<th>Std. Error</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-3.23</td>
<td>0.06</td>
<td>(&lt; 0.01)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-6.43</td>
<td>0.07</td>
<td>(&lt; 0.01)</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3.23</td>
<td>0.06</td>
<td>(&lt; 0.01)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-3.20</td>
<td>0.07</td>
<td>(&lt; 0.01)</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>6.43</td>
<td>0.07</td>
<td>(&lt; 0.01)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3.20</td>
<td>0.07</td>
<td>(&lt; 0.01)</td>
</tr>
</tbody>
</table>
4.4 Contraction Period Characterization

Table 4.35: Repeated measures one-way ANOVA analysis results for maximum temperature achieved by the actuator operating at a constant contraction period of 2.0 s. All values had a Bonferroni correction applied, with a threshold of 0.05 signifying significance.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference</th>
<th>Std. Error</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.00</td>
<td>0.20</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.19</td>
<td>0.17</td>
<td>0.79</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.00</td>
<td>0.20</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.19</td>
<td>0.20</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.19</td>
<td>0.17</td>
<td>0.79</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.19</td>
<td>0.20</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Experiments, Microsoft Excel was used to perform a linear regression on the mean produced values, seen in Table 4.36, to see to what degree they followed a linear trend. The results of this can be seen in Figure 4.40, which shows the constructed confidence intervals overlaid with the linear regression. The coefficient of correlation, or $R^2$ value, is also presented. With an $R^2$ value of 0.9999, it can be seen that the stroke produced by the actuator was extremely linear, increasing by 3.22 mm per additional active pair according to the slope of this linear trend. This reinforces the results of the stroke characterization experiment, which found that the designed actuator has linearly scaling stroke production based on number of TCAs active with a static load, even at elevated speeds.

Figure 4.40: Confidence intervals of the percent contraction compared to activation level for the constant period characterization experiment, with an overlaid linear regression to show linearity of the results. The formula for the linear regression and its $R^2$ value are presented as well.
4.4 Contraction Period Characterization

Table 4.36: Constructed 95% confidence interval for maximum stroke production values for the actuator running at a constant contraction period. All values in mm.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>3.90</td>
<td>0.27</td>
<td>0.05</td>
<td>3.80</td>
</tr>
<tr>
<td>2</td>
<td>7.13</td>
<td>0.23</td>
<td>0.04</td>
<td>7.05</td>
</tr>
<tr>
<td>3</td>
<td>10.33</td>
<td>0.24</td>
<td>0.05</td>
<td>10.24</td>
</tr>
</tbody>
</table>

Despite these favourable results for activation level difference and linearity, the actuator fell short of the desired performance, which can be seen in the constructed confidence intervals in Tables 4.36 and 4.37 for stroke and maximum temperature respectively. Dividing the stroke values by the mean values of similar activation levels (levels two, four, and six for the prior experiment compared to levels one, two, and three from this experiment, due to the halved loading) found from the stroke characterization experiment in Section 4.1, one can find the percent comparison between the attained strokes in this period experiment and those found in that previous experiment. Table 4.38 shows this comparison.

Table 4.37: Constructed 95% confidence interval for maximum achieved temperature values for the actuator running at a constant contraction period. All values in °C.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>72.35</td>
<td>0.86</td>
<td>0.17</td>
<td>72.00</td>
</tr>
<tr>
<td>2</td>
<td>72.35</td>
<td>0.82</td>
<td>0.15</td>
<td>72.04</td>
</tr>
<tr>
<td>3</td>
<td>72.54</td>
<td>0.88</td>
<td>0.17</td>
<td>72.19</td>
</tr>
</tbody>
</table>

Table 4.38: Stroke production of the actuator operating at a constant contraction period of 2.0 s, compared to the corresponding results from the stroke characterization experiment. All values expressed as a percentage.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>0.90</td>
<td>0.87</td>
</tr>
<tr>
<td>2</td>
<td>0.84</td>
<td>0.82</td>
</tr>
<tr>
<td>3</td>
<td>0.84</td>
<td>0.84</td>
</tr>
</tbody>
</table>
These results show that, when operated at a contraction period of 2.0 s at a power level of 40 W per TCA, the actuator is limited in its contraction range. While this could still make it suitable for some neurorehabilitative applications, it is desirable to find the minimum period attainable that produces the same actuation capabilities seen in previous experiments. To this end, a further experiment was deemed necessary to determine at what point the actuator could achieve its full range of motion. From the literature review performed in Chapter 2, the average required contraction period required for wrist neurorehabilitation is 3.57 s, meaning that as long as the found minimum period was less than this value it would remain a viable means of actuating a neurorehabilitative wrist device.

In order to design this minimum period experiment, it was decided that the same temperature threshold of 80°C would be used as the threshold of having achieved full motion. Based on this decision, the required sample size for the experiment was calculated from the maximum temperature confidence interval data. This was done using Equation 3.1, which had been used previously to calculate statistically significant sample sizes. The same $Z_\alpha$ and $Z_\beta$ values were used, with 1.96 representing a 95% certainty there was no Type I error for the $Z_\alpha$ value and 0.84 representing a power of 80% for the $Z_\beta$ value. The minimum discernible effect $\Delta$ was set to 1°C, rather than a percentage, as the final temperature value would be similar for each of the activation levels tested. The standard deviations $\sigma$ were taken from Table 4.37, which detailed the temperature results from the constant period experiment. These gave the calculated $n$ values found in Table 4.39. From these calculations, it was determined that a minimum of 13 repetitions would have to be performed for each activation level for all of them to have the desired level of statistical significance. This was rounded to 15, to allow for even splitting between actuators.

Table 4.39: Minimum sample size calculations by activation level, using achieved temperature results from the constant period characterization experiment.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Std. Deviation</th>
<th>Calculated $n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.86</td>
<td>11.67</td>
</tr>
<tr>
<td>2</td>
<td>0.82</td>
<td>10.58</td>
</tr>
<tr>
<td>3</td>
<td>0.88</td>
<td>12.13</td>
</tr>
</tbody>
</table>
4.4.8 Minimum Period Experimental Methods

In this experiment, the number of pairs of TCAs being heated, or levels of activation of the actuator, were iteratively tested to determine the heating rate, stroke production capacity, and efficiency of the actuator when heated from room temperature to $80^\circ$ C under the maximum power that could be supplied by its controlling circuitry, in order to determine the minimum contraction period that could be obtained at this power level. This was done in a manner similar to that of the constant period experiment previously completed, using largely the same equipment. 18 TCAs were fabricated and trained according to the methods outlined in Sections 3.1 and 3.2 respectively, enough to create three actuators with three bi-pennate pairs of TCAs each. Using the same 3D printed components as in Section 4.1.2, set up in the same manner, the TCAs were strung from the embedded metal pins to connect the rigid plastic tendons. The central tendon was then loaded with 300 g of calibration weights to ensure the actuator was taught during weaving.

The IR sensor holder used in the previous stroke measurement experiments was slid over the bottom of the calibration weights and aligned with a piece of cardboard with two layers of masking tape to record the stroke of the actuator during the experiment. The thermocouples were then attached, with each numbered thermocouple attached to its respective TCA. This was done in a similar manner to the previous experiments, by tying them to their TCA using nylon sewing thread, and then applying thermal paste liberally to eliminate air pockets, ensure good thermal conductivity, and provide electrical insulation between the two components. An example of an actuator having had thermocouples attached to it is seen in Figure 4.41. The actuator was then woven using new lengths of the same yarn as before, with the same over-under and compression technique. An example of a half-woven actuator can be seen in Figure 4.42. The mass applied to the actuator was then increased to 600 g, to give the same mass as in the previous period experiment. The upright holsters were then adjusted to ensure the $45^\circ$ pennation angle was still maintained, after which the actuator was considered fully fabricated and ready for experimentation. Examples of a fully fabricated actuator can be seen in Figure 4.43.
To complete the objectives of the experiment, a pair of TCAs was randomly selected and heated at 40 W per TCA (20 V and 2 A per TCA, or 4 A per pair) supplied by the parallel DC power supply setup described in Section 4.4.4. This was allowed to continue until the active TCAs were 80°C or hotter before the MOSFETS were turned off, cutting off the power to the TCAs. The TCAs were then allowed to cool until they both reached temperatures of less than or equal to 30°C. Throughout the heating and cooling phases of the trial, the temperature, distance, and power consumed were recorded using the attached ESP32 microcontroller, which also controlled the heating and cooling process. After preliminary testing, it was found that the sampling rate was 0.11 seconds. After the active pair of TCAs had successfully heated and cooled, this was
repeated with another random activated pair, repeating for a total of five trials. After these five trials had been run, the number of activated pairs of TCAs was increased by one, and the process done again with an additional 40 W of power per TCA (increase of supply by 2 A per TCA) for an additional five trials. After five trials with two activated pairs, three activated pairs were tested with a further increase of 40 W per TCA, or a total power supply of 240 W (20 V at 12 A) giving an additional five trials and a total of 15 trials per actuator. After these were complete, the actuator was disassembled and a new actuator was constructed using a new set of six TCAs, following the same construction steps as above. The experimentation process was then repeated on the new actuator, giving an additional 15 trials worth of data. This was repeated until three actuators had been tested, giving a total of 15 trials per activation level, the number decided on.
from the results of the calculations done using the results from the constant period experiment. The microcontroller code used to control this experiment can be seen in Appendix C.3.5.

4.4.9 Minimum Period Analytical Methods

After finding the maximum temperature, maximum stroke produced, and time required to reach the maximum temperature, SPSS was used to perform one-way repeated measures ANOVAs on the data to determine the relationship between the activation levels for each set of information. Similar to the constant period experiment, the between-subject factor was set to the actuator number, and the within-subject factor was set to be the activation level. The software was also used to construct confidence intervals (set to 95% confidence) to determine the span of likely values for each piece of information. Both of these were done in a manner similar to the methods outlined in Section 4.1.3. For reasons explored in Section 4.4.3, due to the nature of time data a Kruskal-Wallis test was also applied to the heating time data. This was done with the same method explained in Section 4.4.2, using the actuator number as the grouping variable.

In addition to investigating whether the contraction period of the actuator could be made suitable for its intended purpose, another objective of these experiments was to investigate if the
increased actuation speed could also increase efficiency by lowering the amount of time for heat to dissipate from the actuator before it reached its maximum temperature. For each trial in each contraction period experiment, the energy consumed was calculated by multiplying the power consumption of the actuator between two time samples in the trial by the amount of time that elapsed in that interval. This was done for every sample for the heating phase of each trial, then summed to get the total energy consumed for that trial. The work done by the actuator in this time frame was calculated by multiplying the maximum stroke by the gravitational force exerted by the calibration weights on the actuator. Dividing the work done by the energy consumed gave the total efficiency of the actuator for that trial. After doing this for each trial in each experiment, the results were used in SPSS to perform a repeated measures one-way ANOVA analysis on the differences between each activation level, and to construct confidence intervals for their likely values. This again was done using the methods detailed in Section 4.1.3.

Finally, as discussed in Section 4.3, it was desirable to see to what degree the contraction period could be increased to enable the use of torque amplifying transmission components, such as gear reduction, to better enable the actuator to meet average therapeutic requirements. To do this, the average therapeutic period requirement of 3.57 s (as calculated in Section 2.7) was divided by the minimum contraction period found, to determine what the maximum speed reduction via gearing could be while still being able to achieve the average therapeutic period requirement. The average required torque found from the literature review was then divided by this value, to find the total torque the actuator would have to supply, and used to calculate the number of additional pairs of TCAs required to generate that torque at the radial arm required to produce the minimum and average required therapeutic range of motion. This was done in a similar process to that found in Section 4.3.

4.4.10 Minimum Period Results

After the data were collected for each of the trials, the results were processed and made usable in a similar fashion to previous experiments. The individual temperatures collected by the thermocouples were averaged across the number of active TCAs to get the average activated temperature of the actuator, and the initial distance read by the IR sensor was subtracted from subsequent readings
in each trial to get the stroke of the actuator during a trial. The total data collected for each
activation level can be seen in Figure 4.44, which also gives examples of individual trials’ data.
Each trial was separated to get the heating phase by determining when the temperature began
to decrease after the power consumption of the actuator had dropped to zero, and the duration,
maximum temperature, and stroke of the actuator in the heating phase were recorded. The data for
the heating phase of each trial can be seen in Figure 4.45, which also gives examples of individual
trials’ data.
4.4 Contraction Period Characterization

(a) Total data for the trials with one pair of TCAs active.
(b) Total data for the trials with two pairs of TCAs active.
(c) Total data for the trials with three pairs of TCAs active.
(d) Example of data for one trial with one pair of TCAs active.
(e) Example of data for one trial with two pairs of TCAs active.
(f) Example of data for one trial with three pairs of TCAs active.

Figure 4.44: Total data collected for the minimum period experiment. Different time scales used for clarity purposes.
4.4 Contraction Period Characterization

(a) Total data for the trials with one pair of TCAs active.

(b) Total data for the trials with two pairs of TCAs active.

(c) Total data for the trials with three pairs of TCAs active.

(d) Example of data for one trial with one pair of TCAs active.

(e) Example of data for one trial with two pairs of TCAs active.

(f) Example of data for one trial with three pairs of TCAs active.

Figure 4.45: Total data collected for the contraction phase of the minimum period experiment.
4.4.11 Minimum Period Analysis

After completing the repeated measures one-way ANOVA test for the maximum stroke produced at each activation level, seen in Table 4.40, it can be seen that, similar to the results of the constant period experiment, the results show that the null hypothesis can be rejected ($p < 0.01$) for the stroke measured for each of the activation levels in contraction, signifying that each activation level changes the contraction of the actuator a significant amount when moving a static load. The maximum temperature analysis, seen in Table 4.41, also gave similar results to the constant period experiment, indicating that there was no statistically significant difference between the temperatures reached by the actuator in contraction. This was expected, but for different reasons; in this experiment, the maximum temperature threshold was the same for all trials, which would lead one to expect that all of the maximum temperatures should be similar. Also similar to previous experiments, the between-subjects factor testing performed indicated that the stroke produced, maximum temperature, and heating time were not statistically significantly different from one another between different actuators, with $p$ values for the linear trend and quadratic trend comparisons of 0.13 and 0.61 for the stroke produced, 0.45 and 0.22 for the maximum temperature achieved, and 0.61 and 0.78 for the time required to heat to that value. The purely between-subjects comparisons resulted in $p$ values of 0.52, 0.27, and 0.95 for maximum stroke, maximum temperature, and heating time, respectively. This further shows that there is little variance between different fabricated actuators, as also found in previous characterization experiments. The ANOVA analysis, seen in Table 4.42, and the Kruskal Wallis test, seen in Table 4.43, found that the times required to reach these maximum values were not statistically significantly different between activation levels or different actuators either, with both tests exceeding the critical threshold of 0.05. This too was expected, as the power available to the actuator was kept constant at 40 W per active TCA and the maximum temperature reached was expected to be similar. At a constant final temperature, initial temperature, and power, the time required to reach the final temperature should always be similar, which was confirmed.
4.4 Contraction Period Characterization

Table 4.40: Repeated measures one-way ANOVA analysis results for maximum stroke produced by the actuator operating at its minimum contraction period. All values had a Bonferroni correction applied, with a threshold of 0.05 signifying significance.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference (mm)</th>
<th>Std. Error (mm)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-4.16</td>
<td>0.11</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-7.84</td>
<td>0.11</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>4.16</td>
<td>0.11</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-3.68</td>
<td>0.08</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>7.84</td>
<td>0.11</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3.68</td>
<td>0.08</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 4.41: Repeated measures one-way ANOVA analysis results for maximum temperature achieved by the actuator operating at its minimum contraction period. All values had a Bonferroni correction applied, with a threshold of 0.05 signifying significance.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference °C</th>
<th>Std. Error °C</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.23</td>
<td>0.37</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.20</td>
<td>0.38</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-0.23</td>
<td>0.37</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.43</td>
<td>0.25</td>
<td>0.33</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.20</td>
<td>0.38</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.43</td>
<td>0.25</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Table 4.42: Repeated measures one-way ANOVA analysis results for the time required to achieve maximum temperature in heating by the actuator operating at its minimum contraction period. All values had a Bonferroni correction applied, with a threshold of 0.05 signifying significance.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference (s)</th>
<th>Std. Error (s)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.00</td>
<td>0.05</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.07</td>
<td>0.07</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.00</td>
<td>0.05</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.07</td>
<td>0.08</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-0.07</td>
<td>0.069</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.07</td>
<td>0.08</td>
<td>1.00</td>
</tr>
</tbody>
</table>
4.4 Contraction Period Characterization

Table 4.43: Results of Kruskal-Wallis test performed on the heating time results of the actuator operating at its minimum possible contraction period. A threshold of 0.05 denotes significance.

<table>
<thead>
<tr>
<th>Test Parameter</th>
<th>Activation Level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>H-Test</td>
<td>0.42</td>
</tr>
<tr>
<td>DoF</td>
<td>2</td>
</tr>
<tr>
<td>Significance</td>
<td>0.81</td>
</tr>
</tbody>
</table>

While operating at a slower speed than the previous experiment, it was able to produce greater stroke values due to the higher temperatures reached, which can be seen in Table 4.44, which was the goal of this experiment. Dividing the stroke values obtained from this experiment by the same mean values from the stroke characterization experiment (see Section 4.4.7 for further explanation), it appears that the produced stroke values are actually slightly higher than those from the previous experiment at an activation level of one, as seen from the results of doing so in Table 4.45. This could be from the use of different TCAs than those manufactured for the stroke characterization experiment, as there could be some difference in material quality despite the efforts made to standardize their manufacture. Alternatively, it could be because the increased speed caused momentum to have an effect on the weights, carrying them higher than in the previous experiment. In either case, the differences between the two experiments are slight, with a mean difference of 2%, indicating that this effect would be unlikely to affect performance to a notable degree in its intended use.

Table 4.44: Constructed 95% confidence interval for maximum achieved temperature values for the actuator running at the minimum possible contraction period. All values in °C.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower Bound</td>
<td>Upper Bound</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>83.20</td>
<td>0.92</td>
<td>0.25</td>
<td>82.66</td>
</tr>
<tr>
<td>2</td>
<td>82.97</td>
<td>1.14</td>
<td>0.27</td>
<td>82.37</td>
</tr>
<tr>
<td>3</td>
<td>83.40</td>
<td>0.76</td>
<td>0.19</td>
<td>82.98</td>
</tr>
</tbody>
</table>
4.4 Contraction Period Characterization

Table 4.45: Production of the actuator operating at the minimum possible contraction period, compared to the corresponding results from the stroke characterization experiment. All values expressed as a percentage.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>1.02</td>
<td>0.98</td>
</tr>
<tr>
<td>2</td>
<td>1.00</td>
<td>0.99</td>
</tr>
<tr>
<td>3</td>
<td>1.00</td>
<td>0.99</td>
</tr>
</tbody>
</table>

In addition to these favourable mean stroke production results, linearity testing found that they were exceedingly linear as well. As per the other characterization experiments, Microsoft Excel was used to fit a linear trend line to the confidence interval data (Seen in Table 4.46) to find to what degree the data followed a linear trend. The results of this are presented in Figure 4.46, which shows the constructed stroke confidence intervals for this experiment with the found trend line overlaid. It also shows the coefficient of correlation ($R^2$) value, which was found to be 0.9988. This value is very close to 1, which shows that the results were extremely close to linear behaviour, increasing the stroke per activation level at maximum speed by 3.92 mm according

![Figure 4.46: Confidence intervals of the percent contraction compared to activation level for the minimum contraction period experiment, with an overlaid linear regression to show linearity of the results. The formula for the linear regression and its $R^2$ value are presented as well.](image-url)
Table 4.46: Constructed 95% confidence interval for maximum stroke production values for the actuator running at the maximum possible speed. All values in mm.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower Bound</td>
<td>Upper Bound</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4.27</td>
<td>4.57</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8.43</td>
<td>8.71</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>12.12</td>
<td>12.38</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

to the slope of the relationship. This bodes well for the actuator, as three separate experiments have found that its isotonic stroke production capabilities scale linearly with the number of TCAs active, even at different actuation speeds. This enables easier, faster design based on limited data, which is appealing for future designs incorporating this device. After establishing the stroke production capabilities of the actuator at these elevated maximum actuation speeds, the times required to achieve maximum contraction were collated and confidence intervals were constructed. These contraction period confidence intervals can be found in Table 4.47. These are higher than the 2.0 s contraction period explored in the previous experiment, but are still much lower than the calculated average requirement of 3.57 s for neurorehabilitation, indicating that the actuator powered as it was is able to be used in wrist neurorehabilitation based on its contraction period, leaving room for the addition of a torque amplifying transmission component, as discussed in Section 4.3.

Table 4.47: Constructed 95% confidence interval for the time required to achieve maximum temperature in heating for the actuator running at the maximum possible speed. All values in seconds.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower Bound</td>
<td>Upper Bound</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2.34</td>
<td>2.52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2.32</td>
<td>2.55</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2.25</td>
<td>2.48</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As discussed in that section, if a gearing system was used to improve the torque output of the actuator, the actuation speed would decrease [135], meaning the contraction period would increase. Dividing the average required contraction period of 3.57 s (Section 2.7) by the contraction period of
the slowest activation level (2.43 s), one finds that the actuation speed can be reduced by a factor of 1.46 while still achieving the average required contraction period. Dividing the average required torque of 1.30 Nm by that factor, one finds that the actuator would still have to provide 0.89 Nm of torque in order to achieve average therapeutic requirements. Applying the same process as in Section 4.3, the additional number of pairs of TCAs required to meet that torque requirement while having a radial arm of 0.56 cm to let the actuator achieve the average range of motion required in therapy can be calculated as follows:

\[ F_{\text{linear,req}} = \frac{\tau_{\text{gear}}}{r_{\text{avg}}} \]

\[ F_{\text{linear,req}} = \frac{0.89}{0.0056} \]

\[ F_{\text{linear,req}} = 158.93 \text{ N} \]

\[ \delta F = F_{\text{linear,req}} - F_{\text{prod}} \]

\[ \delta F = 158.93 - 11.28 \]

\[ \delta F = 147.65 \text{ N} \]

\[ i_{\text{add}} = \frac{\delta F}{m_{\text{force}}} \]

\[ i_{\text{add}} = \frac{147.65}{1.85} \]

\[ i_{\text{add}} = 79.81 \]

where \( F_{\text{linear,req}} \) is the required linear force, \( \tau_{\text{gear}} \) is the output torque required from the actuator to achieve the average torque requirement with the use of a gearing system, \( r_{\text{avg}} \) is the radial arm required by the actuator to produce the average range of motion, \( \delta F \) is the difference in force between the required linear force and the force the actuator is capable of producing, \( i_{\text{add}} \) is the number of additional pairs of TCAs required, and \( m_{\text{force}} \) is the amount that the force changes per additional TCA, found from the linear regression in Section 4.2.5 to be 1.85 N per active pair. As can be seen, despite the use of a gear reduction, if the contraction period is to be kept
lower than or equal to the average therapeutic requirement, the actuator cannot produce both the average required torque and average required range of motion without adding a very large number of additional TCAs. Applying the same process, save with the maximum contraction period of 10.31 s, which results in a gearing ratio of 4.23 and a $\tau_{\text{gear}}$ of 0.31 Nm, the following number of additional pairs of TCAs are found:

$$F_{\text{linear,req}} = \frac{\tau_{\text{gear}}}{r_{\text{avg}}},$$

$$F_{\text{linear,req}} = \frac{0.31}{0.0056},$$

$$F_{\text{linear,req}} = 55.36 \text{ N},$$

$$\delta F = F_{\text{linear,req}} - F_{\text{prod}},$$

$$\delta F = 55.36 - 11.28,$$

$$\delta F = 44.08 \text{ N},$$

$$i_{\text{add}} = \frac{\delta F}{m_{\text{force}}},$$

$$i_{\text{add}} = \frac{44.08}{1.85},$$

$$i_{\text{add}} = 23.83.$$

This results in an additional 24 pairs of TCAs to achieve the maximum contraction period, average torque, and average range of motion required for wrist neurorehabilitation. This is a large increase in the number of TCAs required, and would result in a large, bulky actuator, which is undesirable for incorporation into a wearable rehabilitative mechatronic device. Without an increase in the actuation speed, stroke, or force produced by the actuator, even with the use of a gear reduction to amplify the torque the actuator can only achieve two of the design requirements with the same gear reduction and TCA pair configuration. This requires the use of either adjustable or separate designs for different exercise types, optimized for either torque or range of motion while meeting actuation speed requirements, as discussed in Section 4.3, or it requires the use of more power to
improve the actuation speed to allow for higher gearing ratios to be used.

### 4.4.12 Efficiency Analysis

From the ANOVA performed on the constant period experiment’s efficiency, seen in Table 4.48 it can be seen that there was no statistically significant difference between consecutive activation levels, while non-consecutive activation levels were statistically significantly different. This indicates a gradual decrease in efficiency as more activation levels are added for the constant period experiment, as adjacent levels are not different but levels one and three are. Conversely, for the minimum period experiment the ANOVA results in Table 4.49 indicates that there is no statistically significant difference between the efficiency of the different activation levels. For the results of the between-subjects factor testing, in ascending order of trend comparison the $p$ values achieved were 0.97 and 0.52 for the constant period experiment, and 0.78 and 0.77 for the minimum period experiment, while the purely between-subjects comparisons resulted in $p$ values of 0.57 for the constant period experiment and 0.28 for the minimum period experiment, indicating there was no statistical difference between different fabricated actuators at each activation level.

Table 4.48: Repeated measures one-way ANOVA analysis results for the efficiency of the actuator operating at a constant contraction period. All values had a Bonferroni correction applied, with a threshold of 0.05 signifying significance. Mean difference and standard error values presented in decimal percentage values.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference</th>
<th>Std. Error</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.003</td>
<td>0.001</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.006</td>
<td>0.001</td>
<td>$&lt;0.01$</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-0.003</td>
<td>0.001</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.003</td>
<td>0.001</td>
<td>0.07</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-0.006</td>
<td>0.001</td>
<td>$&lt;0.01$</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.003</td>
<td>0.001</td>
<td>0.07</td>
</tr>
</tbody>
</table>

While the between-subjects comparisons for both experiments were similar to results found in the efficiency analysis of the stroke characterization experiment in Section 4.1.7, neither of the within-subject comparisons were, as the analysis for that experiment found that each activation level had a statistically significantly different efficiency when operating in isotonic contraction. These discrepancies indicate not only that the value of the efficiency per activated level changes
Table 4.49: Repeated measures one-way ANOVA analysis results for the efficiency of the actuator operating at its minimum possible contraction period. All values had a Bonferroni correction applied, with a threshold of 0.05 signifying significance. Mean difference and standard error values presented in decimal percentage values.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference</th>
<th>Std. Error</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-0.001</td>
<td>0.002</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.000</td>
<td>0.002</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.001</td>
<td>0.002</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.001</td>
<td>0.002</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.000</td>
<td>0.002</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.001</td>
<td>0.002</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Based on contraction period, but also that the difference in efficiency per activated level changes as well. This phenomenon should be investigated in further work, because it could impact the power management design of devices that need to switch between different number of activated pairs of TCAs often. That being noted, looking at the mean values the difference between the maximum and minimum values for the constant period experiment is only 0.6 percentage points, and both are below 2% efficiency, which was the comparable value found in the literature.

