

Introduction to Robotics

Instructor: Stefanos Nikolaidis



TAs:

Shihan Zhao



Sophie Hsu



Varun Bhatt



Course Website:

<http://www.stefanosnikolaidis.net/robotics2023.html>



Robots interacting with the world



CMU Boss DARPA Urban Challenge 2007, video shared by Geoff Hollinger

Robots interacting with the world

Personal Robotics Lab 
Carnegie Mellon University

Robots interacting with the world

Robotic Lime Picking by considering leaves as Permeable Obstacles

**Heramb Nemlekar*, Ziang Liu*, Suraj Kothawade, Sherdil Niyaz,
Barath Raghavan and Stefanos Nikolaidis**

** equal contribution*

**ICAROS Lab
University of Southern California**

Robots interacting with people

Automatic Feeding with Assistive Robot Arm

Laura V. Herlant
The Robotics Institute
Carnegie Mellon University

Siddhartha S. Srinivasa
University of Washington

Robots interacting with people



Robots interacting with people

Design and Evaluation of a Hair Combing System Using a General-Purpose Robotic Arm

Nathaniel Dennler, Eura Shin, Maja Matarić, and Stefanos Nikolaidis

Robots interacting with people

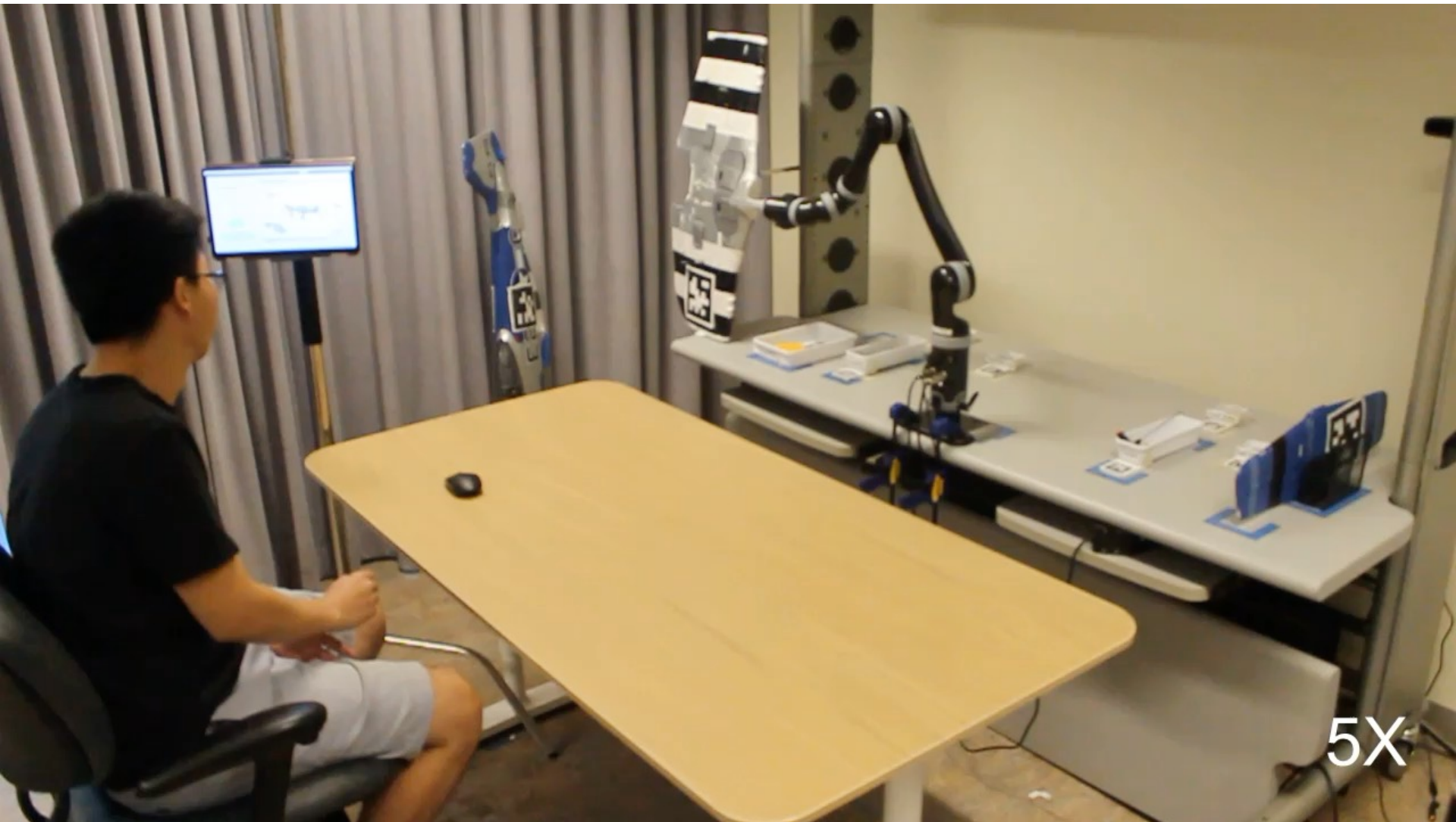
Two-Stage Clustering of Human Preferences for Action Prediction in Assembly Tasks

