Proceedings of the 2018 IEEE
International Conference on Robotics and Biomimetics
December 12-15, 2018, Kuala Lumpur, Malaysia

Pseudo-Locomotion Design with a Quadrotor-Assisted Biped Robot

Azumi Maekawa[1], Ryuma Niiyama[2], and Shunji Yamanaka[3]

***Abstract*— Real-world mechanical characters play an important role in our daily lives and provide rich experiences. This study presents a novel approach to develop a biped robot that has the ability to display walking appearance with dynamic mobility, regardless of its environment. We utilize a quadrotor as an overhead tether to provide mobility and stability. Using the thrust of the rotors, we developed a biped robot for entertainment purpose. It is able to move agilely using novel foot trajectory generation. This method focuses on the velocity of the robot's foot that is in contact with the ground. The control policy is designed to generate in the robot's legs such that it reduces the velocity of the foot in contact with the ground to zero. The policy used for the foot trajectory generation is trained by deep reinforcement learning with a physics-based simulator.**

I. INTRODUCTION

There has been a desire to create mechanical characters that are capable of life-like motions and behaviors. These characters play an important roles in animatronics in theme parks and as electro-mechanical toys, and robotic pets. However, though the progress in modern robotics technology has been remarkable, there are still many practical difficulties and problems remain with regard to creating nuanced, high-quality character motions akin to those in the virtual environment.

This study focuses on a bipedal walking motion, which is one of such difficulty. The ability to locomote freely is an important element for a legged character. The bipedal locomotion problem continues to exist even today, as seen in the DARPA Robotics Challenge[1]. Realizing movement in a bipedal robot akin to that of a living thing is one of the biggest challenges so far, and many studies have been conducted in this area. In addition to control problems, the process of creating these robots is not only very time-consuming, but also requires considerable resources including manpower. However, it is possible to provide the desired visual experience while avoiding these problems by focusing on the appearance features.

Ghassami et al. proposed a bipedal robot prototype that used the buoyancy of a helium filled balloon[2]. This approach enabled various motions, such as walking, jumping, and dancing. It stabilized the robot intrinsically and prevented it from falling. However, the technique could not confer dynamic motions since it was greatly influenced by air resistance and wind.

Fig. 1. Appearance of the bipedal robot using a quadrotor.

Some approaches use mechanical structures consisting of gears and link structures[3][4][5], cams and crank-sliders[6] to realize the appearance of walking motion in a robot intended for entertainment. Although users can design an arbitrary motion trajectory of the robot's end effectors by using these methods, it is limited to a cyclic and static motions, and stable locomotion skill is lacking in most cases.

Megaro et al. proposed an interactive design system that could generate robots to locomote stably with arbitrary, user-created morphologies[7]. Bharaj et al. proposed a computational design method that allows casual users to create personalized walking toys using pre-defined template mechanisms[8]. Although these methods enable users to design a variety of robotic creatures, the motion of the robot, especially the bipedal locomotion, is relatively slow and limited to quasi-static motion.

Our goal is to develop a robot that has the ability to display walking appearance with dynamic mobility, regardless of its environment. This study refers to this motion *pseudo-locomotion*. This paper presents an approach that utilizes a quadrotor as an overhead tether to provide mobility and stability. This approach aims to enable a rich physical expression by allowing the robot to enhance its maneuverability. To achieve pseudo-locomotion with the biped robot, we developed a novel foot trajectory generation method using deep

[1]Azumi Maekawa is with the Graduate School of Engineering, the University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo, JAPAN azumi@star.rcast.u-tokyo.ac.jp

[2]Ryuma Niiyama is with the Graduate School of Information Science and Technology, the University of Tokyo, 7-3-1 Hongo, Bunkyo-Ku, Tokyo, JAPAN niiyama@isi.imi.i.u-tokyo.ac.jp

[3]Shunji Yamanaka is with Institute of Industrial Science, the University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo, JAPAN yam@iis.u-tokyo.ac.jp

reinforcement learning (DRL). The details of this method are discussed in the following sections.

II. SYSTEM OVERVIEW

A. Hardware Design

The robot consists of a quadrotor and legs. This study used DJI Phantom 3 Advanced as the quadrotor. It is one of the a consumer unmanned aerial vehicle (UAV), and its onboard processing unit incorporates an inertial measurement unit (IMU) for attitude control and a distance sensor that measures the current height from the ground. The weight is approximately 1280 g, and the flight time is up to about 23 min with no payload.

