International Journal of Control, Automation, and Systems 23(5) (2025) 1551-1562
http://dx.doi.org/10.1007/s12555-023-0822-8
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Preferred Jacobian Differentiation and Direct Collocation Methods for an Efficient and Accurate Walker Gait Optimization

Veronica Knisley, Kwonseung Cho, Kang-Woo Lee, and Pilwon Hur*

Abstract: Devices based on walker robotics research sometimes require a reference trajectory for the control systems of the devices to track. One valuable class of methods used to find an optimal reference trajectory is direct collocation, but even after selecting a method like direct collocation, several optimization design decisions remain. In order to determine the most desirable optimization settings, 600 optimizations were performed for the trajectory of a two degree-of-freedom (DOF) compass gait walker, and 200 optimizations were performed for a five-DOF link walker. These runs evaluated various combinations of optimization settings, including: numerical vs. symbolic vs. automatic differentiation; trapezoidal vs. Hermite Simpson collocation; numerical vs. symbolic calculation of joint accelerations; and inclusion or exclusion of joint accelerations in the decision variables. The different generated gaits were then compared in terms of computational efficiency and accuracy. The results showed that including joint accelerations as decision variables eliminated alternative gaits but increased computational complexity and variability. Symbolic acceleration evaluation was preferable when automatic differentiation was excluded. Automatic differentiation was shown to be significantly faster than the other two differentiation methods for both walking models. In addition, Hermite-Simpson collocation, although slower than trapezoidal, was the more accurate of the two approaches. These results can be applied to the derivation of optimal reference joint trajectories in future robotics applications.

Keywords: Computational efficiency, direct collocation, Jacobian, trajectory optimization.

1. INTRODUCTION

When designing robots to traverse challenging terrains [1] or developing wearable robotics to assist users with walking [2-5], it is crucial for roboticists to ensure that these systems follow a predetermined trajectory. This trajectory must incorporate desired joint angles and velocities during each step while complying with any physical limitations imposed by the system, environment, or user. To create such a gait trajectory, a variety of techniques are utilized. One such method is the heuristic approach, in which a continuous function is generated by passing through designated control points using spline or smoothing curves [2,6-8]. In situations where contact is crucial, impedance control is employed for terrain adaptation, accompanied by appropriate finite state machine rules [9-11]. Alternative approaches use oscillators or phase variables to account for internal or external intentions, such as recognizing human movement intention [12-16]. These techniques rely on rhythmic movement patterns to generate the gait trajectory. Ultimately, formulating a trajectory optimization problem is an effective method for producing a gait trajectory and optimizing the control input of the trajectory.

Trajectory optimization is one type of continuous-time optimal control problem (OCP). For practical situations like this, however, transcribing the OCP into a nonlinear program (NLP) allows the problem to be solved more efficiently. In simple cases, techniques called shooting methods can achieve this [17,18]; however, for a complex process like walking robots, direct collocation is a fairly popular choice [1,19-21]. In direct collocation, various points along the trajectory are taken to be collocation points. The system dynamics only need to be satisfied at these specific points in time. In this specific study, the collocation points are evenly spaced in time, but this is not required. After solving the NLP, polynomial splines can be used to approximate a continuous-time trajectory [22].

The dynamical constraints at the collocation points can be expressed as

Manuscript received December 6, 2023; revised April 9, 2025; accepted April 18, 2025. Recommended by Editor-in-Chief Hyo-Sung Ahn. The authors would like to thank Dr. Kenneth Chao, a former member of the authors' lab (HUR group), for mentoring Veronica Knisley and for introducing her to direct collocation.

Veronica Knisley is with Southwest Research Institute, San Antonio, TX 78238, USA (e-mail: veronica.knisley@swri.org). Kwonseung Cho, Kang-Woo Lee, and Pilwon Hur are with the Department of Mechanical and Robotics Engineering, Gwangju Institute of Science and Technology, Gwangju 61005, Korea (e-mails: kwonseung93@gm.gist.ac.kr, {rainleekw, pilwonhur}@gist.ac.kr). Veronica Knisley and Kwonseung Cho contributed equally to this work.
* Corresponding author.

$$x_{i+1} - x_i = \int_{t_i}^{t_{i+1}} f(x,u,t)\,dt, \tag{1}$$

where x is one of the states, i is the index of a given collocation point and $f(x,u,t)$ gives the system dynamics. Since we do not actually have a continuous expression for these dynamics, it is necessary to approximate this integral using a collocation method. The lower-order collocation method used in this study is trapezoidal collocation (TPZD); with this approach, the system dynamics and states are represented by linear and quadratic splines, respectively. Integrals are approximated as

$$x_{i+1} - x_i = \frac{h}{2}\left(f_i + f_{i+1}\right), \tag{2}$$

where h is the time step between two consecutive collocation points. Hermite-Simpson collocation (H-S) approximates the system dynamics and states one order higher than TPZD. With this method, integrals are approximated by

$$x_{i+1} - x_i = \frac{h}{6}\left(f_i + 4f_{i+\frac{1}{2}} + f_{i+1}\right), \tag{3}$$

where

$$x_{i+\frac{1}{2}} = \frac{1}{2}(x_i + x_{i+1}) + \frac{h}{8}(f_i - f_{i+1}). \tag{4}$$

Even after selecting direct collocation method, however, there are still multiple implementation decisions which can be made. For example, Chao and Hur [19] included joint accelerations, in addition to the standard joint positions and velocities, in their set of decision variables in order to increase the problem's sparsity. Some researchers, such as Nie and Kerrigan [23] and Pardo *et al.* [24], utilized different collocation methods in their optimizations. In the case of the latter, Hermite-Simpson collocation was shown to be a far more accurate method than the lower-order trapezoidal collocation method.

Implementation strategies in nonlinear programming (NLP) extend beyond problem formulation and often require user-defined gradients and Jacobians. Three primary differentiation techniques—symbolic differentiation (SD), numerical differentiation (ND), and automatic differentiation (AD)—offer distinct trade-offs in terms of accuracy, computational speed, and implementation complexity. SD provides exact derivatives using tools such as MATLAB or Mathematica but becomes impractical for highly complex expressions. ND employs finite differences, making it simpler to implement, though at the cost of reduced accuracy. AD combines the advantages of both methods, delivering high accuracy without the burden of storing large symbolic expressions [25].

Comparative studies have evaluated these techniques across various robotic applications. Giftthaler *et al.* [26] reported that AD outperformed ND in computational speed, while ND exhibited lower accuracy, when applied to the quadruped robot HyQ and a robotic manipulator. Durrbaum *et al.* [27] found AD to be more scalable in high-dimensional systems, whereas SD showed superior performance in lower-dimensional scenarios. Falisse *et al.* [28] further demonstrated that AD was both faster and more robust than ND in a biomechanical simulation context.

