

Supplementary Materials for  
**OpenExo: An open-source modular exoskeleton to augment human function**

Jack R. Williams *et al.*

Corresponding author: Jack R. Williams, [jack.williams@nau.edu](mailto:jack.williams@nau.edu); Zachary F. Lerner, [zachary.lerner@nau.edu](mailto:zachary.lerner@nau.edu)

*Sci. Robot.* **10**, eadt1591 (2025)  
DOI: 10.1126/scirobotics.adt1591

**The PDF file includes:**

Materials and Methods  
Figs. S1 to S12  
Table S1

**Other Supplementary Material for this manuscript includes the following:**

MDAR Reproducibility Checklist

## **SUPPLEMENTAL MATERIALS AND METHODS**

### **Electrical Architecture: Microcontrollers, Actuators, Sensors, and Printed Circuit Board**

#### ***Microcontrollers***

We developed a simplified electrical architecture, which featured unused or networked electrical connections to support multiple potential joint configurations. At the center of this architecture are two microcontrollers: a Teensy 4.1 development board which was used to perform tasks such as data acquisition, control, and actuator communication, and an Arduino Nano BLE 33 which was responsible for communication with a companion application via Bluetooth Low Energy (BLE). Two microcontrollers were selected, rather than one, to increase device speed. The main goal was to find one microcontroller for exoskeleton control and one for Bluetooth communication. The Teensy was selected for exoskeleton control due to its fast clock and good analog-to-digital converter, whereas the Nano was chosen for its Bluetooth capabilities. Both microcontrollers can directly interface with Arduino to facilitate easy usage of the developed software system.

#### ***Actuators***

We designed this system around CubeMars' AK-series motors (Nanchang, China) which are powerful enough to provide torques for larger adult participants and during more challenging tasks such as multi-terrain walking (stairs, inclines, declines...). Our electrical architecture, software, and hardware were all designed to be able to interface with different versions of these motors to provide the researcher with maximum flexibility. All real-time communication with these motors is performed using the Controller Area Network (CAN) protocol, which is a bus-based communication protocol that relies on differential signaling for noise rejection. It should be noted that our software system is robust enough to integrate different motors other than the CAN-based ones utilized currently. We have developed simple guidelines for users to follow if interested in adding a new motor to the system.

#### ***Sensors***

The system was designed to allow for multiple, independent, sensors. Commonly used sensors for our devices include a proprietary, strain-based, torque transducer (29) to estimate applied torque at a joint to facilitate low-level closed-loop control, force sensitive

resistors (FSRs) placed around the ball of the foot and heel to enable high-level control, and hall-effect magnetic angle encoders to estimate joint position and velocity. To account for variance in how the device fits users, which will influence the signals and estimates from each of these sensors, we developed range-based calibration protocols to be employed after donning the device to ensure consistency. Each sensor is available to use but not a required component for the designed architecture to function properly. That is, if no angle sensor is present, that wouldn't prevent the device from functioning as long as the high-level control did not require it. A step-by-step guide is available as part of our software structure should researchers have interest in adding additional sensors to their device.

### ***Printed Circuit Board***

We designed a printed circuit board (PCB) that was used to power and communicate with the actuators and to condition analog signals from external sensors while directly interfacing with the microcontrollers (**Fig. S2**). This included ports to power and communicate with up to four actuators at once (allowing for bilateral multi-joint applications), as well as pins to interface with multiple sensors (simultaneous usage of bilateral torque sensors and FSRs). Documentation breaking down the components of the PCB and guides to modifying the software to incorporate new PCBs are included to help facilitate modification of the system to meet a user's needs.

### **Software: A Modular Code Base for System Flexibility**

#### ***Overview***

The chief source of modularity in our open-source system comes from the software architecture. Our software was developed using a combination of C++ and Arduino languages, with the principle of inheritance-based polymorphism (realized through parent-child classes and abstract classes) used to achieve a high-degree of modularity and reduce redundancy. **Fig. S1** outlines the structure of the software. We designed the software using a hierarchical structure in order to have a logical and easy to follow form. We also used a parallel structure to facilitate the dual usage of microcontrollers. Thus, there were essentially two components to the code: the main classes for handling computation (Exo, Side, Joint, Controller) and classes designed to store the data and parameters used in control of the system (ExoData, SideData, JointData, ControllerData). The structure of these two components mirrored each other with the data classes (ExoData) inheriting their interfaces from their computational (Exo) counterparts via abstract classes. This was done so that the classes responsible for the data could be mirrored onto the Arduino Nano microcontroller in order to interface with the Bluetooth communication without the need to mirror the components of the software used for the primary exoskeleton control (thus improving operating speed).

#### ***Structure***

The exoskeleton device was designed to interface with a companion python application (**Fig. 1**). This device communicates with the system through the Arduino Nano microcontroller via BLE. This microcontroller handles all the Bluetooth communication of the device and contains a mirrored copy of the exoskeleton data which originates from the Teensy microcontroller, with data communication between the two microcontrollers occurring via universal asynchronous receiver/transmitter (UART). The Teensy microcontroller handles most of the computation that occurs within the system. This includes configuring the device via user defined parameters stored in a Secure Digital (SD) card that slots into the Teensy, reading the primary sensors associated with the system,

computing the primary outputs of the system (motor command), and communicating the primary outputs with the relevant hardware (the motors).

### ***SD Card***

An SD card is used to store information regarding the device configuration as well as to store default values for the parameters associated with the controllers for each joint configuration. This SD card is slotted into the Teensy and the software operating on the microcontroller then accesses the stored data to get information about the desired configuration of the device. This includes PCB information (version information which changes pin readouts associated with both microcontrollers), which joints are used (hips, ankles, hips and ankles, knee, elbow), the sides used (unilateral or bilateral), transmission gearing ratios associated with each joint, default controller information (which controller should be default for the given configuration), sensor usage information, and directionality information (whether the calculated motor commands should be flipped to match a specific direction). Additionally, parameter information for each controller for each joint is stored on the SD card; this may include information such as max torque setpoint for a specific controller, when to begin torque onset, and proportional-integral-derivative (PID) gains for closed-loop control. These parameters are the default settings for the specified controller and can be accessed and changed in real-time via the companion application. Step-by-step guides are available to assist users in adding new joint configurations or controllers/controller parameters to the software to help facilitate researcher usage.

### ***A Python GUI for Exoskeleton Control***

A companion python application was developed to facilitate easy usage of the exoskeleton device. Core functions of this software include connecting to the device, calibrating the sensors, starting an active trial, updating controller parameters in real-time, and plotting. Upon the completion of the trial, data from up to ten variables of interest can be saved and exported as a .csv file to enable researchers to analyze the data and verify the accuracy of their intended control paradigm. Like the base software for the exoskeleton, this companion application is open-source and available for researchers to download and modify to match their needs.

## **Hardware and Control**

### ***Hip Exoskeleton***

*Control:* Our primary mode of control for this hip device was a variation of a control scheme developed by Bryan et al. (27) and Franks et al. (28) (**Fig. S10A**). Briefly, heel and toe FSRs were used to estimate the gait cycle of the user. This was done by measuring the time duration from successive heel-strikes. Then, the gait phase was estimated by calculating the current time from the last heel-strike relative to the expected duration based on the average of the previous three steps. Users could then prescribe hip torques based on the percentage of the gait cycle. As outlined in Bryan et al. (27) and Franks et al. (28), the starting point of hip assistance was shifted to 84% of the gait cycle to prevent discontinuities related to the need for hip extension assistance during heel strike. Based on this, the user could define several controller parameters such as the magnitude and percent gait cycle of peak flexion and extension torques as well as the rise/fall time, in terms of percent gait cycle, to reach those peaks. These points were connected using splines to facilitate a smooth transition during assistance. For timing-based measures (percent gait where peaks occurred and the rise/fall times to get there), we elected to use the average parameter values reported by Franks et al. (28) as they found that optimized torque timing was relatively consistent between subjects. Thus, for the purposes of this study, only flexion and extension torque

magnitudes were varied on a subject-by-subject basis. This controller, and others, are available for use as part of our software architecture.

### ***Ankle Exoskeleton***

*End-Effector Hardware:* Our untethered, bi-directional sagittal-plane ankle configuration was similar in design to our previously reported ankle device (29) (**Fig. S3B**). Briefly, motors mounted at the waist actuated a pulley located at the user's ankle via a cable-chain interface which, in-turn, created sagittal plane movement (either plantarflexion or dorsiflexion) for a carbon fiber footplate that could slip into any user's shoes. The ankle end-effector assembly portion of the device was identical to the previously reported device. To summarize, a single-degree of freedom (sagittal-plane rotation) pulley interacted with a user via carbon-fiber footplates and custom-designed calf cuffs. To facilitate a variety of user sizes, multiple sets of footplate and calf cuff sizes were created with the ability to quickly be swapped out with the rest of the device assembly. This pulley was actuated by steel cables driven by the motors located at the waist. In-line with the pulley were custom-designed torque and angle sensors which were utilized for low- and high-level control schemes, respectively. For more information on the torque and angle sensors and their validation, we point you to (29). The device was designed to minimize distal mass and lateral protrusion in order to reduce the metabolic burden and potential for out-of-plane bending moments/environmental collisions, respectively. To facilitate high-level control schemes, FSRs were placed on each footplate near the ball of the foot (under the distal head of the first metatarsal).

