

SMART GRASPING SANDBOX





- **THE SMART GRASPING
SANDBOX**
- **DOES MY GRASP
SUCK?**
- **HUMANS CAN BE
USEFUL TOO**



- **THE SMART GRASPING SANDBOX**
- DOES MY GRASP SUCK?
- HUMANS CAN BE USEFUL TOO

THE SMART GRASPING SANDBOX - SPECS



ENVIRONMENT

+



TOOLS

+



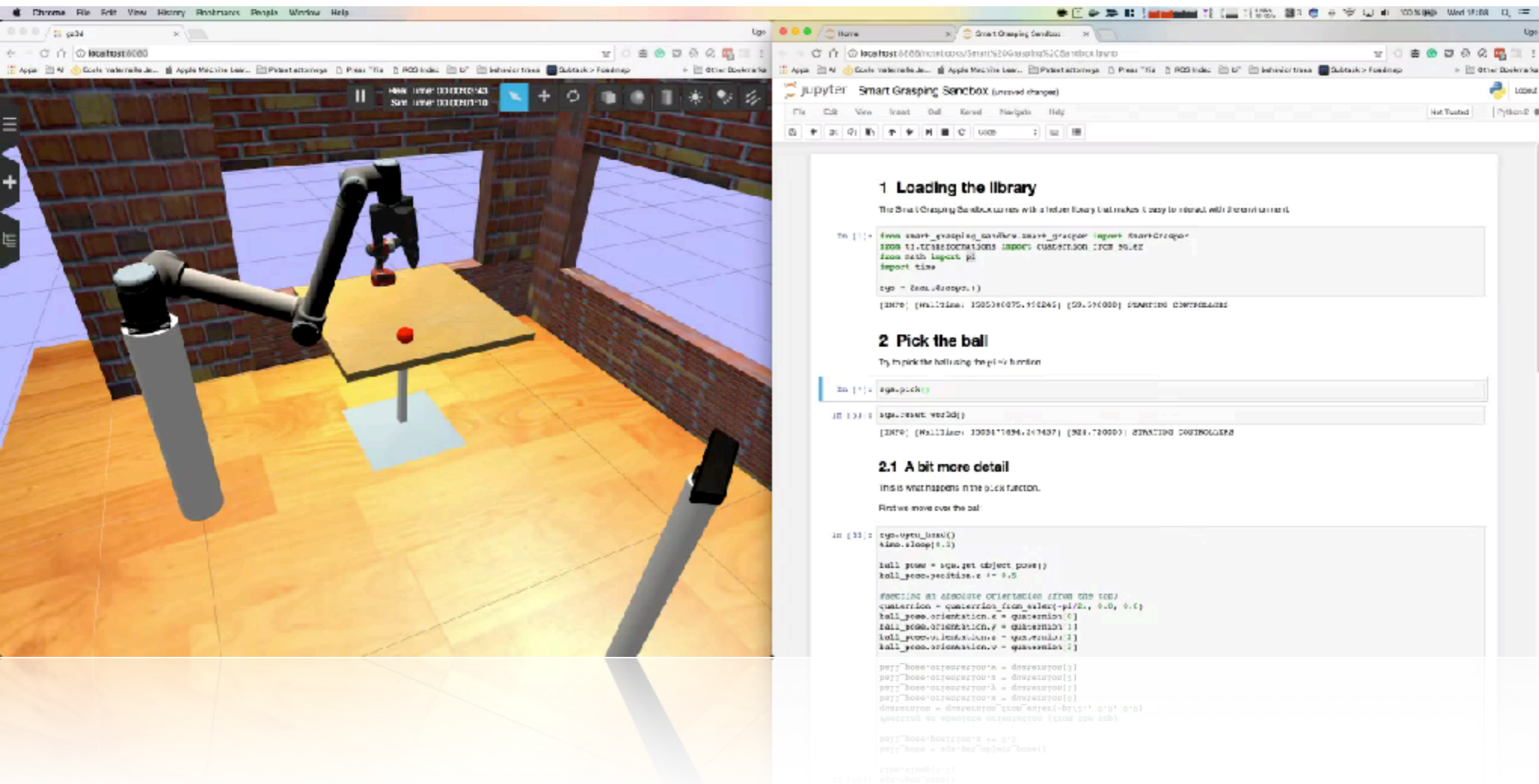
HEADLESS

=



THE SMART GRASPING SANDBOX

THE SMART GRASPING SANDBOX – OVERVIEW



The image displays a dual-view interface for the Smart Grasping Sandbox. On the left, a 3D simulation shows a robotic arm with a gripper positioned over a red ball on a wooden table. The environment features brick walls and a wooden floor. On the right, a Jupyter notebook titled 'Smart Grasping Sandbox (unsaved changes)' contains Python code for controlling the robot. The code is organized into sections: '1 Loading the library', '2 Pick the ball', and '2.1 A bit more detail'. The code includes imports for the 'smart_grasping_sandbox' and 'scipy.optimize' libraries, and defines a 'pick' function to move the robot's gripper to a specific position and orientation.

```
In [1]: from smart_grasping_sandbox.smart_grasping import SmartGrasping
from ti.roboticarm import CUSTOMERION_IPC_SOCKET
from math import pi
import time

csg = SmartGrasping()

