UKAEA-RACE-CP(21)03

Marco Visca, Arthur Bouton, Roger Powell, Yang Gao, Saber Fallah

Conv1D Energy-Aware Path Planner for Mobile Robots in Unstructured Environments

Conv1D Energy-Aware Path Planner for Mobile Robots in Unstructured Environments

Marco Visca, Arthur Bouton, Roger Powell, Yang Gao, Saber Fallah

This paper has been submitted to
2021 IEEE International Conference on Robotics and Automation (ICRA),Xi’an China, 30 May - 5 June 2021

Conv1D Energy-Aware Path Planner for Mobile Robots in Unstructured Environments

Marco Visca, Arthur Bouton, Roger Powell, Yang Gao and Saber Fallah

Abstract— **Driving energy consumption plays a major role in mobile robots navigating challenging environments, especially if they are left to operate unattended under limited on-board power. This paper reports on first results of an energy-aware path planner, which can provide estimates of the driving energy consumption and energy recovery of a robot traversing complex uneven terrains. Energy is estimated over trajectories making use of a self-supervised learning approach, in which the robot autonomously learns how to correlate perceived terrain point clouds to energy consumption and recovery. A novel feature of our method is the use of 1D convolutional neural network to analyse the terrain sequentially in the same temporal order as it would be experienced by the robot when moving. We assess the performance of our approach in simulation over several digital terrain models collected from real natural scenarios, and we show evidence of its benefit to reduce driving energy consumption.**

I. Introduction

Autonomous path planning of mobile robots in unstructured environments is a crucial task in many sectors such as: rescue robots for disaster area, agriculture, nuclear plants, and space exploration. Among different factors, energy consumption can play a major role on the success and efficiency of robotic missions. For example, planetary exploration rovers are required to drive several kilometres to reach potentially interesting scientific goals, but they often have limited on-board power resources [1]. Monitoring robots inside industrial plants or rescue robots in disaster areas are occasionally disconnected from umbilical cables and can be difficult to reach by human operators [2]. Estimating in advance driving energy can be vital to their operational safety and to allow for coverage of longer distances.

In this work, we are interested to analyse the effect of unstructured geometries on driving energy consumption. While previous works have developed heuristic energy models that consider the terrain inclination as the main energy-relevant geometric factor [3] [4] [5], their prediction performance can decrease in highly uneven surfaces. For example, scattered rocks, steps, bumps and rough terrain can pose a greater challenge to the robot locomotion and control system leading to increasing driving energy requirements. However, the complex interplay coming into action between wheel and terrain over unstructured geometries, and their effect on driving energy, can be challenging to model based on first principles. In this context, deep neural networks (DNNs) can be a well-suited asset, as they do not require explicit domain knowledge to be implemented into the prediction algorithm, and for their ability of extracting relevant features from high-dimensional data [6].

In this paper,

- We propose a DNN architecture to estimate from point clouds the driving energy consumption and recovery of a mobile robot traversing unstructured terrains (Section III-A and Section III-B). The main novelty of this paper is the use of a Convolutional 1D (Conv1D) neural network to analyse the point cloud data. In our approach, we rearrange the point cloud in the same temporal order as it would be experienced by the robot when traversing the terrain, and we train the network to capture the sequential context of the wheel-terrain interaction from which energy depends.
- We demonstrate the robustness of this method over data from several real unstructured scenarios (Section IV-A) and we compare it with a heuristic model which only considers the terrain inclination as geometric energy-relevant feature [3]. Particularly, we show that our model is robust to increasing levels of terrain roughness, while the latter substantially degrades its performance as the terrains deviate from planar (Section V-B).
- We integrate the Conv1D neural network energy estimator into a lattice space A* path planner (Section III-C). The benefit of our approach is that both energy estimation and path planning are achieved directly over feasible trajectories and unordered point clouds, without any need of generating artificial 2D cost maps.
- We test our energy-aware path planner over several unstructured scenarios and we show evidence of the benefit of our approach to reduce driving energy consumption (Section V-C).

To the best of the authors knowledge, this work is the first to propose an integrated energy-aware prediction and planning framework for mobile robots in unstructured environments using deep neural networks.

II. Related Work

Existing algorithms in the field of path planning in unstructured environments operate on a 2D-grid superimposed over the operational map [7] [8]. Each cell of the grid is assessed with specific traversibility tests and, based on their results, assigned with a cost value. While this could

Marco Visca and Saber Fallah are with the Connected and Autonomous Vehicles Lab, University of Surrey, Guildford, GU2 7XH, UK. Email: {m.visca, s.fallah}@surrey.ac.uk

Arthur Bouton and Yang Gao is with Surrey Space Centre, University of Surrey, Guildford, GU2 7XH, UK. Email: {a.bouton, yang.gao}@surrey.ac.uk

Roger Powell is in the Cybernetics Group, Remote Applications in Challenging Environments, UK Atomic Energy Authority, Culham Science Centre, OX14 3DB.

be an effective and simple method to ensure reliable and safe navigation, expressing energy cost in this form can be challenging, as it assumes that each cell has a well defined intrinsic cost value. Besides, the cost functions used by grid methods are usually heuristic weighted combinations of terrain geometric properties such as slope, and roughness [9] [10] [11]. However, they do not provide an explicit estimation of energy consumption.

