

# 64A 深層学習を用いた二足歩行ロボットの両足動作生成アルゴリズムの開発

## 1. 緒言

従来の二足歩行ロボットの制御手法は、力学モデルに基づくアプローチが主流である。これらの手法では、ロボットの各部位の運動を力学的に解析し、そのモデルに基づいて適切な制御入力を導き出している。しかし、このような力学ベースの手法には、環境の変化や不確実性に対する適応能力の限界があり、柔軟かつ効率的な歩行の実現には限界がある。

そこで、本研究では深層学習を活用することで、従来の力学モデルに依存しない、センサデータを基にしたより安定した歩行実現を目的とする。

## 2. 二足歩行ロボットの制御

### 2-1 ロボットの概要

シミュレーションを行うためのロボットモデルは本研究室で製作された二足歩行ロボットと同じものを使用した[1]。ロボットの外見を図1に示す。このロボットは7つのリンクと6つのジョイントで構成されており、各ジョイントには回転軸と正回転方向が設定されている。また、ジョイントにトルク値を与えることで、ロボットが動作するようになっている。

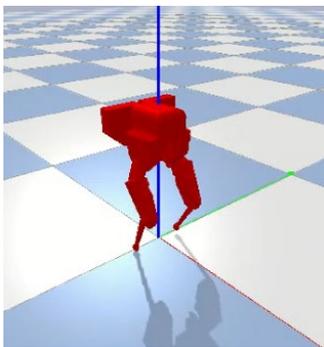


図 1 二足歩行ロボットの外見

### 2-2 力学計算によるロボット制御概要

二足歩行ロボットの歩行制御には、モデル予測制御(MPC)を用いた手法が有効であることが示されている[2]。歩行制御のブロック線図を図2に示す。DCM-MPC では振り足の着地位置 $U_t$ と着地までの時間 $T$ の計算を行う。計算には歩行変数、振り足の着地に必要な最小時間 $T_{min}$ 、重心位置や重心速度といったロボットのセンサデータ $\xi_{mea}$ を用いる。歩行変数にはxy平面における足先の位置、足が地面から離れてから経過した時間などが含まれる。求めた足の位置 $U_t$ と着地時間 $T$ から、Swing-MPC では振り足を目標位置に動かすための両足動作の計算を行う。WBC ではロボットの各関節に適用するトルクの計算を行う[3]。計算にはSwing-MPC で求めた軌跡、位置や関節角度といったロボットからのセンサデータ $x_{mea}$ を用いる。ロボットは計算されたトルク値に従って動作し、その時の足先の位置や速度、重心位置といったセンサデータがフィードバックされる。

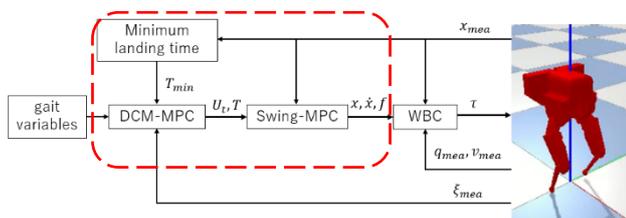


図 2 力学制御のブロック線図

### 2-3 深層学習によるロボット制御概要

既存のロボット制御プログラムにおいて、深層学習を導入する部分を検討するために調査を行った。結果、図2のブロック線図の赤点線内に示した Minimum landing time, DCM-MPC, Swing-MPC は制御プログラムにおいて一つのライブラリにまとめられていることが判明した。本研究ではこのライブラリを深層学習モデルに置き換えることで両足動作の生成を試みる。

## 3. 学習データ作成

### 3-1 環境構築

OSは開発環境の構築の容易さという利点からUbuntuを使用した。また、プログラミング言語にはPython 3.8を使用した。プログラムで使用した主なライブラリは、深層学習用の tensorflow 2.10 と keras 2.10、ロボットシミュレーションの物理エンジンで使用する pybullet 3.2 である。

### 3-2 データ収集方法

学習データを収集するためのプログラムは、モデル予測制御による二足歩行ロボットのシミュレーションプログラムを基に作成した[4]。作成したプログラムはロボットの歩行データを収集し、深層学習に使用するセンサデータを得ることを目的としている。図3に、このプログラムのフローチャートを示し、各ステップについて詳細に説明する。

プログラムが開始されると、最初に使用するライブラリの読み込み、必要な変数や定数の設定、および収集したデータを格納するためのリストが作成される。その後、ロボットは与えられた目標速度に従って歩行を行うが、最初は目標速度[0, 0, 0]が設定され、ロボットは原点でその場歩きを行う。

次に、さまざまな歩行データを収集するため、ロボットにはランダムな動作を指示する必要がある。そのため、ロボットにランダムな時間現在の動作を続けた後、目標速度を更新する。このとき、z方向の速度は0に固定され、xy方向の速度は±0.2 m/sの範囲でランダムに設定される。これにより、ロボットは動作中に目標速度に従い、さまざまな歩行パターンを生成する。

ロボットはその後、1分間にわたってランダムに歩行を行い、その間にセンサデータを収集する。このプロセスは、ロボットが原点に戻されて再度データ収集を開始するまで続く。この一連のデータ収集は60回繰り返され、合計で大量の歩行データが収集されることになる。

データ収集が完了すると、すべてのセンサデータはリストに格納され、その後 CSV ファイルとして保存される。

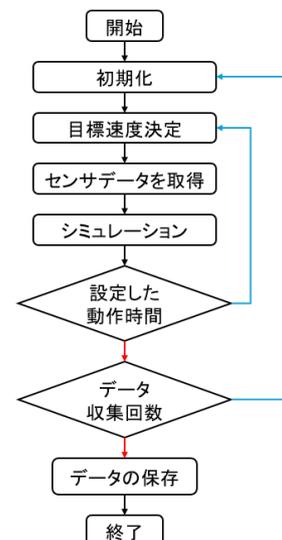


図 3 データ収集プログラムのフローチャート

### 3-3 学習データの詳細

収集するデータには入力データと出力データがある。入力データは、ロボットの動作を制御するために使用されるセンサデータである。収集するデータは、ロボットに与える目標速度、胴体の位置と姿勢、各ジョイントの角度、両足の足先位置、重心位置、加えてそれらの速度に関するものである。また、出力データは、WBC(全身制御コントローラー)に入力される制御値である。収集するデータは、振り足に作用する力、次の両足の足先位置とその速度、接地足と振り足を判別するものである。

