Artificial Intelligence-Based Sensorless Force Estimation for Smart Walkers: A Robust LSTM Disturbance Observer

Muhammad Ishaq, Francesco Cancelliere, Giuseppe Sutera, Dario Calogero Guastella, Giovanni Muscato

***Abstract*—Active smart walkers assist older adults with mobility. Traditional devices use physical force sensors to measure user intent. Physical sensors increase manufacturing costs and introduce mechanical fragility. Sensorless force estimation offers an alternative using disturbance observers to calculate external forces based on internal motor data. Traditional mathematical observers require exact physical parameters and velocity derivatives. Sensor noise and varying physical mass parameters cause estimation errors in these mathematical models. This paper presents a sensorless force estimation framework using a long short-term memory network. The network maps motor velocities, torques, and pitch angles to user forces. This approach removes the need for velocity derivatives and adapts to physical parameter changes. The study evaluates the framework through MATLAB simulations configured with the exact mechanical parameters of a physical prototype. The network filters sensor noise and tracks user force under a 6-kilogram mass mismatch. The integrated admittance control system separates user input from gravitational forces on inclined paths. The system executes differential turns and includes an automatic braking function that holds the walker stationary on a slope without user input. This study provides a framework for sensorless control in assistive mobility devices.**

***Index Terms*—Elderly Care, Sensorless Force Estimation, LSTM, Disturbance Observer, Smart Walker.**

I. Introduction

Active smart walkers (SWs) provide mobility assistance for older adults. These devices help prevent falls and maintain user independence [1]. SWs interpret user intent to deliver active physical support. Traditional SWs use multi-axis force and torque sensors on the handlebars to read user push and pull forces [2]. However, these physical sensors present hardware challenges. Load cells increase manufacturing costs and introduce mechanical fragility. Moreover, force sensors suffer from signal noise and require routine calibration [3]. Some researchers propose optical sensors to replace strain gauges to reduce costs [4]. Nevertheless, reliance on hardware sensors remains a barrier to the mass adoption of assistive mobility devices.

This work was supported by the Italian Ministry of University and Research, under the complementary actions to the NRRP "Fit4MedRob - Fit for Medical Robotics" Grant (#PNC0000007).

(Corresponding author: Muhammad Ishaq)

The authors are with the Department of Electrical Electronic and Computer Engineering, University of Catania, Catania, Italy.
(e-mail: muhammad.ishaq@unict.it, francesco.cancelliere@phd.unict.it, giuseppe.sutera@unict.it, dario.guastella@unict.it, giovanni.muscato@unict.it)

Instead, sensorless force estimation (SFE) offers an alternative control strategy without direct sensors within hardware part. This method uses an algorithm called a disturbance observer (DOB) to calculate external forces based on internal motor data, such as current and velocity [5]. The DOB treats human user forces and environmental factors as disturbances to the nominal motor dynamics. This approach simplifies hardware design in assistive robotics [6].

Traditional DOBs rely on strict mathematical models. These models require exact physical parameters, including mass and friction coefficients. In physical applications, system parameters change. The SWs encounters varied user loads and different floor conditions. Consequently, when the physical system differs from the mathematical model, the traditional DOB produces estimation errors. Furthermore, the mathematical equations require the calculation of velocity derivatives. Sensor noise in the motor encoders causes these derivatives to amplify errors. This amplification leads to unstable control inputs.

This paper presents a data-driven disturbance observer for a smart walker. Past research validated a mathematical disturbance observer on a physical differential-drive walker platform [7]. That physical implementation proved the viability of sensorless control but demonstrated the sensitivity of the mathematical model to parameter mismatches and sensor noise. To resolve these hardware limitations, the proposed system replaces the mathematical equations with a long short-term memory (LSTM) neural network (NN). The network maps noisy motor signals to estimated human interaction forces. This approach removes the need to calculate velocity derivatives and adapts to unmodeled mass variations. This study evaluates the neural network using a simulation environment configured with the exact physical parameters of the existing hardware prototype [7].

The study validates the AI DOB through MATLAB simulations. The simulations compare the NN against a traditional mathematical DOB. The evaluation includes scenarios with sensor noise and a mass parameter mismatch. Additional tests examine the system response during straight motion, inclined paths, turning maneuvers, safety braking, and dynamic force tracking. The results verify the capability of the NN to track user intent and reject environmental loads under imperfect physical conditions.