Other works have proposed the use of accurate onboard traverse simulators, which take as input geometric and other terrain information, and explicitly run a forward navigation simulation so as to assess the energy consumption of specific trajectories [11] [12]. However, in spite of their accuracy, their computational workload is often excessive for real-time navigation, making them cumbersome to integrate into a path planning optimization framework.

Several machine learning techniques have been proposed to improve the path planning autonomy of mobile robots. This includes terrain classification [13] [14] [15], obstacle detection [16], slip prediction [17] [18], and terramechanical parameters estimations [19] [20] [21] among the others. Specifically, artificial neural networks have gained an increasing popularity in this context for their ability of extracting features from high-dimensional inputs, and their efficient parallel computing [6]. Similarly to us, Higa et al. [22] have proposed a deep learning method to estimate driving energy of a mobile robot on uneven terrains. However, they do not show how their method can be integrated into a path planner. Moreover, their architecture consists of a Convolutional 2D (Conv2D) neural network to estimate driving energy consumption over robot-sized patches of terrain. On one hand, their method has the advantage to consider not only terrain geometry, but also its visual aspect. However, correlating energy consumption over robot-sized patches can be considered as a static approximation, which does not necessarily capture the actual sequential context of a robot traversing a terrain [23].

A recent work has developed an energy-aware path planner architecture which explicitly estimates energy over planned trajectories [3]. However, they make use of a heuristic energy model that only considers the terrain elevation as geometric energy-relevant feature [4] [5]. We argue that, while this approach can be satisfactory for relatively planar surfaces, it cannot capture the influence of more complex unstructured geometries. A similar implementation of their energy model, valid for our robotic platform, is used as main reference for qualitative and quantitative comparisons. We refer to it as ElevModel in the rest of the paper. For a detailed description it can be referred to [3]. Furthermore, we observe that in their work two non-geometric factors are additionally considered for energy estimation: surface friction, and internal friction due to the track speed. However, as this work focuses on the effect of terrain geometry, we maintain surface friction and track speed constant during our experiments.

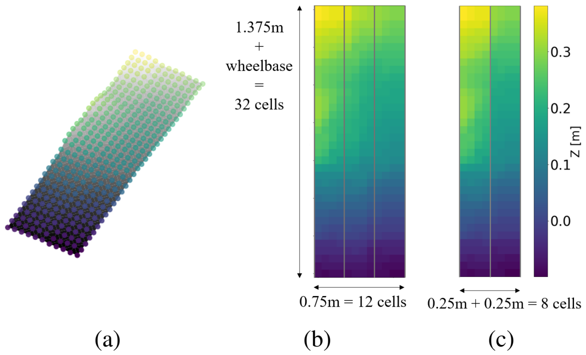


Fig. 1: Point Clouds data processing pipeline. (a) Point clouds relative to trajectory is extracted from memory. (b) Point cloud is downsampled into a 2D voxel-grid. (c) The central region not in contact with the wheels is removed.

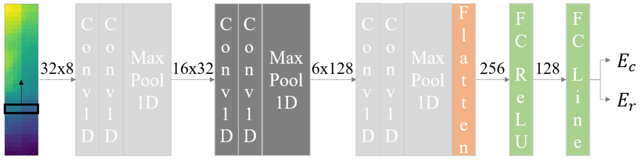


Fig. 2: Conv1D neural network architecture.

III. Constructing the Energy-Aware Path Planner

We propose to estimate driving energy end-to-end from point clouds by neural network training. Particularly, the neural network must be able to analyse the sequential context of the traversed terrain directly over feasible trajectories. In this way, driving energy estimation can be achieved by considering the actual traverse dynamics and the robot mobility constraints, thereby increasing the prediction accuracy. Moreover, the proposed approach allows to plan directly over unordered point clouds, without any need of cost map generation. This can be advantageous as point clouds can be easily obtained from any kind of range sensor, they are easy to maintain, and they can be split and merged with minimal computational effort [24]. The remainder of this section exhaustively describes the proposed methodology.

