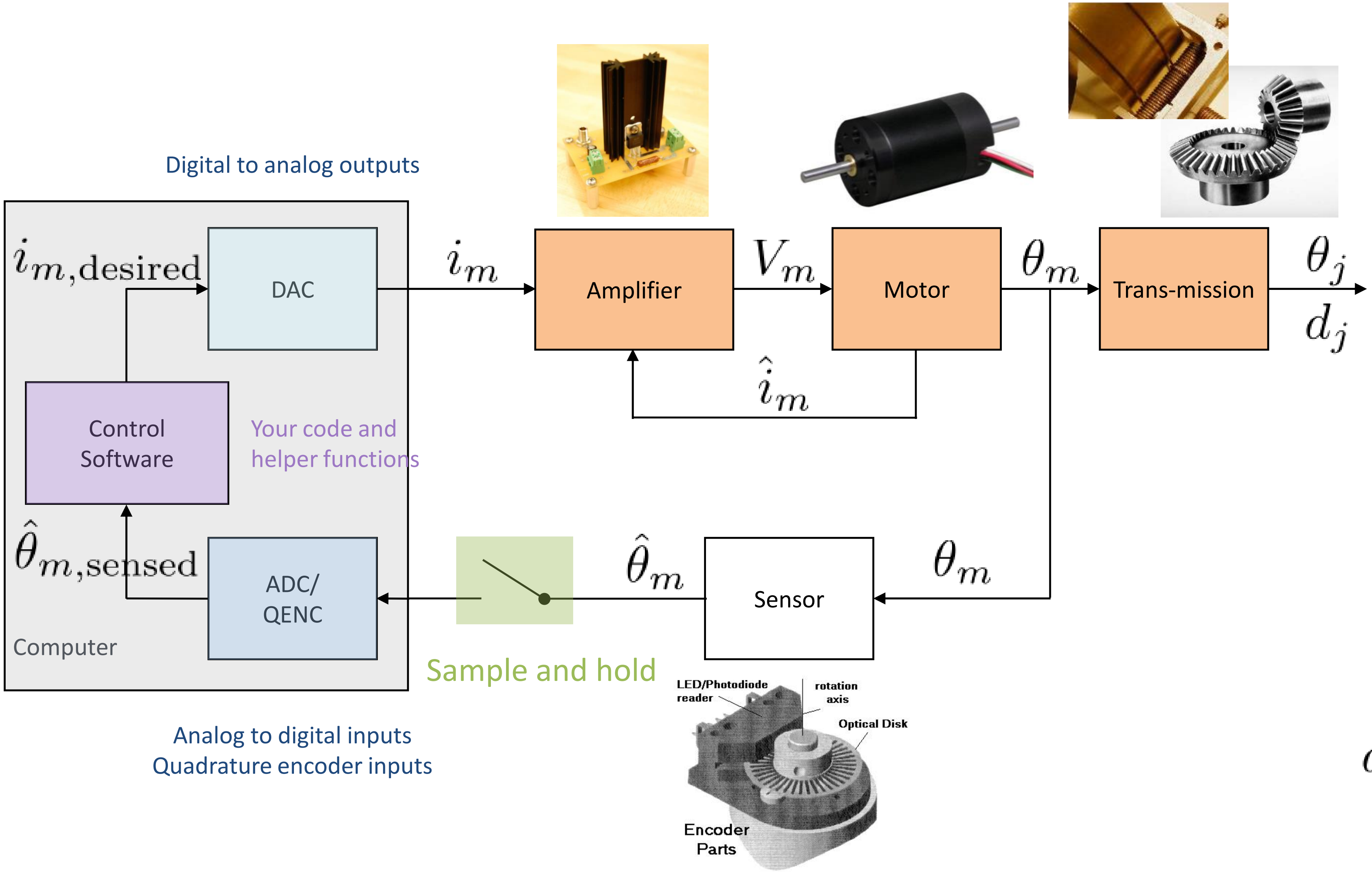
MEAM 520
Lecture 24: Paper Reading

Cynthia Sung, Ph.D.
Mechanical Engineering & Applied Mechanics
University of Pennsylvania

Previously: Actuation



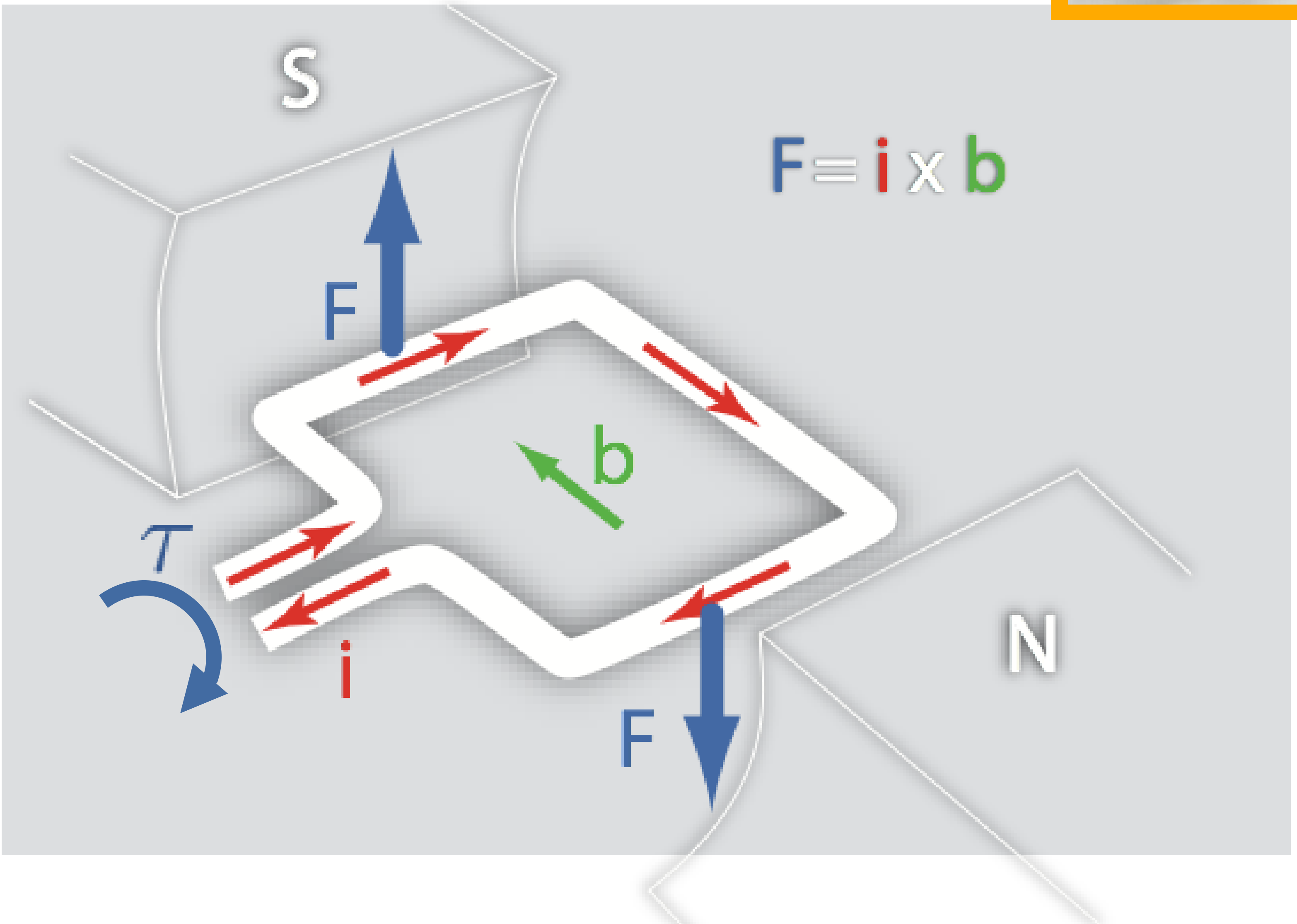
i : current
V : voltage
m : motor
j : joint
θ : angle
d : displacement
$\hat{}$: estimate

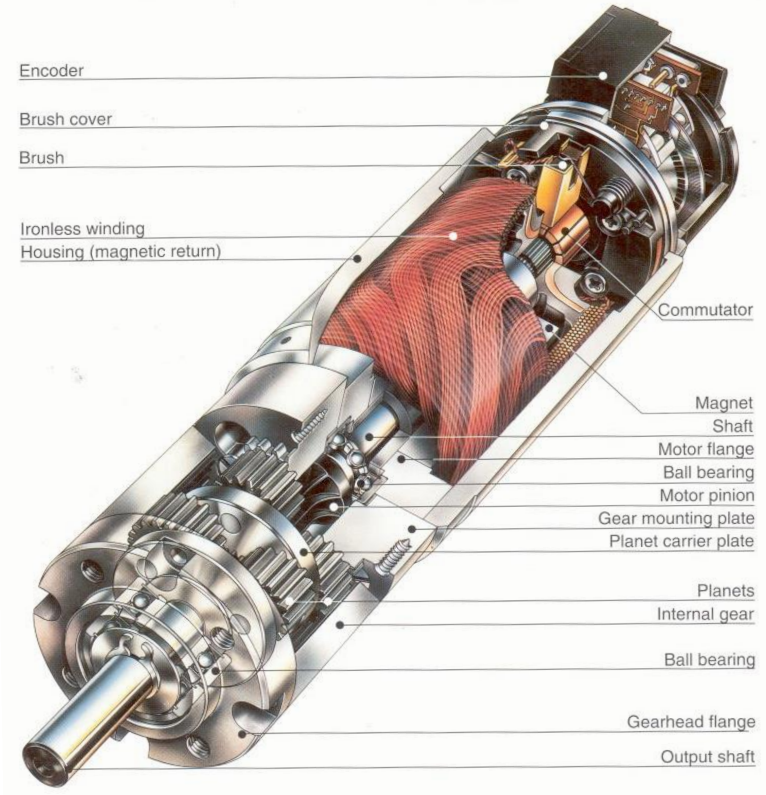
Previously: DC Brushed Motors

DC Brushed

Most common!

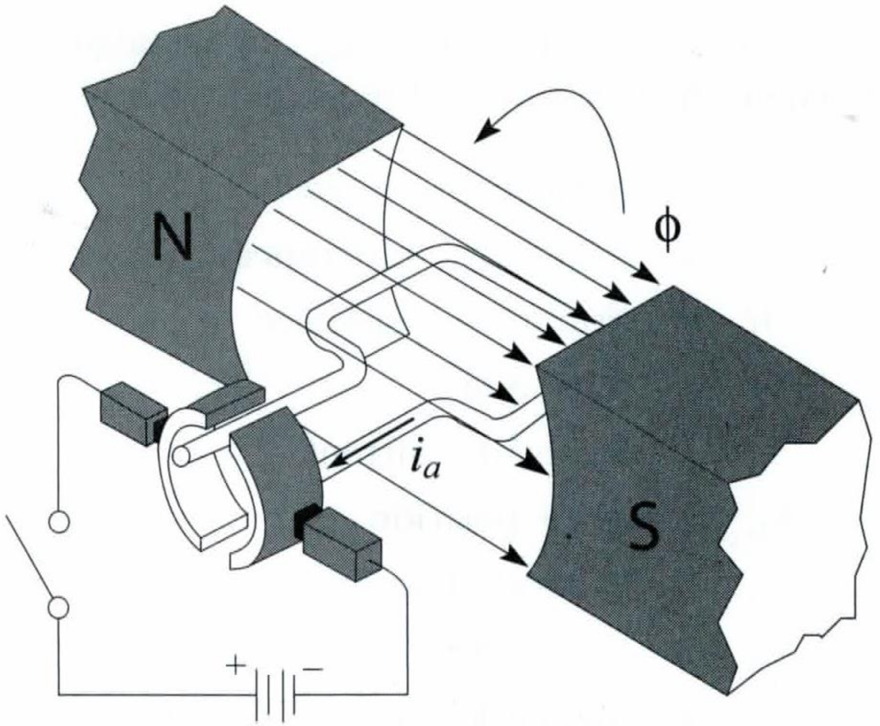
Coil Rotor
Magnetic Stator
Brushes carry current to the rotor

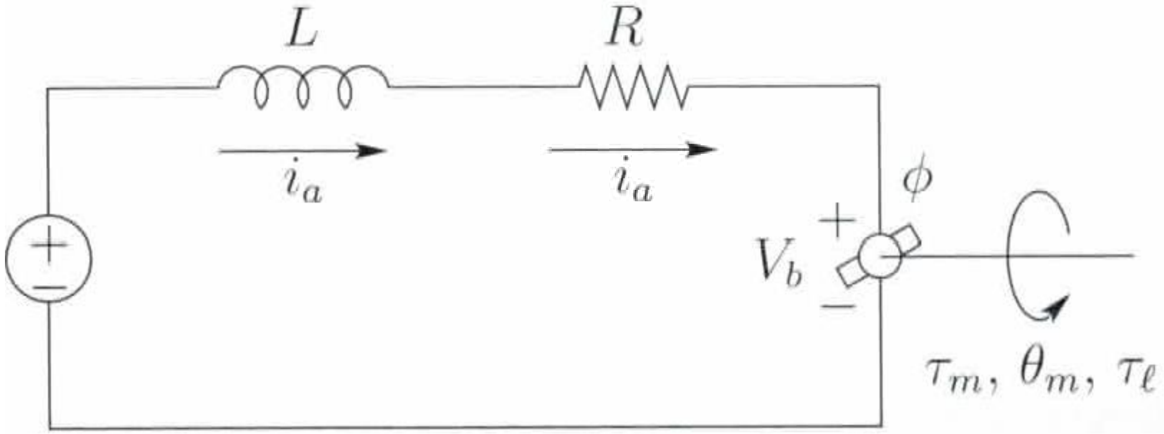




(SHV Section 6.1)

DC Motor





magnetic flux (webers)

torque constant (N•m/A)

$$\tau_m = K_1\,\phi\,i_a = k_t\,i_a$$

generated torque (N•m)

physical constant

armature current (A)

armature current (A)

$$k_t = k_v$$

if using meters, kilograms and seconds

back emf (V)

magnetic flux (webers)

back-emf constant (V•s)

