Learning energy-efficient trotting for legged robots

Athanasios Mastrogeorgiou, Aristotelis Papatheodorou, Konstantinos Koutsoukis, and Evangelos Papadopoulos

School of Mechanical Engineering,
National Technical University of Athens,
Athens 15780, Greece,
{amast, mc15407, kkoutsou, egpapado}@central.ntua.gr,
WWW home page: csl-ep.mech.ntua.gr

Abstract. Quadrupedal locomotion skills are challenging to develop. In recent years, Deep Reinforcement Learning (DRL) promises to automate the development of locomotion controllers and map sensory observations to low-level actions. However, legged locomotion still is a challenging task for DRL algorithms, especially when energy efficiency is taken into consideration. In this paper, we propose a DRL scheme for efficient trotting applied on Laelaps II quadruped in MuJoCo. First, an accurate model of the robot is created by revealing the necessary parameters to be imported in the simulation, while special focus is given to the quadruped's drivetrain. Concerning, the reward function and the action space, we investigate the best way to integrate in the reward, the terms necessary to minimize the Cost of Transport (CoT) while maintaining a trotting locomotion pattern. Last, we present how our solution increased the energy efficiency for a simple task of trotting on level terrain similar to the treadmill-robot environment at the Control Systems Lab [1] of NTUA.

Keywords: Legged robots, learning locomotion, energy efficiency, deep reinforcement learning

1 Introduction

Locomotion skills for quadruped robots require fast reactions, coordinated control of legs, precise manipulation of contact forces, and robust balance control. Setting up such controllers requires significant expertise and often tedious manual tuning. Data-driven methods, such as model-free DRL already have produced promising results showing that they can overcome the simplification of prior model-based approaches by learning effective controllers directly from experience [2] [3] [4] [5]. According to [6], using a low-level toe trajectory planner, markedly reduces training time but requires an investigation on which gait parameters will be left for the agent to learn. Another aspect of DRL is the fact that learning based locomotion controllers usually focus on the target task to be achieved without investigating the energy consumption during that task [7], or

they just penalise high joint accelerations [3] [8] without mechanical antagonism in mind [9]. Additionally, achieving a gait such as trotting, requires accurate orchestration of the robot legs, i.e., a task harder to train in joint space since it requires even more training data, while during training it may produce invalid configurations and thus, special care is needed to avoid this [8] [10]. As a result, learning energy efficient locomotion skills for legged robots while maintaining a well orchestrated locomotion pattern has room for improvement especially when dynamic motions are involved. In this work, the trotting motion is investigated so that the quadruped robot Laelaps II [1] is able to move forward on level terrain, without diverging from its goal, while achieving reduced energy consumption.

1.1 Contribution & Overview

We propose learning efficient controllers directly in Cartesian space by selecting specific toe trajectory attributes while maintaining a trotting pattern as well as constant toe clearance from the ground. It is the first time that DRL with energy efficiency in mind is investigated for Laelaps II quadruped. As a result, it is shown that the Cost of Transport (CoT) is significantly reduced with the proposed reward function even when compared to similar approaches [3] [8], since in this work actuator mechanical antagonism and electric loses are taken into account in the reward function.

The rest of the paper is organised as follows: Section 2 introduces Laelaps II quadruped and justifies all the choices towards creating an accurate model for training. Section 3 describes the developed training environment that produces the policies for energy efficient trotting. The final two sections present the results and discuss future research directions.