A. Point Clouds Collection and Preprocessing

This paragraph describes the point clouds collection and preprocessing procedure. A diagram of the procedure is illustrated in Fig. 1. Points are retrieved from memory along the trajectory with a width equal to the wheel track (Fig. 1a). A safety check is performed prior to continue with the preprocessing pipeline. The point cloud is analysed with standard geometric features extraction techniques to measure pitch, roll, and residual features [7]. If at least one of these values is above specific thresholds the trajectory is labelled as untraversable and is not considered for the energy planning optimisation. Conversely, if the point cloud passes the safety check, the preprocessing procedure can be continued. The point cloud is downsampled with a 2D-voxel regular grid (Fig. 1b). This choice is motivated by the need of our neural network to have a fixed number of inputs to be used. Downsampling does not introduce notable artefacts as long as the point cloud has a similar or higher density than the voxel discretization [25]. For our application, and in line with current on-board perception technologies [26], we assume a uniform point cloud density of $256\,\mathrm{pts/m^2}$ in a region up to $4\,\mathrm{m}$ distant from the robot. Next, we set the voxel discretization to a value of $6.25\,\mathrm{cm} \times 6.25\,\mathrm{cm}$. Finally, we can observe that all elevation points in the central region of the voxel-grid have lower elevations than the rover belly, as they have passed the safety check. As a result, they do not interact with the robot and thereby do not impact the driving energy. Hence, only the two lateral bands, of width $25\,\mathrm{cm}$ each, are retained (Fig. 1c). The result of the point cloud preprocessing analysis is a 32 × 8 matrix of elevation values, which represents one input to our proposed neural network.

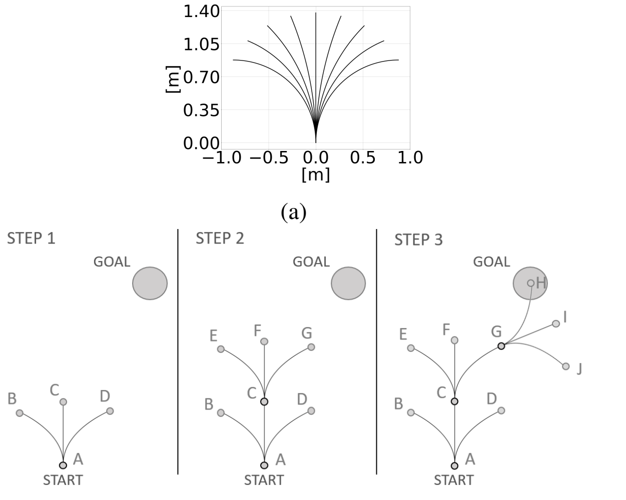
B. Neural Network Architecture

We propose to learn driving energy from point clouds by means of 1D convolutional neural networks. Conv1D is a well known neural network architecture which has demonstrated remarkable performance in analysis of series of sensor data, signal processing and natural language processing [27]. Additionally, they have the advantage over alternative sequential methods (e.g. recurrent neural networks) to require lower training effort and to better exploit parallel computing [28]. Similarly, we propose to formulate our problem in a sequential fashion. In the processed point cloud in Fig. 1c, each row can be seen as the elevation points experienced by the rover along its width at a specific time, while each column represents the time evolution of the terrain feature during the traverse. Hence, Conv1D can be used to analyse the terrain sequence by means of a fixed-dimension 1D context window over the temporal (i.e. vertical) axis.

Fig. 2 shows the proposed Conv1D architecture. It consists of three stacks of two 1D convolutional layers followed by 1D max pooling. In this way, the bottom layers have the ability to learn a narrow temporal context, while wider temporal relationships can be learned at deeper layers. The first stack has kernel dimension $K = 2$ and zero padding, while the second and third stacks have $K = 3$ and no padding. The output of Conv1D is flattened and processed by one Fully Connected (FC) layer with 128 units and ReLU activation, followed by a FC layer with 2 units and linear activation. The outputs are the energy consumption E_c and the energy recovery E_r.

C. Planning Algorithm

Motion planning is performed in a state lattice space, a well known approach to the problem of differentially constrained mobile robot planning in unstructured environment [29]. State lattice is a search graph where vertices representing kinematic states of the robot are connected by edges representing trajectories that satisfy the kinematic constraints of the robot. In our applications, we define a set of 9 elementary trajectories $1.375\,\mathrm{m}$ long, and with curvature uniformly spaced in $[-1.14\ ,\ 1.14]\ \mathrm{m^{-1}}$, according to the mobility capability of our robot (see Fig. 3a).



(a)

(b)

Fig. 3: (a) The 9 elementary trajectories of the lattice space and (b) diagram of A* optimisation in lattice state space with only three actions for simplicity.

The A* graph search optimisation is then used in the lattice space. Fig. 3b shows a diagram of the A* planning process for a lattice space with three actions. The start position is labelled as node A in the diagram. Node A is added to a list called OPEN, which contains all nodes that have been found but not yet expanded. The node is then expanded for the set of all elementary trajectories of the lattice space. When node A is expanded, three new nodes are created and are labelled B, C, and D in the diagram. Node A is then removed from the OPEN list because it has been expanded, while nodes B, C, and D are added to the OPEN list, and are labelled as children of node A. The next step is to choose a new node from the OPEN list to expand. This is done by computing a priority for each of the nodes in the OPEN list. In A* the priority is expressed in this form:

$$P(n) = C(n_p, n) + H(n, g) \tag{1}$$

The priority P of node n is given by the real cost $C(n_p, n)$ of moving from the parent node n_p to n, and the heuristic estimation $H(n, g)$ of the remaining cost from n to the goal g. The new node is then chosen as the one having the lowest P. The search continues until a goal node is retrieved from the OPEN list. At this point, by traversing backward the stored parent-child relationships, the path and the associated cost of A-C-G-H is found from the start to the goal. For more details on A* refer to [30].

