2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Adaptive Traversability of Unknown Complex Terrain with Obstacles for Mobile Robots

Karel Zimmermann[1], Petr Zuzanek[2], Michal Reinstein[3], and Vaclav Hlavac[4]

***Abstract*— In this paper we introduce the concept of Adaptive Traversability (AT), which we define as means of autonomous motion control adapting the robot morphology—configuration of articulated parts and their compliance—to traverse unknown complex terrain with obstacles in an optimal way. We verify this concept by proposing a reinforcement learning based AT algorithm for mobile robots operating in such conditions. We demonstrate the functionality by training the AT algorithm under lab conditions on simple EUR-pallet obstacles and then testing it successfully on natural obstacles in a forest. For quantitative evaluation we define a metrics based on comparison with expert operator. Exploiting the proposed AT algorithm significantly decreases the cognitive load of the operator.**

I. INTRODUCTION

Tracked robots with several articulated parts such as legs or subtracks—referred to as flippers, see Fig. 1—have been studied intensively since the design of robot morphology directly influences the ability to traverse complex terrain, especially with natural unstructured obstacles. Possessing a high number of articulated parts inevitably yields more degrees of freedom that have to be controlled. To reach a suitable pose to traverse such terrain in a safe way may become easily intractable, even for an expert operator. Controlling such many degrees of freedom also requires more time and poses a significant cognitive load onto the human operator. This may have crucial effect on the success of any Search & Rescue mission, that we primarily aim for [1], as well as influence on the robot safety.

We call this task *Adaptive Traversability* (AT), which we define as means of autonomous motion control adapting the robot morphology (configuration of flippers and their compliance) to traverse unknown complex terrain with obstacles in an optimal way. Our metrics for optimality is based on comparison of the AT autonomous regime to the control of an expert operator with respect to time taken for traversal, robot safety and smoothness of transitions. Beside having an edge in these criteria, the ultimate merit of using AT lies in minimal cognitive load for the operator.

Many approaches focus on optimal robot motion control in an environment with a known map, leading rather to the research field of trajectory planning. In contrary to planning [2] [3], the AT can easily be exploited in unknown environment and hence provide a crucial support to the actual procedure of map creation. From the conceptual point of view, the AT is intended to run one level below any SLAM or trajectory planning algorithms and its input commands can either be directly from the operator (usually unknown-map case) or from a planner. We would like to emphasize to perceive AT rather as independent complement to trajectory planning and in no way a substitution. If the task of AT was to be solved by means of trajectory planning, a reliable map is required, providing detailed information on Robot-Terrain Interaction (RTI) (e.g. estimation of stability, slippage coefficient, power consumption, robots full 3D pose etc). Such RTI can be theoretically estimated from the terrain shape and a physical model of the robot and used to build traversability maps [4] [5]. However this modeling is analytically very complex, computationally demanding and in specific cases such as high slippage or aerial motion phases often inaccurate and unreliable. This is not viable solution for many applications, especially when the robot is controlled in an unknown environment. Therefore, in our approach to AT we rather propose to process only the instantaneous RTI properties locally as the robot traverses and explores the environment. The only way to obtain such RTI properties is prediction online using machine learning techniques [6], [7], [8]. We adopted this approach to RTI already in applications such as predicting correction coefficients of robot odometry [9] or estimating stride length of a legged robot while slipping [10] [11]. Since the RTI is predicted only locally, greedy optimization of inaccurately estimated RTI criterion can easily lead the robot into the

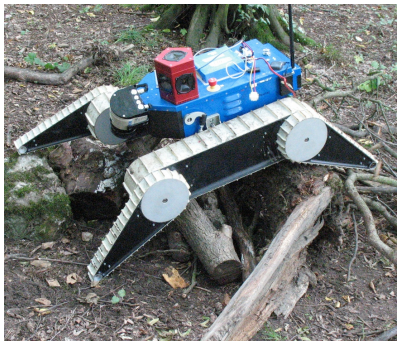


Fig. 1. **Adaptive traversability:** Robot configuration of the 4 flippers (subtracks) and their compliance is controlled autonomously in order to adapt to the terrain and traverse it in an optimal way. The robot is equipped with rotating laser scanner SICK LMS-151, Ladybug 3 omnicam, Xsens MTi-G IMU, and independent flipper stiffness control for each subtrack.

