KTH ROYAL INSTITUTE
OF TECHNOLOGY

Introduction to Robotics

DD2410

Lecture 6 - Control, Grasping

Schedule - Lab assigments

Aug 31 - ROS Introduction (Sep 09) — Parag Khanna

Sep 09 - Kinematics (Sep 16, 17:00) — Marco Moletta

Sep 16 - Planning (Sep 23, 17:00) — Alberta Longhini

Sep 22 - Mapping (Sep 30, 17:00) — Ludvig Ericson

Sep 30 - Pick.and-place Project (Oct 14) — Ignacio Torroba

TA Help sessions:
Aug 31, Sep 5, 6, 9, 12, 16, 19, 22, 26, 30
Oct 7, 10, 14

NOTE: Assignments must be submitted by the DEADLINES to be eligible for higher grades

Schedule - Lab assigments

Sep 9 - Kinematics (Sep 16, 17:00)

Kattis will test your solution on 4 test cases (2 E-level and 2 C-level)
- You get 10 Points per **E**-level Solution (SCARA-Robot)
- You get 1 Points per **C**-Level Solution (KUKA-Robot)

Max score is 22 points.

E grade for >= 20 points. **Accepted (20)**

C grade for >= 22 points. **Accepted (22)**

'Passing' with <20 points: **Accepted (0)**
This means your solution didn't crash, but it is not good enough for an **E** grade

NOTE: Assignments must be submitted by the DEADLINES to be eligible for higher grades

Examination - Assignments (LAB1)

Grades for the assignments will be reported 3 times during the fall:

at their respective deadline
at the time of the exam in P1
at the time of the make-up exam in P2.

Assignments that have been given at least a passing grade by the respective deadline can be resubmitted for a higher grade up until the time of the make-up exam in P2.

Assignments not passed by their initial deadlines are limited to an E grade

December 22 is the **hard** final deadline for all assignments.

Kattis

You must register in Kattis - merely logging in is not enough!

Kattis is the autograding system used by the EECS school. It is used for assignments 2, 3, and 4 in this course.

Use your personal KTH log-in. Kattis is equipped with a plagiarism checker, and if another student's solution is submitted with your account, this will count as attempted plagiarism.

Kattis is **not a debugging tool**. Ensure that all your code works in your own development environment, with all the supplied practice test cases, before submitting to Kattis.

Kattis

You must register in Kattis - merely logging in is not enough!

We will check all submissions to the "Hello World" assignment by wednesday 17:00. We will tell you when the results are published. If your results are not registered by then, it means that something is likely missing in your registration. Fix it before 17:00 on Friday to be sure to get your grades for assignment 2!

Schedule - Lectures

Aug 30 - 1. Intro, Course fundamentals, Topics, What is a Robot, History, Applications.

Aug 31 - 3 ROS Introduction

Aug 31 - 2 Manipulators, Kinematics
Sep 07 - 4. Differential kinematics, dynamics
Sep 09 - 5. Actuators, sensors I (force, torque, encoders, ...)
Sep 12 - 6. Grasping, Motion, Control

Sep 14 - 7. Planning (RRT, A*, ...)
Sep 19 - 8. Behavior Trees and Task Switching

Sep 21 - 9. Mobility and sensing II (distance, vision, radio, GPS, ...)
Sep 26 - 10. Localisation (where are we?)
Sep 28 - 11. Mapping (how to build the map to localise/navigate w.r.t.?)
Oct 03 - 12. Navigation (how do I get from A to B?)

Oct 05 - Q/A - Open questions to your teachers.

Syllabus

R-MPC: 8, 9
RH: A6, A7, C28
JJ Craig: 9,10,11

- Control
  - Joint level vs full system
  - Position control
  - Velocity control, CTC
  - practical considerations
- Grasping
  - Definitions
  - Examples

- Motion control
  - The system state $\mathbf{x(t)}$ should follow a desired state $\mathbf{x_d(t)}$ with as small errors as possible
  - Trajectories can be generated as a set of waypoints that are interpolated, or be generated by advanced planners (see later lecture).

- Motion control
  - The system state **x(t)** should follow a desired state **x_d(t)** with as small errors as possible
  - Trajectories can be generated as a set of waypoints that are interpolated, or be generated by advanced planners (see later lecture).

Chapter 4
in R-MPC

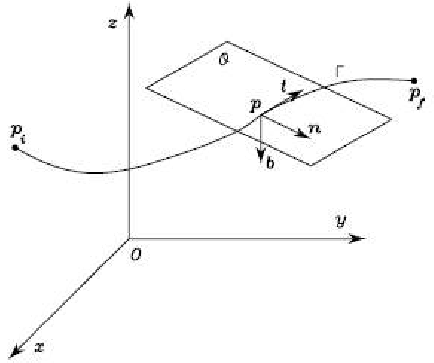
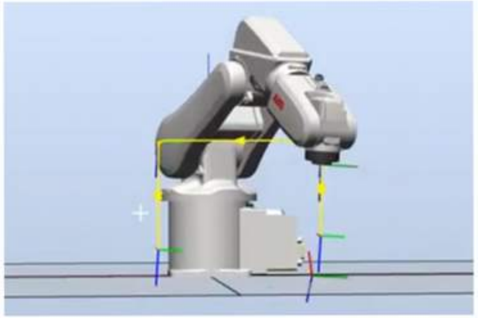
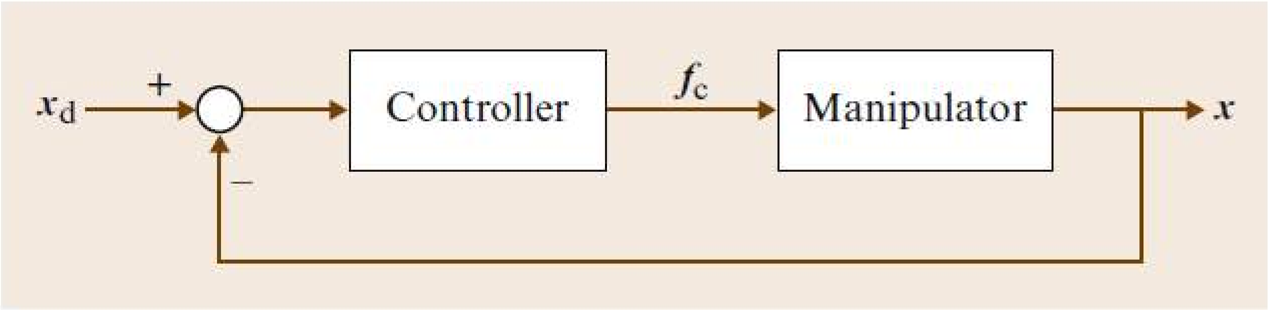