In our application, the $C(n_p, n)$ and $H(n, g)$ must have a meaning of driving energy. Particularly, the term $C(n_p, n)$ must represent the true value of energy when moving from state n_p to n. In ElevModel this is achieved by the fitting of suitable robot-dependent heuristic functions to the trajectory inclination. However, we argue that the accuracy of this

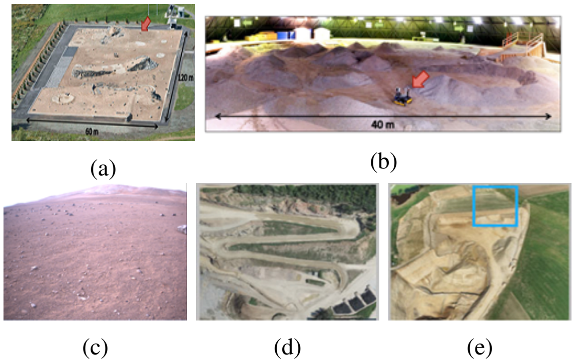


Fig. 4: Training dataset is (a) Canadian Space Agency (CSA) Martian Emulation Terrain (MET). Test datasets are (b) UTIAS indoor rover test facility, (c) Atacama desert SEEKER test trial, (d) mining quarry, and (e) gravel pit.

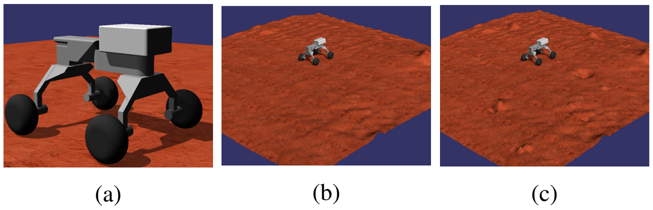


Fig. 5: (a) The robotic model in the ODE simulator, and example of digital terrain model (b) before and (c) after obstacle augmentation.

approach can not be satisfactory for highly unstructured surfaces. We thereby propose a novel method to provide an estimate to this problem. For each new trajectory in the graph, the relative point cloud is retrieved from memory and preprocessed according to III-A and fed to our Conv1D neural network described in III-B. Its transition cost is then defined as: $C(n_p, n) = E_c - E_r$. On the contrary, the heuristic $H(n, g)$ does not require to provide the true energy value, but must be sufficiently informative to address the search algorithm in promising directions. We use for our approach the same $H(n, g)$ of the ElevModel implementation, that is we assume that the goal can be reached by moving on a straight line, and we use the ElevModel functions as heuristic of remaining energy.

IV. Experiment Assets

A. Simulator and Environments

We develop and test our algorithm on a C++ simulator based on Open Dynamics Engine (ODE) [31] [32]. Different simulation environments are integrated in ODE from data collected from real uneven terrains (Fig. 4). Three of them are from Martian-like scenarios and collected using laser scans on-board wheel robotic platforms (Figs 4a, 4b and 4c). The other two are from a mining quarry and a gravel pit scenarios and collected with a fixed-wing flying robot using structure-from-motion techniques (Figs 4d and 4e). The datasets are all publicly available online at [33] and [34] with the exception of the SEEKER dataset [35].

B. Robot Model

We use the four-wheels robot model described in [36]. The model has a wheel base of $58\,\mathrm{cm}$, a wheel track of $61\,\mathrm{cm}$ and weighs $25\,\mathrm{kg}$. Its wheel diameter is $21\,\mathrm{cm}$, its overall height is $55\,\mathrm{cm}$ and it can steer with an angle up to $40°$. Fig. 5 shows the robot in the ODE simulator.

C. Ground Truth Energy Collection

We assume that the robot is powered by five DC-motors. One for each of the four wheels, and one for the central hinge which controls the steering angle. Next, we compute the power of each motor independently using the steady state equivalent circuit of the DC-motor armature [37]. The power of a DC motor is given by:

$$P = \frac{\omega\tau}{\eta} + R_a I_a^2 \tag{2}$$

The first term is the mechanical power, where ω is the motor angular speed, τ is the torque to the motor, and η is the motor efficiency due to mechanical losses. Mechanical power can be either positive or negative according to the sign of ω and τ. Particularly, if they have the same sign, the power is consumed and η has a value lower than 1 (i.e. more power is consumed than in absence of mechanical losses). If ω and τ have opposite sign, the power is recovered and η has a value higher than 1 (i.e. less power is recovered than in absence of mechanical losses). The second term is the electrical power loss due to the motor resistance dissipated as heat, where R_a is the armature resistance and I_a the current flowing in the motor. The current can also be expressed as follows:

$$I_a = \frac{\tau}{K_T g \eta} \tag{3}$$

Where, K_T is the motor torque constant, and g is the gear ratio. It follows that the power can be expressed as function of ω and τ as:

$$P = \frac{\omega\tau}{\eta} + \left(\frac{\tau}{K_T g \eta}\right)^2 R_a \tag{4}$$

