**\section\*{Kinematic Modeling}**

The biarticular exoskeleton was designed to assist hip and knee joints. The exoskeleton was inspired by the biarticular muscles and their functionality, and the aim of the design was to keep the large portion of the device weight around the proximal joint (Hip) while delivering the required power to the distal joint (Knee). A parallelogram mechanism was purposed in the exoskeleton to accomplish this goal and take advantage of the biological features of biarticular muscles.. The purposed assistive device is shown in Figure \ref{Fig\_Exos\_Kinematics\_Model}\subref{Fig\_Biarticular\_Exo\_Mechanism}.\\

{figure}

A monoarticular exoskeleton can be modeled as a two-link serial manipulator as shown in Figure \ref{Fig\_Exos\_Kinematics\_Model}\subref{Fig\_Monoarticular\_Exo\_Mechanism} in which each joint is assisted by the joint actuator directly. The kinematics modeling of the monoarticular and biarticular exoskeletons atboth the configuration and motion levels has been represented in \nameref{S1\_Appendix}.\\

As can be interpreted from the kinematics of the exoskeletons represented in \nameref{S1\_Appendix}, a linear mapping between monoarticular and biarticular exoskeletons can be established to relate these two devices through a linear jacobian, as is represented in Eqn \eqref{Eqn\_Mono\_Bi\_Jacobian}.

{equation}

Using Eqn.\eqref{Eqn\_Mono\_Bi\_Jacobian} which is a mapping among the angular velocities of the exoskeletons, we can derive the mapping between the torque provided by exoskeletons as shown in Eqn. \eqref{Mono\_Bi\_Torque\_Mapping}.

{equation}

This relation between two exoskeleton was used to verify the modeling of the exoskeleton through a/its musculoskeletal simulation framework.

**\section\*{Musculoskeletal Simulation}**

**\subsection\*{Musculoskeletal Model}**

The exoskeletons were studied through musculoskeletal simulations by conducting the simulations of seven subjects walking normally and while carrying a 38kg load on the torso at their chosen speed. The data used in this study was experimentally collected and processed by Dembia et.al. \cite{93}, and their experimental protocol was approved by the Stanford University Institutional Review Board \cite{93}.\\

The musculoskeletal model used in the simulations, which was the same as the model used by Dembia et al. \cite{93}, was a three-dimensional model developed by Rajagopal et al. \cite{130} with 39 degrees of freedom in which the lower limbs were actuated using 80 massless musculotendon actuators, and the upper limb was actuated by 17 torque actuators\cite{130}. \\

This three-dimensional musculoskeletal model was adapted by locking some unnecessary degrees of freedom for both normal walking and walking with a heavy load scenarios and modeling the extra load on the torso of the musculoskeletal model for the walking with heavy load condition \cite{93}.\\

Since this research was built upon the study performed by Dembia et al., we followed similar terminologies in most of the cases to avoid any confusion. Therefore, the \textit{loaded} condition refers to subjects walking while carrying a 38Kg load on their torso while the \textit{noload} condition references subjects walking without any extra load at their chosen speed.

**\subsection\*{Simulation Procedure}**

The first step of conducting the simulations for each subject is scaling the generic dynamic model to acquire a musculoskeletal model matching the anthropometry of each subject, which was performed using OpenSim Scale Tool, and the maximum isometric forces of the muscles were scaled according to the mass and height of each subject \cite{93}. After obtaining the scaled model for each subject, inverse kinematics for each subject was computed using OpenSim Inverse Kinematics Tool and the motion capture data was collected experimentally to obtain the angle trajectories of joints.\\

At the next stage of the simulation workflow, the scaled model, inverse kinematics, and ground reaction forces were employed to run the RRA algorithm\cite{103}. The RRA algorithm reduces the incompatibility of experimental data, including ground reaction forces and trace data, and the musculoskeletal model by slightly adjusting inertial properties and kinematics. The adjusted model and kinematics generated by RRA were then employed to perform muscle driven simulations using a computed muscle control algorithm in OpenSim\cite{104}.\\