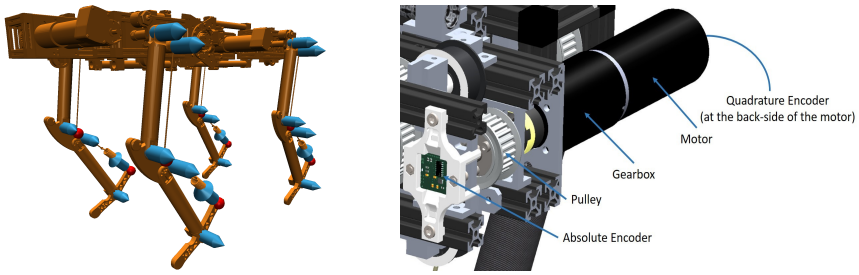
2 Laelaps II Quadruped

2.1 Robot Description

The Laelaps II quadruped, developed by the Legged Robots Team of the Control Systems Laboratory (CSL) [1] at the National Technical University of Athens (NTUA), is a 40 kg, state of the art quadruped that serves as a research platform for control, software development & DRL. Laelaps II legs consist of 3 main segments (femur, tibia, foot) and 3 revolute joints namely hip, knee and ankle joint. The hip and knee joints are actuated by actuation units mounted on the robot's body whereas the ankle joints are passive (a spring tendon connects foot and tibia), see Figure 1a. Each actuation unit (see Figure 1b) consists of the electric motor, the planetary gearbox, the encoder and the timing belt transmission (26/48 reduction ratio). For the hip, a Maxon EC45 250W brushless motor is used, along with its GP52C planetary gearbox (with reduction ratio 8/343). The knees share the same family of gearboxes (with reduction ratio 12/637) with the hips, but they are actuated by Maxon RE50 200W brushed motors. In the knees' case, a parallel mechanism (close kinematic chain consisting of the transmission

arm, femur, tibia and rod) is responsible for the power transmission. A more detailed descriptions of the robot's architecture is presented in [11] [12].

The objective of this work is to create an accurate model of the robot's dynamics, which will be used in DRL training, aiming to produce energy-efficient motions on flat terrain similar to the treadmill-robot setup at CSL [1]. Accurate modeling is a key requirement in extracting valid results and applying the trained policies to the real robot in the future. Towards this, the MuJoCo [13] rigid body dynamics simulator was chosen since it offers notable contact stability [14] and is able to deal with closed kinematic chains present in Laelaps II legs.



(a) (b)

Fig. 1: Laelaps II: (a) MuJoCo model, (b) Drivetrain.

2.2 Laelaps II MuJoCo Model

Our main aim during the modeling process was to create a MuJoCo model as realistic as possible. To this end, all the electrical and mechanical properties of the model (see Figure 1a) correspond to the Laelaps II quadruped [9] [11]. The properties were derived from the components' datasheets and/or were experimentally verified. Special mention should be made to the identification of the frictional parameters of each actuation unit, since they are directly connected to energy consumption.

In order to characterize the friction type we conducted various experiments with constant velocity. The sampling points were carefully chosen to be at the boundary of the breakaway torque as suggested in [15]. For a wide range of voltages, i.e. 0-48V, applied at the motor's terminals, current consumption was measured using the high-precision FLUKE 289 True RMS multimeter[1]. Additionally, using a HEDL 5640 quadrature encoder[2] the joint's angular velocity was recorded. The current measurements were converted to torque at the actuation unit's output shaft using (1), with T and n being the actuation unit's output torque and reduction ratio respectively , K_T the motor's torque constant and i_m

[1] https://www.fluke.com/en/product/electrical-testing/digital-multimeters/fluke-289
[2] https://docs.broadcom.com/doc/AV02-0993EN.

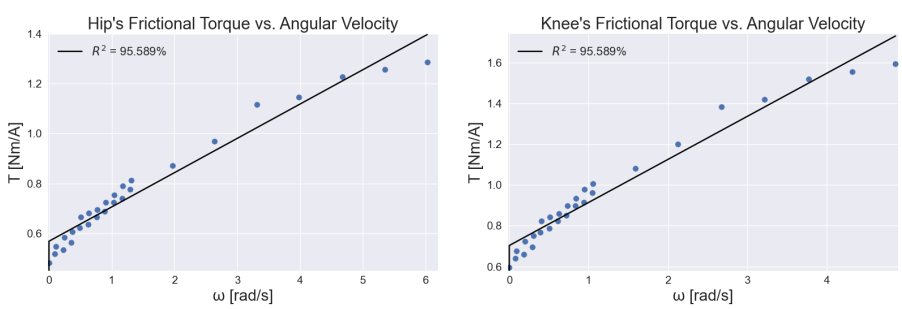
the applied current. The results of the described procedure are displayed in Figure 2. Note that the sampling indicates that there were no significant deviations in the reverse direction. As a result, the same model is used for both directions of motion.

