Robotics

Erwin M. Bakker| LIACS Media Lab

18-2 2020

Universiteit Leiden

Bij ons leer je de wereld kennen

Organization and Overview

Period: February 4th – April 28th 2020
Time: Tuesday 14.15 – 16.00
Place: LIACS, Room 407-409 (Workshops Room 302-304)
Lecturer: Dr Erwin M. Bakker (erwin@liacs.nl)
Assistant: Laduona Dai

NB E-mail your name and student number to erwin@liacs.nl

Schedule:

4-2	Introduction and Overview
17-2	No class
18-2	Locomotion and Inverse Kinematics & *Yetiborg Introduction*
25-2	*Robotics Sensors and Image Processing*
3-3	***Project Proposals*** *(presentation by students)*
10-3	**Yetiborg Qualification Challenge**
17-3	Robotics Image Processing and Understanding
24-3	***Yetiborg Race***
31-3	**Project Progress Report** (by students)
7-4	Robotics Reinforcement Learning
14-4	**Robotics Reinforcement Learning Workshop II**
21-4	TBA
28-4	***Project Demos*** *(by students)*

Website: http://liacs.leidenuniv.nl/~bakkerem2/robotics/

Grading (6 ECTS):

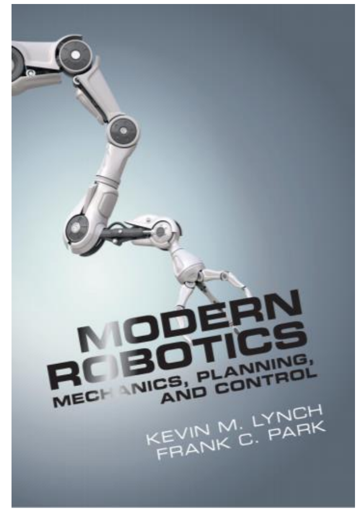
- Presentations and Robotics Project (60% of grade).
- Class discussions, attendance, workshops and assignments (40% of grade).
- It is necessary to be at every class and to complete every workshop.

Universiteit Leiden. Bij ons leer je de wereld kennen

Overview

- Robotic Actuators
- Configuration Space
- Rigid Body Motion
- Forward Kinematics
- Inverse Kinematics

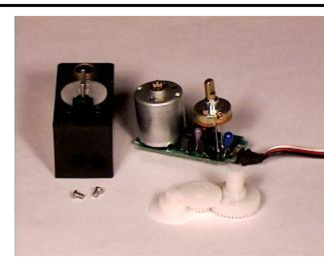
- Link: http://modernrobotics.org

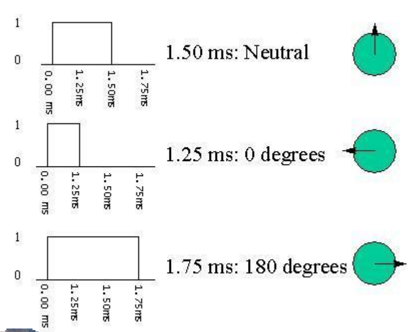


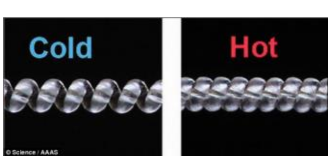
K.M. Lynch, F.C. Park, Modern Robotics: Mechanics, Planning and Control, Cambridge University Press, 2017

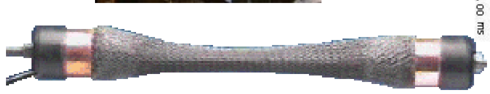
Robotics Actuators

- Electro motors
- Servo’s
- Stepper Motors
- Brushless motors
- Solenoids
- Hydraulic, pneumatic actuator’s
- Magnetic actuators
- Artificial Muscles
- Etc.



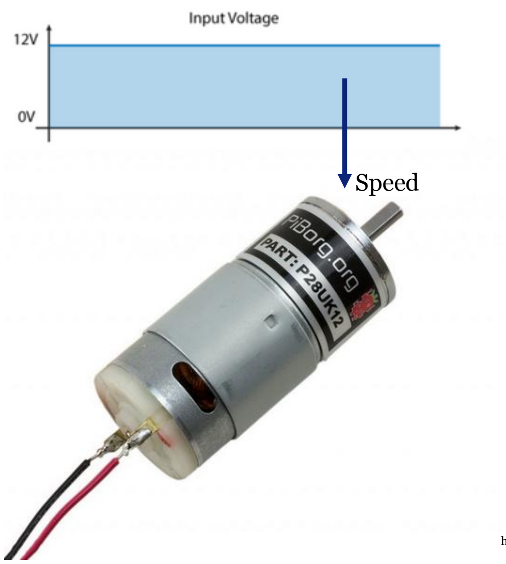


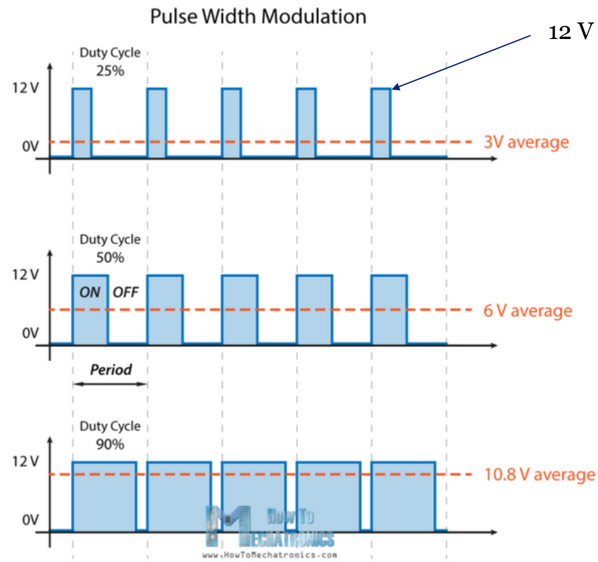




Universiteit Leiden. Bij ons leer je de wereld kennen

DC Motors

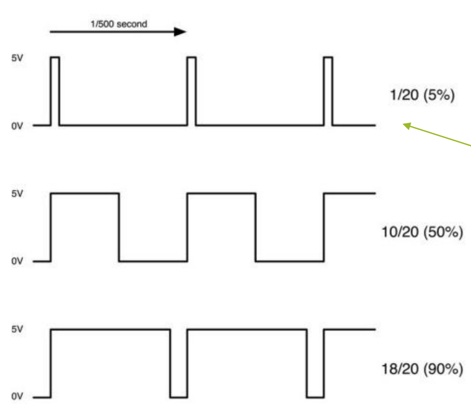


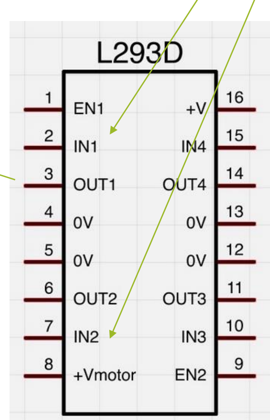


https://howtomechatronics.com/how-it-works/electronics/how-to-make-pwm-dc-motor-speed-controller-using-555-timer-ic/

DC Electro Motors:

- Duty Cycle



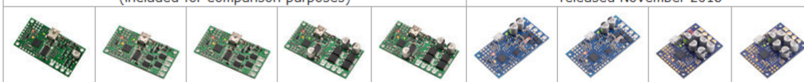



```
import RPi.GPIO as io
io.setmode(io.BCM)

in1_pin = 4
in2_pin = 17

io.setup(in1_pin, io.OUT)
io.setup(in2_pin, io.OUT)

def set(property, value):
    try:
        f = open("/sys/class/rpi-pwm/pwm0/" + property, 'w')
        f.write(value)
        f.close()
    except:
        print("Error writing to: " + property + " value: " + value)

set("delayed", "0")
set("mode", "pwm")
set("frequency", "500")
set("active", "1")

def clockwise():
    io.output(in1_pin, True)
    io.output(in2_pin, False)

def counter_clockwise():
    io.output(in1_pin, False)
    io.output(in2_pin, True)

clockwise()

while True:
    cmd = raw_input("Command, f/r 0..9, E.g. f5 :")
    direction = cmd[0]
    if direction == "f":
        clockwise()
    else:
        counter_clockwise()
    speed = int(cmd[1]) * 11
    set("duty", str(speed))
```

DC Motor Controllers

Pololu Simple Motor Controllers

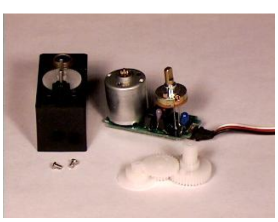
- USB, TTL Serial, Analog, RC Control, I2C