A. Mathematical Notations

The mathematical notations used in this paper are defined as follows. The parameter m is the total mass of the smart walker. The parameter r is the wheel radius. The parameter J is the motor inertia. The parameter F_r is the rolling friction coefficient. The parameter θ is the incline angle. The parameter g is gravitational acceleration. The variable F_{ext} represents the external applied user force. The variable τ_m represents the motor torque. The variable τ_{env} represents the environmental resistance. The variable ω represents the wheel angular velocity.

II. Relevant Literature

A. Smart Walker Technology Gaps

Recent reviews identify a gap between advanced laboratory prototypes and practical commercial walkers [8]. Many current systems use multiple physical sensors to detect human intent and monitor gait. The integration of complex hardware increases device weight and power consumption. Researchers emphasize the need for lightweight and energy-efficient designs to improve user acceptance [8]. Sensorless control strategies support this objective by reducing hardware complexity.

B. Sensor-Based Intent Recognition

SWs need user intent data to govern their motors. The authors in [2] use a six-axis force and torque sensor on the SW handle to detect user inputs for shared control navigation. In [3], the researchers used multi-axis load cells to feed input data to an admittance controller. However, these hardware components add to the cost and create points of mechanical failure. Consequently, researchers explore alternative sensor technologies to reduce these hardware costs. The authors in [4] replaced strain gauges with an optical sensor using photodiodes and light-emitting diodes to measure deformation. Xing et al. [9] deployed a force-based admittance control architecture on a mobile manipulator to assist human mobility. Furthermore, physical sensors introduce a need for ongoing hardware maintenance.

C. Sensorless Force Estimation

SFE calculates external forces using existing motor data. The DOB is a standard algorithm for this task. Authors in [10] describe the mathematics of DOBs in motion control systems. The DOB compares the expected motor behavior against the measured motor behavior to isolate the external disturbance. In [6], researchers used this principle in a rehabilitation exoskeleton to measure human interaction forces without load cells. A limitation of the mathematical DOB is its reliance on precise physical parameters. Past research validated a mathematical DOB on a physical active walker [7]. The physical tests confirmed the observer estimates user push forces without external load cells. However, when the physical mass or friction values do not match the parameters programmed into the controller, the mathematical DOB calculates an incorrect force value.

D. Neural Network for Disturbance Observer

Machine learning models offer solutions to the parameter sensitivity of mathematical models. Researchers integrate neural networks with disturbance observers to learn nonlinear friction patterns in industrial robots [10]. Data-driven admittance controllers adapt to unknown environmental disturbances without exact parameter tuning [11]. Other frameworks use neural networks to separate internal system uncertainties from external human forces [12]. Standard multilayer perceptron models process static data points. However, sensor data in physical robotics involves time-series sequences. LSTMs address temporal dependencies in sensor sequences. Researchers use these temporal networks to predict human motion intent during cooperative physical tasks [13]. These time-series models track human forces under uncertain physical conditions better than static networks. This paper expands on these concepts by deploying a LSTM network as a standalone disturbance observer for an active smart walker.

III. Methodology

Fig. 1 illustrates the closed-loop control architecture of the SW. The system consists of three primary components. The physical plant represents the SW dynamics driven by the user and the environment. The SFE block normalizes the noisy sensor data and processes it through a NN to estimate the applied user force. The motion control block uses an admittance controller to translate this force into a velocity command for the motor control board.

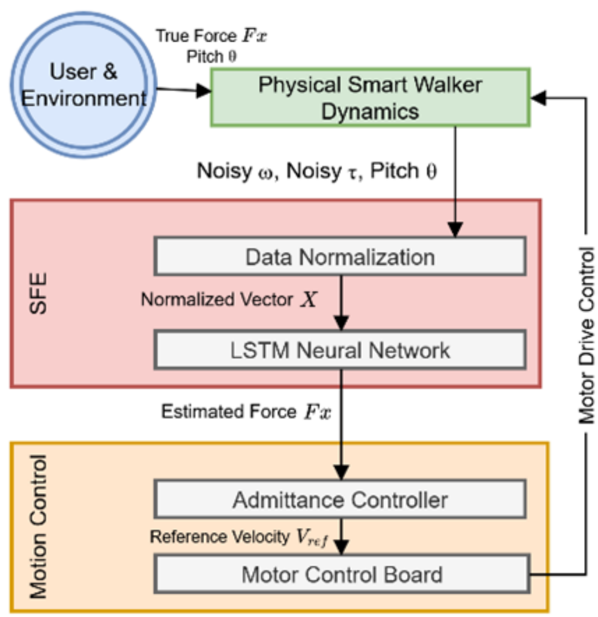

Fig. 1. Control scheme of the AI SFE strategy. The system forms a closed loop where noisy physical states act as inputs for the NN to isolate the user force and drive the admittance controller.

