Staircase Negotiation Learning for Articulated Tracked Robots with Varying Degrees of Freedom

Andrei Mitriakov, Panagiotis Papadakis, Sao Mai Nguyen, Serge Garlatti

Staircase Negotiation Learning for Articulated Tracked Robots with Varying Degrees of Freedom

Andrei Mitriakov, Panagiotis Papadakis, Sao Mai Nguyen and Serge Garlatti

***Abstract*— Tracked robots capable of negotiating 3D terrains require delicate control, most often tailored to a specific platform or setting. For staircase traversal in particular, autonomous robot behaviours are difficult to obtain due to the increased risk of accident and stochasticity. Based on a previously developed reinforcement learning based framework that allows learning staircase ascent for an articulated tracked robot, in this work we extend our work to allow also staircase descent and further investigate the role of a manipulating arm in the stability and smoothness of the traversal. By relying on a precise simulation of geometry and kinematics of a real robot, we demonstrate prototype policies for staircase ascent and descent, optionally under the influence of an integrated active arm and different penalty criteria. The obtained results are qualitatively and quantitatively compared and show that the robot can learn plausible behaviors effectively, when guided by appropriate reward and penalty criteria.**

I. Introduction

Robots capable of 3D navigation bear a strong potential for applications pertaining to hazardous or extreme environments. In such cases, a resilient robot construction may compensate collisions with the environment as the robot passively negotiates the traversed terrain. On the other hand, applications where robot as well as environment safety is crucial require active control of the articulated robot parts.

Most previous works on the development of active 3D obstacle negotiation by tracked robotic vehicles are customized to specific platforms [1], [2], inevitably relying on complex inverse kinematics and accurate estimates of environment geometry. Such solutions are not easily transferable to different platform/environment setups and developing alternative negotiation strategies is not straightforward. Additionally, the development of a robot controller necessitates domain expertise or experience which may not be easy or even feasible to acquire, e.g. for search and rescue environments [3]. Learning from demonstration [4] can alleviate some of these challenges yet providing good demonstrations is costly, partly because controlling multiple degrees of freedom (DOF) in parallel is not intuitive and hence cumbersome. To increase autonomy in the process of learning robot mobility, we perform the acquisition of such skills via reinforcement learning (RL) where we can favor the emergence of the desired control properties through the appropriate design of reward/penalization functions and state/action domains. Since learning in reality is risky and tedious due to the need to perform manual environment reset, we opt for learning such skills through simulation.

Lab-STICC, UMR 6285, F-29238, team RAMBO, IMT Atlantique Bretagne/Pays de la Loire, Brest, France

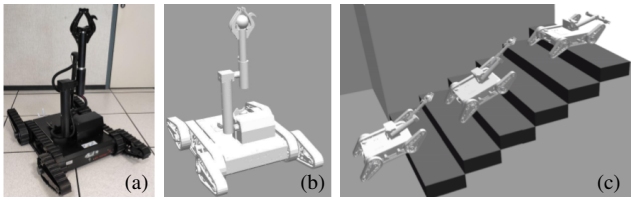

Fig. 1: Jaguar V4 Manipulator with raised arm (a), the corresponding CSM simulated model (b) and simulated ascent staircase negotiation using a learnt policy (c)

A recent approach for simulating robot tracks that provides more realistic robot track simulation and has shown highly plausible results and faster simulation is the Contact Surface Motion (CSM) [5], which we employ in the present work.

Relatedly, we recently developed a RL-based framework for tracked robots equipped with flippers, for learning ascent control policies of staircases of varying size [6]. In continuation of that work, we present here our latest developments for learning more elaborate mobility skills that are altogether critical in enabling tracked robots to undertake the commonly required task of object search and transport. The main contributions of this work can be summarized as follows:

- We upgrade our previous RL-problem formalization to allow learning of staircase ascent and descent and further investigate the influence of an active arm during staircase negotiation.
- We update our simulation model using the geometrical and physical properties of the robot *Jaguar V4 with Arm* (`jaguar.drrobot.com/specification_V4Arm.asp`) and adopt the CSM approach for the simulation of robot tracks (see Fig. 1).

The remainder of the paper is organized as follows. Section II presents most relevant works that deal with the problem of staircase negotiation for articulated tracked robots. In section III we augment our previous formalization [6] to further allow learning of staircase descent and consider the influence of an articulated arm. Finally, in section IV the experimental setup is presented together with the obtained qualitative and quantitative results.

II. Related works

The domain of Urban Search and Rescue (USAR) has drawn considerable attention to the problem of kinematic modelling in order to allow 3D terrain negotiation [7]–[9]. Our particular focus here regards staircase negotiation, generally considered as an important milestone in enabling robots to explore 3D indoor environments.