Despite these insights, the effects of differentiation methods on gait optimization for robotic (non-musculoskeletal) walkers remain largely unexplored. This gap is notable, given that computational efficiency is particularly critical in such systems [2,29]. Moreover, most prior work has focused on the isolated effects of individual factors, without considering the compounded impact of multiple implementation choices. Recent literature underscores the importance of examining trade-offs—such as those between accuracy and efficiency—when designing optimization frameworks [30,31].

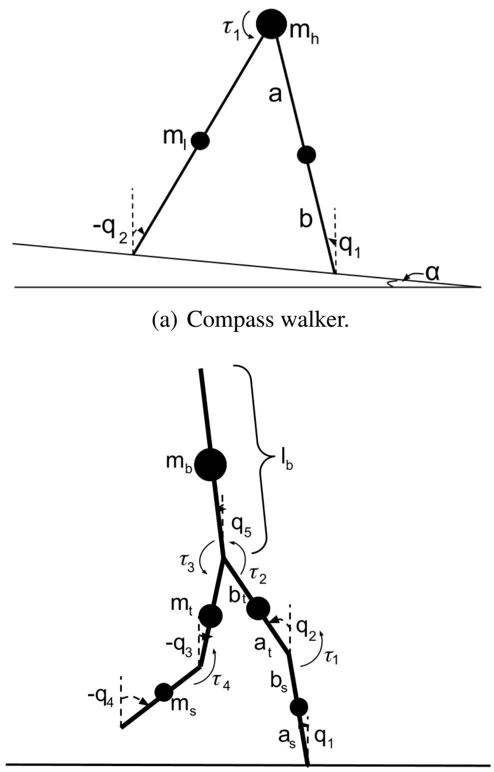
The goal of this study [32] is to examine the effects of different combinations of gradient and Jacobian differentiation methods, collocation techniques, and treatment of joint accelerations on the efficiency of walker trajectory optimization and the accuracy of the resulting optimal gaits. Two different walking models, the two degree-of-freedom (DOF) compass walker and the five-DOF kneed walker with torso, were examined. All investigations were carried out within an offline trajectory-optimization framework—no real-time solving was attempted. The intention behind this study is to help walker robotics researchers, and potentially others, to generate the most accurate trajectories in the least amount of time, likely taking into account tradeoffs between these two desired characteristics.

The rest of this paper is organized as follows: The Methods section describes the NLPs formed for each walker, as well as how these were implemented in IPOPT, which is an interior-point solver for nonlinear programs [33], and how the simulation experiment was structured. The Results section presents the most important pieces of data given in the experiment, while the Discussion section interprets these results. Lastly, the Conclusion and Future Work section summarizes the key findings of this study and suggests future areas of exploration related to this topic.

2. METHODS

2.1. Walking model

The first walking model which was used in this study was the compass walker [34,35], as shown in Fig. 1(a). This walking model has two degrees of freedom, which in this case are represented as the absolute angles of the stance and swing legs with respect to the vertical. Additionally, the presented model is underactuated, as its only



(a) Compass walker.

(b) Five-link walker.

Fig. 1. Walking models.

actuator is at the hip. The masses of each leg, as well as the hip, are point masses, and the walker's feet are point feet. Many of the parameters used are the same as those in [34].

The five-link walker [36] has five degrees of freedom and four degrees of actuation, as illustrated in Fig. 1(b). For this model, since the masses in each link are distributed, the links have inertias I_i, where i is the link number. Some of the inertial parameters used are based on the results found in [37].

2.2. Nonlinear program formulation

The general optimization formula for direct collocation is shown in (5).

$$\begin{aligned} x^* &= \underset{x}{\operatorname{argmin}} J(x), \\ \text{s.t.} \quad & x_{lb} \le x \le x_{ub}, \\ & H_{eq}(x) = 0, \\ & H_{iq}(x) \ge 0, \end{aligned} \tag{5}$$

where the system consists of M collocation points. For all runs, the decision variable x defined as $x = [q_i, \dot{q}_i, \tau_i, \Delta t]$ for $i \in [1, 2, \cdots, M]$. For some tests with the five-link walker, the accelerations of all joints at all collocation points ($\ddot{q}_i$) were also included in the decision variables. x_{lb} and x_{ub} are the lower and upper bounds for x, while $H_{eq}(x)$ and $H_{iq}(x)$ represent the equality and inequality constraints.

2.2.1 Objective function

There are different objective functions which can be used in the trajectory optimization of a walker, such as cost of transport, which takes into account power consumption and distance traveled; input torque squared [22]; some combination of the two [19]; or a robust modification of these [1]. In this study, the square of input torques is used as the cost function. The continuous-time version of this function is

$$J(x) = \int_0^T \sum_{i=1}^{m} \tau_i^2(t)dt, \tag{6}$$

where τ_i is the torque of actuated joint i and the integral is taken over the entire period T of a step. Since the walking models are underactuated, $m = 1$ for the compass walker, and $m = 4$ for the five-link walker. This integral must be approximated using a collocation method.

2.2.2 Constrained dynamics

Both walker walking models used in this study are hybrid systems, which include phases of both continuous and discrete dynamics [38]. The continuous dynamics are described with the Euler-Lagrange equations of motion

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = B\tau, \tag{7}$$

where M is a 2×2 inertia matrix, C is a 2×2 Coriolis and centrifugal force matrix, G is a 2×1 gradient of the system's potential energy, B is a 2×1 vector mapping the input of the system to each state, and τ is a scalar representing the hip torque.

2.2.3 Boundary constraints

There is also an assumed inelastic collision when the heel of the swing leg hits the ground. Given the pre-impact final velocities $\dot{q}_e^-$, the post-impact velocities $\dot{q}_e^+$ and impact forces F can be calculated using (8)

$$\begin{bmatrix} M_e & -J^T \\ J & 0 \end{bmatrix} \begin{pmatrix} \dot{q}_e^+ \\ F \end{pmatrix} = \begin{pmatrix} M_e \dot{q}_e^- \\ 0 \end{pmatrix}, \tag{8}$$

where M_e is the 4×4 inertia matrix for a compass walker with a free stance tip, and J is the 2×4 Jacobian matrix mapping the tip of the swing leg to each state.

