Towards Long-horizon Motion Planning in Dynamic Environments

Xiaoliang Zhang[0009−0008−9111−1828], Alan G. Millard[0000−0002−4424−5953], and Pengcheng Liu[0000−0003−0677−4421]

Department of Computer Science, University of York, York, UK
{xiaoliang.zhang, alan.millard, pengcheng.liu}@york.ac.uk

Abstract. In contrast to mobile robots, the planning space of manipulators (robot arms) is high-dimensional, which poses great challenges for motion planning. Classical motion planning approaches face computation difficulty in such scenarios, and dynamic obstacles exacerbate the problem. We propose a Deep Reinforcement Learning (DRL) based motion planning approach, with a Gaussian Mixture Regression (GMR) sampling tool, to tackle this problem. Benefiting from the reduction of training difficulty in DRL networks, the proposed architecture is effective for a long-horizon manipulation tasks in dynamic environments.

Keywords: Motion planning · Manipulators · Deep Reinforcement Learning · Gaussian Mixture Regression · Dynamic environments.

1 Introduction

Motion planning (MP) for manipulators with dynamic constraints is a challenging problem, but essential for safe human-robot collaboration (sHRC), where robots must avoid human bodies (dynamic constraints) in real-time. Sampling-based motion planners (SBMP) are often used for MP in high-dimensional space. However, SBMP still faces limitations in very high-dimensional space [1], and faces difficulty in processing dynamic environments [2] due to its high computational cost. On the other hand, Deep Reinforcement Learning (DRL) has been widely used in MP, and model-free DRL is especially powerful in unknown environments. Unfortuantely, DRL methods face difficulty in long-horizon manipulation tasks due to the extremely large search space to explore [3], especially when there is a sparse reward. Gaussian Mixture Models (GMM) are simple to implement, and take uncertainty in the environment into consideration, but they are not good at avoiding geometric constraints [4].

In this paper, we propose a hybrid MP model for manipulators in dynamic environments with GMM/GMR (Gaussian Mixture Regression) and DRL. Concretely, a GMM/GMR is trained from demonstrations of an expert in a certain task, then it is used as an aiding tool to train a DRL model. In the training of DRL model, the GMM/GMR is applied to bias the training of DRL, thus reducing its searching space to help it converge.

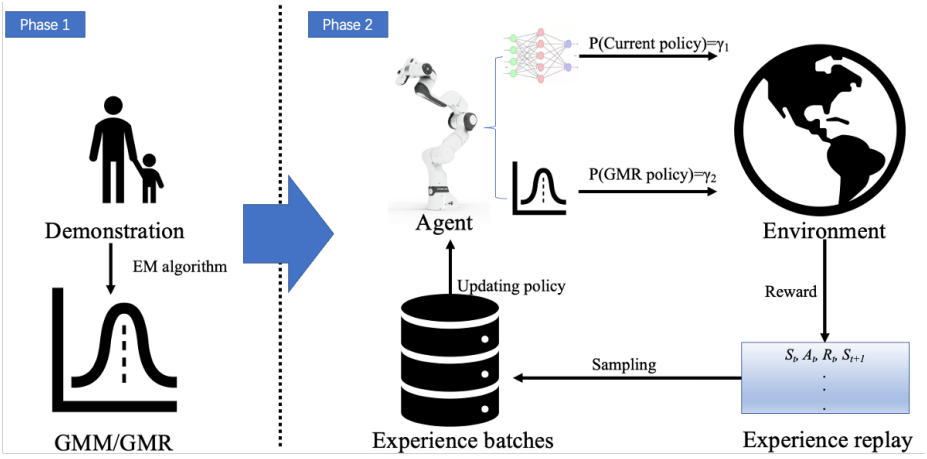

Fig. 1. Architecture of the Proposed Model

The advantages of this work include: (1) by the help of a trained DRL, it can respond to a dynamic environment in real-time. (2) GMR can help train the DRL, by reducing the search space and decreasing its training difficulties, thus for long-horizon tasks that pose challenges in the training of DRL, our proposed method is more powerful. (3) it can process the MP in high-dimensional space, as DRL is well-suited to high-dimensional space.

2 Methodology

We propose a hybrid model for a long-horizon manipulator MP with dynamic constraints, consisting of DRL and GMR. The GMM/GMR, trained by demonstration from humans, functions as a exploration guiding tool for the following DRL-based MP model. The DRL-based MP model can generate motions at the next time step based on the current state, and goal configuration. Instead of exploring the space just by its own policy, the exploration is biased towards imitating the behavior of the GMM/GMR, the search space is thus reduced. The architecture of our proposed model is shown in Figure 1.