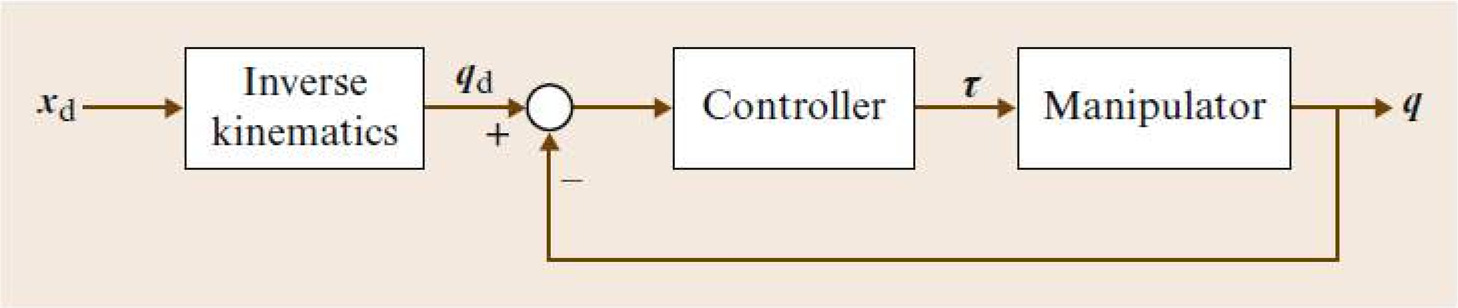
Fig. 4.11. Parametric representation of a path in space



Control

- Control domain
  - Joint space or cartesian?





Independent joint control

- Each joint is controlled individually
- The dynamic effects of other joints are treated as disturbances
- Easy to implement, non-expensive computation
- Large errors when working close to dynamic limits

Dynamics (reminder)

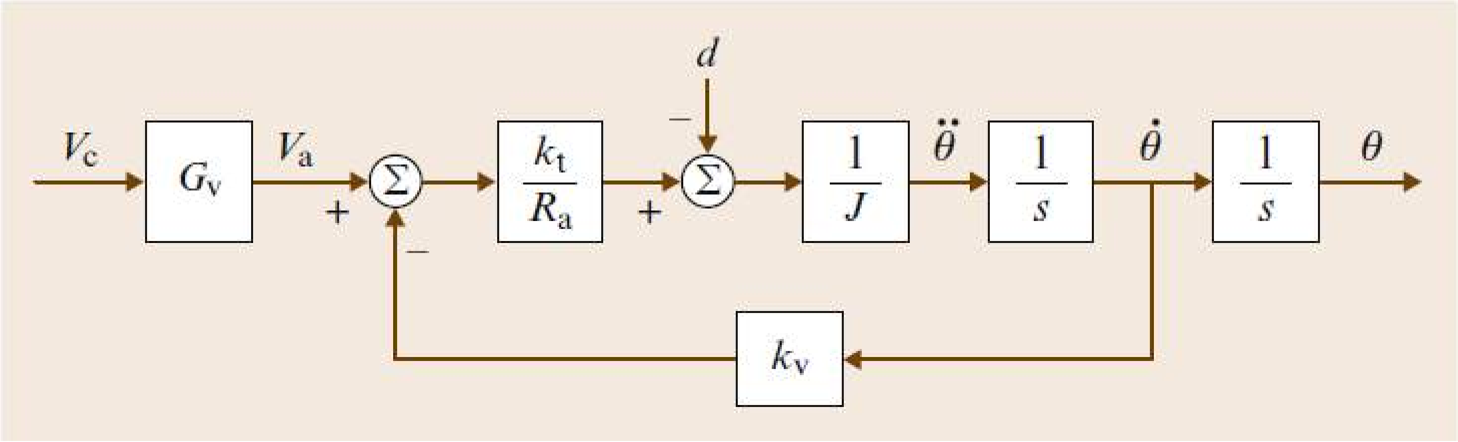
Outward iterations: $i : 0 \rightarrow 5$

$$
\begin{aligned}
{}^{i+1}\omega_{i+1} &= {}^{i+1}_{i}R\,{}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1},\\
{}^{i+1}\dot{\omega}_{i+1} &= {}^{i+1}_{i}R\,{}^{i}\dot{\omega}_i + {}^{i+1}_{i}R\,{}^{i}\omega_i \times \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1},\\
{}^{i+1}\dot{v}_{i+1} &= {}^{i+1}_{i}R({}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{i+1}) + {}^{i}\dot{v}_i),\\
{}^{i+1}\dot{v}_{C_{i+1}} &= {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{C_{i+1}}\\
&\quad + {}^{i+1}\omega_{i+1} \times ({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{C_{i+1}}) + {}^{i+1}\dot{v}_{i+1},\\
{}^{i+1}F_{i+1} &= m_{i+1}\,{}^{i+1}\dot{v}_{C_{i+1}},\\
{}^{i+1}N_{i+1} &= {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\omega_{i+1}.
\end{aligned}
$$