2.2.4 Additional constraints

Additional constraints are needed in order to determine the optimal trajectory of the walking models. Contact constraints are essential to model contact conditions. In the contact constraints, constraints for the ground reaction forces and friction at the ground are introduced. The friction cone model, based on Coulomb friction, ensure that the contact forces remain within reasonable range. The contact constraints are expressed as in (9)

$$\begin{aligned} &\mu\lambda_z - \|\lambda_x\| \geq 0, \\ &\phi_z(q) = 0, \\ &J\dot{q} = 0. \end{aligned} \tag{9}$$

Continuity constraints ensure the gait generated must be periodic, meaning that the initial and final positions of each state must mirror each other.

$$\begin{aligned} &R(q_{initial}) - q_{final} = 0, \\ &x_{tip}(q_{final}) - x_{tip}(q_{initial}) \geq d_{dim}, \end{aligned} \tag{10}$$

where R is the relabeling matrix to swap joint variables between left and right leg. dim is the minimum horizontal travel distance that ensures an acceptable step length by constraining the initial leg-tip distance within a prescribed range.

2.3. Implementation of tested settings

The objective of this study was to compare the optimal walking gaits generated using different collocation methods (TPZD vs. H-S), Jacobian and gradient differentiation methods (ND vs. SD vs. AD), and dynamics evaluation methods (numerical vs. symbolic) in the NLP setup. Additionally, for the five-link walker, some runs included joint accelerations in the decision variables. This was because of concerns about the inversion of the dynamical model's inertia matrix. As is implied by (7), the joint accelerations in the model are calculated by

$$\ddot{q} = M(q)^{-1}(-C(q,\dot{q})\dot{q} - G(q) + B\tau). \tag{11}$$

In practice, this equation is implemented in MATLAB using the backslash operator (\), which avoids explicit matrix inversion and provides a more numerically stable solution to the linear system.

A symbolic matrix inversion for a system of more than just a few dimensions will yield an extremely complex expression, and numerical evaluation of the matrix could introduce inaccuracies. Including the accelerations in the decision variables eliminates the need to compute the inverse of the inertia matrix, and comparing all three options for the acceleration calculations will show how speed and consistency compare in each case.

Table 1 summarizes the different settings applied for each NLP setup. ADVs refers to "accelerations in decision variables." All tests on the five-link walker used trapezoidal collocation (TPZD), and only the differentiation method for the collocation constraints and the acceleration calculation method were modified. One of the tested configurations, originally planned to use numerical dynamics evaluation, was changed to use symbolic dynamics due to the large size and complexity of the Jacobian file generated with the numerical approach. As a result, this configuration used TPZD, automatic differentiation (AD), and symbolic dynamics.

Table 1. Experimental combinations of collocation methods, differentiation methods, and acceleration calculation methods.

Model	Method		
	Collocation	**Differentiation**	**Acceleration**
Compass walker	H-S	ND	Symbolic
	TPZD	ND	
	H-S	SD	
	TPZD	SD	
	H-S	AD	
	TPZD	AD	
Five-link walker	TPZD	SD	Sym Dyn
		ND	Sym Dyn
		ND	Num Dyn
		ND	ADVs
		AD	Sym Dyn

H-S: Hermite-Simpson collocation method, TPZD: Trapezoidal collocation method, ND: Numerical differentiation, SD: Symbolic differentiation, AD: Automatic differentiation, Sym Dyn: Symbolic dynamics evaluation, Num Dyn: Numerical dynamics evaluation, ADVs: Accelerations in decision variables.

It is also important to note that more combinations of differentiation and acceleration calculation methods could be tested. However, the selected combinations represent practical and commonly used configurations. For example, it would be difficult to use symbolic differentiation without a symbolic expression for the system dynamics, so that particular combination was not tested.

Collocation methods: The two collocation methods described in the introduction section were applied to the compass walker. For TPZD, (2) was used in both the objective function and collocation constraints. h was taken to be the time between two consecutive collocation points, although twice this value was used in the objective function. When TPZD was used with the five-link walker, h was always the time between two points. For all runs, h was taken to be a constant value, although this is not required.

The collocation constraints for H-S used (3). However, due to the indexing in MATLAB, $i+1$ became $i+2$ and $i+\frac{1}{2}$ became $i+1$. Additional constraints were included on the states and control values at the midpoints. The former were based on (4), while the latter were enforced by constraining the midpoint controls to fall along quadratic splines.

Differentiation methods: Three differentiation methods—numerical differentiation (ND), symbolic differentiation (SD), and automatic differentiation (AD)—were implemented to compute the constraint Jacobian and the gradient of the objective function. ND employed a central difference finite approximation by evaluating the constraints with forward- and backward-perturbed decision variables, modifying one variable at a time. A custom MATLAB function, *get_numerical_diff()*, was developed to automate this process for most applications, with the exception of the H-S + ND compass tests and all gradient computations, which followed similar procedures implemented in separate functions. To leverage the sparsity of the Jacobian, only the derivatives corresponding to nonzero elements (i.e., relevant rows and columns) were computed.

For SD, symbolic expressions of the constraints and the objective function were first constructed using MATLAB's symbolic class. Their derivatives were then computed using the *diff()* function and saved as an *.m*-file via the *matlabfunction()* command. The corresponding row and column indices were stored in a separate *.mat*-file. During each optimization iteration, the decision variables were passed to the saved derivative function, the index file was loaded, and the sparse Jacobian matrix was constructed using the *sparse()* function.

AD was implemented using the ADiGator package (v1.5) [39]. The decision variables and model parameters were formatted using ADiGator's input preparation functions. Key ADiGator utilities were employed to generate Jacobian and gradient functions compatible with IPOPT. During optimization, the inputs were provided as structured data in ADiGator's required format, and the resulting derivative information—returned as structured outputs—was used to extract row, column, and value arrays for the sparse matrix.

In addition to supplying the Jacobian, IPOPT requires a sparsity pattern matrix, where nonzero entries are indicated by ones. For the compass walker, this pattern was symbolically derived, with all decision variables set to one for the midpoint input torque constraints. For the five-link walker, the sparsity pattern was primarily obtained numerically, except for collocation constraints, which followed the same method as their respective Jacobian computations.

Handling accelerations: Computing acceleration is important since it describes the dynamic behavior of the walker. For the five-link walker, the accelerations of each joint at the collocation points were evaluated in three different ways: symbolically, numerically, and via introduction of extra decision variables. As mentioned previously, this is a necessary test to include because of the concerns with the inversion of a large inertia matrix. Even though the symbolic matrix inverse would be the most accurate, the expression will likely be extremely complex, meaning it would be difficult for even a computer to calculate and could result in an extremely large file size. Numerical "inversion," or to be more accurate, MATLAB's left division, can avoid this issue. However, it could potentially introduce numerical errors. Lastly, the accelerations were explicitly included in the decision variables. In this case, the dynamics constraints were satisfied by solving the Euler-Lagrange Equation explicitly.

