

Bridging Discrete and Continuous Interfaces to Generate Adaptive Gait Synthesis for Humanoid Robots

Jude Onyenze

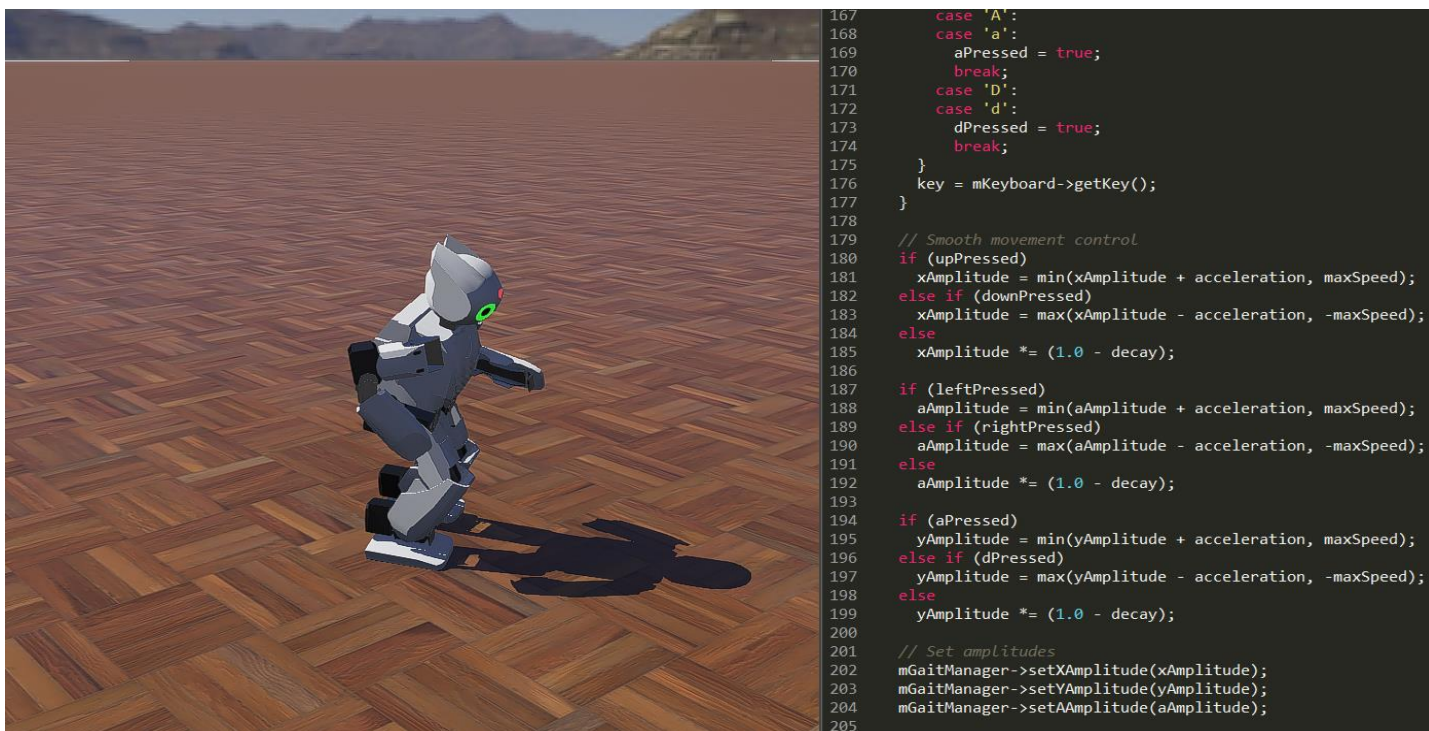


Figure 1: The Darwin-OP humanoid robot within the Webots simulation environment. This figure shows the full-body model of the Darwin OP, a 20-degree-of-freedom bipedal humanoid robot equipped with virtual sensors and actuators. The simulation replicates key physical properties and motion dynamics, including joint torque, balance control, and foot-ground interaction. The robot stands in a ready posture within a flat virtual environment, where dynamic walking behaviors can be tested and visualized in real-time. Webots provides integrated tools for real-time control, sensor feedback visualization, and gait analysis, making it suitable for developing and evaluating locomotion controllers.

