Design and Implementation of a Novel Rehabilitation Robot for Acute Stroke Patients

by

Nicholas Berezny

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of
Master of Applied Science

Ottawa-Carleton Institute for
Mechanical and Aerospace Engineering

Department of Mechanical and Aerospace Engineering
Carleton University
Ottawa, Ontario, Canada
September 2019

Copyright © 2019
Nicholas Berezny
Abstract

Stroke is a prevalent cerebrovascular disease which leads to neurological deficits, and is a major cause of disability in adults. Stroke victims often lose motor control, affecting their ability to perform activities of daily life such as walking and grasping. Rehabilitation is used to help restore lost abilities through intensive and repetitive exercises with the assistance of a therapist. Robots have been introduced to help therapists administer therapy by replicating the role of physically assisting the patient. There is a gap in the literature regarding rehabilitation robots for bed-bound, acute stroke patients – this motivated the design and creation of the novel rehabilitation device presented in this thesis. The device builds off of previous work with the Virtual Gait Rehabilitation Robot (ViGRR), seeking to take the fundamental concepts and apply them to acute, bed-bound stroke rehabilitation. Observational fieldwork on the stroke ward at a local hospital informed the initial design, a linear 1-DOF robot targeting the knee flexion/extension exercise typical to traditional bed-bound rehabilitation. The device sits on the bed, under the leg, with the patient’s foot resting on a footplate. An admittance controller was developed to apply assistive or resistive forces to the patient’s leg. Modular real-time software was created to handle the controller, communication, and data logging. Three games were created which are used to make the exercise more engaging, along with haptic feedback control which can render virtual forces to the patient through the robot. The games run on a user interface, which is also used to set up rehabilitation sessions by selecting assistance levels, resistance level, and type of game. A series of experiments were run, including safety tests and functional tests which confirm that the controller and software are functioning correctly. We conducted an experiment on healthy subjects, which
looked to evaluate the effects of assistance and resistance levels, along with the participant’s view on the device through a subjective questionnaire. The experiments indicate that assistance does work to help the user follow the desired trajectory, that resistance can be used to make the task more difficult, and that the games have potential for making the exercise more engaging. However, there is still design work to be completed before the device is ready for clinical trials; recommendations for future work were enumerated and included.
Acknowledgements

I’d like to thank my supervisor, Mojtaba Ahmadi, for the opportunity to work on this project, and for all the assistance navigating grad school. I would also like to thank all the members of ABL for their support, advice, and for bravely volunteering as the first test subjects.
1 Introduction
   1.1 Stroke and Rehabilitation .............................................. 2
       1.1.1 Rehabilitation Guidelines .................................. 4
       1.1.2 Acute Stroke Rehabilitation .................................. 6
   1.2 Robotic Rehabilitation .................................................. 7
       1.2.1 Robots for Overground Walking ................................ 9
       1.2.2 Robots for Sitting or Bed-bound Therapy .................... 12
       1.2.3 Previous Work ..................................................... 13
       1.2.4 Stroke Recovery Measures ..................................... 14
       1.2.5 Controls ......................................................... 16
       1.2.6 Virtual Environments .......................................... 17
       1.2.7 Serious Games and Exergames ................................ 18
   1.3 Proposed Device Concept ............................................... 19
   1.4 Contributions and Outline ........................................... 20

2 Fieldwork and Initial Design ........................................... 22
   2.1 Introduction ............................................................ 22
   2.2 Fieldwork ............................................................... 23
       2.2.1 Methods .......................................................... 23
       2.2.2 Typical Rehabilitation ........................................ 24
       2.2.3 Patient Engagement ............................................ 24
2.2.4 Hospital Environment ............................................ 25
2.2.5 Therapy Schedule .................................................. 25
2.2.6 Other Comments and Concerns .................................. 25

2.3 Design Requirements ................................................ 26

2.4 Design Decisions .................................................... 27
  2.4.1 Degrees of Freedom ............................................. 28
  2.4.2 Linear Actuator Type .......................................... 29
  2.4.3 Placement Relative to Bed ..................................... 30
  2.4.4 Computer versus Microcontroller ............................ 31

2.5 Final Design .......................................................... 33

3 Device Design and Implementation ................................. 35
  3.1 Overview ............................................................. 35
  3.2 Structural Design .................................................. 35
    3.2.1 Frame ......................................................... 35
    3.2.2 Footplate ..................................................... 36
  3.3 Hardware ............................................................ 38
    3.3.1 Actuator ....................................................... 38
    3.3.2 Sensors ........................................................ 41
  3.4 Electronics and Power ............................................. 42
    3.4.1 PCB Layout ................................................... 44
  3.5 Data Acquisition .................................................. 45
  3.6 Computer System .................................................. 47

4 Controller and Software ............................................. 50
  4.1 Overview ............................................................ 50
  4.2 Impedance & Admittance Controller ............................. 50
    4.2.1 Velocity Profile .......................................... 55
    4.2.2 Operational Modes Using Admittance Control .......... 56
<table>
<thead>
<tr>
<th>Section</th>
<th>Heading</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3</td>
<td>Haptic Feedback</td>
<td>57</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Operational Modes Using the Haptic Coupling</td>
<td>62</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Physics Engine</td>
<td>62</td>
</tr>
<tr>
<td>4.4</td>
<td>Tuning Control Gains</td>
<td>63</td>
</tr>
<tr>
<td>4.5</td>
<td>Software</td>
<td>66</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Real Time Operation and Linux</td>
<td>66</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Controller</td>
<td>68</td>
</tr>
<tr>
<td>4.5.3</td>
<td>User Interface</td>
<td>71</td>
</tr>
<tr>
<td>5</td>
<td>Experiments</td>
<td>77</td>
</tr>
<tr>
<td>5.1</td>
<td>Overview</td>
<td>77</td>
</tr>
<tr>
<td>5.2</td>
<td>Safety Tests</td>
<td>77</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Goals</td>
<td>77</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Method</td>
<td>79</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Results &amp; Discussion</td>
<td>79</td>
</tr>
<tr>
<td>5.3</td>
<td>Functional Tests</td>
<td>80</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Goals</td>
<td>80</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Method</td>
<td>80</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Results &amp; Discussion</td>
<td>81</td>
</tr>
<tr>
<td>5.4</td>
<td>Healthy Subject Experiments</td>
<td>83</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Goals</td>
<td>83</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Method</td>
<td>84</td>
</tr>
<tr>
<td>5.4.3</td>
<td>Results &amp; Discussion</td>
<td>89</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion &amp; Future Work</td>
<td>96</td>
</tr>
<tr>
<td>6.1</td>
<td>Design Requirement Outcomes</td>
<td>97</td>
</tr>
<tr>
<td>6.2</td>
<td>Future Work</td>
<td>100</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Design Additions</td>
<td>100</td>
</tr>
<tr>
<td>6.3</td>
<td>Software Extension</td>
<td>101</td>
</tr>
</tbody>
</table>
6.3.1 User Interface Additions ........................................ 102
6.3.2 Experiments .................................................... 102
List of Figures

1.1 The leg lift exercise for bed-bound rehabilitation .......................... 4
1.2 The knee extension/flexion exercise for bed-bound rehabilitation ........ 5
1.3 The stages of traditional stroke rehabilitation .............................. 5
1.4 Stroke rehabilitation guidelines for scheduling therapy, and potential adjuncts 6
1.5 An endplate rehabilitation device (left) and an exoskeletal device (right) . . 9
1.6 The Lokomat (by Hocoma), an exoskeletal lower-limb robot .................. 10
1.7 The Hapticwalker, an endplate based robot for gait rehabilitation .......... 11
1.8 The MotionMaker for reclined lower-limb rehabilitation ...................... 13
1.9 The Lambda rehabilitation robot with VR setup ............................... 14
1.10 The Virtual Gait Rehabilitation Robot, precursor to the new device ......... 15

2.1 Design process for the new rehabilitation robot .............................. 23
2.2 Off-bed robot design ................................................................... 30
2.3 On-bed robot design .................................................................... 31
2.4 Forces and moment arms for on-bed design .................................... 32
2.5 Forces and moment arm for off-bed design ..................................... 32
2.6 The rehabilitation robot on the test bed ......................................... 33
2.7 The rehabilitation robot with user’s leg ......................................... 34

3.1 Rendering of the rehabilitation robot ........................................... 36
3.2 The structure of the frame ........................................................... 37
3.3 The structure of the footplate ....................................................... 38
List of Tables

<table>
<thead>
<tr>
<th>Table Number</th>
<th>Table Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Design categories and decisions based on the fieldwork</td>
<td>28</td>
</tr>
<tr>
<td>2.2</td>
<td>Benefits of Actuator Types (x = benefit)</td>
<td>30</td>
</tr>
<tr>
<td>2.3</td>
<td>Structural Configuration</td>
<td>30</td>
</tr>
<tr>
<td>3.1</td>
<td>DC Motor Properties</td>
<td>40</td>
</tr>
<tr>
<td>3.2</td>
<td>Belt Drive Properties</td>
<td>40</td>
</tr>
<tr>
<td>3.3</td>
<td>System Maximums</td>
<td>41</td>
</tr>
<tr>
<td>3.4</td>
<td>T7 DAQ Properties</td>
<td>47</td>
</tr>
<tr>
<td>3.5</td>
<td>Computer Specifications</td>
<td>47</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparing Impedance and Admittance Control</td>
<td>54</td>
</tr>
<tr>
<td>4.2</td>
<td>Control gains and parameter limits</td>
<td>66</td>
</tr>
<tr>
<td>5.1</td>
<td>Summary of the Experiments</td>
<td>78</td>
</tr>
<tr>
<td>5.2</td>
<td>Measures across the range of assistance and resistance</td>
<td>89</td>
</tr>
<tr>
<td>5.3</td>
<td>Subjective Scores for the Three Games</td>
<td>92</td>
</tr>
<tr>
<td>5.4</td>
<td>General Subjective Scores</td>
<td>94</td>
</tr>
<tr>
<td>6.1</td>
<td>PCB Components</td>
<td>117</td>
</tr>
</tbody>
</table>
Acronyms

ABL Advanced Biomechatronics and Locomotion Laboratory
ADL Activity of Daily Living
ALEX The Active Leg Exoskeleton
BWSTT Body-weight Supported Treadmill Training
DOF Degree of freedom
FM Fugyl Meyer Scale
LOPES The Lower Extremity Powered Exoskeleton
MAL Motor Activity Log
MUTEX Mutual Exclusion Object
OS Operating System
OT Occupational Therapist
PCB Printed Circuit Board
POSIX Portable Operating System Interface
PT Physiotherapist
STREAM Stroke Rehabilitation Assessment of Movement
TCP Transmission Control Protocol
TUG Timed Up and Go
UI User Interface
ViGRR Virtual Gait Rehabilitation Robot
Chapter 1

Introduction

Suffering a stroke can be devastating to one’s autonomy and quality of life. Often, the resulting motor impairment will degrade the victim’s ability to perform activities of daily living – things we rely on everyday such as getting dressed, eating, and walking. Rehabilitation is administered by physiotherapists to stroke victims to help them restore their motor control and abilities.

Rehabilitation robots have been a topic of research for several decades. They show promise as tools for therapists to help improve stroke rehabilitation. Among lower-limb rehabilitation robots, most target out-of-bed activities such as gait trainers. Others target reclined activities, where the patient can sit in a chair and perform stationary leg exercises. There is a gap in the research regarding devices targeting bed-bound patients, or devices which are compatible with the hospital bed and can be used to deliver the standard rehabilitation exercises for immobile patients. This type of device would primarily benefit patients in the early acute phase of stroke, which occurs within the first 2 weeks of stroke onset. Rehabilitation during this phase is crucial and can greatly affect how well the patient recovers.

This gap in the literature is not due to oversight, however is more likely due to the many issues with building, testing, and implementing a rehabilitation robot for acute stroke patients. Acute stroke patients lack motor control, can have significant muscle spasticity,
and also often lack muscle strength. The utmost care must be taken to ensure the robot does not harm the patient. Since the robot will be largely responsible for driving the motion and supporting the patient’s limb, both the physical structure and the controller must be designed to ensure safe interaction. Gathering meaningful data from clinical trials can also be difficult – patients who have recently suffered severe stroke may be reluctant to participate in an extensive study, and obtaining ethics approval to do so can be lengthy due to the risks involved.

This research looks to determine if the successes from the broader field of rehabilitation can be applied to bed-bound, acute stroke patients. One significant point of inspiration is the Virtual Gait Rehabilitation Robot (ViGRR), a test platform developed at the Advanced Biomechatronics and Locomotion Laboratory. This device successfully incorporates ideas such as using admittance control to provide assistance, and using haptic feedback and virtual environments to increase engagement. This projects seeks to take these aspects and apply them to a smaller, more lightweight device which specifically targets acute patients with bed-bound exercises. To overcome the difficulty associated with acute stroke rehabilitation, we have and will continue to work closely with doctors and therapists with experience on the stroke ward. With their help, we have begun designing a device which will soon be ready for clinical trials with stroke patients. This device, and the results from the clinical trial, will be novel – to our knowledge, no robots exist which explicitly target the range of acute, bed-bound rehabilitation exercises.

This thesis presents the first stages of this process – collecting initial design requirements by working with physiotherapists, designing and implementing the prototype which incorporates previously proven concepts such as admittance control and virtual environments, and finally taking the first steps before clinical trials by testing the device with healthy subjects.

1.1 Stroke and Rehabilitation

Stroke is a cerebrovascular disease which affects approximately 62,000 Canadians and 795,000 Americans each year [1,2]. It can result in limited mobility and muscle control [3], with
some form of motor impairment affecting 80% of patients. Due to these complications, it is among the leading causes of adult disability [4]. Ischaemic stroke is the most common form, accounting for around 80% of all cases [5]. It is caused by blood vessel occlusion, either directly in the brain (thrombotic), or due to the migration of a clot formed somewhere else in the body (embolic). The subsequent lack of oxygen causes the death of brain tissue and neurological deficits [6], which manifest as functional impairments in activities associated with the affected brain region. Stroke complications can include hemiparesis (weakness on one side of the body), muscle spasticity, loss of motor control, and loss of dexterity, all of which can effect the patient’s ability to perform activities of daily living (ADL’s) and compromise their autonomy. In addition to medical intervention in the form of pharmaceuticals and surgery, the stroke patient may require rehabilitation to reduce functional impairment [7]. It is imperative to administer treatment during the acute phase of the stroke (i.e. within days of the stroke) [6].

Older treatment techniques relied on compensatory measures to assist stroke patients in regaining autonomy [8]. Using rehabilitation to restore autonomy began to be emphasized once the concept of neural plasticity became more widely accepted. Neural plasticity is the process by which new connections are formed in the central nervous system in response to experience or activities performed by the subject [9]. When a stroke patient repetitively performs motor tasks, new neural connections in the brain are formed which take on the role of the regions of the brain which were damaged. This allows the patient to re-learn tasks such as walking by going through intensive rehabilitation.

Stroke rehabilitation encompasses a broad range of therapeutic activities, typically carried out by occupational- and physiotherapists. Physiotherapists (PT’s) tend to focus on gross motor movements in the upper and lower limbs, while occupational therapists (OT’s) focus on fine motor control, dexterity, and specific ADL’s. These two goals are inextricably linked, and so both PT’s and OT’s form a team which works together to improve patient outcomes. The rehabilitation carried out by both OT’s and PT’s involves the repetition of a target movement or task. Depending on the abilities of the patient, the therapist may
provide support or assistance. For example, lower-limb rehabilitation begins with simple leg movements performed in bed, such as leg lifts (Fig. 1.1), knee flexion/extension (Fig. 1.2), hip abductions, etc. The therapist can either assist the patient’s leg through the movement, or allow patients to perform the movement on their own, or even provide light resistance if the patient is more advanced. Next, the patient will practice sitting upright, going from sit to stand, and practice bearing weight and balancing while standing. Eventually the patients will be ready for more complex tasks with assistance from the therapist, including walking, heel strike, and stair climbing. The patient may advance to the point where they can walk without therapist assistance, using either parallel bars, a walker, or potentially without any assistance. The stages of lower-limb rehabilitation are summarized in Fig. 1.3.

1.1.1 Rehabilitation Guidelines

Canadian stroke best practices recommend that therapy involves the repetition of intensive, task-specific activities in a complex and stimulating environment [2]. Lower-limb rehabilitation should be goal-orientated towards selected tasks such as walking and sit-to-stand. Robotics, virtual reality, and biofeedback are all mentioned as potential adjuncts, but are not recommended to replace traditional therapy. Active participation on the part of the patient is also imperative [4], as it has been found that this facilitates neural plasticity.
This requires that the patient be engaged in the therapy (hence the need for stimulating environments). A summary of the guidelines is given in Fig. 1.4.

