SpeeLo: Speech Interaction During Locomotion in Resource-Constrained Bipedal Robot

Masato Kobayashi[1,2*]

***Abstract*— In bipedal robots, achieving stable operation while simultaneously executing real-time locomotion control and computationally intensive speech processing is a critical design challenge. This paper proposes *SpeeLo* (Speech Interaction During Locomotion in Resource-Constrained Bipedal Robot), a distributed architecture and corresponding evaluation methodology for integrating speech interaction capabilities into bipedal robots with limited computational resources. In the proposed approach, a lightweight server deployed on the robot handles locomotion control and speech output, while speech processing is offloaded to an external client. Furthermore, dialogue history management enables context-aware response generation. Experiments with a physical robot quantitatively demonstrate that the proposed system has minimal impact on the locomotion control cycle and that incorporating conversational context improves response quality. These results confirm that *SpeeLo* enables safe and effective integration of speech interaction under constrained computational resources. Additional material is available at `https://mertcookimg.github.io/speelo`**

I. Introduction

Speech is one of the most natural modalities for human–robot interaction, allowing users to communicate with robots without specialized interfaces or prior training [1]–[4]. For robots operating in human environments, spoken interaction is valuable not only as a convenient input/output channel but also as a means of improving accessibility, engagement, and interaction continuity. In practical settings, however, robots are often expected to communicate while moving, rather than only when standing still. Therefore, enabling speech interaction during locomotion is an important requirement for mobile robots deployed in real-world environments.

This requirement is especially challenging for bipedal robots. Unlike fixed-base or wheeled robots, bipedal platforms must maintain balance and stable whole-body motion through continuous real-time control. Their locomotion performance depends on timely sensor processing, policy inference, and actuator command updates, all of which must be executed under strict timing constraints. In compact research platforms, onboard computational resources are often limited, and these resources are already heavily utilized by locomotion-related processes. Under such conditions, adding computationally intensive speech functions can easily create resource contention and processing delays, potentially degrading locomotion stability.

At the same time, recent advances in speech and language technologies have made natural spoken interaction increasingly feasible for robotics applications. Large-scale speech recognition models such as Whisper provide robust multilingual recognition performance even under relatively noisy conditions [5]. In parallel, large language models have significantly improved the fluency and flexibility of response generation. These developments make it possible to build robot dialogue systems that are substantially more natural than traditional rule-based pipelines. Nevertheless, directly running these models on resource-constrained onboard computers remains difficult, especially for small bipedal robots where real-time locomotion must take precedence over interaction-related computation.

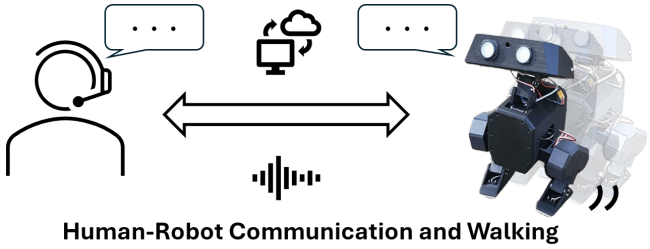

Fig. 1. Concept of SpeeLo. The robot performs real-time locomotion while engaging in speech interaction.

In addition to computational cost, practical spoken interaction requires maintaining conversational context. Human speech frequently includes short utterances, omissions, and elliptical expressions whose interpretation depends on previous turns [6]. If each utterance is processed independently, the robot may generate responses that are grammatically valid but contextually inconsistent. Therefore, a useful spoken interaction system for mobile robots should satisfy two requirements simultaneously: it should avoid interfering with real-time locomotion control, and it should preserve sufficient dialogue history to support context-aware responses.

These considerations reveal a gap between progress in individual component technologies and their integration on real robot platforms. Prior studies have demonstrated effective locomotion control for legged robots and strong performance of speech recognition and language generation models. However, less attention has been paid to how such capabilities can be combined on small bipedal robots with limited onboard resources, while quantitatively verifying that the additional interaction functions do not compromise locomotion control. From both robotics and human–robot interaction perspectives, this integration problem is important because a robot that can only talk when stationary is less

The University of Osaka, [1] D3 Center, The University of Osaka, [2] Graduate School of Maritime Sciences, Kobe University, * corresponding author: kobayashi.masato.cmc@osaka-u.ac.jp

natural and less useful in dynamic real environments.