*Control:* The ankle configuration was controlled using a previously described proportional joint moment controller (PJMC) developed by our research group (29, 30) (**Fig. S10B**). Briefly, we used FSRs placed at the forefoot to detect the stance and swing phase transitions of gait. During stance, the controller generates an adaptive plantarflexor torque based on a real-time estimate of the biological sagittal plane ankle moment. Users specify a setpoint to limit the maximum torque output and the adaptive plantarflexor torque is scaled based on this setpoint. This results in an assistive torque profile that follows the shape of the user's ankle joint moment but with a torque magnitude that is typically only a percentage of that moment. While in swing, the user has the option to prescribe a constant dorsiflexion torque to aid in toe clearance. Throughout, a low-level proportional-derivative (PD) controller is used to facilitate closed-loop control to ensure the prescribed torques matched those measured from a torque transducer placed in-line with the sagittal plane of the ankle joint. The gains of the P and D terms were determined via manual tuning while a user walked on a treadmill, receiving exoskeleton assistance (P:28, D: 200).

### ***Hip-and-Ankle Device***

The hip and ankle devices were combined into one system to provide multi-joint actuation (**Fig. S3C**). The footplates of the ankle device were modified to include a heel FSR in addition to the toe FSRs already embedded within the plate. The same control schemes used in the single-joint configurations were utilized to provide multi-joint assistance.

## **Engineering Validation**

### ***Hip Configuration***

*Benchmark Testing:* To characterize the responsiveness of the direct-drive hip configuration, a benchtop step-response test was performed. To confirm the accuracy of the device's response, the design of the hip was modified to include an in-line torque transducer between the waist-mounted motor and the carbon fiber upright (**Fig. S6A**). This torque transducer was only used to verify the torque being produced at the motor and was not used for any

low-level control approaches (PID control); that is, all tests performed with this modified design were under open-loop control. The mounting plate containing the hip assembly was removed from the waist belts and secured to a table via external clamps. The carbon-fiber upright of the assembly was also secured to the table via a vice to prevent movement (**Fig. S6B**). A step response controller was developed in the previously described software and allowed users to specify the amplitude, duration, and spacing of the step commands applied by the system. For benchtop testing, the peak amplitude of the step command was set to six Nm (0.5 Nm higher than the maximum assisted torque from the experimental validation). A total of five commands were sent, each lasting two seconds with two seconds of spacing between each command. The duration, prescribed magnitude, and the measured torque were recorded for each step command and averaged across all five steps. From this data, the rise time, defined as the time it took for the measured torque to go from 10% of the prescribed torque to 90% of the prescribed torque, and the percent overshoot/undershoot, defined as the difference between the average measured torque at the peak of the step and the prescribed torque, were determined. Residual motor vibration while applying high torques to a rigidly fixed system caused a high degree of low-magnitude oscillation when at the maximum torque setpoint for this test, resulting in unrealistic overshoots compared to how the device would respond when worn by a user. To reduce this noise and more accurately capture the performance of the system when worn by a user, a real-time exponentially weighted moving average was applied to the measured torque once it reached its maximum torque setpoint to help smooth out this noise. Importantly, this had no influence on the calculation of the rise time (due to the lack of applied filter before reaching the setpoint) and minimal influence on the average overshoot of the torque setpoint (as the noise was relatively uniform around this mean).

*Torque Tracking:* To characterize the accuracy of the previously described hip controller, the modified hip design featuring the in-line torque transducer was used during walking to record the measured torque experienced by the user. Despite the presence of the torque transducer, the user walked with open loop control to provide the most accurate representation of how the hip device would function (without the in-line transducer). That is, the transducer was only present to verify the accuracy of the prescribed open loop controller rather than to help influence the control. The individual walked at the most commonly reported maximum flexion and extension torque magnitudes from our experimental validation (Flexion: 3.4 Nm; Extension: 4.1 Nm). While they walked, the estimated percent gait cycle, the prescribed torque, and the measured torque were recorded. The user walked in the device for two minutes and the averages and standard deviations of the three measures of interest were determined over the course of the trial. The accuracy of the prescribed controller was characterized by the average root mean squared error (RMSE) between the prescribed and measured torques.

*Duration Testing:* A duration test was performed with a user walking at their maximum comfortable hip assistance (Flexion: 5 Nm, Extension: 5.5 Nm, Treadmill speed:  $1.25 \text{ ms}^{-1}$ ; as determined from experimental validation). As they walked the motor temperature for both the left and right motors and the total battery voltage (from the 22.2v, 1800 mAh LiPo battery) was recorded every minute via an infrared thermometer (Etekcity Lasergrasp 1080) and LiPo Battery Voltage Tester, respectively. The test was concluded once the motors shut off due to high temperature or once the battery voltage reached the manufacturer's minimum voltage recommendation ( $3.7 \text{ V/cell} = 22.2 \text{ V total}$ ).

*Mechanical Power Characterization:* To further characterize the capabilities of the device while configured for hip assistance, the power delivered by the device to the user was calculated during level ground and incline walking. Before testing, the device was modified

to incorporate the in-line torque transducer described previously and the software was modified to record the angular velocity, in radians per second, from the motor's internal encoder. Data were recorded for one-minute of hip assistance and the average power, normalized to 0-100% of the gait cycle, was calculated by multiplying the angular velocity with the measured torque. During level ground treadmill walking at the user's comfortable, open-loop, hip assistance level (Flexion: 5 Nm, Extension: 5.5 Nm, Treadmill speed: 1.25 ms<sup>-1</sup>; as determined from experimental validation), the device provided a peak power of 8.38 W to the user (**Fig. S11A**, Normalized: 0.08 W/kg, Participant Mass: 101.3 kg). When the user performed 7.5° treadmill walking (Treadmill speed: 1.00 ms<sup>-1</sup>) with the same assistance, the device provided a peak power of 9.15 W (Normalized: 0.09 W/kg). It should be noted that these power capabilities are not the maximum capabilities of this configuration, but the peak power delivered while walking with the maximum comfortable torque encountered experimentally with this specific controller. To illustrate this point, we had the same user repeat each condition with an extra 5 Nm of torque beyond the maximum self-selected comfortable level determined experimentally (Flexion: 10 Nm, Extension 10.5 Nm). Under this level of assistance, the device provided a peak power of 20.47 W (**Fig. S11B**, Normalized: 0.20 W/kg) during level walking (Treadmill speed: 1.25 ms<sup>-1</sup>) and 15.85 W (Normalized: 0.16 W/kg) during incline walking (Treadmill speed: 1.00 ms<sup>-1</sup>).

### ***Ankle Configuration***

*Benchtop Testing:* Like the hip configuration, we tested the responsiveness of the ankle configuration via benchtop step-response testing. This consisted of having a user don one of the legs of the exoskeleton device and having the footplate and carbon fiber upright firmly fixed to their lower leg while the user was seated with their leg bent 90° at the knee. The same step response controller described previously was then applied with the maximum torque amplitude set to 28 Nm (the experimentally determined maximum comfortable plantarflexion torque encountered during experimental validation testing). Unlike hip testing, a low-level PD control was utilized during this test to capture the responsiveness of the system based on the closed-loop control scheme utilized for the ankle exoskeleton controller. The P and D gains were set to the same values as those determined via experimental tuning (P: 28, D: 200). Like the hip, five step response commands were performed while recording the duration, prescribed magnitude, and the measured torque of each step. Once the average prescribed magnitude and measured torque were determined across all five step responses, the rise time and overshoot/undershoot response of the system were calculated as described previously.

*Torque Tracking:* The accuracy of the prescribed ankle controller was characterized by evaluating the average root mean squared error between the prescribed and measured torques from two minutes of walking with closed loop ankle plantarflexion assistance (magnitude: 28 Nm).

*Duration Testing:* The operation length of the ankle configuration was examined while a user walked on a fixed-speed treadmill (1.25 ms<sup>-1</sup>) with maximum comfortable bilateral ankle plantarflexion assistance (28 Nm). As the user walked, the motor temperatures and battery voltage (from the 22.2v, 1800 mAh LiPo battery) were monitored on a minute-by-minute basis. The test lasted until one of the previously described criteria was met (motor temperatures or battery voltage reaching manufacturer's recommended limits).

*Mechanical Power Characterization:* Like the hip configuration, we further characterized the capabilities of the ankle configuration by assessing power delivered by the device during level ground treadmill walking (Treadmill speed: 1.25 ms<sup>-1</sup>). To calculate angular velocity during walking, the device was modified by mounting a magnetic rotary position sensor

module (AS5600; ams OSRAM) onto the exoskeleton's carbon fiber upright in-line with the center of the pulley (**Fig. S12**). A diametrical magnet was rotated in-line with the pulley's axis of rotation, changing its magnetic field, which was then detected by this sensor. The relative rotation of the magnet was expressed as a voltage drop across the OUT and GND pins of the module (Range: 0 to 3.3 V) which was then scaled to the device's range of motion ( $\sim 102^\circ$ ) to provide an estimate of the angle (in degrees). From there we could then estimate angular velocity, in radians per second. All data were recorded for one-minute of assistance and were normalized to 0-100% of the gait cycle.