Therapy dosage also has an effect on functional outcomes. For example, positive correlation between time scheduled for therapy and most functional outcomes was found in [10]. Despite this, many patients do not receive enough therapy. For example, an animal model was used in [11] to determine if the doses of rehabilitation they observed at a hospital were enough to trigger functional improvement, and found the amount of rehabilitation inadequate. Stroke inpatients spend on average 13% engaged in physical activities and 5.2% of their time with a therapist at one hospital researched in [12]. Patients were observed to spend a significant amount of time in or sitting near their bed in [13]. These results indicate the importance of scheduling time specifically for rehabilitation for stroke inpatients.
Other factors may restrict the PT’s and OT’s ability to accomplish the recommended best practices. The intensity and duration of the therapy is limited both by the endurance of the patient and of the therapist, as the physical burden of manually assisting the patient may be high [14]. Furthermore, providing task-specific and stimulating therapy is difficult, particularly when doing bed-bound exercises which are not easily relatable to ADL’s. One study found that therapy dosage may be affected by understaffed units, resulting in less rehabilitation time with therapists [15]. There is an opportunity to introduce new methods or technologies to assist therapists in delivering more rehabilitation.

1.1.2 Acute Stroke Rehabilitation

The acute phase of stroke is usually defined as the stages within several days of stroke onset. The exact time period varies, from within the first 24 hours in [16], to within the first 3 weeks in [17]. For the purposes of this thesis, acute phase is defined as the early stages of recovery, before the patient has been mobilized and while they are still in-patients on the stroke ward, similar to the definition given in the Canadian Stroke Best Practices [18].

The benefits of beginning rehabilitation during the acute phase are emphasized in many studies. The Canadian Stroke Best Practice Publication recommends that patients should receive rehabilitation as early as possible once they are deemed medically ready. Early mobilisation is recommended (within 24-48 hours of stroke, [18], but not before 24 hours due to findings that this actually reduces functional outcomes, [19]). In [20], stroke in-patients at five rehabilitation facilities are followed through their recovery. Various measures of recovery
indicate that earlier initiation of rehabilitation resulted in better outcomes.

Increasing rehabilitation time during the acute phase can be difficult. Many patients are bed-bound, so rehabilitation is restricted to activities done while reclining, which are difficult to make engaging and relevant to ADL’s. Bed-bound patients typically have not recovered motor coordination, and so they rely on therapists for physical assistance. Due to the importance of acute phase rehabilitation, and the difficulties associated with it, there is an opportunity to introduce new technologies which have an impact on patient outcomes.

1.2 Robotic Rehabilitation

Rehabilitation robots have been introduced to improve the stroke rehabilitation process by automating the physical assistance usually supplied by therapists. Rehabilitation requires the repetition of a precise motion, a task well-suited to robotics. Over the past few decades, a variety of robotic platforms have been developed for many different therapeutic activities, from lower-limb gait training to fine motor skills in the hand. There are many other potential benefits for using robotics for rehabilitation, including the following.

- **Relieving Physical Burden**: Using robots to provide the physical support and assistance required for stroke therapy will relieve the physical burden from therapists, removing the chance of injury or fatigue.

- **Increased Intensity and Duration**: Both the intensity and duration of a therapy session may be limited when relying on a therapist to provide physical assistance. With a robot, duration and intensity would only be limited by the endurance of the patient.

- **Increased Dosage**: Robots could also increase the amount of rehabilitation given to patients. For example, a single therapist could supervise several patients each using a robotic device, therefore increasing the number of patients a therapist can manage, thereby increasing the dosage each patient receives.

- **Providing Performance Measures**: Robots record a number of metrics not avail-
able when using traditional therapy, such as velocity, force, trajectory error, and effort. Tracking these measures could allow the therapist to better administrate therapy and track patient improvement.

- **Virtual Environments**: Patient engagement could be increased by incorporating virtual reality, games, and other forms of visual feedback. The robot would serve as the control input to the environment. The relevancy of the therapy could also be improved by simulating ADL’s. Finally, haptic feedback could further immerse the patient in the virtual environment by rendering virtual forces to the patient (see section 1.2.6).

Rehabilitation robots can be broadly categorized as targeting the lower-limb or the upper-limb. Upper-limb robots tend to be lighter weight since they encounter much smaller loads, while lower-limb devices are heavier as they need to support either the weight of the leg or of the entire body. Lower-limb robots can be further categorized as either targeting standing activities such as treadmill walking, or lying/sitting activities [21]. A final subdivision exists between endplate based devices and exoskeletal devices (Fig 1.5). Endplate-based devices interact with the patient at a single point, usually the end of the limb (the hand or foot), and only apply forces through this point. Exoskeletal devices attach to the user at multiple points, offering greater control over the overall motion of the limb, although they are often more restrictive, more complex, and potentially less safe [22]. One study found evidence that endplate-based devices may actually yield better outcomes than exoskeletal devices, although the comparison was done via meta-analysis of many different studies with significantly different methodologies [23].

Rehabilitation robots have not yet been fully endorsed by stroke best practices. For example, Canadian best practices list robots as a potential benefit, but do not recommend they replace traditional therapy [2]. This is largely due to the limited evidence for outcomes of robotic rehabilitation. Many studies are small scale case studies, and the few large clinical trials have found conflicting results. One study following 63 patients compared using the Lokomat (see Sec. 1.2.1) to conventional therapy, and found that the robot was less effective [24]. There are many challenges in studying the efficacy of robots in a clinical setting. It is
unethical to replace traditional therapy with an unproven method, so typically extra therapy is added to the subject’s regimen either from a robot or from a therapist (control). Since most robots are designed to replace time spent in conventional therapy, the robot must consistently be shown to be at least as effective as a therapist at delivering rehabilitation. Furthermore, since most robotic devices are substantially different, results from the study of one lower-limb robot may not apply to another robot. All of this makes “proving” the efficacy of a rehabilitation robot difficult. That said, there have been many different robotic devices successfully tested, with some even being available commercially.

1.2.1 Robots for Overground Walking

One of the most important activities for lower-limb rehabilitation is gait training, wherein the patient walks with assistance from the therapist. Rehabilitation robots were introduced to provide this assistance by attaching to the patient’s leg and guiding them through the correct gait trajectories. This can be accomplished by an exoskeletal orthosis attached around the knee joint, or by an endplate device which supports and guides the feet.

The Lokomat (Fig 1.6) is a powered orthosis that attaches to the patient’s leg, and works in conjunction with body-weight supported treadmill training (BWSTT). It uses impedance control to provide programmable assistance to the leg through gait trajectories. The Lokomat is one of more widely researched platforms. One recent study found that the Lokomat was
more effective than conventional gait therapy with regards to certain outcome measures \cite{25}, although another found that it was less effective \cite{24}. Despite conflicting results, the Lokomat continues to show promise. It is commercially available – 872 devices can be found worldwide according to their website (as of 2019).

![Lokomat](image)

**Figure 1.6:** The Lokomat (by Hocoma), an exoskeletal lower-limb robot

The ALEX \cite{26} is an exoskeletal robot used for treadmill gait training. It consists of an 8-DOF orthosis, and a harness to stabilize the user. Its controller provides assistance-as-needed using a force-field scheme, which only applies force when the user deviates from a desired trajectory. This is done by creating “virtual walls” at a set distance around the trajectory. They also use friction compensation to make the device backdrivable, to ensure the motion is transparent. A small study on healthy subjects showed that the force-field controller improved the user’s ability to perform the trajectory.

The LOPES \cite{27}, or the Lower Extremity Powered Exoskeleton, is gait rehabilitation robot similar to the Lokomat and the ALEX. It uses impedance control to provide modes like patient-in-charge and robot-in-charge. The device has three DOF’s per leg, and also
contains a pelvic support system to hold the patient up, which is itself also actuated by three linear DOFs. A small study compared EMG data from normal treadmill walking with treadmill walking with the LOPES, and found that both produced similar data, suggesting unhindered walking is possible with the LOPES.

The ReoAmbulator is another powered orthosis for treadmill training used to assist with BWSTT [28]. This device differentiates itself from previous devices by its “sanitary construction”, standard electrical connections, patient accessibility, and only requiring a single trained operator; all essentially making the device more compatible with the hospital and easier to use.

The HapticWalker [29] is an endplate-based gait trainer with two actuated platforms that can move the user’s leg through the sagittal plane. The system has 3-DOF per leg, but can be extended to 6-DOF to allow for a greater range of motions. The control software is implemented in RTLinux, and includes trajectory generation and a user interface. As can
be seen in Fig. 1.7, the HapticWalker, along with other endplate devices targeting standing activities, tend to be much larger than and also require more power.

The G-EO is an endplate-based robot designed for the repetitive practice of floor walking and stair climbing [30]. Similarly to the HapticWalker, it has two actuated platforms which can guide the patient’s legs through trajectories. This robot uses position control, which does not necessarily encourage patient engagement.

Overground walking robots show promise for improving gait rehabilitation. However, these only target a single type of lower-limb rehabilitation. Patients must be somewhat mobile (enough to leave their beds and their rooms) to access these devices, and the complexity and setup require that a therapist be present. Hence these robots are designed to replace time spent on conventional therapy, which makes proving consistent performance difficult as was previously mentioned.

1.2.2 Robots for Sitting or Bed-bound Therapy

Some types of rehabilitation are administered while the patient is sitting or reclined in bed, often for acute stroke patient’s who are not yet mobile. Robots for assisting in activities while the patient sits or lies down are less common in the literature. The MotionMaker, Lambda, the Rutger’s Ankle and the Virtual Gait Rehabilitation Robot are some examples.

The MotionMaker (Fig. 1.8) consists of a reclinable chair and two orthoses, which attach to the leg only at the foot [31]. It uses a regulator with force feedback to control the exercise. It is designed to be used in conjunction with functional electrical stimulation for spinal cord injury patients, but it’s application extends to lower-limb rehabilitation in general.

The Lambda (Fig. 1.9) is robot developed for general lower-limb rehabilitation and for fitness purposes [32]. It uses parallel linkages with a total of three degrees of freedom to achieve a variety of motions. An adjustable seat allows the user to recline as needed. Recent developments include the addition of virtual reality and games to engage the patient, enhanced feedback regarding the patient’s progress, and multiple types of exercises.

The Rutger’s Ankle [33] is an endplate based robot used to rehabilitate the ankles. It’s
Figure 1.8: The MotionMaker for reclined lower-limb rehabilitation
design is based on the Stewart platform, in that it is actuated by six linear actuators to
achieve 6 DOF’s. The foot is attached to the top plate, and a 6-DOF force sensor is used to
measure user input. The system is coupled with a virtual environments, each which focus
on a different therapeutic activity. The goals set in the virtual environments are meant
to motivate patients to perform the correct rehabilitation activity. The device is designed
for at home use. While initial experiments showed some patients became fatigued, or were
confused by the virtual environments, the overall results were positive.

1.2.3 Previous Work

Carleton University’s Advanced Biomechatronics and Locomotion Laboratory (ABL) has
developed devices and robots for rehabilitation. The precursor to this project is the Vir-
tual Gait Rehabilitation Robot (ViGRR), a device developed at Carleton University’s Ad-
vanced Biomechatronics and Locomotion Laboratory [34,35]. It is a 4-DOF end-plate based
robot which can assist a reclined patient moving through trajectories in the sagittal plane
(Fig.1.10). The user’s foot interfaces with a footplate which is magnetically attached to the
The magnet can be released if triggered by a safety system, including high force, high velocity, or pressing an emergency stop button. An admittance controller is used to provide assistance-as-needed, roughly in proportion to the trajectory error. These desired trajectories can include gait trajectories, linear motion, circular motion, or any other path in the sagittal plane. To increase user engagement, a virtual environment was created which can display visual feedback regarding the desired trajectory, or games such as pushing a block or brick breaker. Users can be further immersed in the virtual environment through haptic feedback. For example, when pushing on virtual wall, the robot will apply a force to the user’s foot emulating the wall reaction force.

### 1.2.4 Stroke Recovery Measures

A stroke patient’s progress is monitored through tests, surveys, and measures. These are important because they help the therapist determine the best rehabilitation routine, and also can be used in research to compare rehabilitation techniques. The Fugl-Meyer (FM) scale is a 226-point test used to directly measure recovery after stroke \[^{36}\]. It covers both upper and lower limbs, and takes approximately 30 minutes to administer. The Motor Activity Log (MAL) is similar to the FM, however it is used specifically to test motor capacity in the
Other measures commonly used for rehabilitation monitoring include the Berg Balance Scale, Barthel Index, Stroke Rehabilitation Assessment of Movement (STREAM), and the Timed Up and Go (TUG) \[38\]. The Berg Balance scale is used to describe the patient’s ability to remain balanced during different tasks. The Barthel index is used to rate the patient’s ability to perform a number of ADL’s, such as dressing and grooming. Rehabilitation progress is directly monitored in STREAM, which takes into account mobility, range of motion, and coordination. The TUG score measures the time taken to stand up out of a chair, walk 3 metres, return to the chair, and sit down again. This measures mobility, balance, and also fall risk.

Robots can record measures unavailable to traditional therapy, including measures of position, velocity, and force, as well as measures derived from these like error, effort, smoothness of movement, etc. One study compared robot-based measures and traditional rehabilitation measures, specifically they looked at smoothness of movement and trajectory error versus the FM and MAL \[39\]. They found significant agreement between the robot-based measures and the FM score, and moderate agreement against MAL, suggesting that robot-based measures
can be used to augment the traditional measures for monitoring therapy progress.

Robot measures could add to traditional measures, providing more information on the patient’s recovery. They could, if the correlation seen in [39] holds, be used to replace traditional measures, saving the therapist valuable time (as in the FM, which takes 30 minutes to administer).

1.2.5 Controls

The type of controller is especially important for devices interacting with a human (or any unknown environment), as it will determine the interaction behaviour and stability. The behaviour of the interaction should be optimized to ensure the patient engages in meaningful therapy driving neural plasticity and recovery. Position control through the desired trajectory is not adequate, as the robot will effectively force the limb through the motion, thus becoming a passive motion machine with no necessary engagement from the patient. Neural plasticity requires engagement, i.e. the patient needs to contribute to some extent to the movement of their limbs. Therefore, rehabilitation robotics should only give some assistance to the patient, instead of leading the motion.

Three popular control laws for rehabilitation robotics are listed in [40]:

- **Impedance and admittance** control were introduced by Hogan in [41], and have been the foundation of most interaction controllers since. These controllers determine the dynamic interaction between robot and environment instead of regulating position or force. In this way, an impedance controller can be used to give a programmable amount of assistance to the user through a trajectory, which can be varied by changing the parameters so that an optimal amount of assistance can be obtained for each individual. The desired impedance or admittance is defined in terms of a mass-spring-damper system. Therefore, changing the spring gain will change the level of assistance. The damper gain can be used to resist motion if the patient is more advanced or if strength training is the goal. This control scheme is explained in detail in Sec. 4.2.
• **EMG-based** control uses electromyography (EMG) to measure muscle activity in the patient. The robot then only provides assistance when the intent to move is detected (*i.e.* the patient flexes their muscle). This ensures that the patient is actively engaged in the therapy. For example, the ankle-foot orthosis described in [42] uses processed EMG signals to drive pneumatic actuation, using control inputs proportional to the EMG activation.

• **Adaptive** control here refers to a controller which changes its behaviour over time based on some predefined law. For example, the amount of assistance supplied by an admittance controller can be changed based on patient performance. Some robots use information from movements in the patient’s healthy limb to calculate the optimal control path for the affected limb. Others use assist-as-needed control schemes to only provide assistance when the patient requires it. Artificial Neural Networks have been used to determine when the assistance is needed in robots such as the ALEX and LOPES.

### 1.2.6 Virtual Environments

Virtual environments or virtual reality refer to digitally created simulations or visualizations, typically displayed through headsets or on computer monitors. It combines visuals, auditory stimulation, and haptic feedback to create an immersive experience. Virtual reality has been introduced into rehabilitation for two primary reasons [43]:

• **Increase Relevancy**: simulate ADL’s and real world activities to introduce context into the exercise, and better prepare the patient for these activities outside of the hospital environment.

• **Increase Engagement**: use games and other visuals to make the therapy more exciting and engaging, encouraging more active participation, higher intensity, and higher doses of activity.
Virtual reality can be coupled with rehabilitation robotics to further increase immersion. The robot can be used as a control input to the virtual environment, and the robot can in turn deliver haptic feedback to the user. Haptic feedback allows users to interact directly with the virtual world; for example when walking, the ground reaction force can be applied to the user’s foot. This increases the realism of ADL simulations.

A metastudy found evidence that VR may improve functional outcomes over conventional therapy, but admits that most studies are small and prone to bias [43]. One platform used an upper-limb exoskeletal device to apply assistive forces [44]. Another study used a Nintendo Wii gaming system to promote therapy, and found it to be a feasible and safe method [45]. Another study used the Phantom Omni, a pen attached to a robotic arm that can deliver haptic feedback from a virtual environment [46]. A serious gaming approach is used in [47], where the Unity3D engine was used to create a virtual world and a MS Kinetic sensor to detect the user’s motions.

