Proposal

Yaobin Liang

committee

D. PAUL BENJAMIN

DAMIAN M. LYONS

FRANCIS PARISI

Agenda

- Background
- Related work
- Approach of this study

High-dimensional Spaces Motion Planning for Robotic Arm in Dynamic Environment

- Robotic Lab in Pace: 2 Kinova Gen3 arms

Franka Panda

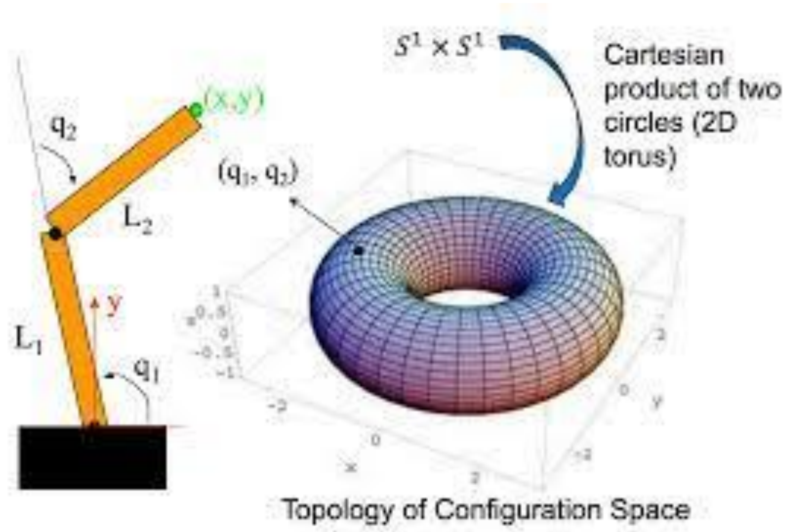
Kinova Gen3

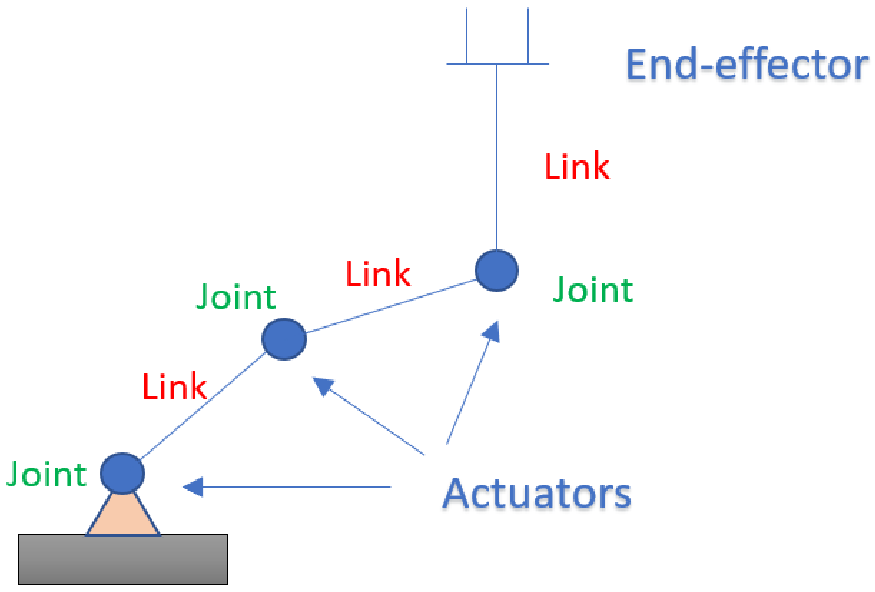
UR10e

KUKA LBR IIWA 7 R800

Basic Concepts

- Link, Joint and End-effector
- Configuration, Configuration Space (C-Space)
- Degrees of Freedom (DOF)





Piano Mover's Problem

Classical Problem Formulation

The aim of motion planning: find the collision-free path $\sigma\colon [0, T] \longrightarrow \mathcal{C}_{free}$

