Inferring Cost Functions Using Reward Parameter Search and Policy Gradient Reinforcement Learning

Emir Arditi
Dept. of Computer Science
Ozyegin University
Istanbul, Turkey
emir.arditi@ozu.edu.tr

Tjasa Kunavar
Dept. of Robotics
Jozef Stefan Institute
Ljubljana, Slovenia
tjasa.Kunavar@ijs.si

Emre Ugur
Dept. of Computer Science
Bogazici University
Istanbul, Turkey
emre.ugur@boun.edu.tr

Jan Babic
Dept. of Robotics
Jozef Stefan Institute
Ljubljana, Slovenia
jan.babic@ijs.si

Erhan Oztop
Ozyegin University &
Osaka University
Osaka, Japan
erhan.oztop@otri.osaka-u.ac.jp

***Abstract*—This study focuses on inferring a cost functions pertaining to movement data using reward parameter search and policy gradient based Reinforcement Learning (RL). The behavior data for this task is obtained through a series of squat-to-stand movements of human participants under dynamic perturbations. The key parameter searched in the cost function is the weight of total torque used in performing the squat-to-stand action. An approximate model is used to learn squat-to-stand movements via a policy gradient method, namely Proximal Policy Optimization(PPO). A behavioral similarity metric based on Center of Mass(COM) is used to find the most likely weight parameter. The stochasticity in the training result of PPO is dealt with multiple runs, and as a result, a reasonable and a stable Inverse Reinforcement Learning(IRL) algorithm is obtained in terms of performance. The results indicate that for some participants, the reward function parameters of the experts were inferred successfully.**

I. Introduction

Recent Reinforcement Learning methods have been proven successful for solving challenging artificial and real life problems. With Q-Learning fundamentals, [1] surpassed the capabilities of humans in several environments in the *Atari* game console by learning from images and acting on top of a discrete action space. Although this approach was a big success, one thing it did not cover was how to deal with environments with continuous action spaces. This fundamental problem was tackled by moving towards direct policy approximation. [2], [3], and [4] proposed different modifications for vanilla actor-critic systems, and all of them reported good results for obtaining policies that perform the task at hand to a satisfactory level.

Although the aforementioned approaches find impressive policies, there is no guarantee about the optimality of the found solution. As deep neural networks are employed as function approximators, each training session finds a different solution -albeit maybe slightly different- due to the stochastic nature of learning and randomness-based exploration. This is not usually an issue for many reinforcement learning problems that does not require strict optimality as in game playing, and qualitative task descriptions.

For example, experimentation conducted in study [4], mainly focuses on tasks of walking with several different variations of the dynamic body. The main goal focuses on either reaching to some point B, starting from another point A, or being able to run freely and not falling down. Their results show that the method they propose is sufficient enough for their goal as they mainly focus on feasibility instead of complete optimality. If, however when optimal solutions are strictly needed, for instance when finding the parameters of a parameterized reward function associated with an optimal behavior by using an RL solver within a search loop, the variability of the solution found by the RL method becomes critical. In general, finding a reward or cost function that best describes an observed (optimal) behavior is called Inverse Reinforcement Learning (IRL).

Recent focus on IRL research addresses two main research directions. The first one is policy extraction and imitation. In this process, the aim is to focus on a faithful imitation rather than understanding the underlying reason of the demonstrator, as such the focus is on directly searching/learning a policy to generate a behavior similar to the observed one. These processes try to directly search and learn the policy [5] [6] [7]. These approaches have been proven successful in a range of both artificial and real life based tasks for imitation of an expert.

