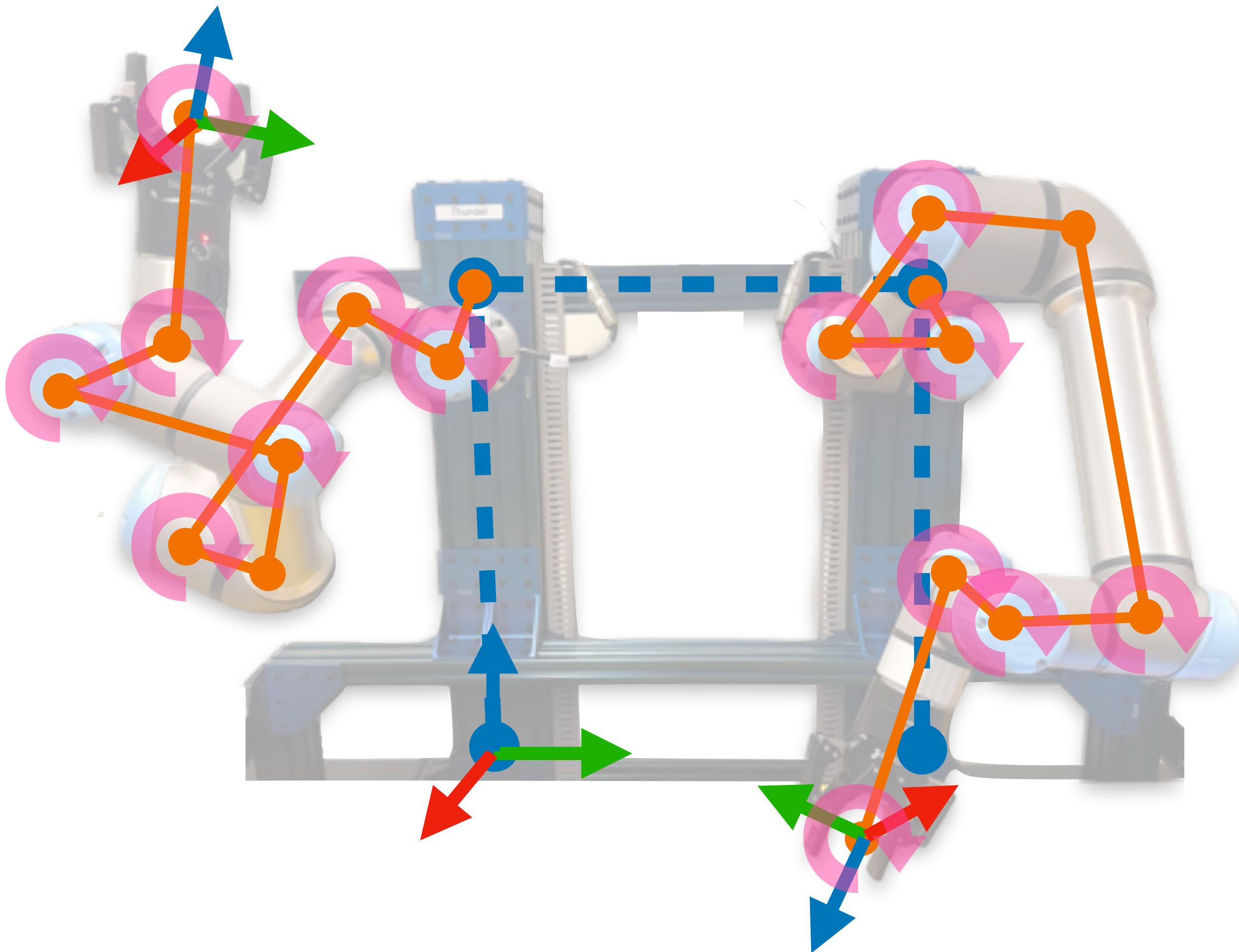


# Introduction to Intelligent Robotic Systems

**CSCI 5551**

Fall 2023 (Section 002)

University of Minnesota



# Welcome to 5551!

## Section - 002



# Course Staff

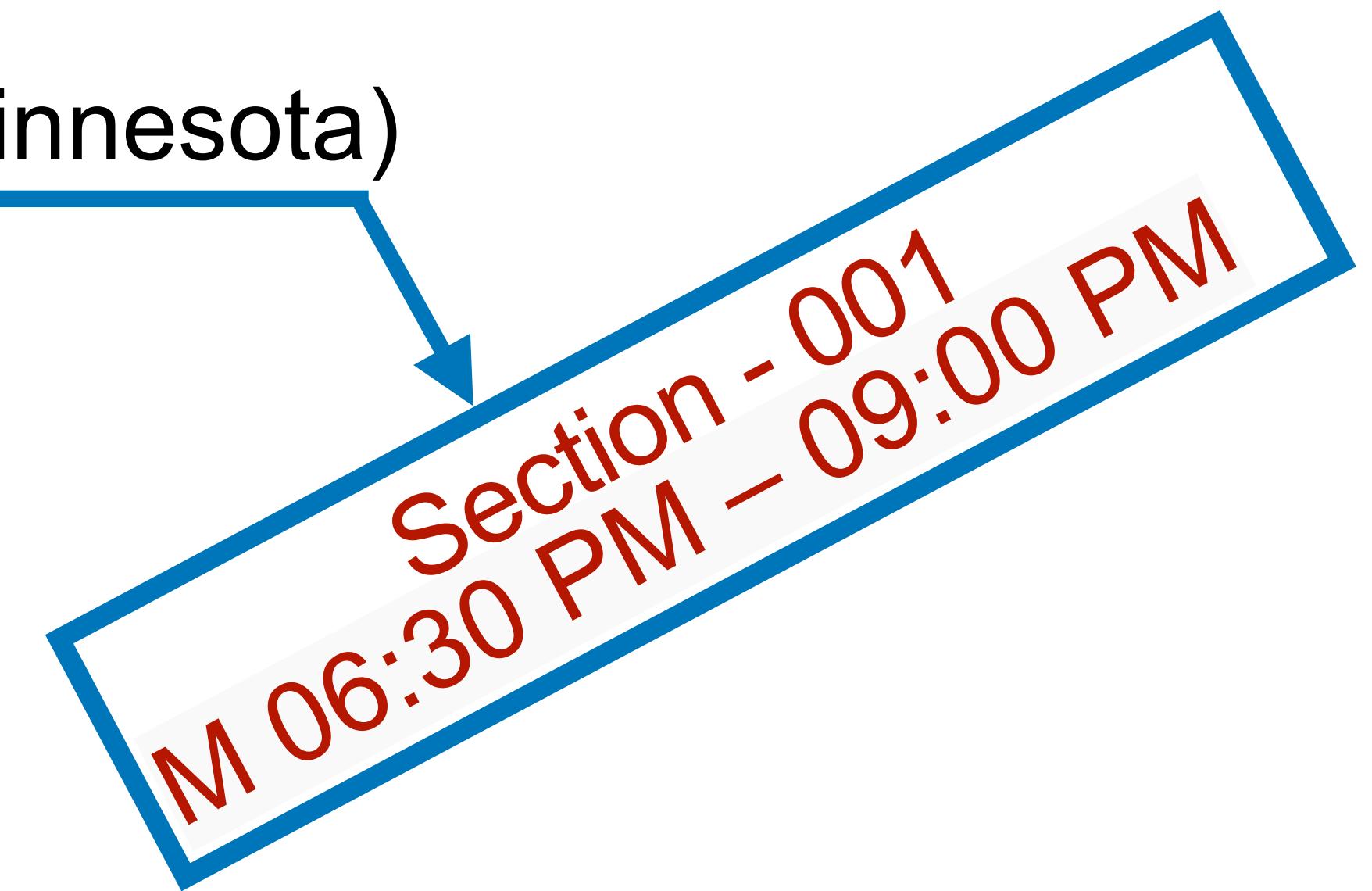
## Section - 002

- **Instructor: Karthik Desingh (*he/him*)**
  - Assistant Professor, CS&E | MnRI
  - [kdesingh@umn.edu](mailto:kdesingh@umn.edu)
  - OH: Mondays 8:30-10:00 am CT Shepherd 2-234
- **TA: Chahyon Ku (*he/him*)**
  - MS Student in Robotics
  - [ku000045@umn.edu](mailto:ku000045@umn.edu)
  - OH: T, Th 10:00-11:00 am CT Keller Hall 2-209



# Acknowledgement

- This course builds on and is indebted to materials from:
  - Prof. Chad Jenkins (University of Michigan) and the staff of autorob.org
  - Prof. Nikolaos Papanikolopoulos (University of Minnesota)
  - Prof. Junaed Sattar (University of Minnesota)



# What are intelligent robotic systems?



# What are intelligent robotic systems?

“systems that provide intelligent services and information by interacting with their environment, including human beings, via the use of various sensors, actuators and human interfaces”



# What are intelligent robotic systems?

“systems that provide intelligent services and information by interacting with their environment, including human beings, via the use of various sensors, actuators and human interfaces”



# What are intelligent robotic systems?

It is getting very hard to define this term.

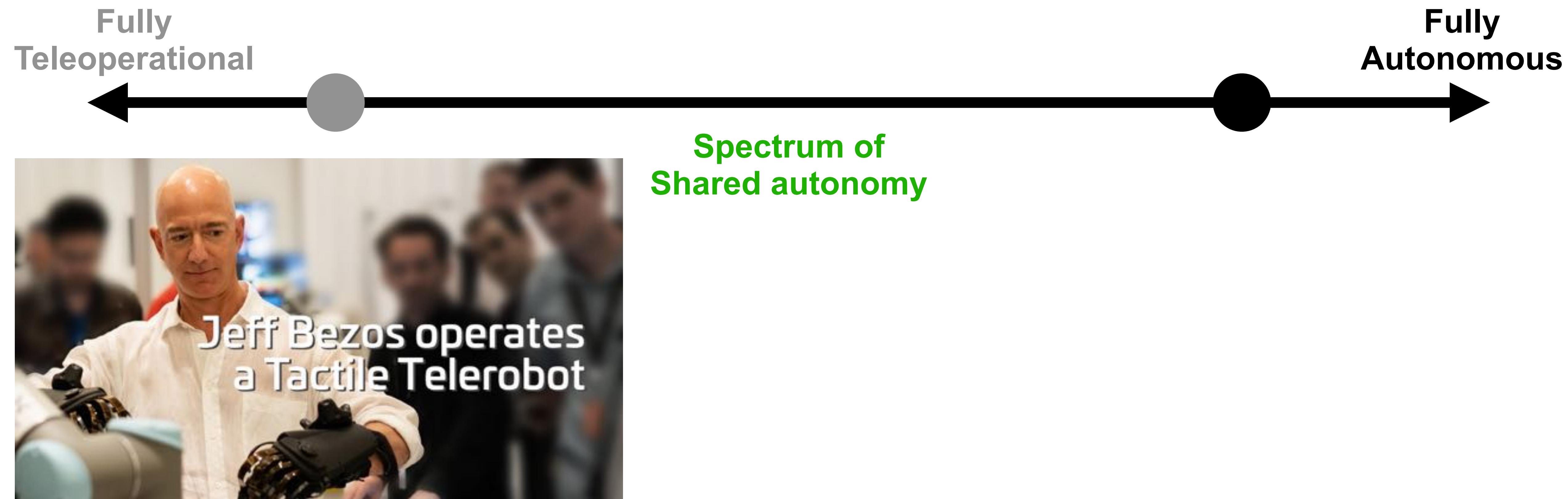
For the sake of this course,  
let us call this “**ability to operate with some autonomy**”



# What are intelligent robotic systems?



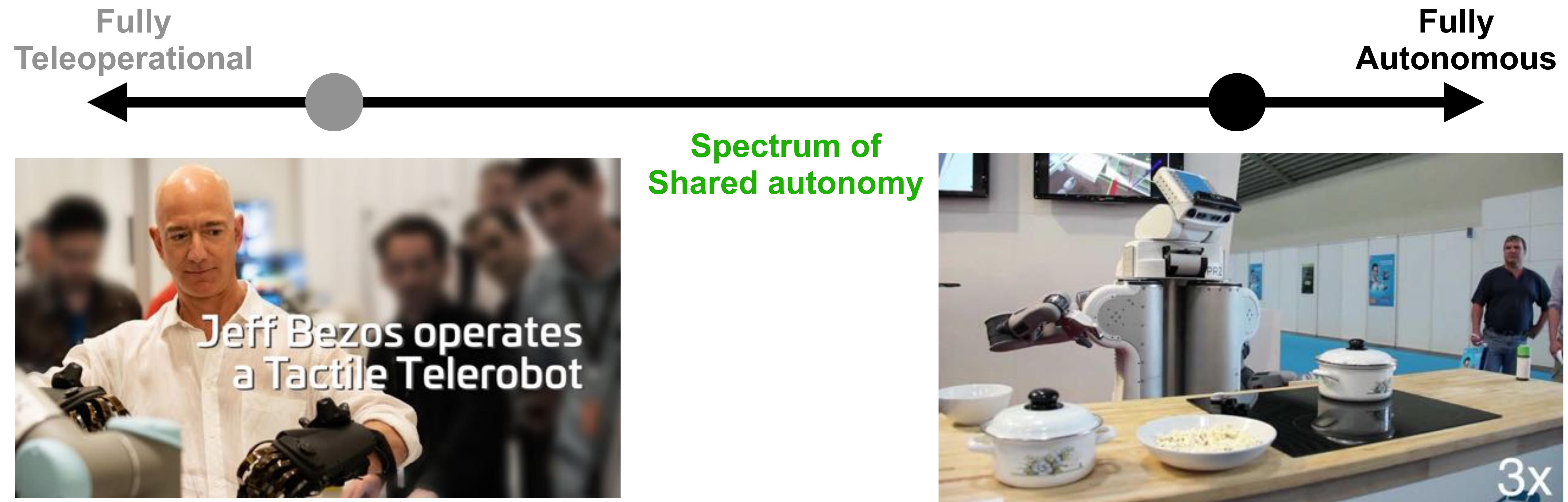
# What are intelligent robotic systems?



HaptX: <https://www.youtube.com/watch?v=uwYtwQtoOh0>



# What are intelligent robotic systems?



HaptX: <https://www.youtube.com/watch?v=uwYtwQtoOh0>

TUM/IAS group: <https://www.youtube.com/watch?v=cTCJSNjTdo0>

# What are intelligent robotic systems?

**...systems that can perform Sense-Plan-Act....**



# What are intelligent robotic systems?

...systems that can perform **Sense-Plan-Act**....



Zhiqiang Sui et al. 2017

# What are intelligent robotic systems?

...systems that can perform **Sense-Plan-Act**....



Zhiqiang Sui et al. 2017

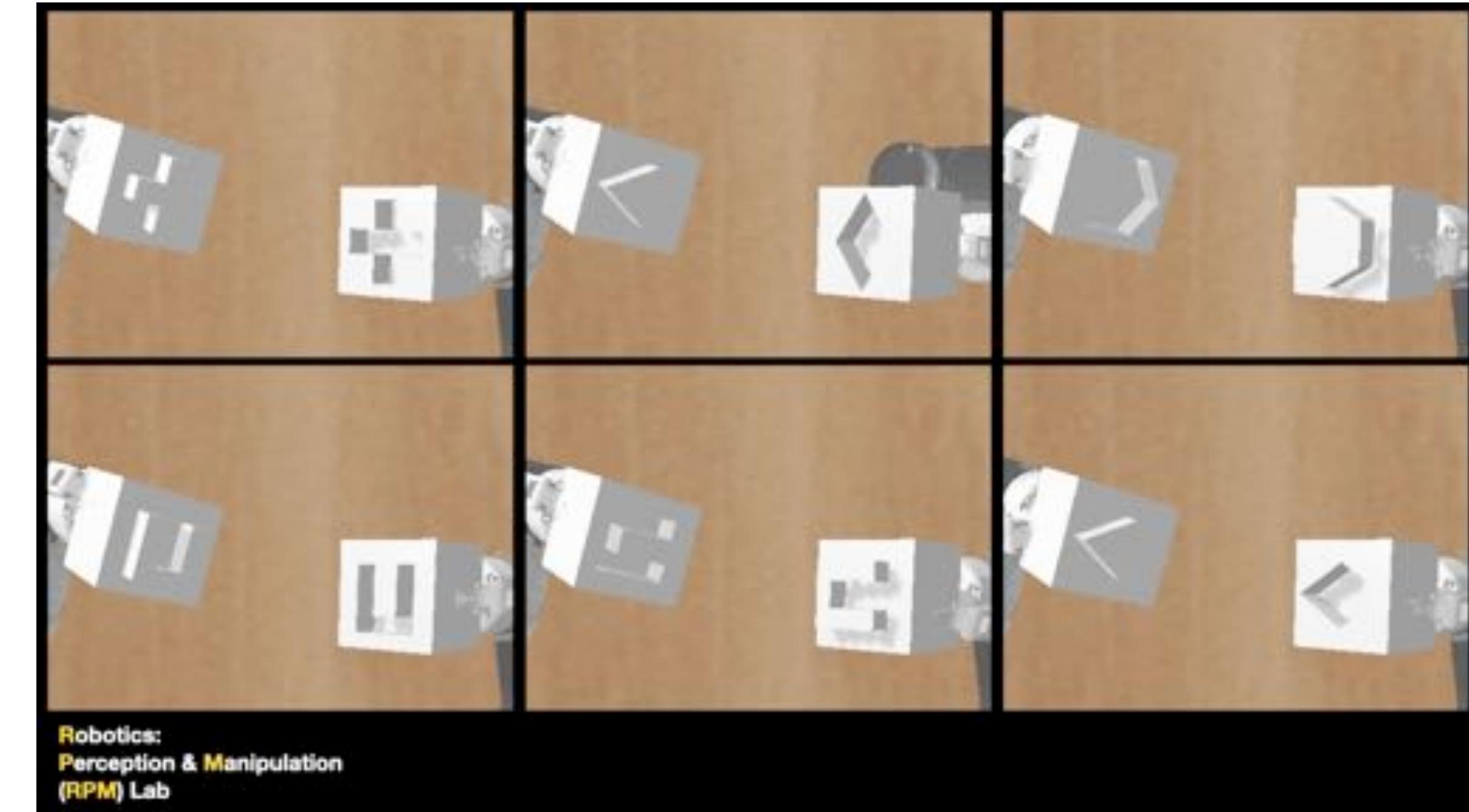
# What are intelligent robotic systems?