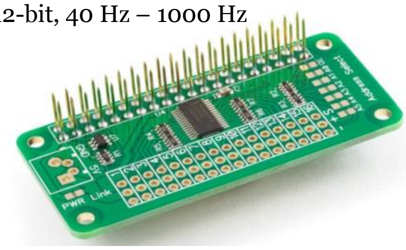
	Original versions, not recommended for new designs (included for comparison purposes)					G2 versions, released November 2018			
	SMC 18v7	SMC 18v15	SMC 24v12	SMC 18v25	SMC 24v23	SMC G2 18v15	SMC G2 24v12	SMC G2 18v25	SMC G2 24v19
Minimum operating voltage:	5.5 V	5.5 V	5.5 V	5.5 V	5.5 V	6.5 V	6.5 V	6.5 V	6.5 V
Recommended max operating voltage:	24 V(1)	24 V(1)	34 V(2)	24 V(1)	34 V(2)	24 V(1)	34 V(2)	24 V(1)	34 V(2)
Max nominal battery voltage:	18 V	18 V	28 V	18 V	28 V	18 V	28 V	18 V	28 V
Max continuous current (no additional cooling):	7 A	15 A	12 A	25 A	23 A	15 A	12 A	25 A	19 A
USB, TTL serial, Analog, RC control:	✓	✓	✓	✓	✓	✓	✓	✓	✓
I²C control:						✓	✓	✓	✓
Hardware current limiting:						✓	✓	✓	✓
Reverse voltage protection:						✓	✓	✓	✓

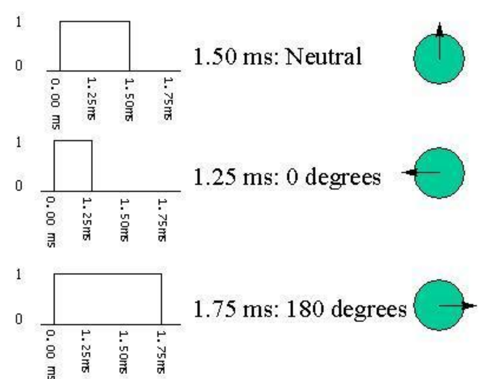
https://www.pololu.com/category/94/pololu-simple-motor-controllers

Servo’s

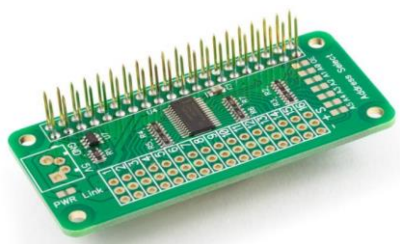


16-ch, 12-bit, 40 Hz – 1000 Hz

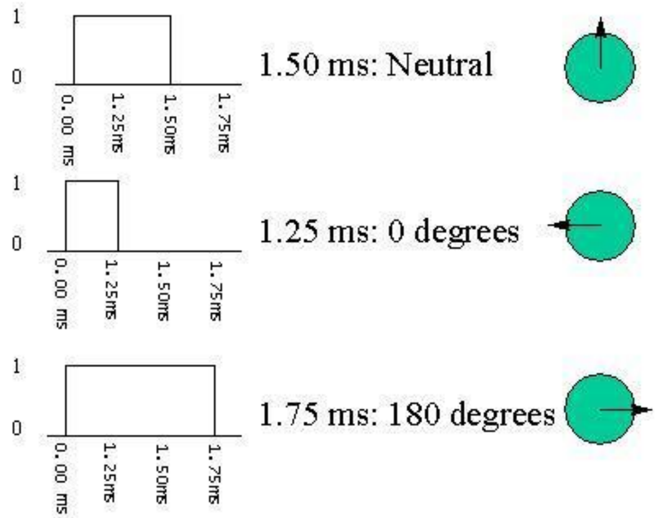




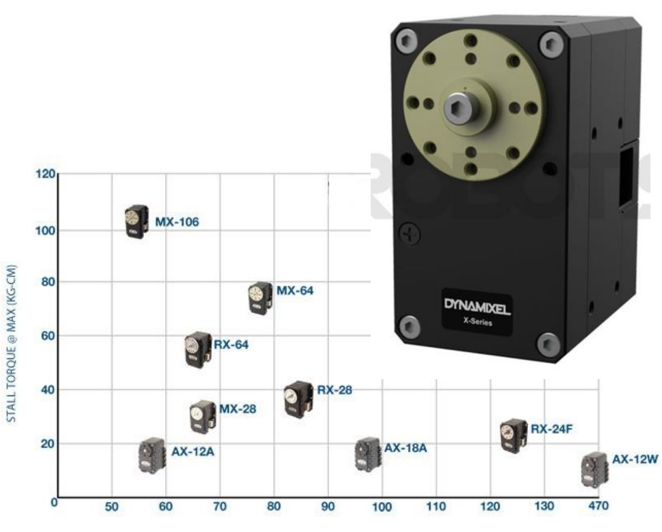
Servo's

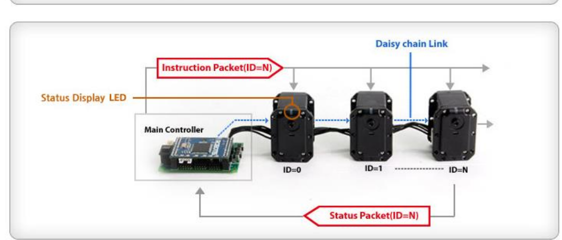


16-ch, 12-bit, 40 Hz – 1000 Hz



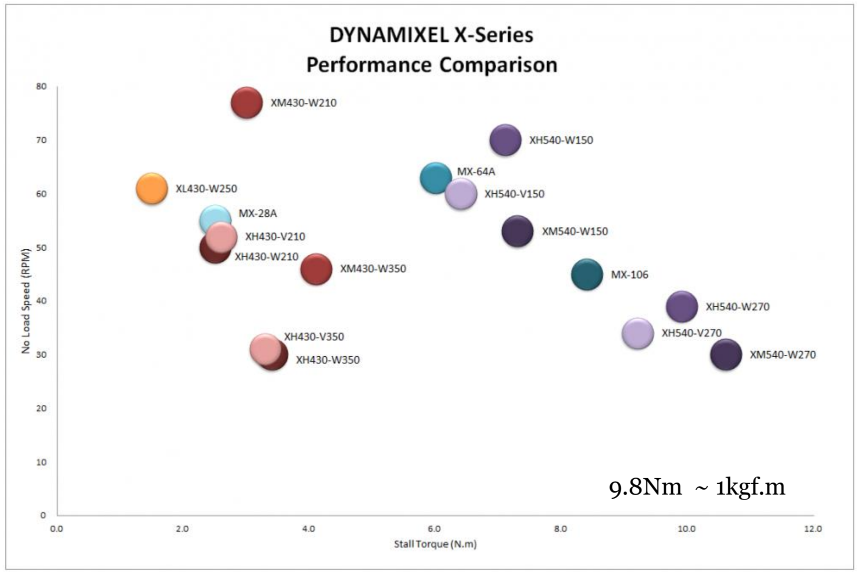
Dynamixel Servo's





Servo’s

Performance Comparison

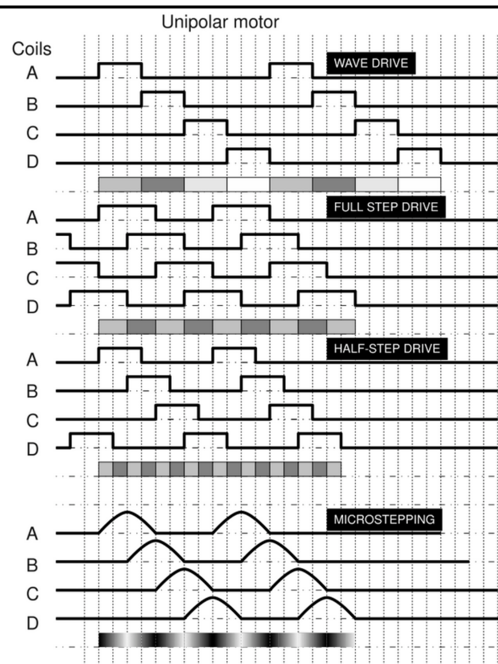


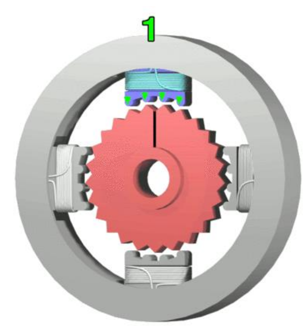
Universiteit Leiden. Bij ons leer je de wereld kennen

