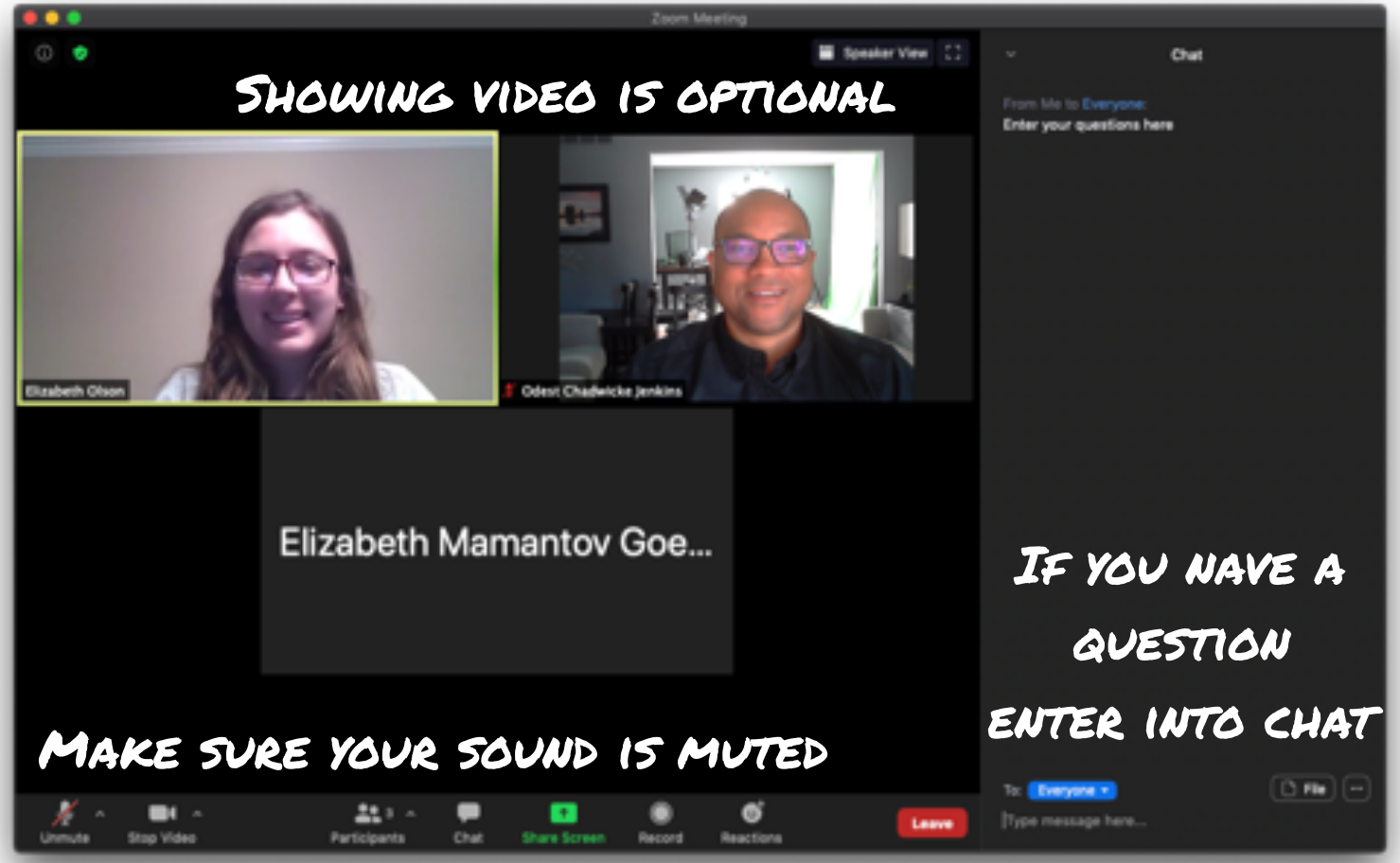
Initialization

EECS 367
Intro. to Autonomous Robotics

ROB 511
Robot Operating Systems

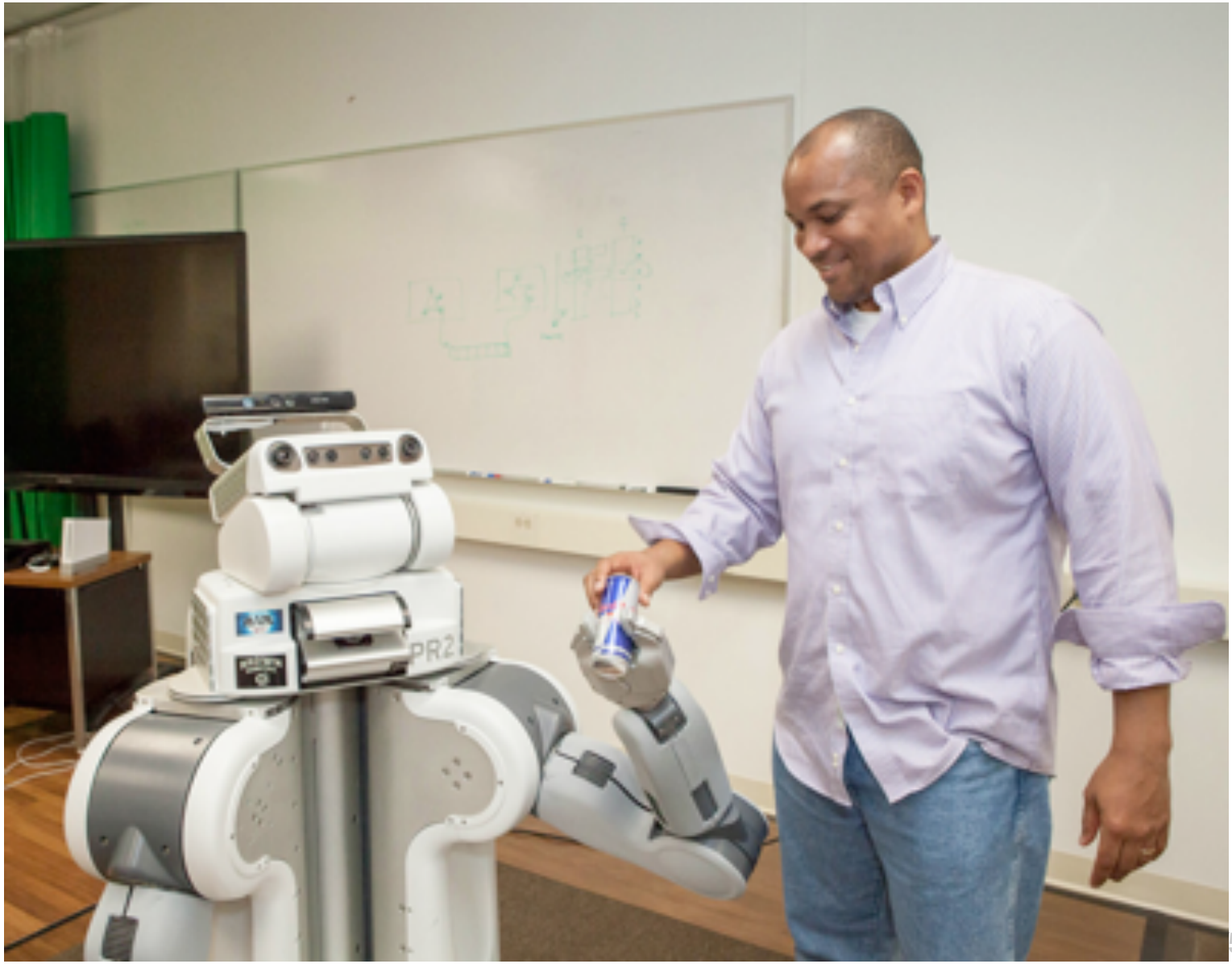
Fall 2020

Zoom etiquette



Today

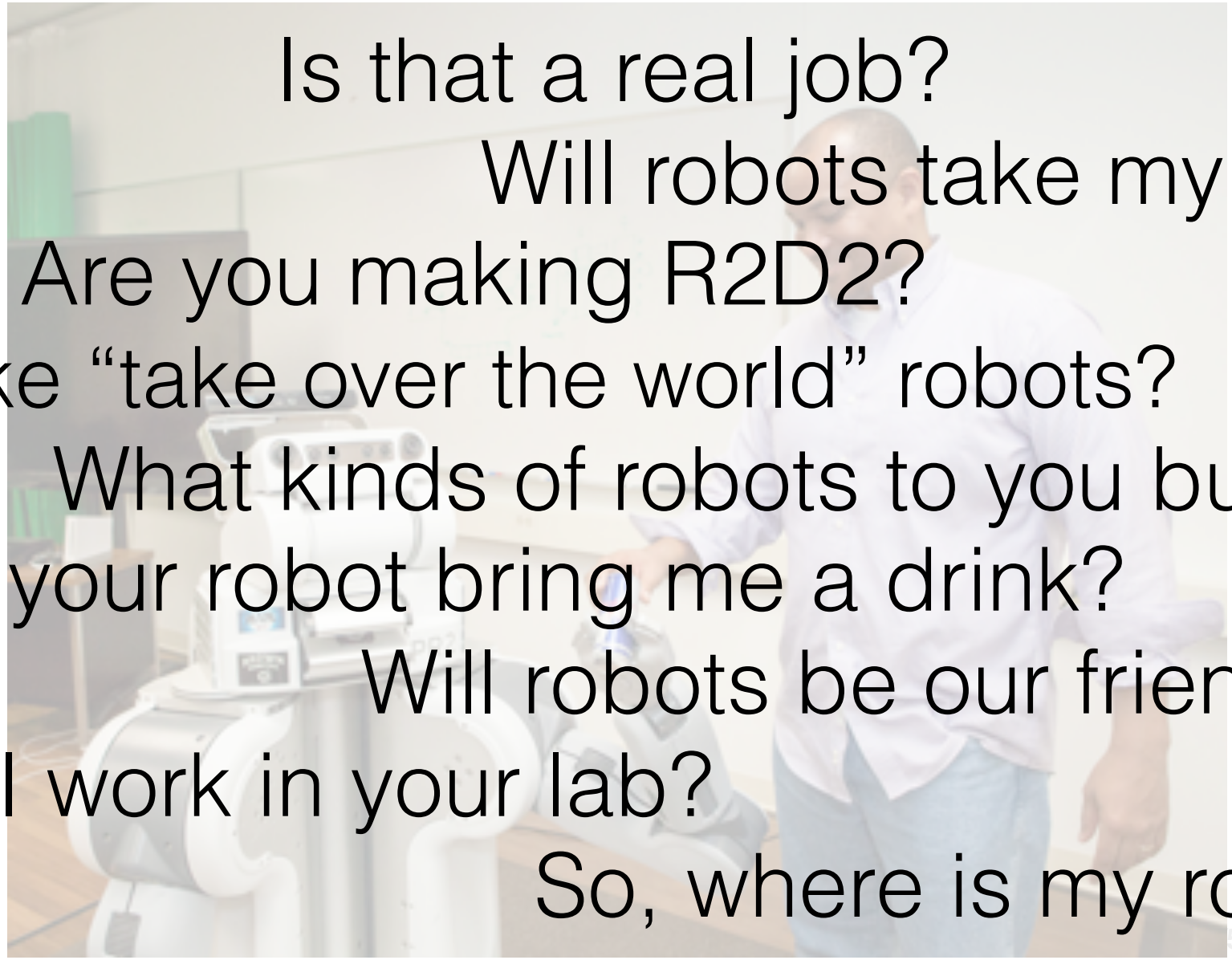
- So, where is my robot?
- Course administrative overview
- Assignment 1 (Path Planning) assigned today, due Sep 16 11:59pm
  - Assignment concepts will be covered in recorded lectures
- Action items: what I need from you now
  - Student workflow survey, Join AutoROB MS Team, Pod assignment


PR2

Is that a real job?

Is that a real job?

Yes.



Is that a real job?

Will robots take my job?

Are you making R2D2?

Like “take over the world” robots?

What kinds of robots to you build?

Can your robot bring me a drink?

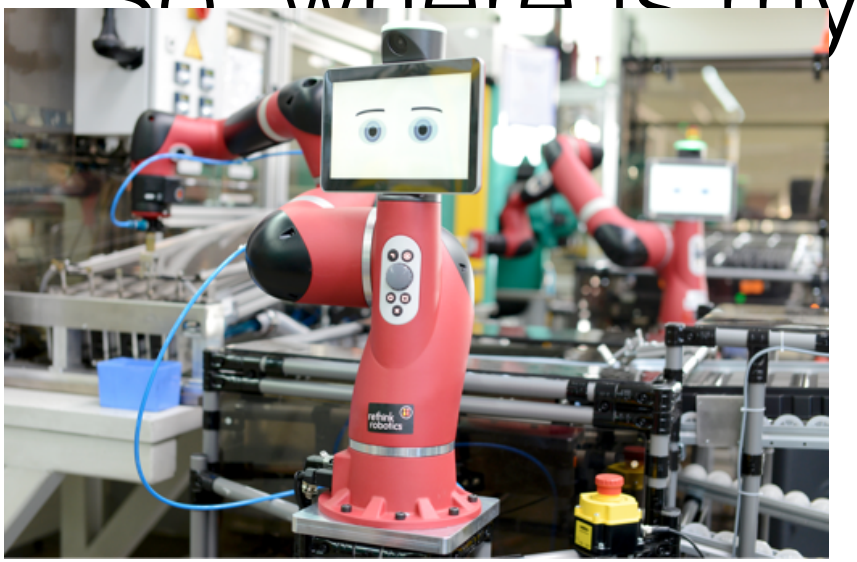
Will robots be our friends?

Can I work in your lab?

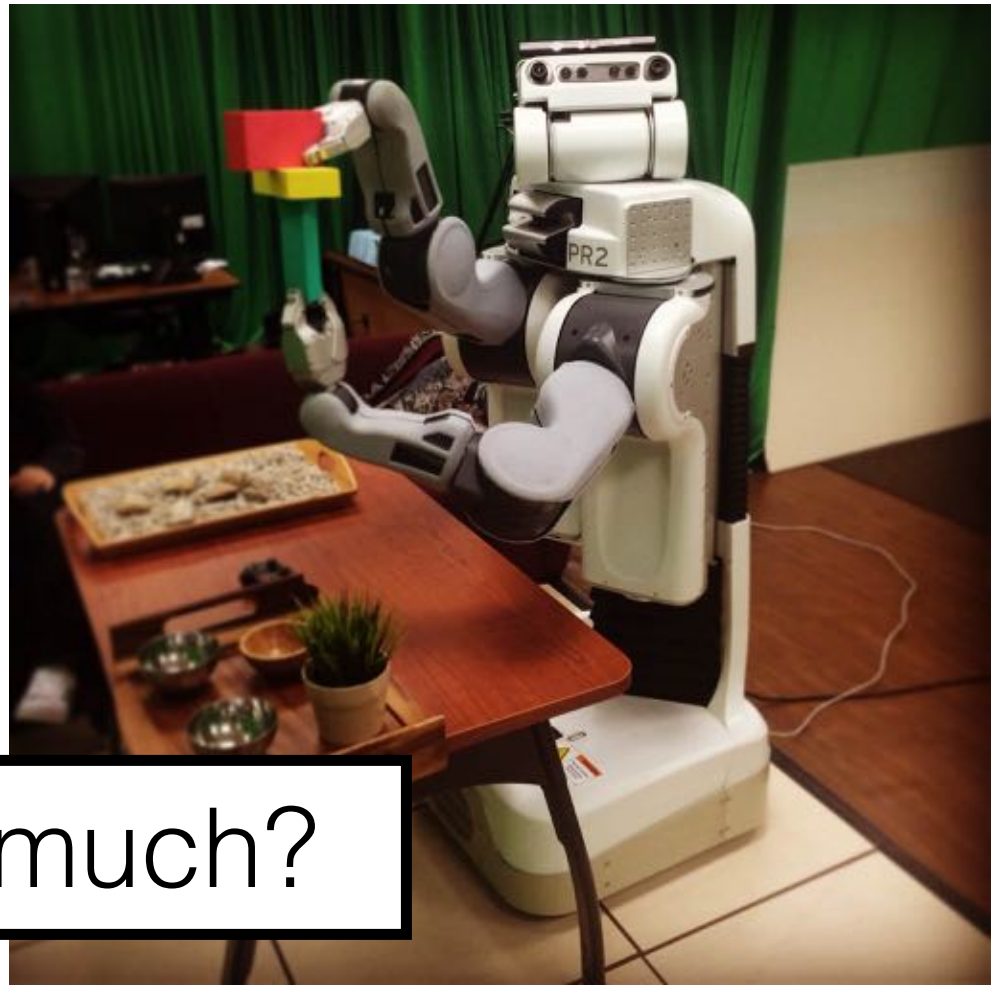
So, where is my robot?

So, where is my robot?