...systems that can perform **Sense-Plan-Act**....

... can also learn skills ... transfer these skills ... adapt to new environments ...



Carl Winge et al. 2022



Chahyon Ku et al. 2023

# What are intelligent robotic systems?

...systems that can perform **Sense-Plan-Act**....

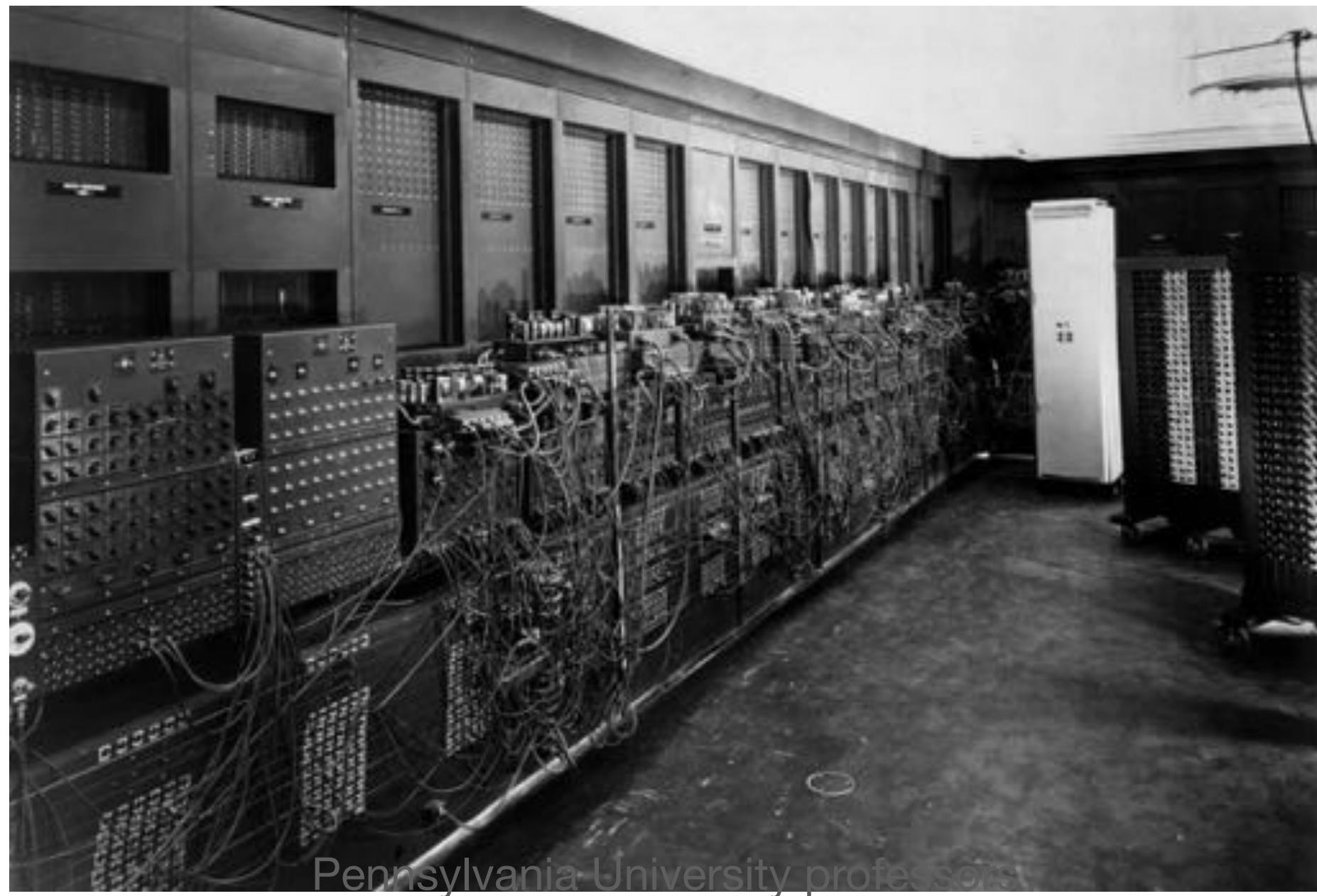
... can also learn skills ... transfer these skills ... adapt to new environments ...



Bahaa Aldeeb et al. 2023

# History of Computers, AI and Robotics





Pennsylvania University professors John Mauchly and J. Presper Eckert build the 'grandfather' of digital computers, the Electronic Numerical Integrator and Calculator (ENIAC)

1944



Researchers William Shockley, John Bardeen and Walter Brattain at Bell Laboratories invent the transistor.

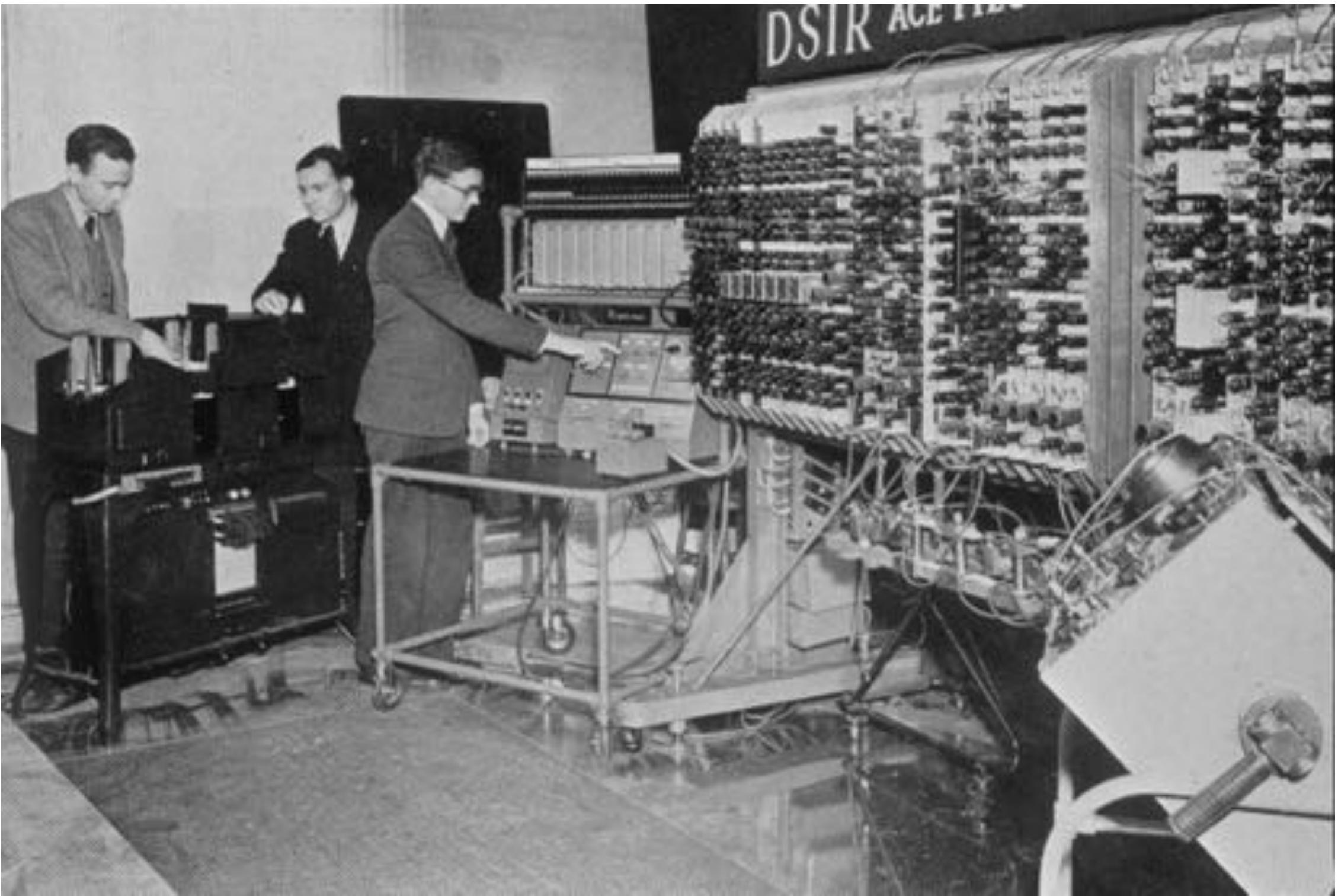
1948



'I, Robot' by Isaac Asimov is published, laying the foundations for the idea of robots in culture.

1950

<https://everydayrobots.com/>



Alan Turing introduces 'The Turing Test' –  
a test of a machine's ability to exhibit intelligent behavior  
equivalent to, or indistinguishable from, that of a human.



Grace Hopper develops COBOL, the first computer language.  
The second, FORTRAN, is developed by a team of  
IBM programmers a year later.

1950

1953

<https://hotcorn-cdn.s3.amazonaws.com/wp-content/uploads/sites/2/2020/09/22125112/bletchleypark-pilotace-scaled.jpg>  
<https://everydayrobots.com/>



Dartmouth conference coins the term  
'artificial intelligence' and  
launches the field of AI.

IBM mainframes are used in early experiments



Five of the attendees of the 1956 Dartmouth Summer Research Project on Artificial Intelligence reunited at the July AI@50 conference. From left: Trenchard More, John McCarthy, Marvin Minsky, Oliver Selfridge, and Ray Solomonoff. (Photo by Joseph Mehling '69)

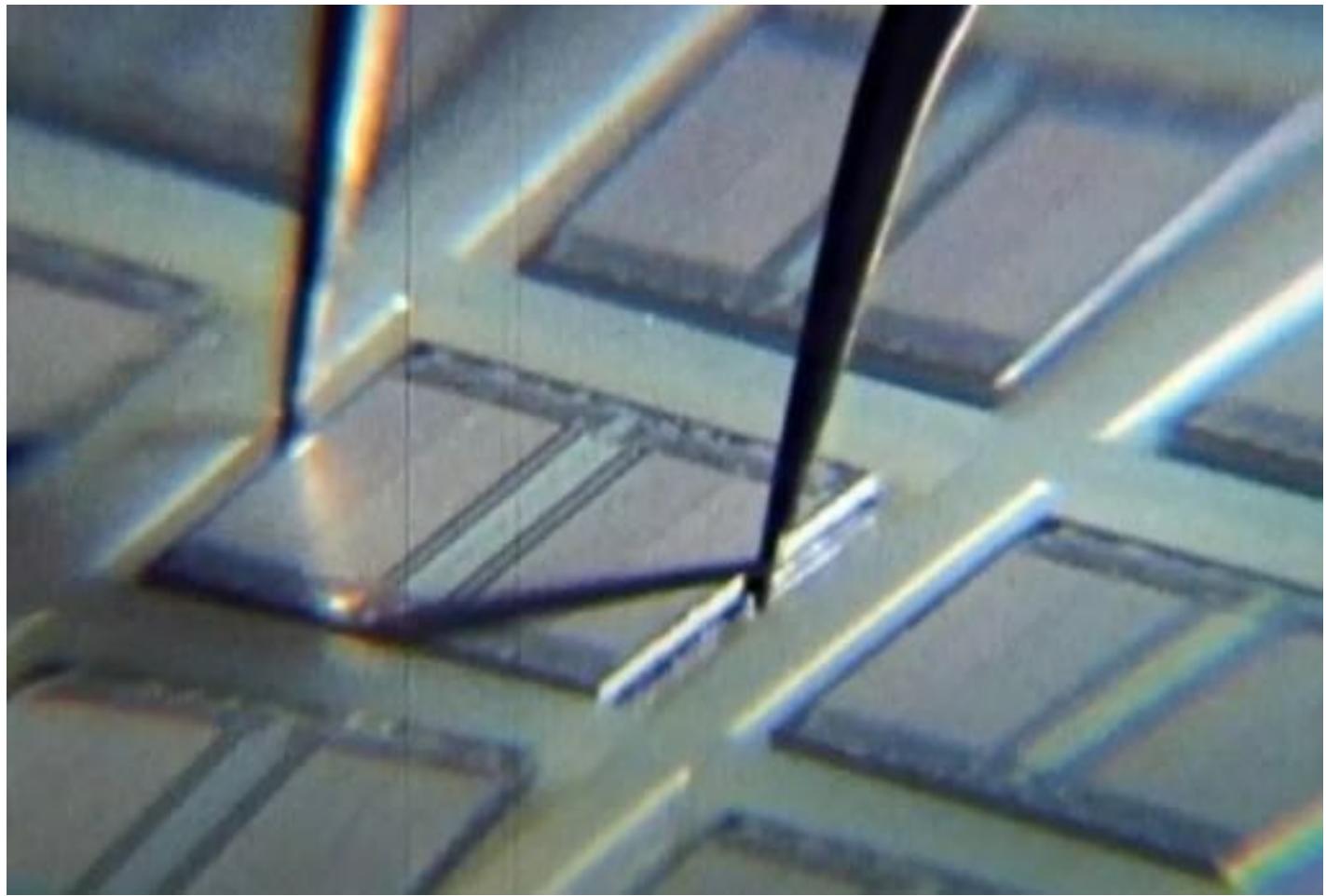


IBM's chairman and CEO, Thomas J. Watson Jr., bets the company's future on the IBM Series/360 — the largest privately-financed commercial project in history. The risk pays off, changing the computer industry forever. Work is revolutionized, productivity is enhanced and countless new tasks become possible.

1956

1964

<https://everydayrobots.com/>



Intel and Ted Hoff introduce the first microprocessor, the Intel 4004.

Intel co-founder, Gordon Moore, theorizes that computing would dramatically increase in power, and decrease in relative cost, at an exponential pace. The insight, known as Moore's Law, becomes the golden rule for the electronics industry, and a springboard for innovation.

1971

1976

1977



Steve Wozniak and Steve Jobs release the Apple 1



A year later, Apple releases the Apple II

<https://image.cnbcfm.com/api/v1/image/100932798-128279719-1.jpg?v=1583960525>

<https://everydayrobots.com/>



The U.S. Defense Department funds the first experimental computer network—ARPANET. It connects computers everywhere, and is a forerunner to the internet.



1966

```
<!DOCTYPE html>
<html>
<body>
<h1>What will people do now we have the internet?</h1>
<p>Surprisingly, it turns out that we really like looking at pictures of cats.</p>
```

Tim Berners-Lee and his colleagues at CERN develop hypertext markup language (HTML) and the uniform resource locator (URL), giving birth to the first incarnation of the World Wide Web.



IBM's Deep Blue Computer, a form of AI, beats reigning world chess champion Gary Kasparov.



[https://i.insider.com/55947fbf2acae7b7188b5388?width=750&format=jpeg&auto=webp](https://i.insider.com/55947fbf2acae7b7188b5388?width=750&format.jpeg&auto=webp)  
<https://everydayrobots.com/>



### Search Stanford

10 results ▾ clustering on ▾

### Search The Web

10 results ▾ clustering on ▾

Larry Page & Sergey Brin, two computer science graduate students from Stanford University, pioneer a new way to search for and find information on the web. They call their invention 'Google'.

1998



Four founders start a company called 'Android'

2003

<https://i.insider.com/55947fbf2acae7b7188b5388?width=750&format.jpeg&auto=webp>  
<https://indonesiamendesain.com/wp-content/uploads/2020/06/original-google-logo-font.jpg>  
<https://everydayrobots.com/>



Steve Jobs unveils the iPhone at Macworld

2007

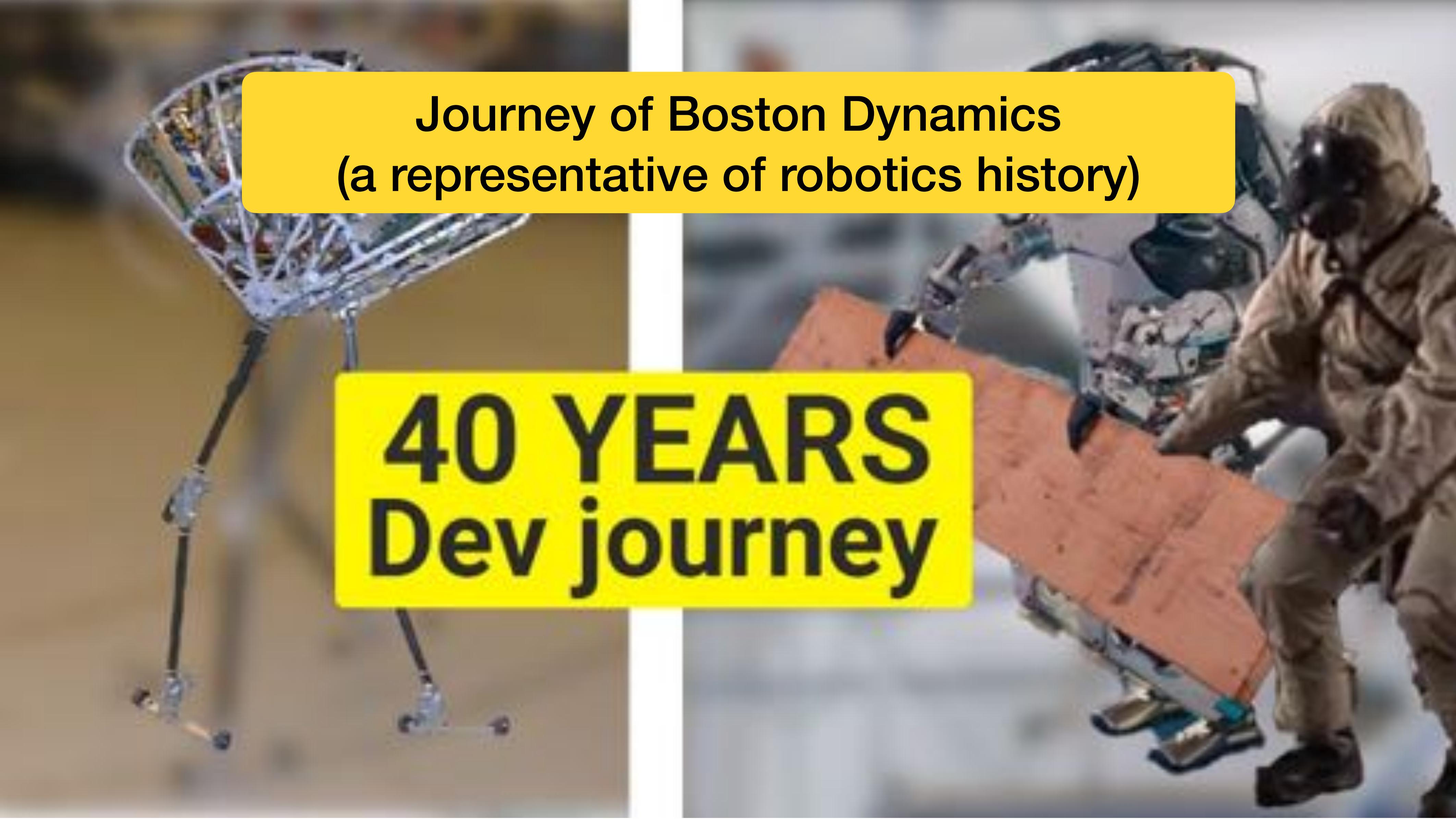


Search becomes intuitive.  
Maps are intelligent.  
Work is more productive than ever

<https://i.insider.com/587374fadd0895e1148b47e7?width=1136&format=jpeg>

<https://everydayrobots.com/>

So, this is computing and AI...  
What was happening in robotics?