Indeed, both experiments failed to break the 2% threshold, indicating that despite the increases in efficiency compared to the 1.1% maximum efficiency found in the stroke characterization experiment both failed to make improvements over examples found in the literature. Looking at the confidence intervals presented in Tables 4.50 and 4.51 for the constant and maximum period experiment efficiencies, respectively, only the actuator contracting over 2.0 s with one activated pair of TCAs was comparable with a confidence interval ranging between 1.6% and 2% efficiency and a mean of 1.8%. However, the maximum period experiment improved over the previous stroke characterization experiment, and the constant period experiment improved over that, indicating that increased actuation speeds do indeed increase efficiency for the designed actuator. While these speeds were comparable to or less efficient than examples in the literature, future iterations with higher actuation speeds could see that efficiency improve beyond that 2% threshold. To accomplish this, more durable power transmission components and larger power supplies should be used to facilitate greater actuation speeds.
4.5 Conclusions

From these experiments, the maximum stroke and force production of the actuator was found at each activation level, and confidence intervals of the possible values were constructed for both contraction and relaxation. Statistical analyses were performed, and it was determined that each result was statistically significantly different from one another in contraction, while in relaxation the production values of some of the activation levels were not statistically significantly different from one another. While this non-difference requires further investigation, the actuator worked as expected in contraction, increasing the stroke and force production to comparable or greater values than those found in the literature along its line of action, while improving over these found values when taking into account that each TCA was contracting at a 45° angle. Further work is required to determine how changing the pennation angle impacts these results, but there is a possibility that the collinear stroke and force could increase even further.

As well as finding the stroke and force production values at each activation level, it was found that the activation levels scaled linearly in their stroke and force production, increasing the stroke by 1.97 mm, or 1.97% and 188.9 g, or 1.85 N per active pair of TCAs. In addition to this, it

Table 4.50: Constructed 95% confidence interval for the efficiency of the actuator operating at a constant contraction period. All values presented as decimal percentage values.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower Bound</td>
<td>Upper Bound</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.015</td>
<td>0.005</td>
<td>0.001</td>
<td>0.013</td>
</tr>
<tr>
<td>2</td>
<td>0.012</td>
<td>0.004</td>
<td>0.001</td>
<td>0.010</td>
</tr>
<tr>
<td>3</td>
<td>0.018</td>
<td>0.004</td>
<td>0.001</td>
<td>0.016</td>
</tr>
</tbody>
</table>

Table 4.51: Constructed 95% confidence interval for the efficiency of the actuator operating at the minimum possible contraction period. All values as decimal percentage values.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower Bound</td>
<td>Upper Bound</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.012</td>
<td>0.004</td>
<td>0.001</td>
<td>0.010</td>
</tr>
<tr>
<td>2</td>
<td>0.013</td>
<td>0.005</td>
<td>0.001</td>
<td>0.011</td>
</tr>
<tr>
<td>3</td>
<td>0.013</td>
<td>0.005</td>
<td>0.001</td>
<td>0.016</td>
</tr>
</tbody>
</table>
was found that each fabricated actuator following the design worked statistically similar to one another, indicating the design is repeatable. These attributes are desirable, as it shows that the actuator can be modularly adjusted for different static loads or desired ranges of motion with the same load, allowing it to be curtailed or extended for different scenarios, and that each actuator made will behave similarly to what was found in these experiments. As well, the linearity of the actuator, combined with the characterization of the stroke and force of each activation level, allows the performance of the actuator to be predicted with certainty, which is desirable for future work incorporating this module into clinical devices.

In addition to finding confidence intervals to allow for the maximum stroke and force produced at each activation level, functions were created describing the relationship between temperature, activation level, and production using the temperature and production values collected in these experiments for when the actuator was in both contraction and relaxation. These allow the stroke and force produced by an actuator constructed in a similar manner to be estimated using the temperature and activation level; or inversely, allow the number of active pairs and the temperature required to contract them sufficiently to be estimated from a required force or stroke value. This is desirable as it can help assist in the design of purpose-built woven bi-pennate actuation devices, should the operating stroke or force range be known prior to the design. In particular, this is beneficial in the field of therapeutic robotics as it allows for the tailoring of designs to individual therapy needs, which may differ based on patient physiology and range of motion.

After finding the stroke and force production at each activation level, characterizing models for the production values based on temperature and number of active pairs, and examining the linearity of the production values compared to activation level, the results were compared to clinical values to determine the viability and requirements for implementing the actuator in therapeutic scenarios. It was found that the actuator could meet average wrist rehabilitation range of motion requirements with a radial arm of 0.56 cm, but could not be made to meet average torque requirements while meeting actuation speed requirements, even with the use of a gear reduction to boost torque output. This requires the use of separate devices to target different exercise scenarios, optimized for either range of motion or torque to meet the needs of that specific scenario. If a single, universal device is desired, it would have to incorporate an adjustable radial arm, or have a higher power
4.5 Conclusions

draw to increase the actuation speed to allow for higher gear reductions to be used. Comparing radial arms to the literature, it was found that a device with a radial arm of 0.56 cm was smaller than examples that could be found by an order of magnitude, while the actuator was able to be easily scaled to meet torque output and size constraint requirements, especially with the use of a gear reduction. Due to the availability and inexpensive nature of yarn and thread, and the linear nature of the design, it could be concluded that while not able to meet both simultaneously without additional components, the actuator could be scaled to meet wrist neurorehabilitation requirements for torque and range of motion while being smaller than or comparable to other soft actuator devices while still meeting contraction period requirements.

In addition to the stroke and force production of the actuator, the contraction periods the actuator was capable of were explored as well, looking at the maximum stroke, temperature, and required heating time for the actuator running at a constant contraction period of 2.0 s and at the minimum contraction period with the power transmission components used. Similar to previous results, it was found that the stroke produced was statistically significantly different at each activation level, while the maximum temperature of the actuator and the time required to reach it were not. The different actuators were also found to perform statistically similar in all regards as well, indicating consistency between different fabricated actuators again. With the given power, the actuator was unable to achieve previously recorded maximum stroke values at a contraction period of 2.0 s; however, it could with contraction periods of between 2.38 and 2.43 s. The actuator had a similar or slightly better range of motion for all activation levels when compared to the stroke characterization experiment, which was comparable to similar actuators found in the literature. At its peak, it was more than able to satisfy the average required actuation speed for wrist neurorehabilitation, eclipsing the found contraction period value of 3.57 s by enough to allow for a torque amplifying transmission system, such as a gear reduction, to be implemented while still allowing the actuator to operate at the required speed. However, this gear reduction cannot be made large enough without increasing the power supplied to the actuator to satisfy both therapeutic range of motion and torque requirements without sacrificing its ability to meet the required contraction period.

While able to nonconcurrently meet clinical requirements, other areas where the actuators
were found lacking were in the introduction of slack in after contraction, efficiency, and cooling speed. As found from the stroke and force characterization experiments, the slack introduced after contraction did not scale linearly or remain constant with the number of activated TCAs. This requires further work to understand its cause, but poses a problem as slack introduced in the TCAs would result in shifting of the range of motion after each repetition, which would affect therapeutic outcomes. While solvable with a tautening device, its less-than-predictable nature makes it more difficult to account for its behaviour without some kind of user adjustment, as the behaviour is unknown when the number of activated pairs of TCAs is changed. In addition to this, the peak efficiency was found to be 1.8%, meaning that the actuator underperformed compared to examples in the literature, which reported efficiencies of 2% [60]. The efficiency went down as the heating time increased, which may mean that a cause for this lack of efficiency is the prolonging of the heating process, allowing more heat to be lost through convention or conduction into the yarn than otherwise would be if the TCAs were heated swiftly. This hypothesis is supported by the efficiency increasing as the contraction speed increased, but further work is required to see if this could be improved further by increasing the actuation speed even more. If the yarn is being heated, it could also be a source of the slow cooling times, as the latent heat from the yarn would slow the cooling of the TCAs when they are required to relax. Regardless, to achieve a relaxation period as low as that found for contraction, an active cooling system is required, as it is unlikely that passive cooling would be sufficient. While the actuator was able to perform adequately in contraction, if these deficiencies could be addressed in future work the viability of the actuator for incorporation into wearable rehabilitative devices would only increase.
Chapter 5

Force Control Implementation

This chapter presents the creation and testing of two force control schemes constructed using the results of the force characterization of the actuator. It details the development of the control laws used, the experimental methods used to test them, the statistical methods used to analyze the results, and the conclusions that can be drawn from the comparison of these two control schemes, recommending one based on superior performance in regards to its applicability towards wrist neurorehabilitation.

5.1 Purpose

The purpose of this experiment was to design an extension of the most popular form of predictive control found in the literature from one TCA muscle fibre to the designed woven bi-pennate actuator. This was done to determine whether the same benefits of reduced overshoot and steady state error could be achieved with a bi-pennate, rather than individual, TCA based actuator, while minimizing any impact it would have on response time. In addition, it was designed to be purely force-feedback based, to minimize the number of sensors required to control the actuator, reducing the cost and decreasing the complexity of implementing the actuator into wearable therapeutic devices. To test how the controller performed, a baseline controller was constructed and compared to the performance of the new control design based on five metrics, which represented the response time, transient response, and steady state behaviour of the controlled response. This provided a way of gauging how the new control system worked relative to standard methods,
and its behaviour helped to determine the viability of the new control method in controlling a bi-pennate TCA actuator embedded in a fabric mesh. It also provided a way of determining the efficacy of the characterization and control law design process in its ability to properly capture the relationship between activation level, temperature, and force produced by the actuator and predict its performance. Based on these comparisons, the new controller, and the methods used to construct it, were assessed on their ability to improve control of the actuator compared to a traditional control method.

5.2 Control Law Development

As mentioned in Chapter 2, in addition to characterizing the performance of the actuator it is imperative to determine a method of controlling it, in order for it to be applicable to rehabilitative devices. Control systems serve to optimize the response of actuation systems, making them respond as swiftly as possible to stimuli, limiting their overshoot when trying to achieve a specified set point, and limiting the oscillations and steady state error after having achieved the specified value. They do this by monitoring the response of the system via sensors, and altering the input to the system according to mathematical principles in order to achieve the desired response. In the context of the designed actuation system, purely force feedback control would be desired, both to ensure maximum performance and safety when manipulating the wrist during therapeutic exercises, and to limit the number of sensors required to operate the system, reducing cost and complexity when implementing it into wearable devices. This is done by setting a desired force setpoint for the actuator to achieve, and constantly polling a force feedback sensor to monitor its progress. This information is then interpreted by a microcontroller, which calculates and sets the power input to the actuator according to the difference between the current and desired force output, i.e. the error signal. Because of the use of a constant power source in the system in previous experiments, pulse width modulation (PWM) was determined to be the method by which the power would be changed. This method of power modulation works by changing the percent of time that the power is supplied, which in practice mimics having just that percent of power being supplied to the system. If done in a discrete repeating time period, expressing that percent of power as a percent of a repeating fixed-time cycle, it more closely approximates the effects of delivering just
that percent of the power constantly. As the cycle time decreases and approaches an infinitely small value, it more closely approximates the desired power value [138].

In order to design a controller for the actuation system, the relationship between the force production, power input, and time had to be resolved. The relationship between force and temperature had already been found via the regression process detailed in Section 4.2, meaning that finding the actuator’s relationship between power input and temperature, converting the two relationships into the frequency domain, and multiplying them together to get the final relationship between power input and force production was required before the control law could be developed. To find the relationship between power supplied and temperature, it was assumed that sources of thermal radiation had a negligible impact on the heating of the actuator, which, when substituted and simplified from the general heating equation given in [139], gives the following heating equation:

$$T = (T_i - T_{amb}) e^{-t/\tau_C} + PR_T \left( 1 - e^{-t/\tau_C} \right) + T_{amb} \quad (5.1)$$

where $T$ is the temperature in degrees Celsius, $T_i$ is the initial temperature, $T_{amb}$ is the ambient temperature, $t$ is the time in seconds since the beginning of the application of power, $P$ is the power supply, $R_T$ is the thermal resistance of the actuator, and $\tau_C$ is the thermal time constant of the system. The thermal resistance for each activation level of the actuator was calculated by dividing the difference between the maximum and minimum temperature of the actuator in the heating phase of its trial by its average power consumption [98], repeating for each trial of each activation level before being averaged across each level to get the average thermal resistance of the actuator at that level. The time constants for each activation level were found as the time it took for the actuator to reach 63% of the difference between the initial and final temperature of the actuator in that trial, before being averaged across each activation level to get the average time constant of the actuator for that level. The code used to do this can be found in Appendix B.2.1 In finding both sets of constant values, the datasets used were those collected from the Force Characterization experiment in Section 5.6, with the results presented in Table 5.1.
Table 5.1: Thermal resistances and time constants of the system for each activation level, calculated from experimental data.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Thermal Resistance $R_T$ (°C/W)</th>
<th>Time Constant $\tau_c$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.26</td>
<td>7.48</td>
</tr>
<tr>
<td>2</td>
<td>1.47</td>
<td>13.14</td>
</tr>
<tr>
<td>3</td>
<td>1.26</td>
<td>17.07</td>
</tr>
<tr>
<td>4</td>
<td>1.15</td>
<td>22.32</td>
</tr>
<tr>
<td>5</td>
<td>0.89</td>
<td>31.25</td>
</tr>
<tr>
<td>6</td>
<td>0.82</td>
<td>38.00</td>
</tr>
</tbody>
</table>

To complete the constant values used in Equation 5.1, the ambient temperature was assumed to be 25°, as that was a common reading from the wall thermostat (the experiments took place in the summer), the initial temperature of the system was assumed to be the same, and the power input was assumed to be 80 W per activated pair, giving a time dependent relationship between temperature and power.

After having clarified time-domain relationships between force and temperature and temperature and power for the actuator at each activation level, they had to be converted to the frequency domain so their control laws could be developed. Typically this would be done by taking the Laplace transform of each relationship; however, this could not be done because the relationships found were not linear-time-invariant [140], which precludes the use of the Laplace transform. Looking to the literature [100, 102, 107, 141, 142], it was found that the System Identification Toolbox from MathWorks was widely considered adequate for approximating such systems, and was used to fit frequency domain models to the found relationships. The code used for this procedure can be found in Appendix B.2.2.

For the temperature–force relationships, the system identification was done by creating an input vector of temperature values, spanning from the ambient temperature (25°) to 85°, and an output vector of their calculated force values for each activation level, using the regressed time-domain relationships from Section 4.2.6. The number of activation levels was limited to three, as this was minimum number of activation levels needed to reaffirm the linear behaviour found in previous experiments, as mentioned in Section 4.4. The sampling rate was set to 0.11 seconds, as that matched that of the force characterization experiment. The two vectors were split into
5.2 Control Law Development

learning and validation sets, and the number of poles and zeros were increased iteratively until the fit of each found model was maximized. This was repeated for each of the activation levels one to three, giving the transfer functions found in Table 5.2, modelling the relationship between temperature and force for the system, as well as their level of fit.

The levels of fit were then compared to the literature to determine if they were adequate. Fits of 69.40 [100], 74.00 [141], 69.44 [142], 69.62 [142], 69.37 [142], and 65.38 [142] were seen in the literature, so the values found were considered adequately similar. A similar methodology was used to find the transfer functions for the power–temperature relationships. A simulated step input was created, such that the power was held to 80 W per activated level until the temperature, calculated using Equation 5.1 with values from Table 5.1, was able to increase from 25° to 85°, and then set to zero until the temperature was able to cool to 30°, creating a simulated version of the time–temperature curve from the minimum contraction period experiment from Section 4.4. The power vector was used as an input, the temperature vector was used as an output, the sampling rate was set to 0.11 seconds, as in the force characterization experiment, and the model was regressed from these data, splitting the datasets into learning and validation sets and iteratively increasing the number of poles and zeros in the modelled function until the fit of each found model was maximized. This was repeated for each of the activation levels one to three, giving the transfer functions in Table 5.3 modelling the relationship between power and temperature of the system, as well as their level of fit.

Table 5.2: Temperature-force transfer functions identified from the regressed time-domain relationships, by activation level. The level of fit denotes how well the transfer function relates to the input data on a linear scale from 0–100, with 100 being best possible fit and 0 being worst possible fit.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Transfer Function $G_{TF}(s)$</th>
<th>Fit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\frac{311s+367.90}{s^2+130.3s+134.90}$</td>
<td>68.41</td>
</tr>
<tr>
<td>2</td>
<td>$\frac{-2220s^2-254.7s+6402}{s^4+64.23s^3+494.3s^2+174.3s+1191}$</td>
<td>73.89</td>
</tr>
<tr>
<td>3</td>
<td>$\frac{1835s^2+1932s+4308}{s^4+254.6s^2+245s+599}$</td>
<td>68.31</td>
</tr>
</tbody>
</table>
Table 5.3: Power-temperature transfer functions identified from the simulated values, by activation level. The level of fit denotes how well the transfer function relates to the input data on a linear scale from 0–100, with 100 being the best possible fit and 0 being the worst.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Transfer Function $G_{PT}(s)$</th>
<th>Fit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\frac{0.1966s^2-0.3707s+1.729}{s^3+0.2338s^2+5.62s+1.309}$</td>
<td>71.87</td>
</tr>
<tr>
<td>2</td>
<td>$\frac{0.1328s+0.2193}{s^2+1.7s+0.1021}$</td>
<td>89.04</td>
</tr>
<tr>
<td>3</td>
<td>$\frac{0.05251s^2-0.02736s+0.4945}{s^3+0.05077s^2+6.122s+0.3108}$</td>
<td>97.01</td>
</tr>
</tbody>
</table>

Again, the level of fit was appraised to determine adequacy. As these transfer functions had a higher degree of fit than the identified temperature–force transfer functions, these too were determined to have an acceptable degree of fit. After having found the frequency-domain transfer function of the actuator operating at each activation level, it was time to implement the force control law by designing controllers. As found in the literature review in Chapter 2, two degree of freedom controllers incorporating feedback and feedforward elements (referred to in this thesis as FF controllers) were found to achieve high degrees of accuracy when following a reference signal [95], limited overshoot [100], decreased recovery time [8], and limited oscillations about a target force setpoint [102], decreasing overall steady state error. Many two degree of freedom controllers were able to improve performance on single TCAs [8, 12, 95, 100, 102, 104], but none were found that had been tested on woven bi-pennate actuating modules, and it was determined that it would be desirable to see if these same benefits could be seen when extending that control methodology to the designed woven actuator. In addition to the literature-recommended FF controllers, baseline controllers were designed for comparison purposes. Proportional-Integral-Derivative (PID) controllers are widely used due to their simplicity in implementation and efficacy, and can be considered a sort of standard or baseline controller due to their ubiquity. They operate in a trifold manner: a proportional component that multiplies the error signal by a constant value, an integral component that alters the control signal based on accumulated error, and a derivative component, that alters the control signal based on how the error signal is changing.
between consecutive time points. By altering the weighting of each component, the controller can be tuned to achieve the desired performance in the system. A block diagram of a PID control loop can be found in Figure 5.1.

For the FF controllers, it was decided to follow a similar design to ones found in the literature that had similar applications: a two degree of freedom controller incorporating both force feedback and feedforward components for TCAs. This type of controller incorporates both a PID controller and a feedforward controller, hence the two degrees of freedom [100]. The PID controller operates as described above, with an additional input fed into the system after the PID controller to alter the control signal. This input is an inverted model of the actuation system, which predicts the state of the system at a given point in time, in conjunction with a second-order controller to modulate the effect it has on the control signal. This way, the PID control signal is merged with the predicted state of the actuator, theoretically giving a more accurate control signal. As well, the same second-order controller is applied to the input signal, to curtail the weighting of the PID component so as not to let it dominate the control signal.

The PID controllers were tuned first, as it would be used in the construction of the FF controller. This was done by implementing the transfer functions designed previously into block diagrams in Simulink with a step input to give a target setpoint in grams force equivalent, with each being set according to the mean force production of the activation level found in the force characterization experiment. An example of this setup can be seen in Figure 5.1. After doing this, the auto-tune function in Simulink was used to find a starting point for the coefficients $K_P$ (the proportional coefficient), $K_I$ (the integral coefficient), and $K_D$ (the derivative coefficient) for each PID controller. Each coefficient was then tuned iteratively to minimize overshoot, oscillations, and settling error, which were determined to be the most important attributes of the system because those could possibly injure the user if they exceeded the target force setpoints. The optimal coefficients found for the controllers can be seen in Table 5.4.

After tuning the PID controllers, the FF controllers were constructed and tuned. The PID controllers from before were again implemented, with a second-order controller put before the error signal summation and an inverted model of the system with an identical second-order controller feeding into the control signal included as well, as described previously. An example of this
setup can be seen in Figure 5.2. Transfer functions represent the rational relationship between an input and output, so to invert them, or show the relationship between output and input, one simply swaps the numerator and denominator of the transfer function. However, in the case of the derived functions, inverting them this way would result in the numerator being of higher order than the denominator, also known as an improper transfer function. These are unstable under most conditions, and cannot be handled in the Simulink simulation software.

Figure 5.1: The three PID control loops implemented in Simulink, used to tune the coefficients $K_P$, $K_I$, and $K_D$ of the PID controllers for each activation level. Transfer functions were calculated as per the steps outlined above, implemented in the code available in Appendix B.2.2.
Table 5.4: Tuned coefficients for the designed PID controller, by activation level.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>$K_P$</th>
<th>$K_I$</th>
<th>$K_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30.0</td>
<td>0.05</td>
<td>1.50</td>
</tr>
<tr>
<td>2</td>
<td>3.5</td>
<td>0.10</td>
<td>0.90</td>
</tr>
<tr>
<td>3</td>
<td>4.0</td>
<td>0.10</td>
<td>0.06</td>
</tr>
</tbody>
</table>

In order to avoid having these improper transfer functions in the system, as would happen from normally inversing a transfer function, an alternate representation for each was found using partial fraction expansion [143, 144] and implemented as proper systems, seen in Table 5.5. The procedure for this is described below.

Partial fraction expansion works by representing a rational function as the sum of simpler rational functions and a polynomial function [145, 146], taking the following form:

$$ G(s) = \sum_{k=0}^{\infty} \left( \frac{\text{Res}(p_k)}{s - p_k} \right) + H(s), $$

where $G(s)$ is the original transfer function, $p_k$ is the $k^{th}$ pole of the transfer function (a pole is a root of the function comprising the denominator of the transfer function), $\text{Res}(p_k)$ is the residue of the function evaluated at $p_k$, and $H(s)$ is a polynomial function of $s$. Going through each component in turn, residues are calculated as follows [145]:

$$ \text{Res}(p_k) = \lim_{s \to p_k} (s - p_k)G(s). $$

This gives a set of scalar values equal in length to the number of poles of the original transfer function $G(s)$. These serve as the numerators for a set of first order rational functions, which, when summed together with $H(s)$ as seen above, give the original transfer function. $H(s)$, as mentioned previously, is a polynomial function of $s$ which takes the following general form:

$$ H(s) = a_0s^n + a_1s^{n-1} + a_2s^{n-2} + \ldots + a_{n-1}s + a_n, $$

where $a_i$ is a scalar coefficient and $n$ is the maximum degree of the polynomial. This polynomial serves to make the equivalency equal to the original transfer function $G(s)$. 
Table 5.5: Calculated representations of the inverted system transfer functions, by activation level. The subscript \( PT \) denotes the transfer function relating power and temperature, and the subscript \( TF \) denotes the transfer function relating temperature and force.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Relationship</th>
<th>Transfer Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( G_{PT}^{-1}(s) )</td>
<td>(-0.3054s + 67.23 \over s^2 + 1.886s + 8.795 + 5.086s - 8.402 )</td>
</tr>
<tr>
<td></td>
<td>( G_{TF}^{-1}(s) )</td>
<td>(-0.05744 \over s + 1.183 + 0.003215s + 0.4152 )</td>
</tr>
<tr>
<td>2</td>
<td>( G_{PT}^{-1}(s) )</td>
<td>(0.1639 \over s + 1.651 + 7.53s + 0.3663 )</td>
</tr>
<tr>
<td></td>
<td>( G_{TF}^{-1}(s) )</td>
<td>(0.01977s - 0.1107 \over s^2 - 0.1107s + 2.884 + 0.0004505s^2 + 0.02898s + 0.2244 )</td>
</tr>
<tr>
<td>3</td>
<td>( G_{PT}^{-1}(s) )</td>
<td>(-57.08s - 96.63 \over s^2 - 0.521s + 9.417 + 19.04s + 10.89 )</td>
</tr>
<tr>
<td></td>
<td>( G_{TF}^{-1}(s) )</td>
<td>(-0.01324s + 0.002044 \over s^2 + 1.053s + 2.348 + 0.000545 + 0.1382 )</td>
</tr>
</tbody>
</table>

After applying these inverted transfer functions to the control scheme seen in Figures 5.2 and 5.3, the FF controllers were tuned so that the response of the system would have minimal overshoot, oscillations, and settling error, as done before with the PID tuning. These controllers took the form shown in Equation 5.2, with \( K_E \) being tuned to get the desired system performance. This tuning was done in an iterative process, by setting \( K_E \) to zero, then first increasing then decreasing the value and taking note of which response resulted in better performance. More steps were taken centred around the better performing value, repeating until the difference between steps were negligible. Initial steps were taken with a value of one, then reduced in size once it became clearer that the value of \( K_E \) was getting closer to its optimal value, based on the amount of change between responses for different iterations of the controller. The optimal coefficients found for the controllers can be seen in Table 5.6.

\[
\frac{K_E^2}{(s + K_E)^2} \tag{5.2}
\]
Figure 5.2: The three FF control loops implemented in Simulink, used to tune the coefficients $K_E$ of the FF controllers for each activation level. Transfer functions were calculated as per the system identification steps outlined above, the results of which can be seen in Tables 5.2 and 5.3 for the regular transfer functions and Table 5.5 for their inverses, and were implemented in the code available in Appendix C.4.2.

Figure 5.3: An example of an inverted subsystem block, used in the construction of the FF control loops. Inverted transfer functions were calculated as per the steps outlined in the system identification process above, the results of which can be seen in Table 5.5. These were implemented in the code available in Appendix C.4.2.

Table 5.6: Tuned feedforward controller coefficient, by activation level.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>$K_E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.0</td>
</tr>
<tr>
<td>2</td>
<td>3.5</td>
</tr>
<tr>
<td>3</td>
<td>6.0</td>
</tr>
</tbody>
</table>
5.3 Sample Size Calculations

After designing the two control systems, the number of samples required for statistical significance of the experiment were calculated. Equation 3.1 was used in conjunction with the results from Section 4.2.4, using the mean and standard deviation values for each activation level for the calculations. A $Z_{\alpha}$ value of 1.96 and a $Z_{\beta}$ value of 0.84 were used once more, to represent a 95% certainty of an absence of Type I errors and an 80% certainty of an absence of Type II errors respectively, and the minimum discernible effect size was set to 5% of the mean produced force, which was done by multiplying the mean force produced at each activation level by 0.05. The results of these calculations can be seen in Table 5.7.

From these calculations, the largest number of required samples for the force control experiment was found to be eight. This was significantly less than the lowest number of sample sizes used previously, and was deemed too small to allow for appreciable room for error or outliers. Thus, it was decided that the previous smallest calculated sample size of 15, used in the minimum contraction period experiment, would be used. This number of trials would be used for each activation level tested for each control scheme, and would be divided into groupings of five trials across three actuators. Similar to the contraction period experiment, it was decided that the actuator investigated would be comprised of three pairs of TCAs, as that experiment showed that the linearity between activation levels observed in the stroke and force characterization experiments was also present at smaller numbers of activation levels, allowing for results to be scaled.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Calculated Sample Size n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.32</td>
</tr>
<tr>
<td>2</td>
<td>5.79</td>
</tr>
<tr>
<td>3</td>
<td>6.10</td>
</tr>
<tr>
<td>4</td>
<td>7.88</td>
</tr>
<tr>
<td>5</td>
<td>4.42</td>
</tr>
<tr>
<td>6</td>
<td>6.31</td>
</tr>
</tbody>
</table>
5.4 Experimental Methods

Having designed the control laws, implemented them in Simulink, and determined the required number of trial repetitions to establish statistical significance, the control schemes were then implemented and tested in physical actuators. To create the required three actuators of three pairs of TCAs, 18 TCAs were fabricated and trained as per the procedure in Sections 3.1 and 3.2. The same power control PCB and 3D printed components as in Chapters 3 and 4 were used, and the Rigol DP811 and Agilent E3620A were attached in parallel as in Section 4.4 to provide required increased power levels. The same wooden structure used in Section 4.2 was used once more to provide a rigid horizontal support structure.

