

|                     |                                                                                   |
|---------------------|-----------------------------------------------------------------------------------|
| <b>Title</b>        | Rollopod: 多脚歩行と登坂ローリング走行が可能な新しいヘキサポッドロボット の無次元輸送コスト削減効果検証                         |
| <b>Author(s)</b>    | 原田, 恒迪                                                                            |
| <b>Citation</b>     |                                                                                   |
| <b>Issue Date</b>   | 2025-03                                                                           |
| <b>Type</b>         | Thesis or Dissertation                                                            |
| <b>Text version</b> | author                                                                            |
| <b>URL</b>          | <a href="http://hdl.handle.net/10119/19811">http://hdl.handle.net/10119/19811</a> |
| <b>Rights</b>       |                                                                                   |
| <b>Description</b>  | Supervisor: 丁 洛榮, 先端科学技術研究科, 修士 (情報科学)                                            |

修士論文

Rollopod: 多脚歩行と登坂ローリング走行が可能な新しいヘキサポッドロボットの無次元輸送コスト削減効果検証

原田 恒迪

主指導教員 Chong Nak Young 教授

北陸先端科学技術大学院大学  
先端科学技術専攻  
(情報科学)

令和7年3月



## Abstract

This paper introduces a novel hexapod robot developed by the authors that can perform both walking and uphill-capable rolling locomotion to improve cost of transport (CoT). Simulation experiments using a deep reinforcement learning-based controller demonstrated that the rolling locomotion mode achieves a 53%–67% reduction in the CoT and provides an 7% slope-climbing capability.

Mobile robots with wheels have been utilized in structured environments such as factories and logistics warehouses. In particular, they contribute to reducing labor burdens and improving process management in indoor goods transportation. However, their applications in unstructured environments and outdoor settings remain limited. Recently, regulatory reforms for remote operations using cameras in facility inspections and patrols have been progressing. As a result, the use of robots for experimental verification is being advanced in plants, construction sites, and infrastructure inspection fields. Furthermore, mobile robots are expected to play a role in hazardous tasks, such as disaster responses or plant accident scenarios. For tasks in unstructured environments, multi-legged robots are a suitable choice. Multi-legged robots are known to outperform wheeled or tracked robots in terms of mobility and versatility. However, due to their high degrees of freedom and control complexity, deploying these robots in real-world environments remains a challenge. The advent of deep reinforcement learning (DRL), which integrates deep learning and reinforcement learning, has shown promise in addressing these challenges.

One remaining issue with multi-legged robots is walking efficiency. These robots consume power to support their weight through their joints, even when stationary. In contrast, wheeled robots transmit their weight directly to the ground, resulting in lower energy consumption for the same distance traveled. To address this issue, wheeled-legged robot have been researched. These robots incorporate passive rollers or active wheels at the leg tips, enabling wheeled locomotion on flat terrain, a combination of wheeled and walking locomotion on moderately rough terrain, and locked-wheel walking on highly rough terrain. Related studies have demonstrated a 50%–83% reduction in CoT through wheeled locomotion compared to pure trotting on flat terrain. These studies indicate that combining legged walking and wheeled locomotion is an effective approach to reducing transportation costs.

While wheeled locomotion is known to be highly efficient, rolling locomotion is another similar method. Wheeled robots have stable main bodies with attached wheels, whereas rolling robots rotate their main bodies entirely. Studies on rolling robots have primarily focused on nonholonomic systems and biomechanics, with most research limited to rolling locomotion on horizontal planes via center-of-gravity control.

In this study, the authors propose a hybrid locomotion method that combines multi-legged walking and kicking motions to enable uphill-capable rolling locomotion for CoT reduction. To realize this locomotion method, the authors developed a novel hexapod robot called "Rollopod-A." This robot can perform general six-legged walking and transform into a circular rolling mode by tilting  $90^\circ$  and folding its legs. In rolling mode, appropriate legs can kick radially outward to generate torque, enabling the robot to overcome small obstacles and climb slopes.

Simulation experiments using a deep reinforcement learning-based controller were conducted on terrains including harsh rough terrain, gentle rough terrain, uphill slopes, and downhill slopes. The results showed that the rolling locomotion mode achieved an average CoT reduction of 51% on gentle rough terrain. Furthermore, on uphill and downhill slopes with a gradient of 7%, maximum CoT reductions of 67% and 63%, respectively, were observed. Additionally, the robot demonstrated an 7% slope-climbing capability, which is not found in other rolling robots.

These results demonstrate that combining multi-legged walking and rolling locomotion is an effective method for achieving high traversal capability on uneven terrain and efficient movement on flat and hill terrains. This study proposes a novel CoT reduction method for multi-legged robots that differs from conventional wheeled-legged robots. However, challenges remain regarding the stability and steering performance of Rollopod-A during rolling locomotion. In particular, improvements are needed for recovery from falls. Additionally, this study was limited to verification in a simplified simulation environment, rather than real-world settings involving stairs or obstacles. Future work will focus on experimental validation with hardware, including the development of an improved version, Rollopod-B, which enhances steering performance using banked rolling. Research will also explore unified control systems based on deep reinforcement learning that integrate walking, rolling, and mode transitions, including recovery from falls.

# 目次

|            |                                      |           |
|------------|--------------------------------------|-----------|
| <b>第1章</b> | <b>はじめに</b>                          | <b>1</b>  |
| 1.1        | 研究背景                                 | 1         |
| 1.1.1      | モバイルロボットの概要                          | 3         |
| 1.1.2      | 多脚式 UGV の課題                          | 4         |
| 1.2        | 研究目的と提案内容                            | 4         |
| 1.3        | 論文の構成                                | 5         |
| <b>第2章</b> | <b>関連研究</b>                          | <b>6</b>  |
| 2.1        | 多脚式ロボット                              | 6         |
| 2.1.1      | Multi-Legged Robot                   | 8         |
| 2.1.2      | Wheeled-Legged Robot                 | 8         |
| 2.2        | ローリングロボット                            | 10        |
| <b>第3章</b> | <b>Rollopod-A</b>                    | <b>12</b> |
| 3.1        | 機械設計                                 | 12        |
| 3.2        | 電気設計                                 | 14        |
| <b>第4章</b> | <b>ローリングによる CoT 削減ロコモーション (提案手法)</b> | <b>17</b> |
| 4.1        | ローリングメカニズム                           | 17        |
| <b>第5章</b> | <b>シミュレータを用いたローリング走行モードの CoT 評価</b>  | <b>19</b> |
| 5.1        | 実験環境                                 | 19        |
| 5.1.1      | NVIDIA Isaac Sim                     | 19        |
| 5.1.2      | NVIDIA Isaac Lab                     | 21        |
| 5.2        | コントローラポリシ学習環境 (NVIDIA Isaac Lab)     | 21        |
| 5.2.1      | Rollopod-A シミュレーションモデル               | 21        |
| 5.2.2      | 深層強化学習アルゴリズム (PPO)                   | 24        |
| 5.2.3      | 学習地形モデル                              | 24        |
| 5.2.4      | コントロールポリシ学習パラメータ・学習設定                | 29        |
| 5.3        | CoT 測定シミュレーション実験                     | 40        |
| 5.3.1      | CoT 計算式                              | 40        |
| 5.3.2      | CoT 計測地形モデル                          | 40        |
| 5.3.3      | CoT 測定手順                             | 40        |

|              |                           |           |
|--------------|---------------------------|-----------|
| <b>第 6 章</b> | <b>結果と考察</b>              | <b>42</b> |
| 6.1          | ウェーブ地形 CoT 測定結果 . . . . . | 43        |
| 6.2          | 勾配地形 CoT 測定結果 . . . . .   | 45        |
| 6.3          | 考察 . . . . .              | 46        |
| <b>第 7 章</b> | <b>結論</b>                 | <b>62</b> |
|              | 謝辞                        | 65        |
|              | 参考文献                      | 65        |

# 目 次

|      |                                                                                       |    |
|------|---------------------------------------------------------------------------------------|----|
| 1.1  | Rollopod-A の多脚歩行モードとローリング走行モード . . . . .                                              | 1  |
| 2.1  | 研究されている多脚式 UGV の例 . . . . .                                                           | 7  |
| 2.2  | 4 脚 UGV の歩行効率に注目した関連研究 . . . . .                                                      | 8  |
| 2.3  | 車輪付き脚ロボットの関連研究 . . . . .                                                              | 9  |
| 2.4  | ローリング走行の関連研究 . . . . .                                                                | 11 |
| 3.1  | Rollopod-A 機体コンポーネント . . . . .                                                        | 13 |
| 3.2  | Rollopod-A 電気設計 . . . . .                                                             | 15 |
| 3.3  | ESP32 用オリジナル拡張ボード . . . . .                                                           | 16 |
| 4.1  | Rollopod-A の蹴り出し動作によるローリングメカニズム図 . . . . .                                            | 18 |
| 5.1  | Rollopod-A の USD 記述されたシミュレーションモデル、シミュレーション計算コスト軽減のため簡略化 . . . . .                     | 23 |
| 5.2  | 学習に用いられる地形例 . . . . .                                                                 | 26 |
| 5.3  | 多脚歩行モードにおいて発生した地面との不適切干渉によるスタック例 (1)、1 体のアクタがスポン直後に地形と干渉してその場に拘束されている . . . . .       | 27 |
| 5.4  | 多脚歩行モードにおいて発生した地面との不適切干渉によるスタック例 (2) . . . . .                                        | 28 |
| 5.5  | 多脚歩行ポリシの報酬推移グラフ (1) . . . . .                                                         | 32 |
| 5.6  | 多脚歩行ポリシの報酬推移グラフ (2) . . . . .                                                         | 33 |
| 5.7  | 多脚歩行コントローラアーキテクチャ . . . . .                                                           | 34 |
| 5.8  | 平坦地形における多脚歩行モードのタイムラプス . . . . .                                                      | 35 |
| 5.9  | ローリング走行ポリシの報酬推移グラフ . . . . .                                                          | 37 |
| 5.10 | ローリング走行コントローラアーキテクチャ . . . . .                                                        | 38 |
| 5.11 | 平坦地形におけるローリング走行モードのタイムラプス . . . . .                                                   | 39 |
| 6.1  | 高周波ウェーブ地形の振幅と CoT の関係：移動モード別の比較（縦の破線は歩行ポリシの最大学習レベル、縦の二点鎖線はローリングポリシの最大学習レベル） . . . . . | 43 |

|      |                                                                                |    |
|------|--------------------------------------------------------------------------------|----|
| 6.2  | 低周波ウェーブ地形の振幅と CoT の関係：移動モード別の比較（縦の破線は歩行ポリシの最大学習レベル、縦の二点鎖線はローリングポリシの最大学習レベル）    | 44 |
| 6.3  | 上り勾配地形の傾斜度と CoT の関係：移動モード別の比較（縦の破線は歩行ポリシの最大学習レベル、縦の二点鎖線はローリングポリシの最大学習レベル）      | 45 |
| 6.4  | 下り勾配地形の傾斜度と CoT の関係：移動モード別の比較（縦の破線は歩行ポリシの最大学習レベル、縦の二点鎖線はローリングポリシの最大学習レベル）      | 46 |
| 6.5  | 高周波ウェーブ地形における振幅 $0.07m$ のローリング走行モードタイムラプス                                      | 48 |
| 6.6  | 高周波ウェーブ地形における振幅 $0.20m$ のローリング走行モードタイムラプス                                      | 49 |
| 6.7  | 低周波ウェーブ地形における振幅 $0.20m$ の多脚歩行モードタイムラプス                                         | 51 |
| 6.8  | 低周波ウェーブ地形における振幅 $0.20m$ のローリング走行モードタイムラプス                                      | 52 |
| 6.9  | 上り勾配地形における傾斜度 7% のローリング走行モードタイムラプス                                             | 54 |
| 6.10 | 上り勾配地形における傾斜度 9% のローリング走行モードタイムラプス                                             | 55 |
| 6.11 | 上り勾配地形における傾斜度 11% のローリング走行モードタイムラプス                                            | 56 |
| 6.12 | 下り勾配地形における傾斜度 12% のローリング走行モードタイムラプス                                            | 58 |
| 6.13 | 下り勾配地形における傾斜度 16% のローリング走行モードタイムラプス                                            | 59 |
| 6.14 | 下り勾配地形における傾斜度 10% の多脚歩行モードタイムラプス、スタックおよび転倒しているアクタは、5.2.3 節で述べた不適切干渉によるものと推測される | 60 |
| 6.15 | 下り勾配地形における傾斜度 14% の多脚歩行モードタイムラプス                                               | 61 |
| 7.1  | Rollopod-B のプロトタイプ                                                             | 63 |
| 7.2  | 可変疑似重力・模擬地形実験装置プロトタイプ                                                          | 64 |

# 表 目 次

|     |                                                       |    |
|-----|-------------------------------------------------------|----|
| 3.1 | 電子デバイス構成 . . . . .                                    | 14 |
| 5.1 | シミュレーションサーバスペック . . . . .                             | 19 |
| 5.2 | PhysX 物理エンジンの機能 . . . . .                             | 20 |
| 5.3 | ポリシ学習におけるドメインランダム化とサンプリング範囲 . . . . .                 | 22 |
| 5.4 | ポリシ学習地形パラメータ . . . . .                                | 25 |
| 5.5 | ポリシ学習パラメータ . . . . .                                  | 29 |
| 5.6 | 多脚歩行ポリシ観測空間 (Observations) . . . . .                  | 30 |
| 5.7 | 多脚歩行ポリシ報酬設計 (Rewards) . . . . .                       | 30 |
| 5.8 | ローリング走行ポリシ観測空間 (Observations) . . . . .               | 36 |
| 5.9 | ローリング走行ポリシ報酬設計 (Rewards) . . . . .                    | 36 |
| 6.1 | 移動モード、地形別 CoT 測定結果とローリング走行による最大 CoT<br>削減効果 . . . . . | 42 |

# 第1章 はじめに

本稿では、著者らが開発中の多脚移動ロボット「Rollopod-A」(図 1.1) について紹介し、その特徴的なローリング走行モードにおける無次元輸送コスト (Cost of Transport: CoT) の削減効果をシミュレーション実験的に検証した結果を報告する。また、ローリング走行モードにおいて傾斜度 7% の登坂能力を確認した。この性能は、既存のローリングロボットでは見られない優位性である。Rollopod-A は、多脚歩行ロボット (多脚式 UGV) の移動効率向上および適応可能な環境の拡大を目標に設計・開発された。本機は、後述する車輪付き脚ロボットに代わる新たな移動効率低減手法として高い可能性を秘めている。



図 1.1: Rollopod-A の多脚歩行モードとローリング走行モード

## 1.1 研究背景

近年、バッテリー性能や屋内自己位置推定技術、Simultaneous Localization and Mapping (SLAM) の発展により、物流倉庫や工場内の物資輸送にモバイルロボットが活用されている。Amazon の Fulfillment Center (FC) では、商品ラックを Proteus ロボット [1] がリフトアップし任意の位置に移動させることが可能である。この機能を用いて、商品ラックごとパッキング従業員の元へ輸送し、不要になった商品ラックを倉庫内に自動返却している。これにより、従来従業員による徒歩でのピッ



クアップと比較しピックアップ効率や従業員負担が大きく改善している。また、大規模工場内部では磁気マーカ方式の Automatic Guided Vehicle (AGV) や SLAM 方式の Autonomous Mobile Robot (AMR) により製造ライン間の物資輸送を効率化 [2] が行われている。このように、構造化さらた屋内環境ではモバイルロボットが活用されている。

一方、国土交通省による建造物の定期調査報告（12 条点検）における点検項目・調査方法・結果の判断基準の一部改正の告示 [3] より、2025 年 7 月 1 以降目視による検査についてカメラ等の代替手段の使用を許可する方針を示した。さらに、デジタル庁ではアナログ規制見直しの取組を行っており、これまでに目視検査 1958 項目、実地監査 139 項目が見直されている [4]。これらの規制緩和により、モバイルロボットを用いたカメラや各種センサによる巡回業務やインフラ検査が可能になるため、今後はロボットに最適化されていない屋内環境や屋外環境で活動できるモバイルロボットの需要が高まると予想される。

また、災害時やプラントにおける事故発生時の情報収集および危険作業への活用が期待されている。例えば、2011 年に発生した東日本大震災に伴う福島第一原子力発電所の炉心融解事故 [5] では、原子炉建屋内での作業を行うため「決死隊」が結成された [6]。これらの教訓を踏まえ、多脚式 UGV に関する研究が進められている [7]。さらに、能登半島地震においては、自衛隊による 4 脚ロボット「Vision 60」の実地試験 [8] が実施されるなど、多脚式 UGV の実用化に向けた検証が進展している。

しかし、屋外環境においてはエンターテインメントにおける空撮や観測について Unmanned Aerial Vehicle (UAV) が用いられているが、地面を移動するモバイルロボットの実用例は少ない。これは、構造化された安定的な屋内環境と異なり、屋外環境は変化し続けており外乱が多様であることから、モバイルロボットを安全・効率的に動作させるのが難しいことに起因する。そこで、現在最も注目されている研究テーマは多脚歩行ロボットと深層強化学習ベースコントローラの組合せ手法である。多脚歩行ロボットはホイール式や履带式ロボットと比較し、機動性と汎用性の観点で優れていることが知られている [9]。しかし、ロボットの高い自由度と制御の複雑さにより、実環境で運用可能なロボットの実現は依然として課題であった。2020 年にチューリッヒ大学とインテルの研究者らにより、深層学習と強化学習を組合せた深層強化学習（Deep Reinforcement Learning: DRL）ベースのコントローラ [10] が開発され、2022 に本コントローラを搭載した ANYmal-C ロボット [11] が「DARPA Subterranean Challenge[12]」で優勝している。一連の研究では、外乱に対して従来コントローラを超える性能を発揮しており、学習環境外の環境に対しても適応することが示された。これら、社会的背景や高い機動性を持つ多脚ロボットとそれを屋外環境でコントロール可能なコントローラの登場により、多脚式モバイルロボットの研究と活用が一層活発になると予想される。

### 1.1.1 モバイルロボットの概要

モバイルロボットとは、活動地点まで運搬され、自律的に移動して作業を行うロボットを指す。移動空間に応じて、空中を移動する UAV (Unmanned Aerial Vehicle)、地上を移動する UGV (Unmanned Ground Vehicle)、海上や海中を移動する USV(Unmanned Surface Vehicle) の3種類に分類される。

#### 1) UAV: 無人航空機

UAV は空中を飛行するタイプの無人ロボットであり、さまざまな分野で活用されている無人航空機の一つである。その中でも、垂直離着陸 (VTOL: Vertical Take-Off and Landing aircraft) が可能な小型無人航空機であるマルチロータ機は、UAV に分類される。マルチロータタイプの UAV は、固定翼タイプの UAV と比較して高い機動性を持つ点が特徴である。この高機動性を活かし、エンターテインメント業界における空撮、橋脚や高層建造物といった人間がアクセスしにくい高所の検査、災害時の迅速な情報収集など、多岐にわたる用途で活躍している。一方で、飛行中のエネルギー効率が固定翼機と比較して低いという欠点があり、長距離移動が必要となる任務では、固定翼タイプの UAV が採用されることが多い。

#### 2) UGV: 無人地上車両

UGV は、地面を移動する無人ロボットであり多岐にわたる用途で活用が期待されている。UGV はその移動方式によって大きくホイール式と多脚式の2つに分類される。ホイール式 UGV はタイヤや履帯を回転させることで効率的に地面を移動するロボットであり、多脚式 UGV は1本以上の脚を用いた歩行移動を特徴とするロボットである。ホイール式 UGV は、荷重を地面に直接伝える構造上比較的高い重量を安定して移動させることが可能である。また、静止状態でも高い安定性を有しておりその際の電力消費も極めて少ない。一方、多脚式 UGV は複数の自由度を持つ脚を用いて移動するため、階段や岩場などの不整地を走破する能力に優れている。しかしながら、多脚式ロボットはホイール式に比べて制御が極めて複雑であり、動作の安定性やエネルギー効率の面で技術的な課題が残されている。とはいえ、PUDU 社が開発するヒューマノイドロボット「PUDU D9」[13] や Boston Dynamics 社が販売する4脚ロボット「Spot」[14] など具体的な製品が市場に登場し始めており、多脚式 UGV の研究は今後さらに進むことが予想される。

#### 3) USV: 無人船舶

USV は海上・海中を移動するロボットである。USV は海上用と海中用で構造が大きく異なり、海上用では一般的な船舶と同様の船体形状と推進機構を備え、海中用ではバラストタンクや2次元、3次元方向への移動用スクリーを備えている。USV は主に海底調査、ダムなど建造物の水中部分の検査に用いられる。

### 1.1.2 多脚式 UGV の課題

モバイルロボットの社会実装はエンターテインメントや屋内物流分野で進展しているが、屋外環境での応用は依然研究段階にある。不整地や非構造化環境における巡回業務や探査任務では、高い機動性が求められるため、多脚式 UGV が注目されている（図 2.1c、図 2.1d）。しかし、多脚式 UGV は他の UGV と比較して制御が複雑であり、電力効率が低いという課題がある。

多脚式 UGV の制御が困難な要因として多数かつ高い自由度が関係している。一般的な 4 脚ロボットの場合、自由度は 12 であり 6 脚ロボットの場合は 18 個である。これは、車輪式 UGV（3～6）やマルチロータ UAV（4～8）と比較して多いことがわかる。また、各脚はシリアルリンクマニピュレータであるため逆運動学計算により複数の解が求められる。さらに、多脚ロボットは地形との相互作用により最終的な姿勢状態が決定されるため非線形システムとなる。そのため、モデルベース制御 [15], [16] や中枢パターン生成器コントローラ（Central pattern Generator: CPG）[17] などが研究された。近年では、多脚ロボットの新たな制御手法として深層強化学習を用いたコントローラ [10], [11] が提案されており、従来型コントローラと比較し高性能であることが示されている。

また、多脚式 UGV の電力効率の低さは動的支持機構に起因する。ホイール式 UGV は静的支持機構により重量を地面に直接伝えるが、多脚式 UGV では関節を介して重量を支えるため、常時電力を消費する。このため、同距離移動時の消費電力がホイール式 UGV に比べて多くなる。この効率の低さは、屋外環境での運用効率を低下させる要因となっている。

## 1.2 研究目的と提案内容

本稿では、多脚式ロボットの歩行効率に関する課題を解決するため、登坂可能なローリング走行を提案する。これに対応するため、著者らは新しい 6 脚ロボット「Rollopod-A」を開発した（図 1.1）。Rollopod-A は 6 本の脚を持ち、標準的な多脚歩行に加え、脚を折りたたんで円形に変形し、ローリング走行を可能とする。ローリング走行中は適切な脚を放射方向に蹴り出すことで移動を行う。本ロボットの特徴は、ローリング走行による登坂が可能であり、多脚歩行モードとローリング走行モードが相互に干渉しないことである。シミュレーション実験により、地形粗さ別に多脚歩行モードとローリング走行モードの無次元輸送コスト（Cost of Transport: CoT）を比較した結果、平地におけるローリング走行モードで平均 53% の CoT 削減効果が確認された。また、7% の登坂が可能であり、これは他のローリングロボットには見られない能力である。

### 1.3 論文の構成

以下に、本論文の構成を述べる。2章では、多脚式ロボットに関連する研究、文献を取り上げる。3章では、筆者らが開発している Rollopod-A について詳述する。4章では、Rollopod-A の移動に要するエネルギー削減のための提案手法について述べる。5章では、NVIDIA Isaac Sim シミュレータを用いた環境構築と CoT 評価実験を述べる。6章では、評価実験の結果と考察を示す。最後に、7章に結論について述べる。

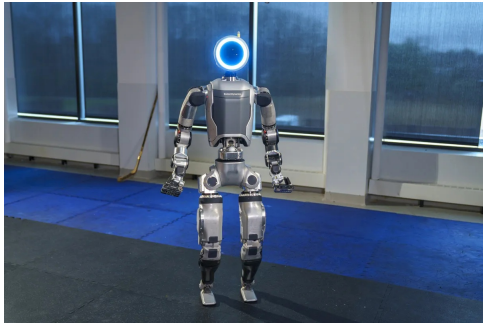
## 第2章 関連研究

本章では、本研究に関連する多脚式ロボット、車輪付き脚ロボット、およびローリングロボットに関する研究を述べる。

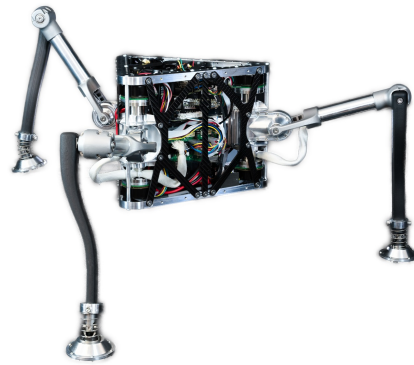
### 2.1 多脚式ロボット

多脚式 UGV は、1 脚以上かつ 1 軸以上の自由度を持つマニピュレータによる歩行機構を有するロボットである。特に、2 足歩行を行うロボット（ヒューマノイドロボット, 図 2.1a）や、4 脚ロボット（Quadroped, 図 2.1c）が多く研究されている。他にも、3 脚（Tripod, 図 2.1b）や 6 脚（Hexapod, 図 2.1d）などが研究対象となっている。