The authors were supported as follows: [1]K.Z. by the Czech Science Foundation Project 14-13876S, [2]P.Z. by the SGS13/142/OHK3/2T/13 of the CTU in Prague, [3]M.R. by the EU-FP7-ICT-609763 TRADR, and [4]V.H. by the Project TE01020197 of Technology Agency of the Czech Republic.

K.Z. and P.Z. are with the Center for Machine Perception, Dep. of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague (e-mails: {zimmerk,zuzanp1}@cmp.felk.cvut.cz). M.R. and V.H. are with the Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University in Prague (e-mails: {michal.reinstein,hlavac}@ciirc.cvut.cz).

dangerous state. Therefore, we propose to rather compute the RTI criterion directly during the training phase and learn to predict the expected sum of RTI criterion values, which can be obtained with the robot from its current state for different controlling strategies. This formulation naturally leads to the reinforcement learning task, where the RTI criterion corresponds to the reward.

Our main contributions lie in (i) defining the concept of AT for mobile robots, and (ii) proposing a reinforcement learning based AT algorithm for autonomous morphology adaptation that improves the motion control even in complex unknown environment. We (iii) demonstrate the functionality by training the AT under lab conditions on simple EUR-pallet obstacles and testing it on natural obstacles in a forest. For quantitative evaluation we (iv) define a metrics based on comparison with expert operator.

The paper is structured as follows: Section II introduces the related work, Section III describes our proposed solution, Section IV summarizes the experimental evaluation and Section V concludes the implications of our work.

II. Related Work

An ample amount of work has been devoted to the field of Robot Terrain Classification (RTC) [12], [13], [14], [15] where terrain features are mapped on discrete classes of given properties. On the conceptual level, this is relevant to extracting the terrain properties for the RTI. However, these RTC defined classes are often weakly connected with the way the robot can actually interact with the terrain or the connection is lacking. Few papers describe the estimation of RTI directly, for example, Kim et al. [7] estimate whether the terrain is traversable or not, and Ojeda et al. [8] estimate power consumption on a specific terrain type. In the literature, the RTI properties can be specified explicitly (e.g. robot consumption [8]) or implicitly (e.g. state estimation correction coefficient [9]). The problem of the AT in the way we approach it using reinforcement learning is a road less traveled in robotics, but though the target application differs, highly relevant is the work of Abbeel et al. [16], [17]. There are also alternative solutions, based for instance on kinematic model of the robot [18], [19], or by achieved learning a direct mapping between terrain features and robot actions [20], [21]. However, analytical modeling of the RTI is in general very difficult and simplifications cannot be avoided. On contrary to [16] we omit this modeling since it is not needed in our approach and instead of using Value-based algorithms, we rather focus on Q-learning technique.

III. Adaptive Traversability by Reinforcement Learning

We solve the adaptive traversability problem for a tracked robot[1] equipped with four flippers, see Fig. 1. The sensor suite of the robot consists of a rotating 2D laser scanner (SICK LMS-151) mounted in front of the robot (the rotation of the scanner provides the 3D scans), a Point Grey Ladybug 3 omni-directional camera, and a Xsens MTi-G inertial measurement unit (IMU) with GPS.

It is expected that the speed and azimuth of the robot is controlled by the operator (or provided by a path planner), and the task is to control the configuration of the four flippers and their compliance. Compliance of flippers is obtained by measuring the actual current in flipper drives and setting a threshold on the maximum allowed current. This threshold is called the *compliance threshold*.

To simplify such 8-dimensional task, we defined five discrete flipper modes specifying the angle and the compliance threshold for all four flippers. The task is to switch between these flipper modes (denoted by $c \in \mathbb{Z}$) in order to collect maximum sum of *rewards* over the obstacle being traversed. We define reward function $r(c, s) : (\mathbb{Z} \times \mathbb{R}^n) \to \mathbb{R}$, which assigns a real valued reward for achieving state s while using mode c. We experimented with several types of the reward function, which are described and evaluated in Section IV. For now, we define the reward function as a weighted sum of (i) user denoted penalty (reward) specifying that the state is (not) dangerous, (ii) high pitch angle penalty (considering robot safety from flip-over), (iii) excessive flipper mode change penalty, (iv) robot forward speed reward (for making progress in traversing), and (v) motion roughness (smoothness) penalty (reward).