[INFO: (MainThread: 100538075.952845) (29.090000) CUSTOMERION_IPC_SOCKET]

2 Pick the ball
Try to pick the ball using the pick function

In [2]: csg.pick()

[INFO: (MainThread: 100538075.952845) (29.090000) CUSTOMERION_IPC_SOCKET]

2.1 A bit more detail
This is what happens in the pick function.
First we move over the ball

In [3]: csg.move_base()
time.sleep(0.1)

ball_pos = csg.get_object_pos()
ball_pos.y_pos += 0.5

# setting an absolute orientation from the top
quaternion = quaternion_from_euler(-pi/2, 0.0, 0.0)
ball_pos.orientation.q = quaternion()
ball_pos.orientation.y = quaternion[1]
ball_pos.orientation.z = quaternion[2]
ball_pos.orientation.w = quaternion[3]

print('base orientation')
print('base orientation.x =', quaternion[0])
print('base orientation.y =', quaternion[1])
print('base orientation.z =', quaternion[2])
print('base orientation.w =', quaternion[3])
quaternion = quaternion_from_euler(-pi/2, 0.0, 0.0)
quaternion = quaternion

print('base orientation')
print('base orientation.x =', quaternion[0])
print('base orientation.y =', quaternion[1])
print('base orientation.z =', quaternion[2])
print('base orientation.w =', quaternion[3])

In [4]: csg.pick()
```

```
docker run -it --name sgs -p 8080:8080 -p 8888:8888 -p 8181:8181 -p 7681:7681 shadowrobot/smart_grasping_sandbox
```


