Identifying Conditions for Autonomous Dynamic Gait Transitions in Quadruped Robots

Emanuele Cuzzocrea[1] and Alberto Finzi[2]

[1]Master's Student, Università degli studi di Napoli Federico II
[2]Full Professor, Università degli studi di Napoli Federico II

***Abstract*—This paper presents a simple framework for managing dynamic gait transitions for a quadruped robot. Controlling a legged robot with different gaits is a fundamental problem, which reinforcement learning techniques are successfully addressing. Another very important aspect concerns how and when to perform transitions between gaits completely autonomously by the robot, thus without requiring manual switching by an operator. Here, the robot was trained to perform different gaits using reward machines technique. Meanwhile, a condition based on the height difference of the four feet, handled by an higher level control system, is used for autonomous gait switching. The ANYmal B quadruped from ANYbotics was trained using the Raisim simulator. Codes and videos are publicly available link.**

***Keywords*—Legged Robots; Reinforcement learning; Reward Machines; gait transition**

I. Introduction

LEGGED robots can perform a wide range of gaits and traverse both flat and highly irregular terrains at various speeds [4]-[5]-[6]. In this context, reinforcement learning techniques have proven to be extremely advantageous due to their effectiveness and simplicity in implementation [1]-[5]. Specifically, by only properly designing reward functions, it is possible to allow the robot to automatically learn the desired behavior. For this reason, in recent years, research has been focusing on finding the right synergy between the use of model-based techniques, such as model predictive control (MPC), and data-driven techniques, in order to fully exploit the advantages of each depending on the circumstances.

An equally important problem concerns how and when to perform transitions between various gaits [3]-[7]. In fact, this issue falls under the more general problem of policy switching management, regardless of the domain being addressed. Indeed, one of the key challenges in robotics beyond the learning a single specific task, is the ability to orchestrate many different behaviors, in such a way that the robot can be truly considered autonomous, and thus able to successfully tackle different tasks intelligently. For a legged robot, this problem of managing different tasks mainly involves the ability to decide, based on certain criteria, the appropriate gait to assume, and, of course, to perform the transition without falling.

In this work, the ability to reproduce a specific gait has been achieved using reward machine methods [9]. Learning a specific gait through a standard reward function is not something trivial, mainly because the reward function is basically a black box for the robot. Attempting to make it walk in a specific way by only exploiting standard state variables such as linear velocity, angular velocity, height, and encoder information is often difficult, and the results obtained are never completely satisfactory. Some techniques in the literature that address this problem are based on phase-guided controllers, which refer to a central pattern generator (CPG) [3]. These systems can be complex to implement and, in some cases, overly restrictive during the robot's learning process. What is needed is a simple guide, a small aid that makes the black box of the reward function a little less opaque. This is exactly what reward machines allow. What they do is simply extend the robot's state with variables belonging to a higher level of abstraction. These variables are essentially associated to finite state diagrams, and the final reward is designed to favor transitions from one state to another within the diagrams.

Regarding the ability to perform robust transitions between different gaits, there are several possibilities [3]. When dealing with very complex transitions, the most appropriate approach would be to conduct training specifically on the transitions, so that the robot can learn to manage all situations and change gaits without ever falling. In the case considered in this work, however, only standard gaits and low speeds were taken into account. Consequently, it was observed that to make the transitions robust, it was sufficient to make the individual gaits robust enough to recover control from difficult positions. Specifically, this was achieved thanks to training on uneven terrains generated randomly in each episode using the concept of Perlin noise, and by applying some disturbances during locomotion.

The main contribution regards on finding a simple condition for autonomous gait transitions. This is based on the height difference of the four feet. The goal was to enable the robot to autonomously realize that it is walking on an irregular terrain and thus to perform the transition without the help of a human operator. This condition can be easily tuned thanks to two parameters, that we can interpret as the minimum magnitude of irregularity that require the transition, and the length of such irregularity. Moreover, the focus will be mainly on the transition from the trot gait to the bound gait, but the same approach can be extended to all the other existing gaits as well.