Another focus is towards extracting a reward function with an inner solver for generating policies according to the extracted function. generation [8] [9] [10]. These studies does not address behavior imitation only; but, they also aim to understand the reason behind the demonstrations from the expert, sometimes described as understanding or uncovering the optimality principles of the expert. These studies tested their frameworks in a wide range of environments. Since IRL is an ill-posed problem, these approaches naturally come with several assumptions to constrain the solution space. However,

there are some fundamental assumptions which should be emphasized. The fundamental assumption for IRL is that the demonstrations given are optimal, although to ensure the validity of this assumption is not always possible. This is a problem for all IRL methods. The assumption for RL based IRL methods (such as [8] [9] [10]), is that the inner RL solver will always converge to a single optimal policy with respect to the current structure of the reward function. While sometimes this assumption can be true for basic environments, it can also become invalid if there are multiple optimal policies or the task is too complex for the RL solver to reach to one solution. These cases usually end up with the RL agent converging into a near-optimal policy.

Due to the assumptions described, policy gradient methods seem to be the winner for continuous state and continuous action based RL problems. However, it is not clear how they must used as the RL solver in the reward parameter search loop to obtain a robust IRL algorithm. In particular, it is of interest to know the applicability of policy gradient based IRL algorithms for analyzing human sensorimotor data. Note that these methods can be successfully applied to non-trivial imitation learning problems(e.g [7]), but, the application of reward function extraction based IRL techniques for real life tasks is scarce. One reason for this is, the amount of IRL methods that can be applicable to continuous action and state space problems are very few.(e.g. [11], local method, need to take the derivative of the dynamic system). There are very limited amount of studies which utilizes the concept of parameter-search based IRL.(e.g [12]). Such straightforward methods are attractive in the sense that they can provide baseline solutions without any strong assumptions or approximations. However, the RL method used in the inner loop, is often a stochastic method which does not always find the optimal solution. This case is especially present for continuous state and continuous action based systems. So it is critical to know how much variance the inner RL solution induces on the reward function.

This study proposes a parameter-search based IRL method that aims to tackle the second issue of these systems by analyzing multiple training sessions and trying to extract meaningful information from them. For this study, we have defined a parameterized reward function for our inner RL solver and tried to obtain reward function parameters to yield behaviors similar to the demonstration. We test our method by reproducing the human experiment setup in a simulation environment and compare our results with a human squat-to-stand data under perturbation and try to understand how much the subjects *value* the effort(amount of torque they used) they need to spend standing up while at the same time compensating for the perturbation and keeping their balance. The perturbation was given in the form of a backward continuous pull with a force relative to the vertical speed of the subjects center of mass.

II. Method

A. Simulation Environment and Task Definition

We have implemented a basic 3-DOF dynamic system using PyDy [13]. In the start of the simulation, the agent is in a squat pose as can be seen in Figure 1. The task is to turn the pose of the dynamic system to a stand pose by providing "correct" torque values to each joint at each time step. The agent is given a limited amount of time and additional constraints including joint limits and height constraints in order to reproduce a realistic dynamic model. The addition of perturbation, aims to deliver a smaller solution space for the environment defined. The state and action of the system is defined as $s = [x_1, x_2, x_3, \dot{x_1}, \dot{x_2}, \dot{x_3}]$ and $u = [u_1, u_2, u_3]$, where x_n and $\dot{x_n}$ represent the angle and the angular velocity of the n$^{\text{th}}$ joint; and u_n denotes the torque applied to the corresponding joint.

B. Data Collection for Analysis

In order to test the applicability of our system for real behavioral data, we used Center of Mass(COM) trajectories of human subjects, from an experiment consisting of a squat-to stand task with external perturbations. The setup of this experiment can be found on Figure 1. This tasks extends the generic standing up task with an additional backwards perturbation to assess the change in behavior. In this task, the subjects must learn to stand up without making a corrective step while regulating their effort. We tried to quantify this effort by defining a composite reward function explained in the next section.