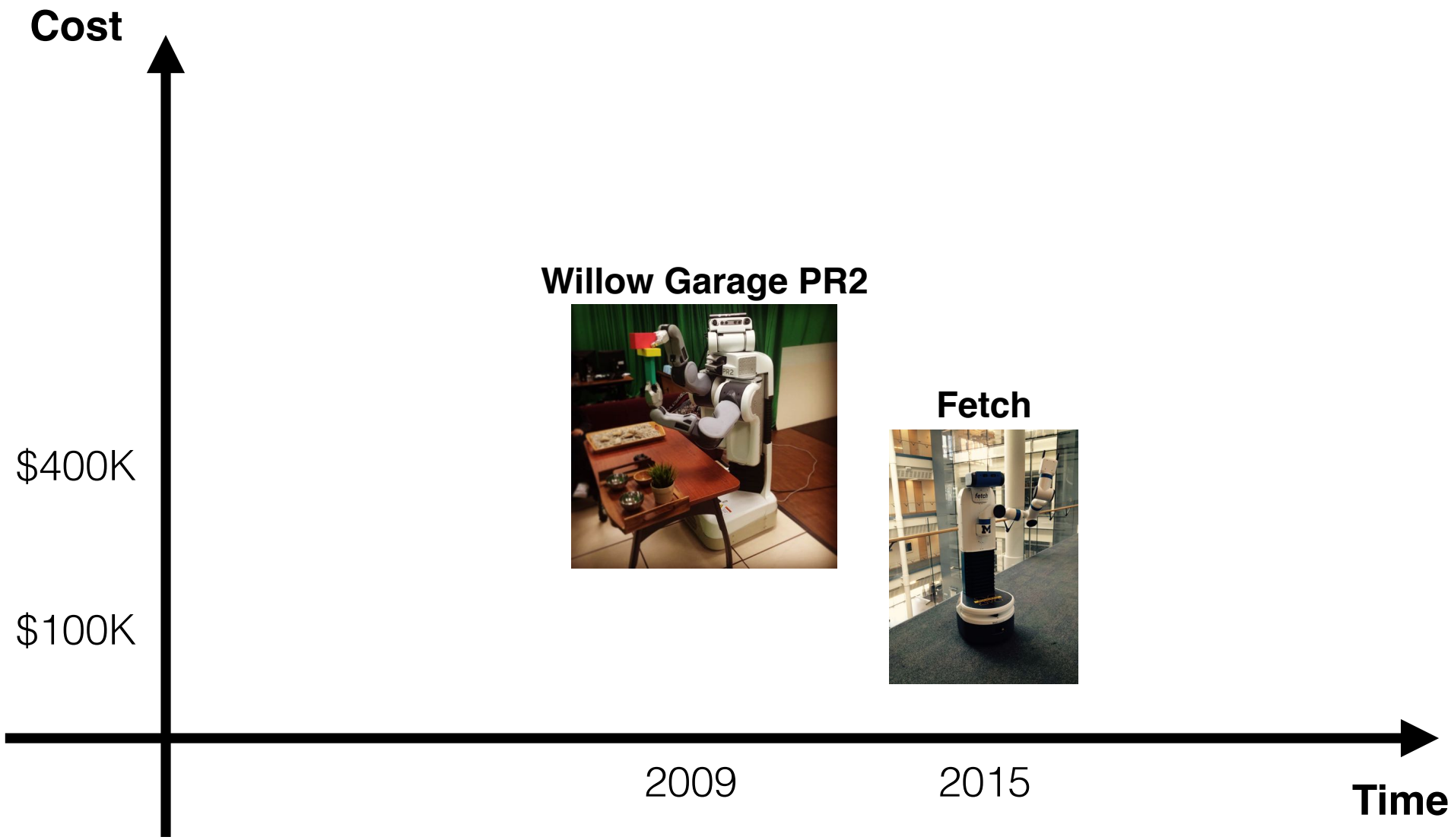
So, where is my robot?



Mobile Manipulation Robots



How much?