1.2.7 Serious Games and Exergames

Serious games are digital games used for purposes other than entertainment such as education or obtaining skills through simulation [48]. When designing these games, it is important to make them both attractive to users, and effective at encouraging the desired activity [49]. An example of the successful application of serious games is the prevalence of exergames, or serious games applied to exercise. Exergames were linked to increased childhood activity and physical fitness in [50].

Rehabilitation robots have also sought to leverage serious gaming to engage the patients in therapy. For example, a tele-robotic system for home based rehabilitation has been designed which uses games to encourage children with cerebral palsy to participate in the activity [51]. Many other rehabilitation robots use some form of visual feedback or games, and there is promise that similar results can be obtained. However, there are other questions which require further research, like how well serious games work with adults and stroke victims.
1.3 Proposed Device Concept

This rehabilitation robot prototype presented here should align with ViGRR’s fundamental concept: a lower-limb robot capable of assisting stroke patients with rehabilitation. It should do this with specific application to bed-bound, acute stroke patients. The device should be a viable tool for therapists within the hospital to use to increase functional outcomes for their patients. It should replicate the role of the therapist when providing assistance for bed bound exercises. There are three primary roles:

1. Support the leg
2. Provide assistance through the motion
3. Engage the patient

Supporting the leg will be accomplished through the physical structure of the device. Assistance will be provided using force feedback and interaction controllers which can ensure that the patient is contributing to some extent. The patient will be engaged using virtual environments and haptic feedback.

There are several use cases for the device. The therapist could use it during therapy instead of manually moving the legs. This would relieve them of the physical burden, the virtual environments may encourage the patient to participate more, and the data collected could be used to track progress. The therapist could use multiple devices simultaneously to provide care to multiple patients. This would allow them to see more patients and thus increase therapy dosage. Finally, since the device replicates the role of the therapist, the device could be used separately from the therapist, under the supervision of other hospital staff. This would encourage patient’s to practice independently, thereby increasing therapy dosage.

In addition to these requirements, the device should be usable within a hospital environment. The device should be safe such that the patient is not harmed, and should also be perceived as safe so that the therapists and patients are willing to use it. The device should
fit on a hospital bed, and should be lightweight and small enough to be manageable within a hospital room. The set-up of the device should be quick and easy.

1.4 Contributions and Outline

A 1-DOF robot was designed and created according to the requirements determined from the literature review, shadowing, and previous work on ViGRR. The device consists of a DC motor, force sensor, belt drive, footplate, frame, and a number of smaller components. A printed circuit board (PCB) was created to route signals, preprocess signals, and distribute power. System software was implemented on a real-time linux system. A controller was created with the programming language C, which can operate the device using admittance control, haptic feedback, a basic physics engine, communication with a UI server, and data logging. Real-time functionality is ensured using POSIX standards for memory locking, multi-threading, etc. The UI is implemented as a website using the React framework, and some simple games are created using the THREE.js 3D web graphics library. The controller was designed as a modular system that can be easily re-factored to accommodate design changes in future iterations. The device was evaluated through a number of tests and experiments, including functional tests, safety tests, and a pilot study with healthy subjects. The results show that the device functions as planned, but that several things must be changed/added before it is ready for patient testing in the hospital.

- Chapter 2 covers the fieldwork done on the stroke ward with therapists, including interviews and observation of typical stroke rehabilitation. The results are summarized and the influence on the device design are given. Preliminary design choices and the design requirements are defined.

- Chapter 3 covers the design of the device. This includes the design of the physical structure, the actuation and sensors, the design of the electronics (power distribution, signal routing and preprocessing), and the chosen computer hardware.
• Chapter 4 covers the controller and its implementation in the software, including the basic admittance control, and the haptic feedback with physics engine. The design of the user interface (UI) and some basic games/visualisations are also discussed. The communication and the overall software design within real-time Linux is explained, including how the modular design will allow it to be easily adapted to future design iterations.

• Chapter 5 covers testing and experiments, including functional testing of the software, controller, and UI. The methods and results from a small study on healthy subjects are presented, with questionnaire answers regarding the subjective experience of participants, along with quantitative performance metrics.

• Chapter 6 concludes with a summary of contributions and recommendations for future work. This includes features that were required which have not yet been implemented due to time constraints, and ideas for future directions of the project.
Chapter 2

Fieldwork and Initial Design

2.1 Introduction

The overall design process is shown in Fig. 2.1. After literature review and previous work done on ViGRR, the next step towards designing the new robot is an analysis of hospital requirements, including an evaluation of the hospital environment, of traditional acute stroke therapy, and of the needs of the therapists.

The rise of the human-centred design paradigm underscores the importance of involving end users in the design process, especially when the designers lack expertise in the field of interest. After the literature review, the next step of the design process was to consult the end user and collect qualitative data and advice. In the case of rehabilitation robots, there are two types of end-users: the stroke patient using the device, and the therapists who decide when and how to incorporate the device into their rehabilitation regimen. The therapist, however, has greater expertise with stroke rehabilitation in general, and will ultimately be the one who decides the device’s fate on the stroke ward. Therefore, it was decided to interview and shadow several PT’s and OT’s concerning the direction of the project. Patients were not consulted, due to their lack of expertise in stroke rehabilitation, although their input will be vital in future evaluation experiments. This chapter covers fieldwork done at the Ottawa Civic Hospital, where we investigated stroke rehabilitation practices and sought advice from
experienced stroke therapists. It also covers the initial design decisions based in part on the results from the fieldwork and from the literature review. The complete rehabilitation robot is introduced, with more technical details available in the next chapter.

![Design process for the new rehabilitation robot](image)

Figure 2.1: Design process for the new rehabilitation robot

### 2.2 Fieldwork

#### 2.2.1 Methods

Fieldwork was conducted at the Civic campus of the Ottawa Hospital (TOH), in cooperation with Dr. Dariush Dowlatshahi, and with ethics approval from the Ottawa Health Science Network Research Ethics Board (OHSN-REB). Three PT’s and two OT’s were recruited for two days of interviews and shadowing. Inclusion criteria included experience working on the stroke ward delivering rehabilitation to stroke patients. Not enough data was collected to perform any meaningful statistic – instead, key insights guided the fundamental concept of the device, and information was gathered not otherwise available through a literature review. Some topics of interest included:

- Typical acute, bed-bound rehabilitation
- Patient engagement – how the therapists keep patients motivated and actively participating
- The hospital environment
- The therapists’ views on rehabilitation robotics
2.2.2 Typical Rehabilitation

Bed-bound lower-limb rehabilitation is one of the first therapeutic activities administered at the stroke ward, and it continues to be used even after patients are mobile. Bed-bound exercises tend to be simple movements, including knee flexion/extension, leg lifts, hip adduction/abduction, and ankle rolls. All exercises involve the therapist manually manipulating the patient’s leg, providing assistance to whatever extent the therapist deems necessary. If the patients are advanced, the therapist may choose to resist motion instead of assisting.

Parameters from typical rehabilitation will help determine the requirements of the new robotic system. Therapists estimated that they exert a maximum of 10 - 15 lbs of force to the leg. The range of motion of patients varied, but many could go through the full range of the leg. The exercises were done slowly at a consistent speed. The role of the therapist can be summarized as providing support to leg and providing assistance through the motion.

Some patients are capable of performing bed-bound exercises independent of the therapist. Performing independent practice can increase total rehabilitation dosage and is recommended by stroke guidelines [2]. However, some patients who are either at an earlier stage of recovery or have suffered more severe strokes are incapable of performing bed-bound exercises without assistance, and so their total dosage is limited to time spent with the therapist.

2.2.3 Patient Engagement

It is important for the patient to be actively engaged in the exercise. Therapists will try to only provide assistance as needed, by waiting for the patient to initiate the movement. Many of the patients required verbal or physical engagement throughout the process. This is in part due to a phenomenon known as neglect, wherein the stroke victim has trouble focusing on the effected side of the body. Therapists will verbally motivate patients in a manner similar to a personal trainer; otherwise they may engage the patient in conversation. Physical or tactile cues can be used to guide the patient, for example by tapping the leg if it needs adjusting. Sometimes, pictures and posters (e.g. of cats and dogs) were posted in front of the exercise machines in the therapy room. Another common practice is to switch
to a new exercise if the patient is showing signs of losing focus.

### 2.2.4 Hospital Environment

Space in the hospital is limited. Many rooms house four patients, and are often crowded by hospital staff, family, and equipment. The beds themselves vary in model and size, but all have some common characteristics: movable guards on the side, removable baseboard, and a tiltable frame controlled from a panel. There appear to only be outlets behind the beds near the floor, but this will again vary from room to room. There is also a therapy room where more advanced patients go to practice sit-to-stand, walking, stair-climbing, etc. This room is more spacious and also includes beds (albeit simpler beds that cannot be tilted and that do no have guards).

### 2.2.5 Therapy Schedule

Therapists generally have a high workload and so must carefully schedule time with patients. At this particular hospital, patients received on average 30 minutes every other day with a physiotherapist for lower-limb rehabilitation. Most patients were fatigued by the end of the session, and so increasing the duration beyond 30 minutes is not feasible. However, the therapists recognized that the patients could benefit from more sessions throughout the week. Some activities that are physically burdensome need two therapist, or more often a therapist and an assistant. These activities are even more difficult to schedule.

### 2.2.6 Other Comments and Concerns

Most therapists had concerns over safety. They recognized that the robot could potentially apply forces which could harm the patient. One point of concern was rotation of the leg out of position such that the force was applied incorrectly. For example, during the knee flexion/extension exercise, if the leg were to rotate out of the sagittal plane, the robot could apply a torque about the hip which would bend the leg further and potentially injure the
patient. Therapists suggested having multiple support points along the leg to keep the leg in position.

Other comments included the need for comfort, particularly using comfortable material where the robot and human interact. They also mentioned the use of tactile methods of engaging the patient (e.g. tapping or vibrations applied to the leg). Some indicated that combining upper and lower-limb activities could be beneficial (e.g. by using a joystick with the robot).

### 2.3 Design Requirements

The following design requirements were collected for the lower-limb rehabilitation device, based on the literature review, fieldwork, and previous work on ViGRR. Any anthropometric data is derived from [52], ranging from a 5th percentile female to a 95th percentile male. After meeting these requirements, the robot should be ready for clinical trials at the hospital with acute stroke patients. The final outcomes of this work and how they relate to these requirements is given in Sec. 6.1.

1. **Support the user’s leg**
   1.1. Hold the weight of the leg, which can be up to 102kg.
   1.2. Keeps the leg from bending out of plane
   1.3. Adjustable to accommodate range of sizes and weights.
   1.4. Can accommodate a large range of motion – leg lengths vary between 80.3 cm and 98.8 cm.

2. **Help acute patients who may not be able to do exercise**
   2.1. Provide assistive forces through the exercise, up to 10-15 lb.
   2.2. Adjust difficulty based on patient

3. **Engage Patient**
3.1. Only provides assistance as needed
3.2. Keeps the patient engaged with visuals and games
3.3. Relates the exercise to ADL’s and real life scenarios

4. **Long Duration & Intensive Sessions**

4.1. Device is comfortable and ergonomic
4.2. Can apply resistance to make exercise more difficult

5. **Hospital Compatible**

5.1. Fits on hospital bed
5.2. Lightweight, can be easily transferred on to bed
5.3. Compliant with sterilization/cleanliness protocols
5.4. Easy to set up

6. **Can be used by Therapists**

6.1. Intuitive UI
6.2. Progress monitoring using patient data

### 2.4 Design Decisions

<table>
<thead>
<tr>
<th>Category</th>
<th>Design Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Motion</td>
<td>Simple Bed-bound Exercise</td>
</tr>
<tr>
<td>Dynamics</td>
<td>Low Speed &amp; Low Force</td>
</tr>
<tr>
<td>Engagement</td>
<td>Visuals, Audio, Tactile</td>
</tr>
<tr>
<td>Structure</td>
<td>Limited space - small form</td>
</tr>
<tr>
<td>Support</td>
<td>Two points of support</td>
</tr>
<tr>
<td>Operational Modes</td>
<td>Assist as needed &amp; Resist</td>
</tr>
</tbody>
</table>
This rehabilitation robot was designed from scratch, from the initial concept to the final realization. A summary of design decisions derived from fieldwork is given in Table 2.1. This sections explains further the early design decisions and iterations, along with the justification. There were four significant design decisions to be made:

1. The number of degrees of freedom (DOF) in the robot. The final decision was to use one linear DOF.

2. The type of linear actuator to use. Options included pneumatics, ball screws, and belt drives; the final design uses a belt drive.

3. The robots placement within the room (on the bed versus beside the bed). The final design sits on the bed.

4. The controller can be run from either a microcontroller or a full computer. Due to the plans to increase complexity in the future, as well as the need for a user interface, a computer was chosen.

2.4.1 Degrees of Freedom

Creating a multi-DOF system which targets many types of acute stroke exercises is the ultimate goal of this project. However, we also wanted feedback from both patients and therapists at the hospital throughout the design. It was decided to build a smaller, simpler robot, which could be taken to the hospital and tested promptly. Feedback could then be used to extend the simpler robot with additional DOF’s in order to achieve more complex movements.

There were three options for a simpler robot configuration: 1 linear DOF, 1 rotary DOF, or 2-DOF. The primary motions we need to target include circular, horizontal linear (knee extensions), vertical linear (leg lifts), and more complex movements like gait trajectories. These also need to be adjustable to accommodate a variety of body sizes. A single linear DOF could target knee extension or leg lifts, whereas a rotary DOF would only be able to
target a weak approximation of a gait trajectory (i.e., a circular motion more similar to pedalling a bike). A 2-DOF system could accomplish more types of motions, for example doing both leg lifts and knee extensions, or more closely approximating a gait trajectory.

Ultimately the single rotary DOF was excluded because its movement was not related to any of the bed-bound exercises. The 2-DOF was excluded because (a) it was determined to be too complex for the scope of this thesis, and (b) it was decided the final configuration of a multi-DOF device would be left to future iterations after receiving feedback from the hospital. Therefore, the final design is a single horizontal linear DOF.

2.4.2 Linear Actuator Type

There are several ways of actuating a linear motion. This includes hydraulics & pneumatics, ballscrews, and belt drives. The pros and cons of each are given in Table 2.2.

**Hydraulics & Pneumatics:** Hydraulics and pneumatics use pressurized liquid or air to achieve linear actuation. They require more consideration than other off-the-shelf solutions. These were discounted primarily because of lack of experience and issues seen in previous projects. Pneumatics can have issues concerning efficiency and cost, and hydraulics can leak fluid which may be hazardous and unacceptable in the hospital.

**Ballscrew:** Ball screws offer high precision and very low backlash. However, the rehabilitation needs to move quickly, and so the ballscrew would require a very large pitch, or the motor would need to be powerful enough to supply high torque at high speed. Cost restraints therefore make using a ballscrew impractical.

**Belt Drive:** Belt drives are in general less expensive and can achieve high speeds. The belt introduces flexibility into the actuator which may affect the precision of the controller. However, precision in this application is not imperative since small errors will not be perceived by the patient. Furthermore, flexibility and belt slip at high torques could actually act as worst-case safety measures. Ultimately an off-the-shelf belt drive was used in the design.
Table 2.2: Benefits of Actuator Types (x = benefit)

<table>
<thead>
<tr>
<th>Type</th>
<th>Cost</th>
<th>Ease of Setup</th>
<th>Backdrivable</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ballscrew</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Belt Drive</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Pneumatic/Hydraulic</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.2: Off-bed robot design

2.4.3 Placement Relative to Bed

The device needs to support the patient’s leg and provide horizontal assistive forces through the leg flexion/extension exercise. Several possible structural configurations were identified. The device could sit beside the bed, with the foot plate extending horizontally to support the patient’s foot (see Fig 2.2). The device could sit on the bed, under the patient’s leg (see Fig 2.3), similar to a passive range of motion machine. Each configurations has pros and cons, as listed in Table 2.3.