以下では、特に 4 脚以上の多脚式 UGV に関するエネルギー効率に着目した関連研究について述べる。



(a) ヒューマノイドロボット (Electric Atlas[18])



(b) 3脚ロボット (SpaceHopper[19])



(c) 4脚ロボット (ANYmal[9])



(d) 6脚ロボット (RHex[20])

図 2.1: 研究されている多脚式 UGV の例

### 2.1.1 Multi-Legged Robot

初期の代表的な研究として、2014年に発表された StarlETH[7] (図 2.2a) や MIT Cheetah Robot[21] (図 2.2b) が挙げられる。MIT Cheetah Robot では、多脚式 UGV のエネルギー損失をアクチュエータの熱損失、トランスミッションの摩擦損失、ロボットと環境間の相互作用損失の三つに分類して分析している。この分析結果に基づき、高トルク密度モータ、エネルギー回生システム、低損失トランスミッション、低慣性脚を採用した設計が行われた。その結果、Total Cost of Transport (TCoT) の比較において、ASIMO ( $TCoT = 2$ ) や BigDog ( $TCoT = 15$ ) と比較して、MIT Cheetah Robot は  $TCoT = 0.5$  を記録し、効率が大幅に向上していることが確認された。さらに、RSL が発表した ANYmal ロボット [9] (図 2.1c) や Boston Dynamics が 2020 年に一般販売した Spot など、以降多くの研究がこれらの既存の 4 脚 UGV を基に進められている。

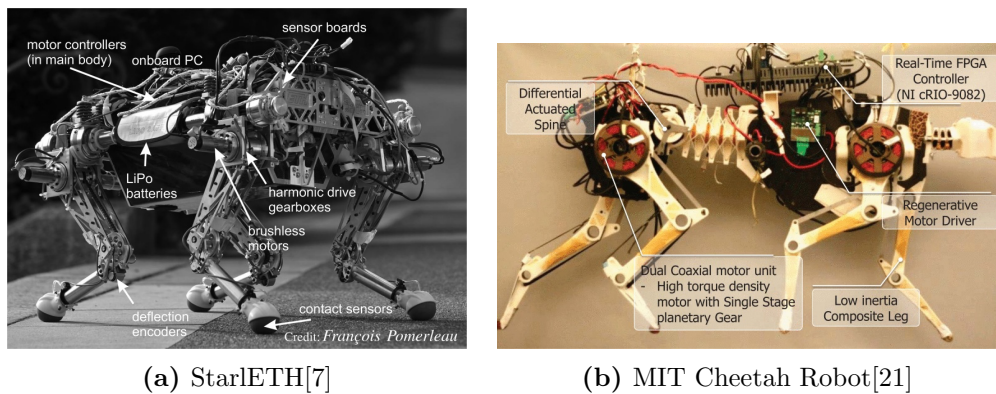


図 2.2: 4 脚 UGV の歩行効率に注目した関連研究

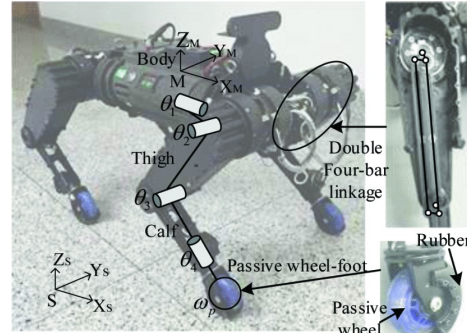
### 2.1.2 Wheeled-Legged Robot

車輪付き脚ロボット (Wheeled-Legged Robot) は、多脚式 UGV の歩行効率向上を目的とした研究分野である。ANYmal ロボットの脚先にアイススケートまたはローラースケートを装着した研究 [22], [23] (図 2.3a, 図 2.3b) では、トロット歩行と比較して CoT が 50% ~ 80% 削減されることが示されている。また、脚先に動力付きホイールを装備した研究 [24], [25] (図 2.3c, 図 2.3d) では、トロット歩行で CoT が 83% 削減され、最高速度が純歩行型 ANYmal ロボットの  $1.5m/s$  を超える  $4m/s$  に達した。

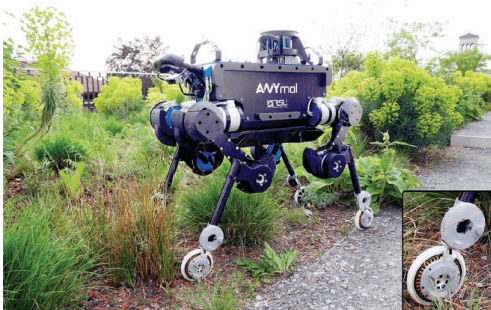




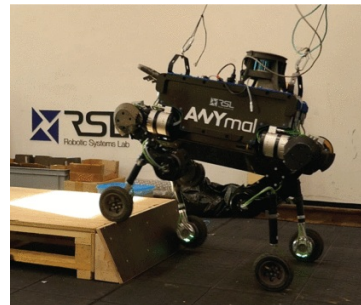
(a) ANYmal にアイススケータを用いたスイングロコモーション研究 [22]



(b) QSkater-E (4脚ロボット) に無動力ローラを用いたスイングロコモーション研究 [23]



(c) 動力ホイール付き脚ロボットの階層型全身コントローラに関する研究 [24]



(d) 動力ホイール付き脚ロボットの軌道最適化に関する研究 [25]

図 2.3: 車輪付き脚ロボットの関連研究

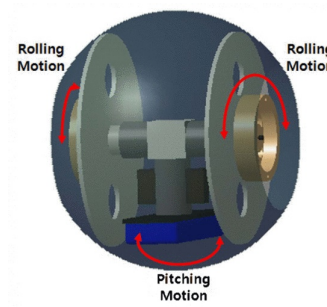


## 2.2 ローリングロボット

ローリングロボットは、ロボット全体が回転運動を行うことで移動するロボットを指す。これらのロボットは主に二つの研究分野で開発が進められている。一つ目は、非ホロノミクスシステムの研究対象としての全球ロボットであり、内部の重り位置制御を基に動作するものが挙げられる [26], [27] (図 2.4a)。二つ目は、バイオメカニクスロボットの分野であり、車輪蜘蛛 (Cebrennus Rechenbergi) (図 2.4e) に着想を得た 4 脚ロボット Scorpio[28] (図 2.4c) や、6 脚ロボット [29] (図 2.4d) がある。これらのロボットは、内部の重心移動や外殻の変形を利用してローリングを実現しているが、地面を蹴る能力は備えていない。そのため、これらのロボットは平面での移動に限定されている。



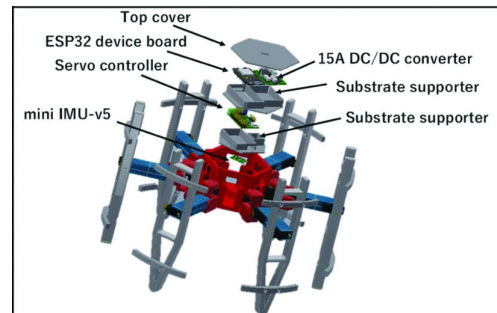
(a) 重心移動による姿勢制御研究用の全球ロボット [26]



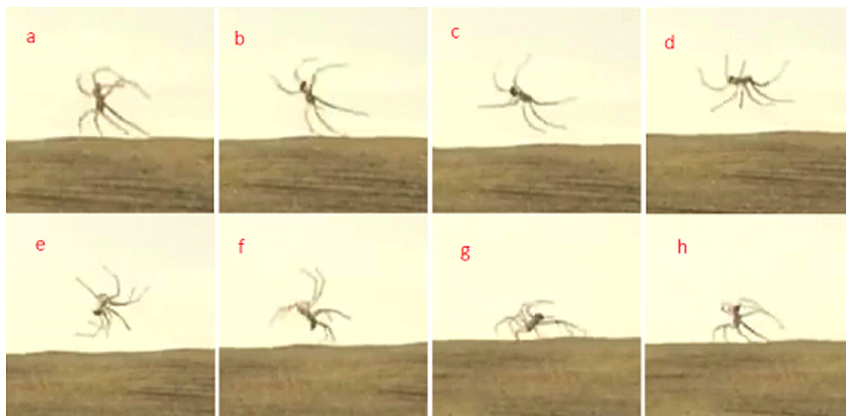
(b) 全球ロボットの内部構造、この関連研究では3自由度の錘を制御する [26]



(c) Scorpio[28]、車輪蜘蛛から着想を得たバイオニクスロボット



(d) Scorpio を6脚に拡張した派生研究 [29]



(e) 車輪蜘蛛のローリングモーション [28]

図 2.4: ローリング走行の関連研究

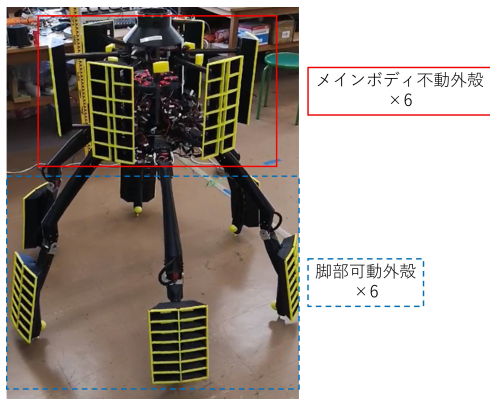
## 第3章 Rollopod-A

本章では、筆者らが開発している Rollopod-A ロボットについて、機械設計と電気設計について述べる。

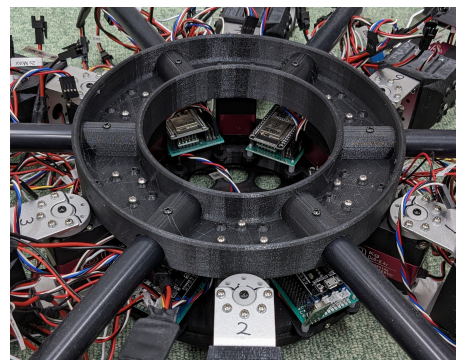
### 3.1 機械設計

Rollopod-A (図 1.1) は、脚を円周上に等間隔で配置した6脚ロボット (Hexapod) である。軽量化と製造性を考慮し、主要部品には 3D プリンタを用いてポリ乳酸 (PLA) を採用し、重量は 15kg に抑えられている。メインボディはリングプレート構造を採用しており、RC サーボモータを挟み込んだベースフレーム (図 3.1b) を中心に構成されている。このベースフレームには、3 層構造のバッテリーユニット (図 3.1c) と、カメラ・コントローラユニット (図 3.1d) が搭載されている。さらに、ローリング走行時の耐衝撃性を考慮し、外殻を等間隔に配置した。この外殻はメインボディに取り付けられた 6 枚と、各脚第 3 アームに装着された 6 枚の計 12 枚で構成され (図 3.1a)、ローリング走行時には直径 0.7m、幅 0.3m の円柱形状を形成する。

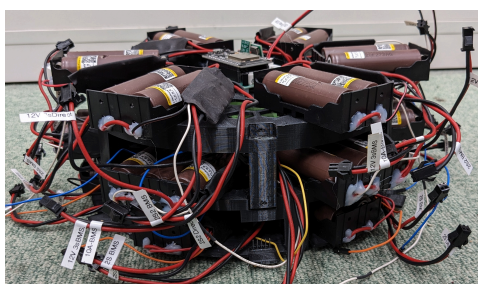
Rollopod-A の脚部 (図 3.1e) は、Leg1、Leg2、Leg3 の 3 つのアームからなる 3 自由度マニピュレータである。脚の全長は 0.95m と、他の 6 脚ロボットと比較して特徴的に長く設計されている。この設計は、低重力下での効率的な多脚歩行を実現するため、深海に生息するタカアシガニに着想を得たものである。



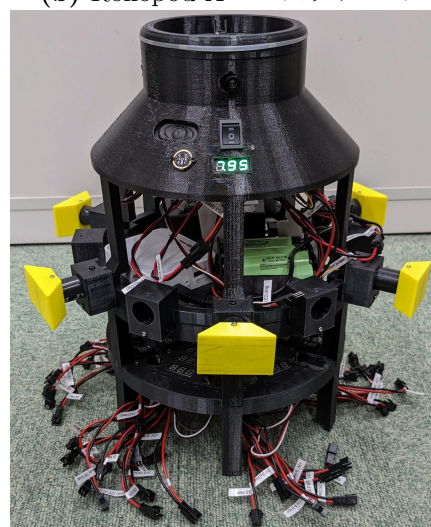
(a) Rollopod-A の外殻構成図



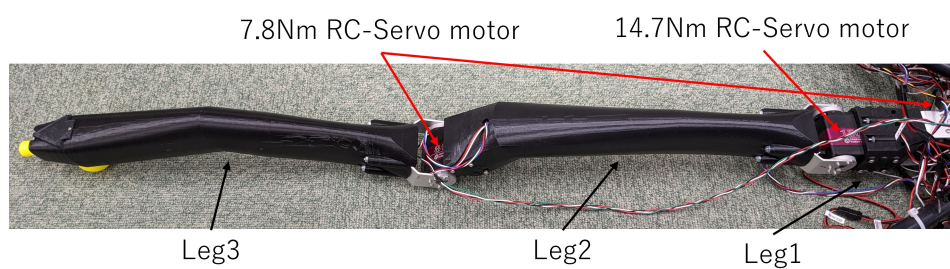
(b) Rollopod-A ベースフレーム



(c) バッテリーユニット、各層には軽量化のためハニカム構造を用いている



(d) カメラ・コントローラユニット



(e) Rollopod-A の脚

図 3.1: Rollopod-A 機体コンポーネント

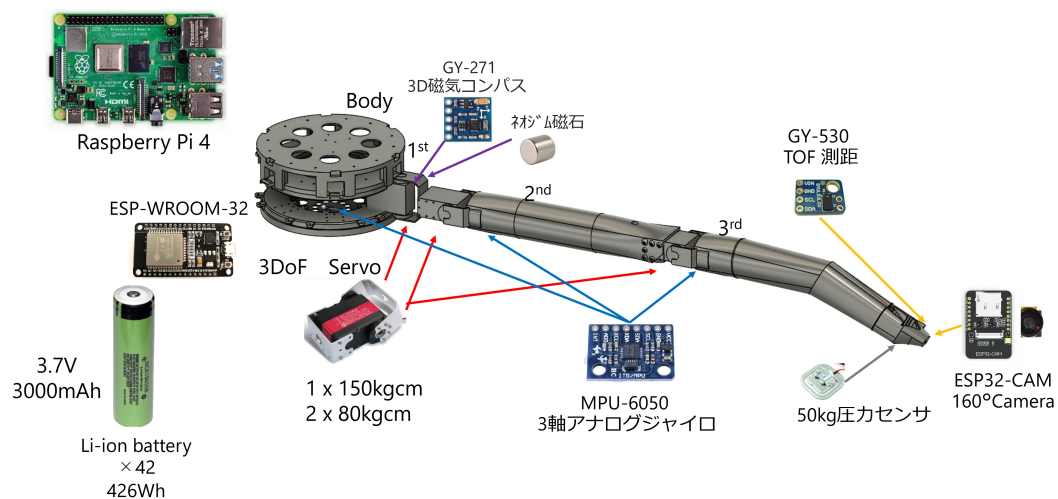
## 3.2 電気設計

Rollopod-A は、1 脚当たり 3 つ、計 18 個の位置制御 RC サーボモータで駆動される。駆動用バッテリーには 18650 リチウムイオンバッテリーを 42 本使用しており、合計 426Wh のエネルギー容量を有している。各脚および胴体下部には、センサデータの集計と RC サーボモータの PWM 制御を行うために ESP32 マイコンが搭載されている（図 3.2a）。これらの ESP32 マイコンは、USB 通信または Wi-Fi 5（IEEE 802.11ac）無線通信を介してメインコンピュータである Raspberry Pi 4（RP4）に接続されている（図 3.2b）。

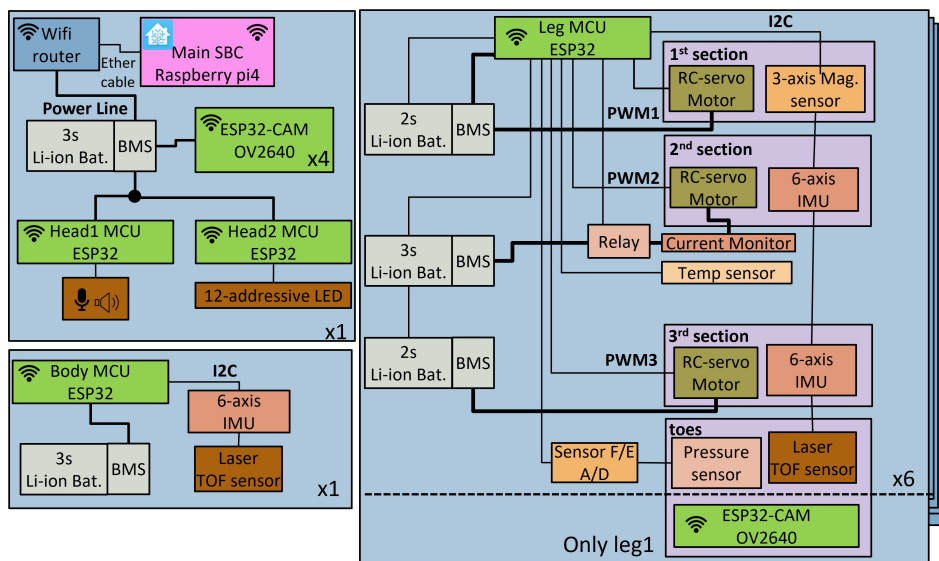
この機能を実現するために、筆者らは独自に ESP32 拡張ボードを設計および製作した（図 3.3）。この拡張ボードには、ESP32 を駆動するための降圧コンバータ、脚先圧力センサを接続するためのホイートストンブリッジ用アンプ、さらにセンサ入力および I2C 通信に対応した I/O 端子が備えられている。電子デバイスの構成に関する詳細は表 3.1 に示す。

表 3.1: 電子デバイス構成

| デバイス             | 型番                            |
|------------------|-------------------------------|
| 7.8Nm RC サーボモータ  | RDS5180 $\times$ 12           |
| 14.7Nm RC サーボモータ | RDS51150 $\times$ 6           |
| バッテリー            | Li-ion(18560) 426Wh           |
| メインコンピュータ        | Raspberry Pi 4 $\times$ 1     |
| マイコン             | ESP32 $\times$ 8              |
| カメラ              | ESP32-CAM(FoV.160) $\times$ 9 |
| IMU センサ          | MPU-6050 $\times$ 13          |
| ToF 距離センサ        | GY-530 $\times$ 7             |
| 荷重計              | ひずみゲージ (max: 50kg) $\times$ 6 |
| 磁気コンパス           | GY-271 $\times$ 6             |



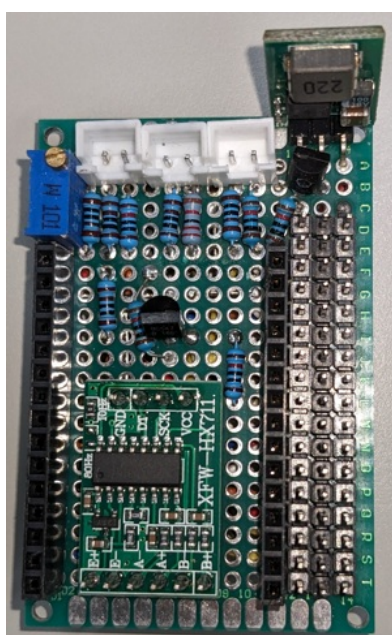
(a) Rollopod-A 1 脚当たりのデバイス構成図



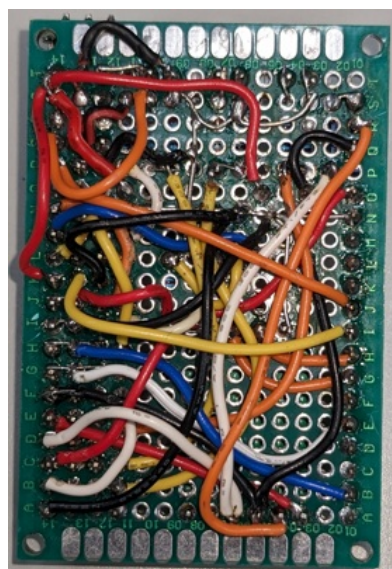
(b) Rollopod-A システム構成図

図 3.2: Rollopod-A 電気設計





(a) オリジナル拡張ボード（表）



(b) オリジナル拡張ボード（裏）

図 3.3: ESP32 用オリジナル拡張ボード

## 第4章 ローリングによる CoT 削減 コモーション（提案手法）

本章では、Rollopod-A がローリング走行を実現することで CoT を削減する手法について詳述する。

### 4.1 ローリングメカニズム

本節では、Rollopod-A のローリングメカニズムについて説明する。Rollopod-A は、適切な脚を放射方向に蹴り出すことでロボット本体に回転トルクを発生させる仕組みを持つ。ローリング時のロボット各コンポーネントの位置関係を図 4.1 に示す。加速時における力の関係は図 4.1 の左図に示されているようにモデル化される。脚の接地点を  $O_W$ 、ロボットの接地点に滑りが無いと仮定した場合、脚の蹴り出しにより発生する力  $F_0$  はメインボディと脚の接合点  $B$  で  $F_1$  と  $F_2$  に分解される。力  $F_1$  は内円の半径  $r_1$  を用いてトルク  $T_1 = F_1 \times r_1$  に変換される。残りの力  $F_2$  はロボットの斜め上方向に作用するが、自重と地面の摩擦力  $F_f$  および外円の半径  $r_2$  によりトルク  $T_2 = (F_f - F_1) \times r_2$  に変換される。この関係は逆回転時にも同様であり、力  $F_0$  を適切にコントロールすることで減速を行うことが可能である。このように、Rollopod-A は重心移動によるローリングを行う従来のローリングロボットとは異なり、ロボット自体に直接回転トルクを発生させる設計となっている。そのため、ローリングロボットでありながら登坂能力を有する点が特徴的である。

ローリングを制御する上で重要な脚の蹴り出し長さ  $l$  および第 1RC サーボモータの角度  $\theta_2$  は、以下の手法を用いて算出される。まず、脚とメインボディの接合点  $B$  のローカル座標を式 4.1 を用いて算出する。次に、脚が地面に対して垂直な時点からのロボット回転角度  $\theta_1$  を、IMU センサの値から式 4.3 を用いて求める。ここで、 $\omega_0$  は現在の角速度、 $t$  はシミュレーションフレーム間隔（センサ動作周波数）、 $\alpha$  は現在の角加速度を示す。さらに、ローカル座標原点（脚の接地点  $O_W$  を原点とする座標）を式 4.2 により算出する。これらの情報を基に、ローカル座標系の点  $B$  の座標とローカル座標原点を同次座標変換式 4.4 を用いて変換し、点  $B$  のワールド座標を求める。これにより、脚の接地点  $O_W$  と脚のメインボディ接合点  $B$  の位置関係が明確になるため、脚の長さ  $l$  および角度  $\theta_2$  を算出することが可能となる。



$$\begin{pmatrix} x_L \\ y_L \end{pmatrix} = \begin{pmatrix} x \cos \theta_1 - y \sin \theta_1 \\ x \sin \theta_1 + y \cos \theta_1 \end{pmatrix} \quad (4.1)$$

$$(O_{LxW}, O_{LyW}) = (2\pi r_2 \frac{\theta_1}{2\pi}, -r_2) \quad (4.2)$$

$$\theta_1 = \omega_0 t + \frac{1}{2} \alpha t^2 \quad (4.3)$$

$$\begin{pmatrix} x_W \\ y_W \\ z_W \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_L \\ y_L \\ z_L \\ 1 \end{pmatrix} + \begin{pmatrix} O_{LxW} \\ y_{LyW} \\ z_{LzW} \\ 1 \end{pmatrix} \quad (4.4)$$

最後に、脚の各関節角度は以下のように求められる。第1関節の角度  $\theta_{L1}$  は  $\theta_2$  に等しく、残りの第2関節  $\theta_{L2}$  および第3関節  $\theta_{L3}$  は、アーム2を斜辺、地面までの距離を底辺とした直角三角形の幾何学的関係に基づき、三角関数を用いて計算される。

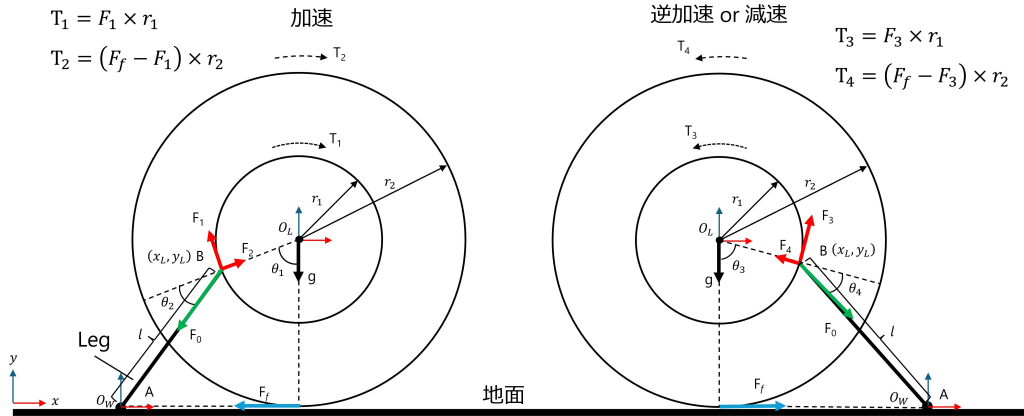


図 4.1: Rollopod-A の蹴り出し動作によるローリングメカニズム図

## 第5章 シミュレータを用いたローリング走行モードのCoT評価

本研究では、提案するローリング走行モードにおける CoT 削減効果を検証するため、シミュレーション実験を実施した。本章では、このシミュレーション実験に用いた環境と手法について述べる。第一節では、実験に用いたシミュレータとシステム構成について説明する。第二節では、Rollopod-A の制御用ポリシーに関する詳細を述べる。第三節では、CoT 測定実験環境について記載する。