A. Smart Walker Dynamic Model

The physical model represents a differential drive walker [7]. The parameters of the platform already elaborated in section I-A. The environmental resistance as described by (1) includes friction and gravity components.

$$F_{env} = cmg\cos(\theta) + mg\sin(\theta) \tag{1}$$

Moreover, the relationship between the applied force, the motor torque, and the wheel angular velocity ω is defined by (2).

$$\tau = (mr^2 + J)\dot{\omega} + D_m\omega + rF_{env} - rF_x \tag{2}$$

B. Traditional Mathematical DOB

The mathematical DOB calculates the estimated user force $\hat{F}_x$ by rearranging the dynamic model [7]. The DOB uses the measured torque and wheel velocity.

$$\hat{F}_x = \left(mr + \frac{J}{r}\right)\dot{\omega} + \left(\frac{D_m}{r}\right)\omega + F_{env} - \frac{\tau}{r} \tag{3}$$

However, equation (3) requires the time derivative of the angular velocity ω. Physical encoders produce electrical noise. The derivative calculation magnifies this noise. Moreover, a moving average filter mitigates the noise but introduces phase lag. Additionally, the equation (3) relies on a fixed mass parameter m. Consequently, if the physical mass of the SW changes, the DOB calculates an incorrect force and produces a steady-state estimation error.

C. AI-Based DOB

The proposed methodology uses a LSTM NN to replace the mathematical DOB. The network acts as a data-driven DOB. It learns the inverse dynamic mapping from sensor data to the applied user force without calculating velocity derivatives. The input vector X_k at time step k contains five variables as expressed by (4). These variables are the left wheel velocity ω_l, the right wheel velocity ω_r, the left motor torque τ_l, the right motor torque τ_r, and the pitch angle θ.

$$X_k = [\omega_{l,k}, \omega_{r,k}, \tau_{l,k}, \tau_{r,k}, \theta_k]^T \tag{4}$$

Furthermore, the target output vector Y_k contains the true left and right user forces as evident by (5).

$$Y_k = [F_{x,l,k}, F_{x,r,k}]^T \tag{5}$$

The system standardizes the input data using the mean μ_X and standard deviation σ_X of the training dataset. This normalization process centers the data to improve network convergence as given in (6).

$$X_{\text{norm},k} = \frac{X_k - \mu_X}{\sigma_X} \tag{6}$$

The network architecture comprises a sequence input layer, a LSTM layer with 128 hidden units, a fully connected layer, and a regression output layer. Fig. 2 illustrates the data flow and gating mechanisms within the neural network architecture. The LSTM cells process the sequential data over time. Each cell updates its cell state C_k and hidden state h_k using a set of gating functions. The forget gate f_k, input gate i_k, and output gate o_k regulate the flow of information. The weight matrices W and bias vectors b dictate the gate activations as defined by (7) through (12).

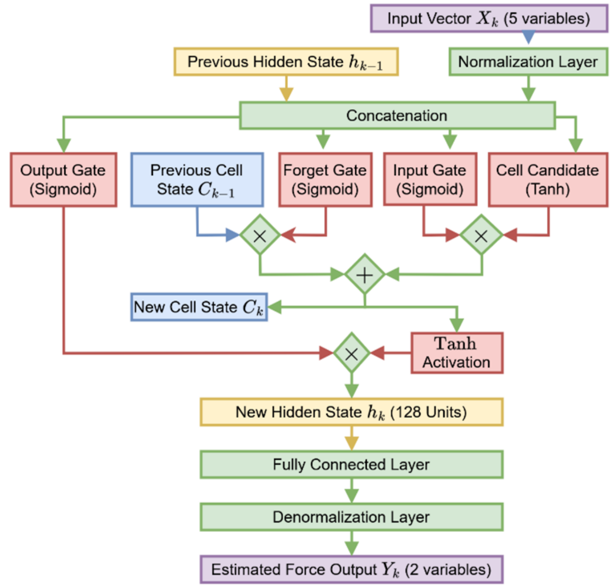

Fig. 2. Architecture of the LSTM NN. The system normalizes a five-variable input sequence and processes the data through 128 hidden units. The internal gating mechanisms map the nonlinear sensor relationships to produce a two-variable estimated force output.

