2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

# CNN-based Foothold Selection for Mechanically Adaptive Soft Foot

Jakub Bednarek[1], Noel Maalouf[1], Mathew J. Pollayil[2], Manolo Garabini[2]
Manuel G. Catalano[3], Giorgio Grioli[3], Dominik Belter[1]

***Abstract*— In this paper, we consider a problem of foothold selection for the quadrupedal robots equipped with compliant adaptive feet. Starting from a model of the foot we compute the quality of the potential footholds considering also kinematic constraints and collisions during evaluation. Since terrain assessment and constraints checking are computationally expensive we applied a Convolutional Neural Network (CNN) to evaluate the potential footholds on the elevation map. We propose an efficient strategy for data clustering and segmentation with CNN. The data for training the neural network is collected off-line but the inference works on-line when the robot walks on rough terrains and allows for efficient adaptation to the terrain and exploitation of the properties of the soft adaptive feet.**

## I. Introduction

Walking robots, like animals, adapt to the terrain when walking on rough terrains. Robots can deal with small irregularities without careful foothold selection [1] and react dynamically to unexpected slippages. However, this approach is risky when the robot is climbing a rough and challenging environment. Additional feedback from the perception system allows analyzing the terrain profile and selecting the optimal position for the foot [2], [3]. In this scenario, the robot considers the elevation map of the terrain and modifies the nominal position of the foot taking into account the terrain slope and risky edges of the obstacles. Despite the visual feedback, the deliberative approach to footstep planning can be applied together with dynamic gaits [2].

Dynamic walking [1] and foothold adaptation based on visual feedback can be also supported by the mechanical adaptation of the robot. Even robotic manipulators use hands with compliant fingers which can adapt to the shape of various objects [4]. This approach improves the stability of the grasp with the mechanical feedback only. A similar method can be applied when designing the foot for a walking robot [5]. The mechanical design of the foot used in this research improves the properties of the contact between the robot and the terrain and increases the stability of the robot on rough terrain.

In this work, we combine two approaches and take advantage of mechanical and visual feedback. In this work, we employ the SoftFoot-Q adaptive foot on the ANYmal quadruped robot. SoftFoot-Q is a mechanically adaptive foot for quadrupedal robots inspired by [5]. These compliant feet increase the stability of the robot by improving grip and traction. The mechanical design compensates small irregularities of the terrain and the robot can walk on moderately rough terrains without visual feedback. However, when walking on risky and rough terrains, the robot should still avoid regions that are outside the workspace of the legs, avoid high slopes, collisions and select footholds on the obstacles which provide a better grip of the foot. To this end, in this work, we also utilize visual feedback to select footholds.

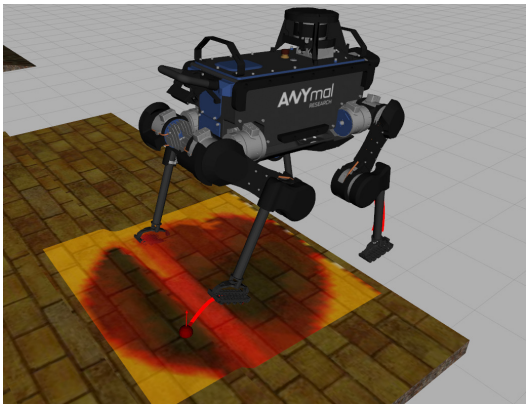


Fig. 1: Foothold selection procedure on the ANYmal robot: the CNN provides the cost of each foothold on the local patch of the terrain taking into account properties of the compliant foot and robot constraints

The ANYmal robot and the example scenario considered in this research are presented in Fig. 1. In contrast to our previous research [3], the robot is equipped with an adaptive soft foot. While planning the next step of the robot, the nominal position of the foot is computed which results from the desired length of the step and direction of motion. The nominal foothold is modified according to the local elevation map below the hip joint of the leg. The local elevation map is provided to the input of the neural network (CNN) that computes the cost map which is later used to find the best foothold (red sphere in Fig. 1) taking into account the shape of the terrain, motion constraints, and position of the nominal foothold.

## II. Related Work

Foothold selection plays an integral part in quadrupedal robot locomotion especially on challenging terrains. Most

This research was supported by EU Horizon 2020 project THING. This work has been conducted as part of ANYmal Research, a community to advance legged robotics. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

[1]Institute of Robotics and Machine Intelligence, Poznan University of Technology, Poznan, Poland
[2]Research Center "E. Piaggio", University of Pisa, Pisa, Italy
[3]Research Line SoftBots, Italian Institute of Technology, Genova, Italy
Contact: jakub.bednarek@put.poznan.pl, dominik.belter@put.poznan.pl

foothold cost evaluation methods utilize terrain properties such as height-maps, slopes, and slippage [6]. Barasuol et al. evaluate candidate footholds through visual pattern classification based on height-maps. The algorithm is tested on the quadruped HyQ on rough terrains with obstacles. The 3D map is built using a motion capture system and RGB-D sensor. During locomotion, the robot requests a local height-map around the nominal foothold and evaluates this candidate foothold through visual pattern classification [7].