## 4. 学習モデル

### 4-1 モデルの詳細

本研究で用いた深層学習モデルは Dense 層と Dropout 層が交互になった単純な DNN モデルである。Dense 層のノード数は 1088 であり、使用した活性化関数は LeakyReLU(alpha=0.01)である。以下の図4にモデルの概略図を示す。使用した最適化法は Adam(learning\_rate=0.001)、損失関数は mean\_squared\_error、評価関数は mae である。

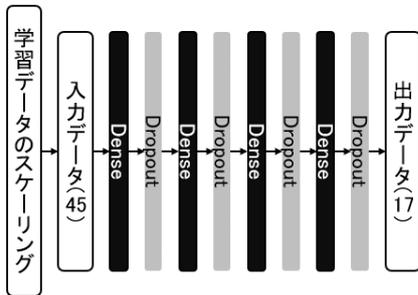


図 4 DNN モデルの概略図

### 4-2 モデルの学習結果

モデルのトレーニングでは、学習データのうち 20%を検証データとして用いた。また、バッチサイズを 64、エポック数を 100 と設定した。ただし、検証データにおける損失が 3 回連続で改善されない場合、トレーニングを終了し、最も良いトレーニングが出来たエポックでのモデルを保存した。訓練データと検証データにおける損失の推移を図5に示す。

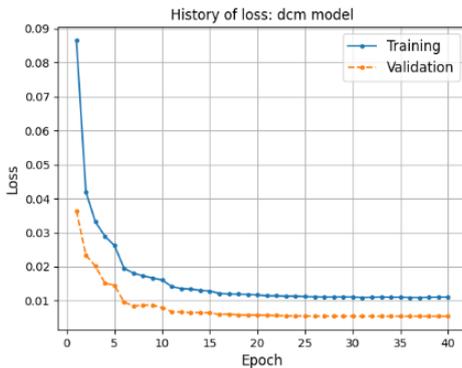


図 5 訓練データと検証データにおける損失の推移

## 5. 学習済みモデルによるシミュレーション

シミュレーションを行う際の学習済みモデルの読み込みでは、出力値の予測速度を向上させるため、tensorflowLite によるモデルの軽量化を行った。この軽量化によって、予測速度が 10 m/s から 1 m/s となり、よりリアルタイムでの制御が行いやすくなった。シミュレーションでは、キーボードでの操作によって、目標速度を与えた。目標速度を x 方向+0.2 m/s と設定した時の、ロボットの速度を図6、実際に移動している様子を図7に示す。図6、7より、与えた目標速度に応じてロボットが歩行していることが確認できる。

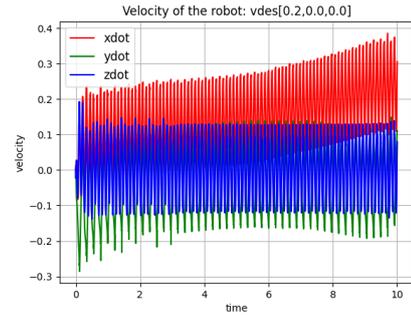


図 6 目標速度[0.2,0.0,0.0]の時のロボットの速度

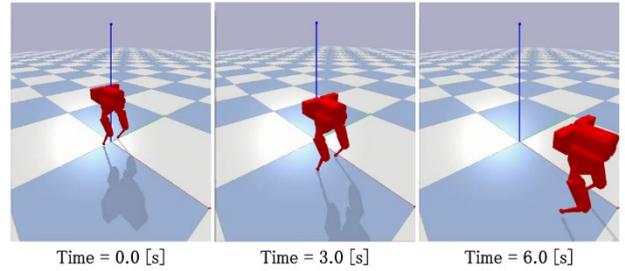


図 7 目標速度[0.2,0.0,0.0]の時のロボットの移動の様子

## 4. 結 言

本研究では、従来の力学モデルに基づく二足歩行ロボットの制御手法の限界を踏まえ、深層学習を活用した二足歩行ロボットの両足動作生成アルゴリズムの開発を行った。深層学習を用いることで、シミュレーション上での二足歩行ロボットの歩行制御を実現し、センサデータを基に安定した歩行動作を得ることが可能となった。本研究で得られた成果は、今後の二足歩行ロボットの設計や制御において、力学モデルに依存しない柔軟かつ効率的なアプローチを提供するものである。今後は、提案手法の実環境での適用や、WBC を含む深層学習モデル開発を計画している。

### <参考文献>

- [1] Maegaki, K., Alemayoh, T. T., Lee, J. H., & Okamoto, S. (2023). Actuator module development for a 3D printed biped robot using low-cost BLDC motor. *Proceedings of the Twenty-Eighth International Symposium on Artificial Life and Robotics 2023 (AROB 28th 2023)*, 815–819.
- [2] Daneshmand, E., Khadiv, M., Grimminger, F., & Righetti, L. (2021). Variable Horizon MPC with Swing Foot Dynamics for Bipedal Walking Control. *arXiv preprint arXiv:2010.08198*. <https://arxiv.org/abs/2010.08198>
- [3] Grimminger, F., Meduri, A., Khadiv, M., Viereck, J., Wüthrich, M., Naveau, M., Berenz, V., Heim, S., Widmaier, F., Flayols, T., Fiene, J., Badri-Spröwitz, A., & Righetti, L. (2020). An open torque-controlled modular robot architecture for legged locomotion research. *IEEE Robotics and Automation Letters*, 5(2), 3650-3657. <https://doi.org/10.1109/LRA.2020.2976639>
- [4] machines-in-motion, reactive\_planners, Website, [https://github.com/machines-in-motion/reactive\\_planners/blob/master/demos/demo\\_reactive\\_planners\\_bolt\\_step\\_adjustment.py](https://github.com/machines-in-motion/reactive_planners/blob/master/demos/demo_reactive_planners_bolt_step_adjustment.py)