1) Learning-free approaches: Relatedly, Mourikis et al. [8] presented an algorithm for autonomous stair ascent through the use of gyroscopic data and estimation of stair step edges. Stair descent was studied in [10] together with stair traversal with the aid of an actively rotating pendulum in a complex track system, yet the obtained behavior is specific to the particular kinematics and hence not generalizable. An approach using dense laser sensory data and passive flippers was proposed in [11] where the robot orientates flippers tangentially to the traversing obstacle while being tele-operated. A planning and control method for 4-DOF tracked robots in autonomous ascent and descent of known staircase was proposed in [12] where the robot estimated its state through internal sensors. The main limitation of that work is that the robot can only negotiate staircases known beforehand. Finally, an elaborate analysis of robot stability and control was presented in [9], taking into account active arm adaptation during stair ascent and descent. The derived analytical solution is characteristic of the high complexity of direct modelling of the interaction between a multi-DOF robot and a negotiated obstacle. Still, it requires full knowledge of the platform kinematic, geometric and physical characteristics, it does not employ congruent control of flippers, main tracks and the arm, nor does it account for the influence of a transported object.

2) Learning-based approaches: Learning-based control approaches on the other hand, tend to be more straightforward when the dynamics of physical interaction are hard to be estimated. Authors of [13] were among the first who treated the problem of staircase traversal via deep reinforcement learning in the end-to-end fashion. Using front and rear cameras they showed that the robot can learn to ascend a previously known staircase, yet at a high computational cost. Reinforcement learning was also applied to the problem of flipper configuration and compliance autonomous control by [14] in order to reduce the operator's cognitive load. That approach enabled negotiation of single palettes and uneven terrain obstacles, nonetheless, assuming that main tracks are independently controlled. Another more sophisticated approach was demonstrated in [15] where authors incorporated constraints into the optimization problem of relative entropy policy search using a RL algorithm [16]. Still, results were limited to the traversal of a palette.

Endowing a system with constraint respect such as stability can be performed in different manners. It is a common practice to introduce safety constraints into the RL algorithm [15], [17], often giving promising results at the cost of exhaustive exploration effort. Alternatively, one can associate safety and reward inside a single function [14], [18], which raises the question of determining their relative influence. In the following, we describe our approach for RL-based staircase ascent and descent under the influence of an integrated arm transporting an object. The diversification of the task incurred by the inclusion of new DOF leads to a harder problem and higher risk of accident, requiring a more comprehensive treatment. In comparison with related works, our framework is easily applicable to different platforms, allows to generalize over a variety of obstacles, accounts for arm control and considers ascent and descent traversals.

III. Methodology

At time step t, the system being in the state $\boldsymbol{s}_t \in S$ executes the action $\boldsymbol{a}_t \in A$ with respect to its policy π, obtains a scalar reward $r_t \in \mathbb{R}$ and transits into a new state $\boldsymbol{s}_{t+1} \in S$. The total set of transitions from the episode start to the end is called a roll-out $\tau = \{\boldsymbol{s}_0, \boldsymbol{a}_0, .., \boldsymbol{s}_{T-1}, \boldsymbol{a}_{T-1}\}$ where T is its number of time steps. To solve this RL problem, we will use policy gradient algorithms where the policy is directly optimized. The main idea consists in performing policy ascent over policy parameters $\boldsymbol{\theta}$ to maximise the expected gradient $\nabla_\theta J(\pi_\theta)$ return. In the sequel, we present the variables used to describe the problem of ascent and descent staircase negotiation learning. Then, we propose a reward function to entice stair traversal while taking into account the stability constraints.

A. Problem description

In our previous work [6] where we opted for independent tracks and flipper control, the robot was able to learn a behavior required for accomplishing staircase ascent while respecting safety constraints. However, with the inclusion of DOF of an articulated arm, separate control of the main tracks with the arms actuators can be sub-optimal.

This motivated us to explore two types of action spaces; (i) when the robot controls 3 DOF consisting of the robot base linear velocity, front flipper and rear flipper angles and (ii) $3 + 2 = 5$ DOF where the 2 additional DOF correspond to the joints of an arm. With reference to (i), the robot selects 3 action parameters, namely, front and rear flippers rotation angles and velocity of the base forming an action vector:

$$\begin{aligned} \boldsymbol{a} = (\psi_a^{front}, \psi_a^{rear}, v_a) \in \\ [\psi_a^{min}, \psi_a^{max}]^2 \times [v_{min}, v_{max}] \end{aligned} \tag{1}$$

where ψ_a^{front} and ψ_a^{rear} are front and rear flipper angles, v_a is the applied velocity. For case (ii), the previous action vector is extended with the joint angles of the arm:

$$(\phi_a^1, \phi_a^2) \in [\phi_1^{min}, \phi_1^{max}] \times [\phi_2^{min}, \phi_2^{max}] \tag{2}$$