Inward iterations: $i : 6 \rightarrow 1$

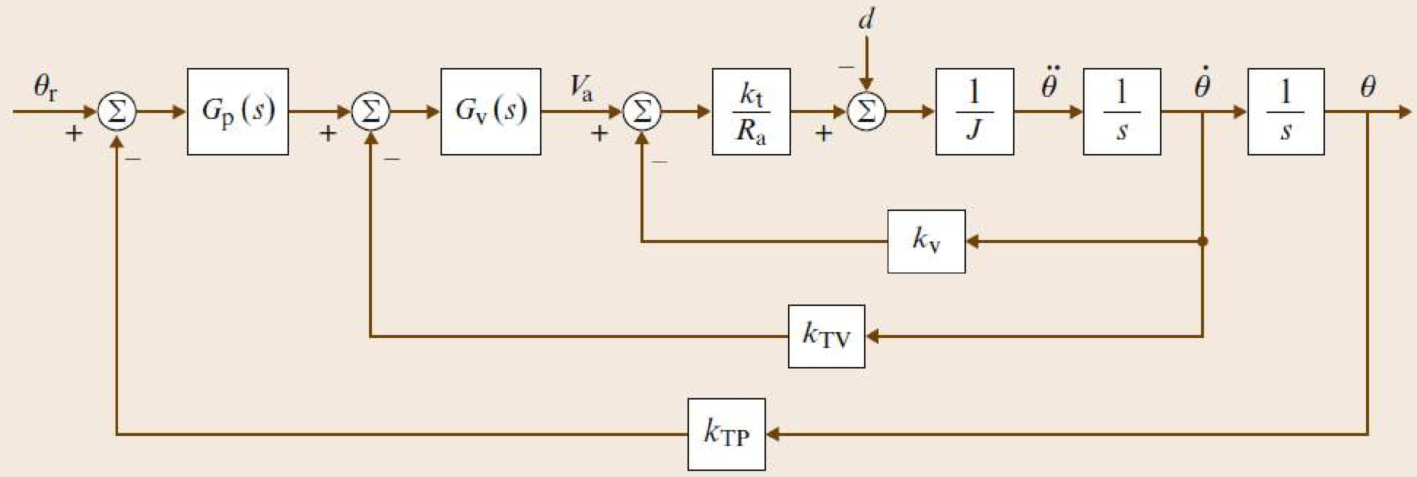
$$
\begin{aligned}
{}^{i}f_i &= {}^{i}_{i+1}R\,{}^{i+1}f_{i+1} + {}^{i}F_i,\\
{}^{i}n_i &= {}^{i}N_i + {}^{i}_{i+1}R\,{}^{i+1}n_{i+1} + {}^{i}P_{C_i} \times {}^{i}F_i\\
&\quad + {}^{i}P_{i+1} \times {}^{i}_{i+1}R\,{}^{i+1}f_{i+1},\\
\tau_i &= {}^{i}n_i^T\,{}^{i}\hat{Z}_i.
\end{aligned}
$$

Independent joint control - model of a single joint



- V_c, V_a - Input and amplifier voltage
- k_t, k_v - torque and motor constants
- d - disturbance
- J - link inertia as seen from the motor

Independent joint control - feedback control



- G_p - Position controller (P)
- G_v - Velocity controller (PI)
- k_{TV}, k_{TP} - transducer constants

$$G_p(s) = K_P\,, \qquad G_v(s) = K_V \frac{1 + sT_V}{s}\,,$$

Full manipulator control

- Given the system dynamics notation:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + \tau_g(q) = \tau \, .$$

- PID control can be used to reach a given setpoint, without explicit knowledge of system dynamics

$$\tau = K_P(q_d - q) + K_I \int f(q_d - q)\,\mathrm{d}t - K_V \dot{q}$$

- Integrator part will correct static effects of gravity
- Gains will be good for local regions around a configuration
- Poor performance for highly dynamic actions

Full manipulator control

- Given the system dynamics notation:

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{\tau}_{\mathrm{g}}(\boldsymbol{q}) = \boldsymbol{\tau} \ .$$

Full manipulator control

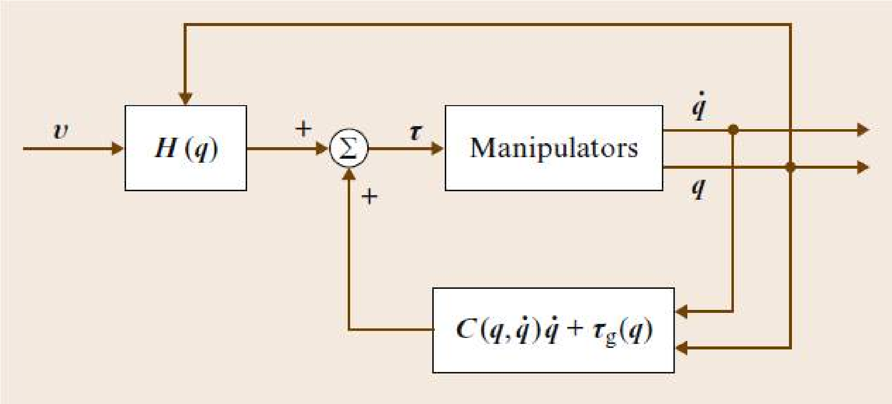
$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + \tau_g(q) = \tau \, .$$

Computed Torque Control - CTC

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + \tau_g(q) = \tau \,.$$

Computed Torque Control - CTC

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tau_g(q) = \tau \,.$$



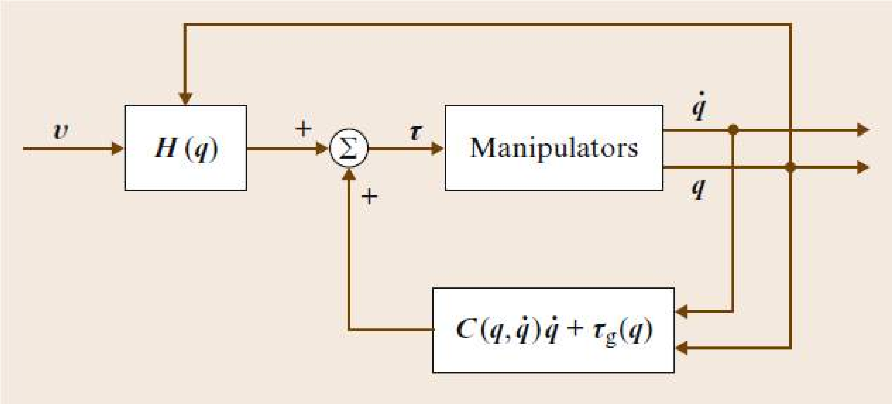
- Assume the control signal $\ddot{q} = v$
  and we get a decoupled system where we can directly assign the desired accelerations