Visual-based foothold selection requires long processing time, therefore most evaluation algorithms resort to other sensors such as LiDARs and force-torque sensors. In [8], footholds are evaluated based solely on height-maps. The costs are computed through matching with pre-selected terrain templates. Kolter et al. [9] present a foothold evaluation method based on terrain elevations and slopes for the quadrupedal robot "Little Dog". The algorithm consists of a two-level control strategy. The high-level controller evaluates foothold costs, while the low-level controller is responsible for calculating feet trajectories to reach the desired footholds. The controller aims at providing proper footholds for traversing rough terrains.

Real-time foothold evaluation is designed for traversing challenging terrains in the absence of prior information about the terrain properties. Fankhauser et al. [10] present a real-time foothold planning for the quadruped ANYmal that aims in traversing rough terrains with stairs. The updated elevation map is used to assess the traversability of nominal footholds, while the trajectory planner guides the foot to the selected safe location [10]. Villareal et al. present another real-time dynamic foothold adaptation algorithm based on self-supervised classification [2]. The proposed strategy enables the integration of visual information in terrain perception whilst maintaining online foothold evaluation.

All the approaches covered in the related work evaluate candidate footholds for point-type feet. In this paper we present a foothold selection algorithm for passive adaptive feet. In particular, we employed the SoftFoot-Q, a compliant foot for quadrupedal robots, which is loosely inspired by [5]. These feet are mounted on the quadruped ANYmal to provide increased compliance on different terrain types. The presented approach utilizes the foot geometric information in addition to terrain properties to select the most compatible candidate foothold.

The problem of foothold selection is similar to the problem of multi-fingerd grasping and was studied widely by the robotics community. Recent development in this field includes the method which uses local geometrical properties of the objects to find the acceptable positions of the fingertips on the object's surface [11]. The grasp configurations are trained from real examples. The collision and kinematic constraints are taken into account during the inference procedure. Recently, deep neural networks, such as the Convolutional Neural Networks (CNN) gained high popularity in robotics applications. In grasp, the CNN is applied to select feasible grasp and robotics finger positions on the object's surface using point clouds [12] or depth images [13].

Most approaches for the foothold selection are based on the local features computed for the terrain surface, such as the inclination of the terrain, roughness, and local curvature from the elevation maps [14]. These features are provided to the input of the simple neural network which was trained on the data provided by human experts. Another approach takes the elevation map and estimates a probability map that is related to the capability of each cell to provide stable support for the robot's feet [15]. The StarlETH robot is equipped with a haptic device on the feet, which explores and evaluates the potential footholds without human supervision [16]. The HyQ robot focuses more on the reflexes which stabilize the robot [17], and visual information about the terrain is used to place the foot on the terrain surface without avoiding risky footholds [7]. The robot corrects the nominal foothold positions according to the output from the visual pattern classifier applied on the terrain patches.

Great progress in the field of autonomous legged locomotion on rough terrain was done on the quadruped robot LittleDog [18]. The authors proposed a terrain scorer which computes the spatial relationship between a considered point and its neighboring points and then rejects points which are located on edges, large slope, the base of a cliff, or inside of a hole.

A learning-based method was proposed to evaluate terrain templates based on the human demonstration [8]. The terrain scorer approach is also adapted in [9], where the weights of geometric features of the terrain are obtained during training and then used for the footsteps planning. The first CNN classifier for the footholds, which efficiently evaluates constraints and a terrain patch during dynamic walking on rough terrain, has been proposed by Villareal et al. [2]. Then, we proposed the method which evaluates footholds for the ANYmal robot using ERFNet neural network architecture [3]. In this paper, we extend the previous method [3] and we show that neural network can be applied to evaluate footholds for a compliant and adaptive feet.

The foothold selection method for a six-legged robot is represented by the method implemented on the Lauron IV robot [19]. The foothold selection module considers points around an initial foothold and takes into account elevation credibility, the mean height, and the height variance of the cells. The six-legged Messor robot learns which points on the elevation map can provide stable support from simulation data [20]. Then, the trained Gaussian Mixture is used to select the footholds in the RRT-based motion planner [21]. The kinematic and self-collision constraints are also taken into account. However, this process significantly slows-down the foothold selection process.