The computed muscle control (CMC) algorithm simulates the muscle recruitment of the subject by resolving the muscle redundancy problem using static optimization to find the required muscle excitations to track the adjusted kinematics. The CMC simulation output was then used to run the analysis tool of OpenSim to compute the subject’s metabolic power consumption, muscle moments, and joint reaction forces.\\

{figure}

The OpenSim computed muscles control algorithm solves the muscle redundancy problem to track experimentally measured motion using effort-based objective, as represented in Eqn.\eqref{Eqn\_CMC\_Objective}. This objective function was optimized to obtain a set of muscle excitations to track measured motions and forces within a specified tolerance at each time step during the motion of interest using a static optimization method\cite{92}. Therefore, the kinematics and dynamics of the subject remain consistent during the simulations, and any additional mass and inertia on the subject that has not been captured by experiments will cause a systematic error in the results.\\

{equation}

With the knowledge of the OpenSim neural control algorithm, we used the adjusted model and kinematics provided by Dembia et al.\cite{93} instead of reproducing all data from the beginning of the simulation procedure, which also helped ease the verification of the simulations procedure thanks to \cite{93} for verified simulations data.\\

\paragraph\*{Metabolic model.} To calculate the estimated instantaneous metabolic power of subjects, Umberger \cite{105} muscle energetic model, which was modified by Uchida et al. \cite{106}, was employed in which average power consumption of a muscle during a gait cycle was calculated using Eq.\eqref{Eqn\_avg\_muscle\_power} \cite{106}.\\

{equation}

where m is muscle mass, and $\dot{E(t)}$ is the normalized metabolic power consumed. This model generates the metabolic power of all muscles; whole body metabolic power is then calculated by summing the metabolic power of all muscles \cite{106}. To compute the gross metabolic energy consumption of all subjects, we integrated the metabolic power over the gait cycle and then divided the cross metabolic power by the mass of subjects.\\

As is mentioned in \cite{93}, due to experimental data insufficiency, some subjects and trial simulations were not a complete gait cycle;therefore, the metabolic energy was calculated for a half of a gait cycle for these subjects and trials, which is a verified method for computing the energy according to \cite{93}. \\

**Joint reaction forces and moments analysis.** Since the equations of motion of the musculoskeletal model were formulated in terms of the generalized coordinates and generalized forces, the internal forces and moments were not solved while performing the computed muscles control or residual reduction algorithm simulations. Consequently, we employed joint reaction analysis provided by OpenSim to compute the resultant forces and moments between two consecutive bodies in a/the kinematic chain connected via a joint. The contact forces and moments of joints were obtained by formulating them through the Newton-Euler equation of motion and solving recursively from the distal to proximal joints.\\

The free body diagram of $i$th body and joint is provided in Figure \ref{Fig\_JRF\_FBD}, which has been adopted from the figure represented in the Supplementary Material of \cite{151}. The Newton-Euler formulation for $i$th body can be represented as Eqn. \eqref{Eqn\_Body\_Reaction\_Force} adapted from \cite{151}, which has been solved to obtain the contact forces and moments acting on the body.

{equation}

{figure}

where $M\_i(q)$ and ${\mathit{\mathbf{a}}}\_i$, respectively, represent the mass matrix of the body $i$ and vector of the linear and angular acceleration of body $i$ expressed at ground frame, and $F\_{constaint}$ accounts for the forces applied by constraints, if applicable. Through this equation $F\_{muscle}$, $F\_{external}$, and $F\_{gravity}$ represent the force and moment applied by a muscle, forces applied externally (e.g. ground reaction forces and moments), and gravitational forces applied to the body, respectively. Lastly, $R\_{i+1}$ accounts for the applied reaction forces from the $(i+1)$th to the $i$th joint.\\

Since these reaction forces and moments are expressed at the origin of the body frame to maintain all terms in a common reference frame, they need to be transformed to the location of the joint frame (i.e. offset frame) where the joint has been defined between two consecutive bodies as represented in Eqn. \eqref{Eqn\_JointReactionForce} \cite{151}.