- A dynamically well-performing tracker can be given as

$$v = \ddot{q}_d + K_V \dot{e}_q + K_P e_q$$,where e_q is the error

Computed Torque Control - CTC

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tau_g(q) = \tau \,.$$



- In practice, modelling errors will have to be treated by an extra term, see RH A6.6 for details

$$v = \ddot{q}_d + K_V \dot{e}_q + K_P e_q + \Delta v$$

- Assuming that we can measure forces/torques, we can define controllers that track a desired force $\mathbf{F_d(t)}$

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T \boldsymbol{f} = \boldsymbol{\tau}$$

Force Control

- Assuming that we can measure forces/torques, we can define controllers that track a desired force $\mathbf{F_d(t)}$

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T\boldsymbol{f} = \boldsymbol{\tau}$$

Force Control

- Assuming that we can measure forces/torques, we can define controllers that track a desired force $\mathbf{F_d(t)}$

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T\boldsymbol{f} = \boldsymbol{\tau}$$

$$\boldsymbol{\tau} = \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{K}_v\dot{\boldsymbol{q}} + \boldsymbol{J}^T\left[\boldsymbol{f}_d - k_I\int_0^t(\boldsymbol{f} - \boldsymbol{f}_d)d\tau\right]$$

- Assuming that we can measure forces/torques, we can define controllers that track a desired force $\mathbf{F_d(t)}$

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T \boldsymbol{f} = \boldsymbol{\tau}$$

$$\boldsymbol{\tau} = \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{K}_v \dot{\boldsymbol{q}} + \boldsymbol{J}^T \left[\boldsymbol{f}_d - k_I \int_0^t (\boldsymbol{f} - \boldsymbol{f}_d) d\tau \right]$$

Force Control

- Assuming that we can measure forces/torques, we can define controllers that track a desired force $\mathbf{F_d(t)}$

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T\boldsymbol{f} = \boldsymbol{\tau}$$

$$\boldsymbol{\tau} = \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{K}_v\dot{\boldsymbol{q}} + \boldsymbol{J}^T\left[\boldsymbol{f}_d - k_I\int_0^t(\boldsymbol{f} - \boldsymbol{f}_d)d\tau\right]$$

0 for static forces

Force Control

- Assuming that we can measure forces/torques, we can define controllers that track a desired force $\mathbf{F_d(t)}$

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T \boldsymbol{f} = \boldsymbol{\tau}$$

$$\boldsymbol{\tau} = \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{K}_v \dot{\boldsymbol{q}} + \boldsymbol{J}^T \left[\boldsymbol{f}_d - k_I \int_0^t (\boldsymbol{f} - \boldsymbol{f}_d) d\tau \right]$$

0 for static forces

- Note that forces and position (velocity) can typically not be tracked independently!

Velocity control

- Given the above schemes, it is possible to realize velocity controlled robots, by setting $\mathbf{x_d(t)}$ to be the integrated target velocity
- Velocity controllers allow us to implement a range of reactive robot behaviors
- Industrial manipulators that do not expose their internal controls can be seen as velocity controlled

- Velocity control for other controllers, assuming force measurements:
  - Virtual spring around x_0

$$f_d = -k(x - x_0)$$

$$v_d = \alpha(f - f_d)$$

- Velocity control for other controllers, assuming force measurements:
  - Admittance control, as virtual damping

$$f_d = -k(\dot{x})$$

$$v_d = \alpha(f)$$

- Velocity control for other controllers, assuming force measurements:
  - Virtual fixture

$$v_d = k f$$

where *k* projects on fixture

- Velocity control for other controllers, assuming force measurements:
  - Full impedance (mass, damper, spring)

- Assuming a robot with several degrees of freedom, different control strategies can be used in different subspaces:
  - position in (**x**,**y**), force in **z**
  - admittance control in (**x,y,z**), fixed orientation
  - trajectory following in **pose,** obstacle (singularity) avoidance in nullspace.