2.4. Simulation experimental setup

To ensure a certain level of accuracy in this study, a baseline gait was determined for each walking model and compared to the corresponding optimal gait obtained from the experiments. The baseline for the compass walker was found using a passive forward simulation initiated from a fixed point on the walker's passive periodic trajectory on a three-degree downslope.The simulation was carried out with precision up to four decimal places. In contrast, the baseline for the five-link walker was obtained via an optimization process using IPOPT. A denser set of collocation points, 101, was used for this run in order to help it converge to the continuous-time gait. Additionally, to enhance accuracy, both accelerations and the Jacobians of the collocation constraints were evaluated symbolically.

The sum of the squared deviation between the value of each state on the baseline and on the experimental gait at certain percentages of gait completion were recorded.

All optimizations were performed using the mexIPOPT interface [40] in MATLAB (r2019a, MathWorks, Natick, MA, USA) on a Dell Latitude E6540 laptop. The solver *mumps* was used. In trajectory optimization for walker robots using direct collocation, the number of collocation points is crucial in balancing computational efficiency and solution accuracy. 11 and 21 collocation points were selected for comparison to evaluate this balance effectively. This choice aligns with standard walking motion generation practices, where 20 to 50 collocation points are typically used per half gait cycle (i.e., a single step) [20]. For the compass walker, one-hundred optimizations were completed with a randomized initial guess for each of the six test settings. Eleven collocation points were used, which means that there were eleven total points with TPZD and twenty-one total points with H-S. For the five-link walker, twenty optimizations were run with randomized initial guess for each of the five settings at two levels of refinement: eleven and twenty-one collocation points.

After running each optimization, plots were generated for spline-interpolated states and control signals, as well as walking tiles. Additionally, key pieces of information such as the optimal objective value, required CPU time, and baseline deviation "accuracy" measures were programmed to automatically be entered into an excel spreadsheet, sorted by the test setting.

The averages and sample standard deviations for the recorded quantities were calculated within Excel. Prior to conducting statistical comparisons, normality and equality of variance tests were performed. The data in this study did not satisfy the assumptions of normality and homogeneity of variance, necessitating the use of non-parametric statistical methods.

In order to determine whether there were statistical differences between the means of data generated using different collocation, differentiation, or dynamics evaluation methods, Kruskal-Wallis tests were performed. If a significant difference was detected, Mann-Whitney U tests were conducted as a post-hoc analysis, with Bonferroni correction applied to account for multiple comparisons. A significance level of $\alpha = 0.05$ was used, and p-values at or below this threshold were considered statistically significant.

3. RESULTS

3.1. Compass walker

Based on the optimization settings for the compass walker, one hundred optimizations were completed with randomized initial guesses for each of the six settings, which varied in collocation methods, differentiation approaches, and acceleration treatments. While most of the generated walking gaits generally followed the baseline trajectory, some converged toward an alternative gait. The H-S + ND and TPZD + ND settings produced the highest number of alternative gaits, each generating 17 instances. The H-S+SD setting resulted in 14 alternative gaits, followed by TPZD + SD and TPZD + AD, each yielding 13 instances. The H-S + AD setting generated the fewest alternative gaits, with 8 occurrences. The data was analyzed both with and without these alternative gaits.

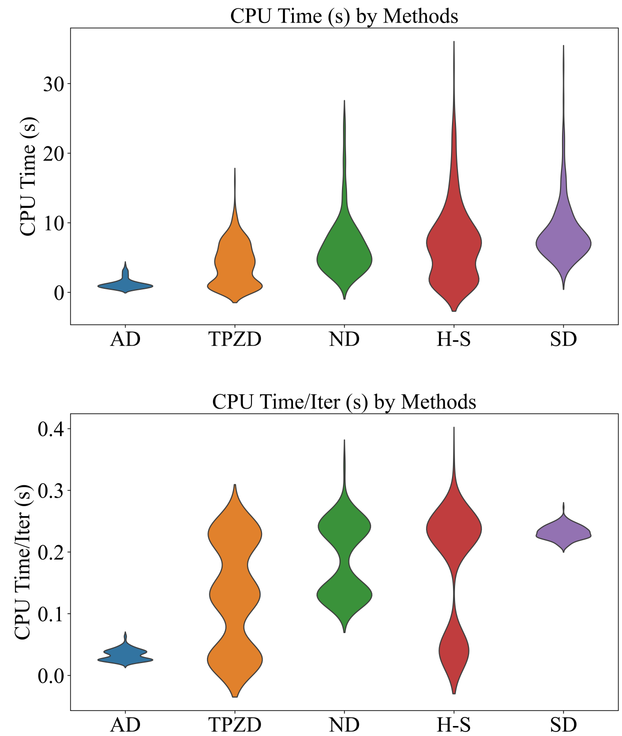
The key findings pertain to computational efficiency. Statistical tests showed that, when comparing individual settings, all configurations exhibited statistically significant differences in both CPU Time and average CPU Time per iteration, except for the cases of H-S and ND (Fig. 2(a)).

In the comparison of setting combinations, there was no significant difference in CPU Time for the H-S + ND and H-S + SD combinations. Similarly, for average CPU Time per iteration, all combinations exhibited significant differences except for H-S + SD and TPZD + SD (Fig. 2(b)).

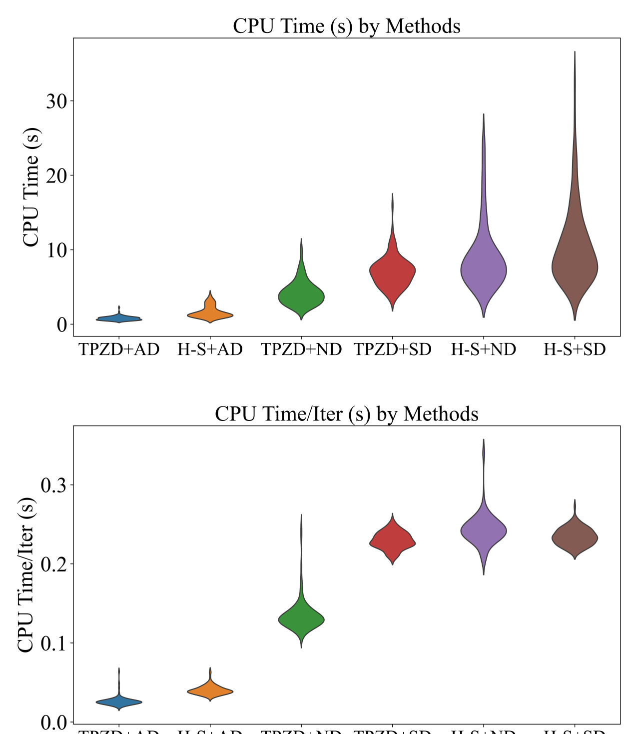
This study evaluated accuracy based on deviation from the expected baseline. After excluding outliers, the analysis revealed statistically significant differences among the collocation methods ($p < 0.001$). The H-S method produced errors up to 1,000 times smaller than those of the TPZD method. In contrast, the differentiation methods showed no statistically significant differences (Fig. 3).