TECHNOLOGIES USED



GZWEB 1.3.0



GAZEBO 7



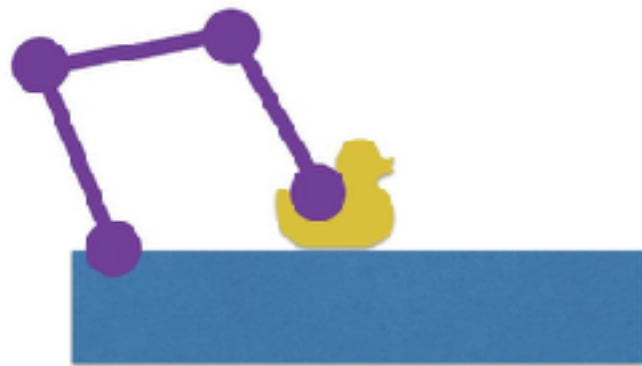
OR





- THE SMART GRASPING
SANDBOX
- DOES MY GRASP
SUCK?
- HUMANS CAN BE
USEFUL TOO

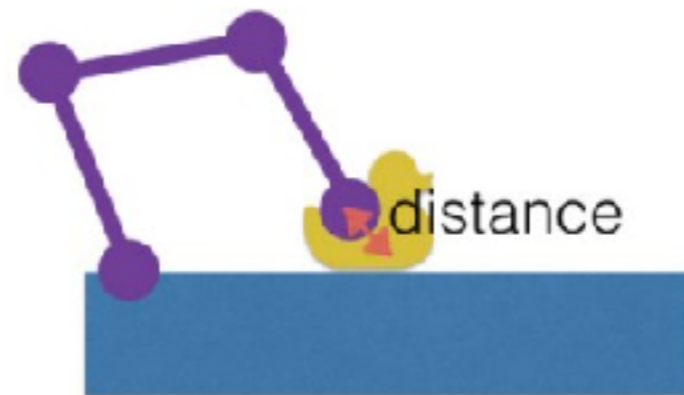
GATHERING DATA – AN OBJECTIVE GRASP STABILITY MEASURE



1. GRASP

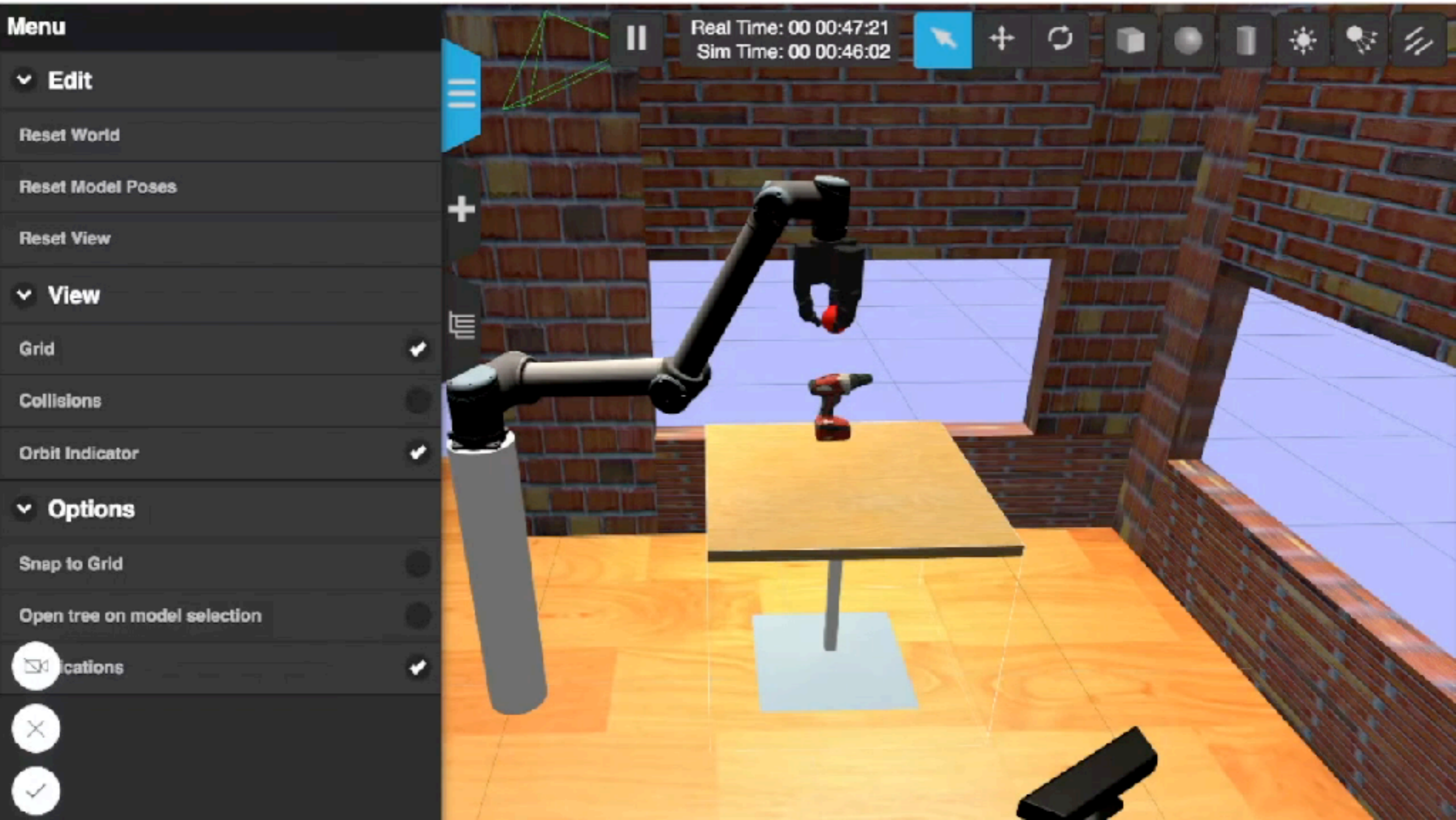


2. SHAKE



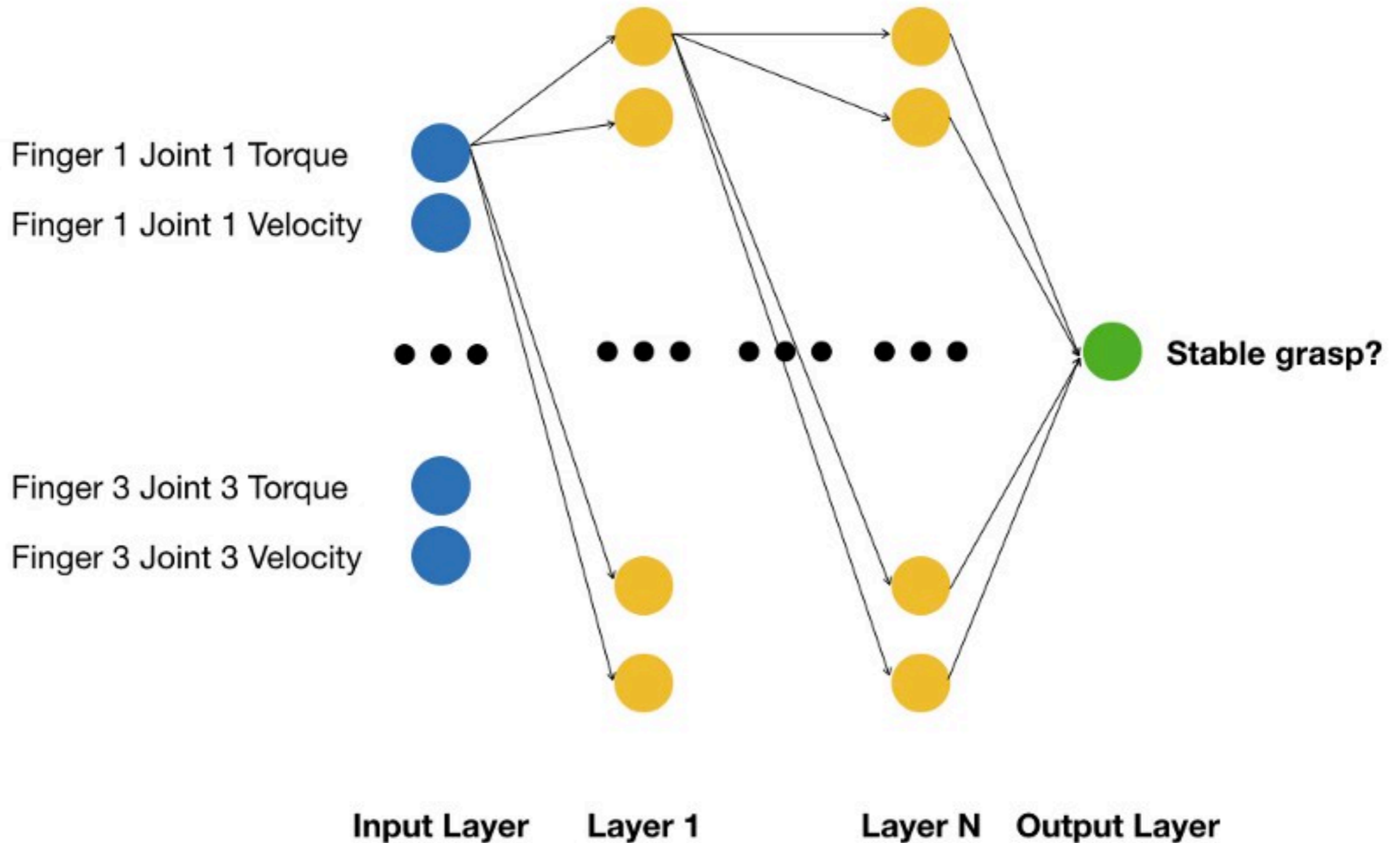
3. MEASURE

GATHERING DATA



kaggle

LET'S LEARN



LIVE PREDICTION

The image displays a live prediction interface, likely a Jupyter Notebook, showing code execution and a 3D simulation of a robotic arm.

Code Execution: The code defines a function `cb(msg)` that updates a model's prediction based on new quality data. The model is used to predict the next quality value. The code is executed in a Python 2 environment.

```
def cb(msg):  
    = []  
    joints:  
    = msg.name.index(joint)  
    tor.append(msg.velocity[id_joint])  
    tor.append(msg.effort[id_joint])  
    _default})  
    .pop(0)  
    asp_model.predict(np.array([input_vector]))  
    .append(qual)  
    debug( 'new quality: ' + str(qual))  
    .data['y'] = qualities  
    ()  
  
grasp_quality', anonymous=True)  
rider('/joint_states', JointState, joint_states_cb, queue_size=1)
```

Simulation: The 3D simulation shows a robotic arm in a virtual environment. The arm is positioned over a red ball on a yellow table. The simulation is running in real-time, with a real-time clock showing 00:02:16:57 and a simulation time of 00:02:12:35.