### *A. Approach and Contribution*

In this work, we extend our previous approach [3] to footstep planning for a ball-like foot. We show that the proposed framework can be also applied to various feet models other than round-shaped feet. Additionally, we demonstrate how to design the function which evaluates each potential foothold for the soft adaptive foot. Moreover, we improve segmen-

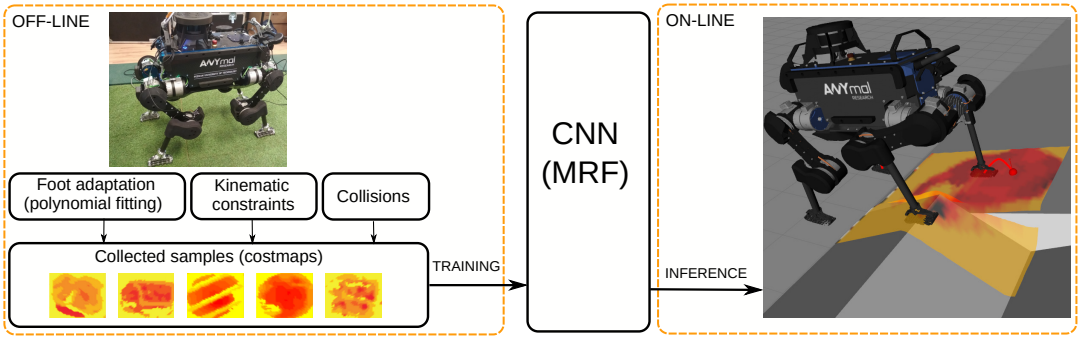


Fig. 2: Architecture of the proposed system: data collection and CNN training are performed off-line but inference is performed on-line when the robot walks on rough terrain

tation results provided by the neural network and provide a comparison to the other state-of-the-art results.

The main contributions of this paper include the following:

1) To the best of our knowledge, CNN that evaluates potential footholds for compliant feet which can mechanically adapt to the terrain profile has not been demonstrated before.
2) Analytical evaluation of the foothold cost which corresponds to the adaptation to the terrain patch and allows generating training data on the terrain map without the need for sampling terrain using the real robot.
3) Improved segmentation-based evaluation of the potential footholds on the elevation map using a CNN.

## III. Foothold Selection

### A. General architecture

The architecture of the proposed system is presented in Fig. 2. The core component of the system is the CNN neural network which evaluates the elevation map below the hip joint of the considered leg. Data for training the neural network are collected off-line. After training, the CNN is used to infer about the cost of potential footholds. We train two neural networks: one for the right front leg and one for the right hind leg. Because the mechanical design of the robot is symmetrical, we use the obtained neural network to find footholds for the left legs by flipping the local elevation maps and transforming the results to the frame of the left leg.

Neural networks need many training examples to find a proper relation between input and output data and generalize results. Collecting data from the real robot is difficult and time-consuming. It requires measuring the quality of the selected foothold (e.g. by measuring slippage). This problem can be solved in simulation [20] but the simulation of the soft foot used in this research is also slow because of its mechanical complexity. Instead, we propose the analytical function which describes how the feet adapt to the considered terrain patch and how far is the obtained configuration of the foot from the preferred one. Additionally, we check the workspace of the leg and collisions with the terrain to avoid constrained configurations of the robot when selecting footholds. We use the proposed method to generate pairs of input elevation maps and corresponding cost maps (Fig. 2).

During inference, we provide the local map of the terrain below the hip joint to the input of the neural network. The size of the local map is 40×40 cells and the size of each cell is 2×2 cm. CNN provides the cost map which is used to modify the nominal foothold – the desired position of the foot for a given step length and assuming that the robot is walking on flat terrain [3]. To find the best foothold we consider the cost $c_{\text{final}}$ which takes into account the output from the neural network $c_f$ and the distance from the nominal foothold $d_n$:

$$c_{\text{final}} = c_f + k \cdot d_n. \tag{1}$$

### B. Mechanically adaptive soft foot

The SoftFoot-Q is a novel passive robotic foot designed specifically for quadrupedal robots, loosely inspired by [5]. The core idea behind its design is that a mechanically adaptive sole can increase the support area, conform to the shape of the ground and thus aid in reducing slippage and enhance the stability of locomotion.

Fig. 3 highlights the main components and dimensions of the foot, which is a closed kinematic chain composed of the following components: i) an *ankle base* Ⓐ meant to be connected to the robot leg; ii) two *arch links* Ⓑ connected to the ankle by means of a revolute joint, which provides pitching movements on the longitudinal plane; iii) two *roll links* Ⓒ connected to the extremities of the arches through two revolute joints that make rolling motions possible along the frontal plane; iv) four *paddled chains* Ⓓ connecting the two roll links and creating an adaptable sole presenting a stiffening by compression behaviour: these flexible chains deform as they contact the terrain, conform to its shape and gradually become rigid in extension.

The roll joints are designed to be positioned as low as possible, to increase the static stability of the foot. Indeed, since all forces applied by the leg onto the foot have to pass through the roll rotation axis, a higher position of the roll joint would increase the chance for the force applied by the foot to lie outside the support area. This problem is exacerbated by the small width of the foot, which has to

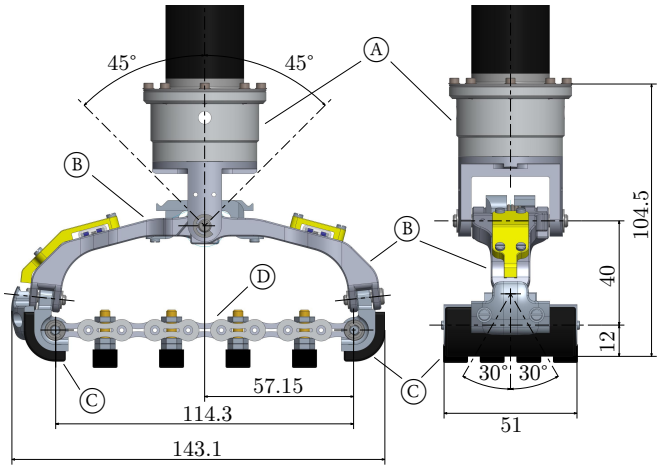


Fig. 3: CAD of the main components and dimensions of SoftFoot-Q.