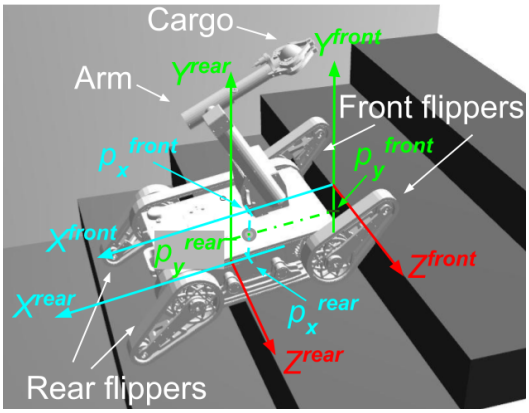

Fig. 2: Front and rear obstacle (step edge) coordinate systems

that correspond respectively to the angles of the first and second joints (see Fig. 2).

The sensory space structure is set as follows. For case (i), the observation vector is represented as:

$$\boldsymbol{s} = (p_x^{front}, p_y^{front}, p_x^{rear}, p_y^{rear}, v_s, \psi_s^{front}, \psi_s^{rear}) \quad (3)$$

where p_x^{front}, p_x^{rear} are distances of the robot centroid relative to the next and previous nearest step edge along the X-axis of the robot as shown in Fig. 2, with p_y^{front}, p_y^{rear} corresponding to the relative distances along the Y-axis and v_s being the current linear velocity. The state further includes ψ_s^{front} and ψ_s^{rear} as the current flipper angles. For case (ii), the state vector is extended with the arm joint angles ϕ_s^1, ϕ_s^2.

B. Reward function design

In this section we present reward functions used for learning staircase ascent and descent. In detail, we employ the same positive reward as in [6] representing the total travelled distance on the stairs which drives learning, namely:

$$R^{tr}(\tau) = \frac{\sum_{t=1}^{N_\tau} x_t}{D_{max}} \quad (4)$$

where $R^{tr}(\tau)$ is a cumulative return along a rollout τ whose maximal value can attain 1, N_τ is the number of time steps in the rollout, x_t is a travelled distance during one time step, D_{max} is the maximal possible travelled distance. As stated earlier, learning with constraints can be performed through accounting for a negative reward (penalty) within an episode. We first propose a reward function that addresses the problem of robot stability for ascent and descent negotiation. Then, a reward function meant to reduce the pitch angular velocity experienced by the platform during descent is proposed.

1) **Center of Gravity stabilization:** The center of gravity of the robotic platform is known to be instrumental in preventing tip-over accidents and improving stability [9], [19]. In [6] we enabled a robot to learn a stable behavior by favoring robot poses with low center of gravity (COG) with respect to the underlying surface A' that represents the Normalized Energy Stability Margin (NESM) [20]. The presence of an arm nonetheless may place the COG low but also close to the "safety zone" border [9], without violating the NESM. To overcome this problem, we use the Stability Margin (SM) [21] that estimates the distance between the lower footprint and the COG projection on the ground. Since it is generally difficult to estimate where exactly the lower footprint touches the ground, we instead minimize the deviation of the COG projection on the stair surface C_x from the projection point O of the centroid of the robot base. In this manner, we can guarantee that the robot respects the SM criterion. We consider that the arm has to place the COG as close as possible to the point O - the most stable point (see Fig. 3) along both the X and Y axes, optimizing both for the SM and the NESM. Thus, we minimize the deviation $D = \sqrt{d^2 + h^2}$ where d and h are the distances between O and the projections of the COG on X and Y axes. This minimization serves the purpose of improving stability through the SM and NESM criteria but also redistribute symmetrically the weight forces exerted by the moving robot tracks to the staircase.

We further wish to penalize the robot when it loses stability and tips over that happens when the pitch angle of the robot reaches $\pi/2$. Thus, letting the COG deviation at every time step be D_t, the penalty term is defined as:

$$r_t^D = \begin{cases} -1, & \text{if tip over} \\ -K_D * D_t, & \text{otherwise} \end{cases} \quad (5)$$

where the scaling coefficient K_D is used for normalisation as will be explained later in III-B.3.