A. Reinforcement learning algorithm

To tackle this problem, the reinforcement learning technique is used. We define function $Q(c, s) : (\mathbb{Z} \times \mathbb{R}^n) \to \mathbb{R}$, which estimates expected sum of discounted rewards, when the robot is in state s and flippers are set to mode c and the robot will be controlled optimally from the following state onward. Such function allows for the following recursive definition:

$$Q(c, s) = \sum_{s'} p(s'|c, s) \Big[r(c, s) + \gamma \max_{c'} Q(c', s') \Big] \quad (1)$$

where $p(s'|c, s)$ is transition probability that the robot, which is in state s with flippers set to mode c will get to the following state s'. Discount factor $\gamma \in \{0, 1\}$ is used to reduce the influence of distant future rewards. If such function is known, the optimal flipper mode c^* for the robot in the state s is chosen as follows

$$c^* = \arg\max_{c} Q(c, s) \quad (2)$$

Since we want to avoid the learning of $p(s'|c, s)$, function $Q(c, s)$ is learned using modification of the *fitted Q-iteration* algorithm summarized in Alg. 1. The proposed algorithm repeats the Q-learning procedure for several episodes. Training data collected for the first episode (line: 3) are obtained by an expert operator. To speed up the learning process, also *reasonably* negative training samples (with negative rewards) are provided. Once we are satisfied with the performance on validation data (also collected and annotated by the expert operator), we start to collect the training data with autonomously chosen flipper modes, i.e. chosen according to Eq. (2). When a batch of the training data is collected, the

[1]Developed as part of NIFTi project `http://www.nifti.eu`

$Q(c,s)$ function is trained in lines 4-7. Since we defined the $Q(c,s)$ as a collection of piecewise constant functions, the solution of the problem from line 6 is detailed in Section III-B. Section III-C describes features representing the state and the feature selection method we used.

//Initialization
1: $Q(c,s) = 0 \quad \forall_{c,s}$
2: **while** (adaptive traversability is not good) **do**
3: Drive the robot over training obstacles and assign rewards. Captured training data consists of sequences $[(s^1,c^1,r^1),(s^2,c^2,r^2),\dots]$.
// Train $Q(c,s)$
4: **repeat**
5: $y^i = r^i + \gamma\Big[\max_{c'} Q(c', s^{i+1})\Big] \quad \forall_i$
6: $Q(c,s) = \arg\min_{\overline{Q}} \sum_{i=1}^{N} \|\overline{Q}(c^i,s^i) - y^i\|_2^2$
7: **until** (convergence reached)
8: **end while**

Algorithm 1: Procedure of learning of the Q-function.

B. Piecewise constant function learning

In our approach, the $Q(c,s)$ is collection of mode specific functions $Q_c(s)$ corresponding to the number of flipper configuration modes. Since the procedure of learning is same for all functions $Q_c(s)$, we omit index c and focus on the learning of a regression function $Q(s)$ for N training samples $(s^1,y^1)\dots(s^N,y^N)$ prepared in the line 5 of Alg. 1. The upper index i is used to denote training samples, the lower index k is used to denote features. Concatenation is denoted by square brackets.

We define $Q(s) = \sum_{k=1}^{K} q_k(s_k)$ as the sum of piecewise constant functions $q_k(s_k)$ of features $s_k \in \mathbb{R}$, where K denotes the number of features. Features s_k are normalized to have zero mean and unit covariance. Feature values are divided into U equally sized bins [2]. To simplify the notation, we define a bin assigning function $\Omega(s_k) : \mathbb{R} \to \mathbb{N}$ which assigns corresponding bin u to feature value s_k.

Response of the regression function $Q(s)$ is then computed as follows:

$$Q(s) = \sum_{k=1}^{K} q_k(s_k) = \sum_{k=1}^{K} \Lambda_{k,\Omega(s_k)}, \tag{3}$$

where $\Lambda_{k,\Omega(s_k)} \in \mathbb{R}$ is the constant response of feature function q_k on feature value s_k.

Substituting Eq. (3) into the problem in Alg. 1, line: 6, we obtain the corresponding least squares problem:

$$\Lambda = \underset{\overline{\Lambda}\in\ \mathbb{R}^{\mathbb{K}\times\mathbb{U}}}{\arg\min} \sum_{i=1}^{N} \Big\| \sum_{j=1}^{k} \overline{\Lambda}_{j,\delta(s_j^i)} - y^i \Big\|^2. \tag{4}$$

To write the solution of (4) in a compact form, we further introduce a binary matrix

$$[\mathbb{A}]_{i,(ku)} = \begin{cases} 1 & \text{if } \Omega(s_k^i) = u \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where index i determines the row and indices (ku)[3] determine the column. We also introduce a vector $\lambda = [\Lambda_{1,1}\dots\Lambda_{K,U}]^\top$, which is concatenation of all unknown lambdas from all bins and all features. Finally, we form the vector $\mathbf{y} = [y^1\dots y^N]^\top$ with all desired values. The solution of problem (4) is then

$$\lambda = \underset{\overline{\lambda}}{\arg\min} \Big\| \mathbb{A}\overline{\lambda} - \mathbf{y} \Big\|^2 = \mathbb{A}^+\mathbf{y}, \tag{6}$$

where $\mathbb{A}^+$ denotes Moore-Penrose pseudo-inverse of matrix $\mathbb{A}$.