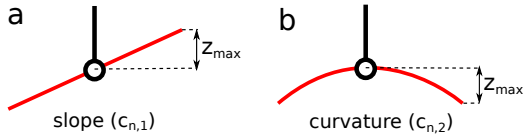


Fig. 4: Parameters of the model and corresponding shape of the foot: slope (a), curvature (b)

comply with specifications for its use on the quadrupedal robot ANYmal [22]. Finally, although the relative motion between the two arches tends to misalign the two roll links, roll motion is still allowed by the play between the chain links. Play is also of extreme importance in guaranteeing adaptiveness to variations of the terrain profile on the lateral direction.

### C. Foothold selection with polynomial fitting

The Pisa foot can adapt to the terrain profile. In the approach to foothold selection, we compute how the foot can fit the local shape of the terrain and how far the obtained configuration is from the preferred one. The example configurations of the soft foot are presented in Fig. 5. The preferred configuration of the foot is presented in Fig. 5b. since the mechanically adaptive sole has a stiffening-by-compression behaviour and it tends to act like a rope in a tension state when approaching a convex hull of the soil. We model the shape of the foot using a second-order polynomial:

$$y = \sum_{n=1}^{N} \sum_{m=0}^{M} c_{n,m} \cdot x_n^m, \tag{2}$$

where $x_n$ is the input coordinate ($x_1$ corresponds to the $x$ axis and $x_2$ corresponds to the $y$), $c_{n,m}$ is the constant value, $N = 2$, and $M = 2$.

During the polynomial fitting, we search for the values of the parameters $c_{n,m}$ which fit the given terrain patch. The example polynomials and corresponding foot shapes are presented in Fig. 5. We limit the values of the polynomial parameters $c_{n,m}$ to better represent the shape and constraints

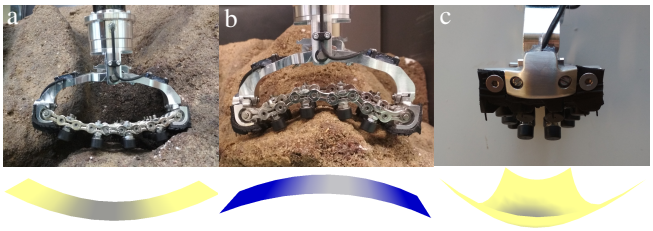


Fig. 5: Example foot configurations and corresponding surfaces for $c_{1,2} = 2.4$ (a), $c_{1,2} = -2.4$ (b), $c_{1,2} = -2.4$ and $c_{2,2} = 12$ (c)

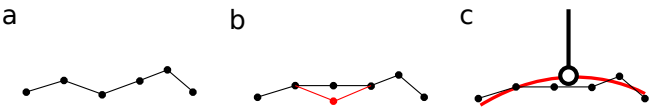


Fig. 6: Illustration of the convex hull computation: initial set of points representing the terrain surface (a), computation of the convex hull (b), and regression result (c)

of the real foot. The parameters of the polynomial have the physical meaning and it is easy to tune these values (Fig. 4). $c_{n,0}$ corresponds to the offset, $c_{n,1}$ corresponds to the slope value and $c_{n,2}$ represents curvature along $n$-th dimension. Finally, the cost of the foothold is evaluated using the following formula:

$$p_p = \alpha_R c_R + \sum_{n=1}^{2} \sum_{m=1}^{2} \frac{\alpha_{n,m}}{c_{n,m}^{\max} - c_{n,m}^{\min}} \cdot |c_{n,m}^{\text{pref}} - c_{n,m}| \tag{3}$$

where $\alpha_{n,m}$ are constants and $c_{n,m}^{\text{pref}}$ are preferred configuration of the foot. The preferred configuration of the foot is presented in Fig. 5b. The $c_R$ value represents the polynomial fitting to the terrain patch:

$$c_R = \sum_{x_1=L/2}^{L/2} \sum_{x_2=-W/2}^{W/2} |p(x_1, x_2) - elev(x_1, x_2)|, \tag{4}$$

where $W$ is the width of the foot $L$ is the length of the foot, and $p(x_1, x_2) - elev(x_1, x_2)$ is the difference between polynomial and elevation of the map for the $(x_1, x_2)$ coordinates.