II. Related Work

A. Reward Machines

Since the introduction of reward machines [9], numerous new scenarios have emerged. In particular, the ability to automatically learn the structure of the RMs themselves [29], the use of RMs for partially observable environments [30], multi-agent intention scheduling [24], and probabilistic RMs [25]. [1] proposes the use of RMs to manage the learning of different gaits for a quadruped robot. However, this article does not address a fundamental issue, namely how and when to transition between one gait and another.

B. Reference-Based RL Controllers

Reinforcement Learning has proven to be a very promising tool for the control of legged robots. Its advantage lies in the need to design reward functions accurately, allowing the robot to automatically learn the most appropriate behavior to maximize the reward through a trial and error mechanism. However, while conceptually simple, accurately designing a reward function can be challenging and time-consuming. This is why motion imitation techniques have emerged [19], combining RL with the use of priors, thus reducing the work of reward shaping. For legged robotics, references from motion-capture, human sketch, and model-based controllers can all serve as good references to train controllers.

C. Central Pattern Generators in Legged Robotics

In biology, central pattern generators are biological neural networks that can reproduce rhythmic signals in the body without rhythmic inputs [20]. They are thus the main resource for performing periodic tasks such as locomotion. In robotics, CPG is usually considered a mathematical framework for formulating periodic patterns [3]. Well-structured CPGs are excellent tools for performing various gaits and their transitions. There are many works in the literature regarding, for example, the structure of CPG networks [31], the mathematical forms [32], and the ability to transition between various gaits for a quadruped robot [27].

D. MPC-based

Various MPC-based approaches have successfully demonstrated the ability to perform different gaits with and without motion priors for legged robots [2]-[13]-[10]. However, these methods can require an accurate knowledge of dynamic models, and parameter tuning can be challenging. This is why the possibility of using RL techniques for tuning is being explored, so that this often tedious operation does not have to be done manually. It is important to note that the conditions proposed in this article for when to perform dynamic gait transitions can be applied independently of the type of control being used, whether it is data-driven or model-based.

Term Description	Definition	Weight
Linear Velocity x	$exp(-\|\|v_{d,x} - v_x\|\|^2)$	10
Linear Velocity y, z	$\|\|v_{y,z}\|\|^2$	-10
Angular Velocity x, y	$\|\|\omega_{x,y}\|\|^2$	-0.1
Angular Velocity z	ω_z^2	-25
Joint Torques	$\|\|\tau\|\|^2$	-0.00004
Joint Position	$\|\|q - q_{init}\|\|^2$	-0.1
Joint Velocity	$\|\|\dot{q}\|\|^2$	-0.01

TABLE I: Terms of R_{walk}

III. Method

A. Reinforcement Learning

The objective in reinforcement learning is to determine a policy $\pi : S \rightarrow A$ that selects actions to maximize the expected future discounted reward, given a state. It is worth remarking that, unlike dynamic programming problems that can be solved with straightforward value iteration and policy iteration techniques, in this context, the agent does not have access to the transition and reward functions. Therefore, the agent must learn just through trial and error interactions with the environment.

B. Reward Machines

Reward Machines (RMs) are used to manage milestone sub-goals in larger tasks, addressing issues of sparse or non-Markovian reward functions. The main idea behind RMs is that sub-goals are specified through a finite state automaton.

Formally, an RM is defined as the tuple $(U, u_0, F, \delta_u, \delta_r)$, where:

- U is the set of automaton states,
- u_0 is the initial state,
- F is the set of accepting states,
- $\delta_u : U \times 2^P \rightarrow U \cup F$ is the transition function,
- $\delta_r : U \times 2^P \rightarrow [S \times A \times S \rightarrow R]$ is the reward function.

Here, P is a set of propositional symbols representing high-level events. A labeling function $L : S \times A \times S \rightarrow 2^P$ determines the truth values of these symbols at each step. The agent uses δ_u to evaluate state transitions and δ_r to receive rewards.

RMs operate alongside the state space S, which describes the agent's low-level observations. To form an MDP from the non-Markovian reward, the agent considers observations from S and its current RM state from U. Training over $S \times U$ maintains the Markov property by indicating which sub-goal was completed.

C. RM for Quadruped Locomotion

Reward Machines can be used to specify the sequence of foot contacts required to perform a specific gait.

