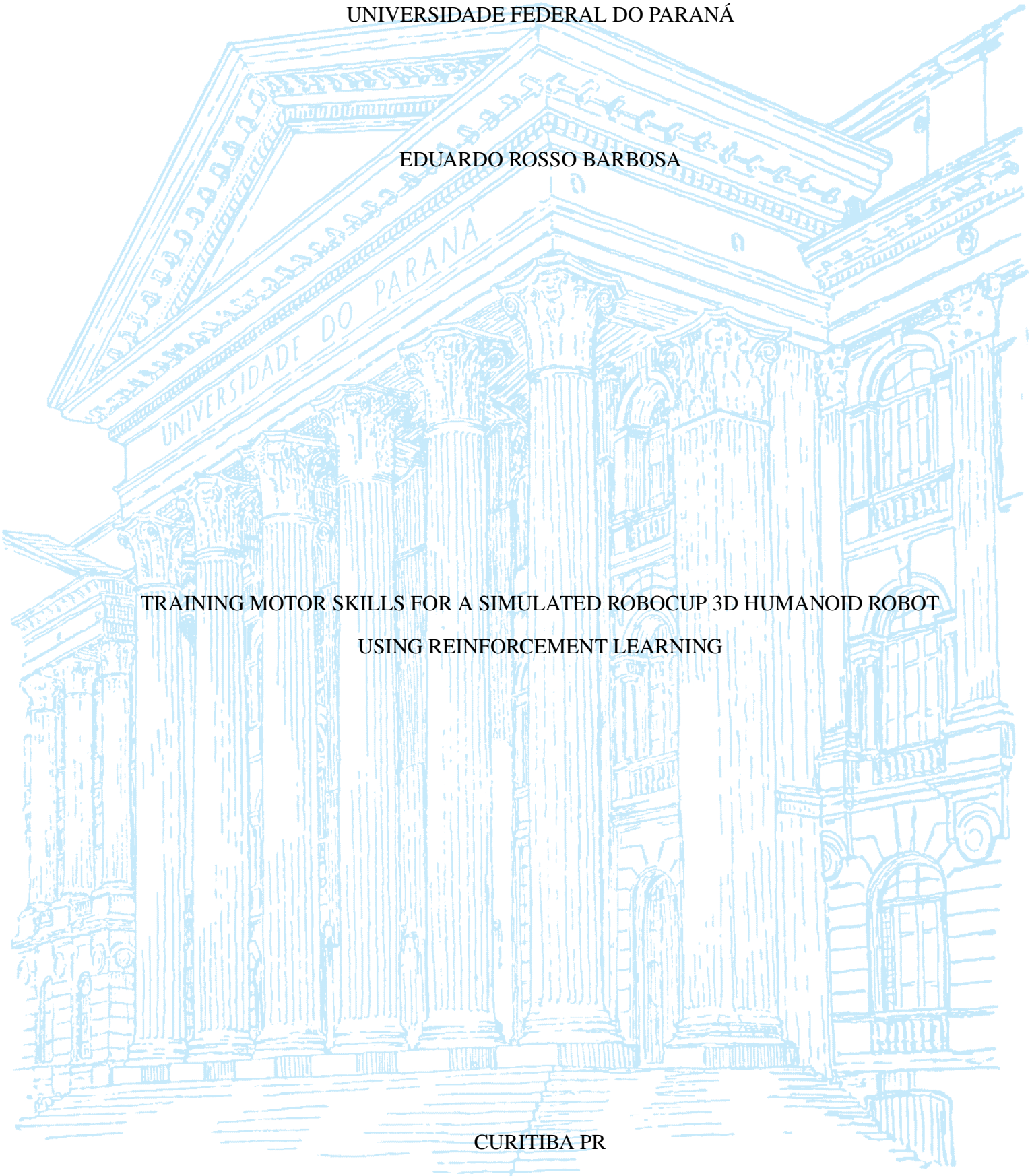
UNIVERSIDADE FEDERAL DO PARANÁ

EDUARDO ROSSO BARBOSA



TRAINING MOTOR SKILLS FOR A SIMULATED ROBOCUP 3D HUMANOID ROBOT USING REINFORCEMENT LEARNING

CURITIBA PR

2024

EDUARDO ROSSO BARBOSA

TRAINING MOTOR SKILLS FOR A SIMULATED ROBOCUP 3D HUMANOID ROBOT USING REINFORCEMENT LEARNING

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Computação*.

Orientador: Eduardo Todt.

CURITIBA PR

2024

RESUMO

A RoboCup Soccer 3D Simulation League fornece um ambiente rico além de desafios que incitam o avanço da área de inteligência robótica, sendo amplamente disponível para estudantes e desempenhando um papel significativo na educação. Neste estudo, abordamos o desafio de desenvolver um agente para controlar de maneira hábil um robô humanoide simulado na habilidade de driblar, utilizando o código aberto disponível do atual campeão mundial da liga, FCPortugal, como base, emparelhado com Aprendizado por Reforço Profundo e Aprendizado Curricular. Como o comportamento do agente treinado com nossas configurações é subótimo, discutimos os sucessos e as deficiências de nossos métodos e propomos melhorias para trabalhos futuros.

Palavras-chave: Controle Motor Robótico. Aprendizagem por reforço. Drible RoboCup

ABSTRACT

RoboCup Soccer 3D Simulation League provides a rich environment and challenges that incite the advance of state of the art of intelligent robots, while being widely available to students and plays a significant role in the robotics education. In this study we address the challenge of developing an agent to dexterously control a simulated humanoid robot in the skill of dribbling utilizing the current world champion in the league, FCPortugal's available open source code as a base paired with deep Reinforcement Learning and Curriculum Learning. As the trained agent behavior with our settings is suboptimal, we discuss the successes and shortcomings of our methods and propose improvements for future works.