Cost
Willow Garage PR2
Fetch
$400K
$100K
2009
2015
Time

Cost

2002

Willow Garage PR2

Fetch

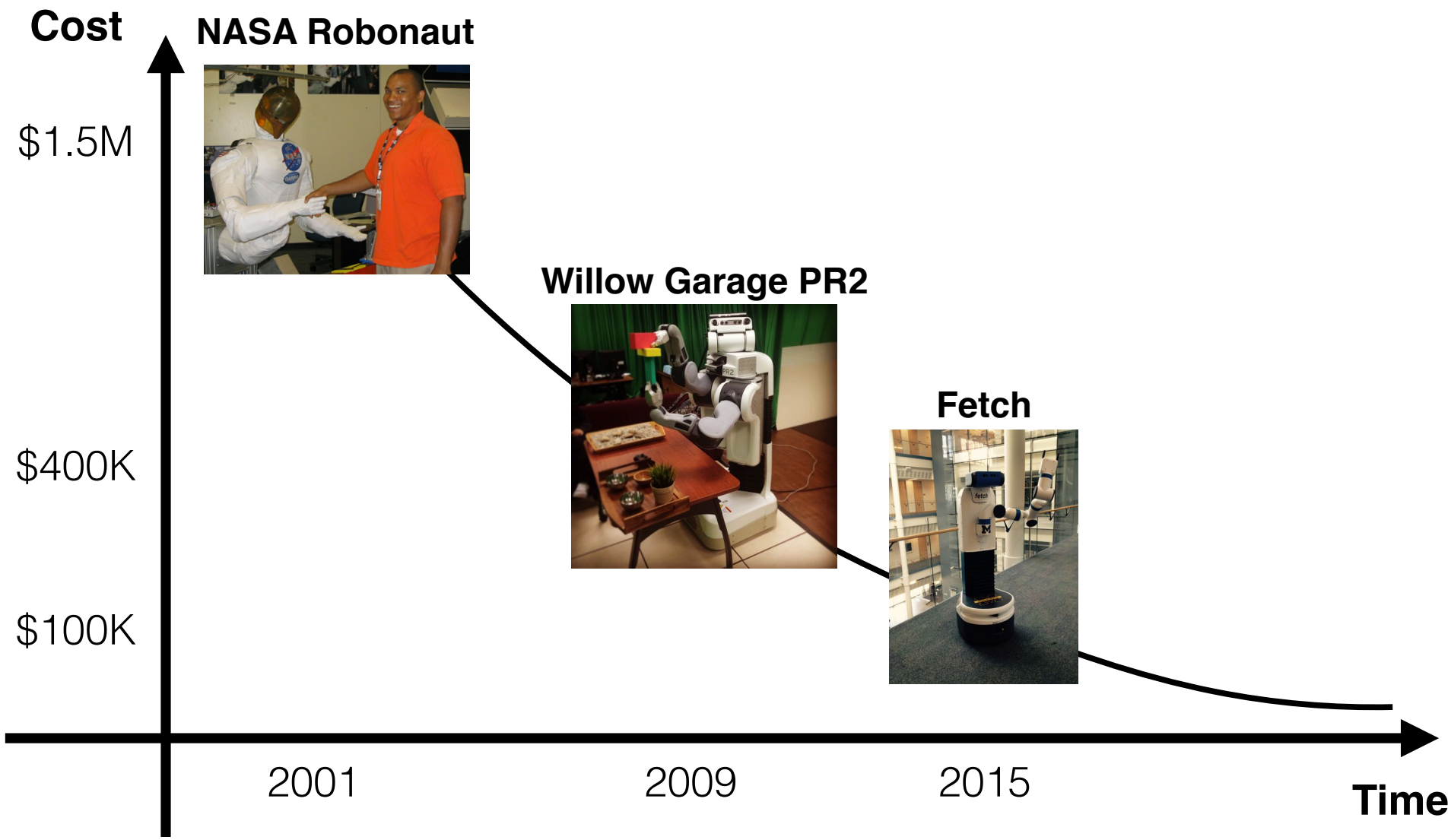
$400K

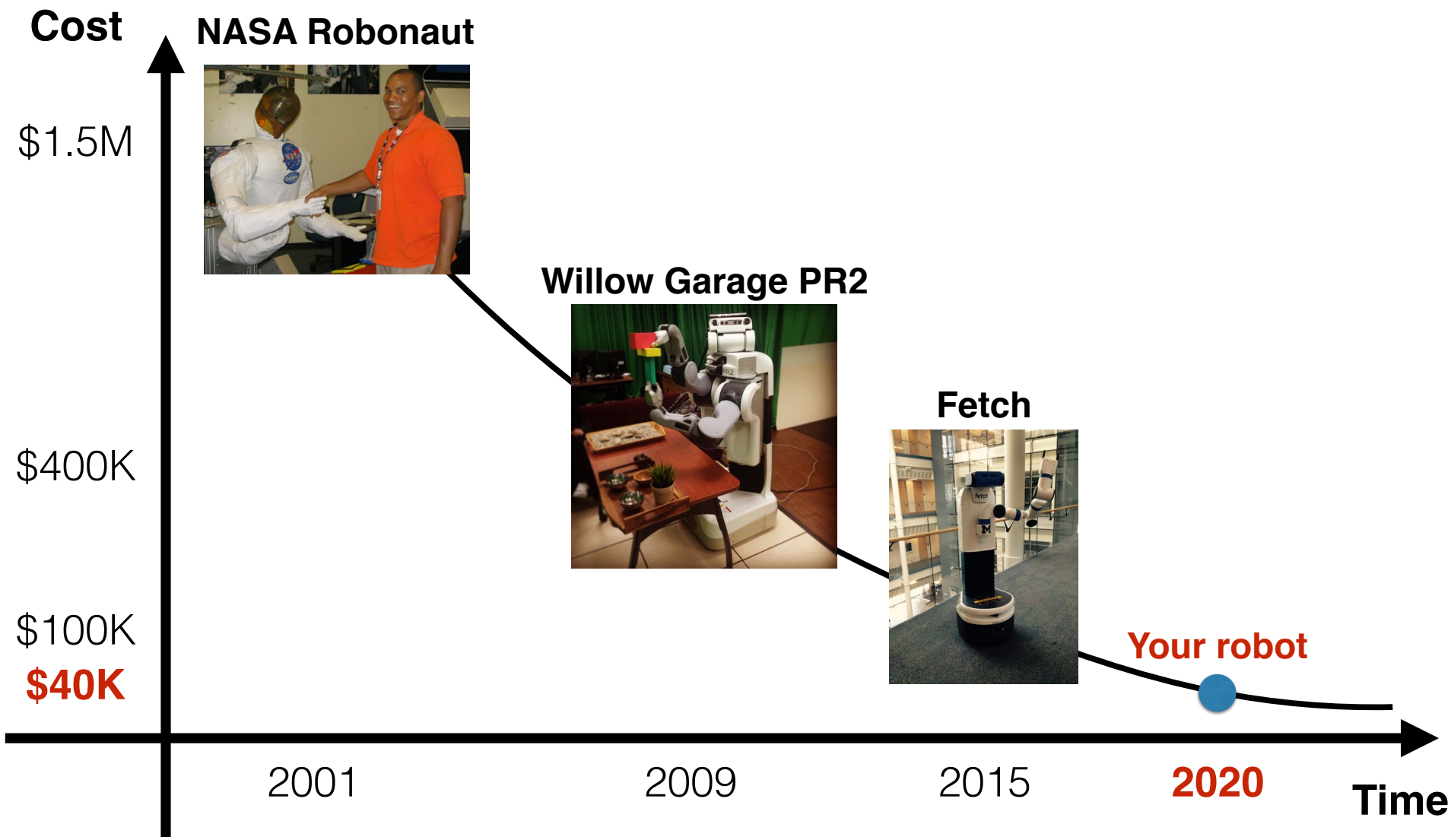
$100K

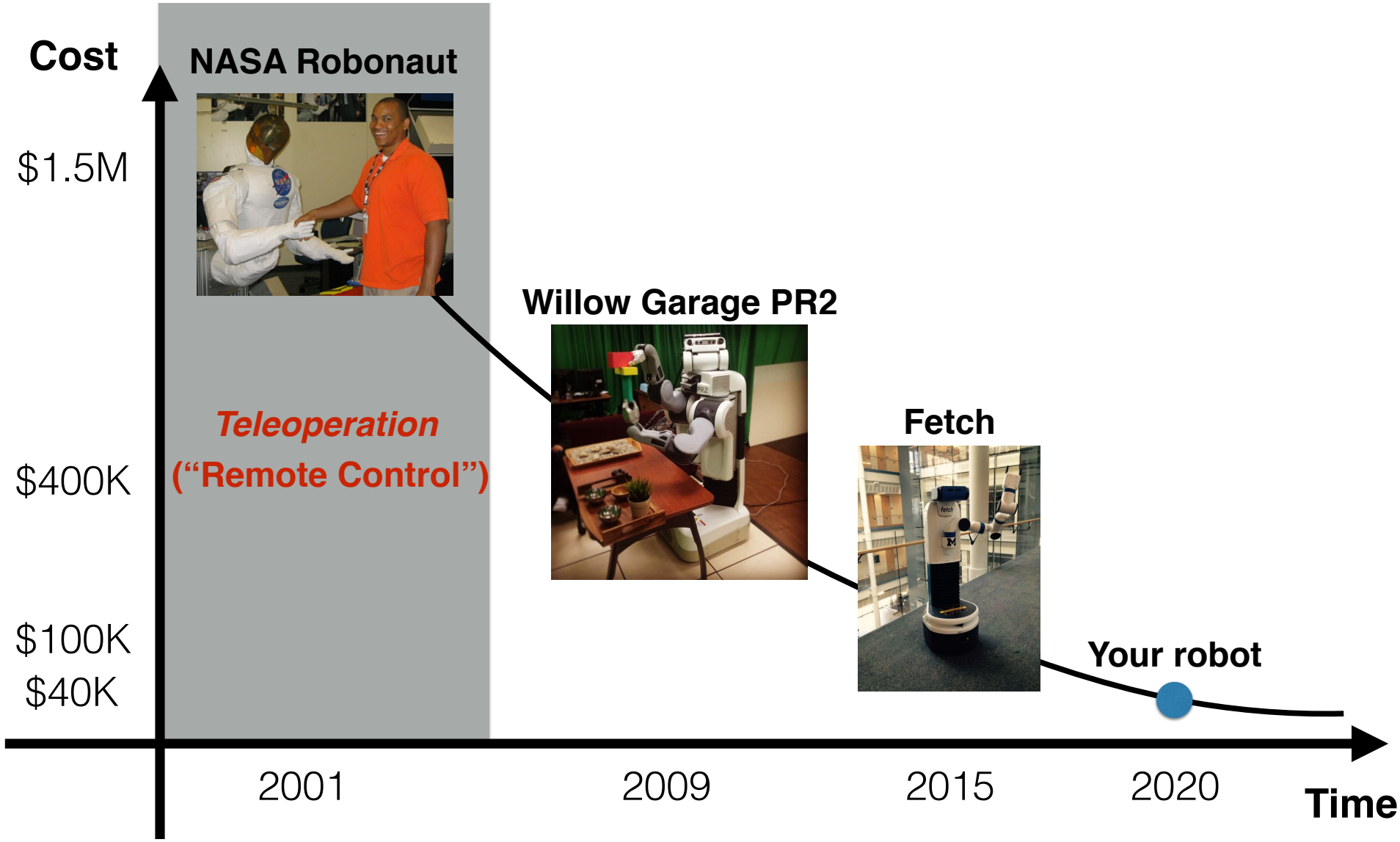
2009

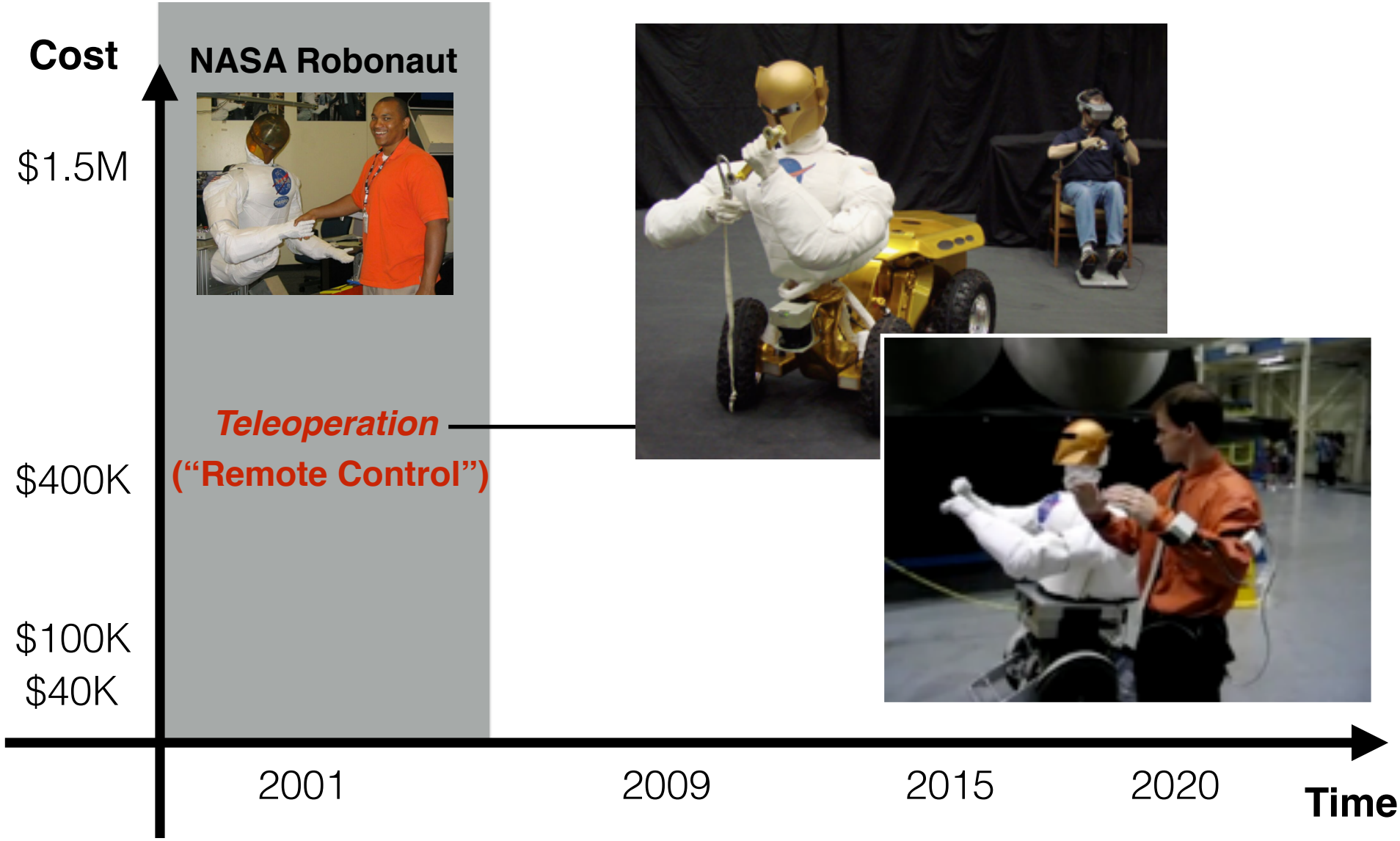
2015

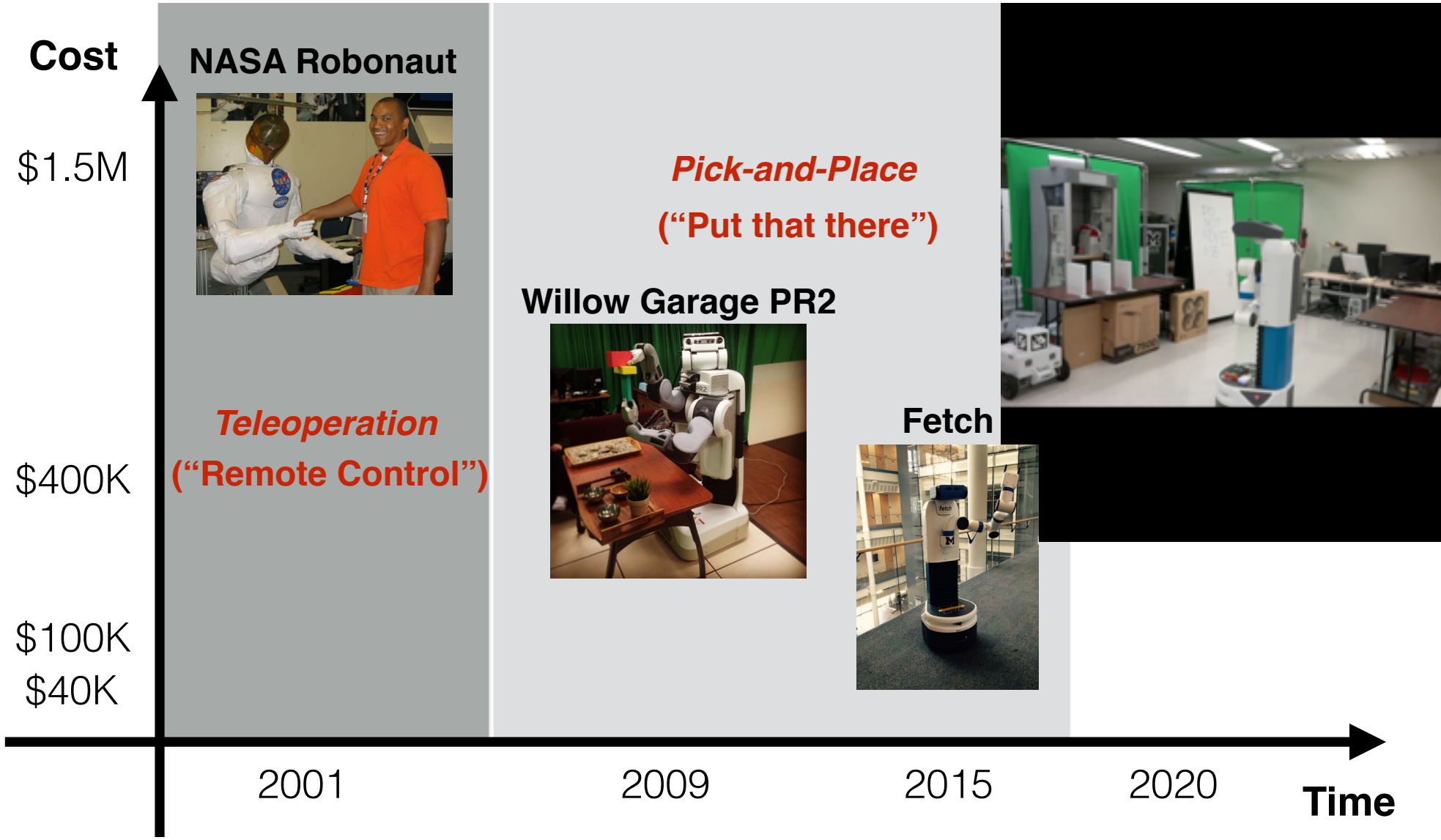
Time

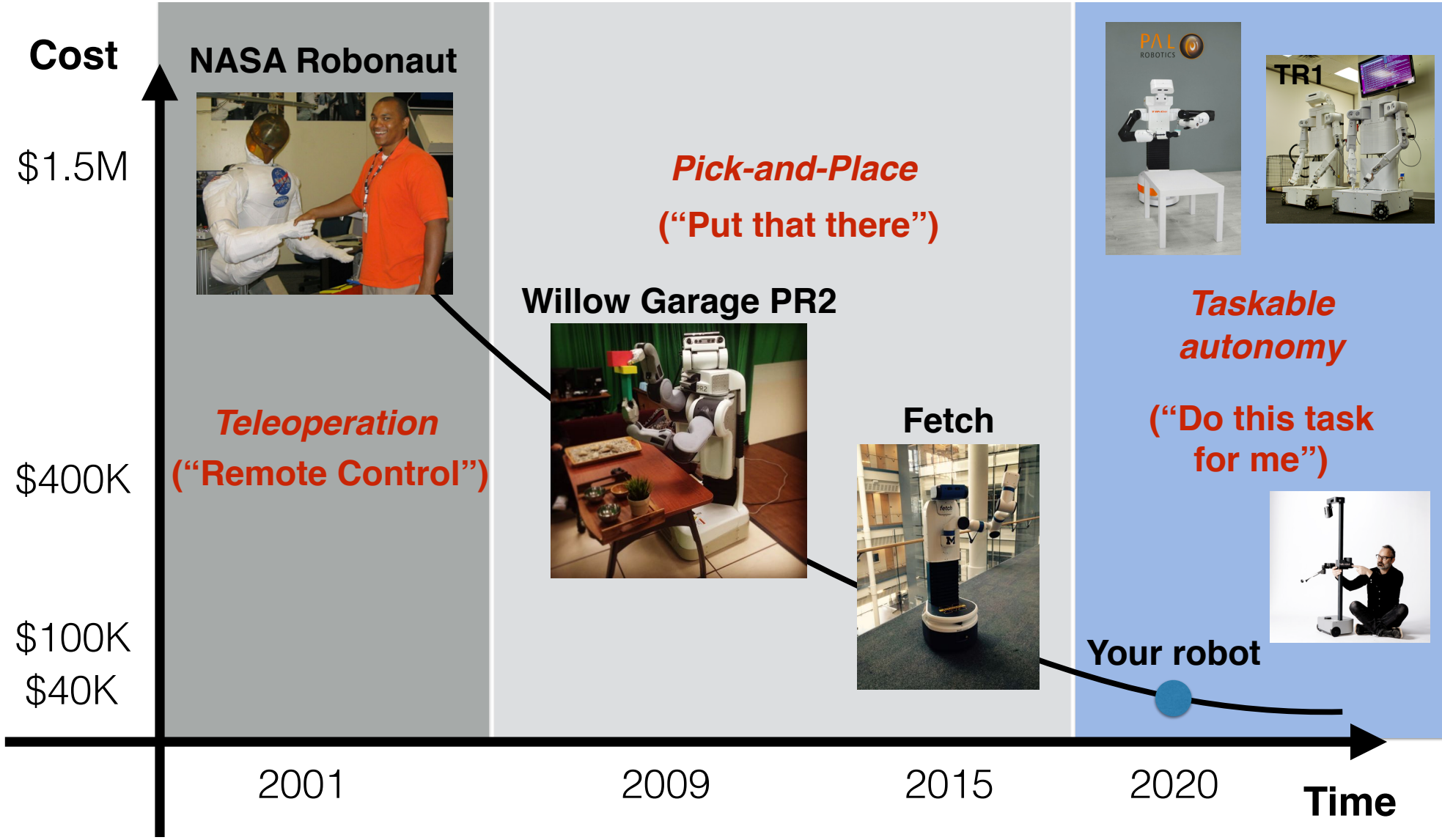

Cost
NASA Robonaut
$1.5M
Willow Garage PR2
$400K
Fetch
$100K
2001
2009
2015
Time


Cost
NASA Robonaut
$1.5M
Willow Garage PR2
$400K
Fetch
$100K
$40K
Your robot
2001
2009
2015
2020
Time


Cost
NASA Robonaut
$1.5M
Willow Garage PR2
Teleoperation
("Remote Control")
$400K
Fetch
$100K
$40K
Your robot
2001
2009
2015
2020
Time


Cost
NASA Robonaut
$1.5M
Teleoperation
("Remote Control")
$400K
$100K
$40K
2001
2009
2015
2020
Time

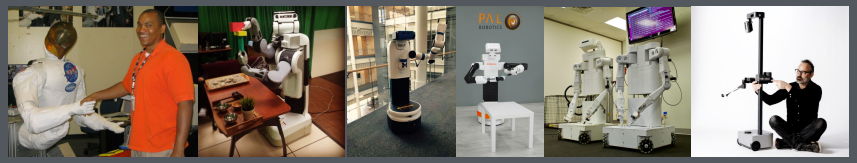

Cost
$1.5M
$400K
$100K
$40K
NASA Robonaut
Teleoperation
("Remote Control")
Pick-and-Place
("Put that there")
Willow Garage PR2
Fetch
2001
2009
2015
2020
Time

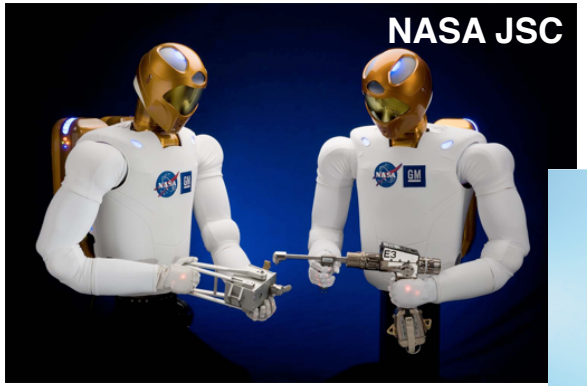

Cost
$1.5M
$400K
$100K
$40K
NASA Robonaut
Teleoperation
("Remote Control")
Pick-and-Place
("Put that there")
Willow Garage PR2
Fetch
Your robot
Taskable
autonomy
("Do this task
for me")
TR1
2001
2009
2015
2020
Time

Taskable autonomy

Pick-and-Place

Teleoperation



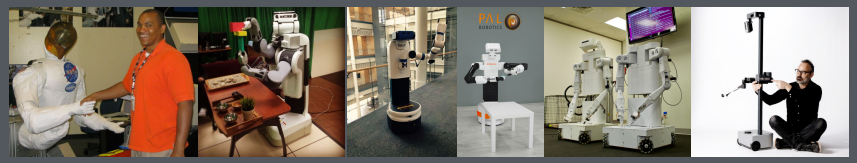


Dexterous Manipulation

Taskable autonomy

Dexterous Manipulation

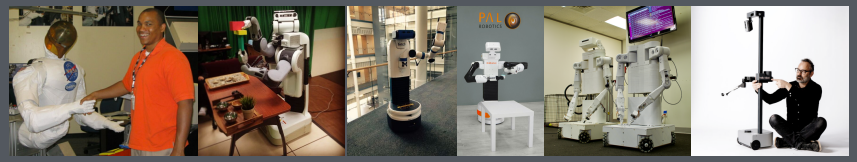
Teleoperation



Taskable autonomy

Dexterous Manipulation

Teleoperation



Operating system

From Wikipedia, the free encyclopedia

An **operating system** (**OS**) is system software that manages computer hardware, software resources, and provides common services for computer programs.

Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources.

For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware,[1][2] although the application code is usually executed directly by the hardware and frequently makes system calls to an OS function or is interrupted by it. Operating systems are found on many devices that contain a computer – from cellular phones and video game consoles to web servers and supercomputers.

The dominant desktop operating system is Microsoft Windows with a market share of around 82.74%. macOS by Apple Inc. is in second place (13.23%), and the varieties of Linux are collectively in third place (1.57%).[3] In the mobile sector (including smartphones and tablets), Android's share is up to 70% in the year 2017.[4] According to third quarter 2016 data, Android's share on smartphones is dominant with 87.5 percent with also a growth rate of 10.3 percent per year, followed by Apple's iOS with 12.1 percent with per year decrease in market share of 5.2 percent, while other operating systems amount to just 0.3 percent.[5] Linux distributions are dominant in the server and supercomputing sectors. Other specialized classes of operating systems, such as embedded and real-time systems, exist for many applications.

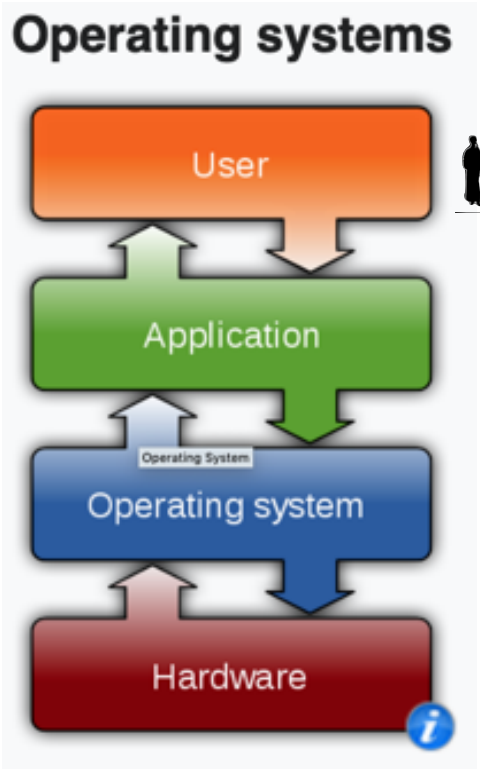
Contents [hide]

Taskable autonomy

Dexterous Manipulation

Teleoperation

An **operating system (OS)** is system software that manages computer hardware, software resources, and provides common services for computer programs.



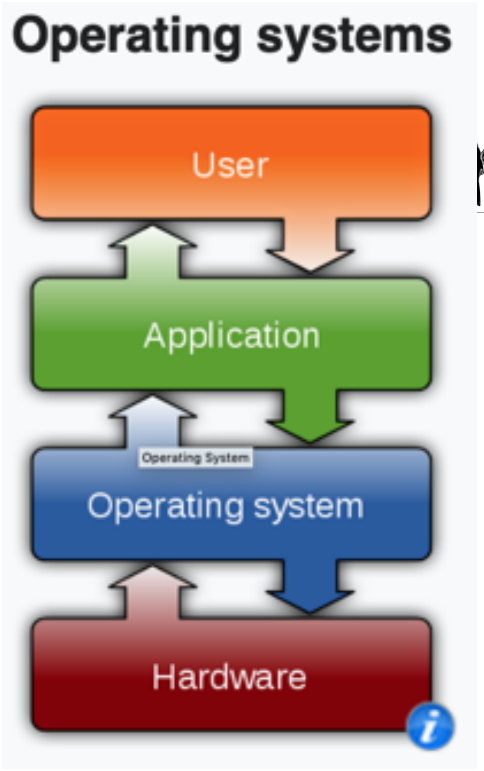
Taskable autonomy

Dexterous Manipulation

Teleoperation

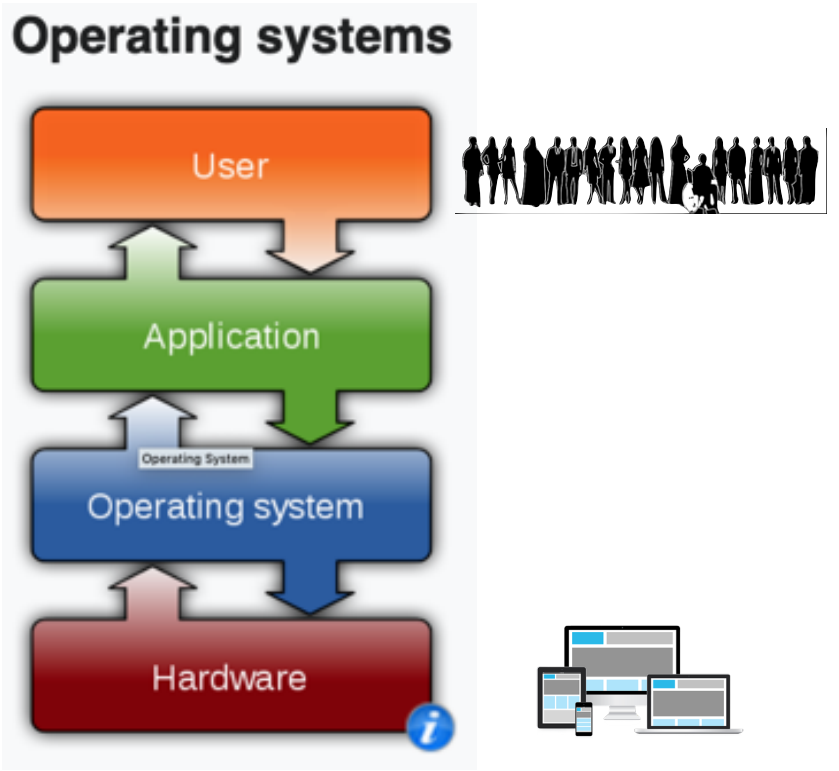
An **operating system (OS)** is a special program that runs on the bare machine and hides the gory details of managing processes and devices.