Robotics Actuators

- Stepper motors
- Drivers: low-level, high level

1
2
3
4

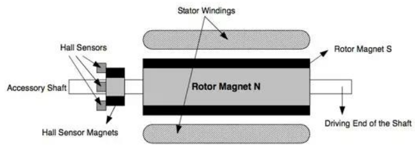


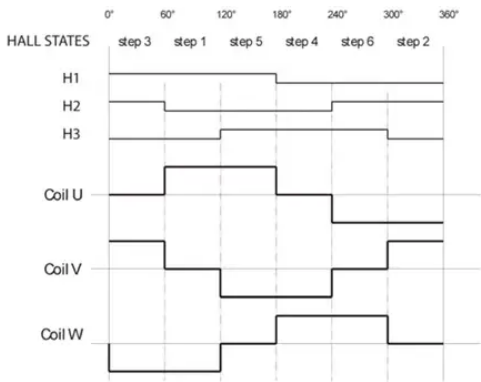


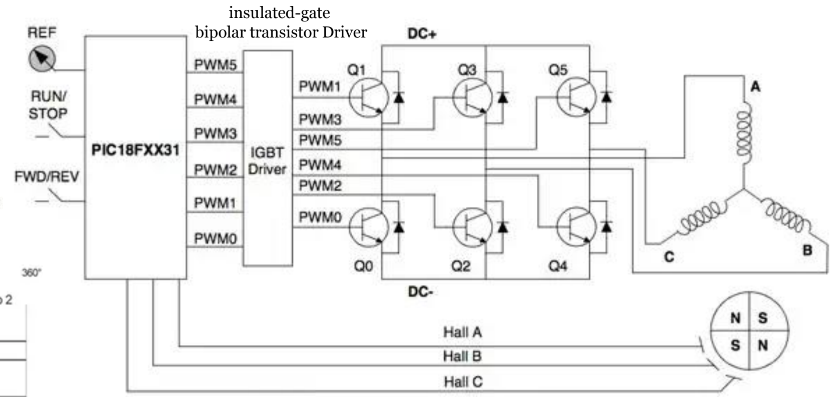
By Wapcaplet; Teravolt. (Wikipedia)

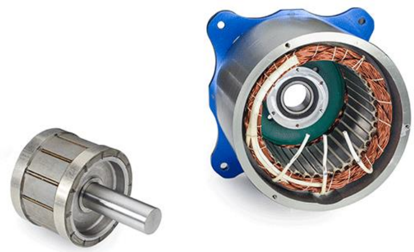
Universiteit Leiden. Bij ons leer je de wereld kennen

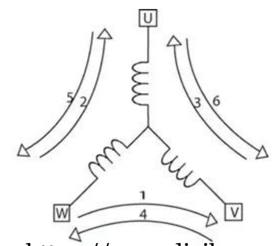
Brushless Motors





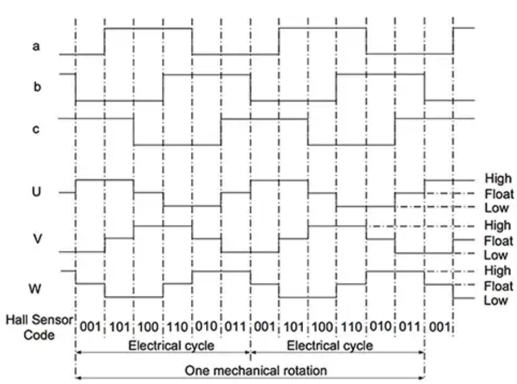


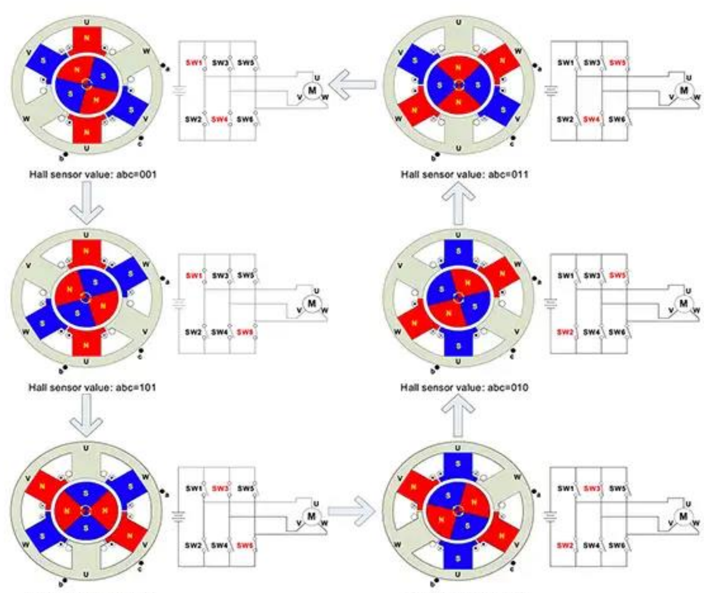




https://www.digikey.com

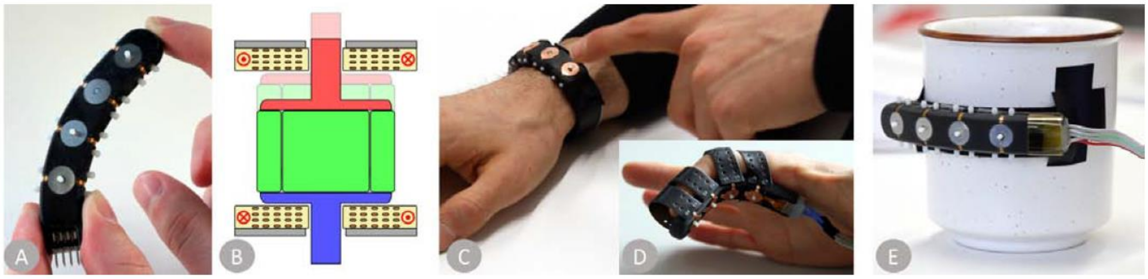
Brushless Motors

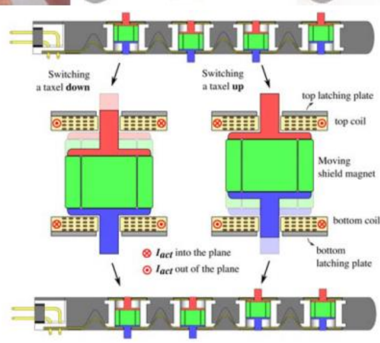




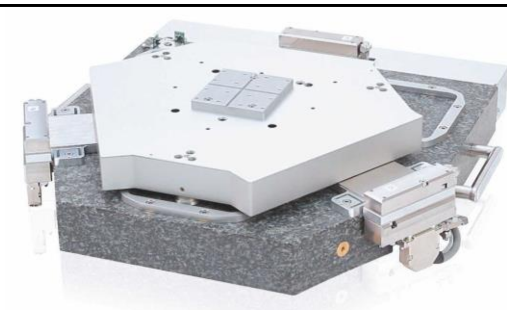
https://www.digikey.com/en/articles/techzone/2016/dec/how-to-power-and-control-brushless-dc-motors

Flexible Magnetic Actuators



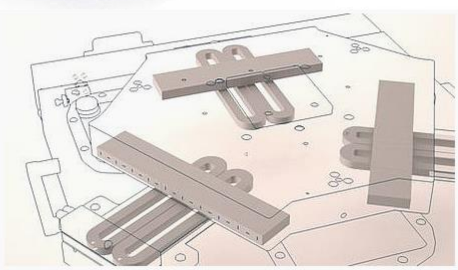


F. Pece et al., MagTics: Flexible and Thin Form Factor Magnetic Actuators for Dynamic and Wearable Haptic Feedback, UIST 2017, Oct. 22–25, 2017, Québec City, Canada

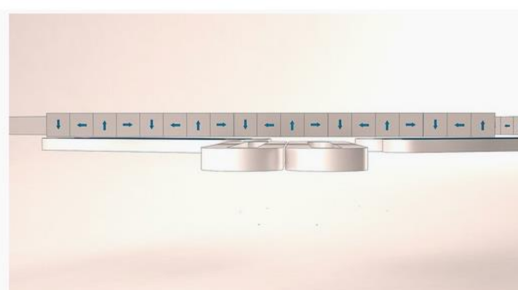


Robotics Actuators

- 6D Magnetic Control
- https://www.pi-usa.us
- pimag-6d-magnetic-levitation

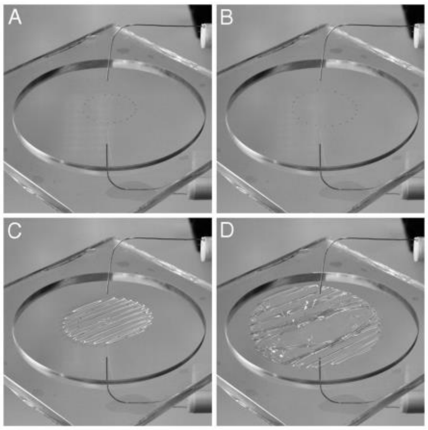


Simple structure: The platform levitates on a magnetic field generated by only six planar coils in the stator