Keywords: Robot Soccer. Deep Reinforcement Learning. RoboCup Gait.

LISTA DE ACRÔNIMOS

3DSSL	3D Soccer Simulation League
SSIM	Soccer Simulation
RL	Reinforcement Learning
ML	Machine Learning
MDP	Markov Decision Process
CL	Curriculum Learning
DoF	Degrees of Freedom
ZMP	Zero Moment Point
LIPM	Linear Inverted Pendulum Motion
DINF	Departamento de Informática
UFPR	Universidade Federal do Paraná

SUMÁRIO

1 INTRODUCTION

RoboCup is an international initiative estabilished in 1997 that promotes scientific advances on robotic intelligence through competition, the main goal is to have a team of robots winning a match of soccer against the human soccer world champions by 2050. Hence many obstacles must be overcomed like team strategy, comunication and cooperation in a collective level, but also indivual skills like gait, ballance, ball control, and many more examples.

To tackle different challanges, different RoboCup leagues, each with their level of abstraction, were created, the subject of our work is the 3D Soccer Simulation League (3DSSL) who provides a simulated enviroment and the standard humanoid robots, the real life counterpart model is utilized in the RoboCup Standard Plataform League.

The current league champion is the *FC Portugal* team, as it was shown in (Abreu et al., 2023) they were able to successfully train the agent in a set of skills that are usefull in a match, and also the high level strategy. Within the spirit of the RoboCup, the codebase is released for the public to improve and advance research.

The skills learned by the agent in this work are trained through Deep Reinforcement Learning (deep RL), which is a machine learning technique inspired by the idea of learning naturally by trial and error. Being in a simulated environment benefits the RL greatly since it can be accelerated, parallelized, and the environment be easily restarted at each attempt, just to name a few advantages.

The codebase for *FC Portugal* provides a strong foundation for developing new skills and behaviors, so it was used and modified to train the agent to achieve our goals.

1.1 OBJECTIVE

This works aims to generate a more natural Dribbling behavior of the the physically simulated robotic character of the RoboCup Simulated 3D League utilizing Deep Reinforcement Learning and Curriculum Learning as well as other classical complementary robotics algorithms.

To generate a gait and the desired dribbling behavior, we will be utilizing the open source code developed by FCPortugal's team as a starting point, taking advantage of the developed basic functions that are made available to the scientific community.

In summary, this study aims to setup the environment, train the model and analyze the result, proposing improvements.

1.2 STUDY OUTLINE

This study is divided in 6 chapters. Chapter 2 contains an overview on current literature of topics that will be important to the understanding of the work, such as the machine learning techniques utilized and the technologies closely related. Chapter 3 presents related work on the field of character animation on physically simulated environments and biped robotics in the context of RoboCup.

Chapter 4 introduced the proposed machine learning pipeline in detail in which the agents will be trained, as well as the reward design of each proposed curriculum, and Chapter 5 discusses the evaluation of the training techniques the resulting final trained model. Finally Chapter 6 summarizes this study by evaluating its achievements and proposes improvements for future works.

2 BACKGROUND

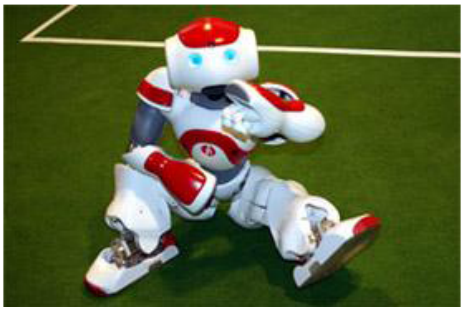
This chapter reviews the literature on the RoboCup Simulation environment and technologies as well as it introduces the machine learning algorithms used in this study.

2.1 ROBOCUP SOCCER SIMULATION - 3D LEAGUE

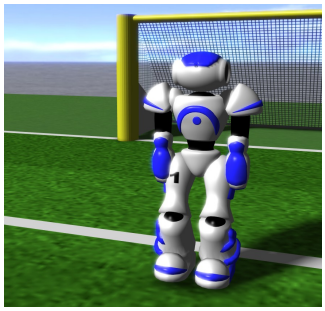
The RoboCup Soccer Simulation (SSIM) is subdivided in the 2D and 3D leagues (Federation, 2024; League, 2024), this study utilizes the environment and tools intended for the 3D league (3dT, 2024) such as the physical multiagent simulator: SimSpark (SimSpark, 2024a), the responsible for creating a server where the simulation runs: rcssserver3d (SimSpark, 2024b) and the monitor and visualizer tool: RoboViz (MagmaOffenburg, 2024).

2.1.1 NAO Robot Model

NAO is the name of the model made by Alderan Robotics (Robotics, 2024), the biped humanoid robot has about 57cm of heigh, weights 4.5Kg and it is a well known model by the academic community, it is the model used by SSIM 3D League and by the real life Standard League.



(a) The real NAO



(b) NAO Model Simulated

The simulated model has 22 degrees of freedom (DoF), a gyroscope and an accelerometer located at the center of the torso, a force resistance perceptor in each foot to indicate pressure, a visual perceptor at the center of the head, a say effector and a hear percetor for communication purposes, each hinge is represented by a hinge joint perceptor and manipulable through the corresponding hinge joint effector. (SimSpark-Aldebaran, 2024)

2.2 REINFORCEMENT LEARNING

Reinforcement Learning (RL) studies how an agent learns and improves through interactions with a given environment (Sutton e Barto, 2018). The agent discretely interacts through actions within the environment and receives a reward and observations, as shown in 2.3, the agent updates its internal function's parameters with the objective of outputting actions that maximize the reward. In our case, the agent is the NAO robot and the environment is soccer field adapted to the robot size, both simulated by the physics simulator SimSpark.