Abstract— As humanoid robots become increasingly prevalent across the world—whether through widespread deployment of

standardized robotic platforms or through the impressive advancements made by startups and research groups like Boston

Dynamics—there is a renewed interest in the adaptability and control of bipedal locomotion. While many modern systems leverage reinforcement learning and advanced AI pipelines, this work revisits a legacy humanoid platform, the Darwin OP, to explore novel methods of gait control using traditional input systems. We present a modular, real-time walking controller that enables dynamic, omni-directional locomotion via both discrete (keyboard) and continuous (joystick) interfaces.

Keywords— Humanoid, Teleoperation, Dynamic Walking, Robotics, Darwin-Op

1. INTRODUCTION

Humanoid robots have seen remarkable growth in both research and emerging commercial applications. While they have traditionally remained within the realms of academic labs and industrial R&D, there is a growing movement—spearheaded by startups and major robotics companies—to bring humanoid robots into real-world environments, including domestic settings. This vision, reminiscent of depictions in science fiction such as the film *I, Robot*, aims to create autonomous robotic assistants capable of supporting human life through versatile, lifelike behaviours[1][2].

However, many modern approaches to humanoid locomotion rely heavily on advanced artificial intelligence techniques such as deep reinforcement learning, imitation learning, and complex control paradigms like model predictive control (MPC) or real-time feedback optimization [8], [9]. These methods often demand substantial computational resources, high-end sensors, and well-calibrated physical platforms, which limit accessibility and generalizability [3].

In contrast, the work presented here proposes a lightweight, accessible alternative. Using the older DARwIn-OP humanoid robot platform—simulated in the Webots environment [4]—we develop a novel dynamic walking controller that enables expressive, adaptive, and omnidirectional locomotion without requiring deep learning models, expensive hardware, or high computational overhead. This controller accepts either binary (keyboard) or analog (joystick) user inputs to generate smooth, momentum-aware gait transitions. By blending user intent with low-level gait shaping, the system achieves stability-aware omni-directional movement, something rarely seen in legacy humanoid platforms [5], [6].

The novelty of this work lies in its minimalist yet expressive design: the system dynamically adapts to varying levels of user input, generating alternating gait cycles that resemble real-world locomotion. Unlike the standard DARwIn-OP walking modules [4], which use fixed gait parameters and exhibit slow, statically stable movement, our controller enables dynamic transitions and omni-directional locomotion that respond continuously to user intent. Compared to the robust but largely state-driven gait systems used by Nimbro in RoboCup [5], [6], which prioritize reliability over adaptability, our approach achieves smoother transitions and more lifelike motion blending using real-time amplitude modulation. Furthermore, it challenges the assumption that only high-end platforms with advanced actuators—like those from Unitree or Boston Dynamics [1], [3]—can perform lifelike walking. Our findings show that even older, affordable humanoid robots can demonstrate natural dynamic behaviors when driven by thoughtfully designed control architectures.

2. RELATED WORK

Enabling dynamic walking in humanoid robots has long been one of the central challenges in robotics. Over the past few decades, a wide range of approaches have been proposed to address this, ranging from classical control strategies to modern learning-based techniques. Early solutions often relied on **trajectory optimization** or **model predictive control (MPC)**, both of which require detailed models of the robot's kinematics and dynamics, as well as precisely tuned sensor feedback systems. These methods often depend on tightly calibrated environments and can be computationally demanding, making them more suited for controlled laboratory conditions than real-world deployment. In more recent years, particularly within the last two to three years, there has been a surge in the use of deep reinforcement learning (DRL) and imitation learning to train policies for complex bipedal locomotion. These approaches have enabled robots like Boston Dynamics' Atlas [1], Agility Robotics' Digit [2], and various models from Unitree Robotics [3] to perform robust, agile, and visually impressive walking behaviors. However, these systems typically rely on

high-performance onboard or offboard computation, expensive sensors, and long training periods involving large-scale data collection [9]. As a result, they are inaccessible to most research institutions and universities, limiting reproducibility and practical experimentation.

Furthermore, while many humanoid robots can be teleoperated or guided using simple user interfaces, their walking controllers are often built on predefined state machines or fixed gait sequences with limited adaptability. These controllers may include basic user input mechanisms but typically lack smooth transitions or responsiveness to dynamic user intent. In some cases, input recognition is limited to symbolic gesture detection or command mapping, which fails to reflect the nuanced motion control required for expressive locomotion [6], [7].