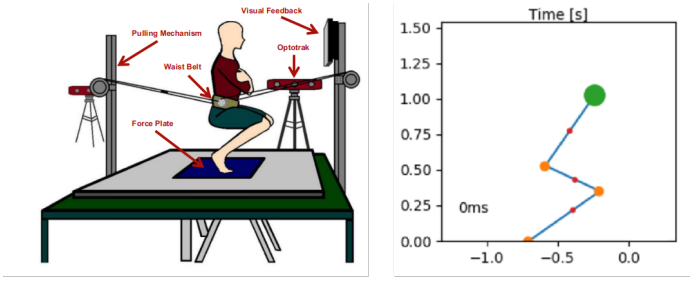

Fig. 1: Left: Initial setup of human experiments, Right: Initial setup of the 3-DOF dynamic system

C. Reward Function

Reward function aims to guide the agent to the goal of stand-up, i.e. bringing the system to the maximum height with minimum torque and minimum final velocity. In case the system reaches 99% of the maximum height with a COM velocity smaller than 0.1 m/s, the episode ends in success condition. If the joint limits are exceeded or the height of the agent drops below 50% of the maximum height, the episode ends in fail condition. Otherwise, the agent receives reward depending on its height. Overall, the reward function is defined as:

$$r_{terminal}(s) = \begin{cases} 1000 - 10 \times t & success \\ -100 - 400 \times h_t & fail \end{cases}$$

$$r_{running}(s,u,w1) = \begin{cases} +1 & h_t >= 0.99 \times h \\ h - w1 \times norm(u) & else \end{cases}$$

where s, u, t, h, h_t, $w1$ denote the state, the action, the current time, the total height, the current height and the cost coefficient of the torque, respectively. The system is provided +1 reward when its height is close to the maximum but its COM velocity is still above 0.1 m/s, in order to narrow down the search space while guiding it towards the success condition. The step reward in our reward function aims to reward the increase in height while punishing the amount of torque used.

D. Inverse Reinforcement Learning via Parameter Search

For our IRL experiments, we have defined our behavior parameter as the trade off of torque usage during the task of standing up. As described in the reward function, we have defined a parameter called $w1$ which indicates this trade off and tried to mimic the expert by searching for the $w1$ which produces the most similar COM trajectory with the expert. The system focuses on a multiple trial approach in order to induce the stochasticity of the inner RL loop. We have used the search loop as given in Algorithm 1

Algorithm 1 IRL via Grid Search

```
weights = [0.1, 0.5, 1.0, 2.0, 4.0, 8.0, 16.0]
for each w1 ∈ weights do
    for trial = 1 to 20 do
        π* = solveRL(w1)
        τ = extractTrajectories(π*)
        for τ_s ∈ subjectTrajectories do
            calculateCost(τ, τ_s)
        end for
    end for
end for
```

E. Inner Reinforcement Learning Loop

For our inner RL solver, we have used the algorithm proposed by [4]. Proximal Policy Optimization(PPO) is an on-policy RL method which works well with continuous state and continuous action based problems. We have decided to use PPO due to ease of implementation and validity of the results presented. As you can observe from Algorithm 1, our RL solver takes $w1$ as a parameter in order to pass this value to the reward function and search for the optimal policy accordingly. Since PPO is also a policy gradient based approach, it is a stochastic system which converges to different near-optimal policies at each iteration. By solving the RL multiple times(20 in our case), we tried to decrease the variance as much as possible which will lead to comparable results.

F. Result Analysis

In order to evaluate our results, we have focused on the variance produced by our inner RL solver. 1000 successful trajectories have been sampled for 140($trial\ count * \#\ of\ weights$) different policies generated during their training phase. We have selected the top 10 trajectories(in terms of reward) from each policy and extracted COM trajectories for each of them. So for each weight, we have generated 20 groups that contain 10 trajectory samples each. In order to narrow down these trajectories into 1, first we have fitted a spline on each of them and sampled same amount of points. Afterwards, we have calculated the average trajectory for each group by taking the mean of the trajectory points. Finally, only a single trajectory was obtained by taking the mean of the group trajectories. During this process, we have also calculated Standard Mean Error(SME) values for torque usage and trajectory areas in order to use them in our inner RL loop analysis.