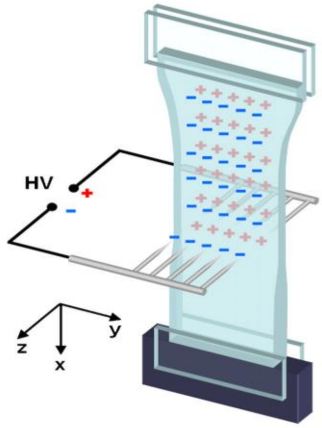


The Halbach arrangement of the magnets makes it possible, to minimize the energy required by the active coils in the stator for carrying the platform, to increase the load carrying capacity and to reduce thermal load

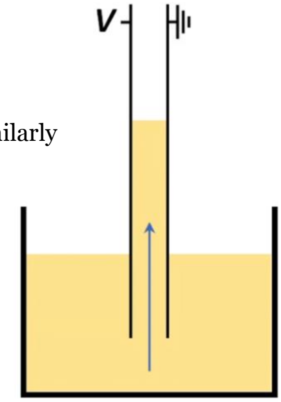
Artificial Muscles



0 – 25 kV



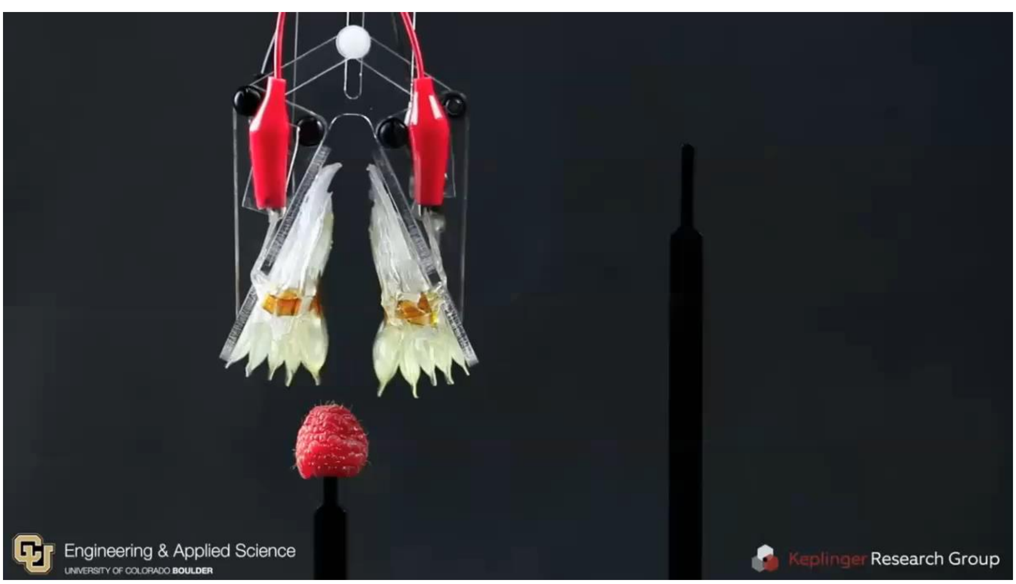
similarly



Maxwell Stress

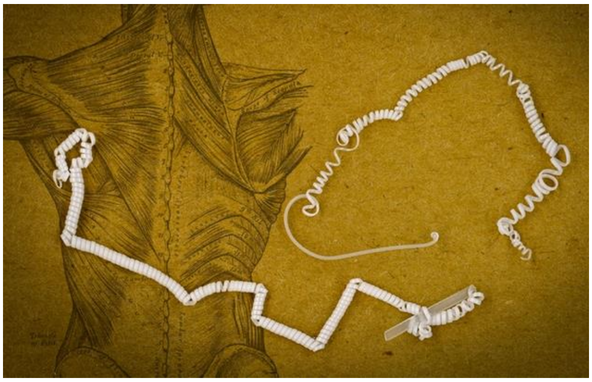
Röntgen's electrode-free elastomer actuators without electromechanical pull-in instability by C. Keplinger, et al. PNAS March 9, 2010 107 (10) 4505-4510; https://doi.org/10.1073/pnas.0913461107

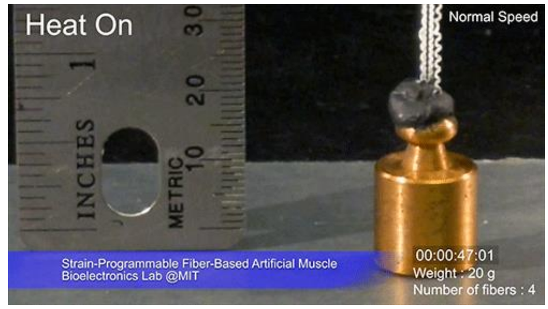
Röntgen WC (1880) Ueber die durch Electricität bewirkten Form—und Volumenänderungen von dielectrischen Körpern. Ann Phys Chem 11:771–786.

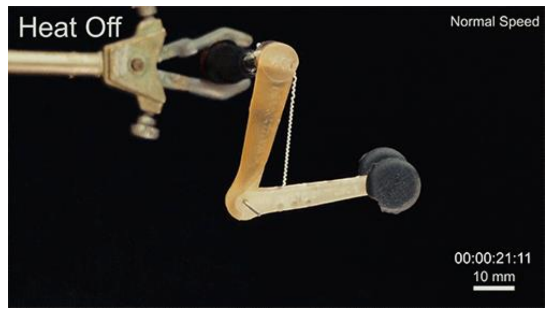


See also TED Talk **The artificial muscles that will power robots of the future by Christoph Keplinger** https://www.youtube.com/watch?v=ER15KmrB8h8

Universiteit Leiden. Bij ons leer je de wereld kennen







MIT Artificial Muscles

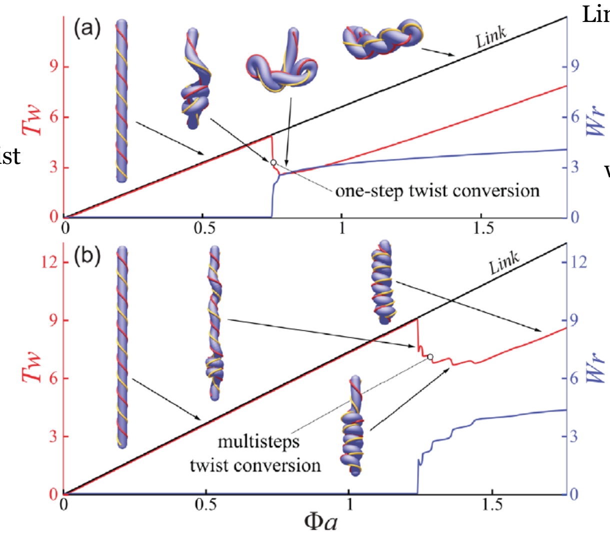
- Combination of two dissimilar polymers into a single fiber
- The polymers have very different thermal expansion coefficients (as in bimetals)
- Developed by Mehmet Kanik, Sirma Örgüç, working with Polina Anikeeva, Yoel Fink, Anantha Chandrakasan, and C. Cem Taşan, and five others

http://news.mit.edu/2019/artificial-fiber-muscles-0711

N. Charles, M. Gazzola, and L. Mahadevan, **Topology, Geometry, and Mechanics of Strongly Stretched and Twisted Filaments: Solenoids, Plectonemes, and Artificial Muscle Fibers**
PHYSICAL REVIEW LETTERS 123, 208003 (2019)