So that, in our case we can introduce the variable $P = \{P_{FL}, P_{FR}, P_{BL}, P_{BR}\}$, where $p \in P$ is a Boolean variable, indicating whether the front-left (FL), front-right (FR), back-left (BL), and back-right (BR) feet are in contact with the ground.

Here, R_{walk} encourages velocity command tracking while minimizing linear velocity in other directions, angular velocity,

and in general energy consumption. The reward function is specified in Table 1. The same reward function has been used for all considered gaits, only the finite state automaton associated to the RM was changed in order to decide the gait. This is a very appealing feature. In fact, it was possible to achieve different type of gaits, just by defining a single reward function. In fact, δ_r encourages transitions inside the finite state automaton since when a transition occurs the total reward is multiplied by a positive scalar. For all simulation the value of b has been set to 200. We can also note that the main goal of chosen the reward function is simply to enable the robot to achieve a certain desired speed along the x axis. In theory, it is a good practice to let the commanded speed to be a variable of the robot's state. During training, this variable should be changed so that the final policy is capable of tracking many different velocities within a certain range. Without loss of generality, only one speed along the x axis was trained, equal to $1.2\,\mathrm{m/s}$, and no speed along the y axis, nor any change in orientation, has been considered. After all, the primary objective of this work was to explore conditions that would allow the robot to perform transitions between gaits autonomously and robustly. The actual speed tracked by the robot is not crucial in this context.

Our state space $S = (u, \delta, q, \dot{q}, z, \phi, v, \omega, f)$ includes the current RM state u, the number of time steps δ since the previous RM state change, 12 joint angles q and velocities $\dot{q}$, the body height z, the body orientation ϕ, base linear and angular velocity v, ω, and f is a vector of four boolean variables, that are 1 if the corresponding foot is touching the ground, 0 otherwise.

D. Robust transition between gaits

In this work, the following three gaits were considered: trot, pace, and bound. The individual gaits were trained separately, and no transitions between the gaits were explicitly trained. What was done was to make each gait as robust as possible so that the policy switch could be performed instantaneously without the robot falling.

To make the transitions between different gaits more robust, a methodology was adopted that involves training the individual gaits on irregular terrains generated using Perlin noise, which changes randomly every 5 iterations (for reducing computational cost), and also applying external disturbances to the robot in the form of forces along the x, y, z axes at the origin of the base frame. This choice was motivated by the need to improve the robot's ability to regain control in difficult conditions and thus better manage the runtime gait transitions.

The irregular terrains created with Perlin noise introduce variability in the training environment, forcing the robot to learn to maintain balance and walk effectively on uneven surfaces. This adaptation process continues throughout the training duration, making the robot more robust and versatile. Additionally, the application of external disturbances in the form of forces along the x, y, z axes simulates real conditions where the robot might encounter obstacles or external forces that alter its trajectory or balance.

This approach results in an increased ability of the robot to execute smooth transitions between different gaits during runtime. Since the robot has already experienced a wide range of situations and perturbations during training, it is better prepared to handle sudden and unexpected changes in the operating environment. Consequently, the robot can maintain balance and direction even during gait transitions, reducing the risk of falls or loss of control.

An important consideration regarding the learning of individual gaits was the choice to use or not curriculum learning techniques for penalties. When starting learning completely from scratch, it happens quite often that the robot might not directly find the right balance between rewards and penalties, but instead, settle on a local minimum by remaining stationary, thus minimizing the penalties. To avoid this phenomenon, it is a very common practice that all penalties are usually multiplied by a factor λ, initially set to a low value, and increased by δ with each iteration. In this case, however, the use of MR can be considered has a very significant help for what concern the exploration towards the right type of locomotion, or at least to performing locomotion, without remaining stationary. So, we didn't feel the necessity to use curriculum learning in the training. By directly using all the penalties, without any scaling at the beginning, we can obtain a much more quick learning phase, and the actual type of locomotion is very similar to the desired one since the beginning.

Finally, it is worth noting that this approach is also valid beyond the context of simple gait transitions. In general, by making individual policies particularly robust, it is possible to effectively switch between different types of behaviors, such as braking, accelerating, manipulation tasks, or even higher-level behaviors [9].

