Knee Exoskeleton Simulation Phase Setup Guide

This document describes how to set up and begin the Simulation Phase for the knee-only exoskeleton project. It provides a clear workflow, tools, requirements, and deliverables to achieve realistic system behavior before hardware fabrication. The simulation combines mechanical modeling, control design, and data analysis to validate torque, energy, and stability performance.

1. Objective

Simulate the mechanical and control behavior of the knee-only exoskeleton to evaluate torque tracking, actuator dynamics, and power efficiency using virtual models. The results will guide actuator sizing, control parameters, and overall design validation before building physical hardware.

2. Simulation Technologies and Tools

Tool	Purpose
OpenSim	Extract gait cycle data (knee torque, angle, velocity) to use as control reference.
MuJoCo	Simulate exoskeleton dynamics, Bowden cable compliance, and actuator respons
Simscape Multibody (MATLAB)	Alternative to MuJoCo for detailed actuator + control co-simulation.
Simulink / Python Control	Implement torque and impedance controllers with noise and delay models.
NumPy / Matplotlib	Data analysis and visualization of torque, power, and stability plots.
Gazebo / PyBullet	Optional 3D visualization and full system control integration.

3. Step-by-Step Simulation Workflow

Step 1 – Define System Architecture
Draw a block diagram linking waist actuators → Bowden cables → knee joint → sensors → controller. Include feedback paths for encoder, load cell, and IMU.

Step 2 – Obtain Human Gait Data (OpenSim)
While OpenSim account approval is pending, use open-source gait datasets (e.g., Stanford Gait Database) or synthetic sinusoidal motion for initial testing. Later, replace with OpenSim-generated torque data.

Step 3 – Build Mechanical Model (MuJoCo or Simscape)
Create rigid links: waist, thigh, and shank. Add rotational knee joint and assign masses from the BOM. Include BLDC motor + SEA spring + Bowden cable compliance as actuation model.

Step 4 – Implement Control System
Design inner torque control and outer impedance control loops. Add noise and latency to sensor feedback. Test stability and responsiveness using different controller gains.

Step 5 – Run Dynamic Simulations
Simulate one gait cycle and evaluate torque tracking, joint angle, and actuator current. Observe system response for walking and sit-to-stand motions.

Step 6 – Perform Parameter Sweeps
Vary SEA stiffness, Bowden friction, and control gains to identify optimal system parameters. Plot performance metrics (error, power, comfort).

Step 7 – Export and Analyze Results
Record torque, velocity, power, and current consumption. Estimate battery life and mechanical efficiency. Validate that joint reaction forces remain within safe limits.

4. Deliverables

• System block diagram (architecture)
• OpenSim gait data or synthetic torque reference
• MuJoCo/Simscape model of leg + actuator
• Controller implementation and tuning results
• Plots: torque tracking, power consumption, and error analysis
• Optimized design parameters (SEA stiffness, control gains)

5. System Requirements

Category	Minimum Specification
CPU	Intel i5 / AMD Ryzen 5 or higher
RAM	16 GB minimum
GPU	Optional (for MuJoCo / PyBullet visualization)
Software	Python 3.10+, MATLAB R2023b+, MuJoCo, Simscape, OpenSim
OS	Windows / macOS / Linux

6. Next Steps (While Waiting for OpenSim Access)

1. Set up Python environment with MuJoCo or PyBullet.
2. Create a simple 2-link (thigh + shank) model with rotational joint.
3. Implement a basic torque controller using synthetic sinusoidal knee torque.
4. Validate actuator and SEA model response.
5. Once OpenSim access is granted, extract real gait torque data and rerun simulations for accuracy.

7. Conclusion

The simulation phase ensures that the knee exoskeleton system is mechanically and dynamically feasible. It allows safe experimentation on virtual models, saving both time and cost before fabrication. Once the OpenSim data is integrated, results can be directly used to fine-tune control strategies and begin hardware-in-loop testing.