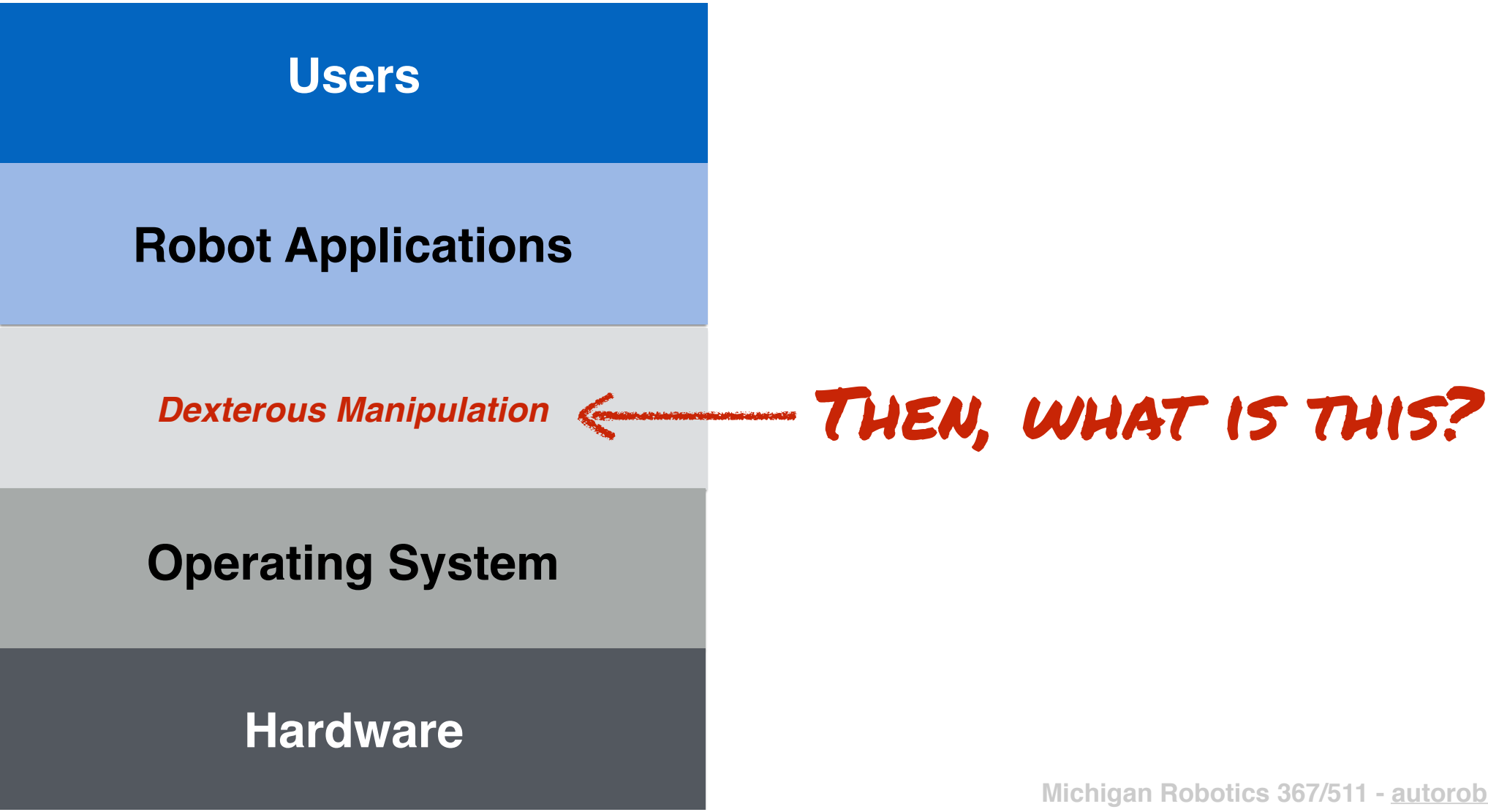
- https://perldoc.perl.org/perlglossary.html#operating-system



An **operating system (OS)** is
a special program that runs on the bare machine and hides the gory details of managing processes and devices.

- https://perldoc.perl.org/perlglossary.html#operating-system




Users
Robot Applications
Dexterous Manipulation
Operating System
Hardware
Then, what is this?

Users

Robot Applications

Robot Operating System

Operating System

Hardware

A **robot operating system (robot OS)** is a special program that runs on the operating system and hides the gory details of controlling robot devices, autonomy processes, and sensorimotor routines.

This abstraction provides a platform for robot applications to run seamlessly across a wide variety of robots capable of mobility and/or dexterous manipulation.

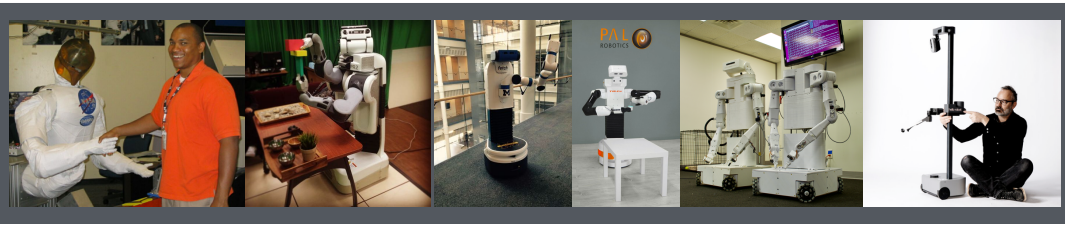
Users

Robot Applications

Robot Operating System

Operating System

Hardware



Users

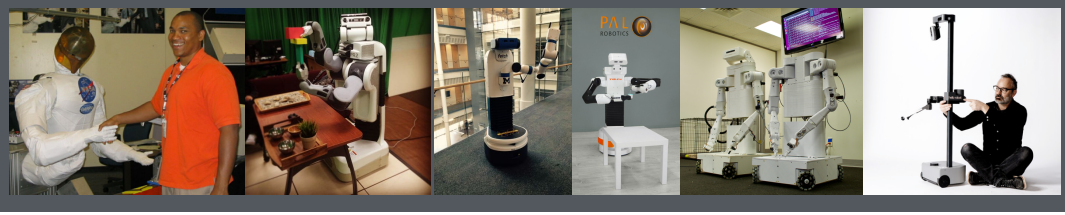
Robot Applications

Robot Operating System

Operating System




Hardware



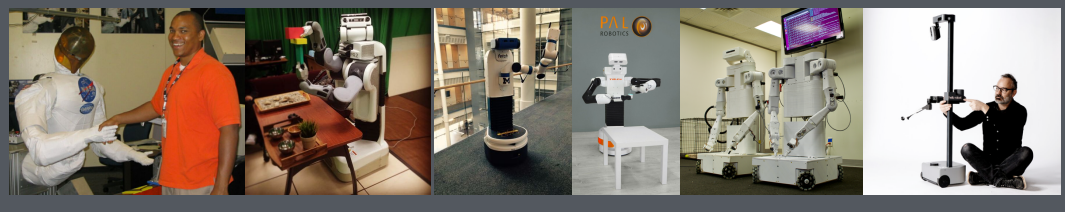
Users

Robot Applications

Robot Operating System

Operating System

Hardware



Users

Robot Applications

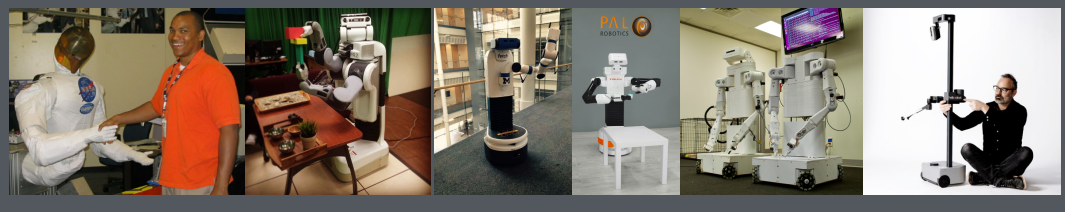
Robot Operating System

LCM ROS Player YARP MOOS

Operating System




Hardware



Users

Robot Applications

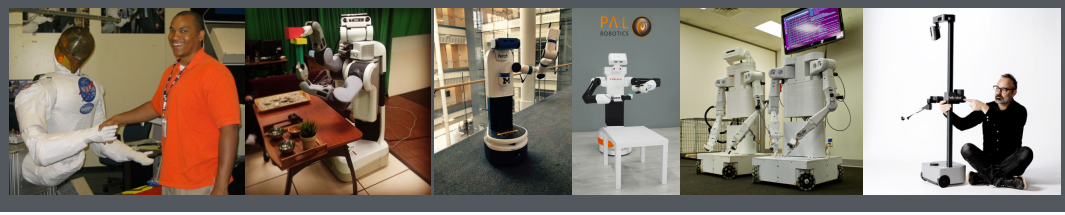
THEN, WHAT IS THIS?

Robot Operating System

LCM ROS Player YARP MOOS10

Operating System

Hardware



Users

Robot Applications

Someday in the Future…

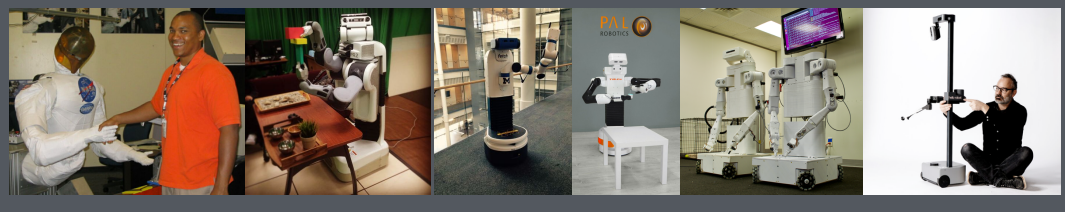
"Do this task for me"

Robot Operating System

LCM ROS Player YARP MOOS

Operating System

Hardware



“Do this task for me”

Can we make your
world programmable ?

Can we make a **robot app store** ?

Willow Garage

What's a robot app?

- In the near future
- Eventually:
  - CleanTheHouse
  - PatrolTheBuilding
  - ...
- For now:
  - demonstrations
  - experiments
  - challenge entries (!)

https://www.ros.org/presentations/2009-07_Gerkey_IJCAI-robotics-workshop.pdf

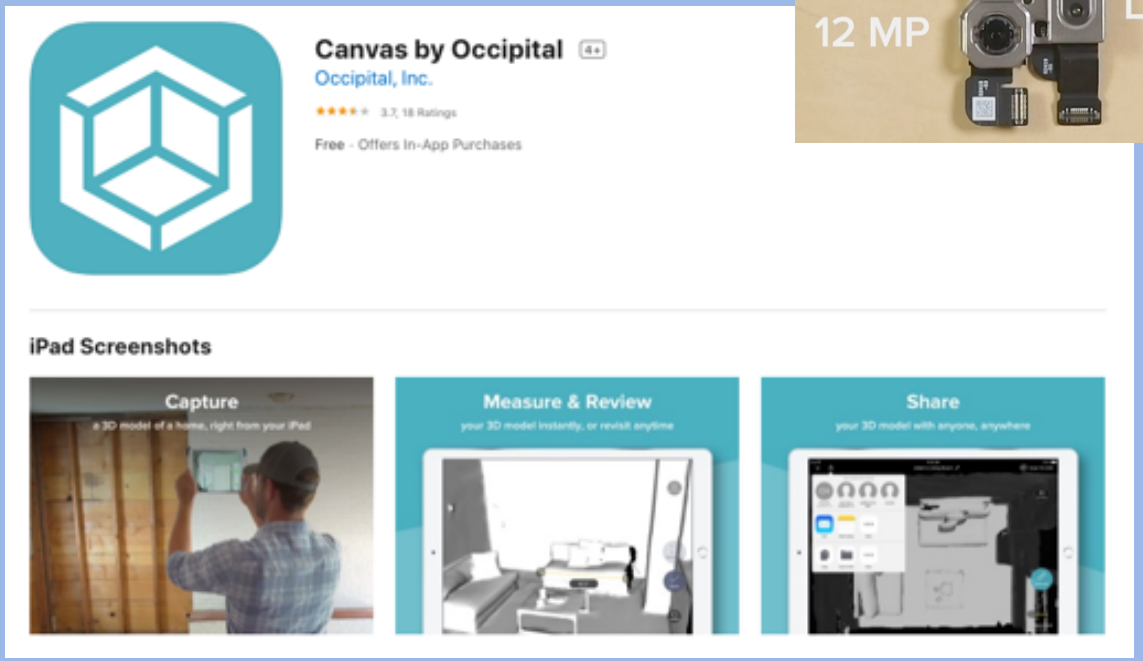
2009

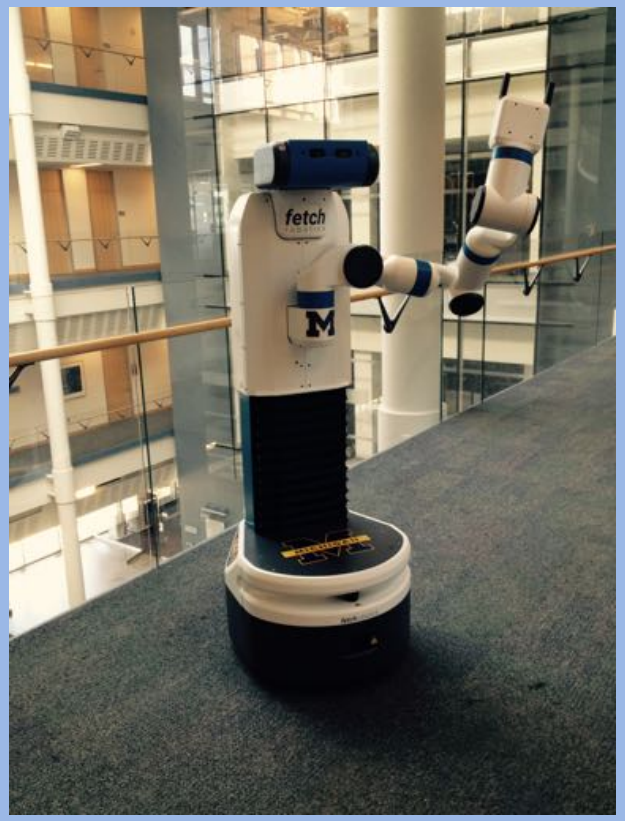

MapIt!
Autonomous exploration and mapping for any indoor environment.
Click to Buy ($49.99)

2020

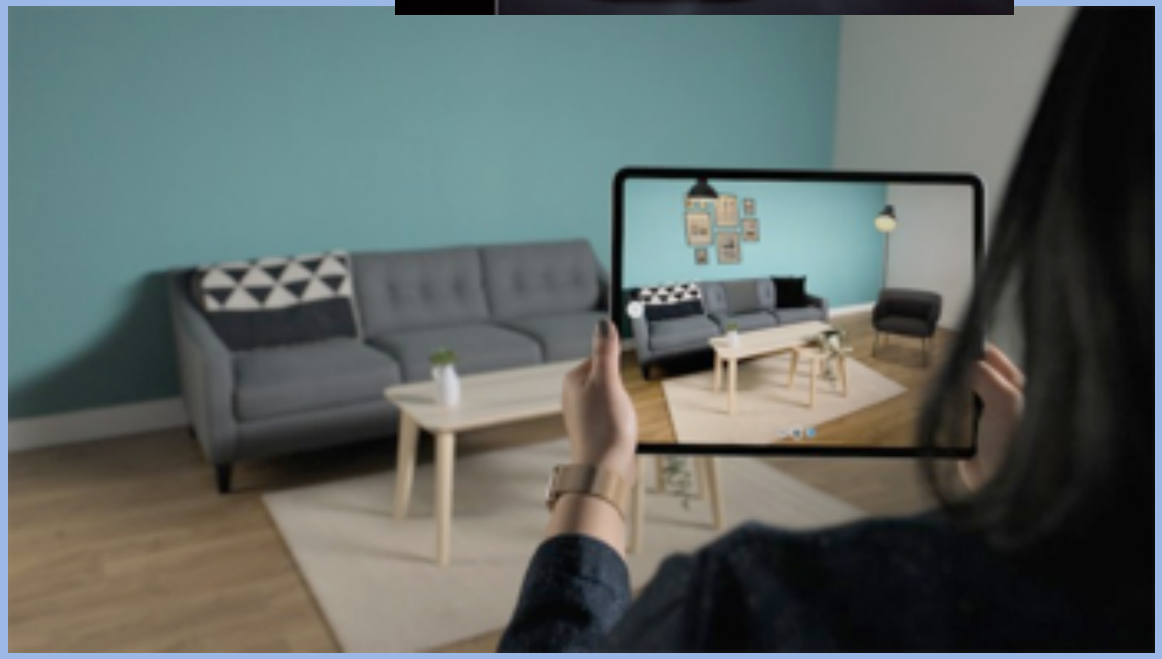

LiDAR Scanner

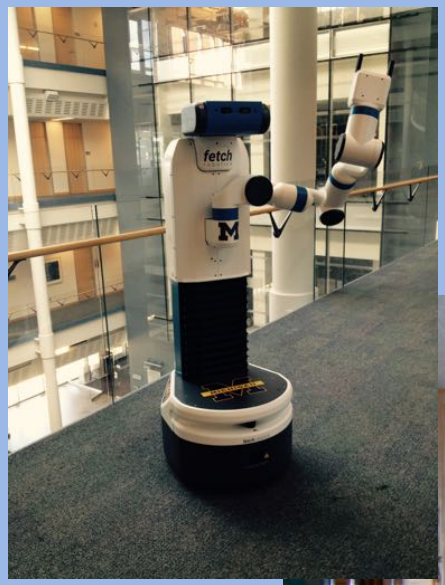

10 MP
12 MP
LiDAR


Canvas by Occipital
4+
Occipital, Inc.
3.7, 18 Ratings
Free · Offers In-App Purchases
iPad Screenshots
Capture
a 3D model of a home, right from your iPad
Measure & Review
your 3D model instantly, or revisit anytime
Share
your 3D model with anyone, anywhere