While walking with closed-loop ankle assistance at their comfortable torque setpoint (Plantarflexion: 28 Nm), the device provided a peak power of 116.43 W (**Fig. S11C**, Normalized: 1.68 W/kg, Participant Mass: 69.4 kg) to the user. Like the hip configuration, this peak power is not necessarily reflective of the maximum capabilities of this configuration as power is a function of the participant's self-selected setpoint (which was the maximum encountered during experimental validation) and the mechanics of the user's movement (ankle angular velocity).

### ***Hip-&-Ankle Configuration***

*Duration Testing:* Like the hip and ankle configurations individually, the length of operation of the combined hip-and-ankle configuration was tested while a user walked on a fixed-speed treadmill ( $1.25 \text{ ms}^{-1}$ ) while receiving maximum comfortable joint assistance at both the hip and ankle joints (Hip Flexion: 5 Nm, Hip Extension: 5.5 Nm, Ankle Plantarflexion: 28 Nm). As they walked the motor temperatures for both joints of both sides and the battery voltage were recorded on a minute-by-minute basis until one of the previously described conditions for trial termination were met.

### ***Elbow Configuration***

Complete information on the methodology used to validate the elbow configuration can be found in Colley et al. (26). The sections below provide a brief description of the methodologies utilized.

*Benchtop Testing:* Like the hip and ankle configurations, we tested the responsiveness of the elbow configuration via benchtop step-response testing. This consisted of locking the forearm and upper arm carbon fiber uprights to a table via vices. The same step response controller described previously was then applied with the maximum torque amplitude set to 10 Nm. A low-level PID control was utilized during this test to capture the responsiveness of the system based on the closed-loop control scheme utilized for the elbow exoskeleton controller. The P, I, and D gains were set to the same values as those determined via experimental tuning (P: 7, I: 25, D: 47). Eight step response commands were performed while recording the duration, prescribed magnitude, and the measured torque of each step. Once the average prescribed magnitude and measured torque were determined across all eight step responses, the rise time and overshoot/undershoot response of the system were calculated as described previously.

*Torque Tracking:* The accuracy of the prescribed elbow controller was characterized by evaluating the average root mean squared error between the prescribed and measured torques from thirteen consecutive 10 kg box lifting motions during closed-loop elbow assistance (magnitude: 12 Nm).

## **Experimental Validation**

### ***Hip Configuration***



To test the utility of the device while configured for hip assistance, we had two individuals (**Table S1**; P1 & P2) walk on a treadmill inclined to 7.5 degrees. Prior to testing, both participants performed 60 minutes of hip assisted walking to acclimate to the device. This consisted of 30 minutes of level treadmill walking at a self-selected speed, and 30 minutes of incline treadmill walking (7.5°) at a self-selected speed that was subsequently maintained for testing. During acclimation, assistance in the extension and flexion directions were tuned based on participant comfort. Acclimation walks occurred the day prior to actual testing. On the test day, participants performed incline treadmill walking while outfitted with a portable, indirect calorimetry metabolic unit (K5, COSMED). Testing order was as follows: five minutes of quiet standing for a metabolic baseline, eight minutes of incline treadmill walking without the device (“shod”), fifteen minutes of rest, five minutes of quiet standing for a metabolic baseline, eight minutes of incline treadmill walking with hip assistance. Oxygen and carbon dioxide volumes during the last two minutes of treadmill incline walking were used to calculate steady-state metabolic power using Brockway’s equation (32). The respiratory data from quiet standing prior to each trial was used to calculate a baseline metabolic rate. The cost of transport (COT) for each trial was then calculated by offsetting the steady-state walking data by the baseline metabolic rate and normalizing it by body mass and walking speed.

### ***Ankle Configuration***

One individual (**Table S1**; P3), with prior ankle exoskeleton usage experience, completed ankle-assisted walking on outdoor terrain. This involved walking with, and without, the ankle exoskeleton about a relatively flat 1650 m loop at a local park (**Fig. 5A and Fig. 5B**). Prior to testing, the individual completed one lap of the loop without exoskeleton assistance to familiarize them with the route. Ankle exoskeleton assistance was set to 30% of the user’s bodyweight (28 Nm). The testing order was as follows: one lap with exoskeleton assistance, twenty minutes of rest, one lap without wearing the exoskeleton device (“shod”). This order was selected as the device had mechanical failures during preliminary pilot testing and we wanted to avoid having the user waste time if a repeated failure were to occur during the actual test (it did not). The time to complete the loop and the number of steps taken with the right leg were recorded during testing. From these, we calculated the average walking speed of the user (dividing the length of the course by the time it took to complete it) and the average step length of the user (dividing the length of the course by the number of steps taken by the user). Metabolic data (oxygen and carbon dioxide volumes) over the course of each loop were collected; however, the data were collected in a way that made direct comparisons between conditions challenging (lack of distance markers for standardization, failure to record for a few minutes after completing the loop to account for the delay in metabolic response) and so this data was excluded as an outcome variable. This metabolic data has been made available for sake of transparency.

In addition to completing outdoor testing, this same participant completed metabolic testing with and without the exoskeleton device during level indoor treadmill walking. Prior to testing, the participant performed 60 minutes of ankle assisted treadmill walking to ensure acclimation to the device. This consisted of three, 20-minute sessions walking on the treadmill at 1.25 ms<sup>-1</sup>. Acclimation walks occurred the day prior to actual testing. On the test day, the participant performed level treadmill walking while outfitted with a portable, indirect calorimetry metabolic unit (K5, COSMED). Testing order was as follows: five minutes of quiet standing for a metabolic baseline, eight minutes of level treadmill walking without the device (“shod”), fifteen minutes of rest, five minutes of quiet standing for a metabolic baseline, eight minutes of level treadmill walking with ankle plantarflexion assistance. Oxygen and carbon dioxide volumes during the last two minutes of treadmill

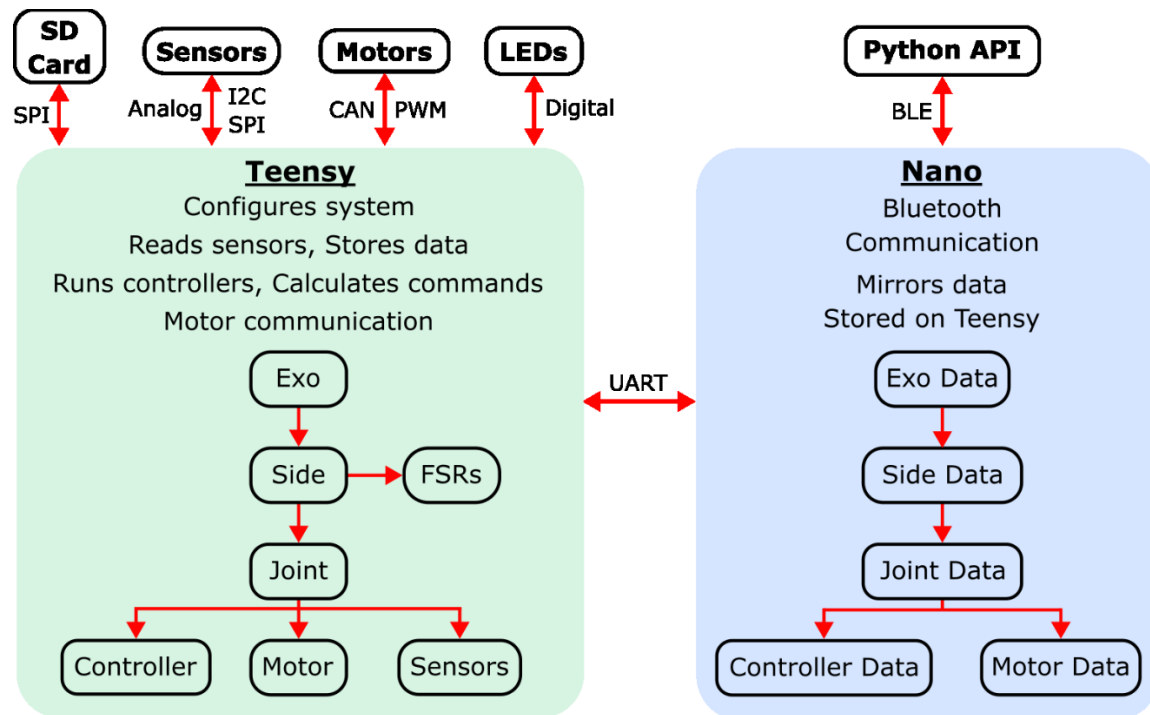
walking were used to calculate steady-state metabolic power using Brockway's equation (32). The respiratory data from quiet standing prior to each trial was used to calculate a baseline metabolic rate. The COT for each trial was then calculated by offsetting the steady-state walking data by the baseline metabolic rate and normalizing it by body mass and walking speed.