A **Markov Decision Process** (MDP) is the mathematical framework in which an RL problem can be expressed: it is a 4-tuple (S, A, P, R) where: set of states S, set of actions A,

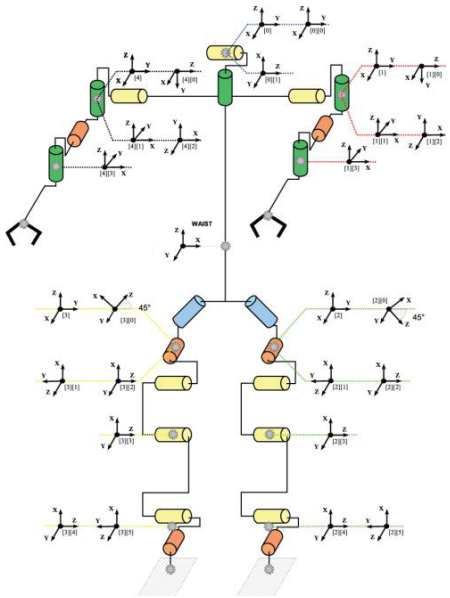


Figura 2.2: NAO Model Anatomy

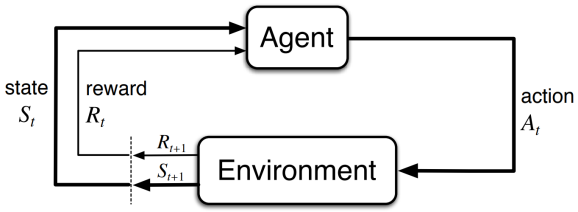


Figura 2.3: Interactions between agent and environment

P (2.1) is the probability that an action a in a state s will lead to the state s' and R (2.2) is the expected immediate reward received after transitioning from state s to s' by action a.

$$p(s'|s,a) = Pr\{S_{t+1} = s'|S_t = s, A_t = a\} \tag{2.1}$$

$$r(s',s,a) = \mathbb{E}[R_{t+1}|S_{t+1} = s', S_t = s, A_t = a] \tag{2.2}$$

2.3 CURRICULUM LEARNING

One of the issues that arises when utilizing Reinforcement Learning is to design a reward function that can generate the desired behavior from the training, that is, the desired behavior must be reinforced by the reward function. The reward function can get convoluted and specific as the goal gets more complex, and one approach that tries to solve this problem is Curriculum Learning (CL). (Bengio et al., 2009)

In Curriculum Learning, the goal is separated in sub-tasks that are ordered by increasing difficulty, the agent will learn first the easier task and after it is mastered, it will move to a following harder task. In CL the researcher must be able to divide a task and order the sub-tasks in difficulty, as well as determine an heuristic to decide when to change the goal forward.(Muzio et al., 2022)

3 RELATED WORK

This chapter discuss related work on different areas that are pertinent to our research, since skill learning on robotics is an active and popular research area and have much work to be comprehensive, this chapter will primarily focus on works that are closely related to our own.

3.1 HUMANOID GAIT WITH RL

In a well known DeepMind paper (Heess et al., 2017), where the goal was to achieve sophisticated locomotion skill of characters on a physically simulated environment utilizing only simple rewards and rich environments. The results were impressive as the agents were able to run, jump and navigate through obstacles, but the downsides of utilizing only deep RL to learn complex behaviors, as demonstrated in the work, is the sample inefficiency, and as the degrees of freedom increases the algorithm tends to generate less impressive gaits and behaviors that suggest the agent to be stuck in local.

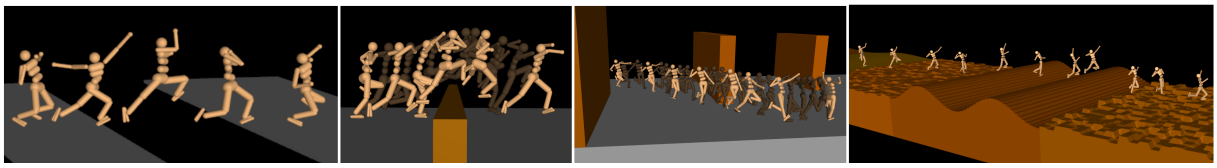


Figura 3.1: DeepMind Humanoid Character gait in several environments

The humanoid character has 28 DoF and 21 joint actuators and was able to succeed in every environment but the gait was not "human like"as can bee seen in 3.1. This work is important to denote the limits of what can be achieved without utilizing specialized algorithms or classical robotics techniques.

3.2 ROBOCUP LOCOMOTION

FCPortugal's team (Abreu et al., 2023) uses a simplified version of the walking engine proposed by (Kasaei et al., 2017) that combines well-known techniques such as the concept of Zero Moment Point (ZMP) firstly utilized in the context of legged locomotion by (Vukobratovic et al., 1970) and Linear Inverted Pendulum Motion (LIPM) (Kajita e Tani, 1991) that are able to generate stable humanoid gait. This approach is not uncommon and is also utilized by similar walking engines such as the developed by (de Albuquerque Maximo, 2015) utilized by ITAndroids.

Realizing that the common trait of skills from the *locomotion skill-set* that includes Omnidirectional Walk, Dribble and Push is alternating lifting each foot, Abreu et al. were able to simplify Kasaei et al.'s walking engine to create a cyclic and smooth stationary walk, this approach resulted in an primitive with improved stability and allowed easier transitions between the skills termed *Step Baseline*.

