

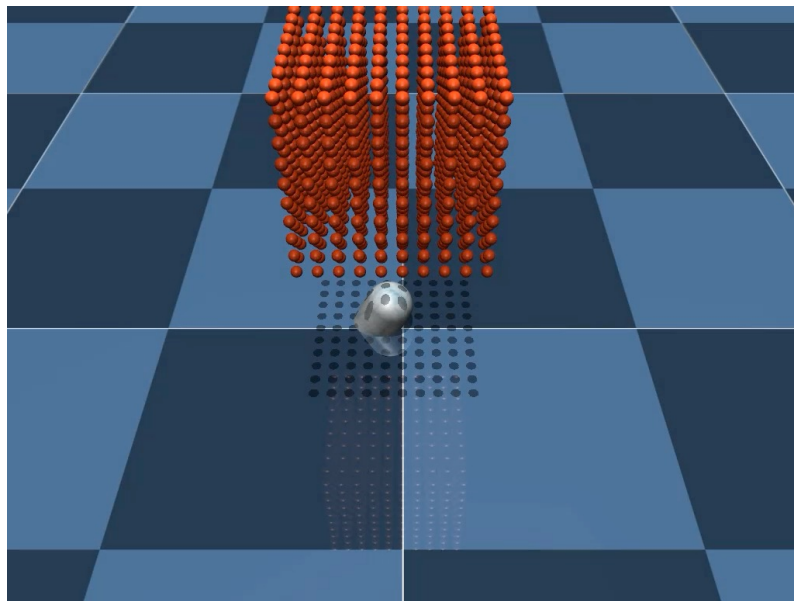
CS 391R robosuite Tutorial

Soroush Nasiriany

September 6, 2023

What is physical simulation?

The simulation of systems of objects that are free to move, usually in three dimensions according to Newton's laws of dynamics.



Why use simulation?

- **Democratize robot learning research.** Prototype ideas at scale and low cost
- **Easy access to state information**, such as object poses, mass, friction, etc
- **Safe** and precisely controlled experiments
- **Challenges:** difficult to model everything in the real world, difficult to transfer models from simulation to real world

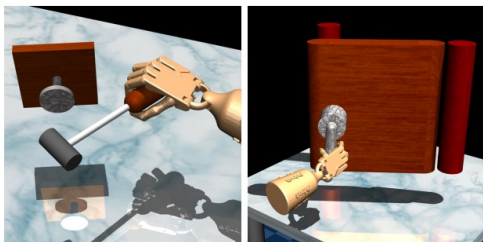
A brief survey of robot simulation frameworks