### ***Hip-and-Ankle Configuration***

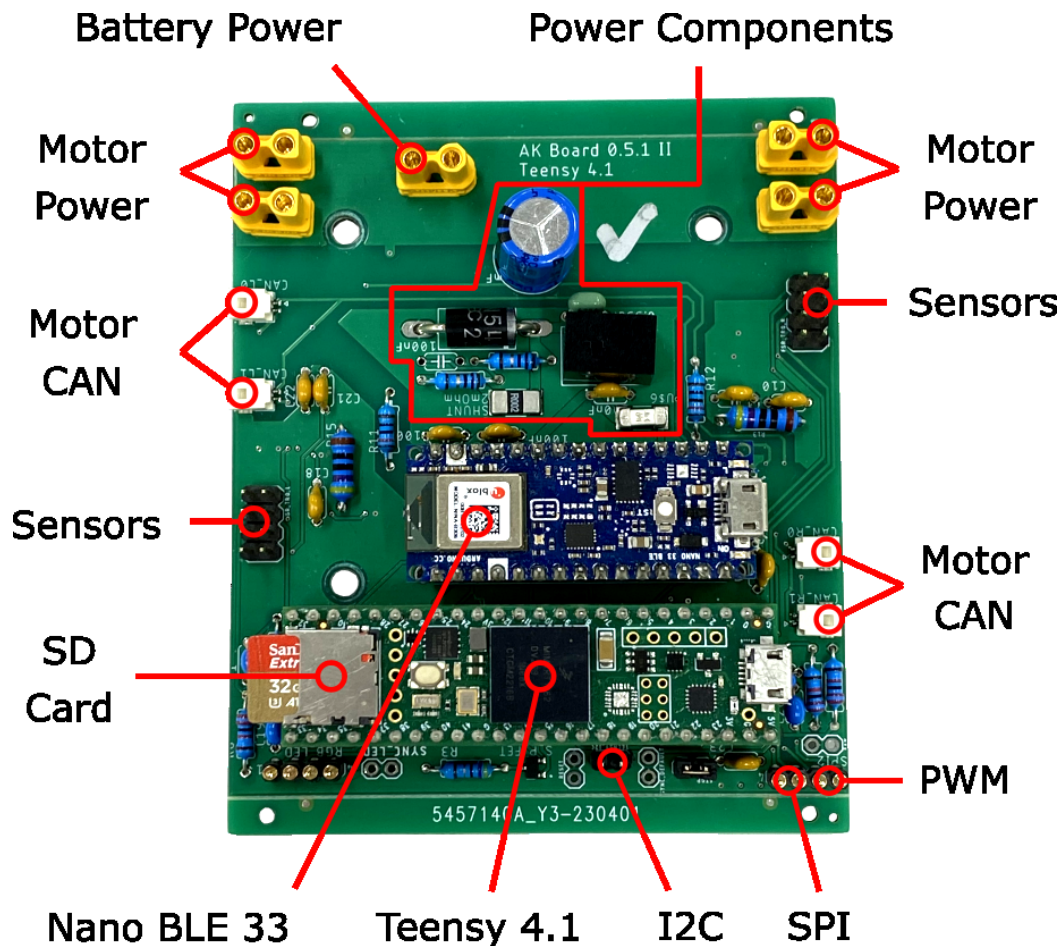
To test the utility of the device while configured for simultaneous hip-and-ankle assistance, we had two individuals (**Table S1**; P4 & P5) walk on a treadmill while carrying a moderate sized load (22.5 %BW, weighted vest). Prior to testing, both participants performed 60 minutes of hip-and-ankle assisted walking to acclimate to the device. This consisted of 30 minutes of level treadmill walking without additional load carriage and 30 minutes of treadmill walking with the additional load. During acclimation, assistance in the extension and flexion directions of the hip and in the plantarflexion direction of the ankle were tuned based on participant comfort. Acclimation walks occurred the day prior to actual testing. Walking speed was held at  $1.25 \text{ ms}^{-1}$  on both the acclimation and testing visits. On the test day, participants performed load-carriage treadmill walking while outfitted with a portable, indirect calorimetry metabolic unit (K5, COSMED). Testing order was as follows: five minutes of quiet standing for a metabolic baseline, eight minutes of load-carriage treadmill walking with hip-and-ankle assistance, fifteen minutes rest, five minutes of standing metabolic baseline, eight minutes of load-carriage treadmill walking without the device ("shod"). Oxygen and carbon dioxide volumes during the last two minutes of treadmill walking were used to calculate steady-state metabolic power using Brockway's equation (32). The respiratory data from quiet standing prior to each trial was used to calculate a baseline metabolic rate. The COT for each trial was then calculated by offsetting the steady-state walking data by the baseline metabolic rate and normalizing it by body mass and walking speed.

### ***Elbow Configuration***

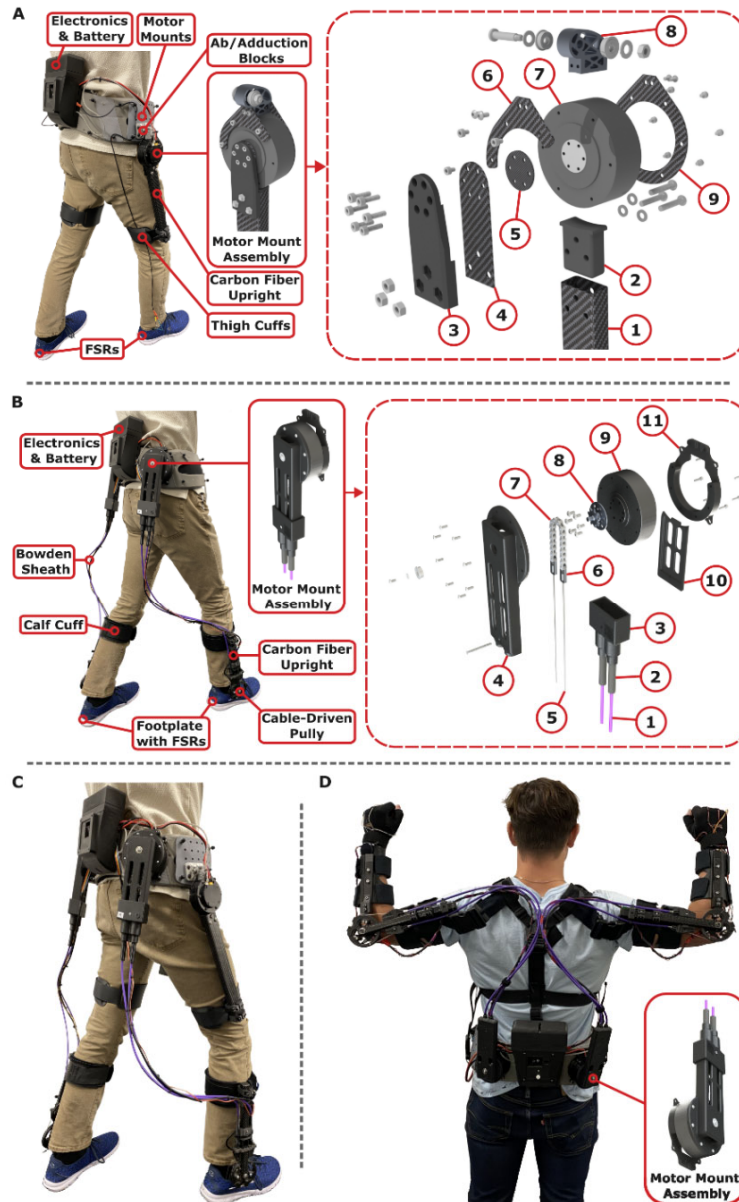
To test the utility of the device while configured for elbow flexion assistance, we had two individuals (**Table S1**; P6 & P7) perform weight curls with a 19.5 kg object until they reached fatigue and could not continue. The object was a wooden box with additional weights fixed to the inside (**Fig. 6A**). A metronome set to sixty beats per minute was used to prompt users to raise and lower the weight (leading to a full cycle every two seconds). A repetition was considered one complete cycle from the down position back to the down position. Testing order was randomized across both participants resulting in the following test orders: P6 - no exoskeleton, twenty-five minutes of rest, exoskeleton and P7 - exoskeleton, twenty-five minutes of rest, no exoskeleton. The number of repetitions completed for each condition was recorded as the primary outcome of interest. Prior to testing, surface electromyography (EMG) electrodes were placed over the muscle belly of the short head of the biceps brachii muscle of each user's dominate arm. Before performing fatigue lifting, both users performed a maximum voluntary isometric contraction (MVIC) by maximally flexing their forearm while holding a static strap with the elbow at  $90^\circ$ . This value was then used to normalize the EMG data collected during fatigue lifting. The EMG data was analyzed by applying a 4<sup>th</sup> order Butterworth band-pass filter (20-460 Hz), rectifying the signal, and then low-pass filtering the data with a 12 Hz cutoff. The EMG signal was normalized to each lifting cycle by identifying the troughs in the signal at the beginning and end of each cycle and then averaging across all the cycles for each condition (Exo, No Exo). Full information on the experimental testing protocol can be found in (26).



