Introducing FootstepNet: an Efficient Actor-Critic Method for Fast On-line Bipedal Footstep Planning and Forecasting

Clément Gaspard

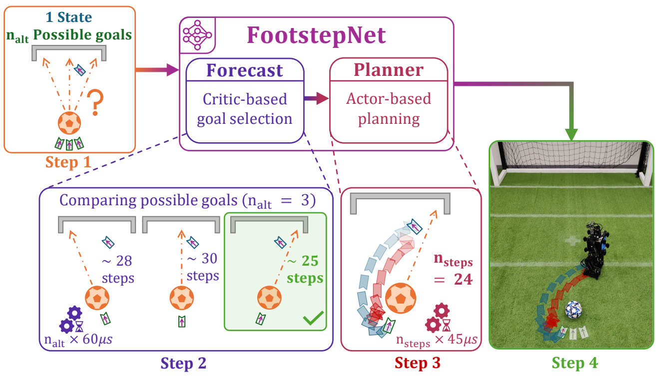

Fig. 1: An example of *FootstepNet* use – **Step 1**: A bipedal robot must score a goal while minimizing its number of steps. To do this, we arbitrarily choose n_{alt} placement possibilities (here $n_{alt} = 3$) which all allow scoring. **Step 2**: Forecasting allows choosing from the n_{alt} possibilities, the one that requires the fewest steps. **Step 3**: The planner compute all the steps in order to go to the position chosen by the forecast. **Step 4**: The step sequence is executed on the real robot.

I. Introduction

Designing a humanoid locomotion controller is challenging and classically split up in sub-problems. Footstep planning is one of those, where the sequence of footsteps is defined. Even in simpler environments, finding a *minimal* sequence, or even a *feasible* sequence, yields a complex optimization problem. In the literature, this problem is usually addressed by search-based algorithms (*e.g.* variants of A*). However, such approaches are either computationally expensive or rely on hand-crafted tuning of several parameters. In this work, at first, we propose an efficient *footstep planning* method to navigate in local environments with obstacles, based on state-of-the art Deep Reinforcement Learning (DRL) techniques, with very low computational requirements for on-line inference. Our approach is heuristic-free and relies on a continuous set of actions to generate feasible footsteps. In contrast, other methods necessitate the selection of a relevant discrete set of actions. Second, we propose a *forecasting* method, allowing to quickly estimate the number of footsteps required to reach different candidates of local targets. This approach relies on inherent computations made by the *actor-critic* DRL architecture. We demonstrate the validity of our approach with simulation results, and by a deployment on a kid-size humanoid robot during the RoboCup 2023 competition.

II. Problem Statement

Footstep planning consists in computing the footstep sequence such that the robot can move from an initial position to a target location efficiently and safely, all while avoiding obstacles and adhering to the physical limitations of the robot's mechanics and its environment. This is a critical aspect of bipedal humanoid robotics.

In this paper, we are interested in footstep planning within a two-dimensional (2D) framework. By constraining our consideration to the 2D pose of the robot—defined by coordinates (x, y) and orientation (θ) in a planar domain—we simplify the inherently complex problem of navigation in three-dimensional space. This approach allows us to effectively decompose the robot's trajectory into a series of planar movements. However, with the minimization of the number of footsteps in mind, local navigation can yield complex maneuvers as presented in Fig. 2.

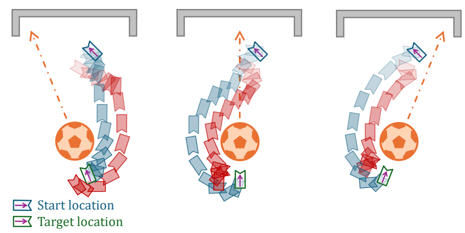

Fig. 2: Example of footsteps generated by *FootstepNet planning*.

III. Method

We define a footstep as $\phi = (f, x, y, \theta)$, where $f \in \{\text{left}, \text{right}\}$ indicates a specific foot and x, y and θ are the position and the orientation of the foot[1]. The robot state can be described with the footstep of its current support foot $\phi_r = (f_r, x_r, y_r, \theta_r)$. In case of double support, the choice of the support foot is arbitrary.