The *Step Baseline* as seen in 3.2 operates in the background as a skill set primitive for the locomotion skill set but not to the other skills such as *Get Up* and *Kick*

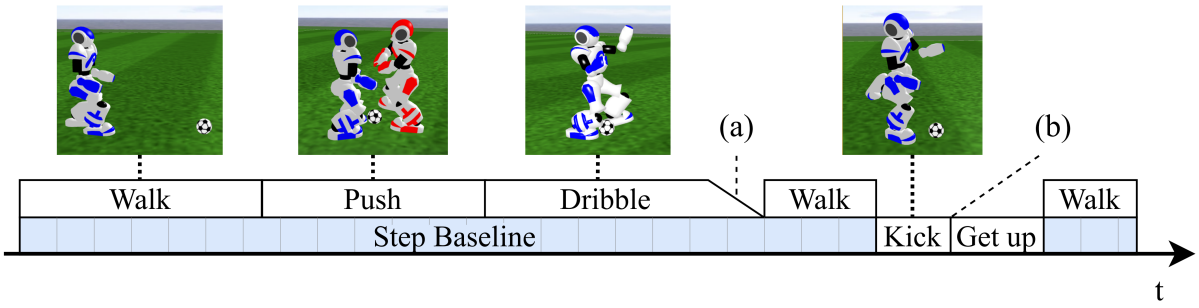


Figura 3.2: Example of the locomotion set in use as the Step Baseline operates in the background as the agent walks, pushes and dribbles. The agent kick, falls and get up, to start walking. **(a)** is the transition from dribble to walk and **(b)** transition between falling after kicking and getting up

4 PROPOSAL

This chapter explains in details what is the object of training and our approach to implement a new behavior to dribble. It defines the tasks of running and dribbling as well as the reward functions that will be utilized in the RL.

4.1 SIMULATION

RoboCup Simulation League 3D utilizes SimSpark as the official physical engine, to interface with it we are using the code provided by the current league winner: FCPortugal. The code contains an walking engine and utilizes Stable Baselines 3 as the implementation of the deep RL algorithms.

4.2 TASKS

We are utilizing the curriculum learning approach, that means that the agent will be trained first to perform the most simple task, running, and after deemed proficient, it will be trained in the more complex task, dribbling, that contains the first and will have an advantageous start than training from scratch.

For simplicity, both tasks have the same observation space, which is the information given by the environment to the agent is an array with 70 positions containing values deemed important to the agent and are updated at every step of the simulation, the information contained in the observation space is described in Table 4.1.

Index	Observation
0	simulation step counter
1	z coordinate (torso)
2	z velocity (torso)
3	absolute orientation in deg
4	absolute torso roll in deg
5	absolute torso pitch in deg
6:9	gyroscope
9:12	accelerometer
12:18	left foot: relative point of origin and force vector
18:24	right foot: relative point of origin and force vector
24:44	position of all joints except head and toes (for robot type 4)
44:64	speed of all joints except head and toes (for robot type 4)
64	step duration in time steps
65	vertical movement span
66	relative extension of support leg
67	step progress
68	if left leg is active
69	if right leg is active

Tabela 4.1: Observation space indexes and description

The action space is an array with the 22 hinge joint effectors available on the NAO robot model and are described in the Table 4.2. The agent can move the hinges within the limits given by the *Step Baseline* primitive, so the gait will be stable and somewhat predictable.

Index	Description	Hinge Joint
1	Yaw	[0][0]
2	Pitch	[0][1]
3	Shoulder Pitch	[1][0]
4	Shoulder Yaw	[1][1]
5	Arm Roll	[1][2]
6	Arm Yaw	[1][3]
7	Hip YawPitch	[2][0]
8	Hip Roll	[2][1]
9	Hip Pitch	[2][2]
10	Knee Pitch	[2][3]
11	Foot Pitch	[2][4]
12	Foot Roll	[2][5]
13	Hip YawPitch	[3][0]
14	Hip Roll	[3][1]
15	Hip Pitch	[3][2]
16	Knee Pitch	[3][3]
17	Foot Pitch	[3][4]
18	Foot Roll	[3][5]
19	Shoulder Pitch	[4][0]
20	Shoulder Yaw	[4][1]
21	Arm Roll	[4][2]
22	Arm Yaw	[4][3]

Tabela 4.2: Action space indexes, description and hinge joint reference

4.2.1 Run

The goal of running is to achieve the maximum velocity without falling, the agent will start at the position X = -14 and its goal is to increase the value of X as can be seen in 4.1 the terminal state is when the agent is considered to be falling or the simulation achieves the step timeout of 300 steps 4.2.

$$r(s,a) = CurrentX - LastX \tag{4.1}$$

$$Terminal = (TorsoZ < 0.3) or (StepCounter > 300) \tag{4.2}$$

4.2.2 Dribble

Our goal in dribbling is to move forward while keeping control of the ball in a certain distance of the agent such as a real human player, in other words, perform small kicks while running. There were two approaches for the reward design, the first one4.3, inspired by (Muzio et al., 2022) and (Hausknecht e Stone, 2015), where the distance from the ball to the agent were

considered together with the advancing of the ball in the X axis, and the second more simple approach 4.4 were only the ball movement in the field is taken into consideration.

$$r(s,a) = (d^{t-1}_{ball-agent} - d^{t}_{ball-agent}) + 0.05e^{(d^{t}_{ball-agent})} + 5\Delta x_{ball} \tag{4.3}$$

The value $d^{t}_{ball-agent}$ is the distance between the ball and the agent in a given time-step t, so the first part of the equation 4.3 is rewarding the agent to increase the distance, that is, to kick the ball, followed by the term $0.05e^{(d^{t}_{ball-agent})}$ that reward the agent to approach the ball, and finally Δx_{ball} is the difference between the ball position in the x axis since the last time-step, therefore the term $5\Delta x_{ball}$ accounts the ball moving forward in the field.

$$r(s,a) = 3\Delta x_{ball} - \Delta y_{ball} \tag{4.4}$$

The intention of the reward function of 4.4 is to keep it more simple and to take in account the ball movement in the y axis. The terminal state is equal in both approaches, it is similar as the 4.2 in the first to terms but also considers the agent losing the ball as a reason to terminate, that is, if the agent runs in front of the ball more than a meter.

$$Terminal = (TorsoZ < 0.3) or (StepCounter > 300) or (x_{ball} - x_{agent} > 1) \tag{4.5}$$

5 RESULTS AND DISCUSSION

This chapter evaluates the training methods and the final agent in their specific tasks. It compares the agents that emerged from the different reward function on Dribbling and concludes by analyzing the achievements of the proposal.