In this paper, we address this problem using *Open Duck Mini* [7], an open-source reinforcement learning-based bipedal robot, as shown in Fig. 1. We propose *SpeeLo* (Speech Interaction During Locomotion in Resource-Constrained Bipedal Robot), a distributed architecture in which only a lightweight speech server and the locomotion controller run on the robot, while computationally expensive processes—including speech recognition, response generation, and speech synthesis—are offloaded to an external client. In the current implementation, Whisper is used for speech recognition, the Groq API is used for response generation, and edge-tts is used for speech synthesis. In addition, the external client maintains multi-turn dialogue history so that recent conversational context can be incorporated into response generation.

The objective of this study is not only to implement speech interaction on a walking bipedal robot, but also to evaluate whether such integration can be achieved without sacrificing the timing requirements of locomotion control. To this end, we define evaluation criteria based on the locomotion control cycle, dialogue processing latency, and context referencing success. By combining system design with quantitative analysis on a physical platform, this paper aims to provide a practical foundation for integrating natural spoken interaction into resource-constrained bipedal robots.

The main contributions of this paper are as follows:

- design and implementation of a distributed speech interaction architecture for a resource-constrained bipedal robot,
- introduction of client-side multi-turn dialogue history management for context-aware spoken responses, and
- quantitative evaluation on a physical robot in terms of control-cycle stability, dialogue latency, and conversational consistency.

II. Related Work

A. Low-Cost and Resource-Constrained Bipedal Robots

Research on bipedal locomotion has advanced rapidly in recent years, particularly through the application of reinforcement learning to legged robot control [8]–[11]. By training policies in simulation environments such as MuJoCo [12], prior studies have demonstrated robust walking behaviors while reducing the cost and risk of real-world trial-and-error [13]. In addition, sim-to-real pipelines that export trained policies to lightweight runtime formats such as ONNX have made it increasingly feasible to deploy learned controllers on actual robots with limited onboard computing resources [14].

Alongside these algorithmic advances, there has been growing interest in open-source and relatively low-cost bipedal robot platforms [7], [15]–[20]. Such platforms are important because they lower the barrier to entry for physical AI and human–robot interaction research, enabling more groups to test ideas on physical hardware rather than only in simulation. In this context, compact bipedal systems are particularly attractive: they provide a realistic embodied platform, while remaining affordable and reproducible compared with larger humanoid robots. However, their compactness also implies strict constraints on onboard computation, power, and thermal budget. As a result, although recent work has shown that reinforcement learning-based locomotion can run effectively on such platforms, the available computational margin beyond locomotion itself is often small.

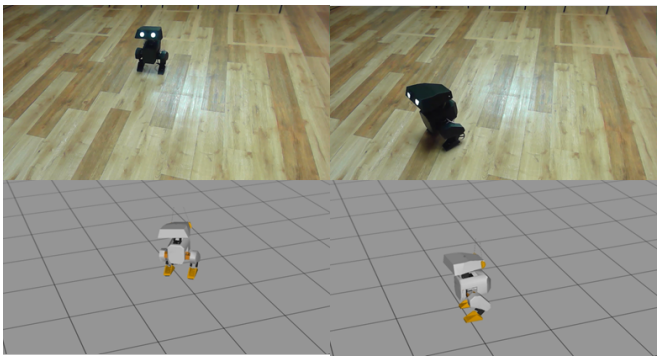

Fig. 2. Simulation and real-world locomotion (Open Duck Mini)

This limitation is especially relevant when additional perception or interaction functions are introduced. In many small bipedal robots, onboard resources are primarily allocated to time-sensitive processes such as sensor acquisition, state estimation, policy inference, and actuator communication. Therefore, extending these platforms with computationally heavy functionalities may lead to resource contention and increased timing jitter. Open Duck Mini [7], shown in Fig. 2, is a representative example of this class of platform: it is an open-source, small-scale bipedal robot that supports reinforcement learning-based locomotion and real-world deployment through a lightweight runtime pipeline. While this design makes it suitable for physical experiments, it also highlights the practical challenge of integrating new capabilities under tight computational constraints.