$$f_k = \sigma\left(W_f[h_{k-1}, X_{\text{norm},k}] + b_f\right) \tag{7}$$

$$i_k = \sigma\left(W_i[h_{k-1}, X_{\text{norm},k}] + b_i\right) \tag{8}$$

$$\tilde{C}_k = \tanh\left(W_C[h_{k-1}, X_{\text{norm},k}] + b_C\right) \tag{9}$$

$$C_k = f_k C_{k-1} + i_k \tilde{C}_k \tag{10}$$

$$o_k = \sigma\left(W_o[h_{k-1}, X_{\text{norm},k}] + b_o\right) \tag{11}$$

$$h_k = o_k \tanh(C_k) \tag{12}$$

The fully connected layer maps the final hidden state h_k to the normalized force estimates $Y_{(norm,k)}$. The system denormalizes this output using the target mean μ_Y and standard deviation σ_Y to yield the final estimated forces as given in (13).

$$\hat{Y}_k = Y_{\text{norm},k}\,\sigma_Y + \mu_Y \tag{13}$$

A MATLAB code generates 200 simulation episodes of dynamic interactions to build the training dataset. It samples user forces from a uniform distribution between $-15N$ and $15N$ to represent typical human pushing limits. Surface inclines are sampled from a uniform distribution between $-10°$ and $10°$. The episodes include varied push durations and overlapping slope transitions to capture transient dynamics. The network architecture bypasses the mathematical velocity derivative calculation. The internal cell states map the sequential changes in velocity and torque to the external force. This temporal processing filters the electrical noise present in the encoder signals. The network trains for 200 epochs using the Adam optimization algorithm.

D. Admittance Control

The control system uses an admittance model to translate the estimated force into a reference velocity V_{ref} for the motors. The continuous-time transfer function uses a virtual mass M_v and virtual damping D_v as shown in (14).

$$G(s) = \frac{V_{ref}(s)}{\hat{F}_x(s)} = \frac{1}{M_v s + D_v} \tag{14}$$

The controller implements this continuous-time transfer function using Tustin's discretization method. The control loop operates with a sampling period T_s of $0.04s$ to match the hardware specifications of the physical prototype [7]. The discrete update shown in (15) calculates the current velocity reference $\omega_{ref}(k)$ based on the past velocity and the estimated forces.

$$V_{ref,k} = -P_1 V_{ref,k-1} + K_{adm}\left(\hat{F}_{x,k} + \hat{F}_{x,k-1}\right) \tag{15}$$

The constants K_{adm} and P_1 depend on the admittance parameters and sample time as detailed in (16) and (17) respectively.

$$K_{adm} = \frac{\Delta t}{2M_v + D_v \Delta t} \tag{16}$$

$$P_1 = \frac{-2M_v + D_v \Delta t}{2M_v + D_v \Delta t} \tag{17}$$

The controller divides the reference velocity by the wheel radius to command the left and right motor angular velocities.

IV. Numerical Results

A. Passive Observer Evaluation

Fig. 3 compares the traditional mathematical DOB and the AI DOB in a passive monitoring state. Both DOBs read sensor data while the physical SW receives a true user force. The physical SW mass is set to $20kg$. The control system assumes a $14kg$ mass. The simulation adds Gaussian noise to the velocity and torque signals. A constant $8N$ force is applied from $3-9s$ on a $5°$ slope. The mathematical DOB produces an oscillating force estimate due to the sensor noise. The mathematical DOB generates a steady-state error because of the $6kg$ mass difference. The AI DOB filters the noise and tracks the $8N$ force without steady-state error. The data-driven model ignores the predefined mass parameter and maps the sensor relationships to the true force.