Having assembled the components required for the construction of the actuators, they could be made and tested. After clamping the wooden support structure to the lab bench, the nylon central tendon was attached to the load cell attach ring and the nylon side tendon holders were clamped to the wooden structure, 141 mm apart, measured using a digital Vernier caliper, and such that their leading pins were 141 mm, measured using the digital Vernier caliper, from the lead pin of the central tendon when held horizontally. Six TCAs were then slid over the first three sets of pins of the central and side tendons and affixed with the power delivery wires and pin caps, creating a bipennate structure on the first three sets of pins, as seen in Figure 5.4. The load cell structure and lateral holsters were then adjusted until there was a slight amount of tension in the TCAs, with the TCAs maintaining a $45^\circ$ pennation angle with the tendons. Unlike previous experiments, thermocouples were not attached, as temperature control was not being investigated and removing their communication delays improved system polling time, allowing for finer PWM control. After preliminary testing, this polling time was improved to approximately 0.03 seconds. After the TCAs were in place and the tendons properly configured, the actuator was woven using the same 100% nylon yarn as in previous experiments (Part No. 1005019 from anniescatalog.com). The weaving process used was the same as in previous experiments, using a manual over-under technique with the TCAs as the warp and compressing the woven weft with every pass to make the weave as tight as possible. One side was done at a time, continuing until the entirety of the TCA was covered up to the clamped ends attached to the side tendons. A half-woven actuator can be seen in Figure 5.4.
Once the actuator was woven, the load cell structure was adjusted to reaffirm that the $45^\circ$ pennation angle for the TCAs was still maintained, after which the actuator was considered fully fabricated and ready for the control scheme testing. A finished actuator can be seen in Figure 5.5, along with examples of its weave density and flexibility. In order to test the designed control schemes, they were first implemented in Arduino C++ code, which can be seen in Appendices C.4.1 and C.4.2 for the PID and FF controllers, respectively. This required translation of the control systems into workable code, which was done using standard techniques for the PID components, and required taking the inverse Laplace of the feedforward component to translate it into the time domain, which was done in MATLAB before being implemented in the microcontroller code. The code for this can be seen in Appendix B.2.2. During the experiment’s run time, a pair of TCAs were chosen randomly and heated via PWM (100 ms cycle time) with a maximum power output of 80 W per pair of TCAs (20 V at 4 A per pair, or 2 A each), supplied by a programmable DC power supply, while the system took continuous force readings, with the load cell being tared on trial start. This was done until it reached a specified force setpoint based on the number of pairs active, seen in Table 5.8, which were also used in simulating the systems.

This was controlled by the PID control code based on the designed control system, and
5.4 Experimental Methods

Figure 5.5: Completed experimental setup presented in two angles, in addition to two views of a disassembled actuator illustrating its weave-density and flexibility.

maintained until five seconds had elapsed after achieving the desired setpoint value. Each setpoint was chosen to be a rounded approximation of the maximum force produced by the actuator at Activation Levels 1–3 in the force characterization experiment in Section 4.2, so that the controlled performance of the actuator operating at maximum capacity at each activation level could be examined. The actuator was then allowed to cool until the force readings again reached zero,
when another random pair of TCAs would be selected and the process would begin again. After five trials of this had been completed, the PID control code was swapped out for the FF control code, and the same process was used to test the performance of that controller. After having completed these ten trials, five for each type of controller, another five trials were completed on the same actuator for each type of controller for two and three active pairs, increasing the power supplied to maintain 80 W available per pair, and increasing the force setpoint to the relevant values in Table 5.8. After five trials for each type of controller at each activation level (30 trials in total) were completed, the actuator was disassembled and the process was begun anew with another actuator, constructed as per the previous instructions and tested in the same manner. Once that actuator’s testing had been completed, the full process was done once more, to give three actuator’s worth of data, or 15 trials per activated pair for each type of controller. The code for controlling the experiments can be found in Appendices C.4.1 and C.4.2 for the PID and FF controller, respectively.

Table 5.8: Target force setpoints used throughout experimentation and simulation, based on the results found in Section 4.2.4. All force values given in grams force equivalent.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Force Setpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>400</td>
</tr>
<tr>
<td>3</td>
<td>550</td>
</tr>
</tbody>
</table>

5.5 Analytical Methods

In order to compare the force responses of the two controllers at each target setpoint, different time domain characteristics had to be calculated in order to quantify the difference in performance between the two controllers. The five characteristics selected were the time constant, rise time, percent overshoot, first peak time, and steady state value, as these gave a sense of the response time, overshoot of the target setpoint, and error when maintaining a specified force, all of which are of interest when investigating a control system’s performance in a rehabilitative context.

The time constant was found as the time it took the force response to achieve 63% of the force setpoint, the rise time was found as the time it took the force response to go from 10% of the
force setpoint to 90%, the percent overshoot was taken as the difference between the maximum
force production of the actuator in a trial and its setpoint, expressed as a percent of the setpoint,
the first peak time was found as the time at which the first local maximum occurred in the force
setpoint, and the steady state value was taken as the average of all of the force values after the
response had initially peaked and returned to within 2% of the force setpoint. If there were no
peaks present, as in the case of some of the simulations, the first peak time was taken as the point
at which the system response first came within 2% of the force setpoint. The code for doing this
can be found in Appendices B.2.3 and B.2.4 for the experimental results and simulation results,
respectively.

These five characteristics were calculated for each trial for each method of control, as well
as for their respective simulations. After having done this, SPSS was used to perform one-way
ANOVA tests, as seen in Section 4.1.3, on the results to determine the relationship between the
characteristics calculated for each activation level. The within-subject factor was chosen to be the
activation level, as in previous experiments, to determine whether there were statistically significant
differences between the calculated characteristics across different activation levels for a specific
control scheme, and the between-subject factors were chosen to be the actuator number and control
scheme used. These were chosen to see whether the results found were statistically significantly
different between fabricated actuator modules, and to see whether there were statistically significant
differences between the two control methods. After completing the within-subjects ANOVA testing
for each of the control characteristics, for reasons expressed in Section 4.4.2, due to the non-
normality of time data, Kruskal-Wallis analyses were also performed on the first peak time, time
constant, and rise time. Following the procedure in Section 4.4.2, the analyses were performed
after grouping by actuator number and controller type.

Confidence intervals were also constructed for each of the control characteristics using SPSS,
as in Section 4.1.3. After creating the confidence intervals, the average steady state error of the
control systems at each activation level were found by finding the difference between the settling
value and target setpoint, then dividing that by the target setpoint, as follows:

\[ \%_{Diff} = \frac{|F_{settle} - F_{target}|}{F_{target}}. \]
where $\%_{Diff}$ is the steady state error expressed as a percentage, $F_{\text{settle}}$ is the average settling value of the controller at that activation level in grams force equivalent, and $F_{\text{target}}$ is the target force setpoint in grams force equivalent. As well, the difference between the average characteristic values and their respective values found through simulation were calculated using the same method, substituting $F_{\text{settle}}$ with the experimentally found mean value, and $F_{\text{target}}$ with the simulated value.

5.6 Results

After the data had been collected, the data were processed and made usable for analysis. The first segment of data consisting of the time it took to reach the force setpoint, plus an additional five seconds, was separated from the rest of the data, as that was the controlled portion of each trial. These data can be seen presented alongside their respective simulation results in Figures 5.6 and 5.7 for the experimental and simulated results respectively. Figure 5.8 gives individual trials’ examples of the controlled force response for each controller. These were then further separated into the transient response of the system, seen in Figures 5.9 and 5.10, and the oscillatory response of the system, seen in Figures 5.11 and 5.12. After separating the data in this manner, the time constant, rise time, first peak time, overshoot, and settling value were calculated from the force responses.
5.6 Results

(a) PID controller, one active pair.

(b) FF controller, one active pair.

(c) PID controller, two active pairs.

(d) FF controller, two active pairs.

(e) PID controller, three active pairs.

(f) FF controller, three active pairs.

Figure 5.6: Simulated step response of each designed controller. The blue signals are the simulated force output of the system, while the yellow signals are the target force setpoints.
5.6 Results

(a) PID controller, one active pair.
(b) FF controller, one active pair.

(c) PID controller, two active pairs.
(d) FF controller, two active pairs.

(e) PID controller, three active pairs.
(f) FF controller, three active pairs.

Figure 5.7: Experimental step response of each designed controller for the full duration of each trial, with each trial shown. The blue signals are the isometric force output of the system, while the yellow signals are the target force setpoint.
5.6 Results

(a) PID controller, one active pair.

(b) FF controller, one active pair.

(c) PID controller, two active pairs.

(d) FF controller, two active pairs.

(e) PID controller, three active pairs.

(f) FF controller, three active pairs.

Figure 5.8: Example experimental step responses of each designed controller for the full duration of a randomly selected trial. The blue signals are the isometric force output of the system, while the yellow signals are the target force setpoint.
5.6 Results

(a) PID controller, one active pair.

(b) FF controller, one active pair.

(c) PID controller, two active pairs.

(d) FF controller, two active pairs.

(e) PID controller, three active pairs.

(f) FF controller, three active pairs.

Figure 5.9: Experimental step responses of each designed controller for the transient phase of each trial, with each trial shown. The blue signals are the isometric force output of the system, while the yellow signals are the target force setpoint.
5.6 Results

(a) PID controller, one active pair.

(b) FF controller, one active pair.

(c) PID controller, two active pairs.

(d) FF controller, two active pairs.

(e) PID controller, three active pairs.

(f) FF controller, three active pairs.

Figure 5.10: Example experimental step responses of each designed controller for the transient phase of a randomly selected trial. The blue signals are the isometric force output of the system, while the yellow signals are the target force setpoint.
5.6 Results

(a) PID controller, one active pair.  
(b) FF controller, one active pair.
(c) PID controller, two active pairs.  
(d) FF controller, two active pairs.
(e) PID controller, three active pairs.  
(f) FF controller, three active pairs.

Figure 5.11: Experimental step responses of each designed controller for the oscillatory phase of each trial, with each trial shown. The blue signals are the isometric force output of the system, while the yellow signals are the target force setpoint.
5.6 Results

(a) PID controller, one active pair.

(b) FF controller, one active pair.

(c) PID controller, two active pairs.

(d) FF controller, two active pairs.

(e) PID controller, three active pairs.

(f) FF controller, three active pairs.

Figure 5.12: Example experimental step responses of each designed controller for the oscillatory phase of a randomly selected trial. The blue signals are the isometric force output of the system, while the yellow signals are the target force setpoint.
5.7 Analysis

From the results of the within-subjects factor testing from the repeated measures one-way ANOVA analyses performed, it can be seen that for the PID controller implementation (Tables 5.9–5.13) the first peak time and percent overshoot cannot have the null hypothesis rejected \( (p > 0.05) \) for any of the activation levels compared to each other, while the time constant had statistically different values for the second activation level compared to the others \( (p < 0.01) \), the rise time and settling values were statistically significantly different for all activation levels \( (p < 0.01) \). For the within-subjects factor testing for the FF controller implementation (Tables 5.14–5.18), it was found that the first peak time, time constant, and overshoot values were not statistically significantly different \( (p > 0.05) \) between activation levels, while activation level three was statistically different from the others for the rise time \( (p < 0.01) \), and all activation levels were statistically significantly different for the settling values \( (p < 0.01) \). The between-subject factors ANOVA testing found that there were no statistically significant differences between different fabricated actuators for either of the controllers, with \( p \) values greater than 0.05 for all trend comparison orders and strict between subject factor testing (Tables 5.19 and 5.20 for the PID and FF controller results, respectively).

It was also found that there were statistically significant differences between the two controllers (Table 5.21) for all of the characteristics tested, except for the first peak time, with \( p \) values < 0.01 for all tested values, except for the between-controllers test for first peak time \( (p = 0.80) \), linear trend comparisons between controller type and activation level for first peak time, time constant, and rise time, and quadratic trend comparisons between controller type and activation level for first peak time, which all had \( p > 0.05 \).

Within the subjects tested for each controller, the fact that the first peak time and overshoot were similar for both the PID and FF controllers indicate that both controllers were able to achieve similar response times and initial errors between activation levels, indicating consistency within the controllers in that regard. Both controllers were able to create statistically significantly different settling values at each activation level, which was expected due to the large difference in force target setpoints. Where the controllers more unexpectedly differed between activation levels was in their transient responses, with the rise time having statistically significant differences between all activation levels in both controllers, making the interstitial phase between 10% and 90% of the
force setpoint less predictable. While undesirable, this is mitigated in the FF controller, as the
time constant was found to be statistically significantly similar between activation levels, meaning
that the discrepancies largely appeared between the 63% and 90% force output points. Having the
only discrepancy appear in this range also rules out signal noise as a contributing cause, because
the smaller time constant values were found to not be statistically significantly different while the
larger rise time values were, meaning the noise did not have an outsized impact on smaller values
in the FF controller. This is encouraging, as that means that there is a larger span of the transient
response that is consistent between activation levels for the FF controller, meaning that it is more
consistent operating in that range than the PID controller. While having any discrepancy between
activation level transient response is undesirable, the fact that this is limited to a narrow range
in the FF controller makes it more suitable for this mode of operation. To further assess the
differences between the controllers, the between-subjects factors testing must be used.

Table 5.9: Repeated measures one-way ANOVA analysis results for the first peak time of the
actuator when controlled by a PID controller. All significance values had a Bonferroni
correction applied.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference (s)</th>
<th>Std. Error (s)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.01</td>
<td>0.03</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.03</td>
<td>0.03</td>
<td>0.93</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-0.01</td>
<td>0.03</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.04</td>
<td>0.03</td>
<td>0.46</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.03</td>
<td>0.03</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.04</td>
<td>0.03</td>
<td>0.46</td>
</tr>
</tbody>
</table>

Table 5.10: Repeated measures one-way ANOVA analysis results for the time constant of the
actuator when controlled by a PID controller. All significance values had a Bonferroni
correction applied.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference (s)</th>
<th>Std. Error (s)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.19</td>
<td>0.02</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.01</td>
<td>0.01</td>
<td>0.94</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-0.19</td>
<td>0.02</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.18</td>
<td>0.02</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-0.01</td>
<td>0.01</td>
<td>0.94</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.18</td>
<td>0.02</td>
<td>&lt; 0.01</td>
</tr>
</tbody>
</table>
Table 5.11: Repeated measures one-way ANOVA analysis results for the rise time of the actuator when controlled by a PID controller. All significance values had a Bonferroni correction applied.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference (s)</th>
<th>Std. Error (s)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.35</td>
<td>0.02</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.19</td>
<td>0.02</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-0.35</td>
<td>0.02</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.17</td>
<td>0.03</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-0.19</td>
<td>0.02</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.17</td>
<td>0.03</td>
<td>&lt; 0.01</td>
</tr>
</tbody>
</table>

Table 5.12: Repeated measures one-way ANOVA analysis results for the percent overshoot of the actuator when controlled by a PID controller. All significance values had a Bonferroni correction applied.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference</th>
<th>Std. Error</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-0.006</td>
<td>0.005</td>
<td>0.75</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.003</td>
<td>0.003</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.006</td>
<td>0.005</td>
<td>0.75</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.008</td>
<td>0.005</td>
<td>0.42</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-0.003</td>
<td>0.003</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.008</td>
<td>0.005</td>
<td>0.42</td>
</tr>
</tbody>
</table>

Table 5.13: Repeated measures one-way ANOVA analysis results for the settling value of the actuator when controlled by a PID controller. All significance values had a Bonferroni correction applied.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference (g)</th>
<th>Std. Error (g)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-205.71</td>
<td>1.54</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-356.52</td>
<td>1.46</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>205.71</td>
<td>1.54</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-150.81</td>
<td>2.36</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>356.52</td>
<td>1.46</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>150.81</td>
<td>2.36</td>
<td>&lt; 0.01</td>
</tr>
</tbody>
</table>
Table 5.14: Repeated measures one-way ANOVA analysis results for the first peak time of the actuator when controlled by a FF controller. All significance values had a Bonferroni correction applied.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference (s)</th>
<th>Std. Error (s)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-0.03</td>
<td>0.02</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.05</td>
<td>0.02</td>
<td>0.06</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.03</td>
<td>0.02</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.02</td>
<td>0.03</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.05</td>
<td>0.02</td>
<td>0.06</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.02</td>
<td>0.03</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5.15: Repeated measures one-way ANOVA analysis results for the time constant of the actuator when controlled by a FF controller. All significance values had a Bonferroni correction applied.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference (s)</th>
<th>Std. Error (s)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.00</td>
<td>0.02</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.02</td>
<td>0.01</td>
<td>0.41</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-0.00</td>
<td>0.02</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.02</td>
<td>0.02</td>
<td>0.94</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.02</td>
<td>0.01</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.02</td>
<td>0.02</td>
<td>0.94</td>
</tr>
</tbody>
</table>

Table 5.16: Repeated measures one-way ANOVA analysis results for the rise time of the actuator when controlled by a FF controller. All significance values had a Bonferroni correction applied.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference (s)</th>
<th>Std. Error (s)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.03</td>
<td>0.02</td>
<td>0.84</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.16</td>
<td>0.02</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-0.03</td>
<td>0.02</td>
<td>0.84</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.14</td>
<td>0.03</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-0.16</td>
<td>0.02</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.14</td>
<td>0.03</td>
<td>&lt; 0.01</td>
</tr>
</tbody>
</table>
Table 5.17: Repeated measures one-way ANOVA analysis results for the percent overshoot of the actuator when controlled by a FF controller. All significance values had a Bonferroni correction applied.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference</th>
<th>Std. Error</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-0.002</td>
<td>0.004</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.001</td>
<td>0.005</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.002</td>
<td>0.004</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.001</td>
<td>0.004</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.001</td>
<td>0.005</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.001</td>
<td>0.004</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5.18: Repeated measures one-way ANOVA analysis results for the settling value of the actuator when controlled by a FF controller. All significance values had a Bonferroni correction applied.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Compared To</th>
<th>Mean Difference (g)</th>
<th>Std. Error (g)</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>-199.21</td>
<td>1.25</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-348.80</td>
<td>1.93</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>199.21</td>
<td>1.25</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-149.58</td>
<td>2.05</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>348.80</td>
<td>1.93</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>149.58</td>
<td>2.05</td>
<td>&lt; 0.01</td>
</tr>
</tbody>
</table>

The between-subject factors testing (Tables 5.19–5.21) indicates that, much like in previous experiments, the actuators behaved similarly to one another, with different actuators performing with consistent results for all control characteristics for each controller. This is desirable, as it further reinforces the repeatability of the actuator’s performance, and extends it further by also proving that, even when controlled, there are statistically no differences between different fabricated actuators. The results of the controller type between-subjects testing indicate that the controllers are not statistically significantly different with respect to first peak time, indicating that both are able to reach their target force setpoints at similar times at each respective activation level, while being statistically different in all other regards. This means that, from the time constant and rise time being different, the transient response of the two controllers are statistically significantly different, that one controller overshoots its target statistically significantly more than the other, and that the two have statistically significant steady state errors, which is implied from the different
settling values. Therefore, one of the controllers is able to limit the error in the system significantly better than the other, while having a similar response time, but different transient responses. The trend comparisons show that there is a statistically significant linear trend between controller type and and activation level for overshoot and activation level, and a statistically significant quadratic trend between controller type and time constant, rise time, overshoot, and settling value, meaning that, not only are the end values different between controllers, but the amount they differ changes per activation level used. These findings came from the ANOVA testing, which, as mentioned before, is less than perfect when dealing with non-normal distributions, such as time values. To that end, the findings for the first peak times, time constants, and rise times must be compared with the Kruskal-Wallis analyses, to further verify their accuracy.

Table 5.19: Between-subject comparison results of the repeated measures one-way ANOVA tests performed for the PID controller, comparing different actuators.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Linear Trend Comparison</th>
<th>Quadratic Trend Comparison</th>
<th>Same Activation Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Peak Time</td>
<td>0.29</td>
<td>0.91</td>
<td>0.10</td>
</tr>
<tr>
<td>Time Constant</td>
<td>0.36</td>
<td>0.87</td>
<td>0.28</td>
</tr>
<tr>
<td>Rise Time</td>
<td>0.58</td>
<td>0.87</td>
<td>0.14</td>
</tr>
<tr>
<td>Overshoot</td>
<td>0.50</td>
<td>0.54</td>
<td>0.54</td>
</tr>
<tr>
<td>Settling Value</td>
<td>0.28</td>
<td>0.49</td>
<td>0.42</td>
</tr>
</tbody>
</table>

Table 5.20: Between-subject comparison results of the repeated measures one-way ANOVA tests performed for the FF controller, comparing different actuators.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Linear Trend Comparison</th>
<th>Quadratic Trend Comparison</th>
<th>Same Activation Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Peak Time</td>
<td>0.09</td>
<td>0.34</td>
<td>0.32</td>
</tr>
<tr>
<td>Time Constant</td>
<td>0.22</td>
<td>0.34</td>
<td>0.73</td>
</tr>
<tr>
<td>Rise Time</td>
<td>0.59</td>
<td>0.50</td>
<td>0.61</td>
</tr>
<tr>
<td>Overshoot</td>
<td>0.84</td>
<td>0.65</td>
<td>0.15</td>
</tr>
<tr>
<td>Settling Value</td>
<td>0.71</td>
<td>0.76</td>
<td>0.14</td>
</tr>
</tbody>
</table>
Table 5.21: Between-subject comparison results of the repeated measures one-way ANOVA test between controllers, comparing the two.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Linear Trend Comparison</th>
<th>Quadratic Trend Comparison</th>
<th>Same Activation Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Peak Time</td>
<td>0.56</td>
<td>0.38</td>
<td>0.80</td>
</tr>
<tr>
<td>Time Constant</td>
<td>0.09</td>
<td>&lt; 0.01</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>Rise Time</td>
<td>0.40</td>
<td>&lt; 0.01</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>Overshoot</td>
<td>&lt; 0.01</td>
<td>&lt; 0.01</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>Settling Value</td>
<td>&lt; 0.01</td>
<td>&lt; 0.01</td>
<td>&lt; 0.01</td>
</tr>
</tbody>
</table>

From the between-actuator grouping Kruskal-Wallis testing performed on the PID controller for the first peak time, time constant, and rise time results (Tables 5.22, 5.23, and 5.24, respectively), it can be seen that the results of the between-subjects factor testing above are accurate, with different fabricated actuators reporting not statistically significant differences between them for any of the time characteristics. Similarly, the results of the Kruskal-Wallis testing for the first peak time, time constant, and rise time (Tables 5.25, 5.26, and 5.27, respectively) also uphold the results of the between-subjects testing from the ANOVA analyses, with each of those tests finding no statistically significant differences between different fabricated actuators for any of those values at each activation level ($p > 0.05$). Comparing the results of the between-subjects factor testing in the ANOVA testing done when comparing controller types to the Kruskal-Wallis results in Tables 5.28, 5.29, and 5.30 (first peak time, time constant, and rise time, respectively), it can be seen that the results are also largely the same, with the first peak time having no difference at any activation level between controllers ($p$ values of 0.07, 0.53, and 0.14 for each activation level in ascending order), and the time constant and rise time being statistically significantly different at most activation levels ($p < 0.05$). However, it was found that at activation level three the two controllers performed similarly ($p = 0.78$ for time constant and $p = 0.45$ for rise time), indicating that while the trend is on average true that the time constants differ, in this instance the null hypothesis could not be rejected. These further uphold the conclusion that while the controllers are able to achieve the final target threshold at the same time at each activation level, they behave less predictably and usually differently in their transient responses.
Table 5.22: Results of Kruskal-Wallis testing performed on the first peak time results of the actuator when controlled by a PID controller, grouped by actuator number.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significance</td>
<td>0.51</td>
<td>0.32</td>
<td>0.17</td>
</tr>
</tbody>
</table>

Table 5.23: Results of Kruskal-Wallis testing performed on the time constant results of the actuator when controlled by a PID controller, grouped by actuator number.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significance</td>
<td>0.29</td>
<td>0.91</td>
<td>0.27</td>
</tr>
</tbody>
</table>

Table 5.24: Results of Kruskal-Wallis testing performed on the rise time results of the actuator when controlled by a PID controller, grouped by actuator number.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significance</td>
<td>0.19</td>
<td>0.77</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Table 5.25: Results of Kruskal-Wallis testing performed on the first peak time results of the actuator when controlled by a FF controller, grouped by actuator number.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significance</td>
<td>0.17</td>
<td>0.76</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Table 5.26: Results of Kruskal-Wallis testing performed on the time constant results of the actuator when controlled by a FF controller, grouped by actuator number.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significance</td>
<td>0.17</td>
<td>0.99</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Table 5.27: Results of Kruskal-Wallis testing performed on the rise time results of the actuator when controlled by a FF controller, grouped by actuator number.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significance</td>
<td>0.14</td>
<td>0.70</td>
<td>0.27</td>
</tr>
</tbody>
</table>
Table 5.28: Results of Kruskal-Wallis testing performed on the first peak time results of the actuator, grouped by controller type.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significance</td>
<td>0.07</td>
<td>0.53</td>
<td>0.14</td>
</tr>
</tbody>
</table>

Table 5.29: Results of Kruskal-Wallis testing performed on the time constant results of the actuator, grouped by controller type.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significance</td>
<td>0.02</td>
<td>&lt; 0.01</td>
<td>0.78</td>
</tr>
</tbody>
</table>

Table 5.30: Results of Kruskal-Wallis testing performed on the rise time results of the actuator, grouped by controller type.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significance</td>
<td>0.03</td>
<td>&lt; 0.01</td>
<td>0.45</td>
</tr>
</tbody>
</table>

This can also be seen in the numerical values found, seen in Tables 5.31, 5.32, and 5.33 for the first peak time, time constant, and rise time results from the PID controller, and Tables 5.34, 5.35, and 5.36 for the first peak time, time constant, and rise time results from the FF controller. From constructing the confidence intervals, the average first peak times in ascending order were found to be 2.61, 2.60, and 2.63 seconds for each activation level in ascending order for the PID control scheme, and 2.55, 2.58, and 2.60 seconds for each activation level in ascending order for the FF control scheme, which were indeed very similar to each other, giving credence to the findings of the between-subjects comparisons and Kruskal-Wallis analysis. These first peak times were slower than in the minimum contraction period experiment, but this was to be expected, as tapering power consumption as the systems reach their force setpoints will reduce their heating rates. That being said, the difference is marginal, meaning that the FF controller is able to performance with minimal compromises to the response time.

For the other time characteristics, the average time constants were found to be 1.35, 1.16, and 1.33 seconds in ascending order of activation levels for the PID control scheme, and 1.32, 1.31, and 1.33 seconds in ascending order for the FF control scheme, which can be seen to reflect the conclusions drawn from the Kruskal-Wallis analysis. The average rise times in ascending order
for the PID controller were found to be 2.13, 1.78, and 1.94 seconds, while they were 2.09, 2.06, and 1.92 seconds for the FF controller, which also upholds the finding that the two controllers’ transient responses were different except for the third activation level. Further work is required to see if this is a result of a trend of their differences converging with activation level, or if it is due to variance in their transient responses between activation levels. Overall, these time characteristic results show that the two controllers behave statistically similarly in reaching their force targets, with the FF controller able to achieve a slightly faster minimum contraction period than the PID controller (2.60 s compared to 2.63 s), while differing for the majority of the activation levels in their transient response.