C. State representation and feature selection

We represent the mutual state of the robot and the local neighboring terrain as N-dimensional feature vector $s \in \mathbb{R}^N$. Features are selected from a feature pool, which consists of: **Terrain shape features:** Since the robot is equipped with the laser scanner, we merge individual scans into a point cloud 3D map exploiting the ICP algorithm [22]. The point cloud map in the local neighborhood of the robot is further transformed into the Digital Elevation Map (DEM), see Fig. 2, capturing the local spatio-temporal representation of the terrain. To represent the terrain shape in a compact form, Haar-like features are computed using the DEM values. In addition to that parameters of the plane fitted into the neighboring terrain are used.

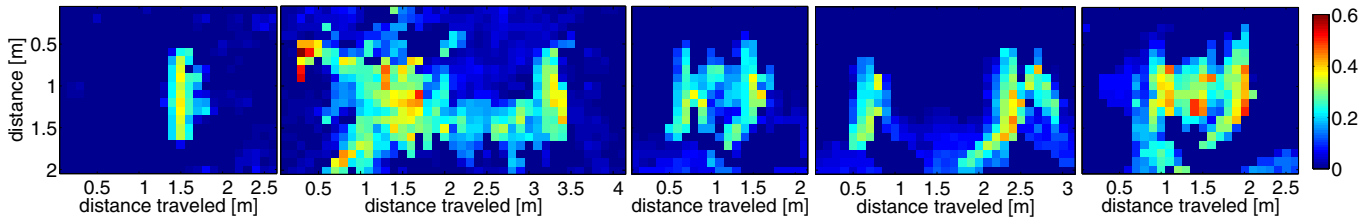
Robot state and configuration features: Robot speed (both actual and requested by the operator), pose (pitch, roll, yaw), flipper angles, compliance thresholds and actual flipper mode. To estimate the velocity of the robot, terrain adaptive odometry method [9] is used and combined with IMU data and information provided by the ICP using the Extended Kalman filter (EKF). The precise and stable pitch and roll angles are obtained using a complementary filter [23]. In addition to this information, currents in the flipper and the main track drives are used to provide the knowledge about the weight distribution and ground contacts.

We select a set of suitable features from the feature pool S by a forward stage-wise feature selection strategy [24] based on Gram-Schmidt orthogonalization process. More formally, we are given a training set $\{(\mathbf{S}^1,y^1),\dots(\mathbf{S}^N,y^N)\}$ consisting of N training samples, where $\mathbf{S}^i$ are M-dimensional vectors containing values of all features from the feature pool. Especially, we denote S_k^i as the k-th feature value of the i-th training sample.

This proposed feature selection method is summarized in Alg. 2. It successively builds the feature set from the features minimizing residuals Δy^i of all training samples $i = 1\dots N$. Initially we equal residual of the i-th training sample $\Delta y^i = y^i$. In each training stage, the algorithm estimates coefficients for all features $k = 1\dots M$ and greedily selects the feature with the lowest residual error. Such feature is added to the list of selected features and the algorithm continues while the validation error is decreasing.

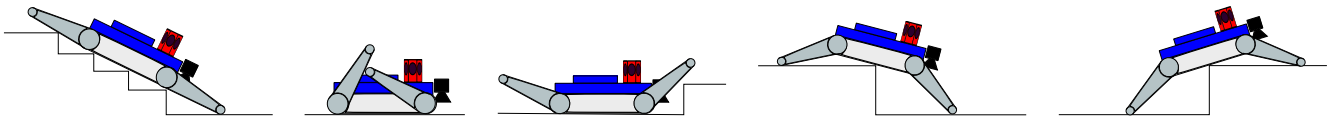
[2] except the size of border bins which are $[-\infty, \texttt{min\_value}]$ and $[\texttt{max\_value}, +\infty]$

[3] (ku) denotes a linear combination of indices k and u corresponding to the vectorization of Λ.