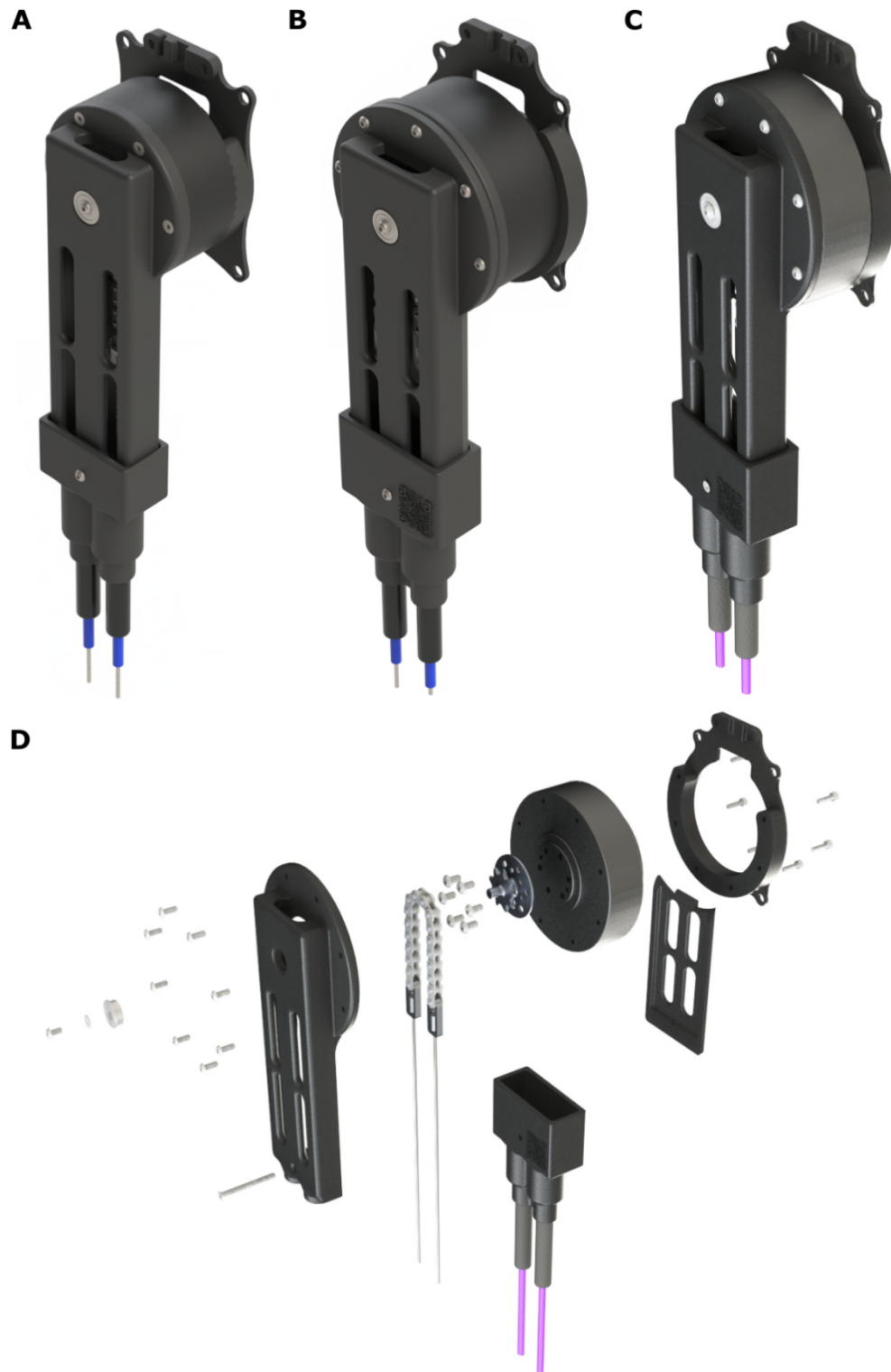
**Fig S1. Breakdown of the structure and function of the software package.** The software was designed to have a high degree of modularity while minimizing redundancy. It operates on two microcontrollers simultaneously (Teensy 4.1 and Arduino Nano BLE 33) to facilitate both computation and Bluetooth communication while maintaining a fast operation speed. Detailed guides on the software are available to aid researchers in effectively utilizing this resource to meet their needs. In addition, a companion python Bluetooth application is also available to help users operate and control the system in real-time.



**Fig S2. Breakdown of OpenExo's printed circuit board.** The printed circuit board supports both microcontrollers needed for device operation and has connections to facilitate operation of up to four individual joints at a time (two per each side of the board). This PCB is specifically designed to interface with CAN-based motors such as CubeMars' AK series. Notably, this setup has space for analog sensors ("Sensors") as well as pins to facilitate I2C and SPI communication protocols for additional sensor integration. A step-by-step guide to integrating new sensors and using I2C to do so is available as part of our documentation.

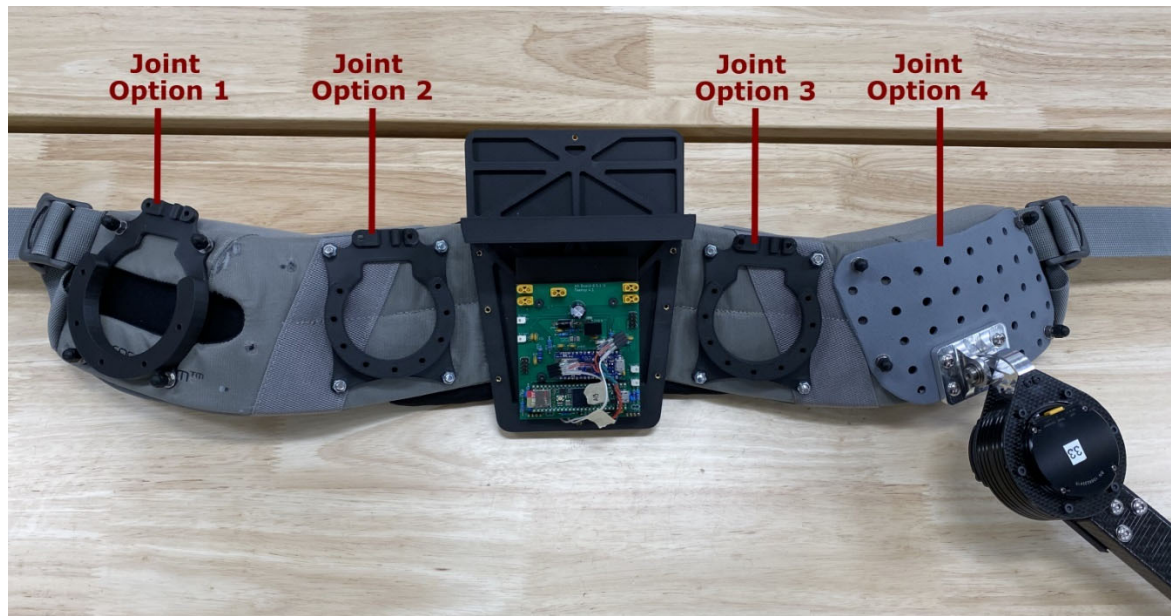


**Fig S3. Different exoskeleton configurations operating on OpenExo's software and electrical architecture.** (A) Direct-drive hip exoskeleton containing the following components: (1) carbon fiber upright, (2) 3D printed screw mount, (3) 3D printed shell for the motor-to-upright connector, (4) carbon fiber motor-to-upright connector, (5) carbon fiber motor space, (6) inner carbon fiber motor bracket, (7) AK60-6v1.1 motor, (8) aluminum abduction/adduction block, (9) outer carbon fiber motor bracket. (B) Bowden cable-based ankle exoskeleton containing the following components: (1) Bowden sheath, (2) TPU strain relief, (3) strain relief casing, (4) outer motor cartridge, (5) steel cables, (6) aluminum cable-chain interface, (7) chain, (8) aluminum sprocket, (9) AK80-9 motor, (10) back motor cartridge, (11) Motor-to-belt connector. (C) Combined hip-and-ankle exoskeleton, all hardware components are the same as A & B. (D) Elbow exoskeleton with the motor mount Bowden-cable transmission highlighted to emphasize the flexibility of the belt-to-motor interface design, which is just the inverted interface utilized for the ankles. The design of this transmission system is openly available for others to replicate and utilize. Configurations are combined on the belt and connected to the PCB secured in the Electronics and Battery housing in the middle of the waist belt.