B. Straight Level Motion

Fig. 4 illustrates the system response of the integrated AI DOB and admittance controller on a flat surface. The admittance parameters are set to a virtual mass of $1kg$ and a virtual damping of $10.0Ns/m$. The input is a $10N$ step force applied to both handles from $3-9s$. Fig. 4 (a) and (b) show the AI DOB estimating the left and right $10N$ forces after a short transient rise time. The admittance controller generates a reference velocity based on this estimate. Fig. 4 (c) shows the left and right wheels accelerating to equal angular velocities and maintaining a constant speed.

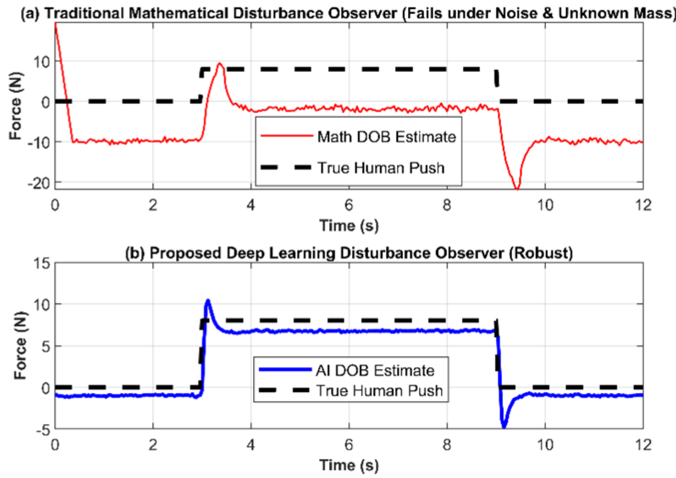

Fig. 3. Passive observer evaluation under parameter mismatch. (a) The mathematical DOB exhibits oscillations and steady-state error. (b) The AI DOB tracks the $8N$ user force despite the sensor noise and the $6kg$ mass

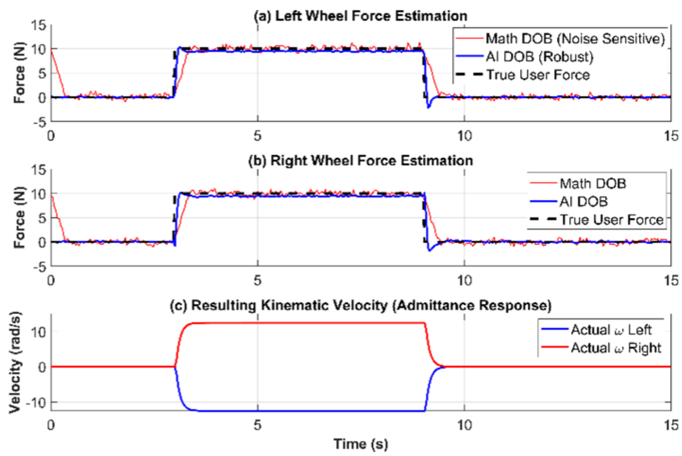

Fig. 4. System response during straight level motion. (a) Left wheel force estimation. (b) Right wheel force estimation. (c) The admittance controller generates equal reference velocities for both wheels to maintain a straight trajectory.

C. Inclined Path Motion

Fig. 5 evaluates the system response to environmental changes. A constant $10N$ user force is applied from $2-12s$. The SW moves onto a $5°$ incline between $5-10s$. The gravitational force pulls the SW backward on the slope. Fig. 5 (a) and (b) display the estimated forces for both wheels. A transient spike occurs at the $5s$ mark as the algorithm detects the new disturbance. The controller increases motor torque to overcome gravity. Fig. 5 (c) demonstrates the wheel velocities recovering after minor transient dips. The DOB separates the continuous $10N$ user push from the physical gravitational pull.

D. Turning Maneuver

Fig. 6 shows the system behavior during a directional turn. From $3-10s$, the left handle receives a $2N$ force and the right handle receives a $10N$ force. Fig. 6 (a) and (b) show the NN estimating the left and right forces as distinct values. The admittance controller processes these separate force inputs to generate two distinct velocity commands. Fig. 6(c) displays the right wheel accelerating to a higher angular velocity than