(1) simple obstacle (2) soft-terrain side-roll obstacle (3) long obstacle (4) double obstacle (5) complex obstacle

Fig. 2. **Obstacles:** Digital elevation maps (DEM) of testing obstacles constructed during experiments.



I-shape V-shape L-shape U-shape soft U-shape hard

Fig. 3. **Defined flipper modes:** Definition of five flipper modes corresponding to different morphological configurations of varying properties.

//Initialization

$\Delta y^k = y^k, \quad K = 1$

while (validation error is decreasing) **do**

// Select the feature k^* with the lowest residual error
// from the feature pool.

$k^* = \arg\min_{(k,\Lambda_K)} \sum_{i=1}^{N} \|Q([\mathbf{s}_1^i \dots \mathbf{s}_{K-1}^i S_k^i]) - y^i\|_2^2$

// Add the selected feature S_{k^*} into $\mathbf{s}$.

$\mathbf{s} = [\mathbf{s} \;\; S_{k^*}]$

// Update residuals

$\Delta y^i = y^i - Q(s^i) \; \forall_i$

$K = K + 1$

end while

Algorithm 2: Feature selection procedure

IV. EXPERIMENTAL EVALUATION

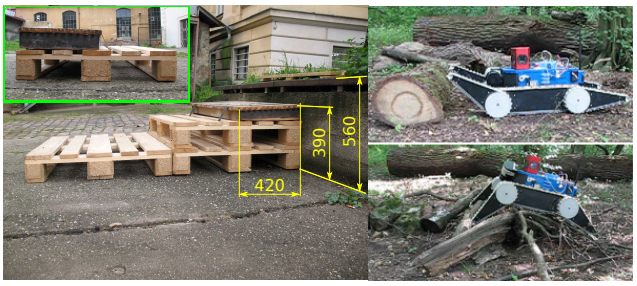
Our proposed solution to the AT was tested on five challenging obstacles created from woods and stones in an outdoor forest environment[4]; see the examples in Fig. 1, Fig. 4b and the digital elevation maps (DEM) of testing obstacles computed online by the robot Fig. 2. Each obstacle was traversed multiple times with autonomous flipper control (AC) following the Eq. (2).; obstacles $1, 2$ and 5 were also traversed with manual flipper control (MC) by the expert operator for the purpose of quantitative comparison. We emphasize that the complexity of testing obstacles was selected in order to challenge robots hardware capabilities. One of the testing obstacles even proved to be too complex to be traversed neither with the AC nor MC.

The rest of this section is organized as follows: Section IV-A describes training procedure, Section IV-B summarize the testing procedure. Section IV-C provides the comparison and evaluation.

A. Training Procedure

We define five morphological configurations—five different flipper modes (Fig. 3) (i) **I-shape** with unfolded flippers (useful for traversing holes or stairs), (ii) **V-shape** with flippers folded in order to provide the best observation

[4]For better comprehension, see the attached multimedia showing one testing drive over obstacle 2 from Fig. 2.



(a) Training objects (b) Testing objects

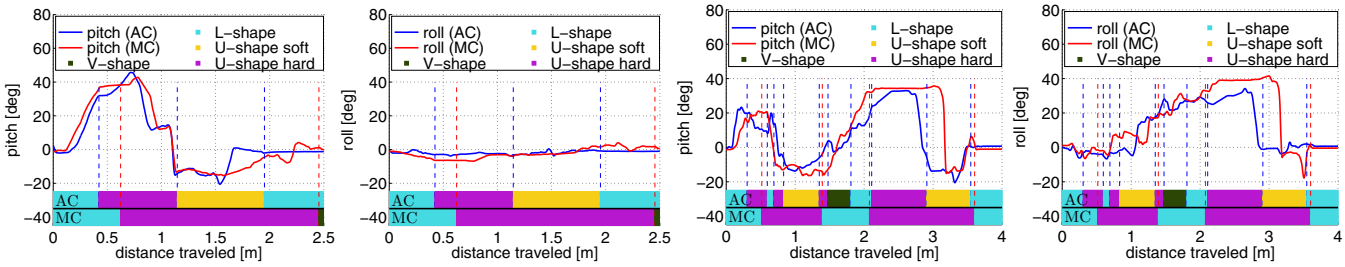
Fig. 4. **Obstacles:** (a) Three EUR pallets with one non standard pallet and concrete shoal used for training part. (b) Natural obstacles in an outdoor forest environment used for testing part.

capabilities to the robot, (iii) **L-shape** with front flippers raised (suitable for climbing up), (iv) **U-shape soft**, pushing the flippers down with low pressure—low compliance threshold (suitable for smooth climbing down), and (v) **U-shape hard**, pushing the flippers down with high pressure—high compliance threshold.