MuJoCo

robosuite

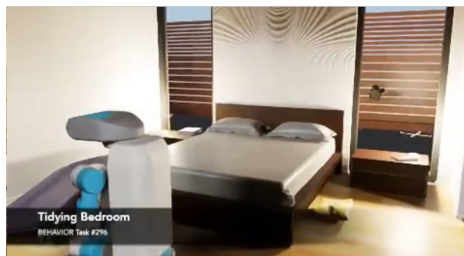


Adroit



PhysX

Behavior-1k

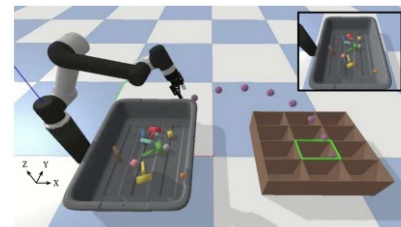


Orbit

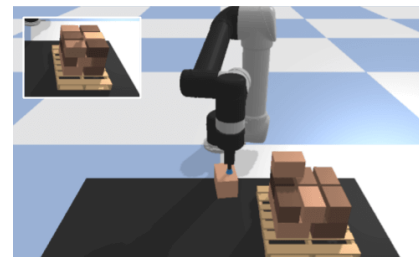


Bullet

Tossing Bot



Ravens

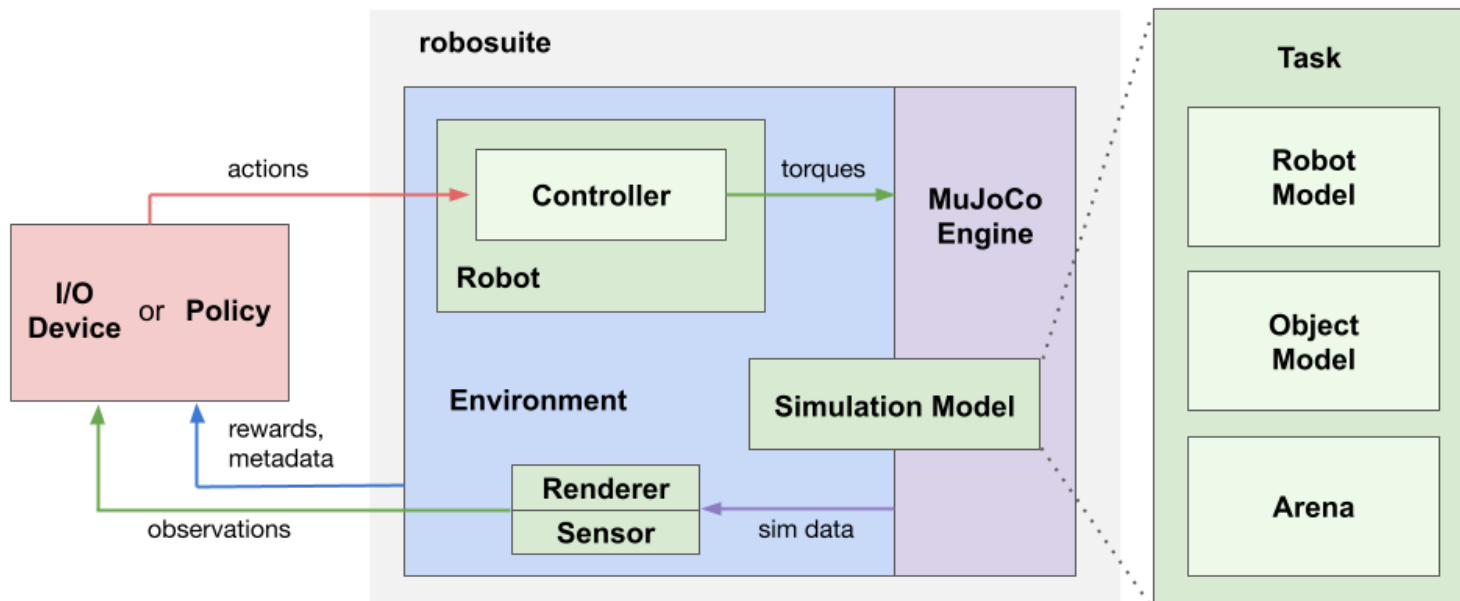


robosuite: A Modular Simulation Framework and Benchmark for Robot Learning

- Development led by Zhu et al., released in 2020
- Built on top of MuJoCo physics engine
- Built with focus on **modularity**: support suite of tasks across multiple robots