The deciding factors were the moment applied by the leg on the footplate, and the space in the room. Two forces are created by the user: the applied force pushing or pulling the footplate ($F_a$), and the weight of the leg on the heel support ($F_w$). The on-bed design sits

Table 2.3: Structural Configuration

<table>
<thead>
<tr>
<th>Pros</th>
<th>On-Bed</th>
<th>Beside-Bed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Takes less space in room</td>
<td>Easy to set up</td>
<td>Requires room beside bed</td>
</tr>
<tr>
<td>Better support of patient’s leg</td>
<td>Easier to extend DOF’s</td>
<td>Large moment on footplate</td>
</tr>
<tr>
<td>Harder to set up</td>
<td></td>
<td>Must move patient to edge of bed</td>
</tr>
<tr>
<td>May obstruct patient</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

30
under the leg, so the moment created by the weight of the leg and its moment arm \((R_z)\) can be minimized by minimizing the \(R_z\) (Fig. 2.4). If the device sits beside the bed, the footplate must extend horizontally outwards, resulting in the weight of the leg creating a moment on the actuator through that distance \((R)\) (Fig. 2.5). This length is harder to minimize, since this would require that the patient is transferred closer to the edge of the bed. Patient transference should be avoided, it is time consuming and usually requires multiple staff members. Additionally, the device would require room beside the bed. After observing several hospital rooms in the stroke ward, it was clear that this space is not always available, due to equipment, furniture, and crowded rooms which can contain up to four beds. The on-bed design sits completely on the bed and so should be compatible with any room. The moment applied to the footplate due to the weight of the leg is smaller as it supports the leg from below. Finally, it can be positioned on the bed in a way that suits the patient, instead of having to move the patient to the edge.

2.4.4 Computer versus Microcontroller

The controller can be executed either from a microcontroller or from a full computer. The microcontroller is cheaper, easier to set up for real-time code execution, and has built in I/O’s. Computers have more processing power, memory, can easily be networked or
connected wirelessly, but require external data acquisition. Additionally, special care must be used to set up a real-time system which can execute control loops with minimized latencies.

For a 1-DOF device, a microcontroller is the typical choice given that the control is simple and does not require much processing power or memory. However, eventually extra DOF’s may be added to our system to the point where a microcontroller is not suitable. The computer could also be used to run the user interface (UI). Therefore we chose to use a computer to run the control software. A Linux operating system with a pre-empted kernel was used to achieve real-time control.
2.5 Final Design

The final design consists of one linear degree of freedom, actuated by a belt drive, which sits on the bed under the patient’s leg. The robot is shown on the test bed in Fig. 2.6. The footplate has heel support and a Velcro strap to keep the foot in place, and is height adjustable to accommodate a variety of leg sizes. The robot with the user’s leg in place and extended is shown in Fig. 2.7. There is also an adjustable end stop to limit the length of the stroke, which can be customized to each individual in order to prevent leg over-extension.
Figure 2.7: The rehabilitation robot with user’s leg
Chapter 3

Device Design and Implementation

3.1 Overview

Figure 3.1 shows a rendering of the design with the frame, footplate, and actuator. The following sections will give an overview of the frame structure, footplate structure, the electronics, and the computer system. Figures of the custom made parts can be found in appendix I, further information on the printed circuit board in appendix II, and hardware data sheets can be found in appendix III.

3.2 Structural Design

3.2.1 Frame

The frame holds the actuator, footplate, and electronics (Fig.3.2). It should fit on a bed, under one of the patient’s leg, without obstructing the other leg or the patient’s body. The frame is constructed out of 1”x1” aluminium profile. The frame can be clamped or strapped to the bed to secure it in place. Future designs could include custom mounts to specific beds by attaching to the footboard or sides. The frame also contains an adjustable physical end-stop which is used to control the maximum stroke length of the actuator. This can be adjusted to prevent over-extension of the leg. Over-extension occurs when the leg is pulled
straight, or beyond its range-of-motion – applying force to the leg at this point may cause harm. The physical end-stop is set to prevent the footplate from moving far enough to over-extend the leg.

3.2.2 Footplate

The footplate attaches to the belt drive and is height adjustable. The footplate both supports the leg and is the interface through which the robot and user interact. It is shown in Fig. 3.3. An aluminium profile extends from the base on the belt drive. An off-the-shelf carriage with a handle lock is used to adjust the height. Mounting plates are used to attach a load cell to the carriage, which is placed near the bottom of the footplate. The footplate is cut from 1/8” aluminium plate.

The load cell measures force applied horizontally; however it may also register force due
to moments placed on the footplate either from the weight of the leg or from applying forces near the top of the footplate. To minimize these erroneous measures, support brackets were created to transfer vertical forces from the footplate directly to the adjustable base, thereby reducing the effect of the moment. These brackets attach rigidly to the footplate, but not to the adjustable base, thus not interfering with the horizontal force reading at the load cell. To hold the foot in place, a custom heel support was created. It contains housing for an infra-red sensor which can detect whether the foot is in place for safety. The heel support was also 3D printed, using ABS material for additional strength.
3.3 Hardware

3.3.1 Actuator

The system is actuated using a Brushed DC Motor, gear reduction, coupling, and an off-the-shelf linear belt drive (Fig. 3.3). The Brushed DC Motor is a Maxon RE 50 200W Graphite Brush model (part number 370354), and the gear reduction is the Maxon GP 52C Planetary Gearhead (part number 223090); more information is provided in the next section. The coupling is an Aluminium Flexible Shaft Coupling with a polyurethane spider from...
McMaster-Carr, which allows for slight shaft misalignment and also has no backlash. The linear belt drive is a LB250-24 Belt Drive from Anaheim Automation. It has a total stroke length of 610mm, which is reduced to 405mm in the final design due to the addition of a second carriage. The belt drive is rated well above the requirements for this rehabilitation robot – it has a maximum speed of 2540mm/sec and a maximum dynamic load of 890N. It is primarily made from aluminium and weighs 40N, making it lightweight and easy to move around. The drive delivers 122mm of linear travel per revolution. The Maxon motor is mounted to the belt drive using custom machined aluminium mounts. Finally, a cable carrier is used to secure wires running from sensors on the footplate to the electronics box mounted at the base of the device.

**Motor Specification**

The motor is required to move the actuator at some desired velocity while also applying adequate levels of force for assistance or resistance. This motor and gear assembly was used
from a previous project, and was found capable of fast linear motion and applied force. Their specifications can be found in Tables 3.1 and 3.2. The maximum linear speed is in this case limited by the maximum input RPM to the gearhead, 6000RPM:

\[ V_{\max} = \frac{p\omega_{\max} \text{1min}}{GR \text{60s}} \]
\[ = \frac{(0.122m)(6000RPM) \text{1min}}{(53) \text{60s}} \]
\[ = 0.230m/s \]

The nominal torque and speed are used to estimate the nominal linear speed and its associated nominal applied force:
\[
V_{nom} = \frac{p\omega_{nom} \, 1\text{min}}{GR \, 60\text{s}} = 0.217 \text{m/s}
\]

\[
F_{nom} = \frac{\tau_{nom} \, GR}{R_{\text{pulley}}} = \frac{(0.405 \text{Nm})(53)}{0.0175} = 1227 \text{N}
\]

The maximum speed of 230 mm/s is adequate for our purposes, as the system is designed for stroke rehabilitation which is a slow moving process. The nominal force of 1227N is much higher than the assistive forces seen during traditional therapy (which therapists estimate to be up to 10-15 lb, or 45-67 N). In the future, a lower gear reduction could be used to increase the maximum speed to accommodate more advanced patients. A lower powered motor could also be used to limit the maximum force for safety purposes.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Linear Speed</td>
<td>230 mm/s</td>
</tr>
<tr>
<td>Nominal Linear Speed</td>
<td>218 mm/s</td>
</tr>
<tr>
<td>Nominal Force</td>
<td>1227N</td>
</tr>
</tbody>
</table>

### 3.3.2 Sensors

A total of five sensors are used on the device, including: a load cell, two limit switches, a quadrature encoder, and an infrared (IR) proximity sensor.

**Load Cell:** An MLP-200 Load Cell from Transducer Techniques is used to measure force applied to the device by the user. The load cell can measure tension and compression force up to 200 lbs. It is used with the TMO-1 sensor amplifier module. The load cell is attached
behind the footplate using two mounting plates.

**Encoder:** The HEDL 5540 encoder came packaged with the motor and gearhead from Maxon. It has three channels and has 500 encoder counts per revolution. The count is recorded by the DAQ using a standard quadrature encoder counter. This gives the relative position of the motor – which, after homing the device to a known position, yields the absolute position of the footplate within the stroke.

**Limit Switches:** Two snap-acting limit switches are used to detect when the carriage has reached either end of the belt drive. The switches are mounted to the belt drive and to the end stop using 3D printed mounts, and a triggering arm is attached to the carriage to engage the switch.

**Infrared Sensor:** This proximity sensor is the GP2Y0A51SK0F IR sensor from SHARP. It can measure distances between 2 and 15 cm. The sensor is embedded in the heel rest on the footplate, and is used to detect whether the user’s foot is in place. As of now, the IR sensor is not integrated into the code: it will be needed before using the device in the hospital.

### 3.4 Electronics and Power

Custom electronics were needed in order to power and operate the sensors, and send command signals to the motor. At first this was achieved using a protoboard (see Fig. 3.8). Wiring and space restrictions became a problem and motivated a redesign using a printed circuit board (PCB), along with a 3D printed case to enclose the electronics. The PCB has four functions: supply power to the sensors, route signals to and from the DAQ, process or filter signals as necessary, and implement basic safety precautions.

- **Power Distribution:** The entire device should run off of a single power supply so that only one wall outlet needs to be used. A 48V supply was chosen, since voltage is required by the motor. The sensors require 24V (Load Cell), and 5V (the IR sensor), and 10V is required to send commands to the motor controller. Therefore the PCB
Figure 3.5: The Maxon motor setup with motor driver, E-stop, and the wiring has three step-down voltage regulators at these three levels.

- **Signal Routing:** To streamline the wiring, the PCB takes in three main wires: a DB9 serial connection from the sensors on the footplate, a DB9 serial connection from the power supply and motor controller, and a DB37 serial connection from the DAQ. These cables are connected to the PCB case and routed to the correct locations.

- **Signal Processing:** The PCB handles analog signal processing (Fig. 3.6). An adjustable first order low-pass filter is used to clean noise from the force measurements. The potentiometer is used to change the cutoff frequency as necessary, we used the filter with potentiometer at about 25K, resulting in a cutoff frequency of approximately 21Hz. The motor commands from the DAQ are also processed through a non-inverting amplifier with a gain of 4. This is used to to map the 0-5V analog out from the DAQ to the motor controller. The new 0-20V motor command is referenced to the 10V signal from a voltage regulator, achieving the final -10 to 10V range required by the motor.
controller.

![Figure 3.6: The force signal low pass filter with adjustable cut-off frequency (left), and the command non-inverting amplifier with a gain of 4 (right)](image)

- **Safety:** The PCB implements four safety checks (Fig. 3.7). First, the motor commands is checked to ensure it is within an adjustable range. This range can be changed using potentiometers. A Watchdog timer is also used to ensure the software has not stalled. Every timestep, the software sends a pulse to the timer – if a pulse is missed within a certain delay, the Watchdog timer is triggered. There is also an emergency stop button which can be pressed at any time by operator or user. Th E-stop signal, Watchdog timer and motor command limits are combined in a logical AND gate, and the resulting output is sent to the enable pin on the motor controller. Any trigger from the three safety signals will disable the motor and stop the device.

### 3.4.1 PCB Layout

The layout was designed in Autodesk Eagle (Fig. 3.9), and the board was manufactured by JLCPCB ([https://jlcpcb.com/](https://jlcpcb.com/)). The first benefit of this design is the size, which contains more components in a smaller space and so can easily be mounted onto the robot. The wiring is also cleaner, with header pins arranged to bundle related signals together (*e.g.* all wires going to the DAQ are together on the same unit of pins). Decoupling capacitors are
Figure 3.7: The safety system controlling motor enable pin

added near all the power pins on the major components, to prevent power spikes or sudden changes in voltage.

3.5 Data Acquisition

Data acquisition is handled by a LabJack T7 DAQ. This model has 14 analog inputs, 2 analog outputs, 23 digital I/O’s, and is also capable of handling quadrature encoder signals and high-speed data streaming (up to 100k samples/s). The device can interface with the computer through an Ethernet, USB, or wifi (Ethernet was chosen for this project as it is the fastest and most reliable). The front-end Modbus TCP interface is conveniently packaged in the C/C++ LJM library, discussed further in chapter 4. The T7’s specifications are listed in Table 3.4.

The analog output of the DAQ is limited to a range of 0-5V, whereas the motor controller accepts commands from -10 to +10V. The DAQ commands are amplified by a gain of 4 using a non-inverting amplifier implemented using an Op-Amp, resulting in a range of 0-20V. This is the positive command input to the motor controller. The negative command is set to a constant 10V from a voltage regulator, leading to the desired -10V to +10V range.
Figure 3.8: The original protoboard layout

Figure 3.9: The PCB layout, with header pins, integrated circuits, and voltage regulators
Table 3.4: T7 DAQ Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
<td>&gt; 1 sample/ms</td>
</tr>
<tr>
<td>Resolution</td>
<td>1µV</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>Analog In: 16-bit</td>
</tr>
<tr>
<td></td>
<td>Analog Out: 12-bit</td>
</tr>
<tr>
<td>Range</td>
<td>Analog In: ±10V</td>
</tr>
<tr>
<td></td>
<td>Analog Out: 0-5V</td>
</tr>
<tr>
<td>Analog Inputs</td>
<td>x14</td>
</tr>
<tr>
<td>Analog Outputs</td>
<td>x2</td>
</tr>
<tr>
<td>Digital I/O</td>
<td>x23</td>
</tr>
</tbody>
</table>

3.6 Computer System

The control software and UI are run from an Intel NUC mini PC, with specification listed in Table 3.5. The RAM and memory are added separately, and can be upgraded as needed. The PC contains ports for USB, Ethernet, and HDMI. It is compact (11.5 x 3.5 x 11.1 cm) and lightweight (450g). The computer’s OS is Ubuntu 16, using a pre-empted linux kernel for real-time control (described in Sec.4.5). In theory, the software should run on any hardware, as long as it has enough RAM and memory, is running Linux with a pre-empted kernel, and has an Ethernet port to connect with the DAQ. This modularity should make upgrading the robot easier, in case a new computer with different capabilities is required.

Table 3.5: Computer Specifications

<table>
<thead>
<tr>
<th>Processor</th>
<th>2.2 GHz 7th Gen Intel Core i5</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td>8GB DDR4</td>
</tr>
<tr>
<td>Memory</td>
<td>250GB Solid State</td>
</tr>
</tbody>
</table>

A GL.iNet GL-MT300N-V2 wireless router is used to set up a local network. The UI is served over this network, and can be accessed by anyone who knows the network passwords and the IP address of the UI. It can be accessed using any web browser. The router is also small and lightweight. The complete computer system, including the router, DAQ, and UI device, is shown in Fig.3.11.
Figure 3.10: The electronics case with the custom PCB and the load cell amplifier
Figure 3.11: The Computer System, with DAQ, router, and tablet
Chapter 4

Controller and Software

4.1 Overview

The controller and software must allow the device to perform the required tasks as laid out in Chapter 3. This includes providing assistance to the patient through the exercise, but not to the extent that the patient can become passive and allow the robot to drive the motion. The level of assistance should be adjustable so that the operator can tailor the session to the capabilities of the patient. The software must also engage the patient using virtual environments and games. These must be related to the motion of the exercise, such that the robot becomes a control input into the game, and the patient is rewarded for following the desired trajectory. Immersion within the virtual environment could also be bolstered by haptic feedback, or the physical rendering of virtual forces to the patient through the robotic device.

4.2 Impedance & Admittance Controller

Both position and force control may behave unpredictably when interacting with an unknown environment, and may not be well suited for the purposes of assisting patients through rehabilitation movements. Position control would not provide assistance as needed, but
would simply force the user’s leg through the motion, similar to a passive motion machine. Force control is not sufficient since it does not allow the controller to follow a predefined trajectory. Interaction control, also known as impedance/admittance control, was introduced in [41]. Instead of controlling position or force directly, interaction control seeks to control the dynamic relation between the position and force variables.

There are two possible causalities – an impedance, which accepts a velocity/position and renders a force, and an admittance, which accepts force and renders a velocity/position. An interaction controller uses one of these causalities, and will sense input from the physical device, calculate the output variables based on some virtual desired system, and then command the device to that output. A commonly-used virtual system is the mass-spring-damper system shown in Fig. 4.1 with mass (m), damping (b), and spring stiffness (k), applied force (F), and the desired position ($x_{des}$).

\[ F = m\ddot{x} + b\dot{x} + k(x - x_{des}). \] (4.2)

Using this control law, the end effector of the robot will behave like the system above, about point $x_{des}$, when displaced by the external environment. If, for example, an operator
moves the robot by pushing on the end-effector, it would respond as if it were the mass-spring-damper system, \textit{i.e.} it would apply a resistive force related to the acceleration, velocity, and the error from the desired point, according to (4.2).

This control law rests on a significant assumption – that the robot itself has no inherent impedance between the actuator and the end-effector. It is possible to cancel some of the impedance of the robot by modelling and compensating for them (such as friction, for example), however this is less common and generally takes more time and is less successful \cite{53}. This assumption is an idealization - there will always be some level of friction, mass, or other impeding factors in the motor, gears, links, \textit{etc}. Therefore, impedance control should only be considered for robots made of low impedance components, such as belt drives and direct-drives.