Our proposed approach to AT was trained in controlled lab conditions using only two artificial obstacles created from EUR pallets [5]. First training obstacle was just a single pallet, the second consisted of stairs created from three pallets (see Fig. 4a).

We trained the Q-function according to the Alg. 1 in three episodes, i.e. three iterations of the while-loop. In the first two episodes, the training data were collected with manual flipper control. To speed up the learning procedure *reasonably* negative (but not dangerous) training samples were provided. In the last episode, the training data were collected autonomously by the robot. Each training sample was accompanied by its reward. The best results were achieved for the reward function defined as a weighted sum of (i) manually annotated labels reflecting success of the operator's goal (either positive equal to 1 or negative equal to -1), (ii) thresholded exponential penalty for pitch angle, and (iii) roughness of motion penalty defined as $\sqrt{v_y^2 + v_z^2}$. In order to reduce oscillations between modes

[5]type EUR 1: 800mm × 1200mm × 140mm



(a) **Obst.1:** pitch + flipper modes (b) **Obst.1:** roll + flipper modes (c) **Obst.2:** pitch + flipper modes (d) **Obst.2:** roll + flipper modes

Fig. 5. **Pitch, roll, and flipper modes along the obstacle traversal:** All graphs show used flipper modes and pitch+roll angles reached by the robot during the obstacle traversal. Graphs (a) and (b) correspond to the obstacle 1, graphs (c) and (d) to the obstacle 2. Autonomous control (AC) is depicted by the blue color and manual control (MC) by the red color. Autonomously selected modes are shown in the first color bar and manual in the second.

with similar Q-values we (i) introduce additional penalty for changing the mode and (ii) evaluate the Q-values over 1 second long time interval. In each episode, the training of the Q-functions was iterated 30 times. The number of iterations was experimentally determined as sufficient for the convergence of the Q-values with $\gamma = 0.8$. To achieve a well conditioned training dataset, the training samples were artificially perturbed several times.

B. Testing Procedure

We tested the AC method on five challenging natural obstacles in a forest. Both the AC and MC allowed to traverse obstacles $1-4$ (see Fig. 2). Obstacle 5 consisted of two woods located in parallel with the mutual distance equaled approximately to the length of robot with folded flippers. Such obstacle turned out to be not traversable neither with the autonomous nor with the manual flipper control. For obstacles 1 and 2 quantitative comparison of the autonomous and manual flipper control is provided in Tab. I, II. To compare AC and MC traversability quality, five different metrics were proposed and evaluated: (i) average pitch angle (sum of absolute values of the pitch angle divided by the number of samples), (ii) average roll angle, (iii) traversal time (start and end points are defined spatially), (iv) average current in flipper engines (corresponds to flipper torque), (v) overall power consumption during the whole experiment, and (vi) number of mode changes.

C. Results

Tab. I shows that the average pitch, roll and the number of changes of the AC and MC on the obstacle 1 are comparable. However, the power consumption and the average current are both lower for the AC. This is achieved by more efficient mode selection—such as using the U-shape soft mode for going down from the obstacle—, see the flipper modes, pitch and roll angle plots of AC and MC in Fig. 5a,b.

Tab. II clearly demonstrates that the AC outperformed MC in most of evaluated metrics. The most significant difference can be seen in the actual time taken. While MC often required to stop the robot and wait for the end of the mode change procedure, the AC was continuous and proceeded as the robot was driven forward. Therefore, the traversal time of the AC is almost twice as short. In addition to that, since

TABLE I

COMPARISON OF AUTONOMOUS AND MANUAL ROBOT CONTROL ON THE OBSTACLE 1 (SIMPLE OBSTACLE).

	Pitch [°]	Roll [°]	Time [s]	Current [A]	Changes [–]	Consumption [Ah]
AC	11.2	1.8	35.7	3.4	3	0.07
MC	11.3	2.8	36.8	5.4	2	0.10

TABLE II

COMPARISON OF AUTONOMOUS AND MANUAL ROBOT CONTROL ON THE OBSTACLE 2 (CONTAINS SOFT-TERRAIN AND SIDE-ROLL).