$$T = K_T i_m n \tag{1}$$

The linear model of equation (2) accounts for the static and linear viscous friction components, with τ_i and $\dot{q}_i$ representing the i_{th} sample of the frictional torque and angular velocity respectively and offers the right balance between accuracy and modeling complexity. The f_1 & f_2 are the unknown friction coefficients.

$$\tau_i = f_1 \dot{q}_i + f_2 sign(\dot{q}_i) \tag{2}$$

This model is ideal for a model-based control design in terms of execution time, while at the same time it exhibits a good R^2 index. The regression problem is solved using linear least-squares, which produce the coefficients given in Figures 2a & 2b for the hip's & knee's joints, respectively. Note that the torques induced by friction are relatively low, when compared to the maximum ones that the system can output at present, namely 70.21 Nm for the hips and 109.76 Nm for the knees. To avoid unrealistic motions, these limits are included in the developed MuJoCo model.



(a) $(f_1,\ f_2) = (0.138,\ 0.568)$ $[Nm]$. (b) $(f_1,\ f_2) = (0.211,\ 0.703)$ $[Nm]$.

Fig. 2: (a) Hip experiment's friction model. (b) Knee experiment's friction model.

2.3 Laelaps II Planner

The planner introduced in [12] has been utilised here to allow the robot's model to move with constant stance velocity in MuJoCo, resulting in smoother gaits. This is a vital step towards the integration and testing of various motion modes, like trotting, originally investigated in [16]. Each leg follows a trajectory which includes separate formulations for swing and stance phases. The planner dictates the appropriate formula, by measuring the time progression of each step

compared to the step's total period. By using the modulo operation, the whole process becomes independent of the step number. Moreover, a time phase shift is introduced, to enable different modes of motion (e.g. trotting, walking, etc.).

Let T_{step} be the total duration of a step, Δt_{phase} its phase shift, while T_{sw} and T_{st} the periods of the swing & stance trajectory phases, respectively. To make the trajectory invariant to the step's number, t_{traj} is introduced. Finally, the variable dir_{leg} is used to change the direction of leg's motion, with $dir_{leg} = 0$ for forward and $dir_{leg} = 1$ for backward motion. The resultant operational space trajectory $(x_{E,des},\ y_{E,des})$ is the half ellipse illustrated in Figure 3, while its formulation is presented in equations (3)-(5). Note that the ellipse's center is represented by $(x_0,\ y_0)$ while its horizontal and vertical radii with a and b, respectively. The operational-space trajectory is converted to a joint-space angle sequence that, in turn, is supplied to the low-level PV controllers of Laelaps II to produce the torques required by the desired motion.

$$t_{traj} = (t + \Delta_{phase}) \bmod (T_{step}), \quad with \;\; T_{step} = T_{sw} + T_{st} \tag{3}$$

Swing Phase

$$\begin{aligned} x_{E,des} &= x_0 + a\cos\left(\theta_{traj} + dir_{leg}\pi\right) \\ y_{E,des} &= y_0 + b\sin\theta_{traj}, \quad with \;\; \theta_{traj} = \frac{\pi}{2}\left(\cos\frac{\pi t_{traj}}{T_{sw}} + 1\right) \end{aligned} \tag{4}$$

Stance Phase

$$\begin{aligned} x_{E,des} &= x_0 + (1 - 2dir_{leg})\left(a - (t_{traj} - T_{sw})\,V_{st}\right), \quad V_{st} = \frac{2a}{T_{st}} \\ y_{E,des} &= y_0 \end{aligned} \tag{5}$$

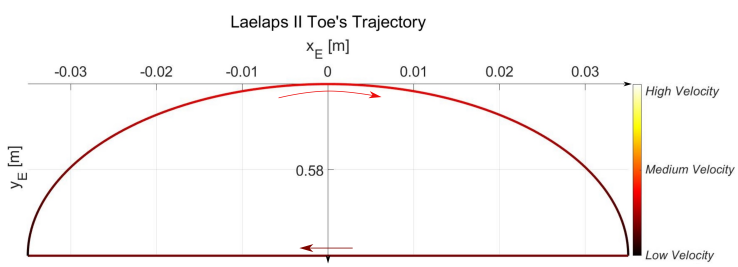


Fig. 3: Laelaps II toe's Cartesian path.