The soft foot effectively compensates small irregularities of the terrain. Thus, we close small holes in the elevation map before the $c_R$ cost (4) is computed. To this end, we compute the convex hull on the 3D point cloud created by the local elevation map. We close holes smaller than two cells only. The procedure is illustrated in Fig. 6.

### D. Convolutional Neural Network

To estimate the cost of selecting the foothold, we initially engaged the solution proposed in [3]. Similarly, the elevation map ($40 \times 40$ cells) is provided to the input of the CNN which returns the cost for each cell. As in the above-mentioned work the regression problem was converted into a classification problem by foothold cost discretization. The discretization process was streamlined by automatically

aligning histograms composed of cost ranges and the number of pixels belonging to each of them. The ranges were created by iteratively adding pixels to the current range in ascending order (in terms of their cost value) until its size exceeds 5% of the total data set. The left end of the current range is initially defined at point 0, the right end is defined as the value of the last added pixel. For the next interval, the left end is the right end of the previous range. The schema is duplicated until the entire collection is divided into classes. Similarly, as in [3], additional classes weighting was performed. In this way, we obtained a data set balanced in terms of the number of occurrences of each class, which was used to teach the prediction model.

The Efficient Residual Factorized ConvNet (ERF) originally proposed in [23] was used to learn the foothold cost. Despite the conversion of the estimation problem to the classification problem, the goal is to minimize the error on the output of the network. Due to the nature of the data, the situation in which the model confuses adjacent classes (adjacent classes represent similar cost) is preferred over the situation in which distant classes are confused. For this purpose, we propose an additional weighting of the loss function, which multiplies the error by the absolute distance between the predicted and the ground truth class.

Because the identifier of the class is proportional to the cost, we can model the co-occurrence relations using known probabilistic mechanisms. Hence, we proposed to use Markov Random Fields (MRF) [24] on top of the ERF model. The formula below defines the MRF operating scheme:

$$P(x) = \frac{1}{Z} \exp\left(-E(x)\right)$$

$$E(x) = \sum_{i} \Phi(y_i) + \sum_{i,j} \Psi(y_i, y_j)$$

$q = ERF(Image)$;
$\Phi = -\log(q)$;
**while** *q not converged* **do**
  $\Psi = conv(q, kernel)$;
  $U = -(\Phi + \Psi) * s$;
  $q = softmax(U)$;
**end**
return $q$

**Algorithm 1:** Mean field approximation is used on top of the ERF model to estimate probability according to the Markov Random Field. The *Kernel* is a randomly initialized trainable tensor, with zero values in diagonal (non-trainable), whereas $s$ is a trainable scaling factor. During training, only one iteration of the loop is performed to propagate the error.

where $P$ is a probability function of the image segmentation, $x$ is the input data (grayscale image), $Z$ is a normalizing term, and $E$ refers to *energy function*. The *energy function* is composed of $\Phi$ – unary and $\Psi$ – pairwise costs. To adapt MRF into our model we used *mean field approximation* method in which we treat negative log of the output from the ERF model (probability distribution for each pixel) as $\Psi$. The output of MRF is estimated using algorithm 1.

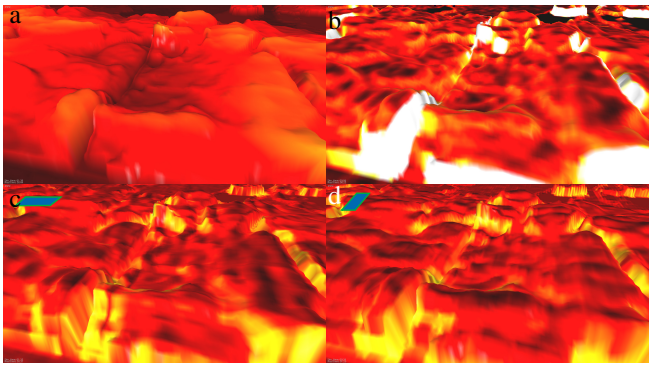


Fig. 7: Comparison between two foothold selection methods: example terrain (a), foothold selection method for a ball-like foot [3] (b), and polynomial fitting method for example orientations of the foot (c,d). Dark red color represent good footholds, yellow weak footholds

## IV. Results

In the first experiment we compute the cost function for the elevation map presented in Fig. 7. We use (3) to determine the cost map. The defined function is anisotropic so the cost value for each cell depends on the orientation of the foot. In Fig. 7c and Fig. 7d we show how the cost map computed using (3) changes when the orientation of the foot changes by $\pi/2$. We also compare the proposed method to the method from [3]. The results are shown in Fig. 7b. When the method from [3] is applied, the robot avoids small bumps and selects flat patches or small concavities. In contrast, the robot equipped with soft foot prefers small bumps but also accepts flat terrain.

In order to justify the application of the proposed neural networks we present qualitative results of the ablation study. We compare the following three neural network configurations:

- ERF with default segmentation loss function used in [3],
- ERF with difference weighting,
- MRF on top of ERF with difference weighting.