2025年 2月20日 (S13 - 64) A方式

## 卒業論文発表会

# 深層学習による二足歩行ロボットの 両足動作生成アルゴリズムの開発

工学部工学科知能システム学コース4回生

# 目次

1. 研究の背景・目的
2. 二足歩行ロボットの制御
3. 学習データ作成
4. 学習モデル
5. 学習済みモデルによるシミュレーション
6. まとめ

# 1. 研究の背景・目的

## ロボットの歩行制御

- ・力学ベースの手法が主流  
→環境の変化に対する柔軟な歩行に限界がある。
- ・深層学習の活用  
→センサデータを基に歩行を最適化するため、  
より安定した歩行を実現できる可能性がある。

深層学習を活用し、柔軟で適応性の高い歩行実現を目指す。

## 2. 二足歩行ロボットの制御

### 2.1. 二足歩行ロボットの概要

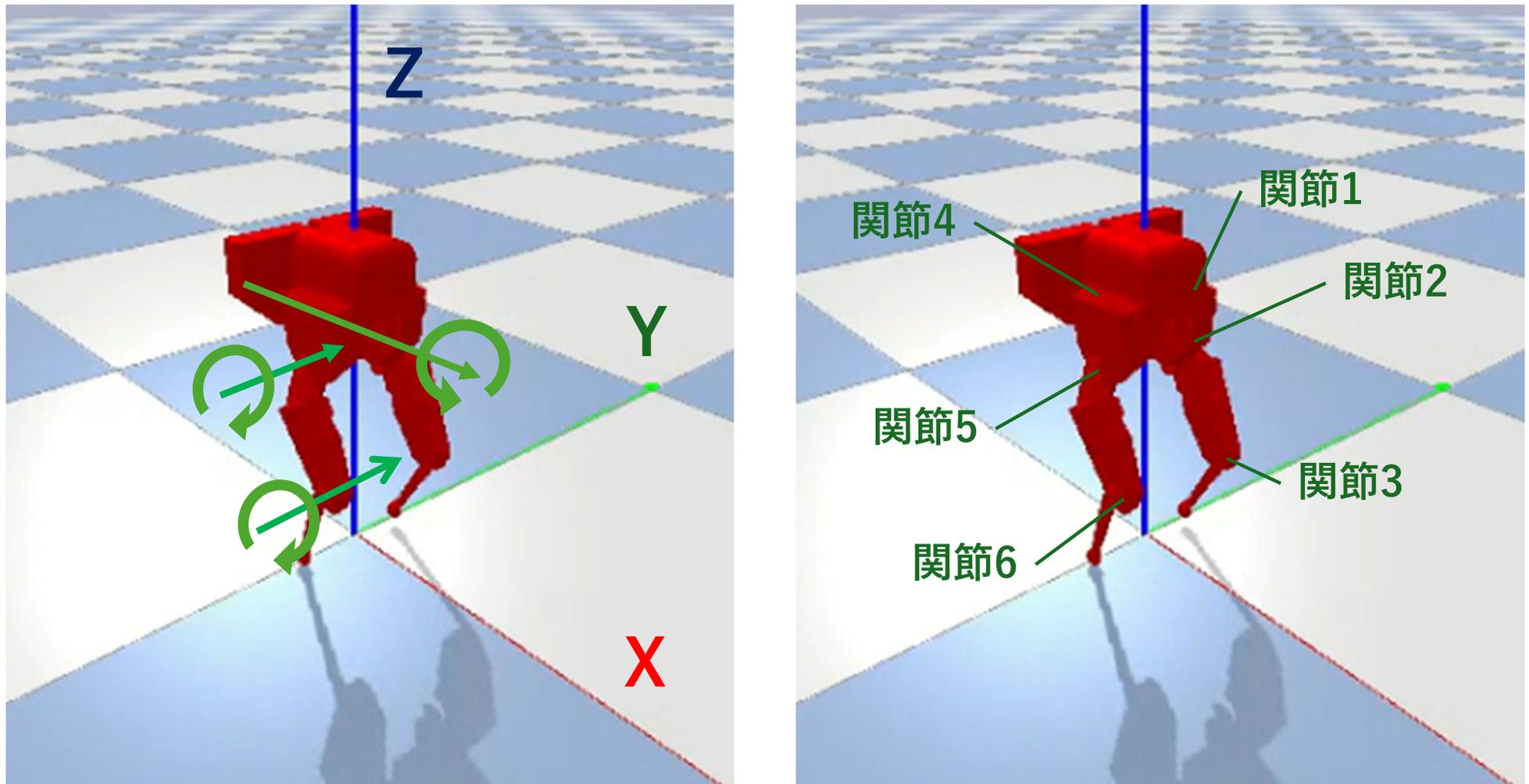


図2.1 二足歩行ロボットの外見・座標軸

## 2.2. 深層学習によるロボット制御概要

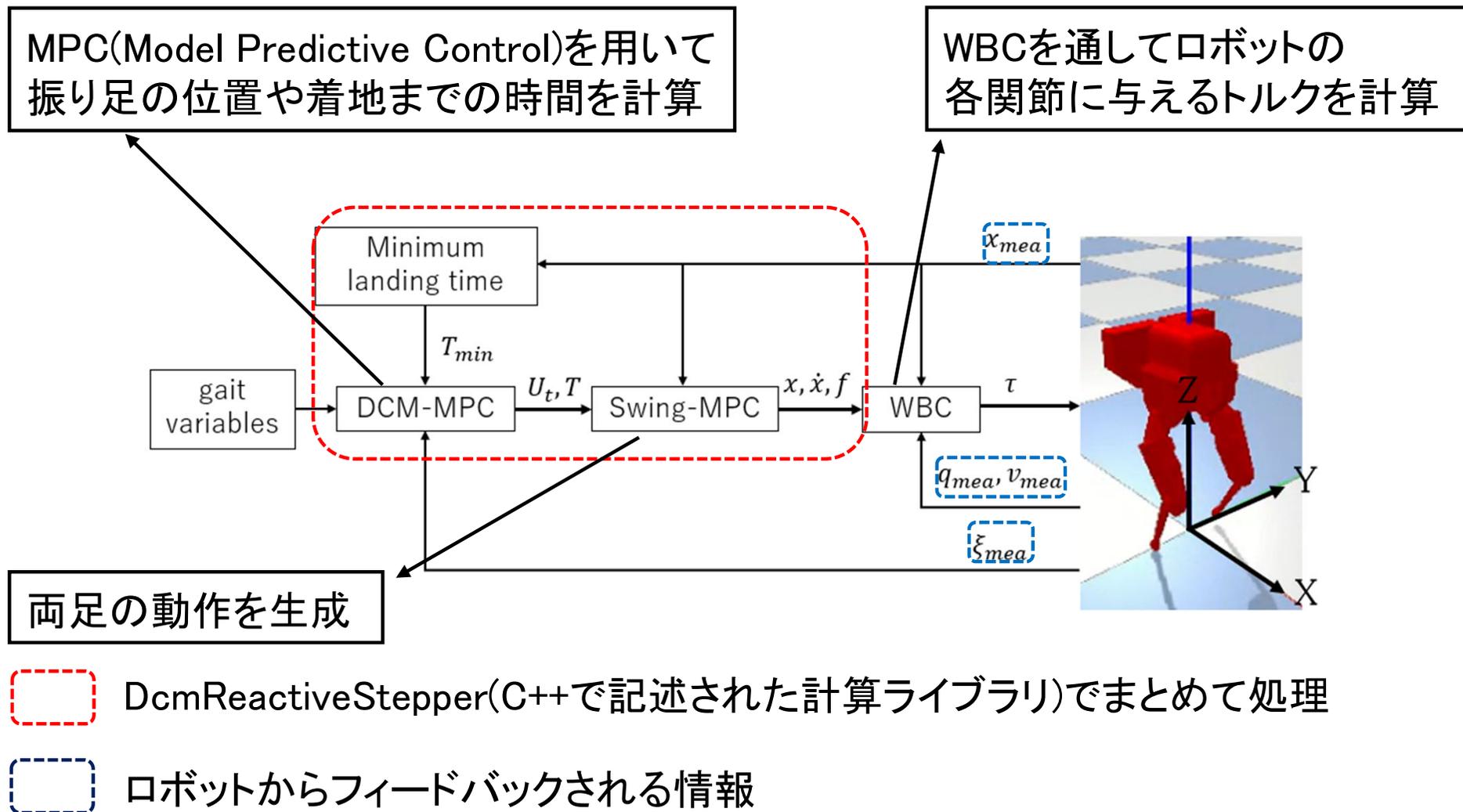


図2.2 ロボットの制御ブロック線図

### 3. 学習データ作成

#### 3.1. データ収集方法

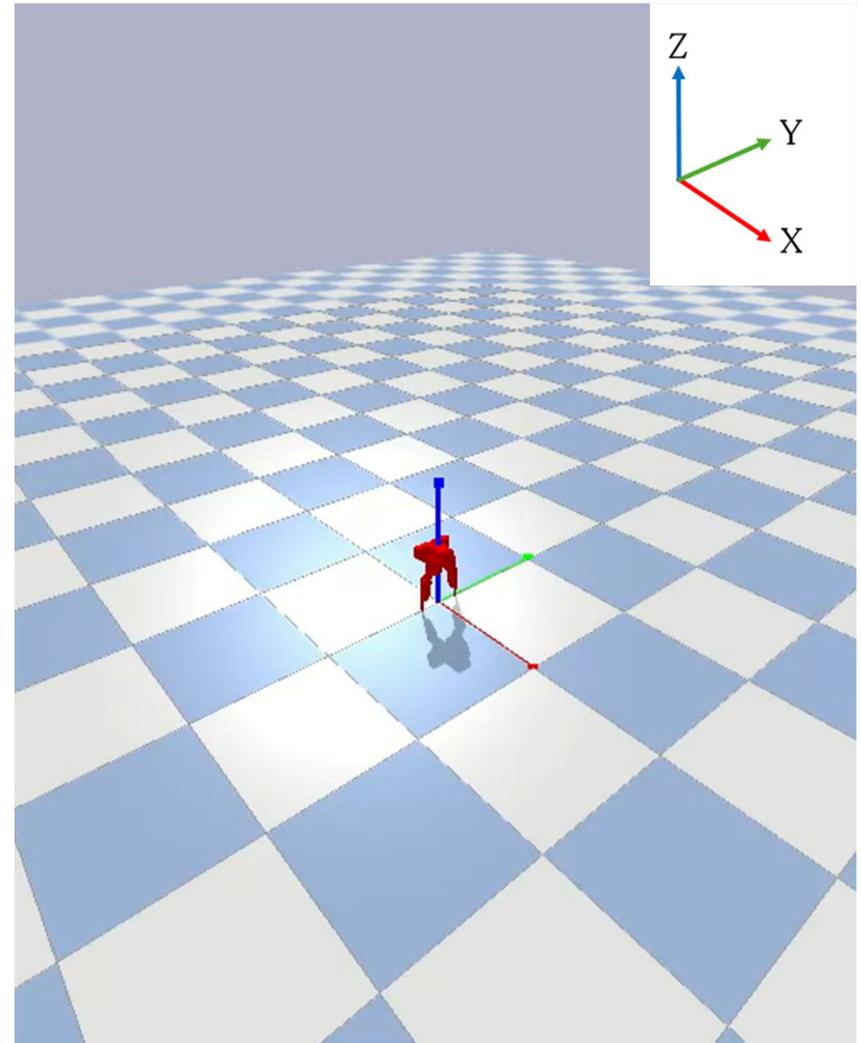