### 5.1 実験環境

シミュレーション実験に用いたサーバのスペックは表 5.1 に示す。本研究では、NVIDIA 社が提供するシミュレーションプラットフォームである NVIDIA Isaac Sim と Isaac Lab を用いて実験を行った。以下に、これらのシミュレータとその特性について詳述する。

表 5.1: シミュレーションサーバスペック

| コンポーネント  | 型番                 |
|----------|--------------------|
| CPU      | AMD Ryzen 7 2700   |
| DRAM     | DDR4 64GB          |
| GPU      | RTX A4000 16GB     |
| OS       | Ubuntu 22.04.5 LTS |
| CUDA     | 12.2               |
| GPU ドライバ | 535.183.01         |

#### 5.1.1 NVIDIA Isaac Sim

NVIDIA Isaac Sim[30] は、NVIDIA 社によって開発された 3D シミュレーションプラットフォームである。このシミュレータは、GPU ベースの物理エンジンである PhysX を採用しており、大規模なマルチセンサシミュレーションや強化学習に適している。Universal Scene Description (USD) 形式による 3D シーン記述にも

対応しており、産業規模のフォトリアルなデジタルツインシミュレーションが可能である。

- PhysX 物理エンジン

PhysX は、有限要素法 (FEM) を用いたソフトボディシミュレーション、クロス、パーティクル、流体のシミュレーションをサポートする統一されたソルバーフレームワークを提供する。また、剛体システムのジャイロ効果や運動量保存を含むアーティキュレーションシステムの安定的なシミュレーションを実現している。これにより、複雑な衝突検出や関節の挙動を正確に再現できる。PhysX がサポートしているシミュレーション機能を表 5.2 に示す。

表 5.2: PhysX 物理エンジンの機能

| 機能              | 動作                                        |
|-----------------|-------------------------------------------|
| 剛体ダイナミクス        | 外力が加わる環境下でのオブジェクト相互作用シミュレーション             |
| シーンクエリ          | シミュレーション内オブジェクトの重複や衝突、レイキャスト検出機能          |
| ジョイント           | オブジェクトの相対的な動きを制約する                        |
| 縮約座標アーティキュレーション | 剛体のツリー構造において、線形時間で計算可能で関節誤差が発生しないシミュレーション |
| 車両ダイナミクス        | 外力が加わる環境下でのオブジェクト相互作用シミュレーション             |
| キャラクタコントローラ     | アクタ用の運動学に準じたコントローラ                        |
| 弾性変形体ダイナミクス     | 有限要素法 (FEM) を用いた弾性変形体のシミュレーション            |
| SDF コライダー       | 符号付き距離場に基づく衝突表現により、非凸形状を凸分解なしでシミュレートする    |
| 位置ベースダイナミクス     | 液体、粒状材料、布、剛体、変形可能な物体などをシミュレートする           |
| カスタムジオメトリ       | ボックスや球体以外の衝突を定義する                         |
| 破壊 & 崩壊         | オブジェクトの破壊や崩壊のシミュレーション                     |
| 煙 & 炎           | 煙や炎などの流体シミュレーション                          |

- 超並列化シミュレーション

Isaac Sim では、NVIDIA GPU と PyTorch を活用した超並列演算により、多数のアクタを同時にシミュレートすることが可能である。これにより、強化学習におけるサンプリング効率が大幅に向上し、学習時間の短縮が実現さ

れる。MuJoCoなどのCPUベースのシミュレータと比較して、高速なシミュレーションが可能である点が大きな特徴である。

- Ray Caster センサ

Isaac Sim および Isaac Lab は NVIDIA の Real-Time Ray Tracing を応用した Ray Caster センサ（RGB カメラ、RGB-D カメラ、LiDAR、ToF センサ）に対応している。Real-Time Ray Tracing は光の経路を反射を含め追跡できる機能だ。一般的には、3DCG における光沢や反射面、光の拡散を正確に再現するのに用いられる。この Real-Time Ray Tracing はその機能上光が物体に到達した場所を特定でき、これを光学センサシミュレートに応用したのが Ray Caster センサである。

### 5.1.2 NVIDIA Isaac Lab

NVIDIA Isaac Lab[31] は、Isaac Sim を基盤としたロボット学習用の統合モジュールフレームワークである。Isaac Sim の高いモジュール性を活用しており、独自機能の拡張や学習環境の統合が容易に行える。さらに、GPU による超並列環境学習をサポートしており、強化学習の効率を飛躍的に向上させる。本研究では、Rollopod-A のポリシーコントローラの学習に Isaac Lab を使用した。

## 5.2 コントローラポリシ学習環境 (NVIDIA Isaac Lab)

本節では、Rollopod-A のコントローラポリシの学習環境について詳述する。

### 5.2.1 Rollopod-A シミュレーションモデル

Rollopod-A のシミュレーションモデル（図 5.1）は、USD 形式で記述されており、機体は剛体モデルとして設計されている。このため、モデルには湾曲や破断が発生しない仕様となっている。地形との相互作用においては、静止摩擦係数が 1.0 ～ 0.5 [32] の範囲からランダムにサンプリングされ、動摩擦係数は静止摩擦係数よりも低い値でランダムにサンプリングされる。ドメインランダム化について表 5.3 に示す。

アクチュエータモデルには、Isaac Lab が標準搭載している二重比例コントローラ（DP コントローラ）が採用されている。このコントローラは関節角度誤差（Stiffness）および関節速度（Damping）の最小化を目指すものである（式 5.1）。本研究では、シミュレーション中の制御パラメータとして  $Stiffness = 10$ 、 $Damping = 1$  を設定している。

$$\tau = stiffness * (q - q_{target}) + damping * (\dot{q} - \dot{q}_{target}) \quad (5.1)$$

表 5.3: ポリシ学習におけるドメインランダム化とサンプリング範囲

| パラメータ   | 対象ポリシ    | サンプリング範囲                                                                                                                    |
|---------|----------|-----------------------------------------------------------------------------------------------------------------------------|
| 指示コマンド  | 歩行       | $-0.6 \leq lin\_vel\_x \leq 0.6,$<br>$-0.6 \leq lin\_vel\_y \leq 0.6,$<br>$ang\_vel\_z = 0.0.$                              |
|         | ローリング    | $-1.0 \leq lin\_vel\_x \leq 1,$<br>$-1.0 \leq lin\_vel\_y \leq 1.0,$<br>$0.0 \leq ang\_vel\_z \leq 2.5.$                    |
| 摩擦係数    | 歩行・ローリング | $0.5 \leq 1.0.$                                                                                                             |
| 重量 (kg) | 歩行・ローリング | $-1.0 \leq 5.0.$                                                                                                            |
| スポーン位置  | 多脚歩行     | $-0.5 \leq x \leq 0.5, -0.5 \leq y \leq 0.5, z = 0.55, -3.14 \leq yaw \leq 3.14.$                                           |
|         | ローリング    | $-0.5 \leq x \leq 0.5, -0.5 \leq y \leq 0.5, z = 0.55, pitch = 1.571, -3.14 \leq roll \leq 3.14, -3.14 \leq yaw \leq 3.14.$ |
| 外乱      | 歩行       | $-0.5 \leq x \leq 0.5, -0.5 \leq y \leq 0.5.$                                                                               |
|         | ローリング    | $0.0 \leq yaw \leq 2.5.$                                                                                                    |

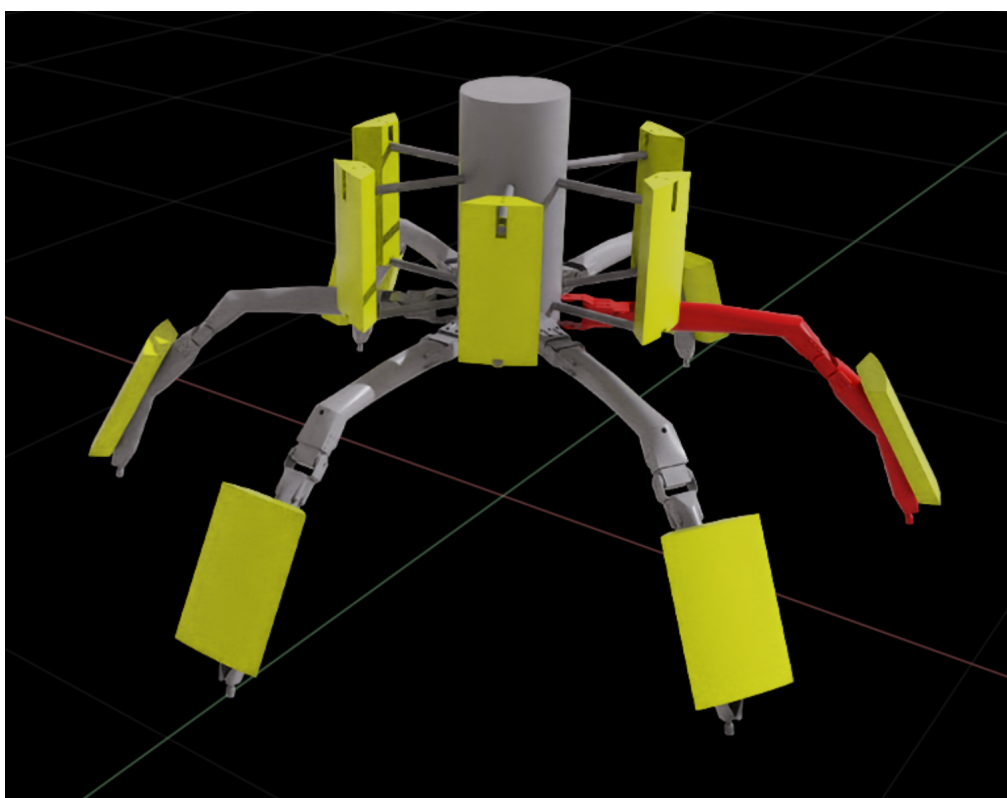


図 5.1: Rollopod-A の USD 記述されたシミュレーションモデル、シミュレーション計算コスト軽減のため簡略化

### 5.2.2 深層強化学習アルゴリズム (PPO)

本研究では、ポリシーコントローラの状態遷移モデルにマルコフ決定過程 (MDP) による方策勾配法を用いる。また、学習アルゴリズムとして Proximal Policy Optimization (PPO) [33] を採用する。PPO は、現在のデータを用いてポリシーを効率的に改善しつつ、過剰な変更による性能劣化を防ぐことを目的とした手法である。従来の Trust Region Policy Optimization (TRPO)[34] が複雑な 2 次の最適化手法を採用しているのに対し、PPO はより簡易な 1 次の手法を用いて安定性と効率性を両立している。具体的には、新旧ポリシー間の類似性を制約することで、過大な更新を抑制しながら安定した学習を実現する。これにより、PPO は実装が容易でありながら、TRPO と同等以上の性能を達成できることが実証されている。

PPO には PPO-Penalty と PPO-Clip の 2 種類存在するが、本研究では PPO-Clip を採用している。

- PPO-Penalty

PPO-Penalty は、TRPO のように KL 制約付き更新をおおよそ解く手法であるが、KL ダイバージェンスを固定制約として扱うのではなく、目的関数にペナルティ項として導入する点が異なる。また、学習の過程でペナルティ係数を動的に調整することで、適切なスケールに調整されるよう設計されている。

- PPO-Clip

PPO-Clip は、目的関数に KL ダイバージェンス項を含まず、また制約も一切設けていない。その代わり、目的関数における特殊なクリッピングを利用することで、新しいポリシーが古いポリシーから大きく乖離することを抑制している。PPO-Clip における目的関数式 5.2 [33] を示す。確率比が指定された範囲  $(1 - \epsilon, 1 + \epsilon)$  を超えた場合、その改善の影響を無視して悪化の影響のみを反映する。この手法により、目的関数はクリッピングされていない場合の下界として機能し、ポリシー更新の安定性を確保する。また、このクリッピング処理は初期状態では従来の目的関数と一致するがパラメータが更新されるにつれて異なる挙動を示す。

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[ \min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t) \right] \quad (5.2)$$

### 5.2.3 学習地形モデル

Rollopod-A のコントローラポリシーの学習には、Isaac Lab のプロシージャル地形生成機能を使用した地形環境を採用している。各地形は  $10m^2$  であり、 $10 \times 20$  マス生成される。各地形の生成確率は高周波ウェーブ地形 (20%)、低周波ウェーブ地形 (20%)、上り勾配地形 (20%)、下り勾配地形 (10%) である。さらに、地

形カリキュラムを導入することで、適応的な学習環境を構築している。具体的には、目標移動距離の50%に到達しない場合は地形の難易度を下げ、50%を超えた場合は難易度を上げる仕組みである。

しかし、現状の多脚歩行コントロールポリシおよびローリング走行コントロールポリシは、最大難易度の地形での学習には至っていない。この制限は、地形難易度の上昇に伴いアクタの移動速度が低下するためである。この仕様は今後の研究で改善を予定している。学習に用いる地形の詳細を以下に示す。

- ウェーブ地形

ウェーブ地形（図 5.2a, 図 5.2b）は、アクタに対して変化の激しい地形や緩やかな地形での安定した歩行およびローリング走行を学習させる目的で使用される。この地形は、各移動モードにおいて共通して用いられるが、学習条件として設定される地形の最大振幅が異なる。ウェーブ地形における具体的なパラメータは表 5.4 に示されている。

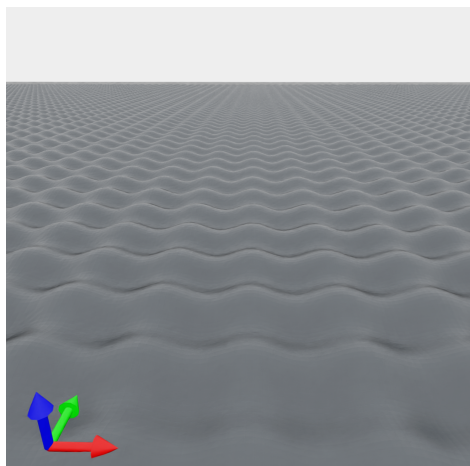
- 勾配地形

勾配地形（図 5.2c, 図 5.2d）は、アクタが上り坂および下り坂の環境において安定した歩行およびローリング走行を学習するために使用される。地形の詳細なパラメータについては表 5.4 に示されている。

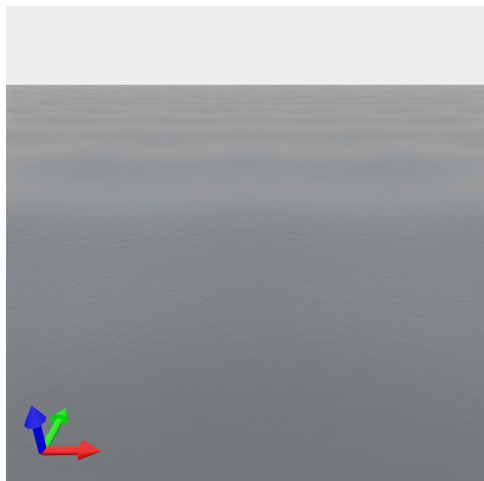
表 5.4: ポリシ学習地形パラメータ

| 地形        | 学習ポリシ    | パラメータ                              |
|-----------|----------|------------------------------------|
| 高周波ウェーブ地形 | 歩行       | 周波数: $6/m$ , 振幅: $0.0m \sim 0.2m$  |
|           | ローリング    | 周波数: $6/m$ , 振幅: $0.0m \sim 0.15m$ |
| 低周波ウェーブ地形 | 歩行       | 周波数: $1/m$ , 振幅: $0.0m \sim 0.2m$  |
|           | ローリング    | 周波数: $1/m$ , 振幅: $0.0m \sim 0.15m$ |
| 上り勾配地形    | 歩行・ローリング | 傾斜度: $0.0\% \sim 20.0\%$           |
| 下り勾配地形    | 歩行・ローリング |                                    |

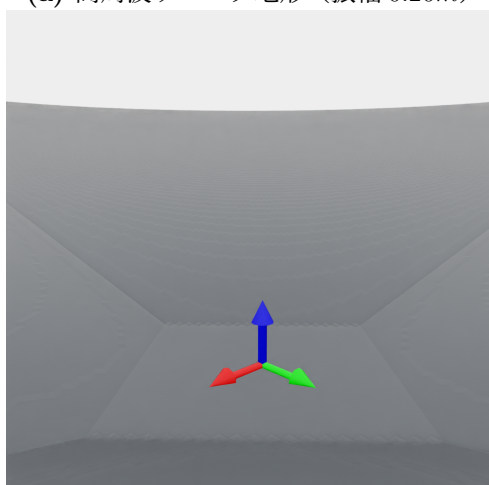




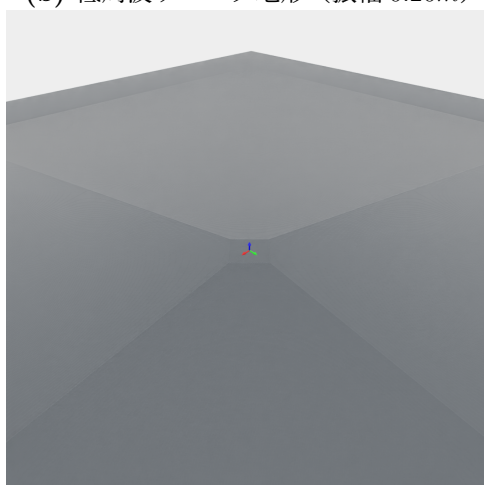
(a) 高周波ウェーブ地形（振幅  $0.20m$ ）



(b) 低周波ウェーブ地形（振幅  $0.20m$ ）



(c) 上り勾配地形（傾斜度  $20\%$ ）



(d) 下り勾配地形（傾斜度  $20\%$ ）

図 5.2: 学習に用いられる地形例

本シミュレーション実験では、シミュレータに起因するアクタと地形間の不適切な干渉によるスタックが多数発生した。図 5.4 に示すのは、多脚歩行モードにおける下り勾配地形（傾斜度 10%）での不適切な干渉による転倒例である。この例では、 $t = 7.0s$  で脚が地形を貫通し、その地点で拘束されることで転倒が生じている。一方、ローリング走行モードでは、このような不適切な干渉は目視観測されなかった。また、アクタのスポン時に類似の不適切な干渉が観測されており（図 5.3）、この現象はローリング走行モードにおいても発生している。本シミュレーションにおいては、アクタのロコモーションに起因する転倒と不適切な干渉による転倒を区別することができない。そのため、以降に提示するデータには、不適切干渉に起因する転倒の影響が含まれている点に留意されたい。

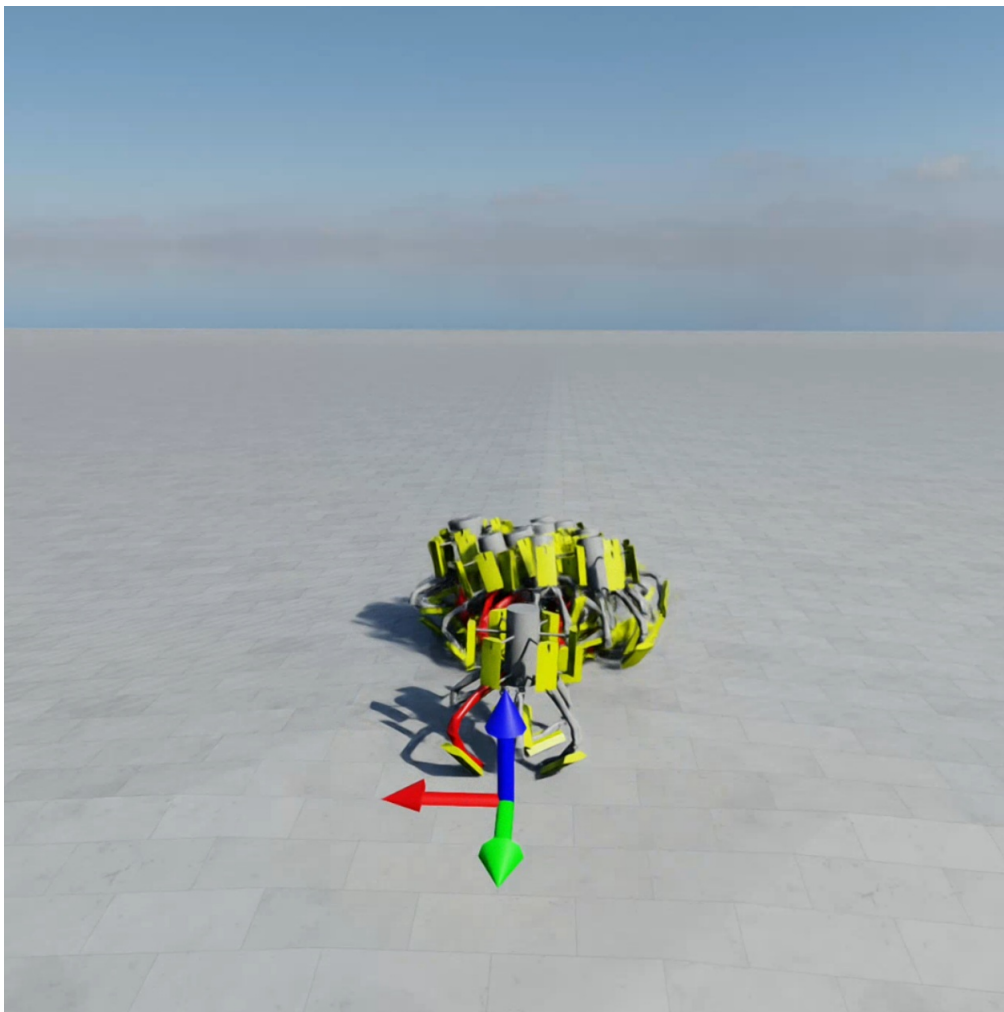
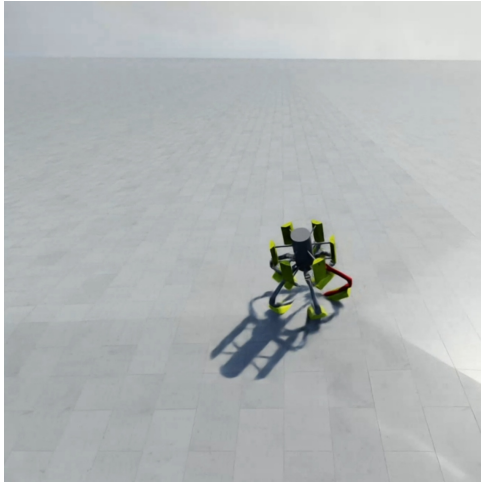
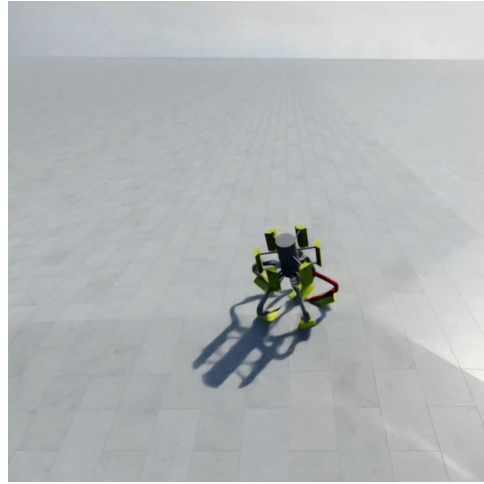


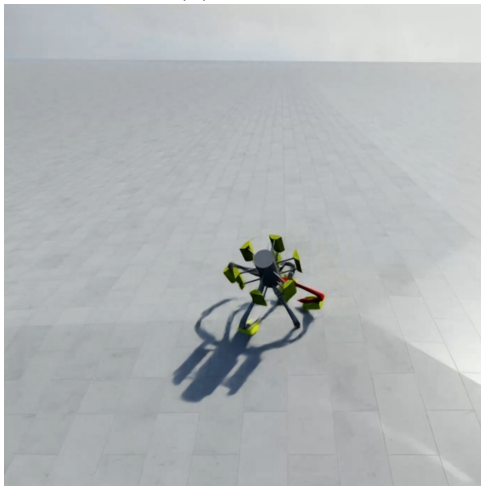
図 5.3: 多脚歩行モードにおいて発生した地面との不適切干渉によるスタック例 (1)、1 体のアクタがスポン直後に地形と干渉してその場に拘束されている