Table 5.31: Constructed 95% confidence intervals of the first peak time achieved by the actuator when controlled by a PID controller. All values in seconds.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>2.61</td>
<td>0.06</td>
<td>2.58</td>
</tr>
<tr>
<td>2</td>
<td>2.60</td>
<td>0.07</td>
<td>2.56</td>
</tr>
<tr>
<td>3</td>
<td>2.63</td>
<td>0.06</td>
<td>2.60</td>
</tr>
</tbody>
</table>

Table 5.32: Constructed 95% confidence intervals of the time constant achieved by the actuator when controlled by a PID controller. All values in seconds.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>1.35</td>
<td>0.033</td>
<td>1.33</td>
</tr>
<tr>
<td>2</td>
<td>1.16</td>
<td>0.055</td>
<td>1.13</td>
</tr>
<tr>
<td>3</td>
<td>1.33</td>
<td>0.030</td>
<td>1.32</td>
</tr>
</tbody>
</table>

Table 5.33: Constructed 95% confidence intervals of the rise time achieved by the actuator when controlled by a PID controller. All values in seconds.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>2.13</td>
<td>0.04</td>
<td>2.10</td>
</tr>
<tr>
<td>2</td>
<td>1.78</td>
<td>0.06</td>
<td>1.74</td>
</tr>
<tr>
<td>3</td>
<td>1.94</td>
<td>0.06</td>
<td>1.91</td>
</tr>
</tbody>
</table>
Table 5.34: Constructed 95% confidence intervals of the first peak time achieved by the actuator when controlled by a FF controller. All values in seconds.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>2.55</td>
<td>0.06</td>
<td>2.52</td>
</tr>
<tr>
<td>2</td>
<td>2.58</td>
<td>0.07</td>
<td>2.54</td>
</tr>
<tr>
<td>3</td>
<td>2.60</td>
<td>0.08</td>
<td>2.56</td>
</tr>
</tbody>
</table>

Table 5.35: Constructed 95% confidence intervals of the time constant achieved by the actuator when controlled by a FF controller. All values in seconds.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>1.32</td>
<td>0.037</td>
<td>1.30</td>
</tr>
<tr>
<td>2</td>
<td>1.31</td>
<td>0.036</td>
<td>1.29</td>
</tr>
<tr>
<td>3</td>
<td>1.33</td>
<td>0.037</td>
<td>1.31</td>
</tr>
</tbody>
</table>

Table 5.36: Constructed 95% confidence intervals of the rise time achieved by the actuator when controlled by a FF controller. All values in seconds.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>2.09</td>
<td>0.053</td>
<td>2.06</td>
</tr>
<tr>
<td>2</td>
<td>2.06</td>
<td>0.056</td>
<td>2.03</td>
</tr>
<tr>
<td>3</td>
<td>1.92</td>
<td>0.071</td>
<td>1.88</td>
</tr>
</tbody>
</table>

The ANOVA analysis found statistically significant differences in the force production based characteristics of the actuator, and this difference can be clearly seen in the average overshoot, settling value, and steady state errors present at each activation level in the two control systems. For the PID controller the average percent overshoot values were 5.6%, 6.2%, and 5.3% for each activation level in ascending order (Table 5.37), while they were 2.4%, 2.6%, and 2.6% for the same activation levels in the FF control scheme (Table 5.38), indicating that the FF controller is much more able to control the overshoot of the system. Showing a similar trend, the settling values were 204.62 g, 410.32 g, and 561.13 g for each activation level in ascending order for the PID control scheme (Table 5.39), and 198.12 g, 397.34 g, and 546.92 g for the same activation levels for the
FF control scheme (Table 5.40) resulting in steady state errors (Table 5.41) of 2.31%, 2.58%, and 2.02% for the PID controller in order of ascending activation level, and 0.94%, 0.67%, and 0.56% for the FF controller in that same order. These results clearly show that the FF control scheme was the more accurate of the two, achieving much lower steady state errors than the PID controller at every activation level. Having the settling values below the target setpoint is also desirable, as it means that even with the small amount of steady state error it has, it tends to err towards less force. This is very desirable for its intended use as a wearable therapeutic device, as safety of the user is paramount, and exceeding force setpoints would result in excess force being applied to the user, possibly causing injury. Therefore, achieving statistically similar response times to the PID controller while having statistically significantly less overshoot and steady state error, which tends towards a conservative settling value, makes the FF controller a superior controller for its intended purpose.

Table 5.37: Constructed 95% confidence intervals of the percent overshoot achieved by the actuator when controlled by a PID controller.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>0.056</td>
<td>0.009</td>
<td>0.051</td>
</tr>
<tr>
<td>2</td>
<td>0.062</td>
<td>0.013</td>
<td>0.054</td>
</tr>
<tr>
<td>3</td>
<td>0.053</td>
<td>0.011</td>
<td>0.048</td>
</tr>
</tbody>
</table>

Table 5.38: Constructed 95% confidence intervals of the percent overshoot achieved by the actuator when controlled by a FF controller.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>0.024</td>
<td>0.013</td>
<td>0.017</td>
</tr>
<tr>
<td>2</td>
<td>0.026</td>
<td>0.012</td>
<td>0.020</td>
</tr>
<tr>
<td>3</td>
<td>0.026</td>
<td>0.012</td>
<td>0.019</td>
</tr>
</tbody>
</table>
5.7 Analysis

Table 5.39: Constructed 95% confidence intervals of the settling value achieved by the actuator when controlled by a PID controller. All values in grams force equivalent.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>204.62</td>
<td>1.76</td>
<td>203.58</td>
</tr>
<tr>
<td>2</td>
<td>410.32</td>
<td>4.91</td>
<td>407.39</td>
</tr>
<tr>
<td>3</td>
<td>561.13</td>
<td>5.91</td>
<td>557.96</td>
</tr>
</tbody>
</table>

Table 5.40: Constructed 95% confidence intervals of the settling value achieved by the actuator when controlled by a FF controller. All values in grams force equivalent.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>95% Confidence Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lower Bound</td>
</tr>
<tr>
<td>1</td>
<td>198.12</td>
<td>2.41</td>
<td>196.69</td>
</tr>
<tr>
<td>2</td>
<td>397.34</td>
<td>4.47</td>
<td>394.93</td>
</tr>
<tr>
<td>3</td>
<td>546.92</td>
<td>6.17</td>
<td>543.35</td>
</tr>
</tbody>
</table>

Table 5.41: Steady state error of the experimental settling values compared to their target setpoints, for both the PID and FF control schemes. All values, unless stated otherwise, in grams force equivalent.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Controller</th>
<th>Experimental Result</th>
<th>Target Value</th>
<th>Raw Diff.</th>
<th>% Diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PID</td>
<td>204.62</td>
<td>200</td>
<td>4.62</td>
<td>0.0231</td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>198.12</td>
<td>200</td>
<td>1.88</td>
<td>0.0094</td>
</tr>
<tr>
<td>2</td>
<td>PID</td>
<td>410.32</td>
<td>400</td>
<td>10.32</td>
<td>0.0258</td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>397.34</td>
<td>400</td>
<td>2.66</td>
<td>0.0067</td>
</tr>
<tr>
<td>3</td>
<td>PID</td>
<td>561.13</td>
<td>550</td>
<td>11.13</td>
<td>0.0202</td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>546.92</td>
<td>550</td>
<td>3.08</td>
<td>0.0056</td>
</tr>
</tbody>
</table>
In addition to comparing the two controllers directly on performance, it was also desirable to determine if the characterization and control law development methods accurately predicted the results found through experimentation. Comparing the experimental results to the simulated responses of the system, it becomes clear that they diverged, likely due to errors introduced during system identification and limitations imposed by the system. Looking at Tables 5.42, 5.43, 5.44, 5.45, and 5.46, it is plain to see that while there are a good portion of the results that were well within the bounds of reason for similarity to their respective simulated values, some values, such as the time constant for the FF control scheme operating on the first activation level of the actuator were as much as 24.58% different from their simulated values. The greatest differences were found in the time based values, with values differing as much as 14.72% greater than simulated values for the first peak time, 23.28% greater than simulated values for the rise time, and 24.58% lesser than simulated values for the time constant. These discrepancies indicate that the actuators are able to increase in temperature (and therefore force) faster than the simulated values at lower temperatures, giving a decreased time constant, while being slower at higher temperatures, giving increased rise time and first peak time values. Expressed graphically, their force–time curves are steeper initially, and become significantly shallower quicker.

Table 5.42: Comparison of the experimental first peak time values to those obtained via simulation, for both the PID and FF control schemes. All values, unless stated otherwise, in seconds.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Controller</th>
<th>Experimental Result</th>
<th>Simulated Value</th>
<th>Raw Diff.</th>
<th>% Diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PID</td>
<td>2.61</td>
<td>2.50</td>
<td>0.10</td>
<td>0.0406</td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>2.55</td>
<td>2.84</td>
<td>-0.29</td>
<td>-0.1032</td>
</tr>
<tr>
<td>2</td>
<td>PID</td>
<td>2.60</td>
<td>2.27</td>
<td>0.33</td>
<td>0.1451</td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>2.58</td>
<td>2.25</td>
<td>0.33</td>
<td>0.1472</td>
</tr>
<tr>
<td>3</td>
<td>PID</td>
<td>2.63</td>
<td>2.76</td>
<td>-0.12</td>
<td>-0.0439</td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>2.60</td>
<td>2.76</td>
<td>-0.17</td>
<td>-0.0601</td>
</tr>
</tbody>
</table>
Table 5.43: Comparison of the experimental time constant values to those obtained via simulation, for both the PID and FF control schemes. All values, unless stated otherwise, in seconds.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Controller</th>
<th>Experimental Result</th>
<th>Simulated Value</th>
<th>Raw Diff.</th>
<th>% Diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PID</td>
<td>1.35</td>
<td>1.66</td>
<td>-0.32</td>
<td>-0.1895</td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>1.32</td>
<td>1.75</td>
<td>-0.43</td>
<td>-0.2458</td>
</tr>
<tr>
<td>2</td>
<td>PID</td>
<td>1.16</td>
<td>1.41</td>
<td>-0.25</td>
<td>-0.1764</td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>1.31</td>
<td>1.35</td>
<td>-0.03</td>
<td>-0.0238</td>
</tr>
<tr>
<td>3</td>
<td>PID</td>
<td>1.33</td>
<td>1.26</td>
<td>0.08</td>
<td>0.0604</td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>1.33</td>
<td>1.33</td>
<td>0.01</td>
<td>0.0053</td>
</tr>
</tbody>
</table>

Table 5.44: Comparison of the experimental rise time values to those obtained via simulation, for both the PID and FF control schemes. All values, unless stated otherwise, in seconds.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Controller</th>
<th>Experimental Result</th>
<th>Simulated Value</th>
<th>Raw Diff.</th>
<th>% Diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PID</td>
<td>2.13</td>
<td>1.89</td>
<td>0.24</td>
<td>0.1251</td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>2.10</td>
<td>1.96</td>
<td>0.13</td>
<td>0.0649</td>
</tr>
<tr>
<td>2</td>
<td>PID</td>
<td>1.78</td>
<td>1.84</td>
<td>-0.06</td>
<td>-0.0322</td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>2.06</td>
<td>1.67</td>
<td>0.39</td>
<td>0.2328</td>
</tr>
<tr>
<td>3</td>
<td>PID</td>
<td>1.94</td>
<td>1.64</td>
<td>0.30</td>
<td>0.1820</td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>1.92</td>
<td>1.56</td>
<td>0.36</td>
<td>0.2325</td>
</tr>
</tbody>
</table>

There are a few possible explanations for this. One is that the system identification process was less than perfect for determining the transfer functions for the system. Because the original force–temperature and power–temperature relationships were regressed from experimental data, they are inherently imperfect approximations, as they did not fit the experimental data with 100% accuracy, meaning each actuator has slightly different relationships, likely due to slight material and manufacturing irregularities. As well, the transfer functions identified from these relationships did not achieve a 100% fit either, indicating further variance from an ideal representation of the actuators' performance. In addition to this, PWM control was used to control the power applied to the system, as opposed to the ideal power control used in simulations. PWM works by supplying power to the system for a percentage of a time period, effectively supplying that same percentage of the maximum possible power to the system. Because this entire time period must elapse, in this case 100 ms, new desired power values can only be calculated every 100 ms instead of
5.7 Analysis

instantaneously, which inherently reduces the accuracy of the experimental implementation. As well, the PWM duty cycle accuracy was limited by the polling speed of the system, as that represented the smallest possible division of the PWM duty cycle. Increasing the clock speed of the system would increase this polling speed, making the PWM duty cycle more accurate and enabling smaller cycle-times, increasing the accuracy of the system in two ways. Finally, looking at Figure 4.45 from Section 4.4, it can be seen that while 80 W of power per TCA pair is available to the actuator, the actuator does not consume this amount at all points, with the values changing as heating progresses. This too could affect the heating process of the actuator, introducing errors when compared to a simulated response that assumes constant power consumption. Using an adaptive power supply based on power consumed could remove this source of error.

That being noted, while the time-based performance metrics of the system were different from the simulated values, the force-based predictions were much more closely related. The percent overshoot values for the FF control scheme in ascending order by activation level were 2.50, 2.81, and 2.70 percentage points greater than simulated values, while the PID control scheme had values 4.12, 6.42, and 5.54 percentage points greater than simulated values. Similarly, the settling values of the FF control scheme in ascending order by activation level were 0.71%, 0.37%, and 0.32% lower than simulated values, with the PID control scheme having settling values 8.13%, 2.82%, and 2.44% greater than simulated values. These are much closer to the simulated values, with the FF control scheme being much more true to the simulated values than the PID control scheme. This indicates that, while the simulations were less well able to predict the time-based performance of the actuation system, they were very close to being able to correctly identify the maximum and

Table 5.45: Comparison of the experimental percent overshoot values to those obtained via simulation, for both the PID and FF control schemes.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Controller</th>
<th>Experimental Result</th>
<th>Simulated Value</th>
<th>Raw Diff.</th>
<th>% Diff.</th>
</tr>
</thead>
</table>
| 1
| PID              | 0.056      | 0.015               | 0.041          | 2.733     |
| FF               | 0.024      | -0.001              | 0.025          | 25.000    |
| 2
| PID              | 0.062      | -0.002              | 0.064          | 32.000    |
| FF               | 0.026      | -0.002              | 0.0281         | 14.050    |
| 3
| PID              | 0.053      | -0.002              | 0.055          | 27.500    |
| FF               | 0.026      | -0.001              | 0.027          | 27.000    |
Table 5.46: Comparison of the experimental settling values to those obtained via simulation, for both the PID and FF control schemes. All values, unless stated otherwise, in grams force equivalent.

<table>
<thead>
<tr>
<th>Activation Level</th>
<th>Controller</th>
<th>Experimental Result</th>
<th>Simulated Value</th>
<th>Raw Diff.</th>
<th>% Diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PID</td>
<td>204.62</td>
<td>202.96</td>
<td>1.65</td>
<td>0.008</td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>198.12</td>
<td>199.53</td>
<td>-1.41</td>
<td>-0.007</td>
</tr>
<tr>
<td>2</td>
<td>PID</td>
<td>410.32</td>
<td>399.06</td>
<td>11.26</td>
<td>0.028</td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>397.34</td>
<td>398.83</td>
<td>-1.49</td>
<td>-0.004</td>
</tr>
<tr>
<td>3</td>
<td>PID</td>
<td>561.13</td>
<td>547.77</td>
<td>13.36</td>
<td>0.024</td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>546.92</td>
<td>548.69</td>
<td>-1.78</td>
<td>-0.003</td>
</tr>
</tbody>
</table>

settling force of the system in both control scenarios. This would indicate that perhaps, of the possible issues identified previously, the limitations of the PWM control and non-constant power consumption were the more impactful, because those have an oversized impact on the time characteristics of the response of the system compared to the force characteristics of the response, which were much more accurate. To overcome these limitations, greater computing power, more efficient code, adaptive modelling based on current power consumption, and more durable power transmission components should be used to facilitate greater response times, more accurate simulations, and decrease the variance between simulated and experimental results.

5.8 Conclusions

These experiments explored the isometric force control of the designed woven bi-pennate actuator, using both a FF and PID control scheme designed based on previously constructed relationships from the results of the force characterization experiment with the assistance of system identification and simulation software. From the results of these experiments, repeated measures one-way ANOVAs were conducted to determine differences between activation levels operating under each controller, differences between fabricated actuators operating under each controller, and differences between the performance of each controller type, confidence intervals of achieved values were constructed, and Kruskal-Wallis analyses were performed on time-related performance metrics, comparing the differences between activation levels for different fabricated actuators and controller types. From these analyses, it was found that the first peak time and percent overshoot for each
control scheme for each activation level were statistically similar, while the settling value was different at each activation level for each control system. These results were expected from the results of the minimum contraction period and force characterization experiments, which found that the maximum contraction times were similar between activation levels, while each activation level was able to produce a distinct, predictable force value. However, it was found that for the PID controller, the time constant and rise time were statistically significantly different for some activation levels, while only the rise time was found to be statistically significantly different for some activation levels for the FF controller. These results indicate that, while the steady state response and response time of each activation level of each controller is similar, and therefore predictable, the transient response is different, and further work is required to properly predict how the actuator changes its behaviour per activated level in that transient range. The fact that the time constants were statistically similar for the FF controller indicates that the range of variance is narrower than in the case of the PID controller.

The between-subjects factors analysis revealed that both controllers did not vary their results to a statistically significantly degree between actuators, further reinforcing the repeatability of the design as seen in the experiments of Chapter 4. This was true for both the between-subjects factors analysis done via ANOVA and Kruskal-Wallis. Applying the between-subjects factors testing to the two controller types, accompanied by numerical analysis from the confidence intervals, revealed that the FF controller was statistically the better performer for rehabilitative force control purposes. The testing found that the FF controller was able to statistically significantly reduce the overshoot and steady state error of the system ($p < 0.01$), while not compromising on response time (first peak time, $p = 0.80$ for ANOVA, $p = 0.07, 0.53, 0.14$ for Kruskal-Wallis). It also found that the two controllers had distinct transient responses according to their time constants and rise times ($p > 0.05$) for all but the third activation level, found via the Kruskal-Wallis testing. This discrepancy, while beyond the scope of this experiment, should be investigated in future works if operation within this subset is found to be desirable. The maximum overshoot of the FF controller was found to be 2.6%, with a maximum steady state error of 0.94%. Based on the first peak time results, the minimum contraction period found for the actuator was found to be 2.60 seconds operating under this control scheme with this power level. While any overshoot or steady state
error is undesirable, these results indicate that, of the two, the designed FF controller performs more conservatively and accurately over an extended period of time without compromising on response time, which is a preferred quality to have in a rehabilitative scenario where exceeding force setpoints can result in user injury.

Despite favourable results, the control systems were imperfect, featuring some amount of overshoot, oscillations, and steady state error, as well as diverging from simulated results, particularly with the time domain characteristics calculated from their responses. This likely was a result of a combination of the approximations used in the system identification process, imperfect regression from experimental data, the use of PWM power control, which inherently confers an output ripple contributing to error due to its nature, slower-than-optimal refresh rates for PWM cycle calculation, and the assumption of constant power consumption throughout the heating process. If these could be rectified, more accurate simulations could be created, enabling better prediction of system performance and enabling the better selection of controller coefficients, which in turn would lessen or eliminate any overshoot, oscillations, or steady state error present in the system. In future research endeavours, improvements to these areas could be realized through the use of different regression methodologies and technologies in the construction of the controlled actuation system.

To improve system identification, machine learning techniques could be used to better regress force–temperature and temperature–power consumption relationships from experimental data. Further enhancing the accuracy of the model and addressing the non-constant power consumption, adaptive model parameter estimation could be used based on the state of the system at any given point, further increasing model accuracy. In order to facilitate these two improvements, as well as decrease the required PWM cycle time and increase refresh rate, even further increasing accuracy, a more powerful microprocessor should be used. With the implementation of these changes, the performance of the designed control systems would only increase, eliminating or lessening overshoot, oscillations, and steady state error, making the designed actuator even more viable for incorporation into a wearable forearm neurorehabilitation device.

Acknowledging the imperfections present in the system, and the improvements that could be made, both controllers were able to control the force output of the system, limiting the power consumption of the actuator to approach the force setpoints gradually and approximately
maintaining the force setpoints for a duration. Both control system implementations were able to achieve their force setpoints while operating at contraction frequencies greater than the maximum allowed contraction period of 3.57 s, albeit at slightly slower paces than found in the minimum contraction period experiment. This was done through purely force control, with the only feedback for the system being the force output of the actuator and the only input being a DC power supply. These results show that the actuator is controllable in variable recruitment scenarios using only force feedback, achieving desired forces in a controlled manner at a suitable rate with few components and fast sampling rates, and that the use of the designed two DOF FF controller was able to achieve the same performance increases in the bi-pennate actuator as in the literature for single TCAs, reducing overshoot and steady state error with minimal impact on response time, which is highly desirable for rehabilitative purposes. They also prove that the design process used to construct these force-feedback control systems is a viable way of developing such control schemes, providing predictable, actionable results that can be used to control these actuators.
Chapter 6

Concluding Remarks

The work presented in this thesis aimed to develop a viable bi-pennate TCA-based actuation system embedded in a woven mesh for use in wearable mechatronic rehabilitation devices. This was done by determining the isotonic stroke and isometric force production capabilities of the actuator, its contractive actuation frequency, its electrical-to-mechanical power conversion efficiency, and developing models to predict the transient stroke and force response of the system. From these created models, force control schemes were designed and tested to determine to what degree the actuator could be controlled, and which of the control schemes was best suited for the actuator’s intended purpose. A literature review was performed to determine the shortcomings of current robot-assisted therapy devices, as well as to develop the torque, range of motion, and actuation frequency requirements of wrist neurorehabilitation, and were compared to the production capabilities of the actuator.

The results of the stroke characterization experiment found that the maximum stroke of the actuators increased when embedded in a woven mesh, as predicted in the literature [13], and because of this were able to achieve a maximum stroke of 12.23%, while achieving an efficiency of 1.8%, similar to results found in the literature [60]. This efficiency was found to increase with actuation speed, meaning there could be improvements with the use of more power. It was also found that the force production increased when embedded in a woven mesh, with the maximum force produced by the actuator being 11.28 N. Both the stroke and force production were found to be exceedingly linear on a number of active TCAs basis, indicating that the actuator can be
scaled easily to meet different force and stroke demands. Different instances of the actuator had little variance between them, indicating repeatable, reliable actuation with different modules. It was also found that the actuator could meet average therapeutic torque, range of motion, and frequency requirements with or without the use of a gear reduction, with the number of TCAs being easily scaled to meet them. However, only two of the three could be met at once, requiring devices incorporating the designed actuator to be specific to one type of exercise, or adjustable to meet different needs at different times. If all three requirements are to be met at once, additional power must be used to allow for higher gear reductions to be used without overtly limiting the actuator’s contractive frequency.

After developing a force–temperature relationship for the actuator, FF and PID force controllers were developed and tuned to minimize overshoot and steady state error. These operated exclusively on force feedback, as the goal was to minimize the complexity and cost of the system, which would impose barriers to adoption. The FF controller was found to be the more conservative of the two, with a maximum overshoot of 2.6% and a maximum steady state error of 0.94%. This was compared to the PID controller, which had a maximum overshoot of 6.2% and a maximum steady state error of 2.58%. This indicates that the FF controller would be preferred for therapeutic devices, as overshoot and steady state error represent possibilities of injuring the user. The FF controller was able to achieve a minimum contractive period of 2.60 seconds, greater than the minimum contractive period of 2.43 seconds found for the actuator when uncontrolled but still exceeding average therapeutic requirements by a wide margin. The results show that that the designed fabric actuator is a viable for use in wearable mechatronic therapy devices, scaling to meet therapeutic requirements while being able to be accurately controlled in a timely manner with minimal additional components. This, in combination with the actuator’s innate flexibility, ease of manufacture, and low cost, make it an appealing possibility for inclusion in wearable devices. Although this work demonstrates that potential of the woven bi-pennate TCA actuator design, further work is required to refine it before it can be incorporated in medical and commercial devices.
6.1 Contributions

The contributions of this thesis are as follows:

1. Designed and developed a novel bi-pennate artificial muscle matrix comprised of TCAs that could be embedded in a woven matrix, the first matrix of its kind comprised of TCAs and incorporated into wearable fabrics. This novel design was able to match or exceed other artificial muscle actuators in terms of stroke, force production, and efficiency, while being incorporated into a fabric. This gives the designed actuator a distinct advantage over other actuator types, being easily incorporable into garments for use in wearable mechatronic devices, without compromising performance. While the actuator could not be made to meet all clinical requirements concurrently, with the use of different radial arms or torque transmission components it could be made to reach each clinical requirement in separate scenarios, showing that it could perhaps be applied to therapeutic purposes with further design refinement.

2. Proved the consistency of results between manufactured novel actuators. With minimal variation between manufactured actuators, the results of the characterization of their maximum performance metrics can be used to reliably predict the performance of similarly made actuators. As well, the low level of deviation from standard values entices their use in therapeutic devices, where unforeseen performance can result in patient injury.

3. Developed numerical relationships between the novel actuator’s ability to produce stroke and force, and the temperature and number of its constituent TCAs. These allow for the prediction of the actuator’s performance at any point within its operational range, giving designers more options in how they want to incorporate or scale the actuator to meet the specific requirements of different usage scenarios. Inversely, they also allow for designers to determine the actuator’s state for a given performance requirement.

4. Used the characterized force–temperature relationship to develop a purely force-feedback based composite feedforward-feedback force control scheme. This controller was able to control the force response of the system with minimal overshoot or settling error, and was
able to achieve comparable response time to a PID controller baseline. Its conservative nature makes it ideal for a clinical scenario, where overshoot or error in the force response could result in patient injury. Using a purely force-feedback controller also limits the number of components in the system, making it cheaper and easier to integrate into future device designs.

6.2 Limitations & Future Work

While the designed actuator was able to meet the design requirements and perform comparably to other artificial muscle actuators in terms of stroke, force, and efficiency, the results show that further improvements and avenues of inquiry could be explored to even further increase the capacity for actuators such as these to be implemented in robot-assisted wearable therapeutic devices. Some of the future improvements and topics to explore are listed below:

1. Investigate the architectural gearing properties of the pennate configuration. One of the key benefits of the pennate configuration in biological muscle is its ability to dynamically cushion rapid changes in force and contraction, protecting the body from jerky motions [88]. It does this by changing the pennation angle of the muscle fibres as they contract, rotating them so that they apply their force increasingly perpendicular to the line of motion of the tendon they are attached to, applying less force near the end of its range of motion. Future work should investigate whether this same phenomenon occurs with artificial muscle fibres, to what degree, how the initial pennation angle affects this effect and the actuator’s performance, and how much of a benefit that confers to actuators designed with this shape.

2. Investigate the use of different mesh materials, and how they affect actuator performance. In previous work, researchers have noted that changing the stiffness of a fabric mesh can alter the capabilities of embedded actuators [13]. Typically, there was an inverse relationship found between force and stroke production changes [13], meaning that as one increased the other decreased. Future work should look to see whether this effect also applies to these actuators, to what degree, and whether this can be harnessed in some way to create more specialized versions of the actuator for different tasks.
3. Explore whether the use of mandrel-coiled TCAs could improve the contractive range of the actuator. Because the actuator cannot meet both torque and range of motion requirements at the same time, it could be beneficial to use a form of TCA that has improved stroke capabilities [54] to increase the radial arm required to achieve the desired range of motion, possibly making it feasible to scale the number of actuators to be able to meet both requirements at once. While mandrel-coiling typically reduces the force output of TCAs, increasing their contractive range and adding more pairs of TCAs to the actuator could enable the actuator to achieve all therapeutic requirements at once, making it more desirable for incorporation into future wearable devices.