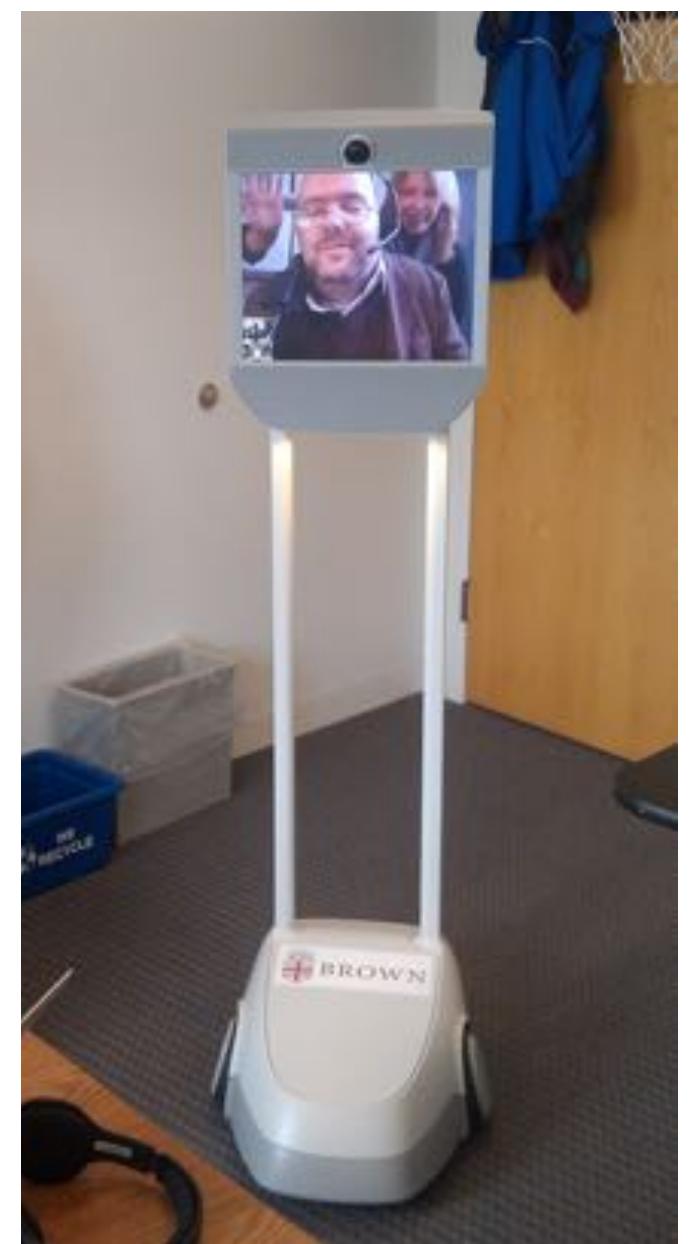


Journey of Boston Dynamics  
(a representative of robotics history)

40 YEARS  
Dev journey

<https://youtu.be/qtos3ezujn4?feature=shared>

# So, where is my robot?

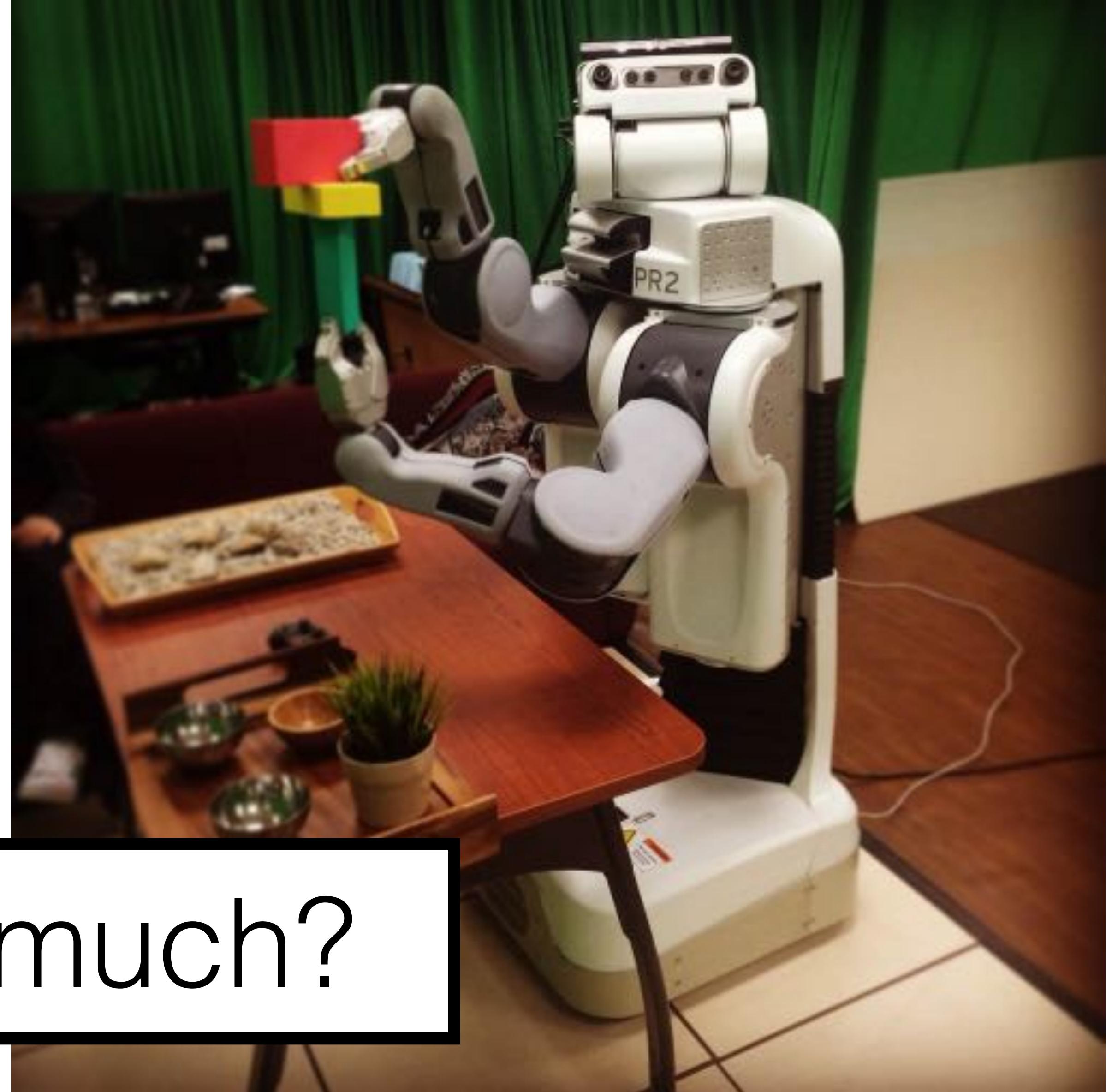


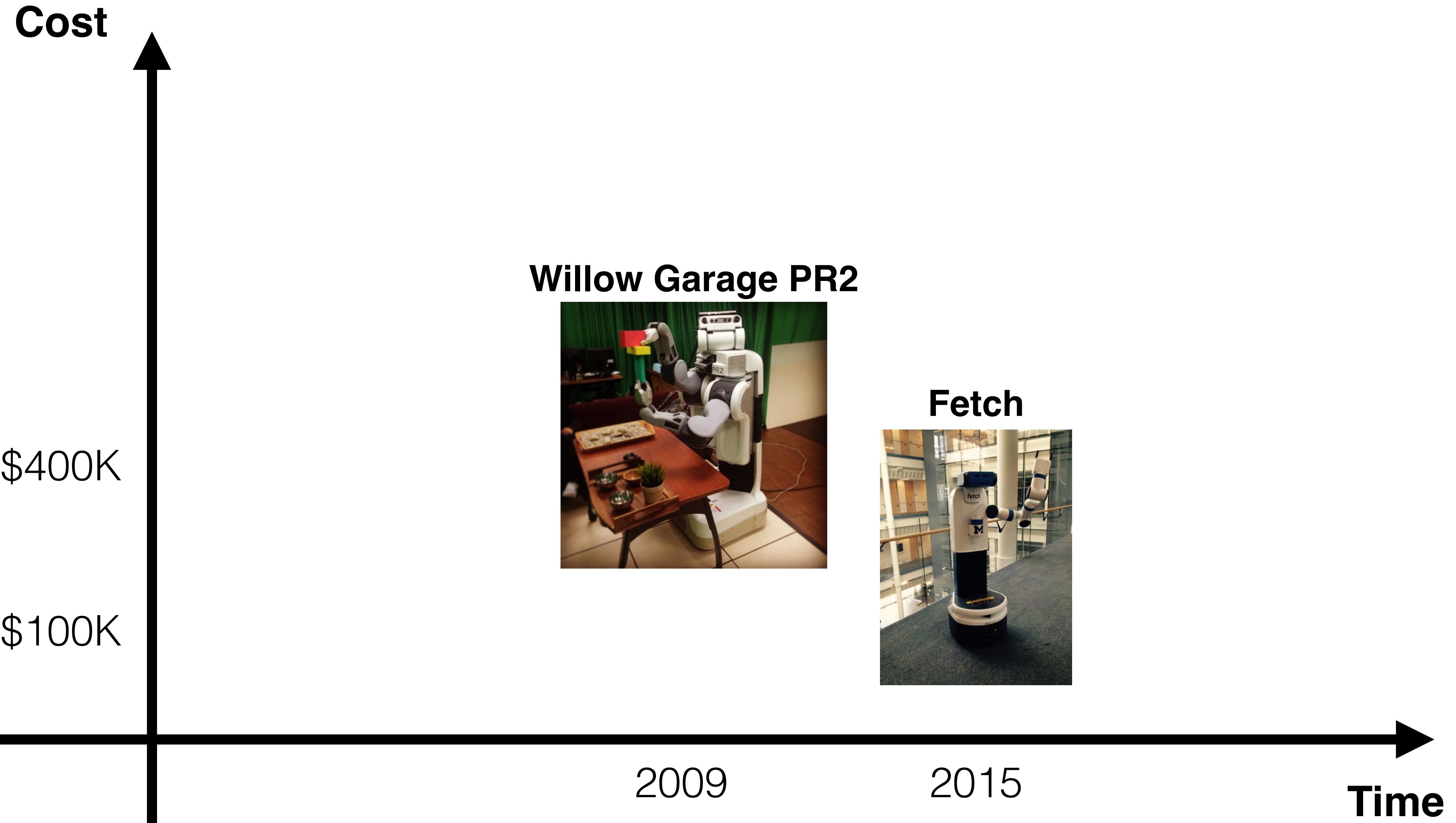
# Mobile Manipulation Robots





How much?