Simulated sensors on-board the robot measure ω and τ for each motor with a constant sampling rate Δt, and the power is computed with the formula in 4. Finally, assuming constant power between each Δt, the energy consumption E_c and the energy recovery E_r over the time interval $[t_0, t_f]$ can be computed as:

$$\begin{cases} E_c = \sum_{t_0}^{t_f} P_t \Delta t & P_t >= 0 \\ E_r = \sum_{t_0}^{t_f} P_t \Delta t & P_t < 0 \end{cases} \tag{5}$$

V. Results

A. Training Dataset

We train the network described in Section III-B by collecting data from the Canadian Space Agency (CSA) Martian Emulation Terrain (MET) (Fig. 4a), while we use the other environments for testing. We also use a simple obstacle

TABLE I: Energy Estimation Test Results.

Dataset		Model	Consumption			Recovery			Total		
Name	Samples		MAE	MSE	R2	MAE	MSE	R2	MAE	MSE	R2
Ramps	1881	Conv1D	5.353	68.445	0.971	0.947	1.746	0.982	5.354	68.564	0.976
		ElevModel	2.112	10.231	**0.996**	0.434	0.360	**0.996**	2.078	9.812	**0.996**
MET (val)	21578	Conv1D	4.804	65.526	**0.939**	0.568	0.654	**0.983**	4.549	60.792	**0.956**
		ElevModel	14.182	385.477	0.640	1.169	2.743	0.930	13.343	342.075	0.750
SEEKER	33673	Conv1D	3.024	16.938	**0.973**	0.509	0.456	**0.982**	2.755	14.234	**0.982**
		ElevModel	10.361	153.691	0.751	1.324	3.044	0.880	9.185	119.125	0.847
UTIAS	11668	Conv1D	6.656	98.209	**0.906**	1.010	1.910	**0.903**	6.172	89.859	**0.919**
		ElevModel	21.969	717.477	0.315	2.843	11.393	0.421	19.423	588.190	0.469
Gravelpit	790	Conv1D	4.023	43.906	**0.922**	0.731	1.082	**0.925**	3.565	36.843	**0.931**
		ElevModel	12.450	321.789	0.427	1.604	4.751	0.670	10.883	256.604	0.517
Quarry	4699	Conv1D	7.497	144.139	**0.851**	1.120	2.517	**0.923**	6.968	133.121	**0.865**
		ElevModel	21.585	817.665	0.156	2.686	11.772	0.639	19.181	680.424	0.308

TABLE II: Path Planner Test Results.

		Planning Performance			Prediction Performance			Total Driving Energy	
Targets	Model	Nodes	Avg. Node Time [s]	Tot Time [s]	MSE	MAE	R2	Predicted [J]	Real [J]
259	Conv1D	5619	0.212	1265	271.938	7.882	0.937	16.268k	17.108k
	ElevModel	2769	0.208	590	2514.917	33.724	0.269	9.368k	18.103k
	DynSim	3692	5.586	20699	0	0	1	15.623k	15.623k
1 (Fig. 6)	Conv1D	16	0.183	3.012	-	-	-	29.23	30.44
	ElevModel	4	0.182	0.808	-	-	-	-0.99	39.33
	DynSim	12	4.324	51.750	-	-	-	27.72	27.72

augmentation algorithm to make the training environment more challenging. Figs 5b and 5c show an example of terrain before and after obstacle augmentation. The robot traverses the *MET* environment at constant speed of $10\,\mathrm{cm/s}$ and collects data from $107\,606$ trajectories, equivalent to approximately $150\,\mathrm{km}$ of traverse. The dataset is then randomly divided between training ($80\,\%$) and validation ($20\,\%$) datasets. Finally, the network is trained by means of stochastic gradient descent for 100 epochs with mean squared error loss, 64 batch size, 10^{-4} learning rate, and RMSprop optimizer [38].