- Grasping
  - Definitions, taxonomy
  - Grippers

- Grasping
  - Form closure
  - Force closure
  - Caging

- Form closure

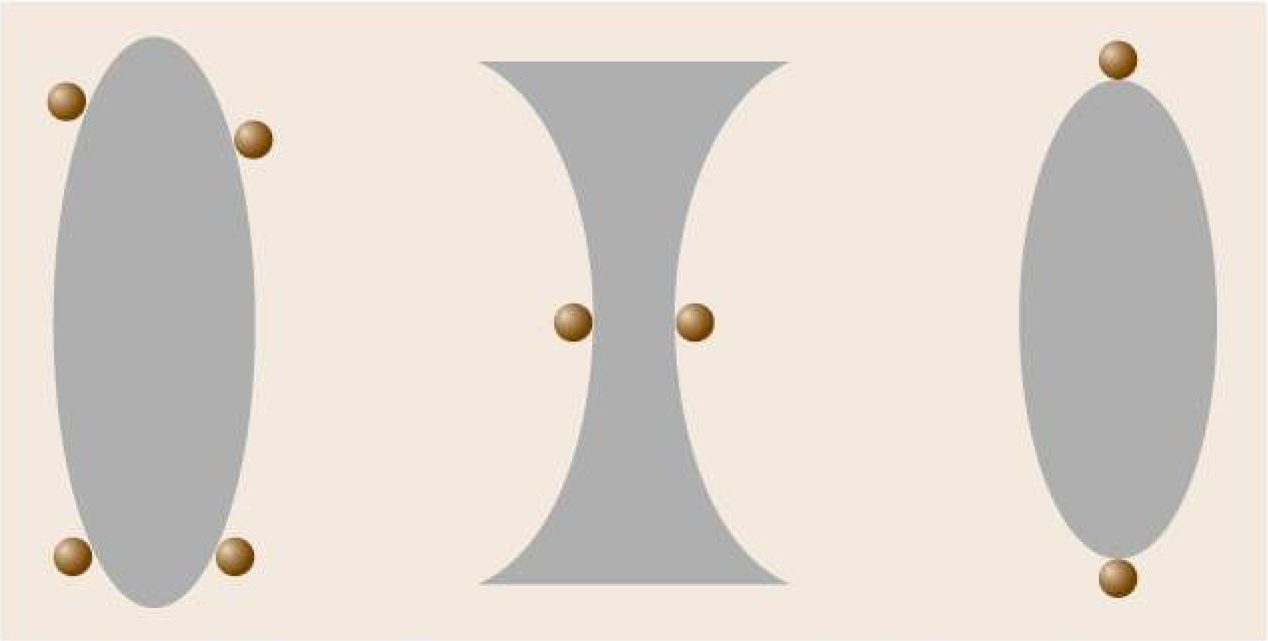


Fig. 28.8 Three planar grasps: two with form closure of different orders and one without form closure

- Gap function is distance between hand and object

$$\psi_i(\boldsymbol{u}, \boldsymbol{q})$$

  - ***u*** is object pose, ***q*** is hand configuration
  - at all contact points,

$$\psi_i(\bar{\boldsymbol{u}}, \bar{\boldsymbol{q}}) = \mathbf{0} \quad \forall\, i = 1, \ldots, n_c$$

  - A grasp has form closure **iff** the following implication holds:

$$\boldsymbol{\psi}(\bar{\boldsymbol{u}} + \mathrm{d}\boldsymbol{u}, \bar{\boldsymbol{q}}) \geq \mathbf{0} \Rightarrow \mathrm{d}\boldsymbol{u} = \mathbf{0}$$

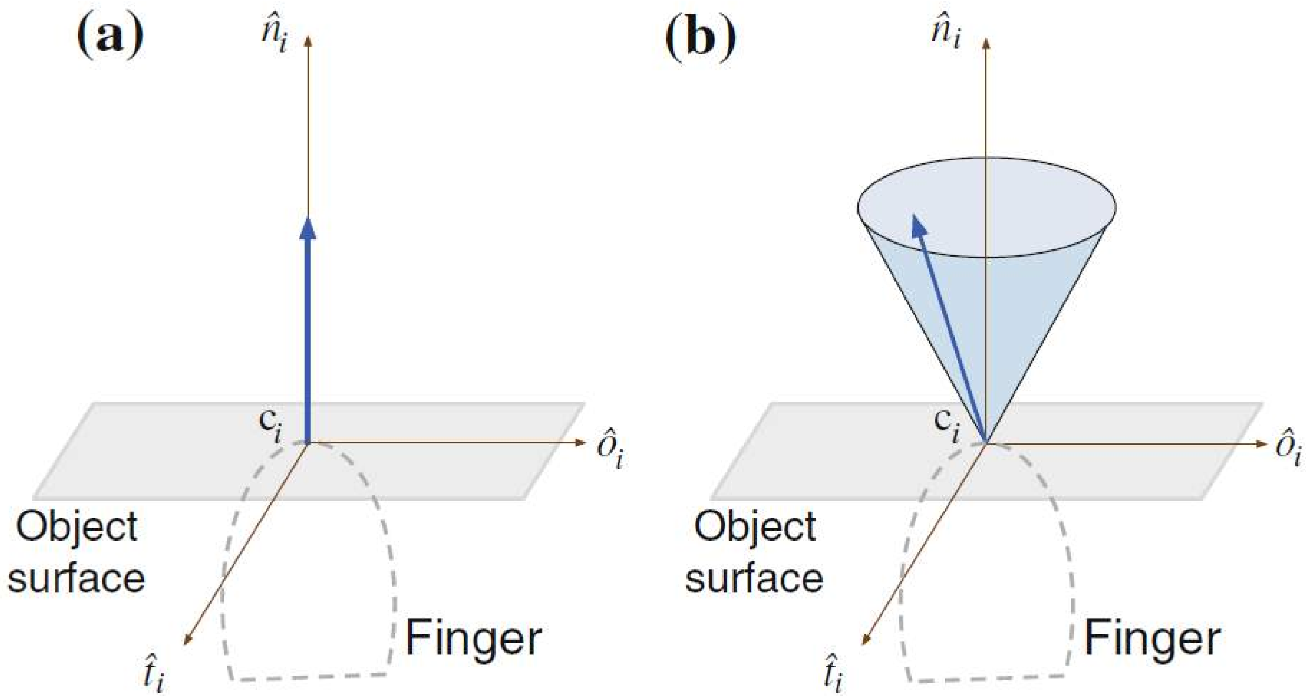
that is, there is no possible motion that increases gap