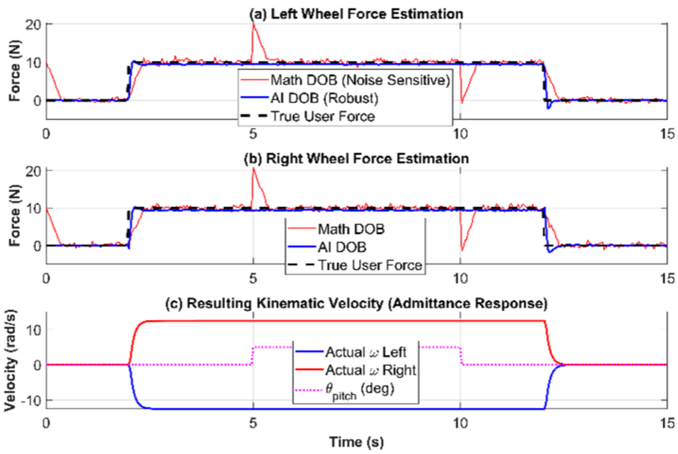

Fig. 5. System response on a 5° inclined path. (a) Left wheel force estimation tracks the gravitational disturbance. (b) Right wheel force estimation matches the left wheel. (c) Wheel velocities show transient dips before the controller compensates for the gravitational pull.

the left wheel. This speed differential controls the turning motion of the walker.

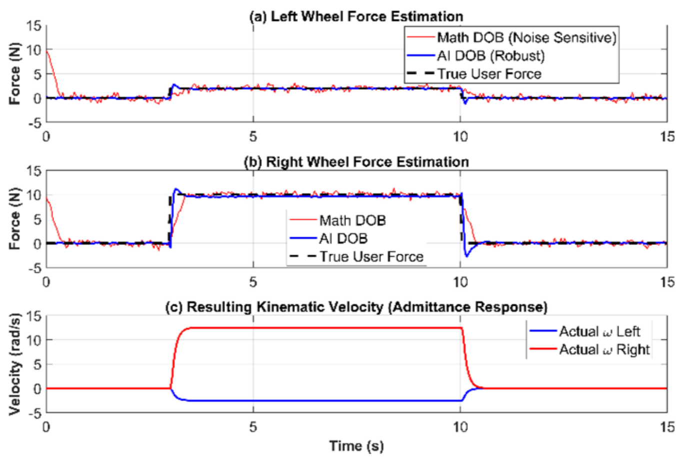

Fig. 6. System response during a turning maneuver. (a) Left wheel force estimation tracks a $2N$ input. (b) Right wheel force estimation tracks a $10N$ input. (c) The right wheel accelerates faster than the left wheel to execute the turn.

E. Safety Braking

Fig. 7 validates the automatic braking function of the admittance control loop. The input is a $10N$ force from $2-4s$. A $5°$ incline begins at $3s$. The user force drops to $0N$ at $4s$. The SW rests on an incline with zero user input. The admittance target velocity drops to zero. Fig. 7 (a) and (b) show the AI DOB identifying the backward gravitational pull on both wheels. The system calculates the required opposing motor torque to maintain the zero-velocity target. Fig. 7 (c) shows the wheel speeds dropping to zero. The SW holds its stationary position on the incline without mechanical brakes.

F. Dynamic Force Tracking with Mass Mismatch

Fig. 8 tests the AI DOB against time-varying inputs under the parameter mismatch. The user input is a sinusoidal force with a $0.25HZ$ frequency, a $4N$ amplitude, and a $6N$ offset. The force oscillates between $2N$ and $10N$. Fig. 8(a) and (b) show the AI DOB tracking the sinusoidal input curve on both

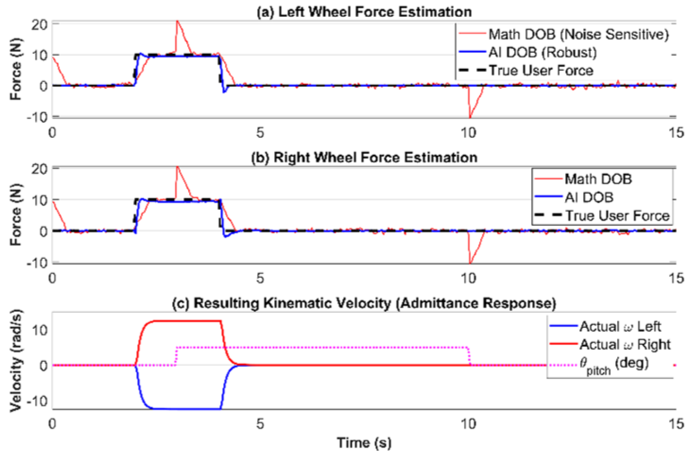

Fig. 7. Validation of the automatic braking function on an incline. (a) Left wheel force estimation registers the gravitational pull after user force drops to $0N$. (b) Right wheel force estimation mirrors the left side. (c) Wheel velocities reach 0 rad/s as the controller outputs opposing torque to hold the walker stationary.

