

Deck 01 · What this course is, and the stack we will fill in

CS 549 · Learning for Robotics · sessions 1.1–1.2

Özgür S. Öğüz · Bilkent University · Fall 2026

What does a robot need in order to pick something up?

Which of those parts can be learned, and what happens when they are?

"Learning for robotics", in this course

Learning to make sequential manipulation decisions in the physical world.

- The classical stack: frames, kinematics, optimization, planning, control
- Perception for manipulation: cameras, point clouds, grasps
- Imitation learning and reinforcement learning
- **Vision-language-action models:** images and words become motor commands

What we do not cover

- The internals of machine learning, vision, or language models
- Control theory as a discipline: no stability proofs, no observer design
- System identification, estimation theory, mechanical design
- Large vision-language models in depth: **that is CS 687**

The thesis of the course

Build the classical manipulation stack, component by component.

Then, for each component, study the learned component that **replaces** it, what it **keeps**, and what it **silently assumes**.

Why now (facts dated 2026-09-04)

What changed	Evidence
Open VLAs at two scales	OpenVLA 7.5 B (2024); SmolVLA 450 M, fine-tunes on one consumer GPU (2025)
The benchmark is saturated	LIBERO above 95 %; perturbation suites exist because it memorises
RL improves a VLA	$\pi^*0.6$ trained with RL from real experience (Nov 2025)
The field moved	about 164 VLA submissions at ICLR 2026, against 9 the year before

Motion as the driving force of intelligence

Why robotics? Embodied intelligence.

Why this course?

- Machine learning, deep learning, vision, AI: *computational*
- Neuroscience, cognitive science: *descriptive*
- Manipulation forces us to acknowledge the **structure** of the real world: geometry, physics, objects

Shakey (SRI, 1966–72): [the first robot that perceived, reasoned, and acted](#)

"We have a brain for one reason and one reason only: to produce adaptable and complex movements." — Daniel Wolpert



A tunicate (sea squirt): swims as a larva, settles, and digests its own brain

We are not just controlling the robot

Dynamics and kinematics

goal: orchestrate joint motion to produce the desired motion in task space

Planning and optimization

goal: plan through contact and around obstacles; optimize trajectories

Control theory

goal: track the planned motion; design optimal controllers

all of it
+ perception
+ learning

The state we care about is **state of the robot × state of the environment.**

Types of robotics, and which one this is

- **Mobile robotics** — (mostly) about localisation and mapping
- **Kinematic and dynamic motion control** — the same as manipulation, but without ever making it to the interaction
- **Manipulation** — all about interacting with the world



Manipulation in one strip: reach, touch, slide, lift. The same task returns as a robot clip on "Why hands are hard".

Practical or theoretical?

"There is nothing more practical than a good theory." — Kurt Lewin, 1951

- Reduce real problems to mathematics that can be solved efficiently
- Then check what the reduction threw away
- Required (basics): linear algebra, probability, Python — and a lot of self-learning and **perseverance**

What earlier students found hardest was the simulator setup, not the theory. This year the install check is on Day 1, and the TAs' tutorial comes before anything is graded.

Where the field talks

Robotics-first

- CoRL, RSS, ICRA, IROS
- T-RO, RA-L, IJRR

Learning-first

- NeurIPS, ICML, ICLR; CVPR, ICCV
- JMLR, TMLR, TPAMI

Tools we use, and why

Tool	Our use
RAI (<code>robotic</code> , TU Berlin): KOMO, RRT, BotOp	A1 and A2 ; scenes as <code>.g</code> files
Drake (MIT / TRI)	its notation for frames (week 2); demos in class
LeRobot + LIBERO (Hugging Face)	A3 ; runs on a Colab Pro L4
ROS 2	mentioned once; not used

Logistics

- **Moodle** for everything: announcements, submissions, forum
- **Midterm 30 %** (at the end of semester)
- **three assignments 24 %**
- **project 30 %**
- **five quizzes 16 %**
- **No final exam**
- **An F for the course:** not submitting a project or an assignment; missing the midterm; missing your project presentation

How the semester runs

- **Monday 8:30–10:30 · Wednesday 13:30–15:30**, each a 100-minute block of two 50-minute sessions
- **Even weeks heavy** (both days lecture) · **odd weeks light** (Monday free, Wednesday lecture)