Grasping - force closure

- Contact types

Frictionless

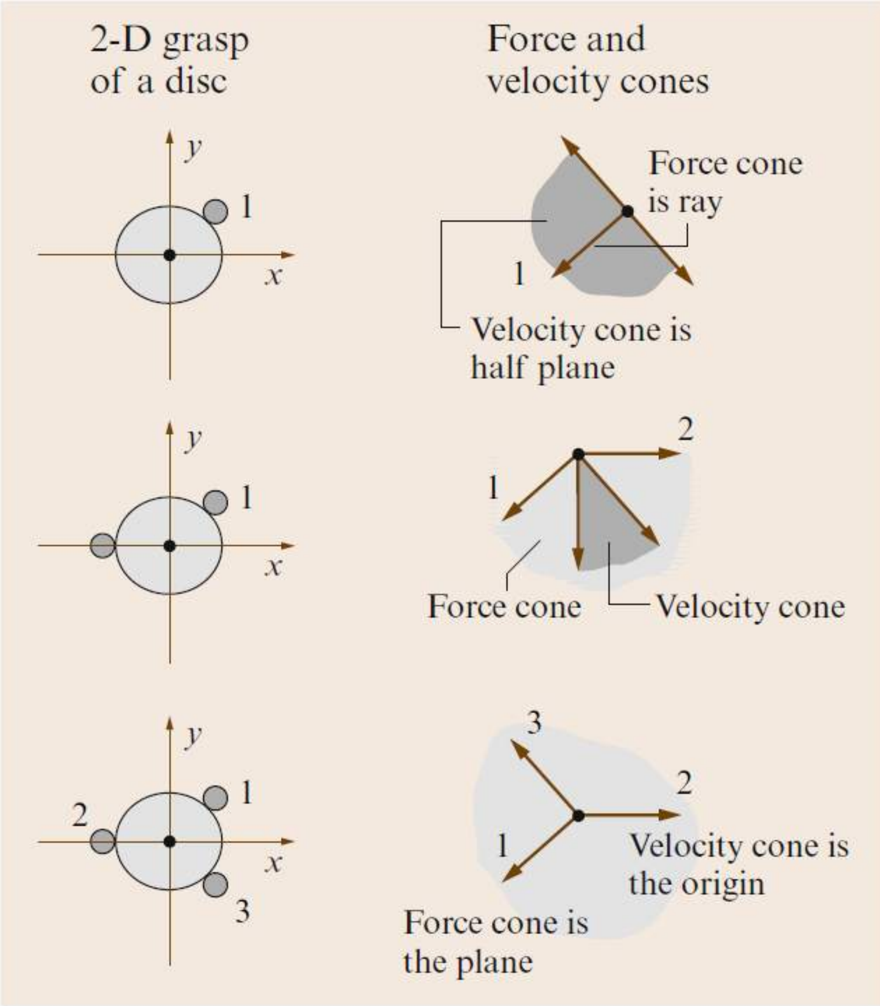
With friction



B. León et al., *From Robot to Human Grasping Simulation*,
Cognitive Systems Monographs 19, DOI: 10.1007/978-3-319-01833-1_2,

- A grasp is in force-closure if the fingers can apply, through the set of contacts, arbitrary wrenches on the object, which means that any motion of the object can be resisted by the contact forces.

  right: frictionless case



B. León et al., *From Robot to Human Grasping Simulation*, Cognitive Systems Monographs 19, DOI: 10.1007/978-3-319-01833-1_2, © Springer International Publishing Switzerland 2014

Grasping - force closure

- friction case

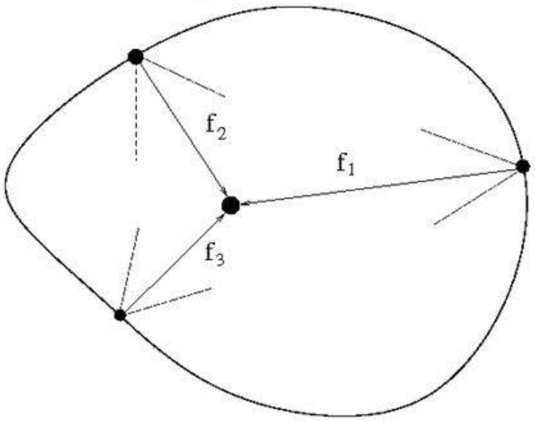
Force Closure

Need balanced forces or else object twists

2 fingers – forces oppose: $\bar{f}_1 + \bar{f}_2 = 0$

3 fingers – forces meet at point: $\bar{f}_1 + \bar{f}_2 + \bar{f}_3 = 0$

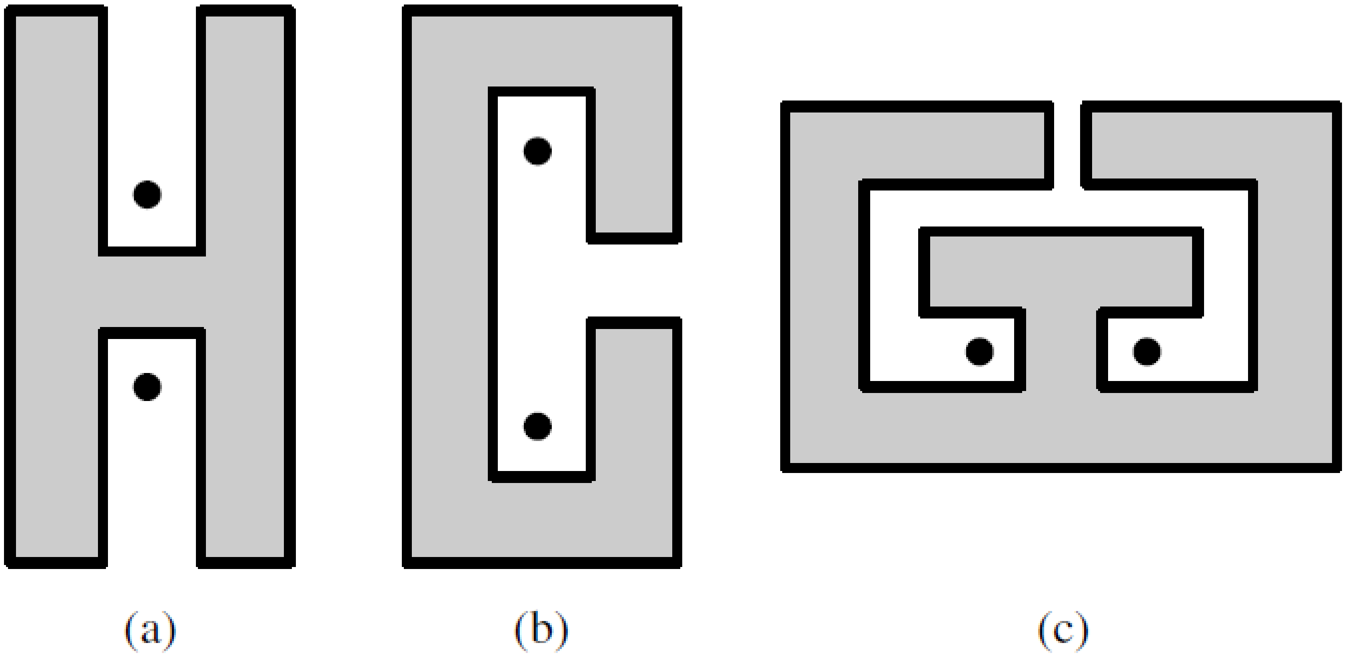
Force closure: point where forces meet lies within 3 friction cones otherwise object slips