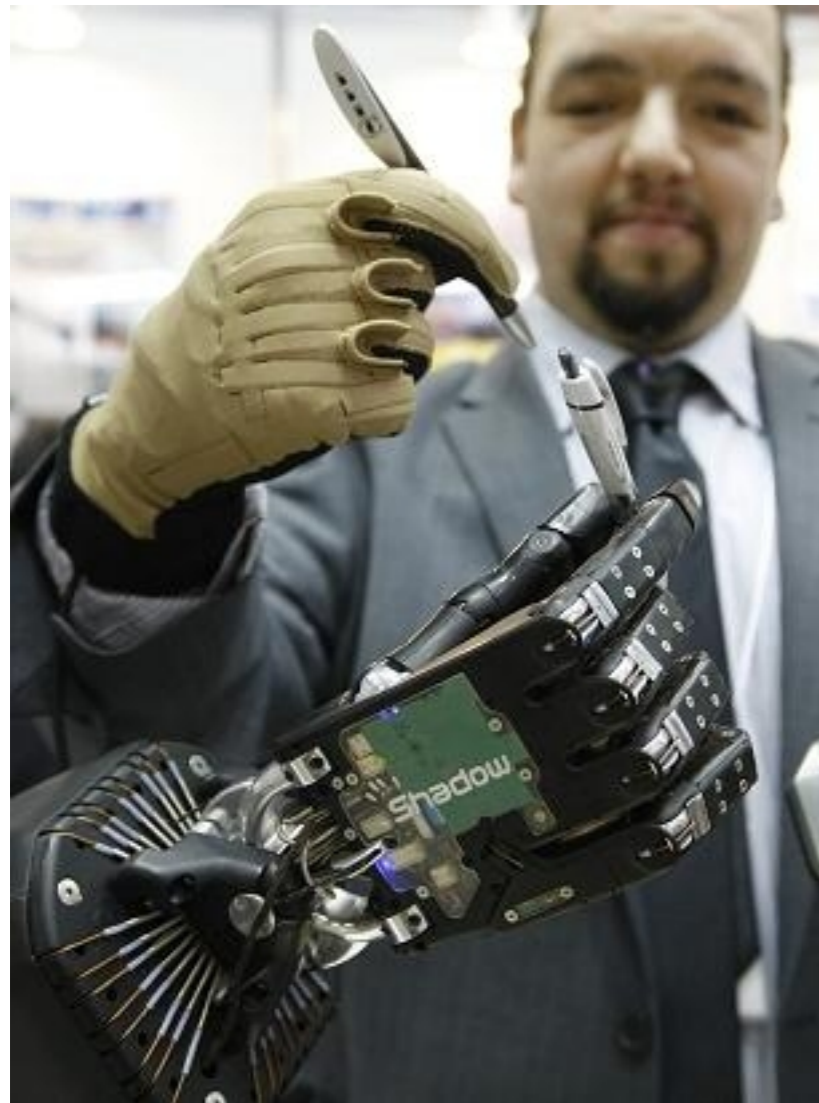
Menu: The simulation interface includes a menu with the following options:

- Edit**
 - Reset World
 - Reset Model Poses
 - Reset View
- View**
 - Grid
 - Collisions
 - Orbit Indicator
- Options**
 - Snap to Grid
 - Open tree on model selection
 - Notifications