The semester on one slide

Wk		Wk	
1–2	intro, frames, kinematics + what learning replaces	8–9	world models in the small; VLA anatomy, inference, data; midterm R 9
3–4	optimization, KOMO, control, RRT	10–11	evaluating VLAs; RL as post-training
5–6	cameras, grasps + learned perception, VLMs ; MDPs	12–13	sim2real, world models; TAMP vs LLM planners
7	imitation, diffusion policies	14	frontier; project presentations

classical component · **learned counterpart**

Assignments and project

	Window	What you build
A1 classical stack	out 2.4 → R 5	frames → forward kinematics → optimization IK → KOMO vs RRT
A2 perception → imitation	out 6.1 → wk 10	point clouds → grasps → pick-and-place → a small visuomotor policy
A3 VLA evaluation	out 10.1 → wk 12	run an open VLA on LIBERO; failure taxonomy; short fine-tune

Project in teams of (?4-5): team formation **?Fri 25 Sep**. Guidelines on Moodle today.

Break

Second half: what manipulation is, and the diagram we will fill in until December.

What is manipulation?

Definition 1 (etymological). *Manipulation refers to the activities performed by hands.*

Definition 5. *Manipulation refers to an agent's control of its environment through selective contact.*

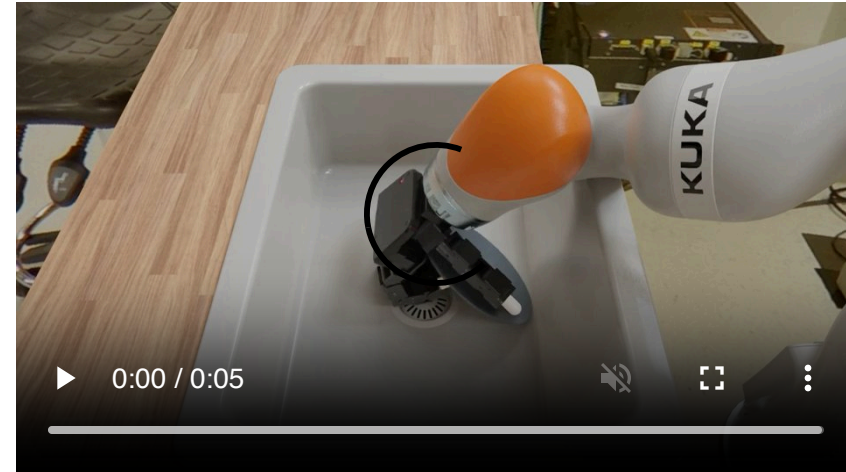
— Matthew T. Mason, *Toward Robotic Manipulation*, 2018

Why hands are hard



Ian's Secure Shoelace Knot tutorial - Professor Shoelace :)

Play: *Ian's Secure Shoelace Knot* · Professor Shoelace
(1m50s)



a hand lifting a plate out of a sink ·
Toyota Research Institute (TRI)

Manipulation without hands



Play: [an eight-stage puzzle](#) · BBC Earth (1m20s)



Play: [Stoffel the honey badger](#) · BBC (35s)

Neither has hands. Both plan a **sequence**, and both act **through contact**.

"Open-world" manipulation requires

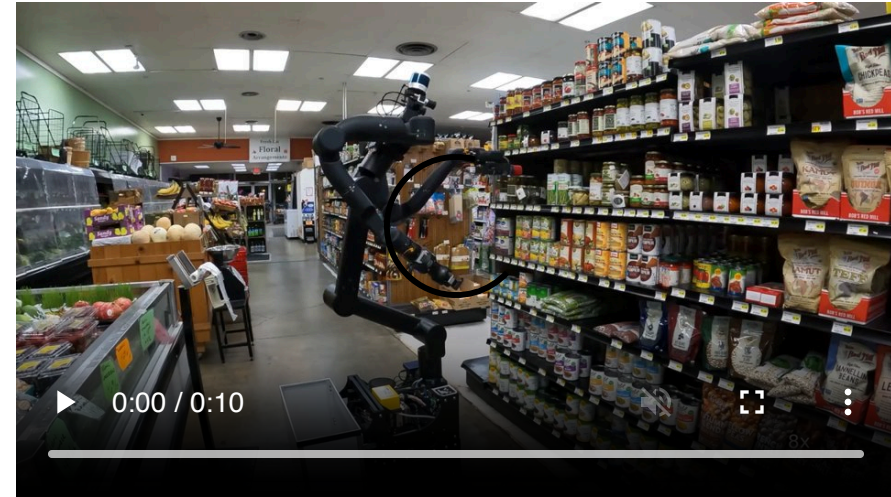
- **Rich perceptual understanding** of the environment
- **Common-sense understanding** of objects: what a mug is for, which way up it goes
- **Long-horizon plans** combined with **fine, joint-level motions**

Examples I



Moley Robotic Kitchen cooks paella

Moley robotic kitchen



a mobile manipulator working a real shop's shelves, at
8x · TRI (10s, local)

The same robot, in the same shop, when it fails



"not that easy ..."