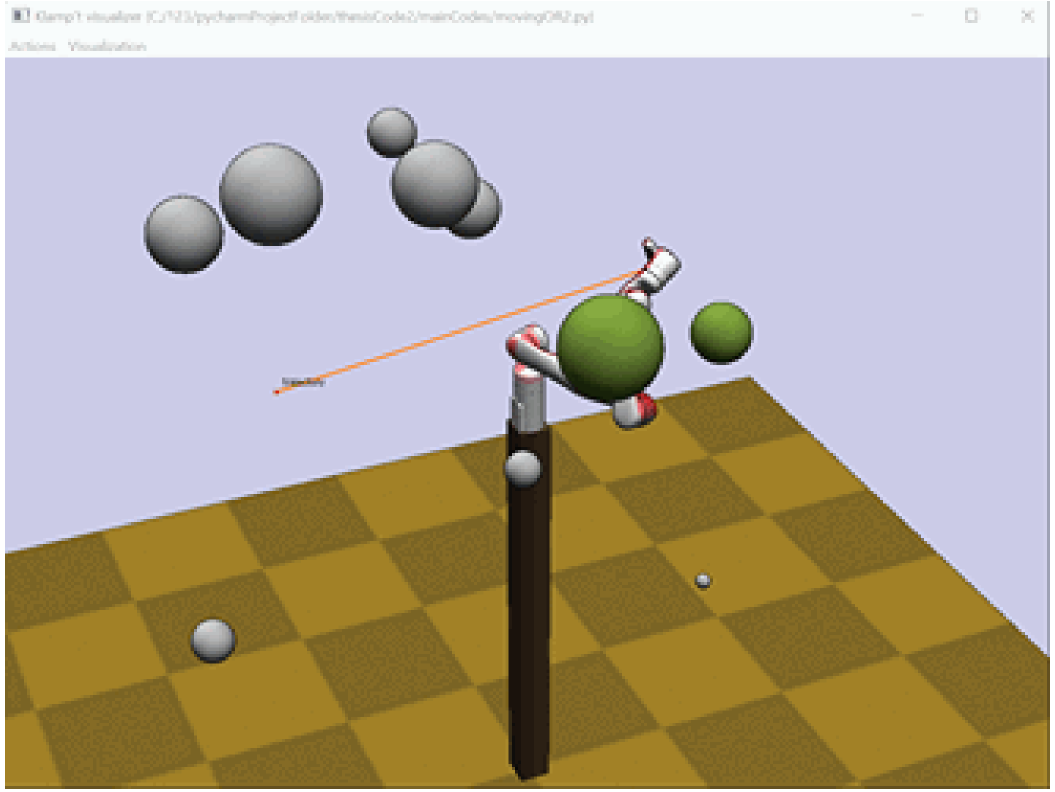
such that $\sigma(0) = q_{init}$ and $\sigma(T) = q_{goal}$.

Problem

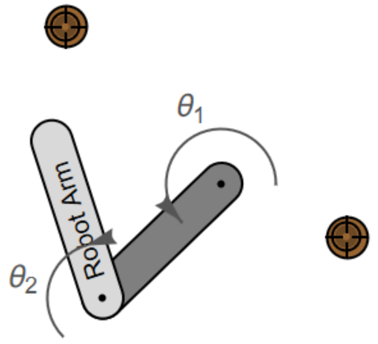
- Static VS Dynamic environments
- Velocity info is important

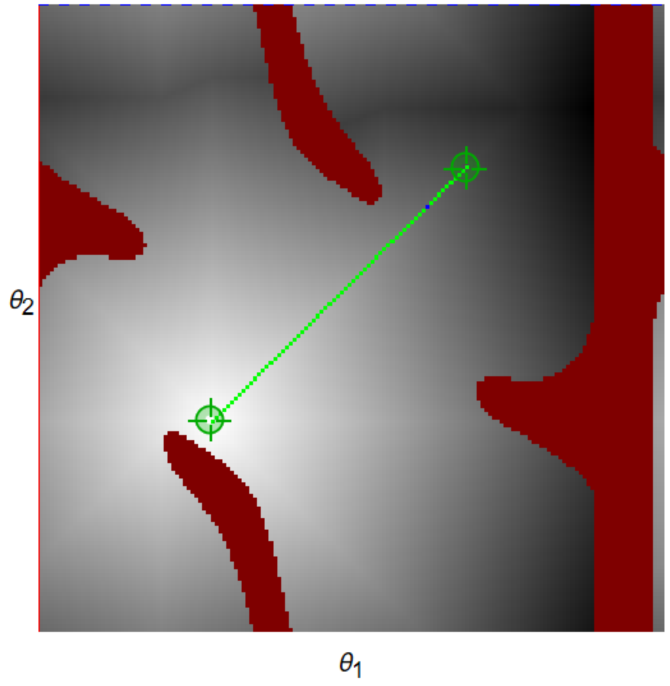
Most manipulators are predetermined motion trajectories, non-reactive

