二足歩行ロボットへの音声対話システムの構築と評価

小林聖人 [1,2]
[1] 大阪大学 [2] 神戸大学
kobayashi.masato.cmc@osaka-u.ac.jp

概要

二足歩行ロボットでは，リアルタイム性を要する歩行制御と計算負荷の高い音声処理を同時に安定動作させることが設計上の重要課題となる．本論文では，計算資源が制約された二足歩行ロボットを対象に，音声対話機能を統合する分散型アーキテクチャと，その評価指針を提案する．提案方式では，ロボット側には歩行制御と音声出力のみを担う軽量サーバーを配置し，音声処理を外部クライアントへ分離するとともに，会話履歴管理により文脈を考慮した応答生成を可能とする．実機実験により，歩行制御周期への影響が小さいことと文脈参照の有効性を定量的に示し，本構成が計算資源制約下で安全に音声対話を統合可能であることを確認した．

1 はじめに

近年，ロボット技術と自然言語処理技術の発展により，人間と音声を介して対話可能なロボットシステムの研究・開発が進められている [1, 2]．歩行可能なロボットにおいては，移動動作を継続しながら人間と音声対話を行うことが，実環境での利用を想定したシステム設計において重要である．そのため，歩行制御と音声対話を同時に実行可能なシステム構成が重要である．

一方で，歩行制御はリアルタイム性が要求されるため，音声認識や応答生成などの計算負荷の高い処理を同一計算資源上で実行すると，遅延や資源競合により制御ループへ影響が生じ得る．近年では Whisper [3] に代表される大規模な音声認識モデルが利用可能になっているが，小型ロボット上の限られた計算資源で常時実行することは容易ではない．さらに，実環境での対話では短い発話や省略表現が頻出するため，直前文脈を参照できない場合には応答の整合性が損なわれやすい [4]．したがって，既存の歩行制御の挙動を損なわずに音声対話機能を追加するためには，処理の分離や優先度設計，通信断を含む運用上の例外に配慮した実装方針，および会話履歴に基づく文脈保持機構が必要となる．

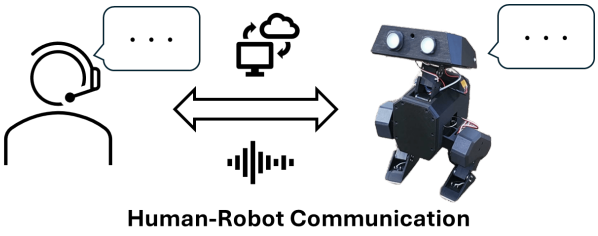

図 1 二足歩行ロボット Open Duck Mini

本論文では，図 1 が示すようにオープンソースの強化学習ベース二足歩行ロボット Open Duck Mini を対象として，音声対話機能を付加する統合方式を提案する．提案方式では，ロボット上では歩行制御と音声出力を担う軽量な音声サーバーのみを動作させ，音声認識（Whisper），応答生成（Groq API），音声合成（edge-tts）といった計算負荷の高い処理は外部クライアントに分離する．また，マルチターンの会話履歴を管理し，省略表現を含む短い発話に対しても直前文脈を参照した応答生成を可能にする．

さらに，本論文では，歩行制御と対話処理の併走を評価するための観点として，歩行制御周期，対話処理時間，文脈参照の成否と，それらをログから算出するための計測点を整理する．本論文は計算資源が制約された歩行ロボットへ音声対話を安全に統合するための構成例と，定量的評価手法を提示することに主眼を置く．

本論文の貢献は以下の 3 点である．

- 歩行制御のリアルタイム性を損なわずに音声対話を設計・実装した点
- マルチターン会話履歴を外部クライアントで管理し，省略発話に対しても文脈を考慮した応答を生成できる構成を実装した点
- 歩行制御周期，対話処理時間，文脈参照の成否という観点から，音声対話統合の影響を実機で定量評価した点

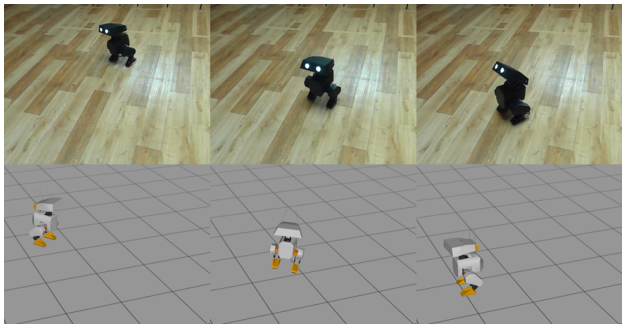

図 2　シミュレーションと実環境歩行 (Open Duck Mini)