Olivia Fox: "Kinematics & Grasping Need to know: Representing mechanism geometry Standard configurations Degrees of freedom Grippers and graspability conditions Goal"

Grasping - caging

- Caging



Alberto Rodriguez, Matt Mason, and Steve Ferry. "From Caging to Grasping" Proceedings of Robotics: Science and Systems, 2011, Los Angeles, CA, USA

"Let P be a polygon in the plane, and let C be a set of n points in the complement of the interior of P. The points capture P if P cannot be moved arbitrarily far from its original position without at least one point of C penetrating the interior of P."

Alberto Rodriguez, Matt Mason, and Steve Ferry. "From Caging to Grasping" Proceedings of Robotics: Science and Systems, 2011, Los Angeles, CA, USA

Grasping - caging

"Let P be a polygon in the plane, and let C be a set of n points in the complement of the interior of P. The points capture P if P cannot be moved arbitrarily far from its original position without at least one point of C penetrating the interior of P."

c.f. form closure:

$$\boldsymbol{\psi}(\bar{\boldsymbol{u}} + \mathrm{d}\boldsymbol{u}, \bar{\boldsymbol{q}}) \geq \mathbf{0} \Rightarrow \mathrm{d}\boldsymbol{u} = \mathbf{0}$$

Alberto Rodriguez, Matt Mason, and Steve Ferry. "From Caging to Grasping" Proceedings of Robotics: Science and Systems, 2011, Los Angeles, CA, USA

"Let P be a polygon in the plane, and let C be a set of n points in the complement of the interior of P. The points capture P if P cannot be moved arbitrarily far from its original position without at least one point of C penetrating the interior of P."

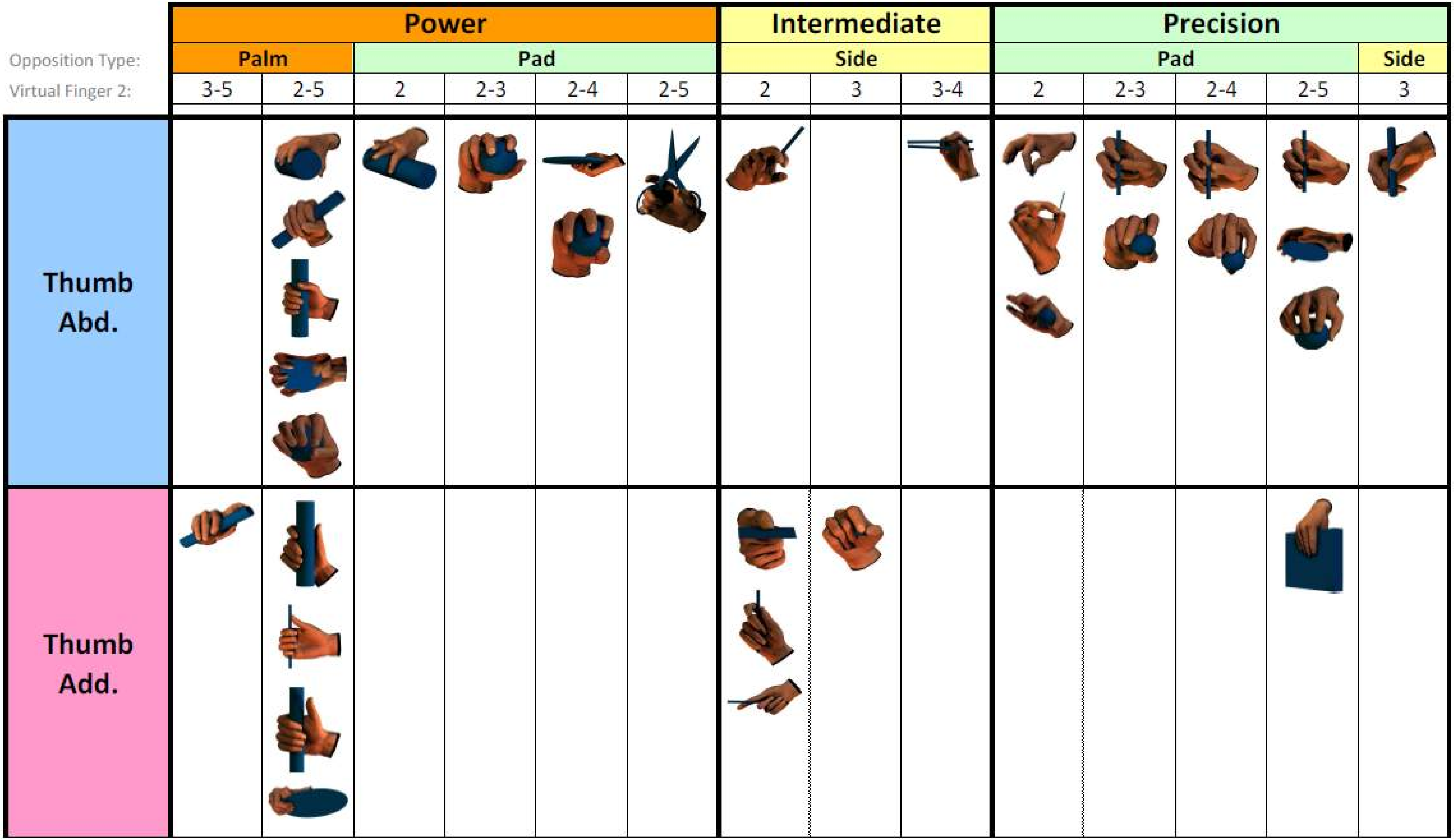
c.f. form closure:

$$\boldsymbol{\psi}(\bar{\boldsymbol{u}} + \mathrm{d}\boldsymbol{u}, \bar{\boldsymbol{q}}) \geq \mathbf{0} \Rightarrow \mathrm{d}\boldsymbol{u} = \mathbf{0}$$

$$\Psi(\bar{u}+du, \bar{q}) \geq 0 \Rightarrow du \text{ bounded}$$

Alberto Rodriguez, Matt Mason, and Steve Ferry. "From Caging to Grasping" Proceedings of Robotics: Science and Systems, 2011, Los Angeles, CA, USA