Heramb Nemlekar, Jignesh Modi, Satyandra K. Gupta, and Stefanos Nikolaidis

ICAROS lab
University of Southern California

Video by ICAROS Lab, University of Southern California

Robots interacting with people

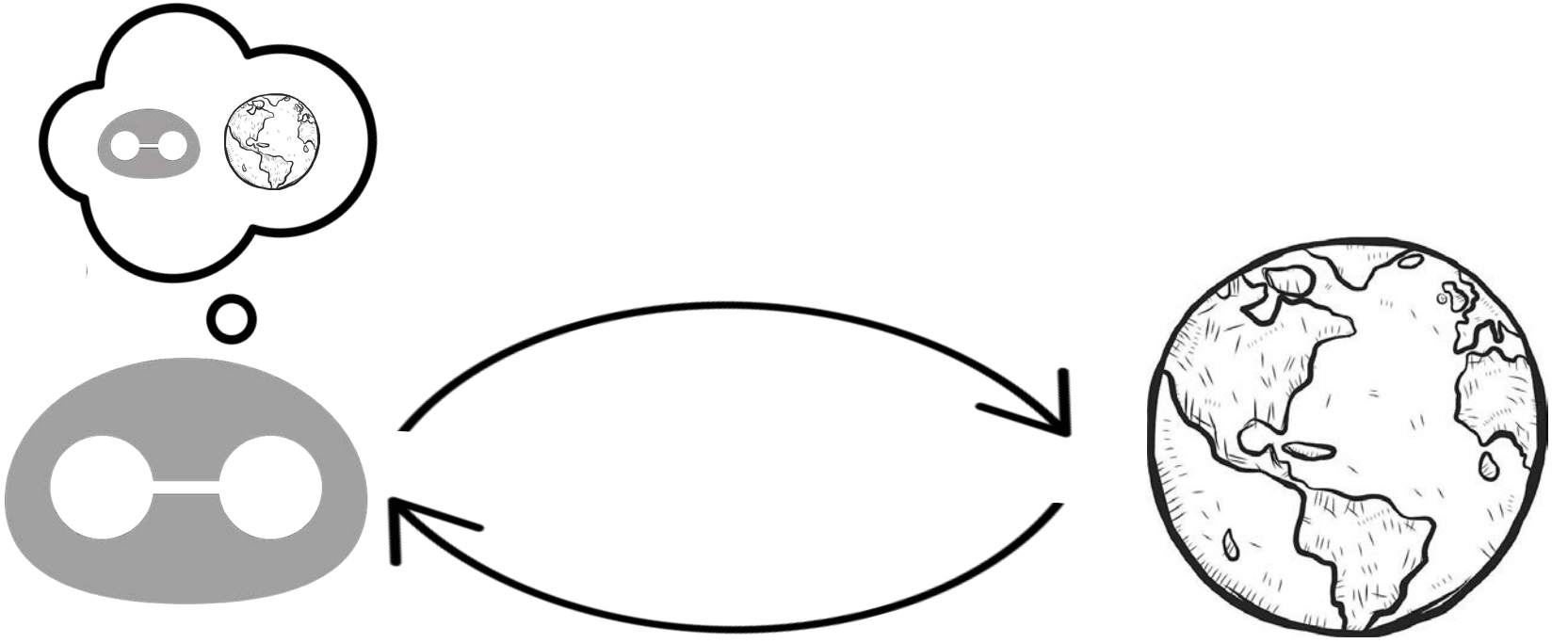


Video by ICAROS Lab, University of Southern California

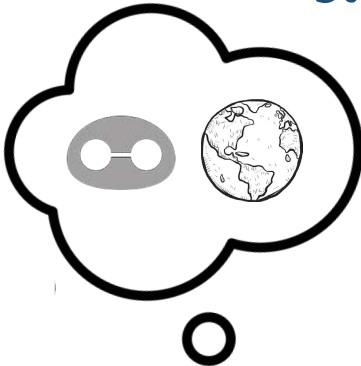
Robots interacting with robots



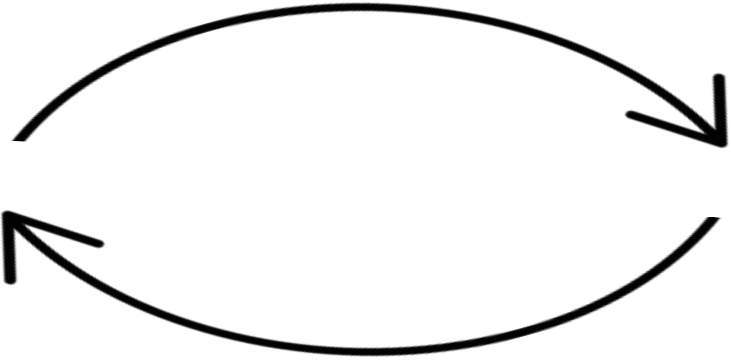
Video by ICAROS Lab, University of Southern California



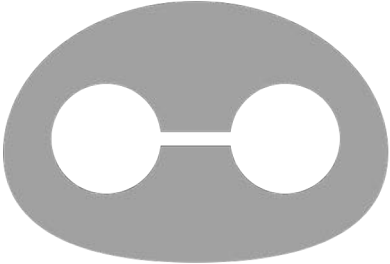
state estimation



action



observation



Learning Objectives

- demonstrate proficiency in the theoretical tools that support state estimation, manipulation and planning with sensor and modeling noise.
- implement these techniques and test them with real-world datasets.
- integrate your algorithms with state-of-the-art simulation environments

Schedule

Matrix Algebra Refresher

Probability Theory

Python / ROS Tutorial

Bayesian Networks

Linear Dynamical Systems

Bayesian and Kalman Filters

EKF and Particle Filters

Motion and Sensor Models

Localization and Mapping

SLAM

Mathematical Programming

Configuration Spaces

Kinematic Transformations

Combinatorial Motion Planning

Sampling-based Motion Planning

Constraint-based Planning

Dynamics

Non-Linear Control

Data-Driven Manipulation

Human-Robot Interaction



Math Fundamentals for Robotics

Robustness in State Estimation

Motion Planning

Robustness in Planning

Required Prerequisites

- Probability Theory
- Linear Algebra
- Calculus
- Python

Office Hours

- Stefanos Nikolaidis: Mon, 5:00 – 5:00
- TAs: 1h / week, TBD

- Things you should ask:
 - Technical material
 - Help with lab setup
 - Genuine concerns about team

- Things you should not ask:
 - Debugging Python
 - Questions covered in the prerequisites
 - Solving the assignments
 - Extensions

Lab Assignments (40%)

HW Assignments (30%)

Final (30%)

Note

- Regardless of the grading system, you are required to submit **all** homework assignments, lab assignments and take the final exam to receive a passing grade for the class.

