Stability Optimisation of an Assistive Lower Limb Exoskeleton

R. Anthony[a], M. Petersen[b]

Received 9 February 2024, in revised form 18 January 2026 and accepted 26 August 2026

Abstract: ***This paper explores the optimisation of stability in powered lower limb exoskeletons for stroke rehabilitation. Employing the linear inverted pendulum model and a genetic algorithm-based optimisation approach, the study utilised MATLAB simulations to predict hardware requirements and stability enhancements. The exoskeleton design was informed by existing systems, and a scaled physical test platform, while not achieving walking functionality, provided crucial insights into the challenges of translating simulation results to hardware. Results showed the linear inverted pendulum model achieved a stable gait, and the optimisation technique exhibited progressive improvements, contributing valuable insights for developing more stable and human-like walking gaits in powered lower limb exoskeletons. This research sets the groundwork for enhancing rehabilitation systems, potentially benefiting patients with spinal cord injuries, and supporting healthcare professionals.***

Additional keywords: Powered Lower Limb Exoskeleton, Genetic Algorithm, Linear Inverted Pendulum Model

1 Introduction

Mobility is a crucial factor for individuals to actively participate in various aspects of professional, social, and economic life. Limited personal travel capabilities pose challenges, hindering full participation in essential activities and diminishing overall quality of life [1]. In South Africa, an overburdened health system restricts access to rehabilitation for those who have suffered a stroke or spinal cord injury, amplifying mobility issues and reducing quality of life [2].

Powered lower limb exoskeletons (LLE) offer potential solutions, particularly for individuals with spinal cord injuries, enabling improved mobility and independence [3]. However, concerns persist about their practicality, prompting the need for research to enhance their design, usability, and potential impact [4].

This research aims to contribute to the advancement of powered LLE technology. Key research questions include optimising stability, predicting hardware requirements for untethered walking through simulation, and evaluating different optimisation techniques. The research employs mathematical modelling, simulation, and a scaled test platform. The successful outcomes of this research could offer valuable insights for design decisions.

Despite the advancements in powered LLEs globally, their impact in South Africa remains limited to sparking interest rather than practical application. Existing solutions face several limitations, including reliance on external aids, restricted adaptability to diverse terrains, and challenges in achieving untethered stability and energy efficiency. Moreover, there is a lack of research addressing these challenges from a South African perspective, where unique socio-economic and environmental factors may influence the design and implementation of powered LLEs. To address these gaps, this research employs MATLAB as a simulation tool to provide an accessible framework for exploring stability optimisation techniques before committing to expensive hardware development. This approach enables researchers to predict hardware requirements, compare different walking algorithms, and identify optimal configurations in a controlled environment—capabilities not readily available when working directly with physical prototypes.

2 Related Work

This section is an examination of existing legged platforms to discern their strengths and compromises. The insights gained from this exploration aim to identify gaps in current research, providing foundational knowledge for the design and simulation of a powered LLE with a focus on stability optimisation for untethered walking.

LLEs, robotic structures worn by humans to augment leg abilities, serve diverse purposes, including gait rehabilitation, strength augmentation, and locomotion assistance for those with lower limb paralysis [5]. Exoskeleton research, dating back to the 1960s, has recently pivoted towards rehabilitation and assistive use cases, emphasising safety, robustness, and user-friendliness [6, 7].

2.1 Classification of Powered LLEs

Chen et al. [7] categorise LLEs into three types: gait rehabilitation, strength augmentation, and locomotion assistive. This research focuses on the third category, aiding individuals with complete lower limb paralysis, aiming to replace wheelchairs and enhance mobility and access to the environment.

Distinctions between gait rehabilitation and locomotion assistive exoskeletons are crucial. Gait rehabilitation targets therapeutic interventions for impaired gait patterns, primarily used in clinical settings [8, 9]. In contrast, locomotion

[a] ORCID 0009-0007-5655-9195, Department of Mechanical and Mechatronic Engineering, Cape Peninsula University of Technology, South Africa. E-mail: ra215223217@gmail.com

[b] ORCID 0000-0002-0427-5461, Department of Mechanical and Mechatronic Engineering, Cape Peninsula University of Technology, South Africa. E-mail: petersenmi@cput.ac.za

http://dx.doi.org/10.69694/2309-8988/2026/v42a4

assistive exoskeletons aim to restore independent walking for those with mobility impairments, incorporating active joints at the hip, knee, and ankle [7].

2.1.1 Survey of Existing Locomotion Assistive LLEs

This section offers a brief review of prominent powered locomotion assistive LLEs and their capabilities.

MINDWALKER: Developed to assist paraplegics, MINDWALKER employs five degrees of freedom (DOF) per leg, combining powered series elastic actuators with passive support. It demonstrated potential for ambulation in paraplegics, revealing interesting electromyography patterns in upper limb muscles during stepping [10, 11].

ReWalk: Food and Drug Administration-approved, ReWalk aids thoracic-level motor-complete spinal cord injury patients, showcasing hip and knee movements. Clinical studies attest to its effectiveness in restoring ambulatory function and improving overall quality of life [7, 12].

Vanderbilt: Modular and lightweight, Vanderbilt exoskeleton allows patients to assemble it, providing essential movements for individuals with paralysis. Its design facilitates sit, stand, walk, and stair navigation, enhancing mobility and independence [7].

REX: Uniquely enabling untethered walking without crutches, REX focuses on aiding thoracic-level motor-complete spinal cord injury patients. Challenges include slow gait speed and limited range, requiring further research for optimisation [13, 14].

Atalante by Wandercraft: A crutchless exoskeleton designed for autonomy, Atalante demonstrates stable bipedal walking and dynamic motions. With an advanced mechatronic design, it offers hands-free walking for individuals with mobility impairments [15, 16].

While these systems demonstrate significant progress in powered LLE technology, common challenges remain, including reliance on external aids for balance, limited terrain adaptability, and the need for further stability optimisation. These limitations informed the focus of this research on stability enhancement through simulation-based methods.

3 Simulation Methodology

This section discusses the procedures and techniques used to achieve the goals of this research which were to use mathematical modelling that would predict the hardware requirements for untethered walking of a powered LLE, and to use this simulation to optimise for stability in a simulated environment.

3.1 Design Considerations

In developing the simulation, several design considerations were made to ensure the model accurately represented a powered LLE. One key aspect was determining the size and weight of the exoskeleton, which were based on realistic measurements and specifications. To incorporate accurate physical parameters such as mass, centre of mass (CoM), moments of inertia, and products of inertia, the 3D model of the exoskeleton was developed using Onshape, a cloud-based Computer-Aided Design (CAD) platform. Onshape was chosen for its accessibility, collaborative features, and ability to provide detailed inertia properties for each subsection of the system. Once the 3D model was created in Onshape, these values were extracted and used to set up the simulation, ensuring that the physical dynamics of the system were faithfully represented.

3.2 Requirements

The requirements relevant to the simulation portion of this research are as follows:

- The simulation should accurately portray the movement of the powered LLE with realistic impact forces, inertial properties, and the effects of gravity.
- The simulation should provide insights into optimising stability in a simulated environment, allowing for the comparison of different walking algorithms.
- The model of the powered LLE system should be full-scale, representing realistic size and proportions. It should be able to accommodate a test dummy with a height of approximately 185 cm, which falls between the dimensions of the Hybrid III 50th Male (175 cm) and the Hybrid III 95th Male (188 cm) test dummies.
- The simulation should be user-friendly and accessible to researchers, facilitating the adoption of powered LLE technology in research at the Cape Peninsula University of Technology.
- The stability margins employed in the simulation must serve as sufficient conditions for stability, providing meaningful and quantifiable results.
- The simulation should account for the trade-off between complexity and computational resources, ensuring a reasonable computational time for repetitive simulations.
- Assumptions may be made to simplify the model and leverage available resources without sacrificing the goal of accurately predicting hardware requirements for untethered walking.
- Specific actuator modelling and production-ready design of the powered LLE are not necessary for the simulation, as long as the simplified model is compatible with human body movements.