Link = Tw + Wr

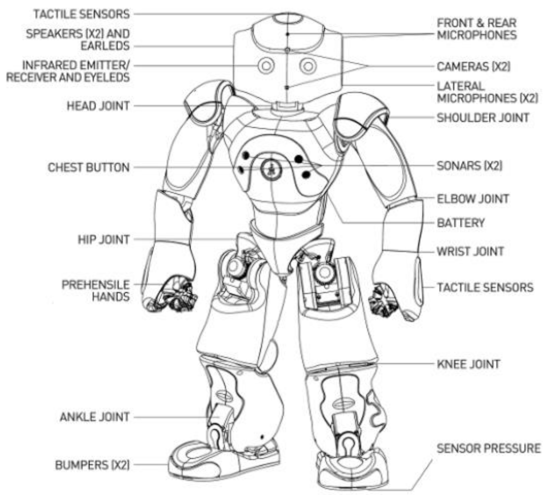
twist

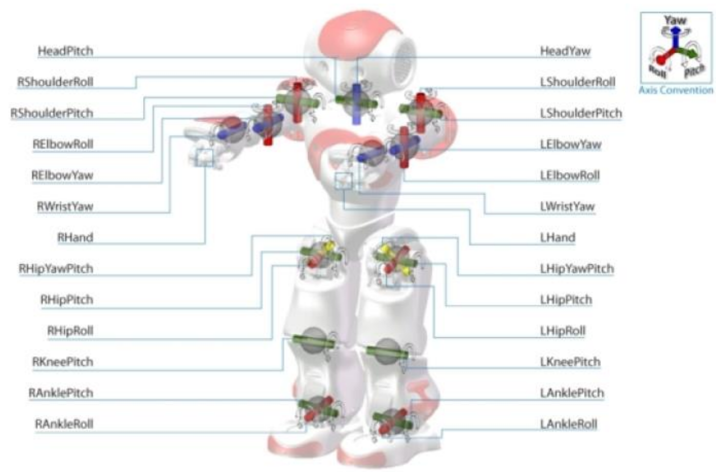


writhe

Twist density

NAO





http://doc.aldebaran.com/2-1/family/nao_dcm/actuator_sensor_names.html

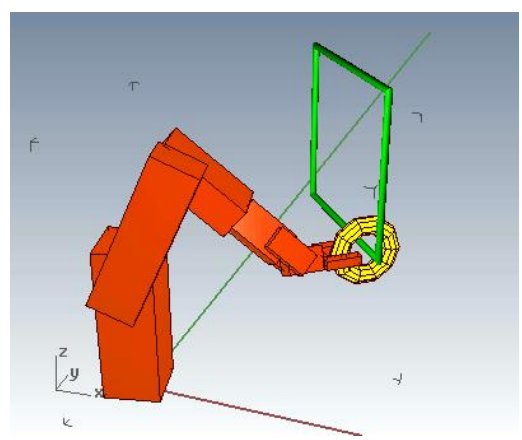
How to move to a goal?

Problem: How to move to a goal?

- Grasp, Walk, Stand, Dance, Follow, etc.

Solution:

1. *Program step by step*
   - *Computer Numerical Control (CNC), Automation.*
2. *Inverse kinematics*:
   - take end-points and move them to designated points.
3. *Tracing movements*
   - by specialist, human, etc.
4. ***Learn the right movements***
   - **Reinforcement Learning**, give a reward when the movement resembles the designated movement.



https://pybullet.org/wordpress/

Configuration Space

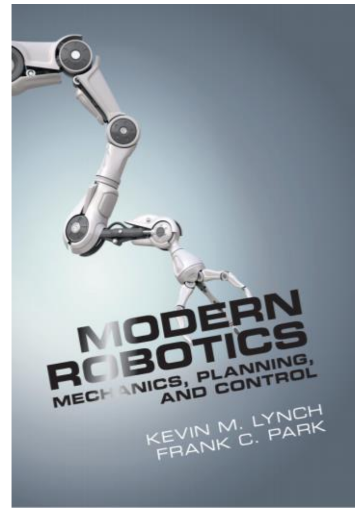
Robot Question: Where am I?

Answer:

The robot's configuration: a specification of the positions of all points of a robot.

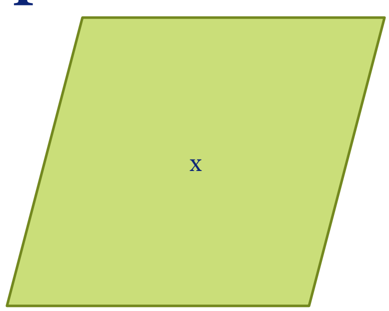
Here we assume:

Robot links and bodies are rigid and of known shape => only a few variables needed to describe it's configuration.



K.M. Lynch, F.C. Park, Modern Robotics: Mechanics, Planning and Control, Cambridge University Press, 2017

Configuration Space

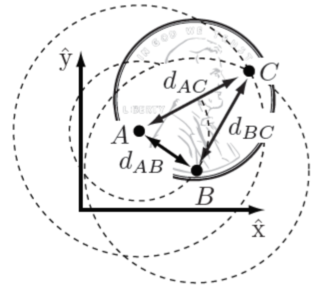


- Degrees of Freedom of a Rigid Body: the smallest number of real-valued coordinates needed to represent its configuration

Universiteit Leiden. Bij ons leer je de wereld kennen

Configuration Space

Assume we have a coin (heads) with 3 points A, B, C on it.



In the plane A,B,C have 6 degrees of freedom:
(x_A,y_A) , (x_B,y_B) , (x_C,y_C)
A coin is rigid => 3 extra constraints on distances: d_{AB}, d_{AC}, d_{BC}
are fixed, wherever the location of the coin would be.

1. The coin and hence A can be placed everywhere => (x_A,y_A) free to choose.
2. B can only be placed under the constraint that its distance to A would be equal to d_{AB}. =>
   freedom to turn the coin around A with angle φ_{AB} => (x_A, y_A, φ_{AB}) are free to choose.
3. C should be placed at distance d_{AC}, d_{BC} from A and B, respectively => only 1 possibility, hence no degree of freedom added.

Degrees of Freedom (DOF) of a Coin
= sum of freedoms of the points – number of independent constraints
= number of variables – number of independent equations = 6 – 3 = 3

Configuration Space

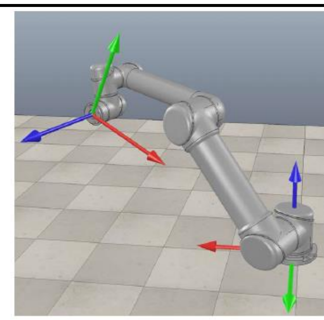
[1] Definition 2.1.

The **configuration** of a robot is a complete specification of the position of every point of the robot.

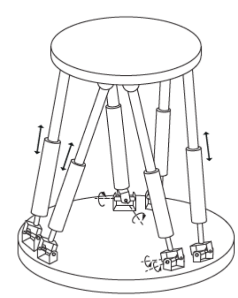
The minimum number n of real-valued coordinates needed to represent the configuration is the number of **degrees of freedom (dof)** of the robot.

The n-dimensional space containing all possible configurations of the robot is called the **Configuration Space (C-space)**.

The configuration of a robot is represented by a point in its C-space.



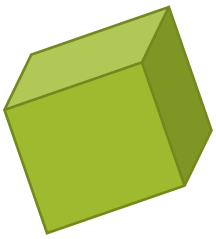
Open-chain robot: Manipulator (in V-REP). [1]

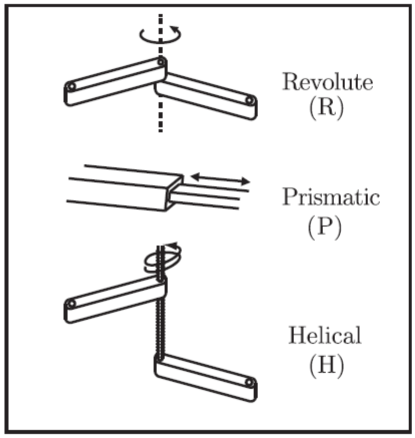


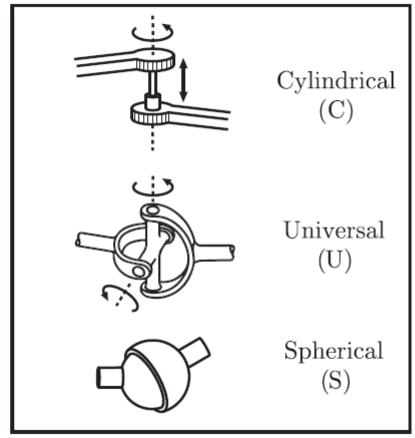
Closed-chain robot: Stewart-Gough platform. [1]

Degrees of Freedom of a Robot

- A rigid body in 3D Space has **6 DOF**



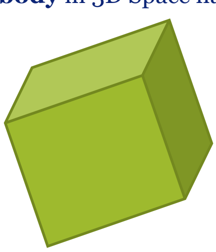


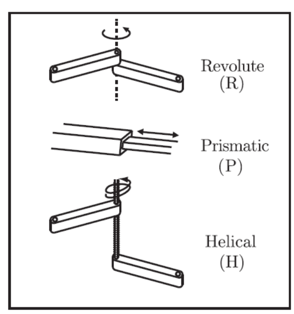


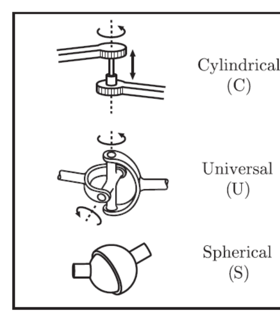
- A joint can be seen to put constraints on the rigid bodies it connects
- It also allows freedom to move relative to the body it is attached to.