- THE SMART GRASPING SANDBOX
- DOES MY GRASP SUCK?
- **HUMANS CAN BE USEFUL TOO**

TEACHING FROM DEMONSTRATION



TEACHING

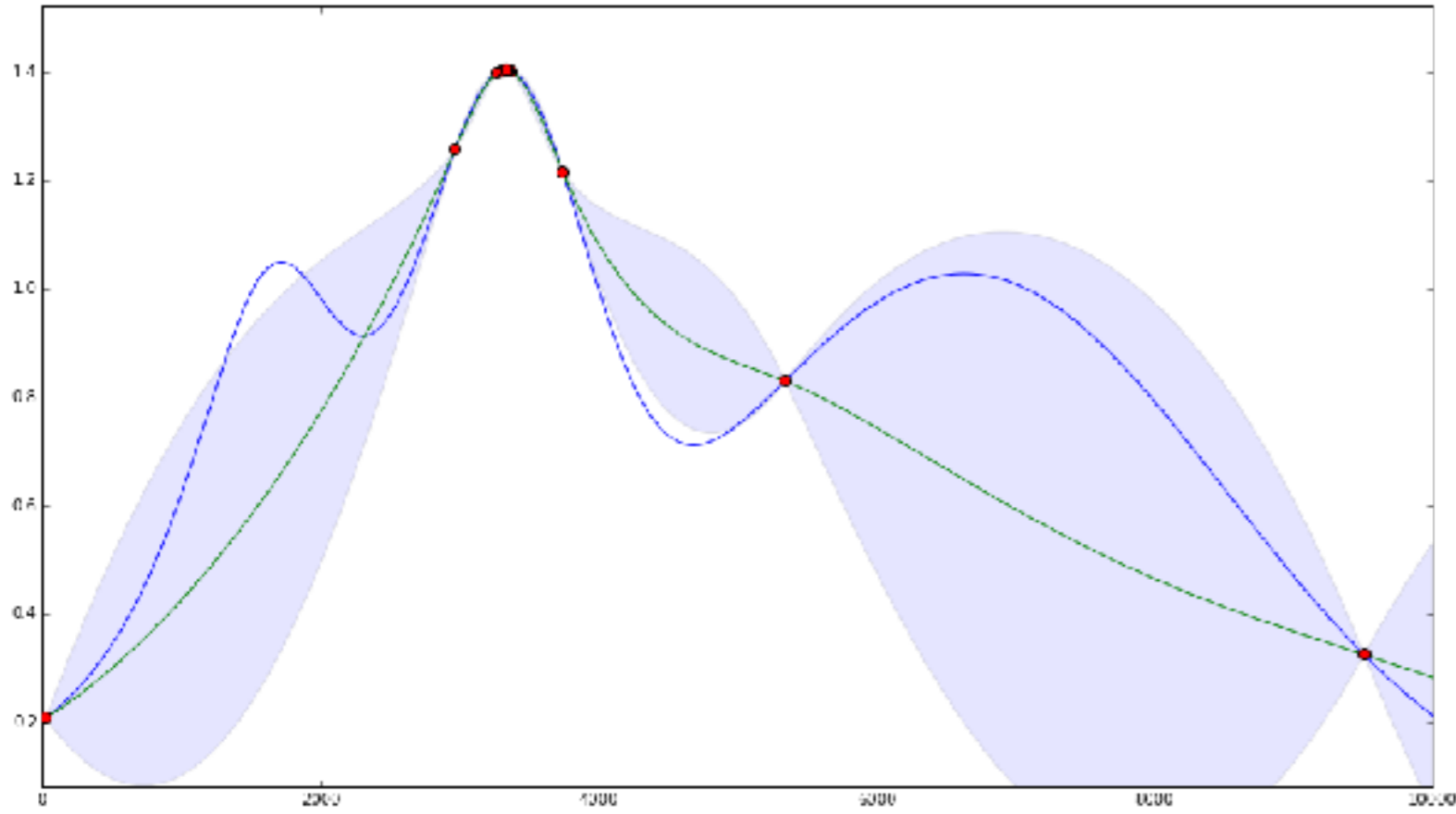


REFINING



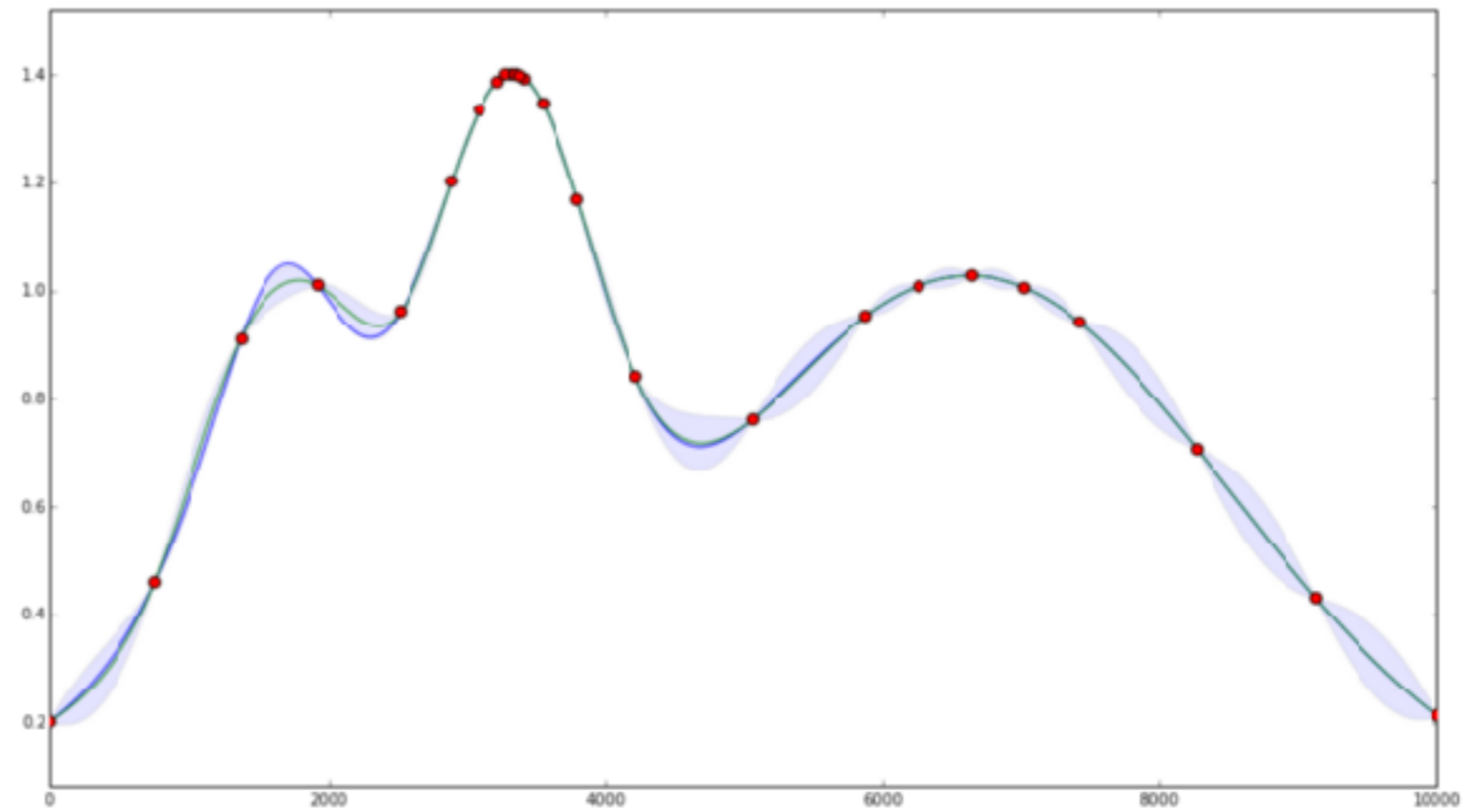