The goal of this simulation was to identify a method for optimising stability in a realistic powered LLE model and, in turn, predict the hardware specifications that may be required—if such hardware currently exists—to achieve walking performance similar to that of the simulation. The focus was not on providing a complete hardware design for constructing a powered LLE, but rather on understanding the fundamental requirements that would enable stable walking. To simplify the model while maintaining its relevance, assumptions such as idealised joint mechanics and a simplified walking pattern were made, allowing for a manageable simulation that could focus on key stability factors. Detailed actuator modelling and production-ready design were not necessary at this stage, as the primary aim was to validate the walking dynamics and determine the necessary hardware specifications for successful implementation.

http://dx.doi.org/10.69694/2309-8988/2026/v42a4

3.3 CAD Model

The final CAD assembly is a simple structure designed by the researcher, mated to a test dummy model that was obtained (figure 1). It was used to primarily represent how weight could be distributed through the system and to specify joint positions. The joints and frame parts are simplified bodies which would, in a real powered LLE, need motors, a power source, control hardware, wiring and various other hardware to function.

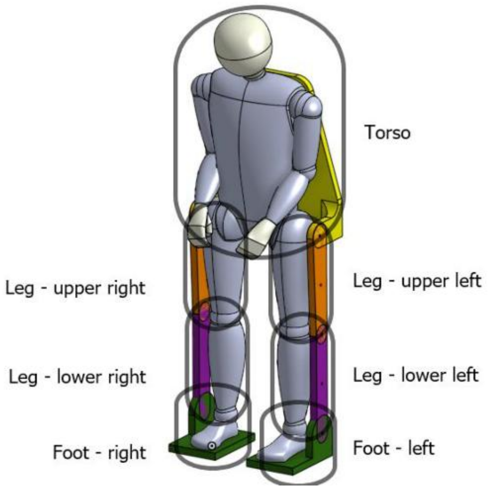

Figure 1 Final CAD Assembly

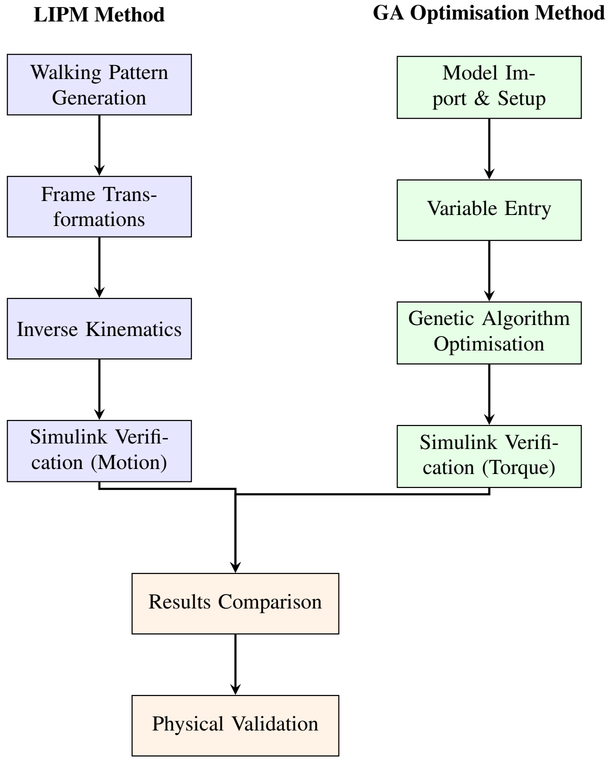

Figure 2 Methodology workflow showing the two parallel approaches: LIPM method (left) and genetic algorithm optimisation method (right), both leading to results comparison and physical validation.

3.4 Methods of Obtaining Walking Gaits

Stable walking gaits were obtained using two different methods, both implemented within the "MATLAB and Simulink Robotics Arena: Walking Robot" software developed by the MathWorks Student Competitions Team [17]. This software, freely available to students, provides a bipedal walker framework that was adapted for this research. Two different methods were used to obtain a walking gait, firstly the linear inverted pendulum model (LIPM) to produce a walking pattern, followed by another technique which makes use of motion planning and a genetic algorithm to produce a more dynamic gait which will later be optimised.

While this research utilised the MathWorks Walking Robot software as a foundation, several original contributions distinguish this work: the development and integration of a custom full-scale powered LLE CAD model with realistic mass and inertia properties derived from Onshape; the systematic comparison of LIPM and genetic algorithm optimisation approaches for stability; the analysis of simulation results against human gait characteristics; and the attempted physical validation using a scaled test platform. The MathWorks software provided the simulation framework and optimisation tools, while this research applied these tools to a novel exoskeleton design and evaluated their effectiveness for assistive powered LLE applications. Figure 2 illustrates the methodology workflow, showing the two parallel approaches and their convergence at results comparison and physical validation.

3.4.1 Linear Inverted Pendulum Model

The LIPM is a widely employed low fidelity technique for modelling bipedal walking robots, known for its simplicity, and frequently used in the design and analysis of legged locomotion [18]. Using the tools created by the MathWorks team, the researcher was able to import the CAD model that was prepared and manipulate variables to obtain a walking gait. The resulting body trajectory assumes that the walking pattern is symmetric and that there are no disturbances.

Although it is possible to optimise LIPM walking gaits, this research chose not. This decision was because the LIPM typically assumes a fixed Z-height of the torso or the CoM. This means that the model assumes the body remains at a constant vertical height while the legs move to generate forward motion, which does not account for the natural up-and-down movement (vertical oscillation) that occurs during actual walking [19]. This results in a less natural walking gait.

Walking Pattern Generation: Using existing software from the MathWorks team, the researcher first created a walking pattern and used it to transform the reference frame from the CoM to the foot positions. Inverse kinematics was then employed to calculate the required joint angles for each leg to achieve these desired foot positions. Finally, the walking robot was simulated using the open-loop trajectory, which was computed by the software based on the calculated joint trajectories.

Figure 3 illustrates the walking pattern generated using the LIPM. The trajectory of the CoM is shown alongside the foot

http://dx.doi.org/10.69694/2309-8988/2026/v42a4

placements for both the left and right feet. The blue line represents the virtual leg connecting the CoM to the stance foot, while the curved lines depict the alternating footstep locations. This simplified model assumes a fixed CoM height and idealised dynamics to demonstrate bipedal walking.

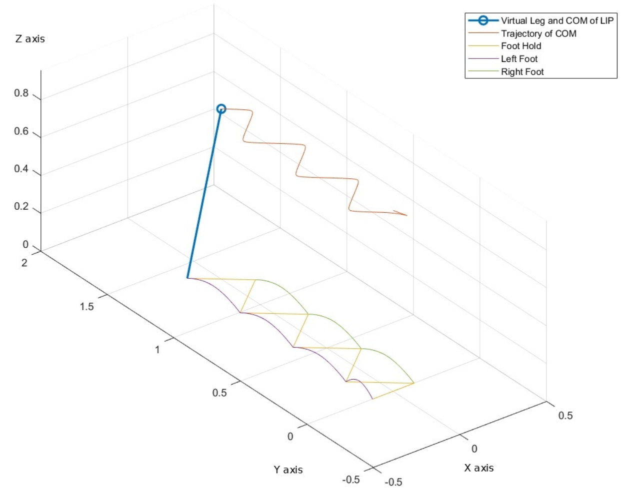

Figure 3 LIPM CoM Walking Pattern with Virtual Leg

Frame Transformations: After the walking pattern in figure 3 was obtained, the reference frame needed to be transformed from the CoM relative to the world, to the foot positions relative to the robot body. This transformation is crucial for translating the global walking pattern into local foot trajectories that the robot can follow.