As in the paper [3], we note the metrics: accuracy and intersection over union (IoU). Both metrics were performed between the discretized ground-truth cost assignments and the classification result of the neural network. The given metrics provide information about the accuracy of the predictions of each class separately and how different areas (in terms of class) are confused with each other. The results are presented in Table I. MRF has significantly improved the results, with an increase in the IoU metric to 66.22.

Detailed results for the MRF and ERF models are presented in Table II. The summary shows a clear improvement for the MRF model.

| | Accuracy [%] | IoU |
|---|---|---|
| ERF basic | 88.85 | 58.32 |
| ERF diff weighting | 89.23 | 60.59 |
| MRF over ERF | **89.89** | **62.22** |

TABLE I: Comparison between neural network architectures on the validation dataset.

| | MRF | | | | | ERF | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 |
| 0 | 0.916 | **0.058** | 0.002 | 0.001 | **0.000** | **0.923** | 0.059 | 0.002 | 0.001 | 0.001 |
| 1 | **0.043** | 0.869 | 0.070 | 0.001 | 0.001 | 0.055 | **0.870** | **0.062** | 0.001 | 0.001 |
| 2 | **0.001** | **0.050** | **0.867** | 0.065 | 0.001 | 0.002 | 0.068 | 0.857 | **0.057** | 0.001 |
| 3 | 0.000 | 0.001 | **0.058** | **0.873** | **0.043** | 0.000 | 0.001 | 0.078 | 0.851 | 0.048 |
| 4 | 0.000 | 0.000 | 0.002 | **0.085** | **0.858** | 0.000 | **0.000** | 0.002 | 0.100 | 0.838 |
| 5 | **0.005** | 0.001 | 0.001 | 0.002 | **0.072** | 0.008 | 0.001 | 0.001 | 0.002 | 0.090 |
| 6 | **0.001** | **0.004** | **0.050** | **0.049** | **0.012** | 0.002 | 0.006 | 0.057 | 0.050 | 0.014 |
| 7 | **0.001** | 0.001 | **0.004** | 0.006 | **0.047** | 0.002 | 0.001 | 0.005 | **0.005** | 0.052 |
| 8 | 0.001 | **0.001** | **0.008** | **0.011** | **0.014** | 0.001 | 0.002 | 0.011 | 0.012 | 0.016 |
| 9 | 0.002 | **0.001** | **0.014** | 0.022 | **0.014** | 0.002 | 0.002 | 0.020 | 0.022 | 0.016 |
| 10 | 0.000 | 0.000 | **0.001** | 0.003 | **0.004** | 0.000 | 0.000 | 0.002 | **0.002** | 0.005 |
| 11 | 0.000 | 0.000 | 0.001 | 0.001 | 0.001 | 0.000 | 0.000 | 0.001 | 0.001 | 0.000 |
| 12 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 13 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

TABLE II: Slice of the confusion matrix for classes 0 to 4. The results show a comparison of the ERF model and the ERF with MRF module built on it. Bold numbers indicate a better result.

We verified the foothold selection algorithm on the various obstacle types in the Gazebo simulation environment which has the same control interface as the real ANYmal robot[1]. The example experimental sets are presented in Fig. 8. We checked the algorithm on stairs, rough terrains, bumps, and various geometric structures. The robot uses a locomotion planner proposed by Fankhauser et al. [10]. We replaced the module for the foothold selection. The direction of the robot's motion is given by the human operator. Then, the robot adapts the position of the feet and optimizes the pose of the robot's body to climb obstacles. This adaptation strategy slightly modifies the direction of motion given by the operator. It is noteworthy that at the present stage the compliance of the feet is not taken into account in the simulation because of its mechanical complexity, which makes it difficult to simulate contacts for the numerous links each foot is made of.

The example elevation maps, obtained cost maps and selected footholds are presented in Fig. 9. The preferred footholds are represented on the cost maps by dark red cells. The selected foothold (red sphere) is always a compromise between the preferences given by the neural network and the position of the nominal foothold. Thus, the robot does not select the cell with the lowest value in the cost map.

The first two examples show the behavior of the robot on the stairs (Fig. 9a and Fig. 9b). The robot avoids the bases of the step due to collisions. In contrast to our previous work, when we considered ball-like foot [3], the robot prefers positions that are closer to the edges. In this case, the robot remains stable even if the adaptive foot is placed on the edge of the step. Similar behavior can be observed when the robot

[1]short video from experiments is available at `https://youtu.be/p6CvUnqtkqQ`

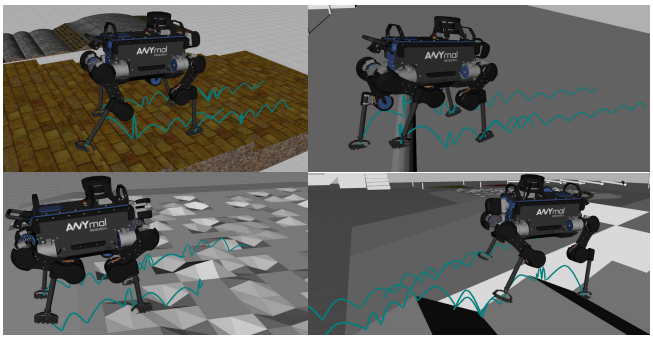

Fig. 8: Obstacles and feet trajectories during experiments with the CNN-based foothold selection for the adaptive soft foot

deals with a single step. Results are presented in Fig. 9c and Fig. 9d. In contrast to the robot equipped with a ball-like foot, the current planner has a larger set of feet positions on the same terrain, which are acceptable and stable for the robot.