2002

\$400K

\$100K

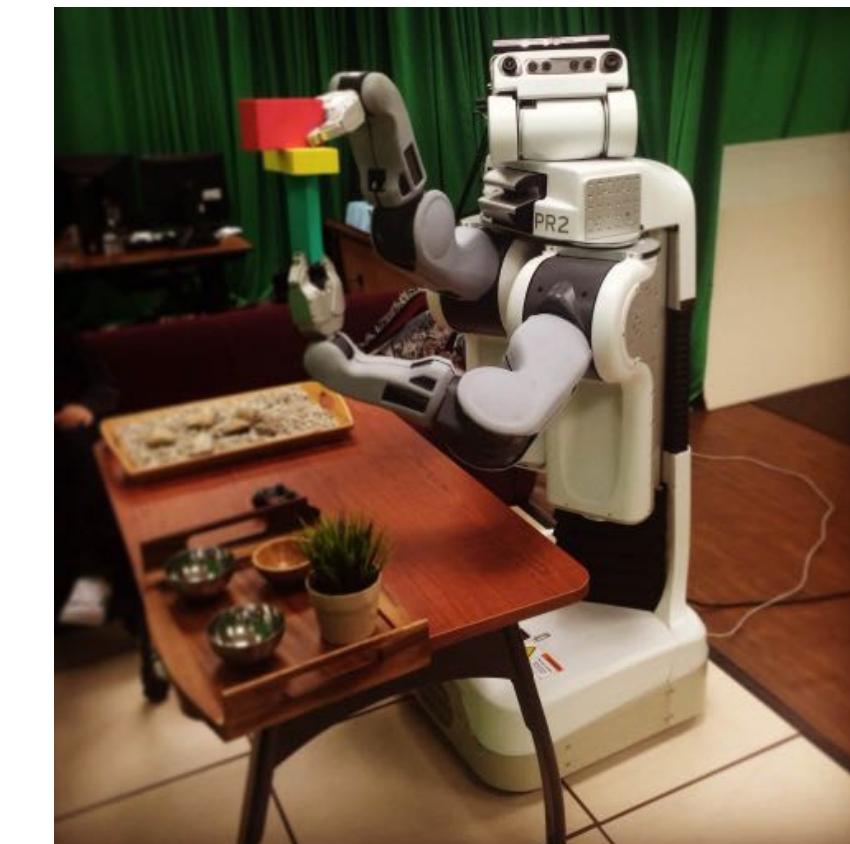


2009

2015

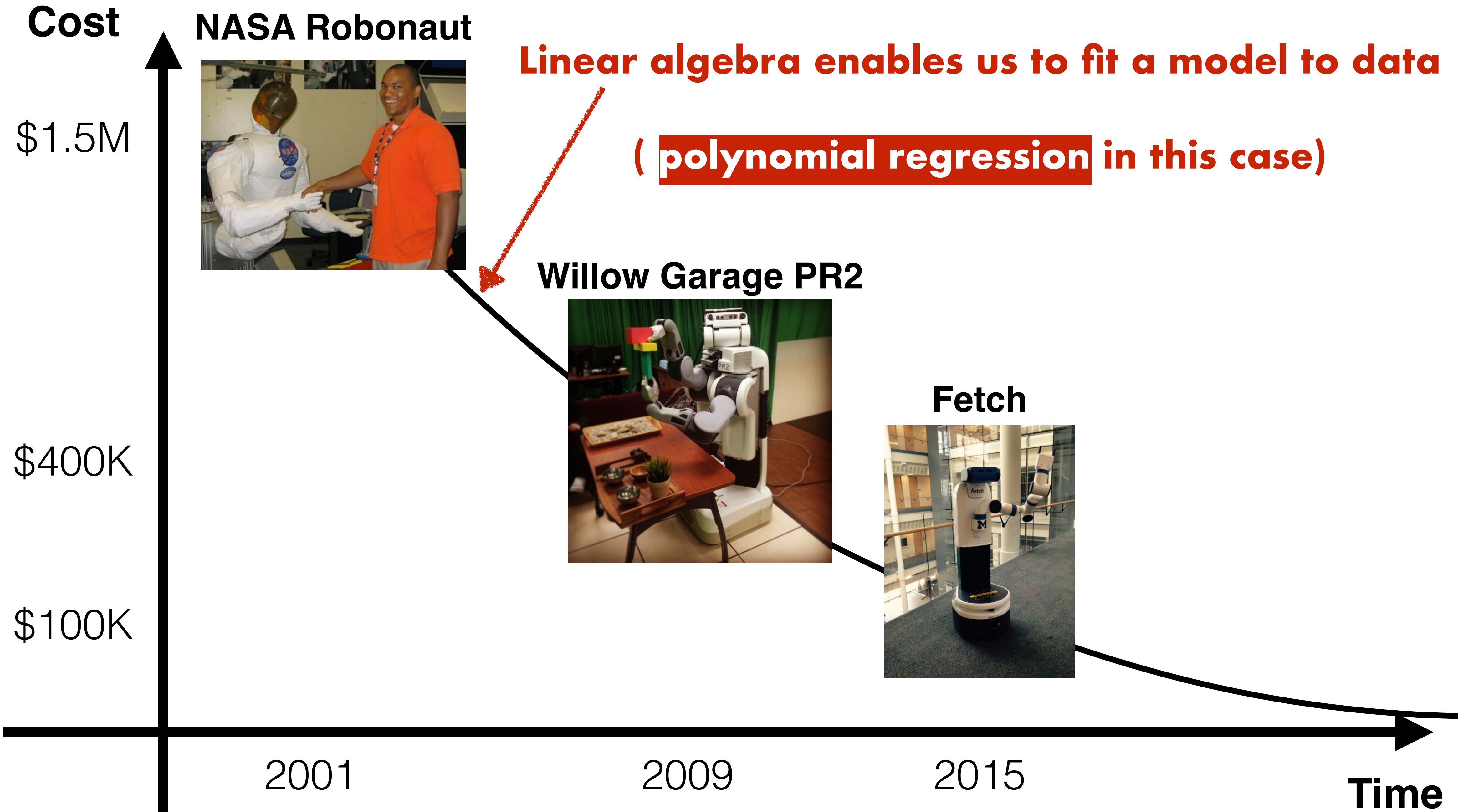
Time

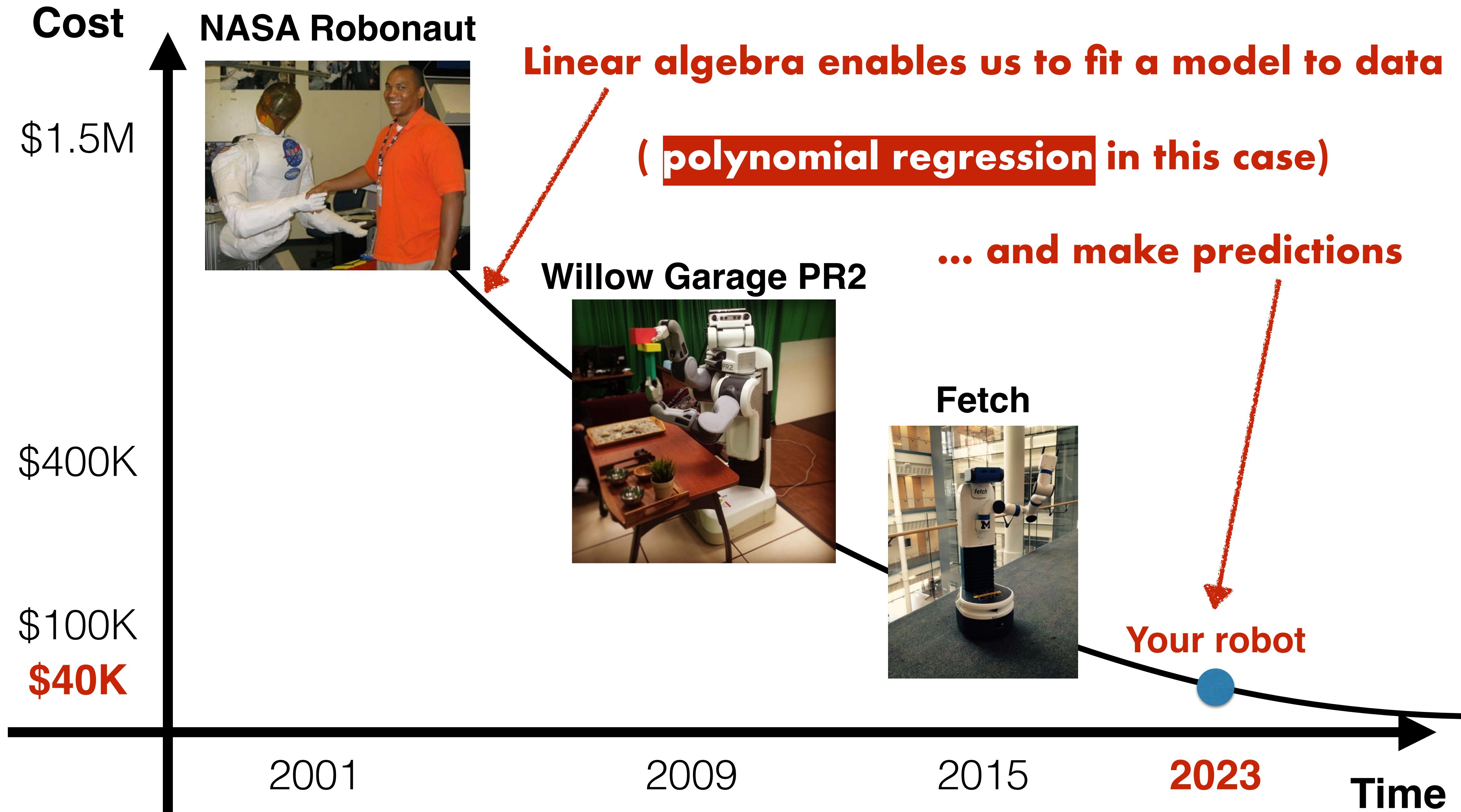
## Willow Garage PR2

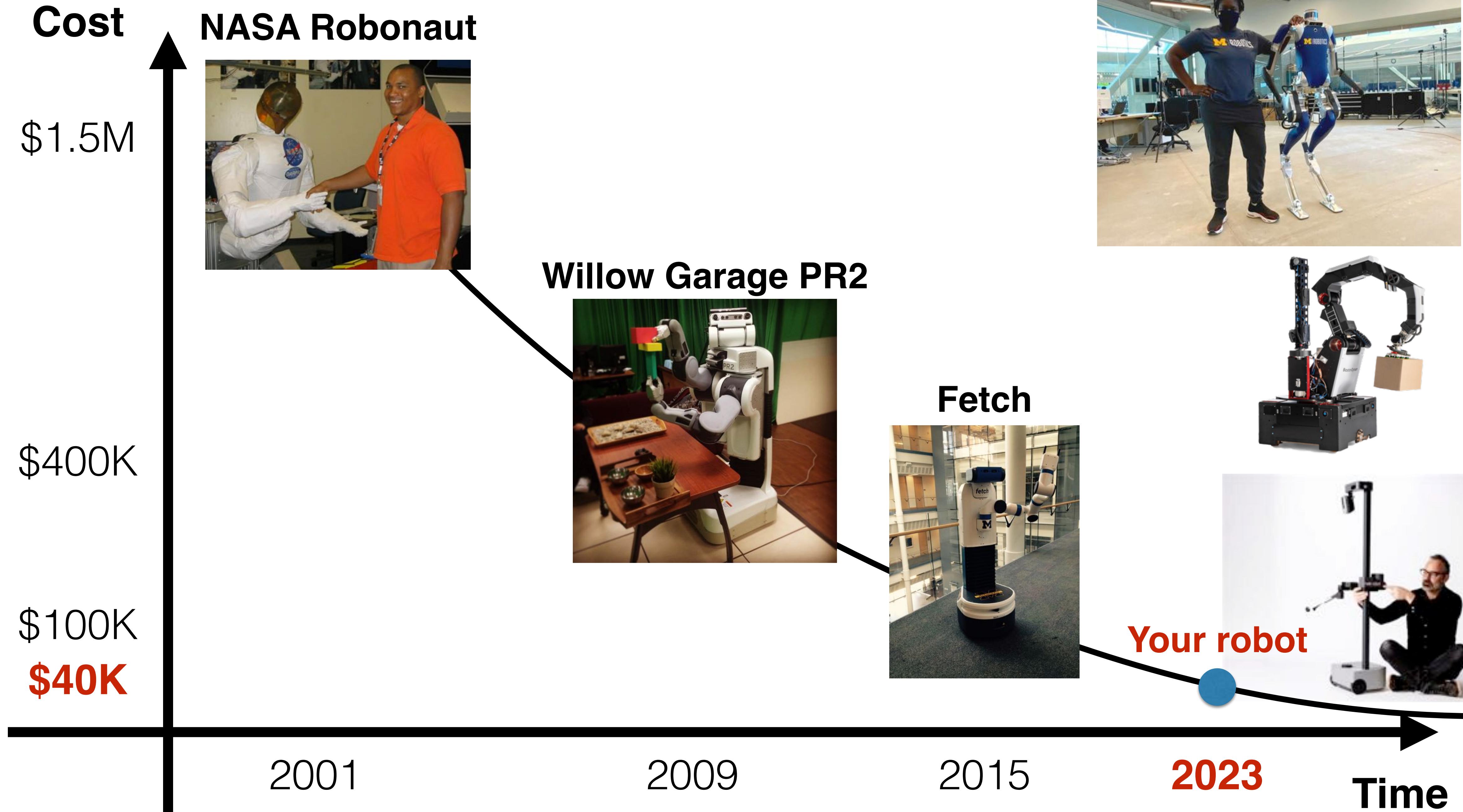


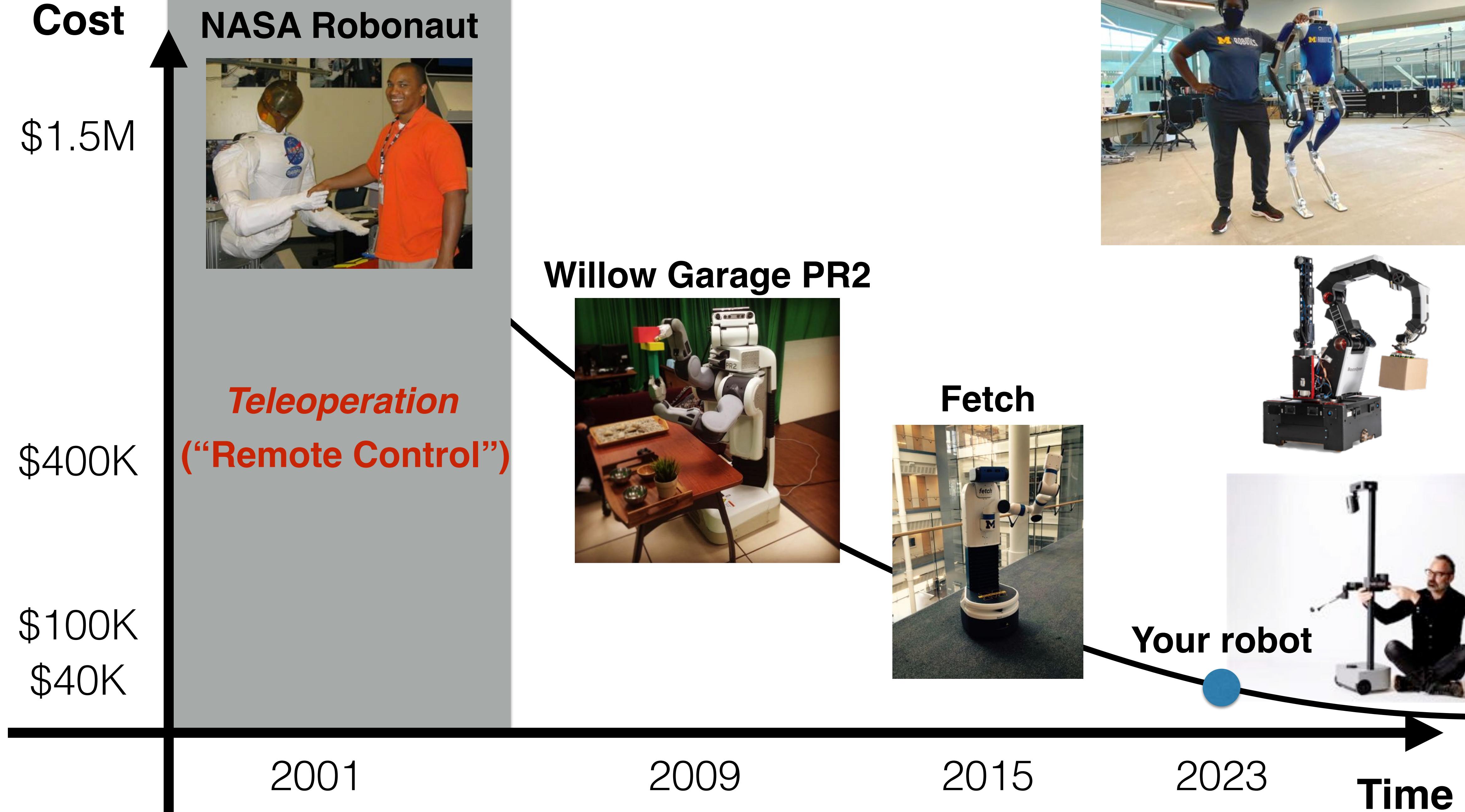
## Fetch

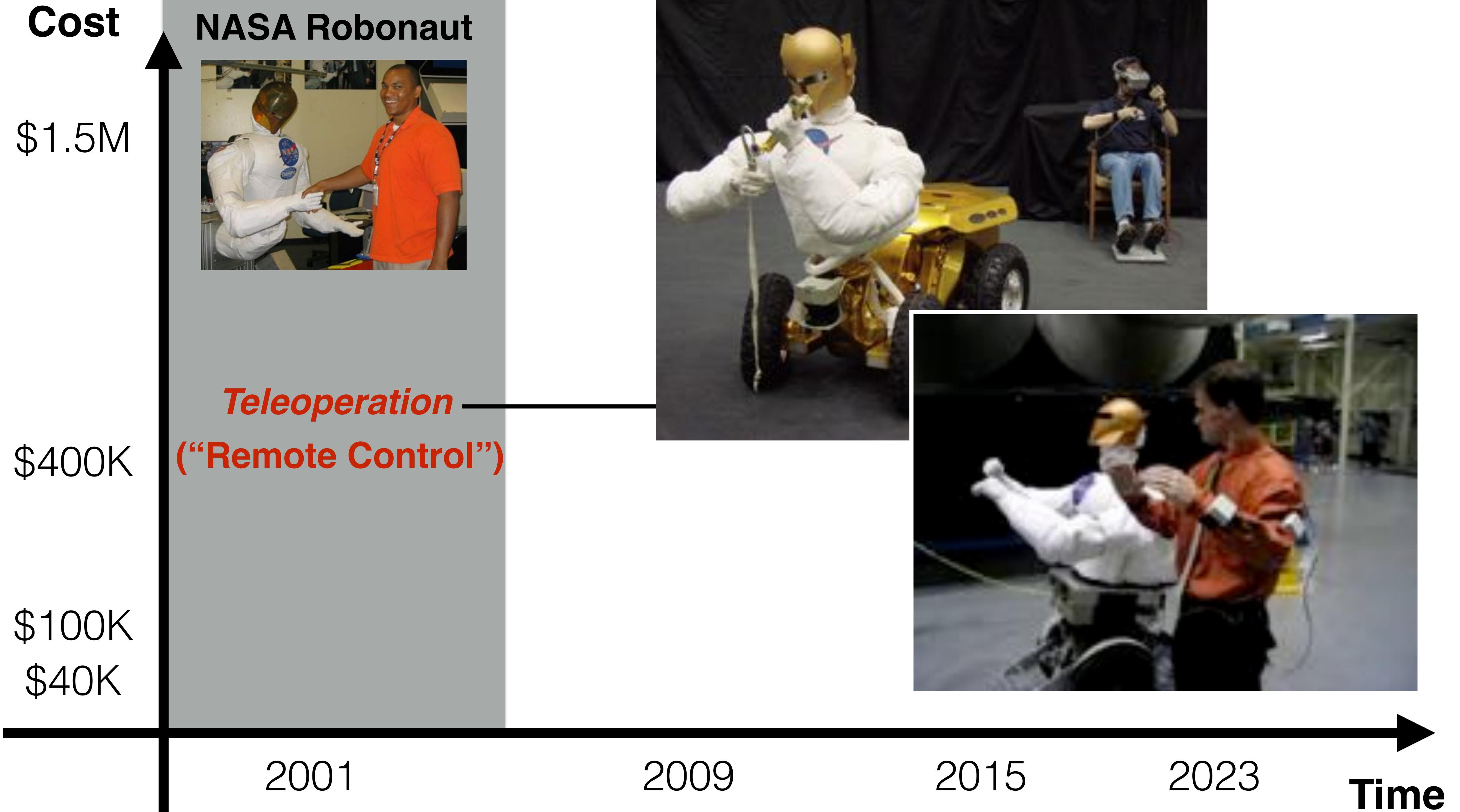


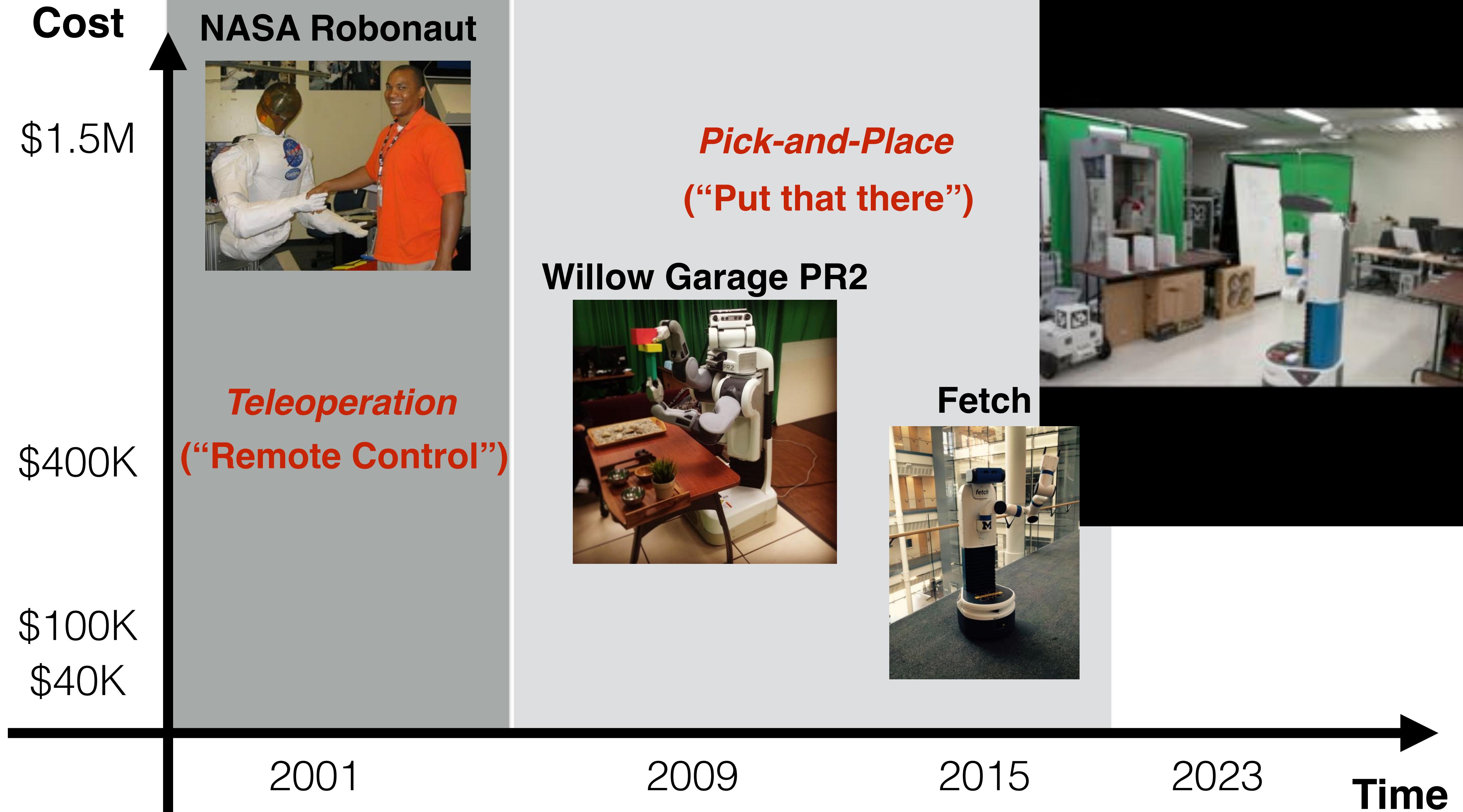


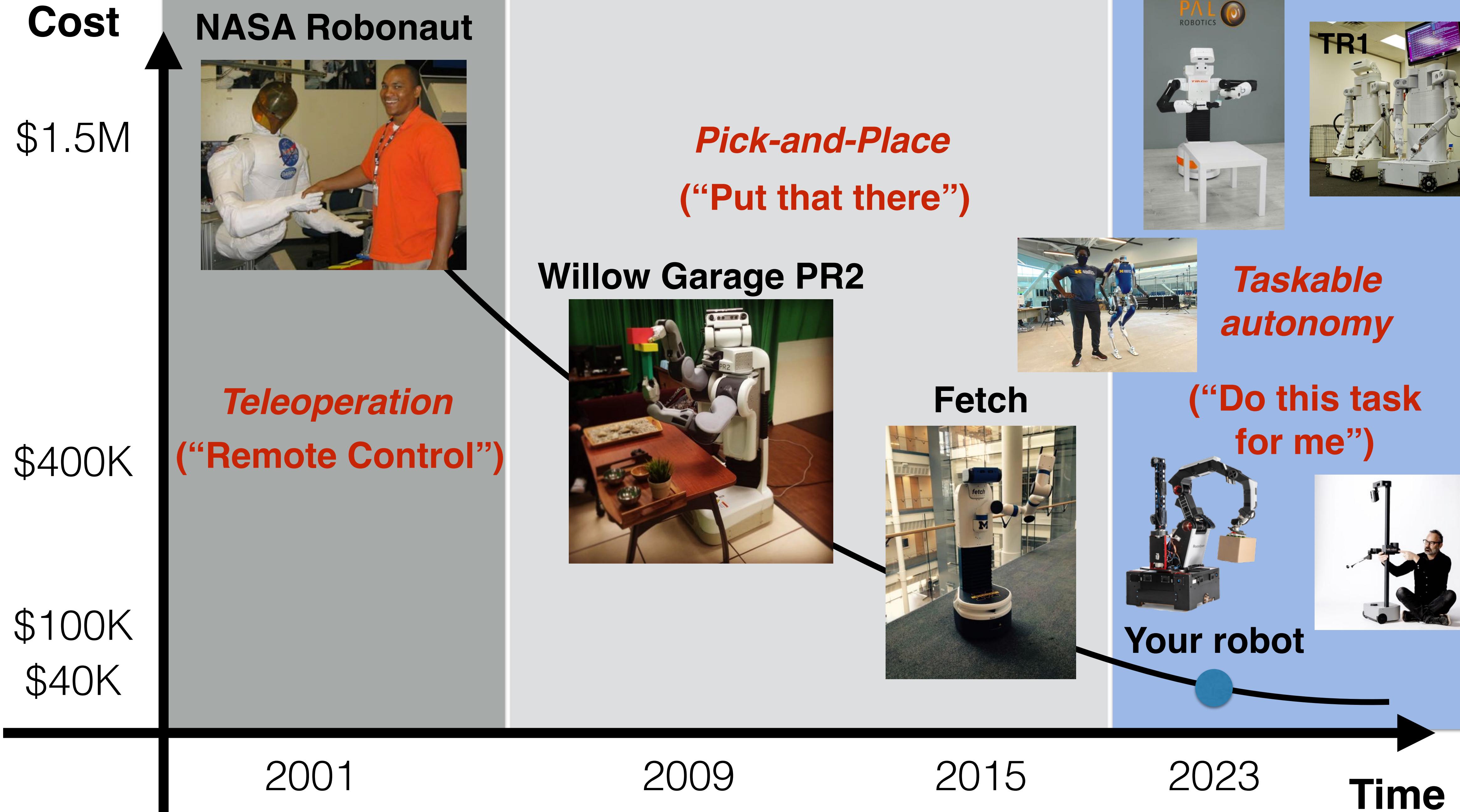










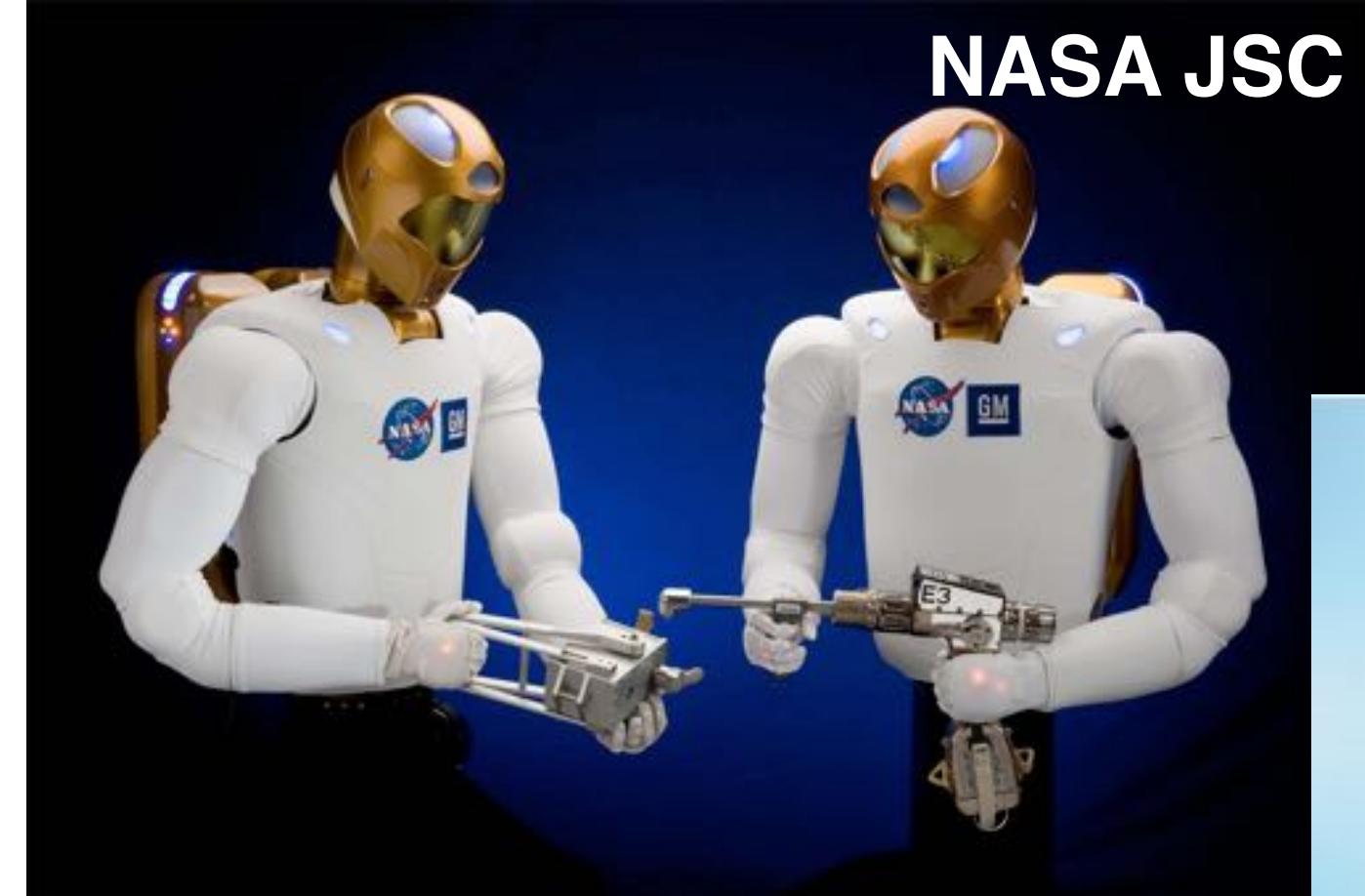




*Taskable  
autonomy*

*Pick-and-Place*

*Teleoperation*



TU Munich

*Dexterous Manipulation*



Harvard/Wyss





*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](#).

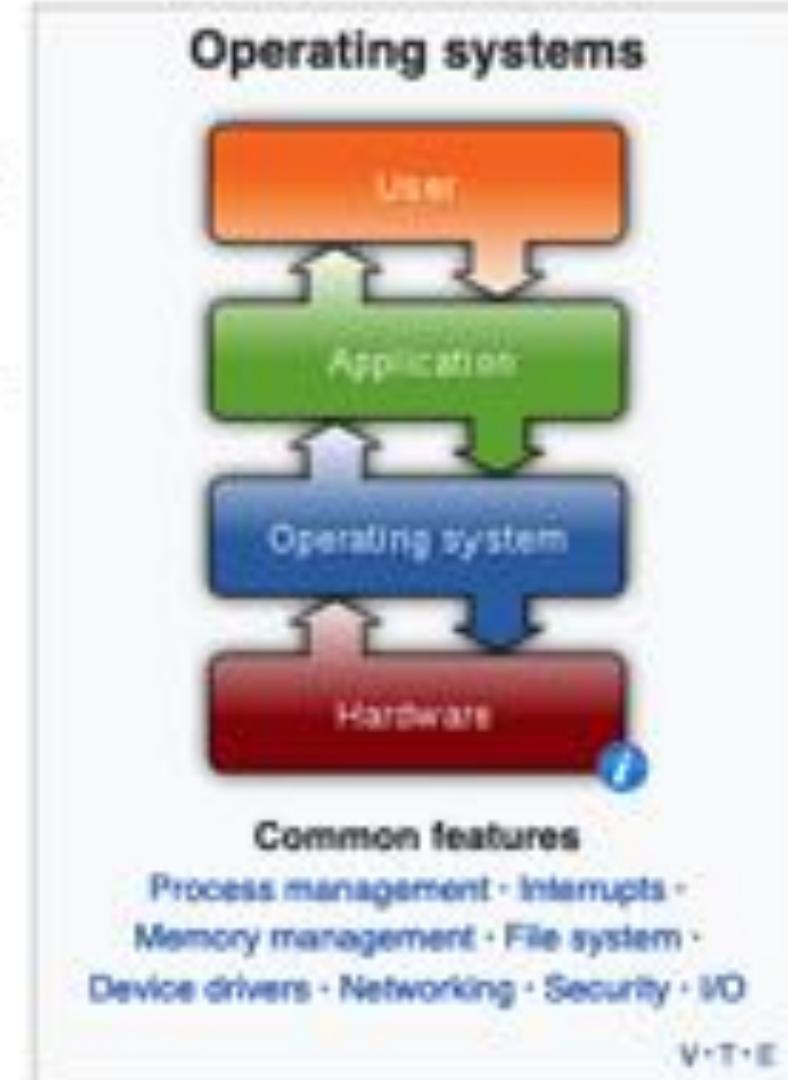
[Time-sharing](#) operating systems [schedule tasks](#) for efficient use of the system and may also include accounting software for cost allocation of processor time, [mass storage](#), printing, and other resources.