Indicative Grading Scheme

check++ : >110%, extra credit

check+ : ~100%, meeting expectations without errors

check: ~90%, a few minor errors

check-: ~80%, multiple minor errors or a major error

check--: ~70%, multiple major errors or large missing parts

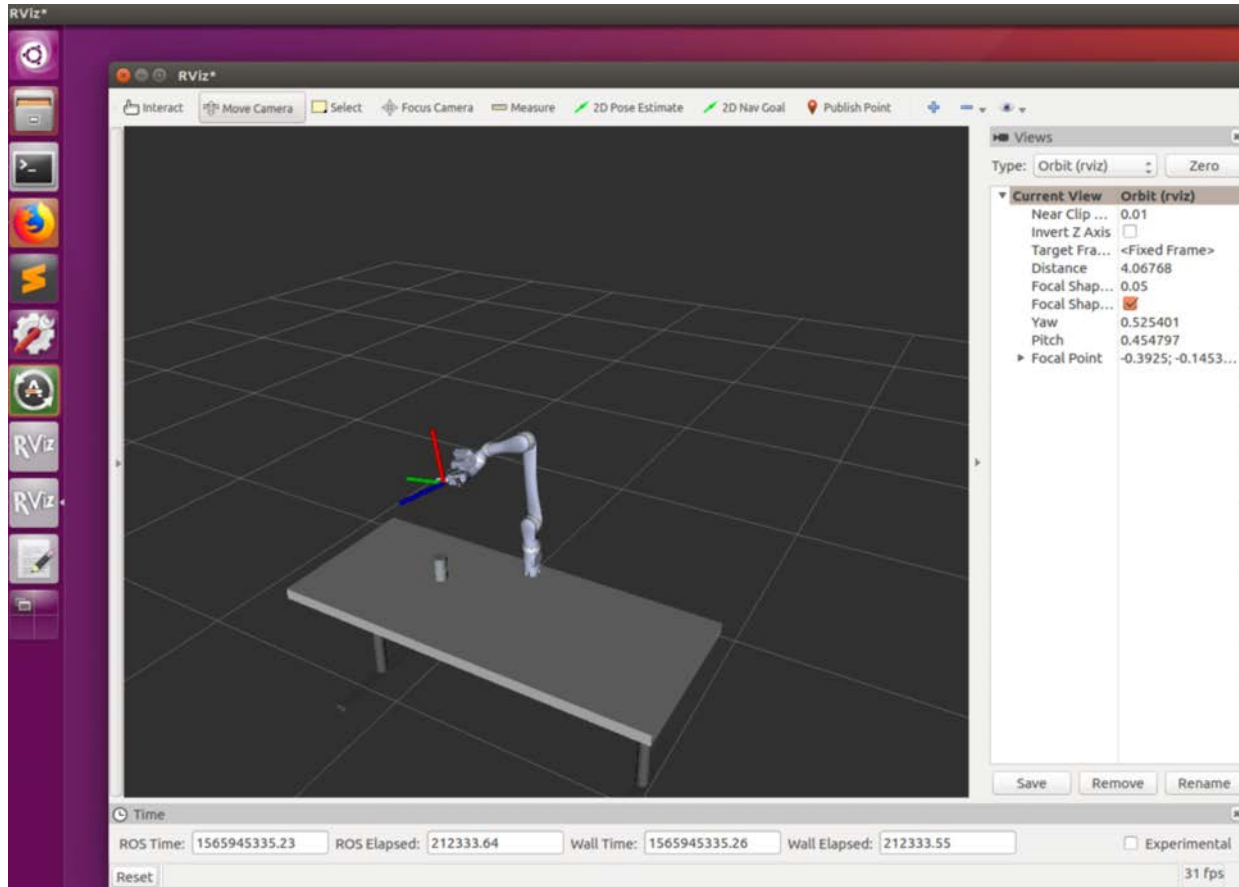
0: no submission

Lab Assignments (40%)

HW Assignments (30%)

Final (30%)

Lab Assignments



Lab Assignments

- For Windows and Linux users, download the free version of VMWare Workstation Player 17 (<https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html>).
- For Mac users (purchased before 2020), download the free version of VMWare Fusion Player 13 (<https://customerconnect.vmware.com/en/evalcenter?p=fusion-player-personal-13>).
- Assign multiple cores and large RAM for the VM if possible
- Start installation of VM early

Lab Assignments

- Groups of ~5 students, one implementation-based solution per group:
https://docs.google.com/spreadsheets/d/1nuseW1N0seG9mAfmG459zRI80gecB_4ob_2NrDzO4Vk/edit?usp=sharing
- Post solutions to your team's private repo in github
- Solutions should include code and report
- Lab 4 builds on Lab 3

Mac M1/M2 Users

VM is not working for M1/M2 Mac architecture

- Team up with members that do not have M1/M2 Macs (from 2020)
- Rent laptops from the library: <https://itservices.usc.edu/spaces/laptoploaner>
- Reach out to us

Lab Assignments (40%)

HW Assignments (30%)

Final (30%)

HW Assignments

- Individual assignments including theory and implementation
- Ok to discuss HW assignments with others but should write your own solutions and cite all resources / collaborators
- Upload solutions to provided google form
- Solutions should include report only (no need to upload code)
- Some Labs will build on HW

Late Assignment Policy

- Each students get 7 “slip” days in total (for the whole term) for late submission of Labs / HWs.
- We will not accept late submissions after the 7 days have been spent.

Plagiarism Policy

- Labs: We will assign 0 points to all groups involved in code sharing between groups or for uncited code retrieved online.
- HWs: We will assign 0 points to all students involved in solution sharing or if solution is retrieved online without proper reference.
- We encourage you to use generic code (e.g., ROS commands, plotting) found online, cited in the comments, and to look online for additional resources that help with the assignments.
- Ask the TAs when in doubt!

Lab Assignments (40%)

HW Assignments (30%)

Final (30%)

Final Exam

- 1 double-page cheat sheet.
- Based on lectures, HW and Lab assignments
- We will provide indicative list of topics near the end of the term