4. Use better performing components to enable better performance and minimize error. As mentioned in Section 4.4, the power control circuitry limited the amount of power that could be applied to a TCA pair. Increasing the power running through the system would both increase contraction speed, by increasing the amount of heat being applied at any one time, potentially increasing the efficiency of the system, if the trends found in Sections 4.1.7 and 4.4.12 continue. Increasing the speed would also allow for larger gear reductions to be employed while still allowing the actuator to surpass clinical contraction period requirements, increasing the torque produced and making the actuator able to meet the required torque, range of motion, and contraction period simultaneously. As well, the clock rate of the microcontroller used to execute the force controller should be increased. This would enable a higher sample rate and lower PWM cycle time, allowing the system to better gauge its progression to a setpoint and more accurately deliver power to the system, reducing the total error. In tandem, these would likely decrease the overshoot and settling error even further, making the FF controller even more viable for force-feedback control. Doing either or both of these things would increase the performance of the actuation system, making it even more viable for use in wearable therapeutic devices.

5. Similarly, the inaccuracies between simulated and experimental values found in Chapter 5 could have resulted in part due to modelling error. This could have come from using too-simplistic modelling methods for regressing the force–temperature relationship of the actuator from experimental results, or it could have been due to measurement errors introduced
from sensors that had been calibrated, but not properly assessed and verified. In the case of the former, different regression techniques should be used to determine which is best suited to this kind of data, and used in subsequent efforts. To remove the possibility of the latter, a comprehensive sensor assessment should always be employed after calibration. If the optimal regression method was found and employed, and the possibility of measurement error removed, the simulation error could be minimized in future work.

6. Develop and compare more control methods in addition to the two already explored. There are a plethora of control options for TCAs found in the literature, including bang-bang [105], internal-model [106], sliding-mode control [109], adaptive Lyapunov [96], fuzzy [95], neural-network [95], and many other kinds of controllers. While less popular, their performance should be compared to the two already tested, to determine the controller with the least error and fastest response time when used to control the actuator.

7. Implement a form of active cooling to enable rapid cycle time. While the actuation speed in contraction was able to exceed the average clinical requirements, the cooling time was far too high to facilitate repeated exercise motions. Future work should aim to create a small, lightweight cooling system that would be able to address this issue, perhaps using a heat pump [99] or forced air [97, 98] to cool TCAs. In either case, care would have to be taken to not increase the weight, noise, power consumption, or rigidity of the system too much, as that would impose limitations on the actuator’s adoption.

The purpose of this thesis was to investigate and characterize the stroke and force production capabilities of the actuator, its maximum actuation speed, and its efficiency in converting electrical to mechanical power. These characterizations were then used to create two force controllers, which were able to control the system using only force feedback. The main objectives of the work, which were to determine the conditions under which the actuator could work for wrist neurorehabilitation, determine its efficiency in doing so, and to implement force control methods to improve performance, were achieved. The number of pairs of TCAs and radial arms required by the actuator were calculated for range of motion and force production requirements, and the overshoot present in the system was limited to 2.6% while the steady state error was limited to
0.94% by a FF controller. Continued work developing active cooling systems, improving efficiency, and improving component quality will be able to improve the viability of this actuator to be incorporated into wearable mechatronic devices for use in robot-assisted wrist neurorehabilitation with stroke patients.
References


REFERENCES


REFERENCES


REFERENCES


Appendix A

Engineering Drawings
A.1 Electrical Components

A.1.1 UNO Pinout
A.1.2 ESP32 Pinout
A.1 Electrical Components

A.1.3 UNO Functional Schematic
A.1.4 ESP32 Functional Schematic
A.1.5 Power Control Board Schematic
A.2 3D Printed Components

A.2.1 IR Sensor Holder
A.2.2 PCB Spacer
A.2.3 Centre Tendon
A.2.4 Side Tendon
A.2.5 Vertical Tendon Holder
A.2.6 Pin Caps

DIMENSIONS ARE IN MM
TOLERANCES: ± 0.1 MM

DESIGNED BY: Vaughan Murphy

Printed on Stratasys Object 260 3D Printer using Vero Clear Material

SOLIDWORKS Educational Product. For Instructional Use Only.

A.2 3D Printed Components
A.2.7 Load Cell Hook
A.2.8 Lateral Tendon Holders
A.3 Traditionally Fabricated Components

A.3.1 Rigid Support Structure
DO NOT SCALE DRAWING
Cross Plank
SHEET 1 OF 1
UNLESS OTHERWISE SPECIFIED:
SCALE: 1:5
WEIGHT:

A A

B B

2

2

1

1

Dimensions are in mm

Constructed from 1" x 3" Pine Planks

Designed by Vaughan Murphy

SOLIDWORKS Educational Product. For Instructional Use Only.
Uses the Attach Structure Cross Planks (x3) and the Actuator Attach Structure Side Pieces (x2).
A.3.2  Load Cell Attach Point
Appendix B

MATLAB Code

B.1 Characterization Experiments Code

B.1.1 Stroke Regression Code

%initialize experimental variables
numPairs = 6;
numDegrees = 10;

heatSquareErr = [];
coolSquareErr = [];
avgHeatSquareErr = [];
avgCoolSquareErr = [];
avgHeatRMSE = [];
avgCoolRMSE = [];

%read in the total data, start regressing
for i = 1:numPairs
    heatfileName = 'pair' + string(i) + 'DisplacementHeatRegression.csv';
    coolfileName = 'pair' + string(i) + 'DisplacementCoolRegression.csv';
    heatregressMat = csvread(heatfileName);
    coolregressMat = csvread(coolfileName);
warning('off', 'all');
for n = 1:numDegrees
  %fit a polynomial of degree n to the data sets
  %x is temperature (col 1), y is stroke (col 2)
  heatPoly = polyfit(heatregressMat(:, 1), heatregressMat(:, 2), n);
  coolPoly = polyfit(coolregressMat(:, 1), coolregressMat(:, 2), n);
  %feed the x values through the regressed polynomial
  heatPolyOut = polyval(heatPoly, heatregressMat(:, 1));
  coolPolyOut = polyval(coolPoly, coolregressMat(:, 1));
  %find the squared errors for that degree polynomial
  heatPolyErr = heatPolyOut - heatregressMat(:, 2);
  heatPolyErr = heatPolyErr.*heatPolyErr;
  coolPolyErr = coolPolyOut - coolregressMat(:, 2);
  coolPolyErr = coolPolyErr.*coolPolyErr;
  %sum square errors
  heatSquareErr(i, n) = sum(heatPolyErr);
  coolSquareErr(i, n) = sum(coolPolyErr);
  %getting the mean square error
  avgHeatSquareErr(i, n) = heatSquareErr(i, n)/length(heatPolyErr);
  avgCoolSquareErr(i, n) = coolSquareErr(i, n)/length(coolPolyErr);
  %getting the root mean square error
  avgHeatRMSE(i, n) = sqrt(avgHeatSquareErr(i, n));
  avgCoolRMSE(i, n) = sqrt(avgCoolSquareErr(i, n));
end
warning('on', 'all');
end
%now go through and get the average errors across all 6 models
totalHeatSE = zeros(10, 1);
totalHeatASE = zeros(10, 1);
totalHeatRMSE = zeros(10, 1);
totalCoolSE = zeros(10, 1);
totalCoolASE = zeros(10, 1);
totalCoolRMSE = zeros(10, 1);
for n = 1:numDegrees
    for i = 1:numPairs
        totalHeatSE(n) = totalHeatSE(n) + heatSquareErr(i, n);
        totalHeatASE(n) = totalHeatASE(n) + avgHeatSquareErr(i, n);
        totalHeatRMSE(n) = totalHeatRMSE(n) + avgHeatRMSE(i, n);
        totalCoolSE(n) = totalCoolSE(n) + coolSquareErr(i, n);
        totalCoolASE(n) = totalCoolASE(n) + avgCoolSquareErr(i, n);
        totalCoolRMSE(n) = totalCoolRMSE(n) + avgCoolRMSE(i, n);
    end
end

hold on;
plot(1:numDegrees, totalHeatRMSE, 'red');
plot(1:numDegrees, totalCoolRMSE, 'blue');
title('Degree vs Average RMSE for Displacement');
xlabel('Degree');
ylabel('Average RMSE Across All Pairs');
legend('Heating', 'Cooling');

% now let's plot that to figure out which degree is the best suited
% now to actually put the coefficients into a table
%change hotChosen and coolChosen to whichever performs 'best'
hotChosen = 1;
hotCoeffs = [];
coolChosen = 1;
coolCoeffs = [];
warning('off', 'all');
for i = 1:numPairs
    %read in values again
    heatfileName = 'pair' + string(i) + 'DisplacementHeatRegression.csv';
    coolfileName = 'pair' + string(i) + 'DisplacementCoolRegression.csv';
    heatregressMat = csvread(heatfileName);
    coolregressMat = csvread(coolfileName);
    %regress according to the chosen polynomials
    hotCoeffs(i, :) = polyfit(heatregressMat(:, 1), heatregressMat(:, 2), hotChosen);
    coolCoeffs(i, :) = polyfit(coolregressMat(:, 1), coolregressMat(:, 2), coolChosen);
    %output each as a figure (in heating), show the regression line as well
    fig = figure;
    hold on;
    scatter(heatregressMat(:, 1), heatregressMat(:, 2), 10, 'green', 'filled');
    xVals = linspace(25, 90, 130);
    yVals = polyval(hotCoeffs(i, :),xVals);
    plot(xVals, yVals, 'blue');
    mainTitle = 'Regression for Activation Level '+string(i)+ ' in Heating';
    title(mainTitle);
    xTitle = 'Temperature ('+string(char(176))+'C)';
    xlabel(xTitle);
    yLabel = legend('Data Points', 'Regression');
    set(legger, 'Location', 'northwest');
hold off;
newFileName = 'regressionPair'+string(i)+'HeatCurve.png';
saveas(fig, newFileName);

%output each as a figure (in cooling), show the regression line as well
fig = figure;
hold on;
scatter(coolregressMat(:, 1), coolregressMat(:, 2), 10, 'green', 'filled');
xVals = linspace(25, 90, 130);
yVals = polyval(coolCoeffs(i, :),xVals);
plot(xVals, yVals, 'blue');
mainTitle = 'Regression for Activation Level '+string(i)+ ' in Cooling';
title(mainTitle);
xTitle = 'Temperature ('+string(char(176))+')';
xlabel(xTitle);
ylabel('Displacement (mm)');
legger = legend('Data Points', 'Regression');
set(legger, 'Location', 'northwest');
hold off;
newFileName = 'regressionPair'+string(i)+'CoolCurve.png';
saveas(fig, newFileName);
end
warning('on', 'all');

B.1.2 Force Regression Code

%setup experimental variables
numPairs = 6;
numDegrees = 10;
heatSquareErr = [];
coolSquareErr = [];

hold off;
newFileName = 'regressionPair'+string(i)+'HeatCurve.png';
saveas(fig, newFileName);

%output each as a figure (in cooling), show the regression line as well
fig = figure;
hold on;
scatter(coolregressMat(:, 1), coolregressMat(:, 2), 10, 'green', 'filled');
xVals = linspace(25, 90, 130);
yVals = polyval(coolCoeffs(i, :),xVals);
plot(xVals, yVals, 'blue');
mainTitle = 'Regression for Activation Level '+string(i)+ ' in Cooling';
title(mainTitle);
xTitle = 'Temperature ('+string(char(176))+')';
xlabel(xTitle);
ylabel('Displacement (mm)');
legger = legend('Data Points', 'Regression');
set(legger, 'Location', 'northwest');
hold off;
newFileName = 'regressionPair'+string(i)+'CoolCurve.png';
saveas(fig, newFileName);
end
warning('on', 'all');
B.1 Characterization Experiments Code

avgHeatSquareErr = [];
avgCoolSquareErr = [];
avgHeatRMSE = [];
avgCoolRMSE = [];
%read in the total data, start regressing
for i = 1:numPairs
    heatfileName = 'pair' + string(i) + 'ForceHeatRegression.csv';
    coolfileName = 'pair' + string(i) + 'ForceCoolRegression.csv';
    heatregressMat = csvread(heatfileName);
    coolregressMat = csvread(coolfileName);
    warning('off', 'all');
    for n = 1:numDegrees
        %fit a polynomial of degree n to the data set
        %x is temperature (col 1), y is force (col 2)
        heatPoly = polyfit(heatregressMat(:, 1), heatregressMat(:, 2), n);
        coolPoly = polyfit(coolregressMat(:, 1), coolregressMat(:, 2), n);
        %feed the x values through the regressed polynomial
        heatPolyOut = polyval(heatPoly, heatregressMat(:, 1));
        coolPolyOut = polyval(coolPoly, coolregressMat(:, 1));
        %find the squared errors for that degree polynomial
        heatPolyErr = heatPolyOut - heatregressMat(:, 2);
        coolPolyErr = coolPolyOut - coolregressMat(:, 2);
        %sum square errors
        heatSquareErr(i, n) = sum(heatPolyErr);
        coolSquareErr(i, n) = sum(coolPolyErr);
        %getting the mean square error
        avgHeatSquareErr(i, n) = heatSquareErr(i, n)/length(heatPolyErr);
avgCoolSquareErr(i, n) = coolSquareErr(i, n)/length(coolPolyErr);

%getting the root mean square error
avgHeatRMSE(i, n) = sqrt(avgHeatSquareErr(i, n));
avgCoolRMSE(i, n) = sqrt(avgCoolSquareErr(i, n));
end

warning('on', 'all');

end

%now go through and get the average errors across all 6 models
totalHeatSE = zeros(10, 1);
totalHeatASE = zeros(10, 1);
totalHeatRMSE = zeros(10, 1);
totalCoolSE = zeros(10, 1);
totalCoolASE = zeros(10, 1);
totalCoolRMSE = zeros(10, 1);
for n = 1:numDegrees
    for i = 1:numPairs
        totalHeatSE(n) = totalHeatSE(n) + heatSquareErr(i, n);
        totalHeatASE(n) = totalHeatASE(n) + avgHeatSquareErr(i, n);
        totalHeatRMSE(n) = totalHeatRMSE(n) + avgHeatRMSE(i, n);
        totalCoolSE(n) = totalCoolSE(n) + coolSquareErr(i, n);
        totalCoolASE(n) = totalCoolASE(n) + avgCoolSquareErr(i, n);
        totalCoolRMSE(n) = totalCoolRMSE(n) + avgCoolRMSE(i, n);
    end
end

totalHeatSE = totalHeatSE./numPairs;
totalHeatASE = totalHeatASE./numPairs;
totalHeatRMSE = totalHeatRMSE./numPairs;
totalCoolSE = totalCoolSE./numPairs;
totalCoolASE = totalCoolASE./numPairs;
totalCoolRMSE = totalCoolRMSE./numPairs;

%now let's plot that to figure out which degree is the best suited
hold on;
plot(1:numDegrees, totalHeatRMSE, 'red');
plot(1:numDegrees, totalCoolRMSE, 'blue');
title('Degree vs Average RMSE for Displacement in Cooling');
xlabel('Degree');
ylabel('Average RMSE Across All Pairs');
legend('Heating', 'Cooling');

%now to actually put the coefficients into a table
%Change the hot chosen and cool chosen based on 'best' performing polys
hotChosen = 3;
hotCoeffs = [];
coolChosen = 2;
coolCoeffs = [];
warning('off', 'all');

for i = 1:numPairs
    heatfileName = 'pair' + string(i) + 'ForceHeatRegression.csv';
    coolfileName = 'pair' + string(i) + 'ForceCoolRegression.csv';
    heatregressMat = csvread(heatfileName);
    coolregressMat = csvread(coolfileName);
    hotCoeffs(i, :) = polyfit(heatregressMat(:, 1), heatregressMat(:, 2), hotChosen);
    coolCoeffs(i, :) = polyfit(coolregressMat(:, 1), coolregressMat(:, 2), coolChosen);
    %output each as a figure (in heating), show the regression line as well
    fig = figure;
    hold on;
    scatter(heatregressMat(:, 1), heatregressMat(:, 2), 10, 'green', 'filled');
xVals = linspace(25, 90, 130);
yVals = polyval(hotCoeffs(i, :), xVals);
plot(xVals, yVals, 'blue');
mainTitle = 'Regression for Activation Level '+string(i)+ ' in Heating';
title(mainTitle);
xTitle = 'Temperature ('+string(char(176))+')\textdegree C';
xlabel(xTitle);
ylabel('Force (g)');
legger = legend('Data Points', 'Regression');
set(legger, 'Location', 'northwest');
hold off;

%output each as a figure (in cooling), show the regression line as well
fig = figure;
hold on;
scatter(coolregressMat(:, 1), coolregressMat(:, 2), 10, 'green', 'filled');
xVals = linspace(25, 90, 130);
yVals = polyval(coolCoeffs(i, :),xVals);
plot(xVals, yVals, 'blue');
mainTitle = 'Regression for Activation Level '+string(i)+ ' in Cooling';
title(mainTitle);
xTitle = 'Temperature ('+string(char(176))+')\textdegree C';
xlabel(xTitle);
ylabel('Force (g)');
legger = legend('Data Points', 'Regression');
set(legger, 'Location', 'northwest');
hold off;
end
warning('on', 'all');
B.2 Force Control Experiment Code

B.2.1 Thermal Constant Calculation Code

%This script gets the constant values required for the heating function
%Get the time constant for the heating functions

timeConst = [];
thermResist = [];
pairNum = 1; %change as required

for trialNum = 1:50
    endBit = string(trialNum) + 'ForceHeat.csv';
    fileName = 'pair' + string(pairNum) + 'Trial' + string(endBit);
    csvMatrix = csvread(fileName);
    isNZ = (csvMatrix(:, 1)~=0);
    csvMatrix = csvMatrix(isNZ, :);
    csvMatrix(:, 1) = csvMatrix(:, 1) - csvMatrix(1, 1);
    %the second is the temperature, get time constant
    initTemp = csvMatrix(1, 2);
    peakTemp = max(csvMatrix(:, 2));
    temp63 = 0.63*(peakTemp-initTemp) + initTemp;
    [minVal, closeIndex] = min(abs(csvMatrix(:, 2)-temp63));
    timeConst(trialNum, 1) = csvMatrix(closeIndex, 1);

    %Get the thermal resistances as well
    maxTemp = max(csvMatrix(:, 2));
    minTemp = min(csvMatrix(:, 2));
    isNZPow = (csvMatrix(:, 4)~=0);
    powMatrix = csvMatrix(isNZPow, 4);
    avgPow = mean(powMatrix);
    thermResist(trialNum, 1) = (maxTemp-minTemp)/avgPow;
end
B.2 Force Control Experiment Code

B.2.2 System Identification Code

%This is where we're going to be identifying the systems used in the
%control systems
%Requires use of the system identification toolbox
%Needs to be run for ffR2.slx to work, uses calculated values for inverses
pairNum = 3; %change to get desired TFs, 1, 2, 3 etc.
sampleRate = 0.11;
%use the heat coefficients found in chapter 4
forceHeatCoeffs = [-9.11030805460331e-07, 0.000239362695829137,
-0.0243346615873439, 1.20890332578751, -26.9597437596095,
245.169057330005; -9.43321076042527e-06, 0.00251043064941403,
-0.255030115365153, 12.2850600088681, -272.938449430584,
2266.0068361139; -1.185831995909688e-05, 0.00328598618659597,
-0.34878064775836, 17.6009420276764, -410.853547768584,
3584.50632769282; -9.48922028586148e-06, 0.00268289699315372,
-0.286683133546483, 14.3226343698246, -318.804077813848,
2548.6242212278; -7.29258585700275e-06, 0.0018572640821784,
-0.174067752063301, 7.07071498282769, -89.3172978140658,
-253.385526517940; -1.37310367184114e-05, 0.00366916259217952,
-0.371124772687944, 17.3832714881347, -345.128499877504,
2183.26144692850];
%generate the polynomials
temperature = linspace(25, 85, 61);
forceVec = [];
for i=1:6
    forceVec(:, i) = polyval(forceHeatCoeffs(i, :), temperature);
end
chosenForce = forceVec(:, pairNum);
temperature = temperature';
%that'll be used to make the function for force & temp

%use the formula $T = T_i \cdot e^{-t/tc} + PR(1-e^{-t/tc})$ to get the relationship
%between temperature and power over a 2.5 second input, before it falls back
%down to 30 degrees celsius

%use the calculated values from thermalResistanceCalcs excel sheet for the
%resistance and time constant values (which calculated values based off
%force characterization data)

thermalResist = [2.261877368; 1.473042; 1.25761; 1.152654; 0.890474; 0.817327;];
timeConstant = [7.48; 13.1406; 17.0698; 22.3168; 31.2466; 37.9962;];
maxPow = 80*pairNum; %maximum allowable power in system
initialTemp = 25;
tr = thermalResist(pairNum)*1.25; %factor needs removing eventually
tc = timeConstant(pairNum);
time = 0;
%get calculated values
calcTemp = [];
powVec = [];
timeVec = [];
flag = 1;
i = 0;
while flag == 1
    i = i+1;
    %23 steps required to hit a 2.5 second step input
    if i == 24
        time = 0;
    end
    if i < 24
powVec(i) = maxPow;
initialTemp = 25;

else
    powVec(i) = 0;
    initialTemp = calcTemp(23);
end
endNum = (powVec(i)*tr*(1-exp((-1*time)/tc)))
calcTemp(i) = (initialTemp*exp((-1*time)/tc)) + endNum;
if i > 1
    timeVec(i) = timeVec(i-1) + 0.11;
else
    timeVec(i) = 0;
end
if i > 24 && calcTemp(i) < 30
    flag = 0;
end
time = time+0.11;
end
powVec = powVec';
calcTemp = calcTemp';

%now to find the inverses for the FF controller
%make sure all coefficients are properly input here and in chapter
%first set of TF coefficients
tF1N = [311 367.96];
tF1D = [1 130.3 134.9];
pT1N = [0.1966 0.3707 1.729];
pT1D = [1 0.2338 5.62 -1.309];
%second set of TF coefficients
tF2N = [2220 -245.7 6402];
tF2D = [1 64.23 494 174.3 1191];
pT2N = [0.1328 0.2193];
pT2D = [1 1.7 0.1021];

%third set of TF coefficients
 tF3N = [1835 1932 4308];
 tF3D = [1 254.6 245 599];
 pT3N = [0.05251 -0.02736 0.4945];
 pT3D = [1 0.05077 6.122 0.3108];

% To get TF^-1, just flip numerator and denominator
% Ref: https://www.cds.caltech.edu/~murray/courses/cds101/fa03/caltech
%/am03_ch6-1nov03.pdf
%This will of course result in improper TFs
%Going to be resolving this using residue approximation theorem
% Ref: http://www.math.uwaterloo.ca/~kmorris/Preprints/Curtain_Morris_final.pdf
%This uses Cauchy's residue theorem to partial fraction expand improper TFs

%get the residues for the first set
[rtf1, ptf1, ktf1] = residue(tF1D, tF1N);
[rpt1, ppt1, kpt1] = residue(pT1D, pT1N);

%get the residues for the second set
[rtf2, ptf2, ktf2] = residue(tF2D, tF2N);
[rpt2, ppt2, kpt2] = residue(pT2D, pT2N);

%get the residues for the third set
[rtf3, ptf3, ktf3] = residue(tF3D, tF3N);
[rpt3, ppt3, kpt3] = residue(pT3D, pT3N);

warning('off', 'all');

%now approximations will take the form G(s) = sum(rtfn(i)/[s-ptfn(i)]) + H(s)
%for first activation level
for i = 1:length(rtf1)
    num = rtf1(i);
    temp = -1*ptf1(i);
    den = [1 temp];
    expanseTF1(i) = tf(num, den);
end
for i = 1:length(rpt1)
    num = rpt1(i);
    temp = -1*ppt1(i);
    den = [1 temp];
    expansePT1(i) = tf(num, den);
end
s = tf('s');
polyExpanseTF1 = ktf1(1)*s + ktf1(2);
simpleExpanseTF1 = expanseTF1(1) + polyExpanseTF1;
polyExpansePT1 = kpt1(1)*s + kpt1(2);
simpleExpansePT1 = expansePT1(1) + expansePT1(2) + polyExpansePT1;
%for second activation level
for i = 1:length(rtf2)
    num = rtf2(i);
    temp = -1*ptf2(i);
    den = [1 temp];
    expanseTF2(i) = tf(num, den);
end
for i = 1:length(rpt2)
    num = rpt2(i);
    temp = -1*ppt2(i);
    den = [1 temp];
B.2 Force Control Experiment Code

```matlab
expansePT2(i) = tf(num, den);
end

polyExpanseTF2 = ktf2(1)*s*s + ktf2(2)*s + ktf2(3);
simpleExpanseTF2 = expanseTF2(1)+expanseTF2(2) + polyExpanseTF2;
polyExpansePT2 = kpt2(1)*s + kpt2(2);
simpleExpansePT2 = expansePT2(1)+polyExpansePT2;

% for third activation level
for i = 1:length(rtf3)
    num = [rtf3(i)];
    temp = -1*ptf3(i);
    den = [1 temp];
    expanseTF3(i) = tf(num, den);
end

for i = 1:length(rpt3)
    num = [rpt3(i)];
    temp = -1*ppt3(i);
    den = [1 temp];
    expansePT3(i) = tf(num, den);
end

warning('on', 'all');

polyExpanseTF3 = ktf3(1)*s + ktf3(2);
simpleExpanseTF3 = expanseTF3(1)+expanseTF3(2) + polyExpanseTF3;
polyExpansePT3 = kpt3(1)*s + kpt3(2);
simpleExpansePT3 = expansePT3(1)+expansePT3(2)+polyExpansePT3;

% can now go through and put all of these into subsystem blocks in simulink
% get the total inverted TF so we can do the inverse laplace for the FF
totalITF(1) = simpleExpansePT1*s*simpleExpanseTF1;
totalITF(2) = simpleExpansePT2*s*simpleExpanseTF2;
totalITF(3) = simpleExpansePT3*s*simpleExpanseTF3;
```
ke = [2, 3.5, 6];
syms s t;
for i = 1:3
  \textit{get the time domain of the top branch}
  ffCont = tf([1 ke(i)^2],[1 2*ke(i) ke(i)^2]);
  topTF = totalITF(i)*ffCont;
  num = poly2sym(topTF.numerator{1, 1}, s);
  den = poly2sym(topTF.denominator{1, 1}, s);
  temp = num/den;
  timeDomain(i, 1) = ilaplace(temp);
  setDig = digits(3);
  timeDomain(i, 1) = vpa(timeDomain(i));
  timeDomain(i, 1) = simplify(expand(timeDomain(i)));% now for the controller placed on the input
  num = poly2sym(ffCont.numerator{1, 1}, s);
  den = poly2sym(ffCont.denominator{1, 1}, s);
  temp = num/den;
  ffTime(i, 1) = ilaplace(temp);
  setDig = digits(3);
  ffTime(i, 1) = vpa(ffTime(i));
  ffTime(i, 1) = simplify(expand(ffTime(i)));%} now for the controller placed on the input
end

B.2.3 Controller Performance Calculation Code

% this is the spss-readying calculations for the force-control data
% we will be looking into OS, Rise Time, Settling Value, first peak time, and TC
% sort out if it's PID or FF
isPID = 1;
epithet = '';
if isPID == 1
    epithet = 'PID';
else
    epithet = 'FF';
end

targets = [200, 400, 550];

%initialize the values
os = []; %OS defined as max value expressed as a percent > target value
riseTime = []; %rise time defined as time between 10% and 20% value
settleVal = []; %settle value defined as average oscillatory value
tc = []; %time constant defined as time it takes to get to 63% target value
fp = []; %first peak time defined as time it takes before it starts oscillating

val10 = [0.1*200, 0.1*400, 0.1*550];
val90 = [0.9*200, 0.9*400, 0.9*550];
val63 = [0.63*200, 0.63*400, 0.63*550];