Admittance control is essentially the inverse of impedance control, in that it accepts an external force and then renders a corresponding displacement. Fundamentally, admittance control can be described by the following equation, where $F_R$ is the force applied to the robot, $Y$ is the desired admittance, and $V$ is the “flow” variable vectors (e.g. position and velocity):

$$V = YF_R$$

(4.3)

Once again, it is common to define admittance in terms of a mass-spring-damper system (4.5). Here, $x_a$ and $v_a$ are the “simulated” position and velocity determined by the admittance controller. Additionally, to actually render the desired position and velocity, an inner-loop PD position controller is used (4.6), with $x_a$ and $v_a$ acting as the desired position and velocity, and with a proportional gain $P$ and a derivative gain $D$. The entire control loop is shown in Fig.4.2. Here, the controller is given in state space form, with state matrix $A$ and input matrix $B$.

$$\dot{X} = AX + BF$$

(4.4)
\[
\begin{bmatrix}
\dot{x}_a \\
\dot{v}_a
\end{bmatrix} = \begin{bmatrix}
0 & 1 \\
\frac{k}{m} & \frac{b}{m}
\end{bmatrix} \begin{bmatrix}
x_a \\
v_a
\end{bmatrix} + \begin{bmatrix}
0 \\
\frac{1}{m}
\end{bmatrix} F_R
\] (4.5)

\[F = P(x - x_a) + D(v - v_a)\] (4.6)

Admittance control works best when the inner position control loop is accurate, and therefore relies on the robot having a fairly high impedance i.e. when using large gear reductions and rigid links.

Figure 4.2: Admittance control flow chart

Another point of consideration when choosing between impedance and admittance control is its stability when encountering different types of environments. Impedance control works better when interacting with an admittance (rigid environments), whereas admittance control works better when interacting with an impedance (weak environment / free space). Admittance control also requires a force/torque sensor, whereas impedance only requires position measurements. A summary of the differences between impedance and admittance control is given in Table 4.1.

This project targets acute stroke patients who typically do not have much leg strength
Table 4.1: Comparing Impedance and Admittance Control

<table>
<thead>
<tr>
<th></th>
<th>Impedance</th>
<th>Admittance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Causality</td>
<td>Position → Force</td>
<td>Force → Position</td>
</tr>
<tr>
<td>Environment</td>
<td>Rigid</td>
<td>Free</td>
</tr>
<tr>
<td>Device</td>
<td>Low impedance</td>
<td>High impedance</td>
</tr>
<tr>
<td>Sensors</td>
<td>Position</td>
<td>Force and Position</td>
</tr>
</tbody>
</table>

and rely heavily on assistance from the therapist during therapy. Therefore, these patients can be modeled as an impedance. Furthermore, the motor used on the robot has a high gear reduction which contains substantial impedance. With impedance control, this impedance would have to be modelled and compensated for. Therefore, it was decided that admittance controller suited the needs of the device.

Admittance control is implemented through the following steps:

1. Set desired admittance parameters (typically K, B and M)
2. Discretize the admittance system \((4.5)\)
3. Run system each time step, using the new force data to update the simulated position and velocity
4. Use a PD position control to achieve the simulated position

The linear admittance control system in \((4.7)\) is discretized according to the \((4.8)\) and \((4.9)\), equations which can be found in a number of books on linear systems, for example in [54], which yield the discrete state and input matrices \((A_d, B_d)\).

\[
\dot{X} = AX + BF \tag{4.7}
\]

\[
A_d = e^{At} \tag{4.8}
\]

\[
B_d = A^{-1}(A_d - I)B \tag{4.9}
\]
4.2.1 Velocity Profile

Figure 4.3: The x-axis shown over the robot

The final step is to set the desired position to a suitable trajectory. The system is 1-DOF, so this can be done simply by creating a velocity profile. The velocity profile is defined in terms of the total length of the linear stroke ($x_{max}$) as well as the maximum velocity ($v_{max}$). The definition of the x-axis is shown in Fig.4.3. At first, a quadratic relation was used, creating a parabola with zero velocity at each endpoint and $v_{max}$ in the center. This was found to feel unnatural. The final profile is shown in Fig.4.4 with constant acceleration to $v_{max}$, a plateau of constant velocity, followed by a constant deceleration. The same profile is inverted to move back at the end of the stroke. The transition points were set at points which were determined empirically based on what felt most natural, with acceleration up to $0.4x_{max}$, and deceleration beginning at $0.6x_{max}$. Improvements could be made on the desired trajectory. For example, a higher order function could be used which minimizes jerk or acceleration to make the movement smoother. Another option is to record the trajectory as either the operator or therapist guides the robot through the exercise.
4.2.2 Operational Modes Using Admittance Control

Two operational modes use the admittance control with the mass-spring-damper system – trajectory tracking with assistance, and trajectory tracking with resistance.

**Trajectory Tracking with Assistance**: The mass-spring-damper system is used with a moving spring origin set to the trajectory as defined by the velocity profile. The spring stiffness represents the level of assistance, with stiffer springs providing more assistive force. The assistance is low when the error is low, therefore only providing assistance as needed.

**Trajectory Tracking with Resistance**: The spring term is omitted \(i.e. k = 0.0\) so that the user only feels damping and mass. The damper term now represents viscous resistance, with higher damping parameters creating greater resistance. Without the spring term, there are no assistive forces and so the patient must provide all motion on their own. This mode could be used for more capable patient’s who need a challenge, or for strength building.
4.3 Haptic Feedback

Haptic feedback is the physical rendering of forces from virtual environments. Typically an actuated device is used as the haptic interface, which allows the user to control the motion and applies interaction forces when necessary. For example, the Phantom Pentium (Fig. 4.5) is a haptic interface pen which allows users to interact with a 3D computer environment [55].

The haptic controller needs to consider the accuracy of the rendering of the interaction forces, the stability of the system, and safety of the user. The modelling and the physics of the virtual environment must also be considered.

Early haptic algorithms used penetration depth between the device and virtual object to determine the interaction forces. However, this leads to ambiguities regarding what surface of the object the user is interacting with (in other words, what face the interaction force should be normal to).

In [56], the God-Object method is introduced to resolve this issue. The device is given a virtual position, the position the device would be in theory if the haptics were infinitely stiff. Hence the virtual position always remains at the surface of the object, and the direction of the interaction force can be set normal to the surface.

A third option, introduced in [57] and known as the haptic coupling, is a virtual system placed between the controller and the virtual environment. Using standard admittance
controller to render the virtual environment is shown in Fig. 4.6. The virtual environment must have the same causality as the controller, in this case the admittance causality. The operator’s state is defined by velocity ($V_h$) and force ($F_h$), and in this case the virtual environment is the admittance controller with velocity $V_e$ and force $f_e$.

Figure 4.6: Block diagram of an admittance haptic device without the haptic coupling

The haptic coupling is placed between the robot and the virtual environment. Its purpose is twofold: to decouple the device and virtual environment, and to allow for different causalities (for example, in Fig. 4.7, the haptic coupling allows for an impedance environment to be used with a robot with admittance control).

Figure 4.7: Block diagram of an admittance haptic device with the haptic coupling

**Decoupling the Device and Environment:** Defining the desired impedance or admittance in terms of a mass-spring-damper system is not always convenient. Ideally, one could define an arbitrary impedance or admittance function relating $F$ and $V$, giving complete control of the behaviour of the robot-environment interaction, while also ensuring stability. The haptic coupling decouples the design of the controller and the design of the virtual environment. This allows the virtual environment designer to create the physics of interaction without regards to the stability of the device. For example, you could create a virtual wall which suddenly applies infinite impedance, something that would normally cause instabilit-
ties. Note that the impedance that is ultimately rendered to the user will not be exactly as designed, but will be filtered through the haptic coupling.

**Allowing for Different Causalities:** The haptic coupling also allows the device controller and the virtual environment to have different causalities. For example, admittance control is better suited to our device. However, creating a virtual environment whose interaction dynamics are calculated based on admittance is difficult. Using an impedance environment is much more intuitive – you measure the current position of the user within the virtual environment, and calculate the interaction force.

![Figure 4.8: Virtual haptic coupling for admittance control](image)

The haptic coupling for admittance devices given in [57] is a mass and damper in series, as in Fig. 4.9. The haptic coupling adds impedance to the controller, and so the virtual environment will be rendered with the highest fidelity when the impedance of the haptic coupling is the smallest possible (without becoming unstable). Placing the mass and damper in series achieves this minimal impedance.

However, in our scenario, it is desired that the minimum impedance provided by the haptic coupling is still somewhat high – if the damping is too low, the patient-robot interaction may become unstable. Also, having the mass and damper in series produces a strange dynamic effect wherein the damping force is diminished as the system reaches equilibrium.
Therefore, we rearranged the haptic coupling to a mass and damper in parallel, as shown in Fig. 4.10. In this case, the impedance of the coupling is intuitive, simply a constant mass and damping. This constant impedance is what the user will perceive when the virtual environment is not adding any interaction force, i.e. when moving through free space. Any interaction forces (e.g. when contacting a wall in the virtual environment) will be added on top of this baseline level of impedance.

\[
Z(s) = \frac{F}{v} = ms + b. \tag{4.10}
\]
The z-transform \[58\] yield the algorithm used in the real-time controller, in terms of mass, damping and the time period \(T\). The bilinear transformation relates the laplace and \(z\) domains through

\[
s = \frac{2(z - 1)}{T(z + 1)}.
\] (4.11)

Substituting this into the haptic coupling dynamic equation yields

\[
Z(z) = \frac{mb(z - 1)}{(m + bT)z - m}.
\] (4.12)

The \(z\)-transform yields the control step algorithm in (4.13). They are used to update the current environment position and velocity \((x_k, v_k)\) based on the previous state of the environment \((x_{k-1}, v_{k-1}, f_{k-1})\).

\[
v_k = \frac{m}{T} v_{k-1} + \frac{f_{k-1} - f}{m + bT + b}
\] (4.13)

\[
x_k = x_{k-1} + v_k T
\] (4.14)

The underlying virtual physics should have an impedance causality, and so the environmental force is calculated based on this position/velocity pair. This mapping can be an arbitrary function of \(x_k\) and \(v_k\), thanks to the haptic coupling (4.15).

\[
f_k = \mathcal{F}(x_k, v_k)
\] (4.15)

For example, to achieve the standard admittance control from the previous section, the impedance mapping can generate the spring component (with the damping and mass contributed by the haptic coupling), as in (4.16).

\[
f_k = K(x_k - x_{des})
\] (4.16)

With the haptic coupling, more complex environments can be created. For example, contact between the user and a virtual object can be rendered. The interaction force is set...
to zero when there is no contact, allowing for free motion. When in contact, the interaction force can be modelled a number of ways, including with a simple spring. This method is used in the balance game, described in Sec. 4.5.3. The implementation of the haptic environment is described in Sec. 4.3.2.

\[ f_k = \begin{cases} 
  f_k = 0, & \text{no contact} \\
  f_k = k(x_{user} - x_{dist}), & \text{contact}
\end{cases} \]

4.3.1 Operational Modes Using the Haptic Coupling

A single mode is implemented using the haptic coupling – a simple interactive balance game. The player is tasked with maintaining a static position within a small margin of error. Disturbances are launched at the player from both directions, which contact the user and produce a haptic force, as described in the previous section. This mode is meant to demonstrate the haptic capabilities of the system, and does not reflect any traditional rehabilitation exercise. In the future, haptic feedback can be added to operational modes that are more related to rehabilitation.

4.3.2 Physics Engine

When using haptic feedback, the physics of the environment become part of the controller and therefore is constrained to the same real time performance needs. The physics engine is implemented as a part of the controller, and the results are sent to the UI so that the environment can be displayed. The physics involves the movement of two objects, the character and the disturbance, as well as the interaction force between them. There are three cases:

1. Disturbance is out of play: The disturbance has left the environment. A new one is randomly spawned to either the right or left of the environment, and given a fixed
initial velocity.

2. No Contact: There is no contact. The position of each object is updated, with the character’s position set equal to the position of the device, and the disturbances position set according to its constant velocity.

3. Contact: The character’s sphere overlaps the disturbances sphere, and are therefore in contact. The contact force is modelled by a linear spring, with the distance between the objects’ centres times the contact stiffness equal to the interaction force.

4.4 Tuning Control Gains

The controller gains were tuned based on the maximum limits of the device and the maximum desired impedance or force when interacting with the human user. The gains and parameters to be set include:

1. Proportional and Derivative gains (P & D) for the position controller
2. The damping and mass (b & m) for the haptic coupling
3. $I_{max}$ on the motor controller, which sets the current command given to the motor at the maximum motor command of $\pm 10V$

Other controller gains are set based on the desired behaviour of the device – these require bounds.

1. K,B,M for the admittance controller
2. $V_{max}$ for trajectory generation
3. $F_{max}$ for the maximum force generated by the system

First, the PD controller can be tuned separately from the admittance. The PD controller can be thought of as another impedance existing within the system (where the proportional
gain is like a spring and the derivative gain is like a damper), and should therefore be
tuned as high (stiff) as possible for optimal performance of the admittance controller. The
P gain should be set such that the largest position error corresponds with the maximum
motor command. The maximum motor command at the motor controller is \( \pm 10V \), however
the motor command sent from the DAQ is multiplied by a factor of approximately 4 and
referenced to a constant 10V. Therefore the maximum command sent from the controller is
between 0 (maximum current in one direction) to 5 (maximum current in the other direction).

The maximum displacement can be calculated by using the maximum velocity achievable
by the actuator. In this case that is 9in/s, which corresponds to a maximum displacement
of about 0.23mm/step for a controller running at 1000 Hz. However, the system may exceed
this limit momentarily. Furthermore, since the system is interacting with a person whose
behaviour is unknown, there is always the chance the system will be driven beyond the limits.
Therefore it was decided to design the PD controller based on a maximum step of 2mm, and
to then try and enforce the maximum velocity of 9in/s through visual feedback to the user,
by encouraging the user to follow the desired velocity. The step size of 2mm was derived
empirically. Given these parameters, the P gain should be set to 2.4. Here the derivative
term is neglected since it’s contribution is negative (damping).

\[
\text{cmd}_{\text{max}} = P \Delta x \\
P = \frac{\text{cmd}_{\text{max}}}{\Delta x} = \frac{4.8}{2} = 2.4
\]

Next, the D gain is determined empirically by increasing it until there is no overshoot to
a step response. The final result was a D gain of 0.2.

The limits on the admittance parameters can now be determined. The admittance system
can be approximated by the following equation,

\[
\Delta v = -\frac{K}{M} x - \frac{B}{M} \dot{x} + \frac{1}{M} F 
\]

(4.18)
To determine the admittance parameters, the maximums on positional error, velocity, and force are needed. The maximum error is here set to 400mm, or approximately the total length of the actuator. The maximum velocity was shown to be 0.23 mm/step above, here it is rounded to 0.3mm and then converted to 300mm/s. The maximum force is set to 350N (note that this represents the highest force possible – with the admittance parameters set to max, and with the system being pushed to the high speed at a high deviation from the desired position – in other words, the user would have to push the system to produce this much force).

This maximums are entered into (4.18). Since this represents the the state of the admittance system at the maximum velocity, the acceleration term is neglected (positive acceleration would indicate the velocity is increasing, but this should not be possible since we are at the maximum velocity, and negative acceleration would reduce the total impedance, hence zero acceleration is the worst case scenario according the assumptions). With acceleration set to zero and the mass term factored out:

\[
0 = -K(400mm) - B(300mm/s) + 350N \tag{4.19}
\]

\[
K \leq 0.875 - 0.75B \tag{4.20}
\]

Rearranging the equation yields a line relating the stiffness (K) and the damping (B). This is the maximum line, any pair of parameters above it may produce behaviours violating the prescribed maximums. The minimum stiffness is zero (negative stiffness is not useful and may cause instability about the desired point). The maximum damping is achieved when there is no stiffness, in this case 1.17 Ns/mm. The minimum damping is determined empirically by lowering the damping until it starts to cause instability, either due to force sensor noise or because it is difficult for the user to control. A minimum of 0.1 Ns/mm was found. At this minimum, the maximum stiffness is then found based on the equation to be 0.8 N/mm.