B. Validation and Test Results

The performance of Conv1D and ElevModel are summarised in Table I. The first dataset, called *Ramps*, is a synthetic environment composed of perfectly planar ramps having different inclinations. The robot traverses these ramps with 1881 trajectories having different combinations of pitch, roll and curvature. Both ElevModel and Conv1D can accurately predict the driving energy for the planar environment. Specifically, ElevModel can predict both driving energy consumption and recovery with a r2 score of $99.6\,\%$, while Conv1D r2 score is of respectively $97.1\,\%$ for energy consumption and $98.2\,\%$ for energy recovery. The lower performance of Conv1D can be explained by the fact that it has been trained exclusively with data from unstructured environments. Adding some of the planar samples to the training dataset would likely increase the Conv1D performance on planar ramps. The results in the *Ramps* dataset show that ElevModel can be completely satisfactory to predict driving energy in planar scenarios. However, we observe that its performance degrades in unstructured surfaces. Particularly, the prediction of the energy consumption of the ElevModel seems more largely affected than energy recovery, with r2 score values respectively ranging from $15.6\,\%$ of *Quarry* to $75.1\,\%$ of *SEEKER*, and from $42.1\,\%$ of *UTIAS* to $93.0\,\%$ of *MET*. A likely reason for this is that energy recovery mostly depends on the downhill inclination of the terrain, making it less challenging, but still far from accurate, to estimate with ElevModel. In contrast, Conv1D learns implicitly, by capturing the sequential context of the traversed terrain, the most relevant features to be correlated with driving energy. In this way, Conv1D retains considerably better generalisation to the unforeseen unstructured geometries, with r2 score ranging from $85.1\,\%$ of *Quarry* to $97.3\,\%$ of *SEEKER* for energy consumption, and from $90.3\,\%$ of *UTIAS* to $98.3\,\%$ of *MET* for energy recovery.

C. Path Planner Results

Conv1D and ElevModel energy estimators are integrated into the A* lattice path planner. Their performance are assessed by conducting statistical analysis over 259 random start-goal positions within the different test environments. Table II summarises our findings. We can observe that the number of nodes expanded by Conv1D is double that of ElevModel. Specifically, over the 259 search problems Conv1D and ElevModel expand respectively 5619 and 2769 nodes. A plausible explanation is that Conv1D has a larger disparity between the energy estimated by the heuristic $H(n, g)$ and the actual cost $C(n_p, n)$. An intrinsic property of the A* optimization is that the more optimistic the heuristic is compared to the actual cost, the more A* has to expand the search. Therefore, Conv1D has to analyse a higher number of possible trajectories prior to find the optimal

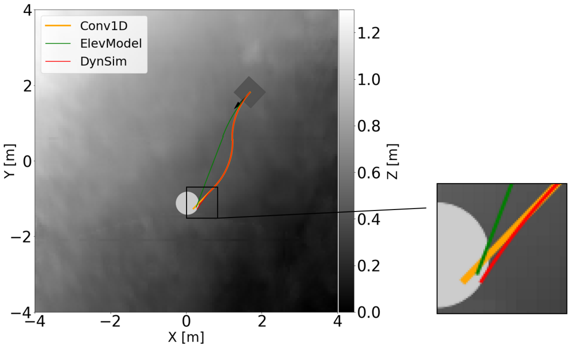


Fig. 6: Three different solutions to the A* optimisation with different energy estimation methods.

solution. While this results in longer planning time, Conv1D is nevertheless able to provide solutions which require lower driving energy consumption, and with considerably higher prediction accuracy. Specifically, the optimal predicted and real driving energy according to Conv1D is of respectively $16.268\,\mathrm{kJ}$ and $17.108\,\mathrm{kJ}$, with a r2 score of $93.7\,\%$. Conversely, ElevModel provides a predicted solution of $9.368\,\mathrm{kJ}$, but which has an actual energy demand of $18.103\,\mathrm{kJ}$ and a r2 score of $26.9\,\%$. We also compare the two approaches with a ground truth method, called DynSim. In this method, the dynamic simulator, described in Section IV, is directly used to perform path planning. This means that the $C(n_p, n)$ values of each expanded node are measured by running the dynamic simulator and by measuring the driving energy with the method in IV-C. As this approach has access to the ground truth energy value during planning, it is able to provide exact predictions and the solution with the lowest driving energy, with a predicted and real energy value of $15.623\,\mathrm{kJ}$. However, DynSim assumes to have full domain knowledge of the environment during planning. While this is possible in the case of a simulator, it is often impractical in real-world scenarios. In contrast, Conv1D does not require any specific domain knowledge to be implemented, as it learns how to make energy estimations by collecting data. Hence, while in this paper we only focus on simulation, the Conv1D methodology can be in principle equally applicable to simulated and real-world scenarios. Moreover, the computational time of DynSim is measured to be more than 20 times higher than the other two methods. Specifically, we observe that, while Conv1D and ElevModel have similar average node expansion times at around $0.2\,\mathrm{s}$, DynSim takes on average $5.6\,\mathrm{s}$. The three methods are tested on an Intel Core i9-9940X CPU and, for the feed forward computation of the neural network, on a GeForce RTX 2080 Ti GPU.

