ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc. THESIS

Metin ÜNLÜ

CONCEPTUAL DESIGN OF ANKLE REHABILITATION ROBOT FOR CEREBRAL PALSY PATIENTS

DEPARTMENT OF MECHANICAL ENGINEERING

ADANA-2022

ABSTRACT

MSc. THESIS

CONCEPTUAL DESIGN OF ANKLE REHABILITATION ROBOT FOR CEREBRAL PALSY PATIENTS

Metin ÜNLÜ

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF MECHANICAL ENGINEERING

Supervisor : Asst. Prof. Dr. Mehmet İlteriş SARIGEÇİLİ
Year: 2022, Pages: 98
Jury : Asst. Prof. Dr. Mehmet İlteriş SARIGEÇİLİ
: Asst. Prof. Dr. Durmuş Ali BİRCAN
: Asst. Prof. Dr. Necdet Sinan ÖZBEK

Physical therapy can assist patients with gait abnormalities and improve their walking skills significantly. Cerebral palsy (CP) patients are one of these patient groups. For CP patients, involuntary and abrupt muscle and joint contractions are the most common cause of gait abnormalities. Recent studies have shown that CP patients benefit from walking training by increasing their walking distance and speed, due to rehabilitation methods that focus on the lower extremities. For this reason, in this study, three different concept designs of ankle rehabilitation robot for CP patients are presented. The advantages and disadvantages of these designs are given and the proposed design is given in detail. The presented final design is aimed to support adolescents with CP aged from 12 to 18 and with weight up to 60 kg. The controllers and algorithms are written in Arduino and the robot operating system (ROS). The embedded software written in the ROS environment processes the gathered data to produce control signals. All the parts in the design are simulated in finite element analysis to check the factor of safety and shown that they are safe to operate.

Keywords: Ankle, exoskeleton robot, rehabilitation robot, cerebral palsy, ROS

ÖZ

YÜKSEK LİSANS TEZİ

SEREBRAL PALSİ HASTALARI İÇİN AYAK BİLEĞİ REHABİLİTASYON ROBOTU KAVRAMSAL TASARIMI

Metin ÜNLÜ

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MAKİNE MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Dr. Öğr. Üyesi Mehmet İlteriş SARIGEÇİLİ
Yıl: 2022, Sayfa: 98
Jüri : Dr. Öğr. Üyesi Mehmet İlteriş SARIGEÇİLİ
: Dr. Öğr. Üyesi Durmuş Ali BİRCAN
: Dr. Öğr. Üyesi. Necdet Sinan ÖZBEK

Fizik tedavi, yürüme anormallikleri olan hastalara yardımcı olabilmekte ve yürüme becerilerini önemli ölçüde geliştirebilmektedir. Serebral palsi (SP) hastaları da bu hasta gruplarından biridir. SP hastaları için, istemsiz ve ani kas ve eklem kasılmaları, yürüme anormalliklerinin en yaygın nedenidir. Son zamanlarda yapılan çalışmalar, alt uzuvlara odaklanan rehabilitasyon yöntemleri sayesinde SP hastalarının yürüme mesafelerini ve hızlarını artırarak yürüme eğitiminden fayda sağladığını göstermiştir. Bu nedenle bu çalışmada, SP hastaları için ayak bileği rehabilitasyon robotu tasarımına ilişkin üç farklı konsept sunulmuştur. Bu tasarımların avantaj ve dezavantajları verilmiş ve önerilen tasarım ayrıntılı olarak verilmiştir. Sunulan nihai tasarım, 12 ila 18 yaşları arasındaki ve ağırlığı 60 kg'a kadar olan SP'li ergenleri desteklemeyi amaçlamaktadır. Kontrolörler ve algoritmalar Arduino ve robot işletim sisteminde (ROS) yazılmıştır. ROS ortamında bulunan gömülü yazılım, kontrol sinyalleri üretmek için toplanan verileri işler. Tasarımdaki tüm parçalar güvenlik faktörünü kontrol etmek için sonlu elemanlar analizinde simüle edilmiş ve kullanıma hazır oldukları gösterilmiştir.

Anahtar Kelimeler: Bilek, giyilebilir robot, rehabilitasyon robotu, serebral palsi, ROS

EXPANDED ABSTRACT

Cerebral Palsy (CP) is a group of movement disorders that is mainly caused by damage to the brain during its developmental stage. CP patients may suffer from abnormal reflexes, loose muscle tones, tremors, rigidity in the body and extremities, gait and stance disorders. Rehabilitation methods, especially strength training results in an increased walking ability for CP patients. The symptoms of CP differ for patients as well as severity which makes rehabilitation therapy differ for patients. The work done by ankle joint is important during walking and CP patients who have gait disorders related to ankle joint benefit from ankle rehabilitation. The ankle rehabilitation is generally performed manually and includes joint manipulation and mobilization. This improves flexibility and alignment, loosens tight tissues, and reduces pain in the joint. However, manual therapy has limitations and robotic devices for ankle rehabilitation are increasing in importance for achieving more effective rehabilitation.

Platform-based ankle rehabilitation robots are stationary complex mechanisms that can achieve specialized spatial movements. On the other hand, wearable ankle rehabilitation robots such as exoskeleton robots can be used to increase the torque for a joint and are mainly used to increase the strength and endurance of limb movements. Active orthoses wearable ankle rehabilitation robots are the smaller version of exoskeletons that applies torque for required phases of the gait and allows the patient to walk more naturally.

Three different concept designs of ankle rehabilitation robot are presented in this thesis. All designs use brushless direct current (BLDC) motors for high torque capability in small sizes. Hydraulics and pneumatics actuation systems were also considered. However, hydraulics and pneumatics increase overall weight, the complexity of the control algorithm and overall cost of robot which are not desirable for the aimed design. In all designs, rotational joints are aligned with ankle axes of rotation.

The software framework of the robot has been developed through Robot Operating System (ROS). In this study, ROS is used with Python programming language to create nodes for the program. ROS offers hardware abstraction, message-passing between processes and nodes, package management and low-level device control.

The first concept design utilizes three linkages and joints to achieve a complex motion of the ankle in 3 DOF. The working mechanism of the design is similar to a robotic arm and can be easily controlled for even specialized spatial movements. This design is to be used in stationary configuration. The ankle motions of dorsiflexion-plantarflexion, inversion-eversion ad abduction-adduction movements can be achieved. The leg will be secured to the foot plate and any motion that is exerted on the plate by robotic links are also exerted on the ankle. However, the main objective is to improve walking gait, and studies have shown walking training can give a more natural rehabilitation process by involving the patient's motor activity, strengthening muscles, and improving neurological control. For this reason, a wearable design is needed and the first concept design is not suitable for a wearable design.

The second concept design uses two aluminum links to create a tripod mechanism. This design is wearable and powered by two BLDC motors in each leg. Two aluminum links are positioned in a vertical slot and are manipulated with steel wires. Motors are BLDC motors with 5500 rpm and an actuator is designed with a gear reduction ratio of 94.024 to increase torque capability and reduce angular speed.

One of the main problems of tripod mechanism is the obstruction of range of motion (ROM) unless designed carefully. Aluminum link connections to upper assembly and footplate are designed carefully to increase the ROM of the design. The aluminum links are driven by steel wires and the total displacement of the aluminum links in the vertical slot is derived from encoder data. Footplate displacement is bound to rotate on the ankle's rotational axes and displacement of aluminum link will develop rotation in sagittal and/or frontal axes. This design is suitable to be used during walking, however, two motors on the knee and ankle area create too much weight which will disrupt the natural walking kinematic chain.

The third and the proposed design is a simple wearable ankle rehabilitation robot with 2 DOF, these are the sagittal plane and the frontal plane. The mean of work done over one stride for sagittal motion is the greatest portion during walking, thus the selected BLDC motor for the sagittal plane covers main torque requirement for push-off. The motors of sagittal plane are equipped with encoders and are capable of controlling position, speed and torque. This design can support adolescents with CP aged from 12 to 18 and with weight up to 60 kg.

This design has two main assemblies, the backpack mechanism, and the ankle mechanism. The backpack mechanism includes motors, an inboard computer, and microcontrollers. The torques generated by motors are transmitted to the ankle mechanism with steel wires going through Bowden cable. A shoe inlet in the ankle mechanism is manipulated by two pulleys. This ankle rehabilitation robot will be used in collaboration with a body-weight support system to increase rehabilitation capabilities. The whole ankle rehabilitation robot with backpack mechanism and ankle mechanism weights around 3300 gr. The steel wire used in sagittal plane is 2.5mm diameter and is capable of carrying the peak force of 889 N for the robot with a safety factor of 5.3 and is safe to use. The steel wire winding in backpack mechanism and path of steel wire between backpack mechanism and ankle mechanism are covered for safety.

The controllers and algorithms are written in Arduino and robot operating system (ROS). The frontal DC motors can be directly controlled through Arduino Nano with and additional motor driver. However, sagittal BLDC motors use Controller Area Network (CAN bus) for communication. A CAN bus communication module is added to achieve communication between Arduino and the sagittal motors. Each Arduino Nano controls motors and sensors responsible with one leg. Thus, there are two Arduino Nano in the system. Arduino Nano sends control signal to the BLDC motor, gathers torque and rotational information from BLDC motor through CAN bus, reads encoder data, sends control signal to DC motor and gather information from sensors.

Arduino publishes the gathered data in the ROS environment. ROS operates on an inboard computer. ROS reads this data to make the next decisions and publish required motor signals to Arduino. In the ROS, two custom messages are created for more suitable communication message type, these custom messages includes motor data and force sensor data. The algorithm is an open control loop and it sends required signals to the motors based on the gait cycle. This control algorithm is strict during operation on following the gait cycle and rehabilitation is required to be done continuously without interruption.

The robot is equipped with safety measures to control and stop motors in any case of malfunction. The safety is achieved mainly by controlling and reading the current data of motors and the rotational values from encoders.

The robot is analyzed for two different stride phases in the gait cycle. One is flat stance and the highest torque is achieved at this phase. The other stride phase is heel strike and the ground reaction force is required to be simulated for safety. All the parts in the design are simulated in FEA for both stride phase and shown that they have a high factor of safety and are ready to use.

GENİŞLETİLMİŞ ÖZET

Serebral Palsi (SP), temel olarak gelişim aşamasında beyinde oluşan hasardan kaynaklanan bir grup hareket bozukluğudur. SP hastalarında anormal refleksler, gevşek kas tonusu, titreme, vücut ve ekstremitelerde sertlik, yürüme ve duruş bozuklukları olabilir. Rehabilitasyon yöntemleri, özellikle kuvvet antrenmanı, SP hastaları için yürüme kabiliyetinin arttırmaktadır. SP'nin semptomları hastadan hastaya farklılık gösterdiği gibi şiddeti de farklılık göstermektedir, bu sebeple rehabilitasyon tedavileride farklılık göstermektedir. Yürüme sırasında ayak bileği ekleminin yaptığı iş önemlidir ve ayak bileği eklemine bağlı yürüme bozukluğu olan SP hastaları için ayak bileği rehabilitasyonunu yürüyüş kabiliyetini artırabilmektedir. Ayak bileği rehabilitasyonu genellikle manuel olarak yapılır ve eklem manipülasyonu ve mobilizasyonu içerir. Bu, esnekliği ve hizalamayı geliştirir, sıkı dokuları gevşetir ve eklemdeki ağrıyı azaltır. Fakat, manuel terapinin sınırlamaları vardır ve daha etkili rehabilitasyon sağlamak için ayak bileği rehabilitasyonu robotik cihazlarının önemi artmaktadır.

Ayak bileği rehabilitasyon robotlarına iki farklı yaklaşım vardır: Platform tabanlı robotlar ve giyilebilir robotlar. Platform tabanlı ayak bileği rehabilitasyon robotları, özel uzaysal hareketler gerçekleştirebilen sabit karmaşık mekanizmalardır. Öte yandan, giyilebilir ayak bileği rehabilitasyon robotları bir eklemin torkunu artırmak için kullanılabilir ve esas olarak uzuv hareketlerinin gücünü ve dayanıklılığını artırmak için kullanılır. Aktif ortez giyilebilir ayak bileği rehabilitasyon robotları, yürüyüşün gerekli aşamaları için tork uygulayan ve hastanın daha doğal yürümesini sağlayan dış iskeletlerin daha küçük versiyonudur.

Bu tezde ayak bileği rehabilitasyon robotunun üç farklı konsept tasarımı sunulmaktadır. Tüm tasarımlar, küçük boyutlarda yüksek tork kapasitesi için fırçasız doğru akım (BLDC) motorları kullanılmıştır. Aynı zamanda Hidrolik ve pnömatik sistemler de dikkate alınmıştır. Bununla birlikte, hidrolik ve pnömatik, amaçlanan

tasarım için arzu edilmeyen özellikler vardır, bunlar yüksek ağırlık, kontrol algoritmasının karmaşıklığını ve yüksek maliyet. Tüm tasarımlarda, cihazın döner eklemleri, ayak bileği dönme eksenleri ile hizalanmıştır.

Robotun yazılımı Robot İşletim Sistemi (ROS) üzerinden geliştirilmiştir. Bu çalışmada, program içeren düğümler için Python programlama dili ile ROS kullanılmıştır. ROS, donanımın etkili kullanımı, prosesler ve düğümler arasında mesaj geçişi, paket yönetimi ve düşük seviyeli cihaz kontrolü özelliklerini sunar.

İlk konsept tasarım, üç serbestlik derecesinde ayak bileğinin karmaşık hareketini elde etmek için üç bağlantı ve eklem kullanır. Tasarımın çalışma mekanizması robotik bir kola benzer ve özel mekansal hareketler için bile kolayca kontrol edilebilir. Bu tasarım sabit konfigürasyonda kullanılacaktır. Ayak bileğinin dorsifleksiyon-plantarfleksiyon, inversiyon-eversiyon ve abduksiyon-addüksiyon hareketleri yapılabilir. Bacak, ayak plakasına sabitlenecek ve plakaya robotik bağlantılar tarafından uygulanan herhangi bir hareket ayak bileğine de uygulanacaktır. Bununla birlikte, temel amaç yürüme yürüyüşünü iyileştirmektir ve çalışmalar, yürüme eğitiminin hastanın motor aktivitesini dahil ederek, kasları güçlendirerek ve nörolojik kontrolü geliştirerek daha doğal bir rehabilitasyon süreci sağlayabileceğini göstermiştir. Bu nedenle giyilebilir bir tasarıma ihtiyaç vardır ve ilk konsept tasarım giyilebilir bir tasarım için uygun görülmemiştir.

İkinci konsept tasarım, bir tripod mekanizması oluşturmak için iki alüminyum bağlantı kullanır. Bu tasarım giyilebilir bir tasarımdır ve her bacakta iki BLDC motor tarafından desteklenmektedir. İki alüminyum bağlantı dikey bir yuvaya yerleştirilmiştir ve çelik teller ile manipüle edilmektedir. Motorlar 5500 rpm'lik BLDC motorlardır ve tork kapasitesini artırmak ve açısal hızı azaltmak için 94.024 dişli redüksiyon oranına sahip bir aktüatör tasarlanmıştır.

Tripod mekanizmasının temel problemlerinden biri, dikkatli bir şekilde tasarlanmadıkça hareket aralığının engellenmesidir. Alüminyum çubukların üst düzeneğe ve ayak plakasına bağlantıları, tasarımın ROM'unu artırmak için dikkatlice tasarlanmıştır. Alüminyum bağlantılar çelik teller tarafından kontrol edilmektedir ve dikey yuvadaki alüminyum bağlantıların toplam yer değiştirmesi, sensör verilerinden elde edilir. Ayak plakasının yer değiştirmesi, ayak bileğinin dönme eksenleri üzerinde dönmeye bağlıdır ve alüminyum bağlantının yer değiştirmesi, sagital ve/veya ön eksenlerde dönme hareketi oluşturmaktadır. Bu tasarım yürüyüş sırasında kullanıma uygundur ancak diz ve ayak bileği bölgesindeki iki motor çok fazla ağırlık oluşturarak doğal yürüme kinematik zincirini bozmaktadır.

Üçüncü ve önerilen tasarım, iki serbestlik dereceli basit bir giyilebilir ayak bileği rehabilitasyon robotudur, bunlar sagital düzlem ve ön düzlemdir. Sagital hareket için bir adımda yapılan işin ortalaması, yürüme sırasındaki en büyük kısımdır, bu nedenle sagital düzlem için seçilen BLDC motor, itme için ana tork gereksinimini karşılar. Sagital düzlemin motorları enkoderlerle donatılmıştır ve konum, hız ve torku kontrol etme özelliğine sahiptir. Bu tasarım, 12-18 yaş arası ve 60 kg ağırlığa kadar SP'li ergenleri destekleyebilir.

Bu tasarım, sırt çantası mekanizması ve ayak bileği mekanizması olmak üzere iki ana düzeneğe sahiptir. Sırt çantası mekanizması, motorları, bir dahili bilgisayarı ve mikro denetleyicileri içerir. Motorların ürettiği torklar, Bowden kablosundan geçen çelik teller ile ayak bileği mekanizmasına iletilir. Ayak bileği mekanizmasındaki bir ayakkabı plakası iki kasnak tarafından manipüle edilir. Bu ayak bileği rehabilitasyon robotu, rehabilitasyon yeteneklerini artırmak için bir vücut ağırlığı destek sistemi ile birlikte kullanılacaktır. Sırt çantası mekanizması ve ayak bileği mekanizması ile beraber tüm ayak bileği rehabilitasyon robotu yaklaşık 3300 gram ağırlığındadır. Sagital düzlemde kullanılan çelik tel 2,5 mm çapındadır ve 889 N'lik maksimum çekme kuvvetini, 5,3'lük bir güvenlik faktörü ile taşıyabilmektedir ve kullanımı güvenlidir. Sırt çantası mekanizmasındaki çelik tel sargı ve sırt çantası mekanizması ile ayak bileği mekanizması arasındaki çelik tel yolu güvenlik için kapatılmıştır.

Kontrolörler ve algoritmalar Arduino ve robot işletim sisteminde (ROS) yazılmıştır. Ön DC motorlar, ek motor sürücüsü ile Arduino Nano üzerinden doğrudan kontrol edilebilir. Ancak sagital BLDC motorlar, iletişim için Kontrolör Alan Ağı'nı (CAN bus) kullanır. Arduino ve sagital motorlar arasındaki iletişimi sağlamak için bir CAN bus iletişim modülü eklenmiştir. Her Arduino Nano, tek bacakla sorumlu motorları ve sensörleri kontrol eder. Bu sebeple sistemde iki adet Arduino Nano bulunmaktadır. Arduino Nano, BLCD motora kontrol sinyali gönderir, BLCD motordan CAN bus üzerinden tork ve dönme bilgilerini toplar, enkoder verilerini okur, DC motora kontrol sinyali gönderir ve sensörlerden bilgi toplar.

Arduino toplanan verileri ROS ortamında yayınlar. ROS, yerleşik bir bilgisayarda çalışır. ROS, sonraki kararları vermek ve gerekli motor sinyallerini Arduino'ya yayınlamak için bu verileri okur. ROS'ta daha uygun iletişim mesaj tipi için cihaza özel iki mesaj oluşturmuştur, bu özel mesajlar motor verilerini ve kuvvet sensörü verilerini içerir. Algoritma açık bir kontrol döngüsünde çalışır ve yürüme döngüsüne göre gerekli sinyalleri motorlara gönderir. Bu kontrol algoritması, operasyon sırasında yürüyüş döngüsünü takip etmek konusunda katıdır ve rehabilitasyonun kesintisiz olarak yapılması gerekir.

Robot, herhangi bir arıza durumunda motorları kontrol etmek ve durdurmak için güvenlik önlemleri ile donatılmıştır. Güvenlik, esas olarak motorların akım verilerinin ve enkoderlerden gelen dönüş değerlerinin kontrol edilmesi ve okunmasıyla sağlanır.

Robot, yürüyüş döngüsünde iki farklı adım aşaması için analiz edilmiştir. Biri düz duruştur ve en yüksek tork bu aşamada elde edilir. Diğer adım aşaması topuk vuruşudur ve güvenlik için yer reaksiyon kuvvetinin simüle edilmesi gerekir. Tasarımdaki tüm parçalar her iki adım aşaması için FEA'da simüle edilmiş ve yüksek güvenlik faktörüne sahip oldukları ve kullanıma hazır oldukları gösterilmiştir.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Asst. Prof. Dr. Mehmet İlteriş SARIGEÇİLİ. I am grateful and indebted to him for his assistance, valuable guidance, and encouragement extended to me.

I must express my very profound gratitude to my parents and my brother who have always been with me and given their unending support.

TABLE OF CONTENTS

PAGE

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

XIX

LIST OF ABBREVIATIONS

A	: Ampere
BLDC	: Brushless Direct Current
BW	: Bodyweight
CAD	: Computer-aided design
CAN bus	: Central Processing Unit
CP	: Cerebral Palsy
CPU	: Central Processing Unit
DC	: Direct Current
DOF	: Degrees of Freedom
FEA	: Finite element analysis
F_z	: Vertical thrust maximum force
g	: gravity
GMFCS	: Gross Motor Function Classification System
IO	: Input and Output
MPa	: Mega Pascal
ms^{-2}	: meter per second squared
N	: Newton
Nm	: Newton-meter
ROM	: Range of Motion
ROS	: Robot Operating System
rpm	: Rotation per minute
V	: Voltage
α	: Angle of rotation

1. INTRODUCTION

Cerebral Palsy (CP) is a group of movement disorders that is mainly caused by damage to the brain during its developmental stage. CP patients may suffer from abnormal reflexes, loose muscle tones, tremors, rigidity in the body and extremities, and gait and stance disorders. The CP has an incidence rate of 1.5-2.5 per 1000 live-born children and it is one of the most common causes of movement disability in childhood (Himmelmann et al., 2005). Cerebral palsy can't be cured but a child's capabilities will often be improved with proper treatment.

Formerly, strength training was not considered to be appropriate for children with the CP as it was thought it could lead to increased spasticity or abnormal patterns. However, studies have shown that progressive strength training can lead to improved functions (Scholtes et al., 2008). As a result, it was shown that the motor function in children with CP caused the main limitation by low muscle strength instead of spasticity. This has changed the rehabilitation methods for children with CP as previously they were focused on spasticity management and currently, they are focused more on strength training (Ross & Engsberg, 2007).

The CP patients that have gait disorders related to ankle joint can benefit from ankle rehabilitation. The ankle is important in gait as it provides a great portion of power during walking (Farris & Sawicki, 2012). It should be noted that the CP affects individuals differently, which makes strength training and therapy differ for patients. Strength training is performed generally by manual therapy and includes joint manipulation and mobilization for measured movements at varying speed, force, and distance to pull or push joints. This type of rehabilitation helps to improve flexibility and alignment, loosen tight tissues around a joint, and reduce pain in the joint and surrounding tissue. However, manual therapy has certain limitations such as inconsistency in the treatment caused by monotonous and repetitive training. Thus, robotic devices for ankle rehabilitation are gaining popularity and are increasing in importance for achieving more effective rehabilitation (Hussain et al., 2017).

There are two different approaches to ankle rehabilitation robots: Platform-based robots or wearable devices that are either exoskeletons or active orthoses. Platform-based robots mostly perform complex passive training while wearable devices can be used both in passive or active simpler training configuration.

Platform-based robots are stationary robot devices that are not limited like exoskeletons or orthoses that have to be wearable. This makes it possible for platform-based robots to achieve more complex and specialized spatial movements. With the platform-based robots, the patient’s foot is mostly attached to a mobile platform generating the required motions and forces. The platform-based robots can achieve better adaptability to the mechanical properties of ankle joints with high dynamic stiffness and have a more balanced force distribution. However, the platform-based robots are often not used for gait recovery exercises since they are not as easily portable as exoskeletons or active orthoses. Also, a specialist is required for the management of the platform-based robots (Alvarez-Perez et al., 2020). On the other side, the exoskeletons are wearable powered robotics devices and are mainly used to increase the strength and endurance of limb movements. The objective of exoskeleton rehabilitation devices may vary as they can be used to increase the torque for a joint, reduce metabolic cost during gait, and contribute to walking speed or during rehabilitation sessions (Shi et al., 2019). The active orthotic devices are reduced versions of exoskeletons and actively control one or more joints by applying power or torque for required phases of the gait cycle which allows the patient to walk more naturally, and/or decrease the metabolic energy cost of the walking gait. The active orthotic devices are portable devices that can be used both for assistance and therapy which creates a great advantage for ankle rehabilitation robots (Dollar & Herr, 2007).

In order to improve walking gait performance, active orthotic robots are suitable design choices for ankle rehabilitation robots because of the advantages of being lightweight, portable, non-interfering with a natural gait, and suitable to be used both for assistance and therapy. In this study, an ankle rehabilitation robot with cable-based actuation is developed for cerebral palsy patients. The target patient group of the robotic device is adolescents with CP aged 12-18 and with a weight under 60 kg.

1.1. Cerebral Palsy

Cerebral palsy describes a group of neurological disorders that appear in infancy or early childhood causing activity limitation and are attributed to non-progressive disturbances. The motor disorders of CP are often accompanied by disturbances of sensation, communication, perception, cognition, and/or behaviors, and/or by a seizure disorder (Bax et al, 2005). CP affects the motor area of the brain's outer layer which is called the cerebral cortex, the part of the brain that directly controls muscle movement (https://www.ninds.nih.gov/health-information/patient-caregiver-education/hope-through-research/cerebral-palsy-hope-through-research (accessed by May, 2022)).

The symptoms of CP differ from patient to patient and include:

- delays in reaching motor skill development milestones.
- stiff or tight muscles;
- weak arms or legs;
- unnatural walking gait such as walking on tiptoes;
- variations in muscle tone;
- difficulties in swallowing or speaking or excessive drooling;
- tremor or random uncontrolled movements;
- muscle spams
- difficulty with precise movements.

The severity of symptoms for CP patients changes which makes it difficult to make distinctions among the CP patients. The Gross Motor Function

Classification System (GMFCS) was developed to provide a method for classifying patients with CP on the basis of motor functional abilities and limitations. The GMFCS includes five levels and helps to make the distinction between different levels of CP symptoms severity and represent motor functional skills in the daily lives of the CP patients. The GMFCS levels and descriptions are summarized in Table 1 for CP patients aged 12-18.

Table 1-1. The Gross Motor Function System (GMFCS) levels for cerebral palsy patients aged 12-18 (Palisano et al., 1997)

GMFCS Level	Description
I	Walks without restrictions; limitations in more advanced gross motor skills
II	Walks without assistive devices; limitations in walking outdoors and in community
III	Walks with assistive devices; limitations in walking outdoors and in community
IV	Self-mobility with limitations; children are transported or use powered mobility outdoors and in community
V	Self-mobility is severely limited, even with the use of assistive technology

Studies have highlighted that while spasticity affects mobility, the greatest limitations in motor functions were caused by low muscle strength (Ross & Engsberg, 2007), which has shifted the rehabilitation of CP patients from spasticity management to strength training. Also, studies in strength training show lower limb muscle strength can be improved without increasing spasticity.