3 Training Environment

The fundamental constituents of every DRL training system is the task in context with an environment that the agent interacts. Here, the task is defined as a low-energy, low-frequency trotting in flat terrain and the environment consists of the

MuJoCo model of Laelaps II traversing level terrain. The low-frequency trotting is promoted as the task, since experimental work has shown that it presents stability issues due to significant body-roll during motion; as a result a restrain mechanism was developed at CSL [1] (Laelaps II cannot perform hip abduction). For the environment, the OpenAI gym [17] together with Stable Baselines 3 [18] were utilised. According to [19], Soft Actor-Critic (SAC) [20] is a good match for training legged locomotion tasks and as a result it was employed in the current work. Concerning the hyperparameters, the buffer size was 100000, the learning rate was 0.0003 and the batch size was 256, while all other parameters were kept at the default values of [18].

3.1 Reward

To achieve efficient trotting, the energy consumption of the robot should be included in the reward function. The total actuation energy (E_{tot}) comes as the sum of the mechanical part of the actuation energy (E_{act}) and the electric losses (E_{el}) [9]. These quantities can be calculated by integrating the corresponding power formulas in (6). For the integration the Simpson 1/3 [21] was used for a given timestep (dt) and period ($\Delta t = t_2 - t_1$). Note that $\tau_{m,i}$, $\dot{q}_{m,i}$, $R_{m,i}$ and $K_{T,i}$ are the i_{th} motor's torque, angular velocity, windings' resistance and torque constant, respectively.

$$E_{act} = \int_{t_1}^{t_2} \sum_{i=1}^{8} |\tau_{m,i}\dot{q}_{m,i}|\, dt \;\; \& \;\; E_{el} = \int_{t_1}^{t_2} \sum_{i=1}^{8} \left[\left(\frac{\tau_{m,i}}{K_{T,i}} \right)^2 R_{m,i} \right] dt \tag{6}$$

The total energy is not a good metric for measuring the agent's performance and reward compared to the Cost of Transport ($CoT = E_{tot}/(mg\Delta x)$), which represents the energy required to move a robot of unit mass for one meter. In the reward function (7), a simplified version of CoT is used since the robot's mass (m) and the gravity's acceleration (g) are constants. Furthermore, the aforementioned reward accounts for the distance traversed from the beginning of the episode (Δx_{ep}), not only the one in the current step. Finally, the small constant ϵ is introduced to avoid division by ~ 0 values, e.g.: when the robot stands still, which would destabilise the training process. A good value for the weight w_{en} has been found to be $3 \cdot 10^{-4}$.

$$rew_{en} = -w_{en} \frac{E_{tot}}{\Delta x_{ep} + \epsilon} \tag{7}$$

The second objective of the training task is to produce a policy that drives the quadruped forward without diverging from its goal. Laelaps II is an appropriate test-bed to evaluate the developed algorithm, since it cannot perform hip abduction which would help in stabilizing the robot in cases of increasing body roll/yaw angles. As a result, the terms (8) in the reward function are introduced to achieve the second objective. The weights w_x & w_y were set to 35 and 15,

respectively, to promote forward motion. Finally, the total reward (rew_{tot}) which is calculated on every agent's timestep is presented in (9).

$$rew_x = w_x \left(|x_{now} - x_{previous}|\right) \;\; \& \;\; rew_y = -w_y \left| |y_{now}| - |y_{previous}| \right| \tag{8}$$

$$rew_{tot} = rew_x + rew_y + rew_{en} \tag{9}$$

3.2 Action Space

As showcased in Section 2.3, the quadruped's planners accept various parameters that form the desired trajectory of each toe. The high-level controller, in this case the DRL policy, is responsible for setting these parameters. However, to achieve a desired gait some of them need to remain unchanged. With that said, the vertical radius (b) remains constant during motion as it is related to the toe's clearance from the ground. For the current scenario its value is chosen to be $0.04m$, for all legs to avoid toe-collisions with the ground during their respective swing-phases, due to potential moderate body-roll/pitch induced by the whole-body motion. Moreover, the stance and swing phases are equal and constant, at $0.5s$ each, to maintain a low-frequency trotting that the real robot can execute. Finally, the time-phase of either set of diagonal legs is the same and $T_{step}/2$ different from the other pair.