%go through and let's calculate things
for i=1:3
    for n=1:15
        endBit = 'Ramping'+string(epithet)+'.csv';
        rampName = 'pair'+string(i)+'trial'+string(n)+string(endBit);
        rampMatrix = csvread(rampName);
        endBit = 'Oscillating'+string(epithet)+'.csv';
        oscName = 'pair'+string(i)+'trial'+string(n)+string(endBit);
        oscMatrix = csvread(oscName);
        endBit = 'Total'+string(epithet)+'.csv';
        totalName = 'pair'+string(i)+'trial'+string(n)+string(endBit);
        totalMatrix = csvread(totalName);
        peakVal = max(totalMatrix(:, 2));
    end
end
os(n, i) = (peakVal-targets(i))/targets(i);
%calculate settling value
averageVal = mean(oscMatrix(:, 2));
settleVal(n, i) = averageVal;
%calculate time constant
[tcValue, tcIndex] = min(abs(rampMatrix(:, 2)-val63(i)));
tc(n, i) = rampMatrix(tcIndex, 1);
%calculate rise time
[startVal, startIndex] = min(abs(rampMatrix(:, 2)-val10(i)));
[endVal, endIndex] = min(abs(rampMatrix(:, 2)-val90(i)));
riseTime(n, i) = rampMatrix(endIndex, 1) - rampMatrix(startIndex, 1);
%get first peak time
fp(n, i) = rampMatrix(end, 1);
end
end
%now output to SPSS ready csvs
%Format goes [actNum|Trial Num|Pair 1 Val|Pair 2 Val|Pair 3 Val]
osCSV = [];
rtCSV = [];
fpCSV = [];
svCSV = [];
tcCSV = [];
count = 0;
for i = 1:3
  for n=1:5
    count = count + 1;
osCSV(count, 1:5) = [i, n, os(count, 1),
                         os(count, 2), os(count, 3)];
rtCSV(count, 1:5) = [i, n, riseTime(count, 1),
B.2 Force Control Experiment Code

\[
\text{riseTime(count, 2), riseTime(count, 3)];}
\]

\[
\text{fpCSV(count, 1:5) = [i, n, fp(count, 1), fp(count, 2),}
\]

\[
\text{fp(count, 3)];}
\]

\[
\text{svCSV(count, 1:5) = [i, n, settleVal(count, 1),}
\]

\[
\text{settleVal(count, 2), settleVal(count, 3)];}
\]

\[
\text{tcCSV(count, 1:5) = [i, n, tc(count, 1), tc(count, 2),}
\]

\[
\text{tc(count, 3)];}
\]

end

end

overshootName = 'overshoot'+string(epithet)+'ANOVA.csv';
csvwrite(overshootName, osCSV);

riseTimeName = 'riseTime'+string(epithet)+'ANOVA.csv';
csvwrite(riseTimeName, rtCSV);

firstPeakName = 'firstPeak'+string(epithet)+'ANOVA.csv';
csvwrite(firstPeakName, fpCSV);

settleValName = 'settleValue'+string(epithet)+'ANOVA.csv';
csvwrite(settleValName, svCSV);

timeConstantName = 'timeConstant'+string(epithet)+'ANOVA.csv';
csvwrite(timeConstantName, tcCSV);

B.2.4 Simulink Performance Calculation Code

%This is where we're going to be manually calculating the rise time and
%other values for the response of the control systems from Simulink
dataset = [out.pair1Step, out.pair2Step, out.pair3Step];
stepVals = [200, 400, 550];
%calculate rise time (time between 10 and 90% of final value) and time
%constant (time it takes to get to 63% of final value)
riseTime = [];
timeConst = [];
numPairs = 3;
for i=1:numPairs
    tcTarg = 0.63*stepVals(i);
    lowRT = 0.1*stepVals(i);
    highRT = 0.9*stepVals(i);
    % find closest indices
    [tcMin, tcIndex] = min(abs(dataset(i).data - tcTarg));
    [rtLow, rtlIndex] = min(abs(dataset(i).data - lowRT));
    [rtHigh, rthIndex] = min(abs(dataset(i).data - highRT));
    % now get the time values, put into their arrays
    timeConst(i) = dataset(i).time(tcIndex);
    riseTime(i) = dataset(i).time(rthIndex) - dataset(i).time(rtlIndex);
end
% calculate overshoot (%), peak value, and time
overshoot = [];
peakVal = zeros(1, numPairs);
peakTime = zeros(1, numPairs);
for i = 1:numPairs
    for n = 1:length(dataset(i).data)
        if dataset(i).data(n) > peakVal(i)
            peakVal(i) = dataset(i).data(n);
            peakTime(i) = dataset(i).time(n);
        end
    end
    overshoot(i) = (peakVal(i) - stepVals(i))/stepVals(i);
end
% calculate settling time and value
% Defining settling time as time it takes to get within 2% stepValue after peaking
settleTime = []; 
settleValue = []; 
threshTime = []; 
for i=1:numPairs 
    toggle = 0; 
    timeIndex = 1; 
    settleTime(i) = 0; 
    settleValue(i) = 0; 
    threshTime(i) = 0; 
    for n=1:length(dataset(i).data) 
        diffPerc = abs(dataset(i).data(n) - stepVals(i))/stepVals(i); 
        %first instance past peak time it gets within 2% of final val 
        if (toggle == 0 && dataset(i).time(n) > peakTime(i) && diffPerc <= 0.02) 
            toggle = 1; 
            settleTime(i) = dataset(i).time(n); 
            timeIndex = n; 
        end 
    end 
    %if not set, just get time to within 2% of final val 
    if (threshTime(i) < 1) 
        for n=1:length(dataset(i).data) 
            difference = abs(dataset(i).data(n) - stepVals(i)); 
            if (difference <= 5) 
                threshTime(i) = dataset(i).time(n); 
                if(settleTime(i) < 1) 
                    settleTime(i) = threshTime(i); 
                    timeIndex = n; 
                end 
            break; 
    end
end

end

end

% now to average the settling values

elong = (length(dataset(i).time) - timeIndex) + 1; % number of elements

summer = sum(dataset(i).data(timeIndex:length(dataset(i).data)));

settleValue(i) = summer/elong; end
Appendix C

Arduino Code

C.1 Sensor Calibration Code

C.1.1 Stroke Sensor Calibration Code

/*
 * Author: Vaughan Murphy
 * Purpose: Test and calibrate the distance measurement capabilities
 * of the IR sensor used throughout the Woven Bi-Pennate
 * Actuator characterization.
 */

//initialize global variables
int irPin = 35;
int counter = 0;
double distance = 0;
double averageDistance = 0;

void setup() {
    //set up required bits
    pinMode(irPin, INPUT);
C.1 Sensor Calibration Code

```cpp
Serial.begin(9600);
}

void loop() {
    // get 1000 readings, average, then display
    if(counter < 1000){
        distance = distance + getDistance();
        counter++;
    }
    else{
        Serial.print("The average IR sensor value is: ");
        averageDistance = distance/1000.0;
        Serial.println(averageDistance);
        while(true){
            // spin while waiting for processor to be reset for next trial
        }
    }
}

double getDistance(){
    // Returns the raw value found by the sensor
    double val = analogRead(infraredPin);
    return val;
}

C.1.2 Force Sensor Calibration Code

// Requires the HX711.h library, published by Bogdan Necula
// under the MIT License.
/*
Author: Vaughan Murphy

Purpose: Test and calibrate the force measurement capabilities of the load cell used throughout the Woven Bi-Pennate Actuator characterization.

```cpp
#include "HX711.h"

//initialize global variables
int loadDOut = 35;
int loadSCK = 32;
int counter = 0;
double calibration = 1;
double force = 0;
double averageForce;
HX711 scale;

void setup() {
    Serial.begin(9600);
    //set up scale
    scale.begin(loadDOut, loadSCK);
    scale.set_scale(calibration);
    scale.tare();
}

void loop() {
    //get 1000 readings, average, then display
    if(counter < 1000){
        if(scale.is_ready()){
            force = force + getForce();
        }
    }
```
C.1 Sensor Calibration Code

} 
    counter++; 
} 
else{
    Serial.print("The sensor reading is: "); 
    averageForce = force/1000.0; 
    Serial.println(averageForce); 
    while(true){
        //spin while waiting for processor to be reset for next trial
    }
}
}

double getForce(){
    //Returns the raw value found by the sensor
    double reading = scale.read() * 1.0; 
    return reading; 
}

C.1.3 Temperature Sensor Calibration Code

//Requires the max6675.h library, published by Adafruit Inc.
//under the BSD License.
/*
Author: Vaughan Murphy
Purpose: Test and calibrate the temperature measurement capabilities
of the thermocouples used throughout the Woven Bi-Pennate
Actuator characterization.
*/
#include "max6675.h"

//initialize global variables, set up thermocouple amplifiers
int thermoDO = 5;
int thermoCS = 10;
int thermoCLK = 7;
int counter = 0;
double averageTemp = 0;
double totalTemp = 0;
MAX6675 thermocouple(thermoCLK, thermoCS, thermoDO);

void setup() {
  //begin serial output
  Serial.begin(9600);
  // wait for MAX chip to stabilize
  delay(500);
}

void loop() {
  // basic readout test, just print the current temp
  if(counter < 10){
    double temp = thermocouple.readCelsius();
    totalTemp = totalTemp + temp;
    counter++;
  }
  else{
    //output average reading
    averageTemp = totalTemp/10.0
    Serial.print("Average sensor reading is: ");
  }
}
```cpp
Serial.println(averageTemp);
//reset values that need it
counter = 0;
averageTemp = 0;
totalTemp = 0;
```

C.1.4 Power Sensor Calibration Code

//Requires the Wire.h library, published by Arduino Inc. under
//the Creative Commons Attribution-ShareAlike 3.0 License.
//Requires the INA219 device driver library, published by Adafruit Inc.
//under the MIT License.
/*
Author: Vaughan Murphy
Purpose: Test and calibrate the power measurement capabilities
of the INA219B power sensors used throughout the Woven Bi-Pennate
Actuator characterization.
*/
#include <Wire.h>
#include <Adafruit_INA219.h>

//initialize ina objects
Adafruit_INA219 ina[] = {
    Adafruit_INA219(0x40),
    Adafruit_INA219(0x41),
    Adafruit_INA219(0x44),
    Adafruit_INA219(0x45),
    Adafruit_INA219(0x42),
};
Adafruit_INA219(0x46),
;

void setup() {
    // use the begin function to get them running
    Serial.begin(9600);
    for(int i = 0; i < 6; i++){
        ina[i].begin();
    }
}

int counter = 0;
float totalPow = 0;

void loop() {
    //get 1000 readings, then average, then display
    if(counter < 1000){
        for(int i = 0; i < 6; i++){
            float tempPower = getPower(i);
            totalPow += tempPower;
        }
        counter++;
    }
    else{
        float averagePow = totalPow/1000.0
        Serial.print("Average sensor reading is: ");
        Serial.println(averagePow);
        while(true){
            //spin the processor until it gets reset
        }
    }
}
```c

double getPower(int index){
    float busVoltage = 0;
    float current = 0; //will be measured in milliamps
    //get the voltage
    busVoltage = ina[index].getBusVoltage_V();
    //get the current, no correction because of chosen resistance
    current = ina[index].getCurrent_mA();
    //calculate the power using Ohm's law
    float power = busVoltage*(current/1000);
    return power;
}
```

C.2 Pilot Study Code

C.2.1 Force Production Pilot Study Code

//Requires the HX711.h library, published by Bogdan Necula
//under the MIT License.

//Requires the max6675.h library, published by Adafruit Inc.
//under the BSD License.

/*
Author: Vaughan Murphy
Purpose: Provide preliminary results and values used to determine
sample size, based on the force production of the TCAs
used throughout the Woven Bi-Pennate Actuator characterization.
*/
#include "HX711.h"
#include "max6675.h"

//initialize global variables
int loadDOut = 35;
int loadSCK = 32;
double maxForce = 0;
HX711 scale;

int thermoDO = 5;
int thermoCS = 10;
int thermoCLK = 7;
double maxTemp = 0;
MAX6675 thermocouple(thermoCLK, thermoCS, thermoDO);

void setup() {
    Serial.begin(9600);
    //set up scale
    scale.begin(loadDOut, loadSCK);
    scale.set_scale(calibration);
    scale.tare();
}

void loop() {
    //free running loop, record temperature, force, and maximum values
    double temperature = getTemperature();
    double force = getForce();
    //compare max values
    if(temperature > maxTemp){
maxTemp = temperature;
}
if(force > maxForce){
    maxForce = force;
}

// output values and maximums
Serial.print("T: 	");
Serial.print(temperature);
Serial.print(" 	 F: 	");
Serial.print(force);
Serial.print(" 	 MT: 	");
Serial.print(maxTemp);
Serial.print(" 	 MF: 	");
Serial.println(maxForce)
if(temperature > 80){
    Serial.println("Turn OFF");
}
}

double getForce(){
    // Returns the corrected weight found by the sensor
    // Uses the zero value found experimentally to get the calibration value
    // then divides sensor reading by calibration to get final force,
    // based on zero'd value
    double reading = scale.read() * 1.0;
    double inputVal = reading - 78550.62;
    if(inputVal <= 200){
        calibration = 0;
        return 0.0;
C.2 Pilot Study Code

```cpp

double getTemperature()
{
    double temp = thermocouple.readCelsius();
    temp = (0.9278*temp)+0.1862;
    return temp;
}
```

C.2.2 Stroke Production Pilot Study Code

//Requires the max6675.h library, published by Adafruit Inc.
//under the BSD License.
/*
Author: Vaughan Murphy
Purpose: Provide preliminary results and values used to determine
sample size, based on the force production of the TCAs
used throughout the Woven Bi-Pennate Actuator characterization.
*/

#include "HX711.h"
#include "max6675.h"

//initialize global variables
int irPin = 35;
double maxDist = 0;
double initDist = 0;

int thermoDO = 5;
int thermoCS = 10;
int thermoCLK = 7;
double maxTemp = 0;
MAX6675 thermocouple(thermoCLK, thermoCS, thermoDO);

void setup() {
    Serial.begin(9600);
    //set up scale
    scale.begin(loadDOut, loadSCK);
    scale.set_scale(calibration);
    scale.tare();
}

void loop() {
    //free running loop, record temperature, distance, and maximum values
    double distance = getDisplacement();
    double force = getForce();
    //compare max values
    if(temperature > maxTemp){
        maxTemp = temperature;
    }
    if(distance > maxDist){
        maxDist = distance;
    }
    if (initDist == 0){
        initDist = distance;
    }
//output values and maximums
Serial.print("T: 	");
Serial.print(temperature);
Serial.print("\t D: 	");
Serial.print(distance);
Serial.print("\t MT: 	");
Serial.print(maxTemp);
Serial.print("\t ID: 	");
Serial.print(initDist);
Serial.print("\t MD: 	");
Serial.println(maxDist)
if(temperature > 80){
    Serial.println("Turn OFF");
}

double getDisplacement(){
    double val = analogRead(infraredPin);
    //linearization equation devised via calibration experimentation
    double mm = 61763*pow(val, -1.031);
    return mm;
}

double getTemperature(){
    double temp = thermocouple.readCelsius();
    //linearization equation devised via calibration experimentation
    temp = (0.9278*temp)+0.1862;
    return temp;}
C.3 Characterization Experiment Code

C.3.1 Temperature Reading and Communication Experiment Code

//Requires the SPI.h library, published by Arduino Inc. under
//the Creative Commons Attribution-ShareAlike 3.0 License.
//Requires the max6675.h library, published by Adafruit Inc.
//under the BSD License.

/*
Author: Vaughan Murphy
Purpose: Provide inter board communications between the Arduino UNO
and ESP32 during the experiments which require temperature readings.
*/

#include <SPI.h>
#include <max6675.h>

volatile byte message;
volatile bool received;

//initialize pin values
int sckpin = A0, sopin = A1;
int cspin[] = {2, 3, 4, 5, 6, 7, 8, 9, A2, A3, A4, A5};

//initialize thermocouples
MAX6675 thermocouple[] = {
    MAX6675(sckpin, cspin[0], sopin),
    MAX6675(sckpin, cspin[1], sopin),
    MAX6675(sckpin, cspin[2], sopin),
byte temperature[] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

ISR(SPI_STC_vect){
    // read the spi buffer that we got this thing
    message = SPDR;
    SPDR = temperature[message];
    if (message == 14){
        
    }
//Serial.print("\n");
}
//Serial.print("\t");
//Serial.print(temperature[message]);
received = true;
}

void loop() {
  // put your main code here, to run repeatedly:
  //long startTime = millis();
  for(int i = 0; i < 12; i++){
    temperature[i] = getTemperature(i);
    //Serial.println(temperature[i]);
  }
  //long endTime = millis();
  //int dur = endTime - startTime;
  //Serial.println(dur);
}

byte getTemperature(int i){
  double celsius = thermocouple[i].readCelsius();
  //rectify temperature using formula found during sensor characterization
  //y = 0.9278x + 0.1862, with an R^2 value of 0.9966
  celsius = (0.9278*celsius) + 0.1862;
  byte bitten = (byte)celsius;
  return bitten;
}
C.3.2 Stroke Characterization Experiment Code

//Requires the Wire.h library, published by Arduino Inc. under
//the Creative Commons Attribution-ShareAlike 3.0 License.
//Requires the INA219 device driver library, published by Adafruit Inc.
//under the MIT License.
//Requires the ESP32 DMA SPI library, published by Hideaki Tai
//under the MIT License.

/*
Author: Vaughan Murphy
Purpose: Characterize the stroke production of the
Woven Bi-Pennate Actuator.
*/

#include <ESP32DMASPIMaster.h>
#include <Wire.h>
#include <Adafruit_INA219.h>

//SPI communications variables
ESP32DMASPI::Master master;
static const uint32_t BUFFER_SIZE = 14;
uint8_t* spi_master_tx_buf;
uint8_t* spi_master_rx_buf;

//temperature related global variables
double tempVals[][3]{{
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
}}
\{0, 0, 0\},
\{0, 0, 0\},
\{0, 0, 0\},
\{0, 0, 0\},
\{0, 0, 0\},
\{0, 0, 0\},
\{0, 0, 0\},
\{0, 0, 0\},
\{0, 0, 0\}
};

//power related global variables

double power = 0;
Adafruit_INA219 ina[] = {
    Adafruit_INA219(0x40),
    Adafruit_INA219(0x41),
    Adafruit_INA219(0x44),
    Adafruit_INA219(0x45),
    Adafruit_INA219(0x42),
    Adafruit_INA219(0x46),
};

//infrared (displacement) related variables

int infraredPin = 35;
double distance = 0;

//control related global variables

double targetTemp = 80; //temperature to be maintained

double tempMin = 30; //floor temperature at which point the TCA is cooled
```cpp
int gatePin[] = {33, 26, 23, 18, 17, 4, 25, 27, 19, 5, 16, 2};
long tempTimer = 0, timer = 0;
bool isReached[] = {false, false, false, false, false, false,
false, false, false, false, false, false};
bool maintaining[] = {false, false, false, false, false, false,
false, false, false, false, false, false};

//general operation variables
int trialNum = 0, trialCap = 30; //change as required
bool heating = true; //this dictates if the trial is heating [true] or cooling [false]

//active set related variables, should always initialize as set of zeros
int activeSet[] = {0, 0, 0, 0, 0};
int list[] = {0, 0, 0, 0, 0,
0, 0, 0, 0, 0,
0, 0, 0, 0, 0,
0, 0, 0, 0, 0,
0, 0, 0, 0, 0,
0, 0, 0, 0, 0};
//doing 5 trials per # pairs per actuator, total of 300 trials
//insert list as set of undemarcated active pairs
//i.e. 13 will have pairs 1 and 3 as a trial

void setup() {
    //initialize pins
    for(int i = 0; i < 12; i++){
        pinMode(gatePin[i], OUTPUT);
    }
```

pinMode(infraredPin, INPUT);
Serial.begin(9600);

// begin INA communications
for(int i = 0; i < 6; i++)
    ina[i].begin();

// begin SPI communications
// to use DMA buffer, use these methods to allocate buffer
spi_master_tx_buf = master.allocDMABuffer(BUFFER_SIZE);
spi_master_rx_buf = master.allocDMABuffer(BUFFER_SIZE);

master.setDataMode(SPI_MODE3);  // for DMA, only 1 or 3 is available
// set the frequency to something lower
master.setFrequency(1000);
master.setMaxTransferSize(BUFFER_SIZE);
master.setDMAChannel(1);         // 1 or 2 only
master.setQueueSize(1);          // transaction queue size

master.begin();  // default SPI is HSPI
// set buffer data here
spi_master_tx_buf[0] = 14;
for(int i = 1; i <= 12; i++)
    spi_master_tx_buf[i] = i-1;
//exit SPI initialization

while(!Serial.available()){
    ;//delay program until serial is triggered
}

Serial.println("Begin Stroke Characterization Experiment");

}

void loop() {

    //If the trials have been completed, make sure to kill the program
    if(trialNum >= trialCap){
        Serial.println("End of experiment, trial number exceeded");
        while(true){
            ;//this infinite loop is to trap the controller once the trials are done
        }
    }

    //check which trial we're in right now, or if we're even in one
    bool inTrial = false;
    for(int i = 0; i < 6; i++){
        if(activeSet[i] > 0){
            inTrial = true;
        }
    }

    //if we're not in a trial, let's get in one
    if(!inTrial){
        getActiveSet();
    }

    else{
        //if we are in a trial, make sure to control the heating or cooling process
        //in intervals of 50 ms collect the temperature data
// (delay required for SPI to run)
if(millis() >= (tempTimer + 50)){
    master.transfer(spi_master_tx_buf, spi_master_rx_buf, BUFFER_SIZE);
    // temperature values will be saved in rx_buf, from elements 2-13
    tempTimer = millis();
}

for(int i = 0; i < 12; i++){
    // for every thermocouple, save the new data to the temperature sliding array
    byte receiver = spi_master_rx_buf[i+2];
    tempVals[i][0] = tempVals[i][1];
    tempVals[i][1] = tempVals[i][2];
    tempVals[i][2] = (double)receiver;
    // if it's an outlier, make it instead an average of the previous two values
    // outlier we'll define as 40% different than last two values
    // that *should* never happen
    double tempAvg = (tempVals[i][0]+tempVals[i][1])/2.0;
    double absoluteDiff = abs(tempVals[i][2] - tempAvg);
    if(absoluteDiff > (tempAvg*0.4)){
        tempVals[i][2] = tempAvg;
    }
}

// Then need to output everything
outputData();

// check which direction we're going, apply controls accordingly
if(heating){
    controlHeating();
}
else{
    controlCooling();
}
//this function deciphers which TCAs are part of the
//current trial from the trial list (for control purposes)
void getActiveSet(){
    //get the current list element
    int temp = list[trialNum];
    //increment through each digit to get active TCAs
    activeSet[0] = (int)(temp/100000);
    temp-=activeSet[0]*100000;
    activeSet[1] = (int)(temp/10000);
    temp-=activeSet[1]*10000;
    activeSet[2] = (int)(temp/1000);
    temp-=activeSet[2]*1000;
    activeSet[3] = (int)(temp/100);
    temp-=activeSet[3]*100;
    activeSet[4] = (int)(temp/10);
    temp-=activeSet[4]*10;
    activeSet[5] = (int)(temp);
    //Output that we are starting a trial, and which pairs are active
    Serial.print("Begin Trail ");
    byte tempTrialNum = trialNum + 1;
    Serial.println(tempTrialNum);
    Serial.print("Active Set is [ ");
    Serial.print(activeSet[0]);
    Serial.print(" ");
    Serial.print(activeSet[1]);
Serial.print("", ");
Serial.print(activeSet[2]);
Serial.print("", ");
Serial.print(activeSet[3]);
Serial.print("", ");
Serial.print(activeSet[4]);
Serial.print("", ");
Serial.print(activeSet[5]);
Serial.println("[");
}

//this function controls the heating process of the actuator
void controlHeating(){
    int maintenanceCount = 0; //keeps track of how many tcas are maintaining
    int activeCount = 0; //keeps track of how many tcas are active
    //cycle through the set of active pairs
    for(int i = 0; i < 6; i++){
        if(activeSet[i] > 0){
            //if the current element is greater than zero, apply logic accordingly
            int left = activeSet[i] - 1;
            int right = left + 6;
            int leftCount = 0;
            int rightCount = 0;
            bool leftHeat = true;
            bool rightHeat = true;
            //cycle through last 3 temperature readings
            //increment counts if higher than temperature targets
            //if all 3 are higher, then we can begin cooling
            //This is done to reject any specific spikes
for (int c = 0; c < 3; c++){
    if (tempVals[left][c] >= targetTemp){
        leftCount++;
    }
    if (tempVals[right][c] >= targetTemp){
        rightCount++;
    }
}

// if we're in the maintaining phase and it gets hot, cool it
if (leftCount > 0 && maintaining[left]){
    digitalWrite(gatePin[left], LOW);
}
if (rightCount > 0 && maintaining[right]){
    digitalWrite(gatePin[right], LOW);
}

// if the counts were less than the target temp, turn up the heat
if (leftCount < 3){
    digitalWrite(gatePin[left], HIGH);
}
else if (leftCount == 3){
    // otherwise, need to begin cooling/maintenance
    maintaining[left] = true;
    digitalWrite(gatePin[left], LOW);
}

// same goes for the right side
if (rightCount < 3){
    digitalWrite(gatePin[right], HIGH);
}
else if (rightCount == 3){

maintaining[right] = true;
digitalWrite(gatePin[right], LOW);
}

//if the pair is active, increase the count of active TCAs by two
activeCount += 2;
}
}

//check how many tcas are maintaining temperature right now
for(int i = 0; i < 12; i++){
    if(maintaining[i]){
        maintenanceCount++;
    }
}

//if greater to or equal to the number of active TCAs
//flip the direction and reset variables
if(maintenanceCount >= activeCount){
    heating = false;
    for(int i = 0; i < 12; i++){
        maintaining[i] = false;
    }
}

//this function controls the cooling process of the actuator
void controlCooling(){
    //cycle through every TCA, if they're all below the threshold
    //((set in global) tick over to a new trial
    int coldCount = 0;
    for(int i = 0; i < 12; i++){
```c
void outputData()
{
    //get the distance and powers
    distance = getDisplacement();
    //getting the power from the power function
    power = 0;
    for(int i = 0; i < 6; i++)
    {
        //this function outputs the data to the serial monitor to be transcribed
    }
}
```
```c
//float tempPower = 0
float tempPower = getPower(i);
power += tempPower;
}
if(power < 0){
power = 0;
}