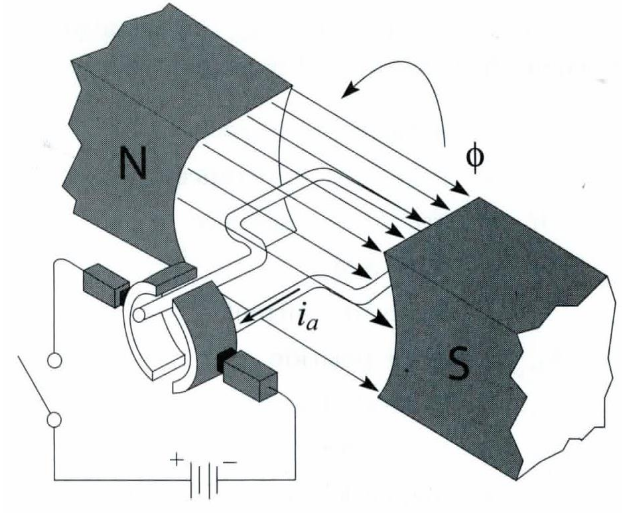
$$V_b = K_2\,\phi\,\omega_m = k_v\,\omega_m$$

physical constant

motor velocity (rad/s)

motor velocity (rad/s)

DC Motor



Electrical Dynamics

$$V(t) = L\frac{di_a}{dt} + Ri_a + k_v\frac{d\theta_m}{dt}$$

Physical Dynamics

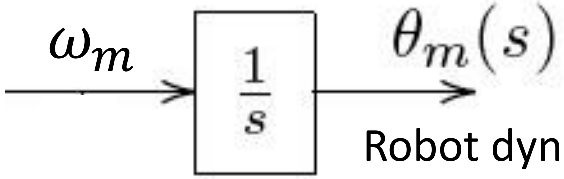
SHV shows the load torque in the wrong direction and confusingly calls gear ratio "r"

$$J_m\frac{d^2\theta_m}{dt^2} + B_m\frac{d\theta_m}{dt} = \tau_m + \tau_{ext} = k_t i_a + \tau_{ext}$$

external disturbances from connections

input

torque constant



Robot dynamics

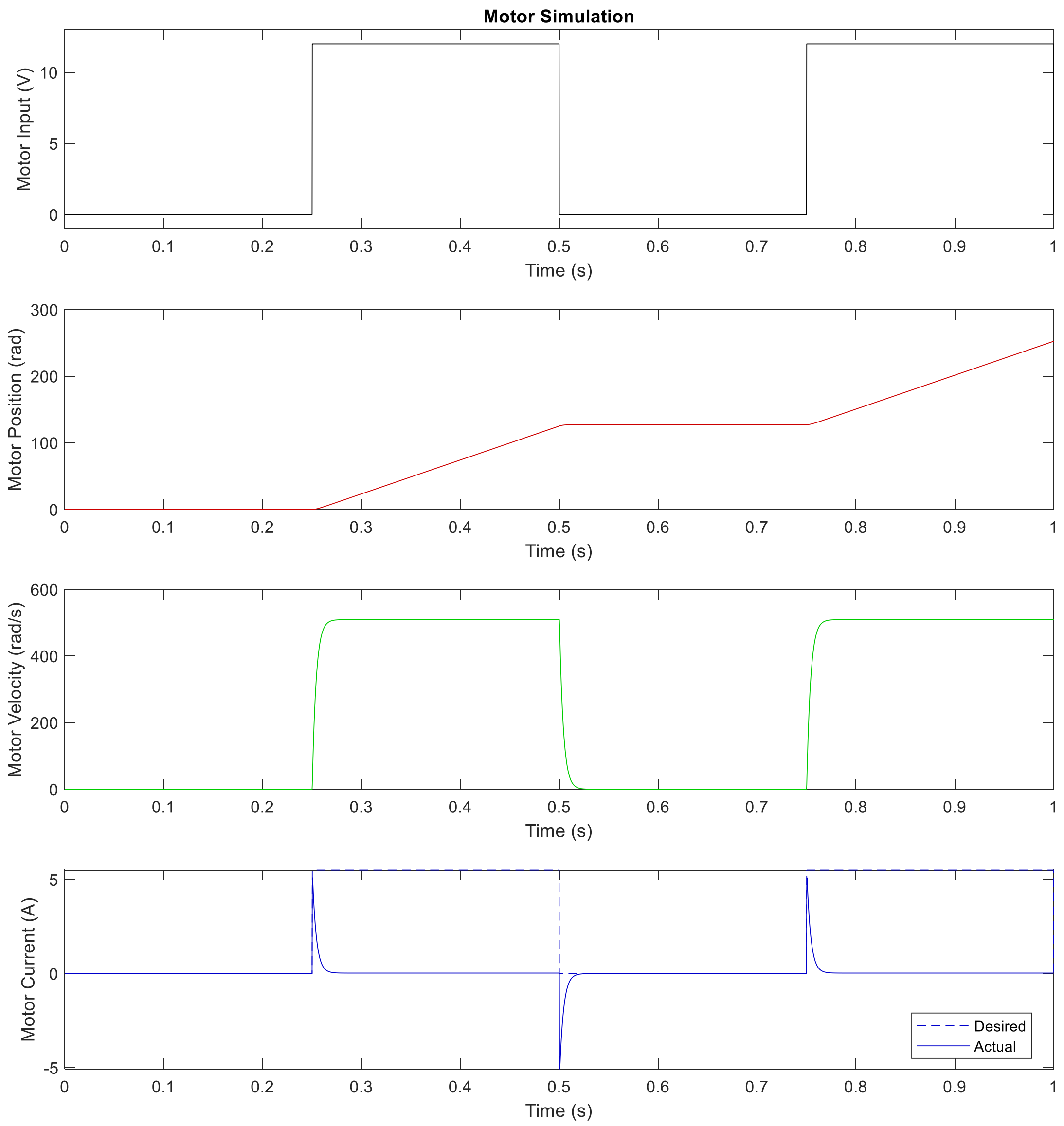
electrical dynamics

motor torque

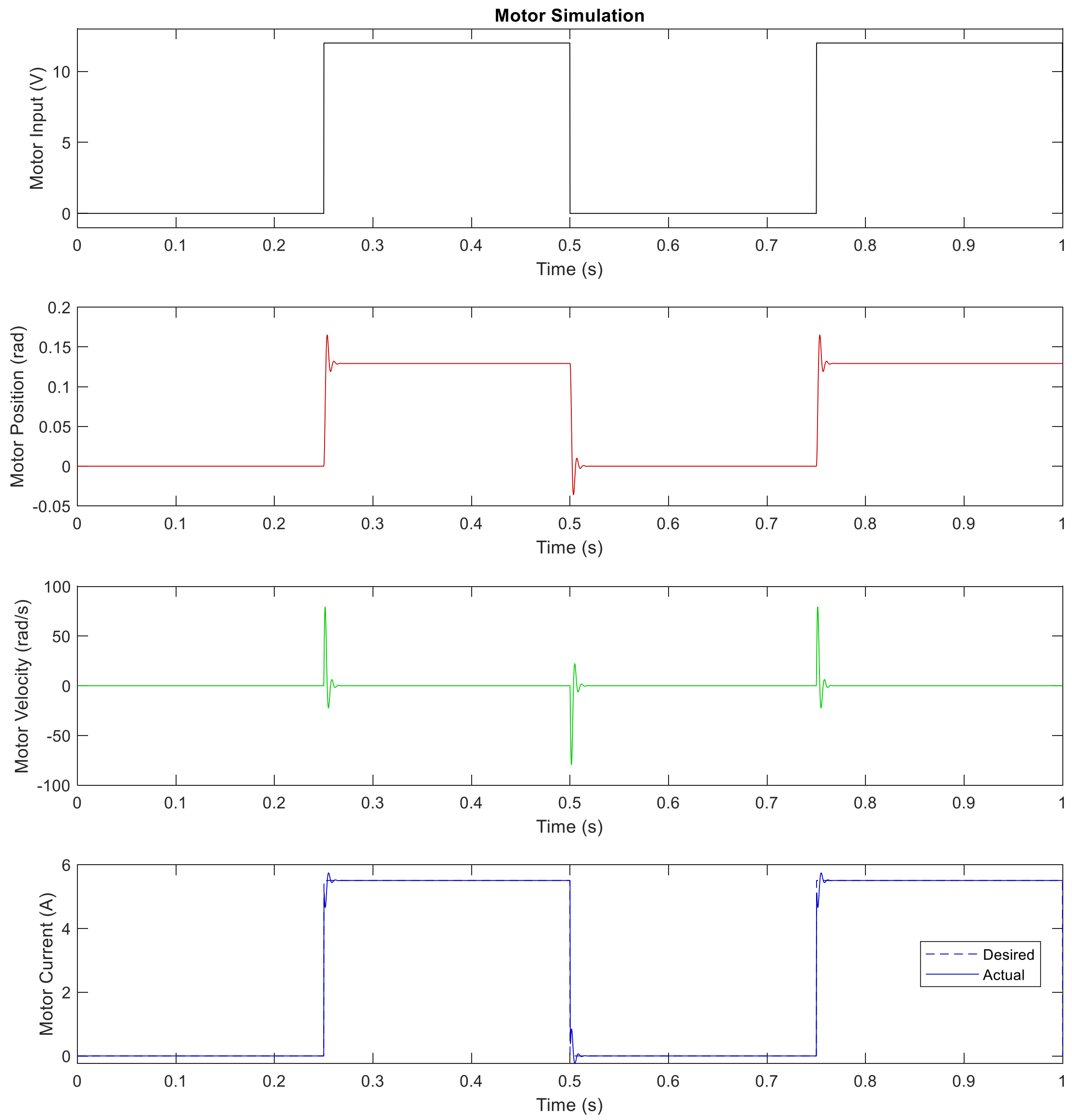
motor physics

back emf constant

No Load



Stall

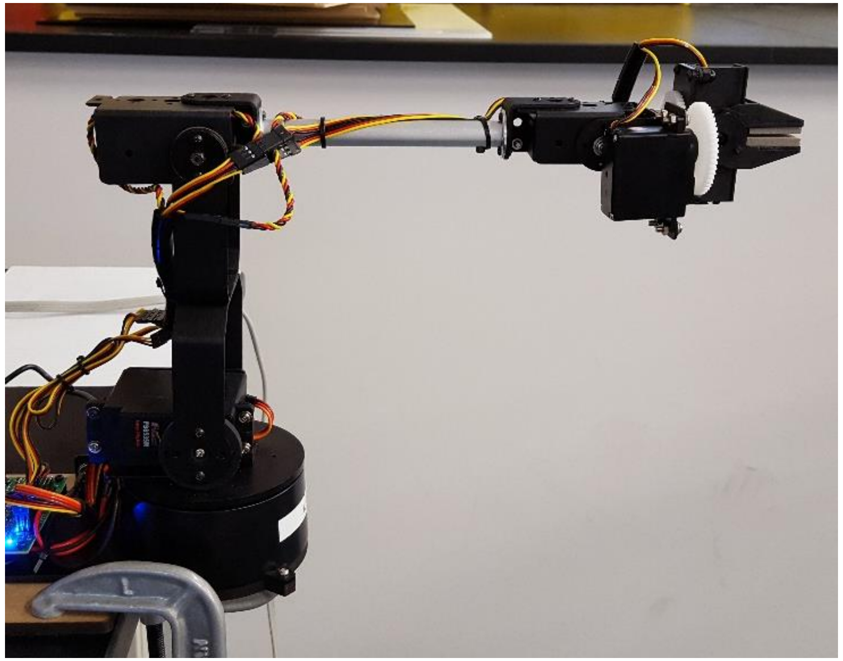


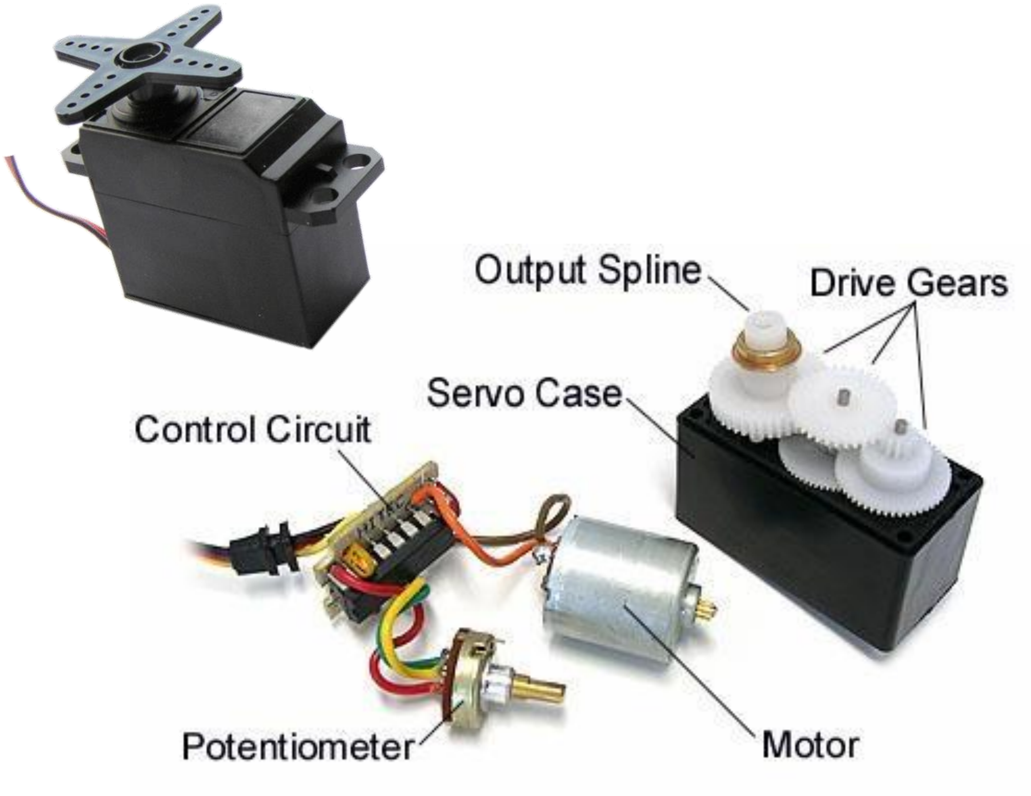
DC Motors

The best brushed DC motors are made by Maxon. They are rather expensive, but they work quite well.

- **Smooth torque output**, independent of motor angle. In other words, very low cogging and torque ripple.
- **Low friction**, both at low and high speeds, due to high quality bearings and low eddy currents.
- **Relatively high stall torque**, which is the torque the motor can deliver when it is not rotating.
- **Larger motors** create higher torques, but they also have higher inertia, higher friction, and higher cost.