5.1 RUNNING

Figura 5.1: Set of frames showing the running skill

Figura 5.2: Training the Running skill. Graph of the Reward (blue) and Length (red) of an episode of training as the time-steps increases (x-axis)

The figure 5.2 is a chart representing the training of the agent, the x-axis is the time step where an evaluation is recorded, since training is divided between 16 instances of environments,

each one having 1024 steps, and the evaluation taking place when every environment run 20 episodes, that means, at every $16 * 1024 * 20 = 327680$ steps an evaluation is recorded, therefore the X-axis starts at $327,680$ and finishes at $20,643,840$ steps.

The blue line represents the *Reward per episode* and it increases steadily until it stabilizes. The red line is the length of each episode and in this case is caped at 300 steps, every episode the agent finishes under 300 steps is an episode where the agent has fallen and terminated the evaluation earlier.

The running gait achieved can bee seen in the figure 5.1. The results were similar to (Abreu et al., 2023) as it uses the same *Step Baseline* primitive and the velocity is about 0,9 m/s. To test the skill, two hundred evaluations were made and the results can be seen in the histogram in the figure 5.3.

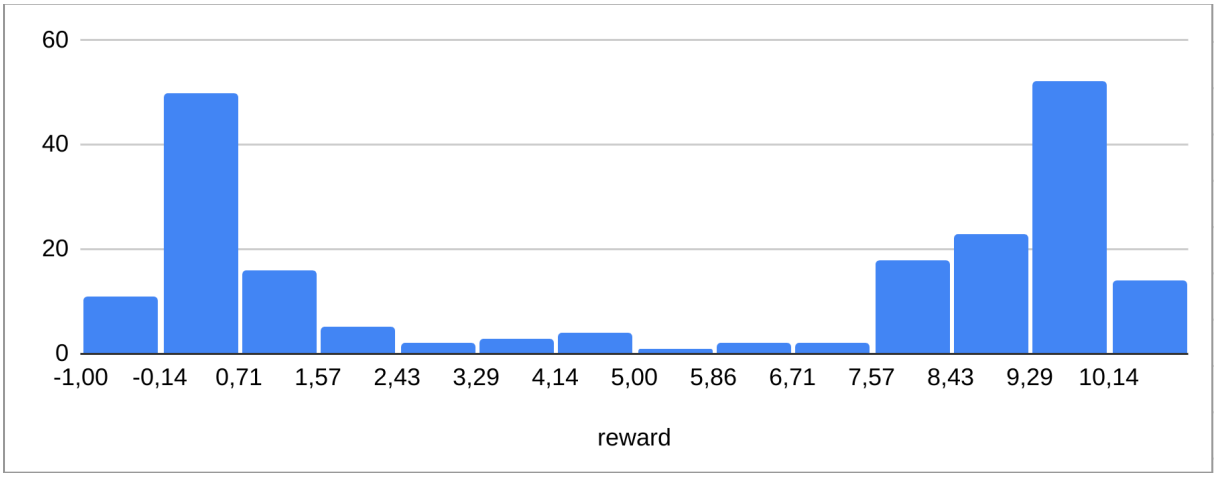


Figura 5.3: Reward histogram of two hundred evaluation of the fully trained agent on the running skill

The results of the two hundred evaluations are interesting because they reveal that the running achieved is not so reliable as the episodes often finalizes with the agent falling, the average reward was $5,41$ and the average steps until termination was 204. The histogram 5.3 shows a high concentration of cases in both ends of the spectrum, meaning that the agent may fall at the beginning of the episode, and if it is able to start running, it will keep running until the end of the episode.

5.2 DRIBBLING

After the agent is able to run while maintaining balance and velocity, we change the curriculum to train the dribbling skill. Since we had two difference reward designs, two behaviors emerged and will be discussed separately in the following subsections.

The figure 5.4 shows an example of an agent after being trained on the dribble skill utilizing the reward function 4.3 termed *Complex Reward* and it is discussed in 5.2.1, consequently, the reward function of the equation 4.4 is termed *Simple Reward* at the 5.2.2 chapter.

5.2.1 Complex Reward

The training evaluation method is the same as the one described in the chapter 5.1: at every $327,680$ steps an evaluation is made and the reward and steps are recorded. The results of the skill training can be seen at 5.5 and are not a representation of a successful trained behavior such as the 5.2.