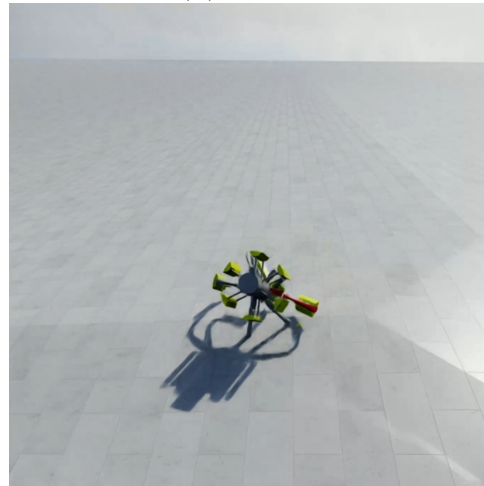
(a)  $t = 5.0\text{s}$



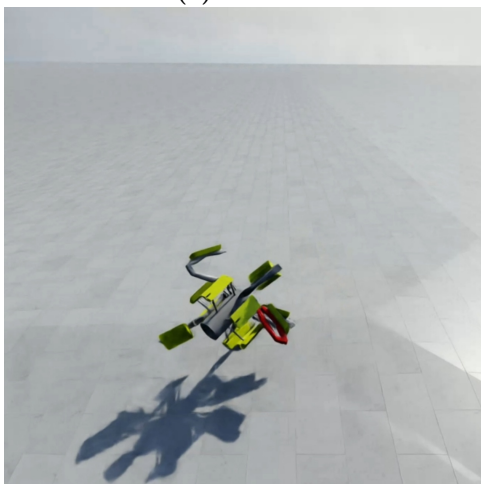
(b)  $t = 6.0\text{s}$



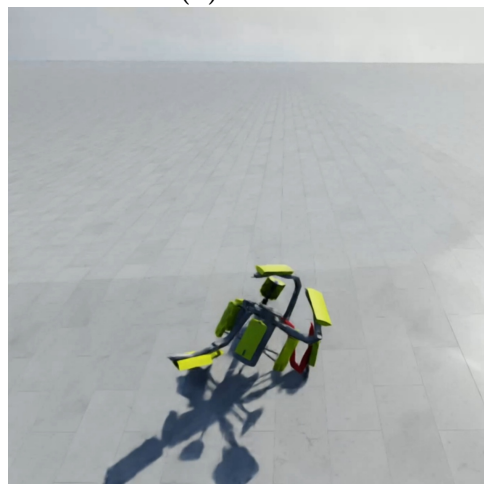
(c)  $t = 7.0\text{s}$



(d)  $t = 8.0\text{s}$



(e)  $t = 9.0\text{s}$



(f)  $t = 10.0\text{s}$

図 5.4: 多脚歩行モードにおいて発生した地面との不適切干渉によるスタック例 (2)

## 5.2.4 コントロールポリシ学習パラメータ・学習設定

本節では、Rollopod-A の制御に用いたコントロールポリシについて、ポリシ学習パラメータと学習設定をモード別に詳述する。両ポリシにおける共通事項としては、1 エピソード 20 秒、終了条件はタイムアウト、最大反復回数は 1500 回である。また、学習に使用されるライブラリは RSL-RL であり、PPO アルゴリズムを用いた。学習パラメータを表 5.5 に示す。

表 5.5: ポリシ学習パラメータ

| パラメータ                        | 値               |
|------------------------------|-----------------|
| num steps per env            | 24              |
| max iterations               | 1500            |
| save interval                | 50              |
| empirical normalization      | False           |
| init noise std               | 1.0             |
| actor hidden <sub>dims</sub> | [512, 256, 128] |
| critic hidden dims           | [512, 256, 128] |
| activation                   | "elu"           |
| value loss coef              | 1.0             |
| use clipped value loss       | True            |
| clip param                   | 0.2             |
| entropy coef                 | 0.005           |
| num learning epochs          | 5               |
| num mini batches             | 4               |
| learning rate                | 1.0e-3          |
| schedule                     | "adaptive"      |
| gamma                        | 0.99            |
| lam                          | 0.95            |
| desired kl                   | 0.01            |
| max grad norm                | 1.0             |

- 多脚歩行コントロールポリシ

多脚歩行コントロールポリシでは、xy 平面速度成分 ( $-1.0 \leq x \leq 1$ ,  $-1.0 \leq y \leq 1$ )、ヨー回転 ( $0 \leq yaw \leq \pi$ ) からランダムにサンプリングされたコマンドが入力される。多脚歩行ポリシの観測空間 (Observations) を表 5.6、多脚歩行ポリシ報酬設計 (Rewards) を表 5.7、報酬推移グラフを図 5.5, 5.6、コントローラアーキテクチャを図 5.7、平坦地形における多脚歩行タイムラプスを図 5.8 に示す。

表 5.6: 多脚歩行ポリシ観測空間 (Observations)

| 観測情報              | 補足                        |
|-------------------|---------------------------|
| base lin vel      | メインボディローカル座標の xyz 速度成分    |
| base lin vel w    | メインボディワールド座標の xyz 速度成分    |
| base ang vel      | メインボディローカル座標の xyz 角速度     |
| projected gravity | メインボディに作用する重力加速度 xyz 成分   |
| velocity commands | アクタに入力された指示コマンド           |
| joint pos         | 関節 18 個の角度                |
| joint vel         | 関節 18 個の角速度               |
| actions           | ポリシの出力値 (18 個)            |
| height scan       | マルチ ToF 代替センサの値 (地面までの距離) |

表 5.7: 多脚歩行ポリシ報酬設計 (Rewards)

| 報酬                                 | 算出式                         | スケール    | std           |
|------------------------------------|-----------------------------|---------|---------------|
| track lin vel xy exp               | 式 5.3.                      | 0.85    | $\sqrt{0.25}$ |
| track direction alignment exp      | 式 5.4.                      | 0.5     | $\sqrt{0.25}$ |
| track ang vel z exp                | 式 5.5.                      | 0.25    | $\sqrt{0.25}$ |
| lin vel z l2                       | $v_z^2$ .                   | -2.0    |               |
| ang vel xy l2                      | $\ \omega_{xy}\ ^2$ .       | -0.05   |               |
| dof torques l2                     | $\sum_{j \in J} \tau_j^2$ . | -1.0e-5 |               |
| dof acc l2                         | $\sum_{j \in J} a_j^2$ .    | -2.5e-7 |               |
| action rate l2                     | 式 5.6.                      | -0.01   |               |
| feet air time                      | 式 5.7.                      | 0.125   |               |
| undesired outershell contacts      |                             | -0.125  |               |
| undesired mainbody contacts        | 式 5.8.                      | -0.5    |               |
| undesired body outershell contacts |                             | -0.5    |               |
| z position err                     | 式 5.9.                      | -0.35   | $\sqrt{0.25}$ |
| flat orientation l2                | 5.10.                       | -0.25   |               |

std: 標準偏差

$$\exp\left(-\frac{(v_{\text{cmd},x} - v_{\text{act},x})^2 + (v_{\text{cmd},y} - v_{\text{act},y})^2}{\text{std}^2}\right) \quad (5.3)$$

$$\theta = \arccos\left(\text{clamp}\left(\frac{\mathbf{v}_{\text{cmd}} \cdot \mathbf{v}_{\text{act}}}{\|\mathbf{v}_{\text{cmd}}\| \|\mathbf{v}_{\text{act}}\|}, -1, 1\right)\right), \exp\left(-\frac{\theta}{\text{std}^2}\right) \quad (5.4)$$

$$\exp\left(-\frac{(\omega_{\text{cmd}} - \omega_{\text{act}})^2}{\text{std}^2}\right) \quad (5.5)$$

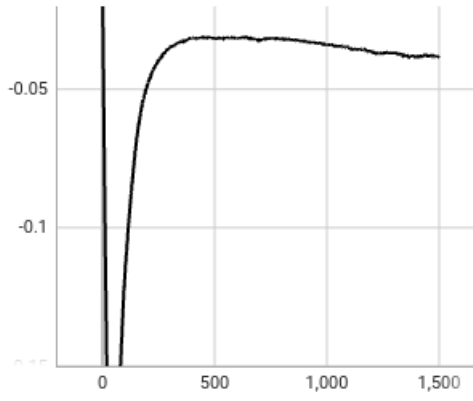
$$\sum_i (a_i^t - a_i^{(t-1)})^2 \quad (5.6)$$

$$\mathbf{1}_{\{\|\mathbf{v}_{\text{cmd}}(t)\| > 0.1\}} \sum_{i \in \text{Feet}} \left( \text{last\_air\_time}_i(t) - \text{threshold} \right) \cdot \text{first\_contact}_i(t) \quad (5.7)$$

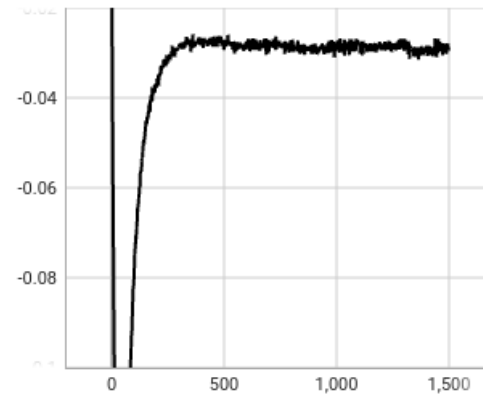
$$\sum_{b \in B} \mathbf{1}\left\{ \max_t \|\mathbf{f}_{n,t,b}\| > \text{threshold} \right\} \quad (5.8)$$

$$\begin{aligned} \text{height\_data}(n, r) &= \text{pos\_w}[n, 2] - \text{ray\_hits\_w}[n, r, 2] \\ \bar{h}(n) &= \frac{1}{R} \sum_{r=1}^R \text{height\_data}(n, r) \\ \begin{cases} 1 - \exp\left(-\frac{(\bar{h}(n) - \text{threshold})^2}{\text{std}^2}\right), & \text{if } \bar{h}(n) < \text{threshold}, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (5.9)$$

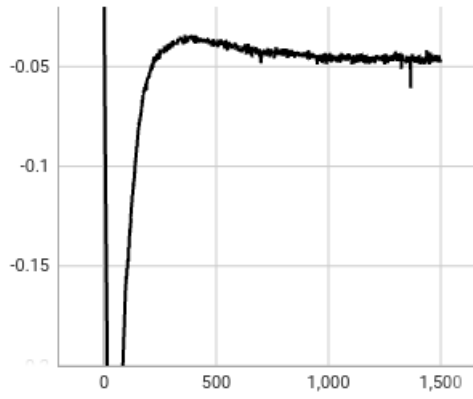
$$(g_{b,x}^n)^2 + (g_{b,y}^n)^2 \quad (5.10)$$



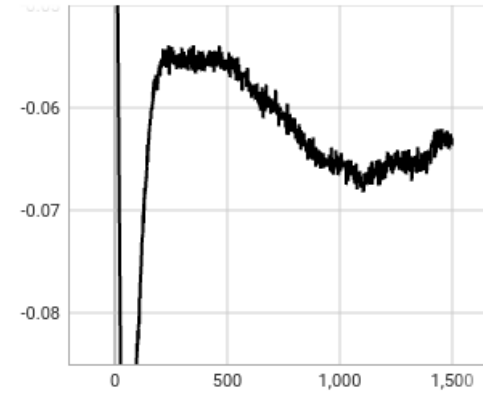
(a) action rate l2



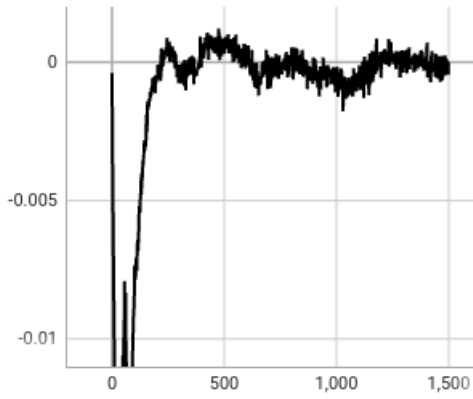
(b) ang val xy l2



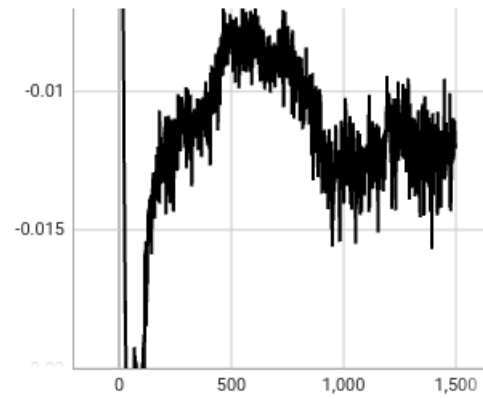
(c) dof acc l2



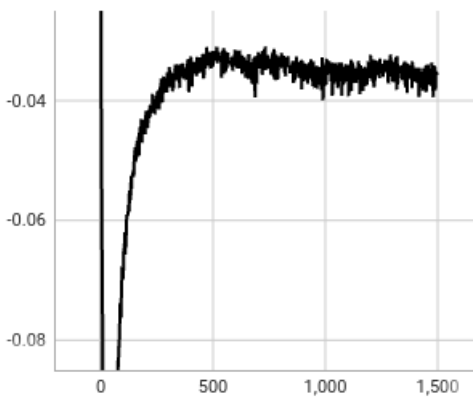
(d) dof torques l2



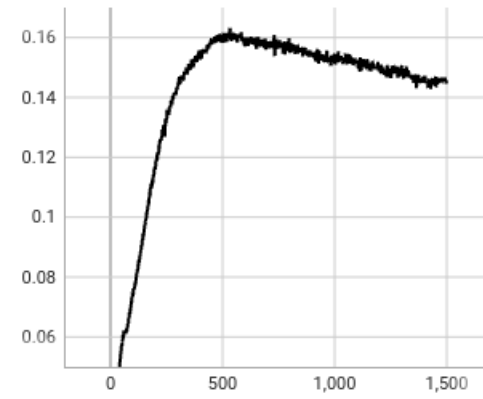
(e) feet air time



(f) flat orientation l2

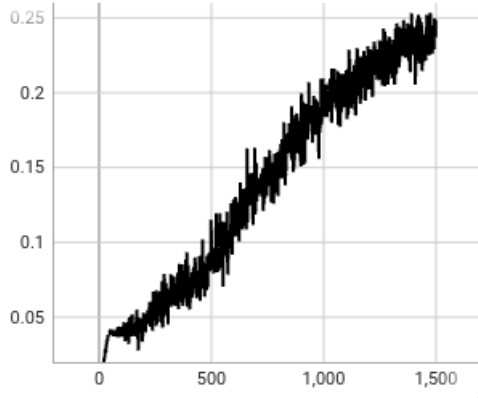


(g) lin vel z l2

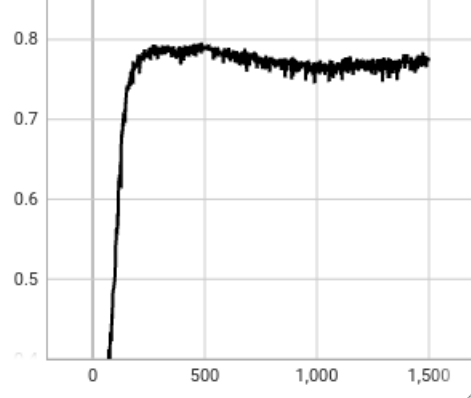


(h) track ang vel z exp

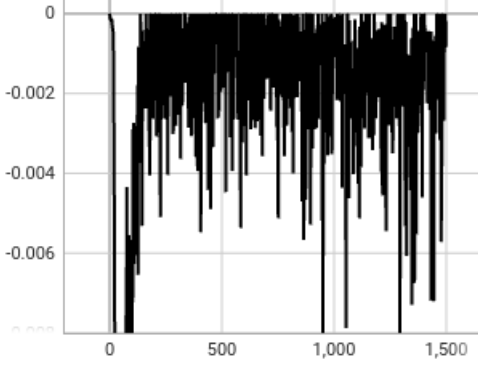
図 5.5: 多脚歩行ポリシの報酬推移グラフ (1)



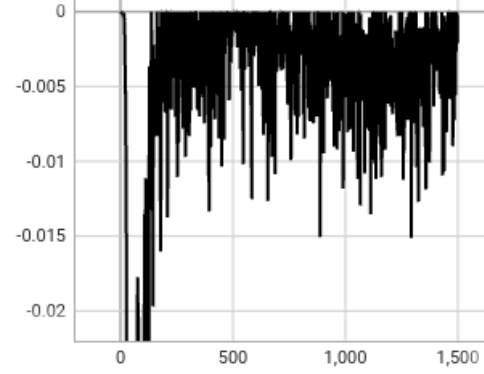
(a) track direction alignment exp



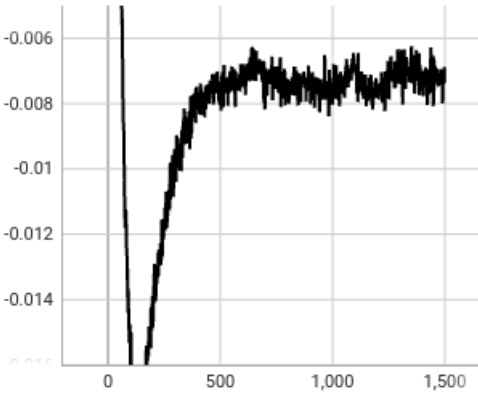
(b) track lin vel xy exp



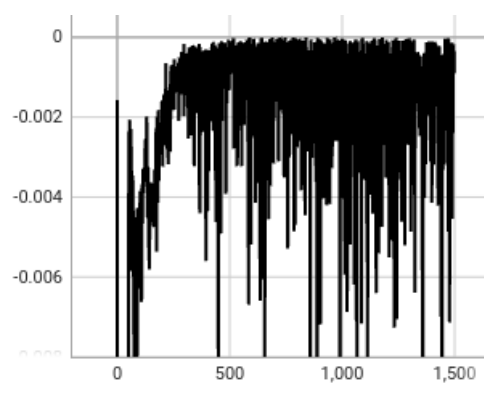
(c) undesired body outershell contacts



(d) undesired mainbody outershell contacts



(e) undesired outershell contacts



(f) z position err

図 5.6: 多脚歩行ポリシの報酬推移グラフ (2)



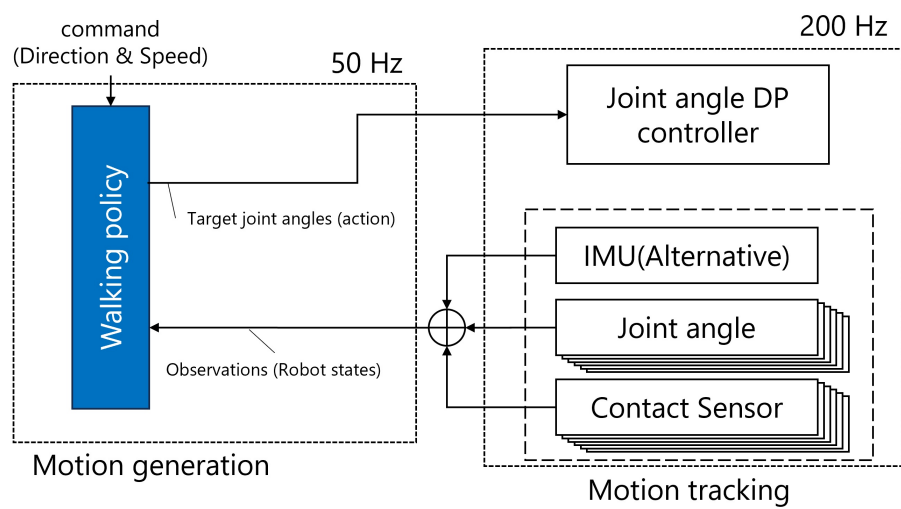
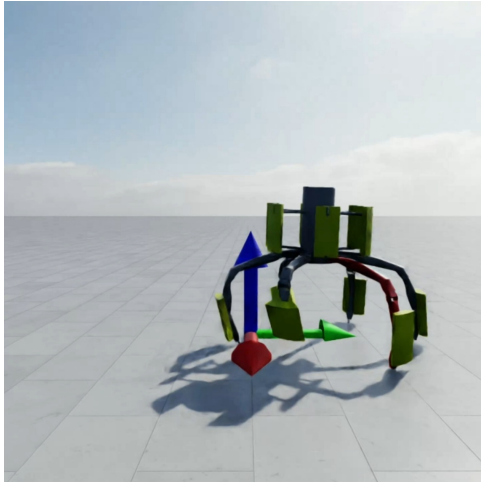
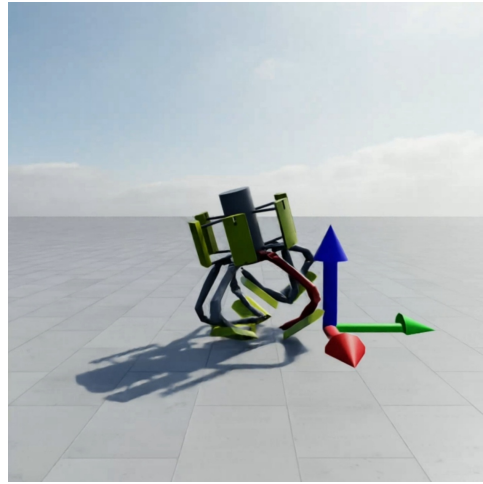


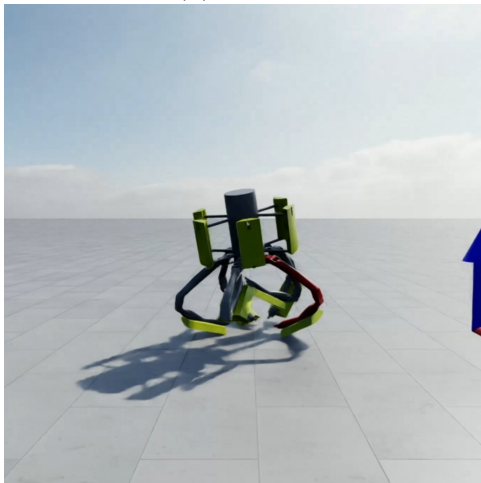
図 5.7: 多脚歩行コントローラーアーキテクチャ



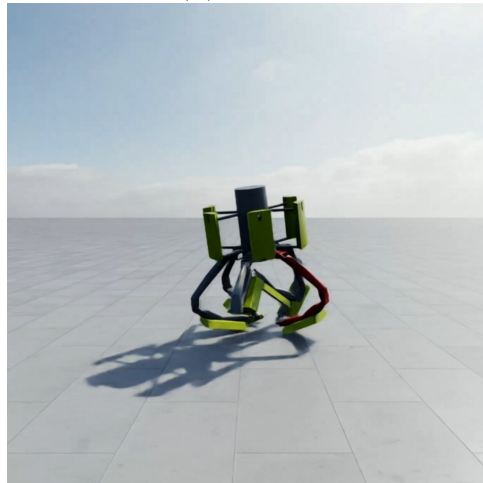
(a)  $t = 0.0\text{s}$



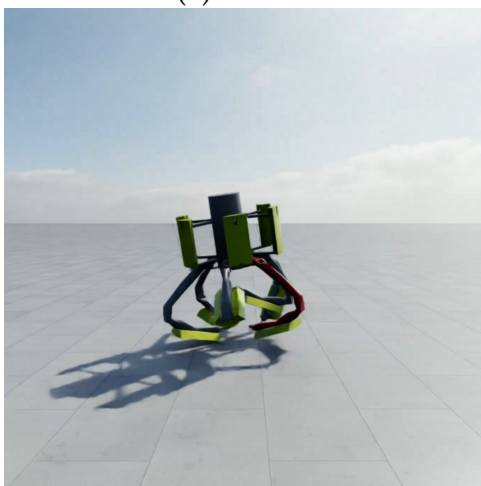
(b)  $t = 2.0\text{s}$



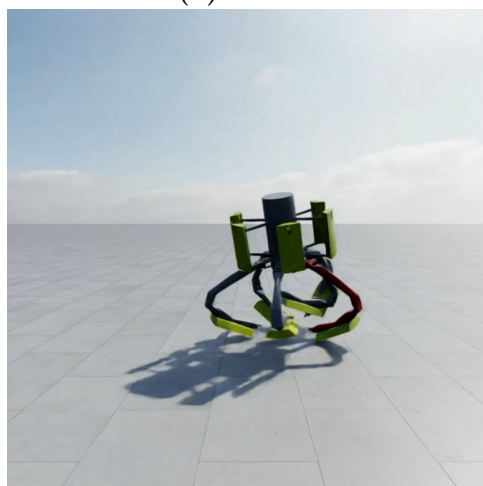
(c)  $t = 4.0\text{s}$



(d)  $t = 6.0\text{s}$



(e)  $t = 8.0\text{s}$



(f)  $t = 10.0\text{s}$

図 5.8: 平坦地形における多脚歩行モードのタイムラプス

- ローリング走行コントローラポリシー

多脚歩行コントロールポリシーでは、xy 平面速度成分 ( $-1.0 \leq x \leq 1, -1.0 \leq y \leq 1$ )、ヨー回転 ( $0 \leq yaw \leq \pi$ ) からランダムにサンプリングされたコマンドが入力される。ローリング走行ポリシーの観測空間 (Observations) を表 5.8、報酬設計 (Rewards) を表 5.9、報酬推移グラフを図 5.9、コントローラアーキテクチャを図 5.10 に示す。