3.2. Five-link walker

Similar data were collected for the five-link walker, with twenty optimization runs performed for each of the



(a) Effect of individual factors.



(b) Effect of factor combinations.

Fig. 2. Comparison of computational performance based on collocation and differentiation methods in compass walker optimizations.

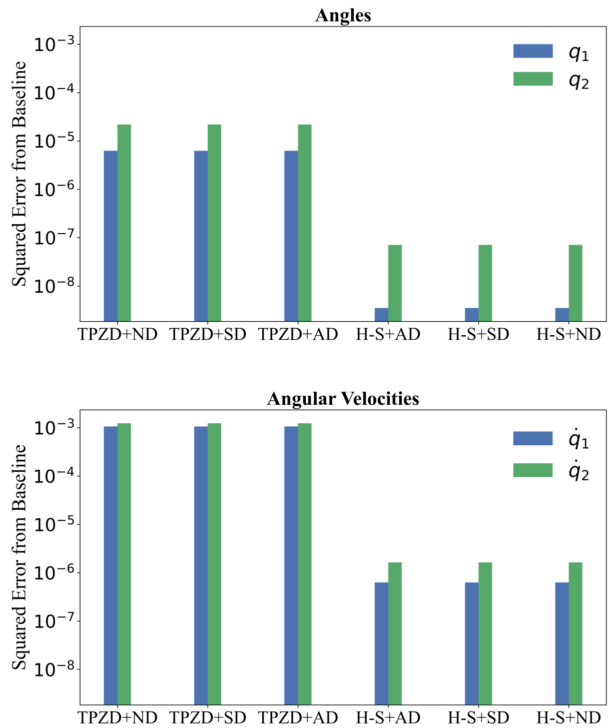
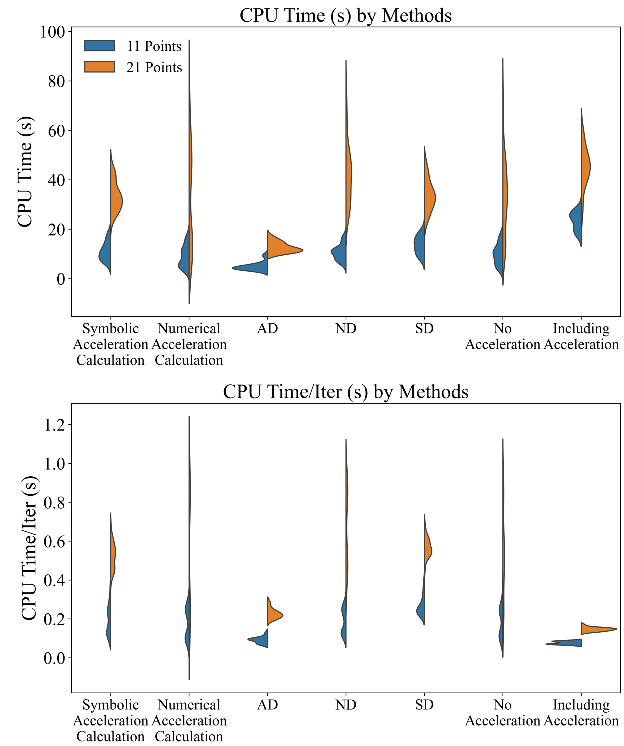
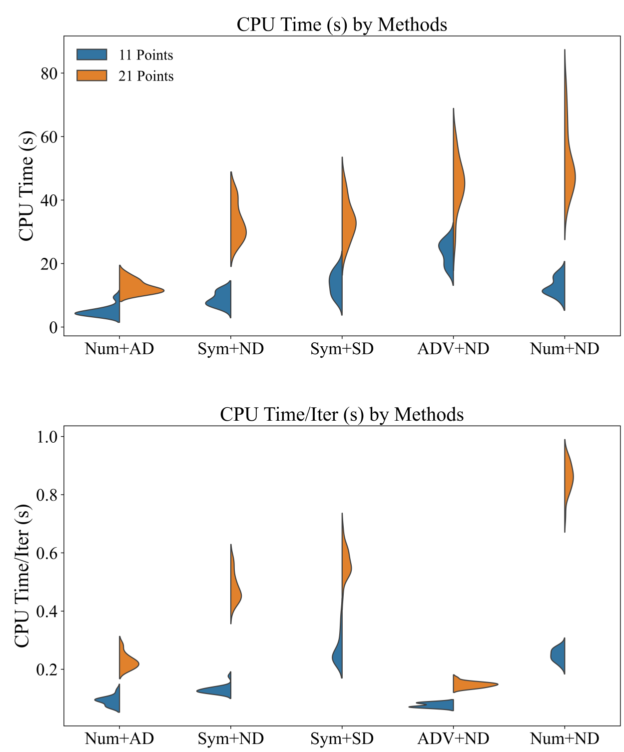


Fig. 3. Squared error with respect to the baseline gait for compass walker runs without alternative gaits. The variables q_1 and q_2 correspond to the angles as described in Fig. 1(a).



(a) Effect of individual factors.



(b) Effect of factor combinations.

Fig. 4. Comparison of computational performance based on collocation, differentiation, and acceleration methods in five-link walker optimizations.

five settings and two collocation point configurations. As with the compass walker, some runs converged to alternative gaits. However, not all alternative gait runs followed the same trajectory. The number of alternative gaits was generally lower with 21 collocation points compared to 11, suggesting that increased constraints reduced local minima. Among the tested settings, some conditions resulted in more frequent alternative gaits, while others showed little to no occurrence. A key finding is that no alternative gaits were observed when joint accelerations were included as decision variables, indicating that this constraint significantly influenced gait variability.

The time-related measures (i.e., CPU Time and the average CPU Time per iteration) for the five-link walker are shown in Fig. 4. It is important to note that the three differentiation methods were not all used with the same dynamics calculation methods. Specifically, the following five combinations were compared: i) Sym + SD, ii) Sym + ND, iii) Num + ND, iv) ADV + ND, and v) Sym + AD. These combinations were chosen due to practical reasons. For example, symbolic differentiation could not be performed when the accelerations were computed numerically. ADV + ND was chosen to examine how adding the accelerations as decision variables compares to calculating them symbolically or numerically for the same differentiation method (i.e., ii vs. iii vs. iv). Finally, Sym + AD was added to investigate the effect of differentiation methods (i.e., i vs. ii vs. v).

