

Search

INTRODUCTION

Installation

Multi-goal API

ENVIRONMENTS

Fetch

Shadow Dexterous Hand

Manipulate Block

Manipulate Block Touch Sensors

Manipulate Egg

Manipulate Egg Touch Sensors

Manipulate Pen

Manipulate Pen Touch Sensors

Reach

Maze

Adroit Hand

Franka Kitchen

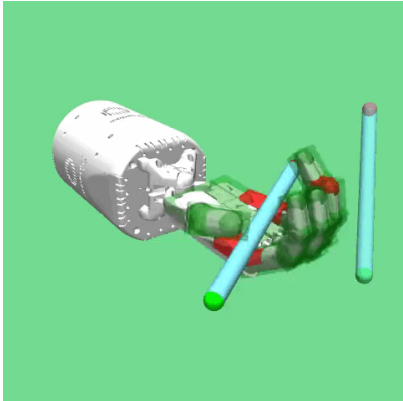
MaMuJoCo (Multi-Agent MuJoCo)

DEVELOPMENT

Github

Release Notes

Manipulate Pen Touch Sensors



Description

This environment was introduced in [“Using Tactile Sensing to Improve the Sample Efficiency and Performance of Deep Deterministic Policy Gradients for Simulated In-Hand Manipulation Tasks”](#).

The environment is based on the same robot hand as in the `HandReach` environment, the [Shadow Dexterous Hand](#). The task to be solved is the same as in the `HandManipulatePen` environment. However, in this case the environment observation also includes tactile sensory information. This is achieved by placing a total of 92 MuJoCo [touch sensors](#) in the palm and finger phalanxes of the hand. The sensors are created by specifying the tactile sensors’ active zones by so-called sites. Each site can be represented as either ellipsoid (for the finger tips) or box (for the phalanxes and palm sensors). When rendering the environment the sites are visualized as red and green transparent shapes attached to the hand model. If a body’s contact point falls within a site’s volume and involves a geometry attached to the same body as the site, the corresponding contact force is included in the sensor reading. Soft contacts do not influence the above computation except inasmuch as the contact point might move outside of the site, in which case if a contact point falls outside the sensor zone, but the normal ray intersects the sensor zone, it is also included. MuJoCo touch sensors only report normal forces using Minkowski Portal Refinement approach . The output of the contact sensor is a non-negative scalar value of type float that is computed as the sum of all contact normal forces that were included for this sensor in the current time step . Thus, each sensor of the 92 virtual touch sensors has a non-negative scalar value.

The sensors are divided between the areas of the tip, middle, and lower phalanx of the forefinger, middle, ring, and little fingers. In addition to the areas of the three thumb phalanxes and the paml. The number of sensors are divided as follows in the different defined areas of the hand:

Functional areas of the hand model	Number of areas	Sensors-per-area	Total Sensors
Lower phalanx of the fingers	4	7	28
Middle phalanx of the fingers	4	5	20
Tip phalanxes of the fingers	4	5	20
Thumb phalanxes	3	5	15
Palm	1	9	9

When adding the sensors to the `HandManipulatePen` environment there are two possible environment initializations depending on the type of data returned by the touch sensors. This data can be continuous values of external forces or a boolean value which is `True` if the sensor detects any contact force and `False` if not. This two types of environments can be initialized from the environment id variations of `HandManipulatePen` by adding the `_ContinuousTouchSensors` string to the id if the touch sensors return continuous force values or `_BooleanTouchSensors` if the values are boolean.

Continuous Touch Sensor Environments:

- `HandManipulatePen_ContinuousTouchSensors-v1`
- `HandManipulatePenRotate_ContinuousTouchSensors-v1`
- `HandManipulatePenFull_ContinuousTouchSensors-v1`

Boolean Touch Sensor Environments:

- `HandManipulatePen_BooleanTouchSensors-v1`
- `HandManipulatePenRotate_BooleanTouchSensors-v1`
- `HandManipulatePenFull_BooleanTouchSensors-v1`

The `Action Space`, `Rewards`, `Starting State`, `Episode End`, and `Arguments` are the same as for the `HandManipulatePen` environment and its variations.

Observation Space

The observation is a `goal-aware observation space`. It consists of a dictionary with information about the robot’s joint and Pen states, as well as information about the goal and touch sensors. The dictionary consists of the same 3 keys as the `HandManipulatePen` environments (`observation`, `desired_goal`, and `achieved_goal`). However, the `ndarray` of the observation is now of shape `(153,)` instead of `(61,)` since the touch sensor information is added at the end of the array with shape `(92,)`.

Version History

- v1: the environment depends on the newest [mujoco python bindings](#) maintained by the MuJoCo team in Deepmind.
- v0: the environment depends on `mujoco_py` which is no longer maintained.