Position Control: Servomotors





DC brushed motor with integrated sensor and controller

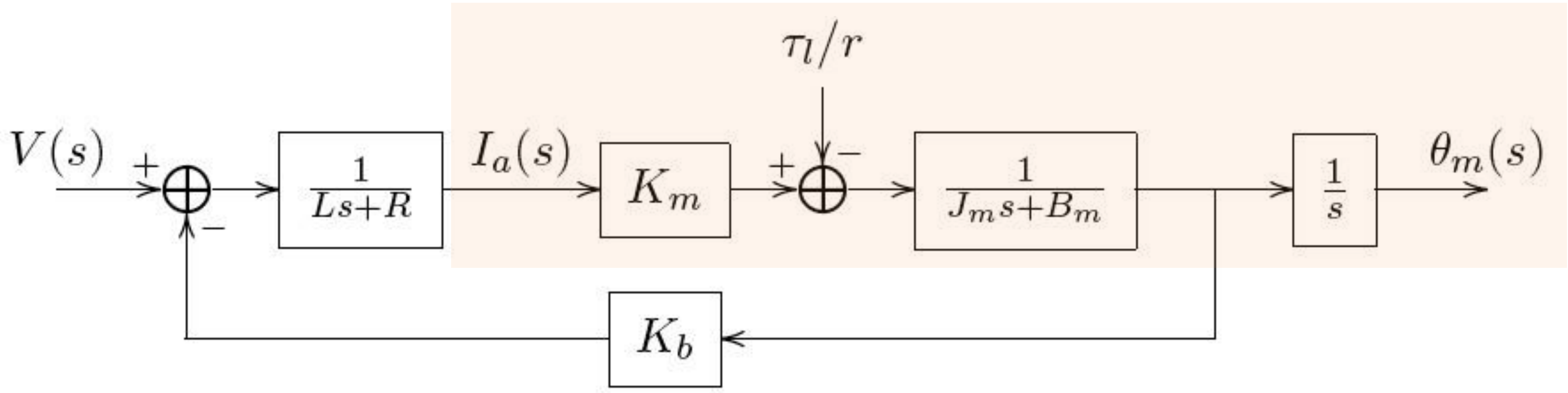
Typically, roboticists treat each joint independently, as a single-input/single-output (SISO) model.

This is adequate for applications that don't involve very fast motions, which helps decouple the links from one another.

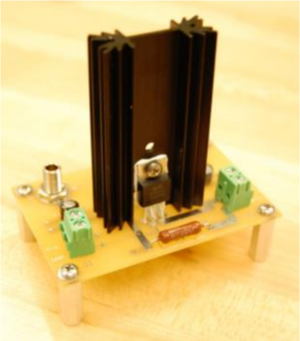
Torque Control: Current Amplifiers

$$\tau_m = k_t \, i_a$$

Motor torque is proportional to current, so if we can control current regardless of speed, we can ignore the motor's electrical dynamics (L, R, V_b).



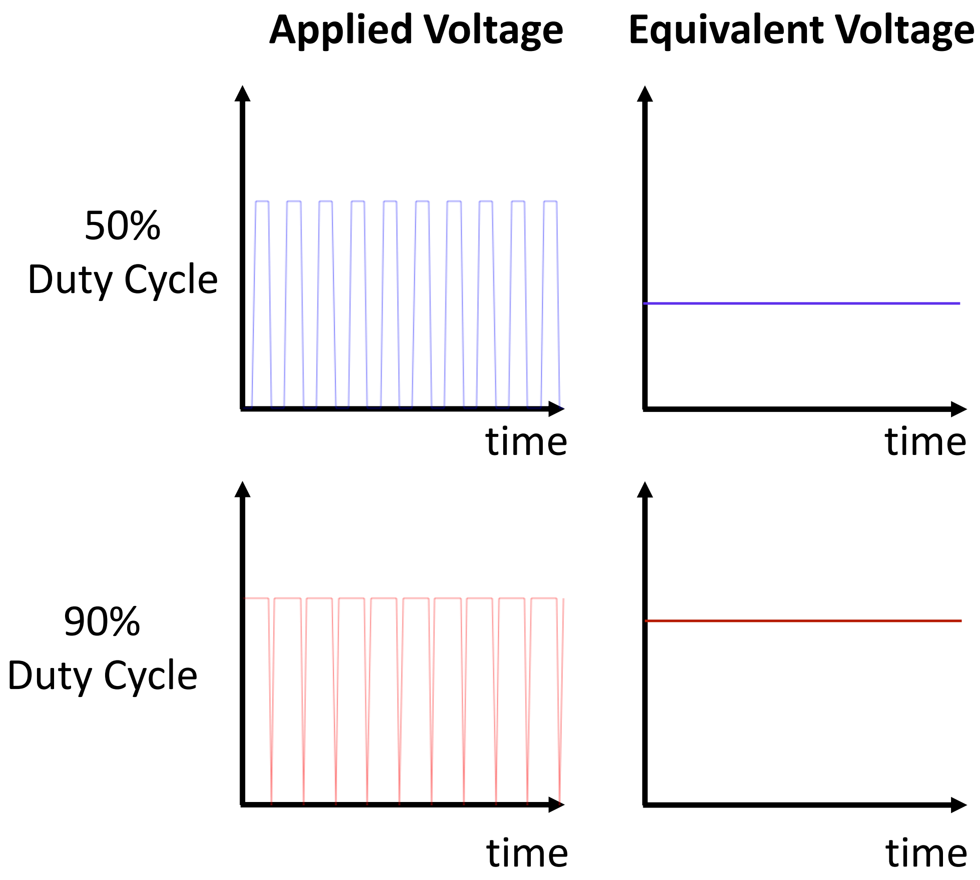
Torque Control: Current Amplifiers



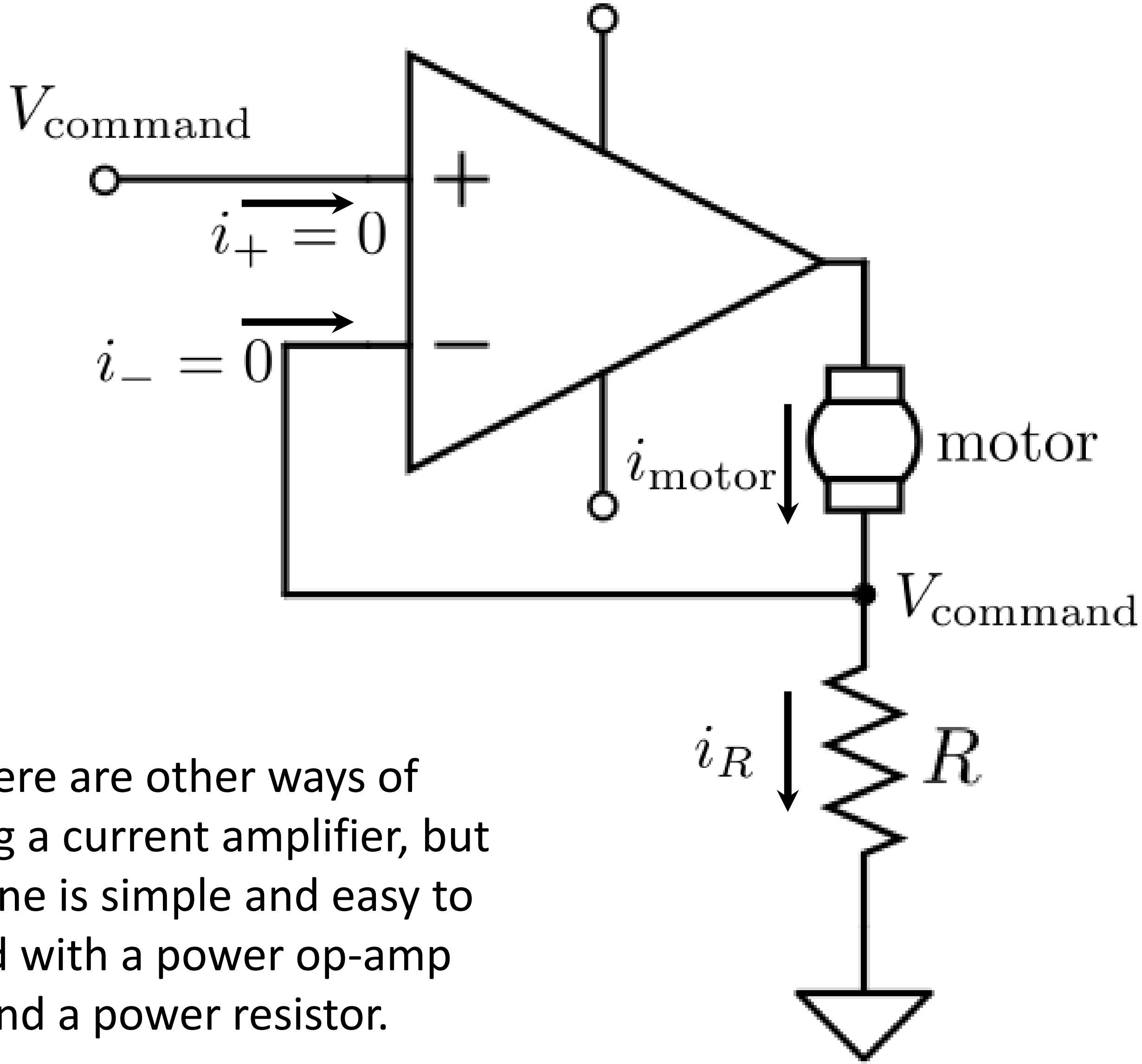
$$\tau_m = k_t \, i_a$$

Takes an information signal (usually an analog voltage) from the computer and drives the requested amount of current through the actuator.

Two common types are Pulse Width Modulation (PWM) and Linear. Linear amplifiers often have higher bandwidth and cause less electrical noise.



Torque Control: Current Amplifier Circuit



Operational amplifier in negative feedback, so it follows the two golden rules of op-amps:

No current enters or leaves the inverting or non-inverting inputs.

The voltages at the inverting and non-inverting inputs are equal.

KCL: $i_{\text{motor}} = i_R$

Ohm's Law: $i_R = \dfrac{V_{\text{command}}}{R}$

$$\boxed{i_{\text{motor}} = \frac{V_{\text{command}}}{R}}$$

There are other ways of making a current amplifier, but this one is simple and easy to build with a power op-amp and a power resistor.

Final Project Proposals all have feedback

- I have matched each group with a TA who has a strong background in your topic of interest
- Do reach out as you are working on your projects
- Final Project Presentations: 12/4 and 12/6
  - 1 min: Problem definition
  - 1 min: Results
  - 1 min: Lessons learned
- Submit your slides the day before
- Order posted on Canvas and Piazza after class today

Today

2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Autonomous Robotic Stone Stacking with Online next Best Object Target Pose Planning

Fadri Furrer*,1, Martin Wermelinger*,2, Hironori Yoshida*
Fabio Gramazio3, Matthias Kohler3, Roland Siegwart1, Marco Hutter2

***Abstract*—Predominately, robotic construction is applied as prefabrication in structured indoor environments with standard building materials. Our work, on the other hand, focuses on utilizing irregular materials found on-site, such as rubble and rocks, for autonomous construction. We present a pipeline that detects randomly placed objects in a scene that are used by our next best stacking pose searching method employing gradient descent with a random initial orientation, exploiting a physics engine. This approach is validated in an experimental setup using a robotic manipulator by constructing balancing vertical stacks without mortars and adhesives. We show the results of eleven consecutive trials to form such towers autonomously using four arbitrarily in front of the robot placed rocks.**

I. Introduction

Over the last decade, robotics has been introduced to architectural construction not only for safer and more efficient construction, but also for exploring diverse forms [1]. However, there are still intensive manual labor works involved for on-site assembly of these components [2].

Digital fabrication has explored applications of autonomous robots in on-site operation scenarios [3], but are restricted to build with regular materials. Building structures with irregular shaped objects was presented in [4], however they apply glue to increase stability. To reduce the environmental impact we aim to use such material without additional adhesives, to build dry-stack compositions. Therefore, our work focuses on developing an automated fabrication process using irregular objects, which are not processed but found on-site.

As a case study, discrete rigid elements, such as stones or concrete rubble, are targeted as a building material. Our goal is to construct a balancing vertical tower with found objects, while maintaining the structure in static equilibrium using a robotic manipulator. To achieve this, we developed a holistic work-flow including precise object detection, motion control, and planning the next target pose (see Figure 1). As part of this work-flow, we describe an algorithm suggesting stable poses for stacking, validated by an implementation of this autonomous stacking work-flow in a real-world experiment. Due to the instability of vertical tower, it is natural to observe errors between a desired target pose and an actual stacked pose. Thus, the work-flow puts emphasis on the resultant pose evaluation and a dynamic re-planning of the target pose.

Fig. 1: In an offline step we scan a set of objects (top). These objects, or a subset of it, can be distributed arbitrarily on the work-space and get detected by our object detection pipeline (middle-left). From the detected objects the presented pose searching algorithm proposes the next stable stack (middle-right). A motion planner (bottom-right) is used to generate the trajectories to replicate the proposed stack with the robot arm (bottom-left). After placing the object, its pose is measured and used as base for the subsequent pose searching step.

This paper makes the following contributions regarding handling irregularly shaped objects:

– a pose searching algorithm considering structural stability using a physics engine
– an object detection pipeline
– an autonomous system for constructing balancing vertical towers using a manipulator

II. Related Work

Research in architecture and digital fabrication investigates novel production techniques in which material behavior is linked to fabrication and assembly tasks [5]. Recently, it has

[1]Autonomous Systems Lab (ASL), ETH Zurich, Switzerland
[2]Robotic Systems Lab (RSL), ETH Zurich, Switzerland
[3]Gramazio Kohler Research (GKR), ETH Zurich, Switzerland

*The authors contributed equally to this work. F.F. was responsible for the object detection, M.W. for the manipulation tasks, H.Y. for the pose searching algorithm.

This work was supported in part by the Swiss National Science Foundation (SNF), the National Centre of Competence in Research Digital Fabrication.