Children with CP classified at different levels of the Gross Motor Function Classification System (GMFCS) were studied for their walking energy cost compared to typically developing children (Johnston et al., 2004). A study has found children with CP had increased energy costs during walking with increasing levels of GMFCS, whereas children at higher GMFCS levels had up to 10 times of energy cost required compared to typically developing (TD) children as shown in Figure 1-1.

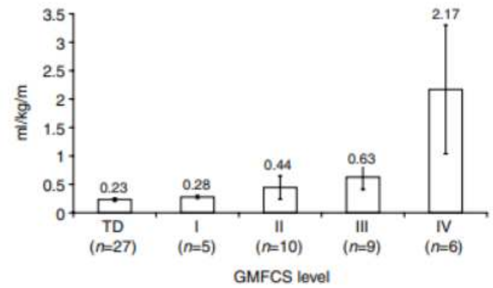


Figure 1-1. Means and standard deviations of energy cost of walking values per GMFCS level. (Johnston et al., 2004)

For improving muscle stimulation, joint range of motion, and increased flexibility, leg exercises are a powerful option. These leg exercises are:

- knee flexion and extension,
- hip rotations,
- ankle rotations,
- hip abductions,
- calf stretch,
- butterfly stretch,
- hamstring stretch,
- leg lifts,
- seated marching,
- walking.

Some of these exercises are passive while some of them are active. Active exercises such as leg lifts can be done with additional weights to increase resistance for better muscle growth. Walking exercises additionally help reduce spasticity and promote correct walking form. With a regular repeat of exercises and neuroplasticity, the brain's ability to rewire itself and adapt is stimulated.

1.2. Biomechanics of Walking

The mechanics of walking is complex and involves a coordinated effort of the hips, knee, and ankles that requires balance and forward propulsion. The walking gait cycle has two phases: stance is where the foot is on the ground and takes around 60% of the cycle and the remaining cycle is swing where the foot is off the ground. The stance phase involves heel strike to the ground, full flat contact, heel off, and toe off respectively. This cycle is shown in Figure 1-2.

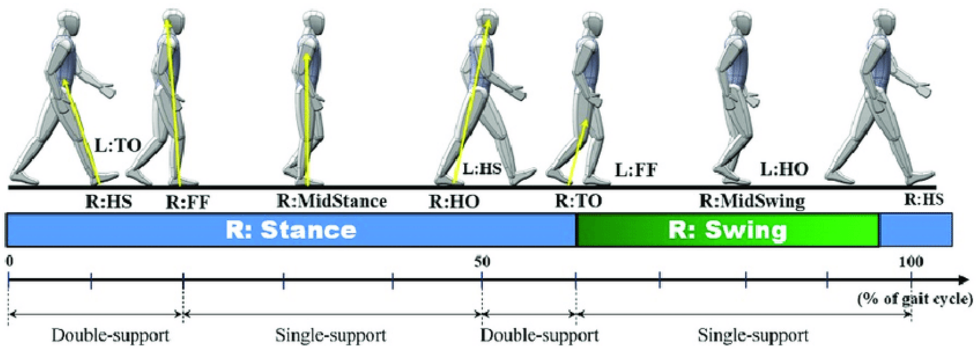


Figure 1-2. Gait cycle for human walking of the right leg. (Alamdari & Krovi, 2017)

In the study of Farris et al. (2012), mechanics of lower limb joints and metabolic costs were examined for walking and running. Instantaneous joint powers during walking and running for the ankle, hip, and knee were shown. From this study, it was observed that the ankle provides a great portion of power during walking gait. It was concluded that the ankle and hip both contribute to the total average power of 40-50% during walking whereas the hip contributed 32-37% while the ankle contributed 42-47% of the power during running. This shows the importance of the ankle in walking gait.

The motion of the ankle joint complex is called as plantarflexion and dorsiflexion, abduction and adduction, and inversion-eversion. These motions and the planes of the ankle are shown in Figure 1-3. With two subtalar and tibiotalar joints, the combination of these motions creates three-dimensional motions called supination and pronation. While the range of motion can vary significantly among individuals based on their activities of daily living, motion in the sagittal plane for plantarflexion range around 40-55° and for dorsiflexion range around 10-20° while motion in the frontal plane for inversion is around 23° and for eversion it is 12°.

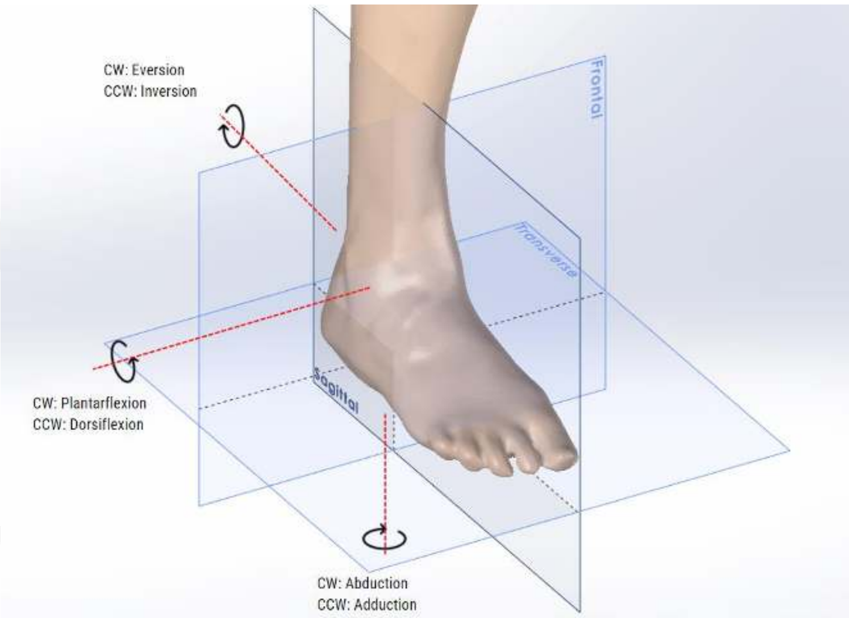


Figure 1-3. Planes and movements for ankle.

Eng et al. (1995) studied lower limbs in a three-dimensional model to see the effects of each axis at joints for the ankle, knee, and hip. In Table 1-2, the work done over one stride is given for the ankle, knee, and hip. It was shown that for the ankle, during walking, plantarflexion and dorsiflexion had the biggest role in the total work done, which is followed by inversion and eversion, and finally transverse rotations. This information helps us to make a walking rehabilitation device that is mainly focused on the sagittal plane and frontal plane, discarding the transverse plane.

Table 1-2. Means and standart deviations of the work (J kg -1) over one stride. (Eng & Winter, 1995)

	Sagittal		Frontal		Transverse		Total	Total	
	gen	abs	gen	abs	gen	abs	gen	abs	Total
Ankle	0.42 (0.09)*	0.10 (0.05)	0.015 (0.008)	0.014 (0.007)	0.005 (0.003)	0.006 (0.002)	0.44	0.12	0.56
Knee	0.11 (0.052)	0.35 (0.094)	0.035 (0.013)	0.026 (0.012)	0.008 (0.004)	0.015 (0.01)	0.15	0.39	0.54
Hip	0.48 (0.12)	0.14 (0.053)	0.092 (0.03)	0.10 (0.05)	0.009 (0.005)	0.022 (0.013)	0.58	0.26	0.84

***Standart Deviations in parentheses**
gen= energy generation, abs=energy absorption

During walking, the ankle pronates and supinates as shown in Figure 1-4. Pronation (heel strike / toe strike) mainly occurs in the stance phase of gait for ground terrain changes, equilibrium, and shock absorption (Donatelli, 1985). Pronation can be observed during the heel strike. Supination (toe strike/push-off) occurs at the end of the stance phase of gait. This motion sets up a rigid level from which to push of.

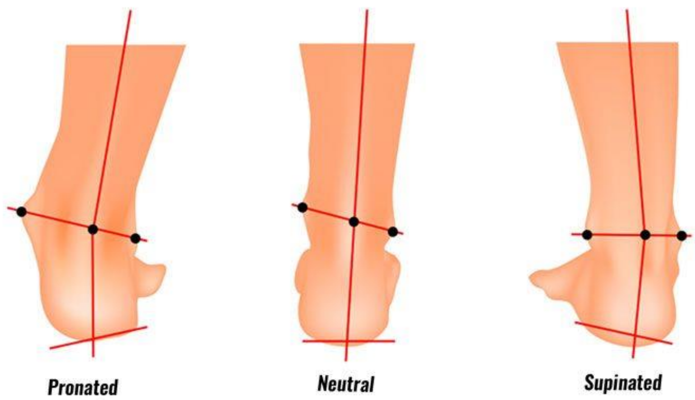


Figure 1-4. Pronation and supination of ankle (https://www.sportsinjuryclinic.net/treatments-therapies/foot-biomechanics-gait-analysis/over-supination-foot-biomechanics (accessed by May, 2022)).

2. PRELIMINARY WORK

Rehabilitation robots are getting increased popularity with technological advances in electrical and mechanical components. Finding high torque motors within a desirable budget gets easier with better improvements every year. There are many different fields where rehabilitation robots are used. Most rehabilitation robots mainly focus on a limb or joint, while some rehabilitation robots, like exoskeletons, can cover and control more than one limb. In this paper, lower limb rehabilitation robots are reviewed. In general, the rehabilitation robots for lower limb aims to regain and/or improve gait. Each robotic device may target a different outcome, such as some of them may target muscle growth whereas others may mainly focus on improving motor functions. The main motivation in the functional variety is determined by the requirement of user (i.e. patient).

There have been a lot of unique ankle rehabilitation robot studies over the years. Most of the ankle rehabilitation robot studies are platform-based robots, while others are wearable devices. The platform-based robots are stationary devices and can achieve complex and specialized spatial movements that can have highly balanced force distribution. In addition, the platform-based robots have better adaptability to the mechanical properties of human ankle joints with high dynamic stiffness compared to wearable devices. On the other hand, the wearable robots are powered robotic devices that can be used to increase the torque for a joint, reduce metabolic cost during gait or during rehabilitation sessions, and/or contribute to walking speed. The wearable devices can be divided into two categories: exoskeletons and active orthotic devices. The active orthotic devices are reduced versions of exoskeletons. The active orthotic robotic devices actively control one or more joints by applying torque or power for the required phase in the gait cycle. These portable devices can be used for assistance and therapy.

Both platform-based robots and active orthotic devices are reviewed for their design, mechanism, and rehabilitation outcomes.

2.1. Platform-based Robots

Improving motor control and muscle strength helps children with CP to gain better mobility. Wu et al. (2011) studied a combination of passive stretching and active movement rehabilitation of lower limb using a portable platform-based robot for children with CP as shown in Figure 2-1.

Figure 2-1. Training setup where participant was seated on the chair with knee extended. (Wu et al., 2011)

The result of this training method demonstrated that it can benefit children with CP who have impaired ankle function. The passive stretching mainly helped children with CP by extending the spastic muscles as well as reducing joint stiffness. This may facilitate ankle function and muscle force generation (Zhang et al., 2002). However, even though the passive stretching is critical for muscle flexibility, stretching alone may not be sufficient. The possible reasons for insufficient effect for children with CP may be due to immobility and/or undesired muscle activations that are associated with dystonia or spasticity (Damiano, 2009). For these reasons, computer games were used to engage the children in exercises in their active movement training using a robotic device. Since the movement training with computer games allows the children to receive biofeedback of the ankle position during playing the game, it was observed to improve functions along the neural pathway between signal reception and movement execution. This can help individuals to develop increased voluntary control and greater awareness. The results of the training were shown in Figure 2-2.

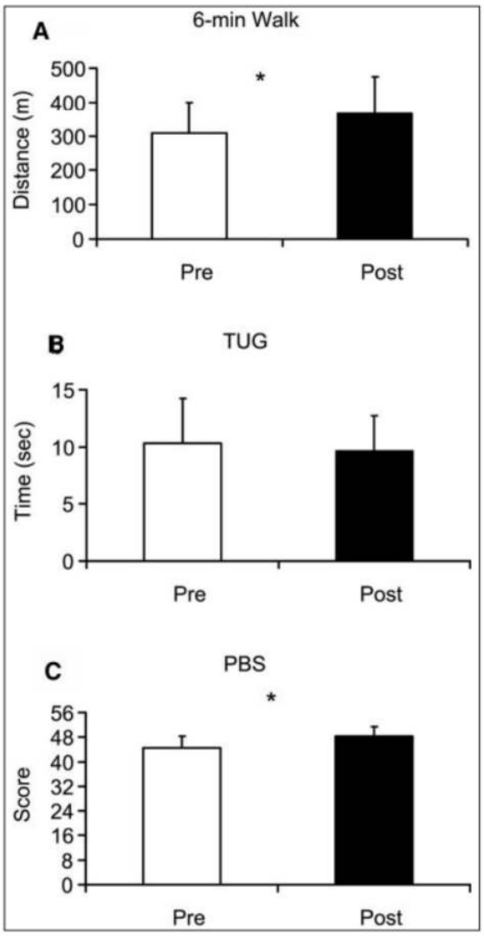
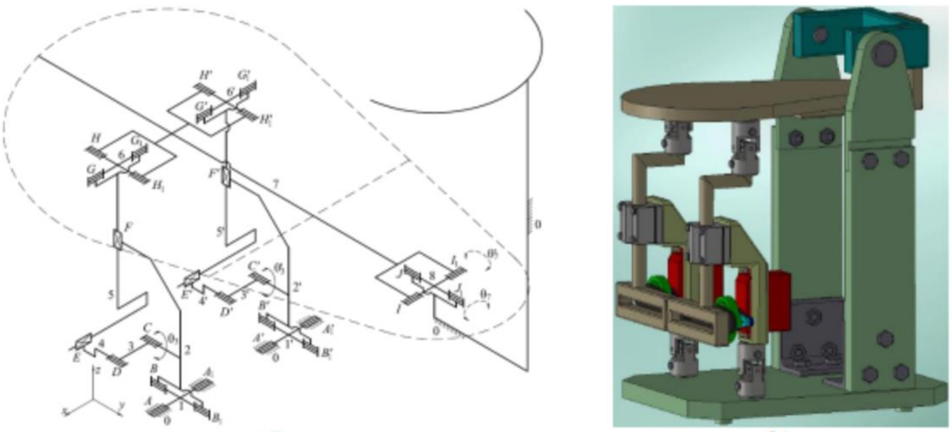


Figure 2-2. Functional outcome measures (A, 6-minute walk; B, Timed Up-and Go; C, Pediatric Balance Scale) before and after the 6-week intervention. *Indicates significant difference (Wu et al., 2011)

Racu and Doroftesi (2015) proposed a stationary ankle rehabilitation device with two degrees of freedom that can provide the dorsiflexion and plantarflexion movements as well as pronation and supination rotations. The proposed kinematics of the device is presented in Figure 2-3a., and the 3D design of the mechanism is presented in Figure 2-3b. The device utilizes a simple scotch yoke mechanism. To servo motors control the mechanism and with both motors rotating in the same direction provide dorsiflexion-plantar flexion movement, while rotating motors with the same speed in different directions will provide inversion-eversion movement A torsion spring is coupled to the flange and to the rotational arm. The deflection of the spring is measured to find joint torque directly. The spring design was implemented by S. Wang and C. Meijneke (2013) in the optimization of the exoskeleton joint. This spring design consists of two connected Archimedean springs.



a) b)

Figure 2-3. The stationary ankle rehabilitation device with two degrees of fredoom: a) kinematic drawing of the design; b) 3D computer-aided design (Racu & Doroftei, 2015)

Sung et al. (2011) designed a mechanism that mimics the inversion/eversion and plantarflexion / dorsiflexion mechanism of an ankle. The designed ankle rehabilitation device is shown in Figure 2-4 and it is equipped with cartwheel flexures as sensors that serve the purpose of measuring the deflection and load applied by the human ankle joint. The device consists of a base aluminum plate with vertical bars, two rotational linkages for plantarflexion flexures and inversion flexures, and finally, a footplate where the foot is strapped in securely onto the plate. Unlike powered rehabilitation devices that create motion, this ankle rehabilitation device aims to achieve a quantitative physical assessment tool to evaluate the performance of the human ankle joint complex.

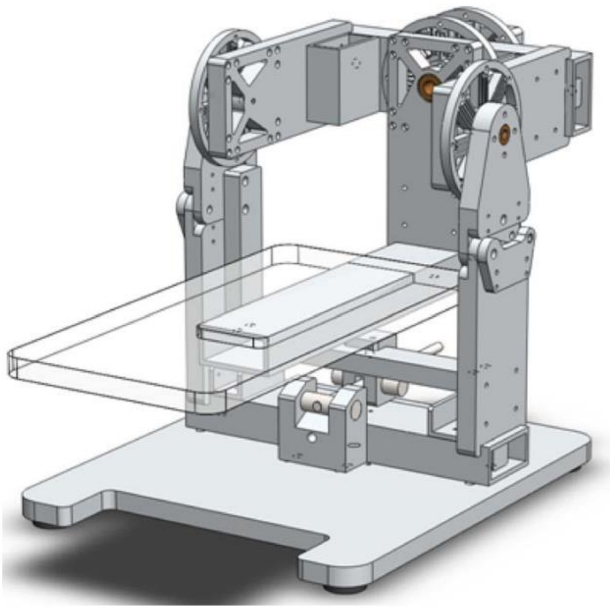


Figure 2-4. Solid model of the prototype ankle rehabilitation device. (Sung et al., 2011)

Another complex mechanism as a platform-based ankle rehabilitation robot is the parallel mechanism. This mechanism has multi degrees of freedom and is capable of carrying out ankle rehabilitation exercises. A design study on a parallel mechanism was presented by Saglia et al. (2009). The proposed ankle rehabilitation device consisted of three DC motors as linear actuators based on a cable-driven system that transforms the rotary motion into linear motion. This design configuration has the advantage of achieving low friction, high back drivability, and high-force capacity. The mechanism is shown in Figure 2-5. It should be noted that the parallel mechanisms can achieve different exercises both in passive and active mode. In the passive mode, pain-free passive motions can be carried out to flex and stretch muscles. In the active mode, the patient has to apply force and torque to the platform and this requires back-drivable actuators. Active mode rehabilitation exercises used in strength training and it is helpful to improve gait.

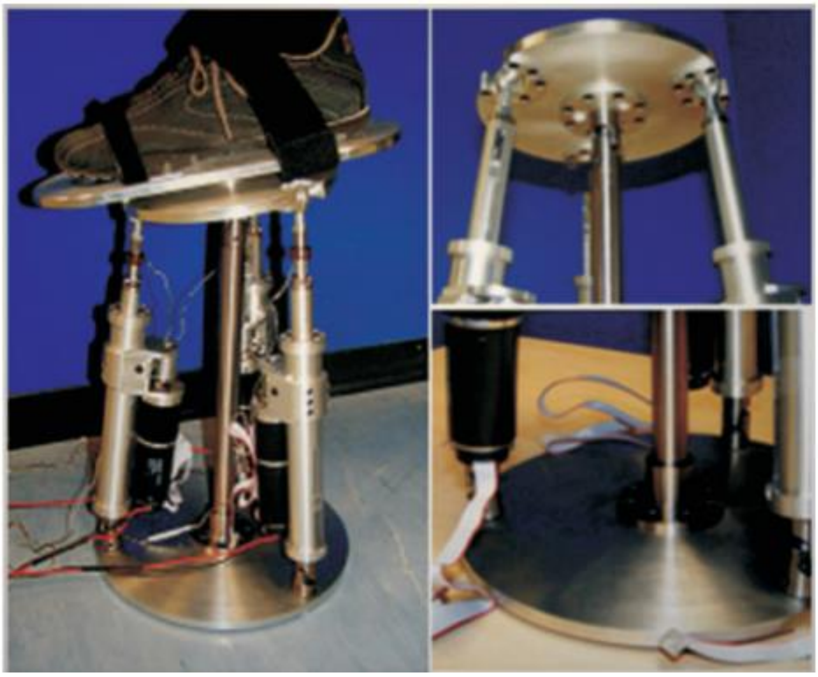


Figure 2-5. The parallel mechanism ankle rehabilitation robot with electric linear actuators (Saglia et al., 2009)

2.2. Wearable Ankle Rehabilitation Robots

A wearable ankle rehabilitation robot can be used for assistance, training, and/or therapy. There are many different choices in wearable ankle rehabilitation robots. Some of them are designed as an exoskeleton while others are designed as active orthoses. In general, it is observed that electrical actuation is the most popular actuator selection followed by pneumatics (Alvarez-Perez et al., 2020). The development of wearable rehabilitation devices is complex and includes challenging problems such as weight, power, and usability. However, there are many commercial exoskeletons and active orthoses robots that have been proven for their effectiveness by clinical studies (Tefertiller et al., 2011, Zhang et al., 2013, Miller et al., 2016).

One of the few commercialized active ankle orthoses is Anklebot (Roy et al., 2009). Anklebot allows the normal range of motion (ROM) of a human ankle in all 3 degrees of freedom (DOF) of the foot relative to the shank and driven by two actuators. With two parallel brushless DC motors, Anklebot can create power and torque in plantarflexion-dorsiflexion and inversion-eversion as shown in Figure 2-6. An advantageous feature of Anklebot is that underactuation allows installation of the device without having the need to precisely align the device with the patient's joint axes. Two of the mounted DC motors are capable of producing a continuous stall torque of 0.5 Nm. The Anklebot is provided with two sets of sensors: rotary encoders and linear incremental encoders. Rotary encoders are directly used to control the motors whereas linear encoders are used as feedback to the controller.

Figure 2-6. The Anklebot active orthoses robotic device (Roy et al., 2009).

Bharadwaj et al. (2005) developed an interesting ankle rehabilitation device design by forming a tripod from two linkages attached to the foot and ankle joint, as shown in Figure 2-7. It has the capability to transmit torque and power efficiently, especially for plantarflexion and dorsiflexion. The design has been developed as an ankle rehabilitation robotic gait trainer structurally based on a tripod mechanism. The design included two pneumatic actuated linkages attached to the left and right of the foot. The device's ROM in plantarflexion and dorsiflexion is 22.8° and 22.9° respectively and ROM in inversion and eversion is 3° and 5°. This ROM is capable of carrying out a natural gait. The actuators act as pneumatic muscles with an internal spring and force can be applied in both directions.

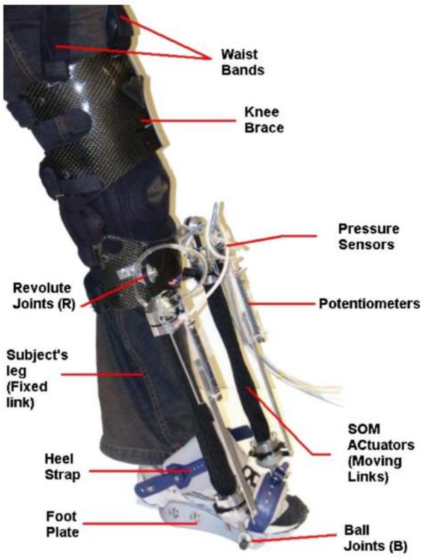


Figure 2-7. An ankle stroke rehabilitation robotic device for gait training using tripod mechanism. (Bharadwaj et al., 2005)

The functions of exoskeletons vary but may include: increasing the torque for the ankle to meet the required "push-off" force, increasing or contributing to walking speeds, improving the swing phase mechanics, and reducing the metabolic cost of walking or during the gait training session. An active leg exoskeleton (ALEX) was developed by Banala et al. (2008) for robot-assisted gait training as shown in Figure 2-8. ALEX's hip joint and sagittal joint are actuated in the sagittal plane with linear actuators. These motors have built-in encoders that are used to compute joint angles during gait. The other DOFs are held by springs. The motors are capable of generating 50 Nm peak torques at the joints of knee and hip. However, the torques generated at those joints also simulate the forces applied at the ankle. All the parameters for the controller of ALEX can be changed during rehabilitation to achieve the most suitable force distribution for the patient. During gait training, forces, the motion of joints, and gait as a whole is observed to find the best controller template. As training progresses and the patient's walking gait gets better, assistance is decreased and the difficulty is gradually increased closer towards the healthy subject's gait.

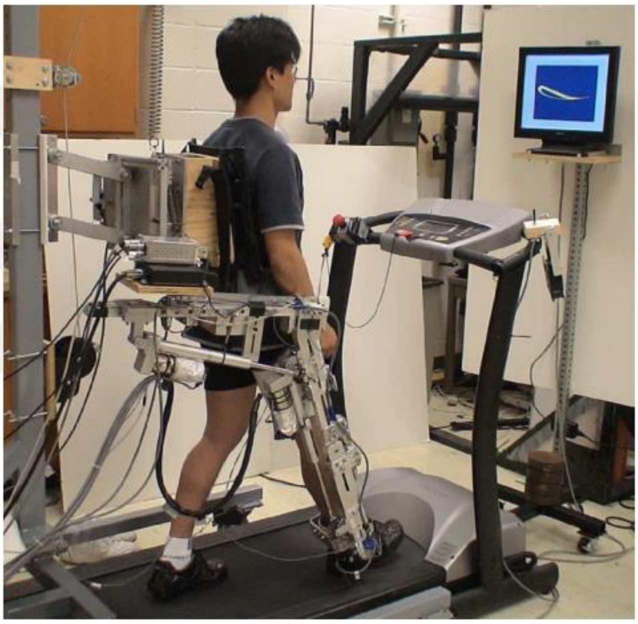


Figure 2-8. Gait training robotic device with active leg exoskeleton (ALEX) (Banala et al., 2008)

Ren et al. (2016) have developed a wearable ankle rehabilitation robotic device by focusing on dorsiflexion and plantarflexion. Their rehabilitation design was developed for patients that had stroke. Since the first few months are critical in stroke recovery, it is required to start rehabilitation as early as possible, and mobilize the ankle for the healthiest recovery. Their developed device performed both passive and active movement training in acute stroke. Isometric torque generation mode was used under real-time feedback to guide patients in motor learning. The mechanical design consists of a force sensor, a leg brace, a foot holder, and a rotary actuator assembly that is powered by a 120W brushless motor with a planetary gearhead and bevel gear set with a 3:1 ratio as shown in Figure 2-9. Interactive games were developed for patients to include the motor activity. With repeated feedback-facilitated training, acute patients slowly regained motor control ability. Intensive daily mobility training increased patients' dorsiflexor from 0.7 ± 1.2 Nm to 2.8 ± 3.5Nm after training.