Examples II – manipulation (by hand) types

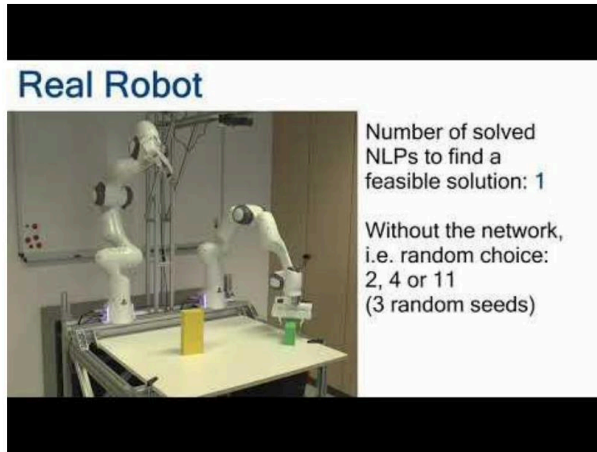


Manipulation

http://www.shap.ecs.soton.ac.uk/about-usage.php?task=obi_spherical_metal
https://www.physio-pedia.com/Sollerman_Hand_Function_Test

Play video: *Sollerman Hand Function Test* (15s)

Examples III — learning *inside* a classical planner



Learning + TAMP

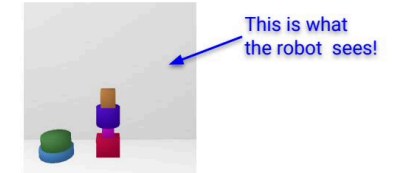
Deep Visual Heuristics: Learning Feasibility of Mixed-Integer Programs for Manipulation Planning
Danny Driess, Ozgur Oguz, Jung-Su Ha, Marc Toussaint
ICRA 2020

Deep Visual Heuristics (ICRA 2020)

Sequential Manipulation Planning from *Images*

Plan:

→ *high-level actions* to take to reach from



Self-Supervised Learning of Scene-Graph Representations for Robotic Sequential Manipulation Planning
ST Nguyen, OS Oguz, VN Hartmann, M Toussaint
Conf. on Robot Learning (CoRL), 2020.

Scene graphs for sequential manipulation (CoRL 2020)

Keep the planner, learn the heuristic.

Sequential manipulation: task-and-motion planning

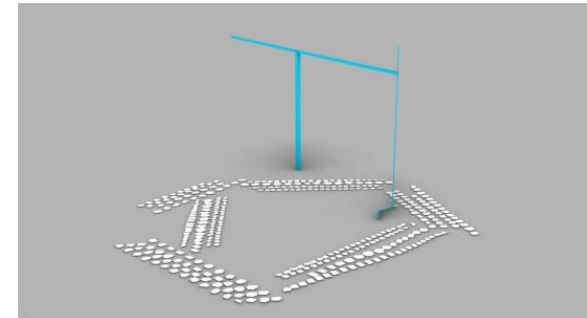
Decide the **sequence of (high-level) actions** *and* the **motion paths** that realise them — a highly interdependent process