2　関連研究

強化学習を用いたロボット歩行制御は近年活発に研究されている [5, 6, 7, 8]．MuJoCo [9] に代表されるシミュレータ上で歩行ポリシーを学習することで，実機実験の負担を軽減しつつ，多様な環境条件に対して頑健な歩行を獲得する手法が提案されてきた [10]．また，学習済みポリシーを ONNX 形式へ変換し，実ロボット上で推論を実行することで，シミュレーションと実機の間を橋渡しする実装形態も広く用いられている [11]．図 2 のように，Open Duck Mini は Apache-2.0 ライセンスで公開されているオープンソースの小型二足歩行ロボットプラットフォームであり，強化学習に基づく歩行制御システムが実装されている．MuJoCo 上で学習した歩行ポリシーを ONNX 形式に変換し，実機上で実行するためのランタイム環境が整備されている点が特徴である．

音声対話システムは，音声認識・言語理解・応答生成・音声合成などの複数モジュールから構成されることが一般的である．とくに音声認識では，Whisper [3] のような大規模モデルが提案され，多様な話者や環境雑音に対する頑健性が報告されている．一方で，この種のモデルは計算コストが高く，計算資源が限られた小型ロボット上で常時稼働させる場合には，リアルタイム処理との競合や遅延に配慮したシステム構成が必要となる [12]．さらに実環境での対話では，短い発話や省略表現が頻出するため，直前文脈を参照できない場合には応答の整合性が損なわれやすく，会話履歴に基づく文脈保持が重要となる．

以上を踏まえると，歩行制御と音声対話を同一ロボット上で同時に動作させるには，計算資源の競合や処理遅延が制御安定性に影響し得るため，処理の分離，監視，および通信断等の例外に対する復帰を含む構成設計が重要となる．そこで本研究は，音声認識・応答生成・音声合成を外部クライアントへ分離する分散アーキテクチャと，ロボット側での非同期処理により歩行制御への干渉を抑える統合方式を示し，定量的評価によりその有効性を検証する点に特徴がある．

3　手法

3.1　システムアーキテクチャ

本システムは，図 3 のように Open Duck Mini の OSS 歩行制御システムを拡張し，歩行制御の挙動を変更せずに音声対話機能を付加することを目的として設計した．本研究では，歩行制御を維持するため，計算負荷の高い音声処理をロボット外へ分離し，ロボット側には音声データの出力に必要な最小限の処理のみを配置する分散アーキテクチャを採用する．システムは，(1) 歩行制御モジュール，(2) 音声サーバー，(3) 音声クライアントの 3 つのコンポーネントから構成される．

歩行制御モジュールは歩行制御を実行しつつ，プロセス内部で音声サーバーをバックグラウンドスレッドとして起動する．これにより，歩行制御のメインループと音声処理が同一スレッドで競合しない構成とし，既存の制御ロジックに変更を加えずに音声機能を追加可能とした．

音声サーバーはロボット上で動作し，バックグラウンドスレッドとして起動される．ネットワーク経由で外部クライアントから送信される音声データを受信して再生する．受信した音声データは音声再生機能を利用して出力し，サーバー側では音声認識や応答生成などの重い処理を行わない．通信エラーに対しては，受信タイムアウトや接続断を検知した場合に例外で停止しないように処理を分岐し，再接続可能な状態を維持する．

音声クライアントは外部端末（PC 等）で実行され，音声認識（Whisper），応答生成（Groq API），音声合成（edge-tts）を担当する．応答生成には Groq API の llama-3.1-8b-instant モデルを使用した．生成された音声データをロボットへ送信することで，ロボット側の計算資源消費を抑えつつ対話機能を実現する．また，対話の文脈を扱うため，会話履歴を保持し，応答生成時に過去発話を参照可能とする．

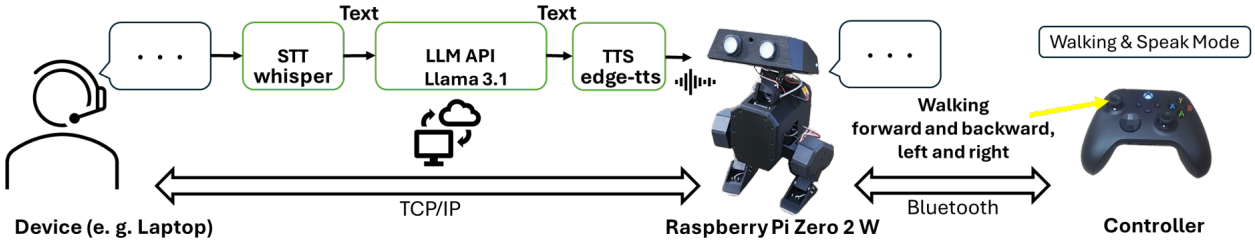

図 3　システム概要

3.2　音声認識

音声認識には Whisper [3] を用いる．Whisper は大規模な多言語データで学習された音声認識モデルであり，日本語を含む多言語に対応する．本システムでは，音声認識を音声クライアント上で実行し，base モデルを使用する．認識結果テキストを後段の応答生成に入力する．音声認識処理をロボット外で実行することで，ロボット側の計算資源使用量を抑え，歩行制御ループへの影響を回避することを意図している．