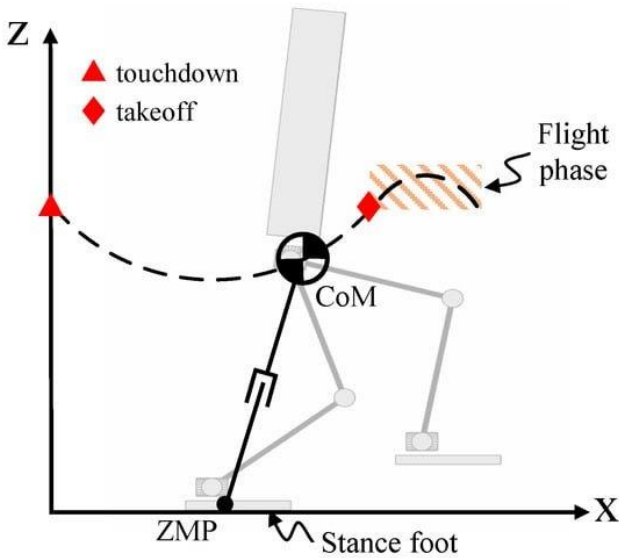


Figure 2: The zero-moment point (ZMP) follows a planned trajectory within the support polygon to ensure dynamic stability as the robot transitions through single and double support phases during walking.

This work positions itself as a lightweight, accessible alternative. It bridges the gap between simplistic finite-state gait controllers—such as those used in DARwIn-OP’s default Webots simulation [4]—and complex machine-learned policies deployed on robots like Atlas and Digit [1], [2]. By

enabling adaptive omni-directional dynamic walking through real-time binary (keyboard) or continuous (joystick) inputs, the controller offers greater expressiveness than static gait libraries [5], [6] and greater accessibility than reinforcement learning-based systems [8], [9]. Crucially, this approach does not rely on deep learning, external feedback loops, or high-cost actuators. Instead, it demonstrates that effective, naturalistic walking behaviors can be achieved on legacy platforms using a thoughtfully designed and smoothly modulated control architecture.

3. System Architecture

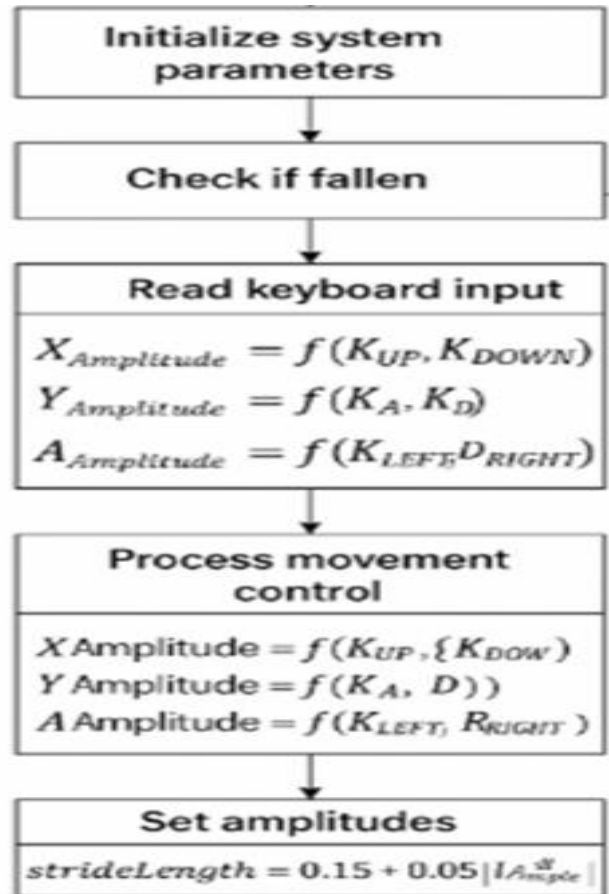


Figure 3: The system begins by initializing all relevant parameters, including amplitude variables and sensor modules. A fall-check is performed to trigger recovery behavior’s if needed.

A. Simulation Environment:

The Darwin OP model in Webots features 20 degrees of freedom, each represented by individually controlled servo motors. These are accompanied by a suite of virtual sensors, including accelerometers, gyroscopes, foot-ground contact sensors, and joint position sensors. Webots simulates accurate foot-ground interaction and supports real-time visualization of motion dynamics, making it ideal for testing balance, trajectory, and walking behavior under different control inputs.