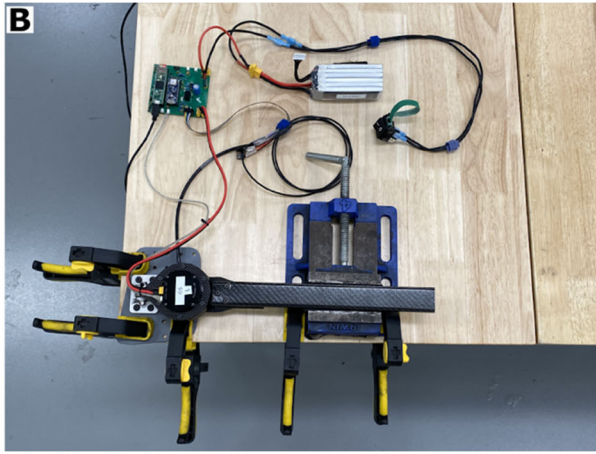


**Fig S4. Open-source transmission design.** (A) AK60-6v1.1 motor, (B) AK70-10 motor, (C) AK80-9 motor, (D) Exploded view of AK80-9 motor assembly. A full breakdown of the required parts and an assembly guide are available within OpenExo's documentation. The same transmission style has been designed to accommodate a variety of CubeMars AK-motors including the AK60v1.1, AK70, and AK80. The parts were designed to fit to the belt in the same manner, minimizing the amount of revision necessary to attach to the waist belt. The same belt-side transmission was used for both the ankle and elbow joints, with only the end-effector hardware differing between the two configurations.



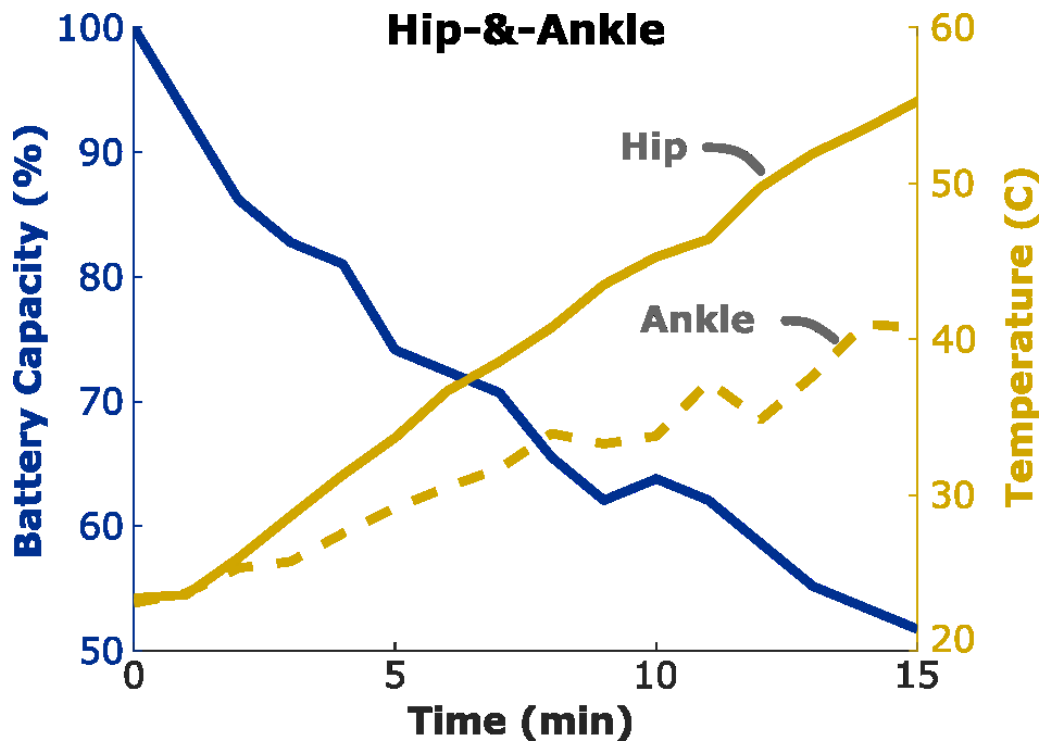


**Fig S5. Modularity of the waist belt design.** The waist belt contains an electronics and battery case containing the PCB (that can support the operation of four joints at a time) and the battery. The belt, however, is modular enough to be utilized in several configurations. The image above highlights how different orientations can be achieved on the belt. This can include bowden-transmission joints (joint options 1, 2, and 3) and direct-drive joints (joint option 4). Users have the flexibility to mix-and-match joints and transmission systems as desired. For example, if it was desired to do simultaneous bilateral operation of the hips and ankles (left and right hips and ankles) then the bowden-cable transmission of joint option 1 could be replaced with a direct drive transmission mirroring joint option 4. Plausible configurations with the established joints could include single joint assistance (hip, ankle, elbow) or multi-joint assistance (hip-ankle, hip-elbow, ankle-elbow).

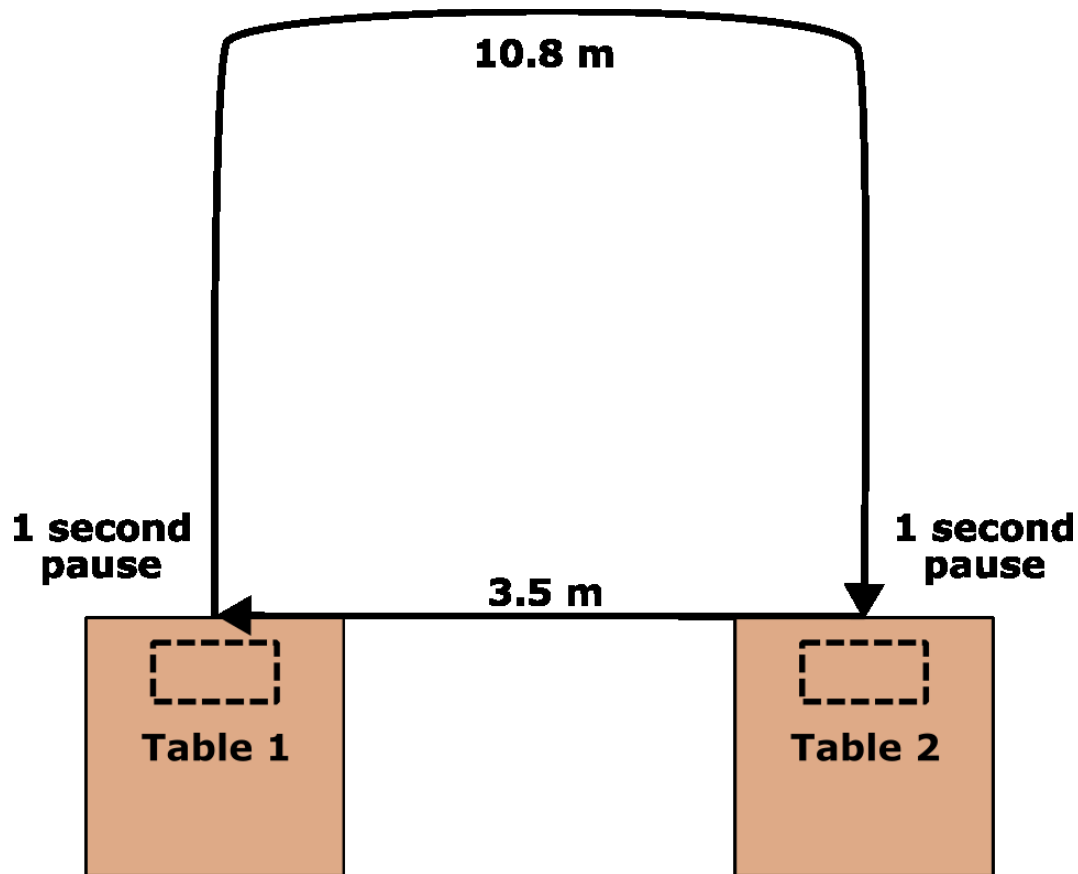
**A****B**

**Fig S6. Experimental setup utilized to perform the engineering validation of the hip configuration.** (A) Modified hip exoskeleton design incorporating a torque transducer. This modified design featured altered 3D printed components to link the motor to the carbon fiber upright while incorporating the torque sensor into these components. This version of the design was only used for the engineering validation work. (B) Image of benchtop testing setup. The motor-upright assembly was secured to a table via clamps and vices to prevent movement. The codebase was modified to include a new exoskeleton controller designed to apply step commands with user defined amplitudes, duration, and spacing (included as part of OpenExo's software). A series of open-loop step commands were sent to the motor while recording the measured torque.