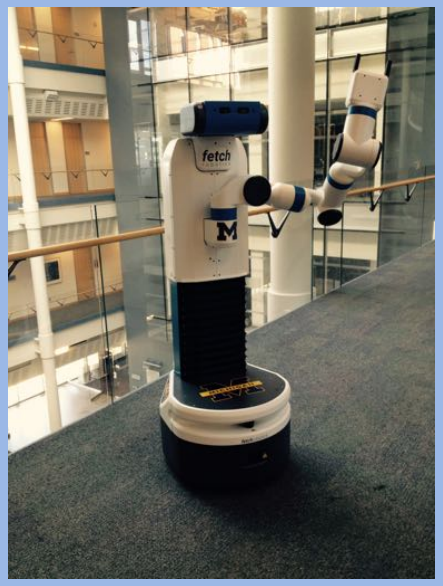

fetch


LiDAR Scanner



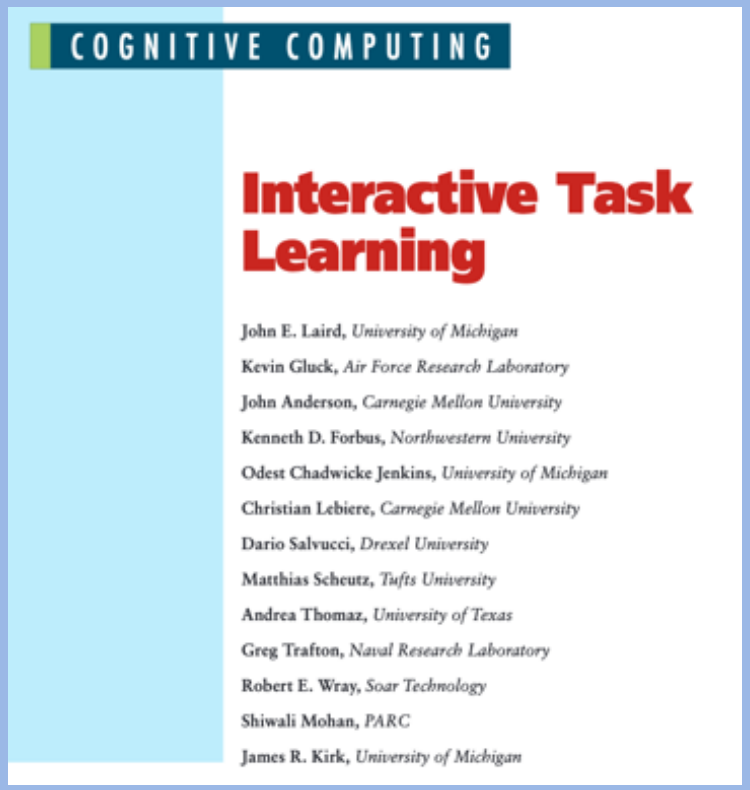

fetch

Use any robot x



to perform any task y

in any environment z



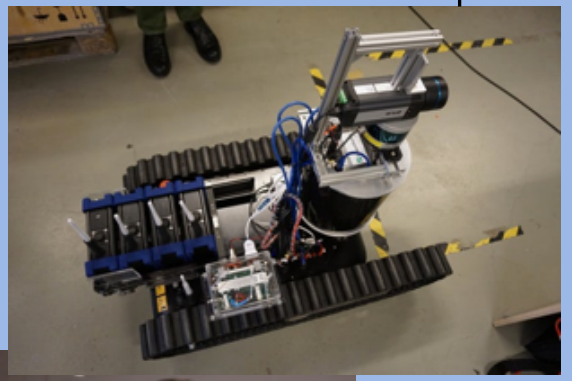
Use any robot x
to perform any task y
in any environment z

The 3Ds: Dirty, Dull, and Dangerous

"Autonomous" Driving

Infrastructure inspection

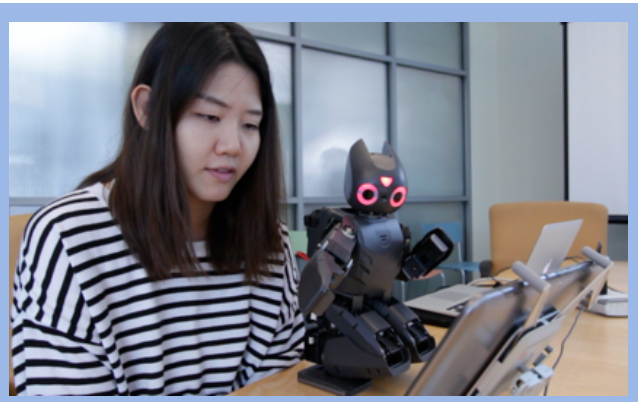
Nuclear cleanup



https://www.shadowrobot.com/blog/robots-saving-humans-from-dangerous-jobs/
https://techcrunch.com/2018/06/05/remote-control-driverless-car-startup-partners-with-vehicle-manufacturers/

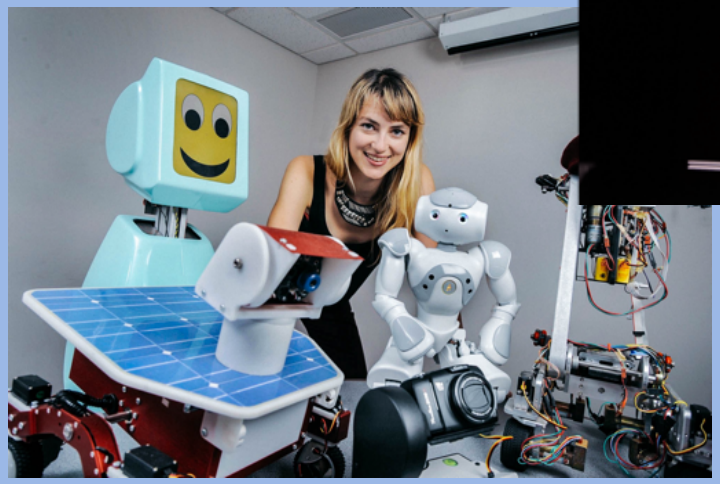
Social Robotics

Autism treatment



Education

Entertainment



Elder care

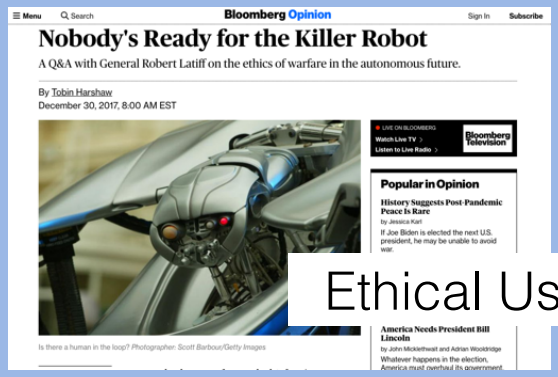
Agriculture

Exploration

Manufacturing

Security

Ethical Use


Bloomberg Opinion

Nobody's Ready for the Killer Robot

A Q&A with General Robert Latiff on the ethics of warfare in the autonomous future.

By Tobin Harshaw
December 30, 2017, 8:00 AM EST

Popular in Opinion

History Suggests Post-Pandemic Peace Is Rare

America Needs President Bill Lincoln

Lethal Force

Social Robotics

Dirty, Dull, Dangerous

Medicine

Users

Robot Applications

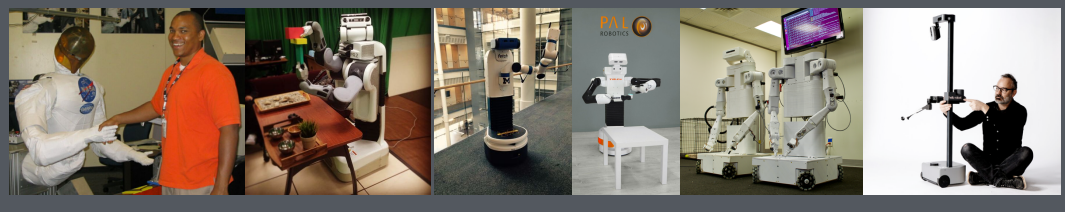
Custom applications,
Taskable autonomy research

Robot Operating System

LCM ROS Player YARP MOOS

Operating System

Hardware



Users

Robot Applications

Custom applications,
Taskable autonomy research

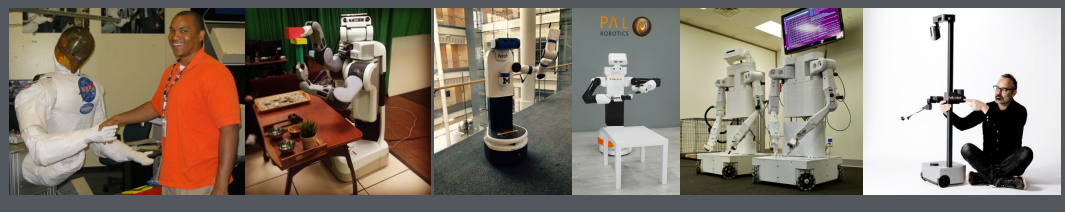
Robot Operating System

LCM ROS Player YARP MOOS10

Operating System




Hardware



Users

Robot Applications

Custom applications,
Taskable autonomy research

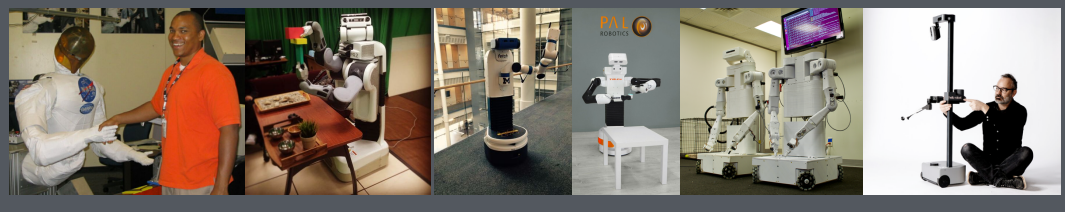
Robot Operating System

Build your own Robot OS

Operating System



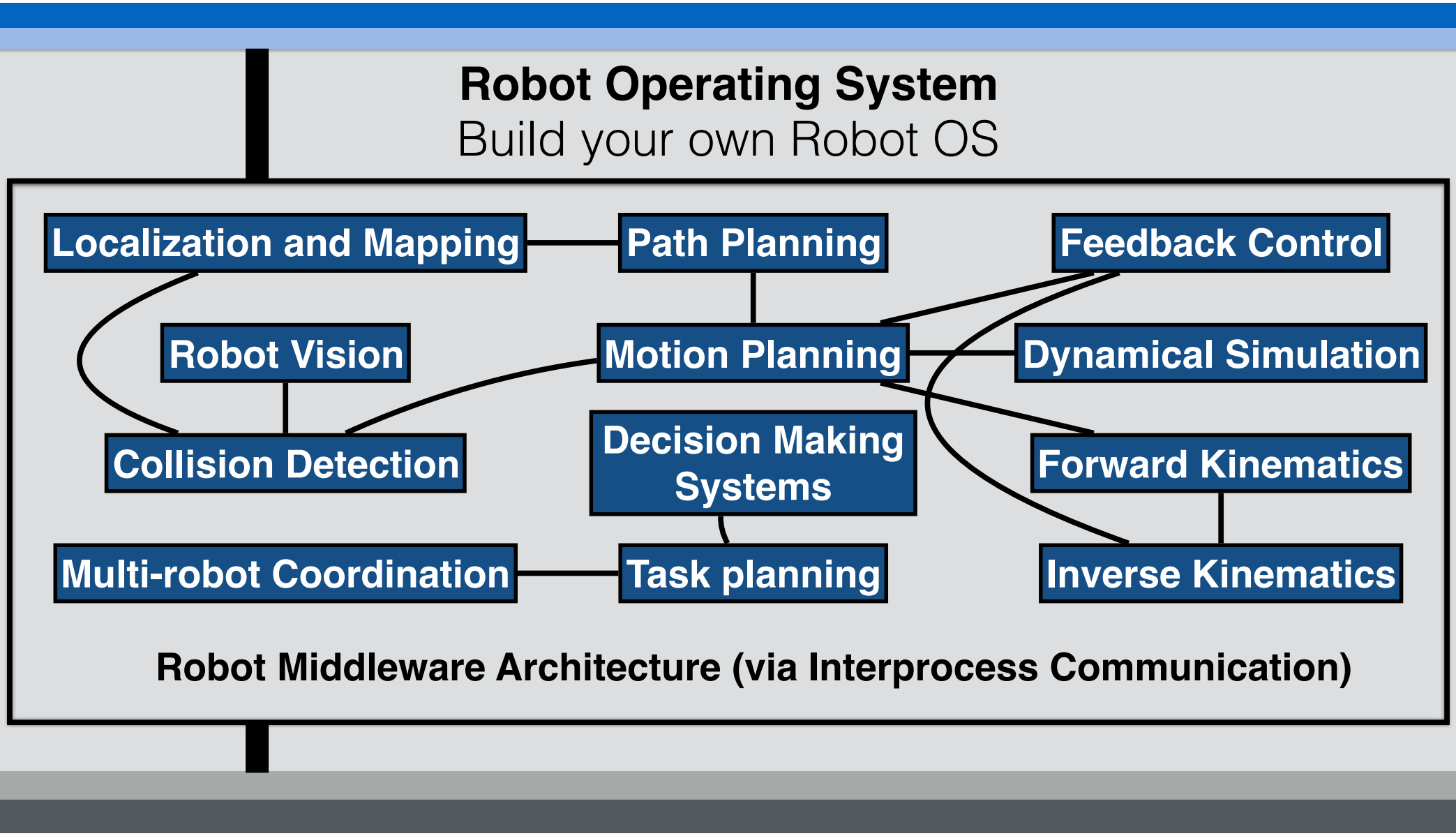

Hardware



Robot Operating System

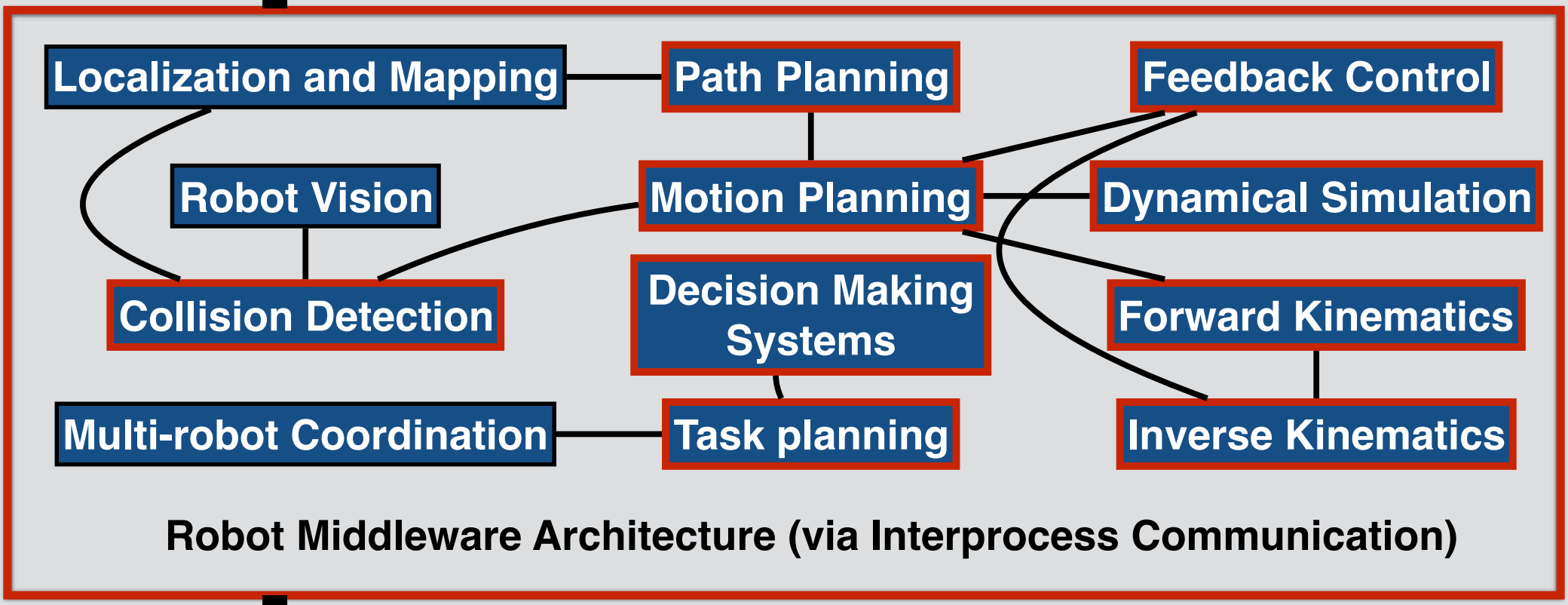
Build your own Robot OS

- Localization and Mapping
- Path Planning
- Feedback Control
- Robot Vision
- Motion Planning
- Dynamical Simulation
- Collision Detection
- Decision Making Systems
- Forward Kinematics
- Multi-robot Coordination
- Task planning
- Inverse Kinematics


Robot Operating System
Build your own Robot OS
Localization and Mapping
Path Planning
Feedback Control
Robot Vision
Motion Planning
Dynamical Simulation
Collision Detection
Decision Making Systems
Forward Kinematics
Multi-robot Coordination
Task planning
Inverse Kinematics
Robot Middleware Architecture (via Interprocess Communication)

Robot Operating System

COVERED AT BREADTH IN AUTOROB



Users

Robot Applications

Robot Operating System

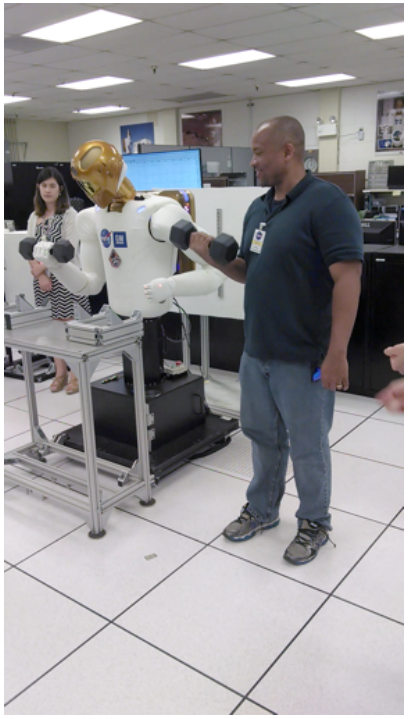
Operating System

Hardware

Work with a real robot once this semester

Course Staff

Instructor: Chad Jenkins (ocj)



Office hours

~~Beyster 3644~~

Mon. 2-5pm

Wed. 2-5pm

GSI
Xiaotong Chen
(cxt)

GSI
Lizzie Goeddel
(mamantov)

GSI
Zheming Zhou
(zhezhou)

https://autorob.org

AUTOROB schedule kinaval git assignments: 1 2 3 4 5 6 7

AutoRob

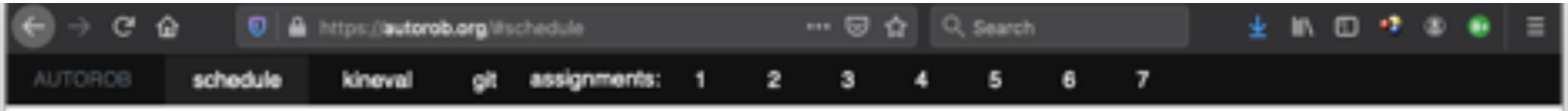
Introduction to Autonomous Robotics
Michigan EECS 367

Robot Operating Systems
Michigan ROB 511

Fall 2020

Flipped Classroom Hybrid COVID-19 Edition

Course website
http://autorob.org



Course Schedule (tentative and subject to change)

Preview slides from lectures during the Fall 2019 offering of AutoRob are provided. These preview slides will be replaced with recorded lectures for Fall 2020 as the videos become available.