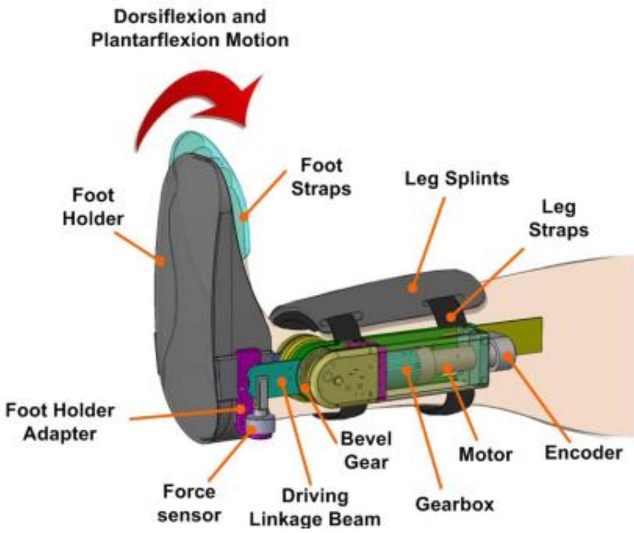


Figure 2-9. Mechanical design of the wearable ankle rehabilitation robotic device for in-Bed acute stroke rehabilitation. (Ren et al., 2016)

It is possible to achieve lightweight exoskeleton robots with different design approaches. One such example was presented by Awad et al. (Awad et al., 2015). The design of a soft wearable robot (exosuit) consisted of garment-like, functional textile anchors and cable-based transmission is shown in Figure 2-10. Exoskeletons in most cases serve to support their own weight and provide high amounts of assistance to the patients. However, for patients who retain their ability to walk but need to restore more normal gait function, a rehabilitation robot does not require to be too powerful. These create opportunities and challenges in design for achieving soft wearable robotic devices that can help individuals with gait impairment in their gait training. The presented exosuit is capable of producing gait-restorative joint torques with a cable-based mechanism and is not restrictive to natural walking dynamics. The exosuit has approximated peak plantarflexion of 0.15 Nm/kg which is %12 of the average peak paretic ankle moment.

Figure 2-10. A soft wearable robot design with cable-based mechanical power transmission. Plantarflexion (PF) actuation and dorsiflexion (DF) actuation is shown. (Awad et al., 2015).

Cable-based mechanisms permit the location of actuators away from the orthoses joints, which especially provides an advantage for ankle rehabilitation devices. For a normal gait, it is crucial to keep additional weights in joints to a minimum. Additional weights at joints will hinder the motion, require more power/torque in addition to complex control algorithms. Some other advantageous features of cable-based actuation are providing high transmission forces, low friction forces with the design capability of implementing speed reductions, and establishing complex coupling between the joint torques and the motor torque. These given features of cable-based actuation opened up pathways to many different designs in ankle rehabilitation robots that utilize steel wires and Bowden cables (Thalman & Artemiadis, 2020).

An ankle exoskeleton was developed by Orekhov et al. (2021) that utilizes cable-based actuation. The developed ankle exoskeleton design aims to help CP patients by applying assistive torque and power to the ankle joints for decreasing patients' metabolic cost of push-off power required in walking. The actuator is located in a waist assembly and power is transmitted with Bowden cables. The whole exoskeleton, as shown in Figure 2-10, weighs approximately 2.5 kg and has motors that are capable of providing peak torque of up to 30 Nm. Conner et al. (2021) have tested the ankle exoskeleton assistance device on children with CP in six-minute walk tests in order to test performance with and without the device. As result, they have observed that powered ankle assistance does in fact significantly increases six-minute walk test performance with participants having an average acclimation rate of 19.6 m.

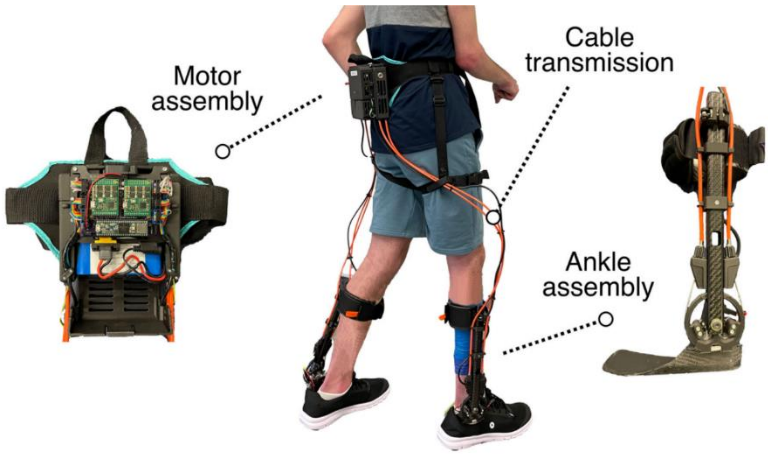


Figure 2-11. An ankle exoskeleton for CP patients. (Conner et al., 2021)

3. MATERIAL AND METHOD

In this thesis, three different conceptual designs for ankle rehabilitation robot for cerebral palsy patients are studied and the final proposed concept design is given in detail. The target patient group is adolescents with CP aged 12-18 and with a weight under 60 kg.

Each concept design is actuated by the same type of motor i.e. brushless direct current (BLDC) motor. Mechanical brushes often create weak points in a motor. Since robotics applications may include peak stress movements, motors need to be light, efficient, and long-lasting. Motor brush housing assemblies are complex and prone to failure from heavy shock loads. Large impacts may cause spring-loaded brushes to bounce, causing arcing. Brushed motors have limited operating speed since brushes arc when transitioning between commutor bars. Brushes will wear quickly and require frequent replacement as they are mostly made from soft carbon alloys. The friction from the sliding contact and the electrical resistance from carbon brushes increase the heat generated during operation, causing lower motor efficiency. These given disadvantages can be eliminated with a BLDC motor. Servo motors have also been considered as they are highly functional, and have better speed, position, and torque control, however, servo motors are much more expensive. Since an economic rehabilitation device is desired, BLDC motor was selected as actuators.

Hydraulics and pneumatics have also been considered for actuation. Both use fluid power and, compared to DC motors, are capable of supplying higher amounts of force. A pneumatics actuation system can produce more force with a lower weight system compared to a DC motor. However, hydraulics and pneumatics both require additional equipment such as pumps or compressors , pressurized tanks, and high pressure lines to properly work. These types of equipment can be arranged in a stationary facility in a rehabilitation session. On the other hand, it is desirable for rehabilitation robotic devices to be mobile and easily transportable. In addition, controlling the speed, torque, and position of a limb actuated by a hydraulics or pneumatics system is more complex and expensive. It generally requires additional pieces of equipment, such as pressure regulating valves, proportional valves, directional control valves, measurement units specific to the application, etc. that are both expensive and increasing the overall weight.

Ankle axes of rotation are another factor for ankle rehabilitation robotic devices. Placement for the rotational link of the robot should be located in the ankles axes of rotation as shown in Figure 3-1. This location of rotational axes naturally changes from patient to patient. A small amount of error in concentric axes are acceptable but too much error will cause strain during motions, especially in plantarflexion and dorsiflexion.

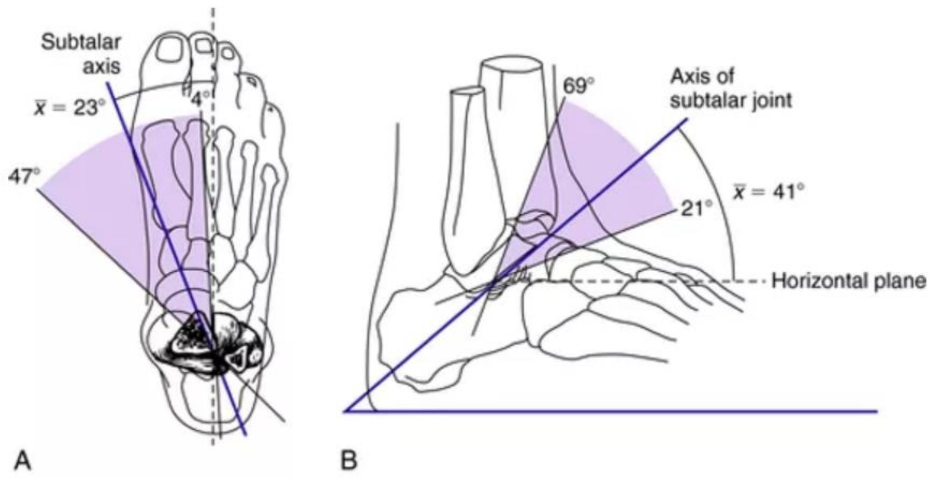


Figure 3-1. Subtular axes in transverse plane in left (A) and in horizontal plane in right (B) (https://www.ankleinfo.net/biomechanics (accessed by May, 2022))

The conceptual designs have been created in a solid modeling package namely SolidWorks. The stress analysis of the selected final design has been performed in SolidWorks as well.

The software framework for the robotic application has been developed through Robot Operating System (ROS). ROS is an open-source framework to build and reuse code for robotics applications. Works and programs done in ROS can be shared in open-source communities, making it easier to access robotic libraries that can be helpful in building robotic software. ROS uses Python and C++ programming languages. In this study, the Python programming language will be used with ROS. The advantage of using ROS is that it provides services such as hardware abstraction, message-passing between processes, package management, and low-level device control.

3.1. Conceptual Design – I

The idea behind the first design is to make a simple ankle rehabilitation robot that can achieve complex motion of the ankle in 3 DOF. The design consists of three linkages and three joints and a footplate as shown in Figure 3-2. The axes of rotations for the joints are placed collinear with the axes of rotations for the ankle as explained in the previous section. Each joint is actuated by a BLDC motor.

The design is to be used in stationary configuration, where the patient is sitting or lying down, and the device will repeat needed exercises for muscle therapy. The ankle motions of dorsiflexion-plantarflexion, inversion-eversion and abduction-adduction movements can be achieved. The leg will be tightly secured to the foot plate with means of cuffs. Then, any motion that is exerted on the plate by the robotic links are also exerted on the ankle. This simple three degrees of freedom rehabilitation design can reproduce any motion that is taught to the device by tracing encoder data of motors.

Figure 3-2. Schematic representation of conceptual design-I.

3.2. Conceptual Design - II

The second conceptual design utilizes the tripod mechanism for motion. This design is a wearable ankle rehabilitation device as shown in Figure 3-3.

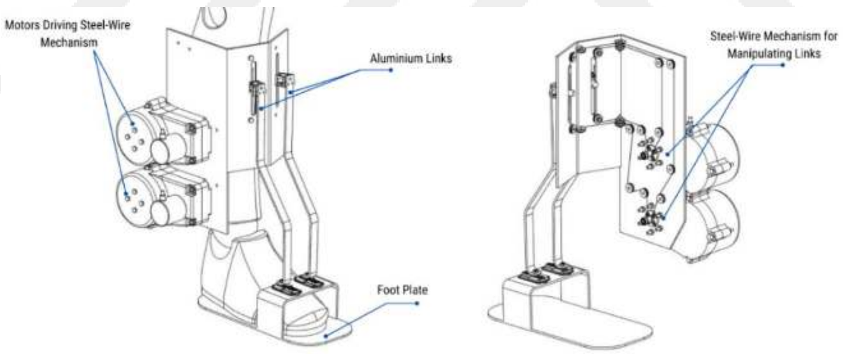


Figure 3-3. Schematic representation of conceptual design-II

The tripod mechanism is achieved with two aluminum links that are connected to the foot plate. These two links are positioned in a vertical slot and are manipulated with steel wires. Steel wires are shown in Figure 3-3 and are connected to the motor capstan. Capstan is a pulley like componet used for winding cable and powered by motor to manipulate the steel wires. The upper part of the design is tightened to the leg and the lower part is tightened to the foot.

The upper part consists of two motors that drive two aluminum links separately. The actuator assembly consist a brushless DC motor, six gears and an encoder. Motors are BLDC motors with 5500 rpm and are controlled with respect to the feedback from encoders that are connected to the opposite side of the capstan that drives steel wire. The gears selected are made of 7075-T6 aluminum and are hard and lightweight. Between the capstan and the motors, three gear reduction is added as shown in Figure 3-4. Their reductions are:

- 18 to 92, reduction ratio of 5.11,
- 20 to 80, reduction ratio of 4,
- 20 to 92, reduction ratio of 4.6.

The total gear reduction ratio is 94.024.

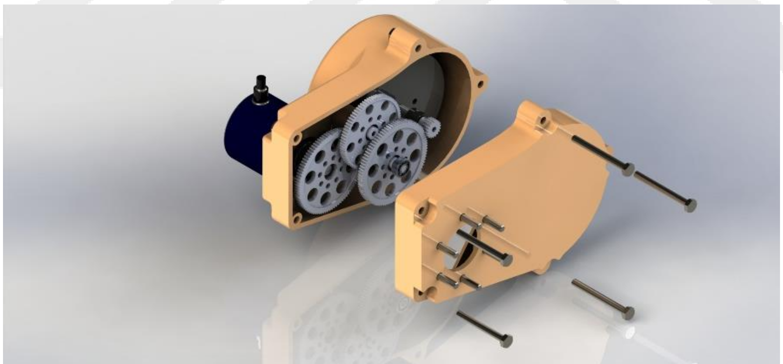


Figure 3-4. Actuator assembly for conceptual design – II.

The first conceptual design has 3 DOF and is able to achieve a more complex motion compared to the second design. However, with the second design, a conceptual design that can be used with gait training was developed. It should be noted that a wearable rehabilitation robot is more efficient for improving gait. The advantage of using a wearable ankle rehabilitation robot is that patients can be involved in walking exercises, giving a more natural rehabilitation process. This improves the patient's motor activity, strengthening muscles, and neurological control (Aisen et al., 2011).

A problem with the tripod mechanism is the obstruction of ROM unless designed carefully. If aluminum links are connected on both sides with fixed joints, a portion of ROM will be blocked. To increase the ROM of the design, aluminum link connections are made as pivot joints on the upper side and ball joints on the footplate side, as shown in Figures 3-5 and 3-6. Ball joints in the footplate are also positioned in a slot.

The aluminum links are driven by steel wires and the total displacement of the aluminum links in the vertical slot is derived from encoder data of how much the capstan has rotated and the radius of the capstan that steel wires are coiled.

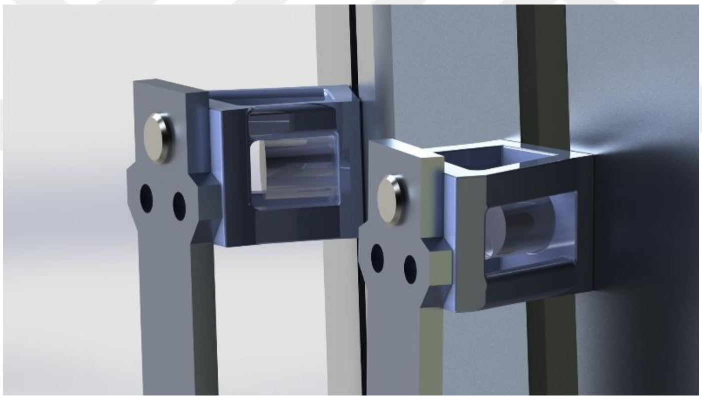


Figure 3-5. Pivot joint of aluminum links in the upper side.

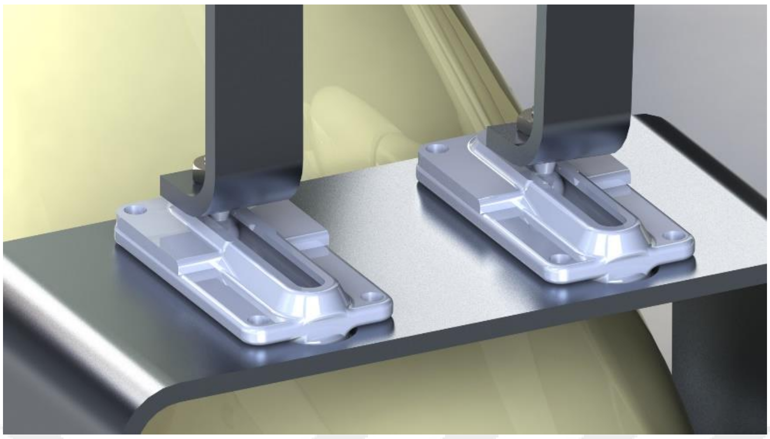


Figure 3-6. Ball joint of aluminum links in the footplate side.

Footplate displacement is dependent on the aluminum link however vertical displacement of aluminum is not directly translated to the footplate displacement. The footplate is bound to rotate on the ankle's rotational axes, and since aluminum links are connected to the footplate with ball joints that are inside a slot, vertical displacement of aluminum link will develop rotation in sagittal and/or frontal axes. To find how much the footplate is rotated depending on the vertical displacement of the aluminum link, the equation (1) given below is derived. The representation of dimensions is given in Figure 3-7.

$$\alpha = tan^{-1}\left(\frac{C_1}{C_2}\right) + tan^{-1}\left(\frac{\sqrt{C_1^2 + C_2^2 - 1}}{-1}\right) \quad (1)$$

$$C_1 = \frac{x}{L} \quad (2)$$

$$C_2 = \frac{(L + y)}{L} \quad (3)$$

Where;

- α, for the angle of rotation in the sagittal plane

- x, for horizontal distance from ankle's rotational axes to flat configuration slot connection position
- y, for the displacement of slot in the vertical axis.
- L, for vertical distance from ankle's rotational axes to flat configuration slot connection position

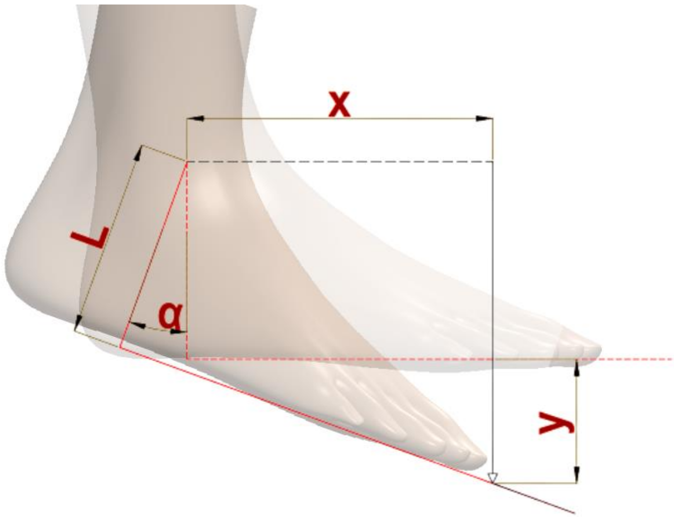


Figure 3-7. The representation of dimensions for equation (1).

This design is wearable and can be used with gait training. As a result, the effectiveness of rehabilitation is increased during walking. A rehabilitation robot for gait training aims to improve walking gait and reach a more natural gait. The main flaw of the conceptual design – II is that it creates too much weight on the ankle and knee. Too much weight at the end effector of a kinematic chain impairs dynamic control as well as the gait pattern. The actuation assembly including motors, gears, and encoders with aluminum plates and links creates a heavy additional weight. Since the additional weight on the ankle and knee disturbs the natural gait pattern, this concept design is not evaluated further and a new concept design is looked for.

3.3. Conceptual Design – III

To overcome difficulties regarding weight and range of motion, a new design is developed. Advantages, disadvantages, and problems inherent in the previous two designs are studied further in detail to reach a final new design. This design is a simple, lightweight wearable ankle rehabilitation robot that does not interfere with natural walking as shown in Figure 3-8.

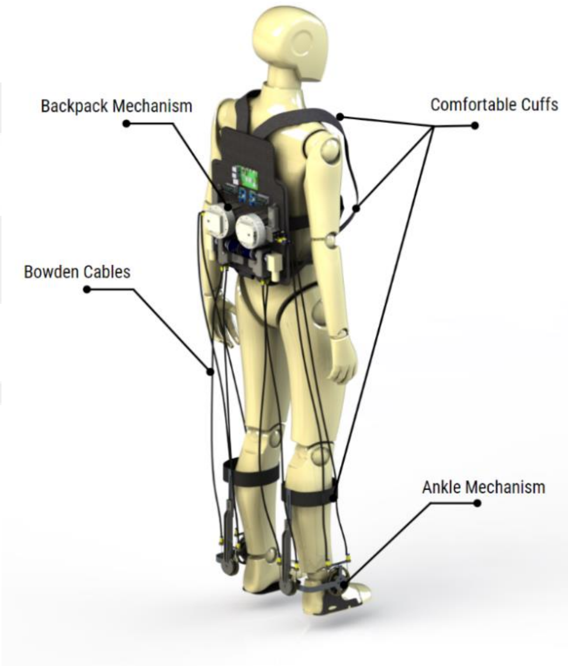


Figure 3-8. Schematic representation of conceptual design-III

The design has two degrees of freedom, the sagittal plane (plantarflexion and dorsiflexion) and the frontal plane (inversion and eversion). The reason why the sagittal plane and frontal plane were selected for two degrees of freedom is that these two planes create the main forces during walking as shown in Table 1-2 that was given in Introduction section. The mean of work done over one stride for sagittal motion is the greatest portion during walking, followed by the frontal plane. The push-off forces are generated in the sagittal plane. The frontal plane, on the other hand, helps with the positioning of the foot for supination and pronation, which will help to create a more natural walking gait. Thus, two degrees of freedom are chosen for the ankle rehabilitation robot for increasing gait performance.

The design has two main assemblies, an upper assembly which is the backpack mechanism, and a lower assembly which is the ankle mechanism, as shown in Figure 3-8. Backpack mechanism includes motors for sagittal plane and frontal plane, capstan, pulleys for steel wires, Bowden cable connections, inboard computer, microcontrollers, and required sensors. The motors of sagittal plane are equipped with encoders and are capable of controlling position, speed, and torque. The motors of frontal plane are not equipped with encoders and an encoder is added to the mechanism for controlling rotation in the frontal plane. The backpack mechanism will also include cuffs to be easily wearable and will be tightened to the body. All parts in the backpack mechanism are boarded to a foam-cored carbon fiber panel for high stiffness and minimum possible weight.

This ankle rehabilitation robot will be used in collaboration with the body-weight support system that is available at Çukurova Robotics Lab (www.cukurovarobotics.com) as given in Figure 3-9. The body-weight support treadmill system makes it possible to support the patients' weight in the amount that is needed. It is possible to support 100% of the weight, which would recreate a weightless walking configuration or decrease the amount of support so that the patient would have to try walking under the specified load. This system combined with our robotic device makes it possible to work on different weight support, torque support configuration, and walking speeds. Depending on patients' capability and level of needed support, different difficulties of gait training can be performed. This is important because some patients may not be able to reach desired gait speed without an unsupported configuration. Therefore, the designed system will especially help following a gradual training program for patients who can't generate enough force for push-off at earlier stages of the rehabilitation program.

Figure 3-9. Body weight support system in Çukurova Robotics Lab.

The ankle rehabilitation robot should be able to support the mean ankle moment of the patients and should be able to support patients up to 60 kg. In previous gait experiments and studies, it was observed that for the sagittal plane motions, the peak ankle moment is around 1.5 Nm/kg, and the mean ankle moment is 0.6 Nm/kg (Crowell et al., 2002), (Lerner et al., 2017). Hence, the selection of the motor should be able to support 36 Nm of torque.

During walking, the ankle's angular velocity rises to 8,34 rpm in the sagittal plane at the peak moment (Crowell et al., 2002). For the ankle rehabilitation robot, an actuator angular velocity of 40 rpm will meet the requirements for a gait pattern with an extra safety margin.

By considering the angular speed and the torque, the selected motor is RMD X8 from MyActuator for the sagittal plane (https://www.myactuator.com/product-page/rmd-x8-1-9 (accessed by June, 2022)). This BLDC motor provides 9 Nm torque at an angular speed of 180 rpm. With 4.44 reduction ratio, a torque of 40 Nm at an angular speed of 40 rpm is achieved where peak torque is 93 Nm. This motor is equipped with an encoder for rotational control and has also an inbuilt speed and torque controller.

The motor selected for the frontal plane motion is much smaller compared to the sagittal plane motion as the torque requirement is much smaller compared to the sagittal plane. A DC motor that can provide 2.1 Nm of torque at 40 rpm will be used. This DC motor doesn't have an encoder built-in, hence an encoder will be also installed. The difference in the capstan and ankle pulley radius will create a gear reduction ratio of 3, increasing torque to 6.3 Nm and decreasing angular speed to 13.34 rpm. This DC motor will help with pronation and supination of the ankle during heel touch and push-off (Donatelli, 1985).

The torque for the ankle mechanism is provided by steel wires connected to the motor in the backpack mechanism. The motion of the steel wires cause friction. The capstan connected to the motor at the backpack mechanism, the ankle pulleys, and other additional pulleys with ball bearing will move together with steel wires. Thus, the frictional losses are neglected at these contacts. However, steel wire moves through Bowden cable between the backpack mechanism and ankle pulley. The Bowden cable is stationary and there will be frictional losses due to the relative motion between the Bowden cable and the steel wires, that will decrease transmitted torque. The frictional behavior of the Bowden cable is similar to the friction of a cable routed around a stationary cylinder (Schiele et al., 2006), which is given in equation (4).

$$T_1 = T_2 e^{\mu\theta} \tag{4}$$

The T_1 represents the larger tension force compared to T_2, μ is the coefficient of friction between steel wire and Bowden cable whereas θ is the angle of contact at a routing inside the Bowden cable. The Bowden cable will have several small bends for routing throughout the path as it can be seen in Figure 3-8. The angles of these small bends are summed up to obtain the total θ for the friction calculation along the transmission.

Since the coefficient of friction depends on the used materials, it is 0.04 for stainless steel to PTFE contact as in our design (Chen et al.,2014). The friction will reduce the transmitted torque value to 35.6 Nm in the ankle pulley. This torque value meets the requirements to support mean ankle moments for patients up to 60 kg.

The motors are placed in an assembly that one can wear as a backpack as shown in Figure 3-10. This design will not interfere with the natural gait, will not cause any additional weight that would increase the cost of energy in the ankle, and will not increase difficulty in walking. Motors will rotate the capstan where the steel wires are fixed to it. The rotation of motors will pull the steel wires from one side and feed them to the other side. This backpack assembly will additionally include several pulleys and connection parts to transfer torque from the motor with low friction force to the ankle pulleys using steel wires.

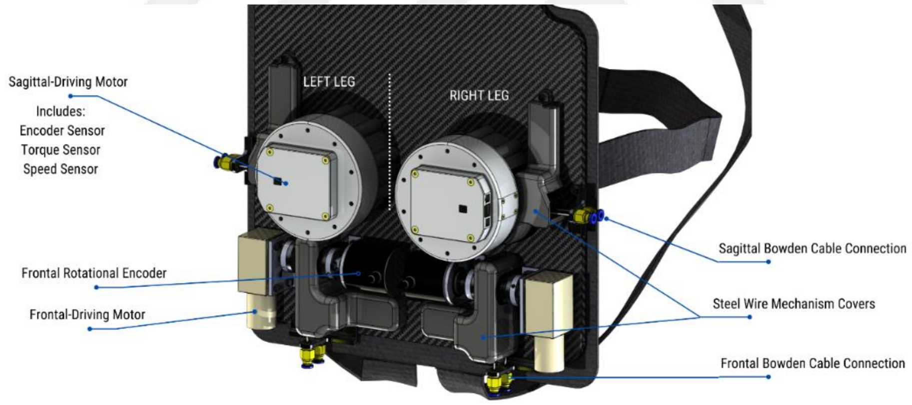


Figure 3-10. Backpack mechanism representing motors.