wheels. Fig. 8 (c) shows the SW wheel velocity following the oscillating force profile. The $0.25HZ$ evaluation frequency achieves the standard human walking pace during continuous motion. Higher frequency components represent sudden changes in user intent. Testing the bandwidth limitations of the network response to abrupt force changes requires future validation on physical hardware.

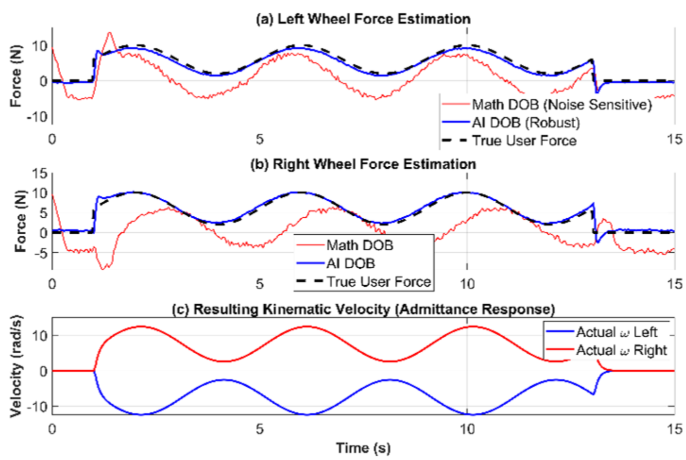

Fig. 8. Dynamic force tracking with a $6kg$ mass mismatch. (a) Left wheel force estimation tracks the sinusoidal input. (b) Right wheel force estimation follows the varying force curve. (c) The resulting wheel velocities oscillate in response to the dynamic user force.

V. DISCUSSION

The simulation results show the differences between the LSTM NN and the traditional mathematical DOB. The mathematical DOB relies on programmed physical parameters. When the physical mass of the SW changes to $20kg$, the mathematical model using a $14kg$ parameter generates an estimation error. Furthermore, the mathematical model requires velocity derivatives. Sensor noise in the motor encoders affects these derivatives and produces oscillating force estimates. Instead, the AI DOB maps sensor data to force values. This data-driven approach removes the derivative calculations and mitigates the sensor noise. Moreover, the NN tracks the user force despite the $6kg$ mass mismatch. This shows robustness to physical parameter changes, reflecting findings by [7], [10] regarding NN managing uncertainties in robot models.

The integrated control system processes mobility scenarios. For example, on flat surfaces and during turns, the NN separates the left and right forces. The admittance controller uses these inputs to drive the wheels at different speeds. On inclined surfaces, the system detects gravitational pull. Additionally, the automatic braking function operates without mechanical brakes. When the user releases the handles on a slope, the target velocity becomes zero. The DOB registers the gravitational force. Consequently, the controller commands the motors to output opposing torque. This action holds the SW in a stationary position.

This study relies on computer simulations to isolate the neural network response. The two-dimensional dynamic model limits the analysis to longitudinal motion. A physical smart walker experiences three-dimensional forces including lateral frame torsion. However, the simulation environment uses the physical parameters of the existing hardware prototype detailed in [8]. This prototype uses a Raspberry Pi 4 processor and a dual-channel motor driver. Past physical testing [14] validated the admittance control scheme but showed the limitations of the mathematical observer under noisy sensor conditions. The LSTM network resolves these limitations. The next phase of this research involves deploying the trained network onto the hardware processor. Physical validation will test the network against actual stiction, unmodeled nonlinear friction, and unseen user inputs. Safety metrics are critical for assistive devices [8]. The automatic braking function depends on estimation accuracy. If the observer fails to detect user force, the target velocity commands a stop. Based on the programmed admittance damping parameter, the theoretical braking distance on a 5°slope is 0.15 meters. Future physical trials will verify this safety metric.