From this perspective, prior research on low-cost bipedal robots has established a strong foundation for locomotion, sim-to-real transfer, and reproducible open hardware. However, comparatively less attention has been paid to how such robots can support rich spoken interaction during locomotion without compromising the timing requirements of the control loop. This gap motivates the present work, *SpeeLo*, which focuses on integrating speech interaction into resource-constrained bipedal robots through a distributed system design.

B. Dialogue Robots and Spoken Human–Robot Interaction

Spoken interaction has long been recognized as a central modality in human–robot interaction because it allows humans to communicate with robots in an intuitive and socially natural manner [21]–[23]. Dialogue robots typically integrate several modules, including speech recognition, language understanding, response generation, and speech synthesis. Advances in these components have significantly improved the quality of robot dialogue, especially with the emergence

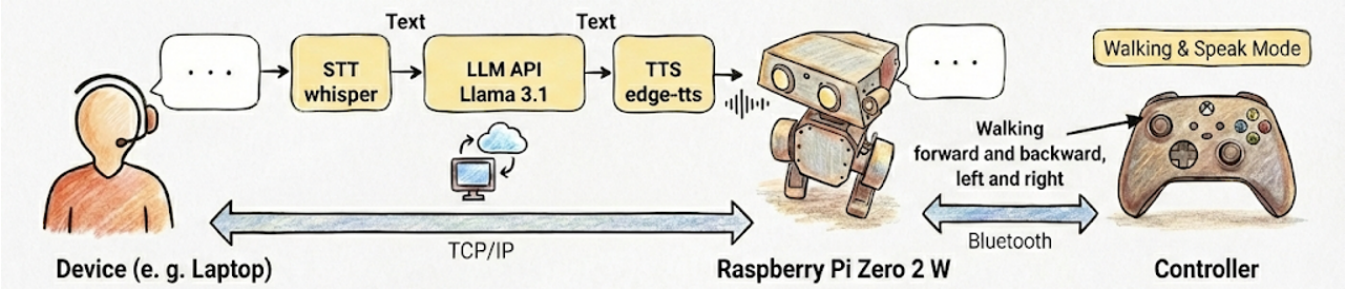

Fig. 3. System overview of SpeeLo. The robot executes locomotion control and audio playback via a lightweight speech server, while speech recognition, response generation, and speech synthesis are offloaded to an external client. Dialogue history is maintained on the client side for context-aware interaction.

of large-scale speech and language models. In particular, Whisper has demonstrated robust multilingual speech recognition across diverse speakers and noisy environments [5], making it attractive for practical robotic systems.

At the same time, recent language models have enabled more flexible and contextually appropriate response generation than earlier pipeline-based dialogue systems [24]–[26]. This progress has expanded the potential of robots as conversational partners in domains such as social robotics, education, and assistance. However, these benefits come with increased computational demands. Running speech recognition and language generation continuously can be expensive even on desktop-class hardware, and this becomes a more serious issue for onboard systems with limited processing power [27]. Consequently, real-world robot dialogue systems must often balance response quality against latency, robustness, and computational feasibility.

Despite substantial progress in spoken dialogue technologies, much of the literature has focused either on improving individual dialogue components or on interaction design in settings where physical motion is not the primary systems constraint. By contrast, for mobile bipedal robots, dialogue must coexist with locomotion control that requires stable periodic execution. This creates a combined systems challenge: speech interaction should remain context-aware and responsive, while the robot must preserve locomotion stability under resource constraints. This challenge is addressed in *SpeeLo*, which integrates speech interaction into a resource-constrained bipedal robot through a distributed system architecture.

C. Position of This Work

This study lies at the intersection of low-cost bipedal robotics and spoken human–robot interaction. While prior work has demonstrated effective locomotion and advanced speech technologies separately, their integration on resource-constrained bipedal platforms remains underexplored. To address this gap, we propose *SpeeLo*, a distributed architecture that offloads computationally intensive speech processing to an external client while preserving real-time locomotion control on the robot.