B. Controller Design:

The architecture of the proposed controller is modular, lightweight, and highly responsive. It is designed to operate in real-time, utilizing either **discrete binary inputs** (via keyboard) or **continuous analog inputs** (via joystick) to drive the walking behavior. Both interfaces share a common low-level control structure, ensuring consistent behavior regardless of the input modality. Input data is processed at each simulation time step, triggering appropriate gait modulation functions based on the user's commands.

C. Hardware Abstraction and Robot Interface:

The Webots environment provides an abstracted hardware layer where motor control and sensor feedback are accessed via prebuilt API calls. The Darwin OP's motors are interfaced using **Motor** and **PositionSensor** objects, while inertial data (from virtual IMUs) is read using **Accelerometer** and **Gyro** modules. This abstraction enables seamless controller deployment and simplifies integration between control logic and hardware behavior.

D. User Input Handling:

The controller supports both binary keyboard input and analog joystick input through dedicated interface modules. In both modes, a deadzone filter is applied to prevent unintentional movements caused by noise or small signal fluctuations. Input values are mapped to three primary gait amplitudes:

A_x : forward/backward amplitude

A_y : lateral (side-stepping) amplitude

A_θ : rotational (yaw) amplitude

These amplitudes are updated in real time using ramp functions:

$$A_{t+1} = A_t + \Delta a - \delta A_t$$

Where Δa is the user-driven acceleration component and δ represents natural decay when no input is active.

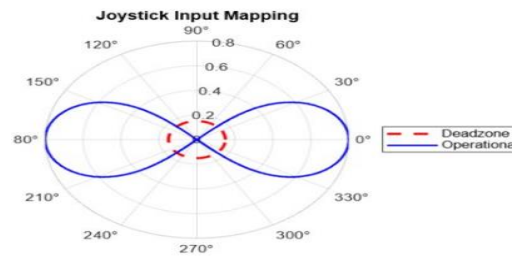


Figure 4: This polar plot illustrates how the controller interprets user joystick input. The **red dashed circle** represents the deadzone threshold, a region near the centre where small inputs are ignored to prevent unintentional drift due to hardware noise or idle hand motion. The **blue operational curve** outlines the range of joystick input values that are mapped to gait amplitude commands. Once the input exceeds the dead zone, amplitude scaling begins using a nonlinear function to ensure smooth, controllable motion transitions. This mapping supports precise low-speed movement and full-speed dynamic walking as input pressure increases.

E. Gait Modulation and Execution:

At each simulation step, the gait amplitudes are passed to the Webots-integrated **GaitManager**, which translates them into lower-level motor commands. This modulation process occurs continuously, allowing the robot to smoothly accelerate and decelerate its walking motions in any

direction. The use of ramp-based amplitude updates enables more lifelike motion transitions, avoiding the abrupt start/stop behavior typical in traditional gait state machines.

4. EXPERIMENTS AND RESULTS

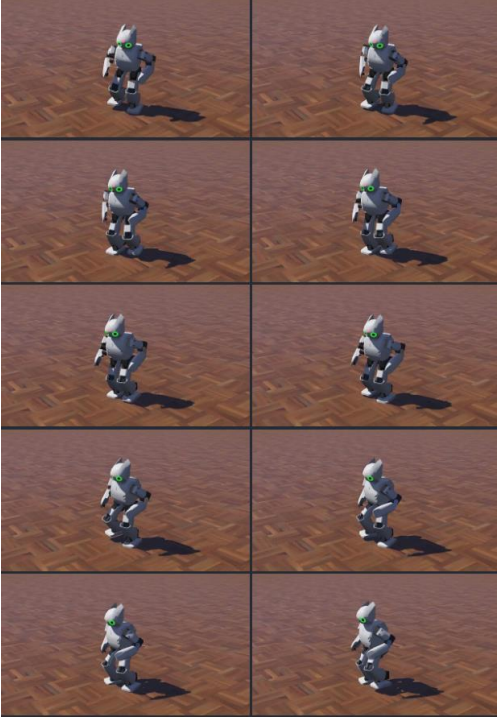


Figure 5: Walking sequence of the Darwin OP robot over a 20-second interval under user input control. This figure shows a time-lapse series of the Darwin OP executing dynamic walking behavior in the Webots simulator. The sequence illustrates the robot's ability to smoothly transition through a full walking gait cycle.

To evaluate the effectiveness and responsiveness of the proposed walking controller, a series of exploratory walking trials were conducted in the Webots simulation environment using the Darwin-OP humanoid robot model. The controller was implemented using C++ within Webots' native control interface, utilizing the built-in Darwin OP model provided by the simulator. The controller accessed the robot's motors and sensors through Webots' API and updated gait parameters at each

simulation step based on real-time user input. The walking behaviour was modulated through amplitude adjustments passed to the Gait Manager, with different input sources—keyboard or joystick—processed through the same motion control logic.