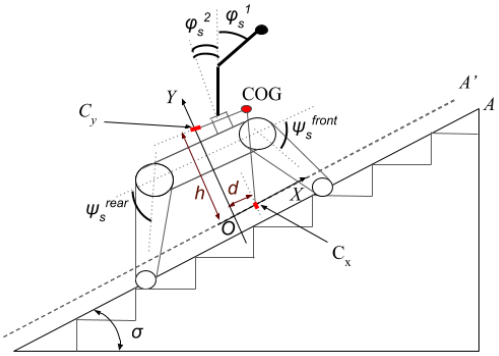

Fig. 3: Side schematic view of the robot on a staircase

2) **Drop impact reduction:** Robot dynamics are different between ascending and descending a staircase, as gravity hinders the accomplishment of the former and it may conduct the robot to fall in the latter. Even if no accident happens, repetitive collisions caused by every step negotiation would have impact on the robot. Such events appear when the COG crosses step edges, at this moment, the base starts rotating downwards which yields an increase of pitch angular velocity that we refer as drop impact (bumpiness). As a way to prevent such behavior, the robot can learn to influence these dynamics through a pitch angular velocity based penalty. The robot may experience a front tip-over when the pitch angle of the robot reaches $-\pi/2$, that we wish to penalize as well. Letting W_t denote the average pitch angular velocity, then the corresponding negative reward is:

$$r_t^W = \begin{cases} -1, & \text{if tip over} \\ -K_W * W_t, & \text{otherwise} \end{cases} \quad (6)$$

3) **Adjustment of scaling coefficient:** The learned policy is largely determined by the reward function that guides learning of a specific task. At the same time, it can endow the policy with specific characteristics such as improvement of traction and respect of safety. While there is no ambiguity if we learn a specific task with only positive rewards, direct summing up the latter with another reward, for example the negative one, may mislead - bias the robot to concentrate on respecting auxiliary constraints rather than the main task.

To control the arbitrary bias of the penalty term, we normalize it by a scaling coefficient whose approximation can balance the learning of the main task along with the desired property. Towards this goal, we initiate a small set of

trial experiments for the first n steps where we do not assign negative rewards, in order to acquire sufficient statistics related to the target value V, for example, COG distance D or angular velocity W in our case. This results in making the robot optimize its behavior only with respect to the main task as specified by the positive reward, setting aside the consideration of the penalty term. Upon the completion of the n^{th} step, we calculate the "sub-optimal" mean target value $\hat{V} = \sum_{t=1}^{n} V_t/n$, then the scaling coefficient as follows:

$$K_V = (\hat{V} \cdot N_{episode})^{-1} \tag{7}$$

where $N_{episode}$ is the maximum episode length. Thus, the cumulative episode return could be expressed as follows:

$$R_V(\tau) = R^{tr}(\tau) + \sum_{t=1}^{N_\tau} r_t^V \tag{8}$$

Normalization guarantees that the absolute episode return $R^{tr}(\tau)$ varies from 0 to 1 as well as the absolute episode penalty without tipping over penalization $\tilde{R}_V^{penalty}(\tau)$. This can be easily seen by defining:

$$\tilde{R}_V^{penalty}(\tau) = \frac{\sum_{t=1}^{N_\tau} V_t}{\hat{V} \cdot N_{episode}} \cdot \frac{N_\tau}{N_\tau} = \frac{\hat{V}_\tau N_\tau}{\hat{V} \cdot N_{episode}} \tag{9}$$

where $\hat{V}_\tau$ is the mean target value during the episode after trial experiments. We consider that in the beginning of learning the policy is sub-optimal, thus the "sub-optimal" mean target value $\hat{V}$ calculated in trial experiments relates to further target value mean observations as $\hat{V}_\tau \leq \hat{V}$. Since the rollout length does not exceed its maximal length $N_\tau \leq N_{episode}$, we can conclude:

$$\hat{V}_\tau N_\tau \leq \hat{V} \cdot N_{episode} \tag{10}$$

It was finally observed that the tip-over penalty prevents learning of sub-optimal policies, therefore we added auxiliary negative reward -1 and end the episode in such cases, thus, $R_V^{penalty}(\tau) \in [-2, 0]$. Finally, we obtain the limits for the episode cumulative return as $R_V(\tau) \in [-2, 1]$.

IV. Experiments

In this section we present the experiments that we conducted, allowing us to obtain staircase ascent and descent policies with desired properties. We also provide a qualitative analysis of learned behaviours in the supplementary video (also available at `partage.imt.fr/index.php/s/JBdmEXaWcjLmgmB/download`).

*1) **Environment set-up**:* We developed a simulation environment as illustrated in Fig. 1, using the physics-based Gazebo simulator (`gazebosim.org`). Staircase size was varied in ranges corresponding to real-world staircases (cf. [6] for details), allowing to learn behaviours in randomly generated staircases further taking into account the influence of noise in the state estimation process.

Furthermore, we increased the maximum total number of steps from 5 to 10. More importantly, we have shifted from a wheeled-based simulation of robot tracks to the more realistic CSM model [22] that is more lightweight in terms of surface collision calculations and increases the level of accuracy. We instantiate a model of the robot Jaguar V4 and add a manipulator arm platform (see Fig. 1) incorporating its geometrical and weight parameters. To study the influence of transporting objects we perform learning with a cargo added to the robot end-effector that constitutes 10% of total robot mass and the heaviest permissible load, then we show resilience to its variation via test roll-outs with cargo masses that constitutes 5% and 15% of total robot mass.