The example cost maps obtained during the experiment on rough terrain presented in Fig. 9e and Fig. 9f show that robot prefers small bumps and avoids concavities. This behavior is expected due to the parameters of the preferred configuration of the foot defined in (3). The soft foot adapts better to small bumps which provide better support for the robot. In the examples presented in Fig. 9g the robot selects the foothold on a slope even though the cost of the foothold in this region is high. This happens because all footholds in this area provide weak support. As a result, the robot selects the foothold which is close to the nominal position of the foot. In the last example in Fig. 9h, the robot selects the preferred foothold on the edge because it is in the range of the considered leg. In all examples presented in Fig. 9 the low-cost footholds are inside circular region. This shape corresponds to the workspace of the robot which is used to define motion constraints while training the neural network.

Additionally, we compare the operating times of the previous model (ERF) and the current one (MRF). Two tests were carried out, for GPU and CPU. The GPU used is NVIDIA GTX 1660, while the CPU is Intel Core i7-9750H CPU with 2.60GHz processor clock. For the purposes of the test, in both cases, the models received batch data with a size of 32. The ERF model achieved a result of 33.82 ms on the CPU and 5.76 ms on the GPU, while the MRF reached 34.43 ms and 5.98 ms, respectively. Tests were repeated 100 times. The difference between the models is very small and allows for real-time operation in both cases.

## V. Conclusions and Future Work

In this paper, we propose a novel foothold selection method for legged robots equipped with passive compliant feet. The proposed method utilizes the neural network which predicts the cost map on the elevation map for each foot. The neural network considers the local shape of the terrain and parameters of the foot. It also stores information about

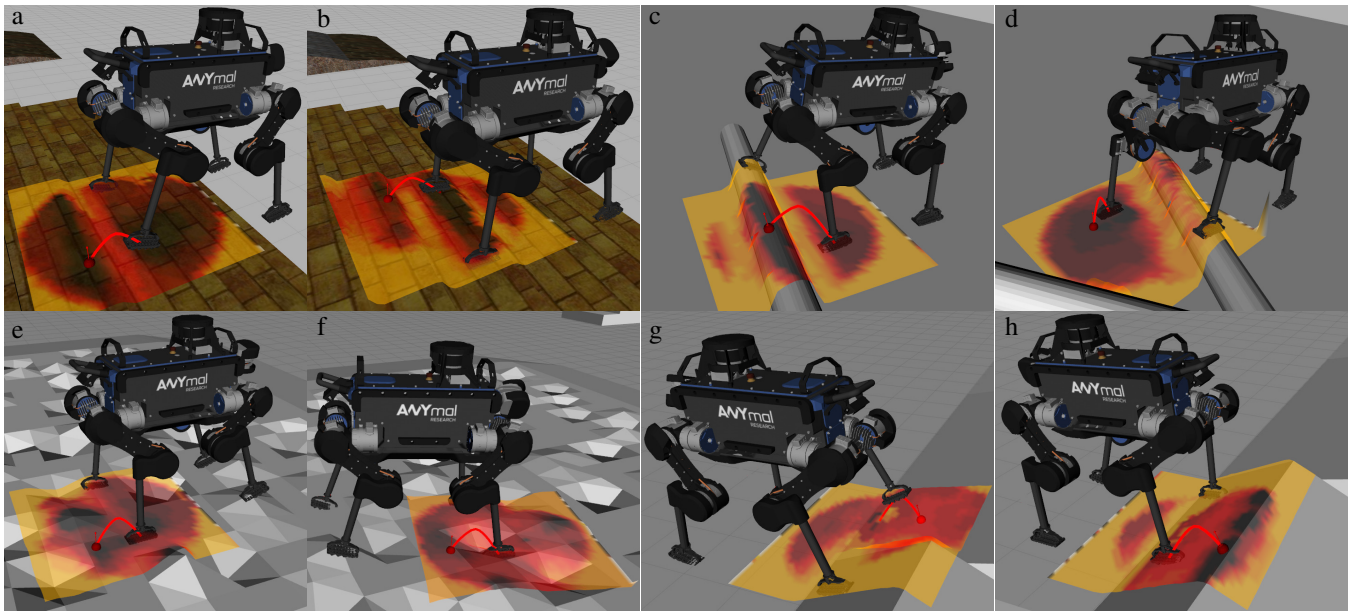


Fig. 9: Example footholds during walking on irregular terrain: hot color map patch represents evaluated region below hip joint (dark red – good foothold, yellow – week foothold, red sphere – selected foothold)

the kinematic model of the foot and can predict collisions between the leg and the terrain.