The experiments involved manually teleoperating the robot using both the keyboard-based controller and the joystick-based controller, observing its gait transitions, stability, and ability to respond to directional inputs over time. The robot was walked in multiple directions — forward, backward, laterally, and rotationally — with emphasis on transitions between these motions. In the keyboard mode, discrete keys were held and released to observe the smooth ramp-up and decay of walking amplitude. In joystick mode, analog stick movements were varied in magnitude and direction to examine continuous gait modulation and the impact of the dead zone filter.

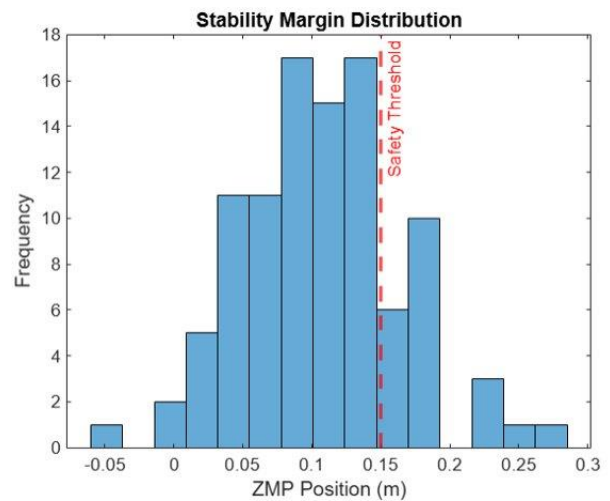


Figure 6: The controllers developed in this project influence the robot's dynamic stability margins during locomotion. Unlike static walking, where the center of mass (CoM) stays within the support polygon, dynamic walking introduces shifts in stability based on motion amplitude and velocity. As gait amplitudes increase—especially with joystick inputs—the center of pressure (CoP) moves more aggressively but remains controlled. The system maintains stability through smooth acceleration and deceleration, without relying on complex balance algorithms. This results in robust, stable locomotion

driven by natural momentum and directional gait shaping.

The robot demonstrated consistent dynamic behavior across all motion axes, with no sudden jolts or unstable transitions. It maintained balance without explicit feedback-based corrections, indicating that the controller’s amplitude ramping and natural deceleration contributed to a stable walking cycle. The omni-directional movement capabilities were fully operational, enabling fluid blending between walking directions in real time.

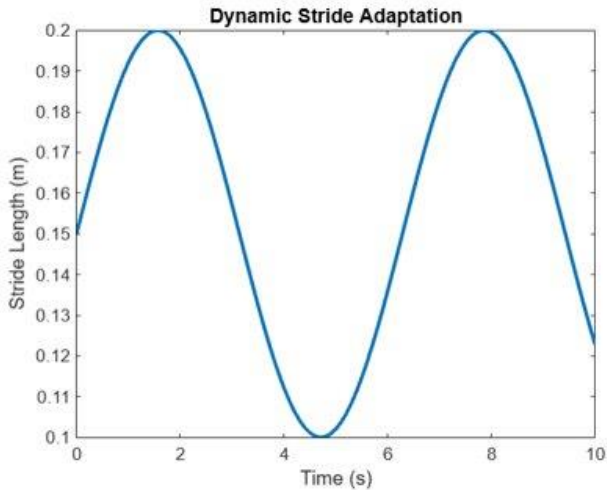


Figure 7: *In this figure it showcases the novelty of these controllers lies in transforming simple user inputs into smooth, adaptive dynamic walking without relying on machine learning or heavy computation. They enable realistic momentum and continuous gait modulation—features rarely seen in small humanoid teleoperation*

No reinforcement learning models, external feedback loops, or dynamic rebalancing algorithms were used, yet the robot was able to produce lifelike and expressive walking motions throughout the trials. These results confirm that the proposed architecture, even when deployed on a legacy platform like Darwin OP, can achieve reliable and adaptive locomotion using simple user-driven input mechanisms.