3.3　マルチターン会話履歴管理

音声対話では，短い発話や省略表現が生じるため，直前の発話文脈を参照できることが重要となる．本システムでは，ユーザ発話とシステム応答のペアからなる会話履歴を保持し，応答生成時に履歴を入力として付与することで文脈保持を行う．

表 1 に，会話の処理フローと履歴の状態を示す．会話履歴は，システムプロンプト，ユーザメッセージ，ロボットメッセージから構成し，応答生成時にはシステムプロンプトを先頭に配置したうえで時系列順に配列する．システムプロンプトには，ロボットの役割（親しみやすいアシスタントロボット，名前はダック），応答スタイル（簡潔で短めの日本語，1-2 文程度），文脈理解の重要性を明記する．履歴の最大長は 10 ターン（ユーザ・アシスタントのペア）とし，これを超える場合には最も古いメッセージから削除することでメモリ使用量を一定に保つ．この設計により，最近の文脈を保持しつつ，計算資源と通信量の増加を抑制する．

3.4　音声合成とロボット音声効果

音声合成には edge-tts を使用する．edge-tts は日本語音声の合成が可能であり，応答テキストから音声データを生成する．

表 1　会話履歴を用いた処理フロー

段階	内容
初期状態	1 会話目の履歴が保持されている 履歴: [user1: 名前は何ですか？, robot1: 私はダックと言います．]
ユーザー発話	「何が得意ですか？」
LLM 送信データ	システムプロンプト + 1 会話目の履歴 + 新しいユーザー発話
LLM 応答	「親しみやすい会話が得意です．」
履歴保存	2 会話目のユーザー発話と応答を履歴に追加 履歴: [user1, robot1, user2, robot2]

4　実験

4.1　実験設定

本システムの評価のため，歩行制御と音声対話の同時実行における影響を定量的に計測した．実験環境は以下の通りである：Open Duck Mini ロボット上で OSS 歩行制御システムを拡張した本システムを動作させ，外部クライアント（Laptop）上で Whisper（base モデル）による音声認識，Groq API（llama-3.1-8b-instant）による応答生成，edge-tts による音声合成を実行した．ロボットと外部クライアント間はネットワーク通信（TCP/IP）により接続した．歩行の目標制御周波数は OSS と同じく 50Hz に設定した．

評価指標として，(1) 歩行制御周期：制御ループ 1 周期の処理時間の平均・標準偏差・分布（目標：50Hz，すなわち 20ms 以内），(2) 対話処理時間：音声送信完了までの各処理段階の時間，(3) 文脈参照の成否：直前文脈を参照する発話に対する応答の整合性を設定した．

表 2　歩行制御周期の比較（単位：ms，目標：20ms 以下）

指標	音声対話なし	音声対話あり
平均	8.677	8.824
標準偏差	0.221	0.734
平均の差	0.147ms (+1.69%)	

表 3　対話処理の時間分析（単位：秒）

処理段階	平均	標準偏差
音声認識	0.87	0.60
応答生成	0.82	0.05
音声合成	1.19	0.29
総処理時間	2.91	0.71

4.2　実験結果

4.2.1　歩行制御周期への影響

歩行制御周期の計測結果を表 2 に示す．音声対話なし条件では平均 8.677ms（標準偏差 0.221ms），音声対話あり条件では平均 8.824ms（標準偏差 0.734ms）であり，平均処理時間の差は 0.147ms（+1.69%）であった．目標とする制御周波数 50Hz（20ms）に対して，両条件とも十分な余裕があった．この結果は，本システムの分散アーキテクチャにより，音声対話処理が歩行制御ループのリアルタイム性に実質的な影響を与えないことを示している．

4.2.2　対話処理の遅延分析

13 ターンの対話セッションにおける処理時間の計測結果を表 3 に示す．音声認識処理は平均 0.87 秒（標準偏差 0.60 秒），応答生成は平均 0.82 秒（標準偏差 0.05 秒），音声合成は平均 1.19 秒（標準偏差 0.29 秒）であった．総処理時間（音声認識から音声送信完了まで）は平均 2.91 秒（標準偏差 0.71 秒）であり，各処理が外部クライアントで実行されるため，ロボット側の計算資源への負荷は音声データの受信・再生のみに限定された．