978-1-5090-4633-1/17/$31.00 ©2017 IEEE

Furrer, F. et al. “Autonomous Robotic Stone Stacking with Online next Best Object Target Pose Planning.” ICRA 2017. Best student paper award.

https://www.youtube.com/watch?v=bXz52KMGUng

Nils Napp, https://www.youtube.com/watch?v=5F13XSHRxEU

Nils Napp, https://www.youtube.com/watch?v=6ijnXfUtYK0

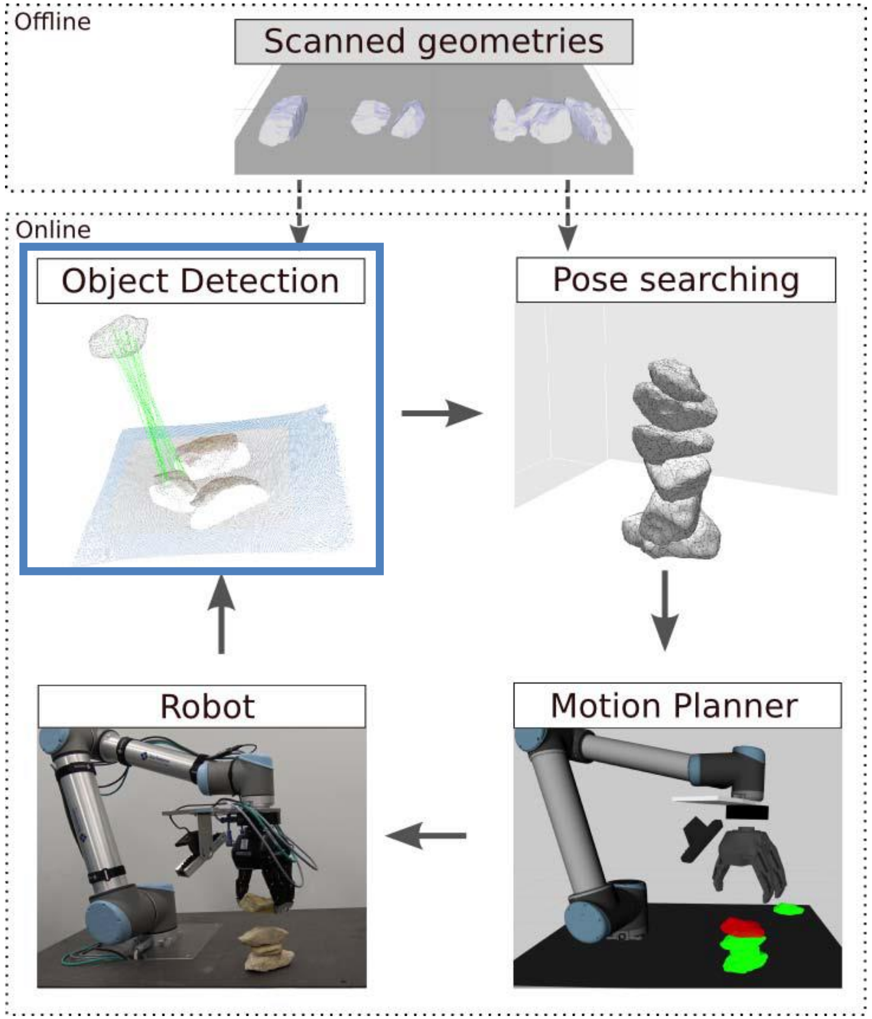


Fig. 1

Transformation matrices

$$\mathbf{P}_{\mathcal{R},\text{object}} = T_{\mathcal{RT}} \cdot T_{\mathcal{TC}} \cdot T_{\mathcal{CO}} \cdot \mathbf{P}_{\mathcal{O},\text{object}}$$

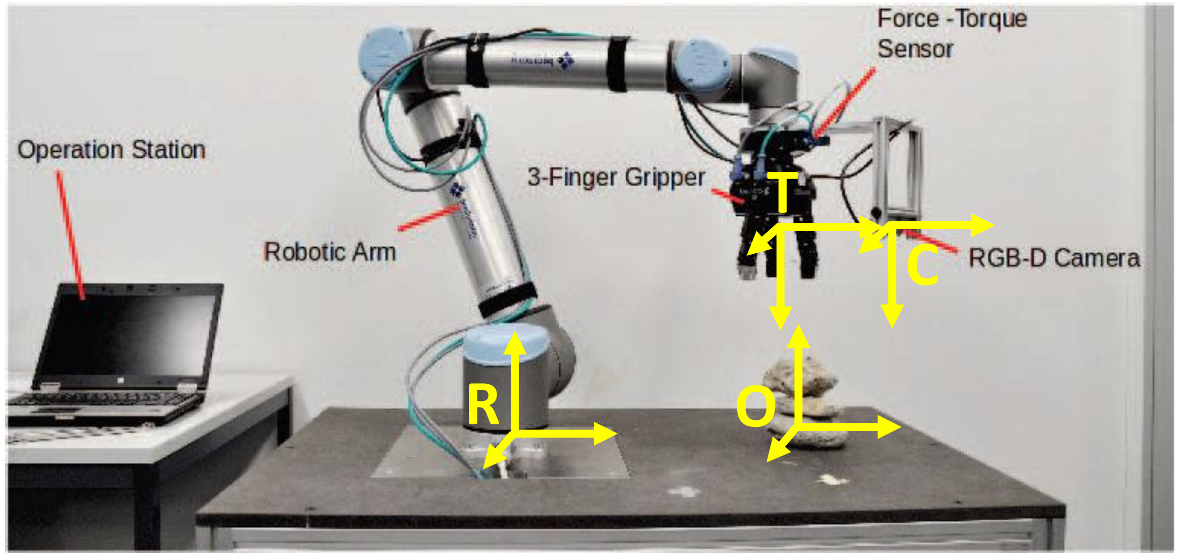
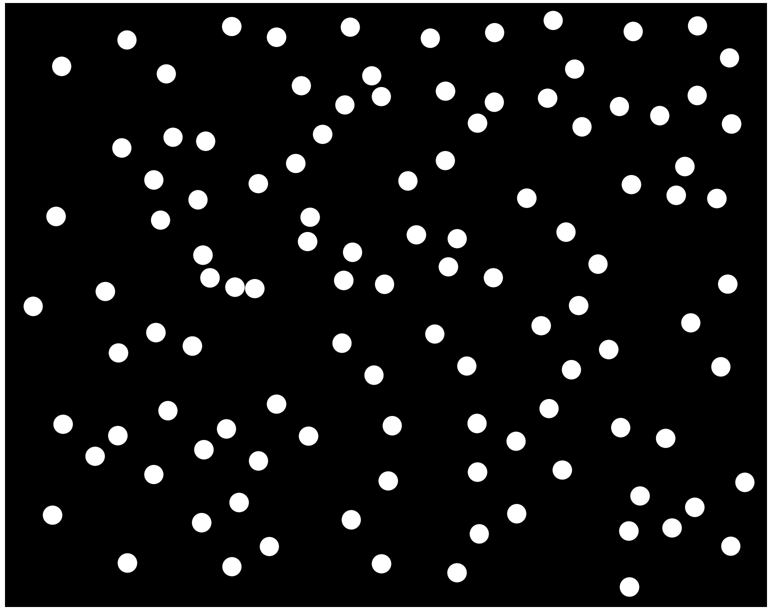
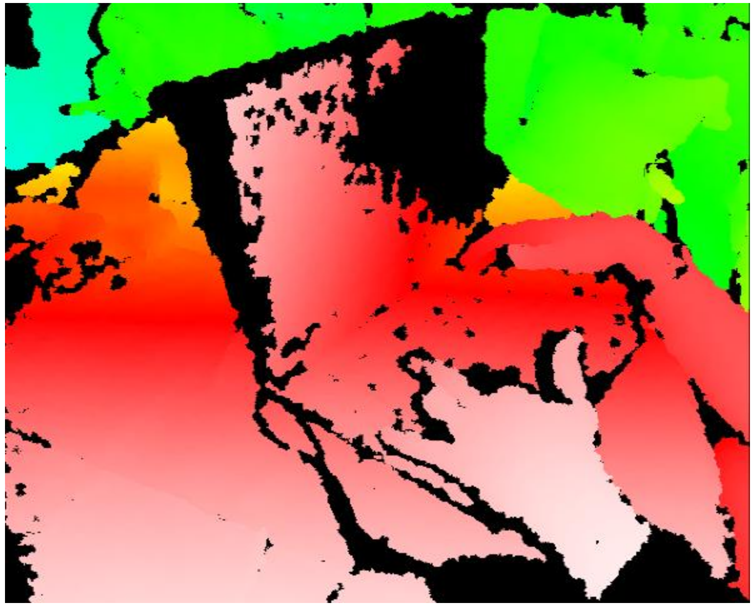


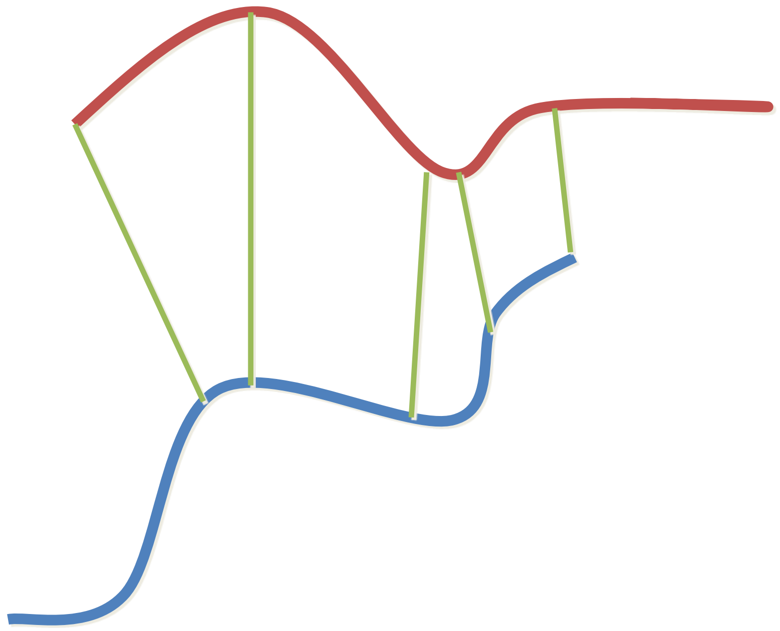
Fig. 6

Depth Camera





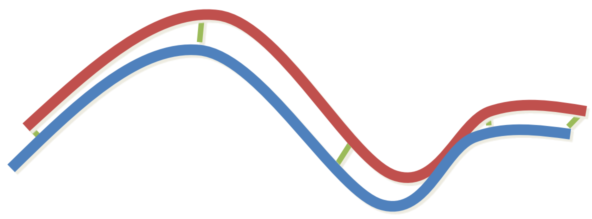
Iterative Closest Point (ICP)



Input: 2 data sets

Match each point in set 1 to
the closest point in set 2

Iterative Closest Point (ICP)



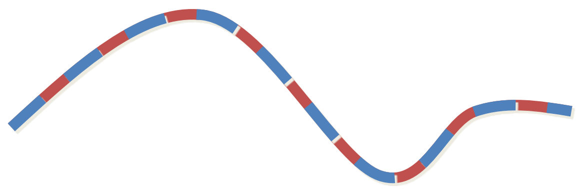
Input: 2 data sets

Repeat:

Match each point in set 1 to
the closest point in set 2

Compute the transformation
that minimizes error

Iterative Closest Point (ICP)



Input: 2 data sets

Repeat:

Match each point in set 1 to the closest point in set 2

Compute the transformation that minimizes error

Iterative Closest Point (ICP)

Sparse ICP

Sofien Bouaziz   Andrea Tagliasacchi   Mark Pauly

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Mark Pauly, https://www.youtube.com/watch?v=ii2vHBwlmo8

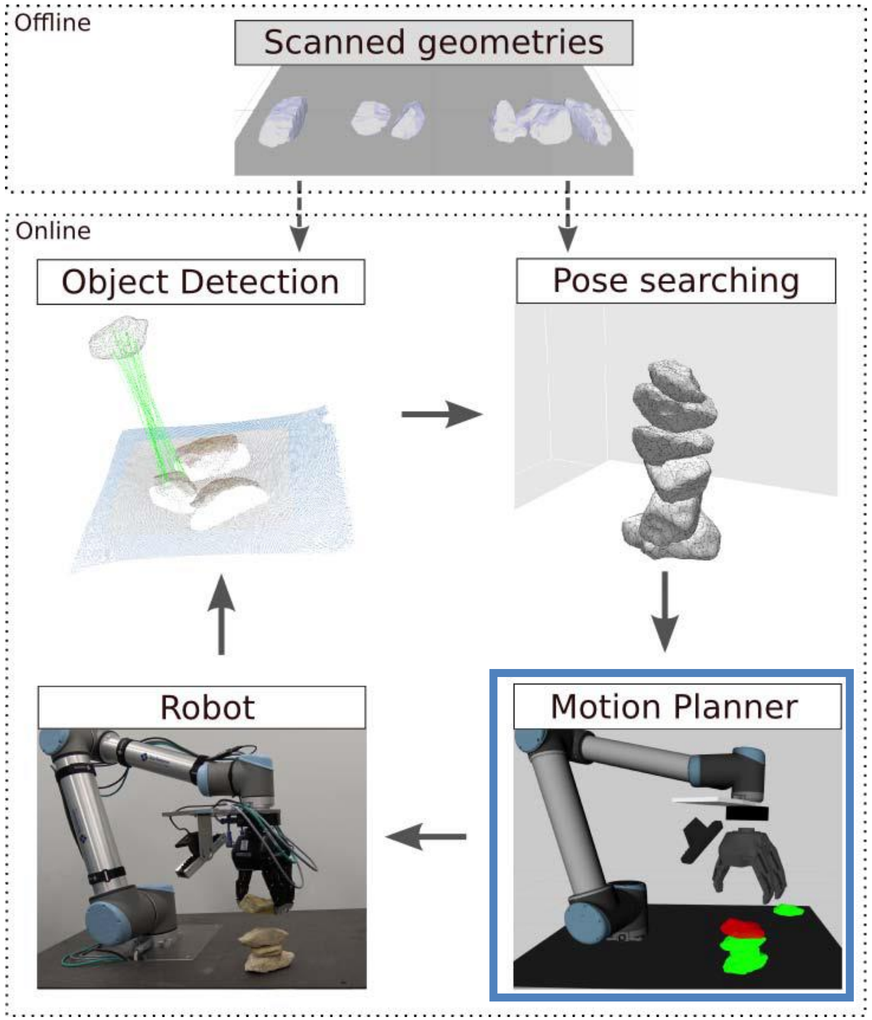

Offline
Scanned geometries
Online
Object Detection
Pose searching
Robot
Motion Planner

Fig. 1

Planners Available in MoveIt!