The haptic coupling parameters are set to the minimums found for the admittance con-
controller. The haptic coupling is then as transparent as possible without leading to instability. The parameter ranges are summarized in Table 4.2.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>P</td>
<td>-</td>
<td>-</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>-</td>
<td>-</td>
<td>0.2</td>
</tr>
<tr>
<td>Coupling</td>
<td>b</td>
<td>-</td>
<td>-</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>m</td>
<td>-</td>
<td>-</td>
<td>0.05</td>
</tr>
<tr>
<td>Admittance</td>
<td>K</td>
<td>0.0</td>
<td>0.8</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>0.1</td>
<td>1.17</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>-</td>
<td>-</td>
<td>0.05</td>
</tr>
</tbody>
</table>

4.5 Software

The controller needs to be implemented in software, along with communication, data logging, and a user interface for setting up and displaying visuals. The software is split into two parts: the control software and the user interface software. The entire codebase is available on GitHub at [https://github.com/nickberezny/RehabRobot](https://github.com/nickberezny/RehabRobot). It is built to run on real time enabled Linux, and should run on any hardware (assuming it has sufficient memory and RAM, and an ethernet port to connect with the DAQ). This modularity should make it easy to use this code in future iterations of the project which may use different hardware. To date, it has been tested on the Intel NUC used in this project, as well as on a Raspberry Pi 3B. The three main branches of the software are the real time controller, the user interface server, and the user interface (Fig. 4.11).

4.5.1 Real Time Operation and Linux

A real time system is one in which the timing of the computations significantly affects the results [59] – in other words, a missed computational deadline can be detrimental to the
performance of the system. Systems can either be soft real time or hard real time. Hard real-time systems are deterministic, whereas soft real-time systems typically use similar methods to hard real-time with less stringent requirements. Choosing which to use is based on the danger associated with missing a deadline. Hard real time system will cause serious harm or damage if a deadline is missed (e.g. the control systems on a Jet), while the consequences are typically minor in soft real time systems.

The consequences of a missed deadline in our system is at worst sending the wrong motor torque command to the actuator, which in the absolute worst case scenario will cause the maximum torque command to be issued. However, motor torque commands are limited for safety, and the operator and user have access to an emergency stop. Therefore, we deemed that the system requires only soft real time capabilities.

Linux is an operating system (OS) based on UNIX which is open source and comes in variety of versions or flavours, including Ubuntu. Linux can be used as a standard home computing operating system like Windows or Mac, and is also found commonly in embedded system where the OS is highly tailored to a specific set of hardware. The Linux kernel is not inherently real time – there is a patch available which adds soft real time capabilities to the OS. This is known as Real-time Linux or RTLinux, and is created using the Preempt RT patch [60]. The patch allows programmers to use the following real time functionalities, as defined in the real time standard extension of the original Portable Operating System Interface for Computing Environments (POSIX) [61].

- **Threads**: POSIX threads, or pthreads, are used for parallel computing. Multiple
threads can run under a single process and share its memory space.

- **Priority Scheduling**: Allows setting thread priorities, with higher priority processes pre-empting lower priority processes. This helps critical control code to run uninterrupted by less important work.

- **Mutexes**: A Mutual Exclusions Object is used to orchestrate data sharing between threads. Once one thread takes ownership of a Mutex, others cannot access it until it is unlocked. This prevent other threads from changing the data before the previous thread is finished.

- **Memory Locking**: Locks memory into RAM to prevent the program from accessing the hard disk (Page Faults), which can cause delays.

### 4.5.2 Controller

The controller is written in C, and is designed to achieve low latency so that it can meet sampling rate deadlines. The target frequency is 1kHz, or 1ms per time step. The controller on average runs at 1.05 ms per time step, or approximately 0.95 kHz. More information of delays and jitter in the control software can be found in Sec 5.4.3.

The controller process is split into three threads: the controller, the server communication, and data logging. Thread priorities can be set from 1 - 98 on Linux, with higher priority threads being able to pre-empt lower priority threads. In our case, the controller, server, and data logging threads have priority levels of 98, 97, and 96, respectively. The controller has highest priority and can pre-empt the others to keep it running up to the desired frequency.

Data sharing between the three threads is handled by mutexes and a 10 dimension array of data structures. Each thread iterates through the data structure array. The control thread, being the highest priority, locks the first element. The other threads cannot lock the mutex and so have to wait. The control loop runs through one step, setting any data from the DAQ or otherwise to the data structure. Once the time step is finished, it unlocks the first element and locks the second, and continues. The communication thread can now lock
the first element, and send the required data to the UI, such as the position of the device. Once complete, it unlocks the first element and waits for the next to be unlocked by the controller. Finally, the data logging thread can lock the first element, and print the required variables to another row in the data log file. This process then continues through the ten elements, with the threads going back to the start once finished with the tenth element, and etc.

Setup

Once code execution is started, there is a number of setup and initialization steps before jumping into the control loop.

1. Global variables are created.

2. The TCP socket is initialized for communication with the UI

3. Connection with the DAQ is established. The I/O’s are setup as necessary (e.g. two digital pins are set as quadrature counters for reading the motor encoder)

4. The data log text file is created, and the header row is printed.

5. A total of 10 mutexes are created to regulate data access to the control data.

6. Memory is locked - no new variables should be allocated after this point

7. The Server code is executed automatically in another terminal so that the user can connect to the UI.

8. The controller waits for input from the UI. Once received, the message is parsed and the control parameters are set.

9. Admittance control matrices $A_d$ and $B_d$ are created according to the discretization formulas.
10. Once the UI sends the home command, the homing procedure is executed, with the carriage moving from front to back, to ensure the device starts from a consistent position, and to measure the total stroke length as set by the adjustable endstop.

11. Once the UI sends the run command, the force sensor is calibrated by taking 10 measurements, checking for outliers, and then setting the force bias to the average.

12. The initialization is now complete, and the three threads are run.

**Control Loop Thread**

The control loop can be broken down into four steps, which are executed repeatedly until the program is stopped either by a set end time or by a stop command from the UI.

1. The current robot state is determined by reading sensor values from the DAQ.

2. The selected control algorithm is run to determine the command (either admittance control or haptics with physics).

3. The command is sent to the DAQ to be realized by the motor.

4. The total change in time since the beginning of the step is determined, and the controller sleeps for any remaining time so that the step lasts approximately 1ms.

**Server Communication Thread**

This thread sends data to the UI so that the games can be updated. To prevent overloading the UI, the data is only sent every 10 time steps (every 10ms). This data is sent through the TCP socket.

**Data Logging Thread**

This thread logs relevant data to a text file. It does this by printing a predefined set of variables at every time step. The text file can then be analysed, using for example MATLAB or Python. Metrics like error and effort can be calculated, and data can be graphed.
4.5.3 User Interface

The User Interface (UI) is used both by the operator and the patient. It can be used to set up the parameters for a session, display the visual feedback to the patient, and view the session data and past session data for progress monitoring. The UI should be intuitive, easy to access, and portable across different hardware and computers (including tablets, mobile devices, etc.).

There are a number of libraries for creating UI’s. For example, GTK is a toolkit for creating UI’s on Linux, Mac or Windows [62]. However, we want the UI to work on mobile devices as well (iPADs and Android Tablets), which are more portable and may be easier to position so that the patient can access it comfortably. Ideally, a library that works on Linux, Max, Windows, Android, and iOS would be chosen. One option is to use web technology, given that all of these OS’s have internet browsers. We therefore decided to create and serve a website from the computer through a local network, that could be accessed by any device which has access to the same local network.

The UI uses the following libraries/tools:

- **Node.js**: A Javascript runtime used to build JS based applications
- **React**: A component-based JavaScript UI framework
- **Redux**: State management for React
- **Express**: Node.js framework, used to create the UI’s HTTP server
- **Next**: Server-side rendering for faster page loading
- **Material-UI**: A css framework based on Google’s Material Design
- **Socket.io**: Library for creating websocket used in communication with host computer
- **Three.js**: A javascript 3D graphic library
Ultimately, the UI is built on React, which is a javascript library for creating graphical user interfaces on the web. Components like buttons or inputs can be customized and re-used throughout the app. This allows for easy customization for future project iteration that will likely need new features.

**Layout & Pages**

The UI consists of a menu and three pages: the setup page, the game page, and the settings page. The UI layout includes a top bar with menu access, the menu (which can be hidden or expanded), and the content.

**Setup Page:** Seen in Fig. 4.12, the setup page allows the therapist or operator to set the parameters for a therapy session. The therapist can set the desired game, the length of the session, velocity and force limits, assistance levels, etc. In developer mode, controller gains can be tuned directly. The set button sends this information to the control process.

![Figure 4.12: The setup page used to start a rehabilitation session](image)

**Games Page:** The game page displays the games or visuals to the patient. It contains
the home and run buttons which are used to set up the device before starting the session. The home button triggers the homing procedure, in which the device is moved to the limits of the device, which is used to determine both the maximum stroke length and the device’s current position.

Settings Page: This page contains any UI settings that one may want to change. For now, this only allows the operator to switch between Developer Mode and User Mode, which change the level of detail in the setup page.

Visuals

The games page is used to give feedback to the user on their position within the trajectory. The guide is used to display the length of the actuator, the user’s position and the desired position (Fig 4.13). The desired position is shown as a translucent box, which is normally red but changes to green if the user’s position is within a certain small margin of error. The User’s position is shown as a blue rectangle. In essence, the knee flexion exercise is carried out by the user attempting to follow the desired position as closely as possible. The guide is also useful in showing the user the end limits of the device, to help them avoid forcing the device through the endstop (Fig 4.14).

Figure 4.13: The trajectory guide with large trajectory error (left) and small error (right)
Three simple games were created, each corresponding to a different, fundamental type of game which could be expanded upon by future students. The goal was to test each type of game to determine which (if any) are engaging, or whether they are distracting. The first two games represent environments which react to the user’s performance. The user is still shown the guide, and the amount of trajectory error determines what happens in the environment. The first is a simple, non-competitive game. The second is a competitive racing game which has a clear goal. The third game links the user’s movements directly to the movement of the virtual character, and is meant to demonstrate a more immersive virtual world with haptic feedback.

- **Simple Environment**: a simple cube, displayed with the guide. If the user is within a certain margin of error, the cube spins, otherwise the cube is still. This is meant to demonstrate a virtual environment which responds to the performance of the user, but is not a competitive or challenging game.

- **Racing Game**: a race between three objects: the user and two computer controlled
characters (Fig. 4.15). Each object moves around a track, with the computer characters moving at predefined speeds, and the user’s character moving at a speed proportional to the accuracy of their motion (i.e. the lower the error, the faster they move). This game is meant to demonstrate a more challenging game which motivates the player to perform the exercise accurately.

- **Haptic Balance Game**: this game is not related to the knee flexion exercise. The user’s foot directly controls the position of their character. They are tasked with maintaining a static position on top of a column (Fig. 4.16). Disturbances are launched at the player, which when in contact create haptic force feedback according to the physics engine. The user must push these disturbances away without falling off the column. This game demonstrates both haptic feedback, and a game in which the user’s foot directly controls a character.

![Figure 4.15: The racing game](image)

Figure 4.15: The racing game
Figure 4.16: The haptic balance game
Chapter 5

Experiments

5.1 Overview

Four experiments were run in total: a safety test, a test of the controller, and two phases of a healthy participant experiment. The goals, procedure and outcomes of each are summarized in Table 5.1. These experiments show that the device is functioning as desired and can successfully deliver assistance and resistance to the user through the knee flexion/extension. The experiments also uncovered future directions for design changes and additions, which are discussed in Chapter 6.

5.2 Safety Tests

5.2.1 Goals

The safety features on the device are designed to limit the forces and velocities applied by the robot to the user, and also to prevent over-extending the leg. The safety features to be tested included the software limits and the end stop. The software limits should saturate the motor command (prevent current commands beyond a set threshold), should limit the desired velocity, and should prevent the carriage from moving beyond the end limits. The end stop is a physical barrier which should be able to withstand large forces.
# Table 5.1: Summary of the Experiments

## Experiment 1: Safety Tests

<table>
<thead>
<tr>
<th>Goal</th>
<th>Verify robot safety systems before human testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure</td>
<td>Ensured software limits were functioning by applying large forces/velocities to device</td>
</tr>
<tr>
<td>Outcomes</td>
<td>Safety features were functioning as expected</td>
</tr>
</tbody>
</table>

## Experiment 2: Functional Tests

<table>
<thead>
<tr>
<th>Goal</th>
<th>Verify admittance control and haptic feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure</td>
<td>Applied forces to the device and measured the response. The response was compared against an admittance simulation</td>
</tr>
<tr>
<td>Outcomes</td>
<td>The robot controller functioned within a small margin of error, which was due the time step lag in the PD controller</td>
</tr>
</tbody>
</table>

## Experiment 3: Healthy Subject Testing - Admittance Parameters

<table>
<thead>
<tr>
<th>Goal</th>
<th>Determine acceptable assistance and resistance levels &amp; determine relation between trajectory error, user effort, and parameter levels.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure</td>
<td>Six subjects followed the knee flexion/extension exercise through 10 one-minute sessions at different assistance/resistance levels</td>
</tr>
<tr>
<td>Outcomes</td>
<td>Assistance reduced trajectory error, and resistance increased user effort. Participants preferred different levels of resistance.</td>
</tr>
</tbody>
</table>

## Experiment 4: Healthy Subject Testing - Games

<table>
<thead>
<tr>
<th>Goal</th>
<th>Test three game activities designed to engage participants in the exercise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure</td>
<td>Six subjects played the three games for 2 minutes each, and answered subjective questions</td>
</tr>
<tr>
<td>Outcomes</td>
<td>Participants preferred games directly related to the movement of the device, opposed to games which only responded to performance. The games require improvements to make them more engaging</td>
</tr>
</tbody>
</table>
5.2.2 Method

Software limits were tested by driving the system over the motor command, velocity, and position limits. This was done both programatically (setting the controller to overshoot the thresholds), and by applying large forces to the device (using the arm instead of the leg).

The end stop was tested by setting the device to move the carriage beyond the end stop with a high motor command, thus applying a large force. The carriage was also pushed into the end stop by the user’s leg (while the device was turned off). These should determine if the end stop can withstand large forces applied by either the actuator or the user.

5.2.3 Results & Discussion

All safety features worked as expected. They effectively limit the velocity and force applied to the user by the robot. The user could still push the device beyond these limits, but this would be under their own volition and hence isn’t considered a safety issue. They would also need to push the device beyond the control limits, i.e. they would be fighting the position controller and would also begin to feel the real impedance of the device, making pushing the device beyond the software limits difficult. The end stop effectively prevents over-extension of the leg, under two assumptions: the end stop is adjusted properly and the person and robot do not shift substantially on the bed. Future work should include a mounting system to keep the device fixed to the hospital bed.

Other safety features have been included but have not yet been integrated, including hard limits on the motor command. This is done using a comparator to compare the motor command against a fixed voltage level. This features should be fully tested and integrated before testing on patients, who may be in greater danger due to high applied forces.
5.3 Functional Tests

5.3.1 Goals

The goal of admittance control is to achieve a desired dynamic behaviour, usually in terms of a mass-spring-damper system, given some input force. It is important that the controller correctly renders the admittance according to the set parameters, since the spring and damper determine the level of assistance and resistance applied to the user and so determine the rehabilitation. This experiment sets out to verify that the robot behaves as desired based on the controller parameters, and additionally that the controller is capable of executing at the desired sample rate (1 kHz) without significant jitter (time step variation) or time delay.

5.3.2 Method

To confirm that the controller is successfully rendering the desired admittance, the output position of the device was compared to a simulation of the desired admittance. The first test looked at a position step response, in which an offset to the reference position of 200mm was used. For the second test, force was applied to the system by pushing the footplate, and data regarding the position and the input force was saved. The force data was used as an input to a Simulink simulation of the mass-spring-damper system, and both the robot’s position and the simulated position were compared. It was expected that the robot position would track the simulation with some minimal delay due to time lag introduced by the finite sampling rate, as well as some possible error due to limitations of the robot’s PD control.

Next, the robot was run with a human in place (i.e. operating the device with their leg). The user went through the knee flexion/extension exercise with the visual feedback and games. The time duration of each step was recorded, and were used to determine the average time step interval, and to investigate time step jitter.
5.3.3 Results & Discussion

For the first test no interaction force was applied, only an initial offset of the reference position of the spring (of 0.2m) was used to elicit a response. A comparison of the robot’s behaviour and the simulated behaviour of the desired admittance behaviour is given in (Fig. 5.1). The system behaved as expected, following the correct admittance with a small delay.

The next test was to apply forces to the system to determine if the system can render the correct admittance with a semi-random input. Hence, a human subject was tasked to move the footplate back and forth. The force data from the 1-DOF load cell was saved and used as an input to the admittance simulation. Fig. 5.2 shows a comparison of the actual device position versus the simulated position. Again, the system behaves as expected, following the simulation with some minimal delay. The average positional error was only 1.91mm.
The soft real-time system was capable of providing reasonably precise time step control, with an average time step of 1.05 ms (5% higher than the goal of 1ms). Jitter was higher during some steps, sometimes reaching up to 2ms, however this did not affect the system’s ability to deliver the proper controls (Fig.5.3).

These tests confirm that the admittance controller adequately recreated the desired dynamics, and that the outer-loop PD controller tracked this behaviour.