In the ankle mechanism, the steel wires around the pulley will be pulled to create the rotational motion. The ankle mechanism is shown in Figure 3-11. There are two pulleys, one for the frontal axis and one for the sagittal axis. The amount of pulling and feeding of the steel wires will provide the necessary torque. The direction of rotation will be controlled by the rotation of the motor. The Bowden cable between the ankle mechanism and backpack mechanism will create a path for steel wires and protect the user from any risk. mechanism will create a path for steel wires and protect the user from any risk.

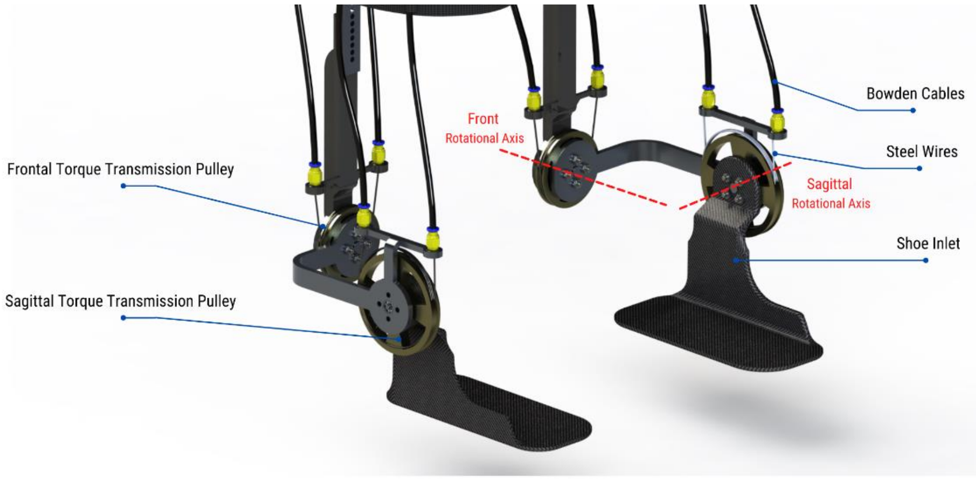


Figure 3-11. Ankle mechanism.

The capstan and mechanism for the sagittal plane inside the assembly are shown in Figure 3-12. The sagittal motor is directly connected to the capstan to transmit rotation and torque. The sagittal motor is fixed to the backplate with the help of a 3D printed part. The steel wire winding and path for the right leg are shown in Figure 3-13, the winding for the left leg is mirrored version of the right leg.

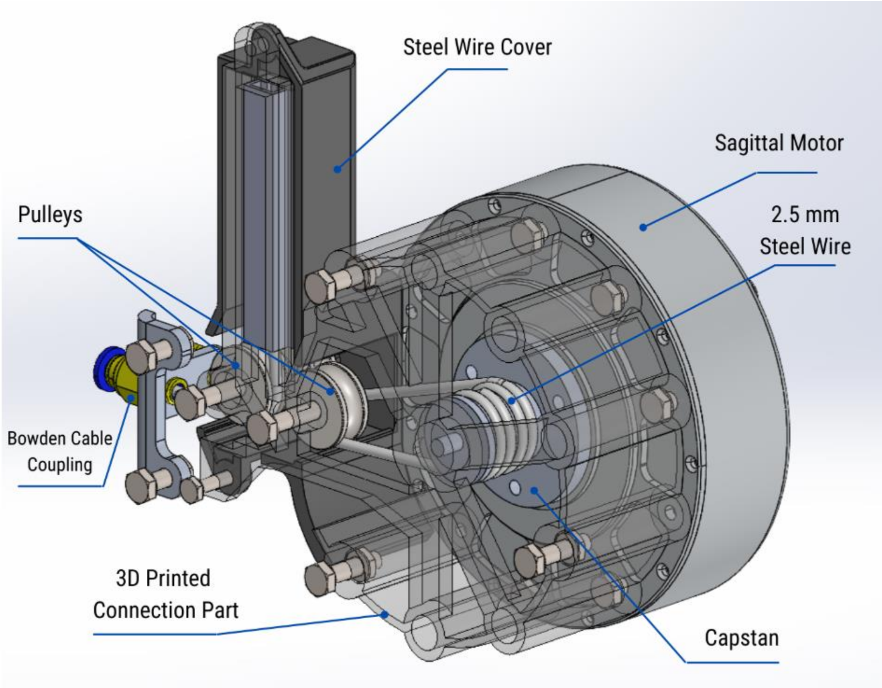


Figure 3-12. The steel wire, sagittal motor, and capstan mechanism.

Material selection is an important design parameter. The ankle rehabilitation robot needs to be durable and hard enough to carry loading stress while being lightweight for the comfort of the user. The material selection for the assembly plate of the backpack mechanism is foam-cored carbon fiber panel due to its high strength and low weight. The steel wire coverings and connection parts for motors in the backpack mechanism, as shown in Figure 3-14, is produced with a 3D printer using ABS for higher strength (Lay et al., 2019).

Shoe inlets, as shown in Figure 3-11, carry the load and requires to be strong. Two different choices were considered for shoe inlets: aluminum and carbon fiber. A comfortable shape gives much more comfort to the user and carbon fiber manufacturing methods permit the ability to shape the comfortable fit. Carbon fiber has high strength and also is lighter than aluminum. Thus, for shoe inlets, the carbon fiber material is selected.

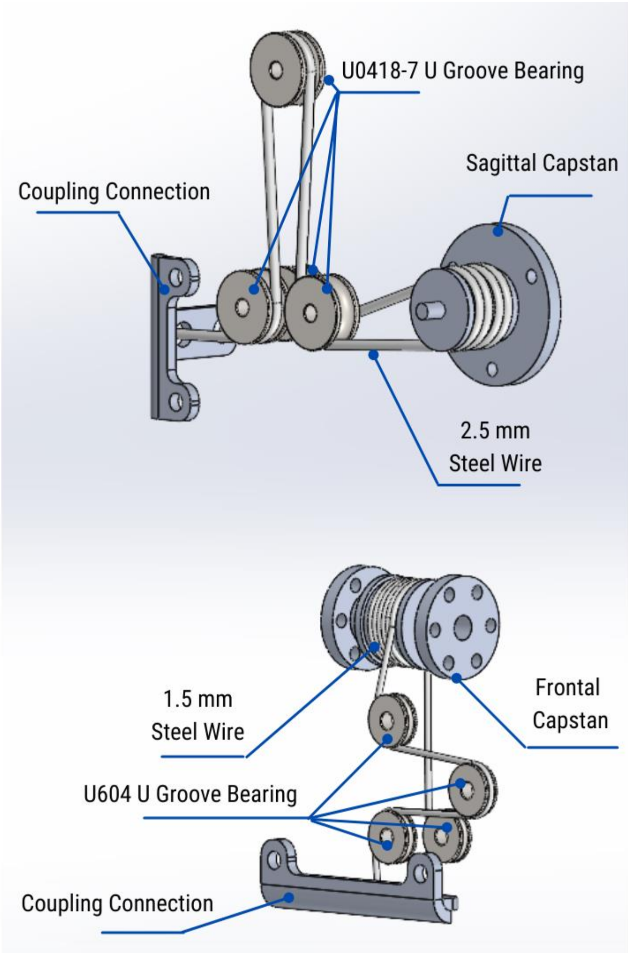


Figure 3-13. The steel wire winding at backpack mechanism, right leg.

The component connecting the frontal pulley and sagittal pulley is selected as aluminum since production methods for this part are more suitable. First, the part will be cut out of an aluminum plate in a laser cutting machine and then bend in two locations.

The whole ankle rehabilitation robot with backpack mechanism and ankle mechanism weights around 3300 gr.

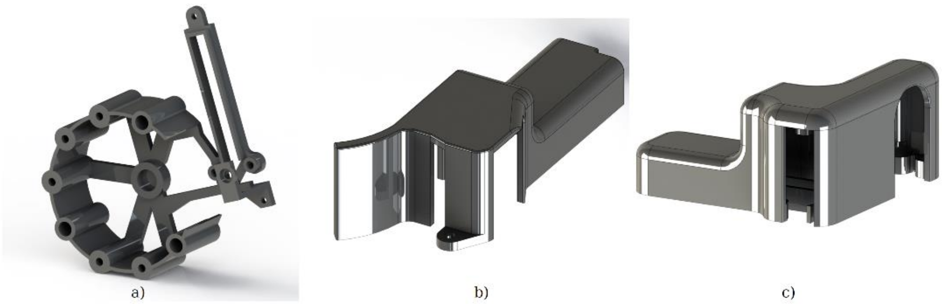


Figure 3-14. The main parts of the backpack mechanism that will be printed. a) Sagittal motor connection, b) Sagittal wire covering, c) Frontal wire covering.

The strength of steel wire has to be considered as it can break under high loads. For this reason, the load, tensile strength, and elasticity are to be considered to make sure there won't be any plastic deformation. The elastic limit of stainless-steel wire is approximately 60% of its breaking strength and for galvanized wires, it is approximately 50% (https://www.bergencable.com/cable-101/ (accessed by May, 2015)). With the radius of the pulley in the ankle being 40 mm, the force transmitted by steel wire reaches to 889 N at the peak torque of 35.6 Nm with frictions. Initially selected 1.5 mm wire is not strong enough to withstand this stress with a safety factor in steel wires of 5 with respect to the minimum yield strength. As a result, the working load of 1.5 mm wire drops to 29 kg. However, since our steel wire faces a tension of 889 N which is around 90 kg, this diameter for the steel wire will be not sufficient. Hence, steel wire with a diameter of 2.5 mm is selected. The properties of the wire are given for 2.5 mm diameter steel in Table 3-1. For the design's configuration, this steel wire will have a safety factor of 5.3 and is satisfactory.

Table 3-1. Properties of 2.5 mm diameter steel wire (https://www.tecni.uk/2.5mm-7x7-Galvanised-Steel-Cable-MBL-438kgs-503.000.025 (accessed by May, 2015))

Construction	7x7
Tensile	1960 N/mm2
Core	WSC (Wire Stand Core)
Material	Steel Galvanized 1960N/mm2
Standard	BS EN 12385-4:2002+A1:2008 Steel wire ropes
Minimum Breaking Load	485 kg (4,75 kN)
Working Load Limit	97 kg (0.95 kN)
Factor of Safety	5
Weight kg per 100m	2.4kg

Systems electrical power is supplied from a voltage regulator with the capability to supply at 20A. Driving motors for sagittal plane require 48V and driving motors for frontal plane require 12V. Battery will not be used as the rehabilitation will be done on a treadmill and the required electrical power will be supplied from the system. The reason is that the initial aim of the proposed design is to help and contribute to walking rehabilitation treatments instead of developing an assistive mobile walking device.

For control of the motors and data collection of sensors, the microcontroller Arduino Nano is used. Arduino Nano microcontroller has the same amount of digital input and output (IO), speed of CPU, and memory as Arduino Uno, however, the dimensions of Arduino Nano are much smaller and compact.

The frontal DC motors can be directly controlled through Arduino Nano with an additional motor driver and the selected DC motor driver is L298N. The sagittal BLDC motors use Controller Area Network (CAN bus) communication. CAN bus is allowing microcontrollers and devices to communicate with each other without a host computer. It is a message-based protocol. Arduino Nano requires a

CAN bus communication module to connect with motors and the selected communication module is MCP 2515.

Since the frontal DC motors don't have in-built encoders, an encoder is installed. The selected encoder is a single-phase 600 pulses rotary encoder with a maximum angular speed of 5000 rpm. The encoder is connected to Arduino Nano. In addition, two force resistive sensors are installed on the underside of the shoe inlets, one on the heel area and one on the toes area. The force sensor is utilized to capture the heel strike and toe-off. This improves the analytical capabilities of the system. Torque and rotation time graphs include gait-cycle with help of force sensors.

The general equipment list and costs are given in Table 3-2. Total cost of the ankle rehabilitation device is $1815.

The connection of motors and sensors to the Arduino Nano is shown in Figure 3-15. Each Arduino Nano controls motors and sensors responsible with one leg. Thus, there are two Arduino Nano in the system.

As a result, the Arduino Nano microcontroller is responsible for sending control signal to the BLDC motor, for gathering torque and rotational information from BLDC motor through CAN bus, for reading encoder data, for sending control signal to DC motor through DC motor driver, and finally for gathering information from the force sensors. However, computational processes are not performed in the microcontroller. All the gathered data are sent to an inboard computer. The selected inboard computer is Raspberry Pi. Arduino Nano has limited memory and the amount of data to be controlled in this configuration is too much for Arduino Nano to control. With its low weight and higher computational power compared to Arduino, Raspberry Pi processes gathered data to produce control signals.

Table 3-2. General equipment list and cost

Description	Material or Id	Amount	Unit	Cost per Unit	Cost
Sagittal Motors	RMD X8	2	pieces	$460	$920
Frontal Motors	DC Motor	2	pieces	$9	$18
Frontal Encoders		2	pieces	$13	$27
Base for Backpack	Foam Cored Carbon Fiber	1	piece	$55	$55
General 3D Print	ABS	2	kg	$12	$24
Aluminum Parts	5083 and 7075	1	kg	$16	$16
Manufacturing Cost		1	piece	$270	$270
Stel wire		20	meter	$0,5	$10
Bowden Cable		6	meter	$8	$49
Raspberry Pi 4 8GB		1	piece	$97	$97
Power Supply with voltage regulator		1	piece	$213	$213
Arduino Nano		2	pieces	$12	$24
MCP2515 CANBUS-SPI Module		2	pieces	$5	$10
DC Motor		2	pieces	$9	$18
Rotary Encoder		2	pieces	$15	$30
Force Sensor		2	pieces	$13	$27
Current Sensor ACS712		2	pieces	$3	$6

All electrical equipment used in the system are given in Table 3-3 and connections are shown in Figure 3-15.

Table 3-3. List of electrical equipment used in the system

Description	Amount
Raspberry Pi	1
Power Supply with voltage regulator	1
Arduino Nano	2
RMD x8 BLDC Motor	2
MCP2515 CANBUS-SPI Module	2
DC Motor	2
Rotary Encoder	2
Force Sensor	2
Current Sensor ACS712	2

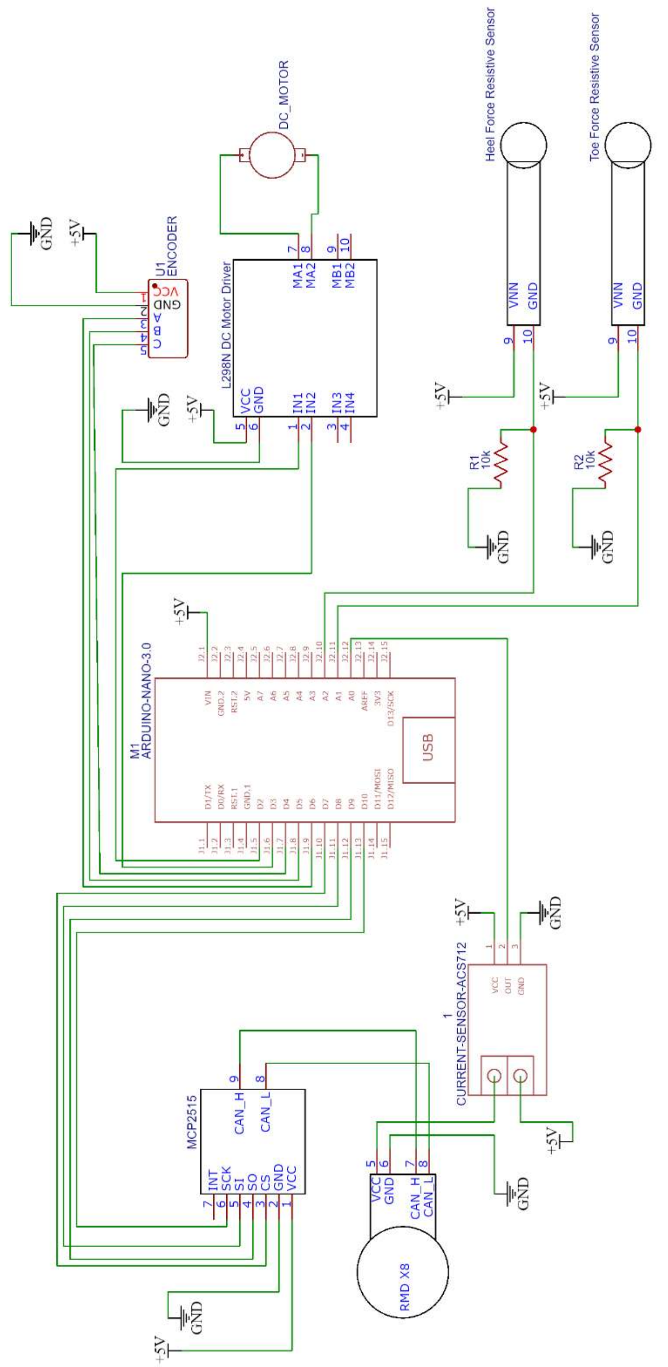


Figure 3-15. The connections of sensors and motors to Arduino Nano

CAN bus connection for the BLDC motor has its own control protocol and commands for communication. This control protocol is described in the documentation of the RMD x8 BLDC motor (https://www.myactuator.com/_files/ugd/cab28a_12650a87d0714b5b9df4aa1e4bafca4a.pdf (accessed by June, 2022)). The protocol works on sending a message to a motor and reading the reply from the motor. The protocol uses identifiers of 0x140+ ID (1~32) to differentiate commands. The frame format is DATA and the data length code is 8 bytes.

There are 52 different single motor commands and a reply for each of these commands. While giving all commands here is nonpractical, to give insight into the protocol, the position control command is given in Table 3-4, and a reply from the motor for the given command is given in Table 3-5.

Table 3-4. BLDC position closed loop control command.

Data field	Description	Data
DATA[0]	Command byte	0xA6
DATA[1]	Rotation direction byte	DATA[1]= spinDirection
DATA[2]	Speed limit low byte	DATA[2]=*(uint8_t*)(&maxSpeed)
DATA[3]	Speed limit high byte	DATA[3]=*((uint8_t*)(&maxSpeed)+1)
DATA[4]	Position control low byte	DATA[4]=*(uint8_t*)(&angleControl)
DATA[5]	Position control high byte	DATA[5]=*((uint8_t*)(&angleControl)+1)
DATA[6]	NULL	0x00
DATA[7]	NULL	0x00

Table 3-5. Motors drive response for position closed loop control command

Data field	Description	Data
DATA[0]	Command byte	0xA6
DATA[1]	Motor temperature	DATA[1] = *(uint8_t *)(&temperature)
DATA[2]	Torque current low byte	DATA[2] = *(uint8_t *)(&iq)
DATA[3]	Torque current high byte	DATA[3] = *((uint8_t *)(&iq)+1)
DATA[4]	Motor speed low byte	DATA[4] = *(uint8_t *)(&speed)
DATA[5]	Motor speed high byte	DATA[5] = *((uint8_t *)(&speed)+1)
DATA[6]	Encoder position low byte	DATA[6] = *(uint8_t *)(&encoder)
DATA[7]	Encoder position high byte	DATA[7] = *((uint8_t *)(&encoder)+1)

In Table 3-3, the data field represents the contents of the message. The "DATA[0]" is the command byte identifier and describes the contents of the message to the motor. The "DATA[1]" represents the direction of rotation that is desired from the motor and if the value of 0x00 means clockwise and 0x01 means counterclockwise. "uint8_t" means 8-bit unsinged type. The position command and the speed to reach the given position are given in "DATA[2]" to "DATA[5]". "DATA[6]" and DATA[7]" is given in NULL, which is because commands always include 8 data, and here position control command only needs 6 data and extra data space is filled with 0x00.

Table 3-4 shows the response of the motor and includes to which command it replies in "DATA[0]". The response message consists of motor temperature, torque value, motor speed value, and encoder position value.

To send this command and read the response in Arduino, the following codes are used:

```
buf[0] = 0xA6; //buf[0] describes the content of our
message
CAN.sendMsgBuf(0x141, 0, 8, buf); //this command sends our
message "buf" with its contents
CAN.readMsgBuf( & len, buf); //and this command will fill
in "buf" with response data.
```

In the given exemplary commands above, the contents of the message “buf” are not given in this text. The detailed Arduino code can be found in Appendix A-1 for further inspection.

The inboard computer needs to gather the motor and sensor data from Arduino for making decisions. This is achieved by using ROS packages. For the communication between ROS and Arduino, the rosserial ROS package gives universal asynchronous receiver/transmitter communication. The ROS uses nodes to define programs and these programs publish/subscribe to other nodes. The rosserial converts the Arduino board to a ROS node that can publish and subscribe to ROS messages.

The Arduino ROS node publishes data from the board to the onboard computer, Raspberry Pi, running the ROS while an Arduino ROS node subscriber can get instruction from the Raspberry Pi. The ros_lib Arduino library will be used on Arduino code to enable the Arduino board to communicate with the ROS.

The message system is what enables easy communication in the ROS. The ROS itself offers different types of messages with different contents however for the best suitable message to our ankle rehabilitation robot, a custom message is created. There are four motors in our system. In the ROS, a custom message named “motordata.msg” is created. This custom message includes data required from a motor, which are:

- Motor ID
- Rotation of Direction (0 for CW, 1 for CCW)
- Torque Value
- Speed Value
- Encoder Value

The version of the custom message in the ROS is shown below:

```
string motorid
int8 rotation
int64 torque
int64 speed
int64 encoder
```

In “string motorid”, the message carries the information of which motor’s data is given.

Another simple custom message for the system is “forcesensors.msg”, which includes two values, one for heel and one for toe:

```
int64 heel
int64 toe
```

The messages are used as contents of a topic. The topic is a system of the ROS to give nodes an option to publish or subscribe. The Arduino node “Arduino-CANBus” publishes topics that carry data from the motors. The ROS code “Motor_control_node” subscribes to these topics to gather motors data, make a decision and publish the motor command message in topics that Arduino subscribes to and uses to control motors. An example of this is shown in Table 3-6 for right leg sagittal and frontal motor data and command topics. A full communication schematic of the motor, sensors, Arduino, and ROS main control node is shown in Figure 3-16. In Figure 3-16, the topic names are given at the top of the lines and there are 12 different topics. It also can be noted that the communication to the sagittal motor goes both ways while communication to the frontal motor is one-sided and rotational data of the frontal motor comes from the encoder.

Table 3-6. A portion of motors data and motors command topics

Topic Name	Publisher	Subscriber
SRmotor_data	Arduino-CANBus	Motor_control_node
SRmotor_com	Motor_control_node	Arduino-CANBus
FRmotor_data	Arduino-CANBus	Motor_control_node
FRmotor_com	Motor_control_node	Arduino-CANBus

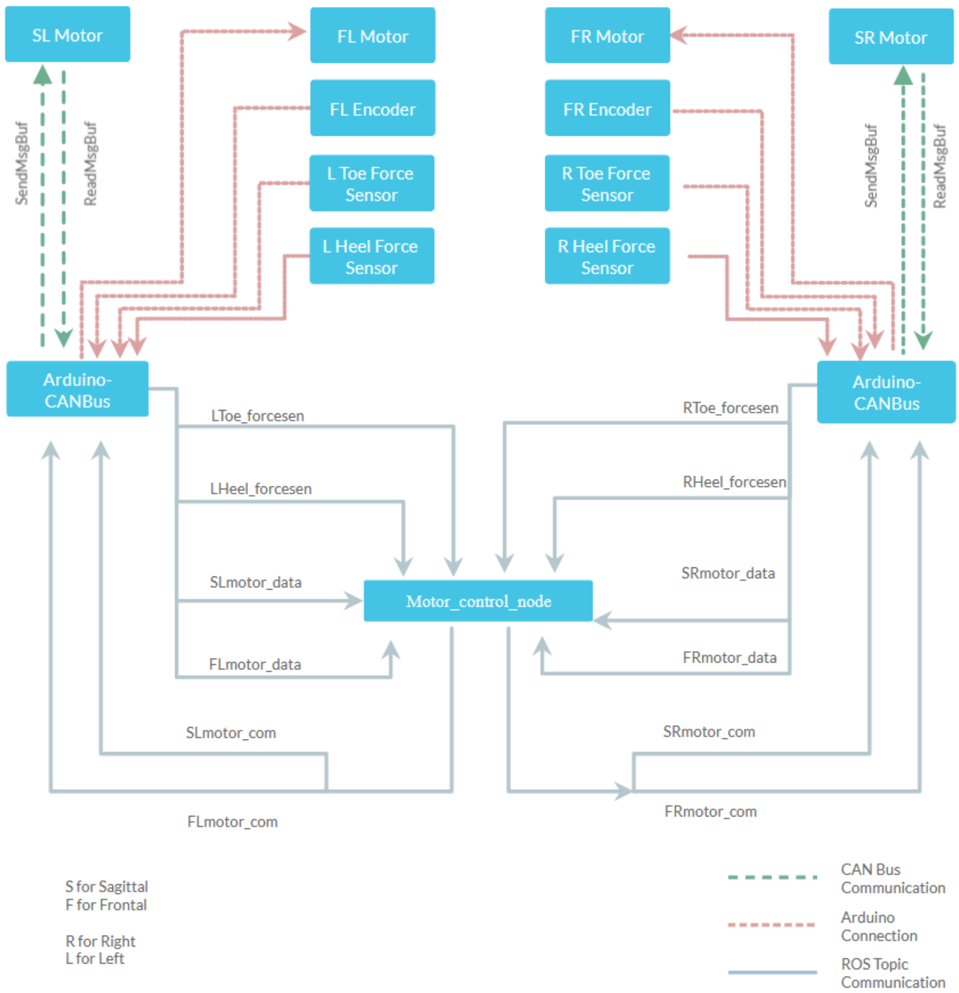


Figure 3-16. Schematic of communication for control.

To control the motors, data set for torque, speed, and position of the ankle is required. This data set for the gait is prepared from the graph given in the study of Crowel et al. (2002), which are shown in Figure 3-17 and Figure 3-18. The data set from these graphs are extracted by a plot digitizer program.

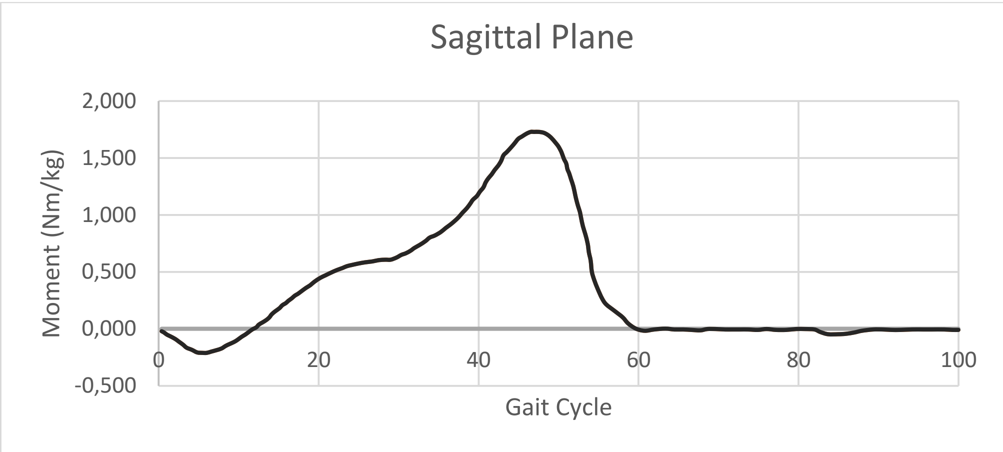


Figure 3-17. Ankle joint moment in the sagittal plane during one cycle of normal walking.