Figure 2 shows the overview of the leg unit. In this paper, we designed simple hardware for simplicity with regard to fabrication and control and to enable a convenient evaluation of our approach. The leg unit consists of 3D printed link parts and off-the-shelf servo motors. The 3D printed parts were printed on Aspect RaFaEl300 using ASPEX-PA material. The material is a kind of nylon. It has excellent toughness, and it is possible to print parts with high flexibility depending on the structure. B and D in Figure 2 are 3D printed parts and have relatively high rigidity. E, which is the foot, has a flexible structure and can deform easily according to the applied force, and D and E are fabricated as one part. Each leg has two degrees of freedom (DOF) and employs Kondo KRS-2572 HV, which is a position-controlled servo, one for each DOF (A and C in Figure 2).

The rotation axes of the servo motors were designed to be parallel to each other and to allow the foot to move in the sagittal plane. Figure 2 shows the leg posture when the angle command to the motor is 0 rad. In this paper, we defined the bending direction of the knee in the direction opposite to the heading, inspired by the appearance of a bird's leg. The range of motion in each servo motor was -1 rad to 1.5 rad. The weight of the leg unit was 206 g, and the total weight of the robot was about 1870 g, including the part that connects leg units and quadrotor.

B. Motion Control

We focused on the velocity of the foot in contact with the ground. In biped locomotion, a gait is generated by each leg repeating the transition between a stance phase and a swing phase. One of the features of walking appearance is that the speed of the grounded foot is zero.

This feature is often used as a constraint when considering natural walking behavior in computer animation[9][10]. To achieve this feature, it is necessary to control the swing leg step properly while the stance leg maintains contact with the ground as the movement of the quadrotor changes successively. The design of the control system becomes a complex and daunting task with the increasing of the DOF. This study therefore proposes a gait generation method focusing on the grounded foot state by using DRL. We trained the control policy in the simulation environment based on the configuration of the hardware design since DRL takes a considerable amount of time for learning. We started

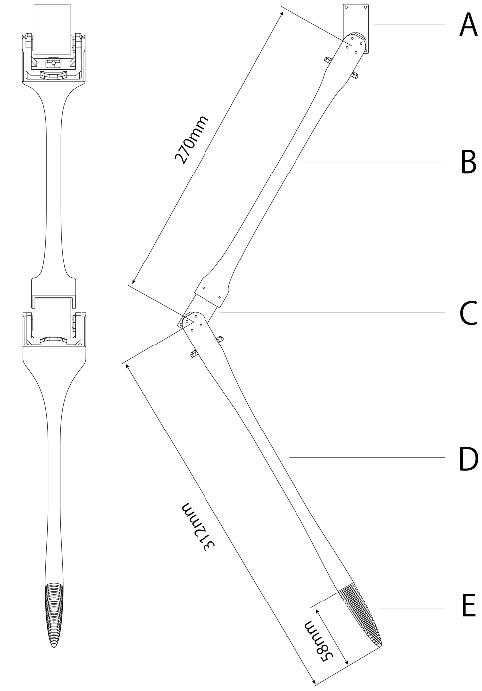

Fig. 2. Components of the leg unit.

by defining the robot model according to the properties of the robot's legs. We defined the robot model as shown in Figure

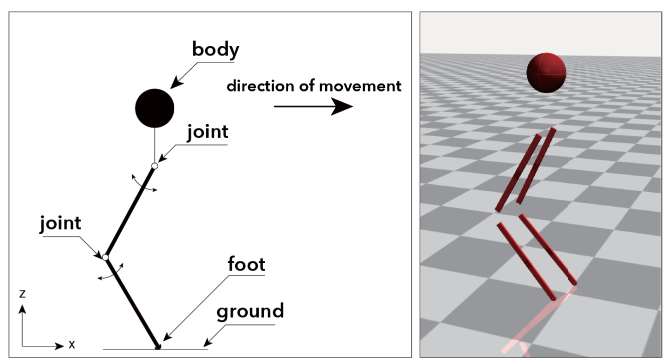

Fig. 3. Robot model used for policy training in the simulation environment.