Figure 5.2: Robot admittance position vs simulated position based on force data
5.4 Healthy Subject Experiments

5.4.1 Goals

The healthy subject experiments were designed as a preliminary study on the interaction between the device and a human operator, specifically the robot’s ability to administer rehabilitation in the form of assistive and resistive forces, visual feedback, and engaging games. While the previous two experiments confirmed that the device was safe and that the controller functioned as expected, there were still questions that needed to be addressed before bringing the robot to the hospital to test on stroke patients. Suitable levels of assistance and resistance are not known – a range of acceptable spring and damper gains needs to be determined. Furthermore, it was not known what parameters levels users would prefer (e.g. high levels of resistance, low levels of resistance, etc.), or to what extent these parameters
would change the behaviour of the user (e.g. does increasing the spring stiffness improve
the user’s ability to follow a desired trajectory, and does it affect how much work the user
has to do?). Other qualitative questions also remained, such as the comfort of the robot,
how well the robot supported the leg, and how safe the user perceives the device to be. It is
critical that any flaws in the design which compromise these aspects of the robot are fixed
before bringing the device to the hospital. Finally, the games need to be tested to determine
if they successfully engage the user in the rehabilitation exercise. Since these games are not
meant to be finished products, but only simple representations of different types of potential
games, the feedback here should be used to influence what types of games are created in
future work.

5.4.2 Method

Six subjects were recruited to participate in a pilot study under ethics approval from Carleton
University’s REB. This was a preliminary study and so was done on healthy subjects with
motor control over their legs. The purpose of the study was threefold: to test the function
and safety of the device, to study the effects of different levels of assistance and resistance,
and to collect subjective data concerning the device and the three types of games. The
experiment consisted of two phases:

Parameter Ranges: Subjects went through ten 1-minute trials with varying levels
of assistance and resistance (i.e. varying levels of the spring and damper parameters in
the admittance control). Five trials included changing levels of assistance with constant
resistance, and the other five had changing levels of resistance with no assistance. The
assistance levels varied from 0.2 to 0.6 N/mm (with a constant damping of 0.2 Ns/mm),
and then the damping varied from 0.2 to 0.6 Ns/mm. The order of the levels was changed
between subjects to prevent time-based biasing due to learning or fatigue. Each trial required
the subject to follow a desired trajectory, moving back and forth through the knee extension
exercise, each at the same velocity.

Games & Visual: The next phase consisted of three 1.5-minute trials covering the three
games described in Sec. 4.5.3. The user played each game and then was asked to comment on their levels of effort, engagement, the difficulty of the task, and the distractibility of the visuals.

Data collection included data logged from the robot and subjective questionnaire answers. Relevant data from the robot included position, desired position, velocity, and force. As has been previously mentioned in the literature review, certain metrics derived from rehabilitation robot data have the potential to be used to monitor patient progress. For example, trajectory error correlates well with some subjective rehabilitation measures. The average trajectory error ($e_{ave}$) can be calculated based on the user’s position at time-step $k$ ($x_k$), the desired position ($x_{k,des}$), over the entire duration of the exercise ($N$ steps)

$$e_{ave} = \frac{1}{N} \sum_{k=0}^{N} |x_k - x_{k,des}|.$$  \hspace{1cm} (5.1)

Another metric that could be of use is what is here termed the user’s effort ($\phi$), or the total work done by the user to the system. This is equal to the sum of instantaneous power over the duration of the exercise. To prevent inter-participant differences from obscuring any trends, each participant’s effort is normalized against their effort on the lowest assistance level

$$\phi = \sum_{k=0}^{N} \frac{v_k F_k}{\phi_{norm}}.$$  \hspace{1cm} (5.2)

User effort could be used to track how well the patient is recovering in terms of strength and motor coordination, or could be used to tune resistance levels to ensure the exercise is not too easy or difficult. However, further research is needed into the utility of effort.

In addition to these metrics, data was collected from the questionnaire in Fig. 5.4 - 5.6. This covers the user’s perceived difficulty of the different levels of assistance/resistance, their experience using the device in terms of comfort, safety, and etc, and their views on the three games.
Subject ID:  
Date:  

**Questionnaire**

1. How difficult was it to follow the trajectory (5 = difficult ... 1 = not difficult) ? Did you feel safe and in control (yes/no)?

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Safety</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assistance Trial #1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>y / n</td>
</tr>
<tr>
<td>Assistance Trial #2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>y / n</td>
</tr>
<tr>
<td>Assistance Trial #3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>y / n</td>
</tr>
<tr>
<td>Assistance Trial #4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>y / n</td>
</tr>
<tr>
<td>Assistance Trial #5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>y / n</td>
</tr>
<tr>
<td>Resistance Trial #1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>y / n</td>
</tr>
<tr>
<td>Resistance Trial #2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>y / n</td>
</tr>
<tr>
<td>Resistance Trial #3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>y / n</td>
</tr>
<tr>
<td>Resistance Trial #4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>y / n</td>
</tr>
<tr>
<td>Resistance Trial #5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>y / n</td>
</tr>
</tbody>
</table>

2. Which assistance level did you prefer (1 = lowest assistance ... 5 = highest assistance)

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
</table>

3. Were any of the resistance levels too high to perform the task? (Mark all that apply)

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
</table>

Figure 5.4: Questionnaire page 1
Subject ID:  

Date:  

**Questionnaire B**

1. How engaged were you during each activity?

<table>
<thead>
<tr>
<th>Activity</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trajectory Tracking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cube Game</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Race Game</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balance Game</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. How difficult was the task?

<table>
<thead>
<tr>
<th>Activity</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trajectory Tracking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cube Game</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Race Game</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balance Game</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. How much did the games/visuals distract from your ability to perform the activity?

<table>
<thead>
<tr>
<th>Activity</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trajectory Tracking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cube Game</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Race Game</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balance Game</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. How much effort did you put into the task?

<table>
<thead>
<tr>
<th>Activity</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trajectory Tracking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cube Game</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Race Game</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balance Game</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.5: Questionnaire page 2
5.

<table>
<thead>
<tr>
<th>Question</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>How comfortable was the device?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How comfortable was using the device on the bed (e.g., while lying down?)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How comfortable was the use/position of the UI?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How easy was it to go through your legs full range of motion?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How safe did you feel while using the device?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How well did the device support your leg?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How in control did you feel while using the device?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How well did the visuals seem to correspond to the movement of the device?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.4.3 Results & Discussion

Admittance Ranges

Table 5.2: Measures across the range of assistance and resistance

<table>
<thead>
<tr>
<th>Mode</th>
<th>Level</th>
<th>Average Error [mm]</th>
<th>Effort [Nmm/s]</th>
<th>Difficulty score/5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assist</td>
<td>1</td>
<td>5.00</td>
<td>5.00</td>
<td>2.75</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4.35</td>
<td>3.15</td>
<td>2.73</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3.55</td>
<td>2.60</td>
<td>1.95</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.52</td>
<td>2.13</td>
<td>2.24</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.40</td>
<td>1.84</td>
<td>2.23</td>
</tr>
<tr>
<td>Resist</td>
<td>1</td>
<td>4.07</td>
<td>2.15</td>
<td>3.50</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>10.40</td>
<td>2.76</td>
<td>2.01</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4.98</td>
<td>5.11</td>
<td>2.61</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6.10</td>
<td>4.44</td>
<td>4.37</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5.61</td>
<td>7.34</td>
<td>5.00</td>
</tr>
</tbody>
</table>

Table 5.2 holds the trajectory error (5.1), user effort (5.2), and subjective difficulty (subjective score out of 5), averaged across all participants, for each level of assistance and resistance.

The effects of changing the level of assistance are shown in Fig.5.7. Increasing assistance levels decreased average error from 9.25mm to 6.32mm at the lowest and highest levels, respectively. This result indicates that the assistance provided by the admittance controller was successful, and can be used to help user’s of the robot go through the knee flexion/extension exercise. An example of the effects of assistance is shown in Fig.5.8, with user positions at high and low assistance compared to the desired trajectory. In this case, higher assistance has helped to prevent overshoot. Assistance levels did not change user effort significantly. The comparison of positions from low and high assistance levels for each participant is given in Appendix IV, Fig.6.20 - 6.25.

Increasing assistance levels also tended to decrease the user’s subjective difficulty score. The participants therefore perceived the increased assistance and found the activity easier to accomplish. Assistance may be useful to help patients who may experience frustration.
Figure 5.7: Mean error from desired position (top) and total effort (middle) and subjective difficulty (bottom) for different assistance levels
from not being able to follow the desired trajectory closely, for example when playing one of the games. One possibility is to develop an adaptive algorithm to change assistance in real time so that the user always remains within some margin of error, to keep them from becoming discouraged.

The effects of changing the level of resistance are shown in Fig. 5.9. Increasing resistance levels increased user effort, and had no significant effect on error. The increase in effort was expected given that higher resistance modes by definition require more force to achieve a given velocity. Resistance levels should be changed based on the user’s strength, and higher levels of resistance could be used to build strength.

The subjective difficulty scores did not change linearly with resistance. Difficulty tended
to be rated higher at the extremes, and was lowest at the middle resistance level. This resulted because some participants preferred lower levels of resistance, reporting that they liked when the robot moved transparently, while other participants preferred higher levels of assistance, reporting that they wanted to be able to push through the movement without overshooting. These results further emphasize the need to tune resistance levels to the individual’s ability and preferences.

These findings suggest that (a) assistance effectively reduces error and helps the user follow the trajectory, (b) resistance should not be minimized, but should be tailored the preferences of the user, and (c) looking at the error and effort metrics yields sensible results, making them candidates to track patient progress in future trials. Future work could include an adaptive controller which updates assistance levels based on the trajectory error.

**Efficacy of Games**

The subjective scores are summarized in Table 5.3, each is averaged over the six subjects, and is given as a score out of five. The engagement scores indicate that the subjects did not enjoy the simple environment, preferring the more challenging racing game and the more interactive balance game. The subjective effort and difficulty did not change significantly across games. Both the simple environment and the racing game were found to be somewhat distracting, with average scores of 3/5.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Simple Env.</th>
<th>Racing Game</th>
<th>Balance Game</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engagement</td>
<td>1.75</td>
<td>4</td>
<td>4.4</td>
</tr>
<tr>
<td>Effort</td>
<td>2</td>
<td>2.3</td>
<td>3.5</td>
</tr>
<tr>
<td>Difficulty</td>
<td>1.75</td>
<td>1.9</td>
<td>2.5</td>
</tr>
<tr>
<td>Distractibility</td>
<td>2.9</td>
<td>3</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Engagement results suggest that interactive and competitive games and visuals are more
Figure 5.9: Mean error from desired position (left) and total effort (right) for different resistance levels
engaging; however, this is based on the opinions of healthy subjects. Future work will also look to quantify engagement instead of relying on subjective scales. The distraction of the games is also an issue, which may be amplified when testing on stroke patients. This could be in part due to the separation of the trajectory guide visual and the game visuals. A way to better integrate the visuals and the trajectory task should be investigated.

General Comments

Participants were asked to score the device’s comfort and safety, the correspondence between the visual guide and the device’s position, and how well the footplate supported their leg. Results are given in Table 5.4. Safety and visual correspondence scored high (4.8 and 4.5/5, respectively), indicating that the controller and the visuals functioned well. The comfort score varied, with some finding the device uncomfortable, particularly the heel rest. The leg support score was consistently low – the mechanical design of the footplate should be improved, and the heel rest made more comfortable. Additionally, a secondary support point should be created to help support the leg and keep the leg from rotating out of plane. This could be a passive linkage which attaches under the user’s calf and prevents lateral motion. This secondary point of support would mimic the therapists, who use one hand at the heel and one hand near the knee when supporting the patient.

<table>
<thead>
<tr>
<th>Subject</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comfort</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3.0</td>
</tr>
<tr>
<td>Safety</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>4.8</td>
</tr>
<tr>
<td>Visual Correspondence</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>4.5</td>
</tr>
<tr>
<td>Leg Support</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1.7</td>
</tr>
</tbody>
</table>

Participants were asked to comment on how they thought the device could be improved. These comments are enumerated here, and will influence the future work section in Sec. 6.2.
1. Some users found the heel rest uncomfortable.

2. The simple visual environment (with the spinning cube) confused some participants.

3. The motor makes a ringing noise when on, which annoyed some participants.

Conclusion

Experiments indicate that the device functions as desired. Findings indicate that assistance can help users follow the desired trajectory more closely, and that resistance should be tailored to the preferences of the user, as some prefer lower or higher levels. The games and haptic interaction increase engagement, but can be distracting. The users felt safe while using the device, although the comfort and the footplate support need to be improved before testing on stroke patients. Future work includes these improvements, more in-depth experiments regarding the engagement of the games, and clinical trials on patients.
Chapter 6

Conclusion & Future Work

Rehabilitation Robots have been used successfully in both research and clinical settings. Many devices exist, primarily for the upper-limbs and for gait training. The gap in the research concerning devices for bed-bound therapy for acute stroke patients motivated the construction of ViGRR, a 4-DOF rehabilitation robot. ViGRR was successfully tested, however its high weight and power makes it unsuitable for the hospital environment. It was decided to construct a new robot which targets acute stroke rehabilitation and is compatible with the hospital. The following work and contributions are included in this project:

Fieldwork at Hospital: Fieldwork was conducted at the Stroke ward at the Civic campus of the Ottawa Hospital. Physio- and Occupational Therapists were shadowed and interviewed in order to learn more about bed-bound stroke rehabilitation. From this information, the device concept was created.

Structural Design: The structural design included preliminary design, creating the CAD models, purchasing the material/hardware, and building the robot. Components included the frame, the actuator, and the footplate.

Electronics: A PCB was created in EAGLE and was manufactured by JCLPCB. It is used to (1) route signals from sensors to the DAQ, (2) process signals, including amplification and filtering, and (3) implement safety features which disable the motor.

Controller: An admittance controller was created, and the acceptable ranges on the
gains were determined. A haptic coupling was added to allow for more complex virtual environments. To demonstrate the haptics, a physics engine was used to create a simple virtual environment containing a moving object, with the contact modelled as a spring.

**Software**: The controller was implemented in C, and runs using real time functionality on Ubuntu with a pre-empted kernel. The code uses multi-threading, mutexes, and scheduling to achieve soft real time, which was verified to run close to the desired 1kHz.

**UI**: A user interface was created using web technologies like React.js and THREE.js. It can be used to set up a therapy session by setting desired parameters and activity; it is also used to display visuals to the patient. Three games were created: a simple environment, a racing game, and a balance game with haptic feedback.

**Pilot Study**: Six subjects were recruited for a study which aimed to look at a range of assistance and resistance levels, and a preliminary test of the games. Results indicate that the robot functions as expected, but more work needs to be done on supporting the leg and making the device comfortable.

**Publications**: This project has resulted in two publications, with one more in the process of submission. (1) A conference abstract and presentation at the Conference of Control, Dynamic Systems, and Robotics (CDSR’19) covering the design and controls, which won the best presentation award [63]; (2) a conference paper and poster presentation at the Canadian Medical and Biological Engineering Conference, which covered the fieldwork and how it inspired the foundational device concept [64]; and (3) a paper which is in the process of submission covering the robot’s design, controller, and the results of the healthy subject experiments. Furthermore, the groundwork has been lain for another experiment and publication in the coming months, which seeks to improve the games, measure engagement quantitatively, and also collect electromyographic (EMG) data.

### 6.1 Design Requirement Outcomes

A list of requirements for the device was developed in Sec. 2.3. The work presented here fulfilled many of the requirements, however some require additional work before the device
is ready for the hospital. The final outcomes of the work as they relate to the requirements are listed below.

1. Support Leg

1.1. Hold the weight of the leg: Works in part, although participant feedback indicates more work needs to be done to secure the foot.

1.2. Keeps the leg from bending out of plane: Requires the addition of another point of support

1.3. Adjustable to accommodate range of sizes and weights: Footplate height is adjustable, and total stroke length is as well.

2. Help acute patients who may not be able to do exercise
2.1. Provide assistive forces through the exercise: experiments show assistance from admittance control lowered trajectory error.

2.2. Adjust difficulty based on patient: Experiments show that increasing resistance increases the required user effort.

3. Engage Patient

3.1. Only provides assistance as needed: In theory, assistance from the admittance spring should only apply large forces when error is large. Future work could include adaptive control to change assistance.

3.2. Keeps the patient engaged with visuals and games: Preliminary games showed promise in keeping the user engaged in an otherwise boring activity.

3.3. Relates the exercise to ADL’s and real life scenarios: Further work is required in making the games more relevant to everyday experiences.

4. Long Duration & Intensive

4.1. Device is comfortable and ergonomic: Participants used device for 20+ minutes without issue. Subjective questions indicate the footplate and heel rest could be more comfortable.

4.2. Can apply resistance to make exercise more difficult: Experiments show that increasing resistance increased the required user effort.