E. Condition for autonomous gait transition

This part regards the main contribution of the paper. Once we explored how to make transitions between gaits robust, the next step was to understand when to make them so that the robot can be considered completely autonomously. In the following, we will consider mainly the case of the transition from the trot gait to the bound gait when the robot moves from a completely flat terrain to an uneven one. This is because the bound gait, being much more dynamic and energy-consuming than the trot, allows for great agility and speed in overcoming particularly rough terrains. The goal is to find a simple condition that enables the robot to switch from trot to bound after a certain interval of time on uneven terrain.

The condition that was developed concerns the height difference of the feet at the moment of contact with the ground. Specifically, a higher-level controller manages four variables, encapsulated in the vector $h = \{h_{FL}, h_{FR}, h_{BL}, h_{BR}\}$. These variables represent the height of each foot relative to the body frame the last time each foot touched the ground. At each step of the simulation, a check is performed to determine which feet are in contact with the ground. For all feet in contact, the corresponding variable h_i is updated. It is important to note that these variables are not part of the robot's state but are

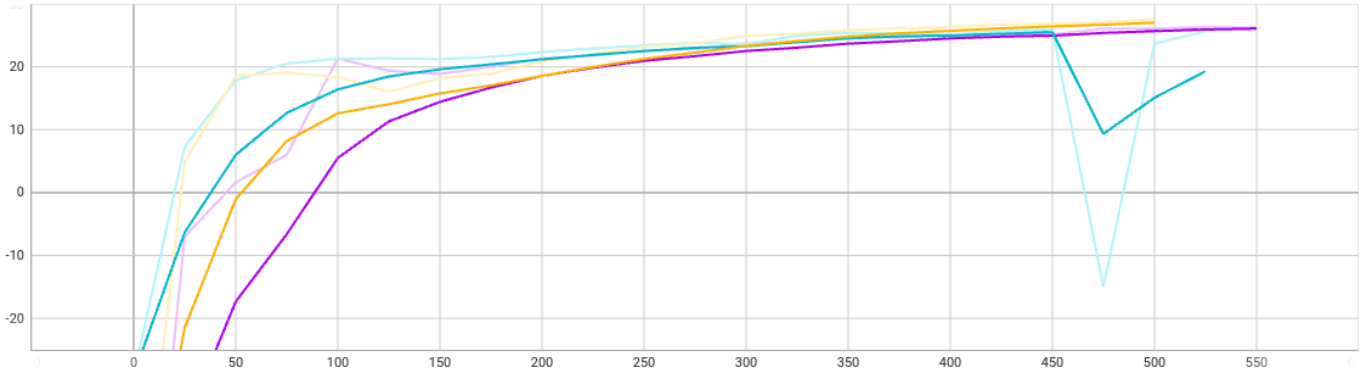

Fig. 1: Reward mean evolution for all three considered gaits. Legend: blue: trot; yellow: bound; violet: pace.

simply necessary for a higher-level controller to perform the policy change.

Now, at each step, the difference between all the values of h is calculated. Since there are only 4 variables, this results in simply 6 differences, and this computation is done in a straightforward manner with two nested for loops. If this difference is greater than a certain threshold, which we denote by α, then a counting variable that we denote by k is incremented by one (or in general by some small number ε). As soon as this counting variable exceeds a certain threshold β, a gait change is performed. This algorithm is better explained in the following pseudocode

Algorithm 1 Condition for autonomous gait transitions

```
Run at each step:
for all i, j do
    if |h_i − h_j| > α then
        k = k + 1
    end if
end for
if k > β then
    Change policy
end if
```

When the policy has to be changed, what happens is that the higher-level controller modifies the weights of the neural network (the algorithm used was Proximal Policy Optimization [8]), replacing those associated with the trot policy with those of the bound gait policy. In this way, the gait change is done completely autonomously, and through a simple computation based on the direct kinematics of the robot's legs. It is not important to know the robot's position in the world because, by knowing the robot's orientation and its height from the ground, we can always easily determine the height of the individual feet when they are in contact with the ground or not.