3. We used MuJoCo [11], which it is widely used for DRL, for the simulation environment. In the MuJoCo environment, the robot configuration is represented by MJCF, where the main part is an XML tree. We modeled the quadrotor as a body part assuming that the height from the ground is constant and the rotation during movement is negligible. The body was set at the position where the feet touched the ground when the command to the actuators was 0 rad. The leg unit comprised the position control actuators and the rigid links. The length of both links of the leg is set to 0.27 m based on the hardware design. The foot, which is the contact part with the ground, was assumed to form an arc shape of radius 0.75×10^{-3} m, which was used to simplify the contact calculation.

In reinforcement learning problems, the robot observes the state s and takes the action a according to a policy in order to

Algorithm 1 Calculate reward

$\alpha, \beta, \epsilon > 0$
if $0 \leq v_{body} < \epsilon$ **then**
  if left foot and right foot are in contact with the ground **then**
    $reward \leftarrow (\alpha - (v_{foot_L} + v_{foot_R}))$
  else
    $reward \leftarrow -\beta$
  end if
else
  if left foot is in contact with the ground **and** right foot is not **then**
    $reward \leftarrow (\alpha - v_{foot_L})$
  else if right foot is in contact with the ground **and** left foot is not **then**
    $reward \leftarrow (\alpha - v_{foot_R})$
  else if left foot and right foot are in contact with the ground **then**
    $reward \leftarrow (\alpha - (v_{foot_L} + v_{foot_R}))$
  else
    $reward \leftarrow -\beta$
  end if
end if

maximize the sum of the reward. The state s consists of the root body's velocity v_{body} and current angles of each joint. The action a from the policy specifies the target angles of actuators at each joint and is used to compute torques to be applied to each joint. The policy was queried at 20 Hz, and the observed state space was mapped to the action space. The reward was designed based on the robot's feet state as shown in Algorithm 1. The reward described in Algorithm 1 indicates that the following states are desired: 1) when the body stops, both feet are in contact with the ground, and the speed of the each foot is 0 $\mathrm{m/s}$, 2) when the body is moving, one foot is or both feet are in contact with the ground, and the speed of the foot in contact with the ground is 0 $\mathrm{m/s}$. A penalty is applied when both feet are not in contact with the ground regardless of the velocity of the body. We set the hyperparameters as follows $\epsilon = 1 \times 10^{-3}, \alpha = \beta = 1$.

The policy was trained under the following conditions.

1) The body has one degree of translation freedom (x axis in Figure3).
2) Each leg has two degrees of rotation freedom.
3) v_{body} changes according to Algorithm 2.
4) The robot gets the reward according to Algorithm1.

Algorithm 2 represents a randomly changing v_{body} for each step k, and X and Y represent the disturbance noise to allow the robot to acquire a robust policy. $U(a, b)$ and $N(\mu, \sigma^2)$ respectively represent uniform distribution in $[a, b]$ and a Gaussian distribution with mean μ and variance σ^2. We set $a = -0.3, b = 0.3, v_{max} = 1.8, c = 0.05, \mu = 0, \sigma^2 = 1$ empirically.

Algorithm 2 v_{body} in each episode

initialize v_{body}
$X \sim U(a, b), Y \sim N(\mu, \sigma^2)$
for $i = 0, 1, 2, \cdots,$ **do**
  if $i \bmod k$ **then**
    $v_{body} \leftarrow v_{body} + X$
    $\mathrm{clip}(\mathrm{v_{body}}, 0, \mathrm{v_{max}})$
  else
    $v_{body} \leftarrow v_{body} + c \times Y$
  end if
end for

III. EXPERIMENTS

To verify the proposed method, we conducted the experiments on policy training and real robot control.

A. Policy Training

We used Proximal Policy Optimization (PPO) [12], a state-of-the-art algorithm in continuous state-action space control, for the policy training. The policy network architecture consisted of 2 fully connected hidden layers, and each layer has 64 hidden units. A hyperbolic tangent was used as the activation function of these layers. We used Adam[13] as an optimizer, and the hyperparameters are shown in Table I. We set an episode as 500 steps.