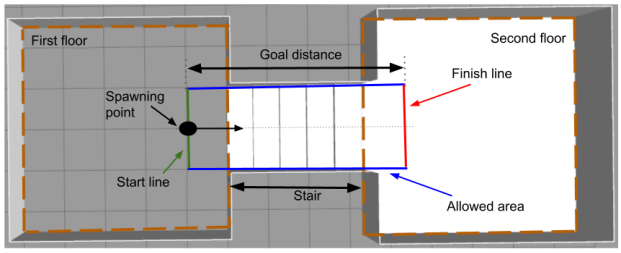

Fig. 4: Illustration of stair traversal task set-up

A two-layer perceptron is used for policy representation. Its parameters are optimised using Proximal Policy Optimization [23]. In the beginning of the ascent tasks, the robot starts at the start point (see Fig. 4) being orientated towards the stairs. It has to traverse the distance from the start to the finish line. For the descent tasks, start and goal positions are swapped. We assign both positive and negative rewards from the beginning of the episode to its end. The episode ends when either the robot reaches the finish line, goes out of the training zone (rectangle defined by red, blue and green lines), tips over or number of time steps exceeds the maximal episode length. We studied learning a total of 5 variants of the staircase traversal task, distinguished by the staircase negotiation direction, DOF involvement and penalty optimization criterion as summarized in Table I.

For all experiments the robot arm starts with the same, vertically stretched, initial configuration perpendicular to the robot base (see Fig. 1 middle). We select this initial arm pose as it does not alter the COG projection of the robot when it moves on flat terrain but it may severely hinder staircase traversal and should therefore incite the robot to learn to move the arm at a better pose. We recall that in the case of a 3DOF action space, the arm remains fixed in this pose.

*2) **Performance for staircase ascent**:* Three independent learning trials were performed for each task (i) and (ii), after which we average the obtained performances. Fig. 5 (a) presents the average over all trials of the exponentially smoothed cumulative reward with min-max bands. The smoothing factor is set to 0.95, considering the trade-off

TABLE I: Description of tasks

Task id	Direction	DOF	Criterion
i	Ascent	3	COG
ii	Ascent	5	COG
iii	Descent	3	Ang. vel.
iv	Descent	5	Ang. vel.
v	Descent	5	COG

between curve smoothing while highlighting local changes. At the end of learning, a total relative reward gain of 0.21 is observed between the tasks, clearly suggesting that the robot learns to control the arm in a way that optimizes the respective criterion. Cumulative reward curves do not reach their maximum positive cumulative return value of 1 because of the minimal COG bias that results from its inability to put the COG closer to the base centroid than a certain distance. It is worth noticing of the considerable distance between the two curves by the end of learning, explained by the fact that the robot with static raised arm and only using its flippers, is unable to attain the same level of positive reward because of the higher COG deviation.

Fig. 5 (b) shows how the COG deviation evolves during the learning process. We can see that the robot learns its dynamics in both tasks. It shows that in both cases the robot converges to a better behaviour in a similar pace. Eventually, the arm control allows to decrease the COG deviation by $0.06m$ compared to the beginning of learning. This figure contains the COG evaluation curve obtained for the task (ii) without reward shaping. This illustrates how absence of reward scaling worsens final optimal control policy. We can see that the sub-optimal policy increases the COG deviation and converges to the mean value which exceeds the corresponding optimal one by $0.15m$.

3) ***Performance for staircase descent****:* Cumulative reward curves are presented in Fig. 6a. We can see that the reward for all tasks reaches approximately the value 0.6 and does not attain the maximal reward 1 because of constant presence of the minimal negative reward. Fig. 6b shows the evaluation of the COG deviation during learning. As expected, we observe that the most unstable robot behavior is obtained for task (iii), where the reduction of COG distance due to the flipper control from the centroid projection point is low. Results

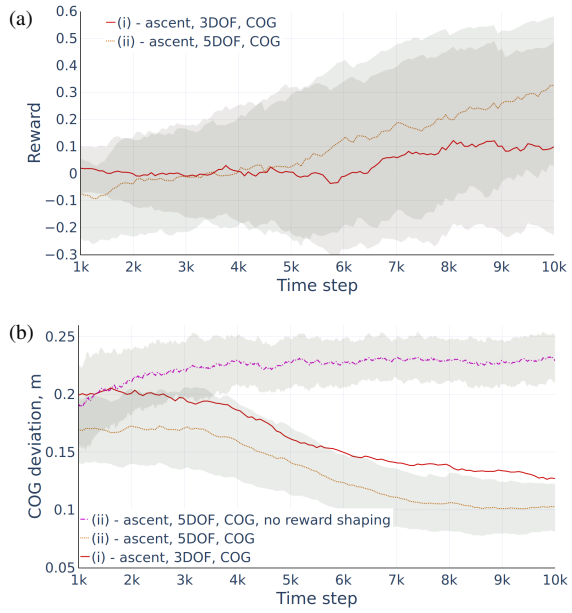

Fig. 5: Ascent task learning analysis

of pitch angular velocity optimization with moving arm (iv) improves stability and the robot has further achieved to reduce the COG deviation by a total of $0.02m$. Nevertheless, direct minimization of the COG distance (v) provides the best overall results, wherein the COG decreases by more than $0.03m$ and attains the lowest absolute value among tasks.