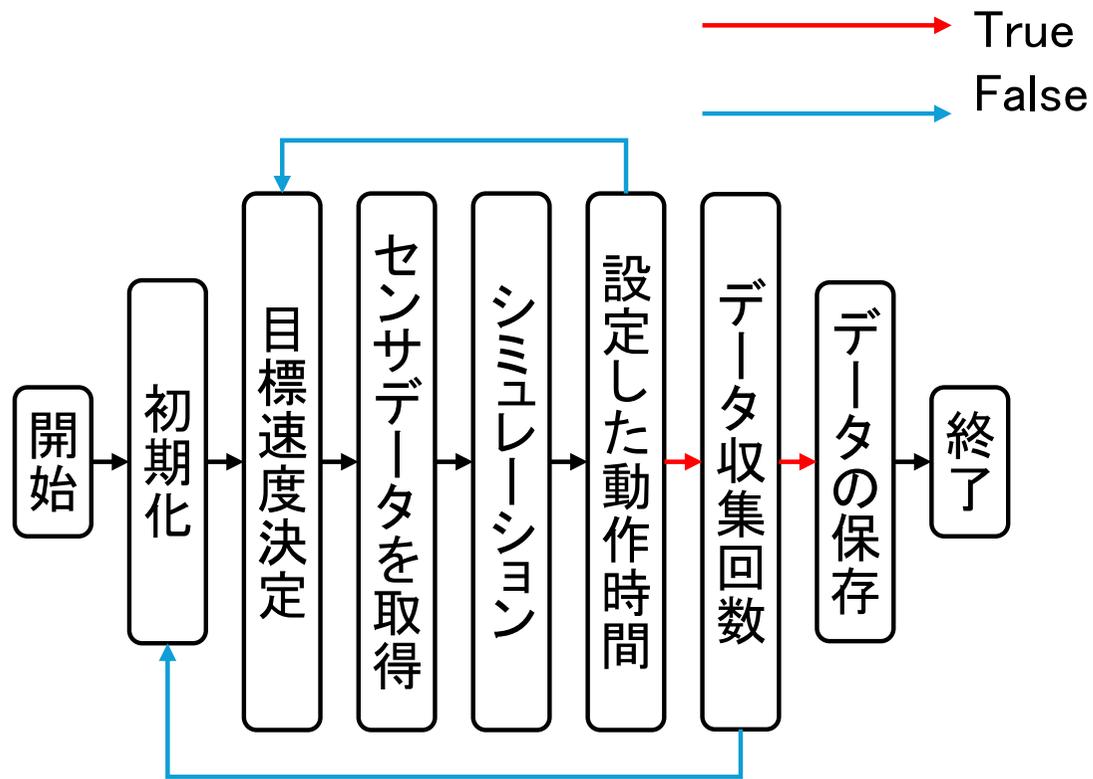


図3.1 データ収集プログラムのフローチャート

## 3.2. 学習データの詳細

収集データ数

入力:(3207473、45)

出力:(3207473、17)

### 入力データ(センサデータ)

- ・目標速度: 2
- ・関節の角速度: 6
- ・胴体の位置: 3
- ・足先の位置: 6
- ・胴体の角度: 4
- ・足先の速度: 6
- ・胴体の速度: 3
- ・重心の位置: 3
- ・胴体の姿勢: 3
- ・重心の速度: 3
- ・関節の角度: 6
- 合計: 45