G. Trajectory Preprocessing

Before conducting the trajectory comparing process, a preprocessing have been done on the the expert and RL trajectories in order to make them comparable. As it was mentioned in the previous sections, our RL setup consists of 3 joints whereas the humans use 4 main joints during the process of standing up. This difference causes the expert COM trajectory to shift towards the back during the process of standing up. In order to resolve this issue, we have rotated both expert and RL trajectories and made sure they both start and end in the same horizontal location. Another preprocessing was done by fitting all vertical coordinates between 0 and 1 so that the heights will be identical and available for comparison. These techniques we have applied allowed us to fairly compare the trajectories generated by RL and experts.

H. Comparing Trajectories

The trajectory of both the RL trials and behavioral data is defined as the COM trajectory they follow while accomplishing this task thus, behavior similarity is defined as COM trajectory similarity. We have implemented 2 different cost metrics in order to compare the trajectory results. In both of these metrics, we have focused on capturing different behavioral similarities between. It should be noted that both of these systems factors into our IRL via Parameter Search method with the *calculateCost* method in Algorithm 1.

1) Signed Area Difference: The first metric we have focused on was based on direct one-to-one correspondence between RL and human trajectories. This metric is also known to be used commonly in trajectory related studies [14] [15]. Since the ultimate goal is inferring reward coefficients while replicating the behavior, this cost system focuses on strict relations. The cost function is formulated as following:

$$area\_difference = \sum_{n=1}^{60} \mid x(n) - y(n) \mid$$

where x and y is the set of sampled horizontal locations from the expert and the RL agent trajectories respectively and n is the order of the point to be calculated.

2) Pearson Product-Moment Correlation: In order to capture some similarity behaviors which cannot be captured by calculating one-to-one difference, we have implemented a second cost function for our IRL system which also takes into

account the shape similarity of the trajectories. We have represented this second cost function suggestion as the following:

$$correlation = \frac{\sum_{i=1}^{n}((x_i - \overline{x})(y_i - \overline{y}))}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

where x and y is the set of sampled horizontal locations from the expert and the RL agent trajectories respectively. Then, we use softmax function over subject rows with a scaling factor in order to rank the correlation results we have achieved in a more understandable visualization.

III. Experimental Results

We have decided to analyze our results in 3 different aspects. First, instead of focusing on the expert data, we will analyze the results obtained from our RL approach in terms of variance and stochasticity. Later, we will analyze the trajectories in terms of signed area difference and finally, we will compare the trajectory correlations between mean human trajectories and our RL trials

A. Analysis of the Inner RL Loop Results

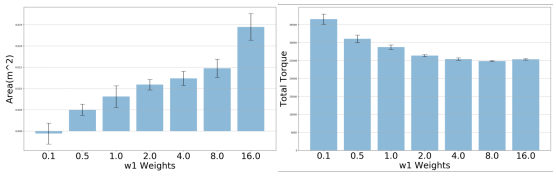

Fig. 2: Analysis of our experiments. Left: Mean Area of Trajectories, Right: Mean Torque Used by the Agent.

Since the algorithms we have conducted our studies on are stochastic, our first aim was to achieve consistency on our experiment results. Our first expectation was that when the torque cost increases, the agent should start to obtain "fatter" trajectories, thus having a greater area. We expect the torque usage to decrease when the torque cost coefficient($w1$) increases. When we observe Figure 2, we observe the trends we have expected occurs in our trials. The mean areas of the trajectories generated increases when the torque cost increases, and torque usage decreases respectively with the increase of w1 torque cost. These results are not enough to fully conclude the validity of our inner RL solver, but they indicate that our technique, repeating 20 trials for each cost, is making the system work in a less stochastic and more consistent way.