A comprehensive statistical analysis demonstrated that computational performance is predominantly governed by the choice of differentiation strategy and the inclusion of acceleration as a decision variable, rather than by the method employed for acceleration calculation. For both 11 and 21 collocation points, no statistically significant differences were observed between symbolic and numerical acceleration in terms of CPU time or CPU time per iteration, thereby challenging prior assumptions regarding the relative efficiency of these approaches. In contrast, AD consistently yielded the lowest computational costs across all conditions, significantly outperforming both SD and ND at each collocation level ($p < 0.001$), while no significant differences emerged between SD and ND. Notably, the inclusion of acceleration as a decision variable led to a substantial increase in computational burden, with $p < 0.001$ for both CPU and iteration time comparisons, underscoring the critical impact of model formulation choices. These effects are illustrated in Fig. 4(a).

The comparison of factor combinations showed statistically significant differences in CPU Time and CPU Time per iteration for 11 collocation points, except between Sym + SD and Num + ND. For 21 collocation points, CPU Time differed significantly across all combinations except Sym + SD vs. Sym + ND and Num + ND vs. ADV + ND, while CPU Time per iteration was significantly different for all cases. Differences between 11 and 21 collocation points were significant across all methods (Fig. 4(b)).

Whether including or excluding optimization runs which resulted in an alternative gait, there were not many differences between data sets with regards to accuracy. The runs including the accelerations had a larger standard deviation by an order of two to four than those excluding them. This is shown in Fig. 5, which uses data for 11 collocation points. On the other hand, these runs were also the only ones where there were no alternative gaits.

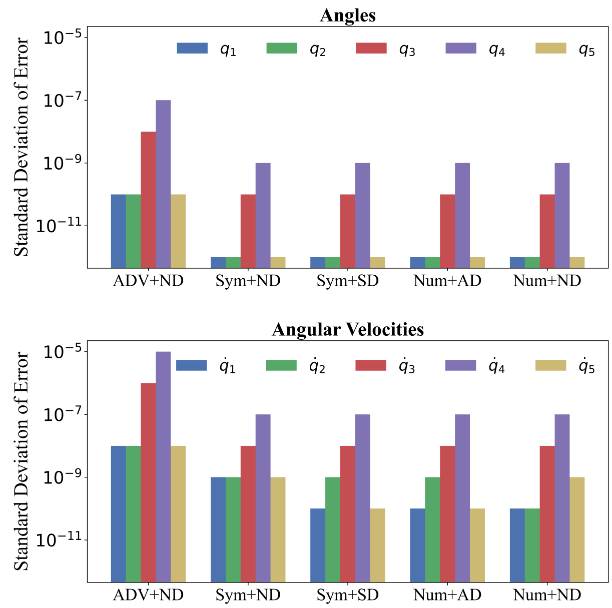


Fig. 5. Standard deviation of error with respect to baseline gait for 11 collocation points (Fig. 1).

4. DISCUSSION

The data for the compass walker show that AD was significantly faster than the other two differentiation methods. This is likely due to the specific algorithms employed in this method. TPZD was also shown to be faster than H-S for each of the given differentiation methods. This was likely due to the smaller number of constraints and decision variables required for those runs, as well as the fact that the dynamics at the midpoints of segments did not need to be calculated. Differences between the collocation methods for the ND runs could have been affected slightly by the slightly different implementations of ND, as described in the Methods section. With regards to the accuracy of the compass gait simulations, and based on previous studies from other researchers [24], it was expected that ND would be less accurate than the other two methods. However, this might not have been the case in this experiment because of the small perturbation size used (10^{-6}) and the simplicity of the model.

Before concluding which methods would make the ideal combination, it is also worthwhile to consider the implementation difficulty of using different factors. Other than the fact that there were fewer constraints needed for the TPZD optimization than the H-S optimization, there were very few differences between the implementation of these two collocation methods. However, as described in the Methods section, there were significant differences in the implementation of the three different differentiation methods. Due to the reuse of code, ND could be considered the easiest to implement. Also, since the ND implementation would call the constraint or objective functions, debugging only required changing the contents of these functions in one location, which was not the case for AD or SD. These latter two methods required the need to re-derive the Jacobian functions each time the constraints were modified. Although these two methods both involved generating new functions for the Jacobians, the black-box nature of ADiGator's built-in functions, as well as their speed, were benefits over SD. One benefit of SD over AD, though, was that it did not require setting up special classes of inputs, as only MATLAB's standard symbolic variables were needed.

The above discussion makes clear that some tradeoffs are required in order to decide which settings are best. Since AD is by far the fastest of the differentiation methods tested, and since it is fairly simple to implement, it would be the ideal choice. Although TPZD is faster than H-S, both settings are so fast that time is perhaps not the most critical factor here. Based on accuracy, this study recommends the use of H-S collocation.

As was the case with the compass walker, AD was shown to be the fastest method for the five-link walker. This fact could have potentially skewed the mean of the runs using numerical dynamics calculation. In an earlier stage of this study, before AD was included, symbolic dynamics calculation was perceived to be the faster method. When examining the inclusion or exclusion of the joint accelerations from the decision variables, the overall CPU time was slower with their inclusion, but the time per iteration was faster. In the same previous stage of this study, when 21 collocation points were used, including the accelerations resulted in the second slowest time among the factors. The per-iteration time was likely due to the increased sparsity of the problem. However, due to the fact that there were more decision variables with that setup, it likely would have taken more iterations to solve the problem.

Including joint accelerations as decision variables significantly impacted the optimization results for the five-link walker. Explicitly incorporating accelerations eliminated alternative gaits, potentially simplifying the search space and preventing convergence to local minima or undesired trajectories. Additionally, increasing the number of collocation points from eleven to twenty-one further reduced alternative gait occurrences, suggesting that more constraints effectively diminished local minima. Computational efficiency, assessed through CPU time and average CPU time per iteration, varied significantly among differentiation and dynamics calculation methods. Generally, including accelerations increased computational demands and variability, although specific method pairs ('Sym + SD and Sym + ND') displayed similar performance. Conversely, alternative gaits largely unaffected accuracy metrics, yet including accelerations led to notably more significant variability in accuracy measures.