5. Hospital Compatible

5.1. Fits on hospital bed: The device has a small footprint and will fit on a standard single bed.

5.2. Lightweight, can be easily transferred on to bed: The device can be carried around by a single (healthy) person. Future work could include a cart or wheeled device for transport.
5.3. Compliant with sterilization/cleanliness protocols: A shell will need to be added before taking the device to the hospital.

5.4. Easy to set up: The set up only requires a few steps, including adjusting the footplate height, end stop distance, and homing the device.

6. Can be used by Therapists

6.1. Intuitive UI: The user interface worked well during experiments; therapists can set up sessions without knowledge of admittance control gains or other technical background.

6.2. Progress monitoring using patient data: two metrics, trajectory error and user effort, show promise for tracking the patients ability and strength.

### 6.2 Future Work

#### 6.2.1 Design Additions

Participants rated the device’s ability to support the leg consistently low, at an average of 1.7/5. This needs to be improved before testing on patient’s, who will be more dependent on the device for support.

1. Improve heel rest comfort by adding comfortable material, and by extending the rest to hold more of the foot.

2. Improving the support of the footplate by using larger brackets to stabilize the cantilever.

3. Improving the footplate-load cell attachment by machining the support brackets from aluminium to eliminate flexibility.

4. As suggested by therapists, a second support point on the leg is needed so that the patient’s leg does not bend out of plane. A passive multi-DOF support point attached
under the knee should be added, which can accommodate a variety of leg sizes and stroke lengths.

A case for the PCB and Load Cell amplifier has been created, however the other electronics also need to be packaged together. A separate box should be built to contain the motor controller, power supply, DAQ and computer. This box should be portable and easily connected to the robot. This would allow the robot to be easily set up and removed as needed in the hospital. A shell needs to be built to protect other’s from any sharp edges, and so that the device can be cleaned or sterilized as needed. Some research should be done on what material to use. Finally, extra DOF’s should be added to accommodate different types of exercises. More research will need to be done concerning the best configuration.

### 6.3 Software Extension

The software was customized for this system, however future design iterations will likely make significant changes, including changing the controller to accommodate more DOF’s, and adding features to the UI. A software template was created from the control and UI software described above. It includes:

- Initialization and creating the threads
- Initializing the Data Log
- Creates Mutex
- Connects to DAQ

The programmer can then make customization for their specific system, as listed below. Essentially, this is a soft real time control software template which handles any boilerplate real time functions.

- Control loop algorithm
• What variables to send to the UI

• Which variables are saved in the data structures

• Which variables to log

The template also includes the UI and UI communication. React.js, the technology behind the UI, can be reconfigured and customized easily. Reusable components can be created, like buttons, dropdown menus, etc. All of the components designed for this UI can be easily moved around or reprogrammed. For example, the home button exists as an independent component and so can easily be moved to a different page or location, and the message it sends to the controller can also be changed.

6.3.1 User Interface Additions

The user interface needs improvement before it is ready for use by therapists in the hospital. The current UI is functional, and can be used as a framework for future additions, which should include:

1. Improved games and visuals
2. Add to the setup page to allow therapist’s to customize the parameters
3. Add a database to store patient data
4. Add a data viewing page which could allow therapist’s to monitor patient progress
5. Add security measures, like a username/password protocol.

6.3.2 Experiments

While the functional tests and pilot study resulted in useful data, more studies and experiments are needed before the robot is ready to be used in the hospital by therapists for acute stroke rehabilitation. Some suggestions for future experiments include:
1. An experiment measuring engagement objectively (using biometrics, for example).

2. An experiment investigating the effort metric used in this thesis (by comparing effort with, for example, EMG data).

3. Another healthy subject test when the secondary leg support and other improvements are complete.

4. Experiments in the hospital on healthy subjects, subjects who have recovered and have experience with rehabilitation.

These experiments are all required to verify that the device functions, is safe, and can potentially benefit acute stroke patients. The results from these experiments will be used to seek ethics approval for a large scale clinical trial on acute stroke patients. We plan to follow patients throughout their recovery, collecting data on outcomes to determine the efficacy of the rehabilitation robot.
Bibliography


[7] “Stroke [online].”


Appendix I: Custom Parts

Figure 6.2: The heel rest used to hold the user’s foot in place
Figure 6.3: The case for the PCB and load cell amplifier
Figure 6.4: Bracket to support footplate by carrying vertical load
Figure 6.5: Limit switch mount
Figure 6.6: Motor and gearhead mount (consisting of four plates)
## Appendix II: PCB Diagram & BOM

Table 6.1: PCB Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>24 V DC/DC Converter</td>
<td>Converts 48V to 24V</td>
</tr>
<tr>
<td>10 &amp; 5V Linear Regulator</td>
<td>Converts 48V to 10V and 5V</td>
</tr>
<tr>
<td>OP-AMP</td>
<td>Signal Processing</td>
</tr>
<tr>
<td>Comparator</td>
<td>Checks to ensure commands are not above/below threshold</td>
</tr>
<tr>
<td>4 Channel AND Gate</td>
<td>Only enables operation if all safety checks read ok</td>
</tr>
<tr>
<td>Watchdog Timer</td>
<td>Ensures controller is still running</td>
</tr>
</tbody>
</table>
Voltage Regulators

Figure 6.7: Printed circuit board schematic – Voltage Regulators
Figure 6.8: Printed circuit board schematic – Comparator & Safety AND Gate
Figure 6.9: Printed circuit board schematic – Signal Processing

Figure 6.10: Printed circuit board schematic – Watchdog Timer
Figure 6.11: Printed circuit board schematic – Header Pins
Figure 6.12: Printed circuit board layout – front (left) and back (right)
Appendix III: Hardware Datasheets

Figure 6.13: Operating Range of the Motor
www.maxongroup.com/maxon/view/product/motor/dcmotor/re/re50/370354
## Figure 6.14: Specifications for the Maxon motor

<table>
<thead>
<tr>
<th>Values at nominal voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal voltage</td>
</tr>
<tr>
<td>No load speed</td>
</tr>
<tr>
<td>No load current</td>
</tr>
<tr>
<td>Nominal speed</td>
</tr>
<tr>
<td>Nominal torque (max. continuous torque)</td>
</tr>
<tr>
<td>Nominal current (max. continuous current)</td>
</tr>
<tr>
<td>Stall torque</td>
</tr>
<tr>
<td>Stall current</td>
</tr>
<tr>
<td>Max. efficiency</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminal resistance</td>
</tr>
<tr>
<td>Terminal inductance</td>
</tr>
<tr>
<td>Torque constant</td>
</tr>
<tr>
<td>Speed constant</td>
</tr>
<tr>
<td>Speed / torque gradient</td>
</tr>
<tr>
<td>Mechanical time constant</td>
</tr>
<tr>
<td>Rotor inertia</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Thermal data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermal resistance housing-ambient</td>
</tr>
<tr>
<td>Thermal resistance winding-housing</td>
</tr>
<tr>
<td>Thermal time constant winding</td>
</tr>
<tr>
<td>Thermal time constant motor</td>
</tr>
<tr>
<td>Ambient temperature</td>
</tr>
<tr>
<td>Max. winding temperature</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mechanical data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bearing type</td>
</tr>
<tr>
<td>Max. speed</td>
</tr>
<tr>
<td>Axial play</td>
</tr>
<tr>
<td>Radial play</td>
</tr>
<tr>
<td>Max. axial load (dynamic)</td>
</tr>
<tr>
<td>Max. force for press fits (static)</td>
</tr>
<tr>
<td>(static, shaft supported)</td>
</tr>
<tr>
<td>Max. radial load</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of pole pairs</td>
</tr>
<tr>
<td>Number of commutator segments</td>
</tr>
<tr>
<td>Number of autoclave cycles</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
</tr>
</tbody>
</table>
General information

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gearhead type</td>
<td>GP</td>
</tr>
<tr>
<td>Outer diameter</td>
<td>52 mm</td>
</tr>
<tr>
<td>Version</td>
<td>Ceramic version</td>
</tr>
</tbody>
</table>

Gearhead Data

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduction</td>
<td>53 : 1</td>
</tr>
<tr>
<td>Absolute reduction</td>
<td>637/12</td>
</tr>
<tr>
<td>Max. motor shaft diameter</td>
<td>10 mm</td>
</tr>
<tr>
<td>Number of stages</td>
<td>3</td>
</tr>
<tr>
<td>Max. continuous torque</td>
<td>30 Nm</td>
</tr>
<tr>
<td>Max. intermittent torque</td>
<td>45 Nm</td>
</tr>
<tr>
<td>Direction of rotation, drive to output</td>
<td>=</td>
</tr>
<tr>
<td>Max. efficiency</td>
<td>75 %</td>
</tr>
<tr>
<td>Average backlash no load</td>
<td>1 °</td>
</tr>
<tr>
<td>Mass inertia</td>
<td>17.2 gcm²</td>
</tr>
<tr>
<td>Gearhead length (L1)</td>
<td>78.5 mm</td>
</tr>
<tr>
<td>Max. transmittable power (continuous)</td>
<td>360 W</td>
</tr>
<tr>
<td>Max. transmittable power (intermittent)</td>
<td>530 W</td>
</tr>
</tbody>
</table>

Technical Data

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radial play</td>
<td>max. 0.06 mm, 12 mm from flange</td>
</tr>
<tr>
<td>Axial play</td>
<td>0 - 0.3 mm</td>
</tr>
<tr>
<td>Max. radial load</td>
<td>900 N, 12 mm from flange</td>
</tr>
<tr>
<td>Max. axial load (dynamic)</td>
<td>200 N</td>
</tr>
<tr>
<td>Max. force for press fits</td>
<td>500 N</td>
</tr>
<tr>
<td>Max. continuous input speed</td>
<td>6000 rpm</td>
</tr>
<tr>
<td>Max. intermittent input speed</td>
<td>6000 rpm</td>
</tr>
<tr>
<td>Recommended temperature range</td>
<td>-15...+80 °C</td>
</tr>
<tr>
<td>Extended temperature range</td>
<td>-40...+100 °C</td>
</tr>
<tr>
<td>Number of autoclave cycles</td>
<td>0</td>
</tr>
</tbody>
</table>

Product

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>770 g</td>
</tr>
</tbody>
</table>

Figure 6.15: Specifications for the Maxon gearhead
### Figure 6.16: Specifications for the Maxon encoder

<table>
<thead>
<tr>
<th>General information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Counts per turn</td>
<td>500</td>
</tr>
<tr>
<td>Number of channels</td>
<td>3</td>
</tr>
<tr>
<td>Line Driver</td>
<td>DS26LS31</td>
</tr>
<tr>
<td>Max. mechanical speed</td>
<td>12000 rpm</td>
</tr>
<tr>
<td>Shaft diameter</td>
<td>8 mm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Technical Data</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply voltage $V_{CC}$</td>
<td>$5.0\text{V} \pm 10.0%$</td>
</tr>
<tr>
<td>Driver used logic</td>
<td>EIA RS 422</td>
</tr>
<tr>
<td>Max. angular acceleration</td>
<td>$250000 \text{rad} / \text{s}^2$</td>
</tr>
<tr>
<td>Output current per channel</td>
<td>$-20...20 \text{mA}$</td>
</tr>
<tr>
<td>Signal rise time</td>
<td>180 ns</td>
</tr>
<tr>
<td>Measurement condition for signal rise time</td>
<td>$CL=25\text{pF}, RL=2.7\text{kOhm}$,</td>
</tr>
<tr>
<td>Signal fall time</td>
<td>40 ns</td>
</tr>
<tr>
<td>Measurement condition for signal fall time</td>
<td>$CL=25\text{pF}, RL=2.7\text{kOhm}$,</td>
</tr>
<tr>
<td>Phase shift</td>
<td>$90^\circ$</td>
</tr>
<tr>
<td>Phase shift, inaccuracy</td>
<td>$45^\circ$</td>
</tr>
<tr>
<td>Index synchronized to AB</td>
<td>Yes</td>
</tr>
<tr>
<td>Max. moment of inertia of code wheel</td>
<td>0.6 gcm²</td>
</tr>
<tr>
<td>Operating temperature</td>
<td>$-40...+100 \degree\text{C}$</td>
</tr>
<tr>
<td>Orientation of encoder output to motor flange</td>
<td>$-1^\circ$</td>
</tr>
</tbody>
</table>
2 Performance Data

2.1 Electrical data

Nominal supply voltage $+V_{cc}$ ................................................................. 12 ... 50 VDC
Absolute minimum supply voltage $+V_{cc\min}$ ........................................... 11.4 VDC
Absolute maximum supply voltage $+V_{cc\max}$ ......................................... 52.5 VDC
Max. output voltage ................................................................................... $0.9 \cdot V_{cc}$
Max. output current $I_{\text{out}}$ ................................................................... 10 A
Continuous output current $I_{\text{cont}}$ .......................................................... 6 A
Switching frequency .................................................................................... 50 kHz
Efficiency .................................................................................................. 95 %
Band width current controller ................................................................... 2.5 kHz
Built-in motor choke ................................................................................... 150 $\mu$H / 5 A

2.2 Inputs

Set value ....................................................................................................... -10 ... +10 V ($R_s = 20 \, k\Omega$)
Enable .......................................................................................................... +4 ... +50 VDC ($R_s = 15 \, k\Omega$)
Input voltage DC tacho “Tacho Input” ......................................................... min. 2 VDC, max. 50 VDC ($R_s = 14 \, k\Omega$)
Encoder signals “Channel A, A, B, B” ......................................................... max. 100 kHz, TTL level

2.3 Outputs

Current monitor “Monitor I”, short-circuit protected ............................ -10 ... +10 VDC ($R_s = 100 \, \Omega$)
Speed monitor “Monitor n”, short-circuit protected ............................... -10 ... +10 VDC ($R_s = 100 \, \Omega$)
Status reading “READY” .................................................................
Open collector, short-circuit protected .................................................... max. 30 VDC ($I_s \leq 20 \, mA$)

2.4 Voltage outputs

Aux. voltage, short-circuit protected .................................................. +12 VDC, -12 VDC, max. 12 mA ($R_s = 1 \, k\Omega$)
Encoder supply voltage ........................................................................... +5 VDC, max. 80 mA

2.5 Trim potentiometers

- $R_s$
- Offset
- $n_{\text{max}}$
- $I_{\text{max}}$
- gain

---

Figure 6.17: Specifications for the Maxon motor controller
Figure 6.18: The load cell and amplifier connection diagram
<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rated Output (R.O.)</td>
<td>2 mV/V nominal</td>
</tr>
<tr>
<td>Nonlinearity</td>
<td>0.1% of R.O.</td>
</tr>
<tr>
<td>Hysteresis</td>
<td>0.1% of R.O.</td>
</tr>
<tr>
<td>Nonrepeatability</td>
<td>0.05% of R.O.</td>
</tr>
<tr>
<td>Zero Balance</td>
<td>1.0% of R.O.</td>
</tr>
<tr>
<td>Compensated Temp. Range</td>
<td>60° to 160°F</td>
</tr>
<tr>
<td>Safe Temp. Range</td>
<td>-65° to 200°F</td>
</tr>
<tr>
<td>Temp. Effect on Output</td>
<td>0.005% of Load/°F</td>
</tr>
<tr>
<td>Temp. Effect on Zero</td>
<td>0.005% of R.O./°F</td>
</tr>
<tr>
<td>Terminal Resistance</td>
<td>350 ohms nominal</td>
</tr>
<tr>
<td>Excitation Voltage</td>
<td>10 VDC</td>
</tr>
<tr>
<td>Safe Overload</td>
<td>150% of R.O.</td>
</tr>
<tr>
<td>Calibration Included</td>
<td>Compression</td>
</tr>
<tr>
<td>Optional Calibration</td>
<td>Tension</td>
</tr>
</tbody>
</table>

Figure 6.19: Specifications for the MLP load cell
Appendix IV: Additional Plots from Healthy Subject Experiment

Figure 6.20: Position comparisons for highest and lowest assistance levels (Participant 1)
Figure 6.21: Position comparisons for highest and lowest assistance levels (Participant 2)

Figure 6.22: Position comparisons for highest and lowest assistance levels (Participant 3)
Figure 6.23: Position comparisons for highest and lowest assistance levels (Participant 4)

Figure 6.24: Position comparisons for highest and lowest assistance levels (Participant 5)
Figure 6.25: Position comparisons for highest and lowest assistance levels (Participant 6)

Figure 6.26: Force comparisons for the highest and lowest resistance levels (Participant 1)
Figure 6.27: Force comparisons for the highest and lowest resistance levels (Participant 2)

Figure 6.28: Force comparisons for the highest and lowest resistance levels (Participant 3)
Figure 6.29: Force comparisons for the highest and lowest resistance levels (Participant 4)

Figure 6.30: Force comparisons for the highest and lowest resistance levels (Participant 5)
Figure 6.31: Force comparisons for the highest and lowest resistance levels (Participant 6)