This simple algorithm is also extremely easy to tune. In fact, the parameter α can be seen as the magnitude of the minimum irregularity of the terrain that we want to consider. Choosing a small value for this parameter means that the robot will be more sensitive, and thus even in the presence of slightly irregular terrain, the counting variable k will be incremented. On the other hand, choosing a larger value allows the gait change to be performed only when the terrain is particularly rough. Additionally, this parameter can be modified online based on different circumstances. Regarding the parameter β, this represents how long it takes before the gait change is performed. Choosing it equal to 1 might not be a wise choice. In fact, a single pebble would be enough to modify the height of the contact points, thereby triggering the gait change. More generally, it might happen that the irregular area is still quite short, and thus it would not be worth performing the gait change.

F. Remark

First of all, although the techniques used for training the individual policies and the various transitions are closely related to the field of reinforcement learning, it is important to note that the condition on the difference in height of the robot's contact points is completely general. Even if the control used were model-based, such as an MPC, this condition could still be applied without significant differences.

Furthermore, in the following we will address the specific case of an autonomous gait change between the trot and the bound. However, without loss of generality, these concepts can be extended to any type of gait transition and for any type of terrain.

IV. Experiments

A. Implementation details

The control policy utilized in this study was designed as a multi-layer perceptron featuring three hidden layers with sizes of 256, 128, and 32 neurons respectively. Leaky ReLU activations were applied between each layer to ensure non-linearity. This control policy operates at a frequency of 100 Hz. For training purposes, the Raisim simulator was employed. In the simulation environment, 30 different versions of the robot were executed in parallel processes to accelerate the collection of diverse datasets. The value of the PD control gains are $K_p = 3$ and $K_d = 0.2$.

The training of the policy network was carried out using the Proximal Policy Optimization (PPO) algorithm [8]. This is an actor-critic algorithm, where alongside the policy network (actor), a separate network (critic) was trained. The critic's

role is to map the state to a singular scalar value, which serves as an estimate of the state's desirability. This scalar value is useful in reducing the variance of the reinforcement learning objective, thus enhancing the stability and efficiency of the training process.

During each training episode, the policy network is run for 100 steps to gather data for optimizing the objective function. An episode is terminated if the robot's body makes contact with the ground. The stable-baselines library, an open-source implementation of the PPO algorithm, was utilized to facilitate this process, whose parameters were all left at their default values. This rigorous training setup ensures that the policy network is robust and capable of handling various scenarios effectively.

B. Multiple Gait Locomotion

As shown in Figure 2,3,4, all three considered gaits: trot, bound, and pace, were successfully achieved. In Figure 5 it is shown the evolution of the corresponding rewards. All the gaits were trained for more or less 500 iterations. The gait that was learned the fastest was the trot. In fact, as it can be seen from the corresponding video, just after 100 iterations the robot is capable to move in the right way. For the bound gait, the corresponding reward is more or less in the middle between the other two. For what concern the pace gait, instead, this was the most difficult one. This is probably because the weight associated to the lateral velocity was to high, and since this gait is mainly based on lateral oscillations, the training required a little bit more time to converge. We can also notice that regarding the trot gait training, a large decrease in reward appeared after 450 iterations. This can happen sometimes, and it is mainly associated to random events, and to the hyperparameters choices. However, after that the reward tends immediately to recover the previous value. For safety reason, only for the trot gait the policy associated to the iteration number 450 has been used for all the other simulations.

It is worth remarking that these are the reward associated to a simple training on a flat ground without disturbances. In fact, all three gaits were firstly learned under easy condition. Then, they were made more robust with terrain randomization and applying external forces.

We can notice that the use of reward machines allows the robot to accelerate the learning process significantly. In most cases, after just 100 iterations, the robot has already learned the right movements. The use of penalties then allows for a more natural walking behavior. By limiting the joint velocities and their positions, it is possible to achieve a dynamic that with further domain randomization could be safely implemented on real hardware [5]-[18].

As for the transitions between different gaits, these were also successfully performed. The use of a rather limited forward speed and the training of the robot to regain control from difficult positions for all three gaits allows it to handle an instantaneous gait change without problems. Consequently, it can make a smooth transition from one gait to another without having been specifically trained during the training phase to change the type of gait. In the provided link it is possible to find a video where seven transitions in a row are performed without any kind of problem.