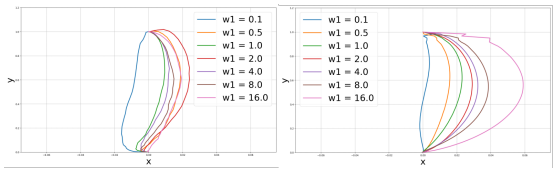

Fig. 3: Left: Trajectories for Top Trials for each W1 value(in terms of reward), Right: Mean of Mean Trajectories Collected From 20 Different Trials

After we conclude the analysis of our inner RL solver, the next step for validating our RL approach is to inspect the trajectories obtained from them. The trend we have observed from Figure 2 suggests that when the torque cost coefficient increases, the area or the *fatness* of the trajectory must also increase. In order to visualize this suggestion, we have plotted the trajectories that performed the "best" in terms of reward in a single sheet and expect them to be ordered in a particular way. When we inspect these trajectories, which can be found in Figure 3, we observe an inconsistency. For example, the best trial from $w1 = 0.5$ was expected to have a smaller area than $w1 = 1.0$ but we can clearly observe that this wasn't the case. Since in Figure 2, the upward trend is present, it is true to say that the average of the successful trajectories will form the expected trend, but the most successful ones don't. This indicates a definite local optima situation in our RL trials and a possible problem with our reward function. It also indicates the crucial need to use the mean results from the trials instead of best results, which is the technique we suggest throughout this study.

B. IRL via Signed Area Difference

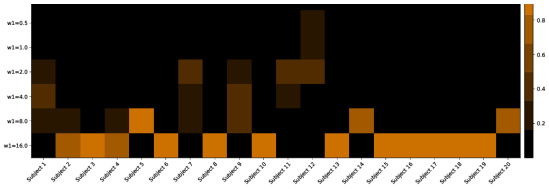

Fig. 4: The comparison between human subjects and RL trials in terms of signed area difference. The outputs have been calculated with softmax function using K=30 as smoothing factor in order to visualize ranking easier

The next step in our experiment evaluation was to compare the trajectories in terms of signed area difference. In order to achieve this, we had to make sure each of our trajectories had the same amount of data points. We have fitted a spline for each trajectory and sampled 60 points from it. Since this cost function focuses on one-to-one correspondence, The heat map on Figure 4 indicates to us that our IRL system was successful in inferring most of the subjects' reward coefficients in an at least near-optimal way.

Our main assumption about the reward function coefficient of the experts was centered on trajectories capturing more and more area, or in other words, getting "fatter" and "fatter" when the torque cost increases so by looking at Figure 4, we can make several deductions. For example, we can say that Subject 6 cares about torque usage more than Subject 12 when he/she is completing our task. Also, we can say that most of the subjects draw a fat trajectory which indicates they focus on limiting their energy usage.

C. IRL via Trajectory Correlation

The final step of our analysis is to compare the trajectories obtained from RL and real trajectories in terms of COM trajectory correlation. Due to small differences between our RL simulation setup and the setup used to collect the behavioral

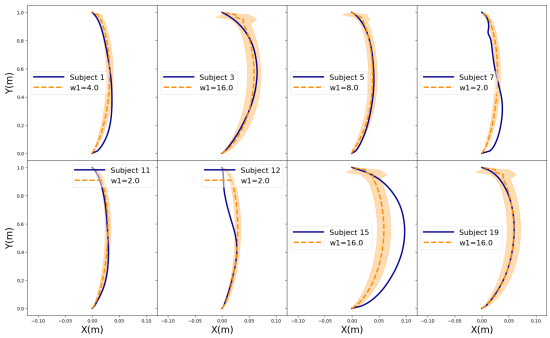

Fig. 5: The visualizations of several subjects and the best fit trajectories we were able to obtain in terms of signed area difference. The shaded area for the RL trials indicate the standard deviation of the trajectories formed. The subjects not given in this figure also produced similar fits.

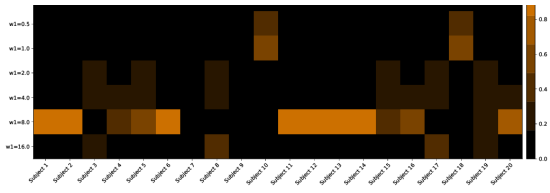

Fig. 6: The comparison between human subjects and RL trials in terms of correlation. The correlation outputs have been sofmaxed over columns with K=100 in order to visualize ranking easier

data, while calculating correlation, we have clipped the first and last 10 data points from our samples. Different from the first analysis we have conducted in the previous section, this cost function focuses more on analyzing the *tactics* humans use during standing up like following a *hip first* or *head first* strategy. When we inspect Figure 6, it can be observed that there are several direct matches found which indicates that the shape of that RL trial trajectory and human trajectory are similar. We also observe that our tests were not very successful in inferring the reward function coefficients of some of the subjects like 7, 9 etc.