Date	Topic	Reading	Project
Aug 31	Initialization: Course overview, Robotics roadmap, Path planning quick start What is a robot?: Brief history and definitions for robotics	Spong Ch.1 Corke Ch.1	Setup git repository Out: Path Planning
Sep 2	Path Planning: DFS, BFS, A-star, Greedy best first	Wikipedia	
Sep 4	367 Lab: Git-ing started with git, JavaScript, and KinEval		
	Week 2		
Sep 7	No course meeting - Labor Day		
Sep 9	JavaScript and AutoRob workflow: Project workflow with git, JS/HTML5, Document Object Model, Data Structures in JS, Animation in JS/H5, Version Control, LaTeX math mode, Licensing, Michigan Honor License	Crockford, HTML Sandbox, hello.html, JavaScript by Example, hello_anim, hello_anim_text	

AutoRob can be done remotely

AutoRob can be done remotely

(in-person needs handled on a case-by-case basis)

https://autorob.org

Search

AUTOROB | schedule | kinaval | git | assignments: 1 2 3 4 5 6 7

AutoRob

Introduction to Autonomous Robotics
Michigan EECS 367

Robot Operating Systems
Michigan ROB 511

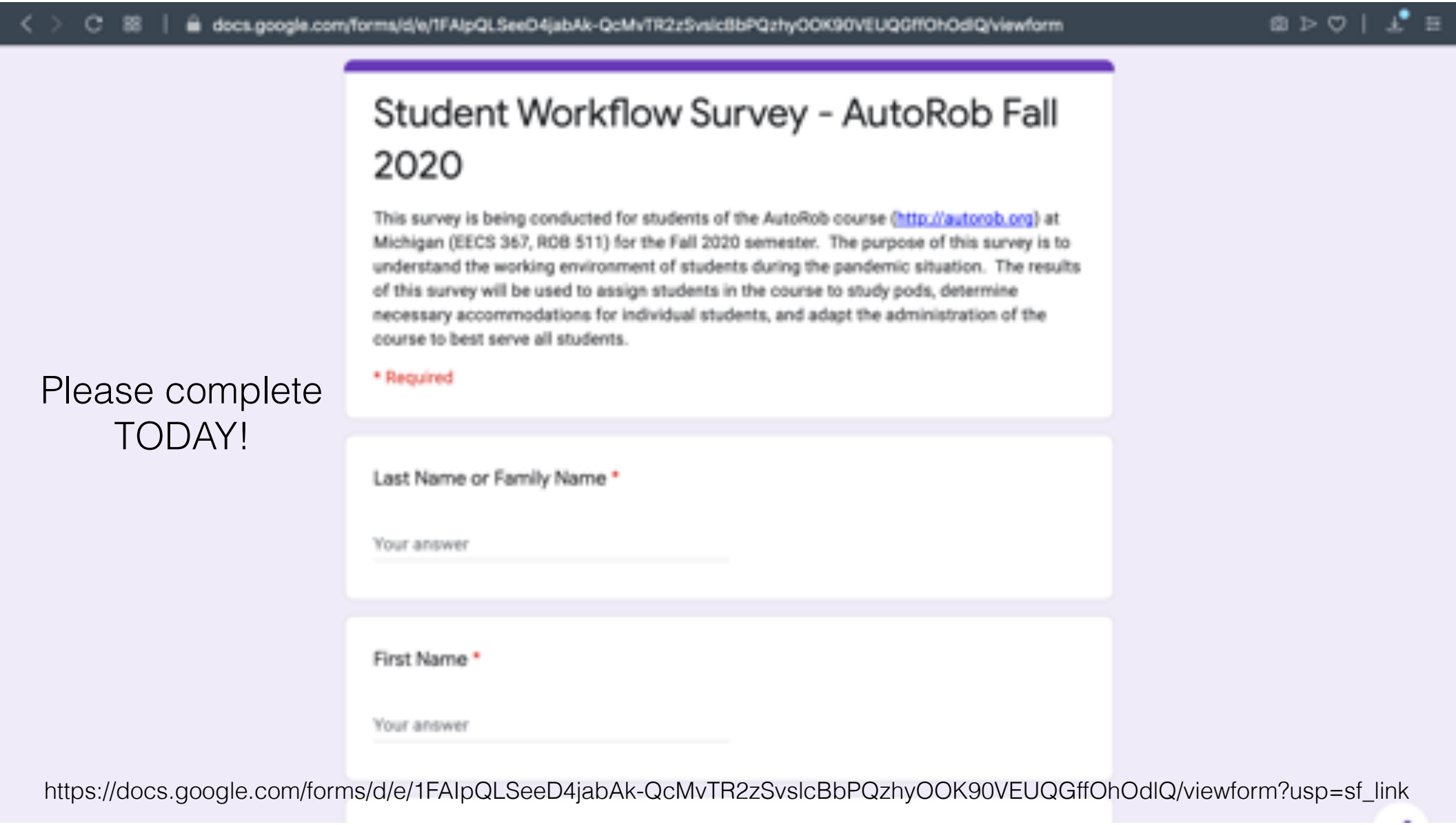
Fall 2020

Flipped Classroom Hybrid COVID-19 Edition

Flipped Classroom Format

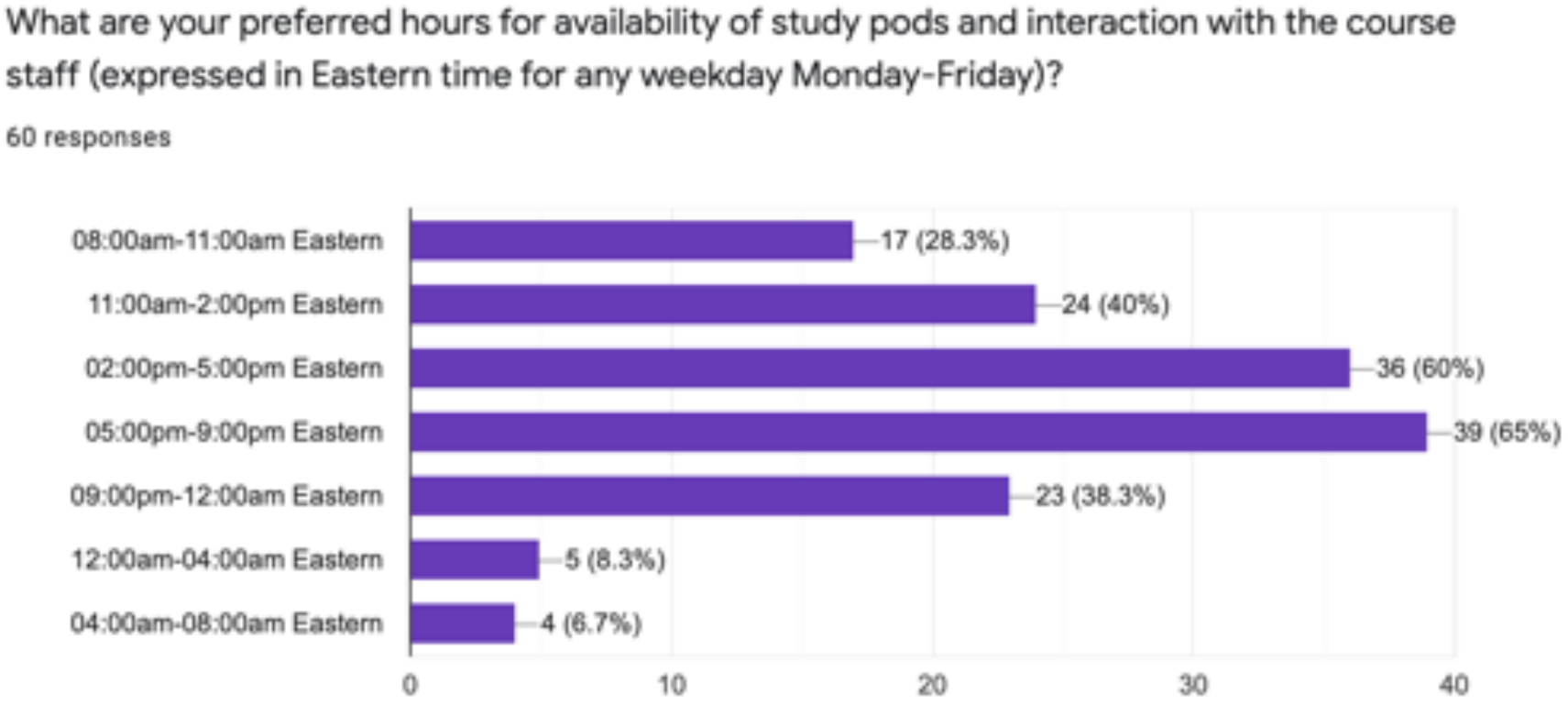
- Lectures will be recorded and available online (autorob.org)
- Course Zoom link: https://umich.zoom.us/j/97911142537
- All-Class Interactive Session MW 1:30-2pm Michigan Mean Time
  - General issues, Q&A
- 367 Lab Section: F 2:30-3:20 MMT
  - Walkthrough of projects
- Interactive study pods
  - Clusters of 5 students led by a GSI
  - Assignments and schedules to be done by this Wednesday

Student Workflow Survey



Please complete TODAY!

https://docs.google.com/forms/d/e/1FAIpQLSeeD4jabAk-QcMvTR2zSvslcBbPQzhyOOK90VEUQGffOhOdlQ/viewform?usp=sf_link



as of 9am Eastern this morning

Fall 2020 Course Format

The AutoRob course will have a flipped classroom hybrid format this Fall semester. At their discretion, a student will be able to complete the course remotely in its entirety. Course meetings, quizzes, and office hours will be held in-person only as needed and with consideration of the public health situation. All lectures will be pre-recorded and available online through this course website. The course staff will be available to students through regularly scheduled all-class interactive sessions twice per week, a laboratory section once per week, and office hours with small "study pods" that are scheduled as needed.

Follow this

IMPORTANT

Students enrolled in EECS 367 or ROB 511 should complete the AutoRob Student Workflow Survey as soon as possible, in preparation for the first week of classes.

Discussion and Communication

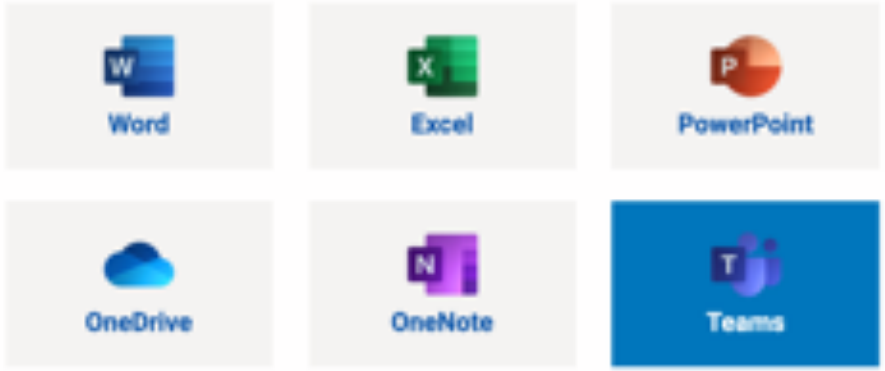
Discussion and Communication

- Discussion Microsoft team: “AutoRob Course at Michigan”
  - https://teams.microsoft.com/l/team/19%3ae1a8302e08d24fd8b42154f6b8edbdbf%40thread.tacv2/conversations?groupId=2a4f0e48-af38-412e-90c5-a2c0a1284404&tenantId=e66e77b4-5724-44d7-8721-06df160450ce
  - Formal course discussions allowed
- Discussion Discord server: https://discord.gg/A8eQhRq
  - Only *informal* discussion allowed (non-FERPA platform)
  - No discussion of grades, student records, etc.
- Course staff email: autorob-course-staff@umich.edu

its.umich.edu/communication/collaboration/microsoft-office-365/getting-started

Log In to the Online Microsoft Office 365 Apps

Additional apps are available to download.



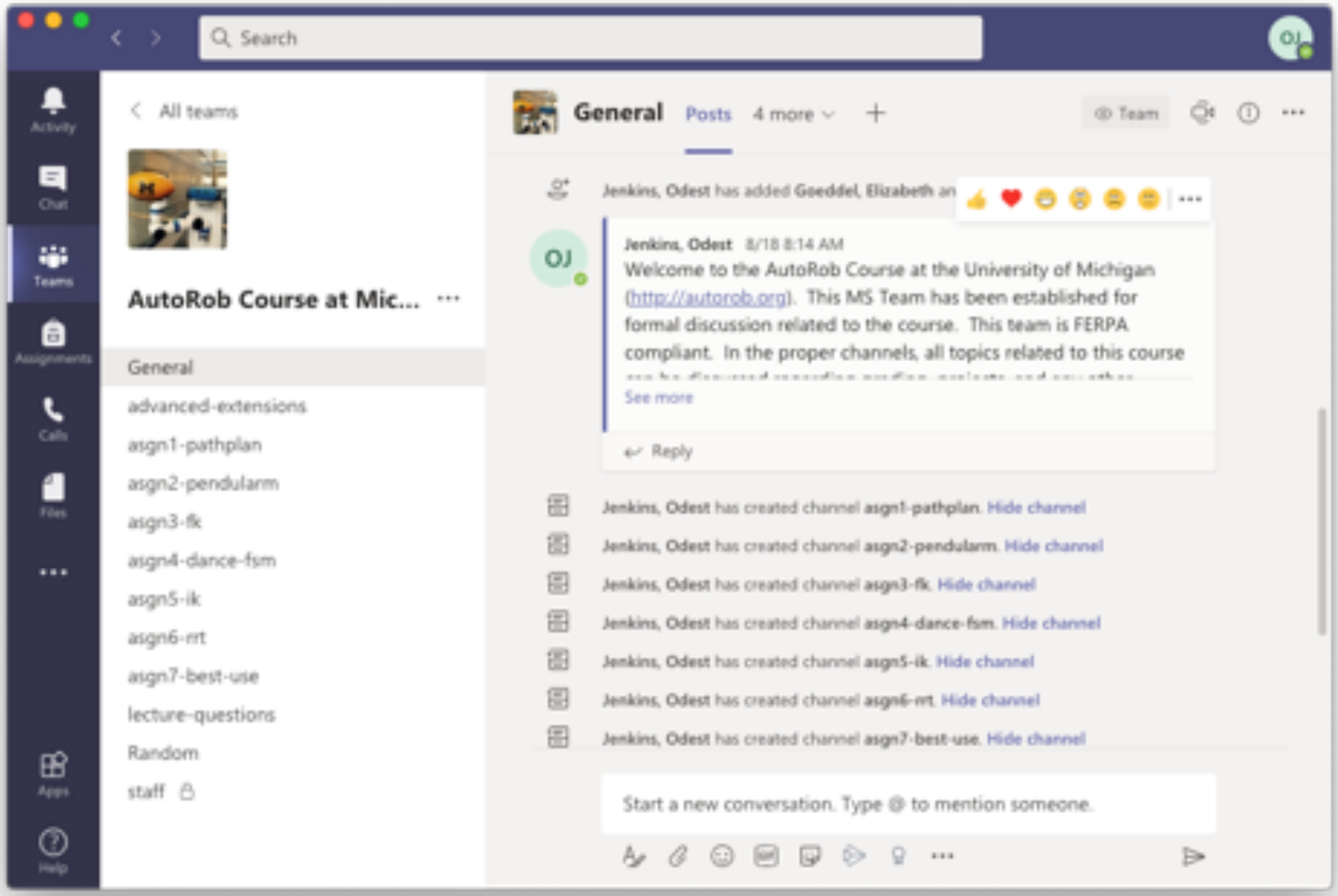
Other online apps available in U-M Office 365:

- Class Notebook
- Delve
- Dynamics 365
- Flow
- Forms
- Minecraft Education Edition
- Planner
- PowerApps
- SharePoint
- Staff Notebook