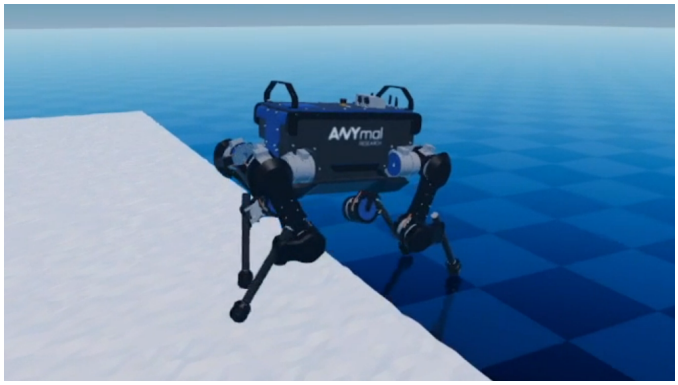

Fig. 2: Autonomous transition when reaching uneven terrain

C. Autonomous transition over uneven terrain

As shown in the corresponding video, the autonomous gait transition, once the robot is on an uneven terrain for a certain amount of time, was successfully achieved. The robot approaches the uneven terrain with a trot gait. Once on it, the variable k quickly starts to increase until it reaches the threshold β. In that particular case, the two parameters where set as $\alpha = 0.2, \beta = 1000$. At that point, the higher-level controller acts, modifying the neural network weights to activate the policy related to the bound gait. Consequently, the transition occurs effectively even on rough terrain, and the robot crosses that part of the map much more energetically than it could have with just the trot gait.

Additionally, it is worth noting that this autonomous gait transition algorithm can be easily modified. For instance, it is possible to introduce a reverse logic to allow the robot to recognize when it is back on flat terrain. Specifically, instead of imposing the condition $|h_i - h_j| > \alpha \quad \forall i, j$, one can impose a condition with the less-than sign $|h_i - h_j| < \alpha' \quad \forall i, j$. This way, the robot can return to a trot gait when it is back on flat terrain, as shown at the end of the corresponding video.

V. Conclusions

In this paper, the problem of how and when to manage transitions between different gaits autonomously and robustly for a quadruped robot has been addressed. The control of locomotion for individual gaits is based on training performed using deep reinforcement learning techniques. In particular, the learning of individual gaits was achieved through the help of reward machines. Transitions between gaits were made robust by randomizing the terrain and applying disturbances to the origin of the robot's base frame during training. The automatic gait transition was achieved through an higher-level control system. This control is based on the height difference of the robot's feet contact points with the terrain and can be easily adapted to different situations as it relies on only two parameters, which can be interpreted as the magnitude of the terrain irregularity and its length. Once a certain condition is met, this controller switches the policy by modifying the

weights of the neural network controlling the robot, allowing for a runtime change of the gait type.

References

[1] David DeFazio, Yohei Hayamizu, and Shiqi Zhang. "Learning quadruped locomotion policies using logical rules". In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 34. 2024, pp. 142–150.

[2] Viviana Morlando, Ainoor Teimoorzadeh, and Fabio Ruggiero. "Whole-body control with disturbance rejection through a momentum-based observer for quadruped robots". In: *Mechanism and Machine Theory* 164 (2021), p. 104412.

[3] Yecheng Shao et al. "Learning free gait transition for quadruped robots via phase-guided controller". In: *IEEE Robotics and Automation Letters* 7.2 (2021), pp. 1230–1237.

[4] Weitao Xi, Yevgeniy Yesilevskiy, and C David Remy. "Selecting gaits for economical locomotion of legged robots". In: *The International Journal of Robotics Research* 35.9 (2016), pp. 1140–1154.

[5] Michel Aractingi et al. "Controlling the Solo12 quadruped robot with deep reinforcement learning". In: *scientific Reports* 13.1 (2023), p. 11945.

[6] Joonho Lee et al. "Learning quadrupedal locomotion over challenging terrain". In: *Science robotics* 5.47 (2020), eabc5986.