MoveIt! is designed to work with many different types of planners, which is ideal for benchmarking improved planners against previous methods. Below is a list of planners that have been used with MoveIt!, in descending order of popularity/support within MoveIt!:

Open Motion Planning Library (OMPL)

OMPL is an open-source motion planning library that primarily implements randomized motion planners. MoveIt! integrates directly with OMPL and uses the motion planners from that library as its primary/default set of planners. The planners in OMPL are abstract; i.e. OMPL has no concept of a robot. Instead, MoveIt! configures OMPL and provides the back-end for OMPL to work with problems in Robotics. Fully supported. More Info

Stochastic Trajectory Optimization for Motion Planning (STOMP)

STOMP (Stochastic Trajectory Optimization for Motion Planning) is an optimization-based motion planner based on the PI^2 (Policy Improvement with Path Integrals, Theodorou et al, 2010) algorithm. It can plan smooth trajectories for a robot arm, avoiding obstacles, and optimizing constraints. The algorithm does not require gradients, and can thus optimize arbitrary terms in the cost function like motor efforts. Partially supported. More Info

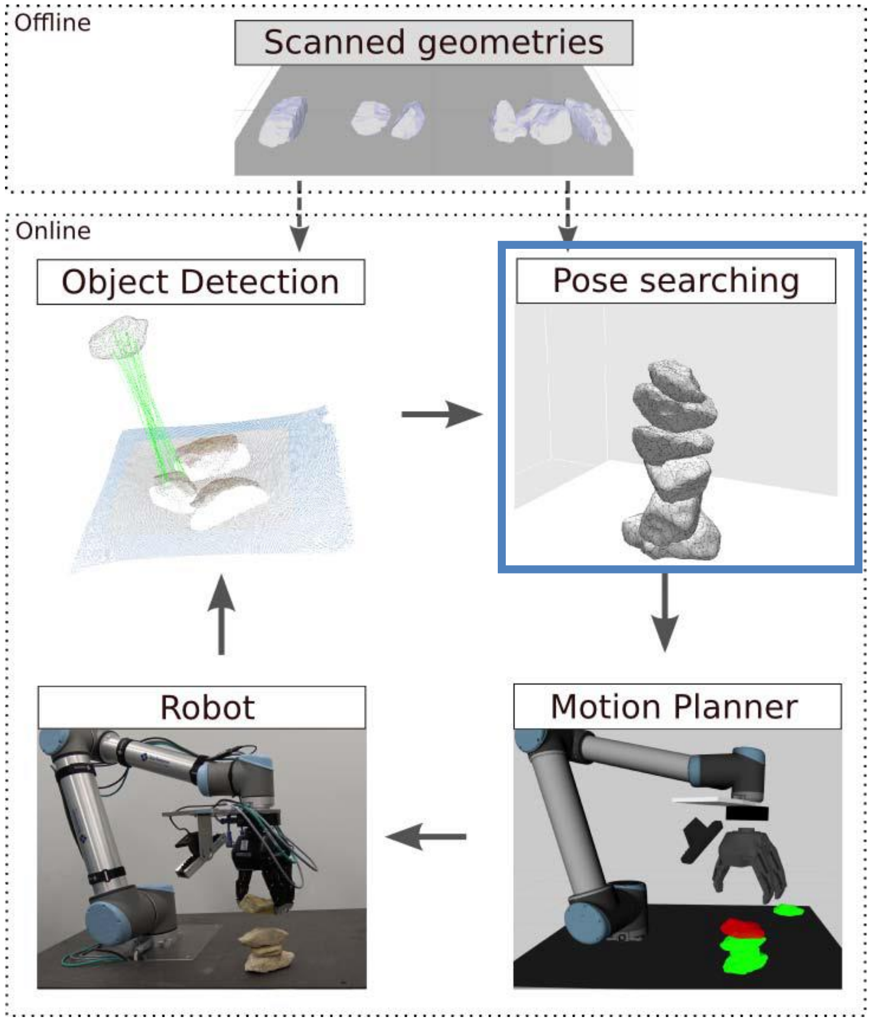
Search-Based Planning Library (SBPL)

A generic set of motion planners using search based planning that discretize the space. Integration into latest version of MoveIt! is work in progress. More Info

Covariant Hamiltonian Optimization for Motion Planning (CHOMP)

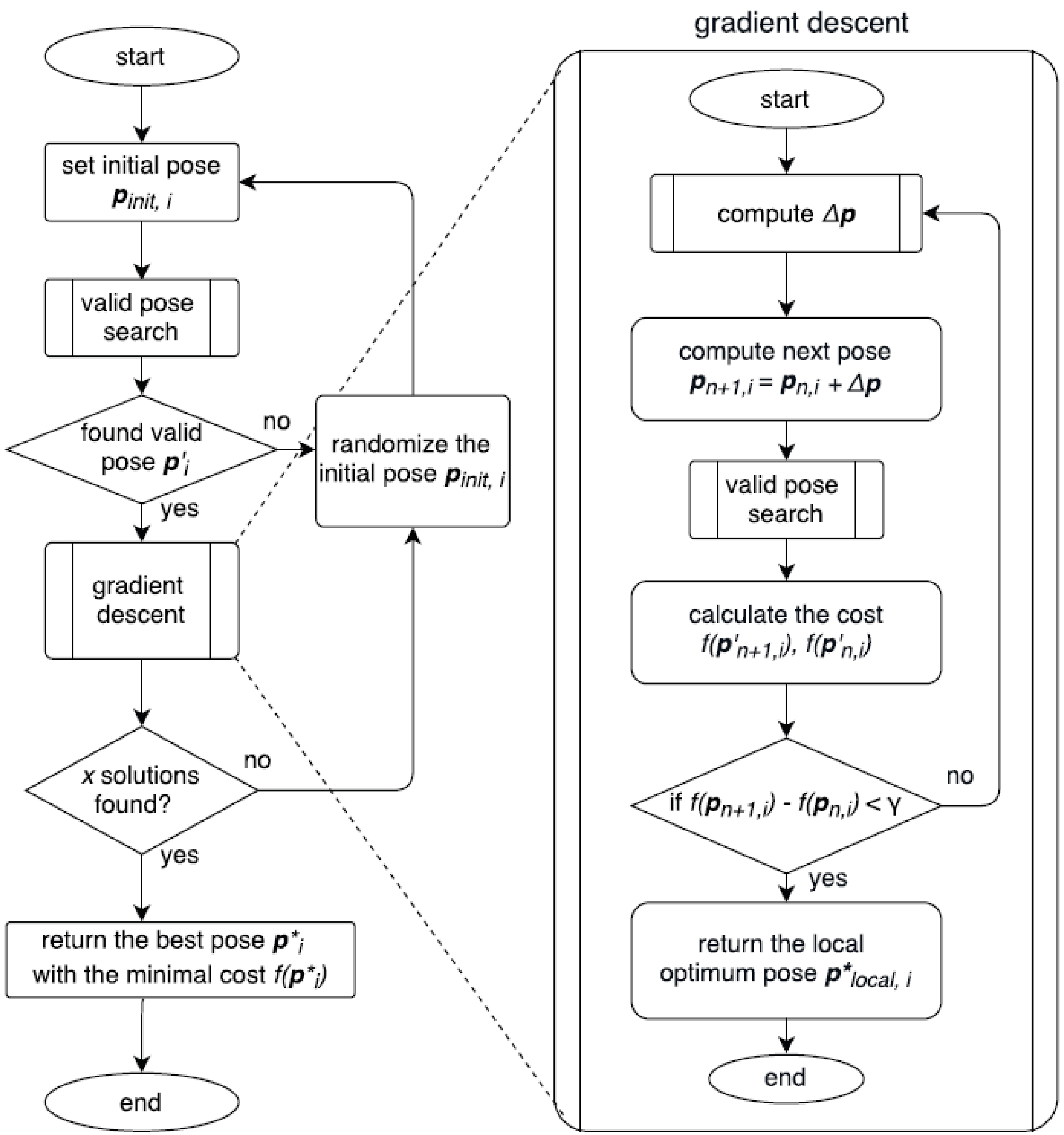
Covariant Hamiltonian optimization for motion planning (CHOMP) is a novel gradient-based trajectory optimization procedure that makes many everyday motion planning problems both simple and trainable (Ratliff et al., 2009c). While most high-dimensional motion planners separate trajectory generation into distinct planning and optimization stages, this algorithm capitalizes on covariant gradient and functional gradient approaches to the optimization stage to design a motion planning algorithm based entirely on trajectory optimization. Given an infeasible naive trajectory, CHOMP reacts to the surrounding environment to quickly pull the trajectory out of collision while simultaneously optimizing dynamical quantities such as joint velocities and accelerations. It rapidly converges to a smooth collision-free trajectory that can be executed efficiently on the robot. Integration into latest version of MoveIt! is work in progress. More Info

https://moveit.ros.org/



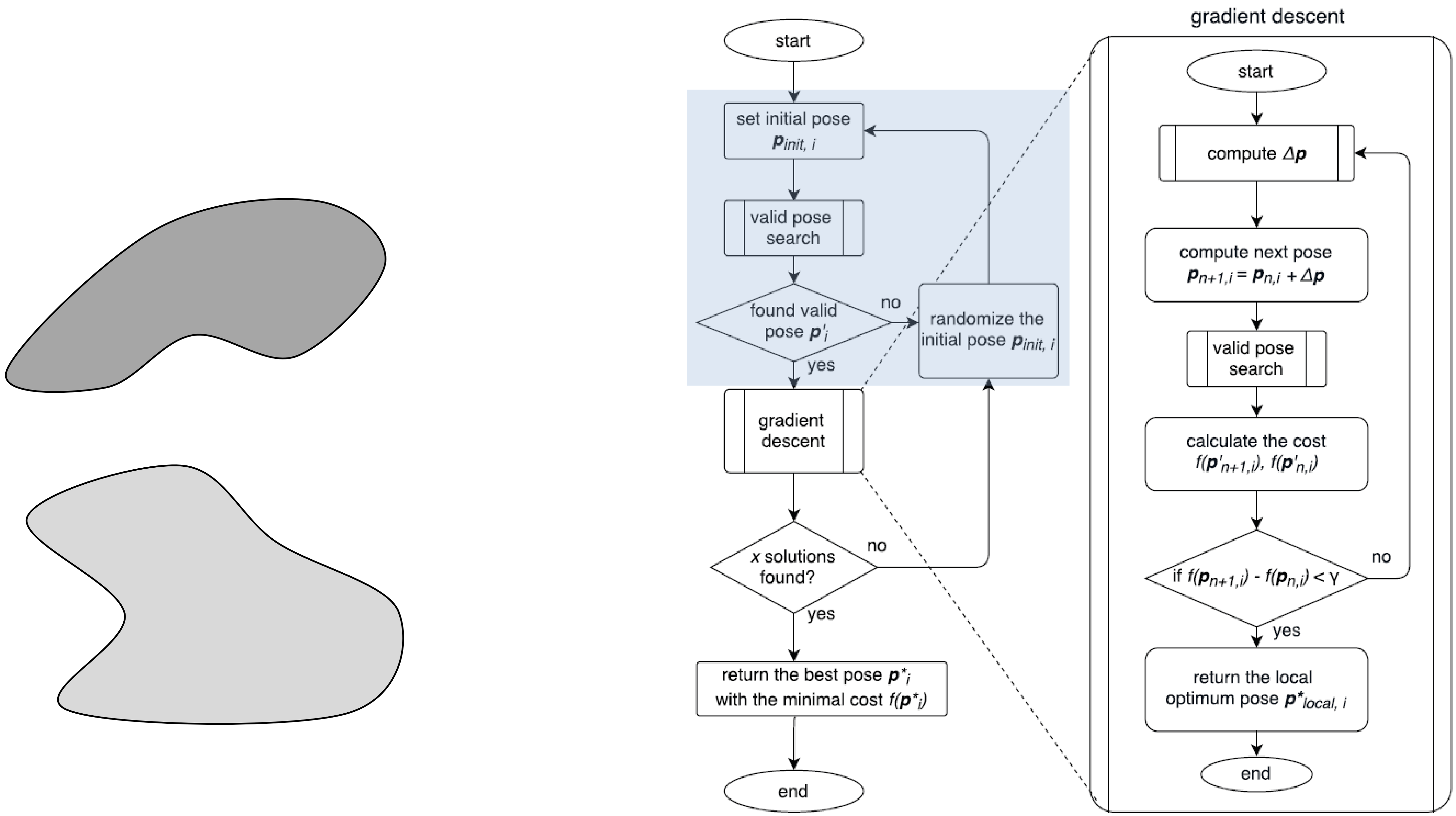
Offline
Scanned geometries
Online
Object Detection
Pose searching
Robot
Motion Planner

Fig. 1



start
set initial pose $p_{init,\, i}$
valid pose search
found valid pose p'_i
no
yes
randomize the initial pose $p_{init,\, i}$
gradient descent
x solutions found?
no
yes
return the best pose p^*_i with the minimal cost $f(p^*_i)$
end
gradient descent
start
compute Δp
compute next pose $p_{n+1,i} = p_{n,i} + \Delta p$
valid pose search
calculate the cost $f(p'_{n+1,i})$, $f(p'_{n,i})$
if $f(p_{n+1,i}) - f(p_{n,i}) < \gamma$
no
yes
return the local optimum pose $p^*_{local,\, i}$
end