The MathWorks team's script facilitates this process by converting the global origin-based coordinates into foot trajectories referenced to the body's CoM. When the script is run, it generates an animation visualising these trajectories. Figure 4 shows the resulting foot trajectories for both the left and right foot, highlighting their motion paths during the walking sequence. The blue and red lines represent the respective trajectories of the left and right foot, demonstrating the alternating step pattern required for stable locomotion.

Inverse Kinematics: The next step in the process was to convert the foot trajectories from the previous section to joint angles which can be used to control a Simulink model. To solve this problem, inverse kinematics would need to be employed on a robot structure containing the desired joints and DOF. The previous section contained only foot trajectories with a reference point, the 'legs' were simply for visualisation purposes.

This was accomplished by using a script which configures a rigid body tree using Denavit-Hartenberg (DH) parameters to set up the kinematic chain. In this case the MathWorks Student Competitions Team chose to use DH parameters which represent the legs of the HUBO humanoid robot, which are said to be common throughout many bipedal walking robots [20].

Tables 1 and 2 describe the DH parameters for the right and left legs of the simulated powered LLE. DH parameters are a standard method for representing the spatial relationships between consecutive links in a kinematic chain. Each row in the tables corresponds to a joint in the leg, with the parameters

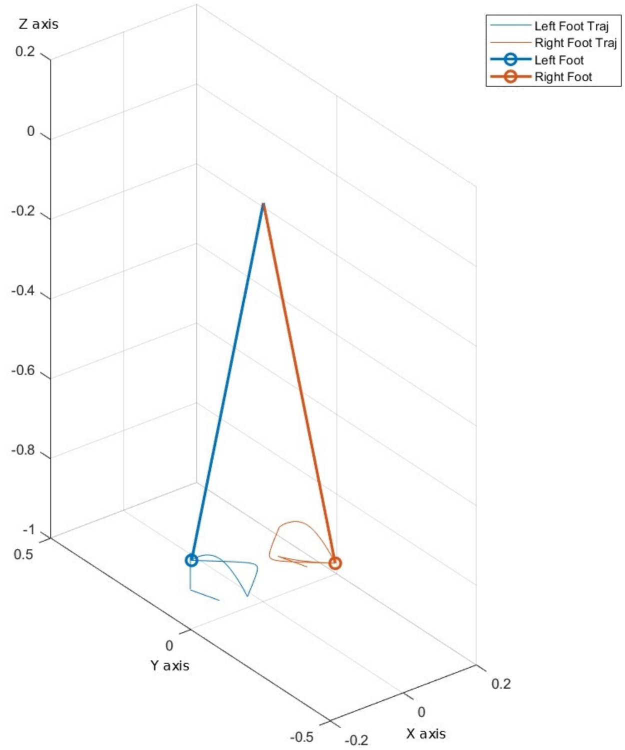

Figure 4 Feet Trajectories Visualisation During Walking

defined as follows:

- a_i: The distance between the current joint and the previous joint, measured along the x-axis of the current joint.
- α_i: The angle of rotation about the x-axis from the previous z-axis to the current z-axis.
- d_i: The distance along the z-axis to align the previous x-axis with the current x-axis. This is often represented as a variable (e.g., L_2), which corresponds to a specific physical dimension in the robotic leg.
- θ_i: The angle of rotation around the z-axis to align the previous x-axis with the current x-axis.

For example:

- In table 1, Joint 0 has $d_i = -L_2$, where L_2 is a variable representing the distance between the base and the hip yaw joint along the z-axis. Similarly, θ_i describes the rotation required around the z-axis at this joint.
- The left leg, represented in table 2, mirrors the configuration of the right leg, with corresponding DH parameters adjusted to ensure symmetry between the two legs.

These parameters are essential for defining the kinematic structure and computing transformations between joints. The use of symbolic variables like L_2, L_3, and L_4 allows the model to remain adaptable to various physical configurations. The DH parameters were chosen to align with the Hubo humanoid robot's design, ensuring compatibility with established robotic frameworks. This setup enables accurate calculation of the foot's position and orientation relative to the body for both legs, as demonstrated in the simulation.

Having a rigid body structure, the foot trajectories could now be mapped onto that structure.

http://dx.doi.org/10.69694/2309-8988/2026/v42a4

Table 1 Right Leg DH Parameters

Joint	a_i	α_i	d_i	θ_i	Description
-	L_1	0	$-L_2$	0	Base to hip yaw
1	0	$-\pi/2$	0	0	Hip yaw to hip roll
2	0	$-\pi/2$	0	0	Hip roll to hip pitch
3	L_3	0	0	0	Hip pitch to knee pitch
4	L_4	0	0	0	Knee pitch to ankle pitch
5	0	$\pi/2$	0	0	Ankle pitch to ankle roll
6	0	0	L_5	0	Ankle roll to end effector (foot)

Table 2 Left Leg DH Parameters

Joint	a_i	α_i	d_i	θ_i	Description
-	$-L_1$	0	$-L_2$	0	Base to hip yaw
1	0	$-\pi/2$	0	0	Hip yaw to hip roll
2	0	$-\pi/2$	0	0	Hip roll to hip pitch
3	L_3	0	0	0	Hip pitch to knee pitch
4	L_4	0	0	0	Knee pitch to ankle pitch
5	0	$\pi/2$	0	0	Ankle pitch to ankle roll
6	0	0	L_5	0	Ankle roll to end effector (foot)

The link coordinate frames of the right leg of a Hubo KHR-4 Robot and its D-H parameters are shown in figure 5 [21].

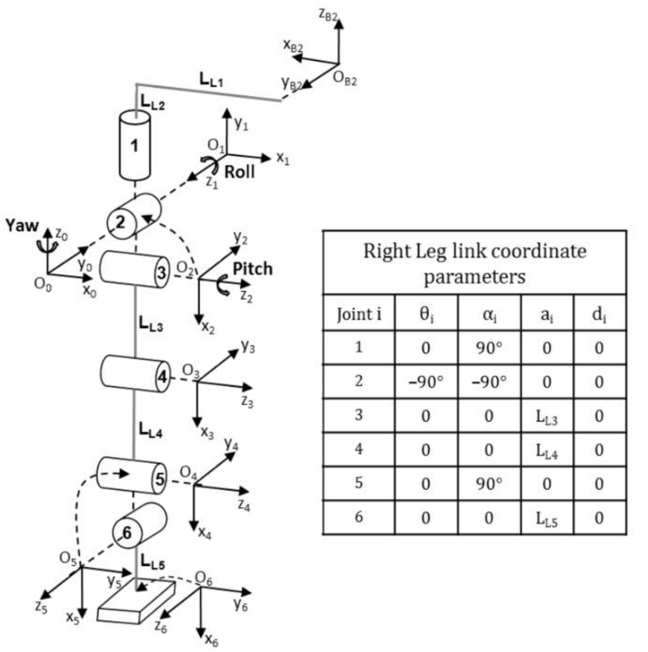

Figure 5 Link Coordinate Frames of the Right Leg of a Hubo KHR-4 Robot and its D-H Parameters [21].