For hardware functions such as [input](#) and [output](#) and [memory allocation](#), the operating system acts as an intermediary between programs and the computer hardware,<sup>[1][2]</sup> 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%).<sup>[3]</sup> In the [mobile](#) sector (including smartphones and [tablets](#)), [Android's](#) share is up to 70% in the year 2017.<sup>[4]</sup> 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.<sup>[5]</sup> [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]

- 1 Types of operating systems
  - 1.1 Single-tasking and multi-tasking
  - 1.2 Single- and multi-user
  - 1.3 Distributed
  - 1.4 Timed
  - 1.5 Embedded





*Taskable autonomy*

*Dexterous Manipulation*

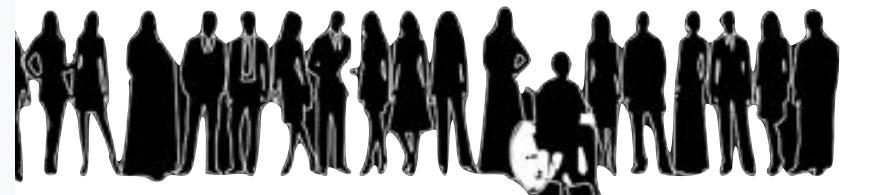
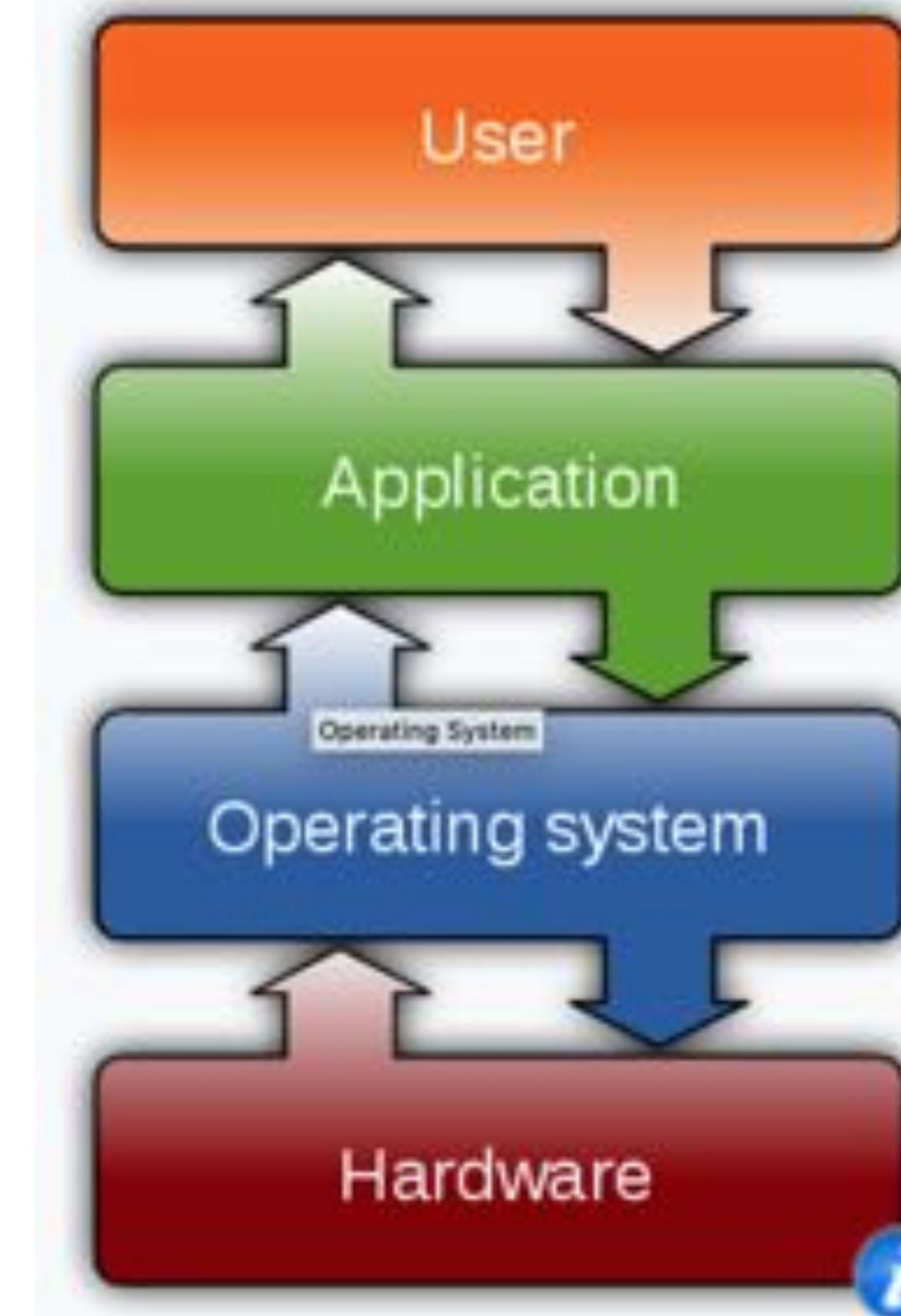
*Teleoperation*



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>

## Operating systems

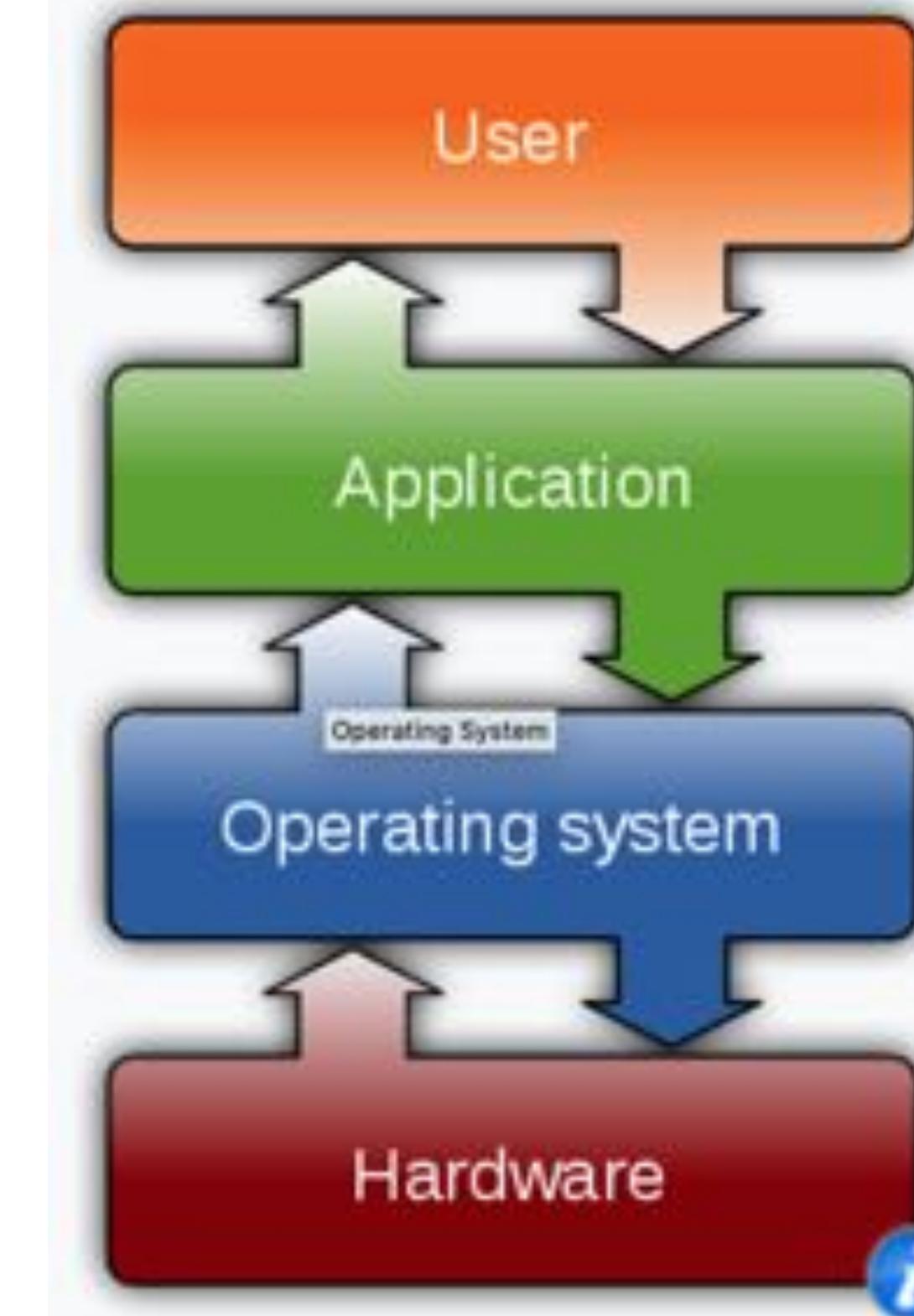




An **operating system (OS)** is a special program that runs on the bare machine and hides the gory details of managing processes and devices.