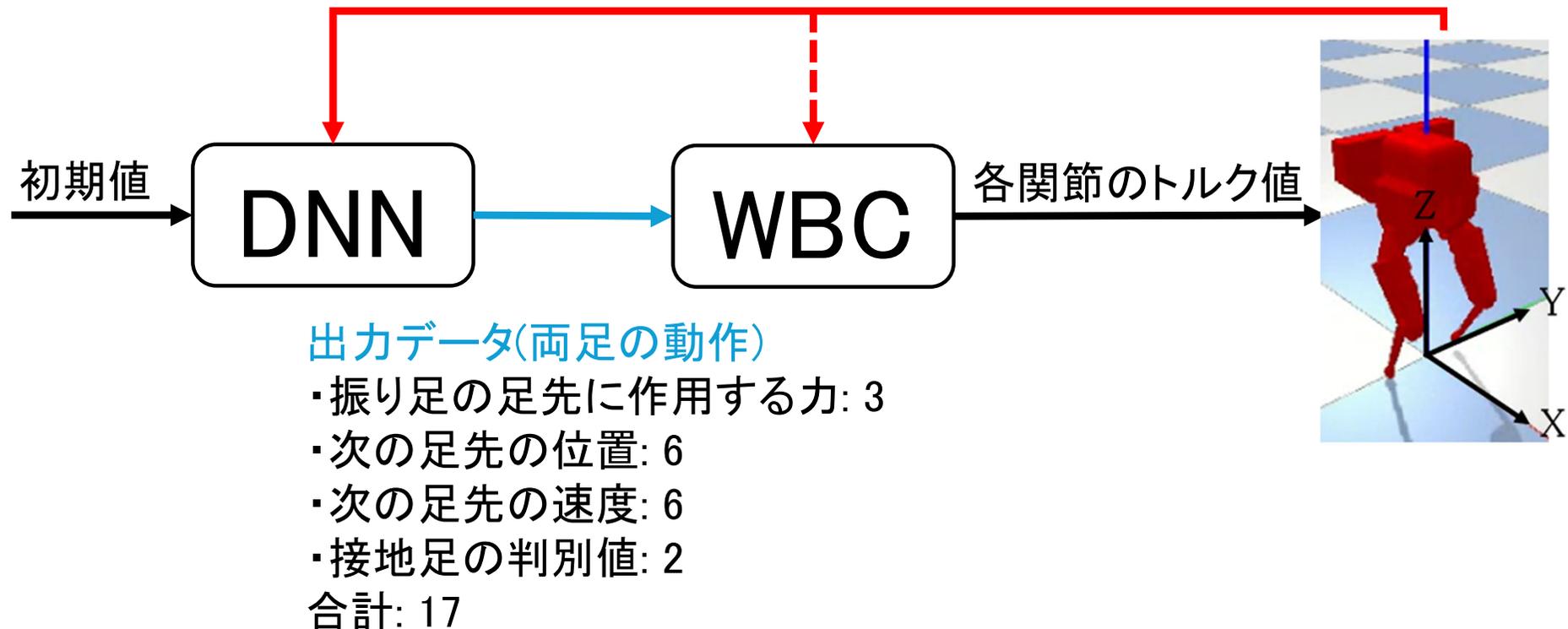
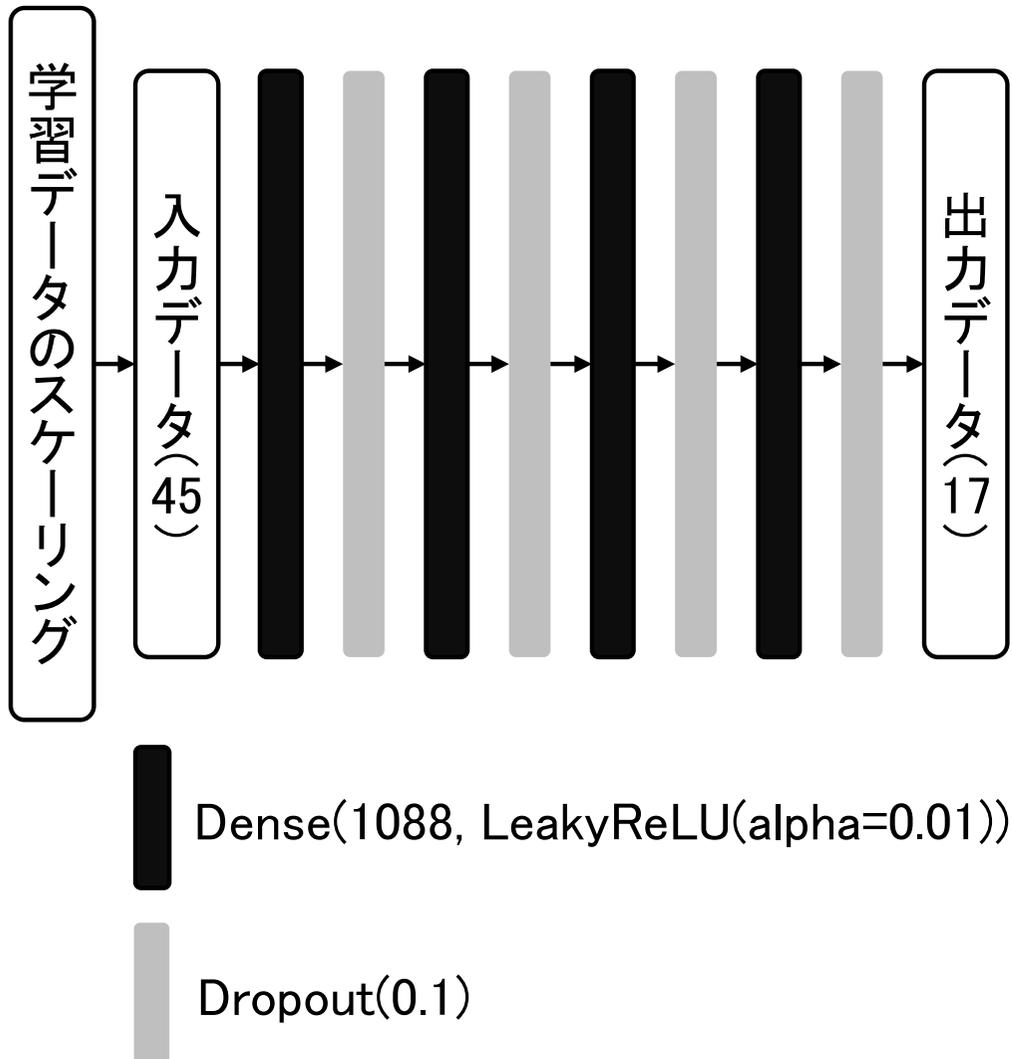