A footstep displacement $\Delta\phi = (\Delta x, \Delta y, \Delta\theta)$, is parametrized as on Fig. 4. It describes the pose of the swing foot in the frame of the support foot. When a support swap occurs, the swing foot becomes the new support foot, producing a new footstep. A sequence of footsteps can then be built from successive displacements, which defines the trajectory.

The displacements are bound in a feasible set $\Delta\phi \in \mathcal{F}$ because the robot has a limited workspace.

An obstacle is defined as $o = (x_o, y_o, \rho)$, where x_o and y_o are the position of the center of the obstacle and ρ is its radius. A collision between a footstep and an obstacle occurs if the rectangular support footstep intersects the circular obstacle.

Given a target $\phi_t = (f_t, x_t, y_t, \theta_t)$, the goal of the *footstep planning* problem is to find a sequence $\Phi_p = (\phi_r, \phi_2, \ldots, \phi_t)$ such that displacements are feasible, and with minimal length $|\Phi_p|$. This problem is non-linear because of the possible rotations of the robot. It also has non-convex constraints because of the obstacle avoidance, but also possibly because of the shape of $\mathcal{F}$. For those reasons, there are no known closed-form solutions.

We formulate it as an MDP which has a concise state and action spaces. This MDP is designed to have a reasonable training time using state-of-the art DRL algorithms. The trained agent has very fast on-board inference time, taking advantage of all modern hardware acceleration for neural network inferences.

[1]Unless specified otherwise, all the quantities are expressed in an inertial world frame attached to the ground

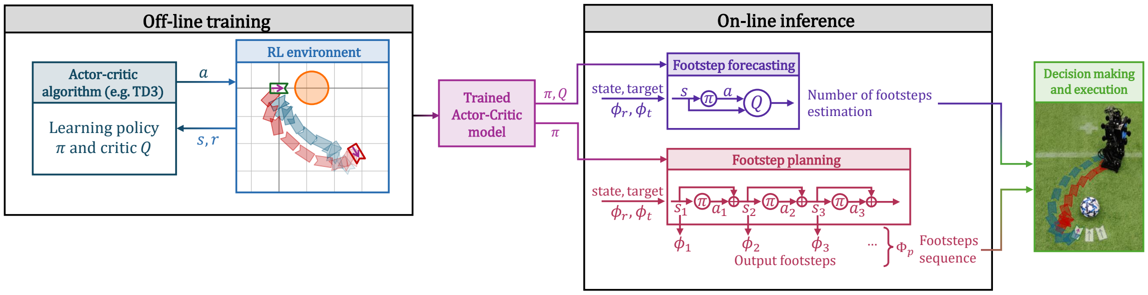

Fig. 3: Overview of the proposed method – First, offline training is carried out during which the agent learns the policy by interacting with the RL environment. During online inference, we then use the trained networks to, on the one hand, estimate the number of steps using the critic and, on the other hand, to determine the sequence of steps to be performed using the actor.

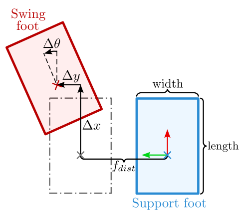

Fig. 4: Parametrization of a footstep displacement $(\Delta x, \Delta y, \Delta\theta)$. The displacement is a pose expressed in the frame of the support foot, with an implicit offset of f_{dist} in the y direction.

Moreover, taking advantage of the actor-critic architecture, the critic network is also an outcome of the RL optimization process. Since our reward lead to meaningful return unit (approximating $|\Phi_p|$), the critic can also be deployed on the robot to perform *footstep forecasting*. We believe this approach produces a useful building block for the whole locomotion controller.

The MDP formulation is as follows:

1) State-space: A state $s \in \mathcal{S} = \mathbb{R}^8$ is a tuple:

$$s = (\mathbf{1}_{f_r=f_t}, x_t, \sigma(y_t), cos(\theta_t), \sigma(sin(\theta_t)), x_o, \sigma(y_o), \rho), \quad (1)$$

where $\mathbf{1}$ is the indicator function : $\mathbf{1}_{f_r=f_t}$ taking the value of 1 if the robot support foot f_r is the target support foot f_t, and 0 else.