Degrees of Freedom of a Robot

- A **rigid body** in 3D Space has 6 DOF





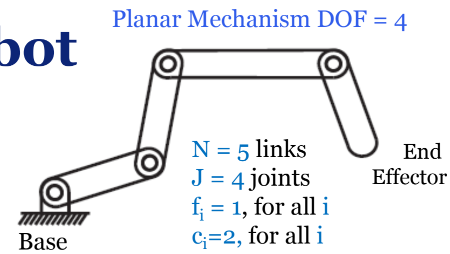


Joint type	dof f	Constraints c between two planar rigid bodies	Constraints c between two spatial rigid bodies
Revolute (R)	1	2	5
Prismatic (P)	1	2	5
Helical (H)	1	N/A	5
Cylindrical (C)	2	N/A	4
Universal (U)	2	N/A	4
Spherical (S)	3	N/A	3

- A **joint** can be seen to put constraints on the rigid bodies it connects
- It also allows freedom to move relative to the body it is attached to.

Universiteit Leiden. Bij ons leer je de wereld kennen

Degrees of Freedom of a Robot



Proposition (Grübler's formula)

Consider a mechanism consisting of

- N links, where ground is also regarded as a link.
- J number of joints,
- m number of degrees of freedom of a rigid body (m = 3 for planar mechanisms and m = 6 for spatial mechanisms),
- f_i the number of freedoms provided by joint i, and
- c_i the number of constraints provided by joint i, where $f_i + c_i = m$ for all i.

Then *Grübler's formula* for the number of degrees of freedom of the robot is

$$dof = m(N-1) - \sum_{i=1}^{J} c_i = m(N-1-J) + \sum_{i=1}^{J} f_i$$

This formula holds only if all joint constraints are independent. If they are not independent then the formula provides a lower bound on the number of degrees of freedom.

Links
1
3
3
6
3
1

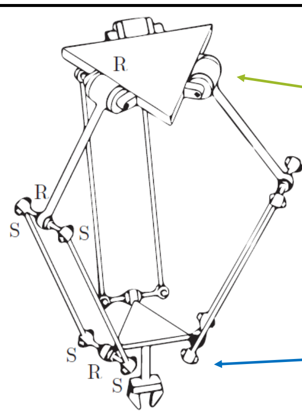
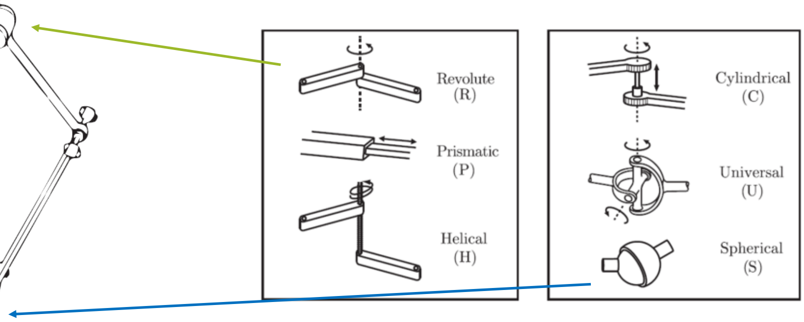


Figure 2.8: The Delta robot.



Example 2.7 (Delta robot). The Delta robot of Figure 2.8 consists of two platforms – the lower one mobile, the upper one stationary – connected by three legs. Each leg contains a parallelogram closed chain and consists of three revolute joints, four spherical joints, and five links. Adding the two platforms, there are $N = 17$ links and $J = 21$ joints (nine revolute and 12 spherical). By Grübler's formula,

$$\boxed{\text{dof} = 6(17 - 1 - 21) + 9(1) + 12(3) = 15.}$$

- Links: 1 + 3 + 3 + 6 + 3 + 1 = 17
- Joints: 21: 9x R(1 dof) and 12 x S(3 dof)
- m= 6

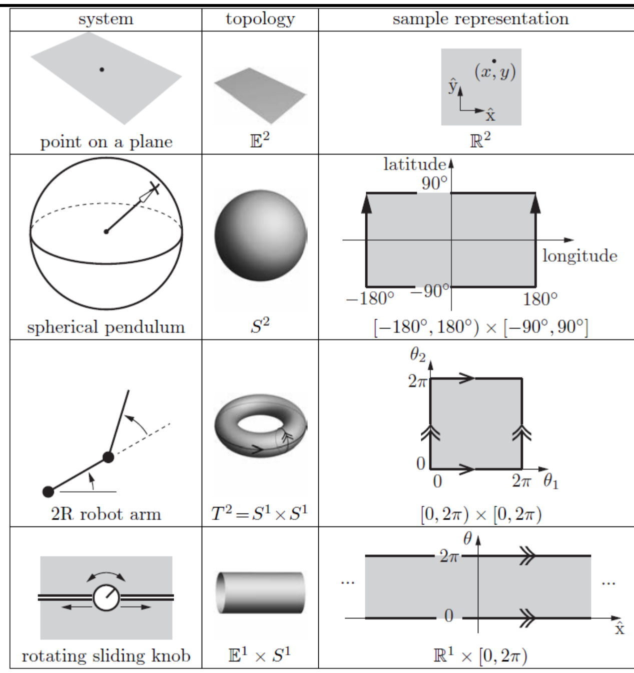
Systems
and their
Topologies

Note: $S^1 \times S^1 = T^2$ (not S^2)

Coordinates

Explicit Coordinates
- Euclidean (x,y)
- Polar (r,φ)
- Combined (x,y) x (r, φ)

Implicit Coordinates
- { (x,y,z) | $x^2+y^2+z^2=1$ }



C-Space (Configuration Space)

How to describe a rigid body's position and orientation in C-Space?

Fixed reference frame {s}

Reference fame attached to body {b}

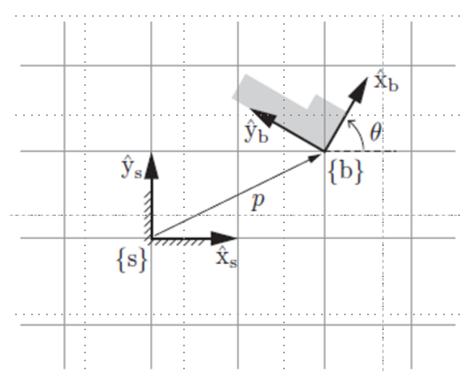
Can be described by a 4x4 matrix with 10 constraints

(constraints, e.g.: unit-length, orthogonal)

Note: a point in $\mathbb{R}^3 x S^2 x S^1$

Matrix can be used to:

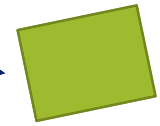
1. Translate or rotate a vector or a frame
2. Change the representation of a vector or a frame
   - for example from relative to {s} to relative to {b}

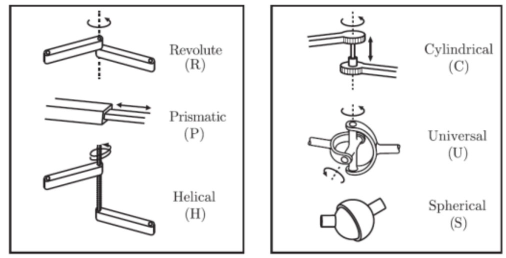


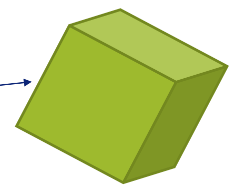
in the plane

C-Spaces

- The C-space of a rigid body in the plane can be written as $\boxed{\mathbb{R}^2 \times S^1}$, since the configuration can be represented as the concatenation of the coordinates (x, y) representing $\mathbb{R}^2$ and an angle θ representing S^1.
- The C-space of a PR robot arm can be written $\boxed{\mathbb{R}^1 \times S^1}$. (We will occasionally ignore joint limits, i.e., bounds on the travel of the joints, when expressing the topology of the C-space; with joint limits, the C-space is the Cartesian product of two closed intervals of the line.)
- The C-space of a 2R robot arm can be written $\boxed{S^1 \times S^1 = T^2}$, where T^n is the n-dimensional surface of a torus in an $(n+1)$-dimensional space. (See Table 2.2.) Note that $S^1 \times S^1 \times \cdots \times S^1$ (n copies of S^1) is equal to T^n, not S^n; for example, a sphere S^2 is not topologically equivalent to a torus T^2.
- The C-space of a planar rigid body (e.g., the chassis of a mobile robot) with a 2R robot arm can be written as $\mathbb{R}^2 \times S^1 \times T^2 = \boxed{\mathbb{R}^2 \times T^3}$.
- As we saw in Section 2.1 when we counted the degrees of freedom of a rigid body in three dimensions, the configuration of a rigid body can be described by a point in $\mathbb{R}^3$, plus a point on a two-dimensional sphere S^2, plus a point on a one-dimensional circle S^1, giving a total C-space of $\boxed{\mathbb{R}^3 \times S^2 \times S^1}$.