図3.2 収集データの詳細と深層学習による制御のブロック線図

## 4. 学習モデル

### 4.1. モデルの詳細



#### Compile

- optimizer: Adam(0.001)
- loss: mean\_squared\_error
- metrics: mae

#### Training

- epochs: 100
- batch\_size: 64
- validation\_split: 0.2

#### LearningRateScheduler

- lambda epoch:  $1e-4 * (0.4^{**}(\text{epoch} // 5))$

#### EarlyStopping

- monitor: val\_loss
- patience: 5
- restore\_best\_weights: True

図4.1 DNNモデル詳細

## 4.2. モデルの学習結果

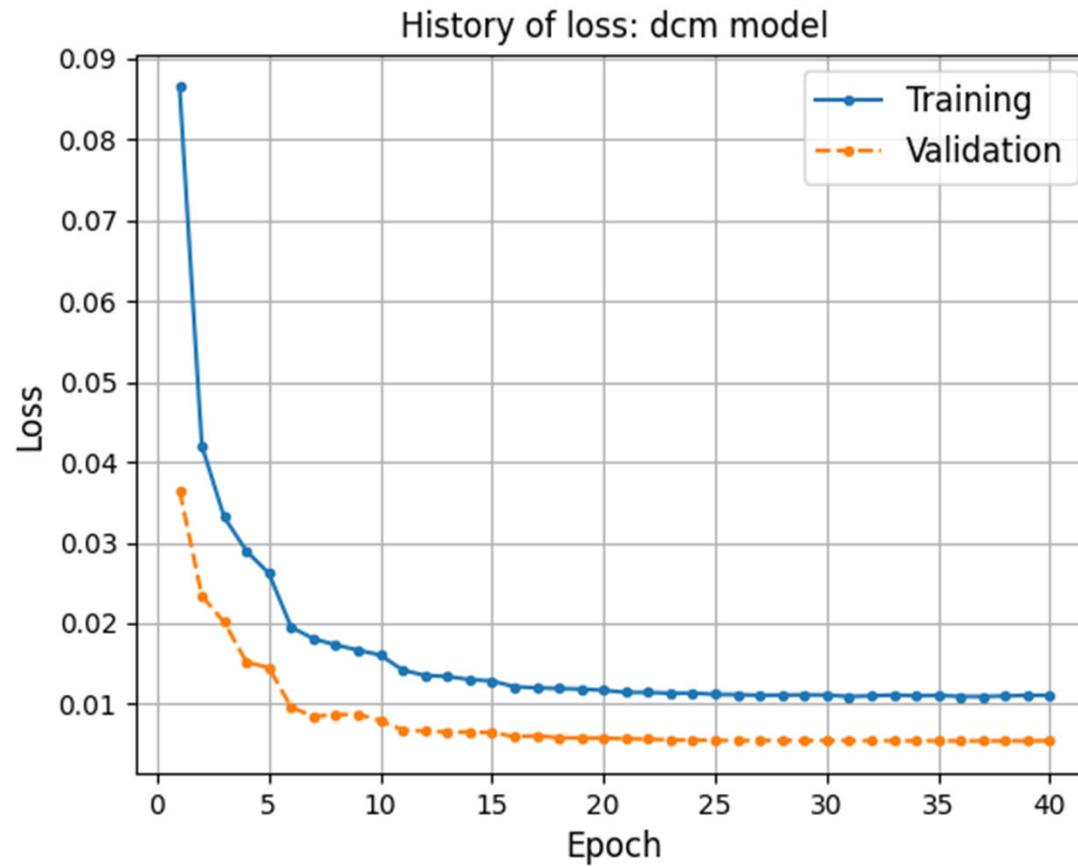


図4.2 lossの推移