An example of the three different solutions according to the three energy estimation methods is given in Fig. 6, while their quantitative results are shown in the second part of Table II. As with the previous results, DynSim has an exact prediction accuracy, and provides the best solution with predicted and real energy values of $27.72\,\mathrm{J}$. However, it also has the highest computational time with a total planning time of $51.8\,\mathrm{s}$. In contrast, Conv1D provides a slightly more energy demanding solution with predicted and real energy value of respectively $29.23\,\mathrm{J}$ and $30.44\,\mathrm{J}$, but in a fraction of the total planning time, with a value of $3.0\,\mathrm{s}$. Particularly, we can notice that the trajectory provided by Conv1D almost entirely corresponds to the one of DynSim. This provides evidence of the estimation accuracy of Conv1D, which results in a similar solution to the one selected by the ground truth method. Finally, ElevModel has the shortest planning time, but also the lowest prediction accuracy, and the most energy demanding solution. We can interestingly observe how ElevModel estimates its solution to provide a small quantity of energy recovery equal to $-0.99\,\mathrm{J}$. The robot proceeds on a slightly downhill trajectory and with a moderate roll inclination, which would lead to a marginal source of energy recovery if the surface was planar. However, the presence of unstructured geometries increases the actual energy to a value of $39.33\,\mathrm{J}$. This confirms the higher robustness of our method which, by considering the actual terrain geometry, can provide more informed estimations and more energy efficient paths.

VI. Conclusion and Future Works

In this paper, an integrated energy-aware prediction and planning framework is proposed for mobile robots in unstructured environments. Specifically, we remark the benefit of our methodology to (1) perform accurate driving energy estimations over highly unstructured terrain geometries by capturing the sequential context of the traversed terrain, and (2) reduce the robot driving energy by predicting and planning directly over feasible trajectories.

We are extending this work in several directions. While in the current implementation we focused on trajectories traversed at constant speed, the ability of our architecture to process data sequentially holds great promises for analysing the effect on driving energy of variable velocities. In addition, while assuming that the terrain geometry is the only external factor influencing driving energy can be an acceptable approximation in some scenarios (e.g. solid ground), it may not be adequate in others (e.g. deformable terrain such as sand or mud). Future works will consider strategies to include different terrain properties as an additional input when performing energy estimation. Finally, we intend to conduct experimental tests on a real robotic platform [36] to validate the results obtained during simulation.

Acknowledgement

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission. The authors are very grateful to the Autonomous Systems Group of RAL SPACE for providing the SEEKER dataset.

References

[1] O. Lamarre and J. Kelly, "Overcoming the challenges of solar rover autonomy: Enabling long-duration planetary navigation," *CoRR*, vol. abs/1805.05451, 2018. [Online]. Available: http://arxiv.org/abs/1805.05451

[2] E. Guizzo. Fukushima Robot Operator Writes Tell-All Blog. [Online]. Available: https://spectrum.ieee.org/automaton/robotics/industrial-robots/fukushima-robot-operator-diaries

[3] V. Gruning, J. Pentzer, S. Brennan, and K. Reichard, "Energy-aware path planning for skid-steer robots operating on hilly terrain," in *2020 American Control Conference (ACC)*, 2020, pp. 2094–2099.

[4] N. Ganganath, C. Cheng, and C. K. Tse, "A constraint-aware heuristic path planner for finding energy-efficient paths on uneven terrains," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 601–611, 2015.

[5] S. Fallah, B. Yue, O. Vahid-Araghi, and A. Khajepour, "Energy management of planetary rovers using a fast feature-based path planning and hardware-in-the-loop experiments," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2389–2401, 2013.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[7] D. Sancho-Pradel and Y. Gao, "A survey on terrain assessment techniques for autonomous operation of planetary robots," *JBIS - Journal of the British Interplanetary Society*, vol. 63, no. 5-6, pp. 206 – 217, May 2010.

[8] S. B. Goldberg, M. W. Maimone, and L. Matthies, "Stereo vision and rover navigation software for planetary exploration," in *Proceedings, IEEE Aerospace Conference*, vol. 5, March 2002, pp. 5–5.

[9] J. Carsten, A. Rankin, D. Ferguson, and A. Stentz, "Global path planning on board the mars exploration rovers," in *2007 IEEE Aerospace Conference*, 2007, pp. 1–11.

[10] M. Winter, C. Barclay, V. Pereira, R. Lancaster, M. Caceres, K. Mcmanamon, B. Nye, N. Silva, D. Lachat, and M. Campana, "Exomars rover vehicle: Detailed description of the gnc system," in *13th Symposium on Advanced Space Technologies in Robotics and Automation*, 05 2015.

[11] G. Ishigami, K. Nagatani, and K. Yoshida, "Path planning and evaluation for planetary rovers based on dynamic mobility index," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 601–606.

[12] D. Helmick, A. Angelova, and L. Matthies, "Terrain adaptive navigation for planetary rovers," *Journal of Field Robotics*, vol. 26, 04 2009.