4.2.3　マルチターン会話履歴の効果

会話履歴管理による文脈参照の効果を確認するため，13 ターンの連続対話セッションを分析した．表 4 に文脈参照の具体例を示す．ターン 5 で「昨日は雨でした」という発話に対して「雨がやんだら外に出てみようかな」と応答し，ターン 7 で「昨日の天気を教えて」という発話に対して「昨日は雨でしたね」と応答した．この応答は，ターン 5 の会話履歴を参照しており，文脈を保持した一貫性のある対話が実現されていることを示している．また，ターン 1 で「名前は何ですか？」に対して「私はダックと言います」と応答し，ターン 10 で「あなたの名前を教えて」に対して「私はダックです」と応答するなど，会話履歴に基づく一貫した応答が確認された．

表 4　文脈参照の具体例

ターン	ユーザ発話	システム応答
5	昨日は雨でした	雨がやんだら外に出てみようかな
7	昨日の天気を教えて	昨日は雨でしたね（文脈参照）
1	名前は何ですか？	私はダックと言います
10	あなたの名前を教えて	私はダックです（一貫性維持）

5　おわりに

本論文では，強化学習に基づく歩行制御を備えた二足歩行ロボット Open Duck Mini を対象に，計算資源の制約がある状況下において音声対話機能を付加する統合方式と評価方法を提案した．提案方式では，ロボット上には歩行制御と音声出力を担う軽量な音声サーバーのみを配置し，音声認識（Whisper），応答生成（Groq API），音声合成（edge-tts）といった計算負荷の高い処理を外部クライアントへ分離する分散アーキテクチャを採用した．これにより，歩行制御ループと音声処理が計算資源・スケジューリングの面で競合しにくい構成を実現した．また，会話履歴管理を導入することで，省略表現を含む短発話に対しても直前文脈を参照した応答生成を可能にした．実機評価では，音声対話の有無で歩行制御周期を比較し，目標制御周波数（50Hz）に対して十分な余裕を保ったまま併走できることを確認した．さらに，対話処理時間を音声認識・応答生成・音声合成の各段階に分解して計測し，遅延要因をログに基づいて分析可能であることを示した．以上より，本手法の有効性が示された．

今後は，対話内容に応じたロボット行動の切り替えに加え，通信断時のフォールバックや再接続後の対話継続を実装し，実環境での継続対話を評価する．本研究で示した構成と計測点は，そのための設計・検証の基盤となる．

参考文献

[1] Callie Y. Kim, Christine P. Lee, and Bilge Mutlu. Understanding large-language model (llm)-powered human-robot interaction. In **Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction**, HRI '24, p. 371–380, New York, NY, USA, 2024.

[2] Lucrezia Grassi, Carmine Tommaso Recchiuto, and Antonio 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)**, pp. 2287–2294, 2024.

[3] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In **Proceedings of the 40th International Conference on Machine Learning**, ICML'23. JMLR.org, 2023.

[4] Gabriel Skantze. Turn-taking in conversational systems and human-robot interaction: A review. **Computer Speech Language**, Vol. 67, p. 101178, 2021.

[5] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[6] Tuomas Haarnoja, Sehoon Ha, Aurick Zhou, Jie Tan, George Tucker, and Sergey Levine. Learning to walk via deep reinforcement learning, 2019.

[7] David Müller, Espen Knoop, Dario Mylonopoulos, Agon Serifi, Michael A. Hopkins, Ruben Grandia, and Moritz Bächer. Olaf: Bringing an animated character to life in the physical world, 2025.

[8] Ruben Grandia, Espen Knoop, Michael Hopkins, Georg Wiedebach, Jared Bishop, Steven Pickles, David Müller, and Moritz Bächer. Design and control of a bipedal robotic character. In **Robotics: Science and Systems XX**, RSS2024. Robotics: Science and Systems Foundation, July 2024.

[9] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In **2012 IEEE/RSJ International Conference on Intelligent Robots and Systems**, pp. 5026–5033, 2012.

[10] Zhongyu Li, Xuxin Cheng, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In **2021 IEEE International Conference on Robotics and Automation (ICRA)**, pp. 2811–2817, 2021.

[11] Lingfan Bao, Tianhu Peng, and Chengxu Zhou. Sim-to-real transfer in deep reinforcement learning for bipedal locomotion, 2025.

[12] Long Mai and Julie Carson-Berndsen. Real-time textless dialogue generation, 2025.