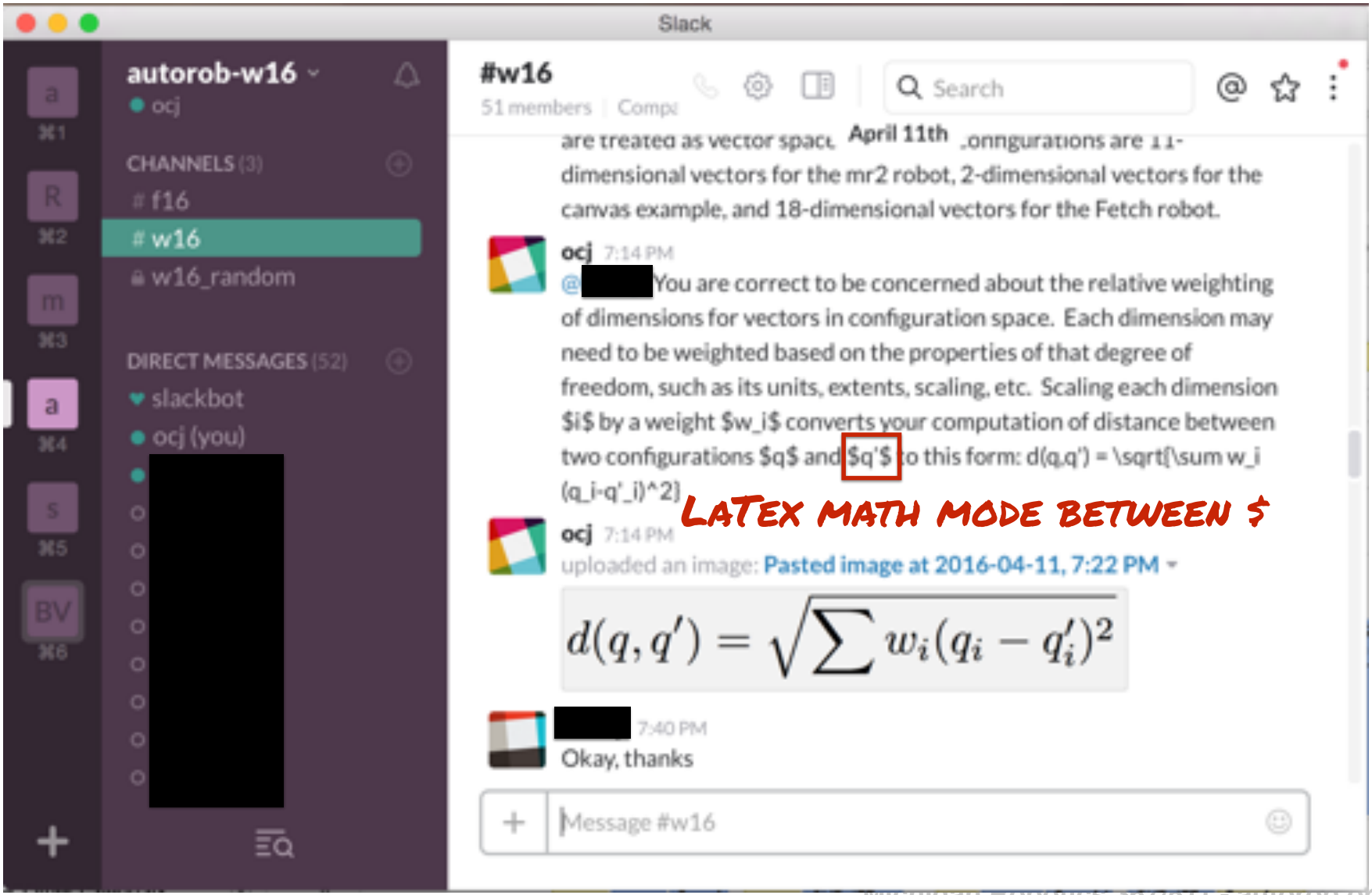
Microsoft Teams available to all U-M students through Office 365

.org

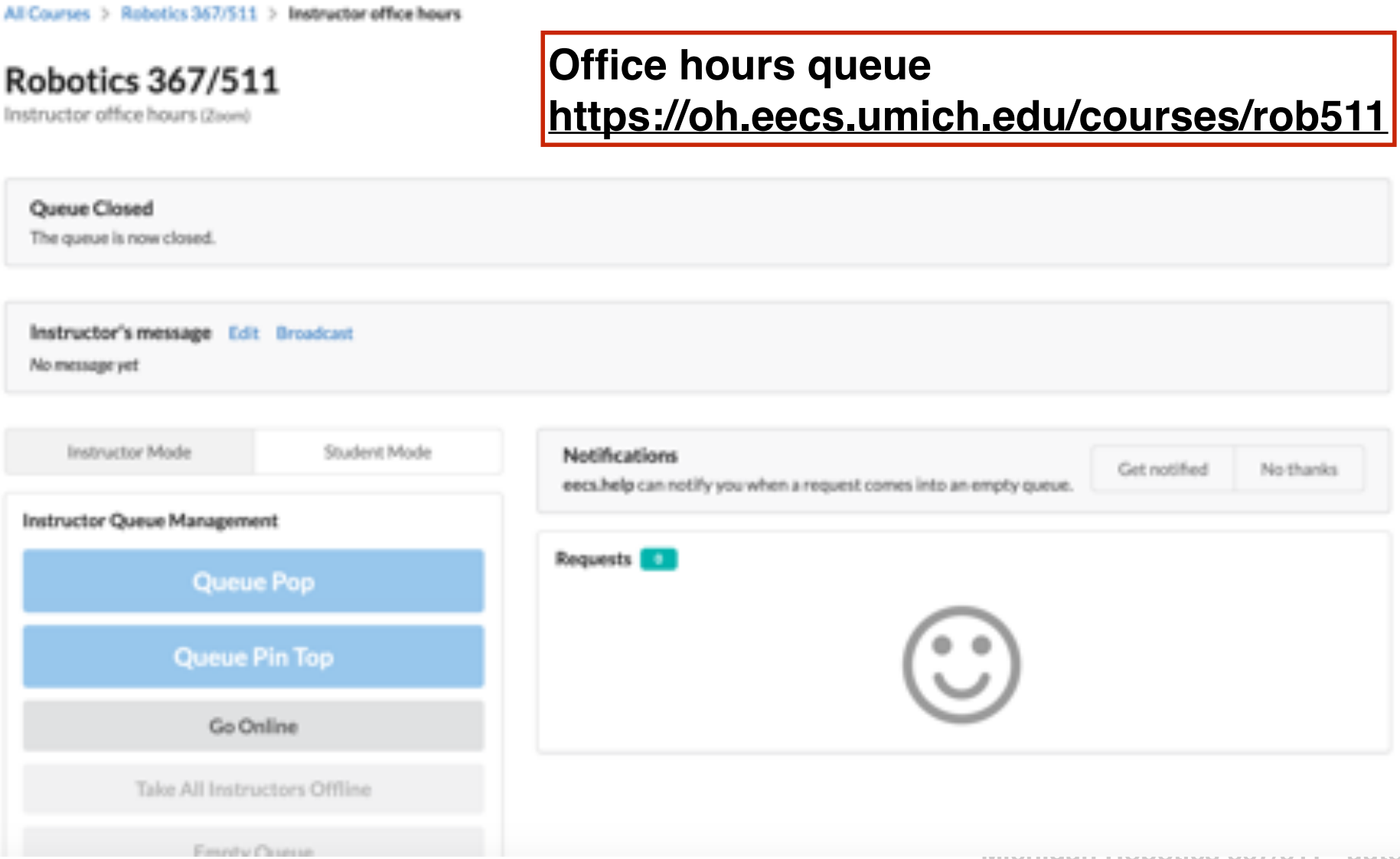
Microsoft Teams can be access via the web or native app



In the past, AutoRob used Slack




Office hours can get really busy
This will only be more challenging remotely


All Courses > Robotics 367/511 > Instructor office hours
Robotics 367/511
Instructor office hours (Zoom)
Office hours queue
https://oh.eecs.umich.edu/courses/rob511
Queue Closed
The queue is now closed.
Instructor's message Edit Broadcast
No message yet
Instructor Mode
Student Mode
Notifications
eecs.help can notify you when a request comes into an empty queue.
Get notified
No thanks
Instructor Queue Management
Queue Pop
Queue Pin Top
Go Online
Take All Instructors Offline
Empty Queue
Requests 0

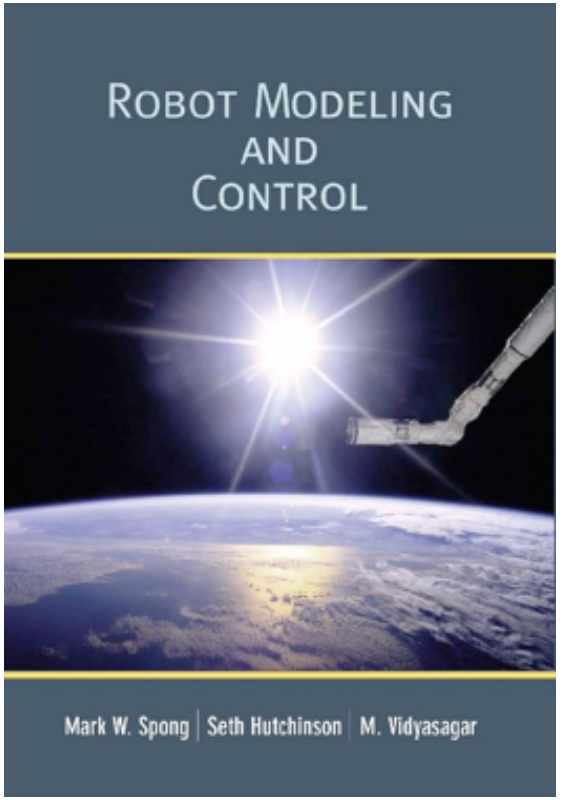
Course Structure

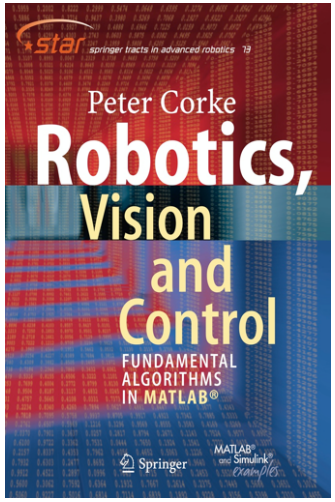
Course Structure

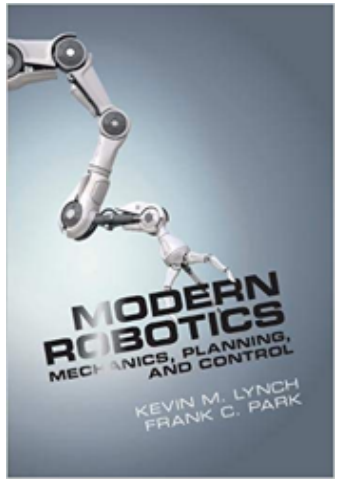
- Autonomous robot modeling and control
  - Objective: Give you the computational skills (and code) to model and control any mobile manipulator
- Project-focused class
  - 7 individual projects: from single joint control up to articulated motion planning with optional robot operating systems tutorials
- Computing-friendly introduction to robotics: projects in JavaScript

Course Textbook

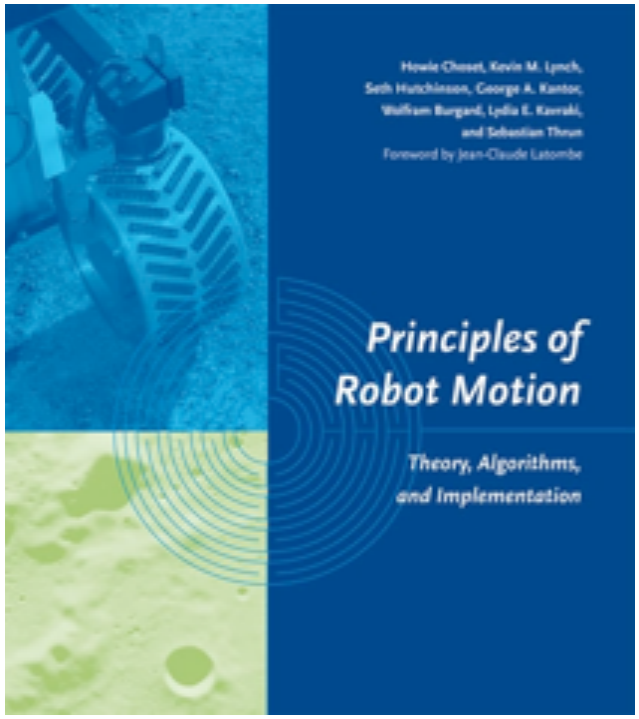
- Robot Modeling and Control (Spong, Hutchinson, Vidyasagar)
- Alternative: Robotics, Vision, and Control (Corke)
- Suggested but unsupported: Modern Robotics (Lynch and Park)
- In-depth coverage of concepts and math contained in textbooks
- Additional handouts and links will appear on the course website







Optional reading

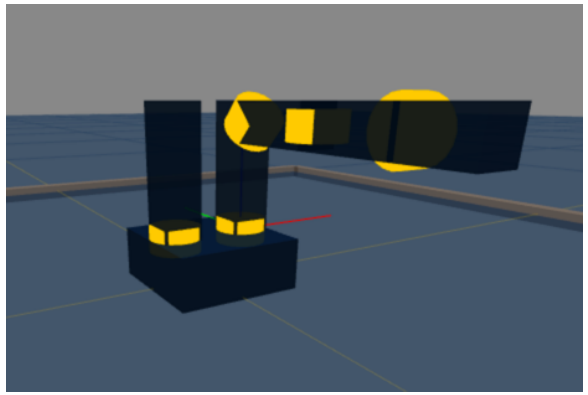


Projects

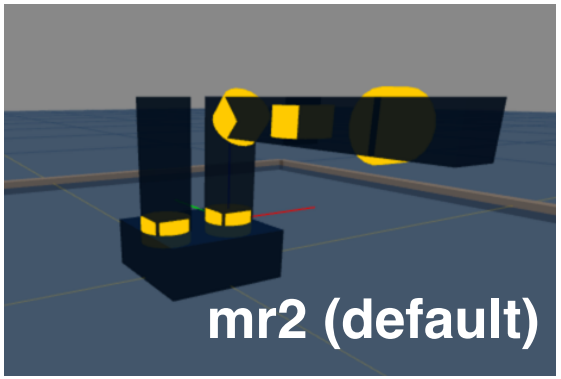
- Projects implemented in JavaScript/HTML5 using KinEval stencil
  - Projects submitted and tracked through git (gitlab|github|bitbucket)
  - Course staff needs admin access (please complete workflow survey!)
- 7 projects
  - 6 Programming, 1 Written/Oral
- Grading: projects are broken down into features that are “checked”
  - points are earned through successful implementation of features
  - Continuous Integration grading

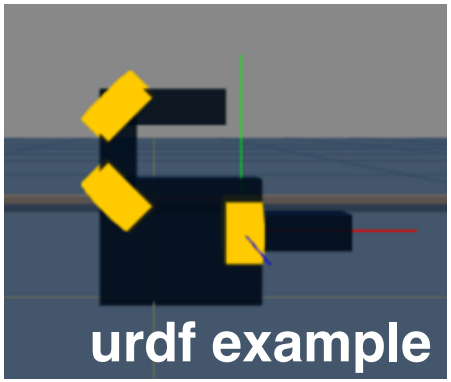
KinEval code stencil

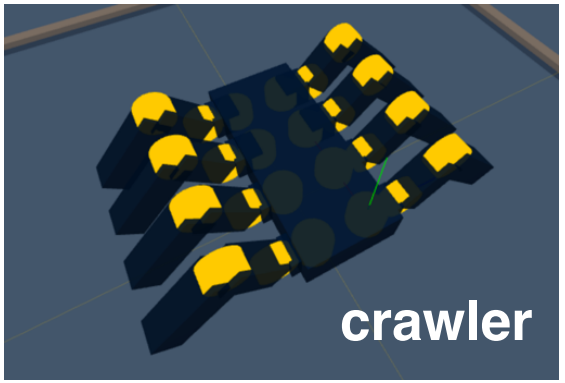
- Code stencil for AutoRob projects in 3D
- Uses threejs 3D rendering library and WebGL
- URDF-like robot description
- Usable, but not perfect, camera and UI controls
- AABB collision detection provided for planning
- Warning: professor-level coding