To determine the position and orientation of the end-effector of a limb, the spatial displacement of the 6th coordinate frame relative to the base/reference coordinate frame can be obtained by multiplying the 6 link-transformation matrices together. This can be achieved by using the established link coordinate frames and the D-H parameters seen in figure 5 [21].

A script is used which calculates the joint trajectories, they are then displayed through an animation to verify if they are moving as expected, this can be seen in figure 6.

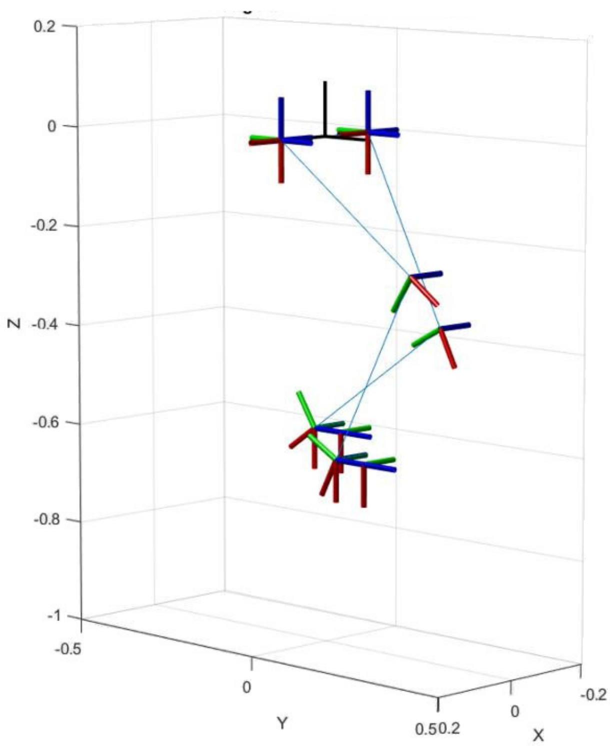

Figure 6 Inverse Kinematics Animation

The walking pattern inverse kinematics was verified by comparing the feet trajectories in the Inverse Kinematics Animation in figure 6, to the desired Feet Trajectories that were seen in figure 4. A visual comparison was made between the step lengths and feet placement positions. Additionally, the joints were checked to ensure they were moving within their defined ranges of motion and that the knees were bending in the correct orientations. Having verified the walking pattern inverse kinematics, the researcher could proceed to using the Simulink model to verify if the robot was able to maintain balance and perform a walk.

Verification using the Simulink Model: Where the inverse kinematic model was used to verify the joint trajectories, the Simulink model is used to model the physical 3D mechanics of the powered LLE to visualise the walking pattern on flat terrain. The same joint trajectories of the walking pattern inverse kinematics animation, which can follow the given trajectory exactly, are applied to the Simulink model. This model incorporates physical properties derived from the CAD design, such as mass, CoM, and moments of inertia, ensuring realistic weight distribution and dynamic behaviour. Although simplified for computational efficiency, the Simulink model provides a realistic representation of the powered LLE's movement by accounting for gravity, inertial properties, and impact forces.

http://dx.doi.org/10.69694/2309-8988/2026/v42a4

Running Simulink Simulation: Once the CAD model was set up, the walking pattern could be tested. By running the Simulink model, the Mechanics Explorer is opened which displays an animation of the powered LLE walking on a surface (figure 7). It can be seen in figure 7 that the final CAD model that was seen previously in figure 1 is traversing a flat runway. The mechanical system is simulated with Simulink's solvers, which considers the mechanical systems contact forces, as well as the physical properties of the solid bodies accelerating within the world environment.

The researcher observed whether the powered LLE fell or not and modified the input variables in the previous steps to alter the walking pattern until the powered LLE was able to perform a walk without falling during 10 seconds of simulation time.

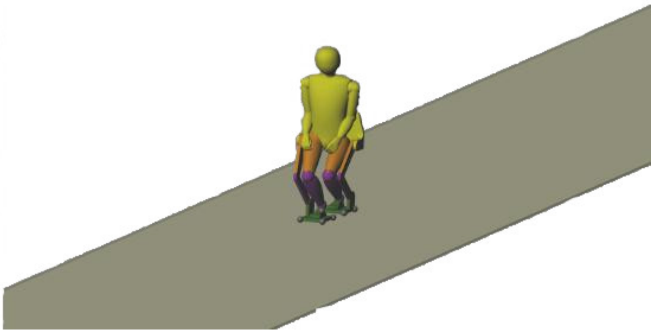

Figure 7 Powered LLE simulation in the Mechanics Explorer showing the CAD model traversing a flat runway

3.4.2 Optimisation using a Genetic Algorithm

An alternative method for obtaining a walking pattern was implemented using genetic algorithm (GA) optimisation. Version 2.0.0.0 of the Walking Robot software [17] was employed, and the results were subsequently compared to those from the LIPM approach.

The optimisation followed an iterative approach. Initially, zero motion initial conditions were used to avoid biasing the algorithm toward a predetermined gait pattern, allowing the GA to explore the solution space independently. The simulation was initially configured with idealised joint motion to enable faster convergence to a viable walking pattern.

Once an initial walking pattern was achieved, this result was used as the initial condition for subsequent optimisation runs. The researcher then transitioned to torque-actuated joints to obtain results more representative of physical actuator behaviour. The distinction between these two actuation modes is significant: motion actuation assumes idealised joints that can follow any commanded trajectory without torque or speed limitations, whereas torque actuation models realistic joint behaviour with torque limits and PID-controlled servo dynamics. Results obtained using motion actuation represent theoretical best-case performance, while torque actuation results more accurately reflect what physical hardware could achieve. A large initial torque limit was used to identify suitable PID values that maintained balance, after which the best-performing simulations were iteratively used as initial conditions while progressively reducing the torque limits until a satisfactory gait was achieved at realistic torque values.

While this iterative refinement approach helped improve solutions progressively, it is acknowledged that the GA may still converge to local minima rather than the global optimum. Future work could explore using multiple random initial conditions or the LIPM trajectory as alternative starting points to potentially identify superior solutions.

Model Import and Setup: The CAD model was imported into the Simulink environment (figure 8) to enable the optimisation workflow.

Optimisation Variable Entry: The simulation variables were configured based on values from the LIPM workflow. The optimisation script employs a GA that iteratively runs the Simulink model to minimise a cost function [22].

GA Function Outline: The GA [23] operates by first creating a random initial population, then iteratively generating new populations through the following process:

a) Each member of the current population is scored by computing its fitness value (raw fitness scores).
b) Raw fitness scores are scaled to produce expectation values.
c) Parents are selected based on their expectation values.
d) Elite individuals with lower fitness values are passed directly to the next population.
e) Children are produced through mutation (random changes to a single parent) or crossover (combining vector entries of two parents).
f) The current population is replaced with children to form the next generation.

The algorithm terminates when a stopping criterion is met.

GA Parameter Settings: The GA was configured with a population size of 20 individuals and a maximum of 50 generations. The initial population matrix was seeded with copies of the initial gait to provide a starting point for optimisation. Mutation and crossover functions were left at MATLAB's default settings: the default crossover function creates offspring by merging pairs of parents, assigning each coordinate from one parent to the offspring, while the default mutation function introduces random alterations with a mutation rate of $1/n$, where n is the number of variables. Parallel computing was enabled to reduce computation time.

GA Function Live Plot: Figure 9 shows the convergence behaviour of the GA during optimisation.