TABLE I
HYPERPAREMETERS USED IN THE TRAINING POLICY EXPERIMENT

Hyperparameter	Value
Learning rate	3×10^{-4}
Num. epochs	10
Minibatch size	64
Discount rate	0.99
Clipping ϵ	0.2

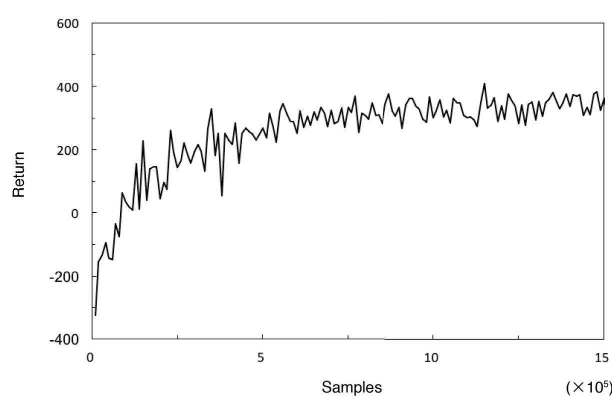

Fig. 4. Learning curve for policy training in the simulation environment.

Figure 4 shows the learning curve of the training. It shows that the learning has converged at about one million samples. The theoretical maximum value of the return that could be obtained in an episode was 500. Thus, the results show that

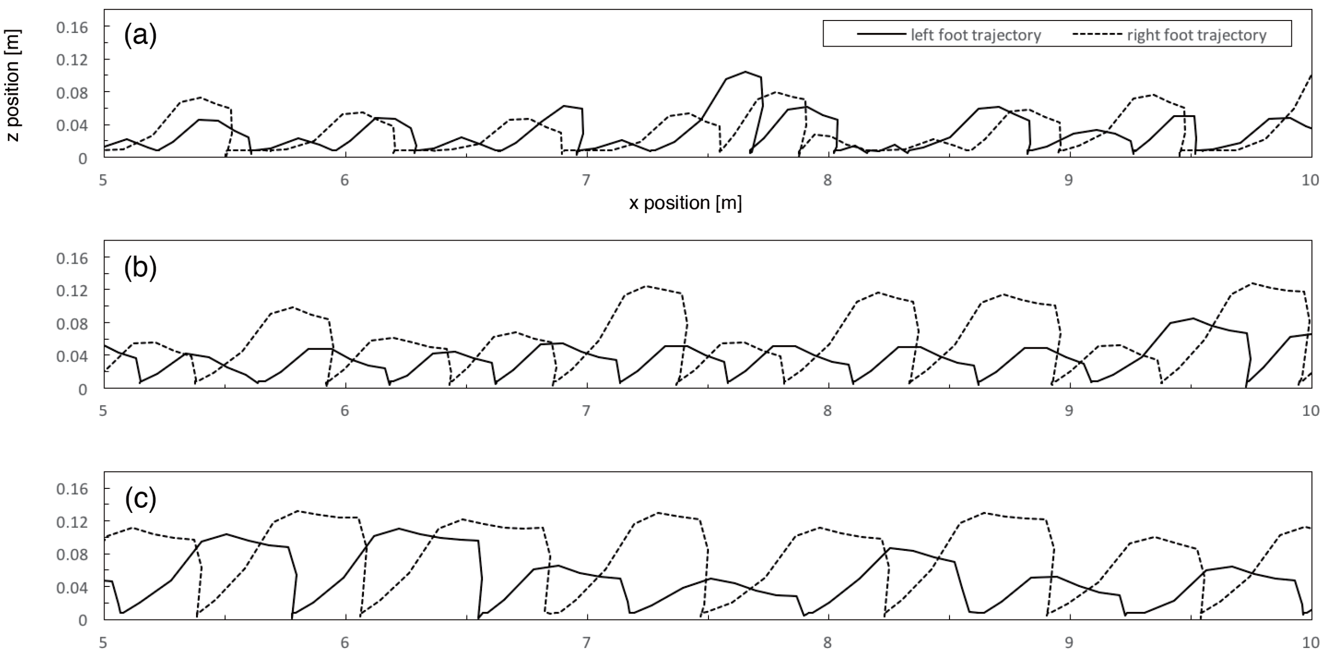

Fig. 5. Foot trajectories at (a) $v_{body} = 0.5\,\mathrm{m/s}$, (b) $v_{body} = 0.75\,\mathrm{m/s}$, (c) $v_{body} = 1.0\,\mathrm{m/s}$ ($5.0\,\mathrm{m} \leq x \leq 10.0\,\mathrm{m}$)

about 70 % of the theoretical value could be obtained with the learned policy.