//get time stamp
float curTime = millis()/1000.0;
//output all values, separated by commas. Will go
//[Time Stamp][Distance][Total Power Consumption][Temperature (12 elements)]
//output time stamp
Serial.print(curTime);
Serial.print(",");
//output distance
Serial.print(distance);
Serial.print(",");
//output the power
Serial.print(power);
Serial.print(",");
//output the temperatures and end the line
for(int i = 0; i < 11; i++){
    Serial.print(tempVals[i][2]);
    Serial.print(",");
}
Serial.println(tempVals[11][2]);
//total output will be a 15 element line
```
double getDisplacement(){
    double val = analogRead(infraredPin);
    //linearization equation devised via calibration experimentation
    double mm = 61763*\text{pow}(val, -1.031);
    return mm;
}

double getPower(int index){
    float busVoltage = 0;
    float current = 0; //will be measured in milliamps
    //get the voltage
    busVoltage = ina[index].getBusVoltage_V();
    //get the current
    current = ina[index].getCurrent_mA();
    //calculate the power using Ohm's law
    float power = busVoltage*(current/1000);
    //rectify using calibration function from preliminary experiment
    double newPow = 0.2841*power*power + 0.6191*power + 0.625;
    return newPow;
}

C.3.3 Force Characterization Experiment Code

//Requires the Wire.h library, published by Arduino Inc. under
//the Creative Commons Attribution-ShareAlike 3.0 License.
//Requires the INA219 device driver library, published by Adafruit Inc.
//under the MIT License.
//Requires the ESP32 DMA SPI library, published by Hideaki Tai
//under the MIT License.
//Requires the HX711.h library, published by Bogdan Necula
//under the MIT License.
/*
Author: Vaughan Murphy
Purpose: Characterize the force production of the
Woven Bi-Pennate Actuator.
*/

#include <ESP32DMASPIMaster.h>
#include <Wire.h>
#include <Adafruit_INA219.h>
#include <HX711.h>

//SPI communications variables
ESP32DMASPI::Master master;
static const uint32_t BUFFER_SIZE = 14;
uint8_t* spi_master_tx_buf;
uint8_t* spi_master_rx_buf;

//temperature related global variables
double tempVals[][3]{{
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
},
{0, 0, 0},
{0, 0, 0},
{0, 0, 0}
}

//power related global variables

double power = 0;
Adafruit_INA219 ina[] = {
    Adafruit_INA219(0x40),
    Adafruit_INA219(0x41),
    Adafruit_INA219(0x44),
    Adafruit_INA219(0x45),
    Adafruit_INA219(0x42),
    Adafruit_INA219(0x46),
};

//load cell (force) related variables

int loadCellDout = 35;
int loadCellSck = 32;
double force = 0;
double calibration = 1;
HX711 scale;

//control related global variables

double targetTemp = 80; //temperature to be maintained
double tempMin = 30; //floor temperature at which point the TCA is cooled
int gatePin[] = {33, 26, 23, 18, 17, 4, 25, 27, 19, 5, 16, 2};
long tempTimer = 0, timer = 0;
bool isReached[] = {false, false, false, false, false, false, false, false,
false, false, false, false, false};

bool maintaining[] = {false, false, false, false, false, false, false, false, false, false, false, false};

// general operation variables
int trialNum = 0, trialCap = 30; // change as required
bool heating = true; // this dictates if the trial is heating [true] or cooling [false]

// active set related variables, should always initialize as set of zeros
int activeSet[] = {0, 0, 0, 0, 0};
int list[] = {0, 0, 0, 0, 0,
0, 0, 0, 0, 0,
0, 0, 0, 0, 0,
0, 0, 0, 0, 0,
0, 0, 0, 0, 0,
0, 0, 0, 0, 0}; // doing 5 trials per # pairs per actuator, total of 300 trials
// insert list as set of undemarcated active pairs
// i.e. 13 will have pairs 1 and 3 as a trial

void setup() {
// initialize pins
for(int i = 0; i < 12; i++){
    pinMode(gatePin[i], OUTPUT);
}
Serial.begin(9600);
//begin INA communications
for(int i = 0; i < 6; i++){
    ina[i].begin();
}

//get the load cell set up
scale.begin(loadCellDout, loadCellSck);
scale.set_scale(calibration);
scale.tare();

//begin SPI communications
// to use DMA buffer, use these methods to allocate buffer
spi_master_tx_buf = master.allocDMABuffer(BUFFER_SIZE);
spi_master_rx_buf = master.allocDMABuffer(BUFFER_SIZE);

master.setDataMode(SPI_MODE3); // for DMA, only 1 or 3 is available

//set the frequency to something lower
master.setFrequency(1000);
master.setMaxTransferSize(BUFFER_SIZE);
master.setDMAChannel(1); // 1 or 2 only
master.setQueueSize(1); // transaction queue size

master.begin(); // default SPI is HSPI

// set buffer data here
spi_master_tx_buf[0] = 14;
for(int i = 1; i <= 12; i++){
    spi_master_tx_buf[i] = i-1;
}

//exit SPI initialization
while(!Serial.available()){  
    ;//delay program until serial is triggered
}

Serial.println("Begin Force Characterization Experiment");

void loop() {
    //If the trials have been completed, make sure to kill the program
    if(trialNum >= trialCap){
        Serial.println("End of experiment, trial number exceeded");
        while(1){
            ;//this infinite loop is to trap the controller once the trials are done
        }
    }
    //check which trial we're in right now, or if we're even in one
    bool inTrial = false;
    for(int i = 0; i < 6; i++){
        if(activeSet[i] > 0){
            inTrial = true;
        }
    }
    //if we're not in a trial, let's get in one
    if(!inTrial){
        getActiveSet();
    }
    else{
        //if we are in a trial, make sure to control the heating or cooling process
        //in intervals of 50 ms collect the temperature data
        //(delay required for SPI to run)
if(millis() >= (tempTimer + 50)){
    master.transfer(spi_master_tx_buf, spi_master_rx_buf, BUFFER_SIZE);
    tempTimer = millis();
}

for(int i = 0; i < 12; i++){
    //for every thermocouple, save the new data to the temperature array
    byte receiver = spi_master_rx_buf[i+2];
    tempVals[i][0] = tempVals[i][1];
    tempVals[i][1] = tempVals[i][2];
    tempVals[i][2] = (double)receiver;
    //if it's an outlier, make it instead an average of the previous two values
    //outlier we'll define as 40% different than last two values
    //that *should* never happen
    double tempAvg = (tempVals[i][0]+tempVals[i][1])/2.0;
    double absoluteDiff = abs(tempVals[i][2] - tempAvg);
    if(absoluteDiff > (tempAvg*0.4)){
        tempVals[i][2] = tempAvg;
    }
}

//Then need to output everything
outputData();
if(heating){
    controlHeating();
}
else{
    controlCooling();
}
}
void getActiveSet() {
    //get the current list element
    int temp = list[trialNum];
    //increment through each digit to get active TCAs
    activeSet[0] = (int)(temp/100000);
    temp-=activeSet[0]*100000;
    activeSet[1] = (int)(temp/10000);
    temp-=activeSet[1]*10000;
    activeSet[2] = (int)(temp/1000);
    temp-=activeSet[2]*1000;
    activeSet[3] = (int)(temp/100);
    temp-=activeSet[3]*100;
    activeSet[4] = (int)(temp/10);
    temp-=activeSet[4]*10;
    activeSet[5] = (int)(temp);
    //Output that we are starting a trial, and which pairs are active
    Serial.print("Begin Trial ");
    byte tempTrialNum = trialNum + 1;
    Serial.println(tempTrialNum);
    Serial.println("Active Set is ");
    Serial.print(activeSet[0]);
    Serial(",");
    Serial.print(activeSet[1]);
    Serial(",");
    Serial.print(activeSet[2]);
    Serial(",");
    Serial.print(activeSet[3]);
    Serial(",");
    Serial.print(activeSet[4]);
    Serial(",");
    Serial.print(activeSet[5]);
}
Serial.print(activeSet[3]);
Serial.print(",");
Serial.print(activeSet[4]);
Serial.print(",");
Serial.print(activeSet[5]);
Serial.println("]");
}

//this function controls the heating process of the actuator
void controlHeating(){
  int maintenanceCount = 0; //keeps track of how many tcas are maintaining
  int activeCount = 0; //keeps track of how many tcas are active
  //cycle through the set of active pairs
  for(int i = 0; i < 6; i++){
    if(activeSet[i] > 0){
      //if the current element is greater than zero, apply logic accordingly
      int left = activeSet[i] - 1;
      int right = left + 6;
      int leftCount = 0;
      int rightCount = 0;
      bool leftHeat = true;
      bool rightHeat = true;
      //cycle through last 3 temperature readings, increment counts if higher
      //than temperature targets
      //if all 3 are higher, then we can begin cooling
      //This is done to reject any specific spikes.
      for(int c = 0; c < 3; c++){
        if(tempVals[left][c] >= targetTemp){
          leftCount++;
        }
        if(tempVals[right][c] >= targetTemp){
          rightCount++;
        }
      }
      if(leftCount >= 3){
        leftHeat = false;
      }
      if(rightCount >= 3){
        rightHeat = false;
      }
      if(leftHeat && rightHeat){
        //apply heating logic
      }
    }
  }
}
if(tempVals[right][c] >= targetTemp){
    rightCount++;
}

//if we're in the maintaining phase and it gets hot, cool it
if(leftCount > 0 && maintaining[left]){  
digitalWrite(gatePin[left], LOW);  
}
if(rightCount > 0 && maintaining[right]){  
digitalWrite(gatePin[right], LOW);  
}

//if the counts were less than the target temp, turn up the heat
if(leftCount < 3){
    digitalWrite(gatePin[left], HIGH);
}
else if(leftCount == 3){  
    //otherwise, need to begin cooling/maintenance  
maintaining[left] = true;  
digitalWrite(gatePin[left], LOW);  
}

//same goes for the right side
if(rightCount < 3){
    digitalWrite(gatePin[right], HIGH);
}
else if(rightCount == 3){
    maintaining[right] = true;  
digitalWrite(gatePin[right], LOW);
//if the pair is active, increase the count of active TCAs by two
activeCount += 2;

//check how many tcas are maintaining temperature right now
for(int i = 0; i < 12; i++){
    if(maintaining[i]){
        maintenanceCount++;
    }
}

//if greater to or equal to the number of active TCAs
//flip the direction and reset variables
if(maintenanceCount >= activeCount){
    heating = false;
    for(int i = 0; i < 12; i++){
        maintaining[i] = false;
    }
}

//this function controls the cooling process of the actuator
void controlCooling(){
    //cycle through every TCA, if they're all below the threshold
    //((set in global) tick over to a new trial
    int coldCount = 0;
    for(int i = 0; i < 12; i++){
        digitalWrite(gatePin[i], LOW);
        int tempCount = 0;
for(int c = 0; c < 3; c++) {
    if(tempVals[i][c] < tempMin) {
        tempCount++;
    }
}

if(tempCount >= 3) {
    coldCount++;
}

if(coldCount >= 12) {
    heating = true;
    trialNum += 1;
    for(int i = 0; i < 6; i++) {
        activeSet[i] = 0;
    }
}

//this function outputs the data to the serial monitor to be transcribed
void outputData() {
    //get the force and powers
    if(scale.is_ready()) {
        force = getForce();
    }

    //getting the power from the power function
    power = 0;
    for(int i = 0; i < 6; i++) {
        // Code continues here...
//float tempPower = 0
float tempPower = getPower(i);
power += tempPower;
}
if(power < 0){
power = 0;
}

//get time stamp
float curTime = millis()/1000.0;

//output all values, separated by commas. Will go
//[[Time Stamp]][[Force]][[Total Power Consumption]][[Temperature (12 elements)]]

//output time stamp
Serial.print(curTime);
Serial.print(",");

//output distance
Serial.print(force);
Serial.print(",");

//output the power
Serial.print(power);
Serial.print(",");

//output the temperatures and end the line
for(int i = 0; i < 11; i++){
    Serial.print(tempVals[i][2]);
    Serial.print(",");
}
Serial.println(tempVals[11][2]);

//total output will be a 15 element line
}
//this function gets the force being exerted on the load cell

double getForce(){
    //Returns the corrected weight found by the sensor
    //Uses the zero value found experimentally to get the calibration value
    //then divides sensor reading by calibration to get final force
    //based on zero'd value
    double reading = scale.read() * 1.0;
    double inputVal = reading - 78550.62;
    if(inputVal <= 200){
        calibration = 0;
        return 0.0;
    }
    else{
        calibration = 372.0513;
        return (inputVal*1.0)/calibration;
    }
}

double getPower(int index){
    float busVoltage = 0;
    float current = 0; //will be measured in milliamps
    //get the voltage
    busVoltage = ina[index].getBusVoltage_V();
    //get the current
    current = ina[index].getCurrent_mA();
    //calculate the power using Ohm's law
    float power = busVoltage*(current/1000);
    double newPow = 0.2841*power*power + 0.6191*power + 0.625;
    return newPow;
C.3 Characterization Experiment Code

C.3.4 Constant Frequency Characterization Experiment Code

//Requires the Wire.h library, published by Arduino Inc. under
//the Creative Commons Attribution-ShareAlike 3.0 License.
//Requires the INA219 device driver library, published by Adafruit Inc.
//under the MIT License.
//Requires the ESP32 DMA SPI library, published by Hideaki Tai
//under the MIT License.
//Requires the SPI.h library, published by Arduino Inc. under
//the Creative Commons Attribution-ShareAlike 3.0 License.

/*
Author: Vaughan Murphy
Purpose: Characterize the stroke production of the
Woven Bi-Pennate Actuator operating at a 0.5 Hz frequency.
*/

#include <ESP32DMASPIMaster.h>
#include <Wire.h>
#include <Adafruit_INA219.h>
#include <SPI.h>

//SPI communications variables
ESP32DMASPI::Master master;
static const uint32_t BUFFER_SIZE = 14;
uint8_t* spi_master_tx_buf;
uint8_t* spi_master_rx_buf;
// temperature related global variables

double tempVals[][3] = {{0, 0, 0},
                         {0, 0, 0},
                         {0, 0, 0},
                         {0, 0, 0},
                         {0, 0, 0},
                         {0, 0, 0},
                         {0, 0, 0},
                         {0, 0, 0},
                         {0, 0, 0},
                         {0, 0, 0},
                         {0, 0, 0},
                         {0, 0, 0},
                         {0, 0, 0},
                         {0, 0, 0}};

// power related global variables

double power = 0;
Adafruit_INA219 ina[] = {
    Adafruit_INA219(0x40),
    Adafruit_INA219(0x41),
    Adafruit_INA219(0x44),
    Adafruit_INA219(0x45),
    Adafruit_INA219(0x42),
    Adafruit_INA219(0x46),
};

// infrared (displacement) related variables

int infraredPin = 35;
double distance = 0;

//control related global variables
double targetTemp = 80; //temperature to be maintained
double tempMin = 30; //floor temperature at which point the TCA is cooled
int gatePin[] = {33, 26, 23, 18, 17, 4, 25, 27, 19, 5, 16, 2};
long tempTimer = 0, timer = 0;
bool isReached[] = {false, false, false, false, false, false, false,
false, false, false, false, false};
bool maintaining[] = {false, false, false, false, false, false, false,
false, false, false, false, false};
bool startCheck = false;

//general operation variables
int trialNum = 0, trialCap = 5;
bool heating = true; //this dictates if the trial is heating [true] or cooling [false]

//active set related variables
int activeSet[] = {0, 0, 0, 0, 0};
int list[] = {0, 0, 0, 0
}; //doing 5 trials per # pairs per actuator

void setup() {
    // setup pins
    for(int i = 0; i < 12; i++){
        pinMode(gatePin[i], OUTPUT);
        digitalWrite(gatePin[i], LOW);
    }
}
pinMode(infraredPin, INPUT);
Serial.begin(9600);

//begin INA communications
for(int i = 0; i < 6; i++){
    ina[i].begin();
}

//begin SPI communications
// to use DMA buffer, use these methods to allocate buffer
spi_master_tx_buf = master.allocDMABuffer(BUFFER_SIZE);
spi_master_rx_buf = master.allocDMABuffer(BUFFER_SIZE);

master.setDataMode(SPI_MODE3); // for DMA, only 1 or 3 is available
//set the frequency to something lower
master.setFrequency(1000);
master.setMaxTransferSize(BUFFER_SIZE);
master.setDMAChannel(1); // 1 or 2 only
master.setQueueSize(1); // transaction queue size

master.begin(); // default SPI is HSPI
// set buffer data here
spi_master_tx_buf[0] = 14;
for(int i = 1; i <= 12; i++){
    spi_master_tx_buf[i] = i-1;
}
// exit SPI initialization
while(!Serial.available()){
    startCheck = false;  // delay program until serial is triggered
}
startCheck = true;
Serial.println("Begin Constant Frequency Experiment");
}

void loop() {
    // If the trials have been completed, make sure to kill the program
    if(trialNum >= trialCap) {
        Serial.println("End of experiment, trial number exceeded");
        while(1){
            // this infinite loop is to trap the controller once the trials are done
        }
    }
    // check which trial we're in right now, or if we're even in one
    bool inTrial = false;
    for(int i = 0; i < 6; i++){
        if(activeSet[i] > 0){
            inTrial = true;
        }
    }
    // if we're not in a trial, let's get in one
    if(!inTrial) {
        getActiveSet();
    } else {
        // if we are in a trial, make sure to control the heating or cooling process
// in intervals of 50 ms collect the temperature data
//(delay required for SPI to run)
if(millis() >= (tempTimer + 50)) {
    master.transfer(spi_master_tx_buf, spi_master_rx_buf, BUFFER_SIZE);
    tempTimer = millis();
}
for(int i = 0; i < 12; i++) {
    // for every thermocouple, save the new data to the temperature array
    byte receiver = spi_master_rx_buf[i + 2];
    tempVals[i][0] = tempVals[i][1];
    tempVals[i][1] = tempVals[i][2];
    tempVals[i][2] = (double)receiver;
    // if it's an outlier, make it instead an average
    // of the previous two values
    // outlier we'll define as 40% different than last two values
    // that *should* never happen
    double tempAvg = (tempVals[i][0] + tempVals[i][1]) / 2.0;
    double absoluteDiff = abs(tempVals[i][2] - tempAvg)
    if(absoluteDiff > (tempAvg * 0.4)) {
        tempVals[i][2] = tempAvg;
    }
}
// Then need to output everything
outputData();
if(heating){
pulseHeat();
}
else{
    controlCooling();
}
// this function deciphers which TCAs are part of the current trial
// from the trial list (for control purposes)

void getActiveSet() {
    // get the current list element
    int temp = list[trialNum];
    // increment through each digit to get active TCAs
    activeSet[0] = (int)(temp/100000);
    temp-=activeSet[0]*100000;
    activeSet[1] = (int)(temp/10000);
    temp-=activeSet[1]*10000;
    activeSet[2] = (int)(temp/1000);
    temp-=activeSet[2]*1000;
    activeSet[3] = (int)(temp/100);
    temp-=activeSet[3]*100;
    activeSet[4] = (int)(temp/10);
    temp-=activeSet[4]*10;
    activeSet[5] = (int)(temp);
    // Output that we are starting a trial, and which pairs are active
    Serial.print("Begin Trail ");
    byte tempTrialNum = trialNum + 1;
    Serial.println(tempTrialNum);
    Serial.print("Active Set is ");
    Serial.print(activeSet[0]);
    Serial.print(", ");
    Serial.print(activeSet[1]);
//this function controls the cooling process of the actuator

void controlCooling()
{
    //cycle through every TCA, if they're all below the threshold
    // (set in global) tick over to a new trial
    int coldCount = 0;
    for(int i = 0; i < 12; i++){
        digitalWrite(gatePin[i], LOW);
        int tempCount = 0;
        for(int c = 0; c < 3; c++){
            if(tempVals[i][c] < tempMin){
                tempCount++;
            }
        }
        if(tempCount >= 3){
            coldCount++;
        }
    }
    //if all of the TCAs are cold, tick trial number
/*clear active set and revert to heating*/

if(coldCount >= 12){
    heating = true;
    trialNum += 1;
    timer = millis();
    for(int i = 0; i < 6; i++){
        activeSet[i] = 0;
    }
}

//this function outputs the data to the serial monitor to be transcribed

void outputData(){
    //get the distance and powers
    distance = getDisplacement();
    //getting the power from the power function
    power = 0;
    for(int i = 0; i < 6; i++){
        float tempPower = getPower(i);
        power += tempPower;
    }
    if(power < 0){
        power = 0;
    }
    //get time stamp
    float curTime = millis()/1000.0;
    //output all values, separated by commas. Will go
    //[[Time Stamp]][[Distance]][[Total Power Consumption]][[Temperature (12 elements)]]
    //output time stamp
Serial.print(curTime);
Serial.print(",");

//output distance
Serial.print(distance);
Serial.print(",");

//output the power
Serial.print(power);
Serial.print(",");

//output the temperatures and end the line
for(int i = 0; i < 11; i++){
    Serial.print(tempVals[i][2]);
    Serial.print(",");
}
Serial.println(tempVals[11][2]);
//total output will be a 15 element line

//this function gets the distance between the IR sensor and the actuator in mm
double getDisplacement(){
    double val = analogRead(infraredPin);

    //linearization equation devised via calibration experimentation
    double mm = 61763* pow(val, -1.031);
    return mm;
}

double getPower(int index){
    float busVoltage = 0;
    float current = 0; //will be measured in milliamps

    //get the voltage
busVoltage = ina[index].getBusVoltage_V();

// get the current

current = ina[index].getCurrent_mA();

// calculate the power using Ohm's law

float power = busVoltage*(current/1000);

// rectify using calibration function from preliminary experiment

double newPow = 0.2841*power*power + 0.6191*power + 0.625;

return newPow;
}

void pulseHeat(){

// pulsing the heat to make it speedy, set local variables

int interval = 2000, startTime = 500, activeTca = 0, tcaCount = 0;

// get the total number of active TCAs, and which ones are active

for(int i = 0; i < 6; i++){
    if(activeSet[i] > 0){
        tcaCount += 2;
    }
}

int *tcaPoint = new int[tcaCount];

int tempCount = 0;

for(int i = 0; i < 6; i++){
    if(activeSet[i] > 0){
        int left = activeSet[i] - 1;
        int right = left + 6;
        tcaPoint[tempCount] = left;
        tempCount++;
        tcaPoint[tempCount] = right;
    }
}
tempCount++;
}

// now going to pulse the active TCAs for 2 seconds
//(duration set by interval value)
// need to set power so that this results in maximum contraction
//( ~20 W per TCA)
if(timer == 0){
timer = millis();
}
for(int i = 0; i < tcaCount; i++){
    int tempNum = tcaPoint[i];
    activeTca = gatePin[tempNum];
    digitalWrite(activeTca, LOW);
}

if((millis() > timer + startTime) && startCheck){
    // once past the start time, initiate heating
    for(int i = 0; i < tcaCount; i++){
        int tempNum = tcaPoint[i];
        activeTca = gatePin[tempNum];
        digitalWrite(activeTca, HIGH);
    }
    if(millis() > (timer + startTime + interval)){
        // once that interval has been elapsed, cool the TCAs down
        for(int i = 0; i < tcaCount; i++){
            int tempNum = tcaPoint[i];
            activeTca = gatePin[tempNum];
            digitalWrite(activeTca, LOW);
        }
    }
}
C.3.5 Maximum Frequency Characterization Experiment Code

//Requires the Wire.h library, published by Arduino Inc. under
//the Creative Commons Attribution-ShareAlike 3.0 License.

//Requires the INA219 device driver library, published by Adafruit Inc.
//under the MIT License.

//Requires the ESP32 DMA SPI library, published by Hideaki Tai
//under the MIT License.

/*
Author: Vaughan Murphy
Purpose: Characterize the maximum frequency of the
Woven Bi-Pennate Actuator.
*/

#include <ESP32DMASPIMaster.h>
#include <Wire.h>
#include <Adafruit_INA219.h>

//SPI communications variables
ESP32DMASPI::Master master;
static const uint32_t BUFFER_SIZE = 14;
uint8_t* spi_master_tx_buf;
uint8_t* spi_master_rx_buf;
//temperature related global variables

double tempVals[][3] = {
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
    {0, 0, 0},
};

//power related global variables

double power = 0;
Adafruit_INA219 ina[] = {
    Adafruit_INA219(0x40),
    Adafruit_INA219(0x41),
    Adafruit_INA219(0x44),
    Adafruit_INA219(0x45),
    Adafruit_INA219(0x42),
    Adafruit_INA219(0x46),
};

//infrared (displacement) related variables
```c
int infraredPin = 35;
double distance = 0;

// control related global variables
double targetTemp = 80; // temperature to be maintained
double tempMin = 30; // floor temperature at which point the TCA is cooled
int gatePin[] = {33, 26, 23, 18, 17, 4, 25, 27, 19, 5, 16, 2};
long tempTimer = 0, timer = 0, heatTimer = 0;
bool isReached[] = {false, false, false, false, false, false, false, 
                    false, false, false, false, false};
bool maintaining[] = {false, false, false, false, false, false, false, 
                       false, false, false, false, false};

// general operation variables
int trialNum = 0, trialCap = 30; // change as required
bool heating = true; // this dictates if the trial is heating [true] or cooling [false]