One last point to consider with this walker is implementation. Compared to the compass walker, the symbolic expressions for the collocation constraints of the five-link walker were extremely complicated. This is a primary reason why numerical dynamics evaluation was used with AD. The implementation strategy of symbolic differentiation was also more complicated because of this, namely because the Jacobian of the collocation constraints were split into many different parts which had to be assembled. The significant increase of complexity with a 3-DOF change in system complexity does not bode well for the generalization of symbolic differentiation to more complex models.

Based on the preceding points, it seems fair to select the AD framework to be the most desirable. This setting had a very quick per-iteration and total time to solution, and its accuracy was comparable to that of most of the other settings.

5. CONCLUSION

The collocation constraints for the five-link walker were significantly more complex than those of the compass walker. This complexity motivated the use of numerical dynamics evaluation with AD. The implementation strategy of symbolic differentiation was also more complicated because of this, namely because the Jacobian of the collocation constraints were split into many different parts which had to be assembled. The significant increase of complexity with a 3-DOF change in system complexity does not bode well for the generalization of symbolic differentiation to more complex models. Through the process of repeatedly optimizing a walker walking gait, different Jacobian differentiation, collocation, and acceleration handling methods were determined to be more efficient or accurate than others. In order to quickly calculate a relatively accurate trajectory for a compass walker, automatic differentiation can be used to calculate the constraint Jacobian and objective gradient, while Hermite-Simpson collocation can be employed within the constraints. For a more complicated walker, this same differentiation method can be used. The better acceleration calculation method is more ambiguous. Explicitly including joint accelerations as decision variables effectively eliminates alternative gait solutions but increases computational complexity and variability in optimization results. The results of this study can aid others in efficiently determining gaits which are to be tracked by various robots and devices.

There are several ways to potentially extend this study. First, different combinations of settings could be tested. For example, it would be interesting to see how the compass walker optimization would run if joint accelerations were included in the decision variables. Additionally, it could be worthwhile to test more discretization methods. Lastly, it would be interesting to look at the effect of using different computer languages, such as the computing language Julia, on the results.

DECLARATIONS

Conflict of Interest

The authors declare that they have no conflict of interest. Pilwon Hur is an Associate Editor of the International Journal of Control, Automation, and Systems. Associate Editor status has no bearing on editorial consideration.

Authors' Contributions

Veronica Knisley set up and ran the simulations, analyzed the results, and contributed to writing the original draft. Kwonseung Cho performed the statistical analysis, interpreted the results, contributed equally to writing the original draft as Veronica Knisley, and revised the manuscript according to the reviewers' comments. Kang-Woo Lee performed the statistical analysis, interpreted the results, contributed to writing the original draft, and worked on revising the manuscript according to the reviewers' comments. Pilwon Hur supervised the project, conceptualized the study, developed the methodology, contributed to the data analysis, and contributed to writing and revising the manuscript.

Funding

This work was partially supported by the Translational Research Program for Rehabilitation Robots (#NRCTR-EX23002, #NRCTR-EX23003), National Rehabilitation Center, Ministry of Health and Welfare, Korea.

REFERENCES

[1] H. Dai and R. Tedrake, "Optimizing robust limit cycles for legged locomotion on unknown terrain," *Proc. of the IEEE Conference on Decision and Control*, Maui, USA, pp. 1207-1213, 2012.

[2] V. Paredes, W. Hong, S. Patrick, and P. Hur, "Upslope walking with transfemoral prosthesis using optimization based spline generation," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Daejeon, Korea, pp. 3204-3211, 2016.

[3] S. Patrick, N. A. Kumar, W. Hong, and P. Hur, "Biomechanical impacts of toe joint with transfemoral amputee using a powered knee-ankle prosthesis," *Frontiers in Neurorobotics*, vol. 16, 809380, 2022.

[4] D. Kim, Y.-P. Hong, and K.-S. Kim, "Bipedal walking and impact reduction algorithm for a robot with pneumatically driven knees," *International Journal of Control, Automation, and Systems*, vol. 19, no. 12, pp. 3937-3946, 2021.

[5] J. Park, D. Lee, and K. Kong, "Shank shock absorption mechanism and associated gait pattern design for reduction of ground impact of a powered exoskeleton," *International Journal of Control, Automation, and Systems*, vol. 21, no. 6, pp. 1959-1969, 2023.

[6] T. Phan-Huu, V. H. Nguyen, and U. Konigorski, "A time-optimal trajectory generation approach with non-uniform B-splines," *International Journal of Control, Automation, and Systems*, vol. 19, no. 12, pp. 3947-3955, 2021.

[7] E. Z. Dong, D. D. Wang, J. G. Tong, C. Chen, and Z. H. Wang, "A stable gait planning method of biped robot based on ankle motion smooth fitting," *International Journal of Control, Automation, and Systems*, vol. 16, no. 1, pp. 284-294, 2018.

[8] W. Hong, V. Paredes, K. Chao, S. Patrick, and P. Hur, "Consolidated control framework to control a powered transfemoral prosthesis over inclined terrain conditions," *Proc. of 2019 International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 2838-2844, 2019.

[9] N. A. Kumar, S. Patrick, W. Hong, and P. Hur, "Control framework for sloped walking with a powered transfemoral prosthesis," *Frontiers in Neurorobotics*, vol. 15, 790060, 2022.

[10] N. A. Kumar, W. Hong, and P. Hur, "Control of a transfemoral prosthesis on sloped terrain using continuous and nonlinear impedance parameters," *Proc. of 2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 3219-3225, 2021.

[11] N. A. Kumar, W. Hong, and P. Hur, "Impedance control of a transfemoral prosthesis using continuously varying ankle impedances and multiple equilibria," *Proc. of 2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 1755-1761, 2020.

[12] W. Hong, N. A. Kumar, S. Patrick, S. Moon, and P. Hur, "A feasibility study of piecewise phase variable based on variable toe-off for the powered prosthesis control: A case study," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2590-2597, 2023.

[13] W. Hong, J. Lee, and P. Hur, "Piecewise linear labeling method for speed-adaptability enhancement in human gait phase estimation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 628-635, 2023.

[14] W. Hong, J. Lee, and P. Hur, "Effect of torso kinematics on gait phase estimation at different walking speeds," *Frontiers in Neurorobotics*, vol. 16, 807826, 2022.

[15] J. Lee, W. Hong, and P. Hur, "Continuous gait phase estimation using LSTM for robotic transfemoral prosthesis across walking speeds," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 1470-1477, 2021.

[16] W. Hong, N. A. Kumar, and P. Hur, "A phase-shifting based human gait phase estimation for powered transfemoral prostheses," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5113-5120, 2021.

[17] M. P. Kelly, "Transcription methods for trajectory optimization: A beginners tutorial," Technical report, *arxiv preprint* arXiv:1707.00284, 2017.