III. SpeeLo: Speech Interaction During Locomotion in Resource-Constrained Bipedal Robot

A. System Architecture

Our SpeeLo is designed to extend the open-source locomotion control system of Open Duck Mini, as shown in Fig. 3, with the goal of integrating speech interaction capabilities without modifying the existing locomotion behavior. To preserve stable locomotion control, we adopt a distributed architecture in which computationally intensive speech processing is offloaded outside the robot, and only the minimal functionality required for audio output is retained on the robot side.

The system consists of three components: (1) a locomotion control module, (2) a speech server, and (3) a speech client.

The locomotion control module executes walking control while launching the speech server as a background thread within the same process. This design ensures that the main control loop for locomotion does not compete with speech-related processing within a single thread, allowing speech functionality to be integrated without modifying the existing control logic.

The speech server runs on the robot as a background thread. It receives audio data transmitted from the external client over a network and plays it back using an onboard audio output interface. The server is intentionally designed to avoid performing computationally expensive tasks such as speech recognition or response generation. To ensure robustness against communication failures, the server includes exception handling mechanisms that detect timeouts and disconnections, preventing system crashes and maintaining a recoverable state for reconnection.

The speech client runs on an external device (e.g., a PC) and is responsible for speech recognition (Whisper), response generation (Groq API), and speech synthesis (edge-tts). For response generation, the *llama-3.1-8b-instant* model provided by the Groq API is used. The generated audio data is transmitted to the robot, enabling speech interaction while minimizing computational load on the robot. In addition, to support context-aware dialogue, the client maintains a history of multi-turn interactions, allowing past utterances to be referenced during response generation.

TABLE I

PROCESSING FLOW USING DIALOGUE HISTORY

Stage	Description
Initial state	History of the first turn is stored History: [user1: "What is your name?", robot1: "My name is Duck."]
User utterance	"What are you good at?"
LLM input	System prompt + first-turn history + new user utterance
LLM response	"I am good at friendly conversations."
History update	Add second-turn user utterance and response History: [user1, robot1, user2, robot2]

B. Speech Recognition

Speech recognition is performed using Whisper [5], a large-scale model trained on multilingual datasets and capable of handling multiple languages, including Japanese. In the proposed system, speech recognition is executed on the external speech client using the *base* model. The transcribed text is then passed to the subsequent response generation module.

By offloading speech recognition to the external client, the computational load on the robot is reduced, thereby preventing interference with the locomotion control loop and ensuring stable real-time performance.

C. Multi-turn Dialogue History Management

In spoken interaction, short utterances and elliptical expressions frequently occur, making it essential to reference recent conversational context. In this system, dialogue history is maintained as pairs of user utterances and system responses, and this history is appended to the input of the response generation model to enable context-aware dialogue.

Table I illustrates the processing flow and the state of the dialogue history. The history consists of a system prompt, user messages, and robot messages, which are arranged in chronological order with the system prompt placed at the beginning during response generation. The system prompt defines the robot's role (a friendly assistant robot named Duck), response style (concise Japanese responses, typically one to two sentences), and the importance of contextual understanding.

The maximum history length is limited to 10 turns (i.e., user–assistant pairs). When this limit is exceeded, the oldest messages are removed to maintain a constant memory footprint. This design preserves recent context while controlling computational cost and communication overhead.

D. Speech Synthesis and Robot Voice Output

Speech synthesis is performed using edge-tts, which supports Japanese speech generation. The synthesized audio data is generated from the response text and transmitted to the robot for playback.

IV. EXPERIMENTS

A. Hardware Setup

The experiments were conducted on the Open Duck Mini v2 bipedal robot platform. The onboard computing is hosted on a Raspberry Pi Zero 2 W running the locomotion runtime and associated hardware-interface modules. Locomotion control is executed as a closed-loop policy at 50 Hz, where the robot reads the onboard IMU and actuator state feedback to build the observation vector and then streams joint-position targets to the actuators.

Fig. 4. Hardware setup of the Open Duck Mini V2 platform and controller used in the experiments. The robot is equipped with onboard computation and actuators for real-time locomotion control.