VI. Conclusion

This study evaluates an SFE strategy for an active SW. The approach replaces traditional physical force sensors with a LSTM NN. The NN functions as a data-driven DOB. Simulations compare the AI DOB against a traditional mathematical DOB. The mathematical DOB requires velocity derivatives and produces oscillating force estimates under sensor noise. It also generates steady-state errors when the physical mass of the SW does not match the programmed mass parameter. Instead, the NN maps noisy motor signals to the true user force without derivative calculations.

The integrated system combines the AI DOB with an admittance controller. The simulations test the system under various mobility scenarios. The system tracks step force inputs on level surfaces and tracks decoupled forces to execute turning maneuvers. On inclined paths, the DOB separates the user push from the gravitational pull. The control loop produces an automatic braking function that holds the SW stationary on a slope when user input reaches zero. The NN also tracks time-varying sinusoidal forces under conditions with a 6-kg mass mismatch. The simulation parameters mirror an existing physical prototype [7]. Future work will implement this trained network on the physical processor to evaluate its performance against nonlinear mechanical friction and unpredictable user interactions.

References

[1] W. H. Organization, "Decade of healthy ageing (2021–2030)," tech. rep., Geneva, Switzerland: WHO Press, 2021.

[2] G. Sutera, D. C. Guastella, F. Cancelliere, and G. Muscato, "Adaptive navigation of a smart walker with shared control," *Actuators*, vol. 14, no. 5, p. 224, 2025.

[3] M. F. Jiménez, M. Monllor, A. Frizera, T. Bastos, F. Roberti, and R. Carelli, "Admittance controller with spatial modulation for assisted locomotion using a smart walker," *Journal of Intelligent & Robotic Systems*, vol. 94, no. 3, pp. 621–637, 2019.

[4] M. E. V. Segatto, A. F. Neto, C. A. Cifuentes, C. A. Diaz, *et al.*, "Development of an optical technology-based sensor and deep learning models to estimate user-interaction forces in smart walkers," in *Biophotonics in Exercise Science, Sports Medicine, Health Monitoring Technologies, and Wearables VI*, vol. 13313, pp. 43–49, SPIE, 2025.

[5] E. Sariyildiz and K. Ohnishi, "Stability and robustness of disturbance-observer-based motion control systems," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 414–422, 2014.

[6] A. U. Pehlivan, D. P. Losey, and M. K. O'Malley, "Minimal assist-as-needed controller for upper limb robotic rehabilitation," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 113–124, 2015.

[7] M. Ishaq, F. Cancelliere, G. Sutera, D. C. Guastella, and G. Muscato, "A sensorless control strategy for an active assistive walker: Design, implementation, and experimental validation," in *2025 11th International Conference on Control, Decision and Information Technologies (CoDIT)*, vol. 1, pp. 1172–1176, IEEE, 2025.

[8] M. Ishaq, D. C. Guastella, G. Sutera, F. Cancelliere, and G. Muscato, "A quantitative review of smart walkers: Trends, gaps, and a proposed evaluation framework," in *2025 International Conference on Electromechanical and Energy Systems (SIELMEN)*, pp. 377–382, IEEE, 2025.

[9] H. Xing, A. Torabi, L. Ding, H. Gao, Z. Deng, V. K. Mushahwar, and M. Tavakoli, "An admittance-controlled wheeled mobile manipulator for mobility assistance: Human–robot interaction estimation and redundancy resolution for enhanced force exertion ability," *Mechatronics*, vol. 74, p. 102497, 2021.

[10] S. Liu, L. Wang, and X. V. Wang, "Sensorless force estimation for industrial robots using disturbance observer and neural learning of friction approximation," *Robotics and Computer-Integrated Manufacturing*, vol. 71, p. 102168, 2021.

[11] G. Peng, C. P. Chen, W. He, and C. Yang, "Neural-learning-based force sensorless admittance control for robots with input deadzone," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 6, pp. 5184–5196, 2020.

[12] T. Li, A. Badre, H. D. Taghirad, and M. Tavakoli, "Neural network learning of robot dynamic uncertainties and observer-based external disturbance estimation for impedance control," in *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 591–597, IEEE, 2023.

[13] P. Franceschi, D. Cassinelli, N. Pedrocchi, M. Beschi, and P. Rocco, "Design of an assistive controller for physical human–robot interaction based on cooperative game theory and human intention estimation," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 5741–5756, 2024.