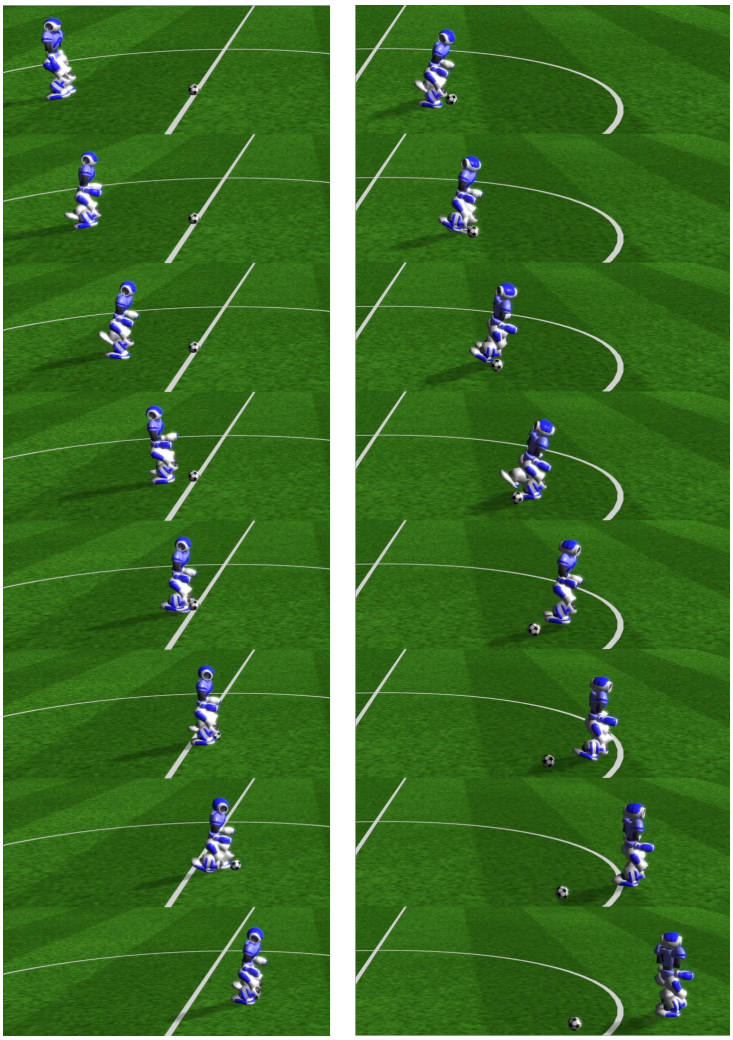


Figura 5.4: Set of frames showing the dribble skill. From top to bottom, left to right, the agent starts running, kicks the ball and loses the ball as it keeps running.

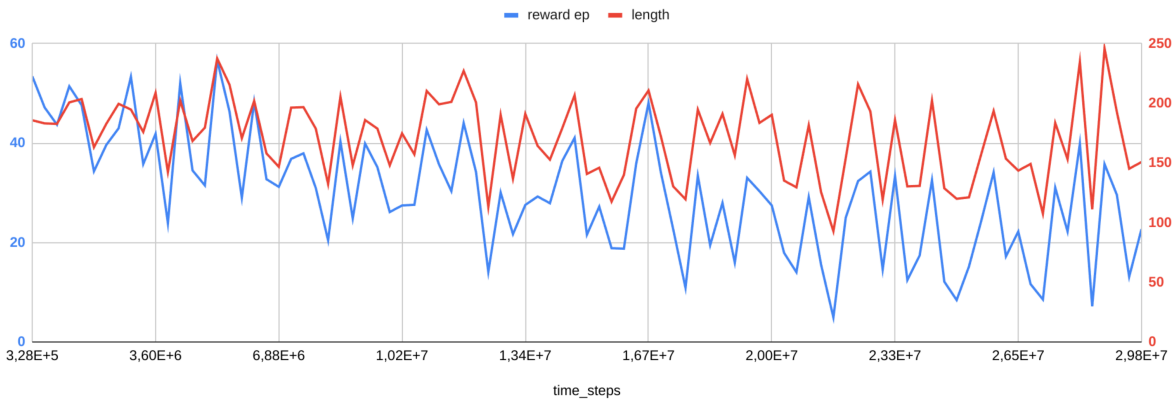


Figura 5.5: Training the Dribble skill with the regular reward (4.3). Graph of the Reward (blue) and Length (red) of an episode of training as the time-steps increases (x-axis)

The chart shows a decline in performance as the training continues, the reward per episode decreases in complete opposite of what is expected, and the trend is to have an agent worst then when it begins. The complex reward function take in account the distance between the

agent and the ball, this can be seen in the final behavior as the agent decreases its velocity when it looses control of the ball while it runs forward without being able to correct its path.

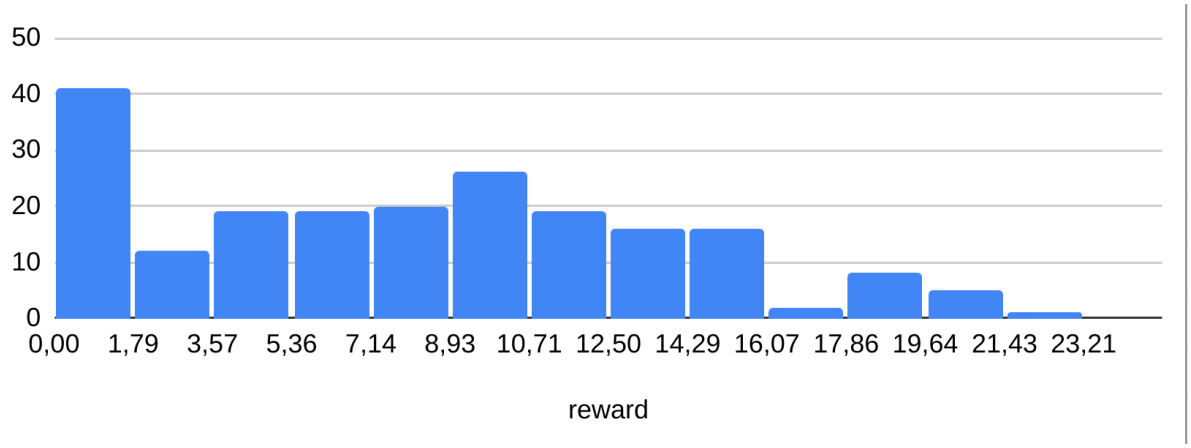


Figura 5.6: Reward histogram of two hundred evaluation of the fully trained agent on the dribbling skill with a complex reward.