The pitch angular velocity evaluation is presented in Fig. 6c. We can see that COG and pitch angular velocity optimizations (iv) and (v) behave quite similarly, with the former yielding a slightly lower angular velocity. Comparing it with the performance of task (iii), we could claim that the mean pitch angular velocity is smaller if the arm does not move. Also, it may seem that the robot control policy fails to converge to an optimal behaviour but this could be partially explained. We have seen that the initial, vertically stretched arm performs worst of all in terms of stability easily leading to front tip-over, end of episode and penalization by -1. Thus, starting in the vertical arm position, the robot would experience more tip-overs that drives the arm control to more stable configuration even if it increases pitch angular velocity during movements. This seems a plausible behavior in reality, as we would prefer to undergo small bump impacts due to increased pitch angular velocity instead of a drastic accident.

4) ***Overall performance****:* We refer the interested reader to the qualitative results provided in the accompanying video, which presents the policies learned for the ascent and descent tasks in simulation. Policy learning time is constrained by action execution duration in the simulation. Using a contemporary machine, a 10000 time step simulation required approximately 20 minutes. As an empirical check that our simulation environment is realistic, we also transferred a learned policy from simulation to a real robot in an example staircase for the task (iii) policy.

Fig. 8a presents mean values with min-max of the COG distance during a test episode for ascent tasks (i) and (ii). We can see that after the initial phase of the task progress, i.e. from 30% of the ascent of the staircase, the COG deviation for the policy (ii) is lower than for the policy (i), further illustrated by Fig. 9 which presents mean values of COG deviations and angular velocities with their standard deviations during simulation test deployment for corresponding experiments of Table I. The mean COG deviation of the policy (i) is higher of the policy (ii) by $0.03m$. Fig. 7 provides indicative snapshots of robot configurations during ascent (a) and descent (b) transitions while respecting the COG criterion. We can see that the robot pushes its arm in front during ascent owing to which tip-over of the platform is largely avoided. The front flippers are raised during ascent while rear flippers are pushed down. Such configuration improves robot traction with front and rear step edges.

Mean transition curves given in Fig. 8b show the evolution of the angular velocity during descent negotiation. To evaluate policy effectiveness in the descent tasks we compare quantitatively corresponding policies with the help of Fig. 9. Angular velocity optimization shows better results for the policies (iii) and (iv) in contrast to the policy (v) which is correspondingly higher by around $0.04rad/s$ and $0.05rad/s$.

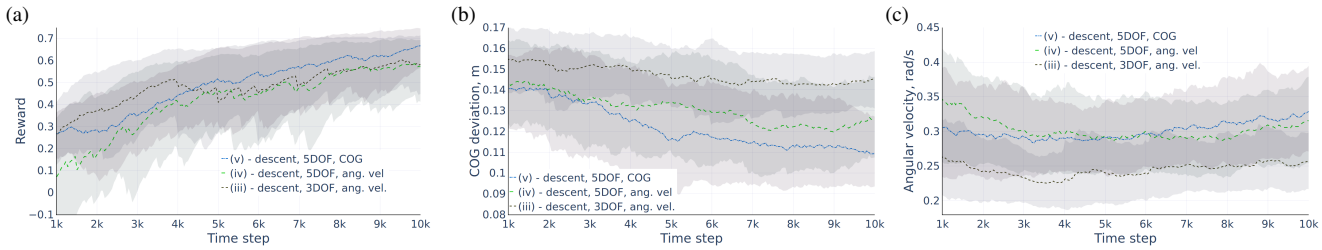

Fig. 6: Descent task learning analysis

Yet, the latter shows the smaller COG deviation by $0.018m$ and $0.028m$. The robot configuration respecting the COG criterion during descent negotiation is shown in Fig. 7 (b). Flipper rotation angles are reversed in the manner that front and rear flippers touch correspondingly lower and upper obstacles making transitions smoother. Thus, front flippers prevent the robot from dropping forward while rear flippers allow the robot to descend smoother from the rear step edge. We favor pitch angular velocity reduction in the descent tasks (iii) and (iv), but the stability (v) remains important. In reality, we would prefer to expose the robot to some drop impacts rather to tip-over that could damage the robot.