	Pitch [°]	Roll [°]	Time [s]	Current [A]	Changes [–]	Consumption [Ah]
AC	10.2	10.6	75.3	3.9	10	0.17
MC	17.1	17.1	132.1	4.6	4	0.33

our definition of the reward function also contains penalty for being in extreme angles (accounting for robot safety), the AC achieved smaller pitch, roll, as well as flipper current—the ground/obstacle contacts were less frequent and less intense. The power consumption of AC compared to MC was hence much lower, enabling the robot to last longer while carrying out the mission.

On the other hand, the number of mode changes of AC is higher. To explain it, we need to analyze the actual motion trajectory and corresponding obstacle in detail (see also Fig. 5 c,d): The first part of the obstacle is created from many flexible sticks behaving as a soft terrain–and thus deforming under the robot weight. Since we used only the EUR pallets, this RTI property was not represented in the training set at all. The robot correctly starts in the L-shape mode to climb on the obstacle.Then it switches to the U-shape hard mode to lift its body on the obstacle. However, the soft terrain collapses under the body weight and L-shape must be used again to traverse the remaining hard part of the obstacle. Similar scenario repeats, when traversing the middle part of the obstacles and can be in general expected on similar terrain.

V. CONCLUSION

In this paper we have concentrated our efforts on defining the task of *Adaptive Traversability* (AT) as means of autonomous motion control adapting the robot morphology (configuration of flippers and their compliance) in order to

traverse unknown complex terrain with obstacles. Similar approaches have been deployed using trajectory planning for scenarios, where a map of the environment was available, hence providing an easy way to compute the Robot-Terrain Interaction. However, we propose a solution based on reinforcement learning that exploits only the information from local RTI and does it online, hence no map is needed—solution ideal for exploring unknown environments with obstacles. Having experience from real deployment of robots in Search & Rescue scenarios, we are aware of the crucial impact of cognitive load on the operator. Therefore, we define a metrics allowing us to compare our solution to an expert operator driving the robot manually. Beside outperforming the manual control in a number of criteria (time taken for traversal, power consumption, smoothness and safety of the robot), the main accomplishment lies in the minimal cognitive load required for the robot control while using our AT solution. Moreover, our approach can easily be used together with any trajectory planning algorithm in general in a complementary way. We would like to also point out, that for the actual training, only simple obstacles made of EUR pallets were used, but the actually testing was successfully performed using challenging natural obstacles in a forest environment.

To conclude, on the testing dataset, the proposed AT algorithm exhibited very stable behavior such as: (i) **Repeatability:** consistent flipper control over multiple traversals of the same obstacle, (ii) **Robustness:** training with similar parameters and similar training data yields similar behavior (iii) **Generalization:** reasonable and explainable flipper control on the challenging testing data—no deformable obstacles in the training dataset, yet surprisingly good performance on such deformable terrain during testing.

As a future work, we clearly see the opportunity in expanding all of our assumptions made: we can define more different modes, exploit more compliance thresholds, we can allow the robot to train by itself on much larger scale of obstacles, as well as to push the challenge of the testing environments. We also intend to integrate the AT algorithm with our SLAM solution and trajectory planner to expand the range of field applications and possibilities.

REFERENCES

[1] G.-J. Kruijff, M. Janicek, S. Keshavdas, B. Larochelle, H. Zender, N. Smets, T. Mioch, M. Neerincx, J. van Diggelen, F. Colas, M. Liu, F. Pomerleau, R. Siegwart, V. Hlavac, T. Svoboda, T. Petricek, M. Reinstein, K. Zimmermann, F. Pirri, M. Gianni, P. Papadakis, A. Sinha, P. Balmer, N. Tomatis, R. Worst, T. Linder, H. Surmann, V. Tretyakov, S. Corrao, and S. Pratzler-Wanczura, "Experience in system design for human-robot teaming in urban search & rescue," in *Field and Service Robotics (FSR), 2012 8th International Conference on*, July 2012.

[2] F. Colas, S. Mahesh, F. Pomerleau, M. Liu, and R. Siegwart, "3d path planning and execution for search and rescue ground robots," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 722–727, Nov 2013.

[3] M. Brunner, B. Bruggemann, and D. Schulz, "Towards autonomously traversing complex obstacles with mobile robots with adjustable chassis," in *Carpathian Control Conference (ICCC), 2012 13th International*, pp. 63 –68, may 2012.

[4] S. Martin, L. Murphy, and P. Corke, "Building large scale traversability maps using vehicle experience," in *Internation Symposium on Experimental Robotics*, 2012.