Task Space and Work Space

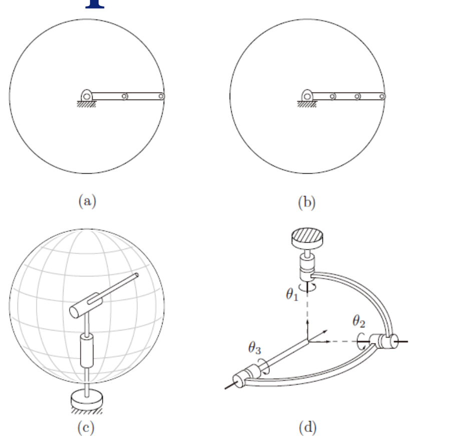
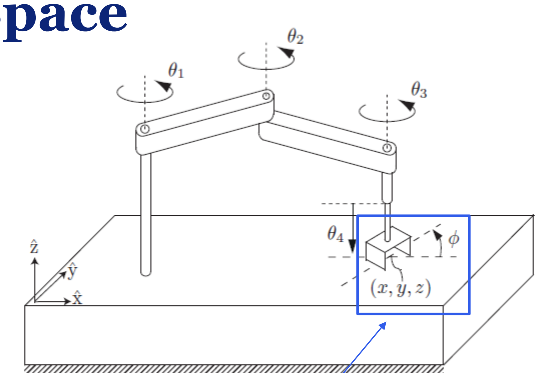


Figure 2.12: Examples of workspaces for various robots: (a) a planar 2R open chain; (b) a planar 3R open chain; (c) a spherical 2R open chain; (d) a 3R orienting mechanism.

The **workspace** is a specification of the configurations that the end-effector of the robot can reach.



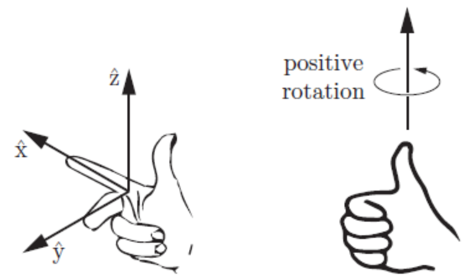
The SCARA robot is an RRRP open chain that is widely used for tabletop pick-and-place tasks. The end-effector configuration is completely described by (x, y, z, φ)

⇒ **task space** $R^3 \times S^1$ and

⇒ **workspace** as the reachable points in (x, y, z), since all orientations φ can be achieved at all reachable points.

Rigid Body Motion

Rigid-body position and orientation (x, y, z, φ, θ, ψ) ∈ $\mathbb{R}^3 x S^2 x S^1$



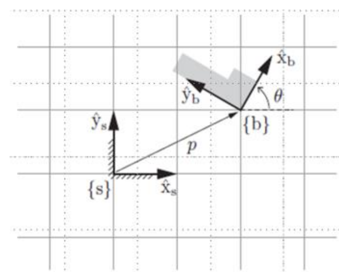
- Can also be described by 4x4 matrix with 10 constraints.
- In general 4x4 matrices can be used for
  - Location
  - Translation + rotation of a vector or frame
  - Transformation of coordinates between frames

- Velocity of a rigid body: $(\partial x/\partial t,\ \partial y/\partial t,\ \partial z/\partial t,\ \partial \phi/\partial t,\ \partial \theta/\partial t,\ \partial \psi/\partial t)$

Exponential coordinates:

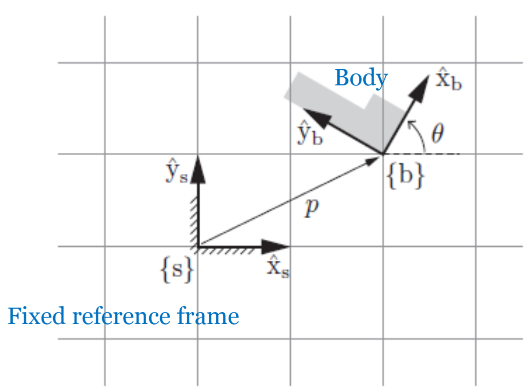
Every rigid-body configuration can be achieved by:

- Starting in the fixed home frame and integrating a constant twist for a specified time.
- Direction of a screw axis and scalar to indicate how far the screw axis must be followed



Similarly in the plane

Rigid Body Motions in the Plane



$$p = p_x \hat{x}_s + p_y \hat{y}_s.$$

$$\begin{aligned} \hat{x}_b &= \cos\theta\, \hat{x}_s + \sin\theta\, \hat{y}_s, \\ \hat{y}_b &= -\sin\theta\, \hat{x}_s + \cos\theta\, \hat{y}_s. \end{aligned}$$

Figure 3.3: The body frame {b} is expressed in the fixed-frame coordinates {s} by the vector p and the directions of the unit axes $\hat{x}_b$ and $\hat{y}_b$. In this example, $p = (2, 1)$ and $\theta = 60°$, so $\hat{x}_b = (\cos\theta, \sin\theta) = (0.5, 1/\sqrt{2})$ and $\hat{y}_b = (-\sin\theta, \cos\theta) = (-1/\sqrt{2}, 0.5)$.

Rigid Body Motions in the Plane

Previously:

$p = p_x \hat{x}_s + p_y \hat{y}_s.$

$$\begin{aligned} \hat{x}_b &= \cos\theta\, \hat{x}_s + \sin\theta\, \hat{y}_s, \\ \hat{y}_b &= -\sin\theta\, \hat{x}_s + \cos\theta\, \hat{y}_s. \end{aligned}$$

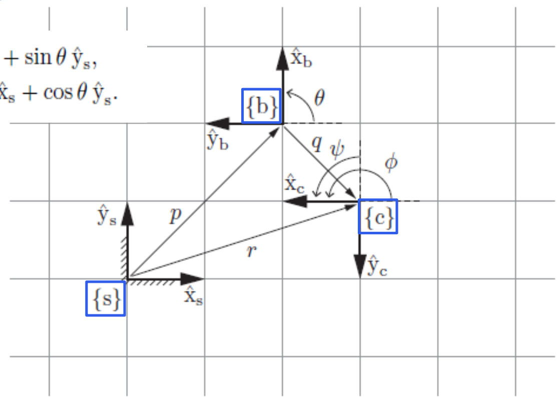


Figure 3.4: The frame {b} in {s} is given by (P, p), and the frame {c} in {b} is given by (Q, q). From these we can derive the frame {c} in {s}, described by (R, r). The numerical values of the vectors p, q, and r and the coordinate-axis directions of the three frames are evident from the grid of unit squares.

{b} relative to {s}

$$p = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

$$P = [\hat{x}_b \ \ \hat{y}_b] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

{c} relative to {s}

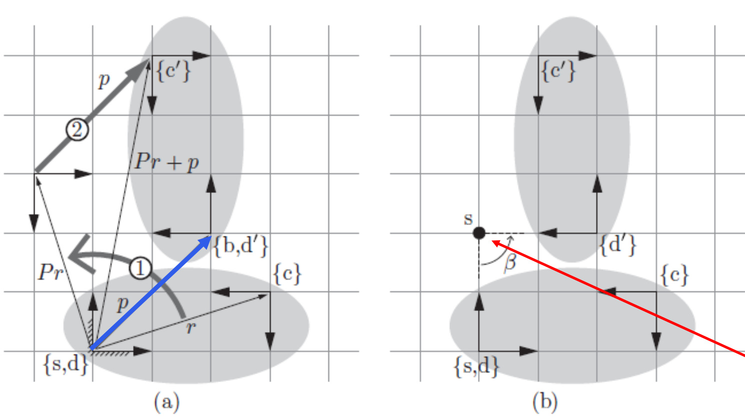
$$r = \begin{bmatrix} r_x \\ r_y \end{bmatrix}, \qquad R = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

{c} relative to {b}

$$q = \begin{bmatrix} q_x \\ q_y \end{bmatrix}, \qquad Q = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}$$

Note and verify: R = PQ, and r = Pq+p

Rigid Body Motions in the Plane



{c} described by (R,r)

$$r = \begin{bmatrix} r_x \\ r_y \end{bmatrix}, \qquad R = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

Move rigid body such that {d} coincides with {d’}.

$$p = \begin{bmatrix} p_x \\ p_y \end{bmatrix} \qquad P = [\hat{x}_b \ \ \hat{y}_b] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Then {c’} described by (R’,r’):

$$R' = PR,$$
$$r' = Pr + p,$$

Note: SCREW MOTION
The above rotation followed by a translation can also be expressed as a rotation of the rigid-body about a fixed point s by an angle β

(P, p) can be used to

1. Represent a configuration of a rigid body in {s}
2. Change the reference frame for vector representation.
3. **Displace a vector or a frame.**

Rigid Body Motions in the Plane

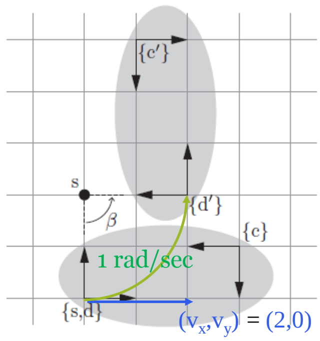
Note: SCREW MOTION
The above rotation followed by a translation can also be expressed as a rotation of the rigid-body about a fixed point s by an angle β

(β, s_x, s_y) , where $(s_x, s_y) = (0, 2)$

In the {s}-frame rotate 1 rad/sec with speed $(v_x, v_y) = (2, 0)$ is denoted as:
$S = (\omega, v_x, v_y) = (1, 2, 0)$

Following the screw-axis for an angle $\theta = \pi/2$ gives the displacement we want:
$S\theta = (\pi/2, \pi, 0)$

These are called the **exponential coordinates** for the planar rigid-body displacement.