Penalty Value Calculation: The penalty value shown in figure 9 represents the cost function that the GA seeks to minimise. The cost function balances two components: a positive reward based on the distance travelled and simulation duration, calculated as the square of the final x-position multiplied by the simulation end time; and a negative reward based on the average angular velocity and trajectory aggressiveness, where aggressiveness is measured by counting the number of times the joint trajectory derivatives change sign. The final penalty is computed as the negative ratio of positive to negative rewards, ensuring that gaits covering greater distances with smoother joint movements receive lower (better) penalty values.

http://dx.doi.org/10.69694/2309-8988/2026/v42a4

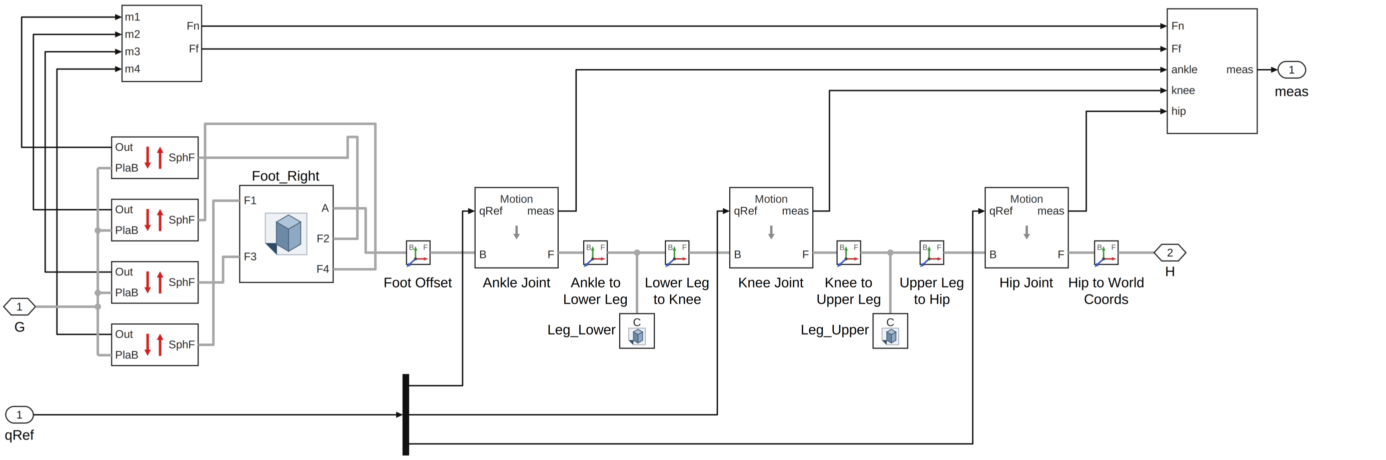

Figure 8 Right Leg Simulink Model

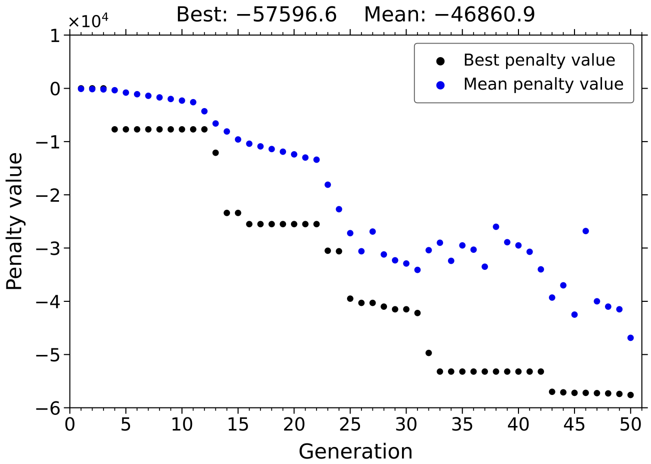

Figure 9 Genetic Algorithm Convergence Plot

4 Simulation Results Comparison

This section presents the results of the two methods employed to achieve a stable walking gait, as discussed in the preceding sections. The first method involves the conventional LIPM, while the second method utilises an optimisation technique incorporating a cost function and GA. It should be noted that the LIPM was not optimised.

The results will be analysed based on the criteria specified in the requirements section. These criteria include the distance covered by the robot within a 10-second simulation time, the maximum torque observed at the ankle, knee, and hip joints during the simulation, the torso height to measure the robot's upright walking posture, and an objective assessment of the human-likeness of the gait.

The results will be presented in graph format, generated using MATLAB software. After many iterations, both the LIPM simulation and the optimisation method simulation successfully achieved the objective of walking for the full 10 seconds of simulation time without experiencing any instances of falling.

Figure 10 shows an overlay of the different stages of the gait from the LIPM simulation onto a standard human gait cycle diagram.

Figure 10 shows that the robot does have discrepancies to a normal human walking gait. Firstly, the robot walks with knees bent, in a squat-like posture. Robots that employ the LIPM often exhibit a walking pattern with bent knees or a slightly squat-like posture as a result of fixing their CoM to a specific height [19]. This is because the LIPM is a simplified model that approximates human walking dynamics using an inverted pendulum analogy [25].

In the LIPM, the robot's CoM is treated as a point mass located at a fixed height above the ground, and its motion is governed by principles of pendulum dynamics [26].

As was done for the LIPM simulation, the first step was to evaluate the human-likeness of the robot's gait. Figure 11 shows an overlay of the different stages of the gait from the optimisation simulation onto the same human gait cycle diagram.

Figure 11 shows an upright posture through the walking gait. The step length appears a lot larger than that of the LIPM simulation. Another advantage with this simulation in terms of human-likeness is the inclusion of the heel-strike and toe-off phases within the double support phases of the gait.

Although the gait appears more human-like than the LIPM simulation, the robot exhibits a skipping motion rather than a smooth steady-paced walk. This behaviour was observed during the simulation animation, where the robot rapidly transitions between double-support phases with minimal time spent in single-support. The static overlay in figure 11 does not fully capture this dynamic behaviour, as the skipping motion is more apparent when viewing the continuous animation. This skipping may be attributed to the limited DOF of the joints and the cost function's emphasis on maximising distance covered while maintaining stability, which results in the robot minimising single-support duration to reduce the risk of falling. The result does not show an adequate walking gait for a user in an assistive powered LLE. Although the LIPM covered far less ground, the walking pattern it achieved appeared much more suitable for carrying a human safely.

In table 3 a summary of the comparison of the results is presented. The distance walked in the optimisation simulation as opposed to the LIPM simulation was approximately 6 meters to 1 meter. The torque values for the ankle, knee and hip pitch are all lower in the optimisation simulation. Finally, it can also