[5] B. Cafaro, M. Gianni, F. Pirri, M. Ruiz, and A. Sinha, "Terrain traversability in rescue environments," in *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE Int. Symposium on*, pp. 1–8, Oct 2013.

[6] K. Ho, T. Peynot, and S. S. Sukkarich, "Traversability estimation for a planetary rover via experimental kernel learning in a gaussian process framework.," in *Internation Conference on Robotics and Automation (ICRA)*, 2013.

[7] D. Kim, J. Sun, S. Min, O. James, M. Rehg, and A. F. Bobick, "Traversability classification using unsupervised on-line visual learning for outdoor robot navigation," in *In Proc. of International Conference on Robotics and Automation (ICRA)*, pp. 518–525, 2006.

[8] L. Ojeda, J. Borenstein, G. Witus, and R. Karlsen, "Terrain characterization and classification with a mobile robot," *Journal of Field Robotics*, vol. 23, pp. 103–122, 2006.

[9] M. Reinstein, V. Kubelka, and K. Zimmermann, "Terrain adaptive odometry for mobile skid-steer robots," in *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, 2013. *accepted—to appear*.

[10] M. Reinstein and M. Hoffmann, "Dead reckoning in a dynamic quadruped robot: Inertial navigation system aided by a legged odometer," in *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, pp. 617–624, 2011.

[11] M. Reinstein and M. Hoffmann, "Dead reckoning in a dynamic quadruped robot based on multimodal proprioceptive sensory information," *IEEE Trans. on Robotics*, vol. 29, pp. 563–571, April 2013.

[12] C. Weiss, H. Frohlich, and A. Zell, "Vibration-based terrain classification using support vector machines," in *Proc. IEEE/RSJ Int Intelligent Robots and Systems Conf*, pp. 4429–4434, 2006.

[13] K. Kim, K. Ko, W. Kim, S. Yu, and C. Han, "Performance comparison between neural network and SVM for terrain classification of legged robot," in *Proc. SICE Annual Conf. 2010*, pp. 1343–1348, 2010.

[14] E. M. DuPont, C. A. Moore, and R. G. Roberts, "Terrain classification for mobile robots traveling at various speeds: An eigenspace manifold approach," in *Proc. IEEE Int. Conf. Robotics and Automation ICRA 2008*, pp. 3284–3289, 2008.

[15] P. Komma, C. Weiss, and A. Zell, "Adaptive bayesian filtering for vibration-based terrain classification," in *Proc. IEEE Int. Conf. Robotics and Automation ICRA '09*, pp. 3307–3313, 2009.

[16] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," in *In Advances in Neural Information Processing Systems 19*, p. 2007, MIT Press, 2007.

[17] P. Abbeel and A. Y. Ng, "Exploration and apprenticeship learning in reinforcement learning," in *Proc. of the 22nd int. conf. on Machine learning*, ICML '05, (New York, NY, USA), pp. 1–8, ACM, 2005.

[18] Y. Okada, K. Nagatani, K. Yoshida, S. Tadokoro, T. Yoshida, and E. Koyanagi, "Shared autonomy system for tracked vehicles on rough terrain based on continuous three-dimensional terrain scanning," *J. Field Robot.*, vol. 28, no. 6, pp. 875–893, 2011.

[19] K. Ohno, S. Morimura, S. Tadokoro, E. Koyanagi, and T. Yoshida, "Semi-autonomous control system of rescue crawler robot having flippers for getting over unknown-steps," in *Intelligent Robots and Systems (IROS), 2007. IEEE/RSJ Int. Conf. on*, pp. 3012–3018, 2007.

[20] R. Sheh, B. Hengst, and C. Sammut, "Behavioural cloning for driving robots over rough terrain," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 732–737, 2011.

[21] E. Uğur and E. Şahin, "Traversability: A case study for learning and perceiving affordances in robots," *Adaptive Behavior*, vol. 18, no. 3-4, pp. 258–284, 2010.

[22] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP Variants on Real-World Data Sets," *Autonomous Robots*, vol. 34, pp. 133–148, Feb. 2013.

[23] V. Kubelka and M. Reinstein, "Complementary filtering approach to orientation estimation using inertial sensors only," in *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, pp. 599–605, 2012.

[24] F. W. Roush, "Applied linear regression : Snaford weisberg, new york: Wiley, 1980, pp. 283," *Mathematical Social Sciences*, vol. 3, July 1982.