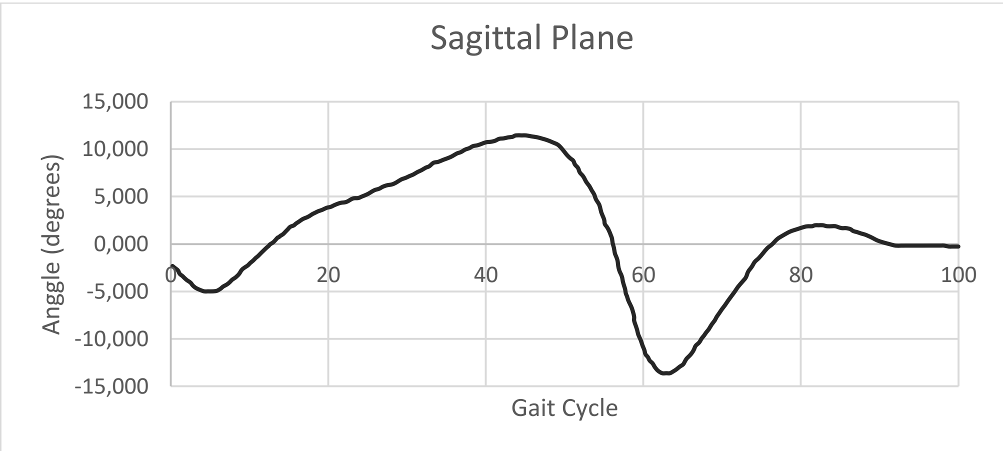


Figure 3-18. Ankle joint angles in the sagittal plane during one cycle of normal walking.

The Arduino is able to send and receive necessary data for controlling motors with the system that is built. For the gait control, torque and angle values are extracted from graphs. The ankle joint speed values are obtained from the angular position data values. In the control algorithm, these values are loaded as “NumPy” array, a Python library. In addition to these values, our configuration also includes body support ratio and motors assistive torque ratio. This increases the torque control ability to find the best assistance for the user that can increase the quality and effectiveness of rehabilitation. Different support ratios are compared in the following two figures. Figure 3-19 shows the torque on the ankle joint for the sagittal plane for a 60 kg patient with 20% body-weight support and 45% motor assistance torque during walking whereas Figure 3-20 shows for a 35 kg patient without weight support and 60% motor assistance torque. In moment graphs, summation of the assistive moment and patient’s work gives the required moment for the normal walking. These gathered torque, speed, and position values are given in the Appendix for inspection and/or reuse opportunities.

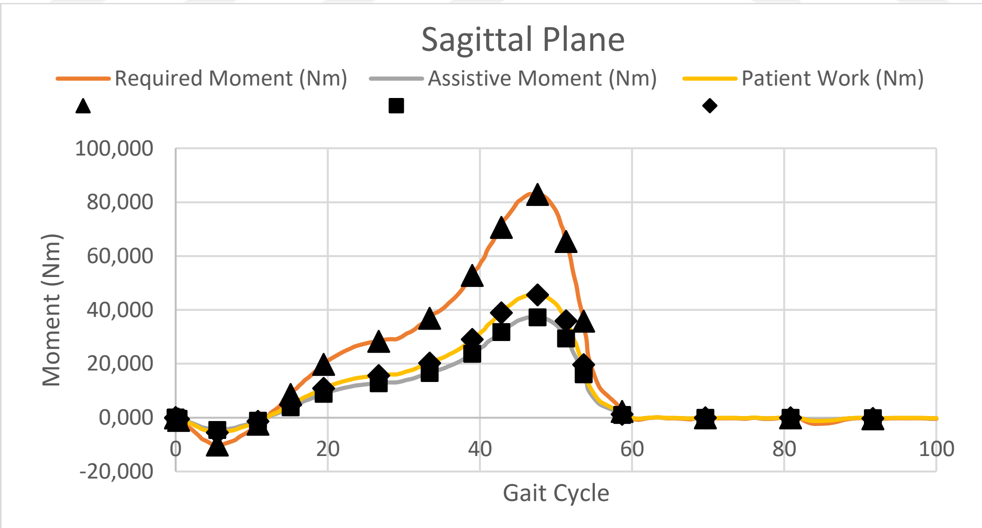


Figure 3-19. Moment values during walking for a 60 kg patient with 20% body-weight support and 45% motor assistance torque.

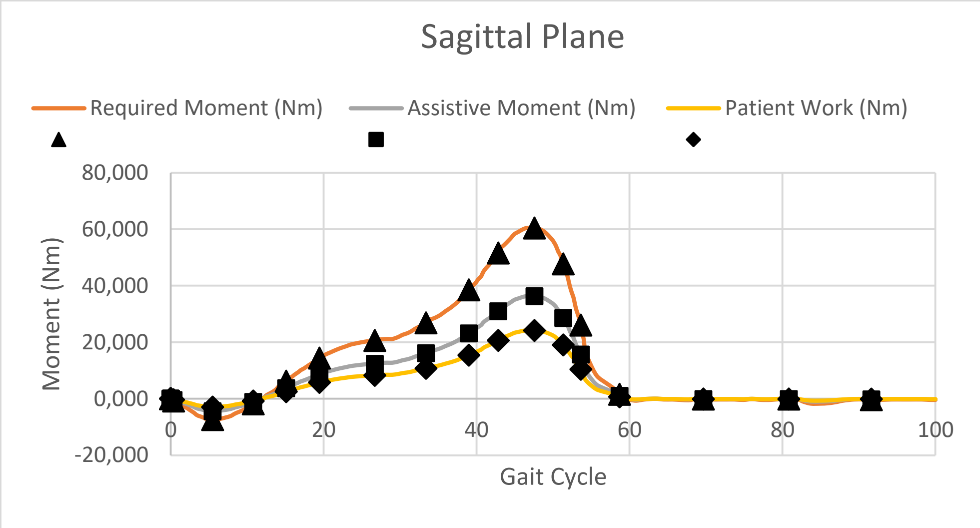


Figure 3-20. Moment values during walking for a 35 kg patient with no body-weight support and 60% motor assistance torque.

There are four motors, each responsible for two legs, hence to control two gaits. The timing of motor controls defines a more natural gait. For this reason, the timing diagram of a gait cycle in Figure 3-21 is inspected. This image shows heel strike and toe off with stance phase, swing phase, and double support timestamps. The image is given for the right leg stride, starting with the heel strike of right leg.

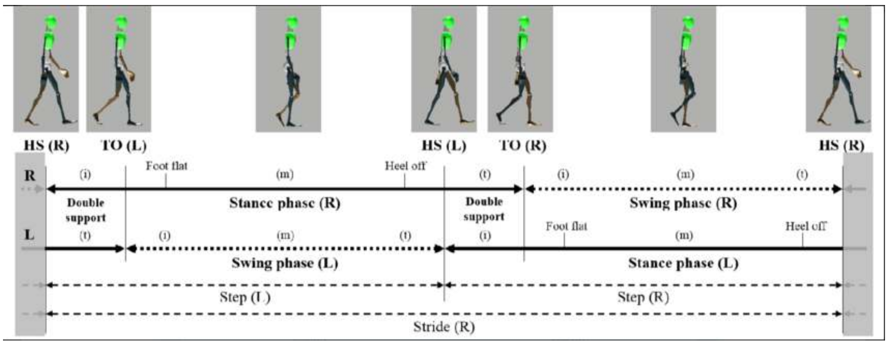


Figure 3-21. Time diagram of a gait cycle. 'HS' stands for heel strike, ‘TO’ is toe-off, ‘R’ is right, ‘L’ is left, (i) initial phase, (m) is mid-phase, and (t) is the terminal phase (Hwang et al., 2018).

Algorithm Case Inputs are given as in the following list:

- The Gait cycle for one side of the leg starts at the heel strike and continues until the next heel strike, where the process then repeats.
- During the gait cycle of one side, the other leg will start its cycle around the middle with a heel strike.
- Beginning of the cycle, at double support, force absorption will take part.
- Towards the end of the stance phase, torque will reach a peak and starts decreasing.
- As the other leg comes to the heel strike and double support starts, torque will fall to zero, and the swing phase starts. The end of the swing phase is a heel strike.

Thus, it is assumed as one side of the ankle is in the middle of its stride, the other leg starts its stride.

Two gaits are constructed in such a way that when one of them starts at %0 and the other starts at %50 of the gait cycle. By this way it will start with a double support position that turns into a stance on the first side that starts at %0 of the cycle and swing at the second side that starts at %50 of the cycle.

Stride time serves as an input to rearrange the gait cycle values in the data array. The program continuously creates required torque, speed and position signals for motors depending on time by comparing their location in the data array. This code is given in Appendix A-2 for inspection.

It should be noted that the system has an open control loop as it just sends the required signals to the motors based on the gait cycle. This control algorithm is strict during operation on following the gait cycle and rehabilitation is needed to be done continuously without interruption. In other words, this algorithm can't be used outside the rehabilitation where the user may need to stop, turn directions, walk in different speeds or climb stairs. This algorithm is to be used during the rehabilitation session. The different control algorithms such as PID control, adaptive control, fuzzy logic control can be applied to predict the motion of the patient and to control motors accordingly. With such improvement, this ankle robot can be used in daily life.

The final design parameter is safety. A powered rehabilitation device requires safety features to stop motors in any case of malfunction. The safety of patients must be all-time guaranteed and any risk parameter must be prevented. There are number of safety measures, one of them is to monitor the current at the motors and limit the maximum current in the microcontroller code. Similarly, responses of the BLDC motors are monitored constantly for current (representing torque) and motors will be stopped if values are over the maximum set limits. These set limits differ in rehabilitation as torque values of motors for a 20 kg patient and 60 kg patient are not the same. Furthermore, a current sensor is installed in the system that the motor’s current will pass through, which constantly makes sure that our output from the program is the same as the actual currents. If there is a difference in actual currents and current values sent by the program or, if the actual currents exceed limit set for the current then the motors will be shut down. The over-flexing are prevented and protected by monitoring the encoders' rotational values and shutting down motors if encoder values are not part of the gait cycle.

4. RESULTS AND DISCUSSION

4.1. Strength Analysis

The components in the ankle mechanism were analyzed using the FEA module of SolidWorks to ensure the safety of the system.

Materials properties available in SolidWorks were used for aluminum parts and 3D print ABS parts. However, for carbon fiber, mechanical properties were specified using a supplier's datasheet regarding their carbon fiber product T700.

Two different stride phases in a gait cycle were selected for the analysis. These were flat stance and heel strike. Peak torque value takes place in the flat stance and is important to analyze to ensure parts are able to withstand high torque values. A heel strike involves reaction force during walking and this force will pass through the foot inlet and needs to be analyzed to make sure the shoe inlet won't break under the force.

The images from the analyses are given with a deformation scale. The deformation scale is used to scale the deformed shape of the model to give a better idea of how the deformation takes place by exaggerating the deformation. Without the deformation scale, the deformation from analyses are not observable.

The safety of the components are checked with respect to the factor of the safety. By the definition of the factor of safety, the maximum stress points from the analyses are identified and then the yield strength of the components is divided by the maximum stress.

$$Factor\ of\ Safety = \frac{Yield\ Strength}{Maximum\ Stress} \tag{5}$$

4.1.1. Flat Stance

The most of the required power is greatly applied during the flat stance and peak torque occurs starting near the end of the flat stance. The peak torque sustains until the beginning of heel off and starts declining until toe-off.

For the strength analysis of the foot plate for the peak torque, the peak torque occurrence during the flat stance was taken into consideration. At this state of gait, the foot is fully in contact with the ground with a knee-to-ankle dorsiflexion angle of 11° degrees. The simulation was done both in full perpendicular weight reaction force to the footplate as well as with an angle of 11° degrees to guarantee the footplate's material safety.

Focusing on the peak torque and peak load occurrence, shoe inlet is brought to the flat position on the surface. The user's weight acting on top of the footplate, the peak torque, and steel wire pulling force is applied in pulley connections. Torque and steel wire pulling force are 35.6 Nm and 889 N in actual peak occurrence however these numbers are rounded for an extra safety factor. Analysis input values are given below:

- Weight: 60 kg
- Torque: 36 Nm
- Steel Wire Pulling Force: 900 N

Under the given input values, simulation gives the highest stress as 173 MPa where our tensile strength is 2550 MPa, and the footplate factor of safety is found as 14.7. During the production of this part with carbon fiber, the actual tensile strength may be less because of non-perfect production procedures. However, with the factor of safety being above 10, the footplate will be safe for unexpected distortions in production. The boundary conditions and loads for this analysis are given in Figure 4-1. The stress results on the component under the applied force and torque are shown in Figure 4-2 and Figure 4-3.

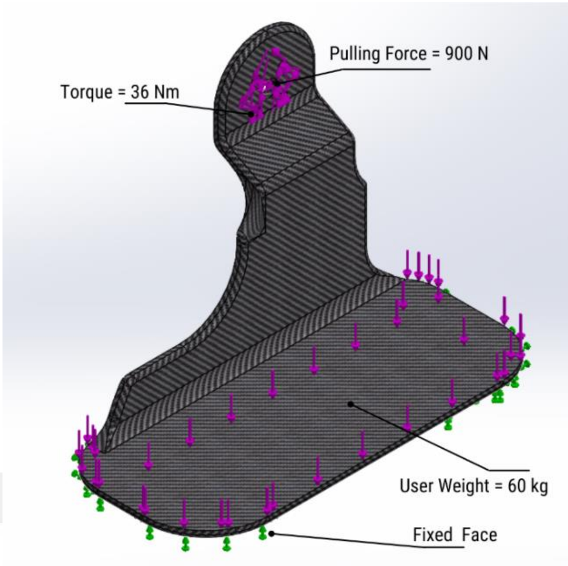


Figure 4-1. The boundary conditions and loads for shoe inlet strength analysis.

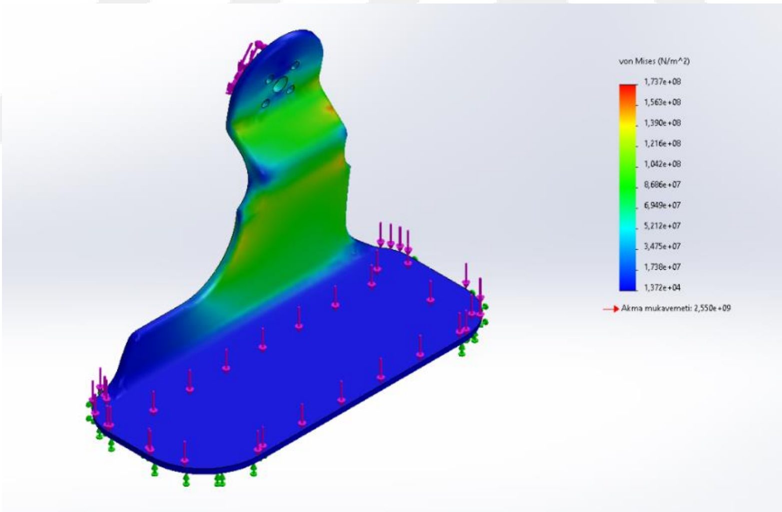


Figure 4-2. Shoe inlet's flat stance perpendicular strength analysis. The deformation scale is 8.43.

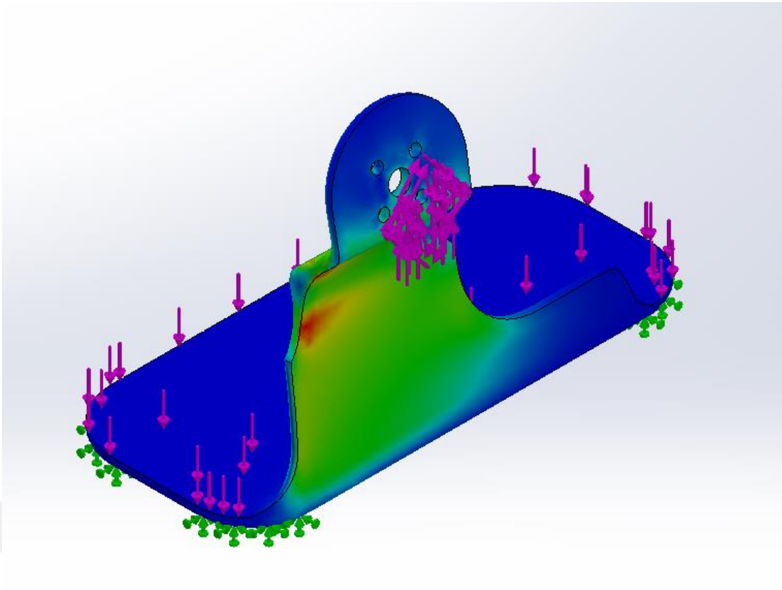


Figure 4-3. Shoe inlet's flat stance perpendicular strength analysis in opposite direction showing maximum stress point. The deformation scale is 8.43.

Dorsiflexion causes the weight of the user to act on the footplate at an angle of 11°. For the shoe inlet, the simulation is repeated with an angle modification. The results are similar to each other and have the same factor of safety. The results are shown in Figure 4-4.

The stress analysis of the shaft on the footplate supporting the pulley has been also performed. For the simulation of the shaft, the area which is connected to the footplate is fixed as it is obscured. 900N peak force is applied on the area where the pulley is connected. The factor of safety is represented as 1.67. In actual configuration, the area connected to the footplate would also absorb a portion of force as the footplate is deformed elastically around 2mm. Thus, the actual factor of safety will be slightly higher than 1.67. The boundary conditions and loads for the shaft analysis are given in Figure 4-5. Stress and deformation simulation images are given in Figure 4-6 and Figure 4-7 where the deformation scale is 87.839.

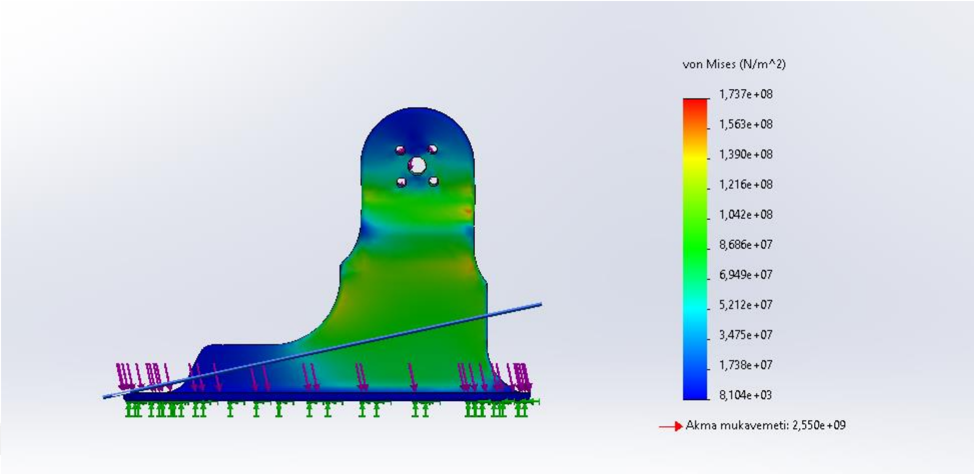


Figure 4-4. Shoe inlet stress analysis under acting weight angle of 11° and peak torque.

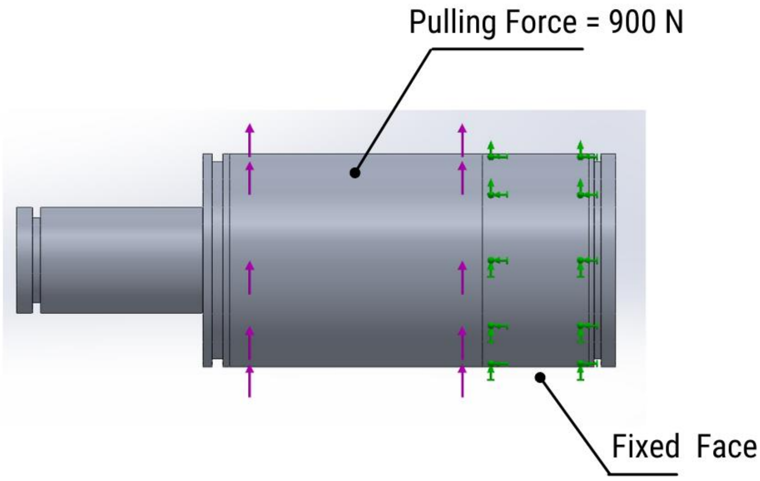


Figure 4-5. The boundary conditions and loads for the shaft.

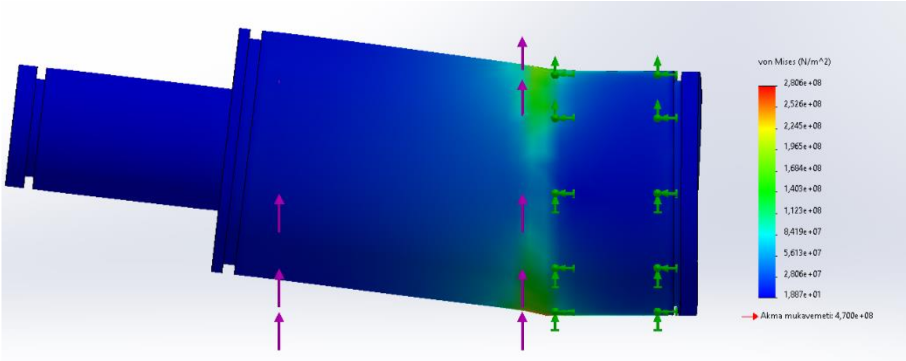


Figure 4-6. Shaft stress analysis under peak torque and pulling force. The deformation scale is 87.839

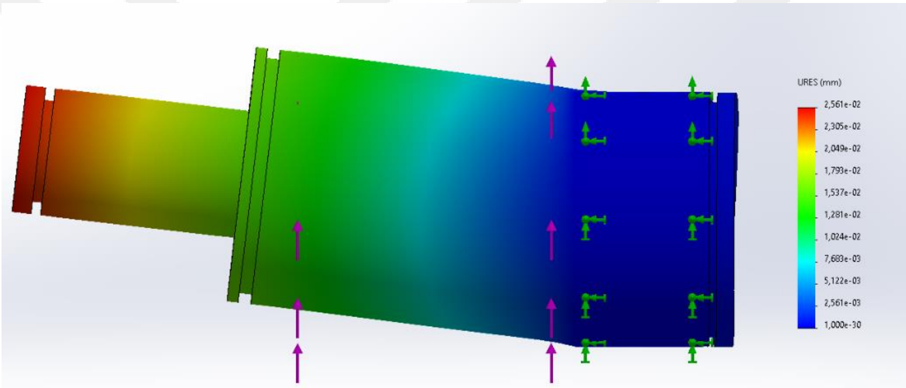


Figure 4-7. Shaft deformation analysis under peak torque and pulling force. The deformation scale is 87.839

The shaft was simulated with two different materials AL 5083 and AL 7075. However, with AL 5083 factor of safety is 1.01 and the desired factor of safety is a minimum of 1.4 for our robotic device. Therefore, the selected material for the shaft is AL 7075 which has higher strength. Even though it is a little bit more expensive, the shaft is around 5 g before machining and would not increase the cost too much.

The part named frontal translator, which transmits frontal torque to the footplate is connected to the far left of the shaft and is affected by shaft deformation. The frontal translator is simulated under the effect of this deformation. In addition, a torque of 2.1 Nm is applied caused by frontal motion. The boundary conditions and loads for the frontal translator analysis are given in Figure 4-8. The part is sheet metal thus sheet metal simulation is done with SolidWorks as shown in Figure 4-9. The frontal translator has a factor of safety of 20. This is more than required and it may be beneficial to make it thinner however part is selected as 5mm and lightweight as it is.

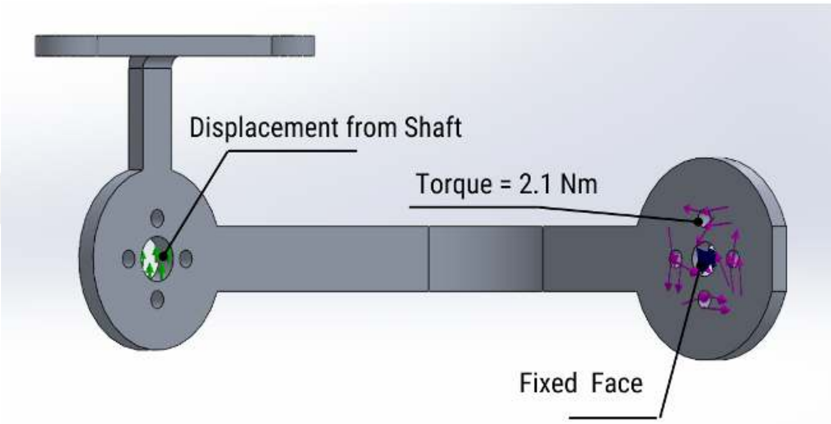


Figure 4-8. The boundary conditions and loads for the frontal translator.

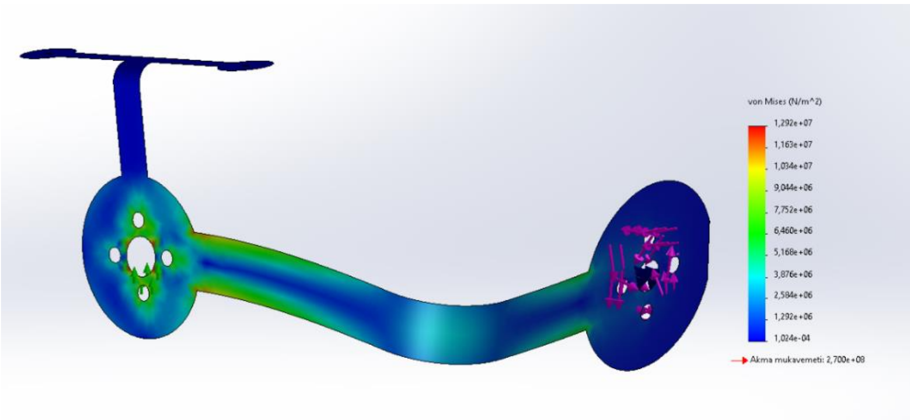


Figure 4-9. The frontal translator stress analysis under experienced deformation and torque of 2.1Nm. The deformation scale is 192.77.