http://dx.doi.org/10.69694/2309-8988/2026/v42a4

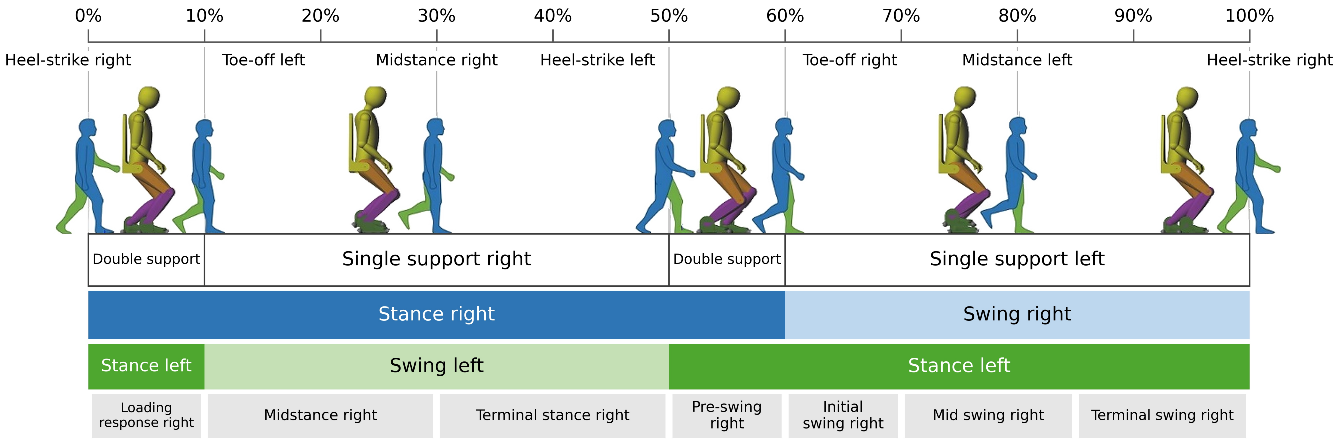

Figure 10 LIPM Simulation Gait Cycle Overlay [24]

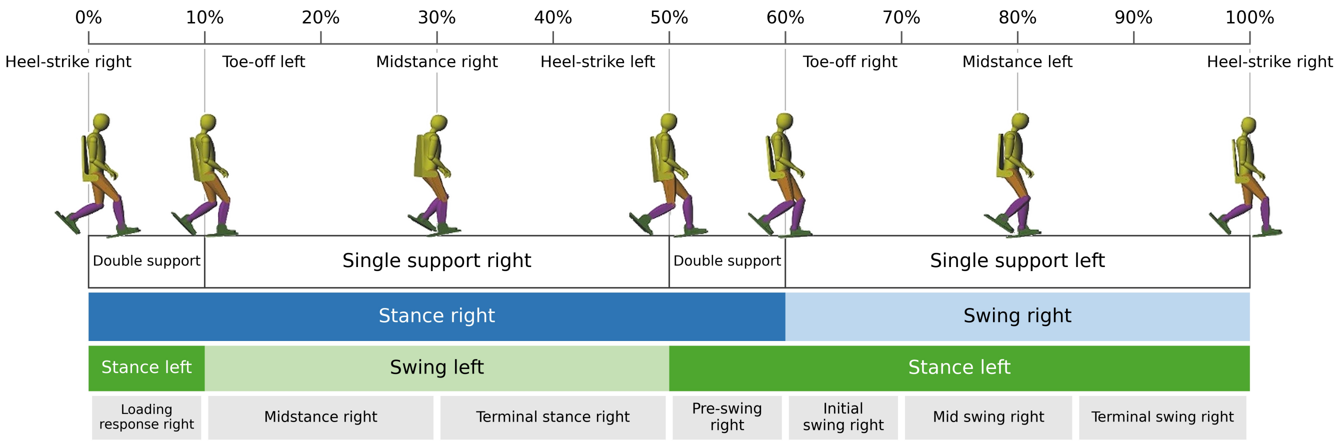

Figure 11 GA Optimisation Simulation Gait Cycle Overlay [24]

be seen that the optimisation simulation walks with a higher torso height.

Table 3 Results Comparison

Criteria	LIPM	Optimisation
Distance walked in 10s (m)	1.10	5.98
Minimum Z torso height (m)	0.73	0.83
Ankle pitch torque maximum (Nm)	984	700
Knee pitch torque maximum (Nm)	1894	700
Hip pitch torque maximum (Nm)	1188	700

In the following figures, the legend labels “Motion” and “Torque” refer to the two simulation methods: “Motion” represents the LIPM simulation using idealised joint motion without torque constraints, while “Torque” represents the GA optimisation simulation using torque-actuated joints with PID control and torque limits.

Figure 12 shows a simple displacement graph. Both the blue line representing the LIPM simulation, as well as the red line representing the optimisation simulation are more or less straight, indicating that in both simulations the powered LLE walked forwards without stumbling backwards. The Optimisation model was just able to progress a lot further in the given time.

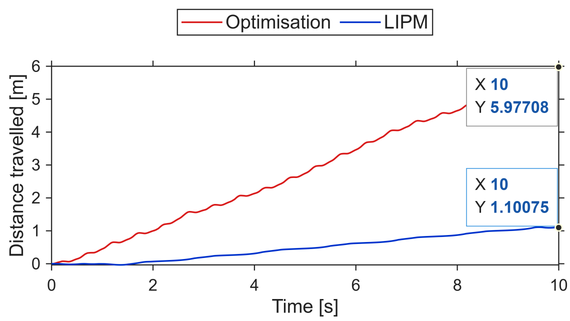

Figure 12 Distance Travelled Comparison

In figure 13 the difference between the LIPM simulation and the optimisation simulation in terms of torso height during a walking gait is made clear. Once in a walking position, the torso height of the LIPM simulation is virtually fixed, whereas in the optimisation simulation, it varies as the gait progresses. The optimisation simulation also walks with a higher torso height, even though it varies.

In figure 14 the optimisation simulation results had lower peak torque values for the ankle torque as opposed to the LIPM simulation. Flat troughs and peaks indicate areas where the optimisation simulation reached the torque limit that was

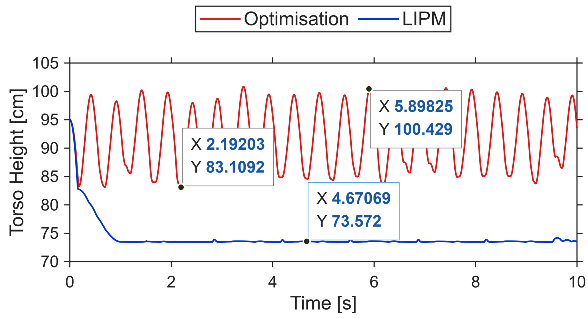

Figure 13 Torso Height Comparison

set at 700 Nm.

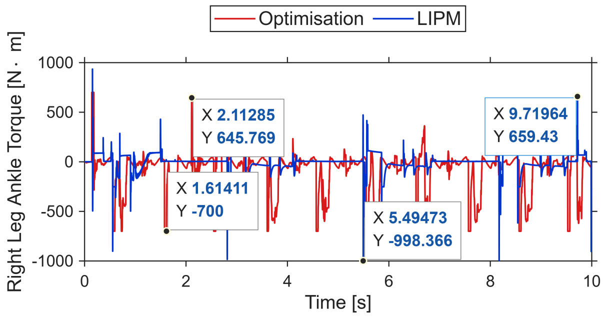

Figure 14 Ankle Torque Comparison

In figure 15 the optimisation simulation results had lower peak torque values for the knee torque as opposed to the LIPM simulation. Flat troughs and peaks indicate areas where the optimisation simulation reached the torque limit that was set at 700 Nm.

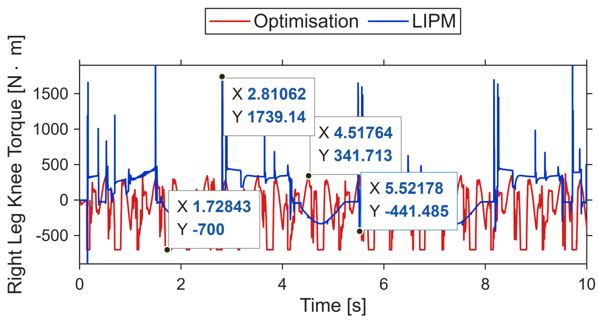

Figure 15 Knee Torque Comparison

In figure 16 the optimisation simulation results had lower peak torque values for the hip torque as opposed to the LIPM simulation. Flat troughs and peaks indicate areas where the optimisation simulation reached the torque limit that was set at 700 Nm.