USING

BAYESIAN OPTIMISATION

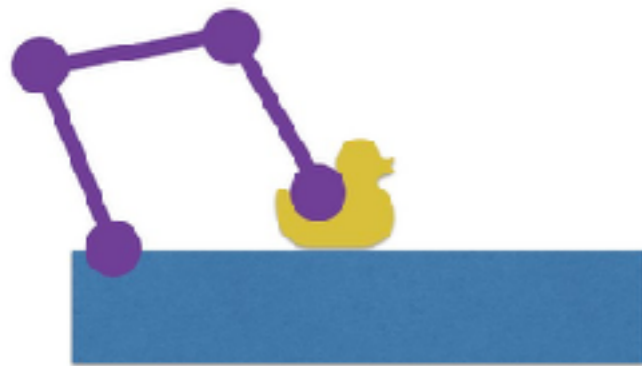


EXPLOITATION

EXPLORATION



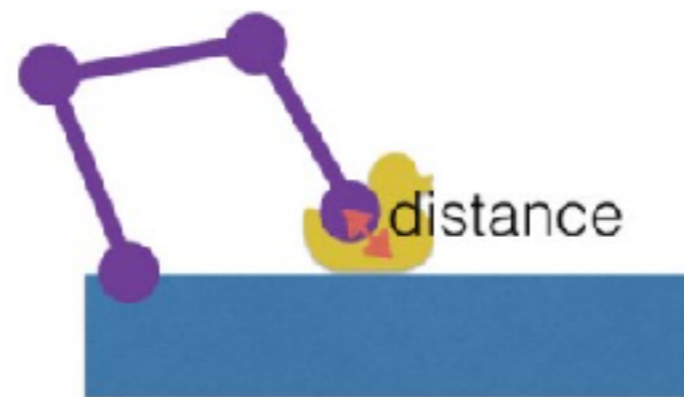
GATHERING DATA – AN OBJECTIVE GRASP STABILITY MEASURE