TABLE II

SYSTEM PROMPT AND DIALOGUE POLICY CONFIGURATION

Item	**Setting**
LLM system prompt	The assistant persona is defined as a friendly robot named Duck; the model is instructed to answer in concise Japanese with approximately 1–2 sentences while maintaining consistency with the prior dialogue context.
Dialogue history	Bounded multi-turn buffer with maximum of 10 user–assistant turns; the system prompt is inserted as the first message at each inference request.
Message composition	Ordered as [system prompt, recent history messages, current user utterance appended last]. After generation, the assistant response is appended to the history for subsequent turns.

The Open Duck Mini v2 system provides 14 controlled joints: left and right leg joints (hip yaw/roll/pitch, knee, and ankle on each side) as well as a neck pitch and a 3-DoF head (pitch/yaw/roll). Actuation is realized with Feetech servo motors driven through a USB-serial interface, as implemented in the robot-side hardware interface. In the IMU pipeline, a BNO055 sensor provides gyro and accelerometer measurements, while foot-contact sensing is implemented as digital inputs on the GPIO lines.

This modular hardware organization supports the coexistence of locomotion control and speech interaction by keeping sensor acquisition and actuator command streaming aligned with the real-time control.

B. Experimental Setup

To evaluate the proposed system, we quantitatively measured the impact of concurrently executing locomotion control and speech interaction. As summarized in Table II, the

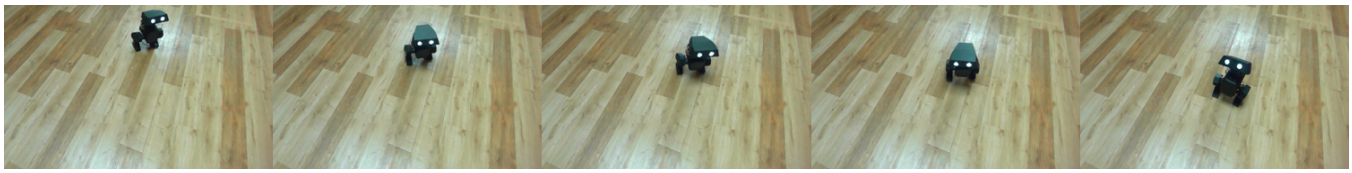

Fig. 5. Snapshots of the Open Duck Mini robot during locomotion.

TABLE III

COMPARISON OF LOCOMOTION CONTROL CYCLE (UNIT: MS, TARGET: $\leq$ 20 MS)

Metric	Without Speech Interaction	With Speech Interaction
Mean	8.677	8.824
Standard deviation	0.221	0.734
Difference in mean	0.147 ms (+1.69%)	

TABLE IV

LATENCY ANALYSIS OF DIALOGUE PROCESSING (UNIT: S)

Processing stage	Mean	Standard deviation
Speech recognition	0.87	0.60
Response generation	0.82	0.05
Speech synthesis	1.19	0.29
Total processing time	2.91	0.71

TABLE V

EXAMPLES OF CONTEXT REFERENCING

Turn	User utterance	System response
5	It rained yesterday	I might go outside once the rain stops
7	What was the weather yesterday?	It rained yesterday (context-aware)
1	What is your name?	My name is Duck
10	Tell me your name	I am Duck (consistent response)

dialogue policy uses a Duck persona defined by a fixed system prompt, maintains a bounded multi-turn history (up to 10 user–assistant turns). The experimental setup is as follows: the proposed system, implemented as an extension of the OSS locomotion control framework, was deployed on the Open Duck Mini robot. On an external client (laptop), speech recognition using Whisper, response generation using the Groq API (*llama-3.1-8b-instant*), and speech synthesis using edge-tts were executed. The robot and the external client were connected via TCP/IP network communication. The target locomotion control frequency was set to 50 Hz, consistent with the original OSS configuration.

The evaluation metrics were defined as follows: (1) locomotion control cycle, including the mean, standard deviation, and distribution of the processing time per control loop (target: 50 Hz, i.e., within 20 ms); (2) dialogue processing time, defined as the latency of each processing stage until completion of audio transmission; and (3) context referencing success, defined as the consistency of responses for utterances that require reference to recent conversational context.