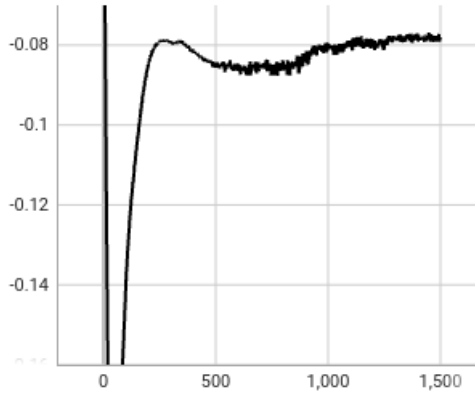
**表 5.8:** ローリング走行ポリシー観測空間 (Observations)

| 観測情報              | 補足                      |
|-------------------|-------------------------|
| base lin vel      | メインボディローカル座標の xyz 速度成分  |
| base lin vel w    | メインボディワールド座標の xyz 速度成分  |
| base ang vel      | メインボディローカル座標の xyz 角速度   |
| projected gravity | メインボディに作用する重力加速度 xyz 成分 |
| velocity commands | アクタに入力された指示コマンド         |
| joint pos         | 関節 18 個の角度              |
| joint vel         | 関節 18 個の角速度             |
| actions           | ポリシーの出力値 (18 個)         |

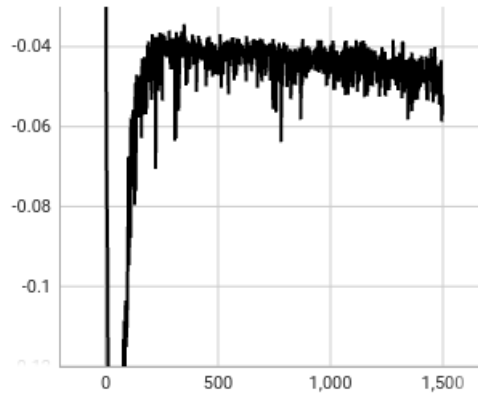
**表 5.9:** ローリング走行ポリシー報酬設計 (Rewards)

| 報酬                      | 算出式                              | スケール    | std           |
|-------------------------|----------------------------------|---------|---------------|
| track ang vel z exp     | 5.11.                            | 0.85    | $\sqrt{0.25}$ |
| dof torques l2          | $\sum_{j \in J} \tau_j^2$        | -1.0e-5 |               |
| dof acc l2              | $\sum_{j \in J} a_j^2$           | -2.5e-7 |               |
| action rate l2          | $\sum_i (a_i^t - a_i^{(t-1)})^2$ | -0.01   |               |
| vertical orientation l2 | $(g_{b,z}^{(n)})^2$              | -0.5    |               |
| std: 標準偏差               |                                  |         |               |

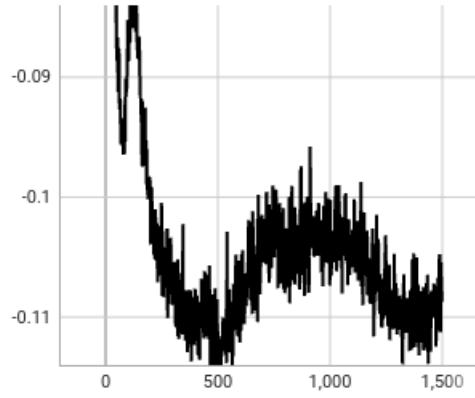
$$\begin{cases} \exp\left(-\frac{(\omega_{\text{cmd}} - \omega_{\text{act}})^2}{\text{std}^2}\right), & \text{if } \text{sign}(\omega_{\text{cmd}}) = \text{sign}(\omega_{\text{act}}), \\ 0, & \text{otherwise} \end{cases} \quad (5.11)$$



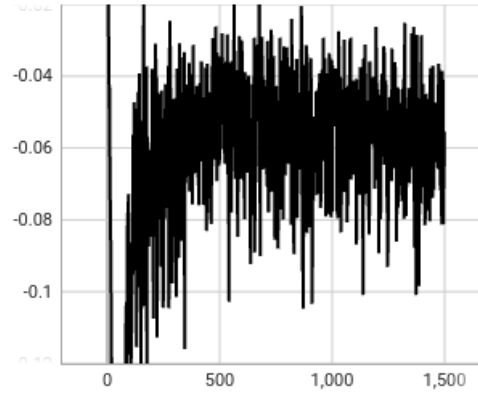
(a) action rate l2



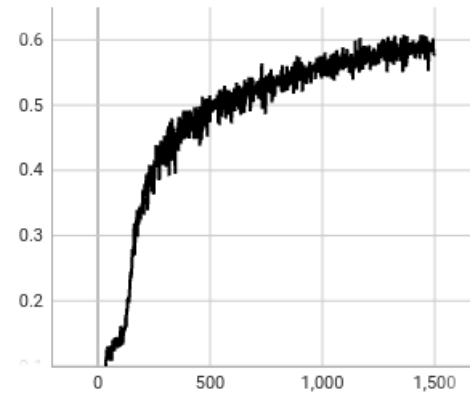
(b) dof acc l2



(c) dof torques l2



(d) erctical orientation l2



(e) track ang vel z exp

図 5.9: ローリング走行ポリシの報酬推移グラフ

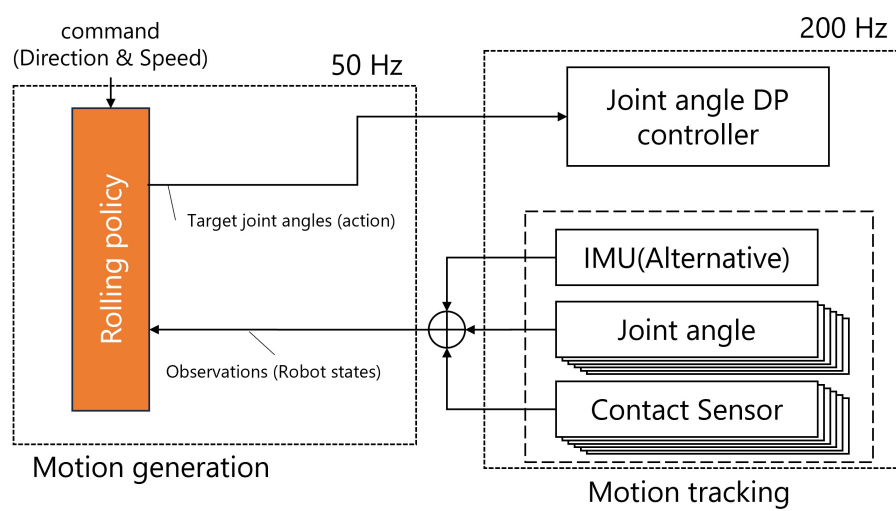
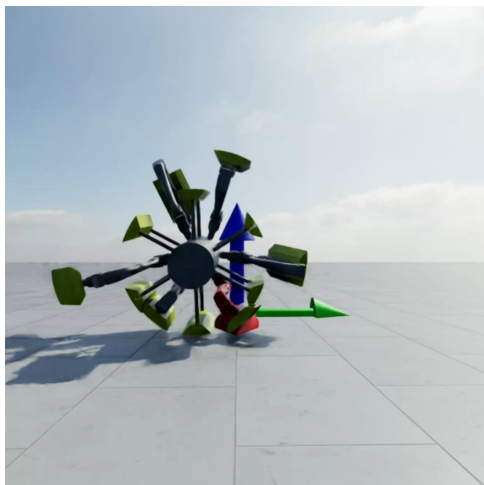
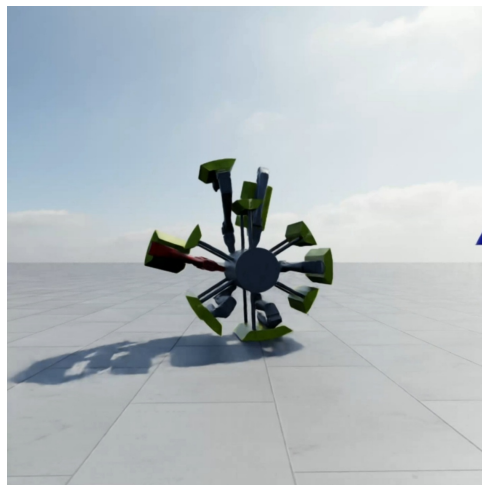


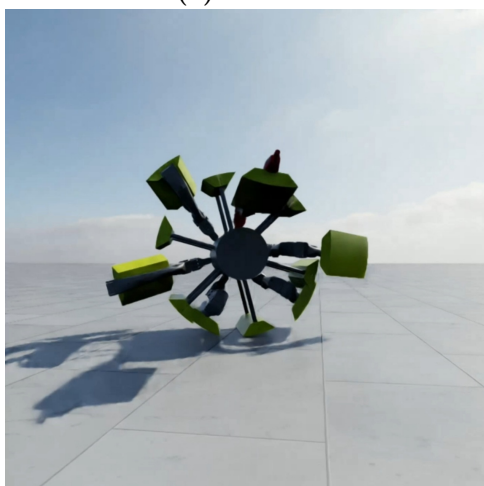
図 5.10: ローリング走行コントローラアーキテクチャ



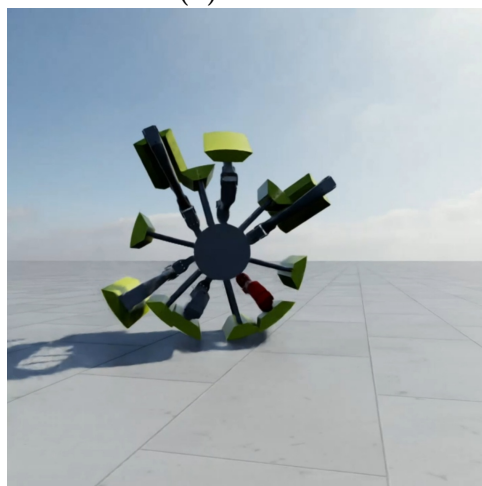
(a)  $t = 0.0\text{s}$



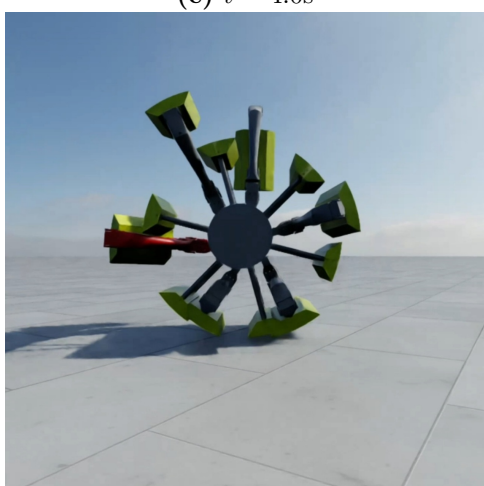
(b)  $t = 2.0\text{s}$



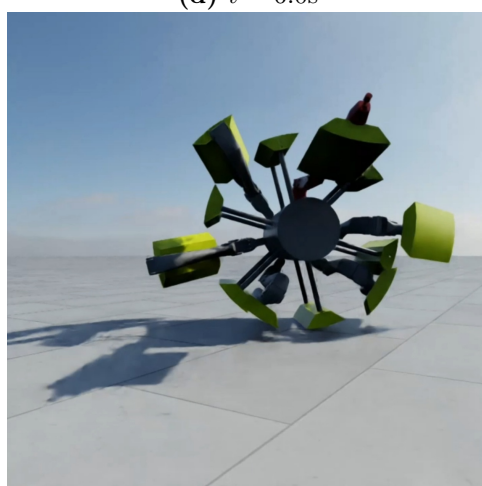
(c)  $t = 4.0\text{s}$



(d)  $t = 6.0\text{s}$



(e)  $t = 8.0\text{s}$



(f)  $t = 10.0\text{s}$

図 5.11: 平坦地形におけるローリング走行モードのタイムラプス

## 5.3 CoT 測定シミュレーション実験

本章では、Rollopod-A の多脚歩行およびローリング走行時における CoT を地形粗さ別に測定するシミュレーション実験について詳述する。

### 5.3.1 CoT 計算式

輸送コスト (Cost of Transport, CoT) は、単位重量を単位距離移動させる際に消費されるエネルギーを基準とした次元のない指標であり、異なる移動形態やエネルギー源を客観的に比較するために用いられる [35]。本実験の CoT 算出式 (式 5.12) を示す。

$$CoT = \frac{\sum_{i=1}^{18} \int_0^t |\tau_i \omega_i| dt}{mg \Delta d} \quad (5.12)$$

本シミュレーション実験では、関節トルク ( $\tau$ )、関節角速度 ( $\omega$ )、およびアクタのワールド座標を、1 エピソード ( $t = 60$  秒) 間に 0.02 秒間隔でサンプリングした。各関節  $i$  において、関節トルクと関節角速度を積分し、その総和を求めることで 1 エピソードあたりにアクタが消費したエネルギーを算出した。次に、 $mg$  はアクタの重量 (15kg) であり、 $\Delta d$  ( $= \sum_{j=0}^{N-1} \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2}$ ) はアクタの総移動距離である。ここで、 $(x_j, y_j)$  はアクタのワールド座標であり、 $N$  はサンプリング数を示す。

### 5.3.2 CoT 計測地形モデル

CoT の測定には、5.2.3 節で述べた学習地形をそれぞれ個別に使用した。テスト地形の面積は  $200m^2$  を確保し、アクタは地形の中心からスタートする設定とした。

### 5.3.3 CoT 測定手順

本節では、多脚歩行モードおよびローリング走行モードにおける地形および難易度別の CoT 測定方法について詳述する。

アクタのデータ取得には、Isaac Lab の recorder 機能を使用した。この機能は、シミュレーションステップ (0.02 秒) ごとにアクタに対してアクションが適用された後の任意の物理値を記録できる。具体的には、この機能を用いて、アクタごとの関節トルク、関節角速度、アクタ座標を 1 エピソード (60 秒間) あたり  $N = 3000$  サンプリングし、HDF5 形式で保存した。このサンプリングデータを基に式 5.12 を用いて CoT を計算した。アクタへの指示速度は、すべての地形および各移動モードにおいて一律  $0.75m/s$  と設定した。

高周波および低周波ウェーブ地形では、地形振幅を  $0.0m \sim 0.20m$  の範囲で  $0.01m$  刻みに変化させた。各振幅地形において、多脚歩行モードとローリング走行モードのアクタをそれぞれ 128 体用い、60 秒間の移動を行った。多脚歩行ポリシに対する指示コマンドは  $(lin_v el_x = 0.0, lin_v el_y = -0.53, ang_v el_z = 0.0)$ 、ローリング走行ポリシに対する指示コマンドは  $(-0.001 \pm lin_v el_x \pm 0.001, lin_v el_y = -1.0, ang_v el_z = 2.14)$  と設定した。

上り勾配および下り勾配地形では、傾斜度を  $0\% \sim 20\%$  の範囲で  $1\%$  刻みに変化させた。各傾斜地形においても、多脚歩行モードおよびローリング走行モードのアクタをそれぞれ 128 体用いて 60 秒間の移動を実施した。多脚歩行ポリシに対する指示コマンドは  $(lin_v el_x = 0.0, lin_v el_y = -0.53, ang_v el_z = 0.0)$ 、ローリング走行ポリシに対する指示コマンドは  $(-0.001 \pm lin_v el_x \pm 0.001, lin_v el_y = -1.0, ang_v el_z = 2.14)$  とした。



## 第6章 結果と考察

本章では、第5章で実施した CoT 測定実験の結果を示し、その考察を行う。本実験では、各移動モードと地形別の CoT 測定結果、そしてローリング走行モードにおける CoT 削減効果を分析した（表 6.1）。実験の結果、ウェーブ地形ではローリング走行モードが最大で 53% の CoT 削減効果を示した。また、勾配地形では、下り勾配（傾斜度 7%）で最大 67%、上り勾配（傾斜度 7%）で最大 63% の CoT 削減効果が確認された。

表 6.1: 移動モード、地形別 CoT 測定結果とローリング走行による最大 CoT 削減効果

| 地形        | 移動モード | 最小 CoT         | 最大 CoT         | 最大 CoT 削減効果 |
|-----------|-------|----------------|----------------|-------------|
| 高周波ウェーブ地形 | 歩行    | $14.0 \pm 0.1$ | $22.1 \pm 0.1$ | 53%（平地）     |
|           | ローリング | $6.6 \pm 0.1$  | $35.9 \pm 0.1$ |             |
| 低周波ウェーブ地形 | 歩行    | $14.0 \pm 0.1$ | $15.4 \pm 0.1$ | 53%（平地）     |
|           | ローリング | $6.6 \pm 0.1$  | $9.0 \pm 0.1$  |             |
| 上り勾配地形    | 歩行    | $14.0 \pm 0.1$ | $28.4 \pm 0.1$ | 67%（勾配 7%）  |
|           | ローリング | $6.0 \pm 0.1$  | $22.8 \pm 0.1$ |             |
| 下り勾配地形    | 歩行    | $14.0 \pm 0.1$ | $30.2 \pm 0.1$ | 63%（勾配 7%）  |
|           | ローリング | $7.0 \pm 0.1$  | $25.3 \pm 0.1$ |             |

## 6.1 ウェーブ地形 CoT 測定結果

ウェーブ地形における高周波および低周波の地形粗さ別に、多脚歩行モードとローリング走行モードの CoT 測定結果を図 6.1 および図 6.2 に示す。両地形において、ローリング走行モードの CoT 削減効果が最大となったのは平面地形であった。

- 高周波ウェーブ地形

高周波ウェーブ地形では、多脚歩行モードにおいて地形振幅が増加すると緩やかに CoT が上昇し、最大振幅時には  $CoT = 22.1$  に達した。一方で、ローリング走行モードでは地形振幅の増加に伴い急激に CoT が上昇し、振幅が  $0.08m$  に達した時点で多脚歩行モードを上回った。

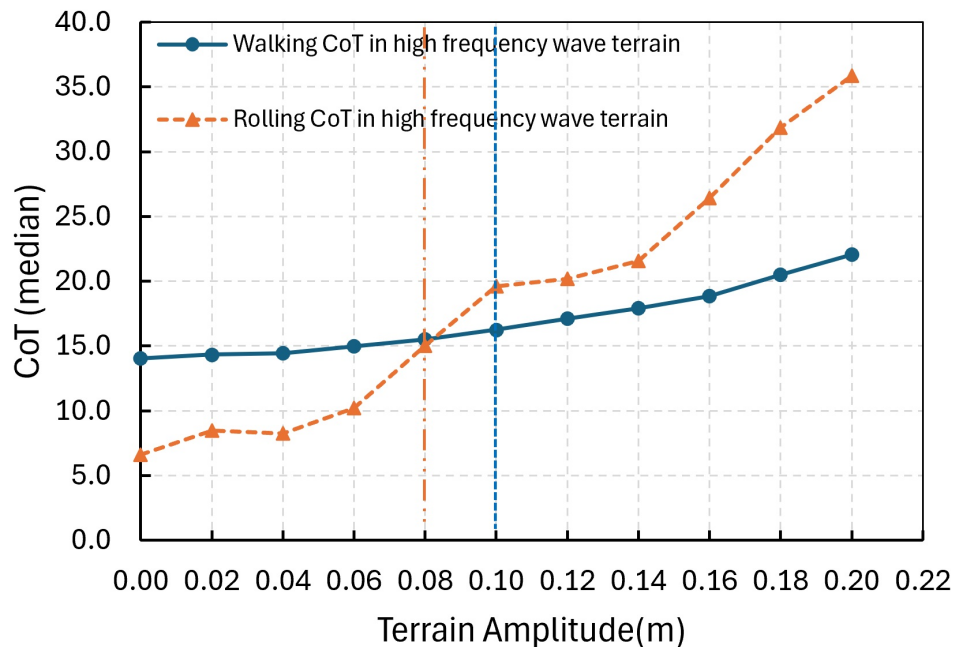


図 6.1: 高周波ウェーブ地形の振幅と CoT の関係：移動モード別の比較（縦の破線は歩行ポリシの最大学習レベル、縦の二点鎖線はローリングポリシの最大学習レベル）

- 低周波ウェーブ地形

低周波ウェーブ地形においては、全振幅においてローリング走行モードが多脚歩行モードより平均 51% 優れた CoT を示した。地形振幅が増加すると両モードともに CoT が増加したが、多脚歩行モードでは最大 1.4 ポイント、ローリング走行モードでは最大 2.4 ポイントの増加にとどまった。

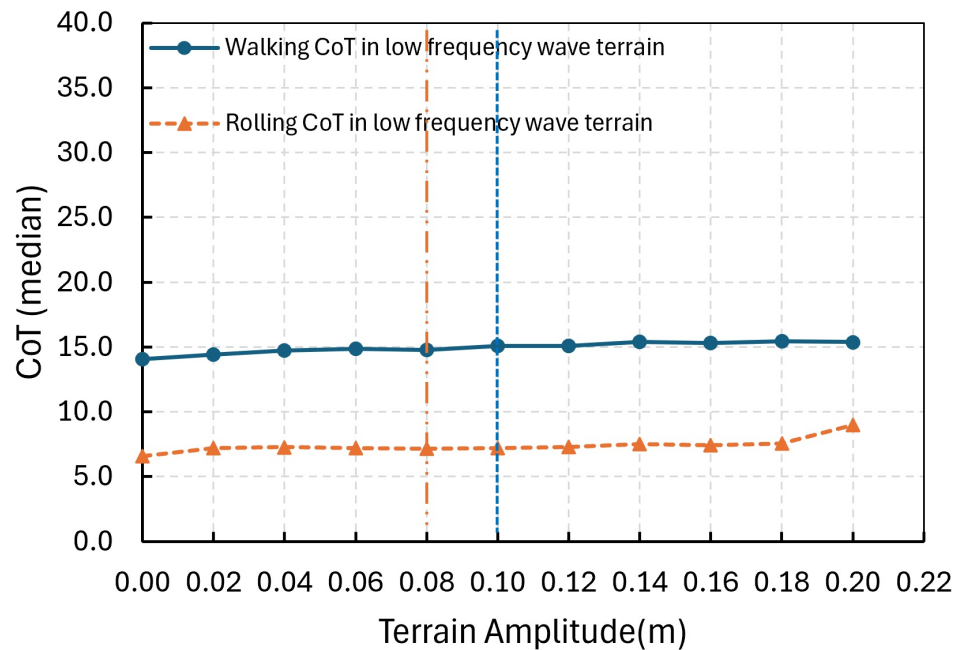


図 6.2: 低周波ウェーブ地形の振幅と CoT の関係：移動モード別の比較（縦の破線は歩行ポリシの最大学習レベル、縦の二点鎖線はローリングポリシの最大学習レベル）

## 6.2 勾配地形 CoT 測定結果

勾配地形では、上り勾配および下り勾配それぞれにおける多脚歩行モードとローリング走行モードの傾斜度別 CoT 測定結果を図 6.3 および図 6.4 に示す。両地形において、ローリング走行モードの CoT 削減効果が最大となったのは傾斜度 7% の地点である。

- 上り勾配地形

上り勾配地形では、傾斜度 8.0% まではローリング走行モードの CoT が平均 9.9 ポイント低くなっている。一方、傾斜度 9% 以降は CoT が急速に増加し、傾斜度 11.0% の時点で歩行モードを上回っている。

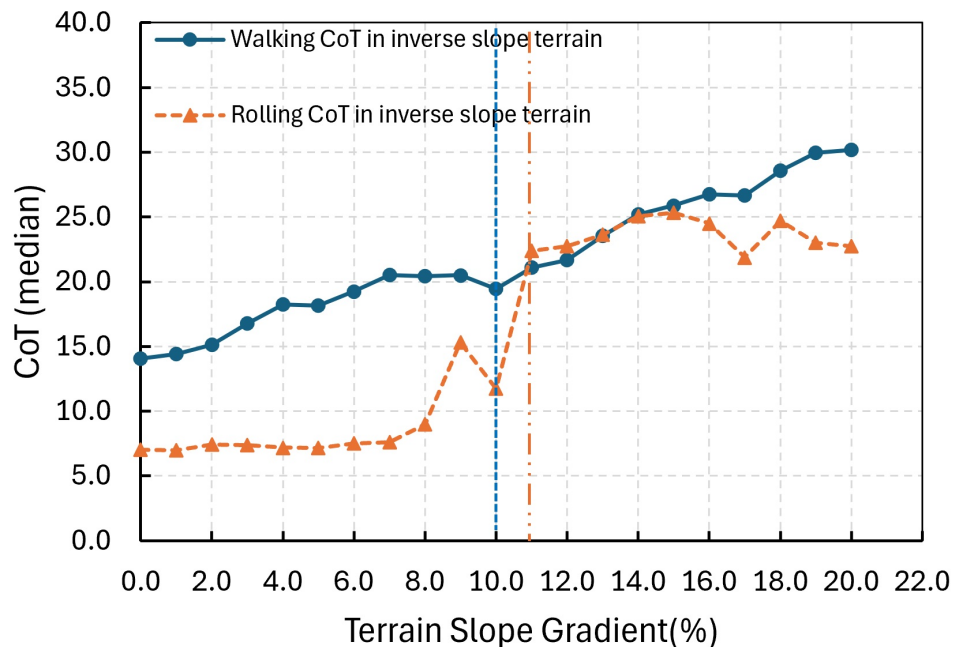


図 6.3: 上り勾配地形の傾斜度と CoT の関係：移動モード別の比較（縦の破線は歩行ポリシの最大学習レベル、縦の二点鎖線はローリングポリシの最大学習レベル）

- 下り勾配地形

下り勾配地形では、多脚歩行モードの CoT が傾斜度の増加に伴って増加するのに対し、ローリング走行モードでは傾斜度 7% まで減少を続け、最大で 0.9 ポイント減少した。しかし、傾斜度 8% 以降急速に CoT が悪化し、CoT 低減効果が最小 8% まで低下した。一方、傾斜度 12% を超えた地点では、ローリング走行モードの CoT が再び減少傾向に転じている。