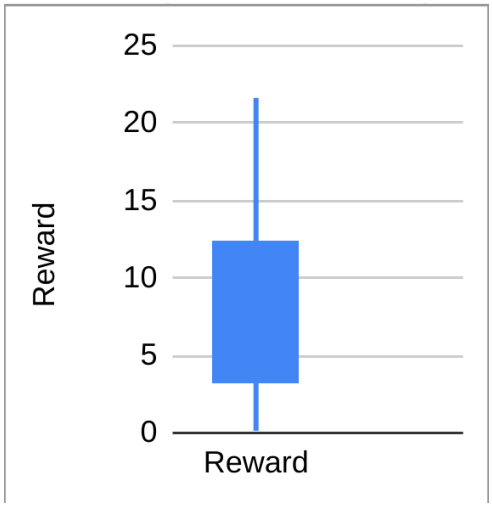


Figura 5.7: Box plot of the two hundred evaluations of a fully trained agent on the dribbling skill with a complex reward.

Two hundred evaluations were made with the fully trained agent and the results can be seen in 5.7 and 5.6, the average reward was $7,95$ and the average of steps taken are 173.

5.2.2 Simple Reward

In this approach, only the ball (x, y) position in the field is taken into account, the intention behind the reward design is to have the agent to perform a strong and straight kick, and run forward. The training evaluation can be seen at 5.8 and it has similar issues as the Complex Reward.

As the training continues the reward tends to decrease and the agent's performance gets worst. The emerged behavior has a poor performance, four hundred evaluations were made and the results can be seen at figures 5.9 and 5.10, the average reward is $6,42$ and the average episode length is 131.

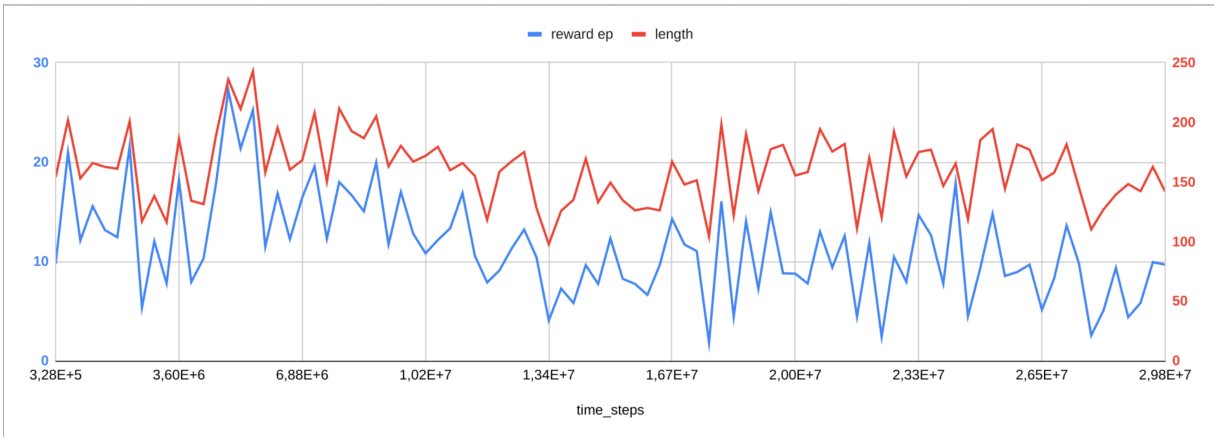


Figura 5.8: Training the Dribble skill with the simple reward (4.4). Graph of the Reward (blue) and Length (red) of an episode of training as the time-steps increases (x-axis)

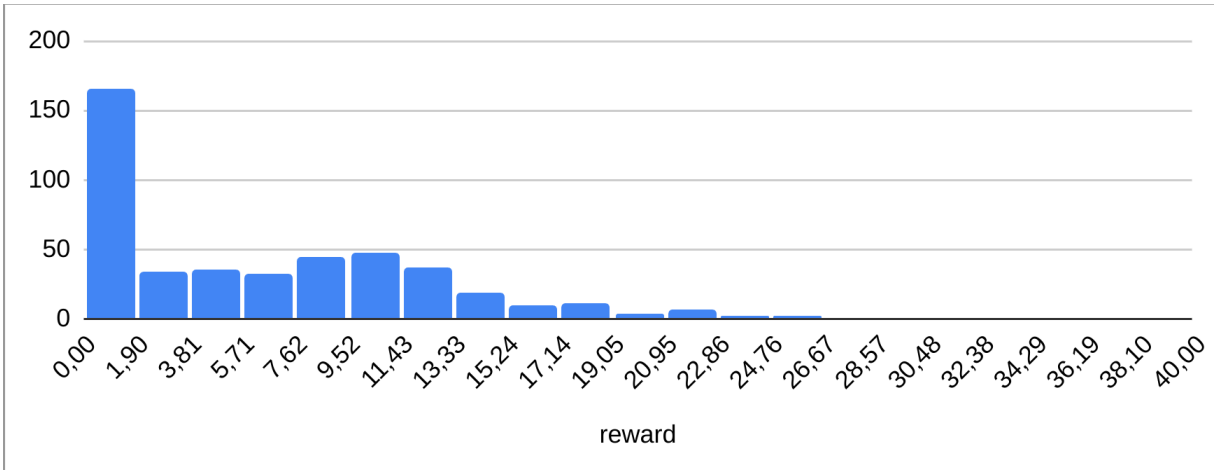


Figura 5.9: Reward histogram of two hundred evaluation of the fully trained agent on the dribbling skill with a simple reward.