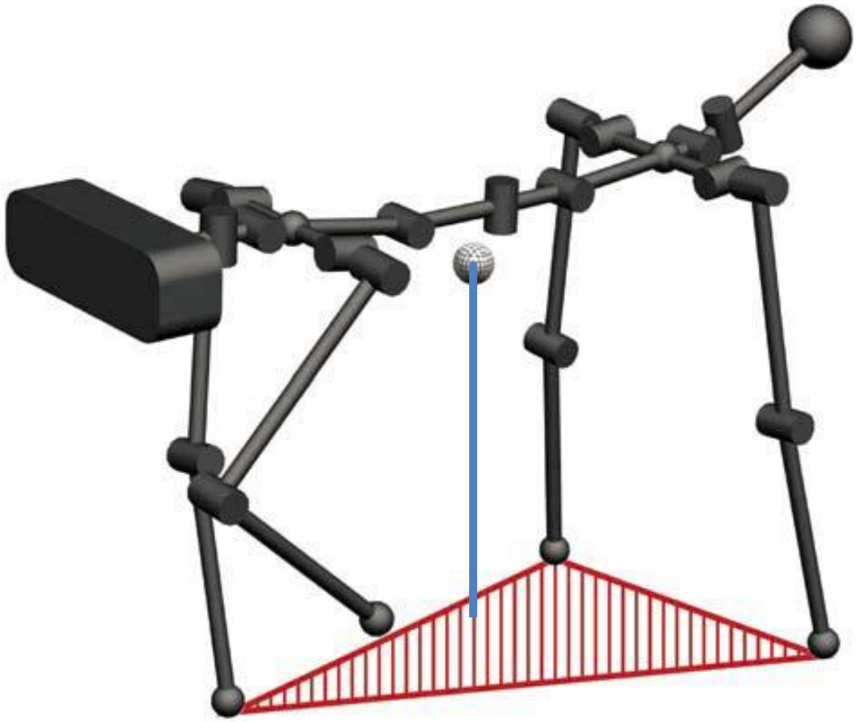
Fig. 2

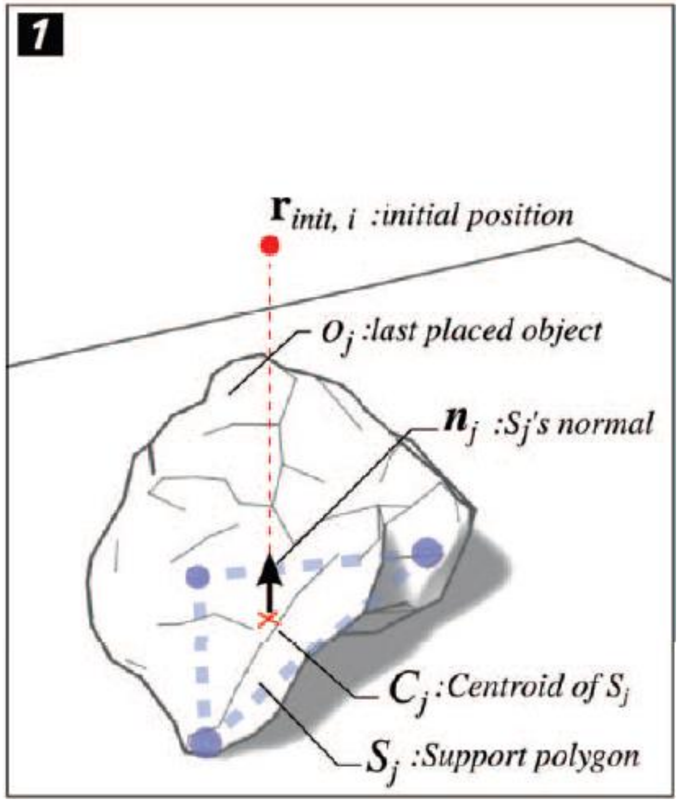


start
set initial pose $\boldsymbol{p}_{init,\,i}$
valid pose search
found valid pose $\boldsymbol{p}'_i$
no
yes
randomize the initial pose $\boldsymbol{p}_{init,\,i}$
gradient descent
x solutions found?
no
yes
return the best pose $\boldsymbol{p}^*_i$ with the minimal cost $f(\boldsymbol{p}^*_i)$
end
gradient descent
start
compute $\Delta\boldsymbol{p}$
compute next pose $\boldsymbol{p}_{n+1,i} = \boldsymbol{p}_{n,i} + \Delta\boldsymbol{p}$
valid pose search
calculate the cost $f(\boldsymbol{p}'_{n+1,i})$, $f(\boldsymbol{p}'_{n,i})$
if $f(\boldsymbol{p}_{n+1,i}) - f(\boldsymbol{p}_{n,i}) < \gamma$
no
yes
return the local optimum pose $\boldsymbol{p}^*_{local,\,i}$
end

Fig. 2

Support Polygon: Measure of static stability





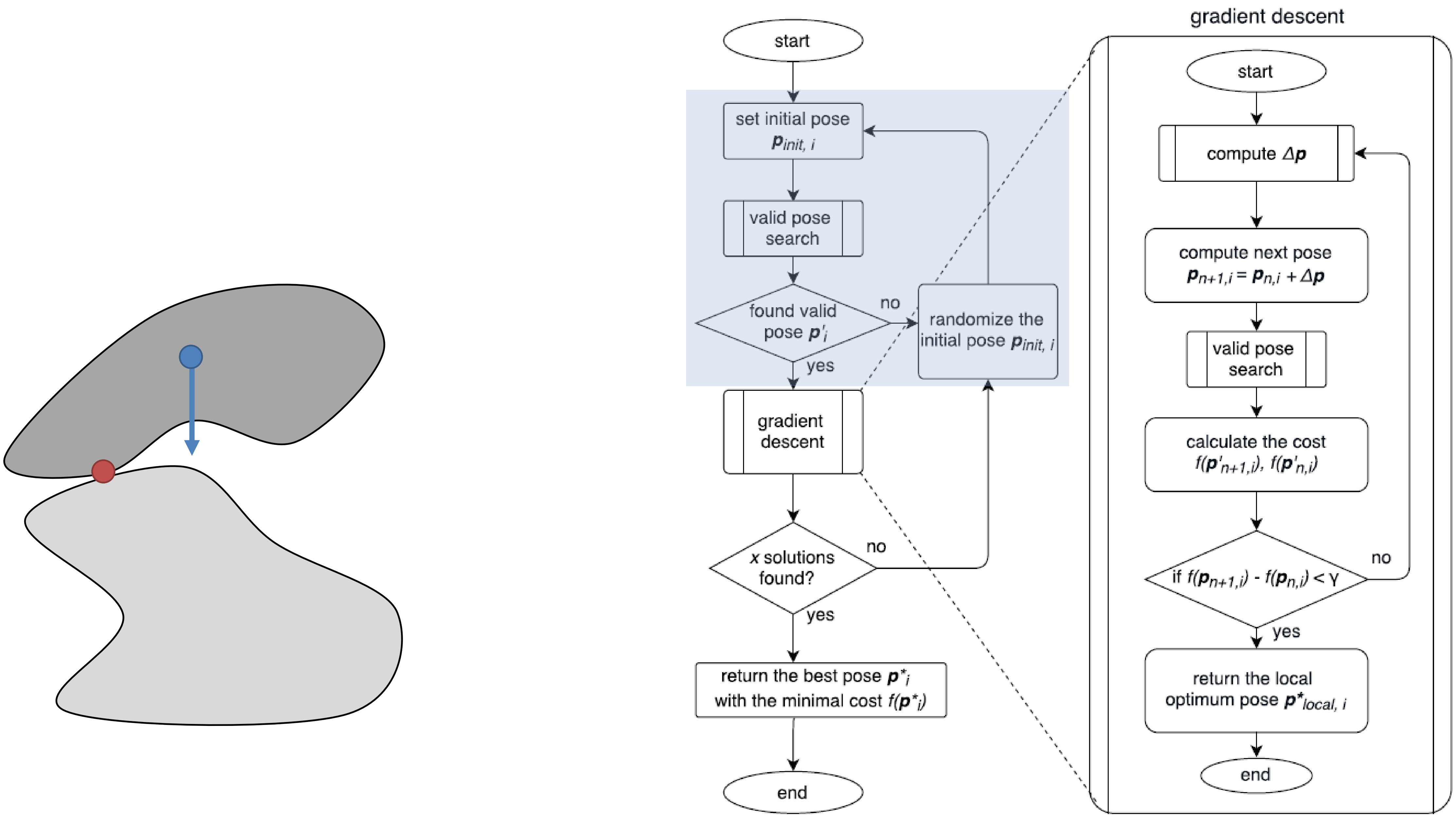
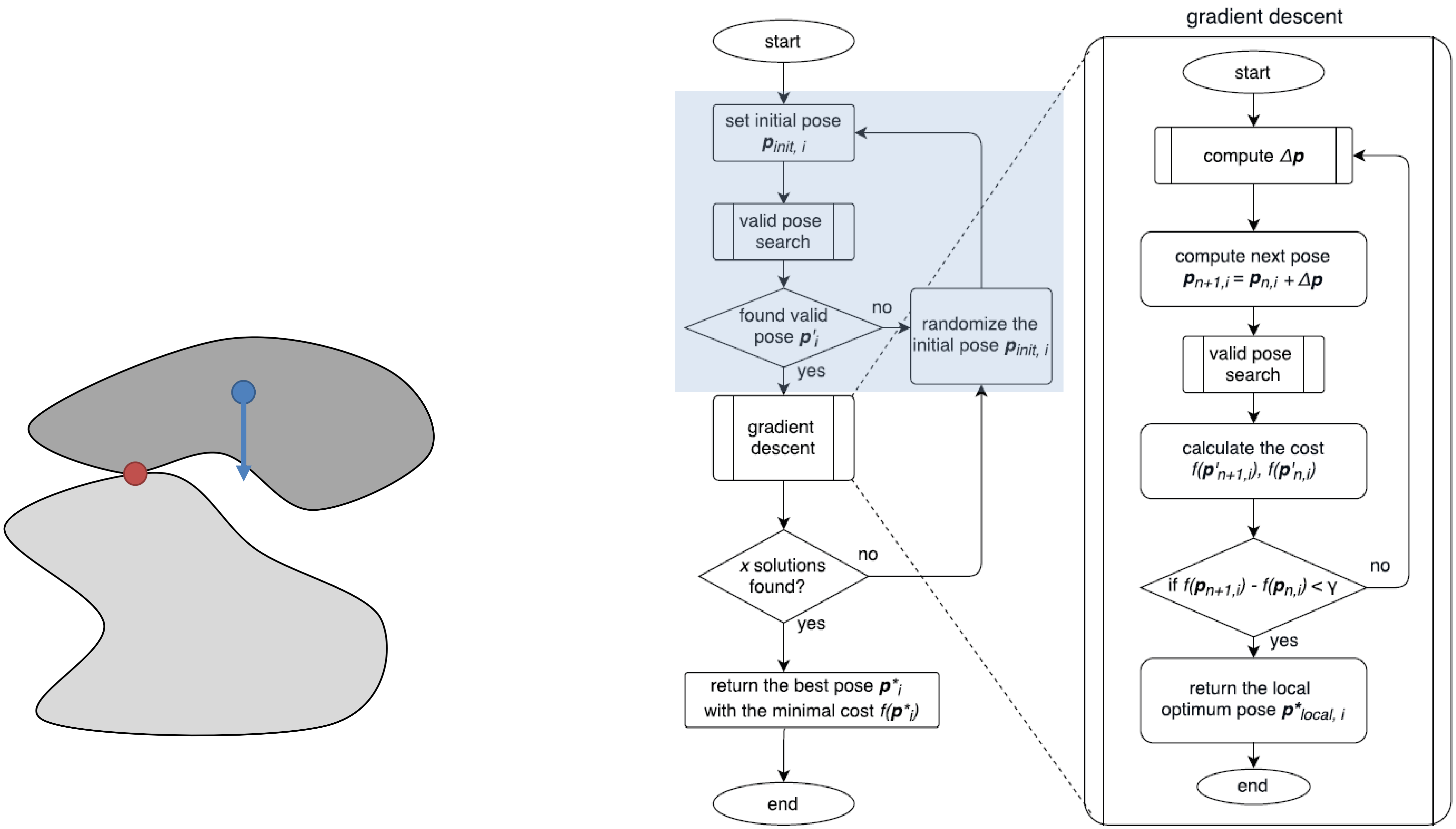
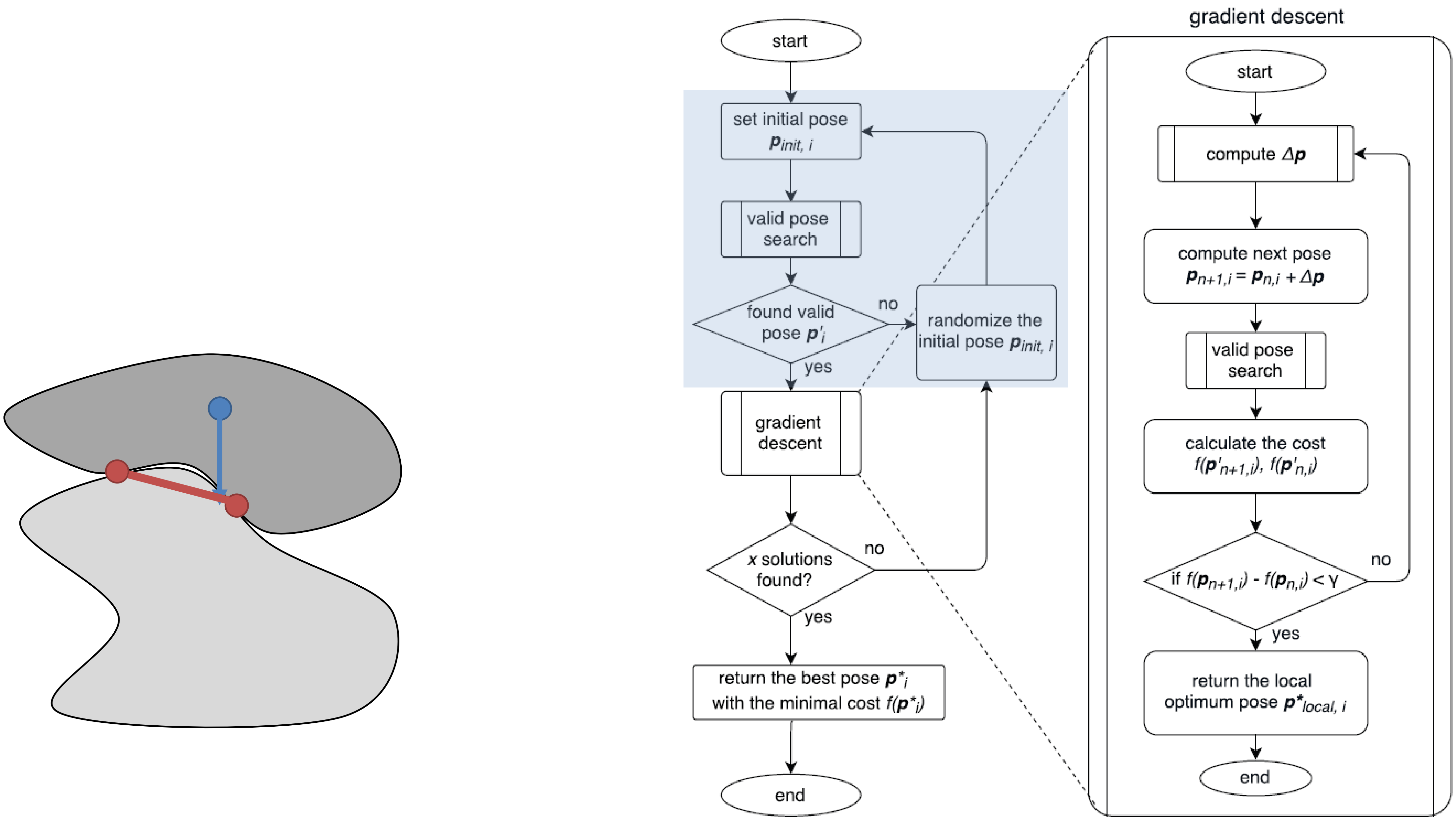

gradient descent
start
set initial pose
$p_{init,\ i}$
valid pose search
found valid pose p'_i
no
yes
randomize the initial pose $p_{init,\ i}$
gradient descent
x solutions found?
no
yes
return the best pose p^*_i with the minimal cost $f(p^*_i)$
end
start
compute Δp
compute next pose
$p_{n+1,i} = p_{n,i} + \Delta p$
valid pose search
calculate the cost
$f(p'_{n+1,i})$, $f(p'_{n,i})$
if $f(p_{n+1,i}) - f(p_{n,i}) < \gamma$
no
yes
return the local optimum pose $p^*_{local,\ i}$
end