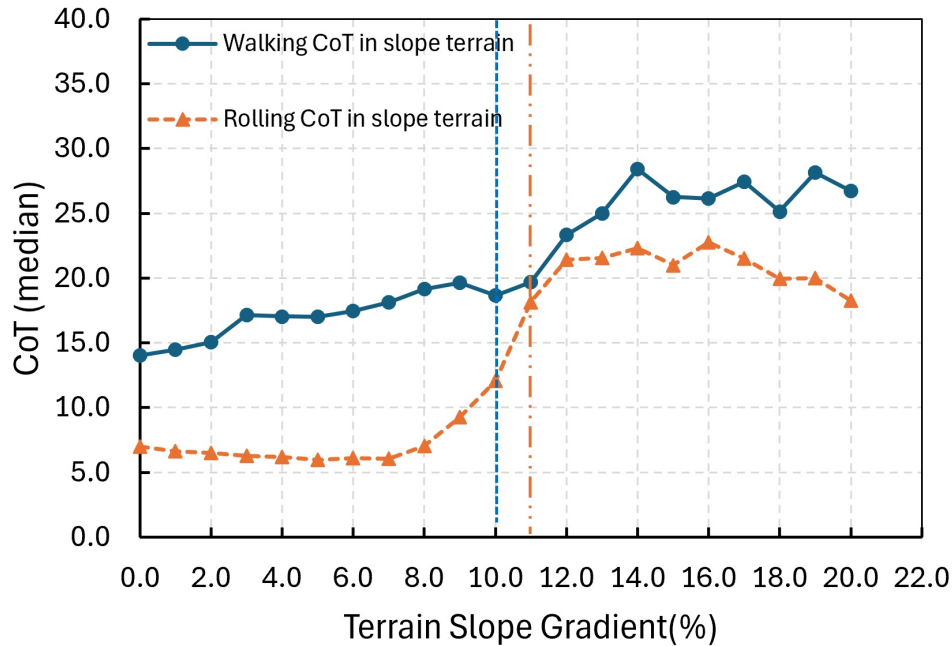


図 6.4: 下り勾配地形の傾斜度と CoT の関係：移動モード別の比較（縦の破線は歩行ポリシの最大学習レベル、縦の二点鎖線はローリングポリシの最大学習レベル）

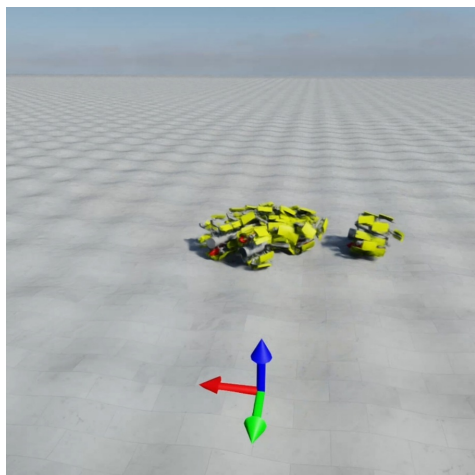
## 6.3 考察

本節では、著者がシミュレーションで目視観察を行った結果を踏まえ、各地形におけるローリング走行モードの挙動について考察する。

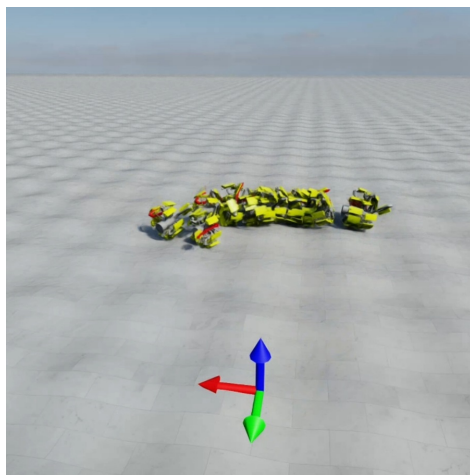
- 高周波ウェーブ地形

高周波ウェーブ地形では、多脚歩行モードにおいて地形振幅が増加するにつれて CoT も穏やかに上昇しており、安定して歩行できていることが確認された。これは、多脚歩行が不整地移動に適しているという関連研究の知見と一致している。一方、ローリング走行モードでは地形振幅 0.06m までは安定的に移動が可能で、CoT も平均 6 ポイント優れていた。しかし、地形振幅が 0.07m を超えると急激に CoT が上昇している。地形振幅 0.07m のシミュレーション映像（図 6.5）を確認すると、一部のアクタが転倒している

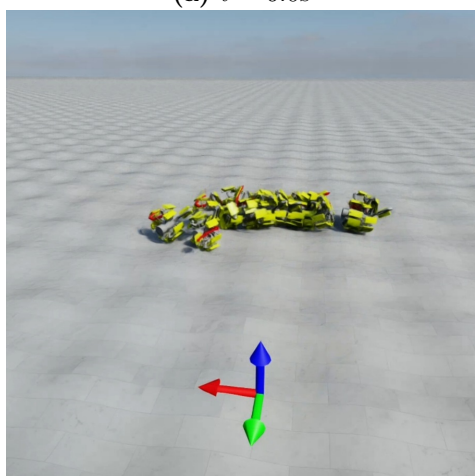
ことが確認できた。また、地形の隆起に伴い、アクタが進行方向に対して左右に揺れ動く挙動が確認された。この現象は、Rollopod-A が有する固定外殻によって地形形状の影響を大きく受けていることに起因すると考えられる。次に、最大振幅  $0.20m$  の映像（図 6.6）では、ほとんどのアクタが開始地点から移動できていない状況が見られた。これらの結果から、地形振幅が大きくなると転倒による移動距離の減少と転倒復帰のためのエネルギー消費が重なり、CoT が急速に悪化するものと考えられる。



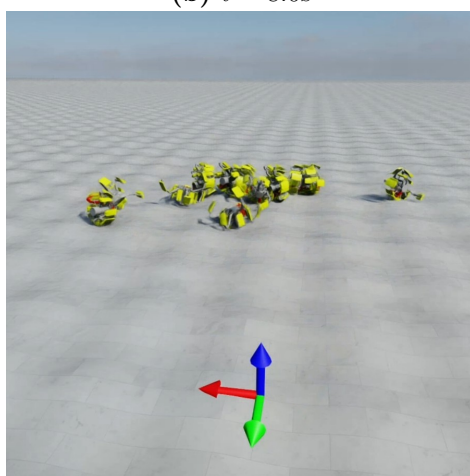
(a)  $t = 6.0\text{s}$



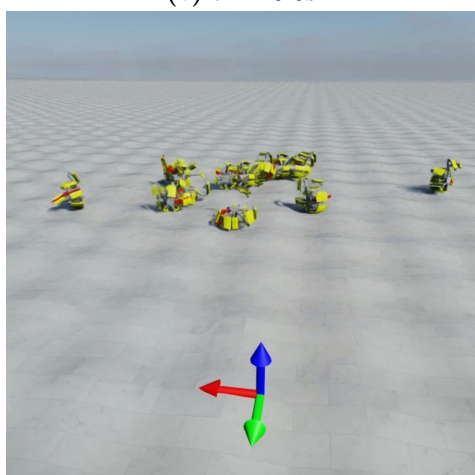
(b)  $t = 8.0\text{s}$



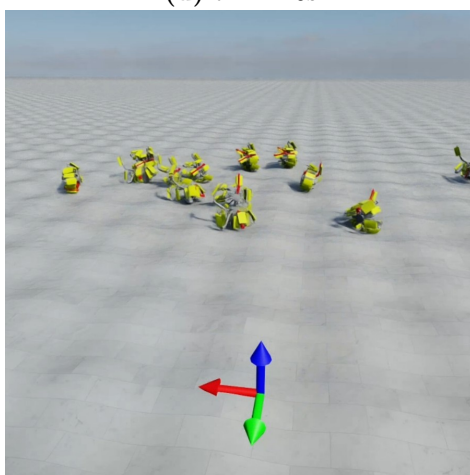
(c)  $t = 10.0\text{s}$



(d)  $t = 12.0\text{s}$



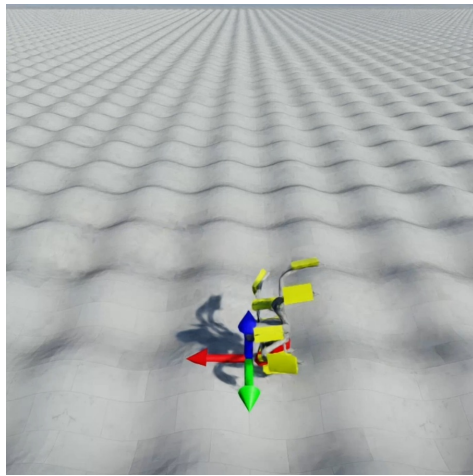
(e)  $t = 14.0\text{s}$



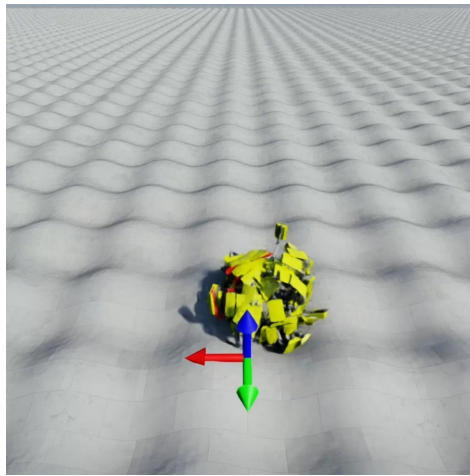
(f)  $t = 16.0\text{s}$

図 6.5: 高周波ウェーブ地形における振幅  $0.07m$  のローリング走行モードタイムラプス

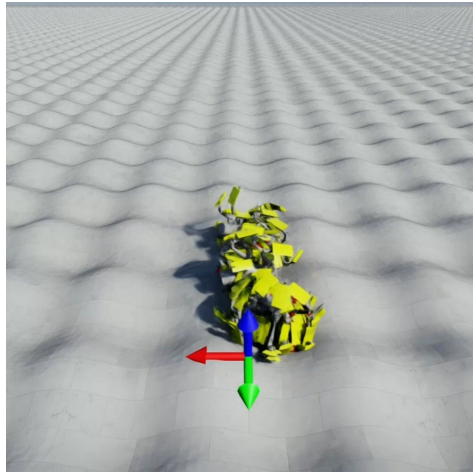




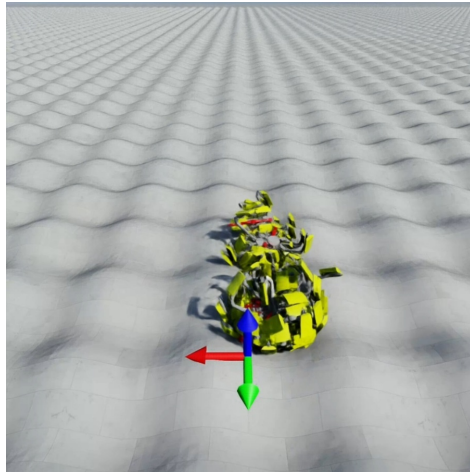
(a)  $t = 0.0s$



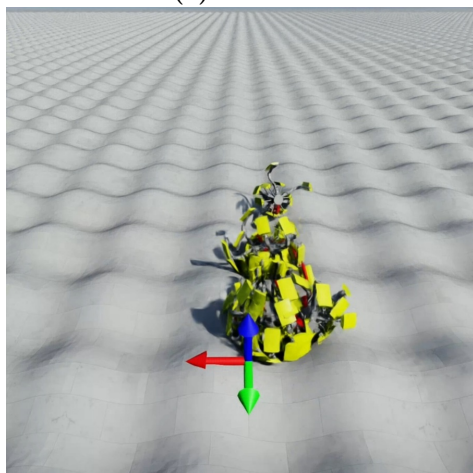
(b)  $t = 2.0s$



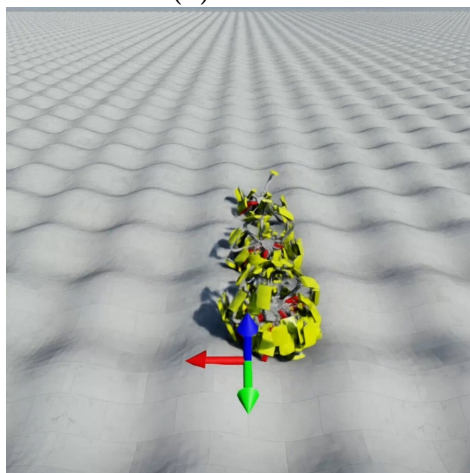
(c)  $t = 4.0s$



(d)  $t = 6.0s$



(e)  $t = 8.0s$



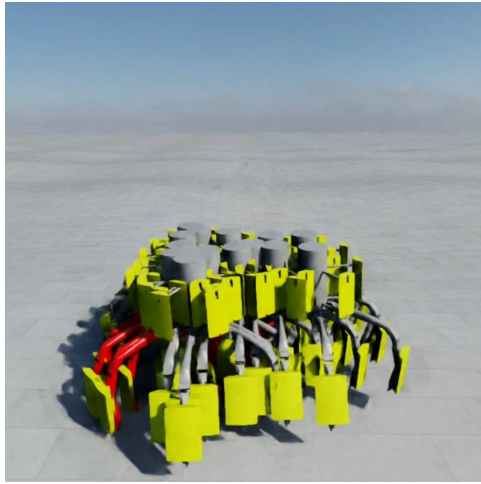
(f)  $t = 10.0s$

図 6.6: 高周波ウェーブ地形における振幅  $0.20m$  のローリング走行モードタイムラプス

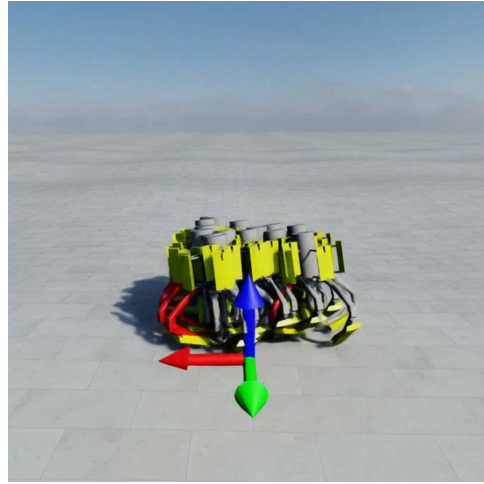


- 低周波ウェーブ地形

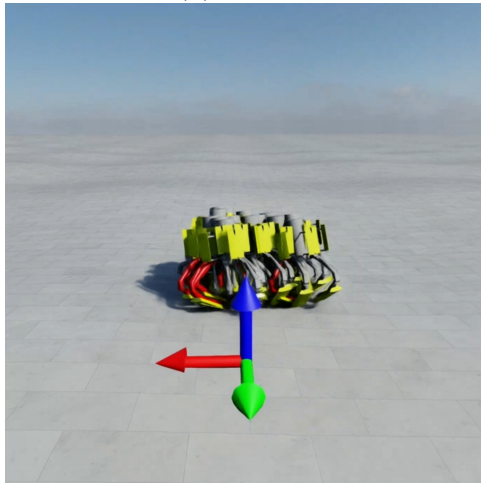
低周波ウェーブ地形では、多脚歩行モードとローリング走行モードのいずれにおいても、地形振幅の増加に伴う CoT の増加がほとんど観測されなかった。多脚歩行モードについては CoT が安定すると予想されていたが、ローリング走行モードにおいても終始安定的な移動が可能であった点は予想外の結果であった。これは、後述する上り勾配地形で傾斜度 7% の登坂が可能であり、CoT が多脚歩行モードよりも低かったことを考慮すると、低周波ウェーブ地形では地形変化の影響をほとんど受けなかったことを示している。この結果から、ある程度穏やかな不整地ではローリング走行が有効であると結論づけられる。最大振幅時の低周波ウェーブ地形における多脚歩行モードタイムラプスを図 6.7、ローリング走行モードのタイムラプスを図 6.8 に示す。



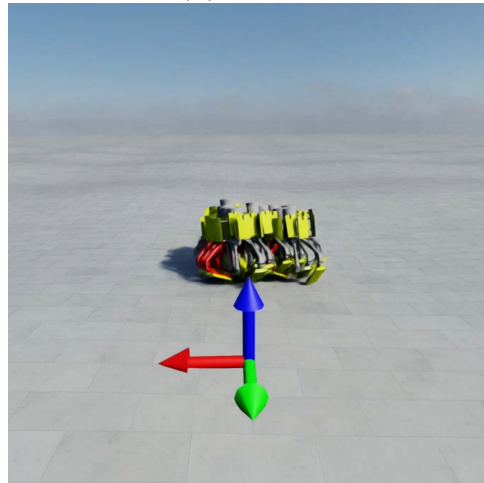
(a)  $t = 0.0\text{s}$



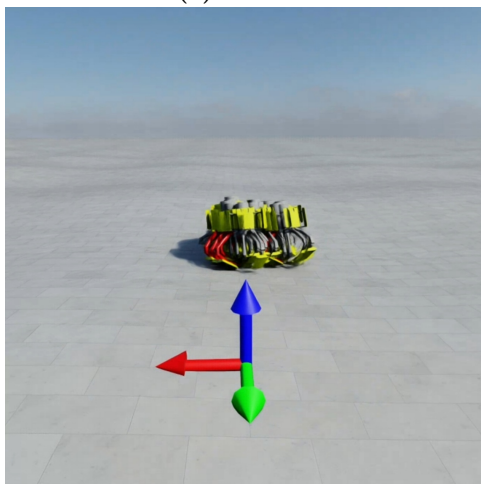
(b)  $t = 2.0\text{s}$



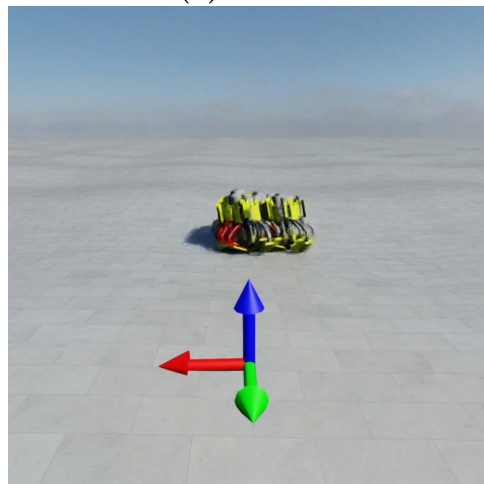
(c)  $t = 4.0\text{s}$



(d)  $t = 6.0\text{s}$

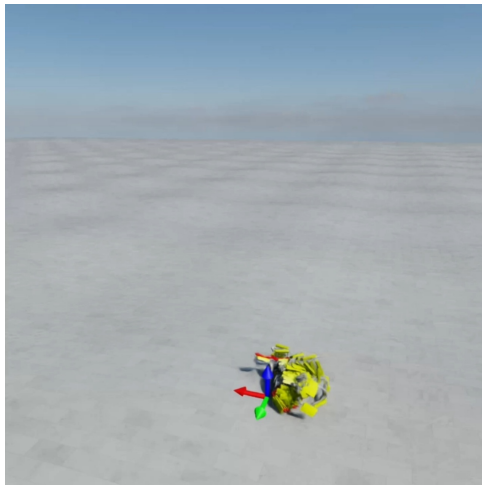


(e)  $t = 8.0\text{s}$

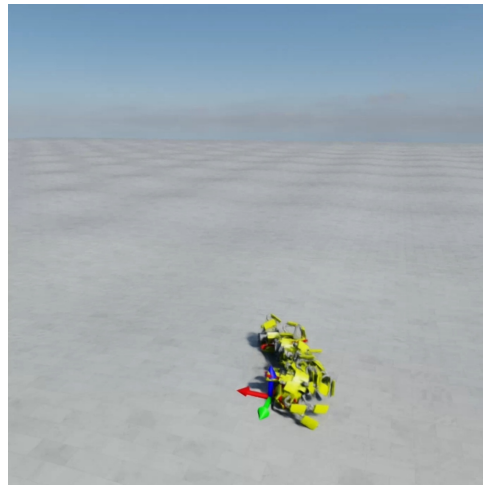


(f)  $t = 10.0\text{s}$

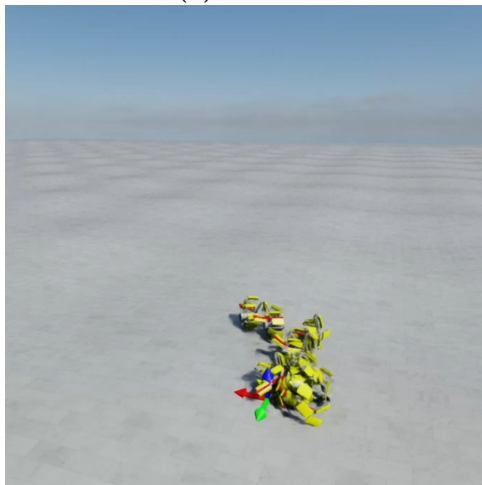
図 6.7: 低周波ウェーブ地形における振幅  $0.20m$  の多脚歩行モードタイムラプス



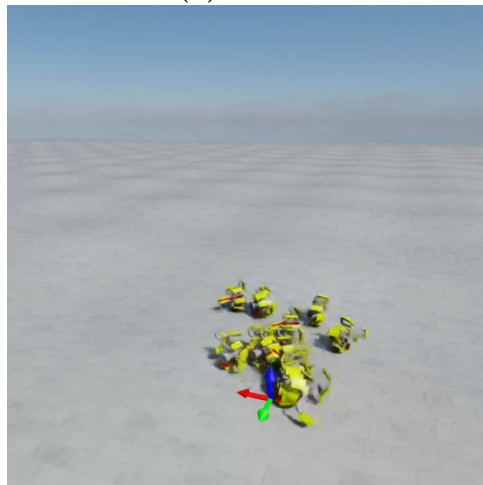
(a)  $t = 0.0\text{s}$



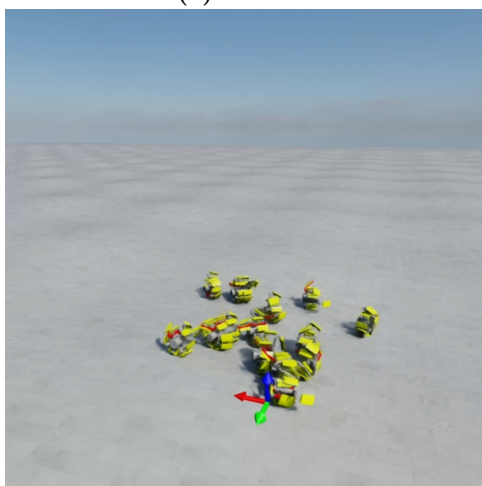
(b)  $t = 2.0\text{s}$



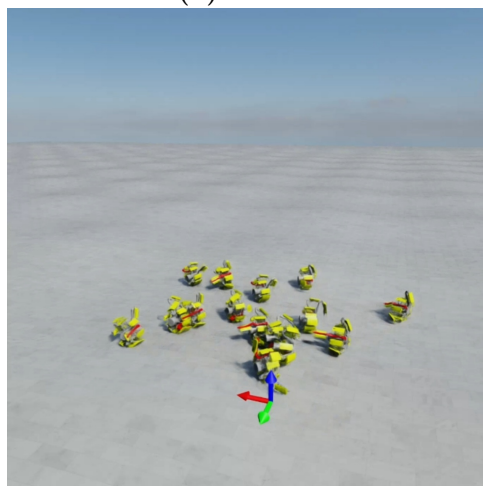
(c)  $t = 4.0\text{s}$



(d)  $t = 6.0\text{s}$



(e)  $t = 8.0\text{s}$

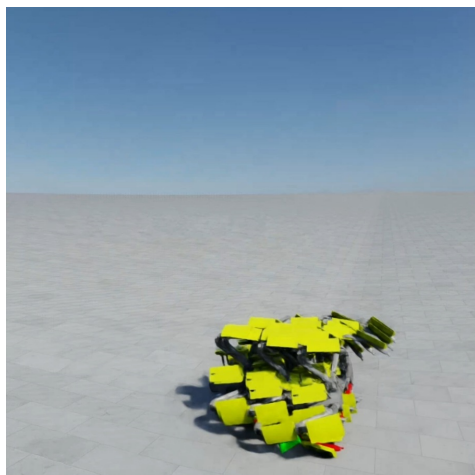


(f)  $t = 10.0\text{s}$

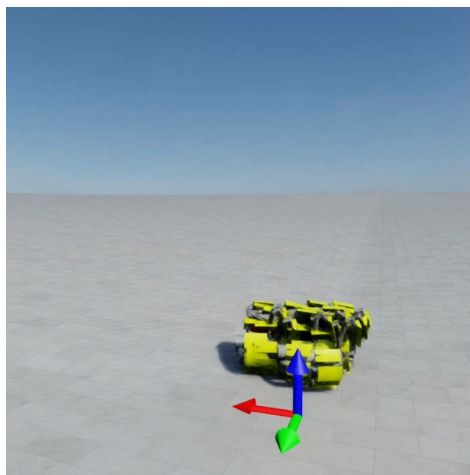
図 6.8: 低周波ウェーブ地形における振幅  $0.20m$  のローリング走行モードタイムラプス