5. Conclusion

This work presents a lightweight, modular, and expressive dynamic walking controller for the DARwIn-OP humanoid robot, implemented

within the Webots simulation environment [4]. By leveraging both discrete (keyboard) and continuous (joystick) input modes, the controller enables smooth, omni-directional locomotion without reliance on complex machine learning models [9], real-time feedback controllers, or expensive hardware [1], [2]. Through the use of amplitude ramping, deadzone filtering, and directional gait modulation, the system produces lifelike and adaptive walking patterns that remain stable across a variety of user-driven inputs.

The key novelty of this controller lies in its ability to translate user intent—whether binary or analog—into continuous, momentum-aware, omni-directional walking behavior. Unlike traditional controllers based on finite-state transitions or fixed trajectories [5], [6], this system enables seamless motion blending across multiple axes (forward, lateral, rotational), resulting in fluid and human-like transitions. It supports dynamic response to varying input magnitudes, allowing for expressive gait shaping that reflects walking speed, direction, and intent—all without requiring model-based optimization or reinforcement learning policies [8].

This work also uniquely demonstrates that expressive and dynamic locomotion can be achieved even on legacy humanoid platforms, such as DARwIn-OP with Dynamixel actuators [4], using minimal computational resources. The controller extends the expressive potential of these systems beyond their original, limited gait frameworks and toward more adaptable motion control.

Future work will explore extending this controller to handle terrain adaptation, multi-step planning, and integration with lightweight learning mechanisms for semi-autonomous gait refinement. Real-hardware deployment will also be a critical step in validating this approach and potentially expanding its application beyond simulation, toward real-world human-robot interaction scenarios.

REFERENCES

- [1] BOSTON DYNAMICS. *ATLAS: THE WORLD'S MOST DYNAMIC HUMANOID ROBOT*, 2024. [ONLINE].
- [2] AGILITY ROBOTICS. *DIGIT: A HUMAN-CENTRIC ROBOT FOR LOGISTICS AND BEYOND*, 2024. [ONLINE].
- [3] UNITREE ROBOTICS. *HIGH-PERFORMANCE QUADRUPED AND HUMANOID ROBOTS*, 2024. [ONLINE].
- [4] D. HANSON, Y. YOU, J. PARK, AND M. KIM, "DEVELOPMENT OF THE DARWIN-OP HUMANOID PLATFORM," IN *PROC. IEEE-RAS INTERNATIONAL CONFERENCE ON HUMANOID ROBOTS (HUMANOIDS)*, NASHVILLE, TN, USA, 2010, PP. 1–6.
- [5] M. SCHWARZ, S. SCHÜLLER, AND S. BEHNKE, "NIMBRO-OP2X: ADULT-SIZED OPEN HUMANOID PLATFORM FOR RESEARCH AND EDUCATION," IN *PROC. IEEE-RAS INT. CONF. ON HUMANOID ROBOTS (HUMANOIDS)*, 2019, PP. 1–8.
- [6] S. BEHNKE, "ROBOCUP HUMANOID LEAGUE: HUMANOID SOCCER ROBOTS IN RESEARCH AND EDUCATION," IN *PROC. INT. CONF. ON ADVANCED ROBOTICS (ICAR)*, 2006, PP. 1–6.
- [7] M. MISSURA, T. RÖFER, AND M. BEHNKE, "REAL-TIME GAIT ADAPTATION FOR A HUMANOID SOCCER ROBOT," IN *ROBOCUP 2012: ROBOT SOCCER WORLD CUP XVI*, LECTURE NOTES IN ARTIFICIAL INTELLIGENCE, VOL. 7500, SPRINGER, 2013, PP. 30–41.
- [8] R. TEDRAKE ET AL., "A ROBUST CONTROL ARCHITECTURE FOR REAL-TIME DYNAMIC WALKING ON THE ATLAS HUMANOID ROBOT," IN *PROC. IEEE-RAS INT. CONF. ON HUMANOID ROBOTS (HUMANOIDS)*, 2015, PP. 574–581.
- [9] S. HWANG, J. LEE, AND D. HONG, "WALKING CONTROL OF THE DARWIN-OP HUMANOID ROBOT WITH REINFORCEMENT LEARNING," IN *PROC. IEEE INT. CONF. ON ROBOTICS AND AUTOMATION (ICRA)*, 2011, PP. 2307–2312.
- [10] F. ALLGEUER AND S. BEHNKE, "ROBUST SENSOR FUSION FOR BIPED HUMANOID ROBOT LOCALIZATION," IN *PROC. IEEE/RSJ INT. CONF. ON INTELLIGENT ROBOTS AND SYSTEMS (IROS)*, 2015, PP. 4582–4587.