// active set related variables, should always initialize as set of zeros
int activeSet[] = {0, 0, 0, 0, 0};
int list[] = {0, 0, 0, 0, 0,
             0, 0, 0, 0, 0,
             0, 0, 0, 0, 0,
             0, 0, 0, 0, 0,
             0, 0, 0, 0, 0,
             0, 0, 0, 0, 0}; // doing 5 trials per # pairs per actuator, total of 300 trials
// insert list as set of undemarcated active pairs
// i.e. 13 will have pairs 1 and 3 as a trial
```
void setup() {
    // initialize pins
    for(int i = 0; i < 12; i++){
        pinMode(gatePin[i], OUTPUT);
    }
    pinMode(infraredPin, INPUT);
    Serial.begin(9600);

    // begin INA communications
    for(int i = 0; i < 6; i++){
        ina[i].begin();
    }

    // begin SPI communications
    // to use DMA buffer, use these methods to allocate buffer
    spi_master_tx_buf = master.allocDMABuffer(BUFFER_SIZE);
    spi_master_rx_buf = master.allocDMABuffer(BUFFER_SIZE);

    master.setDataMode(SPI_MODE3); // for DMA, only 1 or 3 is available
    // set the frequency to something lower
    master.setFrequency(1000);
    master.setMaxTransferSize(BUFFER_SIZE);
    master.setDMAChannel(1); // 1 or 2 only
    master.setQueueSize(1); // transaction queue size

    master.begin(); // default SPI is HSPI
    // set buffer data here
spi_master_tx_buf[0] = 14;
for(int i = 1; i <= 12; i++){
    spi_master_tx_buf[i] = i-1;
}
//exit SPI initialization
while(!Serial.available()){
    ;//delay program until serial is triggered
}
Serial.println("Begin Maximum Frequency Characterization Experiment");
}

void loop() {
    //If the trials have been completed, make sure to kill the program
    if(trialNum >= trialCap){
        Serial.println("End of experiment, trial number exceeded");
        while(true){
            ;//this infinite loop is to trap the controller once the trials are done
        }
    }
    //check which trial we're in right now, or if we're even in one
    bool inTrial = false;
    for(int i = 0; i < 6; i++){
        if(activeSet[i] > 0){
            inTrial = true;
        }
    }
    //if we're not in a trial, let's get in one
if(!inTrial){
    getActiveSet();
}
else{
    // if we are in a trial, make sure to control the heating or cooling process
    // in intervals of 50 ms collect the temperature data
    // (delay required for SPI to run)
    if(millis() >= (tempTimer + 50)){
        master.transfer(spi_master_tx_buf, spi_master_rx_buf, BUFFER_SIZE);
        tempTimer = millis();
    }
    for(int i = 0; i < 12; i++){
        // for every thermocouple, save the new data to the temperature sliding array
        byte receiver = spi_master_rx_buf[i+2];
        tempVals[i][0] = tempVals[i][1];
        tempVals[i][1] = tempVals[i][2];
        tempVals[i][2] = (double)receiver;
        // if it's an outlier, make it instead an average of the previous two values
        // outlier we'll define as 40% different than last two values
        // that *should* never happen
        double tempAvg = (tempVals[i][0]+tempVals[i][1])/2.0;
        double absoluteDiff = abs(tempVals[i][2] - tempAvg)
        if(absoluteDiff > (tempAvg*0.4)){
            tempVals[i][2] = tempAvg;
        }
    }
    // Then need to output everything
    outputData();
    // check which direction we're going, apply controls accordingly
if(heating){
    if(heatTimer == 0){
        heatTimer = millis();
    }
    controlHeating();

    //this experiment is essentially the same as the stroke
    //but make sure to limit the heating to <= 3 sec
    //otherwise damage could occur
    if((millis() - heatTimer) > 3000){
        heating = false;
    }
}
else{
    controlCooling();
}
}

//this function deciphers which TCAs are part of the current trial
//from the trial list (for control purposes)
void getActiveSet(){
    //get the current list element
    int temp = list[trialNum];
    //increment through each digit to get active TCAs
    activeSet[0] = (int)(temp/100000);
    temp-=activeSet[0]*100000;
    activeSet[1] = (int)(temp/10000);
    temp-=activeSet[1]*10000;
    activeSet[2] = (int)(temp/1000);
temp-=activeSet[2]*1000;
activeSet[3] = (int)(temp/100);
temp-=activeSet[3]*100;
activeSet[4] = (int)(temp/10);
temp-=activeSet[4]*10;
activeSet[5] = (int)(temp);

//Output that we are starting a trial, and which pairs are active
Serial.print("Begin Trail ");
byte tempTrialNum = trialNum + 1;
Serial.println(tempTrialNum);
Serial.println("Active Set is [");
Serial.print(activeSet[0]);
Serial.print(" , ");
Serial.print(activeSet[1]);
Serial.print(" , ");
Serial.print(activeSet[2]);
Serial.print(" , ");
Serial.print(activeSet[3]);
Serial.print(" , ");
Serial.print(activeSet[4]);
Serial.print(" , ");
Serial.print(activeSet[5]);
Serial.println("]");
}

//this function controls the heating process of the actuator
void controlHeating(){
  int maintenanceCount = 0; //keeps track of how many tcas are maintaining
  int activeCount = 0; //keeps track of how many tcas are active

//cycle through the set of active pairs
for(int i = 0; i < 6; i++){
    if(activeSet[i] > 0){
        //if the current element is greater than zero, apply logic accordingly
        int left = activeSet[i] - 1;
        int right = left + 6;
        int leftCount = 0;
        int rightCount = 0;
        bool leftHeat = true;
        bool rightHeat = true;
        //cycle through last 3 temperature readings, increment
        //counts if higher than temperature targets
        //if all 3 are higher, then we can begin cooling.
        //This is done to reject any specific spikes
        for(int c = 0; c < 3; c++){
            if(tempVals[left][c] >= targetTemp){
                leftCount++;
            }
            if(tempVals[right][c] >= targetTemp){
                rightCount++;
            }
        }
        //if we're in the maintaining phase and it gets hot, cool it
        if(leftCount > 0 && maintaining[left]){
            digitalWrite(gatePin[left], LOW);
        }
        if(rightCount > 0 && maintaining[right]){
            digitalWrite(gatePin[right], LOW);
        }
    }
}
//if the counts were less than the target temp, turn up the heat
if(leftCount < 3){
    digitalWrite(gatePin[left], HIGH);
}
else if(leftCount == 3){
    //otherwise, need to begin cooling/maintenance
    maintaining[left] = true;
    digitalWrite(gatePin[left], LOW);
}

//same goes for the right side
if(rightCount < 3){
    digitalWrite(gatePin[right], HIGH);
}
else if(rightCount == 3){
    maintaining[right] = true;
    digitalWrite(gatePin[right], LOW);
}

//if the pair is active, increase the count of active TCAs by two
activeCount += 2;

//check how many tcas are maintaining temperature right now
for(int i = 0; i < 12; i++){
    if(maintaining[i]){
        maintenanceCount++;
    }
}

//if greater to or equal to the number of active TCAs
    //flip the direction and reset variables
if(maintenanceCount >= activeCount){
    heating = false;
    for(int i = 0; i < 12; i++){
        maintaining[i] = false;
    }
}

//this function controls the cooling process of the actuator
void controlCooling(){
    //cycle through every TCA, if they're all below the threshold
    //((set in global) tick over to a new trial
    int coldCount = 0;
    for(int i = 0; i < 12; i++){
        digitalWrite(gatePin[i], LOW);
        int tempCount = 0;
        for(int c = 0; c < 3; c++){
            if(tempVals[i][c] < tempMin){
                tempCount++;
            }
        }
        if(tempCount >= 3){
            coldCount++;
        }
    }
    //if all of the TCAs are cold, tick trial number,
    //clear active set and revert to heating
    if(coldCount >= 12){
        heating = true;
    }
}
//reset the heating timer as required
heatTimer = millis();
trialNum += 1;
for(int i = 0; i < 6; i++){
    activeSet[i] = 0;
}

//this function outputs the data to the serial monitor to be transcribed
void outputData(){
    //get the distance and powers
distance = getDisplacement();
    //getting the power from the power function
    power = 0;
    for(int i = 0; i < 6; i++){
        //float tempPower = 0
        float tempPower = getPower(i);
        power += tempPower;
    }
    if(power < 0){
        power = 0;
    }
    //get time stamp
    float curTime = millis()/1000.0;
    //output all values, separated by commas. Will go
    //[[Time Stamp] ][Distance] [[Total Power Consumption] ][Temperature (12 elements)]
    //output time stamp
    Serial.print(curTime);
Serial.print("", ");
//output distance
Serial.print(distance);
Serial.print("", ");
//output the power
Serial.print(power);
Serial.print("", ");
//output the temperatures and end the line
for(int i = 0; i < 11; i++){
    Serial.print(tempVals[i][2]);
    Serial.print("", ");
}
Serial.println(tempVals[11][2]);
//total output will be a 15 element line

//this function gets the distance between the IR sensor and the actuator in mm
double getDisplacement(){
    double val = analogRead(infraredPin);
    //linearization equation devided via calibration experimentation
    double mm = 61763*pow(val, -1.031);
    return mm;
}

double getPower(int index){
    float busVoltage = 0;
    float current = 0; //will be measured in milliamps
    //get the voltage
    busVoltage = ina[index].getBusVoltage_V();
//get the current
current = ina[index].getCurrent_mA();

//calculate the power using Ohm's law
float power = busVoltage*(current/1000);

//rectify using calibration function from preliminary experiment
double newPow = 0.2841*power*power + 0.6191*power + 0.625;
return newPow;

C.4 Force Control Experiment Code

C.4.1 PID Control Experiment Code

//Requires the Wire.h library, published by Arduino Inc. under
//the Creative Commons Attribution-ShareAlike 3.0 License.
//Requires the INA219 device driver library, published by Adafruit Inc.
//under the MIT License.
//Requires the SPI.h library, published by Arduino Inc. under
//the Creative Commons Attribution-ShareAlike 3.0 License.
//Requires the ESP32 DMA SPI library, published by Hideaki Tai
//under the MIT License.
//Requires the HX711.h library, published by Bogdan Necula
//under the MIT License.
//Requires the Wire.h library, published by Arduino Inc. under
//the Creative Commons Attribution-ShareAlike 3.0 License.

/*
Author: Vaughan Murphy
Purpose: Apply and test PID control on the
Woven Bi-Pennate Actuator.
*/
C.4 Force Control Experiment Code

```cpp
/*

#include <ESP32DMASPMaster.h>
#include <Wire.h>
#include <Adafruit_INA219.h>
#include <SPI.h>
#include <HX711.h>

//power related global variables
double power = 0;
Adafruit_INA219 ina[] = {
    Adafruit_INA219(0x40),
    Adafruit_INA219(0x41),
    Adafruit_INA219(0x44),
    Adafruit_INA219(0x45),
    Adafruit_INA219(0x42),
    Adafruit_INA219(0x46),
};

//load cell (force) related variables
int loadCellDout = 35;
int loadCellSck = 32;
double force = 0;
double calibration = 1;
HX711 scale;

//control related global variables
double targetTemp = 80; //temperature to be maintained
double tempMin = 30; //floor temperature at which point the TCA is cooled
*/
```
int gatePin[] = {33, 26, 23, 18, 17, 4, 25, 27, 19, 5, 16, 2};
long maintainTimer = 0, timer = 0, initialForce = 0, controlStart = 0, pwmStart = 0;
bool isReached[] = {false, false, false, false, false, false, false, false, false, false, false, false};
bool maintaining[] = {false, false, false, false, false, false, false, false, false, false, false, false};

//general operation variables
int trialNum = 0, trialCap = 1;
bool heating = true; //this dictates if the trial is heating [true] or cooling [false]

//active set related variables
int activeSet[] = {5, 0, 0, 0, 0, 0};
int list[] = {5, 5, 5, 5, 5, 5}; //doing 5 trials per # pairs per actuator
//insert list as set of undemarcated active pairs
//i.e. 13 will have pairs 1 and 3 as a trial

void setup() {
    // put your setup code here, to run once:
    for(int i = 0; i < 12; i++){
        pinMode(gatePin[i], OUTPUT);
    }
    Serial.begin(9600);

    //begin INA communications
    for(int i = 0; i < 6; i++){
        ina[i].begin();
    }
}
//get the load cell set up
scale.begin(loadCellDout, loadCellSck);
scale.set_scale(calibration);
scale.tare();

//exit SPI initialization
while(!Serial.available()){
    ;//delay program until serial is triggered
}
Serial.println("Begin PID Force Control Experiment");
}

void loop() {
    // put your main code here, to run repeatedly:
    //If the trials have been completed, make sure to kill the program
    if(trialNum >= trialCap){
        Serial.println("End of experiment, trial number exceeded");
        while(1){
            ;//this infinite loop is to trap the controller once the trials are done
        }
    }
    //check which trial we're in right now, or if we're even in one
    bool inTrial = false;
    for(int i = 0; i < 6; i++){
        if(activeSet[i] > 0){
            inTrial = true;
        }
    }
    //if we're not in a trial, let's get in one
\begin{verbatim}
if(!inTrial){
    getActiveSet();
}
else{
    //if we are in a trial, make sure to control the heating or cooling process
    //Need to output all data
    outputData();
    if(heating){
        //if hot, can apply control scheme, will have capacity to control
        //temperature rising and settling
        applyControl();
    }
    else{
        controlCooling();
    }
}

long errorSum = 0, prevError = 0;
//all of the coefficients for the PID controllers, from simulink simulations
double kp[] = {30, 3.5, 4.0};
double ki[] = {0.05, 0.1, 0.10};
double kd[] = {1.5, 0.90, 0.06};
bool oscillating = false;
int targetForce = 200; //change as required
//this function applies the control law to control the heating of the actuator
void applyControl(){
    long pwmInterval = 100, controlInterval = 5000;
    int activeCount = 0;
}\end{verbatim}
double pwmFactor = 0;

// if it's the start of the control section, get initial values
if (controlStart == 0){
    initialForce = getForce();
    controlStart = millis();
    pwmStart = millis();
}

// get the number of active TCAs to be able to select the proper coefficients
for(int i = 0; i < 6; i++){
    if(activeSet[i] > 0){
        activeCount++;
    }
}

if (activeCount == 2) targetForce = 400;
if (activeCount == 3) targetForce = 550;
int tcaNum = activeCount - 1; // needs rectification for array selection

// if the control has lasted longer than the designated control interval
// begin cooling and reset initialization values
if (millis() > (controlStart + controlInterval)){
    heating = false;
    oscillating = false;
    controlStart = 0;
    pwmStart = 0;
}
else{
    // otherwise control the actuator
    // make sure to refresh the pwm start point
    if (millis() > (pwmStart + pwmInterval) || (controlStart == pwmStart)){
        pwmStart = millis();
    }
}
//calculate the pwmFactor required
double curForce = force - initialForce;

//get the error in the system for the feedback component
double error = targetForce - curForce;
if (!oscillating && curForce > (0.95*targetForce)){
    oscillating = true;
}

//calculate based on PID values found from simulations
pwmFactor = error*kp[tcaNum] + errorSum*ki[tcaNum] + (error-prevError)*kd[tcaNum];
prevError = error;
errorSum += error;

//limit the pwmFactor based on saturation (0-1.0)
if(pwmFactor > 1.0) pwmFactor = 1.0;
if(pwmFactor < 0.0) pwmFactor = 0.0;

//apply calculated pwmFactor value to a pwm heating scheme
//so long as pwmStart has been set
if(millis() < pwmStart + (pwmFactor*pwmInterval) && pwmStart != 0){
    //after start, for (pwmFactor) percent of the interval
    //heat the TCAs in the active set
    for(int i = 0; i < 6; i++){
        int left = activeSet[i] - 1;
        int right = left + 6;
        digitalWrite(gatePin[left], HIGH);
        digitalWrite(gatePin[right], HIGH);
    }
}
else{
    for(int i = 0; i < 6; i++){
```c
int left = activeSet[i] - 1;
int right = left + 6;
digitalWrite(gatePin[left], LOW);
digitalWrite(gatePin[right], LOW);
}
}
}

//this function controls the cooling process of the actuator
void controlCooling(){
    //cycle through every TCA, if they're all below the threshold
    //(set in global) tick over to a new trial
    int coldCount = 0;
    for(int i = 0; i < 12; i++){
        digitalWrite(gatePin[i], LOW);
    }
    //if it has cooled enough to meet its initial force, tick trial number
    //clear active set and revert to heating
    if (force <= initialForce){
        heating = true;
        trialNum += 1;
        timer = millis();
        for(int i = 0; i < 6; i++){
            activeSet[i] = 0;
        }
    }
}
```

// this function deciphers which TCAs are part of the current trial
//from the trial list (for control purposes)

void getActiveSet()
{

    // get the current list element
    int temp = list[trialNum];

    // increment through each digit to get active TCAs
    activeSet[0] = (int)(temp/100000);
    temp-=activeSet[0]*100000;
    activeSet[1] = (int)(temp/10000);
    temp-=activeSet[1]*10000;
    activeSet[2] = (int)(temp/1000);
    temp-=activeSet[2]*1000;
    activeSet[3] = (int)(temp/100);
    temp-=activeSet[3]*100;
    activeSet[4] = (int)(temp/10);
    temp-=activeSet[4]*10;
    activeSet[5] = (int)(temp);

    // Output that we are starting a trial, and which pairs are active
    Serial.print("Begin Trial ");
    byte tempTrialNum = trialNum + 1;
    Serial.println(tempTrialNum);
    Serial.println("Active Set is ");
    Serial.print(activeSet[0]);
    Serial.print(", ");
    Serial.print(activeSet[1]);
    Serial.print(", ");
    Serial.print(activeSet[2]);
    Serial.print(", ");
    Serial.print(activeSet[3]);
C.4 Force Control Experiment Code

```java
Serial.print(',', '');
Serial.print(activeSet[4]);
Serial.print(',', '');
Serial.print(activeSet[5]);
Serial.println(']');

//this function outputs the data to the serial monitor to be transcribed
void outputData(){
    //get the distance and powers
    force = getForce()-initialForce;// - initialForce;
    double zeroForce = force - initialForce;
    //getting the power from the power function
    power = 0;
    for(int i = 0; i < 6; i++){
        float tempPower = getPower(i);
        power += tempPower;
    }
    if(power < 0){
        power = 0;
    }
    //get time stamp
    float curTime = millis()/1000.0;
    //output all values, separated by commas. Will go
    //[Time Stamp][Force][Total Power Consumption]
    //output time stamp
    Serial.print(curTime);
    Serial.print(',', '');
    Serial.print('"");
    //output force
```
Serial.print(force);
Serial.print("", "");
// output the power
Serial.print(power);
Serial.print("\n");
// total output will be a 3 element line
}

// this function gets the force being exerted on the load cell
double getForce(){
  // Returns the corrected weight found by the sensor
  // Uses the zero value found experimentally to get the calibration value
  // then divides sensor reading by calibration to get final force
  // based on zero'd value
  double reading = scale.read() * 1.0;
  double inputVal = reading - 78550.62;
  if(inputVal <= 200){
    calibration = 0;
    return 0.0;
  }
  else{
    calibration = 372.0513;
    return (inputVal*1.0)/calibration;
  }
}

double getPower(int index){
  float busVoltage = 0;
  float current = 0; // will be measured in milliamps
/get the voltage
busVoltage = ina[index].getBusVoltage_V();

//get the current
current = ina[index].getCurrent_mA();

//calculate the power using Ohm's law
float power = busVoltage*(current/1000);
double newPow = 0.2841*power*power + 0.6191*power + 0.625;
return newPow;

C.4.2 FF Control Experiment Code

//Requires the Wire.h library, published by Arduino Inc. under
//the Creative Commons Attribution-ShareAlike 3.0 License.
//Requires the INA219 device driver library, published by Adafruit Inc.
//under the MIT License.
//Requires the SPI.h library, published by Arduino Inc. under
//the Creative Commons Attribution-ShareAlike 3.0 License.
//Requires the ESP32 DMA SPI library, published by Hideaki Tai
//under the MIT License.
//Requires the HX711.h library, published by Bogdan Necula
//under the MIT License.
//Requires the Wire.h library, published by Arduino Inc. under
//the Creative Commons Attribution-ShareAlike 3.0 License.

/*
Author: Vaughan Murphy
Purpose: Apply and test feedforward control on the
Woven Bi-Pennate Actuator.
*/
#include <ESP32DMASPI_Master.h>
#include <Wire.h>
#include <Adafruit_INA219.h>
#include <SPI.h>
#include <HX711.h>

// power related global variables
double power = 0;
Adafruit_INA219 ina[] = {
    Adafruit_INA219(0x40),
    Adafruit_INA219(0x41),
    Adafruit_INA219(0x44),
    Adafruit_INA219(0x45),
    Adafruit_INA219(0x42),
    Adafruit_INA219(0x46),
};

// load cell (force) related variables
int loadCellDout = 35;
int loadCellSck = 32;
double force = 0;
double calibration = 1;
HX711 scale;

// control related global variables
double targetTemp = 80; // temperature to be maintained
double tempMin = 30; // floor temperature at which point the TCA is cooled
int gatePin[] = {33, 26, 23, 18, 17, 4, 25, 27, 19, 5, 16, 2};
long maintainTimer = 0, timer = 0, initialForce = 0, controlStart = 0, pwmStart = 0;
bool isReached[] = {false, false, false, false, false, false, false, false,
false, false, false, false};
bool maintaining[] = {false, false, false, false, false, false, false, false,
false, false, false, false};

//general operation variables
int trialNum = 0, trialCap = 5;
bool heating = true; //this dictates if the trial is heating [true] or cooling [false]

//active set related variables
int activeSet[] = {0, 0, 0, 0, 0, 0};
int list[] = {0, 0, 0, 0, 0, 0}; //doing 5 trials per # pairs per actuator
//insert list as set of undemarcated active pairs
//i.e. 13 will have pairs 1 and 3 as a trial

void setup() {
// put your setup code here, to run once:
for(int i = 0; i < 12; i++){
    pinMode(gatePin[i], OUTPUT);
}
Serial.begin(9600);

//begin INA communications
for(int i = 0; i < 6; i++){
    ina[i].begin();
}
//get the load cell set up
scale.begin(loadCellDout, loadCellSck);
scale.set_scale(calibration);
scale.tare();

//exit SPI initialization
while(!Serial.available()){  
  //delay program until serial is triggered
}
Serial.println("Begin FF Force Control Experiment");

void loop() {
  // put your main code here, to run repeatedly:
  // If the trials have been completed, make sure to kill the program
  if(trialNum >= trialCap){
    Serial.println("End of experiment, trial number exceeded");
    while(1){
      //this infinite loop is to trap the controller once the trials are done
    }
  }
  //check which trial we're in right now, or if we're even in one
  bool inTrial = false;
  for(int i = 0; i < 6; i++){  
    if(activeSet[i] > 0){
      inTrial = true;
    }
  }
  //if we're not in a trial, let's get in one
  if(!inTrial){
getActiveSet();
}

else{
    // if we are in a trial, make sure to control the heating or cooling process
    // Need to output all data
    outputData();
    if(heating){
        // if hot, can apply control scheme, will have capacity to
        // control temperature rising and settling
        applyControl();
    }
    else{
        controlCooling();
    }
}

long errorSum = 0, prevError = 0;

// all of the coefficients for the PID controllers, from simulink simulations
double kp[] = {30.0, 3.5, 4.0};
double ki[] = {0.05, 0.10, 0.10};
double kd[] = {1.5, 0.90, 0.06};
// coefficients for the feedforward controllers, from simulink simulations
double ke[] = {2, 3.5, 6};
bool oscillating = false;
int targetForce = 200;  // change as required
// this function applies the control law to control the heating of the actuator
void applyControl(){
    long pwmInterval = 100, controlInterval = 5000;
int activeCount = 0;
double pwmFactor = 0;

// if it's the start of the control section, get initial values
if (controlStart == 0){
    initialForce = getForce();
    controlStart = millis();
    pwmStart = millis();
}

// get the number of active TCAs to be able to select the proper coefficients
for(int i = 0; i < 6; i++){
    if(activeSet[i] > 0){
        activeCount++;
    }
}

if (activeCount == 2) targetForce = 400;
if (activeCount == 3) targetForce = 550;

int tcaNum = activeCount - 1; // needs rectification for array selection

// if the control has lasted longer than the designated control interval
// begin cooling and reset initialization values
if (millis() > (controlStart + controlInterval)){
    heating = false;
    oscillating = false;
    controlStart = 0;
    pwmStart = 0;
}
else{
    // otherwise control the actuator
    // make sure to refresh the pwm start point
    if (millis() > (pwmStart + pwmInterval) || (pwmStart == controlStart)){
    
}
```c
pwmStart = millis();
// calculate the pwmFactor required

double t = (millis() - controlStart)/1000.0;
// get the time since control began in seconds

double curForce = force - initialForce;
// get the input target force based on the FF controller
// values obtained from simulink

double newTarget = targetForce*(exp(-1*ke[tcaNum])*(2*ke[tcaNum]+1));
// get the error in the system for the feedback component

double error = newTarget - curForce;
// calculate based on PID values found from simulations

pwmFactor = error*kp[tcaNum] + errorSum*ki[tcaNum] + (error-prevError)*kd[tcaNum];

prevError = error;
errorSum += error;

if (!oscillating && force > (0.95*targetForce)) {
    oscillating = true;
}

// calculate the feed forward component of the pwmFactor
// from the inverse Laplace functions

double ffComp = 0;
if (activeCount == 1) {
    double preEnd = (0.00795*cos(2.81*t)*exp(-0.943*t));
    double endBit = preEnd+(0.225*sin(2.81*t)*exp(-0.943*t));
    ffComp = (0.524*exp(-2*t))-(2.06*exp(-1.18*t))-endBit;
} else if (activeCount == 2) {
    double preEnd = (0.00178*cos(1.7*t)*exp(0.0553*t));
    double endBit = preEnd+(0.000956*sin(1.7*t)*exp(0.0553*t));
    ffComp = (0.012*exp(-3.5*t))-(0.0126*exp(-1.65*t))+endBit;
}
```
else if(activeCount == 3){
    double preEnd = (0.19*\sin(1.44*t)*exp(-0.526*t));
    double endBit = preEnd+(0.174*\sin(3.06*t)*exp(0.26*t));
    double midBit = (0.18*\cos(3.06*t)*exp(0.26*t));
    ffComp = (exp(-6*t))*(0.0878*\cos(1.44*t)*exp(-0.526*t))-midBit-endBit;
}
else{
    ffComp = 0;
}
ffComp = ffComp*targetForce;
pwmFactor += ffComp;
//limit the pwmFactor based on saturation (0-1.0)
if(pwmFactor > 1.0) pwmFactor = 1.0;
if(pwmFactor < 0.0) pwmFactor = 0.0;

//apply calculated pwmFactor value to a pwm heating scheme
//so long as pwmStart has been set
if(millis() < pwmStart + (pwmFactor*pwmInterval) && pwmStart != 0){
    //after start, for (pwmFactor) percent of the interval
    //heat the TCAs in the active set
    for(int i = 0; i < 6; i++){
        int left = activeSet[i] - 1;
        int right = left + 6;
        digitalWrite(gatePin[left], HIGH);
        digitalWrite(gatePin[right], HIGH);
    }
}
else{
for(int i = 0; i < 6; i++){
    int left = activeSet[i] - 1;
    int right = left + 6;
    digitalWrite(gatePin[left], LOW);
    digitalWrite(gatePin[right], LOW);
}

//this function controls the cooling process of the actuator
void controlCooling(){
    //cycle through every TCA, if they're all below the threshold
    //((set in global) tick over to a new trial
    int coldCount = 0;
    for(int i = 0; i < 12; i++){
        digitalWrite(gatePin[i], LOW);
    }

    //if it has cooled enough to meet its initial force, tick trial number
    //clear active set and revert to heating
    if (force <= initialForce){
        heating = true;
        trialNum += 1;
        timer = millis();
        for(int i = 0; i < 6; i++){
            activeSet[i] = 0;
        }
    }
}
//this function deciphers which TCAs are part of the current trial
//from the trial list (for control purposes)

void getActiveSet(){
    //get the current list element
    int temp = list[trialNum];
    //increment through each digit to get active TCAs
    activeSet[0] = (int)(temp/100000);
    temp-=activeSet[0]*100000;
    activeSet[1] = (int)(temp/10000);
    temp-=activeSet[1]*10000;
    activeSet[2] = (int)(temp/1000);
    temp-=activeSet[2]*1000;
    activeSet[3] = (int)(temp/100);
    temp-=activeSet[3]*100;
    activeSet[4] = (int)(temp/10);
    temp-=activeSet[4]*10;
    activeSet[5] = (int)(temp);
    //Output that we are starting a trial, and which pairs are active
    Serial.print("Begin Trial ");
    byte tempTrialNum = trialNum + 1;
    Serial.println(tempTrialNum);
    Serial.println("Active Set is [");
    Serial.print(activeSet[0]);
    Serial.print(", ");
    Serial.print(activeSet[1]);
    Serial.print(", ");
    Serial.print(activeSet[2]);
    Serial.print(", ");
    Serial.print(activeSet[3]);
    Serial.print(", ");
    Serial.print(activeSet[4]);
    Serial.print(", ");
    Serial.print(activeSet[5]);
    Serial.print("]");
}
C.4 Force Control Experiment Code

```cpp
Serial.print(activeSet[3]);
Serial.print(",");
Serial.print(activeSet[4]);
Serial.print(",");
Serial.print(activeSet[5]);
Serial.println(""]);
}

//this function outputs the data to the serial monitor to be transcribed
void outputData()
{
    //get the force and power
    force = getForce(); // - initialForce;
    //getting the power from the power function
    power = 0;
    for(int i = 0; i < 6; i++){
        float tempPower = getPower(i);
        power += tempPower;
    }
    //rectify power using characterization equation found experimentally
    if(power < 0){
        power = 0;
    }
    //get time stamp
    float curTime = millis()/1000.0;
    //output all values, separated by commas. Will go
    //[Time Stamp] [Force] [Total Power Consumption]
    //output time stamp
    Serial.print(curTime);
    Serial.print(",");
```
// output force
Serial.print(force);
Serial.print(",");

// output the power
Serial.println(power);

// total output will be a 3 element line
}

// this function gets the force being exerted on the load cell
double getForce(){
    // Returns the corrected weight found by the sensor
    // Uses the zero value found experimentally to get the calibration value
    // then divides sensor reading by calibration to get final force
    // based on zero'd value
    double reading = scale.read() * 1.0;
    double inputVal = reading - 78550.62;
    if(inputVal <= 200){
        calibration = 0;
        return 0.0;
    }
    else{
        calibration = 372.0513;
        return (inputVal*1.0)/calibration;
    }
}

double getPower(int index){
    float busVoltage = 0;
    float current = 0; // will be measured in milliamps
```cpp
// get the voltage
busVoltage = ina[index].getBusVoltage_V();

// get the current
current = ina[index].getCurrent_mA();

// calculate the power using Ohm's law
float power = busVoltage*(current/1000);

double newPow = 0.2841*power*power + 0.6191*power + 0.625;

return newPow;
```
VITA

Name: Vaughan Murphy

Post-secondary Education and Degrees: The University of Western Ontario, London, Ontario, Canada


Honours and Awards: 2015–2019 Dean’s Honour List

Awards: 2016 MROO Leadership Scholarship

2015 Reach for the Top Scholarship

2015 Western Scholarship of Excellence