- 上り勾配地形

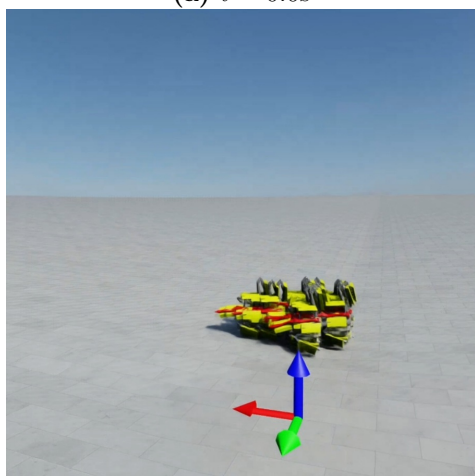
上り勾配地形において、Rollopod-A はローリング走行モードで驚異的な登坂能力を発揮した。傾斜度 7% のシミュレーション動画 (図 6.9) では、継続的に坂を登る姿勢が確認され、これは他のローリングロボットには見られない特徴である。しかし、CoT が急激に上昇した傾斜度 9% の映像 (図 6.10) では、大半のアクタが途中で転倒し、進行方向が大きく乱れていることが確認された。この現象は、ローリングコントロールポリシが傾斜面平行方向に作用する重力成分に対応できず、登坂角度のずれや転倒を引き起こしたためと考えられる。さらに、傾斜度 11% (図 6.11) では大多数のアクタが登坂に失敗したものの、一部のアクタは継続的に登坂を行っていた。この結果から、ローリングコントロールポリシの性能を向上させることで、さらなる登坂能力を獲得できる可能性が示唆される。



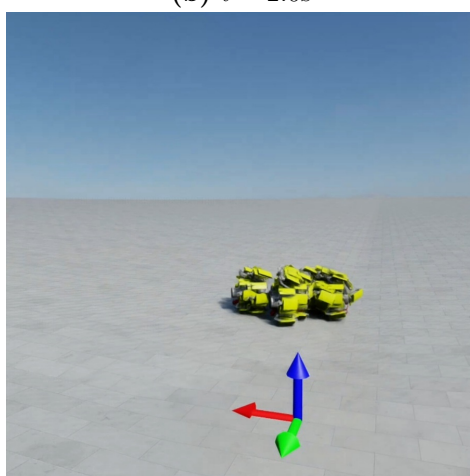
(a)  $t = 0.0\text{s}$



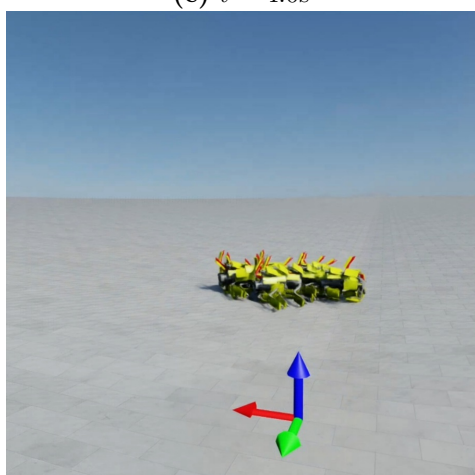
(b)  $t = 2.0\text{s}$



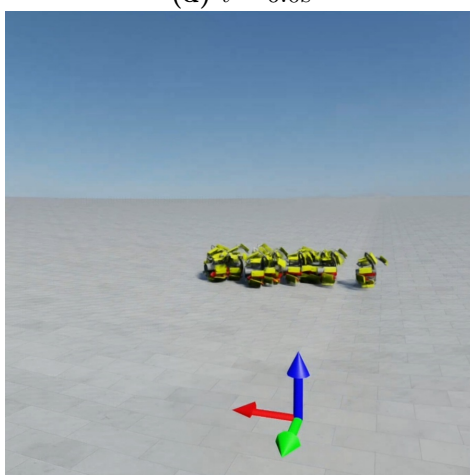
(c)  $t = 4.0\text{s}$



(d)  $t = 6.0\text{s}$



(e)  $t = 8.0\text{s}$



(f)  $t = 10.0\text{s}$

図 6.9: 上り勾配地形における傾斜度 7% のローリング走行モードタイムラプス

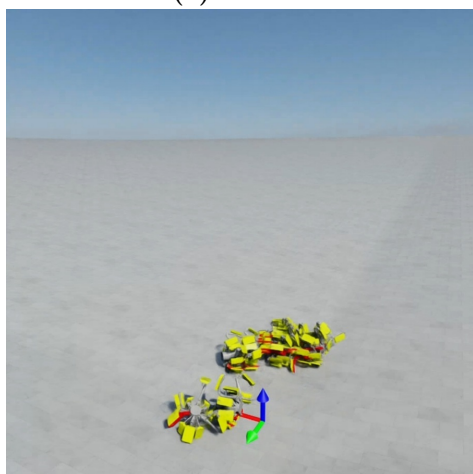




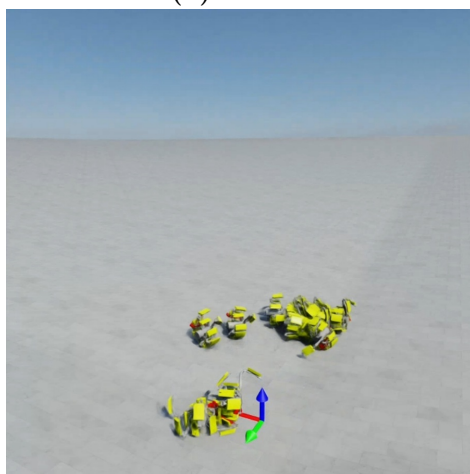
(a)  $t = 0.0\text{s}$



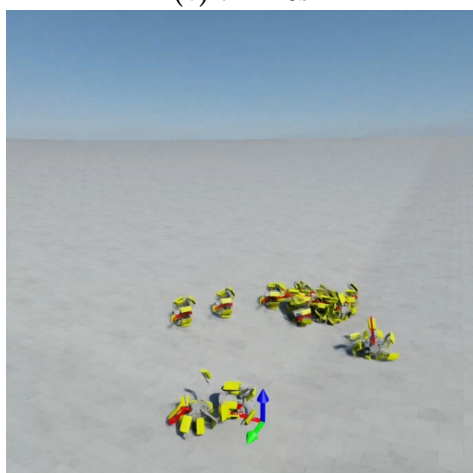
(b)  $t = 2.0\text{s}$



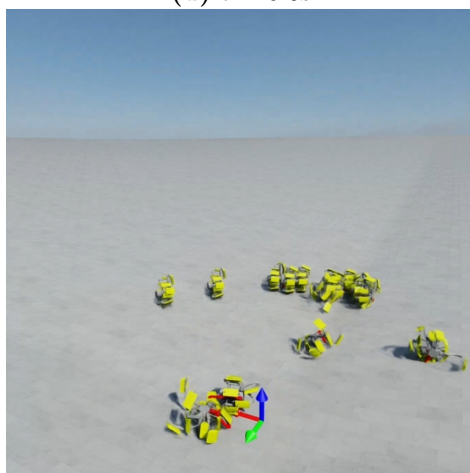
(c)  $t = 4.0\text{s}$



(d)  $t = 6.0\text{s}$

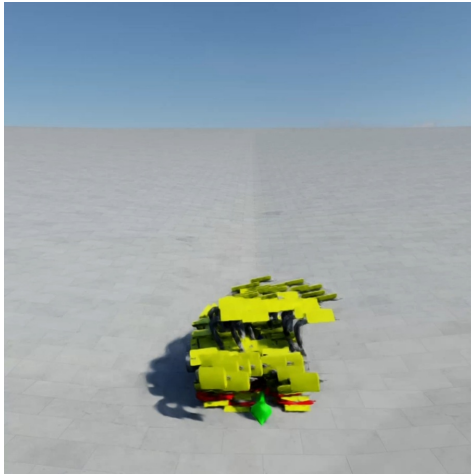


(e)  $t = 8.0\text{s}$

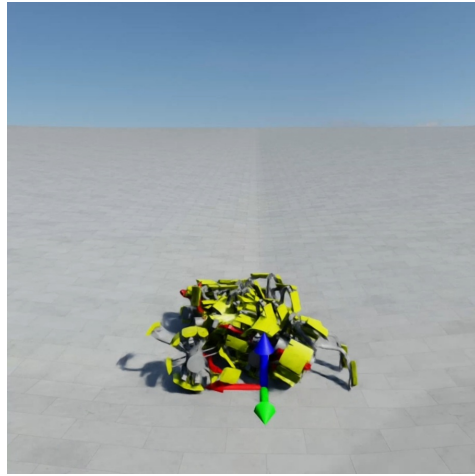


(f)  $t = 10.0\text{s}$

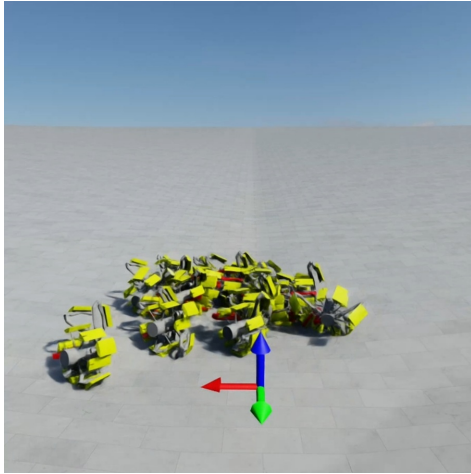
図 6.10: 上り勾配地形における傾斜度 9% のローリング走行モードタイムラプス



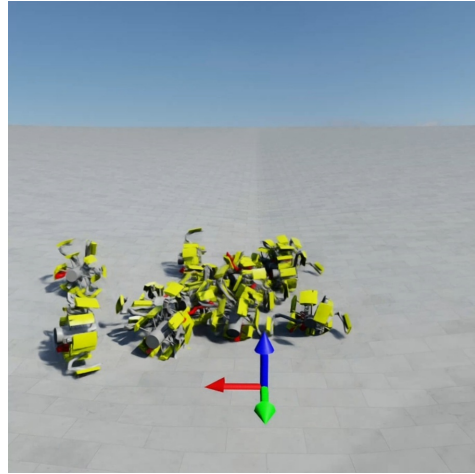
(a)  $t = 0.0\text{s}$



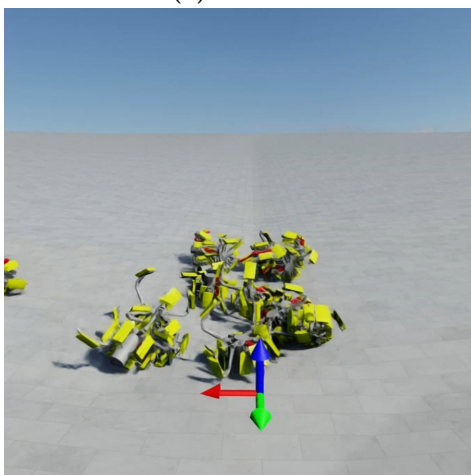
(b)  $t = 2.0\text{s}$



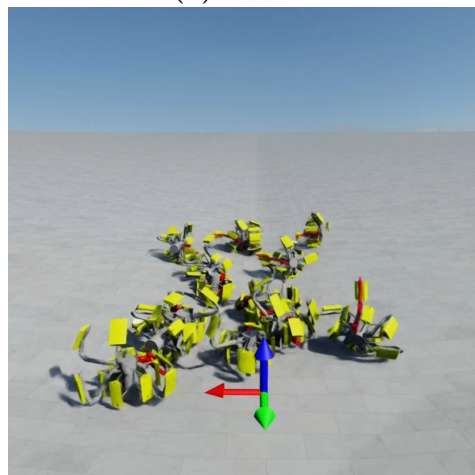
(c)  $t = 4.0\text{s}$



(d)  $t = 6.0\text{s}$



(e)  $t = 8.0\text{s}$



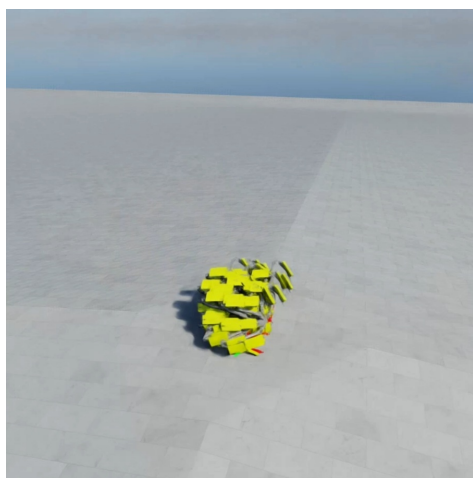
(f)  $t = 10.0\text{s}$

図 6.11: 上り勾配地形における傾斜度 11% のローリング走行モードタイムラプス

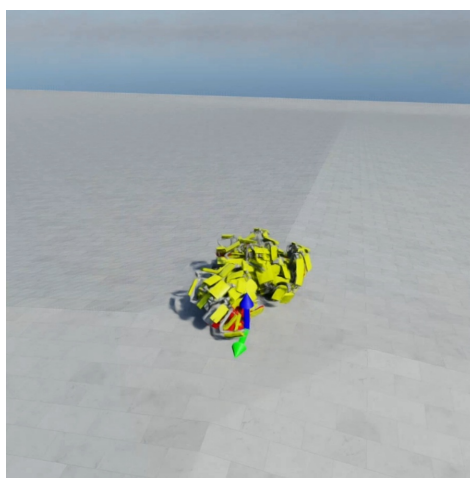
- 下り勾配地形

下り勾配地形では、ローリング走行モードが多脚歩行モードに比べて CoT が優れており、傾斜度 7% までは CoT が減少するという優位性が確認された。この結果は、ローリングモードが重力成分を回転運動に変換できるためであり、一方で多脚歩行モードでは傾斜面平行に作用する重力成分に逆らう必要があることによる。しかし、傾斜度 7% を超えると CoT が上昇し、傾斜度 12% のシミュレーション映像（図 6.12）では、傾斜方向とローリング方向の差異による転倒や進行方向の逸れが観測された。この挙動が CoT の急激な悪化を引き起こしたと推察される。さらに、傾斜度 16% 以降（図 6.13）では転倒後に滑落するアクタが多く、見かけ上の移動距離の増加により CoT が減少したと考えられる。一方、多脚歩行モードでは傾斜度 10%（図 6.14）までは安定して斜面を下ることができたが、それ以上の傾斜度 14%（図 6.15）では転倒が増加し、歩行に失敗するアクタが増加した。この現象は、歩行コントロールポリシーが最大傾斜 10% までしか学習していないことに起因していると考えられる。

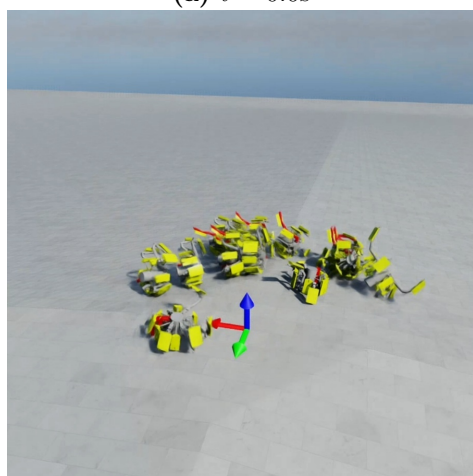




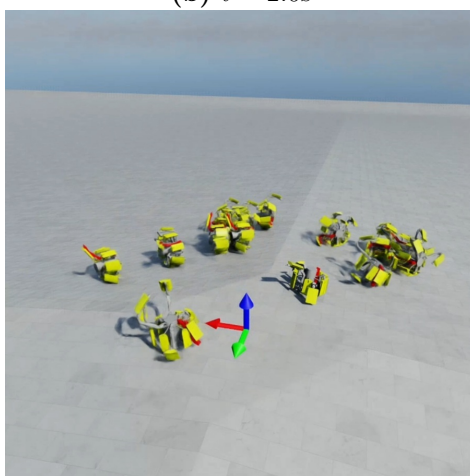
(a)  $t = 0.0\text{s}$



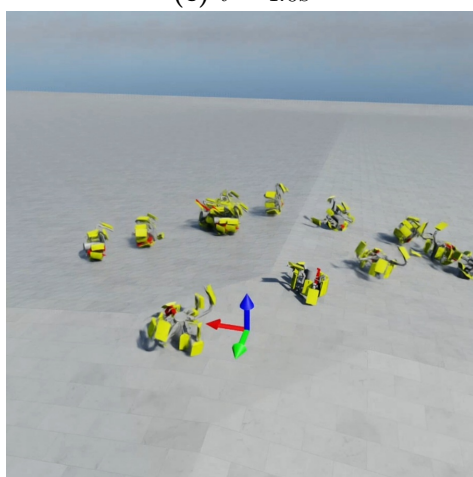
(b)  $t = 2.0\text{s}$



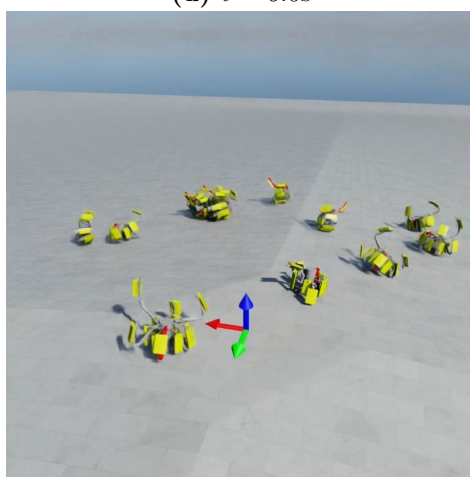
(c)  $t = 4.0\text{s}$



(d)  $t = 6.0\text{s}$

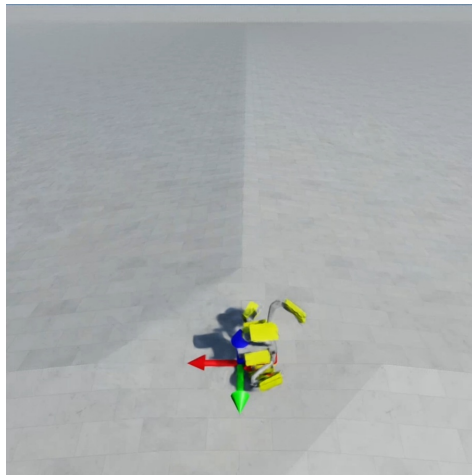


(e)  $t = 8.0\text{s}$

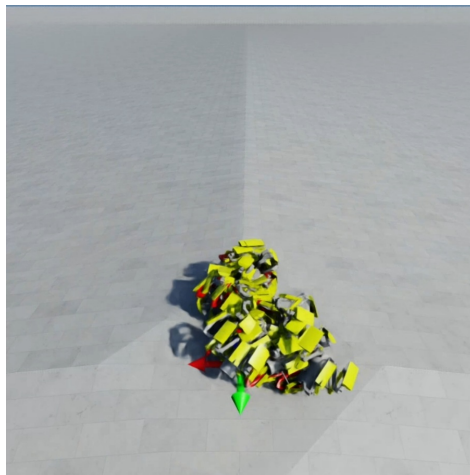


(f)  $t = 10.0\text{s}$

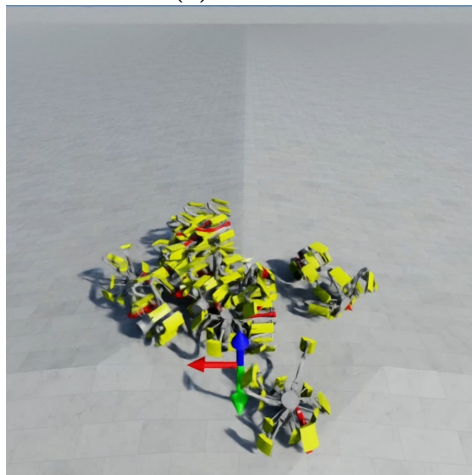
図 6.12: 下り勾配地形における傾斜度 12% のローリング走行モードタイムラプス



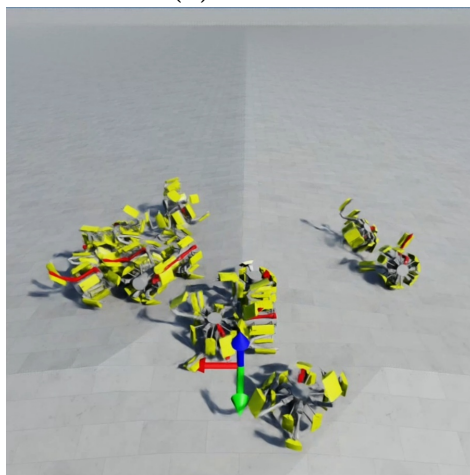
(a)  $t = 0.0\text{s}$



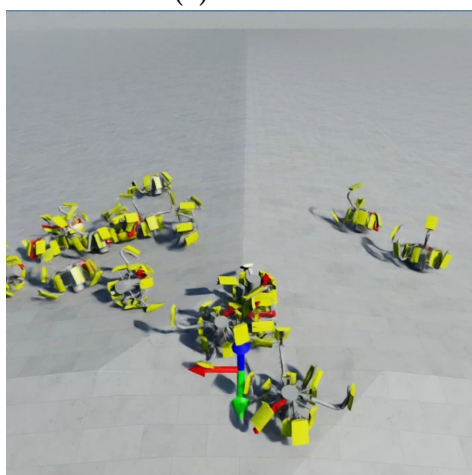
(b)  $t = 2.0\text{s}$



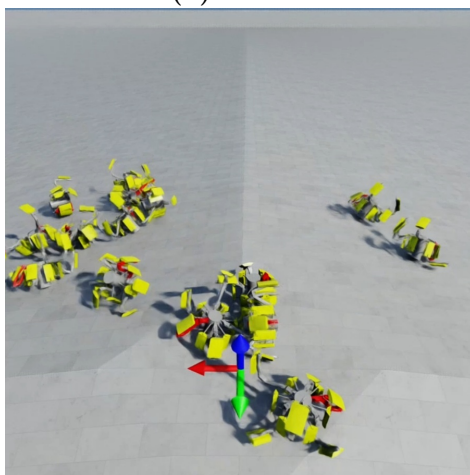
(c)  $t = 4.0\text{s}$



(d)  $t = 6.0\text{s}$

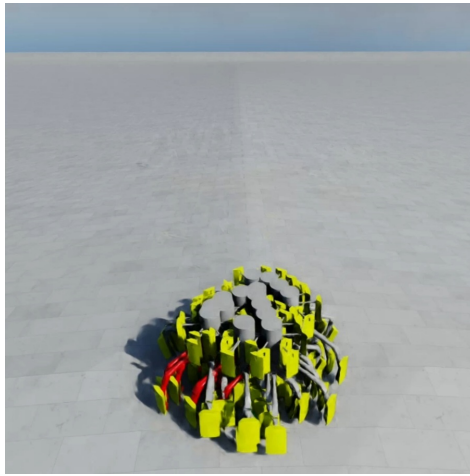


(e)  $t = 8.0\text{s}$

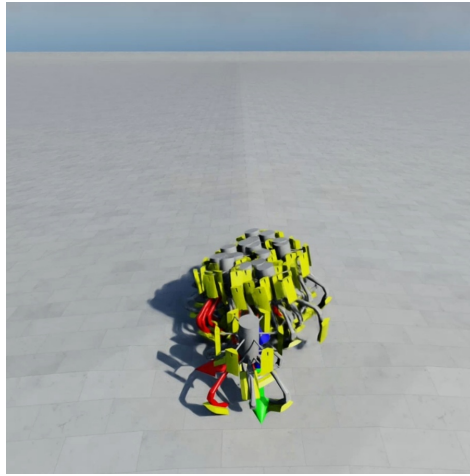


(f)  $t = 10.0\text{s}$

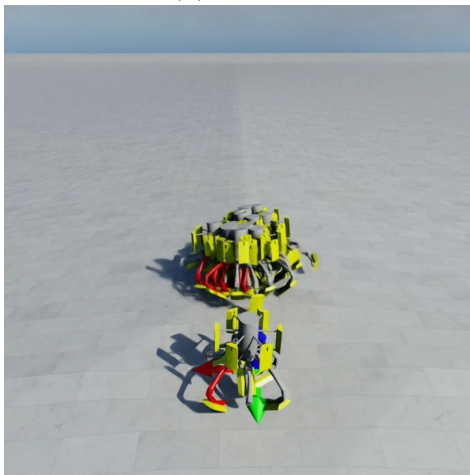
図 6.13: 下り勾配地形における傾斜度 16% のローリング走行モードタイムラプス