[7] Yunho Kim, Bukun Son, and Dongjun Lee. "Learning multiple gaits of quadruped robot using hierarchical reinforcement learning". In: *arXiv preprint arXiv:2112.04741* (2021).

[8] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[9] Rodrigo Toro Icarte et al. "Using reward machines for high-level task specification and decomposition in reinforcement learning". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2107–2116.

[10] Donghyun Kim et al. "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control". In: *arXiv preprint arXiv:1909.06586* (2019).

[11] Alberto Camacho et al. "Reward machines for vision-based robotic manipulation". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 14284–14290.

[12] Donald F Hoyt and C Richard Taylor. "Gait and the energetics of locomotion in horses". In: *Nature* 292.5820 (1981), pp. 239–240.

[13] Jared Di Carlo et al. "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control". In: *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2018, pp. 1–9.

[14] Rodrigo Toro Icarte et al. "Reward machines: Exploiting reward function structure in reinforcement learning". In: *Journal of Artificial Intelligence Research* 73 (2022), pp. 173–208.

[15] Jemin Hwangbo et al. "Learning agile and dynamic motor skills for legged robots". In: *Science Robotics* 4.26 (2019), eaau5872.

[16] Trevor Drew, Stephen Prentice, and Bénédicte Schepens. "Cortical and brainstem control of locomotion". In: *Progress in brain research* 143 (2004), pp. 251–261.

[17] Magnus Gaertner et al. "Collision-free MPC for legged robots in static and dynamic scenes". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 8266–8272.

[18] Jie Tan et al. "Sim-to-real: Learning agile locomotion for quadruped robots". In: *arXiv preprint arXiv:1804.10332* (2018).

[19] Xue Bin Peng et al. "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills". In: *ACM Transactions On Graphics (TOG)* 37.4 (2018), pp. 1–14.

[20] Auke Jan Ijspeert. "Central pattern generators for locomotion control in animals and robots: a review". In: *Neural networks* 21.4 (2008), pp. 642–653.

[21] Wei Zhang et al. "Central Pattern Generators for Locomotion Control in Hexapod Robot Legs". In: *2022 34th Chinese Control and Decision Conference (CCDC)*. IEEE. 2022, pp. 3988–3993.

[22] Andrew Cohen, Lei Yu, and Robert Wright. "Diverse exploration for fast and safe policy improvement". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[23] Xingye Da et al. "Learning a contact-adaptive controller for robust, efficient legged locomotion". In: *Conference on Robot Learning*. PMLR. 2021, pp. 883–894.

[24] Michael Dann et al. "Multi-Agent Intention Progression with Reward Machines." In: *IJCAI*. 2022, pp. 215–222.

[25] Taylor Dohmen et al. "Inferring probabilistic reward machines from non-markovian reward signals for reinforcement learning". In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 32. 2022, pp. 574–582.

[26] Zipeng Fu et al. "Minimizing energy consumption leads to the emergence of gaits in legged robots". In: *arXiv preprint arXiv:2111.01674* (2021).

[27] Yasuhiro Fukuoka, Yasushi Habu, and Takahiro Fukui. "A simple rule for quadrupedal gait generation determined by leg loading feedback: a modeling study". In: *Scientific reports* 5.1 (2015), p. 8169.

[28] Siddhant Gangapurwala et al. "Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control". In: *IEEE Transactions on Robotics* 38.5 (2022), pp. 2908–2927.

[29] Cyrus Neary et al. "Reward machines for cooperative multi-agent reinforcement learning". In: *arXiv preprint arXiv:2007.01962* (2020).

[30] Rodrigo Toro Icarte et al. "Learning reward machines for partially observable reinforcement learning". In: *Advances in neural information processing systems* 32 (2019).

[31] Yasushi Habu, Keiichiro Uta, and Yasuhiro Fukuoka. "Three-dimensional walking of a simulated muscle-

driven quadruped robot with neuromorphic two-level central pattern generators". In: *International Journal of Advanced Robotic Systems* 16.6 (2019), p. 1729881419885288.

[32] Andres Espinal et al. "Design of spiking central pattern generators for multiple locomotion gaits in hexapod robots by christiansen grammar evolution". In: *Frontiers in neurorobotics* 10 (2016), p. 6.