GRASP

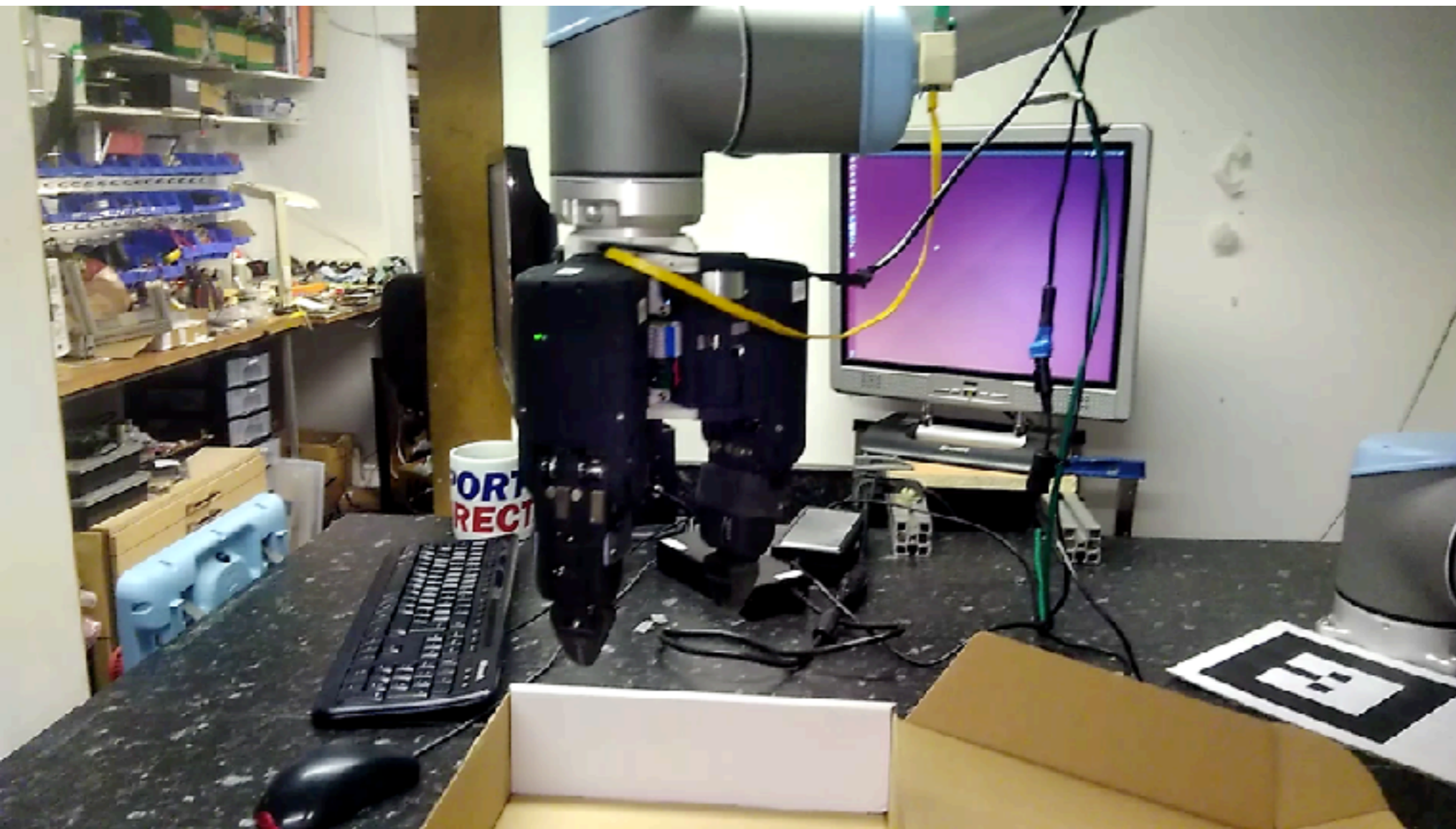


SHAKE



MEASURE

github.com/shadow-robot/smart_grasping_sandbox



@ugocupcic