{equation}

The vector of $F\_i$ and $\tau\_i$ represent the joint reaction force applied to the joint of interest expressed at the ground frame.\\

As was mentioned earlier, this analysis was adopted from the supplementary material of the \cite{151} and we reference the readers to this paper for detailed discussion about this analysis.\\

**\subsection\*{Modeling and simulation of assisted subjects}**

The kinematics of the exoskeletons were already discussed; in order to model ideal exoskeletons in OpenSim framework, we used the Torque Actuators provided by OpenSim API\cite{103}. Torque actuators of biarticular and monoarticular exoskeletons were assigned, as shown in Figure \ref{Fig\_Exos\_Model\_Opensim}.

{figure}

As is represented in Figure \ref{Fig\_Exos\_Model\_Opensim} \subref{Fig\_Bi\_Exo\_Model\_Opensim}, both torque actuators of a/the biarticular exoskeleton were assigned to the torso;, thereaction forces of the actuators were then applied to the torso, which matches the kinematics and dynamics model of the biarticular exoskeleton.

{equation}

A/The Monoarticular exoskeleton (figure \ref{Fig\_Exos\_Model\_Opensim} \subref{Fig\_Mono\_Exo\_Model\_Opensim}) was modeled by assigning a/the hip joint actuator from the torso to the femur body, and the knee joint actuator was assigned from the femur to the tibia body where knee torque actuator's reaction torque applied to femur body:

{equation}

**Computed Muscle Control adjusted objective function.** To investigate the performance of the assistive devices and their effect on the human musculoskeletal system through an/the OpenSim simulation framework, we used the CMC algorithm. The Computed Muscle Control algorithm objective function depends on the sum of squared muscle activation and reserve actuators, which compensates for modeled passive structures and potential muscle weakness\cite{93}:

{equation}

where $w\_i$ determines the weight of reserve actuators, which is generally selected as a small number to highly penalize the use of reserve actuators. By adding assistive device actuators (i.e. torque actuators) to the musculoskeletal model of the subject, they are added to the CMC tool objective function. The adjusted objective function includes the assistive actuators ast is expressed in Eq. \eqref{Eqn\_CMC\_Assisted\_Obj\_Func}, and by selecting proper weights for the assistive actuators, they can be chosen by the optimizer as the actuation of the assigned degree of freedom.

{equation}

In the adjusted objective function, $w\_{exo,i}$ is torque actuator weights, which is named optimal force in OpenSim \cite{93} penalizing the usage of torque actuators. By selecting a large number, penalization of the actuators is insignificant and they are selected for actuating the joint between two bodies assigned for the torque actuator. If we select a small optimal force, the optimizer will highly penalize the usage of exoskeleton actuators. To study each configuration of the exoskeleton at their maximum performance, the assigned torque actuator's optimal force was selected as 1000 N.m, enabling the optimizer to use the assistive actuators as much as possible during a gait cycle simulation.\\

**Power calculation of Metabolics and actuators.** Similar to the unassisted procedure, the instantaneous metabolic power of the subjects was computed using the energetic model of Uchida et al. \cite{106}. The etabolic energy of all subjects was then derived through integration of the metabolic power over the gait cycle. In order to compute the energy consumption of the assistive actuators, the power profiles of the actuators were obtained and their absolute power profiles were integrated over the gait cycle and divided to the subjects’ total mass. Similar to the energy consumption of the exoskeleton procedure, the negative energy or regenerable energy through a gait cycle was calculated by obtaining the negative power profile and integrating it over the gait cycle and normalizing it to the mass of the subject.