Task planning

- discrete, symbolic
- graph search

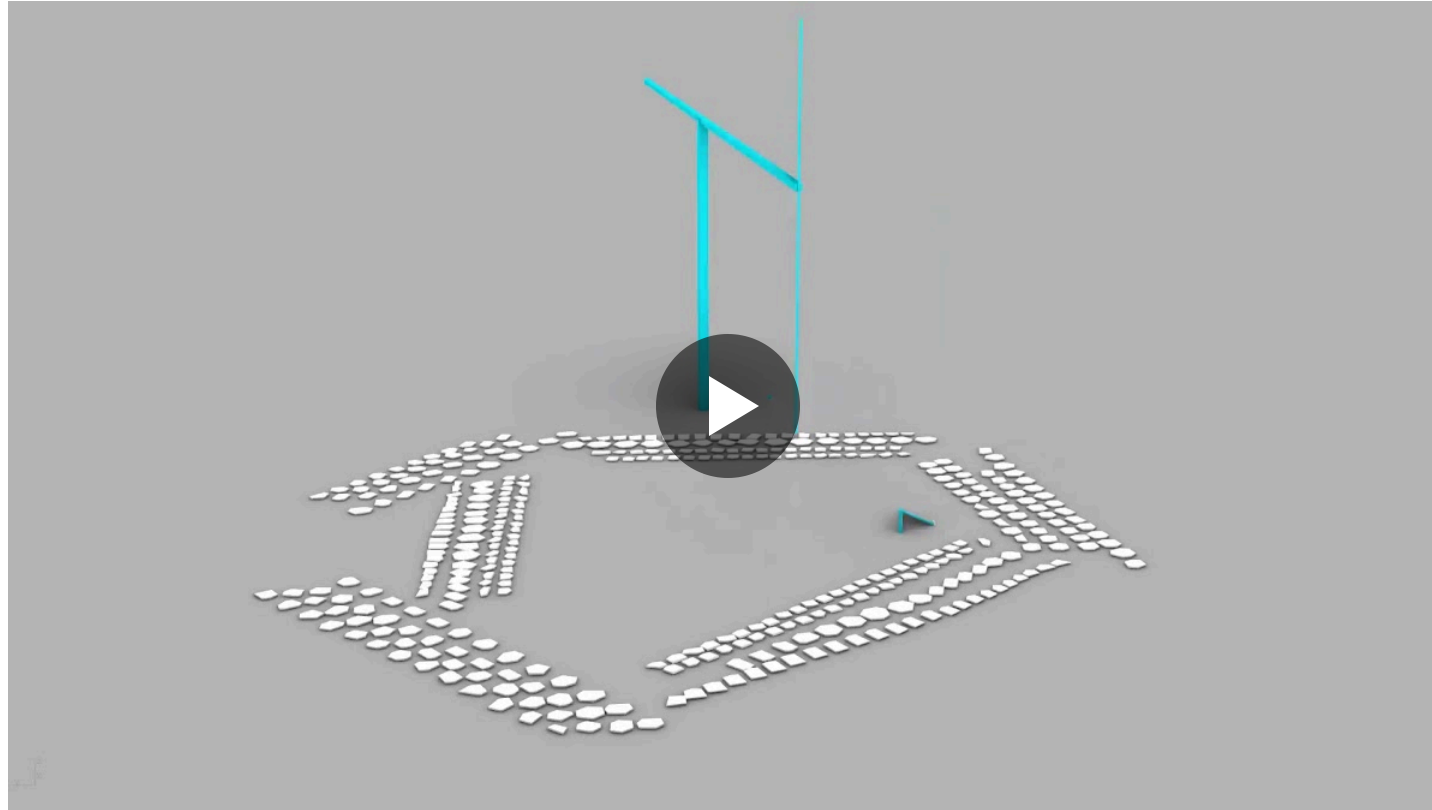
Motion planning

- continuous time, state, action
- two main lines: **sampling-based** and **optimization-based**



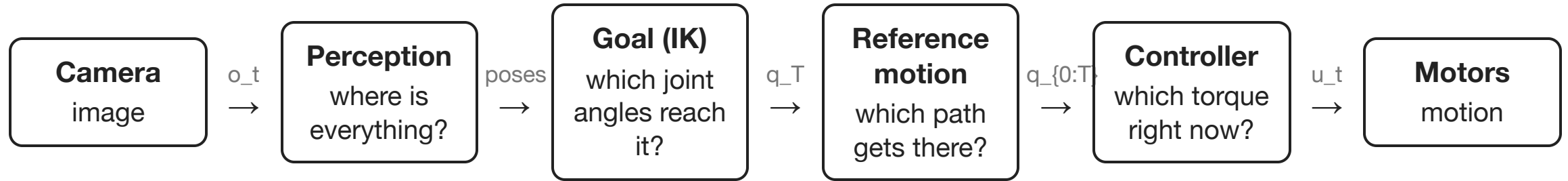
Robust TAMP for long-horizon architectural construction planning · Hartmann, Oguz, Driess, Toussaint, Menges · IROS 2020

The pavilion assembly, executed



Play: [Assembly, 60 s](#) · opens in Google Drive, in a new tab · Hartmann, Oguz, Driess, Toussaint, Menges · IROS 2020

The stack diagram, version 0



Every box is empty today. We fill in some of them.