[13] J.-F. Lalonde, N. Vandapel, D. Huber, and M. Hebert, "Natural terrain classification using three-dimensional ladar data for ground robot mobility," *J. Field Robotics*, vol. 23, pp. 839–861, 10 2006.

[14] M. Ono, T. J. Fuchs, A. Steffy, M. Maimone, and J. Yen, "Risk-aware planetary rover operation: Autonomous terrain classification and path planning," in *2015 IEEE Aerospace Conference*, March 2015, pp. 1–10.

[15] S. Zhou, J. Xi, M. McDaniel, T. Nishihata, P. Salesses, and K. Iagnemma, "Self-supervised learning to visually detect terrain surfaces for autonomous robots operating in forested terrain," *Journal of Field Robotics*, vol. 29, 03 2012.

[16] C. Spiteri, A. Shaukat, and Y. Gao, "Structure augmented monocular saliency for planetary rovers," *Robotics and Autonomous Systems*, vol. 88, 11 2016.

[17] R. Gonzalez and K. Iagnemma, "Slippage estimation and compensation for planetary exploration rovers. state of the art and future challenges," *Journal of Field Robotics*, vol. 35, no. 4, pp. 564–577, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21761

[18] R. González and K. Iagnemma, "Deepterramechanics: Terrain classification and slip estimation for ground robots via deep learning," *CoRR*, vol. abs/1806.07379, 2018. [Online]. Available: http://arxiv.org/abs/1806.07379

[19] K. Iagnemma, Shinwoo Kang, H. Shibly, and S. Dubowsky, "Online terrain parameter estimation for wheeled mobile robots with application to planetary rovers," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 921–927, Oct 2004.

[20] H. Shibly, K. Iagnemma, and S. Dubowsky, "An equivalent soil mechanics formulation for rigid wheels in deformable terrain, with application to planetary exploration rovers," *Journal of Terramechanics*, vol. 42, pp. 1–13, 01 2005.

[21] K. Ho, T. Peynot, and S. Sukkarieh, "A near-to-far non-parametric learning approach for estimating traversability in deformable terrain," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 2827–2833.

[22] S. Higa, Y. Iwashita, K. Otsu, M. Ono, O. Lamarre, A. Didier, and M. Hoffmann, "Vision-based estimation of driving energy for planetary rovers using deep learning and terramechanics," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3876–3883, 2019.

[23] P. Blacker, C. P. Bridges, and S. Hadfield, "Rapid prototyping of deep learning models on radiation hardened cpus," in *2019 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, July 2019, pp. 25–32.

[24] P. Krüsi, P. Furgale, M. Bosse, and R. Siegwart, "Driving on point clouds: Motion planning, trajectory optimization, and terrain assessment in generic nonplanar environments," *Journal of Field Robotics*, vol. 34, no. 5, pp. 940–984, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21700

[25] C. Connolly, "Cumulative generation of octree models from range data," in *Proceedings. 1984 IEEE International Conference on Robotics and Automation*, vol. 1, 1984, pp. 25–32.

[26] A. Shaukat, P. Blacker, C. Spiteri, and Y. Gao, "Towards camera-lidar fusion-based terrain modelling for planetary surfaces: Review and analysis," *Sensors*, vol. 16, no. 11, November 2016, (This article belongs to the Special Issue Vision-Based Sensors in Field Robotics). [Online]. Available: http://epubs.surrey.ac.uk/812938/

[27] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1d convolutional neural networks and applications: A survey," *arXiv preprint arXiv:1905.03554*, 2019.

[28] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[29] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, vol. 26, no. 3, pp. 308–333, 2009.

[30] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968.

[31] A. Bouton, C. Grand, and F. Benamar, "Obstacle negotiation learning for a compliant wheel-on-leg robot," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2420–2425.

[32] ——, "Motion control of a compliant wheel-leg robot for rough terrain crossing," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 2846–2851.

[33] ASRL. ASRL - Autonomous Space Robotics Lab. [Online]. Available: http://asrl.utias.utoronto.ca/

[34] senseFly. Elevation datasets. [Online]. Available: https://www.sensefly.com/

[35] M. Woods, A. Shaw, E. Tidey, B. Van Pham, L. Simon, R. Mukherji, B. Maddison, G. Cross, A. Kisdi, W. Tubby, G. Visentin, and G. Chong, "Seeker—autonomous long-range rover navigation for remote exploration," *Journal of Field Robotics*, vol. 31, no. 6, pp. 940–968, 2014.

[36] A. Bouton and Y. Gao. MARCEL: A Mobile Active Rover Chassis for Enhanced Locomotion. [Online]. Available: https://www.surrey.ac.uk/news/surrey-space-centre-wins-prestigious-robotics-award

[37] A. Sharma, N. Gupta, and E. Collins, "Energy efficient path planning for skid-steered autonomous ground vehicles," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 8045, 05 2011.

[38] T. Tieleman and G. Hinton, "Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural Networks for Machine Learning, 2012.