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

## Operating systems



Users

Robot Applications

*Dexterous Manipulation*



Then, what is this?

Operating System

Hardware



# Users

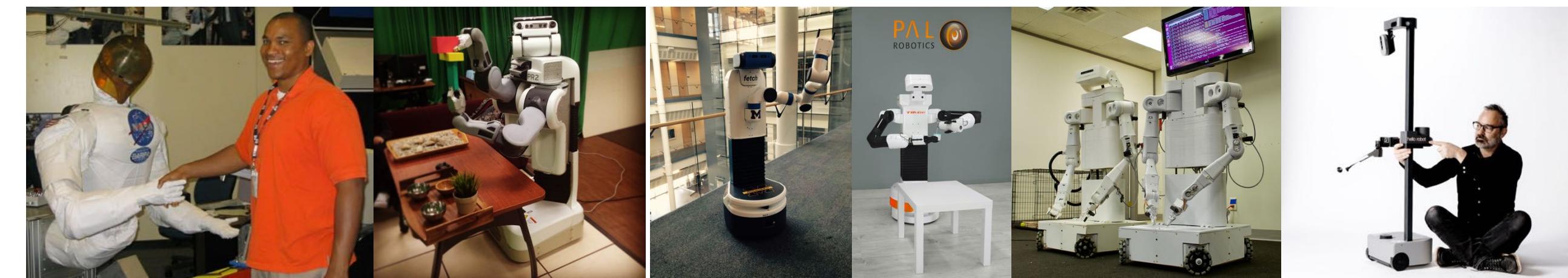
## Robot Applications

## 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



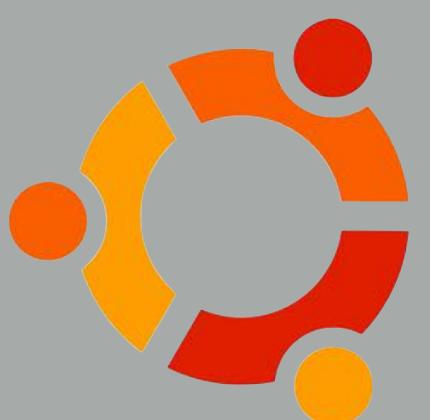
# Users

## Robot Applications

## Robot Operating System

## Operating System

## Hardware



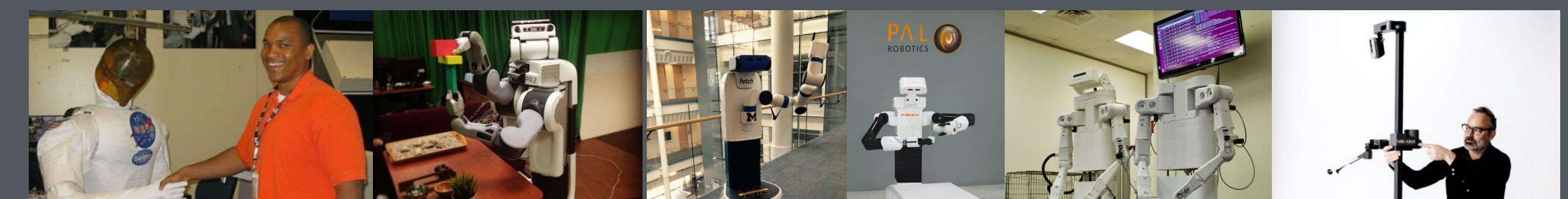
# Users

## Robot Applications

## Robot Operating System

## Operating System

## Hardware



# Users



## Robot Applications

## Robot Operating System

## Operating System

## Hardware

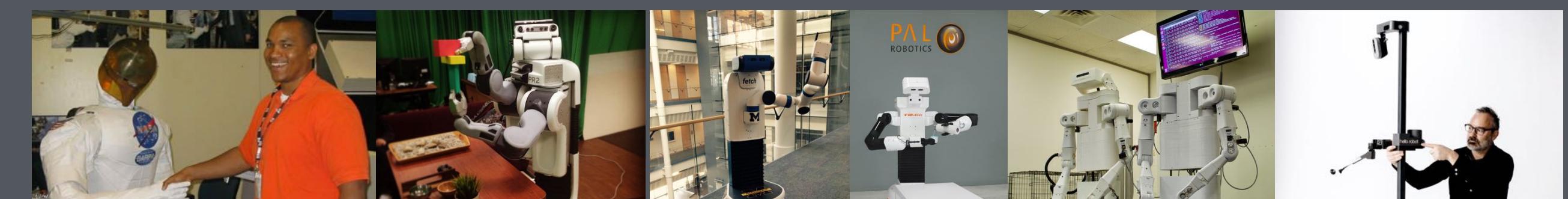
LCM

ROS

Player

YARP

MOOS



Users



Robot Applications

Robot Operating System

Operating System

Hardware

Then, what is this?

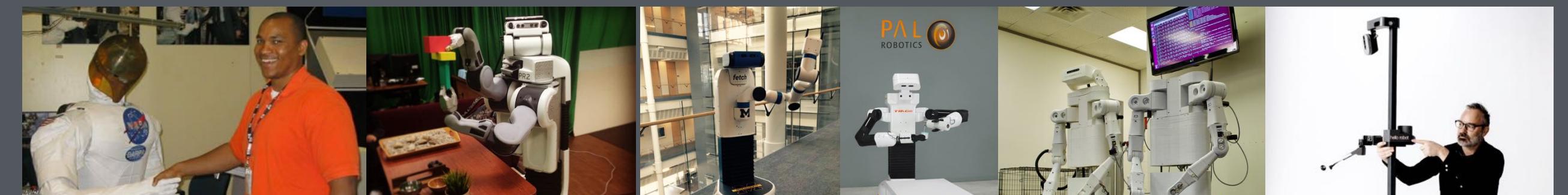
LCM

ROS

Player

YARP

MOOS



# Users



## Robot Applications

## Robot Operating System

## Operating System

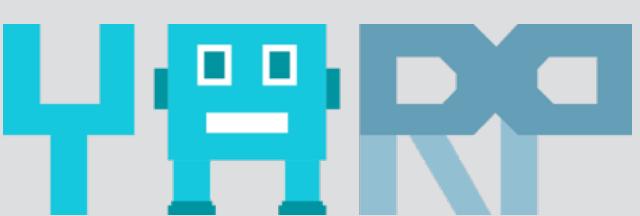
## Hardware

**Apps of the Future...**  
“Do this task for me”

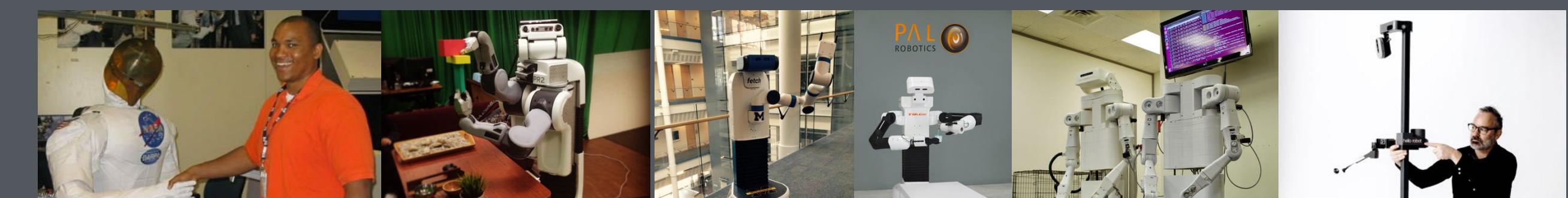
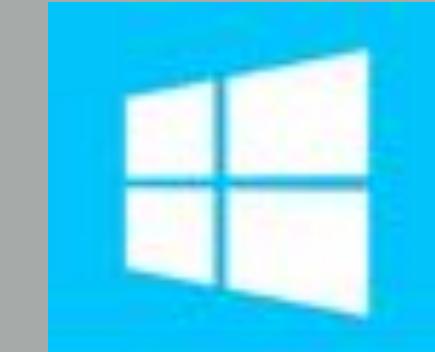
**LCM**

 **ROS**

 **Player**







“Do this task for me”

Can we make your  
**world programmable ?**



**MapIt!**  
Autonomous exploration and mapping for any indoor environment.

Click to Buy (\$49.99)

Bloomberg Technology N.Y. State Faces Outbreaks; U.S. Cases Ticking Up: Virus Upd... Salt Mobile Rebuffed Liberty Prior to Sunrise Deal, FT Repor... TikTok Assets Can't China's Approval

## SoftBank Robotics Plans App Store for Humanoid Pepper Robot

By Giles Turner  
March 1, 2017, 4:54 AM EST Updated on March 1, 2017, 10:37 AM EST

- Pepper is currently focused on business-to-business uses
- SoftBank Robotics plans to open up platform to developers

LIVE ON BLOOMBERG: Watch Live TV > Listen to Live Radio >

RobotShop Community Together, towards a world full of robots

Get Started Sign In Dashboard Forums Tutorials Robots Blogs News Leaderboards Shop Support Search for topic...

Consumer Robotics | GoRobotics >> Education | Professional and Research Robots | Robot Ethics | Robotic News

## New Apps on the MyRobots App Store

Posted on 21/11/2012 by carlos-31 in Cloud Robotics Tags: MyRobots App Store, MyRobots.com

Like Comment Share

The ROBOTIC CLOUD Developers Submit an App Follow Us SHARE IT

RoboControl RoboChat RoboServer RoboServer Lite

About carlos-31 View more by this author You may like iRobot RP-VITA - Telepresence

# Can we make a robot app store ?

**Robot App Store**

With Robot-App™ Store in the Cloud, your robots are always up-to-date with the coolest apps. To start, choose a robot or a Robot-App™.

**HOT!** Roomba Driver-Android

Use this app to tease your pets, race, or ask for sweets from someone in the kitchen

Browse Robot-Apps™ by robot: ENHANCED BY Google

Roomba AR.Drone Sphero OTHER BIOLOID



# What's a robot app?

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

**MapIt!**  
Autonomous exploration and mapping for any indoor environment.



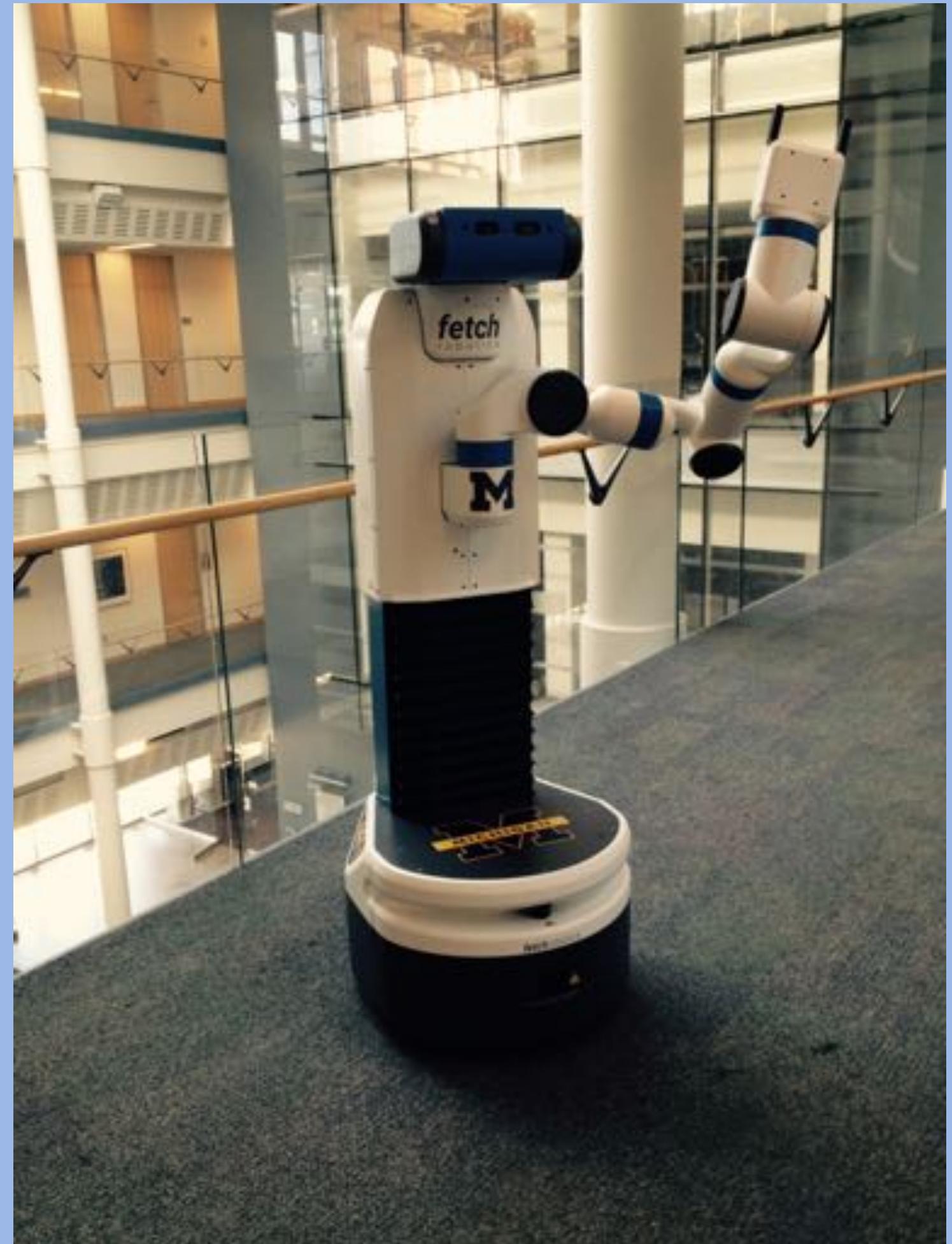
Click to Buy (\$49.99)

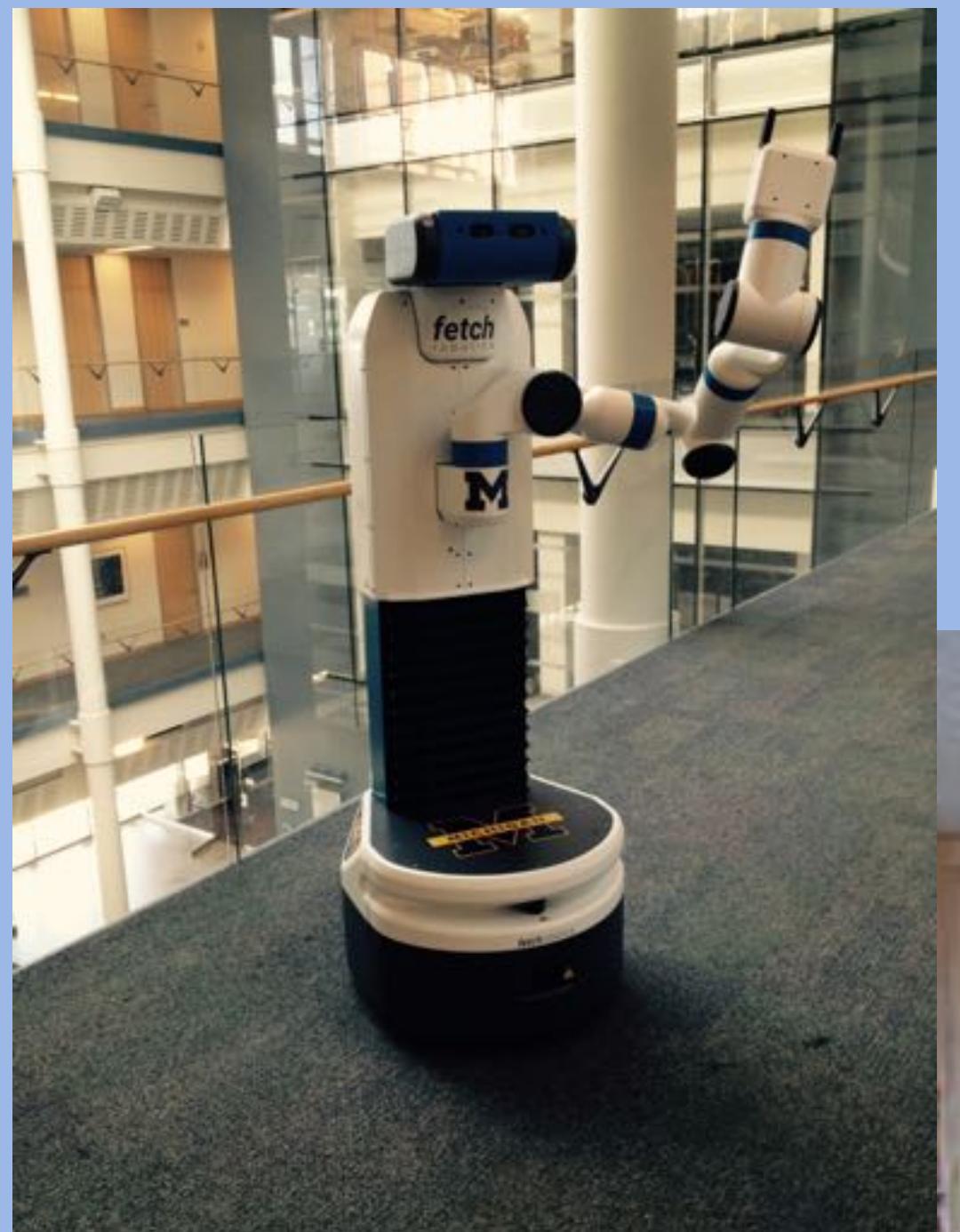
# 2009



# 2020

The image shows a comparison between 2009 and 2020 technology. On the left, there's a screenshot of the MapIt! app from 2009. In the center, there's a screenshot of the Canvas by Occipital app from 2020, which includes iPad screenshots showing "Capture", "Measure & Review", and "Share" features. On the right, there's a close-up of a smartphone camera module with three lenses, labeled "10 MP" and "12 MP". Above the camera module, the text "LiDAR Scanner" is displayed, indicating the presence of a LiDAR sensor in modern phones.