Figure 5 shows the trajectory of the foot when the body was subjected to constant velocities (0.5, 0.75, and 1.0 $\mathrm{m/s}$) with the learned policy. The x position is from 5 to 10 m, where the velocity of the body was in the steady state. We found that the velocity of the foot in contact with the ground was almost 0 $\mathrm{m/s}$, and pseudo-locomotion was generated. The results also show that the stride and walking cycle were changed according to the body's velocity, thus mimicking nature; it is known that a positive correlation exists between movement speed and stride in many living creatures. Therefore, the results of this experiment indicate that the obtained policy follows one aspect of gaits in living things.

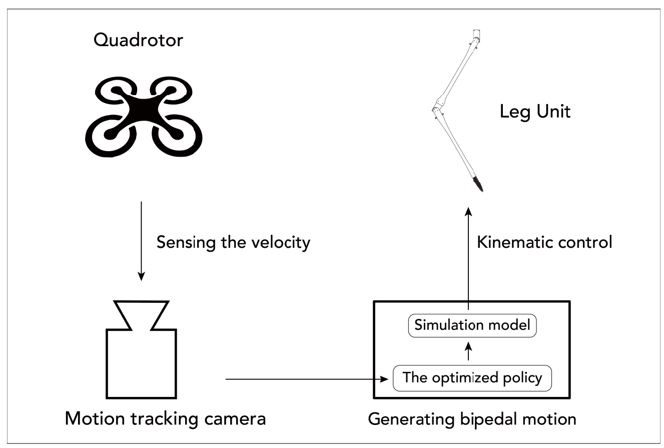

Fig. 6. Control system in the real-world environment.

B. Real World Environment

The real-world environment necessitates measurement of the motion of the quadrotor with sensors so as to estimate the body's velocity. In this paper, we used motion capture cameras, OptiTrack Flex 3, to fulfill this requirement. The command to provide the input to the actual servo motors was calculated by the learned policy using the estimated velocity of the quadrotor. However, it is difficult to precisely simulate properties of an actual servo motor, such as response speed, temperature characteristics, disturbance, etc. Therefore, even if the same input is given, the identical result is not necessarily returned between the MuJoCo motor and the actual motor.

In this experiment, we constructed a control system, as shown in Figure 6, and controlled the real robot with the following steps. The velocity of the quadrotor was estimated from the tracking data obtained from motion capture cameras and inputted into the learned policy. The policy returned the optimized action, and the robot's pose in the next step in the simulation environment was computed using this action. Then, angle positions of the robot model's leg joints were inputted to the real servo motors. By repeating this step, the robot's leg motion could be controlled continuously in the real-world environment, and pseudo-locomotion was generated according to the velocity of the quadrotor in real time. The bipedal motion generation was executed in external CPU, and servo motors connected in a daisy chain were controlled with the microcontroller, Arduino MEGA, using serial communication. In this experiment, the quadrotor was controlled manually using a remote. Figure 7 shows the generated gait of the real robot. The result of the experiment

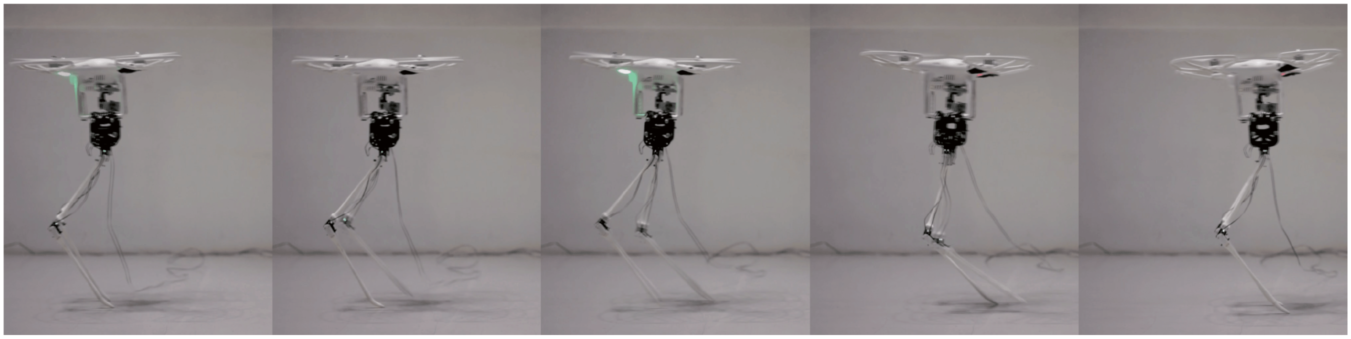

Fig. 7. Generating pseudo-locomotion experiment on hardware.