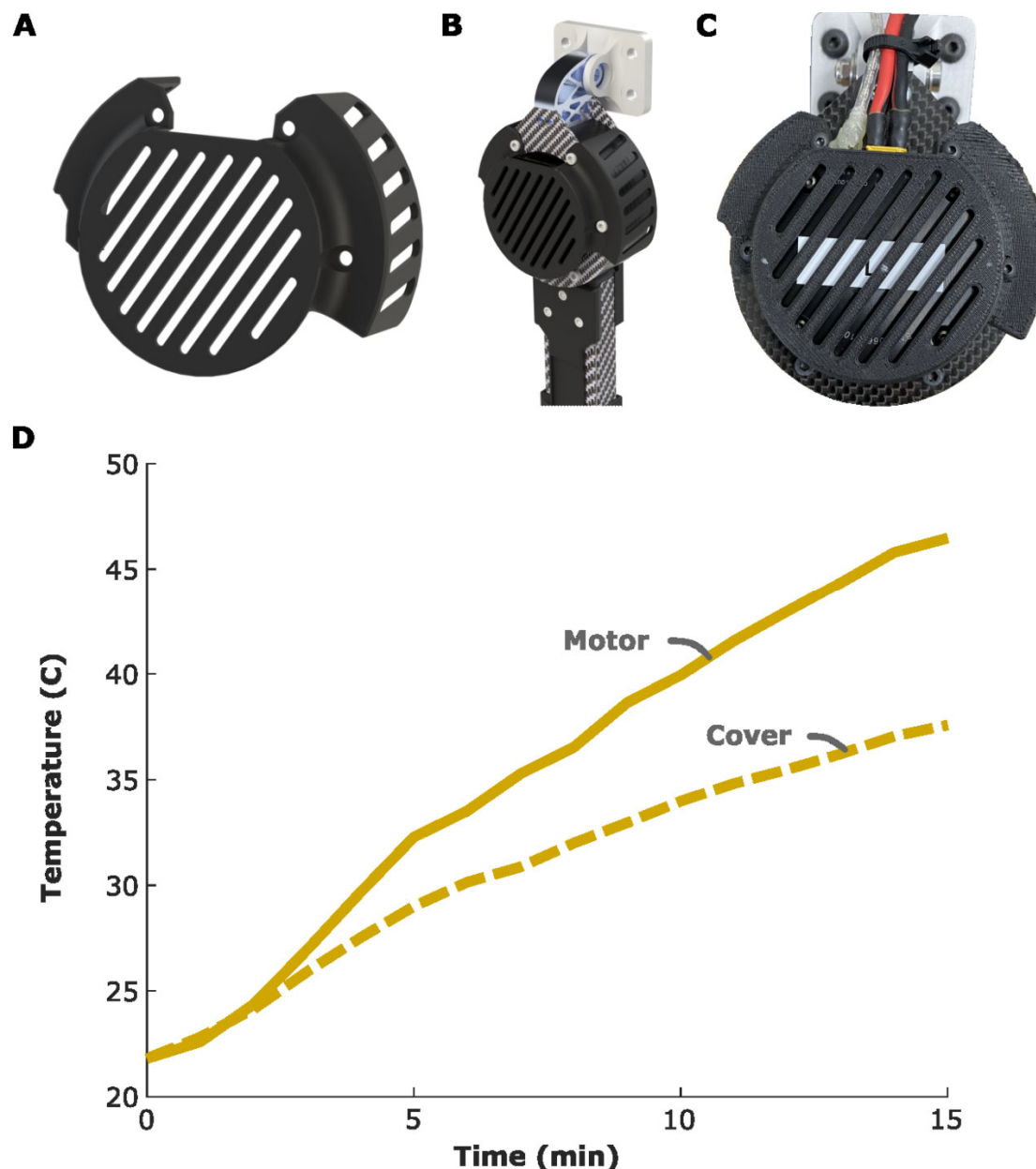




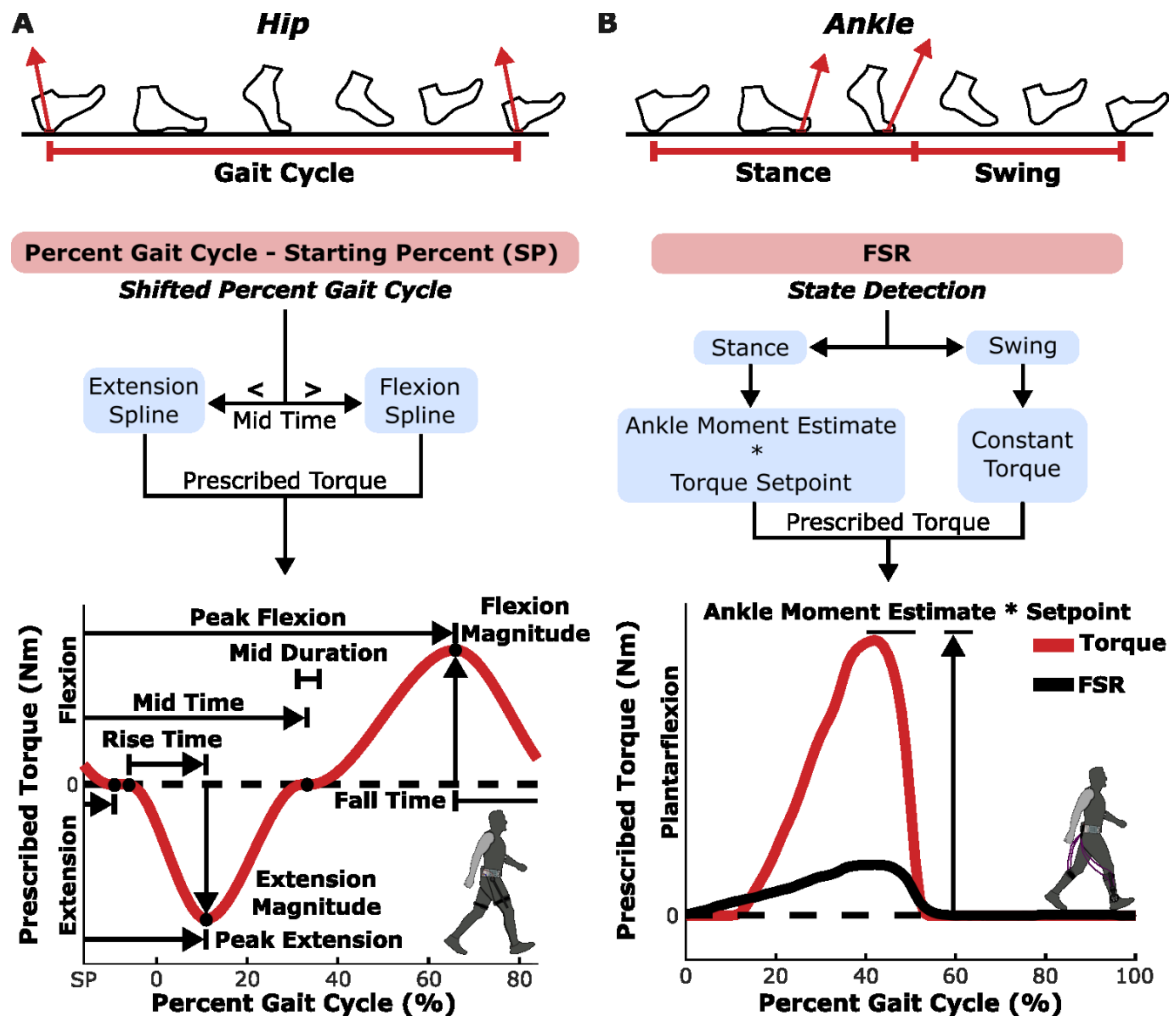
**Fig S7. Duration testing of the OpenExo under its hip-and-ankle configuration indicates operation lifespan is primarily limited by battery life.** Battery capacity (blue) and motor temperatures (gold) over the course of a duration test. The tests were ended when the battery voltage reached the manufacturer's recommended limit of 3.7 V/cell (22.2V total), which represents 50% of the battery's actual estimated capacity (3.2 V/cell = 19.2 V total) or when the motor temperature reached their shutoff temperature (100 °C). The user walked with 5 Nm of hip flexion and extension assistance and 28 Nm of ankle plantarflexion assistance at a treadmill speed of 1.25 ms<sup>-1</sup>. The combined hip-and-ankle configuration could operate for 15 minutes and was limited by the battery life, suggesting this could be a primary target to improve device operation lifespan moving forward.



