Autonomous Robotic Grasping

BTech Final Year Project

Sri Aditya Deevi

ECE (Avionics)

Indian Institute of Space Science and Technology

Advisor : Dr. Deepak Mishra

Two ARG tasks are mainly considered, critical for developing “intelligent” robotics :

Task I : Grasping Various Objects in Diverse Environments

Aims to use Deep Reinforcement Learning techniques to train a robotic arm to grasp novel objects, i.e. objects whose 3D model is not known apriori, in novel random scenes.

Task II : Dynamic Grasping of Moving Objects

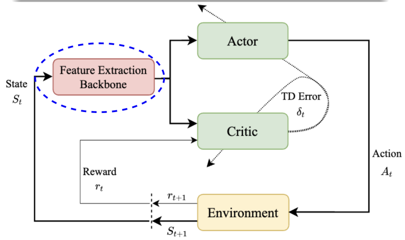
Aims to use Deep Learning and Inverse Kinematics based Motion Planning techniques to train a robotic arm to grasp dynamic objects of interest, whose 3D model is known apriori but motion trajectory is not known.

Also, basic tasks such as *Hand-Eye Calibration* and *Blind Pick and Place* that are necessary for performing ARG tasks in a “real” robotic setup are explored.

Methodology

- The need for developing effective techniques for skilled robotic manipulation tasks, from repetitive industry processing activities to household grasping chores, was the motivation for this project.
- As a part of both tasks, a series of effective Deep Learning (DL) algorithms were developed for accomplishing various subtasks embedded in the main task.

Task I



- Design and development of a series of octree-based convolutional neural network models for effective feature extraction in an Actor-Critic setting was done.
- The model development was focused to demonstrate the need for developing and incorporating complex DL models in DRL algorithms.

Task II

Algorithm 1: Dynamic Grasping Algorithm

```
Function DynamicGrasp(O)
  G_DB ← RetrieveGraspDatabase(O);
  while True do
    p_c ← RetrieveObjectPose(O);
    t_pred ← CalcPredTime(d);
    p_f ← PredictObjectPose(O,t_pred);
    G_W ← ConvertGrasps(G_DB, p_f);
    G_F ← FilterGrasps(G_W, p_f);            /* Using Ranking Functions */
    g_c ← ChooseGrasp(G_F);
    If this chosen grasp is unreachable, continue to next iteration;
    Move arm to g_c;
    if ObjectGrasp() then
      p'_c ← RetrieveObjectPose(O);
      t'_pred ← 1s;
      p'_f ← PredictObjectPose(O,t'_pred);
      g_c ← ConvertGrasps(G_DB, p'_f);
      Move arm to g_c;
      Close hand while moving with the target for t'';
      Break Loop;
    end
  end
  return CheckGraspSuccess();
end
```

- Demonstration of a Dynamic Grasping pipeline, where a robotic arm was deployed in a simulation scene to grasp dynamic objects.
- Various object pose prediction algorithms, both standard and DL methods, were designed and incorporated into the pipeline leading to improved performance.

Notable Results

Task I

The test setup consists of evaluating the robot's performance and the learnt policy in 200 episodes with novel scenes and novel objects.

Robot	Panda	Panda	UR5 with RG2	UR5 with RG2
FE Arch.	Vanilla O-CNN	Residual O-CNN ($n = 2$)	Residual O-CNN ($n = 4$)	O-AHRNet
# Parameters	0.6795M	0.5675M	0.6348M	2.6614M
Success Rate	45.5%	45.5%	80.5%	**87.5%**
Mean Reward	188.42 ± 191.20	185.955 ± 194.76	319.37 ± 152.32	$\mathbf{349.34 \pm 123.41}$
Mean Episode Length	65.71 ± 39.38	63.965 ± 40.615	39.42 ± 34.70	$\mathbf{29.57 \pm 31.03}$
Mean Successful Episode Length	24.78 ± 19.725	20.835 ± 14.54	25.71 ± 21.95	$\mathbf{19.50 \pm 17.04}$

Task II

The test setup consists of evaluating the robot's performance and dynamic grasping algorithm in 100 trials for each of the object and each type of motion.

Method \ Type of Motion	Linear	Linear with Obstacles	Linear with Top Shelf	Sinusoidal
Kalman Filter	**85.8**	82.8	46.6	10.8
	13.7016	**14.6598**	25.0018	24.0029
Multi Layer Perceptron (MLP)	85	81.2	46.2	74.8
	14.7262	15.0821	25.0037	18.0111
Long Short Term Memory Network (LSTM)	85.4	**84.4**	**46.2**	**75.8**
	13.9090	14.7954	**24.5845**	**17.8904**

■ Average Success Rate || ■ Average Dynamic Grasping Time

Summary and Conclusion

Extensive experimentation, comparison, and inference based on obtained results were undertaken for all the developed methods in both the tasks.

Task I : Grasping Various Objects in Diverse Environments

✓ It was shown that in addition to improving the algorithm side of DRL techniques, it might be beneficial to do develop advanced neural network architectures.
✓ Considering the Feature Extractor, the best performing model was the *O-AHRNet* model, which used repeated multi-depth octree fusion features (to maintain high-resolution representations) and an attention module consisting of the channel and spatial attention.

Task II : Dynamic Grasping of Moving Objects

✓ It was discovered that the usage of DL models could prove useful for predicting the motion of dynamic known objects.
✓ Considering the improvements made in the Object Pose Prediction component, the best performing model for objects moving in Sinusoidal non-linear trajectories was the *Long Short Term Memory (LSTM)* model, known to model complex non-linear interactions in sequence to sequence problems.

Furthermore, tasks such as *Hand-Eye Calibration & Blind Pick and Place*, involved in setting up a real-world robotic grasping system, motivated the use case of having a perception-based intelligent grasping system for performing skilled manipulation tasks.