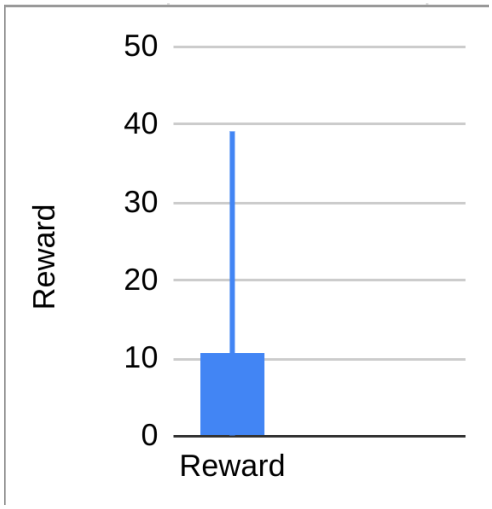


Figura 5.10: Box plot of the two hundred evaluations of a fully trained agent on the dribbling skill with a simple reward.

5.3 CONCLUSION

Both reward functions described in this work were unable to produce the intended complex behavior of dribbling forward, both behaviors share a few major problems, when able to kick the ball, it is common for the agent to lose it as the trajectory of the robot is not corrected to the trajectory of the ball and the episode is terminated, and at many times, the agent is unable to perform the first kick as it runs by the sides of the ball without touching it or the agent loses balance and falls even before.

Since falling at the beginning of the episode without being able to start running is a common trait of the agent trained in running, and the trait was then reproduced by the agent trained in dribbling, this is an example of the downside of utilizing the Curriculum Learning approach, the errors produced at first curriculum will most likely be reproduced forward as the goal gets more complex and can prevent the agent to achieve better results.

The agent being unable to correct its trajectory towards the ball is a problem that would be the limitation even for a completely balanced agent, therefore is the major flaw of the second curriculum. Although in the *Complex Behavior* the reward function takes into account the distance between the agent and the ball, it is not enough to develop an efficient route that intercepts the ball.

6 CONCLUSION

This chapter concludes this work by summarizing its achievements and discussing possible improvements that can be explored in future works on the subject of humanoid motor skills in robotics or physically simulated characters.

6.1 SUMMARY

Utilizing the code base of the current world champion in RoboCup Soccer Simulation 3D League we were able to setup an environment and train a machine learning model with the goal of generating a human-like motor control to dribble a ball forward. The behaviors that emerged from the different rewards were not satisfactory but made possible to analyze the difficulties when working with RL to control a character with a considerably high DoF and a moving object as the ball.

In addition, we were able to analyze the impact of the reward design on the final emerged behavior, with the particularities of the Curriculum Learning.

In conclusion, the approach taken relying on rewards to create a skilled robot able to balance and dribble forward can be seen as naive, as it doesn't actually control the ball but rather march forward hoping to intercept the ball and provoke it to move.

6.2 FUTURE WORKS

To address the problems discussed on Chapter 5.3, the running gait can be improved to be more reliable by lessening the rates where the agent falls at the beginning of the episode, to accomplish this, different rewards can be tested such as penalties for falling or a maximum velocity.

As for the dribbling skill, the main issue was the lack of maneuverability of the agent, redirecting the agent at each contact with the ball can have much impact on the resulting model and may be able to generate a much more dynamic player.

To create a more natural gait and develop robust skills with sample efficiency, the approach of (Peng et al., 2018), where the agent is trained first to mimic an specialist (motion capture data) and to later reproduce in different scenarios.

Finally this work does not take into consideration the RoboCup environment as a soccer match, the achieve a functioning dribble skill the agent must be able to control the ball within a field with adversaries and teammates, as well as to kick precisely to pass or to score.

REFERÊNCIAS

(2024).

Abreu, M., Reis, L. P. e Lau, N. (2023). Designing a skilled soccer team for robocup: Exploring skill-set-primitives through reinforcement learning. *arXiv preprint arXiv:2312.14360*.

Bengio, Y., Louradour, J., Collobert, R. e Weston, J. (2009). Curriculum learning. *Journal of the American Podiatry Association*, 60:6.

de Albuquerque Maximo, M. R. O. (2015). Omnidirectional zmp-based walking for a humanoid robot. Dissertação de Mestrado, Instituto Tecnologico de Aeronáutica.

Federation, R. (2024). Robocup simulation league.

Hausknecht, M. e Stone, P. (2015). Deep reinforcement learning in parameterized action space.

Heess, N., TB, D., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, S. M. A., Riedmiller, M. e Silver, D. (2017). Emergence of locomotion behaviours in rich environments.

Kajita, S. e Tani, K. (1991). Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. Em *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, páginas 1405–1411 vol.2.

Kasaei, S. M., Lau, N., Pereira, A. e Shahri, E. (2017). A reliable model-based walking engine with push recovery capability. Em *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, páginas 122–127.

League, R. S. S. (2024). Robocup soccer simulation league home.

MagmaOffenburg (2024).

Muzio, A., Maximo, M. e Yoneyama, T. (2022). Deep reinforcement learning for humanoid robot behaviors. *Journal of Intelligent Robotic Systems*, 105.

Peng, X. B., Abbeel, P., Levine, S. e van de Panne, M. (2018). Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph.*, 37(4):143:1–143:14.

Robotics, A. (2024).

SimSpark (2024a).

SimSpark (2024b).

SimSpark-Aldebaran (2024).

Sutton, R. e Barto, A. (2018). *Reinforcement Learning, second edition: An Introduction*. Adaptive Computation and Machine Learning series. MIT Press.

Vukobratovic, M., Frank, A. A. e Juricic, D. (1970). On the stability of biped locomotion. *IEEE Transactions on Biomedical Engineering*, BME-17(1):25–36.