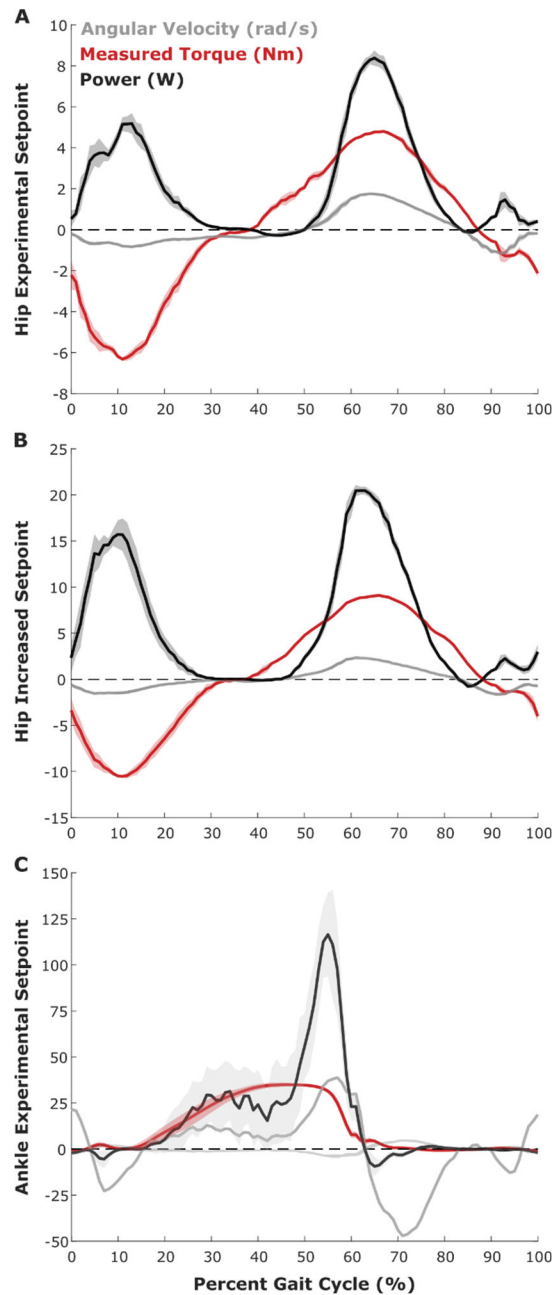
**Fig S8. Schematic of elbow duration testing course.** The user started at Table 1, where a 10 kg box was resting on the table. Using 12 Nm of elbow exoskeleton assistance the user then lifted the 10 kg box and then walked a 10.8 m loop around the room until reaching Table 2. Upon reaching Table 2, they set down the 10 kg box and waited a second before picking it back up and walking 3.5 m back to the original table where they set the box back down in its original position. A full cycle started with and ended with the 10 kg resting on Table 1. Every third cycle the participant walked around the course without carrying the 10 kg load, and thus without any exoskeleton assistance. The participant completed a 30-minute trial where battery voltage and motor temperature were recorded every two minutes. At the end of the 30 minutes the participant completed 122 loops with a final battery voltage of 23.8 V and motor temperature of 37.6 °C.



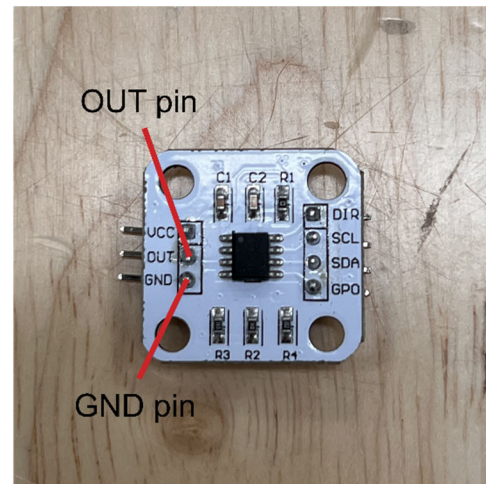
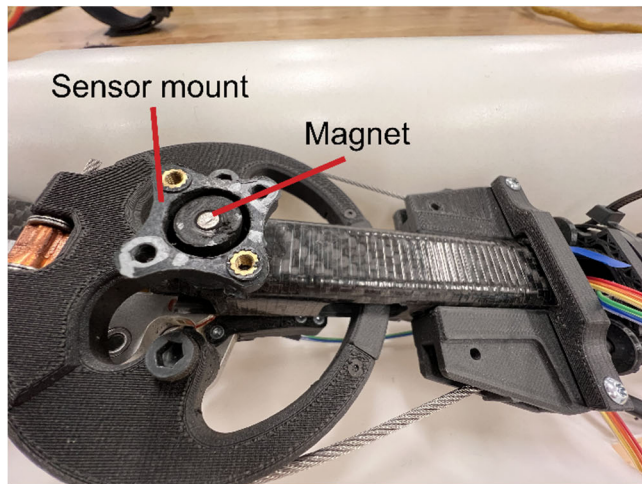
**Fig S9. Hip exoskeleton motor heat shields reduce the temperature of the exposed device surfaces.** (A) Design of AK60-6v1.1 hip heat shield. (B) Rendering of heat shield on assembled hip configuration. (C) Image of heat shield on hip motors. (D) Surface temperatures of motor (solid) and heat shield cover (dashed) over the course of a 15-minute hip exoskeleton walk, replicating the experimental protocol of the hip configuration duration test. The heat shield was effective in reducing the temperature of the exposed surface, with a difference of 9.3 °C at the end of the protocol.



**Fig S10. Control profiles utilized for the hip and ankle configurations.** (A) The hip exoskeleton is controlled based on an estimate of the user's current position in their gait cycle (Percent Gait Cycle) as calculated by the time since their last heel strike relative to their average time from heel strike to heel strike. The prescribed torque profile is then applied via extension and flexion spline curves that are based on a user defined shift in percent gait (Shifted Percent Gait Cycle, to avoid discontinuities at heel-strike) and user-defined parameters (rise time, extension magnitude, peak flexion,...). This controller is based on the works of Bryan et al. (27) and Franks et al. (28). (B) The ankle exoskeleton is controlled via a previously described proportional joint moment controller (29, 30). This operates by detecting the current state of the user (stance or swing) based on a FSR placed at the forefoot. If the user is in stance, an estimate of the ankle moment is determined based on a previously validated regression equation that takes a normalized FSR value as its main input. This moment is then scaled based on a user defined setpoint resulting in an assistive torque profile that follows the shape of the user's estimated ankle moment but with a torque that is only a percentage of that moment. If in swing, a constant user defined torque can be supplied if desired.



**Fig S11. Measured torque (red), angular velocity (grey), and power (black) supplied by the device to the user during functional assistance across different configurations.** The solid lines represent the average value of each measure of interest, while the shaded regions represent  $\pm$  one standard deviation from the one-minute trial ( $N = 1$  for each test). **(A)** During level ground treadmill walking (Treadmill Speed:  $1.25 \text{ ms}^{-1}$ ) with maximum experimentally-determined hip assistance (Flexion: 5 Nm, Extension: 5.5 Nm), the device provided a peak power of 8.38 W. **(B)** During level ground treadmill walking (Treadmill Speed:  $1.25 \text{ ms}^{-1}$ ) with 5 Nm of additional hip assistance beyond the experimentally-determined maximum preference (Flexion: 10 Nm; Extension: 10.5 Nm), the device provided a peak power of 20.47 W. **(C)** During level ground treadmill walking (Treadmill Speed:  $1.25 \text{ ms}^{-1}$ ) while configured for ankle assistance at the maximum experimentally-determined level (Plantarflexion: 28 Nm), the device provided a peak power of 116.43 W. **Note: For (C), angular velocity in the figure is scaled by a factor of 10 to aid in visualization relative to the other components of the plot.** Data for (C) was collected on a different user than (A) and (B).



**Fig S12. Magnetic angle encoder for the ankle configuration.** The ankle configuration was modified by mounting a magnetic rotary position sensor module (AS5600; ams OSRAM, Right) onto the exoskeleton's carbon fiber upright in-line with the center of the pulley. A diametrical magnet was rotated in-line with the pulley's axis of rotation, changing its magnetic field. The relative rotation of the magnet was detected by the sensor and expressed as a voltage drop across the OUT and GND pins of the module (Range: 0 to 3.3 V) which was then scaled to the device's range of motion (1.78 rad) to provide an estimate of the angle.

**Supplemental Table 1:** Participant Characteristics.

| <b>Configuration</b>   | <b>Condition</b>            | <b>Participant</b> | <b>Sex</b> | <b>Age<br/>(years)</b> | <b>Mass<br/>(kg)</b> | <b>Height<br/>(cm)</b> | <b>Torque<br/>(Nm)</b>          | <b>Walking<br/>Speed<br/>(ms<sup>-1</sup>)</b> |
|------------------------|-----------------------------|--------------------|------------|------------------------|----------------------|------------------------|---------------------------------|------------------------------------------------|
| <b>Hip</b>             | Incline walking<br>(7.5°)   | P1                 | M          | 23                     | 65.6                 | 169.0                  | Ext: 4.1<br>Flex: 3.4           | 1.0                                            |
|                        |                             | P2                 | F          | 63                     | 50.6                 | 167.5                  | Ext: 4.1<br>Flex: 3.4           | 0.85                                           |
| <b>Ankle</b>           | Outdoor                     | P3                 | M          | 28                     | 93.0                 | 182.0                  | PF: 28                          | Not Fixed                                      |
|                        | Indoor                      |                    |            |                        |                      |                        | PF: 28                          | 1.25                                           |
| <b>Hip-&amp;-Ankle</b> | Load carriage<br>(22.5% BW) | P4                 | M          | 32                     | 70.8                 | 172                    | Ext: 4.1<br>Flex: 3.4<br>PF: 28 | 1.25                                           |
|                        |                             | P5                 | M          | 37                     | 71.1                 | 173.5                  | Ext: 5.5<br>Flex: 4.9<br>PF: 28 | 1.25                                           |
| <b>Elbow</b>           | Weight lifting<br>(19.5 kg) | P6                 | F          | 22                     | 65.7                 | 156.5                  | Flex: 12                        | NA                                             |
|                        |                             | P7                 | M          | 23                     | 79.1                 | 171.8                  | Flex: 12                        | NA                                             |

Note: Ext = Extension; Flex = Flexion; PF = Plantarflexion