The torque profiles shown in figures 14 to 16 exhibit less smooth trajectories compared to results from the original MathWorks Walking Robot demonstration. This difference can be attributed to several factors: the full-scale powered LLE model used in this research has significantly greater mass and inertia compared to the smaller demonstration robot, resulting in higher dynamic forces; the cost function prioritised stability and distance coverage over trajectory smoothness; and the 700 Nm torque limits imposed constraints that caused the flat peaks and troughs visible in the optimisation results.

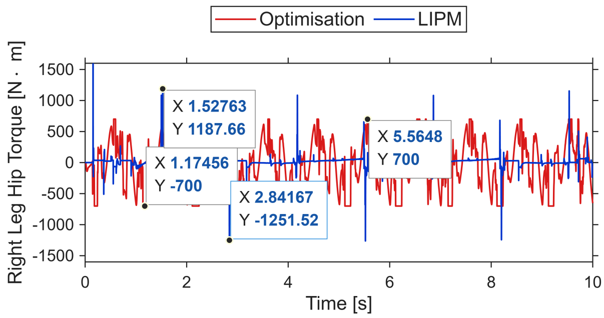

Figure 16 Hip Torque Comparison

Additionally, the MathWorks demonstration uses a simplified, lightweight model optimised for visual demonstration rather than realistic hardware prediction.

Although the data from the graphs in figures 12 to 16 appear to favour the optimisation simulation in terms of having lower torque values and a further distance walked with a more upright posture, it was noted that the optimisation result does not show an adequate walking gait for a user in an assistive powered LLE. Although the LIPM covered far less ground with higher torque values, the walking pattern it achieved appeared much more suitable for carrying a human safely.

5 Attempted Physical Validation of Aspects of the Numerical Modelling

This section covers the test results of an attempted physical validation of aspects of the numerical modelling. The test platform can be seen in figure 17.

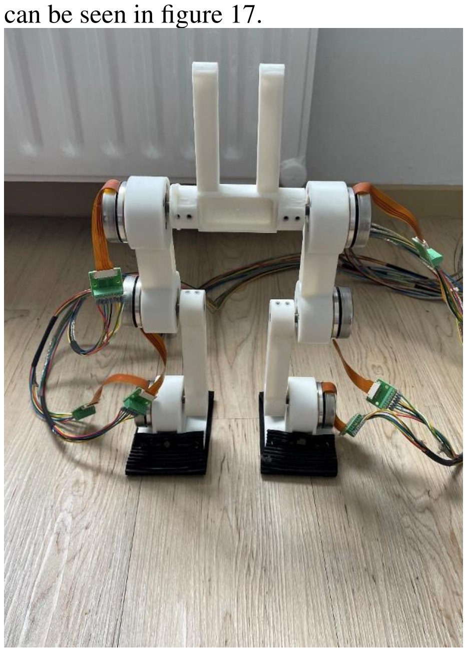

Figure 17 Test Platform Unrestricted

5.1 First attempt at a ground contact walk

The researcher tested if the test platform could perform a walk on a flat surface. It was observed that when the test platform attempted to take a step, the walking pattern was immediately out of sync. Several specific problems were identified: the leg would lift and overstep, causing the robot to stall in a position with the feet widely spread; alternatively, the leg would not be

http://dx.doi.org/10.69694/2309-8988/2026/v42a4

able to step at all, causing the robot to rock back and forth with both feet remaining on the floor; and the timing synchronisation between the left and right legs was inconsistent, preventing coordinated locomotion. These issues appeared to stem from the actuators' inability to achieve the required trajectory speeds under load. Although the peak torque of the motors was sufficient to allow for a steady walking gait in simulation, the dynamic response and torque-speed characteristics did not match real-world requirements. Additionally, the motors were equipped only with hall effect sensors, which provide limited resolution feedback suitable for commutation but not for precise position control. This lack of high-resolution encoder feedback meant that the controller could not accurately compensate for deviations from the intended trajectory.

It is acknowledged that this does not constitute a complete validation of the simulation. The test platform was a scaled-down model with different actuators and mechanical properties compared to the full-scale simulation, limiting direct comparability. The primary purpose of this physical test was to gain preliminary insights into the challenges of translating simulated walking gaits to hardware, rather than to provide a rigorous validation of the numerical model. A complete validation would require a full-scale platform with actuators capable of matching the torque and speed specifications identified in the simulation.

It should be noted that the actuators used in this test platform were selected based on availability rather than suitability. Bipedal walking robots typically require brushless DC motors with high torque density, low gear ratios, and favourable power-to-weight ratios to handle the dynamic demands of locomotion. The motors used in this research do not possess these characteristics and would not be recommended for future implementations of walking powered LLEs.

These limitations in terms of hardware accessibility constrained further experimentation. However, a future direction for this research should focus on overcoming these limitations by addressing hardware-related issues and refining the experimental setup.

To enhance the experimental setup and bridge the existing disparities between simulation and practical implementation, the adoption of more suitable actuators is paramount. These actuators should possess a high torque-to-weight ratio, enabling them to effectively manage the complex dynamics of bipedal locomotion. Furthermore, incorporating encoders into the actuator design is crucial for enabling precise position control and providing valuable feedback. The availability of accurate position feedback is essential as it empowers the researcher to meticulously analyse the platform's movements and behaviour. In addition to position feedback, it is advisable to integrate force/torque sensors to gain comprehensive insights into the interactions between the platform and its environment. These sensors will facilitate a more profound understanding of the forces at play during walking. While focusing on actuator and feedback enhancements, it is equally important to evaluate the entire experimental setup comprehensively. Potential areas for improvement could include refining mechanical components, optimising sensor configurations, and implementing advanced control algorithms to ensure the test platform operates seamlessly in a real-world environment.

6 Discussion and Conclusion

This research utilised mathematical modelling, simulation, and optimisation techniques to explore the hardware requirements for untethered walking of a powered lower-limb exoskeleton. Using Onshape, Simulink, and MATLAB, the study demonstrated the feasibility of achieving stability during walking, while revealing limitations in human-likeness and real-world implementation.

The findings underscore critical trade-offs in simplified models like the LIPM, which achieved stability but exhibited unnatural gait patterns. Optimisation techniques using GAs improved performance metrics such as distance coverage but raised challenges in achieving smooth, natural locomotion. Experimental tests further highlighted the discrepancies between simulation and real-world performance, primarily due to hardware constraints like actuator torque limitations.

These insights advance understanding of the intricate dynamics involved in exoskeleton locomotion and provide a foundation for future research. By addressing challenges in model fidelity, optimisation techniques, and hardware implementation, the field can move closer to achieving efficient and human-like exoskeleton designs, ultimately enhancing mobility and quality of life.

This study contributes to the field of powered LLE design by demonstrating the utility of mathematical modelling and optimisation in predicting hardware requirements and refining stability. However, several challenges were identified that warrant further investigation:

- Refinement of Simulation Models: Future work should address the limitations in simplified models like LIPM by incorporating biomechanical complexities and improving human-likeness in gait patterns.
- Optimisation for Smoothness: Enhancements in GAs and cost functions are necessary to balance performance metrics, such as distance coverage, with gait naturalness and smoothness.
- Hardware Innovations: Advancements in actuator technology, precise control mechanisms, and comprehensive sensor integration are crucial to bridge the gap between simulation and real-world performance.