2.1 The GMM/GMR Exploration Guiding Tool

We develop a GMM/GMR model similar to [4], which maps time to state. To train the GMM/GMR, we first plan to collect N demonstrations from human experts, each of which will be abstracted into T-time steps. GMM/GMR is the mixture of several Gaussian models (Gaussian components) – in our model, we have K Gaussian components, so:

$$P(x \mid \theta) = \sum_{k=1}^{K} \alpha_k \phi(x \mid \theta_k) \tag{1}$$

where θ_k refers to the mean and covariance matrix of the k-th Gaussian component, i.e., (μ_k, Σ_k). And $\phi(\xi \mid \theta_k)$ is the k-th Gaussian component base. $x = (x_i), i = 1, 2...N$ is the demonstrations. By introducing time t, we have:

$$\begin{aligned} P(x \mid t, k) &\sim \mathcal{N}(x; \hat{x_k}, \hat{\Sigma_k}), \\ \hat{x_k} &= \mu_{x,k} + \Sigma_{xt,k}(\Sigma_{tt,k})^{-1}(t - \mu_{t,k}), \\ \hat{\Sigma_{xx,k}} &= \Sigma_{xx,k} - \Sigma_{xt,k}(\Sigma_{tt,k})^{-1}\Sigma_{tx,k} \end{aligned} \tag{2}$$

Through an Expectation-Maximisation (E-M) algorithm, we can train the GMR model and finally the GMR-based exploration tool. This will be implemented with the DRL-based motion planner introduced in the next subsection.

2.2 GMR-guided Soft Actor-Critic

After we get $P(x|t)$, we begin to design the DRL-based planner. We plan to apply the Soft Actor-Critic (SAC) model to train our DRL network, due to its strong adaptability to continuous space and outstanding capability in convergence. To reduce the search space, we make an improvement in the exploration process of the SAC. Typically, the SAC agent will execute actions based on its current policy to collect transitions from environment. Instead, we want the agent to perform exploration with probability γ_1, or follow GMR with probability γ_2 for the whole episode exploration at the beginning of each one, which could reduce the search space. We make γ_1 and γ_2 dynamic, as we want the agent to follow the GMR more often in the beginning, since at this time the GMR model is more experienced. Whereas with the improvement of the agent's policy, the weight explores on its own policy more. This means that γ_1 will increase with training and γ_2 will decrease.

We name our SAC-based planner with GMR as GMR-guided SAC (GSAC), and after it is trained offline, it can be set up for online usage, thus saving computation time when used, compared with SBMP. To make it compliant with dynamic obstacle avoidance, we need to design its reward function carefully like in [5]. Anther concern is the inefficiency brought by sparse reward in a high-dimensional space. We plan to use a composition of dense rewards to provide the agent with timely feedback in case it gets lost in the large space. A typical approach is to give the agent a small value in every step, to query whether it is in collision or reach the goal. And once it happens that the agent collides with obstacles or reaches the goal, a large reward can be given as a inspiration or a punishment. We will design our reward function based on the above considerations, and make improvement to make our model better. The architecture of a SAC is rather complicated, which consists of 4 networks, one policy network, one state value network, and two Q networks. We will implement these four networks and their objective functions in detail in the next steps.

We will conduct the experiments on a Franka Panda robotic arm, a robot with 7 DOFs. The action is represented by the deviation of each joint, $a_t = \Delta q_t$.

There should be an upper limit set on this deviation to avoid dealing damage to the robot. To make the agent collect enough information about the environment, we make state $s_t = (q_t, q_e, d_{obs}, d_{tar})$, where q_t represents the coordinate of each joint, q_e is the position of the end-effector, d_{obs} is the distance between each link of the robot to the obstacles and d_{tar} is the distance of the end-effector to the target. However, the state and action space is still under evaluation and will be further improved in the future based on the experiment results.

Our proposed work is built on the backbone of DRL-based MP in dynamic environments, which is already verified in many past works like in [5] [6]. Based on these works, we introduce a sampling method to enhance its performance, thus it is plausible that our proposed model will perform well in the experiment.

3 Conclusions

In this paper, we introduced a DRL-based MP model to solve a long-horizon manipulation task in dynamic environments, with the help of an innovative GMM/GMR-based exploration strategy. Previous works using DRL-based model for manipulation in dynamic environments demonstrated effectiveness and thus we can conclude that our work will be more effective. We will move on to improve our design and implement the experiment. Moreover, this model also has the potential to be integrated with the ISO requirements for the sHRC.

References

1. Ichter, B., Pavone, M.: Robot motion planning in learned latent spaces. IEEE Robotics and Automation Letters **4**(3), 2407–2414 (2019)
2. Yamada, J., Lee, Y., Salhotra, G., Pertsch, K., Pflueger, M., Sukhatme, G., Lim, J., Englert, P.: Motion planner augmented reinforcement learning for robot manipulation in obstructed environments. In: Conference on Robot Learning. pp. 589–603. PMLR (2021)
3. Nasiriany, S., Liu, H., Zhu, Y.: Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 7477–7484. IEEE (2022)
4. Wang, J., Li, T., Li, B., Meng, M.Q.H.: Gmr-rrt*: Sampling-based path planning using gaussian mixture regression. IEEE Transactions on Intelligent Vehicles **7**(3), 690–700 (2022)
5. Chen, P., Pei, J., Lu, W., Li, M.: A deep reinforcement learning based method for real-time path planning and dynamic obstacle avoidance. Neurocomputing **497**, 64–75 (2022)
6. Wang, Y., Kasaei, H.: Obstacle avoidance for robotic manipulator in joint space via improved proximal policy optimization. arXiv preprint arXiv:2210.00803 (2022)