Grasping taxonomy



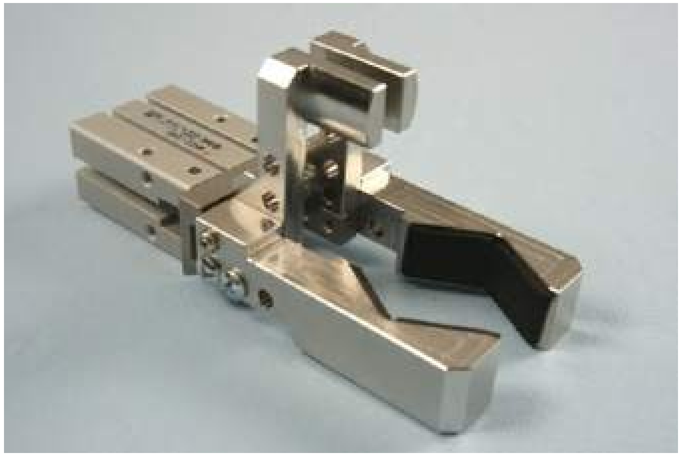
T Feix, R Pawlik, H Schmiedmayer, J Romero, D Kragic, "A comprehensive grasp taxonomy", RSS 2009

Industrial grasping

- parallell grippers

Industrial grasping

- Custom grippers



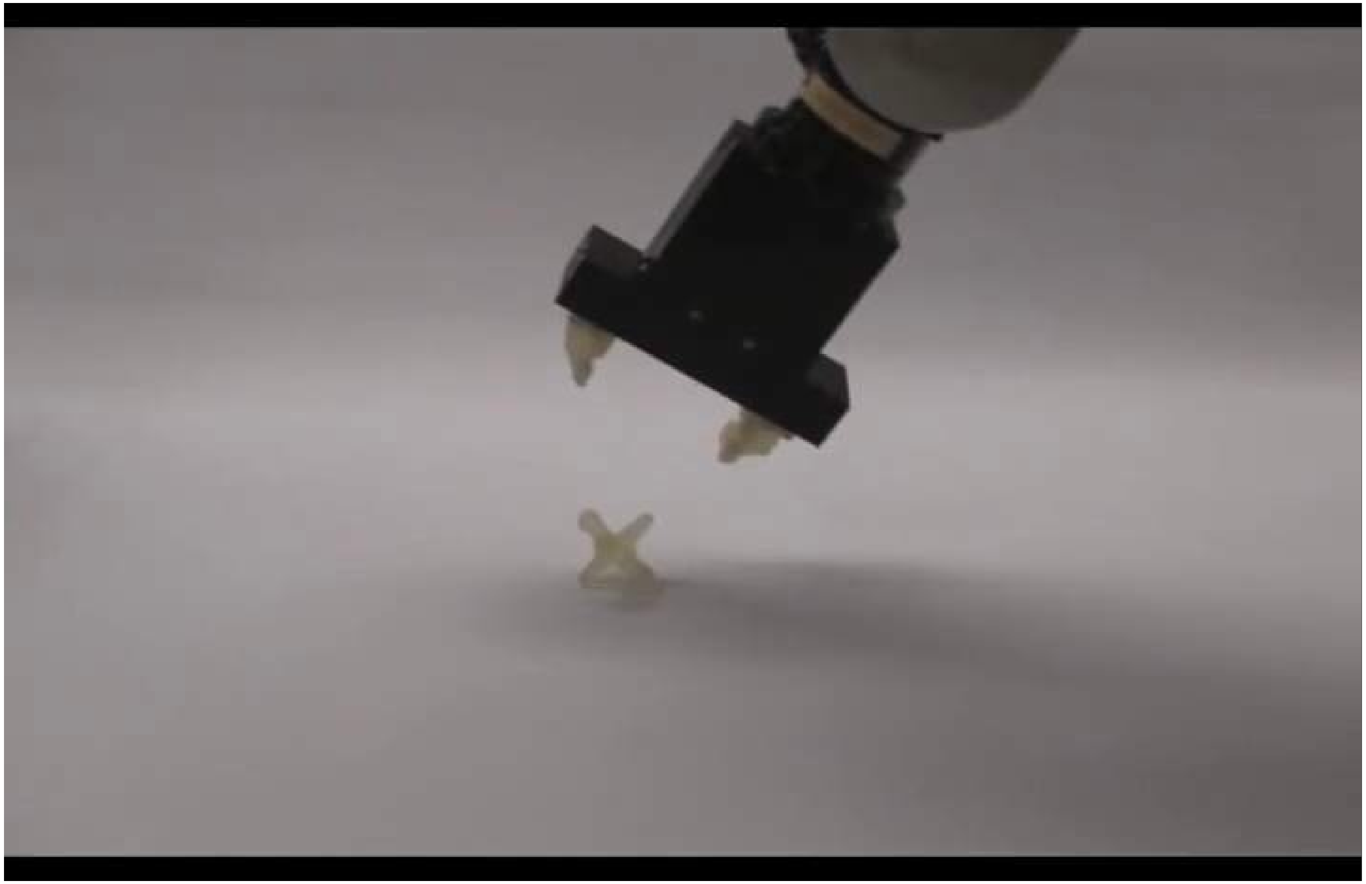
Industrial grasping

- Underactuated grippers

Industrial grasping

- Suction, magnets

Industrial grasping



credit: Cornell Creative Machines Lab