plan their motion or path on the fly



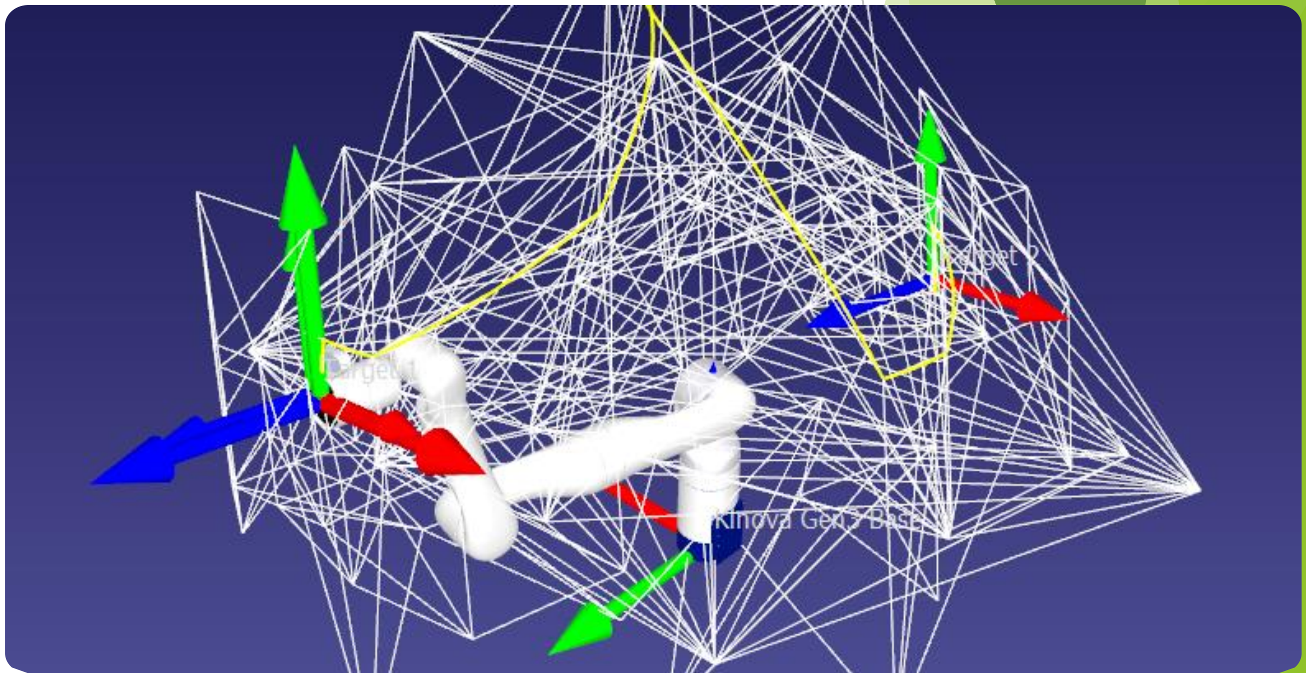
Related Work





- Grid-based
- Sampling-based
- Machine Learning based

Limitation: low DOF, static



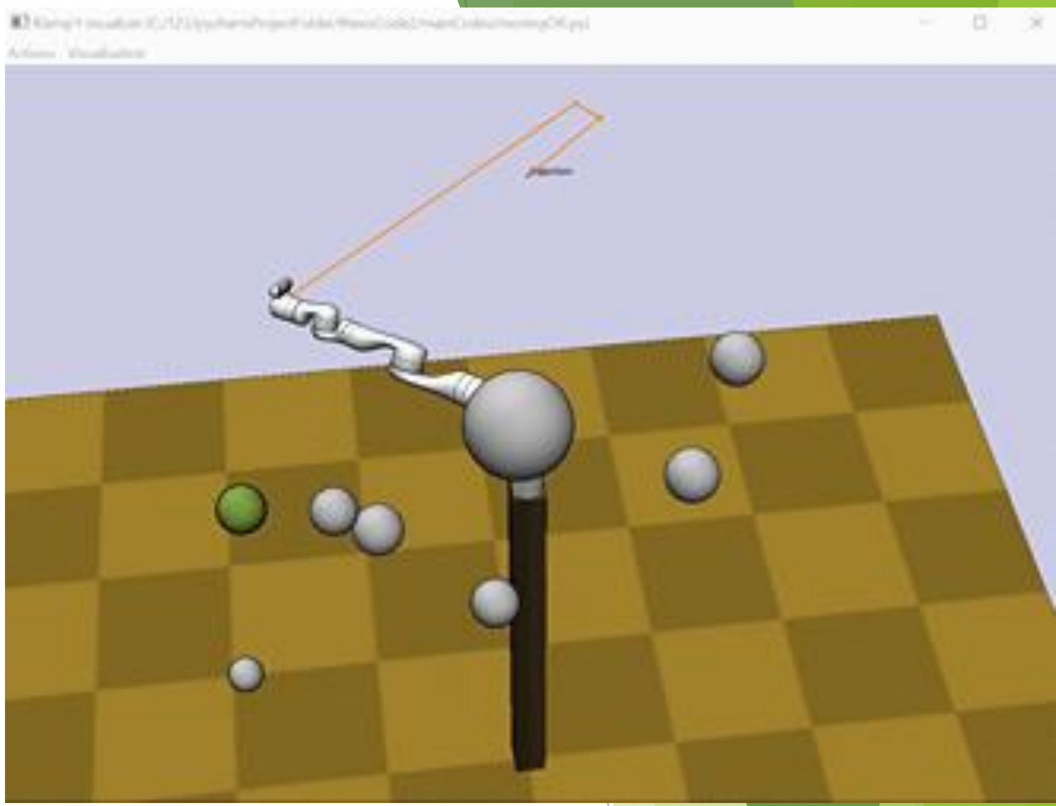
Limitation: velocity

Approach of This Study

Why Supervised learning :

- Require fast reaction
- Successful rate

Supervised learning



Supervised learning is to learn the mapping function from input to output:

$$Y=f(X)$$

Input:

1. Current and previous Configurations
2. Current and previous Obstacles Info
3. Goal Configuration
4. Constant

→

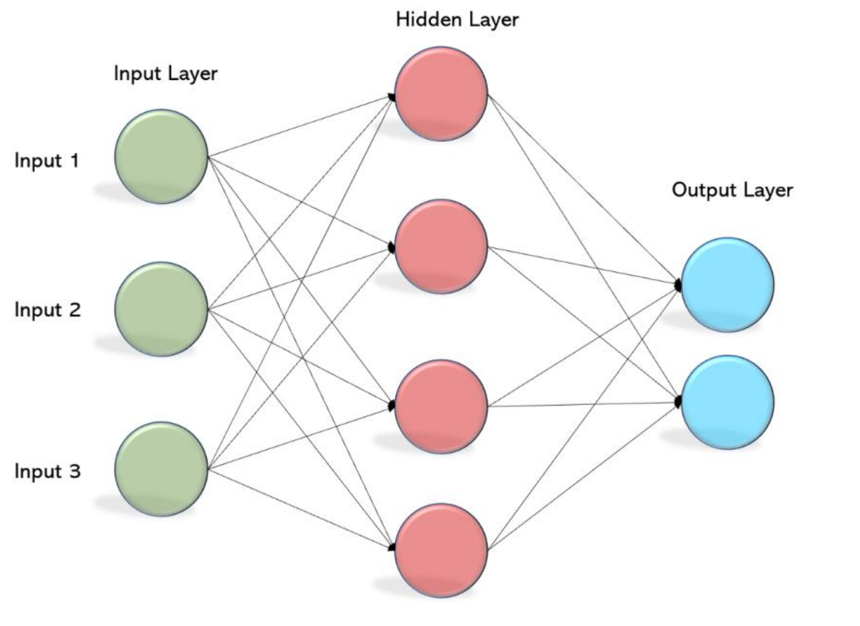
1. Denavit-Hartenberg(DH) Parameters
2. Moment of Inertia

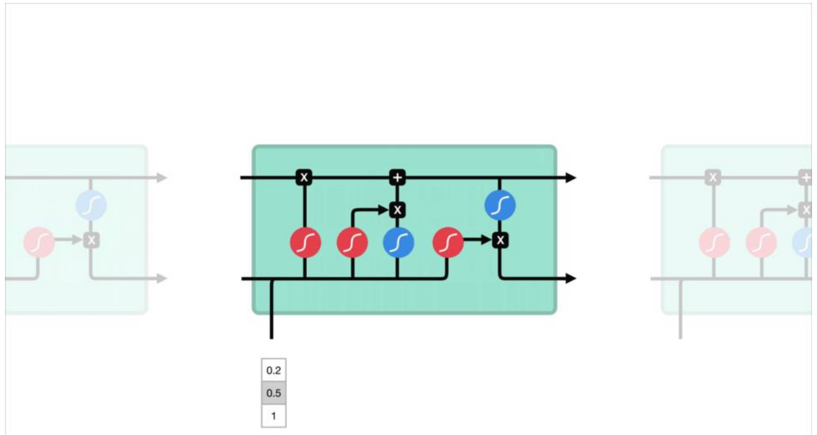
Output:

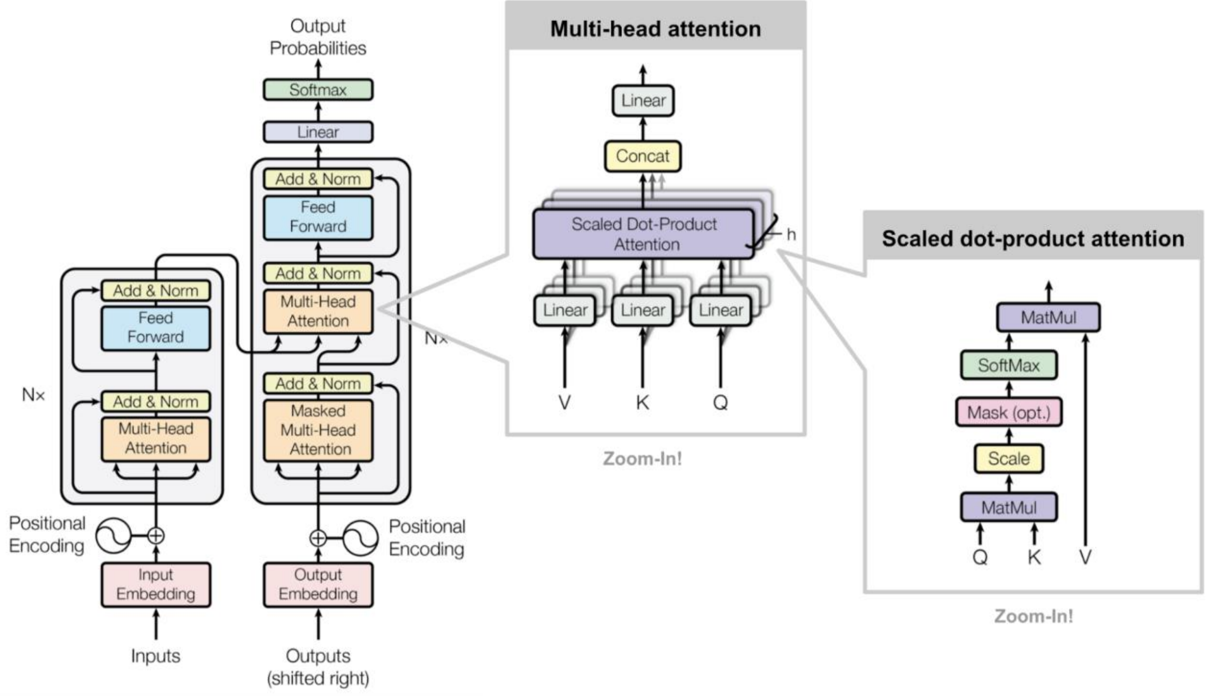
Next Configuration

Deep learning structure

- MLP-based
- LSTM-based
- Transformer-based







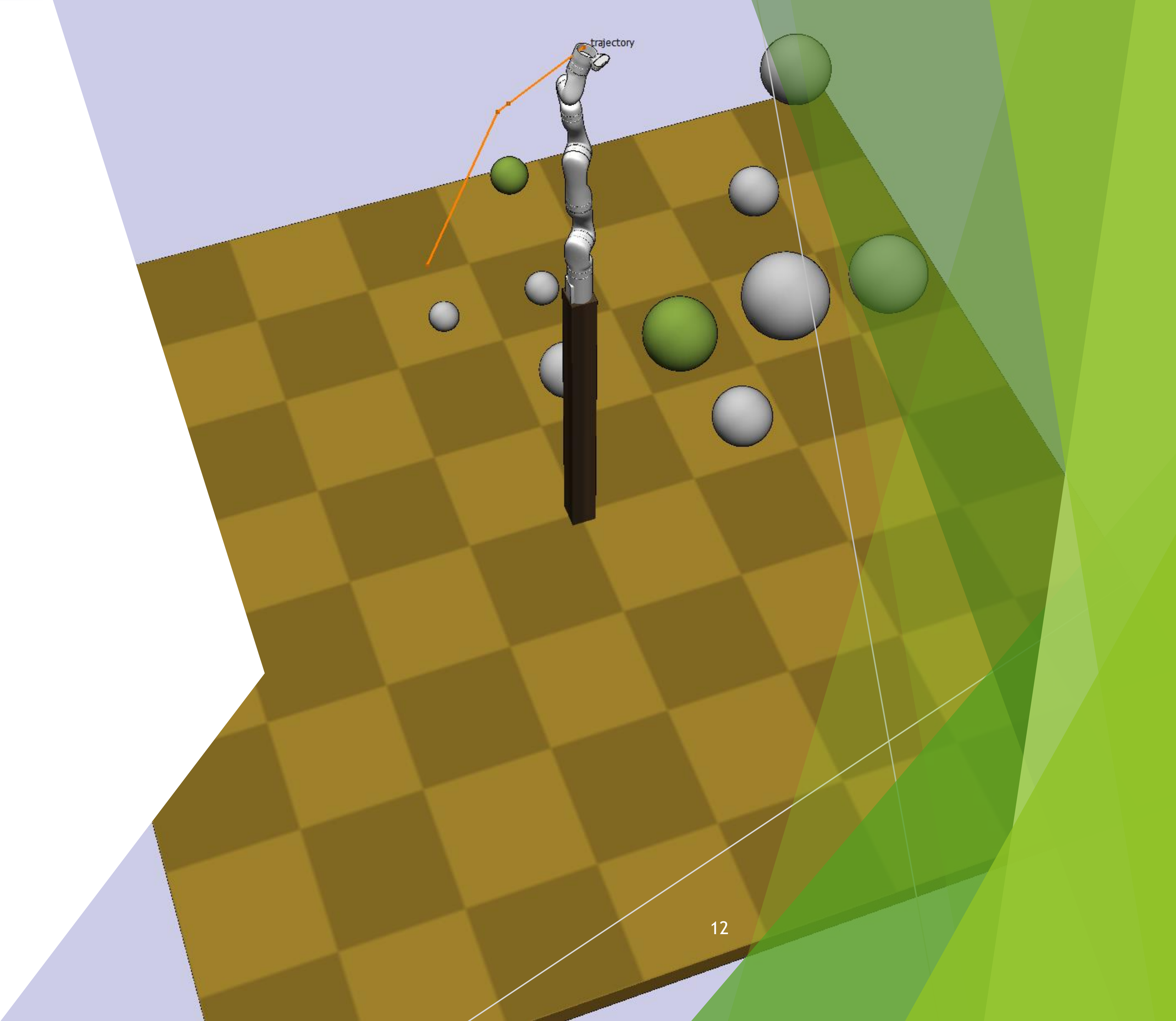
Data Acquisition

Why? -> better performance

How to generate training data?