The rationale behind these choices is for the agent to learn parameters that effectively maximize the total reward of equation (9) and do not degrade the quality and synchronization of the desired gait. With this in mind, the agent selects the center (x_0, y_0) of the semi-ellipse and its horizontal radius (a). This center affects the body's pose and stability, while the horizontal radius is directly coupled to both the body's velocity and direction and thus to its kinetic energy. The resultant action space is given in equation (10), where L & R correspond to left and right legs while F & H to fore and hind ones, respectively.

$$\left[(x_0,\, y_0,\, a)_{RF}\,,\; (x_0,\, y_0,\, a)_{RH}\,,\; (x_0,\, y_0,\, a)_{LF}\,,\; (x_0,\, y_0,\, a)_{LH}\right] \in \mathbb{R}^{12\times 1}, \tag{10}$$

3.3 Observation Space

The observation space includes the standard outputs of the IMU [22] mounted on Laelaps II. Specifically, the linear (v_x, v_y, v_z) and angular $(\omega_x, \omega_y, \omega_z)$ components of the body's velocity are included along with its roll-pitch-yaw angles $(\theta_x, \theta_y, \theta_z)$. In accordance with [23] the observations time progress is also included in the observation space. In general to capture a motion, the sampling frequency is required to be at least two-times faster than the motion's frequency (see Nyquist-Shannon [24]). So, in the current work the observations are sampled ten-times faster than the fundamental frequency of the toe's motion (which is $f_{motion} = 2/T_{step} = 2Hz$), namely $20Hz$, to be on the safe-side and account for unexpected motion changes that may happen due to unmodeled dynamics. Then, the observation space is formed as in equation (11).

$$\left[(v_x, v_y, v_z)_{t_{-9\ldots now}}\,,\; (\omega_x, \omega_y, \omega_z)_{t_{-9\ldots now}}\,,\; (\theta_x, \theta_y, \theta_z)_{t_{-9\ldots now}}\right] \in \mathbb{R}^{90\times 1} \tag{11}$$

3.4 DRL Control Architecture

The described architecture in Section 3 along with the created model presented in Section 2 is adumbrated in Figure 4. The proposed architecture extends the standard OpenAI gym's framework by defining several callbacks and nested loops to execute multiple tasks in different frequencies. Specifically, the dynamics engine runs at $5kHz$ to ensure convergence and avoid zero-crossing errors. At the same time, the planner along with the low-level PV leg-controllers run at $500Hz$. The agent chooses an action at a much slower rate, namely $2Hz$, i.e., on every toe's touchdown or liftoff, since each pair of diagonal legs have $T_{step}/2$ time-phase shift between them. Note that this is the ideal case, i.e., trotting motion. When the agent chooses significantly different attributes for each leg, the aforementioned assumption is invalid. However, this happens only when it corrects the robot's pose and direction, not during the steady-state forward motion. The framework of Figure 4 is used for both training and testing.

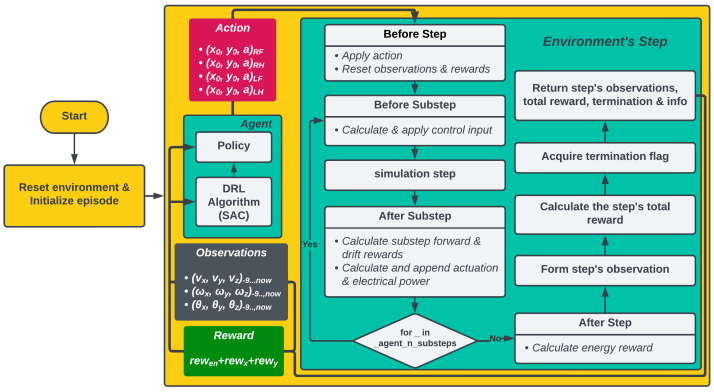


Fig. 4: Laelaps II DRL control architecture.