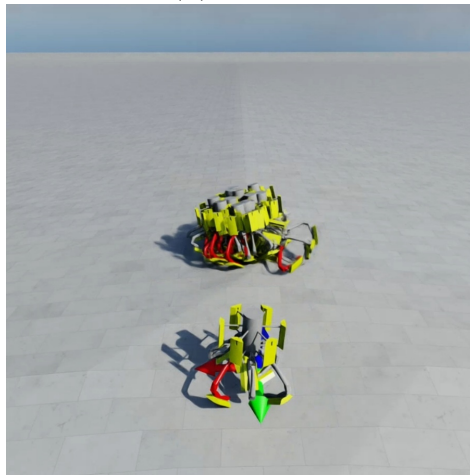
(a)  $t = 0.0\text{s}$



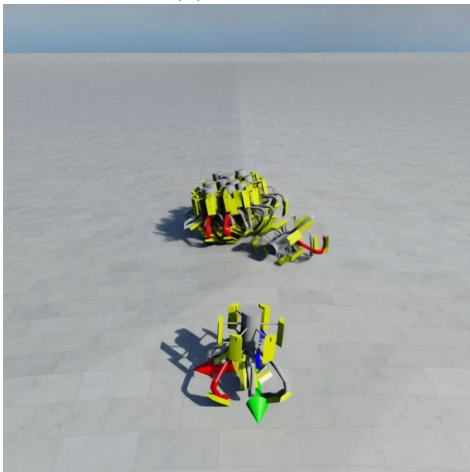
(b)  $t = 2.0\text{s}$



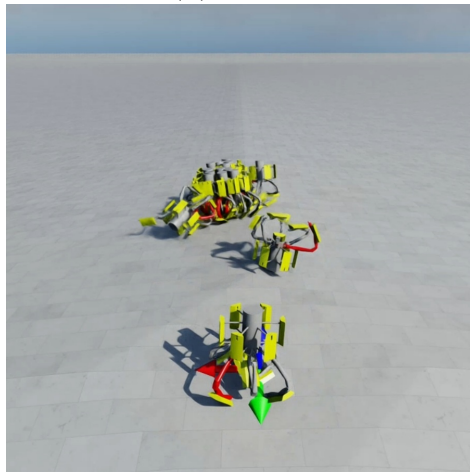
(c)  $t = 4.0\text{s}$



(d)  $t = 6.0\text{s}$



(e)  $t = 8.0\text{s}$



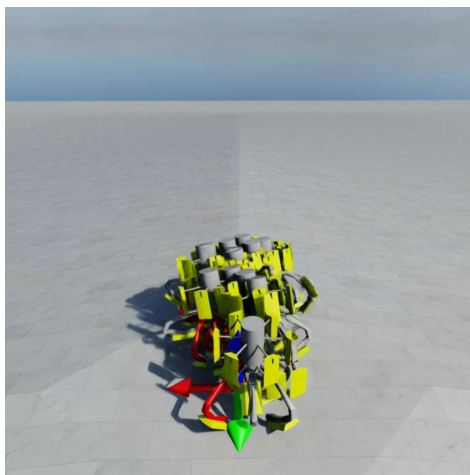
(f)  $t = 10.0\text{s}$

図 6.14: 下り勾配地形における傾斜度 10% の多脚歩行モードタイムラプス、スタックおよび転倒しているアクタは、5.2.3 節で述べた不適切干渉によるものと推測される

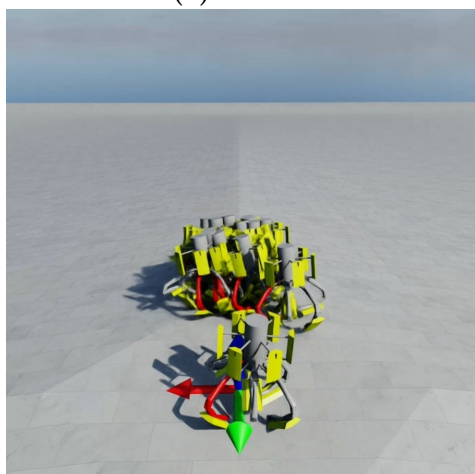




(a)  $t = 0.0\text{s}$



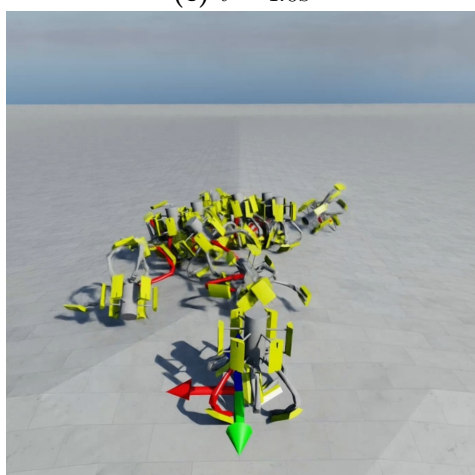
(b)  $t = 2.0\text{s}$



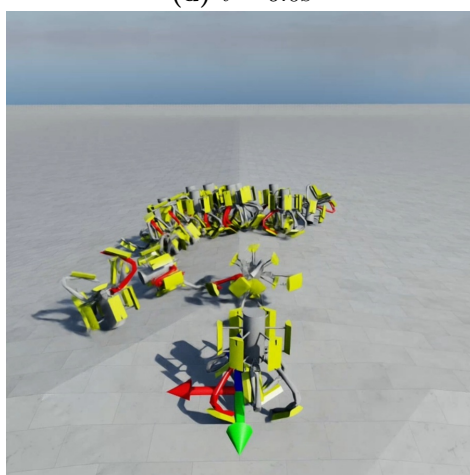
(c)  $t = 4.0\text{s}$



(d)  $t = 6.0\text{s}$



(e)  $t = 8.0\text{s}$



(f)  $t = 10.0\text{s}$

図 6.15: 下り勾配地形における傾斜度 14% の多脚歩行モードタイムラプス

## 第7章 結論

本稿では、多脚歩行ロボットの歩行効率を改善することを目的として、著者らが開発した歩行と登坂が可能なローリング走行を実現する新しい6脚ロボットについて報告した。深層強化学習ベースのコントローラを用いたシミュレーション実験を通じて、本ロボットがローリング走行モードにおいて無次元輸送コスト (Cost of Transport, CoT) を 53% ~ 67% 削減し、さらに 7% の登坂能力を有することを明らかにした。

シミュレーション実験の結果、平地においてローリング走行モードは最大で 53% の CoT 削減効果を示した。高周波ウェーブ地形において、多脚歩行モードでは CoT の増加が最大で 8 ポイントに抑えられており、歩行による高い不整地走破性能が示された。一方、ローリング走行モードでは、地形振幅が  $0.04m$  までは CoT が平均 6.5 ポイント優れているものの、それ以降は転倒に伴い CoT が増加し、激しい荒地での安定性が低いことが明らかとなった。低周波ウェーブ地形では平均 51% の CoT 削減効果が確認され、地形変化が緩やかな環境においてローリング走行モードが有効であることが示された。これは、Rollopod-A のローリング走行モードが傾斜度 7% までにおいて、多脚歩行モードよりも平均で 10.2 ポイント低い CoT を示し、後述する優れた登坂能力を有していることによるものである。

勾配地形においては、下り勾配地形 (傾斜度 7%) で 67%、上り勾配地形 (傾斜度 7%) で 63% の CoT 削減効果が確認された。また、Rollopod-A のキックロコモーションによるローリング走行モードが、傾斜度 7% の登坂能力を有することが明らかとなった。この能力は、関連研究における他のローリングロボットには見られない特性である。これは、関連研究におけるローリングが主に重心移動によって実現されるのに対し、本稿で提案するロボットが第 4 章 4.1 節で説明した脚の駆動力を直接ローリング駆動力に変換する仕組みを備えているためである。これらの結果から、Rollopod-A はローリング走行によって軽度の荒地における移動効率を大幅に改善できることが確認された。また、多脚式 UGV の CoT 削減手法として、関連研究で提案されているホイール付き脚ロボットの CoT 削減効果 (50% ~ 83%) に匹敵する結果を示していることが明らかとなった。

これらの知見から、多脚歩行とローリング走行の組み合わせは、不整地走破能力が高く、効率的に平地や下り勾配地形を移動するのに有効であることが示された。これは、従来のホイール付き脚ロボットとは異なる、新たな多脚歩行ロボットの CoT 削減手法を提案するものである。しかし、現状では Rollopod-A のローリング走行中の安定性や操舵性に課題があり、特に転倒からの復帰動作において改



善の余地がある。また、本研究は階段や障害物を含む現実的な環境を用いた実験ではなく、簡易的なシミュレーション環境（図 5.2）に限定されている。さらに、本稿のシミュレーション実験では、シミュレータに起因する地形との不適切な干渉による転倒が散見されたことから、シミュレーション環境の精度向上が不可欠であることが示唆された。

今後の課題としては、実機実験を視野に入れ、アーチ型全可動外殻を採用したバンクローリングによる操舵性能の向上を目的とした改良型 Rollopod-B（図 7.1）の開発や、ロボットの動作に追従可能な 3 自由度クレーンを備えた可変疑似重力・模擬地形実験装置（図 7.2）の設計が挙げられる。また、コントローラに関しては、歩行、ローリング、モード切替（転倒復帰を含む）を統一制御可能な深層強化学習ベースのコントローラの研究を進める必要がある。

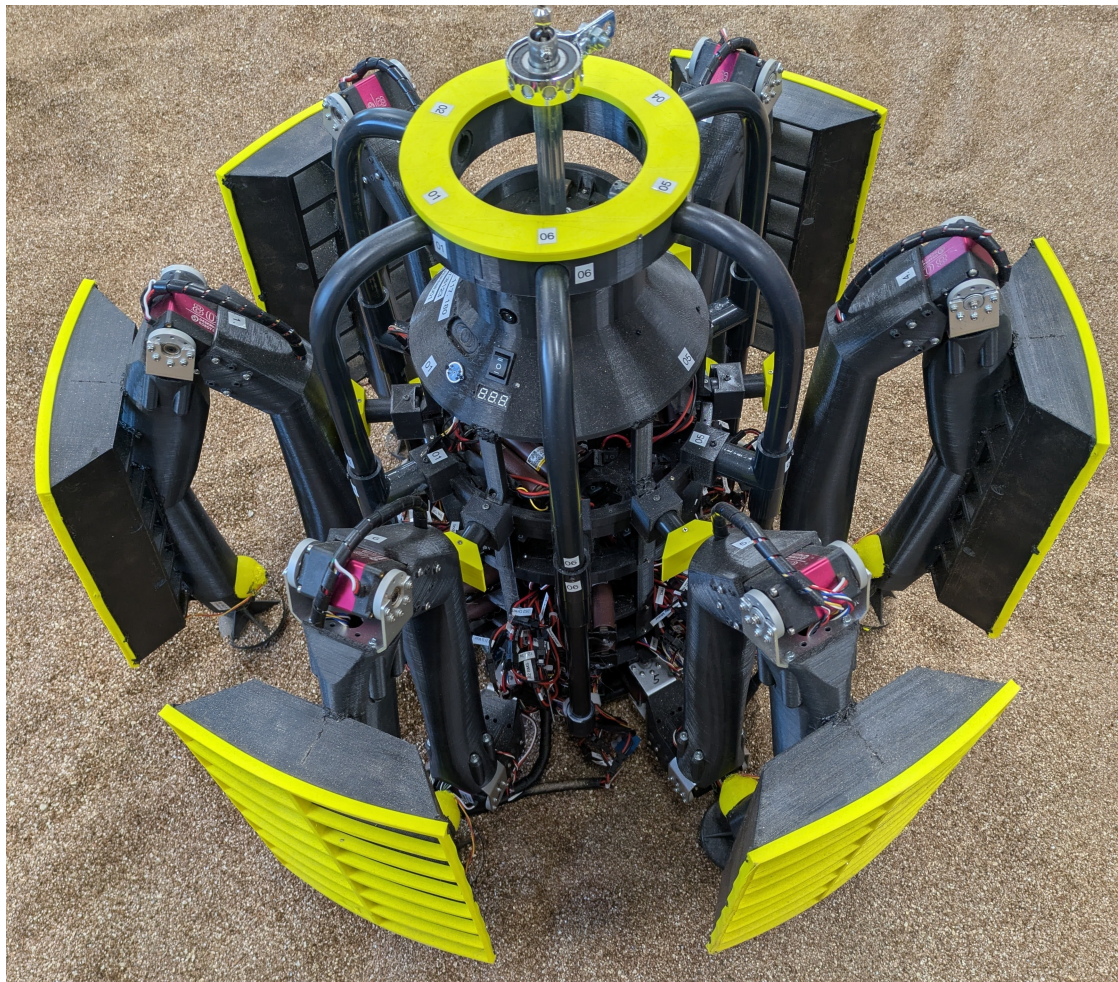


図 7.1: Rollopod-B のプロトタイプ

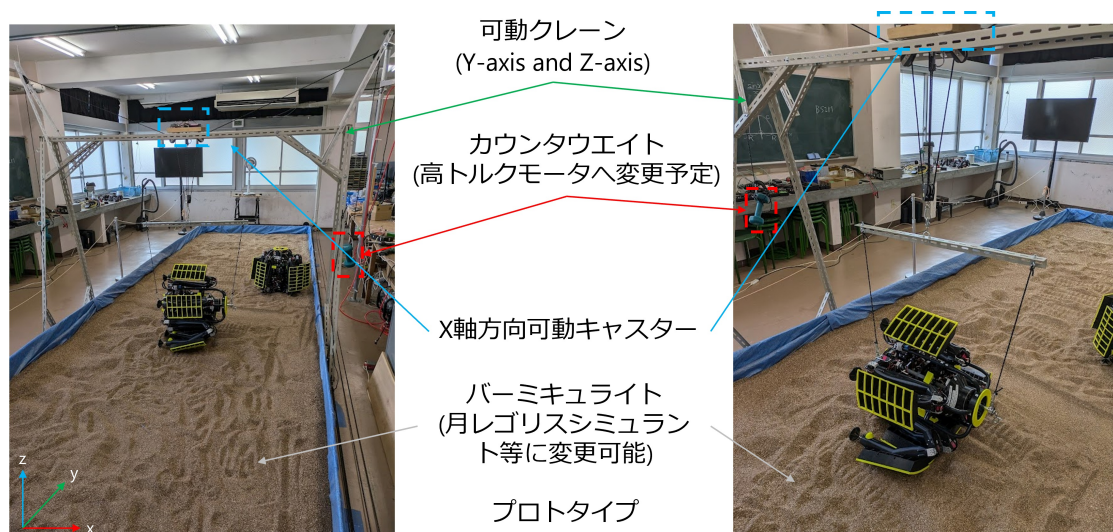


図 7.2: 可変疑似重力・模擬地形実験装置プロトタイプ

# 謝辞

本研究を進めるにあたり、多大なご指導とご助言をいただいた Chong 教授に深く感謝申し上げます。また、ロボット製作において多大なるご協力を賜りました日本文理大学工学部機械電気工学科の伊藤教授ならびに伊藤研究室の皆様に、心より感謝申し上げます。さらに、日々の実験の補佐を担当して頂きました萩さんにも深く感謝の意を表します。

ここに記しきれない多くの方々のご支援とご協力により、本研究を無事進めることができましたことを、重ねて御礼申し上げます。



## 参考文献

- [1] Amazon. “Amazon announces 2 new ways it’s using robots to assist employees and deliver for customers”. 参照日: 2025 年 1 月 11 日. 2023. URL: <https://www.aboutamazon.com/news/operations/amazon-introduces-new-robotics-solutions>.
- [2] TOYOTA L&A. “シンプル AGV キーカートによる、部品搬送の自動化で、生産効率の向上と作業負担の軽減を実現。”. 参照日: 2025 年 1 月 21 日. URL: <https://logi.toyota-lf.com/case/case015/>.
- [3] 国土交通省. “国土交通省告示第九百七十四号（令和六年六月二十八日）”. 参照日: 2025 年 1 月 11 日. 2024. URL: <https://www.mlit.go.jp/jutakukentiku/build/content/001753382.pdf>.
- [4] デジタル庁. “アナログ規制に係る工程表及び通知・通達の見直し方針のフォローアップ取組状況（2024 年 9 月 10 日時点）”. 参照日: 2025 年 1 月 11 日. 2024. URL: [https://www.digital.go.jp/assets/contents/node/basic\\_page/field\\_ref\\_resources/34a225ed-03be-4408-b00d-f9b88a5a2543/7590759e/20240912\\_policies\\_digital-extraordinary-administrative-research-committee\\_01.pdf](https://www.digital.go.jp/assets/contents/node/basic_page/field_ref_resources/34a225ed-03be-4408-b00d-f9b88a5a2543/7590759e/20240912_policies_digital-extraordinary-administrative-research-committee_01.pdf).
- [5] 東京電力ホールディングス. “福島第一原子力発電所事故の経過と教訓”. 参照日: 2025 年 1 月 21 日. URL: <https://www.tepco.co.jp/nu/fukushima-np/outline/>.
- [6] 朝日新聞. “吉田調書：第 3 章 ヒトが止められるか”. 参照日: 2025 年 1 月 21 日. URL: [http://www.asahi.com/special/yoshida\\_report/3-1.html](http://www.asahi.com/special/yoshida_report/3-1.html).
- [7] Marco Hutter, Christian Gehring, Mark A. Höpflinger, Michael Blösch, Roland Siegwart. “Toward Combining Speed, Efficiency, Versatility, and Robustness in an Autonomous Quadruped”, *IEEE Transactions on Robotics*, Vol. 30, Issue 6, pp. 1427-1440, 2014. URL: <https://doi.org/10.1109/TR0.2014.2360493>.
- [8] 防衛日報デジタル. “【能登半島地震】陸自、ロボット犬を被災地に導入 被災者誘導などで活用 | 陸上自衛隊”. 参照日: 2025 年 1 月 21 日. URL: [https://dailydefense.jp/\\_ct/17679459](https://dailydefense.jp/_ct/17679459).

- [9] Marco Hutter, Christian Gehring, Dominic Jud, Andreas Lauber, C. Dario Bellicoso, Vassilios Tsounis, Jemin Hwangbo, Karen Bodie, Peter Fankhauser, Michael Bloesch, Remo Diethelm, Samuel Bachmann, Amir Melzer, Mark Hoepffinger. “*ANYmal - a highly mobile and dynamic quadrupedal robot*”, *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 38-44, 2016. URL: <https://doi.org/10.1109/IROS.2016.7758092>.
- [10] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, Marco Hutter. “*Learning quadrupedal locomotion over challenging terrain*”, *Science Robotics*, Vol. 5, No. 47, 2020. 2020. URL: <https://doi.org/10.1126/scirobotics.abc5986>.
- [11] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, Marco Hutter. “*Learning robust perceptive locomotion for quadrupedal robots in the wild*”, *Science Robotics*, Vol. 7, No. 62, 2022. URL: <https://doi.org/10.1126/scirobotics.abk2822>.
- [12] Defense Advanced Research Projects Agency. *Subterranean Challenge Final Event*. 参照日: 2025 年 1 月 11 日. URL: <https://www.darpa.mil/research/challenges/subterranean>.
- [13] PUDU. “*PUDU D9: The First Full-sized Bipedal Humanoid Robot by Pudu Robotics*”. 参照日: 2025 年 1 月 21 日. URL: <https://www.pudurobotics.com/jp/products/d9>.
- [14] BostonDynamics. “*Spot-The Agile Mobile Robot*”. 参照日: 2025 年 1 月 21 日. URL: <https://bostondynamics.com/products/spot/>.
- [15] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth and G. Pratt. “*Virtual model control: An intuitive approach for bipedal locomotion*”, *Int. J. Robot. Res.*, Vol. 20, No. 2, pp. 129-143, 2001. URL: <https://doi.org/10.1177/02783640122067309>.
- [16] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan and S. Schaal. “*Optimal distribution of contact forces with inverse-dynamics control*”, *Int. J. Robot. Res.*, Vol. 32, No. 3, pp. 280-298, 2013. URL: <https://doi.org/10.1177/0278364912469821>.
- [17] Junzhi Yu, Min Tan, Jian Chen, Jianwei Zhang. “*A Survey on CPG-Inspired Control Models and System Implementation*”, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 25, Issue 3, pp. 441-456, 2014. URL: <https://doi.org/10.1109/TNNLS.2013.2280596>.

- [18] Evan Ackerman. “*Boston Dynamics introduces a fully electric humanoid robot that “exceeds human performance”*”, *IEEE Spectrum*, 2024. 参照日: 2025 年 1 月 11 日. URL: <https://spectrum.ieee.org/atlas-humanoid-robot>.
- [19] Alexander Spiridonov, Fabio Buehler, Moriz Berclaz, Valerio Schelbert, Jorit Geurts, Elena Krasnova, Emma Steinke, Jonas Toma, Joschua Wuethrich, Recep Polat, Wim Zimmermann, Philip Arm, Nikita Rudin, Hendrik Kolvenbach, Marco Hutter. “*SpaceHopper: A Small-Scale Legged Robot for Exploring Low-Gravity Celestial Bodies*”, *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3464-3470, 2024. URL: <https://doi.org/10.1109/ICRA57147.2024.10610057>.
- [20] Uluc Saranlı, Martin Buehler, Daniel E. Koditschek. “*RHex: A simple and highly mobile hexapod robot*”, *The International Journal of Robotics Research*, Vol. 20, No. 7, pp. 616-631, 2001. URL: <https://repository.upenn.edu/server/api/core/bitstreams/8f050c46-0ade-4471-a993-f7bd62147170/content>.
- [21] Sangok Seok, Albert Wang, Meng Yee Chuah, Dong Jin Hyun, Jongwoo Lee, David M. Otten, Jeffrey H. Lang, Sangbae Kim. “*Design Principles for Energy-Efficient Legged Locomotion and Implementation on the MIT Cheetah Robot*”, *IEEE/ASME Transactions on Mechatronics*, Vol. 20, Issue 3, 2015. URL: <https://doi.org/10.1109/TMECH.2014.2339013>.
- [22] Marko Bjelonic, C. Dario Bellicoso, M. Efe Tiryaki, Marco Hutter. “*Skating with a Force Controlled Quadrupedal Robot*”, *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp.7555-7561, 2018. URL: <https://doi.org/10.1109/IROS.2018.8594504>.
- [23] Jiawei Chen, Ripeng Qin, Longfei Huang, Zongbo He, Kun Xu, Xilun Ding. “*Unlocking Versatile Locomotion: A Novel Quadrupedal Robot with 4-DoFs Legs for Roller Skating*”, *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8037-8043, 2024. URL: <https://doi.org/10.1109/ICRA57147.2024.10610706>.
- [24] Marko Bjelonic, C. Dario Bellicoso, Yvain de Viragh, Dhionis Sako, F. Dante Tresoldi, Fabian Jenelten, Marco Hutter. “*Keep Rollin’ — Whole-Body Motion Control and Planning for Wheeled Quadrupedal Robots*”, *IEEE Robotics and Automation Letters*, Vol. 4, Issue 2, pp. 2116-2123, 2019. URL: <https://doi.org/10.1109/LRA.2019.2899750>.

- [25] Vivian S. Medeiros, Edo Jelavic, Marko Bjelonic, Roland Siegwart, Marco A. Meggiolaro, Marco Hutter. “*Trajectory Optimization for Wheeled-Legged Quadrupedal Robots Driving in Challenging Terrain*”, *IEEE Robotics and Automation Letters*, Vol. 5, Issue 3, pp. 4172-4179, 2020. URL: <https://doi.org/10.1109/LRA.2020.2990720>.
- [26] Gye-Do Park, Hyun Lee, Kyoung-Hwan Kim, Jang-Myung Lee. “*The method of roll maintain for spherical robot of torque driven type from the external impact*”, *2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. November 2011. URL: <https://doi.org/10.1109/URAI.2011.6145873>.
- [27] Wenchuan Jia, Zhongshu Huang, Yi Sun, Huayan Pu, Shugen Ma. “*Toward a novel deformable robot mechanism to transition between spherical rolling and quadruped walking*”, *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1539-1544, 2017. URL: <https://doi.org/10.1109/ROBIO.2017.8324636>.
- [28] Ning Tan, Rajesh Elara Mohan, and Karthikeyan Elangovan. “*Scorpio: A biomimetic reconfigurable rolling-crawling robot*”, *International Journal of Advanced Robotic Systems* Volume 13, Issue 5. 参照日: 2025 年 1 月 6 日. September 2016. URL: <https://doi.org/10.1177/1729881416658180>.
- [29] Takeshi Miura, Shoshiro Hatakeyama, Masami Iwase. “*Development of Wheel-Spider-Inspired Hexapod Robot Realizing Walking and Rolling Locomotion*”, *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)*, pp.174-178, 2019. URL: <https://doi.org/10.1109/ICCMA46720.2019.8988636>.
- [30] NVIDIA. “*Omniverse IsaacSim*”. 参照日: 2025 年 1 月 11 日. URL: <https://docs.omniverse.nvidia.com/isaacsim/latest/index.html>.
- [31] NVIDIA. “*Isaac Lab Documentation*”. 参照日: 2025 年 1 月 11 日. URL: <https://isaac-sim.github.io/IsaacLab/main/index.html>.
- [32] 安藤和彦, 倉持智明. “*路面のすべり摩擦と路面管理水準及びすべり事故*”, *土木技術資料* 52-5(2010), pp. 56-59, 2010.
- [33] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov. “*Proximal Policy Optimization Algorithms*”, *arXiv:1707.06347*, 2017. 参照日: 2025 年 1 月 13 日. 2017. URL: <https://arxiv.org/abs/1707.06347>.
- [34] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, Pieter Abbeel. “*Trust Region Policy Optimization*”, *arXiv:1502.05477*, 2015. 参照日: 2025 年 1 月 13 日. 2015. URL: <https://arxiv.org/abs/1502.05477>.

- [35] A.D. Kuo. “ *Choosing Your Steps Carefully* ” , *IEEE Robotics Automation Magazine*, Vol. 14, Issue 2, 2007. URL: <https://doi.org/10.1109/MRA.2007.380653>.