[18] R. W. H. Sargent, "Optimal control," *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 361-371, 2000.

[19] K. Chao and P. Hur, "A step towards generating human-like walking gait via trajectory optimization through contact for a bipedal robot with one-sided springs on toes," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver, Canada, pp. 4848-4853, 2017.

[20] K. Chao, *Bipedal Walking Analysis, Control, and Applications Towards Human-like Behavior*, Ph.D. Dissertation, Texas A&M University, 2019.

[21] A. Hereid, C. M. Hubicki, E. A. Cousineau, and A. D. Ames, "Dynamic humanoid locomotion: A scalable formulation for HZD gait optimization," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 370-387, 2018.

[22] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849-904, 2017.

[23] Y. Nie and E. C. Kerrigan, "How should rate constraints be implemented in nonlinear optimal control solvers?" *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 362-367, 2018.

[24] D. Pardo, L. Moller, M. Neunert, A. W. Winkler, and J. Buchli, "Evaluating direct transcription and nonlinear optimization methods for robot motion planning," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 946-953, 2016.

[25] A. Güne, G. Baydin, B. A. Pearlmutter, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *Journal of Machine Learning Research*, vol. 18, pp. 1-43, 2018.

[26] M. Giftthaler, M. Neunert, M. Stäuble, M. Frigerio, C. Semini, and J. Buchli, "Automatic differentiation of rigid body dynamics for optimal control and estimation," *Advanced Robotics*, vol. 31, no. 22, pp. 1225-1237, 2017.

[27] A. Dürrbaum, W. Klier, and H. Hahn, "Comparison of automatic and symbolic differentiation in mathematical modeling and computer simulation of rigid-body systems," *Multibody System Dynamics*, vol. 7, no. 4, pp. 331-355, 2002.

[28] A. Falisse, G. Serrancolí, C. L. Dembia, J. Gillis, and F. De-Groote, "Algorithmic differentiation improves the computational efficiency of OpenSim-based trajectory optimization of human movement," *PLoS ONE*, vol. 14, no. 10, pp. 1-19, 2019.

[29] A. C. Hildebrandt, R. Wittmann, F. Sygulla, D. Wahrmann, D. Rixen, and T. Buschmann, "Versatile and robust bipedal walking in unknown environments: real-time collision avoidance and disturbance rejection," *Autonomous Robots*, vol. 43, no. 8, pp. 1957-1976, 2019.

[30] J. Ismail and S. Liu, "Efficient planning of optimal trajectory for a furuta double pendulum using discrete mechanics and optimal control," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10456-10461, 2017.

[31] Z. Manchester, N. Doshi, R. J. Wood, and S. Kuindersma, "Contact-implicit trajectory optimization using variational integrators," *International Journal of Robotics Research*, vol. 38, no. 12-13, pp. 1463-1476, 2019.

[32] V. Knisley, *Investigating Differences Between Bipedal Walking Gaits Generated Using Different Jacobian Calculation and Direct Collocation Approaches*, Master's thesis, Texas A&M University, 2020.

[33] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25-57, 2006.

[34] L. B. Freidovich, U. Mettin, A. S. Shiriaev, and M. W. Spong, "A passive 2-DOF walker: Hunting for gaits using virtual holonomic constraints," *Proc. of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, pp. 1202-1208, 2008.

[35] Z. Liu and Y. Tian, "Some control strategy on the compass gait biped," *Proc. of IMACS Multiconference on Computational Engineering in Systems Applications*, Beijing, China, pp. 1712-1718, 2006.

[36] X. Mu and Q. Wu, "A complete dynamic model of five-link bipedal walking," *Proc. of the American Control Conference*, vol. 6, pp. 4926-4931, 2003.

[37] R. F. Chandler, C. E. Clauser, J. T. McConville, H. M. Reynolds, and J. W. Young, "Investigation of inertial properties of the human body," Technical report, Aerospace Medical Research Laboratory, 1975.

[38] A. Goswami, B. Thuilot, and B. Espiau, "Compass-like biped robot part I : Stability and bifurcation of passive gaits," Technical report, INRIA, RR-2996, 1996.

[39] M. J. Weinstein and A. V. Rao, "Algorithm 984: Adigator, a toolbox for the algorithmic differentiation of mathematical functions in matlab using source transformation via operator overloading," *ACM Transactions on Mathematical Software (TOMS)*, vol. 44, no. 2, 21, 2017.

[40] E. Bertolazzi, mexIPOPT, 2018. Available: https://github.com/ebertolazzi/mexIPOPT. [Accessed April 26, 2025]

Veronica Knisley received her B.S. and M.S. degrees in mechanical engineering from Texas A&M University in 2018 and 2020, respectively. She is currently a researcher at Southwest Research Institute, San Antonio, TX.

Kwonseung Cho received his B.S. degree in control and instrumentation engineering from Pukyong National University, Busan, Korea, in 2020, and received an M.S. degree in mechanical engineering from Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, in 2023, and is currently a Ph.D. candidate in mechanical and robotics engineering at GIST, Gwangju, Korea. His research interests include bipedal locomtion, control of exoskeleton, nonlinear optimization, human-robot interaction, mutual trust, and control theory.

Kang-Woo Lee received his Ph.D. degree in human anatomy from Yonsei University, Seoul, Korea, in 2019, and a B.Sc. degree in physics from Incheon National University in 2011. From 2020 to 2023, he was a Research Assistant Professor at the College of Dentistry, Yonsei University. Since 2023, he has been a Postdoctoral Researcher in the School of Mechanical and Robotics Engineering at the Gwangju Institute of Science and Technology (GIST), Korea. His research interests include biomechanics, biomedical engineering, and computational modeling, with a focus on applying engineering principles to improve human health and rehabilitation.

Pilwon Hur received his B.S. degree in mechanical engineering from Hanyang University in 2004, an M.S. degree in mechanical engineering from Korea Advanced Institute of Science and Technology in 2006, an M.S. degree in applied mathematics in 2010, and a Ph.D. degree in mechanical engineering from the University of Illinois at Urbana-Champaign in 2010. He is currently an Associate Professor in the Department of Mechanical and Robotics Engineering at Gwangju Institute of Science and Technology (GIST) and Director of the GIST-SAMSUNG Intelligent Motor Talent Training Center. Previously, he served as an Assistant Professor at Texas A&M University (2014-2020) and a Postdoctoral Research Fellow at the Center for Ergonomics, University of Wisconsin-Milwaukee (2010-2014). His research interests include human-robot interaction, rehabilitation robotics, free energy principle in human neuroscience, bipedal walking, prosthetics and exoskeletons, and sensory augmentation.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.