4 Results

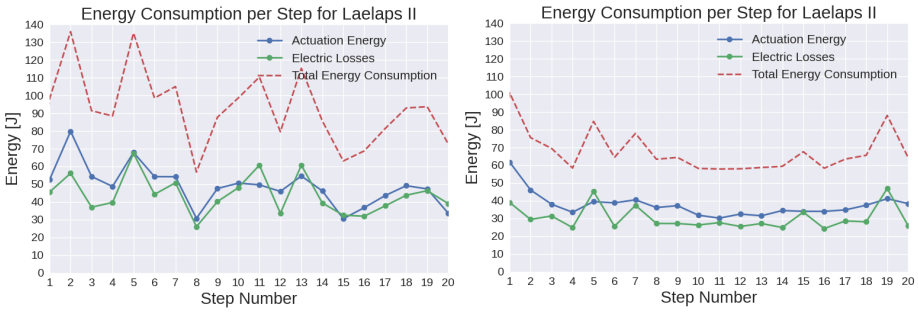
The presentation of the results section focuses on three points: (a) the reduction of the CoT when using the energy reward term presented in Section 3.1, (b) the improvement in the quality of the produced motion with the proposed reward function, and (c) the fact that the produced footsteps tend to be within a specific area of the leg's workspace. The last conclusion coincides with work presented in [9] and opens new research directions, showing that the proposed DRL scheme is a way to validate analytic approaches on energy efficiency for the Laelaps II.

First, the energy levels are presented *without* and *with* the proposed energy term in the reward function in Figures 5a & 5b, respectively. The lower energy

consumed together with the longer distance covered (see Figure 6a for the CoM's trajectory) in the second case results in a drop of $\sim 37\%$ in the CoT in favour of the developed reward function. Specifically, $CoT = 3.01$ in case of no energy term in the reward function compared to $CoT = 1.89$ when the energy term is introduced in the reward function).

The quality of motion is visualised in Figure 6. Specifically, in Figure 6a & 6b the CoM trajectory in the XY plane, i.e., top view in MuJoCo simulation environment, are presented for both simulation experiments. With the proposed reward function the robot moves almost in a straight line. In addition, the bounded yaw angle of the robot's body is shown in Figure 6b. The yaw angle depicts the direction of the robot's body which using the proposed reward function is heading towards the goal with maximum yaw angle $-0.05\,rad$.

Last, Figures 7a & 7b illustrate the footsteps, i.e., ellipse centers and ellipse a radii that the DRL control scheme produces. The fact that the agent is trained to reduce energy consumption for the given task, results in smoother motion with bounded body angles as well footsteps that are within a limited area of the legs workspace. The latter coincides with analytical studies presented in [9]. Figure 8a summarises the training procedure and the produced CoT from each learned policy. Similar approaches have also tried to reduce energy consumption by penalising joint acceleration or the mechanical part of the actuation power during a gait, but not the drivetrain's total energy demands (6). After training and testing them on Laelaps II, our approach still achieved the lowest CoT, i.e., 1.89 (see Figure 8b).

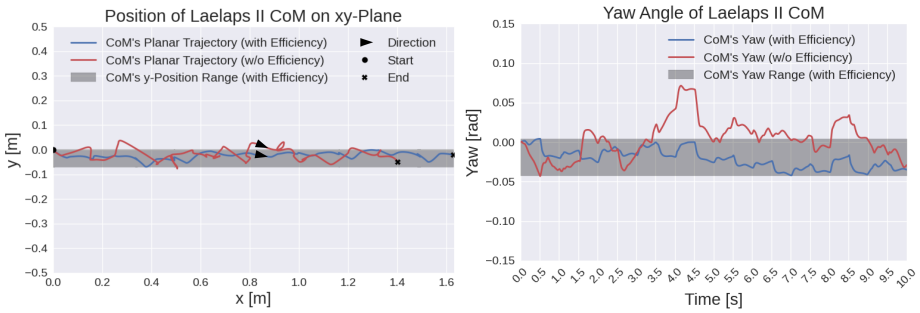


(a) (b)

Fig. 5: Energy levels (a) *without*, (b) *with* the proposed reward function.