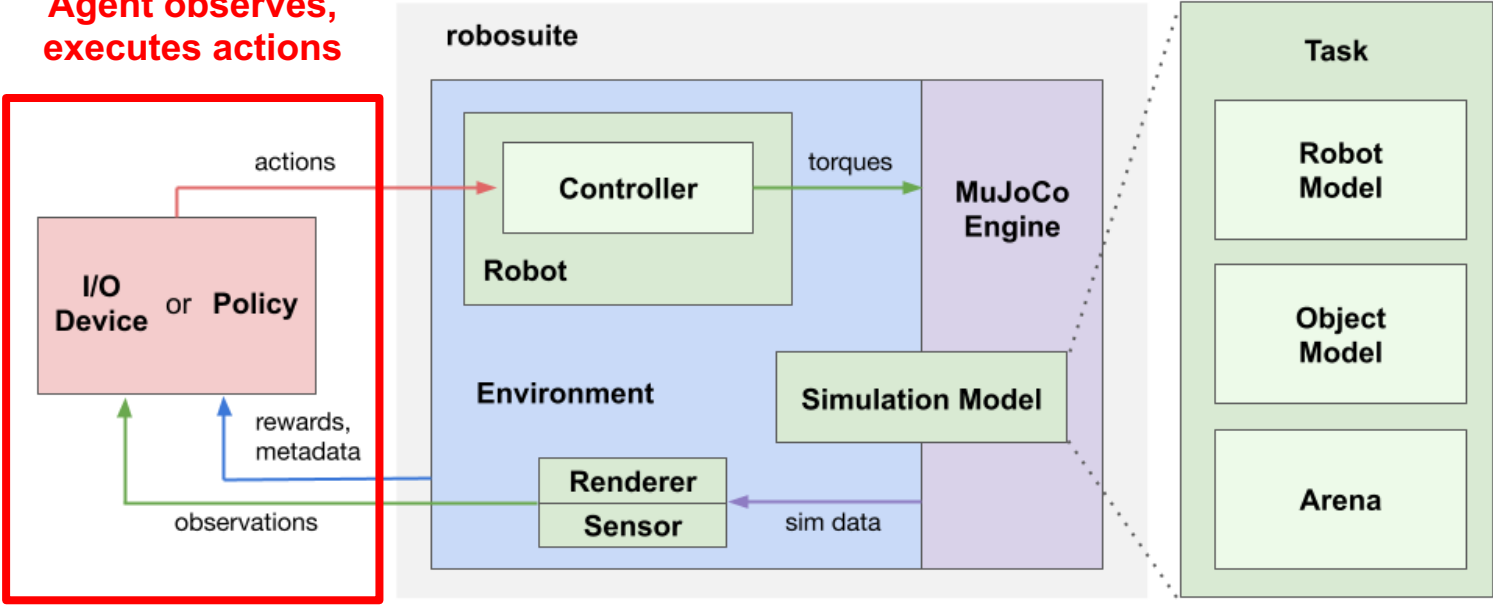


System overview

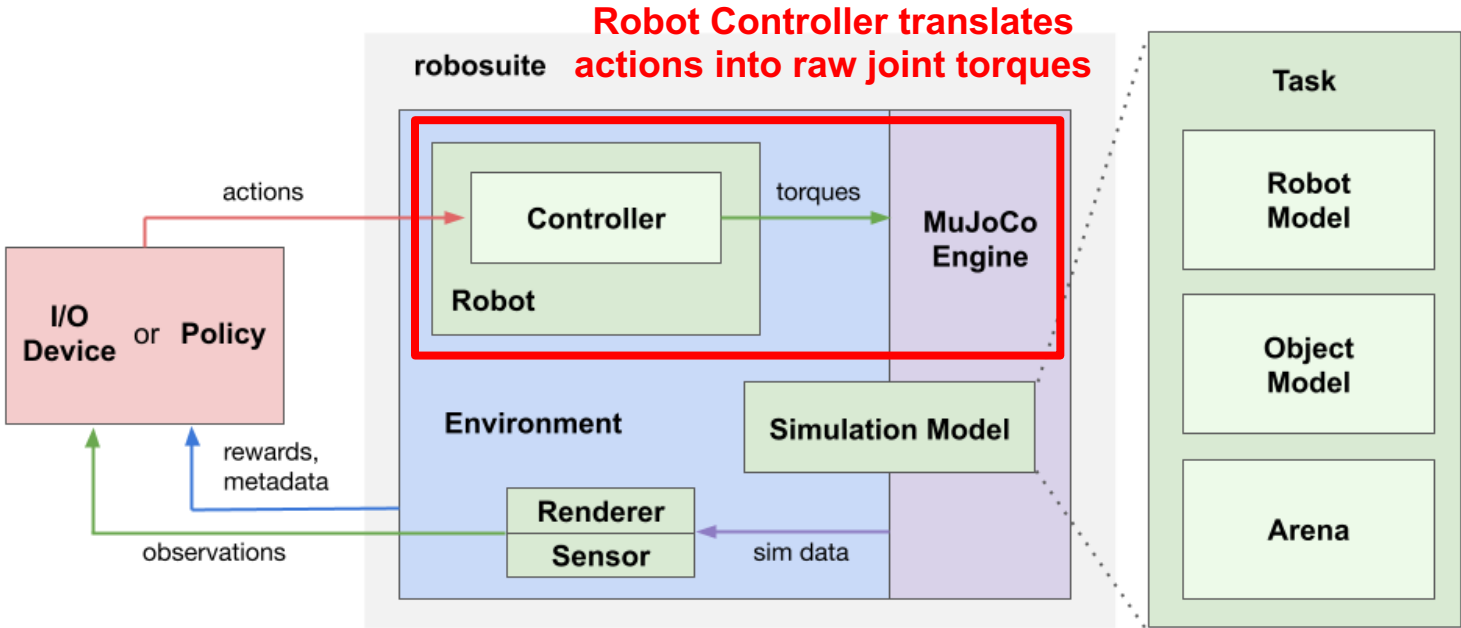


System overview

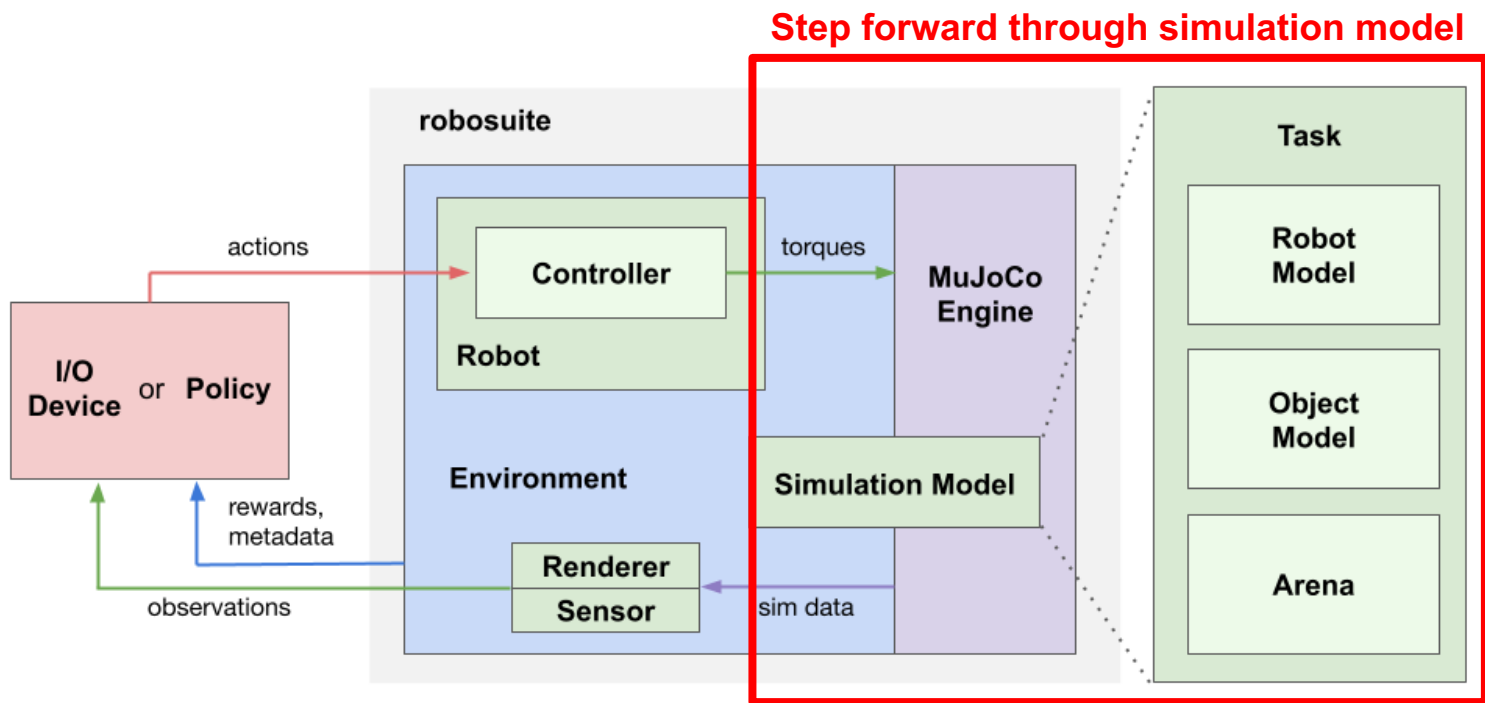
Agent observes,
executes actions



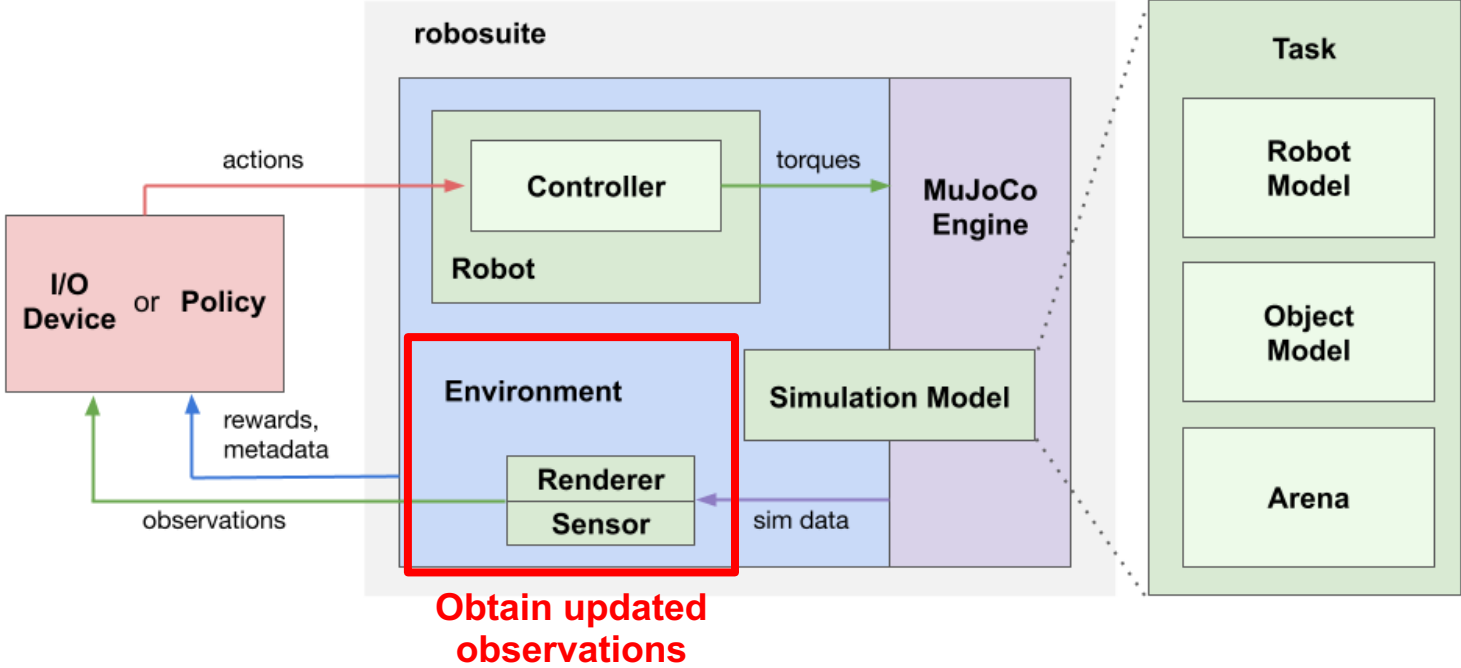
System overview



System overview



System overview



Getting started with robosuite

```
1 import numpy as np
2 import robosuite as suite
3
4 # create environment instance
5 env = suite.make(
6     env_name="Lift", # try with other tasks like "Stack" and "Door"
7     robots="Panda", # try with other robots like "Sawyer" and "Jaco"
8     has_renderer=True,
9     has_offscreen_renderer=False,
10    use_camera_obs=False,
11 )
12
13 # reset the environment
14 env.reset()
15
16 for i in range(1000):
17     action = np.random.randn(env.robots[0].dof) # sample random action
18     obs, reward, done, info = env.step(action) # take action in the environment
19     env.render() # render on display
```