The same diagram as a drive to a new address

Box	Driving	What it needs: the map
Perception	looking at signs and cars	camera model, object shapes
Goal	choosing the parking spot	the arm's kinematic structure
Reference motion	planning the route	shapes of everything, for collisions
Controller	steering and pedals	masses, friction, motor limits
Learned policy	the taxi driver who never opens a map	training data that resembles today

The stack coloured by week

Box	Classical	Learned counterpart
Perception	5.2, 6.1	6.2 learned grasps · 6.3 VLM perception
Goal (IK)	2.2–4.2	5.1 learned samplers · 7.1 a policy
Reference motion	4.2 KOMO · 4.4 RRT	7.2 generative policies
Controller	4.3	8.1 learned model + MPC · 8.3–9.1 VLA action chunks · 10.4–11.2 RL
Whole stack	13.1 TAMP	8.3–8.4 VLAs · 13.2 LLM planners

The methods table, revisited

Perception

- data-driven: ML, DL
- object detection, classification, tracking
- pattern recognition
- function approximation

Action

- graph and tree search
- Bayesian approaches
- linear and mixed-integer programs

Motion

- sampling-based (PRMs, RRTs)
- constraint optimization

Feedback

- control theory: robust, adaptive, optimal, model-predictive
- reinforcement learning

Representations

Decompositions

Learning

Multi-modal generative models (VLAs) ??

Takeaways and expectations

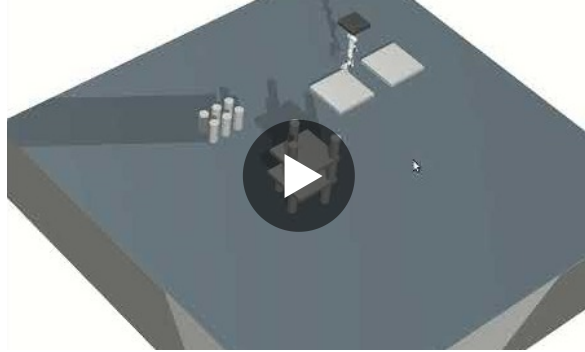
Content

- breadth over depth, without skipping fundamentals
- insight over information dump

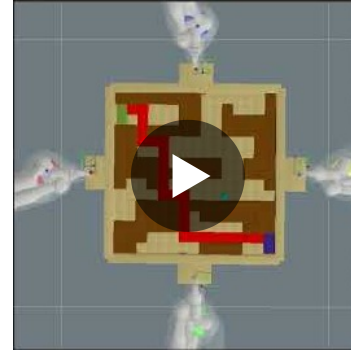
You

- expect half the course to be new to you
- go deep on one topic in the project

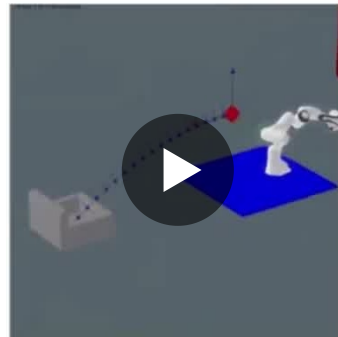
What previous students built



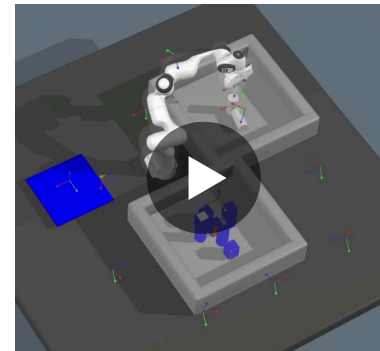
[a full BotOp build](#) · YouTube (6m36s)



[a tilt maze, four arms, SARSA](#) · YouTube (2m47s)



[throwing around obstacles](#) · Group 4 · Drive (27s)



[sorting into two trays](#) · Group 3 · Drive (25s)

All four were built in the simulator you install this week.

The next two decks, and before R 3

- **Deck 02 (2.1–2.2):** which boxes learning replaces; frames and notation
- **Deck 03 (2.3–2.4):** transforms, forward kinematics, the Jacobian; **A1 released**
- **Before R 3:** RAI installed, `hello_rai.py` running;

```
pip install robotic numpy && python hello_rai.py      # from Moodle
```

Team formation for the project: DAY, September XX.