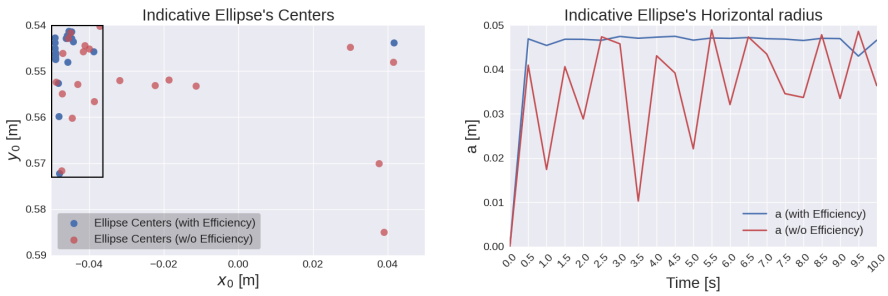
5 Conclusion & Future Work

This work described a systematic approach in producing energy efficient trotting motions using DRL. A detailed model of the Laelaps II quadruped was developed in MuJoCo and trained to move forward without diverging from its goal. A



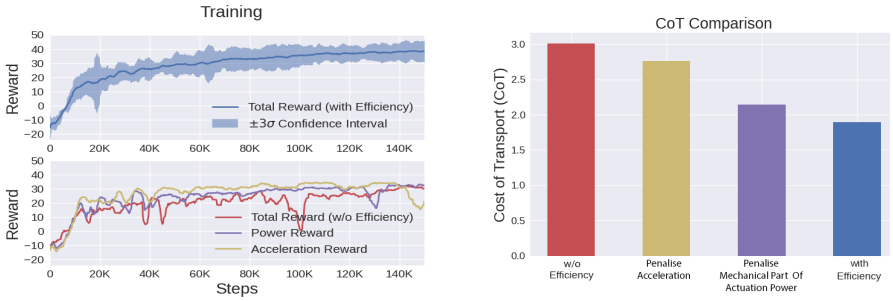
(a) (b)

Fig. 6: Motion quality. (a) CoM trajectories in top view. (b) Body yaw angle.



(a) (b)

Fig. 7: (a) The produced ellipse centers *with* the proposed reward function tend to be in specific area of the leg's workspace (black rectangle on the left). (b) The a radius also converges to a specific value.



(a) (b)

Fig. 8: (a) All algorithms were trained until the mean reward converged so that the quadruped could successfully reach the goal on level terrain. (b) Using the proposed reward function the lowest CoT was achieved.

term for energy efficiency was introduced in the reward function resulting in (a) smoother produced motions, (b) reduction in the CoT by $\sim 37\%$, and (c) smaller ranges in the body yaw angle, showing that the robot was not diverging from its goal. Additionally, the corrective actions needed during its transition to the goal were reduced. Last, the footsteps, i.e., ellipse centers and a radii, produced by the proposed DRL control scheme tended to be within a specific area of the legs workspace. Previous analytical studies reached similar conclusions [9].

Concerning future research directions, the gains of the low-level PV controllers can be also included in the DRL action space to investigate their effect on energy efficiency [11]. The training pipeline as well as the trained policies are open-source and available to download from the CSL's Bitbucket repository [25].

Acknowledgements The authors wish to thank K. Machairas for co-developing the quadruped robot Laelaps II. This work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant" (Project Number: 2182).