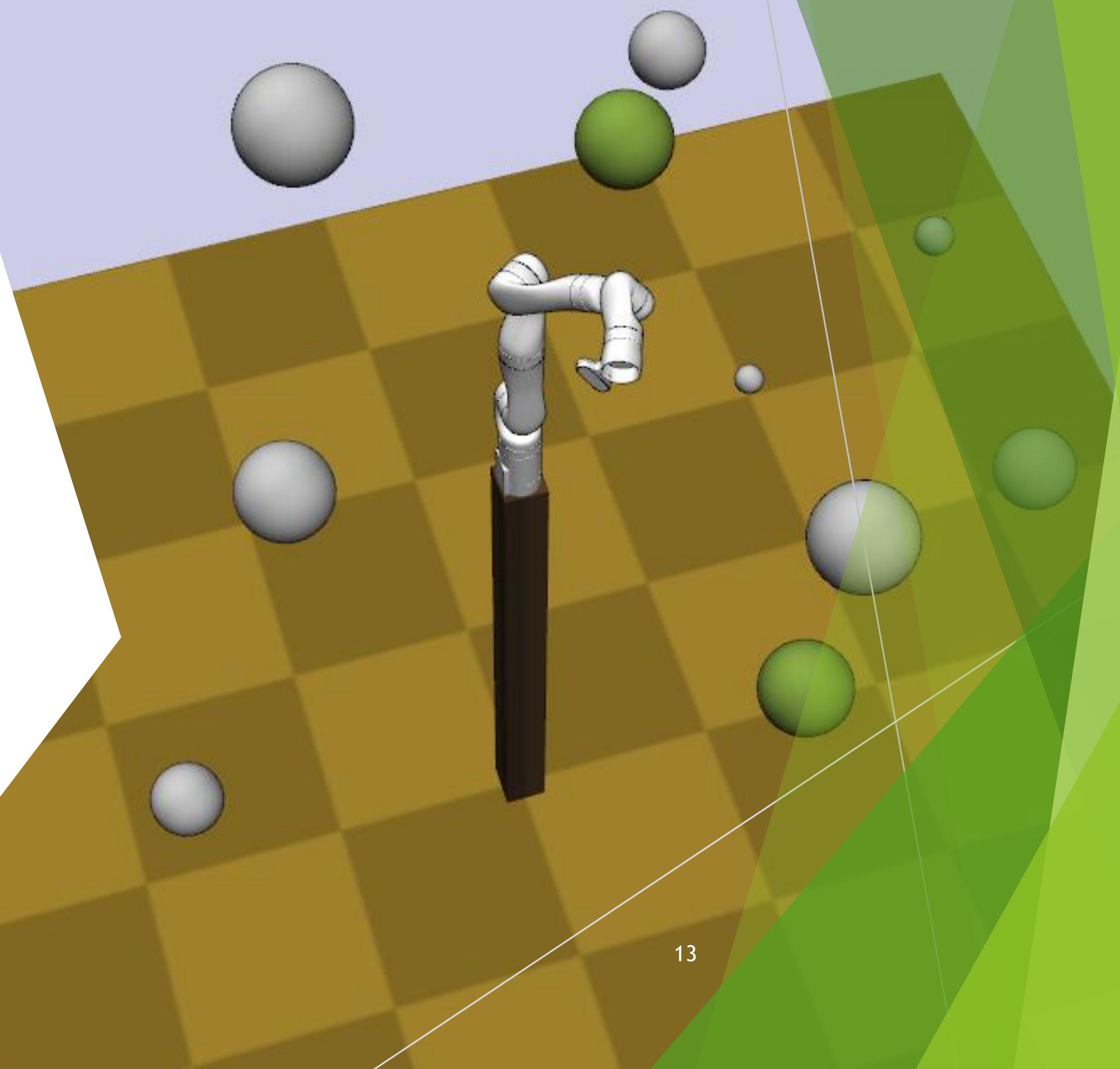
- Static

RRT* algorithm:
asymptotically optimal



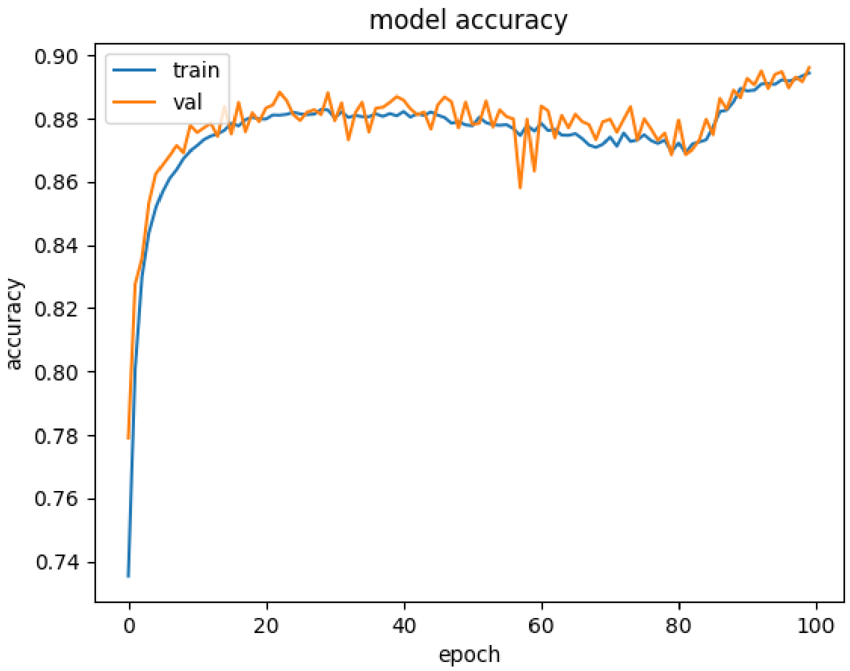
Detail of Generating Data

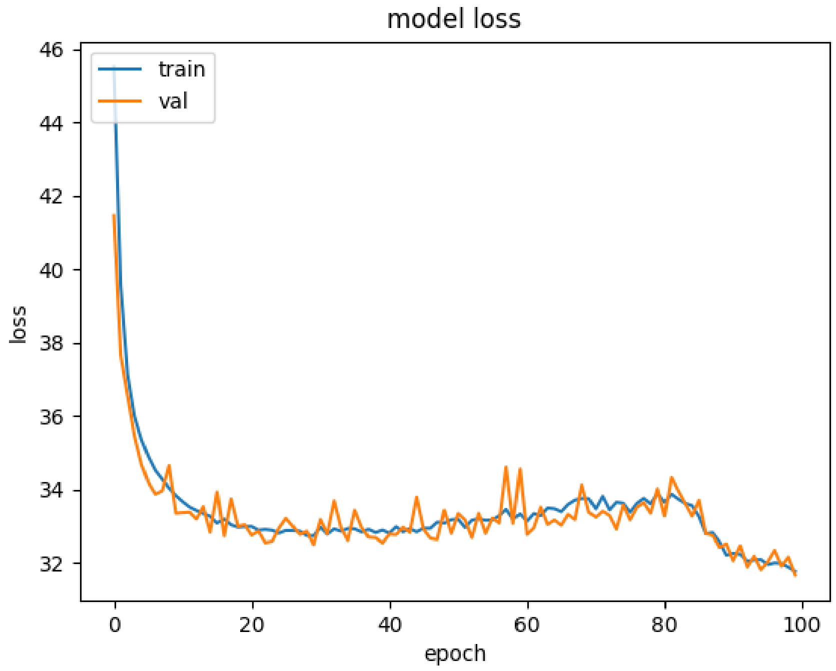
- Generate environments store as .xml.
- 100 different environments each with different obstacles
- Each has 3000 paths(trajectories) with different start and goal configurations
- Total of 300,000 data paths
- Save paths data as .configs



Training parameters for DNN

- backpropagation algorithm
- loss='mae' or 'mse' , optimizer='adam'
- batch_size=500-5000 ,epochs=300-3000