To avoid time-consuming data collection we propose the cost function which computes how the foot can adapt to the given terrain patch. We also take into account the distance from the desired configuration which guarantees the best grip. The proposed cost function allows generating training data for the neural network.

Finally, the obtained neural network is used on-line to select footholds for the quadruped robot ANYmal on various obstacles. With the proposed method the robot avoids risky configurations and selects footholds that provide stable support. In the future, we are going to define the CNN-based method which selects footholds and configuration of the robot in a single step. Moreover, we will also study and validate through rigorous experiments the impact of our approach on the performance of the robot on rough terrain.

## REFERENCES


[1] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert, "Autonomous navigation for bigdog," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 4736–4741.

[2] O. A. Villarreal, V. Barasuol, M. Camurri, L. Franceschi, M. Focchi, M. Pontil, D. G. Caldwell, and C. Semini, "Fast and continuous foothold adaptation for dynamic locomotion through CNNs," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2140–2147, 2019.

[3] D. Belter, J. Bednarek, H.-C. Lin, G. Xin, and M. Mistry, "Single-shot foothold selection and constraint evaluation for quadruped locomotion," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7441–7447.

[4] M. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi, "Adaptive synergies for the design and control of the pisa/iit softhand," *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 768–782, 2014.

[5] C. Piazza, C. Della Santina, G. M. Gasparri, M. G. Catalano, G. Grioli, M. Garabini, and A. Bicchi, "Toward an adaptive foot for natural walking," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 1204–1210.

[6] A. Winkler, I. Havoutis, S. Bazeille, J. Ortiz, M. Focchi, R. Dillmann, D. Caldwell, and C. Semini, "Path planning with force-based foothold adaptation and virtual model control for torque controlled quadruped robots," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6476–6482.

[7] V. Barasuol, M. Camurri, S. Bazeille, D. G. Caldwell, and C. Semini, "Reactive trotting with foot placement corrections through visual pattern classification," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5734–5741.

[8] M. Kalakrishnan, J. Buchli, P. Pastor, and S. Schaal, "Learning locomotion over rough terrain using terrain templates," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 167–172.

[9] J. Z. Kolter, M. P. Rodgers, and A. Y. Ng, "A control architecture for quadruped locomotion over rough terrain," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 811–818.

[10] P. Fankhauser, M. Bjelonic, C. D. Bellicoso, T. Miki, and M. Hutter, "Robust rough-terrain locomotion with a quadrupedal robot," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[11] M. Kopicki, R. Detry, M. Adjigble, R. Stolkin, A. Leonardis, and J. L. Wyatt, "One-shot learning and generation of dexterous grasps for novel objects," *The International Journal of Robotics Research*, vol. 35, no. 8, pp. 959–976, 2016.

[12] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *Robotics: Science and Systems*, 2017.

[13] M. Gualtieri, A. Ten Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 598–605.

[14] E. Krotkov and R. Simmons, "Perception, planning, and control for autonomous walking with the ambler planetary rover," *The International journal of robotics research*, vol. 15, no. 2, pp. 155–180, 1996.

[15] Chun-Hung Chen and V. Kumar, "Motion planning of walking robots in environments with uncertainty," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, April 1996, pp. 3277–3282 vol.4.

[16] M. A. Hoepflinger, M. Hutter, C. Gehring, M. Bloesch, and R. Siegwart, "Unsupervised identification and prediction of foothold robustness," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3293–3298.

[17] M. Focchi, R. Orsolino, M. Camurri, V. Barasuol, C. Mastalli, D. Cald-

well, and C. Semini, "Heuristic planning for rough terrain locomotion in presence of external disturbances and variable perception quality," 2018.

[18] J. R. Rebula, P. D. Neuhaus, B. V. Bonnlander, M. J. Johnson, and J. E. Pratt, "A controller for the littledog quadruped walking on rough terrain," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 1467–1473.

[19] A. Roennau, T. Kerscher, M. Ziegenmeyer, J. Zoellner, and R. Dillmann, "Six-legged walking in rough terrain based on foot point planning," in *Mobile robotics: solutions and challenges*. World Scientific, 2010, pp. 591–598.

[20] D. Belter and P. Skrzypczyński, "Rough terrain mapping and classification for foothold selection in a walking robot," *Journal of Field Robotics*, vol. 28, no. 4, pp. 497–528, 2011.

[21] D. Belter, P. Łabecki, and P. Skrzypczyński, "Adaptive motion planning for autonomous rough terrain traversal with a walking robot," *Journal of Field Robotics*, vol. 33, no. 3, pp. 337–370, 2016.

[22] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, *et al.*, "Anymal-a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 38–44.

[23] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 19, pp. 263–272, 2018.

[24] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2479–2486.