Anatomy of an environment: arena, robots, objects

```
1 # load model for table top workspace
2 from robosuite.models.arenas import TableArena
3 mujoco_arena = TableArena()
4
5 # Arena always gets set to zero origin
6 mujoco_arena.set_origin([0, 0, 0])
7
8 # set up robot
9 from robosuite.models.robots import Panda
10 mujoco_robot = Panda()
11
12 # add gripper to robot
13 from robosuite.models.grippers import gripper_factory
14 gripper = gripper_factory("PandaGripper")
15 mujoco_robot.add_gripper(gripper)
16
```

```
17 # initialize objects of interest
18 from robosuite.models.objects import BoxObject
19 cube = BoxObject(
20     name="cube",
21     size_min=[0.020, 0.020, 0.020],
22     size_max=[0.022, 0.022, 0.022],
23     rgba=[1, 0, 0, 1],
24 )
25
26 # task includes arena, robot, and objects of interest
27 from robosuite.models.tasks import ManipulationTask
28 model = ManipulationTask(
29     mujoco_arena=mujoco_arena,
30     mujoco_robots=[mujoco_robot],
31     mujoco_objects=cube,
32 )
```

Anatomy of an environment: arena, robots, objects

Arena

```
1 # load model for table top workspace
2 from robosuite.models.arenas import TableArena
3 mujoco_arena = TableArena()
4
5 # Arena always gets set to zero origin
6 mujoco_arena.set_origin([0, 0, 0])
7
8 # set up robot
9 from robosuite.models.robots import Panda
10 mujoco_robot = Panda()
11
12 # add gripper to robot
13 from robosuite.models.grippers import gripper_factory
14 gripper = gripper_factory("PandaGripper")
15 mujoco_robot.add_gripper(gripper)
16
```

```
17 # initialize objects of interest
18 from robosuite.models.objects import BoxObject
19 cube = BoxObject(
20     name="cube",
21     size_min=[0.020, 0.020, 0.020],
22     size_max=[0.022, 0.022, 0.022],
23     rgba=[1, 0, 0, 1],
24 )
25
26 # task includes arena, robot, and objects of interest
27 from robosuite.models.tasks import ManipulationTask
28 model = ManipulationTask(
29     mujoco_arena=mujoco_arena,
30     mujoco_robots=[mujoco_robot],
31     mujoco_objects=cube,
32 )
```

Anatomy of an environment: arena, robots, objects

```
1 # load model for table top workspace
2 from robosuite.models.arenas import TableArena
3 mujoco_arena = TableArena()
4
5 # Arena always gets set to zero origin
6 mujoco_arena.set_origin([0, 0, 0])
7
8 # set up robot
9 from robosuite.models.robots import Panda
10 mujoco_robot = Panda()
11
12 # add gripper to robot
13 from robosuite.models.grippers import gripper_factory
14 gripper = gripper_factory("PandaGripper")
15 mujoco_robot.add_gripper(gripper)
16
```

Robots

```
17 # initialize objects of interest
18 from robosuite.models.objects import BoxObject
19 cube = BoxObject(
20     name="cube",
21     size_min=[0.020, 0.020, 0.020],
22     size_max=[0.022, 0.022, 0.022],
23     rgba=[1, 0, 0, 1],
24 )
25
26 # task includes arena, robot, and objects of interest
27 from robosuite.models.tasks import ManipulationTask
28 model = ManipulationTask(
29     mujoco_arena=mujoco_arena,
30     mujoco_robots=[mujoco_robot],
31     mujoco_objects=cube,
32 )
```