The sagittal pulley is simulated under the torque of 36 Nm and 900 N pulling force applied to the steel wire. The sagittal pulley is safe against any deformation.

The boundary conditions and loads for the sagittal pulley analysis are given in Figure 4-10. The simulation image is shown in Figure 4-11 below and the deformation scale is 549.

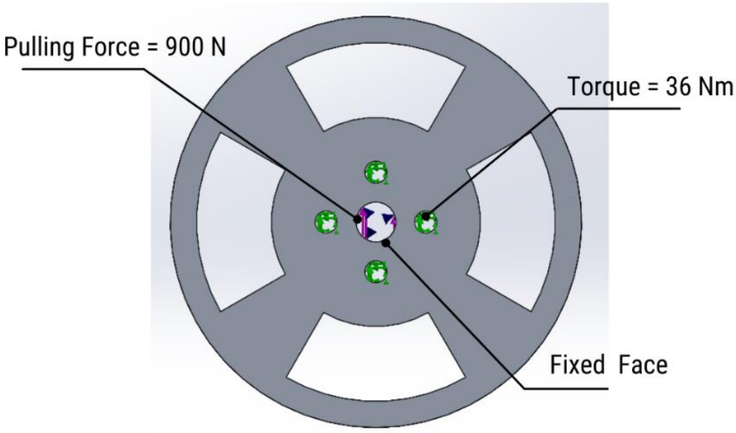


Figure 4-10. The boundary conditions and loads for the sagittal pulley.

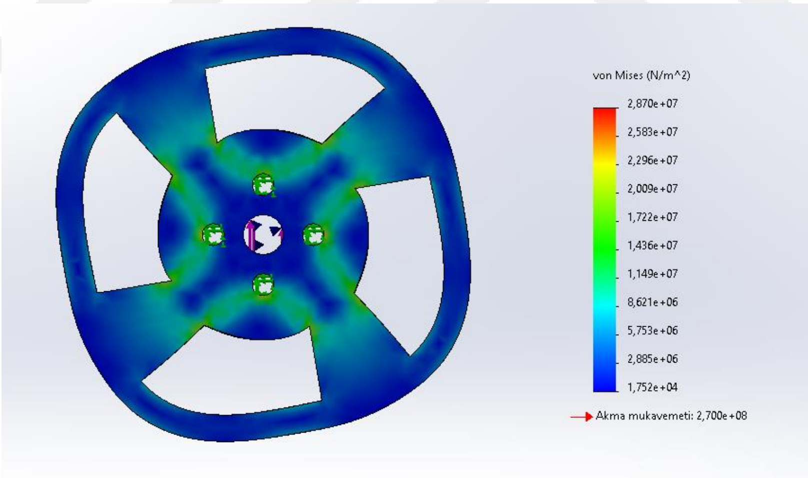


Figure 4-11. Sagittal pulley stress analysis. The deformation scale is 549.

The frontal pulley part is also simulated under the torque of 2.1 Nm and 52 N. The results showed the part is safe against any high deformation and fracture which was to be expected from the low torque.

The frontal pulley is supported with a 3D printed part, named frontal support. This part will support the steel wire's pulling force of 52 N on the frontal pulley. The initial analysis has shown that under 52 N this part experiences high deformation and is not suitable for safe usage. To fix this problem, thickness is increased and two rims are added to the backside of the frontal support in order to increase deformation resistance. Initial design and improved designs are shown in Figure 4-12 below. This solution decreased deformation from 14.4 mm to 1.82 mm as shown in Figure 4-13. The stress analysis result is shown in Figure 4-14.

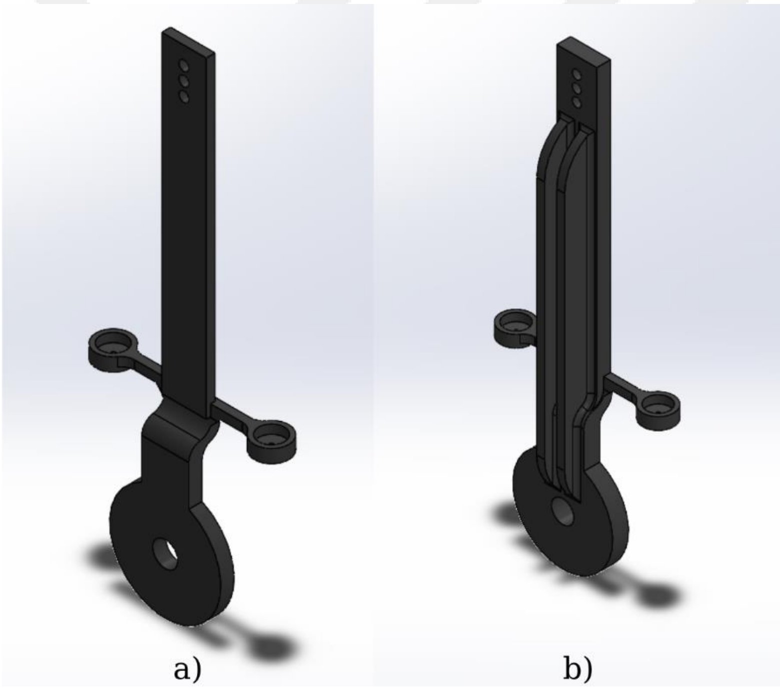


Figure 4-12. Design modification of frontal support. Initial design on the left (a), improved design on the right (b).

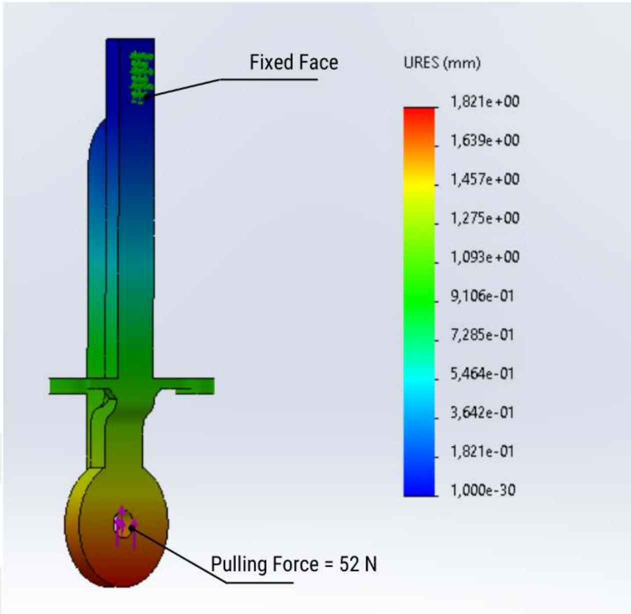


Figure 4-13. The frontal support deformation analysis.

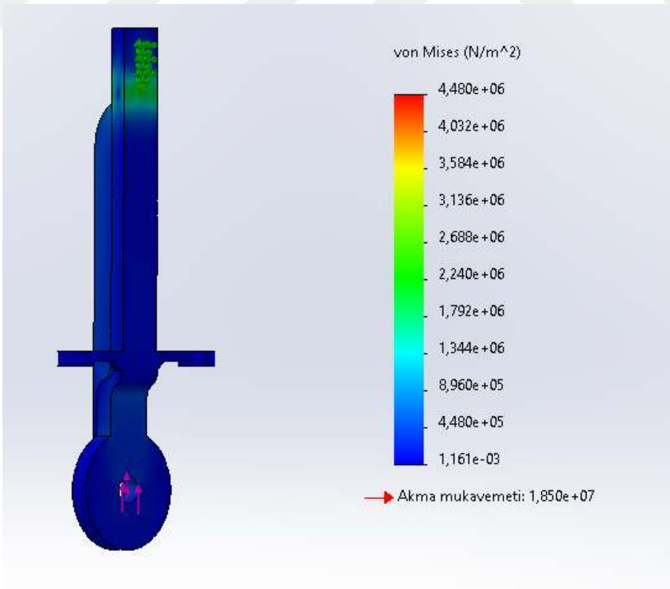


Figure 4-14. The frontal support strength analysis.

4.1.2. Heel Strike

The ground reaction force during walking is a variable of walking speed. A study by Keller et al. in 1996 showed the variation of vertical maximum thrust force for different walking speeds. The normal walking speed is approximately 1.5 m/s. For stress analysis of the designed part, the ground reaction force of vertical thrust maximum force (F_z) during heel strike is considered. This maximum vertical thrust is normalized to body weight (BW). For females, at a walking speed of 1.5 ms^{-2}, F_z is 1.15 BW. For males, at the same walking speed, F_z is 1.23 BW (Keller et al., 1996).

To stay on the safe side both for the strength of part and the rehabilitation process, walking speed is taken as 2.5 m/s. At a walking speed of 2.5 m/s, F_z is 1.73 for females and 1.62 for males. F_z is higher for females, taking this into consideration, the maximum ground reaction force for the design is found using equation (6).

$$F = F_z * BW * g \quad (6)$$

In this equation, F_z is the maximum vertical thrust force and is normalized to body weight taken as 1.73. BW is body weight and selected as 60 kg and g is gravity and taken as 9.80665 ms^{-2}. The maximum force, F, is found 1017.93 Nm. The shoe inlet is analyzed in SolidWorks under this heel strike force. This analysis is shown in Figure 4-15 and has a high safety factor, showing the part is secure for use in walking for patients up to 60 kg.

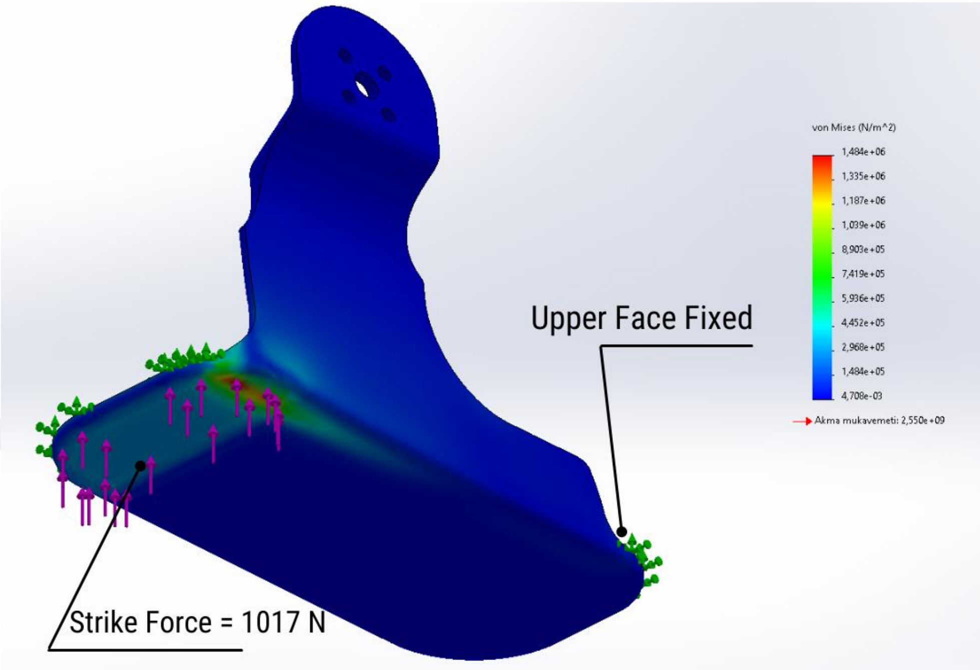


Figure 4-15. Shoe inlet strength analysis for heel strike.

All parts are simulated under peak torque and force. The shoe inlet is simulated additionally for the ground reaction force during heel strike. The design is safe and suitable for use.

5. CONCLUSION

This work presents design features of the ankle rehabilitation robot for improving gait performance for cerebral palsy patients. Robotic rehabilitation device designs are complex and require varying problems to solve. The current state-of-art for ankle rehabilitation robots are presented and details are given. The difficulties encountered during the design of a wearable rehabilitation robotic device are discussed. An iterative approach is followed for the design, expanding on previous concepts. The key design parameters are introduced and explained for the final proposed design.

The proposed ankle rehabilitation robot is wearable and does not obstruct normal gait. This device is intended for use during walking therapy to help people improve their walking ability. At average walking speed, it can support mean torque requirements for adolescents aged 12 to 18 who weigh 60 kg or less. For the best rehabilitative results, assistive torque can be set based on the patient's needs. The proposed design uses steel wire for force transmission which is becoming more popular due to the advantage of achieving complex mechanisms.

Parts used in the proposed ankle rehabilitation robot are simulated in FEA programs and shown that they are secure and ready for use. These stress analyses and simulations have been done for important stride phases with the highest torque or forces. However, this can be improved with a dynamic simulation as the center of mass during walking varies. This can give insight into force distribution, making it possible to have a better torque control algorithm.

For the control of motors, Arduino and ROS are used in collaboration. Additional encoder and force-resistive sensors are connected to Arduino. All information from Arduino is published in the ROS environment. The ROS node for decision-making gathers the required information and publishes the command values for motors. The ROS works on an onboard computer and the robot is able to make complex decisions.

This proposed design has not been manufactured due financial issues. For the proposed design, materials of components are selected and manufacturing methods are discussed for the production of the robot. The sagittal motors are the most expensive part of the design and cover around 50% of the total needed cost of manufacture. This proposed ankle rehabilitation robot costs around 29.862 ₺ ($1.815) which is expensive for the majority of the population. Advances in motor technology are continuing and motors with high torque are much cheaper compared to last decade. If the same trend in motor technology continues, powered rehabilitation robots can be much cheaper, increasing the availability of rehabilitation robots.

Steel wire selection is done according to the tension in the peak torque moment. The tension is dependent on the pulley's radius as the needed torque will be generated by force multiplied by radius. For the sagittal plane, the selected radius was 40 mm however radius selection was tested for 30 mm and 50mm as well. It is desirable to have a smaller radius for a more compact robot. But, smaller radius causes a higher force required to generate the same amount of torque. 30 mm radius would require much thicker steel wire while a 50 mm radius can be too big for use with some patients.

The codes for the control algorithm are developed according to the documentation given on the internet for motors and sensors. In real configuration, depending on sensors output, the gathered data may require filtering such as a low-pass filter or Kalman filter. The selection of filtering varies on the sensors' output and the desired outcome which is best done through testing. With proper funding and production of the robot, these control algorithms can be improved with filters according to the sensors.

REFERENCES

Aisen, M. L., Kerkovich, D., Mast, J., Mulroy, S., Wren, T. A., Kay, R. M., & Rethlefsen, S. A., 2011. Cerebral palsy: clinical care and neurological rehabilitation. The Lancet Neurology, 10 (9), 844-852.

Alamdari, A., & Krovi, V. N., 2017. A review of computational musculoskeletal analysis of human lower extremities. Human Modelling for Bio-Inspired Robotics, 37-73.

Alvarez-Perez, M. G., Garcia-Murillo, M. A., & Cervantes-Sánchez, J. J., 2020. Robot-assisted ankle rehabilitation: a review. Disability and Rehabilitation: Assistive Technology, 15 (4), 394-408.

Andersson, C., Grooten, W., Hellsten, M., Kaping, K., & Mattsson, E., 2003. Adults with cerebral palsy: walking ability after progressive strength training. Developmental medicine and child neurology, 45(4), 220-228.

Awad, L. N., Bae, J., O'donnell, K., De Rossi, S. M., Hendron, K., Sloot, L. H., ... & Walsh, C. J., 2017. A soft robotic exosuit improves walking in patients after stroke. Science translational medicine, 9(400), eaai9084.

Banala, S. K., Kim, S. H., Agrawal, S. K., & Scholz, J. P., 2008. Robot assisted gait training with active leg exoskeleton (ALEX). IEEE transactions on neural systems and rehabilitation engineering, 17(1), 2-8.

Bax, M., Goldstein, M., Rosenbaum, P., Leviton, A., Paneth, N., Dan, B., ... & Damiano, D., 2005. Proposed definition and classification of cerebral palsy, April 2005. Developmental medicine and child neurology, 47(8), 571-576.

Bharadwaj, K., Sugar, T. G., Koeneman, J. B., & Koeneman, E. J., 2005. Design of a robotic gait trainer using spring over muscle actuators for ankle stroke rehabilitation. Journal of biomechanical engineering, 127(6), 1009-1013

Chen, D., Yun, Y., & Deshpande, A. D., 2014. Experimental characterization of bowden cable friction. 2014 IEEE International Conference on Robotics and Automation (ICRA), 5927-5933.

Conner, B. C., Orekhov, G., & Lerner, Z. F., 2021. Ankle Exoskeleton Assistance Increases Six-Minute Walk Test Performance in Cerebral Palsy. IEEE Open Journal of Engineering in Medicine and Biology, 2, 320-323.

Crowell III, H. P., Boynton, A. C., & Mungiole, M., 2002. Exoskeleton power and torque requirements based on human biomechanics. Army Research Lab Aberdeen Proving Ground Md.

Damiano, D. L., 2009. Rehabilitative therapies in cerebral palsy: the good, the not as good, and the possible. Journal of child neurology, 24(9), 1200-1204.

Dollar, A. M., & Herr, H., 2007. Active orthoses for the lower-limbs: challenges and state of the art. IEEE 10th International Conference on Rehabilitation Robotics, 968-977.

Donatelli, R., 1985. Normal biomechanics of the foot and ankle. Journal of Orthopaedic & Sports Physical Therapy, 7(3), 91-95.

Eng, J. J., & Winter, D. A., 1995. Kinetic analysis of the lower limbs during walking: what information can be gained from a three-dimensional model?. *Journal of biomechanics*, *28*(6), 753-758.

Farris, D. J., & Sawicki, G. S., 2012. The mechanics and energetics of human walking and running: a joint level perspective. Journal of The Royal Society Interface, 9(66), 110-118.

Himmelmann, K., Hagberg, G., Beckung, E., Hagberg, B., & Uvebrant, P., 2005. The changing panorama of cerebral palsy in Sweden. IX. Prevalence and origin in the birth-year period 1995–1998. Acta paediatrica, 94(3), 287-294.

Hussain, S., Jamwal, P. K., & Ghayesh, M. H., 2017. State-of-the-art robotic devices for ankle rehabilitation: Mechanism and control review. Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine, 231 (12), 1224-1234.

Hwang, T. H., Reh, J., Effenberg, A. O., & Blume, H., 2018. Real-time gait analysis using a single head-worn inertial measurement unit. IEEE Transactions on Consumer Electronics, 64(2), 240-248.

Johnston, T. E., Moore, S. E., Quinn, L. T., & Smith, B. T., 2004. Energy cost of walking in children with cerebral palsy: relation to the Gross Motor Function Classification System. Developmental medicine and child neurology, 46(1), 34-38.

Keller, T. S., Weisberger, A. M., Ray, J. L., Hasan, S. S., Shiavi, R. G., & Spengler, D. M., 1996. Relationship between vertical ground reaction force and speed during walking, slow jogging, and running. Clinical biomechanics, 11(5), 253-259.

Lay, M., Thajudin, N. L. N., Hamid, Z. A. A., Rusli, A., Abdullah, M. K., & Shuib, R. K., 2019. Comparison of physical and mechanical properties of PLA, ABS and nylon 6 fabricated using fused deposition modeling and injection molding. Composites Part B: Engineering, 176, 107341.

Lerner, Z. F., Damiano, D. L., & Bulea, T. C., 2017. The effects of exoskeleton assisted knee extension on lower-extremity gait kinematics, kinetics, and muscle activity in children with cerebral palsy. Scientific reports, 7(1), 1-12.

Miller, L. E., Zimmermann, A. K., & Herbert, W. G., 2016. Clinical effectiveness and safety of powered exoskeleton-assisted walking in patients with spinal cord injury: systematic review with meta-analysis. Medical Devices: Evidence and Research, 9, 455-456.

Orekhov, G., Fang, Y., Cuddeback, C. F., & Lerner, Z. F., 2021. Usability and performance validation of an ultra-lightweight and versatile untethered robotic ankle exoskeleton. Journal of neuroengineering and rehabilitation, 18(1), 1-16.

Palisano, R., Rosenbaum, P., Walter, S., Russell, D., Wood, E., & Galuppi, B., 1997. Development and reliability of a system to classify gross motor function in children with cerebral palsy. Developmental medicine and child neurology, 39(4), 214–223.

Racu, C. M., & Doroftei, I., 2015. Ankle rehabilitation device with two degrees of freedom and compliant joint. IOP Conference Series: Materials Science and Engineering, 95 (1).

Ren, Y., Wu, Y. N., Yang, C. Y., Xu, T., Harvey, R. L., & Zhang, L. Q., 2016. Developing a wearable ankle rehabilitation robotic device for in-bed acute stroke rehabilitation. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 25(6), 589-596.

Ross, S. A., & Engsberg, J. R., 2007. Relationships between spasticity, strength, gait, and the GMFM-66 in persons with spastic diplegia cerebral palsy. Archives of physical medicine and rehabilitation, 88(9), 1114-1120.

Roy, A., Krebs, H. I., Williams, D. J., Bever, C. T., Forrester, L. W., Macko, R. M., & Hogan, N., 2009. Robot-aided neurorehabilitation: a novel robot for ankle rehabilitation. IEEE transactions on robotics, 25(3), 569-582.

Saglia, J. A., Tsagarakis, N. G., Dai, J. S., & Caldwell, D. G., 2009. A high-performance redundantly actuated parallel mechanism for ankle rehabilitation. The International Journal of Robotics Research, 28(9), 1216-1227.

Schiele, A., Letier, P., Van Der Linde, R., & Van Der Helm, F., 2006. Bowden cable actuator for force-feedback exoskeletons. 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 3599-3604.

Scholtes, V. A., Dallmeijer, A. J., Rameckers, E. A., Verschuren, O., Tempelaars, E., Hensen, M., & Becher, J. G., 2008. Lower limb strength training in children with cerebral palsy–a randomized controlled trial protocol for functional strength training based on progressive resistance exercise principles. BMC pediatrics, 8(1), 1-11.

Shi, D., Zhang, W., Zhang, W., & Ding, X., 2019. A review on lower limb rehabilitation exoskeleton robots. Chinese Journal of Mechanical Engineering, 32 (1), 1-11.

Sung, E., Slocum, A. H., Ma, R., Bean, J. F., & Culpepper, M. L., 2011. Design of an ankle rehabilitation device using compliant mechanisms. Journal of Medical Devices, 5(1).

Tefertiller, C., Pharo, B., Evans, N., & Winchester, P., 2011. Efficacy of rehabilitation robotics for walking training in neurological disorders: a review. Journal of Rehabilitation Research & Development, 48(4), 387-416.

Thalman, C., & Artemiadis, P., 2020. A review of soft wearable robots that provide active assistance: Trends, common actuation methods, fabrication, and applications. Wearable Technologies, 1.

Wang, S., Meijneke, C., & van der Kooij, H., 2013. Modeling, design, and optimization of Mindwalker series elastic joint. *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, 1-8.

Wu, Y. N., Hwang, M., Ren, Y., Gaebler-Spira, D., & Zhang, L. Q., 2011. Combined passive stretching and active movement rehabilitation of lower-limb impairments in children with cerebral palsy using a portable robot. Neurorehabilitation and neural repair, 25(4), 378-385.

Zhang, L. Q., Chung, S. G., Bai, Z., Xu, D., Van Rey, E. M., Rogers, M. W., ... & Roth, E. J., 2002. Intelligent stretching of ankle joints with contracture/spasticity. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 10(3), 149-157.

Zhang, M., Davies, T. C., & Xie, S., 2013. Effectiveness of robot-assisted therapy on ankle rehabilitation–a systematic review. Journal of neuroengineering and rehabilitation, 10(1), 1-16.

CURRICULUM VITAE

Metin Ünlü finished high school in 2015 and started university at Çukurova University in Adana. He studied English between 2015-2016 in the preparatory year and started mechanical engineering in 2016. He studied mechanical engineering and management for one semester in Poland, Politechnika Poznańska. He graduated in 2020 and started MSc studies in the same year. He is studying robotics in his department.