By addressing these areas, the field can progress toward the design of efficient, stable, and human-like exoskeletons, ultimately achieving the vision of enhanced mobility and improved quality of life for users.

References

[1] A. Neven and W. Ectors. 'i am dependent on others to get there': Mobility barriers and solutions for societal participation by persons with disabilities. *Travel Behav. Soc.*, 30:302–311, jan 2023. doi: 10.1016/J.TBS.2022.10.009.

[2] Q. Magaqa, P. Ariana, and S. Polack. Examining the availability and accessibility of rehabilitation services in a rural district of south africa: A mixed-methods study. *Int. J. Environ. Res. Public Health*, 2021. doi: 10.3390/ijerph18094692.

[3] S. O. Schrade, G. Devittori, C. A. Easthope, C. Shirota, O. Lambercy, and R. Gassert. Exoskeleton knee compliance improves gait velocity and stability in a spinal cord injured user: A case report. *arXiv Prepr. arXiv1911.04316*, 2019.

[4] L. van Silfhout, A. J. F. Hosman, H. van de Meent, R. H. M. A. Bartels, and M. J. R. Edwards. Design recommendations for exoskeletons: Perspectives of individuals with spinal cord injury. *J. Spinal Cord Med.*, 46(2): 256–261, 2023. doi: 10.1080/10790268.2021.1926177/SUPPL_FILE/YSCM_A_1926177_SM6436.DOCX.

[5] K. Anam and A. A. Al-Jumaily. Active exoskeleton control systems: State of the art. *Procedia Eng.*, 41(December):988–994, 2012. doi: 10.1016/j.proeng.2012.07.273.

[6] J. L. Pons. Rehabilitation exoskeletal robotics. *IEEE Eng. Med. Biol. Mag.*, 29(3):57–63, may 2010. doi: 10.1109/MEMB.2010.936548.

[7] B. Chen, H. Ma, L. Y. Qin, F. Gao, K. M. Chan, S. W. Law, L. Qin, and W. H. Liao. Recent developments and challenges of lower extremity exoskeletons. *J. Orthop. Transl.*, 5:26–37, 2016. doi: 10.1016/j.jot.2015.09.007.

[8] A. Rodríguez-Fernández, J. Lobo-Prat, and J. M. Font-Llagunes. Systematic review on wearable lower-limb exoskeletons for gait training in neuromuscular impairments. *J. Neuroeng. Rehabil.*, 2021. doi: 10.1186/s12984-021-00815-5.

[9] C. Zhu, Q. Liu, W. Meng, Q. Ai, and S. Q. Xie. An attention-based CNN-LSTM model with limb synergy for joint angles prediction. In *2021 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 2021. doi: 10.1109/aim46487.2021.9517544.

[10] S. Wang, L. Wang, C. Meijneke, E. van Asseldonk, T. Hoellinger, G. Cheron, Y. Ivanenko, V. La Scaleia, F. Sylos-Labini, M. Molinari, F. Tamburella, I. Pisotta, F. Thorsteinsson, M. Ilzkovitz, F. Zanow, F. L. Haufe, A. Topini, I. Dimbwadyo-Terrer, M. Alcobendas-Maestro, A. Gil-Agudo, C. M. Bauer, S. Hesse, and H. van der Kooij. MINDWALKER. Semantic Scholar, 2015. URL `https://www.semanticscholar.org/paper/Design-and-Control-of-the-MINDWALKER-Exoskeleton-Wang-Wang/b6e3ab7d4b11f141418183e3bd14f93125d87d31/figure/0`. (accessed Aug. 15, 2020).

[11] T. Yan, M. Cempini, C. M. Oddo, and N. Vitiello. Review of assistive strategies in powered lower-limb orthoses and exoskeletons. *Rob. Auton. Syst.*, 64:120–136, 2015. doi: 10.1016/j.robot.2014.09.032.

[12] A. Esquenazi, M. Talaty, A. Packel, and M. Saulino. The ReWalk powered exoskeleton to restore ambulatory function to individuals with thoracic-level motor-complete spinal cord injury. *Am. J. Phys. Med. Rehabil.*, 2012. doi: 10.1097/phm.0b013e318269d9a3.

[13] K. S. G. Chua and C. W. K. Kuah. Innovating with rehabilitation technology in the real world. *Am. J. Phys. Med. Rehabil.*, 2017. doi: 10.1097/phm.0000000000000799.

[14] R. Riener. The Cybathlon promotes the development of assistive technology for people with physical disabilities. *J. Neuroeng. Rehabil.*, 2016. doi: 10.1186/s12984-016-0157-2.

[15] T. Gurriet, M. Tucker, A. Duburcq, G. Boeris, and A. D. Ames. Towards variable assistance for lower body exoskeletons. *IEEE Robot. Autom. Lett.*, 2020. doi: 10.1109/lra.2019.2955946.

[16] V. Huynh, E. Grondelle, R. Baud, J. Alban, O. Puget, A. Ortlieb, F. Raschellà, M. Bouri, H. Bleuler, Y. Allemand, and V. Crocher. Versatile dynamic motion generation framework: Demonstration with a crutch-less exoskeleton on real-life obstacles at the Cybathlon 2020 with a complete paraplegic person. *Front. Robot. AI*, 8, 2021. URL `https://www.frontiersin.org/articles/10.3389/frobt.2021.723780`.

[17] MathWorks Student Competitions Team. MATLAB and Simulink robotics arena: Walking robot, 2023. URL `https://github.com/mathworks-robotics/msra-walking-robot`. (accessed May 29, 2023).

[18] S. Castro. Walking robot control: From PID to reinforcement learning. MathWorks, 2019. URL `https://blogs.mathworks.com/student-lounge/2019/04/24/walking-robot-control/`. (accessed May 30, 2023).

[19] C. Brasseur, A. Sherikov, C. Collette, D. Dimitrov, and P.-B. Wieber. A robust linear MPC approach to online generation of 3D biped walking motion. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015. doi: 10.1109/humanoids.2015.7363423.

[20] S. Castro and B. Kim. Model-based control of humanoid walking - YouTube, 2020. URL `https://www.youtube.com/watch?v=jnJbXdp2wak&ab_channel=MATLAB`. (accessed Jun. 02, 2023).

[21] M. A. Ali, H. A. Park, and C. S. G. Lee. Closed-form inverse kinematic joint solution for humanoid robots. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 704–709, 2010. doi: 10.1109/IROS.2010.5649842.

[22] S. Castro. Walking robots, part 3: Trajectory optimization - MATLAB and Simulink robotics arena - YouTube, 2017. URL `https://www.youtube.com/watch?v=-dEX1SZOZEY&ab_channel=MATLAB`. (accessed Jun. 05, 2023).

[23] How the genetic algorithm works - MATLAB & Simulink - MathWorks United Kingdom. URL `https://uk.mathworks.com/help/gads/how-`

http://dx.doi.org/10.69694/2309-8988/2026/v42a4

the-genetic-algorithm-works.html. (accessed Feb. 04, 2024).

[24] Gait_cycle.jpg (1860×412). URL https://www.physio-pedia.com/images/1/17/Gait_cycle.jpg. (accessed Jun. 30, 2023).

[25] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, pages 239–246. IEEE, 2001.

[26] L. Li, Z. Xie, X. Luo, and J. Li. Trajectory planning of flexible walking for biped robots using linear inverted pendulum model and linear pendulum model. *Sensors*, 2021. doi: 10.3390/s21041082.

http://dx.doi.org/10.69694/2309-8988/2026/v42a4