The quantities x_t, y_t, θ_t, x_o and y_o are expressed in the support foot reference frame when included in s. This allows for the current footstep ϕ_r to be omitted, reducing the state space dimensionality. Moreover, the $\sigma(y)$ operator, defined by $\sigma(y) = y$ if $f_r = right$ and $\sigma(y) = -y$ else, allows to handle the symmetry of the problem.

2) Action-space: An action $a \in \mathcal{A} = \mathbb{R}^3$ is a tuple: $a = \Delta\phi$, as specified in Fig. 4. The actions are clipped to lie in the feasible set $\mathcal{F}$. After applying back the symmetry operator σ on Δy and $\Delta\theta$, such a displacement can be integrated to obtain a new footstep.

3) Reward and termination: The reward function is expressed as:

$$R(s) = -1 - w_1\delta_p - w_2\delta_\theta - w_3\mathbf{1}_{s\in C}, \quad (2)$$

where δ_p and δ_θ are respectively the distance and absolute orientation error between the current and the target footstep. $\mathbf{1}_{s\in C}$ indicates if the current state is in a collision with the obstacle, C being the set of states in collision. $0 \leq w_1, w_2 \ll 1$ are reward-shaping weights intended to guide the learning and w_3 is a penalty weight. Every step taken in collision is the equivalent of taking w_3 extra steps, which is prohibitive for $w_3 \gg 1$. Reaching the target footstep, within a fixed tolerance yields a terminal state (which is equivalent to a subsequent return of 0).

The return obtained from a given state can be interpreted as the (negative) approximation of the number of footsteps required to reach the target. Given that the critic is an approximation of this return, it can then provide an estimation of the sequence length $|\Phi_p|$, which is useful for upstream decision-making. For this reason, the simplicity of the reward function is a key feature of *FootstepNet*. This approximation is valid if the shaping weights w_1, w_2 are small, and if the discounting factor γ is close to 1.

The sequence of planned footsteps $\Phi_p = (\phi_r, \phi_1, \phi_2, \ldots, \phi_H)$ can then be obtained by evaluating recursively the policy with a target horizon H. We call this process a *roll-out* of the policy. In practice, the size of the horizon H can be selected to produce the relevant number of footsteps for downstream whole-body planning and control. Alternatively, it is possible to apply this *roll-out* fully until the target is reached. $|\Phi_p|$ then becomes the number of required steps to reach the target. However, this requires one inference per footstep and is thus costlier than using the critic-based estimation.

IV. Experimentations and Conclusion

The comprehensive evaluation and deployment of FootstepNet have underscored its effectiveness and efficiency as a planner in bipedal robotics, particularly in comparison to the state-of-the-art ARA* planner. Through experimentation under various scenarios (detailed here[2]), including obstacle navigation and target reaching, FootstepNet has consistently demonstrated superior performance, achieving equal or better results 99% of the time, while boasting significantly lower execution times (ARA*: 5s/path vs FootstepNet: 45µs/step). The utilization of DRL with a continuous set of footsteps not only streamlines the planning process but also obviates the need for selecting a discrete footsteps set, a notable advantage over traditional methods. Additionally, the accurate forecasting capability of FootstepNet, as evidenced in both experimental setups and real-world competition scenarios such as RoboCup 2023 (that we won scoring 95 goals and taking only 2), highlights its potential for enhancing decision-making in robotics, enabling quick and reliable movements essential for success in dynamic environments.

FootstepNet represents a significant step forward in the domain of footstep planning, combining speed, efficiency, and accuracy in a manner not previously achieved by existing planners, to our knowledge. Its success in both controlled experiments and competitive environments attests to its utility and potential for broader applications.

Additionally, we provide a complementary video[3] about FootstepNet and demonstrates its application on a Sigmaban bipedal robot.

[2] https://arxiv.org/pdf/2403.12589
[3] https://youtu.be/EL1rJh45vug