APPENDIX

A-1 Arduino Codes

```
// CAN-BUS Shield, send data
// More additions and controls for ROS and additional
Encoders by Curobotics & Metin Unlu
//


#include <mcp_can.h>
#include <SPI.h>
#include <ros.h>
#include <curobotics/motordata.h>
#include <curobotics/forcesensors.h>
#include <avr/interrupt.h>
#include <avr/sleep.h>

ros::NodeHandle node_handle;

// Topics for data are specified and named
curobotics::motordata SRmotor_com;
curobotics::motordata FRmotor_com;
curobotics::motordata SRmotor_data;
curobotics::motordata FRmotor_data;
curobotics::forcesensors Rforce_data;


ros::Publisher SRmotor_data_pub("SRmotor_data",
&SRmotor_data);
ros::Publisher FRmotor_data_pub("FRmotor_data",
&FRmotor_data);
ros::Publisher Rforce_data_pub("Rforce_data",
&Rforce_data);


//Frontal Motor Pins
const int A1A=3;
const int A1B=2;
//Setup for Frontal Encoder
#define outputA 6
#define outputB 7
int counter = 0;
int aState;
int aLastState;
```

```
//Force Sensors
int heelfsrAnalogPin = 2; // Heel FSR is connected to
analog 2
int heelfsrReading;      // the analog reading from the FSR
resistor divider
int toefsrAnalogPin = 1; // Toe FSR is connected to analog
1
int toefsrReading;      // the analog reading from the FSR
resistor divider

//Setup For Measuring Current Using ACS712
const int analogIn = 0; //Connect current sensor with A0 of
Arduino
int mVperAmp = 185; // use 100 for 20A Module and 66 for
30A Module
int RawValue= 0;
int ACSoffset = 2500;
double Voltage = 0; //voltage measuring
double Amps = 0;// Current measuring
double AmpsLimit = 0;

//Safety Measure, Stops arduino
void stop() {
    set_sleep_mode(SLEEP_MODE_PWR_DOWN);
    cli();  // Disable interrupts
    sleep_mode();
    }

/*SAMD core*/
#ifdef ARDUINO_SAMD_VARIANT_COMPLIANCE
  #define SERIAL SerialUSB
#else
  #define SERIAL Serial
#endif

#define StepValue 130

// the cs pin of the version after v1.1 is default to D9
// v0.9b and v1.0 is default D10
const int SPI_CS_PIN=10;

MCP_CAN CAN(SPI_CS_PIN);            // Set CS pin
```

```
void subscriberSRCallback(const curobotics::motordata&
drive_com) {
  //SERIAL.println(drive_com.encoder);
  // send data: id = 0x141 (140+ID), standart frame, data
len=8, stmp: data buf
  // Incoming drive command from ROS is send to motors
through CANBus
  unsigned char len=0;
  unsigned char buf[8];
  buf[0]=0xA6;
  buf[1]=drive_com.rotation;
  buf[2]=0x02;
  buf[3]=0x02;
  buf[4]=drive_com.encoder;
  buf[5]=drive_com.encoder>>8;
  buf[6]=0x00;
  buf[7]=0x00;
  CAN.sendMsgBuf(0x141,0,8,buf);

  if(CAN_MSGAVAIL==CAN.checkReceive())        // check if
data coming
  {
    CAN.readMsgBuf(&len,buf);    //read data, len:data
length, buf: data buf

    unsigned long canId=CAN.getCanId();

    //SERIAL.println("------------------------");
    //SERIAL.print("Get data from ID: 0x");
    //SERIAL.println(CanID,HEX);

    for (int i=0; i<len; i++) //print the data
    {
      SERIAL.print(buf[i],HEX);
      SERIAL.print("\t");
    }
    // The message that comes from CANBus is forwarded to
ROS
    SRmotor_data.rotation=drive_com.rotation;
    SRmotor_data.torque=buf[3];
    SRmotor_data.speed=buf[5];
    SRmotor_data.encoder=buf[7];
    SRmotor_data_pub.publish( &SRmotor_data);
    SERIAL.println();
  }
```

```
}

void subscriberFRCallback(const curobotics::motordata&
drive_com) {
  // Depending on direction desired from ROS, motors are
controlled
  if (drive_com.rotation  == 1) {
    digitalWrite(A1A,LOW);
    digitalWrite(A1B,HIGH);
    }
  if (drive_com.rotation == -1) {
    digitalWrite(A1A,HIGH);
    digitalWrite(A1B,LOW);
    }
  if (drive_com.rotation == 0) {
    digitalWrite(A1A,HIGH);
    digitalWrite(A1B,LOW);
    }
  SERIAL.println();
}


ros::Subscriber<curobotics::motordata>
SRmotor_com_subs("SRmotor_com", &subscriberSRCallback);
ros::Subscriber<curobotics::motordata>
FRmotor_com_subs("FRmotor_com", &subscriberFRCallback);

void setup() {
  SERIAL.begin(115200);
  delay(1000);
  while (CAN_OK != CAN.begin(CAN_1000KBPS))
  {
      SERIAL.println("CAN BUS Shield init fail");
      SERIAL.println(" Init CAN BUS Shield again");
      delay(100);
  }
  SERIAL.println("CAN BUS Shield init ok!");

  node_handle.initNode();
  node_handle.advertise(SRmotor_data_pub);
  node_handle.advertise(FRmotor_data_pub);
  node_handle.advertise(Rforce_data_pub);
  node_handle.subscribe(SRmotor_com_subs);
  node_handle.subscribe(FRmotor_com_subs);
```

```
  // Reads the initial state of the outputA
  pinMode (outputA,INPUT);
  pinMode (outputB,INPUT);
  aLastState = digitalRead(outputA);
}

long GenPos= 0;
void loop() {

  //Check The Amp being send to Sagittal Motor
  RawValue = analogRead(analogIn);//reading the value from
the analog pin
  Voltage = (RawValue / 1024.0) * 5000; // Gets you mV
  Amps = ((Voltage - ACSoffset) / mVperAmp);

  //Safety measure where if Amps goes over 30, motors and
arduino are shutdown
  if (Amps  > 30) {

    //Sagittal Motor Stop CANBus Command
    unsigned char buf[8];
    buf[0]=0x81;
    buf[1]=0x00;
    buf[2]=0x00;
    buf[3]=0x00;
    buf[4]=0x00;
    buf[5]=0x00;
    buf[6]=0x00;
    buf[7]=0x00;
    CAN.sendMsgBuf(0x81,0,8,buf);

    //Frontal Motor Stop
    digitalWrite(A1A,LOW);
    digitalWrite(A1B,LOW);

    //Arduino Stop
    stop();
  }

  //For Frontal Encoder
  aState = digitalRead(outputA); // Reads the "current"
state of the outputA
   // If the previous and the current state of the outputA
are different, that means a Pulse has occured
   if (aState != aLastState){
```

```
     // If the outputB state is different to the outputA
state, that means the encoder is rotating clockwise
     if (digitalRead(outputB) != aState) {
       counter ++;
     } else {
       counter --;
     }
     //The given value is incremental in both sides, not
360 position wise. That will be done in ROS node
   FRmotor_data.encoder=counter;
   FRmotor_data_pub.publish( &FRmotor_data);
   }

  aLastState = aState; //Updates the previous state of the
outputA with the current state

  heelfsrReading = analogRead(heelfsrAnalogPin);
  toefsrReading = analogRead(toefsrAnalogPin);
  Rforce_data.heel=heelfsrReading;
  Rforce_data.toe=toefsrReading;
  Rforce_data_pub.publish(&Rforce_data);

  node_handle.spinOnce();
  delay(1);
  }
```

A-2 ROS Node Code in Python

```
#!/usr/bin/env python
from json import encoder
from turtle import right
import numpy as np
from numpy import genfromtxt
from matplotlib import pyplot as plt
import time
import rospy
import message_filters
from curobotics.msg import motordata, forcesensors

# CSV datas containing necessary Gait values from
referenced research paper
data_torque=genfromtxt('/home/lastingk/catkin_ws/src/curobo
tics/data/Data_torque.csv', delimiter='_')
data_angle=genfromtxt('/home/lastingk/catkin_ws/src/curobot
ics/data/Data_anglevelocity.csv', delimiter='_')

user_weight=60.00
weight_support_ratio=0.2
assistance_ratio=0.6
transmission_eff= 0.889600617 # Transmission will be done
with steel wires, there will be friction that decreases
transmission efficiency. General prediction value is given
here.
motor_maximum_torque=40.5
gait_cycle=0.0 #0 to 100
gait_time=1.06 #in seconds
len_torque_array=len(data_torque)
len_angle_array=len(data_angle)
lefttorque_dif=len(data_torque)/2
leftangle_dif=len(data_angle)/2
data_torque[-1,0]=100

# 4th(3rd in array) column, representing required moment
data_torque=np.column_stack((data_torque,data_torque[:,1]*(
user_weight)*(1-weight_support_ratio)))

# 5th(4th in array) column, representing assistive moment
data_torque=np.column_stack((data_torque,data_torque[:,3]*(
assistance_ratio)))
```

```python
# Simple way to make sure we don't send our motor more than maximum limit of motor
for n in range(len(data_torque)):
        if data_torque[n,4] > motor_maximum_torque:
                data_torque[n,4]=motor_maximum_torque

# 6th column, representing patient's moment work
data_torque=np.column_stack((data_torque,data_torque[:,3]-data_torque[:,4]))

data_torque[:,0]=data_torque[:,0]*gait_time/100
data_angle[:,0]=data_angle[:,0]*gait_time/100

gait_td=time.time()

SLmotor_data_sub = message_filters.Subscriber('SLmotor_data', motordata)
SRmotor_data_sub = message_filters.Subscriber('SRmotor_data', motordata)
FLmotor_data_sub = message_filters.Subscriber('FLmotor_data', motordata)
FRmotor_data_sub = message_filters.Subscriber('FRmotor_data', motordata)
LForce_sub = message_filters.Subscriber('Lforce_data', forcesensors)
RForce_sub = message_filters.Subscriber('Rforce_data', forcesensors)

def checkstride(heelforce, toeforce):
        #Assuming sensor data is in kg
        # Stride represent
        # 0 for heel strike
        # 1 for toe strike
        # 2 for heel off
        # 3 for toe off
        stride=0
        if heelforce>2 & toeforce <2:
                stride=0
        elif heelforce>2 & toeforce >2:
                stride=1
        elif heelforce<2 & toeforce >2:
                stride=2
        elif heelforce<2 & toeforce <2:
                stride=3
        return(stride)
```

```
def checkrotation(desired_angle,encoder_angle):
        # After toe off, pronation
        # After heel off, supination
        # This function will check which direction motor
needs to rotate.
        if desired_angle < encoder_angle:
                rotation=-1
        if desired_angle > encoder_angle:
                rotation=1
        return(rotation)

def
frontalcontrol(FLmotor_data_sub,FRmotor_data_sub,LForce_sub
,RForce_sub):
        stride_right=checkstride(RForce_sub.heel,RForce_sub
.toe)
        stride_left=checkstride(LForce_sub.heel,LForce_sub.
toe)
        rotation_left=0
        rotation_right=0
        if stride_right==2:
                # Pronation is desired

        rotation_right=checkrotation(3,FRmotor_data_sub.enc
oder)
        elif stride_right==3:
                # Supination is desired, encoder works in 0-
360 pattern and may need modification
                rotation_right=checkrotation(-
3,FRmotor_data_sub.encoder)
        if stride_left==2:
                # Pronation is desired

        rotation_left=checkrotation(3,FLmotor_data_sub.enco
der)
        elif stride_left==3:
                # Supination is desired, encoder works in 0-
360 pattern and may need modification
                rotation_left=checkrotation(-
3,FLmotor_data_sub.encoder)

        pubfr = rospy.Publisher('FRmotor_req', motordata,
queue_size=10)
```

```
        pubfl = rospy.Publisher('FLmotor_req', motordata, 
queue_size=10)
        fr_motor=motordata()
        fl_motor=motordata()
        fr_motor.rotation=rotation_right
        fl_motor.rotation=rotation_left
        pubfr.publish(fr_motor)
        pubfl.publish(fl_motor)

def sagittalcontrol(SLmotor_data_sub, SRmotor_data_sub):
        global data_torque,data_angle, gait_td
        td=time.time()-gait_td
        torque_index=np.searchsorted(data_torque[:,0],td%ga
it_time,side="right")
        right_torque=data_torque[torque_index,4]
        left_torque=data_torque[(torque_index+len_torque_ar
ray)%len_torque_array,4]

        angle_index=np.searchsorted(data_angle[:,0],td%gait
_time,side="right")
        right_angle=data_angle[angle_index,1]
        left_angle=data_angle[(angle_index+len_angle_array)
%len_angle_array,1]
        right_vel=data_angle[angle_index,3]
        left_vel=data_angle[(angle_index+len_angle_array)%l
en_angle_array,3]

        pubsl = rospy.Publisher('SLmotor_req', motordata, 
queue_size=10)
        pubsr = rospy.Publisher('SRmotor_req', motordata, 
queue_size=10)
        sr_motor=motordata()
        sl_motor=motordata()

        sr_motor.encoder=right_angle
        sr_motor.speed=right_vel
        sr_motor.torque=right_torque

        sl_motor.encoder=left_angle
        sl_motor.speed=left_vel
        sl_motor.torque=left_torque

        pubsr.publish(sr_motor)
        pubsl.publish(sl_motor)
```

```
        #print("RT: ", right_torque, "LT: ", left_torque,
"RA: ", right_angle, "LA: ", left_angle, "RV: ", right_vel,
"LV: ", left_vel)

def talker():
        rospy.init_node('Gait_Motion', anonymous=True)
        ts =
message_filters.ApproximateTimeSynchronizer([SLmotor_data_s
ub, SRmotor_data_sub], queue_size=5, slop=0.1)
        tf =
message_filters.ApproximateTimeSynchronizer([FLmotor_data_s
ub, FRmotor_data_sub,LForce_sub,RForce_sub], queue_size=5,
slop=0.1)
        ts.registerCallback(sagittalcontrol)
        tf.registerCallback(frontalcontrol)
        print("is it there yet?")
        rospy.spin()

if __name__ == '__main__':
        try:
                talker()
        except rospy.ROSInterruptException:
                pass
```

A-3 Position and Speed Data Set

Delimitir "_", Order of item is gait cycle, angle, time difference, angular velocity respectively

```
2.274450341167559841e-01_-2.313312843140888475e+00_2.274450341167559841e-01_-7.028277634961271447e-01
5.307050796057604103e-01_-2.518656534610819620e+00_3.032600454890044261e-01_-9.028277634961271447e-01
9.097801364670212720e-01_-2.775336148948237991e+00_3.790750568612608618e-01_-8.464010282776437410e-01
1.137225170583776368e+00_-3.134999444542980740e+00_2.274450341167550960e-01_-9.487917737789255401e-01
1.516300227445033677e+00_-3.391679058880399111e+00_3.790750568612573090e-01_-1.128534704370182773e+00
1.971190295678543869e+00_-3.751108477833369648e+00_4.548900682335101919e-01_-9.481748071979393400e-01
2.501895375284305167e+00_-4.059046056157679772e+00_5.307050796057612985e-01_-6.769494430162751319e-01
2.805155420773313146e+00_-4.367217511123758555e+00_3.032600454890079789e-01_-5.806830701432262876e-01
3.184230477634572232e+00_-4.623897125461173374e+00_3.790750568612590854e-01_-8.464010282776220917e-01
3.714935557240333530e+00_-4.829006940289346517e+00_5.307050796057612985e-01_-5.410796915167199739e-01
4.245640636846096605e+00_-4.982702873369444063e+00_5.307050796057630748e-01_-2.896070510466413062e-01
4.700530705079606797e+00_-4.982235120085913849e+00_4.548900682335101919e-01_8.813808299662165974e-04
5.307050796057620978e+00_-4.981611449041206896e+00_6.065200909780141814e-01_1.371036846614117743e-03
5.837755875663382277e+00_-4.929651855129016269e+00_5.307050796057612985e-01_8.566838046272422236e-02
6.292645943896892469e+00_-4.723528574853197881e+00_4.548900682335101919e-01_3.883951524054354287e-01
6.671721000758149778e+00_-4.466069371709892266e+00_3.790750568612573090e-01_5.659811482433668717e-01
7.126611068991661746e+00_-4.259946091434073878e+00_4.548900682335119683e-01_5.437532133676106216e-01
7.505686125852920831e+00_-4.002486888290771816e+00_3.790750568612590854e-01_5.659811482433568797e-01
7.808946171341927034e+00_-3.745105644028058123e+00_3.032600454890062025e-01_6.789717223650377775e-01
8.188021228203187007e+00_-3.539060322632831657e+00_3.790750568612599736e-01_6.794344473007606355e-01
8.642911296436693647e+00_-3.178695397112793586e+00_4.548900682335066392e-01_9.506426735218568291e-01
9.021986353297958061e+00_-2.715580666977206903e+00_3.790750568612644145e-01_1.018080548414739495e+00
9.401061410159215370e+00_-2.458121463833904841e+00_3.790750568612573090e-01_6.791773778920202620e-01
9.780136467020472679e+00_-2.252076142438671269e+00_3.790750568612573090e-01_5.435475578406279151e-01
1.008339651250948066e+01_-1.994694898175961129e+00_3.032600454890079789e-01_6.789717223650315603e-01
1.046247156937073797e+01_-1.737235695032659066e+00_3.790750568612573090e-01_8.489717223650352640e-01
1.076573161485974595e+01_-1.479854450769941820e+00_3.032600454890079789e-01_6.789717223650503231e-01
1.106899166034875392e+01_-1.273887088255296618e+00_3.032600454890079789e-01_6.791773778920399129e-01
1.137225170583775835e+01_-1.016505843992586477e+00_3.032600454890044261e-01_8.487146529562834552e-01
1.175132676269901921e+01_-7.076327591012159246e-01_3.790750568612608618e-01_1.018508997429302454e+00
1.220621683093252585e+01_-3.472678335811778538e-01_4.548900682335066392e-01_9.506426735218546087e-01
1.243366186504928095e+01_-1.927923116952001692e-01_2.274450341167550960e-01_3.395886889460102975e-01
1.258529188779378316e+01_-3.839474868980730093e-02_1.516300227445022131e-01_6.788346186803774218e-01
1.288855193328279114e+01_1.161587320767623055e-01_3.032600454890079789e-01_1.019280205655534566e+00
1.311599696739954624e+01_2.706342539627470956e-01_2.274450341167550960e-01_5.093830334190329046e-01
1.334344200151630488e+01_4.765236575967968236e-01_2.274450341167586487e-01_9.052270779777054077e-01
1.357088703563305643e+01_6.824130612308536570e-01_2.274450341167515433e-01_9.052270779777225052e-01
1.372251705837756219e+01_7.339828607401024385e-01_1.516300227445057658e-01_2.267352185090007188e-01
1.394996209249431729e+01_8.884583826260801231e-01_2.274450341167550960e-01_1.018766066838006967e+00
```

1.425322213798332527e+01_1.043011863392649730e+00_3.032600454890079789e-01_6.795201371036844185e-01
1.455648218347232969e+01_1.300393107655366975e+00_3.032600454890044261e-01_8.487146529563068809e-01
1.485974222896133767e+01_1.557774351918080669e+00_3.032600454890079789e-01_8.487146529563051045e-01
1.516300227445034210e+01_1.815155596180794362e+00_3.032600454890044261e-01_8.487146529562952235e-01
1.561789234268385584e+01_1.969864994708540706e+00_4.548900682335137446e-01_5.101542416452475859e-01
1.592115238817286382e+01_2.175832357223182356e+00_3.032600454890079789e-01_4.527849185946837274e-01
1.630022744503412113e+01_2.381877678618412375e+00_3.790750568612573090e-01_6.794344473007684071e-01
1.675511751326763132e+01_2.639414840642302806e+00_4.548900682335101919e-01_6.793830334190251730e-01
1.721000758150114152e+01_2.794124239170049151e+00_4.548900682335101919e-01_3.401028277634957631e-01
1.758908263836239882e+01_2.948755678817207126e+00_3.790750568612573090e-01_3.399314481576689584e-01
1.796815769522365969e+01_3.154801000212437145e+00_3.790750568612608618e-01_5.435475578406185893e-01
1.834723275208491700e+01_3.309432439859595121e+00_3.790750568612573090e-01_4.079177377892002077e-01
1.872630780894617430e+01_3.464063879506753096e+00_3.790750568612573090e-01_4.079177377892040379e-01
1.925701288855193738e+01_3.618851236915091363e+00_5.307050796057630748e-01_4.083290488431976506e-01
1.978771796815769690e+01_3.825052476071498120e+00_5.307050796057595221e-01_3.885420492104284529e-01
2.031842304776345998e+01_3.928425951731757237e+00_5.307050796057630748e-01_1.947851634226891449e-01
2.084912812736922305e+01_4.134627190888163994e+00_5.307050796057630748e-01_3.885420492104284529e-01
2.153146322971948834e+01_4.340984347805747490e+00_6.823351023502652879e-01_3.888358428204172212e-01
2.228961334344200296e+01_4.444591700107775267e+00_7.581501137225146181e-01_1.518423307626385055e-01
2.312357846853677401e+01_4.805346420030755183e+00_8.339651250947710537e-01_4.758354755784120482e-01
2.380591357088704285e+01_4.857461931704122549e+00_6.823351023502688406e-01_6.249123627015578525e-02
2.426080363912054594e+01_5.012171330231868893e+00_4.548900682335030865e-01_2.267352185089959726e-01
2.486732373009855834e+01_5.218450528268864019e+00_6.065200909780124050e-01_4.534704370180014377e-01
2.532221379833207564e+01_5.424573808544682407e+00_4.548900682335172974e-01_3.398457583547562777e-01
2.585291887793783516e+01_5.682188929449162984e+00_5.307050796057595221e-01_5.663239074550077135e-01
2.645943896891585112e+01_5.837054245738086067e+00_6.065200909780159577e-01_2.918105031215578182e-01
2.691432903714936131e+01_6.043177526013904455e+00_4.548900682335101919e-01_3.398457583547542793e-01
2.744503411675512083e+01_6.197964883422237392e+00_5.307050796057595221e-01_3.402742073693186264e-01
2.812736921910538612e+01_6.301494276843678577e+00_6.823351023502652879e-01_1.950789570326879330e-01
2.865807429871114564e+01_6.507695516000085334e+00_5.307050796057595221e-01_3.021993716081117221e-01
2.926459438968916160e+01_6.816802477533224547e+00_6.065200909780159577e-01_5.824458318031606741e-01
2.979529946929492112e+01_6.971589834941561037e+00_5.307050796057595221e-01_2.552056555269938132e-01
3.032600454890068775e+01_7.177791074097967794e+00_5.307050796057666275e-01_3.885420492104310619e-01
3.078089461713419794e+01_7.332500472625714139e+00_4.548900682335101919e-01_2.915167095115651641e-01
3.131159969673996457e+01_7.590115593530192939e+00_5.307050796057666275e-01_5.663239074550125984e-01
3.184230477634572409e+01_7.796316832686601472e+00_5.307050796057595221e-01_3.885420492104292300e-01
3.237300985595148006e+01_8.053931953591080273e+00_5.307050796057559694e-01_4.854204921042987175e-01
3.282789992418499025e+01_8.208641352118824841e+00_4.548900682335101919e-01_2.915167095115676621e-01
3.335860500379074978e+01_8.569084236519447728e+00_5.307050796057595221e-01_7.923736075407027402e-01
3.396512509476876573e+01_8.672535671060300544e+00_6.065200909780159577e-01_1.949320602276935488e-01
3.457164518574678169e+01_8.878814869097295670e+00_6.065200909780159577e-01_3.401028277634944308e-01
3.525398028809704698e+01_9.085172026014877389e+00_6.823351023502652879e-01_3.402313624678615644e-01
3.578468536770280650e+01_9.291373265171285922e+00_5.307050796057595221e-01_3.021993716081143311e-01
3.631539044730856602e+01_9.548988386075766499e+00_5.307050796057595221e-01_4.854204921043020482e-01
3.684609552691433265e+01_9.703775743484099436e+00_5.307050796057666275e-01_2.916636063165601311e-01
3.737680060652009217e+01_9.961390864388578237e+00_5.307050796057595221e-01_4.854204921042922227e-01

```
3.790750568612585170e+01_1.011617822179691473e+01_5.307050796057595221e-01_2.916636063165667925e-01
3.836239575435936899e+01_1.032230150207273311e+01_4.548900682335172974e-01_3.883951524054367055e-01
3.896891584533737785e+01_1.042575293661358238e+01_6.065200909780088523e-01_2.274207369322967520e-01
3.949962092494314447e+01_1.058054029402191887e+01_5.307050796057666275e-01_2.552056555269968108e-01
4.003032600454891110e+01_1.073532765143025536e+01_5.307050796057666275e-01_2.916636063165629067e-01
4.063684609552691285e+01_1.078736520422303435e+01_6.065200909780017469e-01_9.805361733382127187e-02
4.116755117513267948e+01_1.089073867988329702e+01_5.307050796057666275e-01_1.704370179948614117e-01
4.169825625473843900e+01_1.109693991903970378e+01_5.307050796057595221e-01_3.885420492104258994e-01
4.230477634571645495e+01_1.114897747183248455e+01_6.065200909780159577e-01_9.805361733382593481e-02
4.283548142532221448e+01_1.125235094749274545e+01_5.307050796057595221e-01_1.704370179948545005e-01
4.336618650492798110e+01_1.130431054140493607e+01_5.307050796057666275e-01_9.790672052882821996e-02
4.382107657316149130e+01_1.145901993993268420e+01_4.548900682335101919e-01_2.915167095115685503e-01
4.457922668688400591e+01_1.145979952873856789e+01_7.581501137225146181e-01_1.713796058267647287e-03
4.510993176648977254e+01_1.146034524090268647e+01_5.307050796057666275e-01_7.197943444724140809e-04
4.579226686884003072e+01_1.135821910733184126e+01_6.823351023502581825e-01_-1.924348145427766388e-01
4.655041698256255245e+01_1.125617093264157909e+01_7.581501137225217235e-01_-1.495572693516191431e-01
4.723275208491281774e+01_1.110263091732265828e+01_6.823351023502652879e-01_-2.025192802056552943e-01
4.783927217589083369e+01_1.094901294312315265e+01_6.065200909780159577e-01_-2.251356755212754746e-01
4.844579226686884965e+01_1.074398108717557143e+01_6.065200909780159577e-01_-3.380462724935732743e-01
4.912812736921910783e+01_1.048761330836050654e+01_6.823351023502581825e-01_-4.226863753213366270e-01
4.958301743745261803e+01_1.012818388940753245e+01_4.548900682335101919e-01_-5.267637817766419595e-01
5.003790750568613532e+01_9.665926706958416048e+00_4.548900682335172974e-01_-1.016195372750640979e+00
5.034116755117513975e+01_9.357755251992339041e+00_3.032600454890044261e-01_-6.774635818337487736e-01
5.072024260803639351e+01_9.049661755906846849e+00_3.790750568612537563e-01_-1.015938303341918481e+00
5.109931766489765437e+01_8.792982141569430254e+00_3.790750568612608618e-01_-6.771208226221134829e-01
5.132676269901441657e+01_8.381904964226619015e+00_2.274450341167622014e-01_-1.084421593830329433e+00
5.170583775587567033e+01_8.022397586393060109e+00_3.790750568612537563e-01_-1.580634104541497997e+00
5.193328278999242542e+01_7.559906527302175050e+00_2.274450341167550960e-01_-1.220051413881770186e+00
5.231235784685367918e+01_7.200399149468614368e+00_3.790750568612537563e-01_-1.580634104541555285e+00
5.253980288097044138e+01_6.892149735621945439e+00_2.274450341167622014e-01_-8.131619537275228682e-01
5.276724791508719647e+01_6.532486440027208019e+00_2.274450341167550960e-01_-1.581319622964812988e+00
5.314632297194845023e+01_6.121565180445573517e+00_3.790750568612537563e-01_-1.806683804627253087e+00
5.344958301743745466e+01_5.659152080235280380e+00_3.032600454890044261e-01_-1.219845758354768606e+00
5.375284306292645908e+01_5.248152861773057509e+00_3.032600454890044261e-01_-1.355269922879190680e+00
5.398028809704322128e+01_4.734247920934102183e+00_2.274450341167622014e-01_-1.694601542416468654e+00
5.435936315390448215e+01_4.220498897856323595e+00_3.790750568612608618e-01_-2.258783204798562760e+00
5.451099317664898791e+01_3.860757643380999582e+00_1.516300227445057658e-01_-9.489974293058989208e-01
5.466262319939348657e+01_3.346774743661455886e+00_1.516300227444986604e-01_-3.389717223650337896e+00
5.489006823351024877e+01_2.884283684570572603e+00_2.274450341167622014e-01_-3.050128534704470873e+00
5.504169825625474033e+01_2.524542430095248591e+00_1.516300227444915549e-01_-1.581662382176458603e+00
5.511751326762699676e+01_2.113309334991264166e+00_7.581501137225643561e-02_-2.712082262210989470e+00
5.542077331311600119e+01_1.753723998277109786e+00_3.032600454890044261e-01_-4.742930591259399975e+00
5.564821834723276339e+01_1.394060702682370589e+00_2.274450341167622014e-01_-1.185989717223661977e+00
5.587566338134951138e+01_9.315696435914873064e-01_2.274450341167479905e-01_-2.033419023136187054e+00
5.602729340409402425e+01_5.718283891161597410e-01_1.516300227445128712e-01_-1.581662382176573178e+00
5.610310841546625937e+01_1.091814122641032725e-01_7.581501137223511932e-02_-3.051156812339121949e+00
```

```
5.625473843821077224e+01_-3.533876057073683796e-01_1.516300227445128712e-01_-6.101285347045045881e+00
5.633055344958302157e+01_-7.132068190632843141e-01_7.581501137224933018e-02_-2.373007712082117582e+00
5.640636846095527801e+01_-1.021612150671128205e+00_7.581501137225643561e-02_-4.067866323907588288e+00
5.655799848369977667e+01_-1.329939523398383727e+00_1.516300227444986604e-01_-4.066838046272246210e+00
5.670962850644428954e+01_-1.689680777873711293e+00_1.516300227445128712e-01_-2.372493573264859545e+00
5.678544351781653177e+01_-2.100913872977699270e+00_7.581501137224222475e-02_-2.712082262210631534e+00
5.686125852918878820e+01_-2.512146968081687248e+00_7.581501137225643561e-02_-5.424164524422280032e+00
5.701288855193328686e+01_-2.923302104305086857e+00_1.516300227444986604e-01_-5.423136246786302905e+00
5.716451857467779973e+01_-3.180215595284270336e+00_1.516300227445128712e-01_-1.694344473007768137e+00
5.731614859742228418e+01_-3.539956849759594348e+00_1.516300227444844495e-01_-2.372493573264613964e+00
5.739196360879454062e+01_-3.899776063115510283e+00_7.581501137225643561e-02_-2.373007712082562559e+00
5.754359363153905349e+01_-4.362345081086981935e+00_1.516300227445128712e-01_-6.101285347043330809e+00
5.769522365428355215e+01_-4.773500217310381544e+00_1.516300227444986604e-01_-2.711568123393151453e+00
5.777103866565580859e+01_-5.184733312414369522e+00_7.581501137225643561e-02_-2.712082262210885553e+00
5.792266868840030725e+01_-5.493060685141625044e+00_1.516300227444986604e-01_-4.066838046272246210e+00
5.807429871114480591e+01_-5.904215821365024652e+00_1.516300227444986604e-01_-2.711568123393405472e+00
5.830174374526157521e+01_-6.315292998707832339e+00_2.274450341167693068e-01_-2.711053984575901854e+00
5.852918877937831610e+01_-6.726370176050643579e+00_2.274450341167408851e-01_-1.807369323050447285e+00
5.868081880212282897e+01_-7.137525312274043188e+00_1.516300227445128712e-01_-1.807712082262326714e+00
5.883244882486734184e+01_-7.651508211993583330e+00_1.516300227445128712e-01_-3.389717223650155820e+00
5.883244882486734184e+01_-8.062819265978156125e+00_0.000000000000000000e+00_-2.712596401028088522e+00
5.898407884761182629e+01_-8.473974402201555733e+00_1.516300227444844495e-01_-2.712596401028088522e+00
5.921152388172859560e+01_-9.039293224788586656e+00_2.274450341167693068e-01_-3.728277634961935316e+00
5.936315390447309426e+01_-9.553276124508126799e+00_1.516300227444986604e-01_-2.259811482433437213e+00
5.966641394996209158e+01_-1.011851698821456580e+01_3.032600454889973207e-01_-3.727763496144082200e+00
5.989385898407886089e+01_-1.068383581080159672e+01_2.274450341167693068e-01_-1.864138817480792909e+00
6.012130401819560177e+01_-1.114632686989247645e+01_2.274450341167408851e-01_-2.033419023136107562e+00
6.027293404094011464e+01_-1.160889588786395166e+01_1.516300227445128712e-01_-2.033761782348046498e+00
6.057619408642911196e+01_-1.191706734283003044e+01_3.032600454889973207e-01_-2.032390745501162943e+00
6.080363912054588127e+01_-1.227673063842475898e+01_2.274450341167693068e-01_-1.185989717223654427e+00
6.118271417740713503e+01_-1.258482413451025650e+01_3.790750568612537563e-01_-1.354584404455819513e+00
6.148597422289614656e+01_-1.299582335297247226e+01_3.032600454890115316e-01_-1.084215938303338733e+00
6.186504927975740031e+01_-1.335533073080603828e+01_3.790750568612537563e-01_-1.185475578406165464e+00
6.239575435936316694e+01_-1.361185442738227636e+01_5.307050796057666275e-01_-6.767095115681245332e-01
6.292645943896891936e+01_-1.361130871521815777e+01_5.307050796057524167e-01_1.028277634960579091e-03
6.338134950720242955e+01_-1.361084096193462756e+01_4.548900682335101919e-01_8.813808299662342699e-04
6.383623957543593974e+01_-1.340471768165880917e+01_4.548900682335101919e-01_4.531276778063407784e-01
6.421531463229720771e+01_-1.319867236026357915e+01_3.790750568612679672e-01_4.529562982005140293e-01
6.459438968915846147e+01_-1.294121315712028064e+01_3.790750568612537563e-01_6.791773778920046079e-01
6.504927975739197166e+01_-1.268367599509638666e+01_4.548900682335101919e-01_6.793830334190409381e-01
6.527672479150872675e+01_-1.237495882796618929e+01_2.274450341167550960e-01_6.786632390745506171e-01
6.550416982562548185e+01_-1.206624166083599192e+01_2.274450341167550960e-01_1.357326478149101234e+00
6.588324488248674982e+01_-1.175736857594462492e+01_3.790750568612679672e-01_1.358011996572377145e+00
6.618650492797574714e+01_-1.144857344993383208e+01_3.032600454889973207e-01_8.146015424164512186e-01
6.641394996209250223e+01_-1.113985628280363471e+01_2.274450341167550960e-01_1.017994858611857678e+00
6.664139499620925733e+01_-1.072831135217729326e+01_2.274450341167550960e-01_1.809425878320481518e+00
```