shows that pseudo-locomotion was generated even in the actual robot, as shown in Figure 7. We found that the system enabled the user to choreograph the walking motion of the robot interactively in real time by controlling the quadrotor movement.

IV. CONCLUSION AND FUTURE WORK

We proposed a biped robot for intended for entertainment using a quadrotor. This robot can move agilely and generated pseudo-locomotion in real time using the control policy, wherein the velocity of the foot in contact with the ground is zero according to the velocity of the quadrotor. The control policy is optimized with a deep reinforcement learning based method. Our method allows the robot to automatically generate a foot trajectory that realized the appearance of bipedal walking with mobility and stability. Thus, our approach has the potential to realize a virtual reality experience in the real world by enabling the physical movement, which has been impossible so far due to the constraints of the mechanism and real-world properties.

Presently, the motion is limited to velocity in one direction and the walking motion occurs in the sagittal plane. Therefore, increasing the DOF of the body and legs will enable various motions. It is possible to adjust the generated trajectories slightly with the current method by tuning the hyperparameters. However, it is difficult to make the movements feature the desired motions. Thus, our future work will attempt to create a control system that realizes the desired trajectories of the biped robot using multirotor. In this study, the leg was controlled by the off-board controller, and power was supplied externally using wires. In order to obtain high mobility, it is necessary to use a load battery and an onboard controller. Furthermore, in this study, we controlled the robot in the real world using a kinematics-based approach. In the future, We will consider the dynamics of the system and attempt to allow the robot to acquire various motions.

ACKNOWLEDGMENT

I would like to thank M. Inami for useful advice. This work was supported by JST ERATO Grant Number JPMJER1501, Japan.

REFERENCES

[1] E. Krotkov, D. Hackett, L. Jackel, M. Perschbacher, J. Pippine, J. Strauss, G. Pratt, and C. Orlowski, "The darpa robotics challenge finals: results and perspectives," *Journal of Field Robotics*, vol. 34, no. 2, pp. 229–240, 2017.

[2] S. Ghassemi and D. Hong, "Feasibility study of a novel robotic system ballu: Buoyancy assisted lightweight legged unit," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 144–144.

[3] D. Ceylan, W. Li, N. J. Mitra, M. Agrawala, and M. Pauly, "Designing and fabricating mechanical automata from mocap sequences," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 186, 2013.

[4] S. Coros, B. Thomaszewski, G. Noris, S. Sueda, M. Forberg, R. W. Sumner, W. Matusik, and B. Bickel, "Computational design of mechanical characters," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 83, 2013.

[5] B. Thomaszewski, S. Coros, D. Gauge, V. Megaro, E. Grinspun, and M. Gross, "Computational design of linkage-based characters," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 64, 2014.

[6] L. Zhu, W. Xu, J. Snyder, Y. Liu, G. Wang, and B. Guo, "Motion-guided mechanical toy modeling." *ACM Trans. Graph.*, vol. 31, no. 6, pp. 127–1, 2012.

[7] V. Megaro, B. Thomaszewski, M. Nitti, O. Hilliges, M. Gross, and S. Coros, "Interactive design of 3d-printable robotic creatures," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 216, 2015.

[8] G. Bharaj, S. Coros, B. Thomaszewski, J. Tompkin, B. Bickel, and H. Pfister, "Computational design of walking automata," in *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 2015, pp. 93–100.

[9] F. Multon, L. France, M.-P. Cani-Gascuel, and G. Debunne, "Computer animation of human walking: a survey," *The journal of visualization and computer animation*, vol. 10, no. 1, pp. 39–54, 1999.

[10] C. K. Liu and Z. Popović, "Synthesis of complex dynamic character motion from simple animations," *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3, pp. 408–416, 2002.

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

[12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.