References

1. https://nereus.mech.ntua.gr/legged/laelaps-2-2/ (accessed Apr. 15, 2022).
2. X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine, "Learning Agile Robotic Locomotion Skills by Imitating Animals," in *Robotics: Science and Systems*, 07 2020.
3. T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine, "Learning to walk via deep reinforcement learning," in *Robotics: Science and Systems*, 2019
4. Z. Xie, P. Clary, J. Dao, P. Morais, J. Hurst, and M. van de Panne, "Iterative reinforcement learning based design of dynamic locomotion skills for cassie," *arXiv*, preprint arXiv:1903.09537, 2019
5. J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, p. eabc5986, Oct 2020.
6. Mastrogeorgiou, A., ElBahrawy, Y., Kecskeméthy, A., and Papadopoulos, E., "Slope Handling for Quadruped Robots Using Deep Reinforcement Learning and Trajectory Planning," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '20)*, Las Vegas, NV, USA, Oct. 25-29, 2020.
7. Atil Iscen, George Yu, Alejandro Escontrela, Deepali Jain, Jie Tan, and Ken Caluwaerts. Learning agile locomotion skills with a mentor. In 2021 *International Conference on Robotics and Automation (ICRA)*, 2021
8. J. Tan et al., Sim-to-Real: Learning Agile Locomotion For Quadruped Robots. *arXiv*, 2018. doi: 10.48550/ARXIV.1804.10332.
9. Koutsoukis, K. and Papadopoulos, E., "On the Effect of Robotic Leg Design on Energy Efficiency," *Proc. IEEE International Conference on Robotics and Automation (ICRA '21)*, Xi'an, China, May 30-June 5, 2021
10. A. Iscen et al., "Policies Modulating Trajectory Generators," 2019, *arXiv*, doi: 10.48550/ARXIV.1910.02812.

11. Machairas, K. and Papadopoulos, E., "An Active Compliance Controller for Quadruped Trotting," *24th Mediterranean Conference on Control and Automation (MED '16)*, Athens, Greece, June 21-24, 2016.
12. A. Papatheodorou, "Design & implementation of a real-time distributed EtherCAT-based motion control system for a multi-DoF quadruped robot", *Diploma thesis*, CSL, NTUA, Athens, 2021. [Online]. Available: https://dspace.lib.ntua.gr/xmlui/handle/123456789/53738
13. E. Todorov, T. Erez and Y. Tassa, "MuJoCo: A physics engine for model-based control," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026-5033, doi: 10.1109/IROS.2012.6386109.
14. J. Collins, S. Chand, A. Vanderkop and D. Howard, "A Review of Physics Simulators for Robotic Applications," in *IEEE Access*, vol. 9, pp. 51416-51431, 2021, doi: 10.1109/ACCESS.2021.3068769.
15. I. Virgala and M. Kelemen, "Experimental Friction Identification of a DC Motor," *International Journal of Mechanics and Applications*, vol. 3, no. 1, pp. 26–30, 2013.
16. K. Machairas and E. Papadopoulos, "An Analytical Study on Trotting at Constant Velocity and Height," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, Oct. 2018, pp. 3279–3284.
17. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J. & Zaremba, W. OpenAI Gym. (2016)
18. Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M. & Dormann, N. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research.* **22**, 1-8 (2021), http://jmlr.org/papers/v22/20-1364.html
19. Mastrogeorgiou, A., ElBahrawy, Y., Machairas, K., Kecskeméthy, A., and Papadopoulos, E.,"Evaluating Deep Reinforcement Learning Algorithms for Quadrupedal Slope Handling," *23rd International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, (CLAWAR 2020)*, Moscow, Russian Federation, 24-26 August 2020.
20. Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S., "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor". *International Conference on Machine Learning*, 2018.
21. Horwitz, A. "A Version of Simpson's Rule for Multiple Integrals.", *J. Comput. Appl. Math.*, 134, 1-11, 2001.
22. https://www.xsens.com/products/mti-100-series.
23. S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis, RLOC: Terrain-Aware Legged Locomotion using Reinforcement Learning and Optimal Control. *arXiv*, 2020. doi: 10.48550/ARXIV.2012.03094.
24. https://ocw.mit.edu/courses/2-161-signal-processing-continuous-and-discrete-fall-2008.
25. https://bitbucket.org/csl_legged/laelaps_rl_mujoco/ (accessed Apr. 15, 2022).