C. Experimental Results

1) Impact on Locomotion Control Cycle: The measurement results of the locomotion control cycle are presented in Table III. In the condition without speech interaction, the average processing time was 8.677 ms (standard deviation: 0.221 ms), whereas in the condition with speech interaction, it was 8.824 ms (standard deviation: 0.734 ms). The difference in the average processing time was 0.147 ms (+1.69%).

Compared to the target control frequency of 50 Hz (20 ms), both conditions maintained a sufficient margin. These results indicate that the distributed architecture of the proposed system enables speech interaction processing with negligible impact on the real-time performance of the locomotion control loop.

2) Latency Analysis of Dialogue Processing: The processing time measured over a 13-turn dialogue session is summarized in Table IV. Speech recognition required an average of 0.87 s (standard deviation: 0.60 s), response generation required 0.82 s on average (standard deviation: 0.05 s), and speech synthesis required 1.19 s on average (standard deviation: 0.29 s). The total processing time, defined as the duration from speech recognition to completion of audio transmission, was 2.91 s on average (standard deviation: 0.71 s).

Since all computationally intensive processes are executed on the external client, the computational load on the robot is limited to receiving and playing back audio data. This design effectively isolates the robot from latency-intensive processing tasks.

3) Effect of Multi-turn Dialogue History: To evaluate the effect of dialogue history management on context referencing, a 13-turn continuous dialogue session was analyzed. Table V presents representative examples of context-aware responses.

At turn 5, the user uttered "It rained yesterday," and the system responded, "I might go outside once the rain stops." At turn 7, when the user asked, "What was the weather yesterday?", the system responded, "It rained yesterday," demonstrating that it correctly referenced the prior context from turn 5. This result indicates that the system maintains conversational consistency by leveraging dialogue history.

Similarly, at turn 1, the system responded to "What is your name?" with "My name is Duck," and at turn 10, it responded to "Tell me your name" with "I am Duck," confirming consistent responses based on stored dialogue history.

V. Conclusion

In this paper, we proposed *SpeeLo*, an integration method and evaluation framework for incorporating speech interaction into a reinforcement learning-based bipedal robot under constrained computational resources. The proposed system adopts a distributed architecture in which a lightweight speech server on the robot handles locomotion control and audio playback, while computationally intensive processes, including speech recognition, response generation, and speech synthesis, are offloaded to an external client. This design minimizes interference between locomotion control and speech processing.

Furthermore, by introducing dialogue history management, the system enables context-aware response generation for short and elliptical utterances. Experimental results on a physical robot demonstrated that speech interaction has minimal impact on the locomotion control cycle while improving response consistency. In addition, dialogue processing latency was analyzed by decomposing it into speech recognition, response generation, and speech synthesis stages.

As future work, extending the system to include onboard speech acquisition is important to better reflect real-world conditions, including ego-noise and environmental disturbances. The proposed architecture and evaluation framework provide a foundation for integrating speech interaction into resource-constrained bipedal robots.

References

[1] C. Y. Kim, C. P. Lee, and B. Mutlu, "Understanding large-language model (llm)-powered human-robot interaction," in *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '24, New York, NY, USA, 2024, p. 371–380. [Online]. Available: https://doi.org/10.1145/3610977.3634966

[2] L. Grassi, C. T. Recchiuto, and A. Sgorbissa, "Enhancing llm-based human-robot interaction with nuances for diversity awareness," in *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, 2024, pp. 2287–2294.

[3] M. Bossema, S. Ben Allouch, A. Plaat, and R. Saunders, "Llm-enhanced interactions in human-robot collaborative drawing with older adults," in *2025 34th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2025, pp. 700–707.

[4] D. Tozadore, N. Ertug, Y. Chaker, and M. Abderrahim, "Robobuddy in the classroom: Exploring llm-powered social robots for storytelling in learning and integration activities," in *2025 34th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2025, pp. 1543–1549.

[5] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.