# Use any robot $x$



# to perform any task $y$

# in any environment $z$



# The 3Ds: Dirty, Dull, and Dangerous

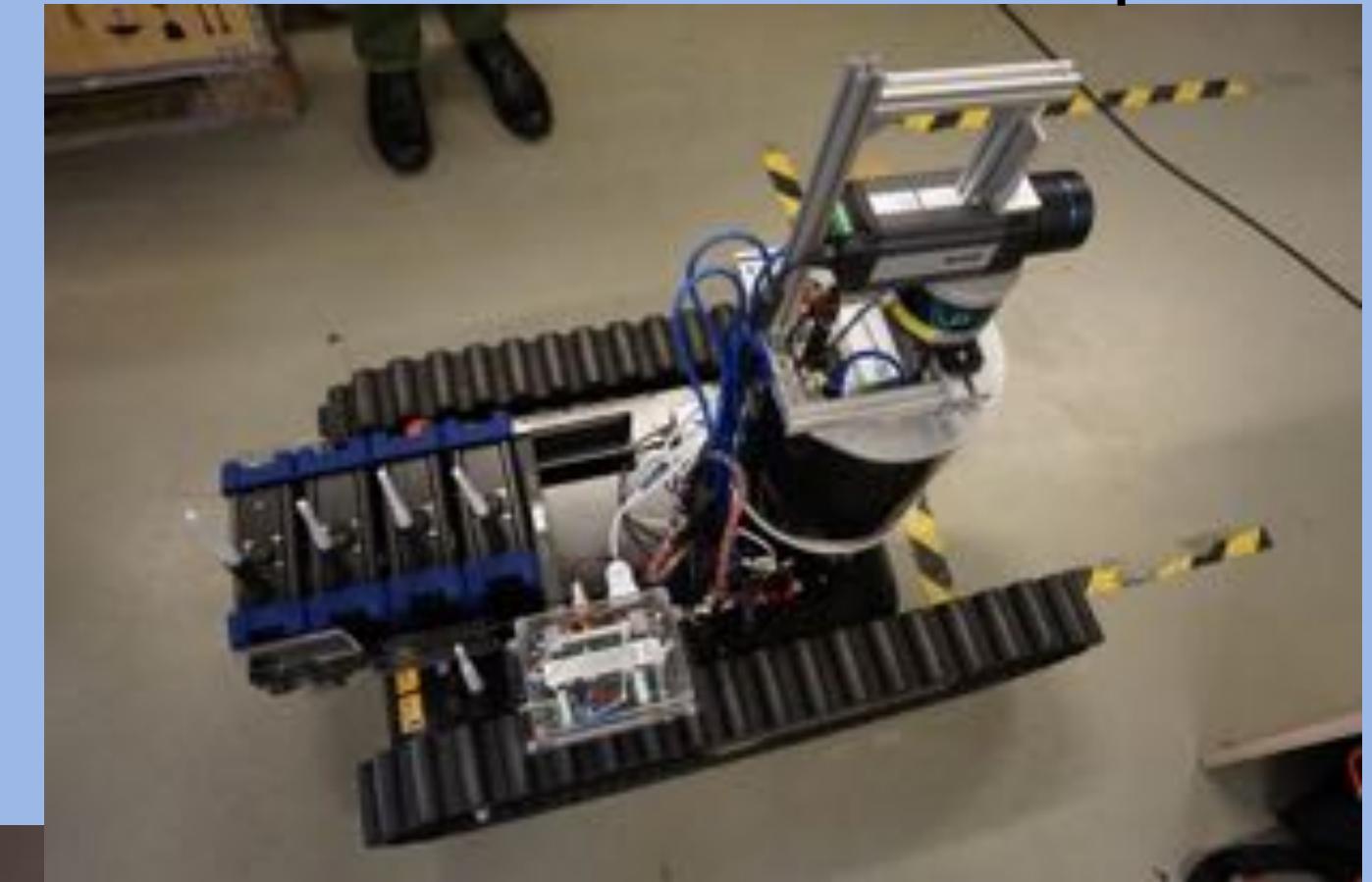
## “Autonomous” Driving



## Infrastructure inspection



## Nuclear cleanup



<https://www.shadowrobot.com/blog/robots-saving-humans-from-dangerous-jobs/>

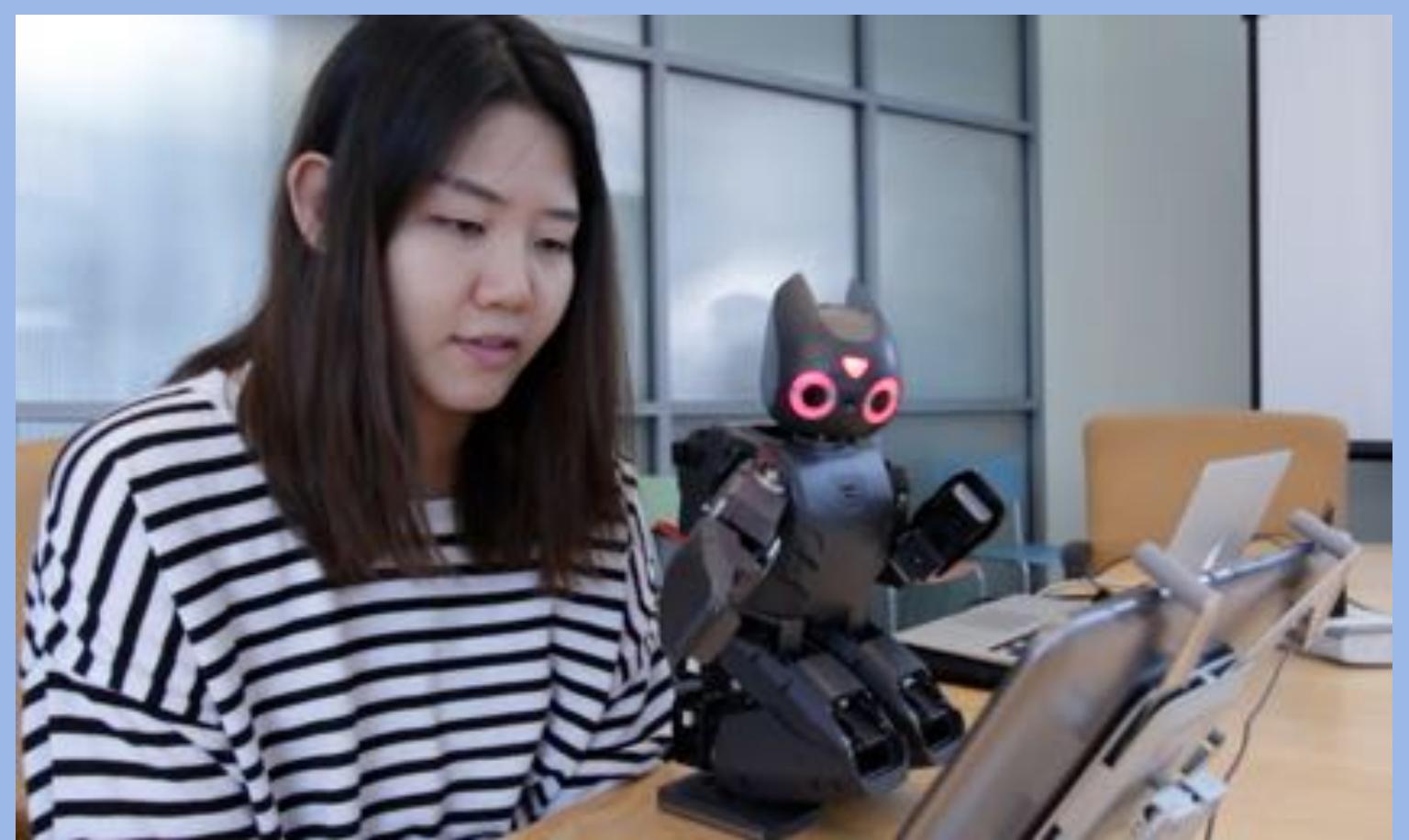
<https://techcrunch.com/2018/06/05/remote-control-driverless-car-startup-partners-with-vehicle-manufacturers/>