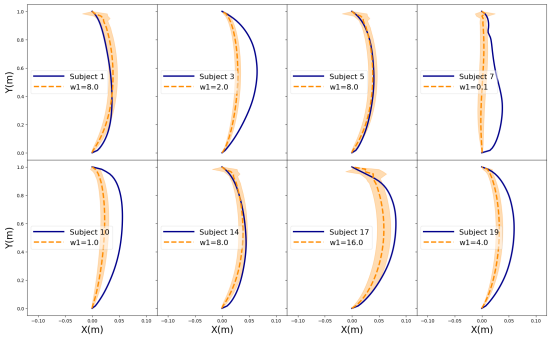

Fig. 7: The visualizations of several subjects and the best fit trajectories we were able to obtain in terms of correlation. The subjects not given in this figure also produced similar fits.

Also, we have visualized some of these trials in Figure 7 and observed the similarities we have expected from Figure 6. Another thing we have observed was that our system was unable to infer the rewards for human trajectories which were very *fat* or *straight* but was successful in inferring more than half of the human subject trajectories in terms of correlation(shape matching). This indicates that these subjects followed a different or a combined strategy that our simulation could not dwell upon.

IV. Discussion

When we analyze the results we have reported in Section III, we observe several key points that needs to be discussed. First of all, in Figure 3, we observe that although the mean trajectories follow a certain trend, the top trajectories obtained for each weight does not and this was the exact behavior we aimed to address while structuring our IRL mechanism. We believe that several reasons cause this behavior. The first one revolves around using a policy gradient method as our inner RL solver. Even with the advancements and modifications suggested by PPO, this method still bounds to reach to a near-optimal solution, which does not fully describe a policy which maximizes the reward function and since the training sessions are stochastic(in other words, end up in a different near-optimal solution each time), the solutions found can produce unexpected behavior with respect to our cost function. Another reason for this behavior can be explained by our cost function itself. In our terminal state *Success*, we have tweaked the terminal reward with a time constraint so that the agent will learn to reach the end point in the fastest way possible. This generates a trade-off since although a fatty trajectory should be more optimal for step reward, the terminal punishment from time consumption can override the importance of it. That's why, further experimentation is needed in order to investigate this issue.

During our experimentation's, we have tried to avoid discretization techniques in our inner RL solver in order for our system to be applicable to real life problems fully and we have applied a grid search technique as parameter-search. We believe that although grid search is not a state-of-art technique, it is not in conflict with the concept of gradient-based outer loop. The solutions found on the grids can be used to seed gradient descent search, which would improve the solution around the grid points. We believe this would be a better approach if we have some prior knowledge on how to choose the discrete grid values. Note that, the task we have used is rather complex, therefore direct gradient descent with an inner RL (which is not guaranteed to find an optimal solution) becomes computationally expensive. One cost of grid search is that, the precision of the ω value is bounded to the precision of the grid. For example, if slices of 0.1 is taken, the system cannot fully capture a subject with the ω value of 0.55.

From both of our comparison functions for our parameter-search, we have seen very descent matches that indicate how much the subjects care about the torque usage and how the subject prefers to stand up. We have seen that the decision of torque usage coefficient not only effects the *fatness* of the trajectory but also the decision on the strategy to use during

standing up. Some of the subject trajectories we have obtained were significantly fatter than the fattest trajectory we have generated, or had a very different shape. This indicates that our weight set for torque cost needs more adjustments in order to capture a wider set of subjects. We also need to point out that these differences can also be caused by the dynamic system properties. Our RL setup was built with single human body data(length, mass, inertia etc.) and subjects can and should perform differently according to their own body. Also, the experiment setup is conducted in a manner that the real life experiments use 4 degrees of freedom but our RL setup only includes 3. This difference, which can be explained as model's point-foot vs. flat contact of human feet may also cause the discrepancies described in terms of area and shape.