6.709628506444276752e+01_-1.036794642665726229e+01_4.548900682335101919e-01_1.584404455869736328e+00
6.739954510993177905e+01_-9.956323537150328917e+00_3.032600454890115316e-01_9.048843187660753351e-01
6.770280515542077637e+01_-9.647528411139543181e+00_3.032600454889973207e-01_1.018251928020550201e+00
6.800606520090978790e+01_-9.287319403380685401e+00_3.032600454890115316e-01_1.187789203084870682e+00
6.830932524639879944e+01_-8.978524277369899664e+00_3.032600454890115316e-01_1.018251928020550201e+00
6.868840030326005319e+01_-8.463995665486237385e+00_3.790750568612537563e-01_1.696658097686350120e+00
6.906747536012132116e+01_-8.052294817098719193e+00_3.790750568612679672e-01_1.086066838046286565e+00
6.929492039423806204e+01_-7.692163768220453335e+00_2.274450341167408851e-01_9.500257069408416521e-01
6.959818043972707358e+01_-7.331954760461595555e+00_3.032600454890115316e-01_1.583718937446543906e+00
6.997725549658832733e+01_-6.868840030326005319e+00_3.790750568612537563e-01_1.527120822622085150e+00
7.035633055344958109e+01_-6.457139181938487127e+00_3.790750568612537563e-01_1.086066838046286565e+00
7.073540561031084906e+01_-5.994024451802900444e+00_3.790750568612679672e-01_1.221696658097693122e+00
7.119029567854435925e+01_-5.479417881038646243e+00_4.548900682335101919e-01_1.357532133676068842e+00
7.164518574677786944e+01_-4.964811310274392042e+00_4.548900682335101919e-01_1.131276778063418975e+00
7.202426080363912320e+01_-4.501696580138805359e+00_3.790750568612537563e-01_1.018080548414731501e+00
7.255496588324488982e+01_-3.987012050493966342e+00_5.307050796057666275e-01_1.357737789203102396e+00
7.300985595147840002e+01_-3.523819361477787737e+00_4.548900682335101919e-01_8.727873668747631042e-01
7.331311599696741155e+01_-2.906540944978573293e+00_3.032600454890115316e-01_1.356983718937439853e+00
7.376800606520092174e+01_-2.391934374214322645e+00_4.548900682335101919e-01_1.696915167095090160e+00
7.414708112206217550e+01_-1.877405762330656813e+00_3.790750568612537563e-01_1.131105398457592059e+00
7.467778620166792791e+01_-1.465548996181965435e+00_5.307050796057524167e-01_1.086478149100261525e+00
7.520849128127369454e+01_-9.508644665371228655e-01_5.307050796057666275e-01_9.698127065736564800e-01
7.573919636087946117e+01_-4.361799368922838482e-01_5.307050796057666275e-01_9.698127065736238395e-01
7.634571645185747002e+01_-2.424521186300054865e-02_6.065200909780088523e-01_7.762027175908855492e-01
7.702805155420773531e+01_5.420091172910836974e-01_6.823351023502652879e-01_9.336118251928038347e-01
7.771038665655801481e+01_9.026079194528868754e-01_6.823351023502794988e-01_5.284775778349093400e-01
7.862016679302502098e+01_1.314854480004523651e+00_9.097801364670061730e-01_6.041702370751085294e-01
7.952994692949205557e+01_1.572859395311944297e+00_9.097801364670345947e-01_2.835904027420776408e-01
8.051554207733133239e+01_1.830942269499953312e+00_9.855951478392768195e-01_2.836760925449821613e-01
8.097043214556482837e+01_1.882823904531555570e+00_4.548900682334959811e-01_5.263990508206389113e-02
8.142532221379833857e+01_1.883291657815085784e+00_4.548900682335101919e-01_1.028277634960620508e-03
8.180439727065959232e+01_1.986509215714171717e+00_3.790750568612537563e-01_2.269065981148239153e-01
8.225928733889310251e+01_1.986976968997701931e+00_4.548900682335101919e-01_1.233933161952721408e-03
8.278999241849886914e+01_1.987522681161820515e+00_5.307050796057666275e-01_1.199657240787352971e-03
8.332069749810463577e+01_1.885240629829798564e+00_5.307050796057666275e-01_-1.927286081527653794e-01
8.392721758908264462e+01_1.885864300874505517e+00_6.065200909780088523e-01_1.175174439954947657e-03
8.438210765731616902e+01_1.886332054158035731e+00_4.548900682335244028e-01_7.712082262204472205e-04
8.476118271417742278e+01_1.783894085064833490e+00_3.790750568612537563e-01_-2.251928020565492350e-01
8.521607278241093297e+01_1.681534074852219618e+00_4.548900682335101919e-01_-2.700257069408787958e-01
8.574677786201668539e+01_1.682079787016338202e+00_5.307050796057524167e-01_1.199657240787352971e-03
8.627748294162245202e+01_1.579797735684312698e+00_5.307050796057666275e-01_-1.927286081527772310e-01
8.665655799848370577e+01_1.374532003094966370e+00_3.790750568612537563e-01_-3.867792875504934558e-01
8.711144806671723018e+01_1.272171992882356051e+00_4.548900682335244028e-01_-2.700257069408694144e-01
8.764215314632299680e+01_1.118476059802258504e+00_5.307050796057666275e-01_-3.378748928877372548e-01
8.809704321455649279e+01_1.016116049589644632e+00_4.548900682334959811e-01_-1.928755049577664249e-01
8.855193328279000298e+01_8.623421576289587165e-01_4.548900682335101919e-01_-3.380462724935851537e-01

```
8.908263836239576960e+01_6.572323428007926793e-01_5.307050796057666275e-01_-4.508997429305850457e-01
8.953752843062927980e+01_4.520445690920311677e-01_4.548900682335101919e-01_-3.866323907455057607e-01
8.999241849886278999e+01_2.982706771313452521e-01_4.548900682335101919e-01_-3.380462724935745511e-01
9.059893858984079884e+01_1.446527029318396274e-01_6.065200909780088523e-01_-3.377035132819132257e-01
9.128127369219106413e+01_-8.887312387077628273e-03_6.823351023502652879e-01_-2.531491002570668281e-01
9.203942380591358585e+01_-1.623493688254100675e-01_7.581501137225217235e-01_-2.249071693801783511e-01
9.272175890826385114e+01_-1.616477389001147458e-01_6.823351023502652879e-01_9.254498714645236542e-04
9.332827899924185999e+01_-1.610240678554077931e-01_6.065200909780088523e-01_9.140245644094118285e-04
9.393479909021988306e+01_-1.604003968107008404e-01_6.065200909780230631e-01_1.028277634960596439e-03
9.438968915845337904e+01_-1.599326435271706259e-01_4.548900682334959811e-01_7.712082262204292228e-04
9.492039423805914566e+01_-1.593869313630484896e-01_5.307050796057666275e-01_1.199657240795200643e-03
9.560272934040941095e+01_-1.586853014377531679e-01_6.823351023502652879e-01_1.322071244949316006e-03
9.643669446550417490e+01_-1.578277537512811080e-01_8.339651250947639483e-01_1.256783776062941373e-03
9.704321455648219796e+01_-1.572040827065741553e-01_6.065200909780230631e-01_7.478382799713411593e-04
9.780136467020471969e+01_-1.564244939006904644e-01_7.581501137225217235e-01_1.285347043700715407e-03
9.818043972706597344e+01_-1.560346994977486190e-01_3.790750568612537563e-01_5.141388174802909551e-04
9.878695981804398230e+01_-2.582387919491857531e-01_6.065200909780088523e-01_-2.696143958868945645e-01
9.916603487490523605e+01_-2.578489975462403549e-01_3.790750568612537563e-01_6.426735218562302848e-04
9.962092494313876045e+01_-2.573812442627101404e-01_4.548900682335244028e-01_1.233933161952721408e-03
1.000000000000000000e+02_-2.569134909791799259e-01_3.790750568612395455e-01_1.028277634960556106e-03
```

A-4 Torque Data Set

Delimitir “_”, Order of item is gait cycle, torque/kg, time difference

```
3.705652950121907097e-01_-2.091254752851723353e-02_3.705652950121907097e-01
6.669237170232067413e-01_-3.231939163498109835e-02_2.963584220110160317e-01
1.037489012035398339e+00_-5.133079847908739168e-02_3.705652950121915978e-01
1.481979738052552342e+00_-6.653992395437269280e-02_4.444907260171540031e-01
2.074696582074582629e+00_-8.935361216730042244e-02_5.927168440220302870e-01
2.519374936089213257e+00_-1.121673003802281521e-01_4.446783540146306279e-01
3.038166349106281139e+00_-1.387832699619773180e-01_5.187914130170678817e-01
3.482938517119648303e+00_-1.653992395437260399e-01_4.447721680133671640e-01
4.149580792146642771e+00_-1.844106463878323332e-01_6.666422750269944686e-01
4.816316881172375552e+00_-2.072243346007605069e-01_6.667360890257327810e-01
5.482583900204418548e+00_-2.110266159695814991e-01_6.662670190320429953e-01
6.000718615230325348e+00_-2.110266159695814991e-01_5.181347150259068002e-01
6.592591133263723258e+00_-1.996197718631176343e-01_5.918725180333979097e-01
7.184463651297117615e+00_-1.882129277566537695e-01_5.918725180333943570e-01
7.850261600335477929e+00_-1.730038022813684684e-01_6.657979490383603149e-01
8.367833431368961072e+00_-1.501901140684411828e-01_5.175718310334831429e-01
8.811573645396219234e+00_-1.349809885931558817e-01_4.437402140272581619e-01
9.403258535432140519e+00_-1.159695817490495884e-01_5.916848900359212848e-01
9.846904935460660369e+00_-9.695817490494285096e-02_4.436464000285198495e-01
1.036438295249540431e+01_-7.034220532319368502e-02_5.174780170347439423e-01
1.080802935252392416e+01_-5.133079847908739168e-02_4.436464000285198495e-01
1.125158193855370570e+01_-2.851711026615966205e-02_4.435525860297815370e-01
1.169513452458348723e+01_-5.703422053231932409e-03_4.435525860297815370e-01
1.221280016961571135e+01_1.330798479087480501e-02_5.176656450322241199e-01
1.250831426563808790e+01_3.612167300380209056e-02_2.955140960223765489e-01
1.295196066566661131e+01_5.513307984790882799e-02_4.436464000285234022e-01
1.346953249670009356e+01_7.794676806083655762e-02_5.175718310334822547e-01
1.383906583772617083e+01_1.007604562737642873e-01_3.695333410260772666e-01
1.413448611974981617e+01_1.273764258555130091e-01_2.954202820236453420e-01
1.457794489178086650e+01_1.539923954372621751e-01_4.434587720310503300e-01
1.509542290881560689e+01_1.806083650190113410e-01_5.174780170347403896e-01
1.546486243584295295e+01_2.072243346007605069e-01_3.694395270273460596e-01
```

```
1.590850883587146924e+01_2.262357414448668003e-01_4.436464000285162967e-01
1.620411674589259121e+01_2.452471482889735377e-01_2.956079100211219668e-01
1.664766933192236564e+01_2.680608365019012673e-01_4.435525860297744316e-01
1.701720267294845002e+01_2.908745247148289970e-01_3.695333410260843721e-01
1.738692364197200391e+01_3.060836501901138540e-01_3.697209690235538915e-01
1.783047622800178900e+01_3.288973384030415836e-01_4.435525860297850897e-01
1.842187967604149890e+01_3.593155893536121859e-01_5.914034480397099003e-01
1.893945150707498115e+01_3.821292775665399155e-01_5.175718310334822547e-01
1.945683571011099033e+01_4.125475285171105178e-01_5.173842030360091826e-01
2.019618383415935980e+01_4.467680608365021122e-01_7.393481240483694705e-01
2.086179415520025060e+01_4.695817490494293978e-01_6.656103210408907955e-01
2.152740447624113429e+01_4.923954372623575715e-01_6.656103210408836901e-01
2.226703404228572225e+01_5.152091254752853011e-01_7.396295660445879605e-01
2.300675742232904142e+01_5.342205323193915945e-01_7.397233800433191675e-01
2.352451688136000030e+01_5.494296577946768956e-01_5.177594590309588796e-01
2.419040864439709893e+01_5.608365019011407604e-01_6.658917630370986274e-01
2.493031965243790182e+01_5.722433460076046252e-01_7.399110080408028978e-01
2.581826915048609550e+01_5.836501901140684900e-01_8.879494980481936750e-01
2.670631246253302749e+01_5.912547528517111406e-01_8.880433120469319874e-01
2.752024271557753110e+01_6.026615969581750054e-01_8.139302530445036155e-01
2.833436059661950424e+01_6.064638783269962197e-01_8.141178810419731349e-01
2.900053380165281425e+01_6.064638783269962197e-01_6.661732050333100119e-01
2.974025718169613697e+01_6.254752851711027350e-01_7.397233800433227202e-01
3.033184825773331283e+01_6.482889733840304647e-01_5.915910760371758670e-01
3.092362696176797954e+01_6.634980988593155438e-01_5.917787040346667027e-01
3.151521803780516251e+01_6.863117870722434954e-01_5.915910760371829724e-01
3.195877062383494405e+01_7.091254752851710030e-01_4.435525860297815370e-01
3.262428713087709298e+01_7.357414448669201690e-01_6.655165070421489304e-01
3.336363525492546245e+01_7.699619771863117634e-01_7.393481240483694705e-01
3.388101945796147163e+01_8.003802281368819216e-01_5.173842030360091826e-01
3.454672359300109719e+01_8.193916349809886590e-01_6.657041350396255552e-01
3.528616553104820497e+01_8.498098859315588172e-01_7.394419380471077829e-01
3.587747516508918011e+01_8.840304182509506337e-01_5.913096340409751406e-01
3.646887861312889356e+01_9.144486692015210139e-01_5.914034480397134530e-01
3.698626281616489564e+01_9.448669201520911720e-01_5.173842030360020772e-01
```

3.750355320520216651e+01_9.790874524714829885e-01_5.172903890372708702e-01
3.802065596624196786e+01_1.020912547528517234e+00_5.171027610398013508e-01
3.853785254128050042e+01_1.058935361216730042e+00_5.171965750385325578e-01
3.898093605731659039e+01_1.100760456273764287e+00_4.430835160360899749e-01
3.927626252534150098e+01_1.131178707224334445e+00_2.953264680249105822e-01
3.979355291437877895e+01_1.165399239543726262e+00_5.172903890372779756e-01
4.016271099940991007e+01_1.203422053231939293e+00_3.691580850311311224e-01
4.060588832944473836e+01_1.241444866920152101e+00_4.431773300348282874e-01
4.090083954147468859e+01_1.287072243346007561e+00_2.949512120299502271e-01
4.126999762650582682e+01_1.325095057034220369e+00_3.691580850311382278e-01
4.163934333953443456e+01_1.355513307984790750e+00_3.693457130286077472e-01
4.200850142456556569e+01_1.393536121673003780e+00_3.691580850311311224e-01
4.245167875460039397e+01_1.431558935361216811e+00_4.431773300348282874e-01
4.282074302563279389e+01_1.473384030418250834e+00_3.690642710323999154e-01
4.311560042366401291e+01_1.522813688212927730e+00_2.948573980312190201e-01
4.355896538169632493e+01_1.553231939163498110e+00_4.433649580323120176e-01
4.414999357374107802e+01_1.598859315589353569e+00_5.910281920447530979e-01
4.451933928676969288e+01_1.629277566539923949e+00_3.693457130286148526e-01
4.496251661680452116e+01_1.667300380228136980e+00_4.431773300348282874e-01
4.548008844783800697e+01_1.690114068441064710e+00_5.175718310334858074e-01
4.599766027887148567e+01_1.712927756653992439e+00_5.175718310334787020e-01
4.651541973790244100e+01_1.728136882129277518e+00_5.177594590309553269e-01
4.695953520792465241e+01_1.728136882129277518e+00_4.441154700222114116e-01
4.755168916795425815e+01_1.728136882129277518e+00_5.921539600296057415e-01
4.814403075598134052e+01_1.720532319391634868e+00_5.923415880270823664e-01
4.866263454100093355e+01_1.701520912547528575e+00_5.186037850195930332e-01
4.910740670901429894e+01_1.674904942965779409e+00_4.447721680133653877e-01
4.962657337802631474e+01_1.633079847908745164e+00_5.191666690120158023e-01
4.999751392903345248e+01_1.598859315589353569e+00_3.709405510071377421e-01
5.036882973603555058e+01_1.549429657794676896e+00_3.713158070020980972e-01
5.066631392603140682e+01_1.492395437262357349e+00_2.974841899958562408e-01
5.096342286003233113e+01_1.450570342205323104e+00_2.971089340009243074e-01
5.111268093202331642e+01_1.401140684410646431e+00_1.492580719909852860e-01
5.133567680702179103e+01_1.363117870722433400e+00_2.229958749984746191e-01
5.155895412401646638e+01_1.313688212927756505e+00_2.232773169946753455e-01

5.185662594200980635e+01_1.249049429657794752e+00_2.976718179933399711e-01
5.208046614299691868e+01_1.176806083650189905e+00_2.238402009871123255e-01
5.222963040098916565e+01_1.131178707224334445e+00_1.491642579922469736e-01
5.245309534598133183e+01_1.074144486692015343e+00_2.234649449921661812e-01
5.267637266297602139e+01_1.024714828897338448e+00_2.232773169946895564e-01
5.282572454896574499e+01_9.714828897338403380e-01_1.493518859897235984e-01
5.304956474995285731e+01_8.992395437262357127e-01_2.238402009871123255e-01
5.334733038194492849e+01_8.307984790874525238e-01_2.977656319920711780e-01
5.364547126993194581e+01_7.471482889733840338e-01_2.981408879870173223e-01
5.379538603991409218e+01_6.711026615969581943e-01_1.499147699821463675e-01
5.401922624090121161e+01_5.988593155893537912e-01_2.238402009871194309e-01
5.416970389487578075e+01_5.000000000000000000e-01_1.504776539745691366e-01
5.446756334086659734e+01_4.277566539923953748e-01_2.978594459908165959e-01
5.491364890486228489e+01_3.479087452471483211e-01_4.460855639956875507e-01
5.543375371386166961e+01_2.680608365019012673e-01_5.201048089993847157e-01
5.595320182486990035e+01_2.148288973384029354e-01_5.194481110082307396e-01
5.706499152390520635e+01_1.539923954372621751e-01_1.111789699035306000e+00
5.802855510493564850e+01_1.007604562737642873e-01_9.635635810304421511e-01
5.869604170595127357e+01_4.752851711026595538e-02_6.674866010156250695e-01
5.965941765898422489e+01_1.901140684410940196e-03_9.633759530329513154e-01
6.077017540403342366e+01_-1.711026615969579723e-02_1.110757745049198775e+00
6.195420188209644152e+01_-5.703422053231932409e-03_1.184026478063017862e+00
6.336037990916928209e+01_1.901140684410940196e-03_1.406178027072840564e+00
6.447085621222225882e+01_-5.703422053231932409e-03_1.110476303052976732e+00
6.558114488727778735e+01_-5.703422053231932409e-03_1.110288675055528529e+00
6.661750813132832150e+01_-9.505703422052924623e-03_1.036363244050534149e+00
6.772789062038259544e+01_-1.330798479087436093e-02_1.110382489054273947e+00
6.861584011843079622e+01_-1.901140684410496107e-03_8.879494980482007804e-01
6.965210954848259917e+01_-1.901140684410496107e-03_1.036269430051802942e+00
7.083651128254055607e+01_-5.703422053231932409e-03_1.184401734057956901e+00
7.172474222258496468e+01_-5.703422053231932409e-03_8.882309400444086123e-01
7.268699240763307046e+01_-5.703422053231932409e-03_9.622501850481057772e-01
7.372326183768488761e+01_-5.703422053231932409e-03_1.036269430051817153e+00
7.490766357174284451e+01_-9.505703422052924623e-03_1.184401734057956901e+00
7.601776461880088220e+01_-1.901140684410496107e-03_1.110101047058037693e+00

```
7.720226016685755610e+01_-9.505703422052924623e-03_1.184495548056673897e+00
7.846058733192047896e+01_-9.505703422052924623e-03_1.258327165062922859e+00
7.971872686898593940e+01_-1.901140684410496107e-03_1.258139537065460445e+00
8.082901554404143951e+01_-1.901140684410496107e-03_1.110288675055500107e+00
8.201341727809939641e+01_-5.703422053231932409e-03_1.184401734057956901e+00
8.275417261212884057e+01_-2.851711026615966205e-02_7.407553340294441568e-01
8.371689186717063080e+01_-4.752851711026639947e-02_9.627192550417902339e-01
8.497521903223353945e+01_-4.752851711026639947e-02_1.258327165062908648e+00
8.593737540328291402e+01_-4.372623574144496317e-02_9.621563710493745702e-01
8.697336339133852334e+01_-3.231939163498109835e-02_1.035987988055609321e+00
8.793523832039167587e+01_-1.711026615969579723e-02_9.618749290531525276e-01
8.934132253346578523e+01_-5.703422053231932409e-03_1.406084213074109357e+00
9.059964969852869388e+01_-5.703422053231932409e-03_1.258327165062908648e+00
9.163601294257924224e+01_-9.505703422052924623e-03_1.036363244050548360e+00
9.274630161763475655e+01_-9.505703422052924623e-03_1.110288675055514318e+00
9.422659270371005391e+01_-5.703422053231932409e-03_1.480291086075297358e+00
9.563295835878037110e+01_-5.703422053231932409e-03_1.406365655070317189e+00
9.666922778883218825e+01_-5.703422053231932409e-03_1.036269430051817153e+00
9.822363193390989977e+01_-5.703422053231932409e-03_1.554404145077711519e+00
9.925999517796046234e+01_-9.505703422052924623e-03_1.036363244050562571e+00
1.000000000000000000e+02_-9.505703422052924623e-03_7.401924500369858606e-01
```