We have performed additional tests in the ascent and descent tasks for the same policies when the end effector holds loads constituting 5 to 15% of the robot mass. We can observe from Fig. 9 that the robot is capable of controlling the COG deviation if load mass changes in the ascent tasks. For example, the robot controls the arm in a manner that it improves the target value for tests with loads of $5, 10$ and 15% of the robot mass by $0.046, 0.044$ and $0.083m$ in comparison with the stretched arm tests. In the descent tasks, the COG deviation is the highest in the task (iii) for all load types. Then, it is decreased if the robot controls its arm to minimize the angular velocity, but the minimal target value corresponds to the policy of the task (v). The mean angular velocity decreases along with load mass augmentation for all tasks. We can conclude that the angular velocity is smaller for tasks (iii) and (iv) than for task (v).

V. Conclusion

We have presented an improved RL-based framework for the problem of control policy learning during staircase descent and ascent, further investigating the influence of an integrated arm. Within 150 episodes, the robot is able to learn its dynamics and safety constraints while negotiating varying staircases. The automated scaling coefficient estimation proved effective by constraining cumulative returns to appropriate scales and avoiding biasing the convergence of the obtained optimal policies. We have studied optimization of the COG and pitch angle velocity criteria for both ascent and descent. The COG based policies have exhibited better performance in terms of stability and qualitatively presented the same optimal arm control behavior in comparison with ordinary control. Despite the addition of more DOF, the control of the arm yields better overall stability to the robot during traversal. The angular velocity minimization showed minor improvements of the policy and indirectly improved the COG deviation. The learned control has shown resilience in application to different carrying loads. Finally, policy test evaluations showed that optimizing the COG criterion allows the execution of ascent and descent in a safer manner.

We are currently working on transferring the learned behavior to the real robot, integrating auxiliary components such as robot localization while we further consider comparing against conventional control or other learning-based techniques to different platform configurations.

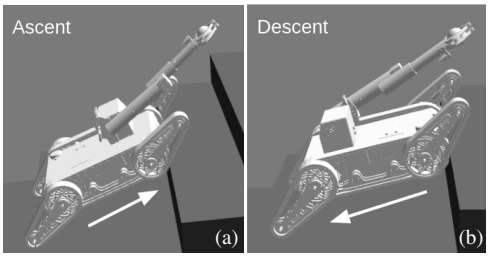

Fig. 7: Robot during transitions on the staircase; front arm configuration (a), rear arm configuration (b)

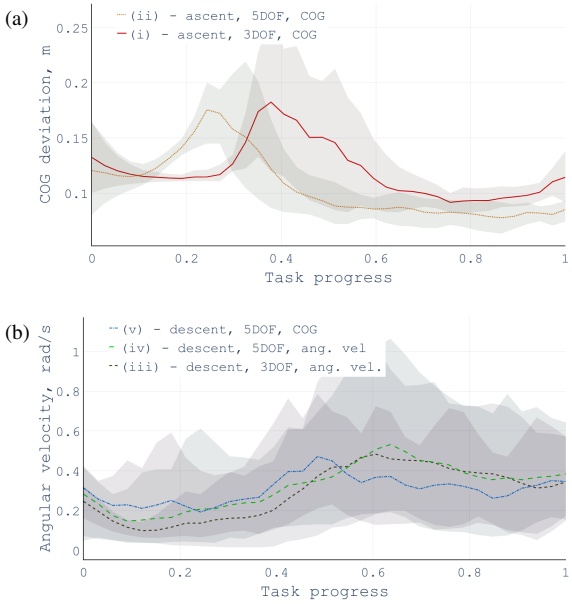

Fig. 8: Evolution of penalty values during task execution

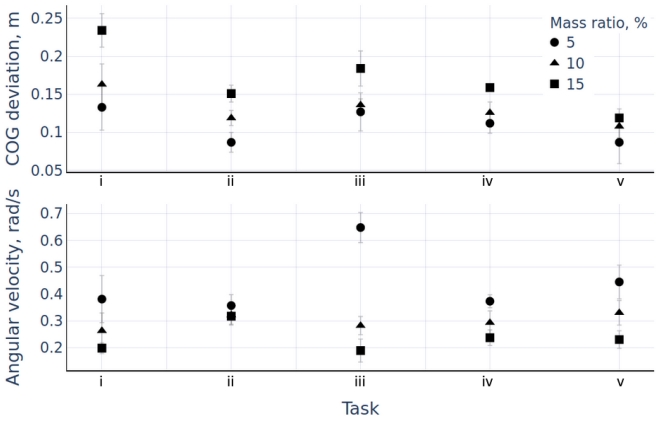

Fig. 9: Performance of learned policies with varying weight of transported object

VI. Acknowledgements

The work is performed in the context of the project REACT, project VITAAL and is financed by Brest Metropole, the region of Brittany and the European Regional Development Fund. Sao Mai Nguyen acknowledges the support of U2IS, ENSTA, IP Paris and of Inria FLOWERS, France.