[6] G. Skantze, "Turn-taking in conversational systems and human-robot interaction: A review," *Computer Speech & Language*, vol. 67, p. 101178, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S088523082030111X

[7] A. Pirrone, "Open Duck Mini: A miniature version of the BDX droid," 2025. [Online]. Available: https://github.com/apirrone/Open_Duck_Mini

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[9] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine, "Learning to walk via deep reinforcement learning," 2019. [Online]. Available: https://arxiv.org/abs/1812.11103

[10] D. Müller, E. Knoop, D. Mylonopoulos, A. Serifi, M. A. Hopkins, R. Grandia, and M. Bächer, "Olaf: Bringing an animated character to life in the physical world," 2025. [Online]. Available: https://arxiv.org/abs/2512.16705

[11] R. Grandia, E. Knoop, M. Hopkins, G. Wiedebach, J. Bishop, S. Pickles, D. Müller, and M. Bächer, "Design and control of a bipedal robotic character," in *Robotics: Science and Systems XX*, ser. RSS2024. Robotics: Science and Systems Foundation, Jul. 2024. [Online]. Available: http://dx.doi.org/10.15607/RSS.2024.XX.103

[12] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[13] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 2811–2817.

[14] L. Bao, T. Peng, and C. Zhou, "Sim-to-real transfer in deep reinforcement learning for bipedal locomotion," 2025. [Online]. Available: https://arxiv.org/abs/2511.06465

[15] G. Mothish, K. Rajgopal, R. Kola, M. Tayal, and S. Kolathaya, "Stoch biro: Design and control of a low cost bipedal robot," 2023. [Online]. Available: https://arxiv.org/abs/2312.06512

[16] Y. Huang, Y. Zeng, and X. Xiong, "Stride: An open-source, low-cost, and versatile bipedal robot platform for research and education," 2024. [Online]. Available: https://arxiv.org/abs/2407.02648

[17] Y. Chi, Q. Liao, J. Long, X. Huang, S. Shao, B. Nikolic, Z. Li, and K. Sreenath, "Demonstrating berkeley humanoid lite: An open-source, accessible, and customizable 3d-printed humanoid robot," 2025. [Online]. Available: https://arxiv.org/abs/2504.17249

[18] B. Xia, B. Li, J. Lee, M. Scutari, and B. Chen, "The duke humanoid: Design and control for energy efficient bipedal locomotion using passive dynamics," 2025. [Online]. Available: https://arxiv.org/abs/2409.19795

[19] P. Allgeuer, H. Farazi, M. Schreiber, and S. Behnke, "Child-sized 3d printed igus humanoid open platform," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 33–40.

[20] H. Shi, W. Wang, S. Song, and C. K. Liu, "Toddlerbot: Open-source ml-compatible humanoid platform for loco-manipulation," 2025. [Online]. Available: https://arxiv.org/abs/2502.00893

[21] L. Lopes and A. Teixeira, "Human-robot interaction through spoken language dialogue," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, vol. 1, 2000, pp. 528–534 vol.1.

[22] E. Billing, J. Rosén, and M. Lamb, "Language models for human-robot interaction," in *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 905–906. [Online]. Available: https://doi.org/10.1145/3568294.3580040

[23] M. M. Reimann, F. A. Kunneman, C. Oertel, and K. V. Hindriks, "A survey on dialogue management in human-robot interaction," *J. Hum.-Robot Interact.*, vol. 13, no. 2, Jun. 2024. [Online]. Available: https://doi.org/10.1145/3648605

[24] T. Yamazaki, K. Yoshikawa, T. Kawamoto, T. Mizumoto, M. Ohagi, and T. Sato, "Building a hospitable and reliable dialogue system for android robots: a scenario-based approach with large language models," *Advanced Robotics*, vol. 37, no. 21, pp. 1364–1381, 2023. [Online]. Available: https://doi.org/10.1080/01691864.2023.2244554

[25] M. Y. Baihaqi, A. García Contreras, S. Kawano, and K. Yoshino, "Llm-driven approach for motion control in human-robot dialogue for elevating engagement," in *2025 34th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2025, pp. 544–550.

[26] V. J. Zhong, E. Studerus, and S. Vonschallen, "Integrating llm into a socially assistive robot for social dialogue: An exploratory study in a nursing home*," in *2025 34th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2025, pp. 1165–1172.

[27] L. Mai and J. Carson-Berndsen, "Real-time textless dialogue generation," 2025. [Online]. Available: https://arxiv.org/abs/2501.04877