Fig. 2



start
set initial pose $\boldsymbol{p}_{init,\,i}$
valid pose search
found valid pose $\boldsymbol{p}'_i$
no
yes
randomize the initial pose $\boldsymbol{p}_{init,\,i}$
gradient descent
x solutions found?
no
yes
return the best pose $\boldsymbol{p}^*_i$ with the minimal cost $f(\boldsymbol{p}^*_i)$
end
gradient descent
start
compute $\Delta\boldsymbol{p}$
compute next pose $\boldsymbol{p}_{n+1,i} = \boldsymbol{p}_{n,i} + \Delta\boldsymbol{p}$
valid pose search
calculate the cost $f(\boldsymbol{p}'_{n+1,i})$, $f(\boldsymbol{p}'_{n,i})$
if $f(\boldsymbol{p}_{n+1,i}) - f(\boldsymbol{p}_{n,i}) < \gamma$
no
yes
return the local optimum pose $\boldsymbol{p}^*_{local,\,i}$
end

Fig. 2



start
set initial pose $\boldsymbol{p}_{init,\,i}$
valid pose search
found valid pose $\boldsymbol{p}'_i$
no
yes
randomize the initial pose $\boldsymbol{p}_{init,\,i}$
gradient descent
x solutions found?
no
yes
return the best pose $\boldsymbol{p}^*_i$ with the minimal cost $f(\boldsymbol{p}^*_i)$
end
gradient descent
start
compute $\Delta\boldsymbol{p}$
compute next pose $\boldsymbol{p}_{n+1,i} = \boldsymbol{p}_{n,i} + \Delta\boldsymbol{p}$
valid pose search
calculate the cost $f(\boldsymbol{p}'_{n+1,i})$, $f(\boldsymbol{p}'_{n,i})$
if $f(\boldsymbol{p}_{n+1,i}) - f(\boldsymbol{p}_{n,i}) < \gamma$
no
yes
return the local optimum pose $\boldsymbol{p}^*_{local,\,i}$
end

Fig. 2

Gradient Descent: Cost Function

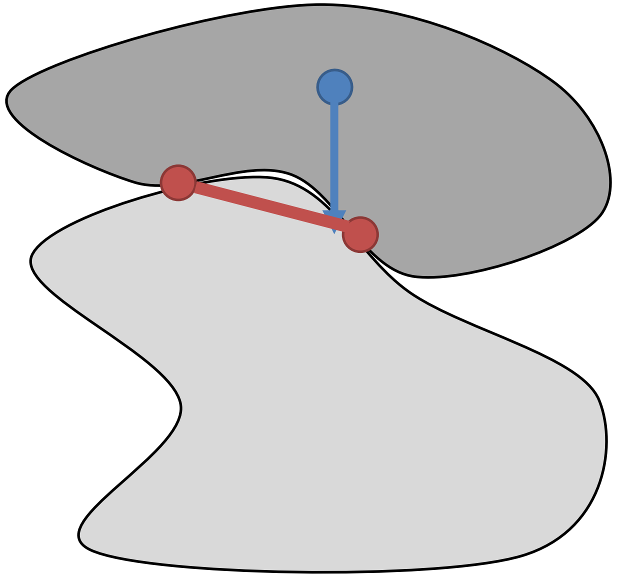
$$
\begin{aligned}
f(\mathbf{p}_{\text{contact},i}) = {} & w_1 A_i^{-1} + w_2 E_{\text{kin}}(\mathbf{p}_{\text{contact},i}) \\
& + w_3 \|\mathbf{r}_{P_j P_i}\| + w_4 \|\mathbf{n}_i \cdot \mathbf{v}_i\|, \qquad (3) \\
\text{s.t. } & w_j \geq 0 \ \forall j \in 1, \ldots, 4
\end{aligned}
$$

Gradient Descent: Cost Function

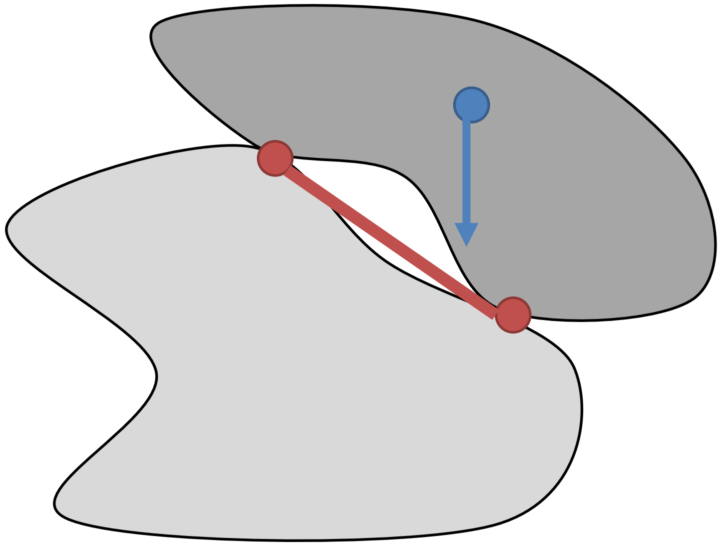
Area of support polygon

$$f(\mathbf{p}_{\text{contact},i}) = w_1 A_i^{-1} + w_2 E_{\text{kin}}(\mathbf{p}_{\text{contact},i}) + w_3 \|\mathbf{r}_{P_j P_i}\| + w_4 \|\mathbf{n}_i \cdot \mathbf{v}_i\|, \quad (3)$$

$$\text{s.t. } w_j \geq 0 \ \forall j \in 1, \ldots, 4$$



vs

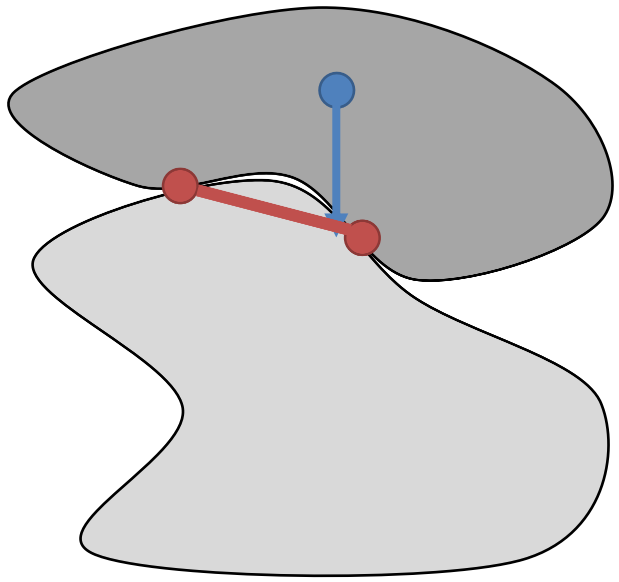


Gradient Descent: Cost Function

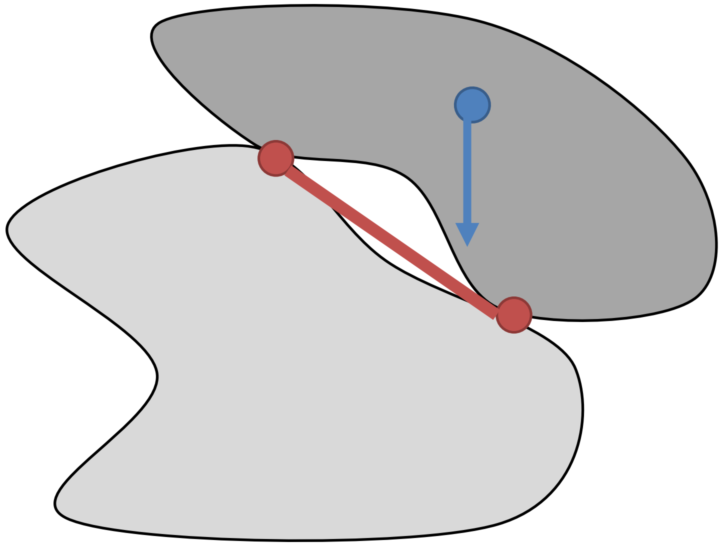
Kinetic energy (based on simulation)

$$f(\mathbf{p}_{\text{contact},i}) = w_1 A_i^{-1} + w_2 E_{\text{kin}}(\mathbf{p}_{\text{contact},i}) + w_3 \|\mathbf{r}_{P_j P_i}\| + w_4 \|\mathbf{n}_i \cdot \mathbf{v}_i\|, \quad (3)$$

$$\text{s.t. } w_j \geq 0 \ \forall j \in 1, \ldots, 4$$



VS

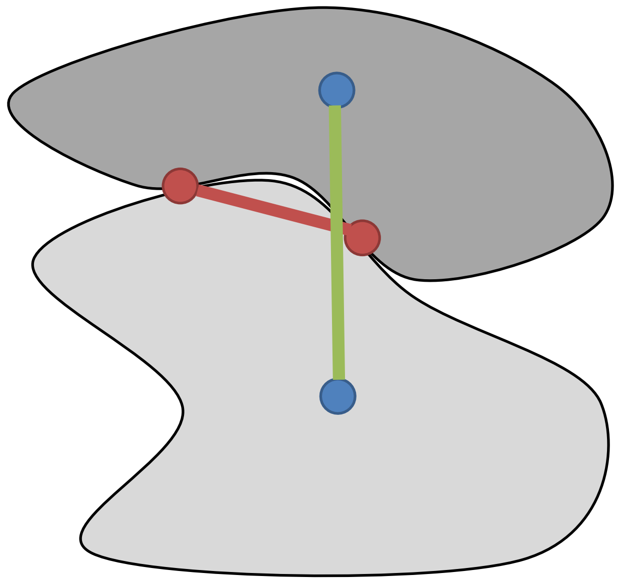


Gradient Descent: Cost Function

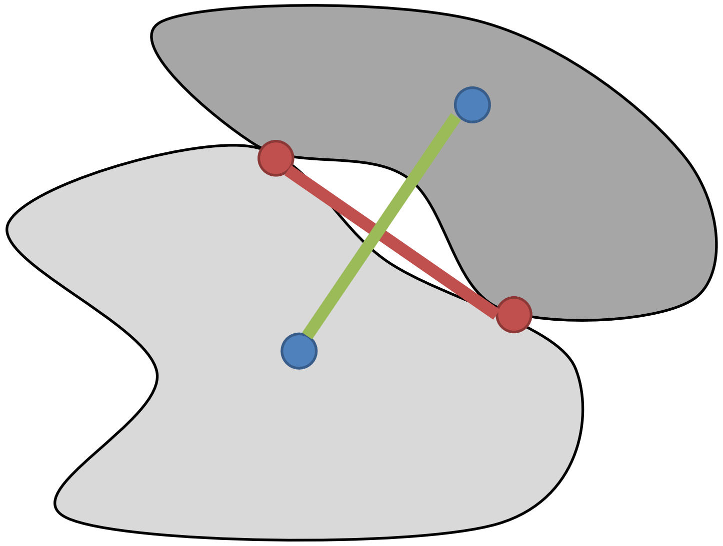
$$f(\mathbf{p}_{\text{contact},i}) = w_1 A_i^{-1} + w_2 E_{\text{kin}}(\mathbf{p}_{\text{contact},i}) + w_3 \|\mathbf{r}_{P_j P_i}\| + w_4 \|\mathbf{n}_i \cdot \mathbf{v}_i\|, \quad (3)$$

$$\text{s.t. } w_j \geq 0 \; \forall j \in 1, \ldots, 4$$

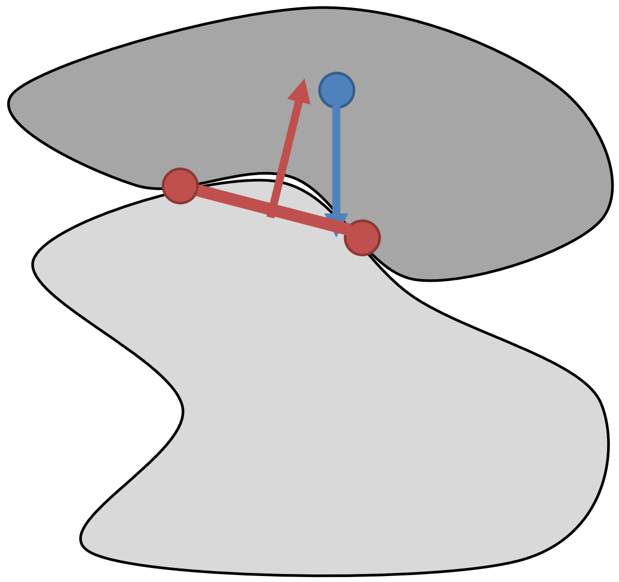
Distance between rocks



vs



Gradient Descent: Cost Function



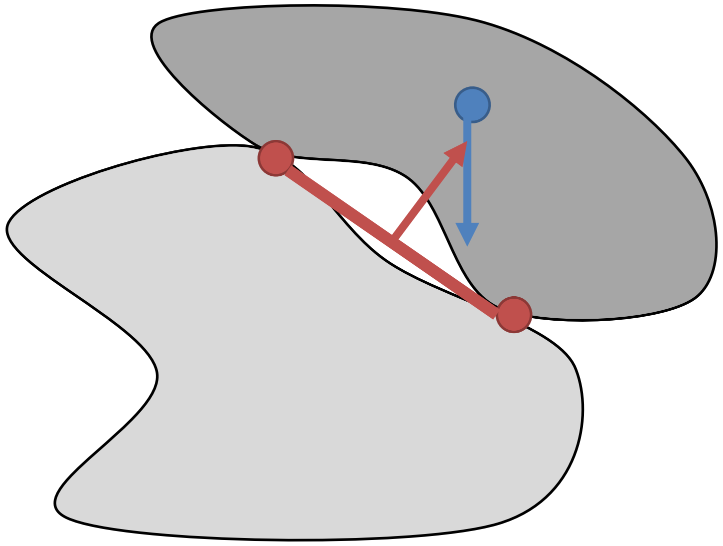
$$f(\mathbf{p}_{\text{contact},i}) = w_1 A_i^{-1} + w_2 E_{\text{kin}}(\mathbf{p}_{\text{contact},i}) + w_3 \|\mathbf{r}_{P_j P_i}\| + w_4 \|\mathbf{n}_i \cdot \mathbf{v}_i\|, \qquad (3)$$

$$\text{s.t. } w_j \geq 0 \; \forall j \in 1, \ldots, 4$$

"surface normal deviation from thrust line"