Evaluation

SUCCESS RATE

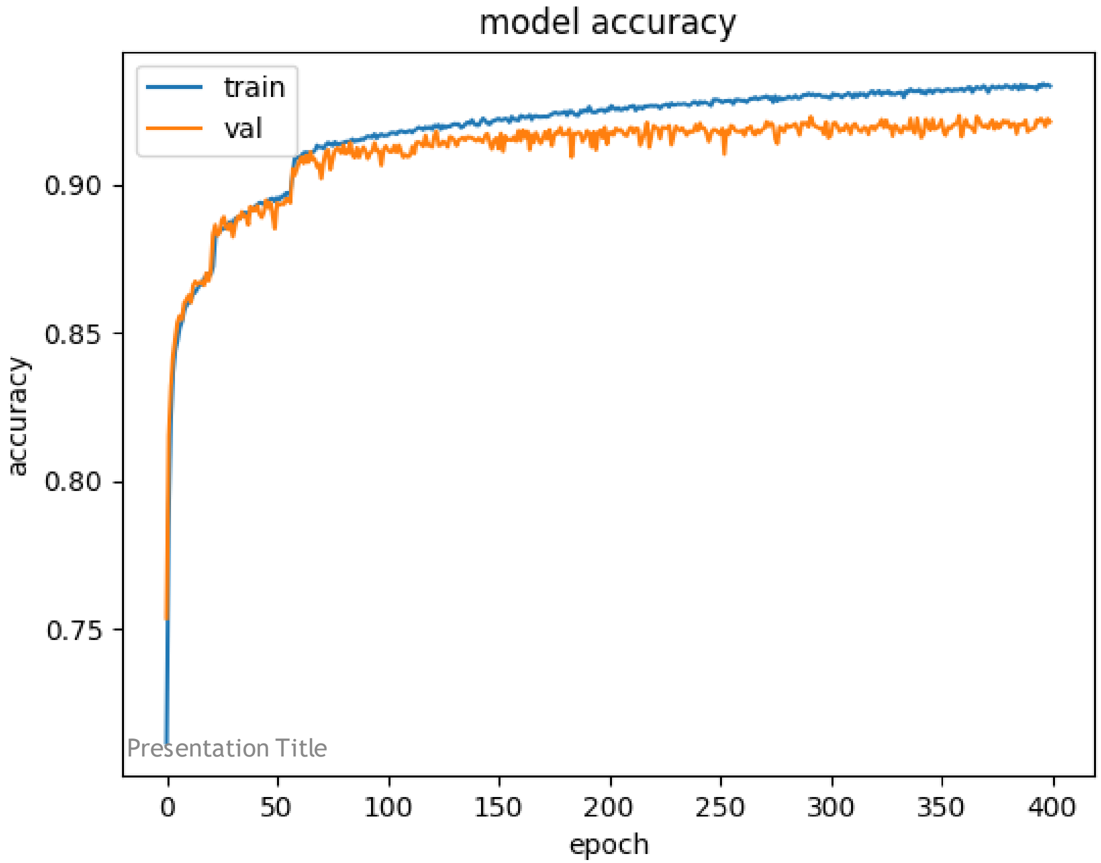
PLANNING TIME

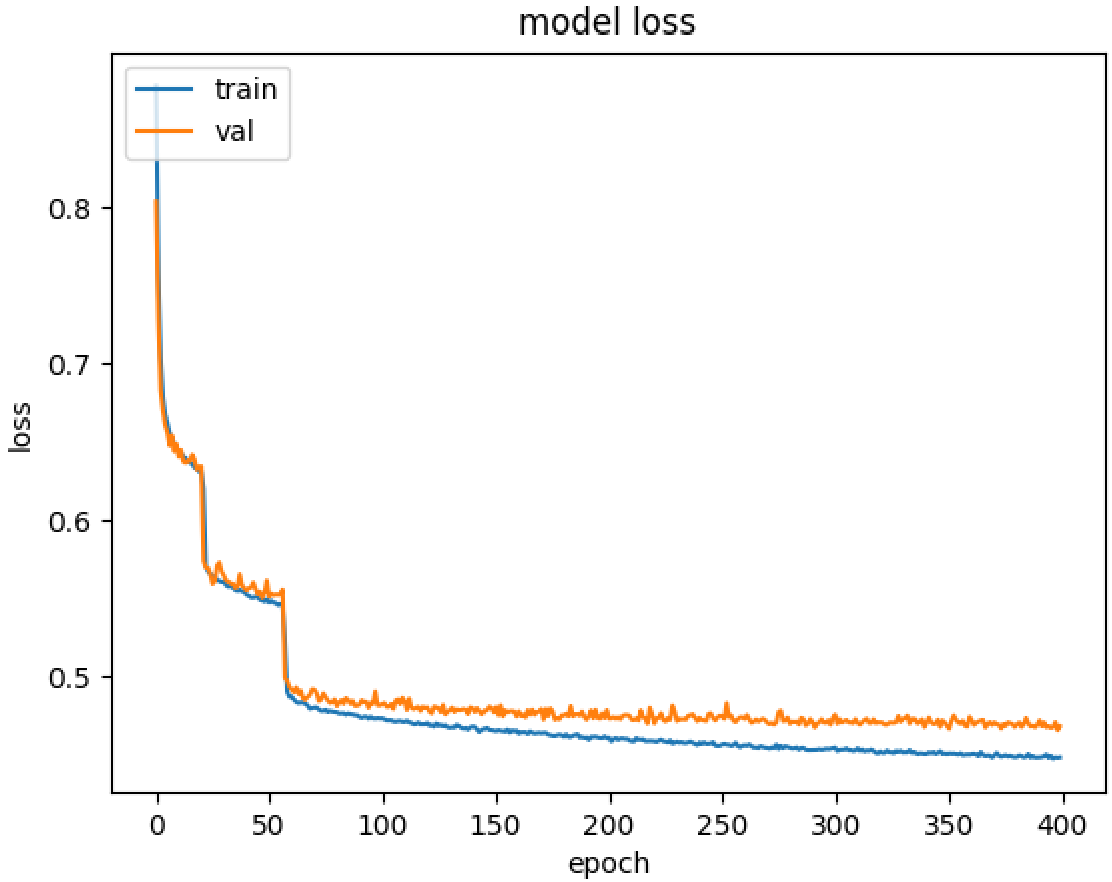
Software and Packages

- Klampt – Simulation, data generation
- Tensorflow – Deep learning

Result

Train my MPL-based DNN:
-about 92% accuracy for predicting the next configuration
-2% better if we take the Constants as input





Thank you

Thanks to Nishark Singhal for implementing the simulation environment

Reference

https://planners-benchmarking.readthedocs.io/en/latest/

https://twitter.com/atsushi_twi/status/1147403214477598720

https://www.google.com/url?sa=i&url=https%3A%2F%2Fnatanaso.github.io%2Fece276b2018%2Fref%2FECE276B_5_ConfigurationSpace.pdf&psig=AOvVaw0zVYCELq7ct8T8aDY18Ll7&ust=1664760141651000&source=images&cd=vfe&ved=0CAwQjRxqFwoTCJD2tvmwwPoCFQAAAAAdAAAAABAD

https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/

https://becominghuman.ai/multi-layer-perceptron-mlp-models-on-real-world-banking-data-f6dd3d7e998f