Another thing we want to emphasize is the optimality basis for RL trials and expert trials. Even if the dynamic system is completely replicated and an inner RL loop is found that reaches to a global optima point, IRL systems still have no guarantee of perfectly inferring the reward function coefficients of the experts. This is due to the fact that it is unclear if the expert data collected is optimal according to the reward function followed by the subjects.

V. Conclusion and Future Work

In this study, we have proposed a parameter-search based IRL model which takes the variance of the inner RL loop into account in order to tackle the stochasticity problem of policy gradient RL methods and tried to infer the reward function coefficients for real life experts during motor control related tasks. We believe that constructing a parameter-search based IRL method on top of a grid is important, since it has the capability to serve as a baseline solver for more advanced IRL methods which will be applicable to real life problems. We also believe that the findings and the methodologies we have presented in terms of handling the variance generated by the policy-gradient based RL loops are meaningful, and is a step forward towards a robust IRL method that will be applicable to continuous state, continuous action based environments. Our findings suggest that for generating a robust IRL method that uses policy gradient methodologies as a RL solver, the converge properties must be studied in depth so that the final IRL method can be regarded as a trustworthy system.

For future studies, we plan to address the issues presented on Section IV including reward function modification, trying to increase the determinism of RL trials and different dynamics systems special to each subject. Also, we plan to dwell more upon the idea of combining our grid with gradient based approaches in order to create more precision on the predicted ω value. With these additions, we believe that our IRL system will have the ability to serve as a computational tool for inferring the reward parameters of expert behavior in a state-of-art manner.

Finally, in order to demonstrate our approach in a real world environment, we plan to employ the reward parameters we have extracted during our simulations to different types of humanoid robots and try to match the trajectories we have generated with RL setups and the expert data with the trajectories humanoid robots generate during standing up.

ACKNOWLEDGMENT

The full body movement work is supported by the Slovenia/ARRS - Turkey/TUBITAK bilateral collaboration grant (ARRS Project no: BI-TR/16-18-001; TUBITAK Project no:215E271). Additional partial support for the work presented in the paper was given by the Bogazici Resarch Fund (BAP) IMAGINE-COG++ Project with project no. 18A01P5

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14236

[2] S. Gu, T. P. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep q-learning with model-based acceleration," in *ICML*, 2016.

[3] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *CoRR*, vol. abs/1509.02971, 2015.

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

[5] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 4565–4573.

[6] S. Levine and P. Abbeel, "Learning neural network policies with guided policy search under unknown dynamics," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 1071–1079.

[7] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *ICML*, 2016.

[8] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *AAAI*, 2008.

[9] M. Wulfmeier, P. Ondruska, and I. Posner, "Maximum entropy deep inverse reinforcement learning," 2015.

[10] E. Uchibe, "Model-free deep inverse reinforcement learning by logistic regression," *Neural Processing Letters*, vol. 47, no. 3, pp. 891–905, Jun 2018.

[11] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," in *ICML '12: Proceedings of the 29th International Conference on Machine Learning*, 2012.

[12] D. Clever and K. Mombaur, "An inverse optimal control approach for the transfer of human walking motions in constrained environment to humanoid robots," 06 2016.

[13] G. Gede, D. Peterson, A. Nanjangud, J. Moore, and M. Hubbard, "Constrained multibody dynamics with python: From symbolic equation generation to publication," in *ASME*, 2013.

[14] J. Babic, E. Oztop, and M. Kawato, "Human motor adaptation in whole body motion," *Scientific Reports*, vol. 6, pp. 32 868 EP –, Sep 2016, article. [Online]. Available: https://doi.org/10.1038/srep32868

[15] M. Bucklin, M. Wu, G. Brown, and K. Gordon, "Adaptive motor planning of center-of-mass trajectory during goal-directed walking in novel environments," *Journal of Biomechanics*, 08 2019.