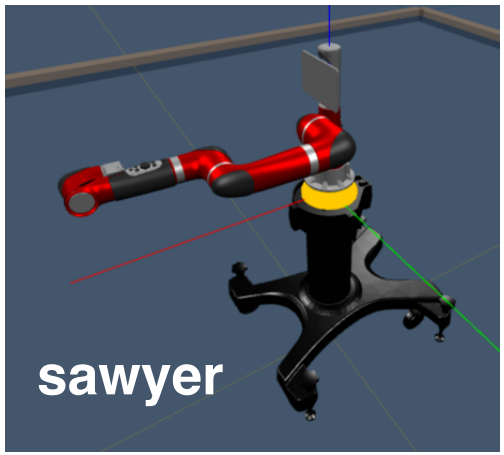


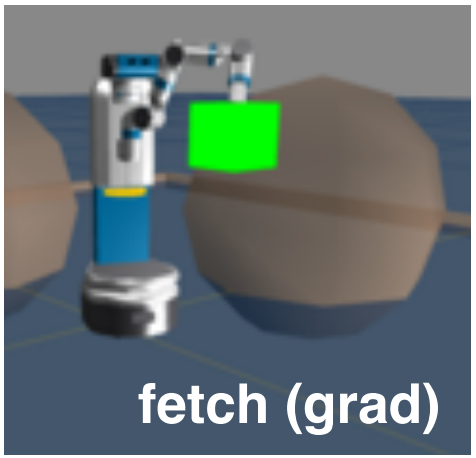
KinEval code stencil

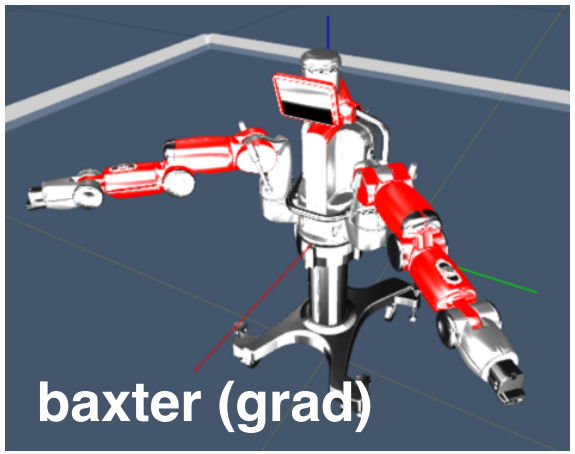












https://autorob.org

Search

AUTOROB | schedule | kinaval | git | assignments: 1 2 3 4 5 6 7

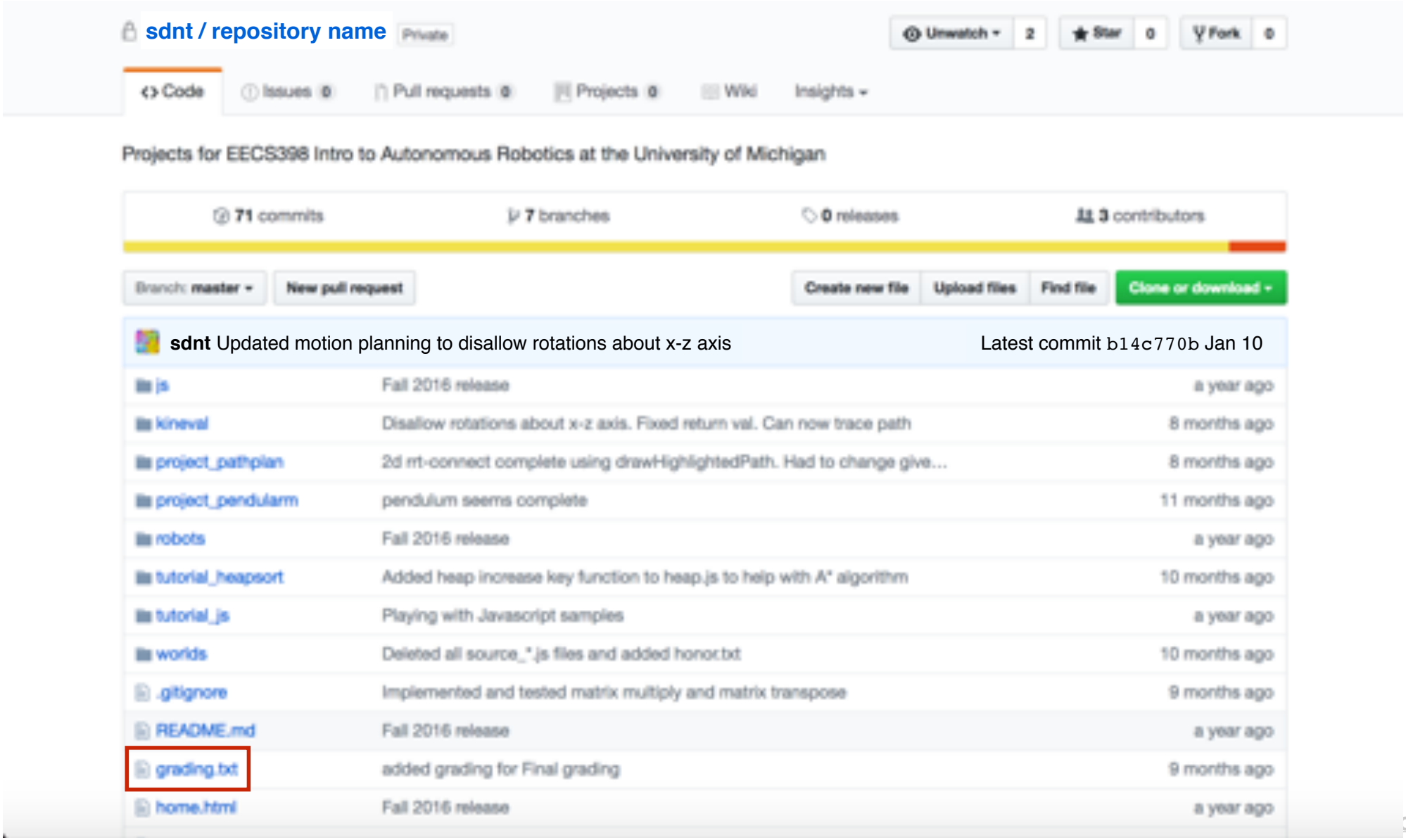
AutoRob

Introduction to Autonomous Robotics
Michigan EECS 367

Robot Operating Systems
Michigan ROB 511

Fall 2020

Flipped Classroom Hybrid COVID-19 Edition


sdnt / repository name
Private
Unwatch 2
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
Projects for EECS398 Intro to Autonomous Robotics at the University of Michigan
71 commits
7 branches
0 releases
3 contributors
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
sdnt Updated motion planning to disallow rotations about x-z axis
Latest commit b14c770b Jan 10
js Fall 2016 release a year ago
kineval Disallow rotations about x-z axis. Fixed return val. Can now trace path 8 months ago
project_pathplan 2d rrt-connect complete using drawHighlightedPath. Had to change give... 8 months ago
project_pendularm pendulum seems complete 11 months ago
robots Fall 2016 release a year ago
tutorial_heapsort Added heap increase key function to heap.js to help with A* algorithm 10 months ago
tutorial_js Playing with Javascript samples a year ago
worlds Deleted all source_*.js files and added honor.txt 10 months ago
.gitignore Implemented and tested matrix multiply and matrix transpose 9 months ago
README.md Fall 2016 release a year ago
grading.txt added grading for Final grading 9 months ago
home.html Fall 2016 release a year ago

Branch: **master** ▾ **repository name** / **grading.txt**

odestcj added grading for Final grading

1 contributor

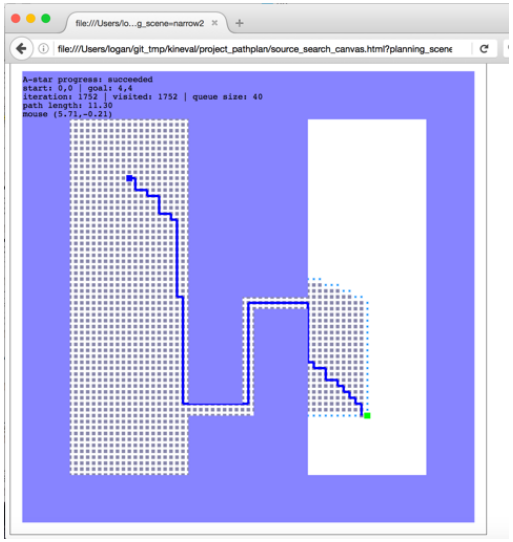
60 lines (58 sloc) | 2.26 KB | Raw | Blame

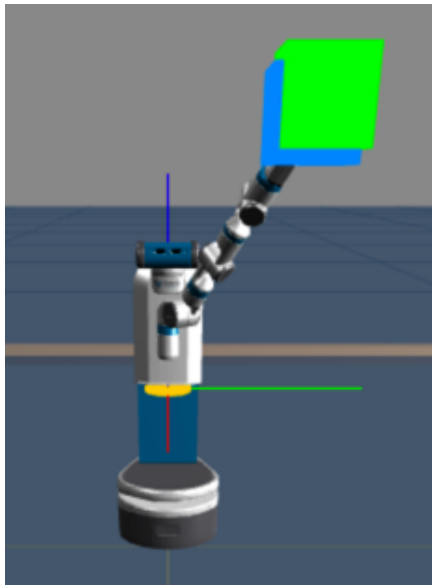
```
53: student name
EECS 398-004 F16
Assignment 1 feature 1 (4.00/4): PathPlan_Heap: CHECK
  comment: good work
Assignment 1 feature 2 (8.00/8): PathPlan_AStar: CHECK
  comment: good work
Assignment 2 feature 3 (4.00/4): Pendularm_Euler: CHECK
  error: Euler integrator is incorrect
  comment: why is this integrator stable?
  error: integration of dynamics is incorrect
  comment: double check computation of pendulum acceleration
  regrade: servo converges after several oscillations, borderline control performance
Assignment 2 feature 4 (4.00/4): Pendularm_VelocityVerlet: CHECK
  error: integration of dynamics is incorrect
  comment: double check computation of pendulum acceleration
  regrade: working
Assignment 2 feature 5 (4.00/4): Pendularm_PID: CHECK
  error: integration of dynamics is incorrect
  comment: modulo correction is not needed
Assignment 3 feature 6 (2.00/2): FK_MatrixRoutines: CHECK
```

Projects

- **Path Planning** A-star search in 2D world
- **Pendularm** physical simulation and PID control of 1 DoF robot
- **Forward Kinematics** convert robot configuration to 3D space
- **Dance Contest** control of robot joints to do a dance
- **Inverse Kinematics** control gripper of a robot to reach a point in 3D
- **Motion Planning** collision-free planning over robot configurations
- **Best Use of Robotics** what will you do with all of this power?

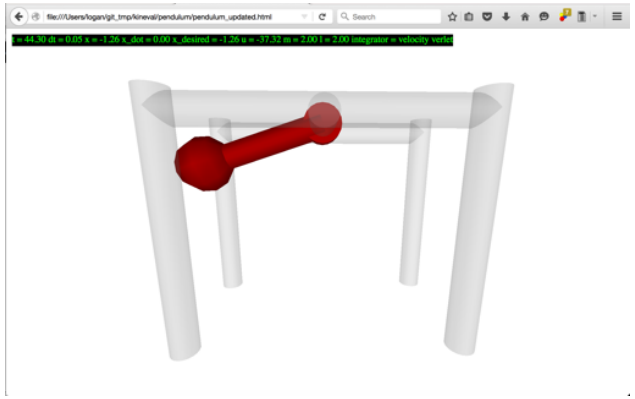
- Project 1: **Path Planning**

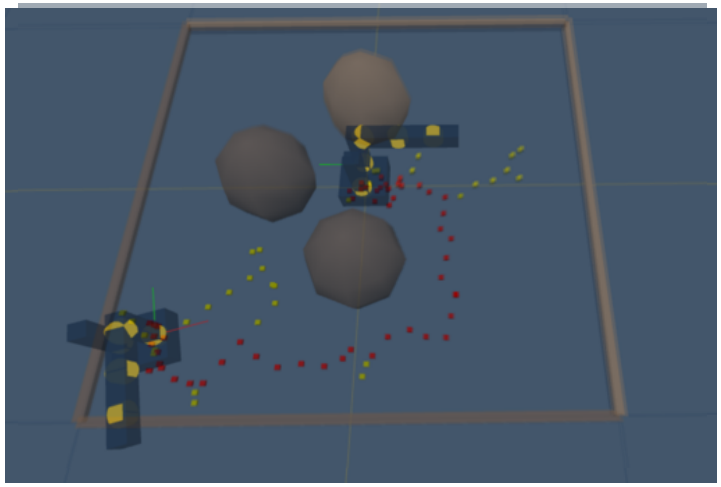




- Projects 3-5: **Forward and Inverse Kinematics**

- Project 2: **Pendularm**





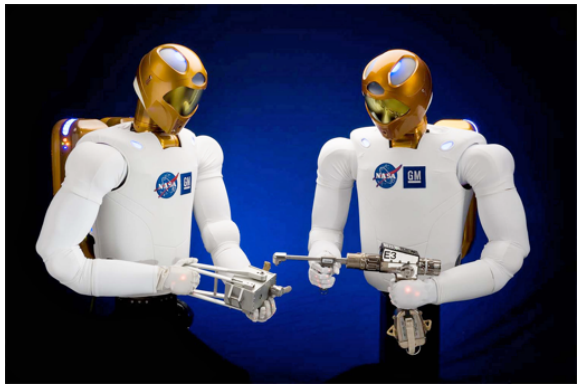
- Project 6: **Motion Planning**

Will you work with a real robot?

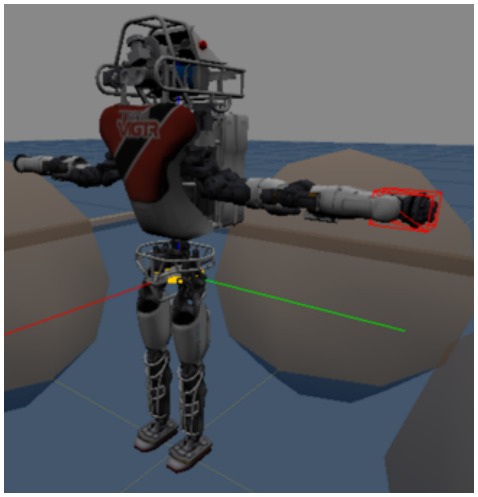
Will you work with a real robot?

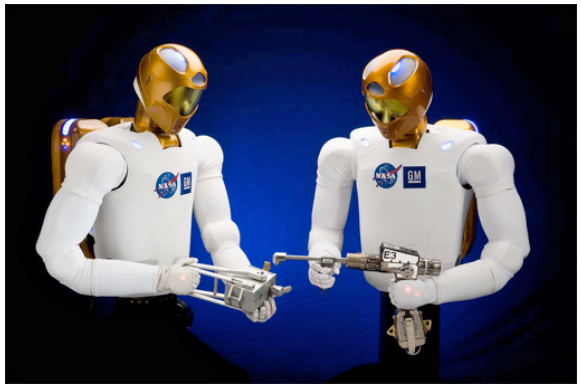
Yes, at least once using rosbridge/ROS

Maybe also?



Maybe also?





Course Policies

Grading Policy

EECS 367: Introduction to Autonomous Robotics

- 7 projects (12 points each)
- 5 quizzes (4 points each)
- Participation (4 points)

A: 95+ points
B: 83+ points
C: 73+ points

ROB 511: Robot Operating Systems

- 7 projects (18 points each)
- 5 quizzes (4 points each)
- Advanced features (4 points)
- Participation (4 points)

A: 138+ points
B: 120+ points
C: 105+ points

Continuous Integration Grading

- AutoRob in Fall 2020 will use a “CI grader” for the first time
- The CI grader will
  - pull code from your repository
  - run tests for all projects that past their due dates
  - return test and grading results to your repository
  - test schedule
- The course staff will review all grades for correctness

Late Policy

- Projects submitted after deadline may not be graded (zero credit)
- If a late submission is allowed, it can receive at most
  - 80% credit if pushed within 2 weeks of the deadline
  - 60% credit if pushed within 4 weeks of the deadline
  - 50% credit if pushed anytime before final grading deadline (Dec 8)

Regrading policy

- Projects features are graded with:
  - “CHECK” (sufficiently completed)
  - “DUE” (insufficiently completed)
  - “PENDING” (not due yet)
- A project feature can be regraded for partial credit for at most
  - 80% credit if pushed within 2 weeks of the last returned grading
  - 60% credit if pushed anytime before final grading deadline (Dec 8)

Collaboration Policy

- All work submitted must be your own
  - All code submitted must comply with Michigan Honor License
- No code can be communicated, including verbally
  - Explicit use of external sources must be clearly cited
  - Repositories must be **private** for proper compliance
- Free flow of discussion and ideas is encouraged

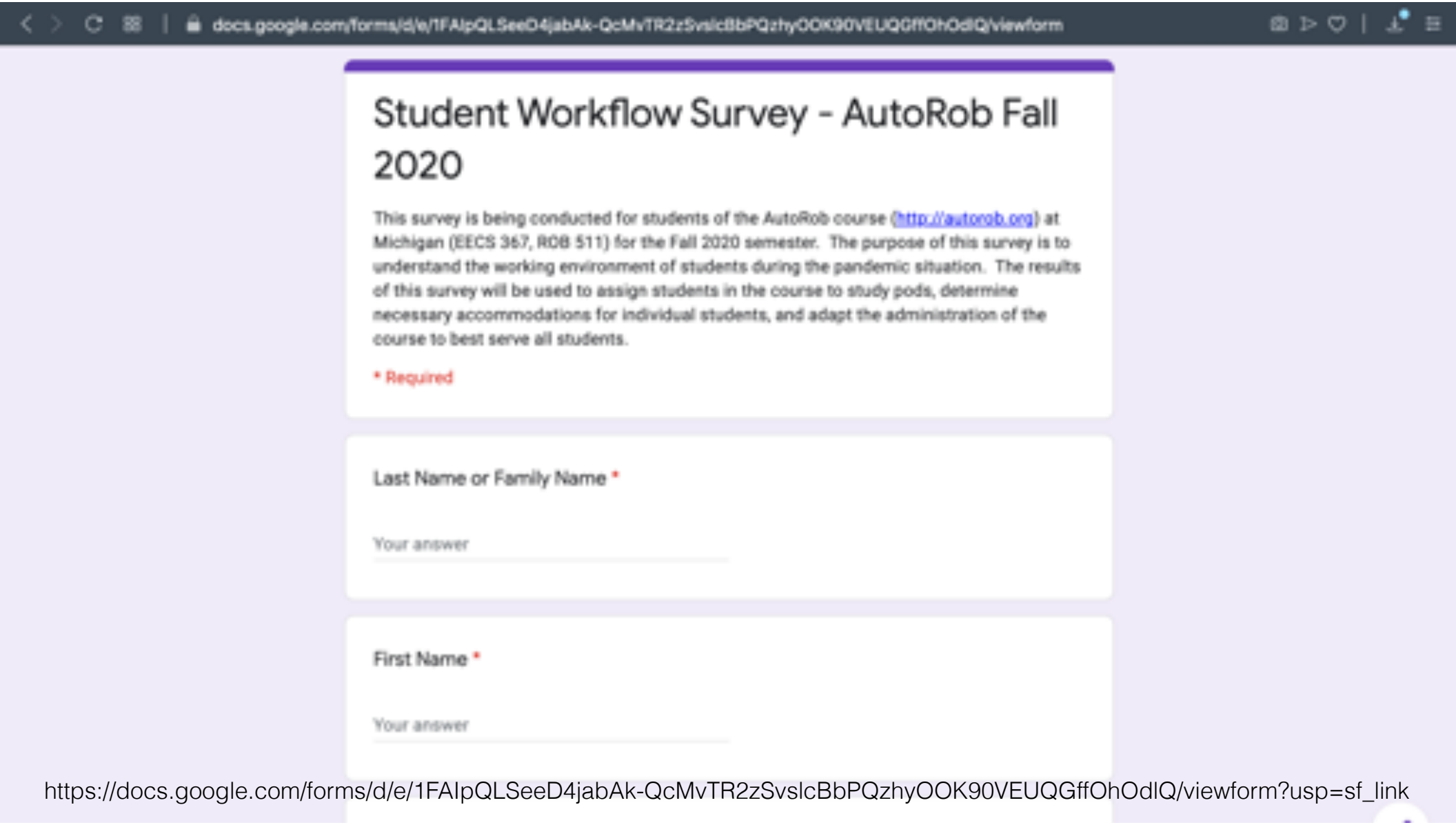
Michigan Honor License

- 3-Clause BSD License + Michigan Honor Code + “CC BY 4.0”
- Assert the compliance of your code with the MHL
  - Append your name to the end of LICENSE in your repository
- Submitted code will not be graded without asserting LICENSE

What I need from you **now**

What I need from you **now**

- Complete student workflow survey - TODAY!
- Join the course MS Team “AutoRob Course at Michigan”
- Ensure you are assigned to a study pod by the end of the week
- Install git and setup your working environment
  - create a git repository: https://gitlab.eecs.umich.edu/
  - ensure you can clone, commit, and push files to your repository
- View recorded course lectures (to be posted on Wednesday)



https://docs.google.com/forms/d/e/1FAIpQLSeeD4jabAk-QcMvTR2zSvslcBbPQzhyOOK90VEUQGffOhOdlQ/viewform?usp=sf_link