Anatomy of an environment: arena, robots, objects

Objects

```
1 # load model for table top workspace
2 from robosuite.models.arenas import TableArena
3 mujoco_arena = TableArena()
4
5 # Arena always gets set to zero origin
6 mujoco_arena.set_origin([0, 0, 0])
7
8 # set up robot
9 from robosuite.models.robots import Panda
10 mujoco_robot = Panda()
11
12 # add gripper to robot
13 from robosuite.models.grippers import gripper_factory
14 gripper = gripper_factory("PandaGripper")
15 mujoco_robot.add_gripper(gripper)
16
```

```
17 # initialize objects of interest
18 from robosuite.models.objects import BoxObject
19 cube = BoxObject(
20     name="cube",
21     size_min=[0.020, 0.020, 0.020],
22     size_max=[0.022, 0.022, 0.022],
23     rgba=[1, 0, 0, 1],
24 )
25
26 # task includes arena, robot, and objects of interest
27 from robosuite.models.tasks import ManipulationTask
28 model = ManipulationTask(
29     mujoco_arena=mujoco_arena,
30     mujoco_robots=[mujoco_robot],
31     mujoco_objects=cube,
32 )
```

Anatomy of an environment: arena, robots, objects

```
1 # load model for table top workspace
2 from robosuite.models.arenas import TableArena
3 mujoco_arena = TableArena()
4
5 # Arena always gets set to zero origin
6 mujoco_arena.set_origin([0, 0, 0])
7
8 # set up robot
9 from robosuite.models.robots import Panda
10 mujoco_robot = Panda()
11
12 # add gripper to robot
13 from robosuite.models.grippers import gripper_factory
14 gripper = gripper_factory("PandaGripper")
15 mujoco_robot.add_gripper(gripper)
16
```

```
17 # initialize objects of interest
18 from robosuite.models.objects import BoxObject
19 cube = BoxObject(
20     name="cube",
21     size_min=[0.020, 0.020, 0.020],
22     size_max=[0.022, 0.022, 0.022],
23     rgba=[1, 0, 0, 1],
24 )
25
26 # task includes arena, robot, and objects of interest
27 from robosuite.models.tasks import ManipulationTask
28 model = ManipulationTask(
29     mujoco_arena=mujoco_arena,
30     mujoco_robots=[mujoco_robot],
31     mujoco_objects=cube,
32 )
```

Combine all components together

Useful resources

From **robosuite.ai** find the code, documentation, and whitepaper



Search the docs ...

INTRODUCTION

Overview

Installation

Quick Start

Demo Showcases

MODULES

Overview

Robots

Controllers

Objects

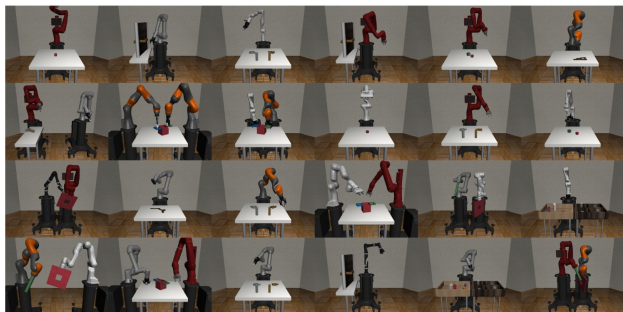
Environments

Sensors

I/O Devices

Renderers

Overview



robosuite is a simulation framework powered by the [MuJoCo](#) physics engine for robot learning. It also offers a suite of benchmark environments for reproducible research. The current release (v1.4) features long-term support with the official MuJoCo binding from DeepMind. This project is part of the broader [Advancing Robot Intelligence through Simulated Environments \(ARISE\)](#) Initiative, with the aim of lowering the barriers of entry for cutting-edge research at the intersection of AI and Robotics.

robosuite: A Modular Simulation Framework and Benchmark for Robot Learning

Yuke Zhu^{*} Josiah Wong^{*} Ajay Mandlekar^{*} Roberto Martín-Martín^{**}
Abhishek Joshi[◇] Soroush Nasiriany[◇] Yifeng Zhu^{◇†}

robosuite.ai

Abstract

robosuite is a simulation framework for robot learning powered by the MuJoCo physics engine. It offers a modular design for creating robotic tasks as well as a suite of benchmark environments for reproducible research. This paper discusses the key system modules and the benchmark environments of our new release **robosuite v1.0**. For the latest updates on **robosuite**, please visit our project website.