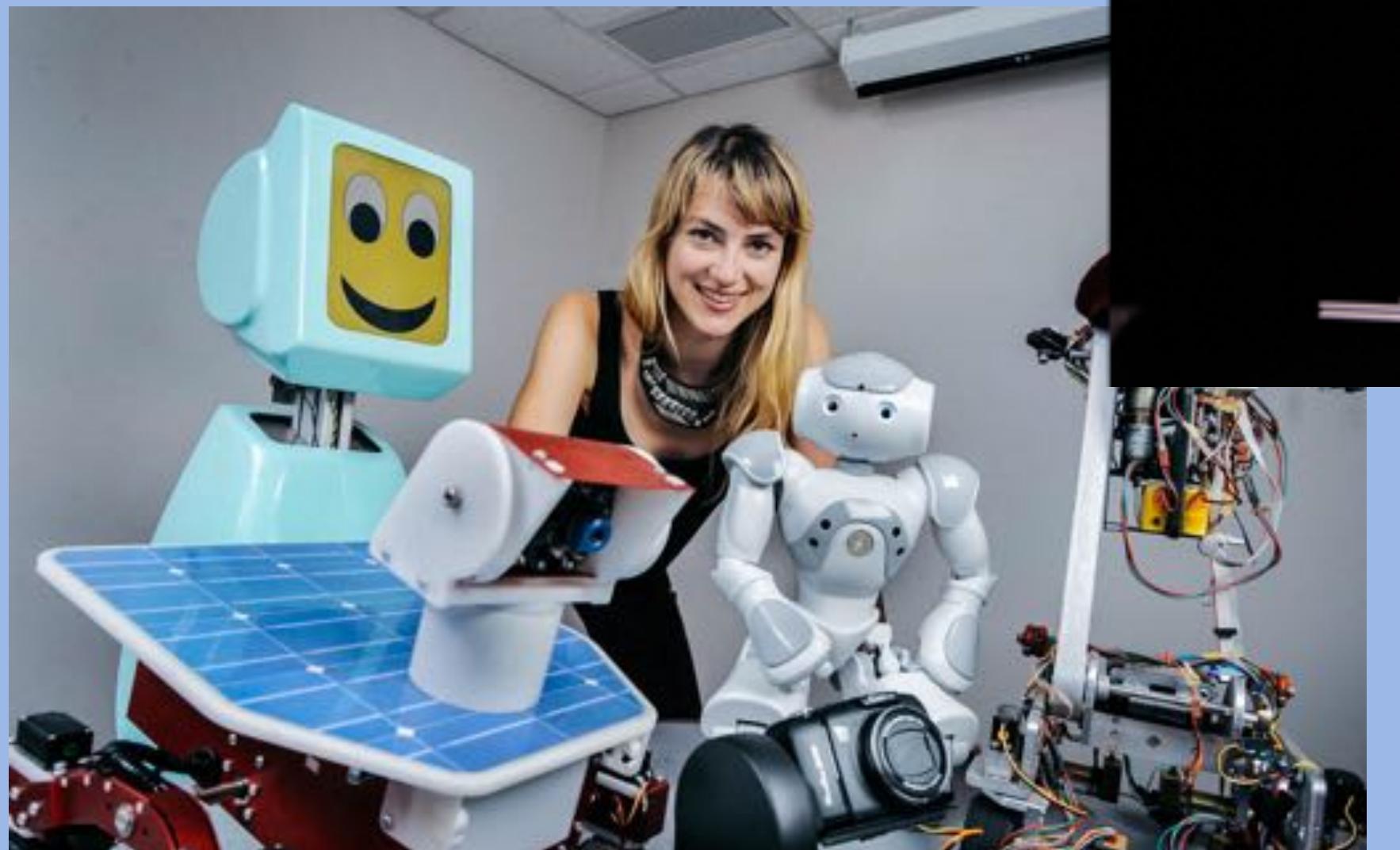
Autism treatment

# Social Robotics

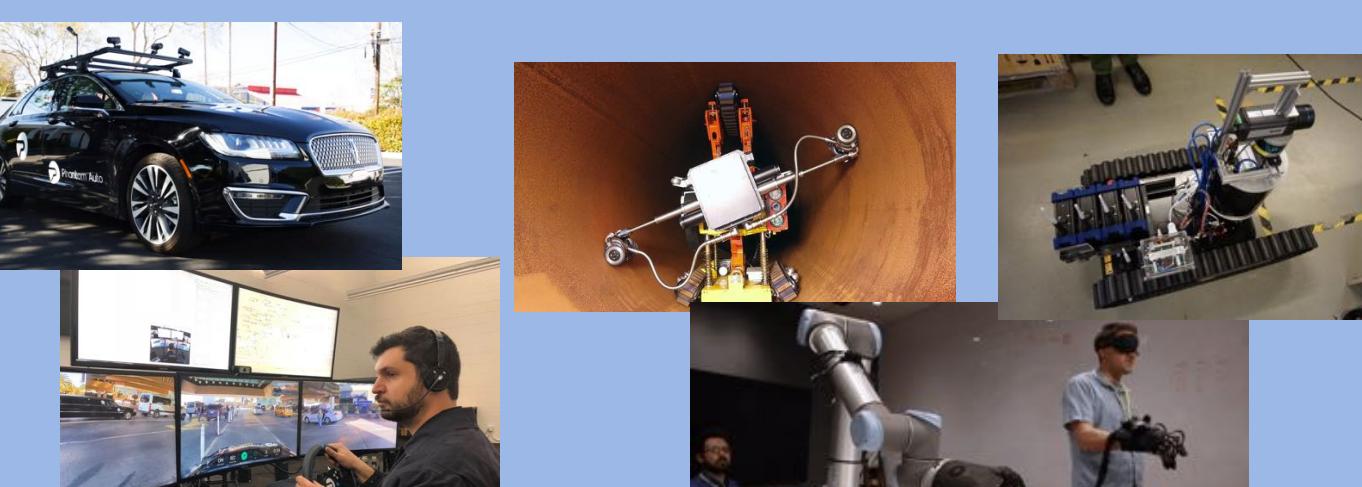


Education

Entertainment



Rehabilitation



Elder care



# Agriculture



# Exploration



# Manufacturing



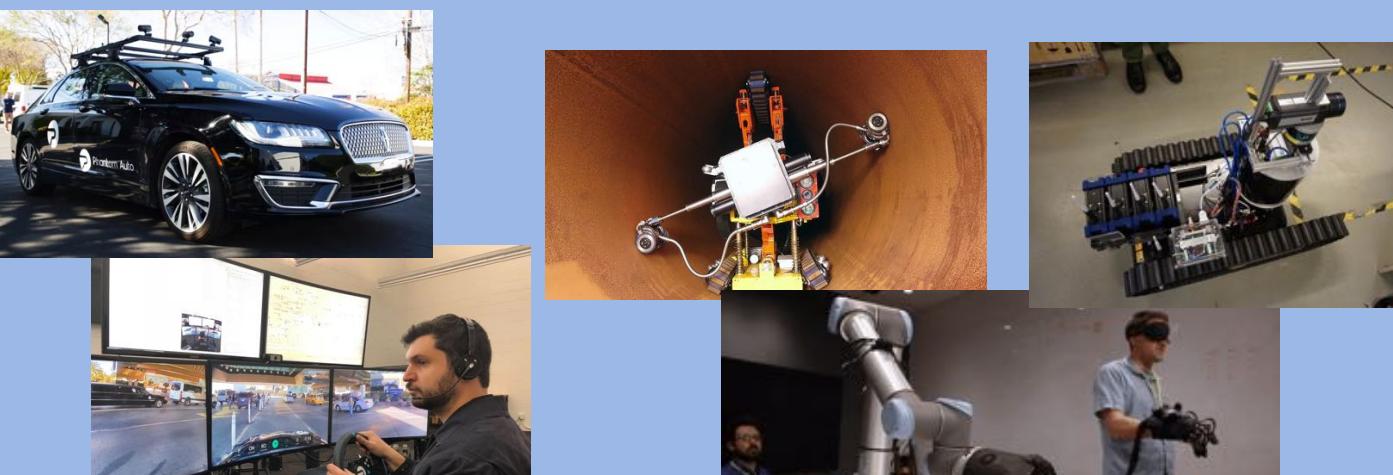
# Security



# Social Robotics



# Dirty, Dull, Dangerous



# Lethal Force



# Medicine



# Users



# Robot Applications

Custom applications,  
Taskable autonomy research

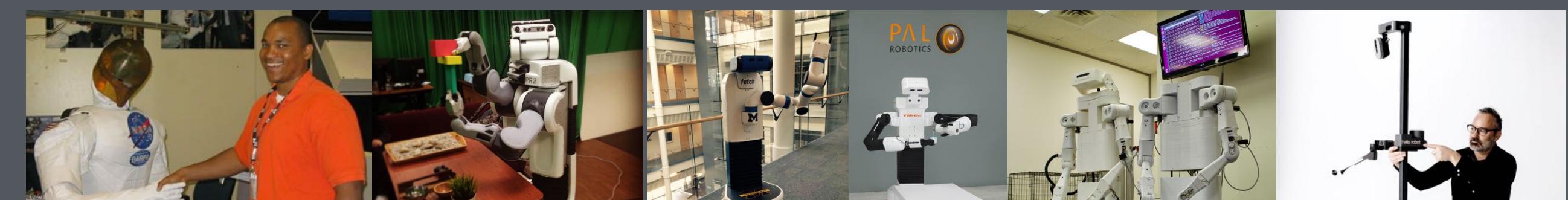
# Robot Operating System



# Operating System



# Hardware



# Users



# Robot Applications

Custom applications,  
Taskable autonomy research

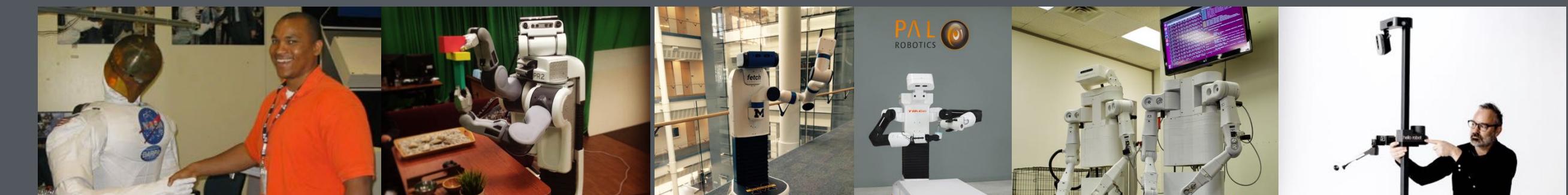
# Robot Operating System



# Operating System



# Hardware



# Users



# Robot Applications

Custom applications,  
Taskable autonomy research

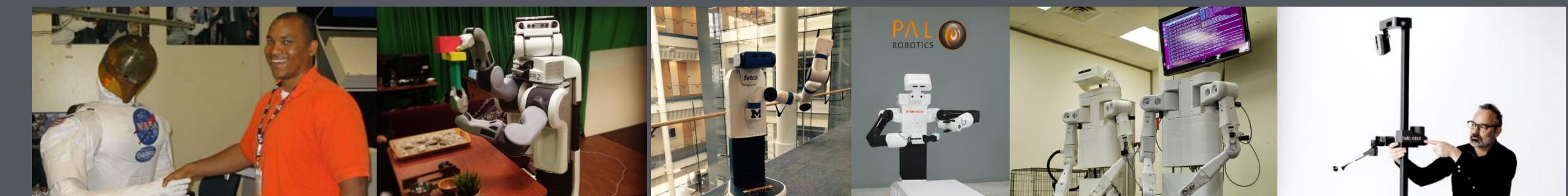
# Robot Operating System

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

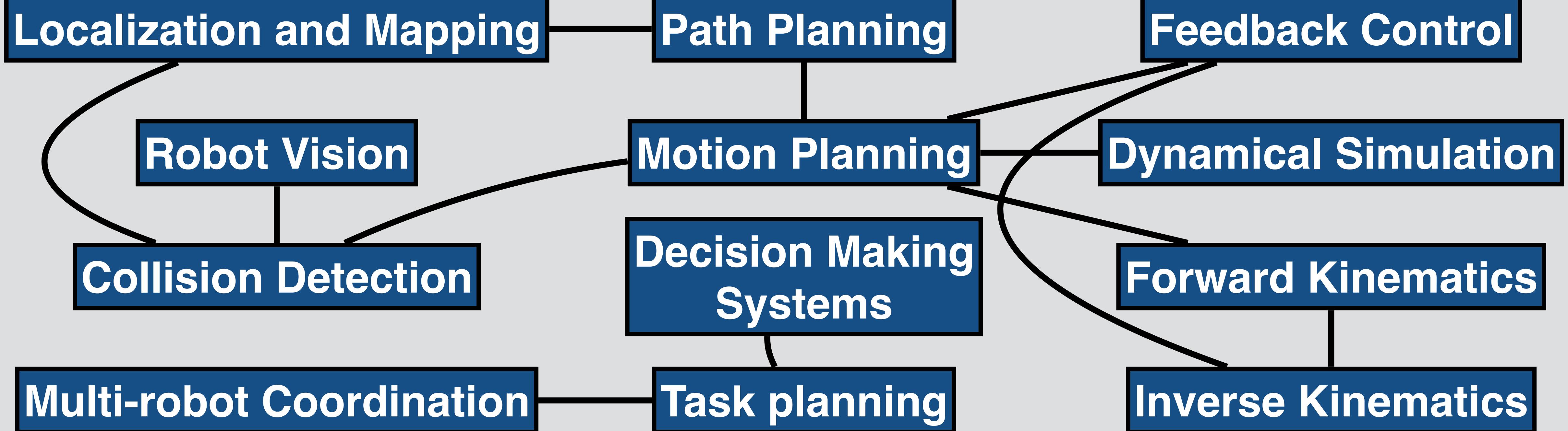
Task planning

Inverse Kinematics



# Robot Operating System

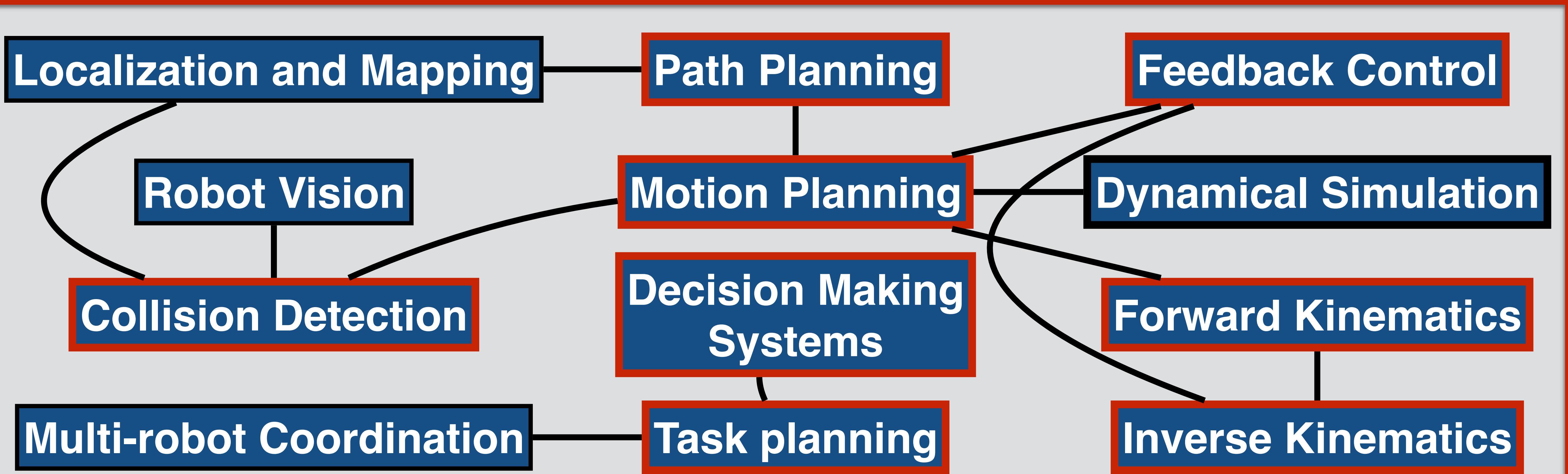
Build your own Robot OS



Robot Middleware Architecture (via Interprocess Communication)

# Robot Operating System

Covered at breadth in CSCI5551



# Robot Middleware Architecture (via Interprocess Communication)



# Users

## Robot Applications

## Robot Operating System

## Operating System

## Hardware

Work with  
a real robot  
once this  
semester



# Course Resources

- Course Website: <https://rpm-lab.github.io/CSCI5551-Fall23-S2/>
- EdStem: <https://edstem.org/us/courses/46318/discussion/>
- Canvas: You should see it now!



# Course Website

The screenshot shows a web browser window displaying a course website. The title bar reads "Home | CSCI5551-Fall23". The address bar shows the URL "rpm-lab.github.io/CSCI5551-Fall23-S2/". The page itself has a yellow header bar with the text "CSCI5551-Fall23" and a search bar. Below the header, there is a sidebar on the left with links to "Home", "Syllabus", "Calendar", "University Policies and Resources", and another "Home" link which is currently highlighted. The main content area features a large heading "Introduction to Intelligent Robotic Systems", followed by "CSCI5551-02 Fall 2023 at The University of Minnesota - Twin Cities", and the meeting time "M, W 1:00PM-2:15PM CT - Rapson Hall 45". A detailed description of the course follows, mentioning robotics principles, 3D transformations, robot kinematics, path planning, configuration spaces, sampling-based planning, motion control algorithms, state estimation, mapping, localization, and SLAM. It also notes hands-on experience in Java threejs, real-world robots, virtual counterparts, and a final project.

**CSCI5551-Fall23**

Search CSCI5551-Fall23

Ed Forum RPM Lab

Home Syllabus Calendar University Policies and Resources Home

## Introduction to Intelligent Robotic Systems

CSCI5551-02 Fall 2023 at The University of Minnesota - Twin Cities

M, W 1:00PM-2:15PM CT - Rapson Hall 45

The goal of this course is to introduce students to robotics principles, covering key topics such as 3D transformations, robot kinematics, forward and inverse kinematics, path planning, configuration spaces, sampling-based planning, basic motion control algorithms, and state estimation for mobile robots, which includes mapping, localization, and SLAM. Students will gain hands-on experience in programming robots in the [Java threejs](#) environment. In later projects, they will have the opportunity to control real-world robots using their virtual counterparts and codebase. There will be an open-ended final project where students can apply their skills acquired throughout the semester to explore new ideas. They will present their projects to a wider audience through poster presentations and demos.



# Meeting Logistics

- In-person Lectures
  - Mon & Wed 1:00-2:15 PM  
Central Time
  - Rapson Hall 45
  - No recordings
- Office Hours
  - Posted on the website, or by appointment



# Course Structure

- **Objective:** Give you the computational skills to understand the nuts and bolts of developing a robotic system using kinematics and dynamics.
- **Project focused class:**
  - 7 total projects: building in complexity from basic transformations-rotations to motion planning of a dual-arm robot manipulator

# Course Structure

The screenshot shows a web browser window with the title "Calendar | CSCI5551-Fall23". The URL is "rpm-lab.github.io/CSCI5551-Fall23-S2/calendar/". The page has a yellow sidebar on the left with links: Home, Syllabus, Calendar (which is highlighted in red), University Policies and Resources, and a footer note about Doceo. The main content is a table with columns: Lec #, Date, Topic, Project Announcement, Project Due, and Quiz. The table lists 31 lectures from September 6 to December 20, covering topics like Introduction, Linear Algebra Refresher, Representations I-III, Manipulation I-VII, Planning I-VII, Mobile Robotics I-IV, and Guest Lectures/OH. It also includes ROS/RosBridge Tutorial, Dual-arm setup, SLAM 1/2, and Poster Day.

Lec #	Date	Topic	Project Announcement	Project Due	Quiz
1	09/06	Introduction			
2	09/11	Linear Algebra Refresher			
3	09/13	Representations I - Transformations	P0: Gt, JS, Setup		Q1
4	09/18	Representations II - Rotations - Quaternions			Q2
5	09/20	Manipulation I - Kinematics	P1: Forward Kinematics	P0: Due	Q3
6	09/25	Manipulation II - Forward Kinematics			Q4
7	09/27	Manipulation III - Forward Kinematics	P2: Robot Dance		Q5
8	10/02	Manipulation IV - Inverse Kinematics		P1: Due	Q6
9	10/04	Manipulation V - Inverse Kinematics			Q7
10	10/09	Manipulation VI - Inverse Kinematics			Q8
11	10/11	Manipulation - New Frontiers	P3: Inverse Kinematics	P2: Due	Q9
12	10/16	Planning I - Path Planning			Q10
13	10/18	Planning II - Bug Algorithms			Q11
14	10/23	Planning III - Configuration Spaces			Q12
15	10/25	Planning IV - Sampling-based Planning	P4: Planning	P3: Due	Q13
16	10/30	Planning V - Potential Fields			Q14
17	11/01	Planning VI - Collision Detection			Q15
18	11/06	ROS, RosBridge, Tutorial			Q16
19	11/08	Planning - New Frontiers	P5: Dual-arm setup	P4: Due	Q17
20	11/13	Motion Control I - Intro, Open-loop vs Closed-loop			Q18
21	11/15	Motion Control II - PID			Q19
22	11/20	Mobile Robotics I - Mapping			Q20
23	11/22	Mobile Robotics II - Localization	P6: Dual-arm challenge	P5: Due	Q21
24	11/27	Mobile Robotics III - SLAM 1			Q22
25	11/29	Mobile Robotics IV - SLAM 2			Q23
26	12/04	Mobile Robotics - New Frontiers			Q24
27	12/06	Guest Lecture / OH	Open Ended Final Project Ideas		
28	12/11	Final Project Challenge Day - Drone Lab		Challenge Day	
29	12/13	Guest Lecture / OH			
30	12/18	Guest Lecture / OH			
31	12/20	Poster Day		Open Ended Final Project Presentation	

# Guided Projects

- Project 0
  - Introduction to Git, ThreeJs, Code Stencil
- Project 1
  - Forward Kinematics
- Project 2
  - Robot dance
- Project 3
  - Inverse Kinematics
- Project 4
  - Planning
- Project 5
  - Dual-arm setup
- Project 6
  - Dual-arm Challenge

# Final Project Challenge

- Open-ended and will let students explore ideas with their learnings from the course.

# Project Grading

- Guided Projects 0-6
  - 2 total late days available
  - 25% daily penalty after deadline.
- Open-ended Final Project
  - No late days

# Overall Grading Policy

The screenshot shows a web browser window with the title "Syllabus | CSCI5551-Fall23". The URL in the address bar is "rpm-lab.github.io/CSCI5551-Fall23-S2/syllabus/#grading-policy". The page content is as follows:

**CSCI5551-Fall23**

**Grading Policy**

Course grades will be determined according to the following criteria:

- Project 0: 5%
- Project 1: 10%
- Project 2: 10%
- Project 3: 10%
- Project 4: 10%
- Project 5: 10%
- Project 6: 10%
- Final Project: 15%
  - Project proposal document: 3%
  - Project evaluation by guests: 6%
  - Poster presentation (+video, +slides): 6%
- 20 Pre-Lecture Quizzes: 20% (1% each)

The grading in this course is on an absolute scale. This means that the performance of others in the class will not affect your grade. Letter grades will be assigned using the following scale:

- A: ≥ 93.0%
- A-: ≥ 90.0% and < 93.0%
- B+: ≥ 87.0% and < 90.0%
- B: ≥ 83.0% and < 87.0%
- B-: ≥ 80.0% and < 83.0%
- C+: ≥ 77.0% and < 80.0%
- C: ≥ 73.0% and < 77.0%
- C-: ≥ 70.0% and < 73.0%
- D+: ≥ 67.0% and < 70.0%
- D: ≥ 60.0% and < 67.0%
- F: < 60.0%

For S/N grading, a satisfactory grade (S) requires a grade of 70.0% or above.

# Collaboration Policy

- All work submitted must be your own.
  - All code submitted must comply with College of Engineering Honor Code.
- Cheating will not be tolerated and can lead to termination from the program.
- No code can be communicated, including verbally.
  - Explicit use of external sources must be clearly cited.
- Free flow of discussion and ideas is encouraged.



# University Policy

The screenshot shows a web browser window with the title bar "University Policies and Resources". The address bar displays the URL "rpm-lab.github.io/CSCI5551-Fall23-52/policies\_resources/". The main content area is titled "CSCI5551-Fall23". On the left, a yellow sidebar contains links: "Home", "Syllabus", "Calendar", and "University Policies and Resources" (which is highlighted). The main content area includes sections for "Standard University Policies", "Mental Health Information", "Disability Information", and "Acknowledgments".

**Standard University Policies**

This class follows standard university policies. It's your responsibility to be familiar with:

- Student conduct code
- Academic dishonesty
- Makeup work for legitimate absences
- Student responsibilities
- Grading and transcripts
- Sexual harassment
- Equity, diversity, equal opportunity, and affirmative action
- Safety in classroom and campus

**Mental Health Information**

The Department of Computer Science & Engineering strives to ensure all students have access to resources that will help them feel safe and supported. We are deeply committed to the wellbeing of all students, staff, and faculty. Your mental health is part of who you are and if you are unsure where to turn, we are here to help you. If you are encountering challenges, I encourage you to visit our department's [Mental Health Resources](#) website and connect with one of our Mental Health Advocates, who are available to answer questions about campus mental health resources and services.

**Disability Information**

University policy is to provide, on a flexible and individualized basis, reasonable accommodations to students who have documented disability conditions (e.g., physical, learning, psychiatric, vision, hearing, or systemic) that may affect their ability to participate in course activities or to meet course requirements. Students with disabilities are encouraged to contact the [Disability Resources Center](#) (DRC) to discuss individual needs for accommodations.

If you have already been working with the Disability Resource Center and have a letter from them to help direct your instructors on to set you up for success, then please send that to me at the beginning of the semester. In many cases, you will find that this course is already designed to be accessible for students that would benefit from additional flexibility (e.g., non-timed quizzes instead of large exams, flexible late policy on programming assignments, etc.). However, if additional accommodations are necessary, I will make every reasonable effort to make sure this class is a good experience for you.

**Acknowledgments**

Aspects of this syllabus (including this statement) were adapted from Evan Suma Rosenberg, Nathan Taylor, Daniel Keefe, Blair MacIntyre, Shana Watters, Lana Yarosh, and the American Association of University Professors Joint Statement on Rights and Freedoms of Students, because writing a good syllabus is hard.

# Discussion Forum

## EdStem

- EdStem is the discussion forum used in this course.
- Discussion of quizzes and verbatim code must be private.
- You will be added to it this week.



# Next lecture: Linear Algebra Refresher