## 5. 学習済みモデルによるシミュレーション

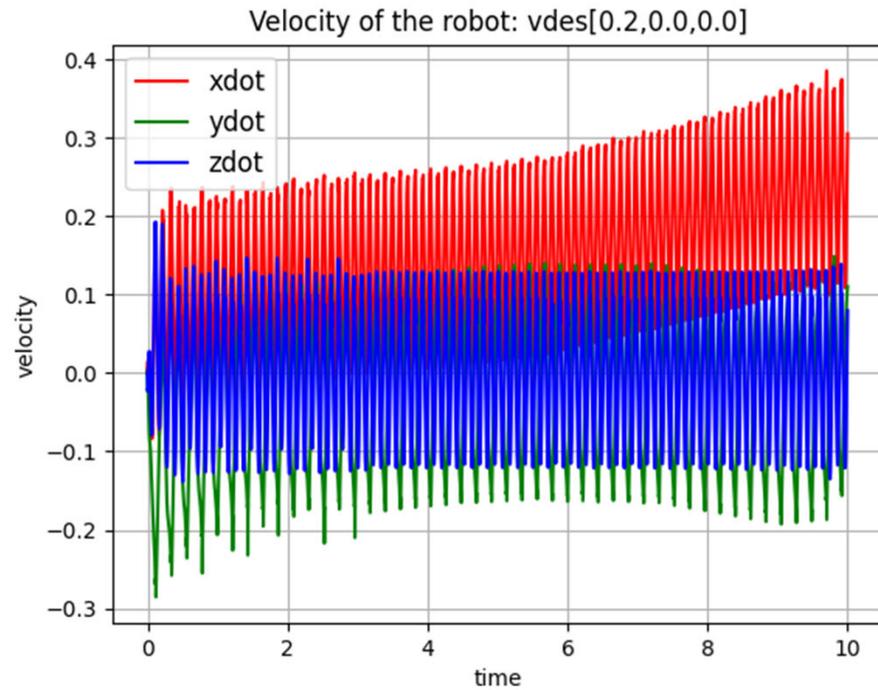


図5.1 目標速度[0.2,0.0,0.0]の時の  
ロボットの速度変化

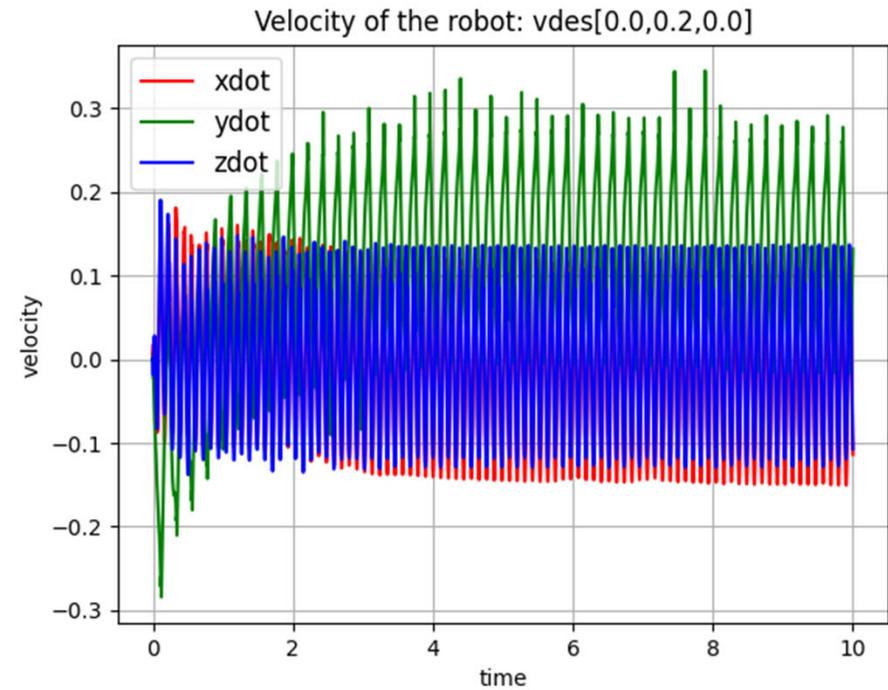
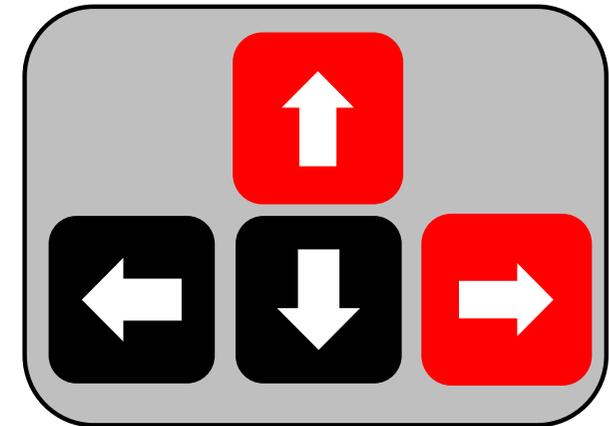
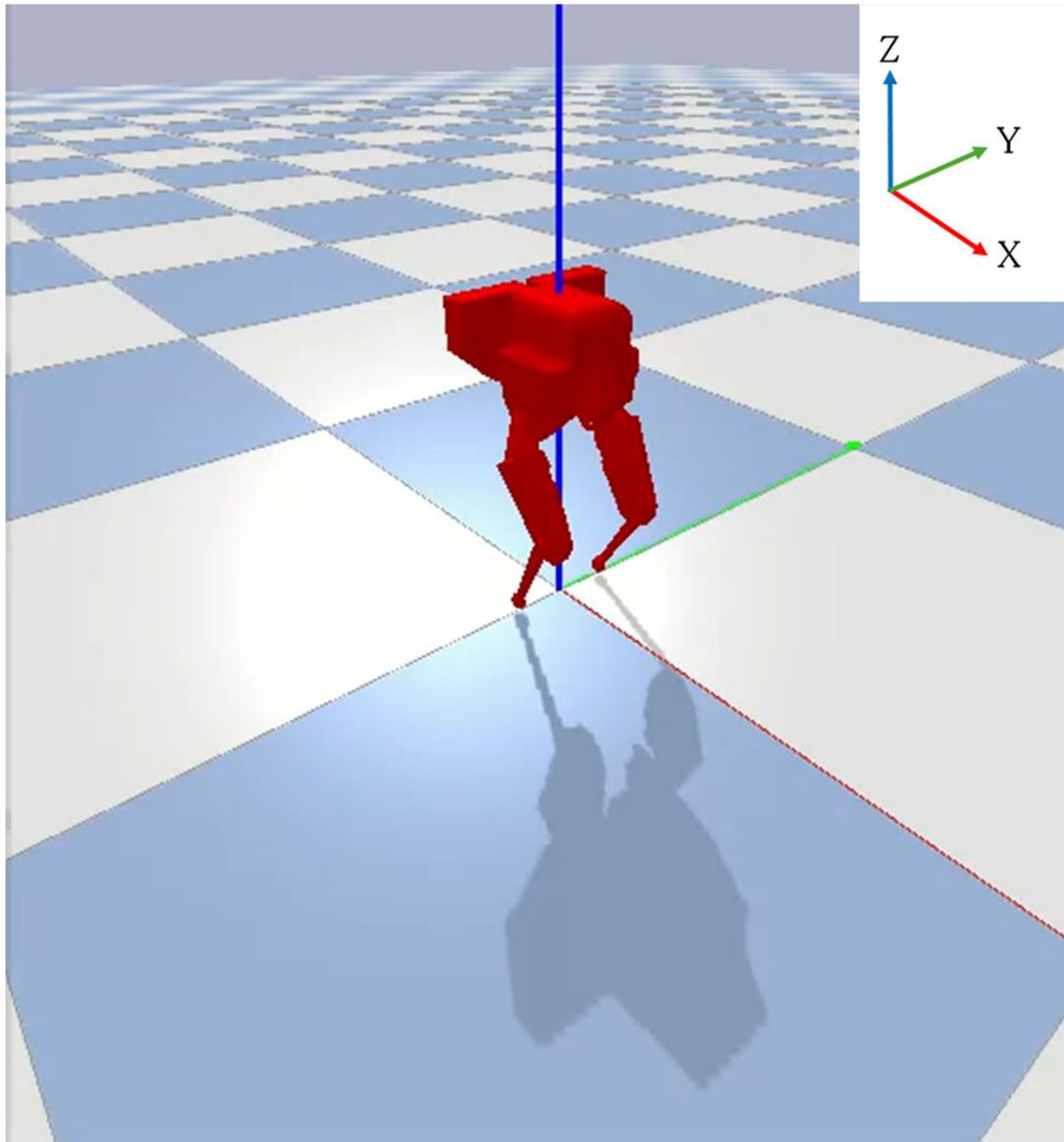


図5.2 目標速度[0.0,0.2,0.0]の時の  
ロボットの速度変化

## 5. 学習済みモデルによるシミュレーション



キーボードで  
目標速度を操作

図5.3 キーボード操作によるロボットのシミュレーション動画

## 6. まとめ

- ・深層学習を用いることで二足歩行ロボットの両足動作の生成アルゴリズムを開発した。
- ・ロボットに目標速度を与えることで、命令通りの歩行を行わせた。

### 今後の予定

- ・トルク予測モデルを開発し、制御をすべて深層学習で行う。
- ・実機を用いてロボットの歩行実験を行う。