References

[1] K. Otsu, G. Matheron, S. Ghosh, O. Toupet, and M. Ono, "Fast approximate clearance evaluation for rovers with articulated suspension systems," *J. of Field Robotics*, 2019.

[2] B. D. Ilhan, A. M. Johnson, and D. E. Koditschek, "Autonomous stairwell ascent," *Robotica*, vol. 38, no. 1, p. 159–170, 2020.

[3] G. M. Kruijff, F. Pirri, M. Gianni, P. Papadakis, M. Pizzoli, A. Sinha, V. Tretyakov, T. Linder, E. Pianese, S. Corrao, F. Priori, S. Febrini, and S. Angeletti, "Rescue robots at earthquake-hit mirandola, italy: A field report," in *IEEE Int. Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2012.

[4] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, *Handbook of Robotics*. MIT Press, 2007, no. 59, ch. Robot Programming by Demonstration.

[5] M. Pecka, K. Zimmermann, and T. Svoboda, "Fast simulation of vehicles with non-deformable tracks," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017.

[6] A. Mitriakov, P. Papadakis, S. M. Nguyen, and S. Garlatti, "Staircase traversal via reinforcement learning for active reconfiguration of assistive robots," in *IEEE Int. Conf. on Fuzzy Systems*, 2020.

[7] J. L. Martínez, A. Mandow, J. Morales, S. Pedraza, and A. García-Cerezo, "Approximating kinematics for tracked mobile robots," *The Int. J. of Robotics Research*, vol. 24, no. 10, pp. 867–878, 2005.

[8] A. I. Mouriakis, N. Trawny, S. I. Roumeliotis, D. M. Helmick, and L. Matthies, "Autonomous stair climbing for tracked vehicles," *The Int. J. of Robotics Research*, vol. 26, no. 7, pp. 737–758, 2007.

[9] H. Zhang and A. Song, "System centroid position based tipover stability enhancement method for a tracked search and rescue robot," *Advanced Robotics*, vol. 28, no. 23, pp. 1571–1585, 2014.

[10] P. Ben-Tzvi, S. Ito, and A. A. Goldenberg, "A mobile robot with autonomous climbing and descending of stairs," *Robotica*, vol. 27, no. 2, p. 171–188, 2009.

[11] Keiji Nagatani, Ayato Yamasaki, Kazuya Yoshida, Tomoaki Yoshida, and Eiji Koyanagi, "Semi-autonomous traversal on uneven terrain for a tracked vehicle using autonomous control of active flippers," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008.

[12] D. Endo, A. Watanabe, and K. Nagatani, "Stair climbing control for 4-dof tracked vehicle based on internal sensors," *J. of Robotics*, pp. 1–18, 10 2017.

[13] G. Paolo, L. Tai, and M. Liu, "Towards continuous control of flippers for a multi-terrain robot using deep reinforcement learning," *CoRR*, vol. abs/1709.08430, 2017.

[14] K. Zimmermann, P. Zuzanek, M. Reinstein, and V. Hlavac, "Adaptive traversability of unknown complex terrain with obstacles for mobile robots," in *IEEE Int. Conf. on Robotics and Automation*, 2014.

[15] M. Pecka, V. Šalanský, K. Zimmermann, and T. Svoboda, "Autonomous flipper control with safety constraints," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016.

[16] M. P. Deisenroth, G. Neumann, and J. Peters, "A survey on policy search for robotics," *Found. Trends Robot*, vol. 2, pp. 1–142, 2013.

[17] S. Ha, P. Xu, Z. Tan, S. Levine, and J. Tan, "Learning to walk in the real world with minimal human effort," *arXiv preprint arXiv:2002.08550*, 2020.

[18] J. Bagnell, "Learning decision: Robustness, uncertainty, and approximation," *PhD Thesis*, 2012.

[19] K. Sasaki, Y. Eguchi, and K. Suzuki, "Stair-climbing wheelchair with lever propulsion control of rotary legs," *Advanced Robotics*, vol. 34, no. 12, pp. 802–813, 2020.

[20] S. Hirose, H. Tsukagoshi, and K. Yoneda, "Normalized energy stability margin and its contour of walking vehicles on rough terrain," in *IEEE Int. Conf. on Robotics and Automation*, 2001.

[21] R. McGhee and A. Frank, "On the stability properties of quadruped creeping gaits," *Mathematical Biosciences*, vol. 3, pp. 331 – 351, 1968.

[22] M. Pecka, K. Zimmermann, and T. Svoboda, "Fast simulation of vehicles with non-deformable tracks," *CoRR*, vol. abs/1703.04316, 2017.

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.