$w_4(\mathbf{n}_i \cdot \mathbf{v}_i)$?

vs



Gradient Descent: Cost Function

Area of support polygon

Kinetic energy (based on simulation)

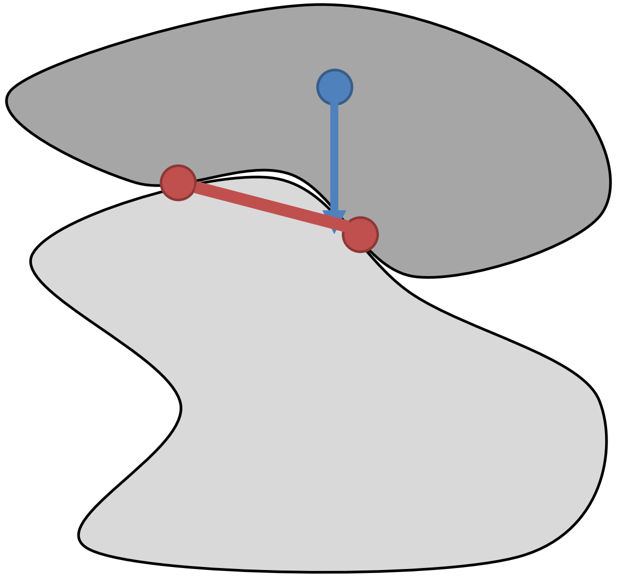
$$f(\mathbf{p}_{\text{contact},i}) = w_1 A_i^{-1} + w_2 E_{\text{kin}}(\mathbf{p}_{\text{contact},i}) + w_3 \|\mathbf{r}_{P_j P_i}\| + w_4 \|\mathbf{n}_i \cdot \mathbf{v}_i\|, \quad (3)$$

$$\text{s.t. } w_j \geq 0 \ \forall j \in 1, \ldots, 4$$

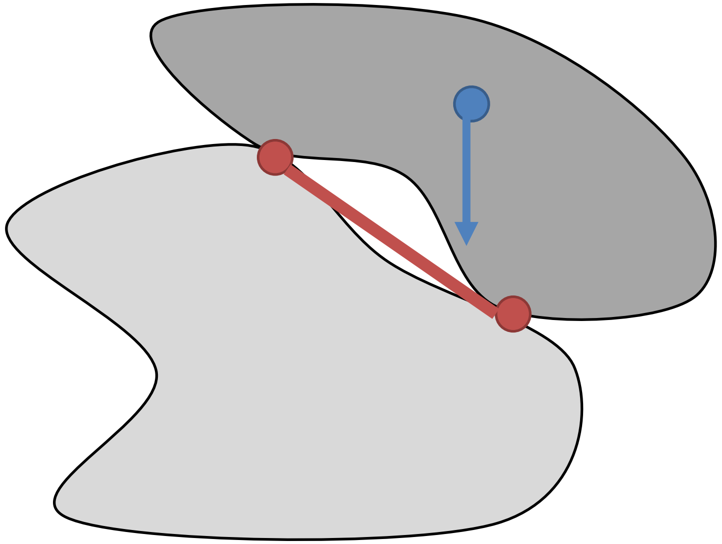
"surface normal deviation from thrust line"

Distance between rocks

$w_4(\mathbf{n}_i \cdot \mathbf{v}_i)$?

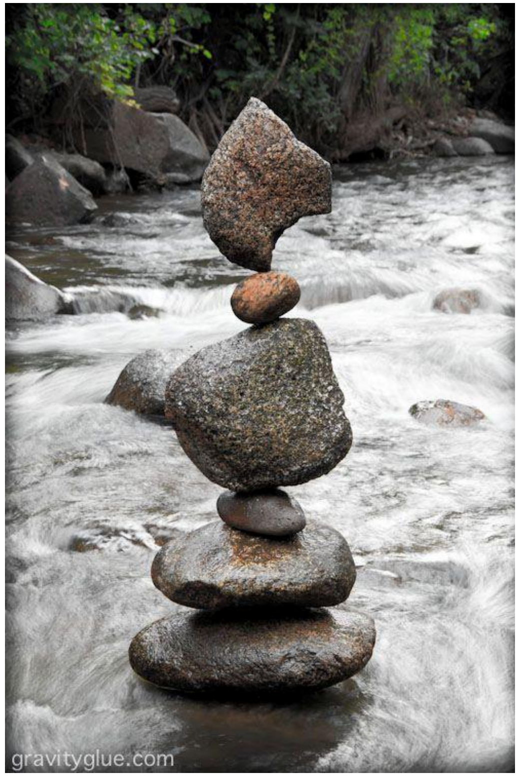


vs



Weight	Value
w_1	0.179
w_2	0.472
w_3	0.094
w_4	0.255

vs


gravityglue.com

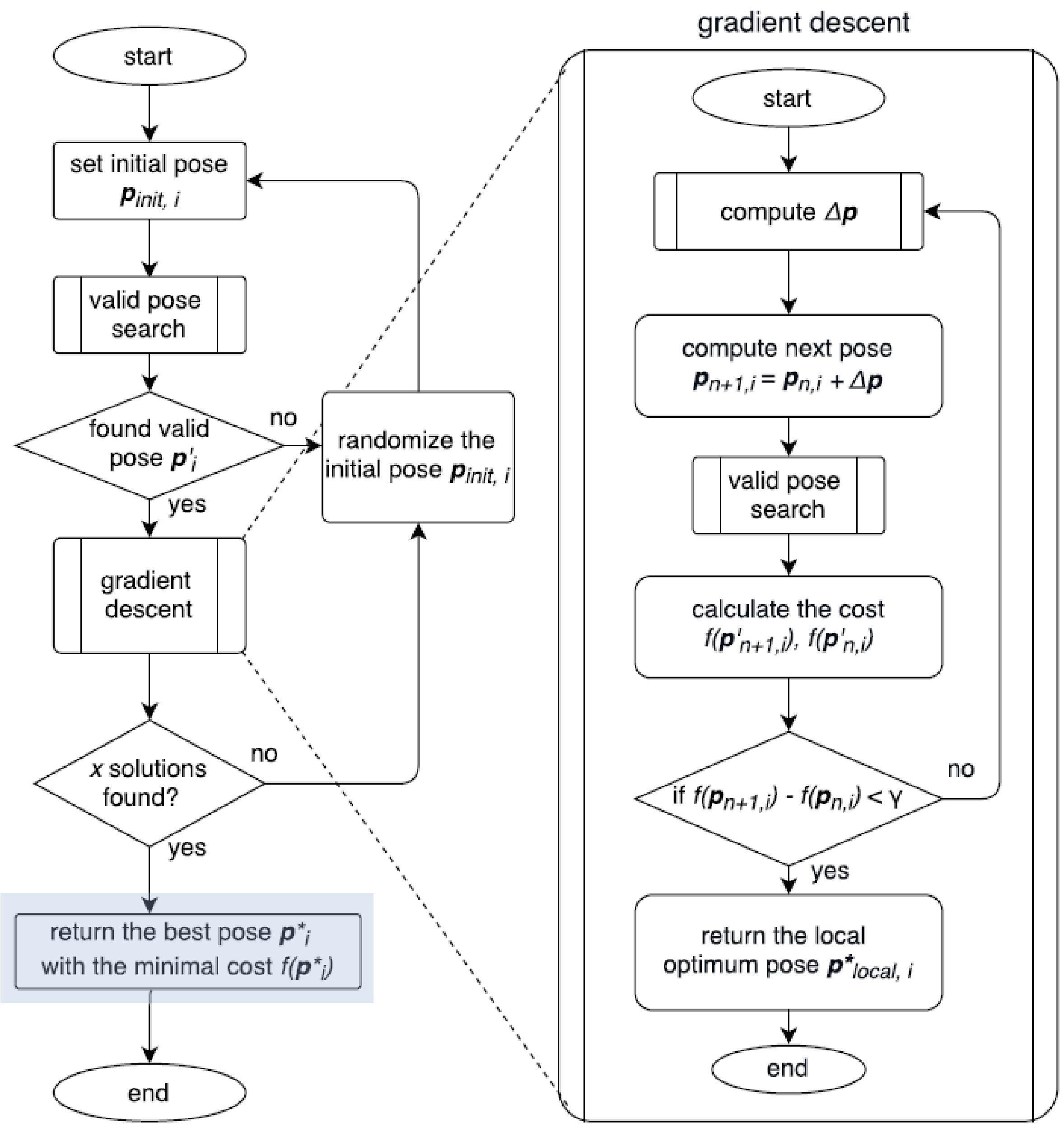

start
set initial pose $\boldsymbol{p}_{init,\ i}$
valid pose search
found valid pose $\boldsymbol{p}'_i$
no
yes
randomize the initial pose $\boldsymbol{p}_{init,\ i}$
gradient descent
x solutions found?
no
yes
return the best pose $\boldsymbol{p}^*_i$ with the minimal cost $f(\boldsymbol{p}^*_i)$
end
gradient descent
start
compute $\Delta\boldsymbol{p}$
compute next pose $\boldsymbol{p}_{n+1,i} = \boldsymbol{p}_{n,i} + \Delta\boldsymbol{p}$
valid pose search
calculate the cost $f(\boldsymbol{p}'_{n+1,i})$, $f(\boldsymbol{p}'_{n,i})$
if $f(\boldsymbol{p}_{n+1,i}) - f(\boldsymbol{p}_{n,i}) < \gamma$
no
yes
return the local optimum pose $\boldsymbol{p}^*_{local,\ i}$
end

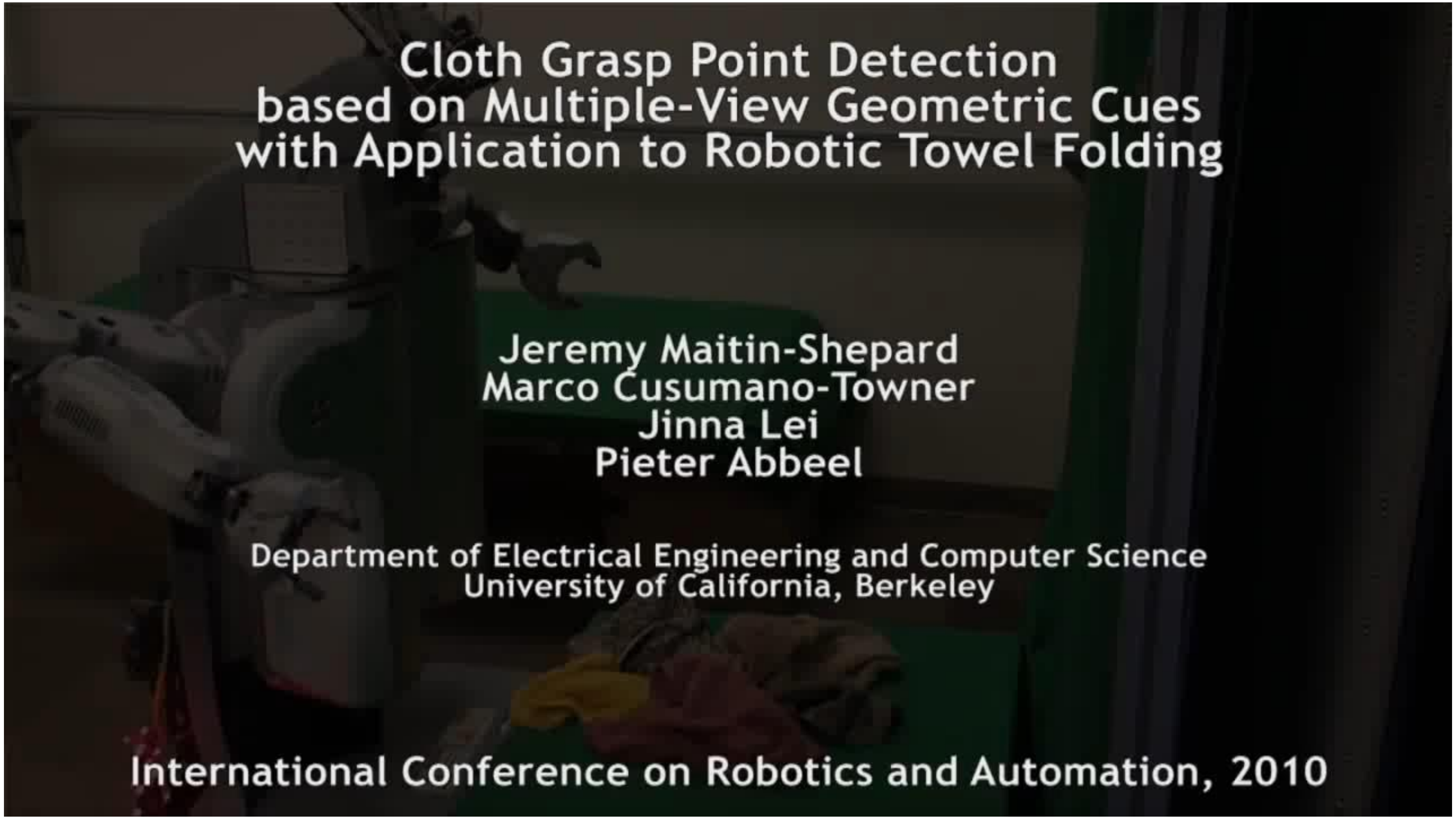
Fig. 2

Open Questions

- Sources of error
  - Physics simulation (unknown parameters, e.g. friction)
  - Pose detection
  - Grasping pose
- Unknown objects

Unknown Objects

- Recognition/Detection: Are those rocks?
- Segmentation: Which are the separate rocks?


Cloth Grasp Point Detection
based on Multiple-View Geometric Cues
with Application to Robotic Towel Folding
Jeremy Maitin-Shepard
Marco Cusumano-Towner
Jinna Lei
Pieter Abbeel
Department of Electrical Engineering and Computer Science
University of California, Berkeley
International Conference on Robotics and Automation, 2010

Challenge: What does a rock look like?

Take CIS 580: Machine Perception

Open Questions

- Error in the physics simulation
  - Physics simulation (unknown parameters, e.g. friction)
  - Pose detection
  - Grasping pose
- Unknown objects
- Physical properties of unknown objects
- Planning for contacts

Upcoming Schedule

- 11/22: Thanksgiving, no classes
- 11/27: Multi-robot systems with Dr. Ani Hsieh
- 11/29: Emerging trends in robotics
- 12/4, 12/6: Final project presentations

Final projects due 12/10
(no penalty deadline 12/12)

Next Time: Dr. Ani Hsieh
Multi-Robot Systems