{c} described by (R,r)

$$r = \begin{bmatrix} r_x \\ r_y \end{bmatrix}, \qquad R = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

Move rigid body such that {d} coincides with {d'}.

$$p = \begin{bmatrix} p_x \\ p_y \end{bmatrix} \qquad P = [\hat{x}_b \ \ \hat{y}_b] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Then {c'} described by (R',r'):

$$R' = PR,$$
$$r' = Pr + p,$$

Note:
- distance = vt
- distance along quarter circle with radius 2 equals π.

Forward Kinematics

The forward kinematics of 3R Planar Open Chain can be written as a product of four homogeneous transformation matrices: $T_{04} = T_{01}T_{12}T_{23}T_{34}$, where

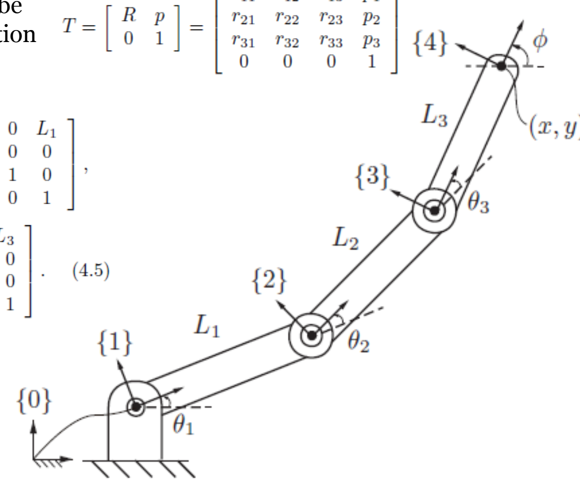
$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_1 \\ r_{21} & r_{22} & r_{23} & p_2 \\ r_{31} & r_{32} & r_{33} & p_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{01} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{12} = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_1 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{23} = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_2 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{34} = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (4.5)$$

Home position M:

$$M = \begin{bmatrix} 1 & 0 & 0 & L_1 + L_2 + L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Forward Kinematics: Product of Exponentials

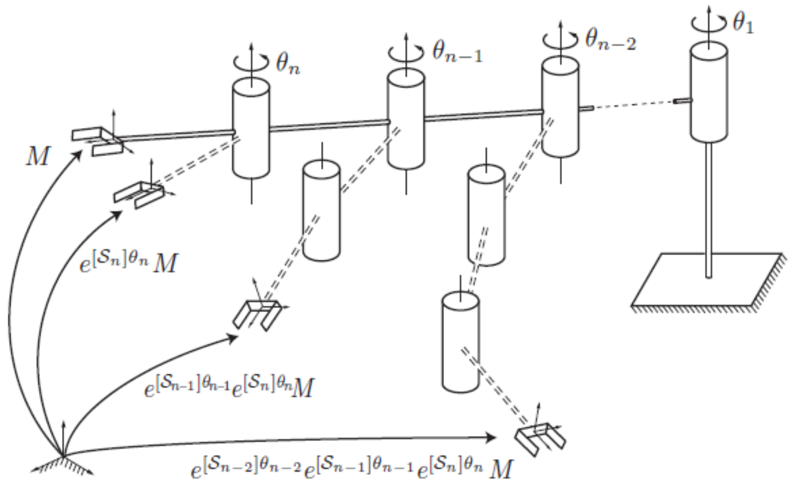


Figure 4.2: Illustration of the PoE formula for an n-link spatial open chain.

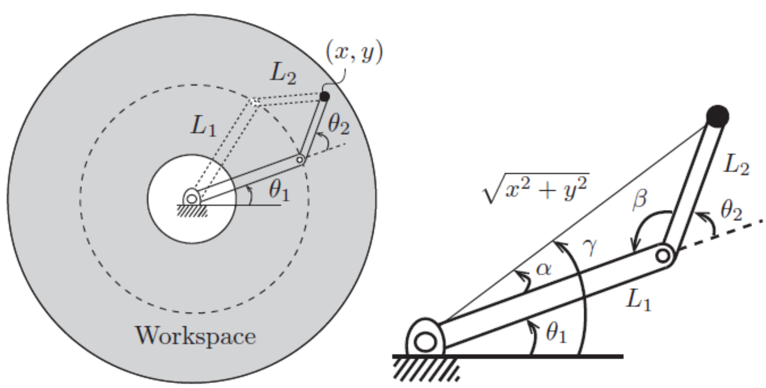
PoE parameters also known as Euler-Rodrigues parameters.

There are many other representations:
- for example Denavit-Hartenberg (1955) representation is very popular, but can be cumbersome

In velocity kinematics Jacobians are used.

Inverse Kinematics

Which angles θ_1, and θ_1 will lead to location (x,y)?



(a) A workspace, and lefty and righty configurations.

(b) Geometric solution.

Figure 6.1: Inverse kinematics of a 2R planar open chain.

Law of cosines gives:

$$L_1^2 + L_2^2 - 2L_1L_2\cos\beta = x^2 + y^2$$

, hence

$$\beta = \cos^{-1}\left(\frac{L_1^2 + L_2^2 - x^2 - y^2}{2L_1L_2}\right)$$

,and similarly

$$\alpha = \cos^{-1}\left(\frac{x^2 + y^2 + L_1^2 - L_2^2}{2L_1\sqrt{x^2+y^2}}\right)$$

γ= atan2(y,x)

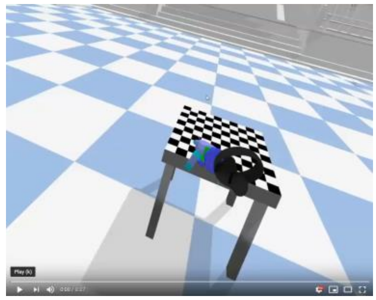
Answer:

$$\theta_1 = \gamma - \alpha, \qquad \theta_2 = \pi - \beta$$

In general: IK-Solvers, Newton-Raphson, etc.

Real Time Physics Modelling

https://pybullet.org/wordpress/



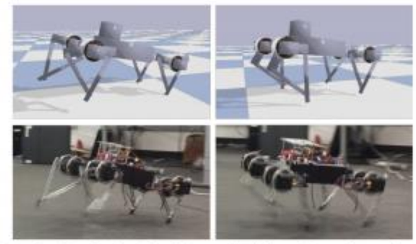
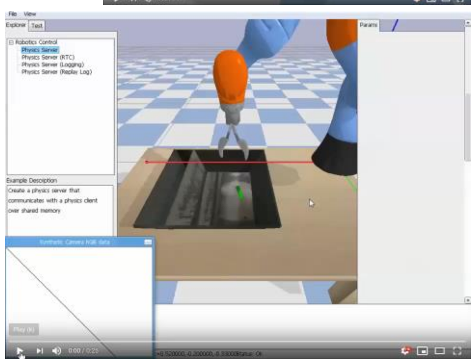
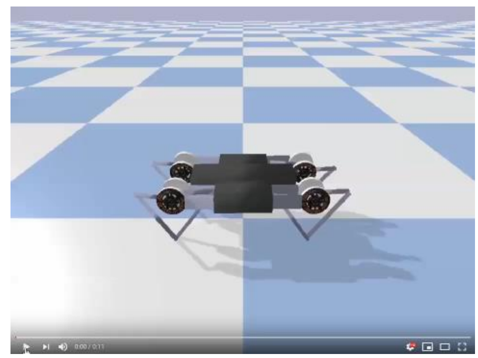


Fig. 1: The simulated and the real Minitaurs learned to gallop using deep reinforcement learning.

pybullet KUKA
grasp training

Using Tensorflow
OpenAI gym
Baselines
DeepQNetworks (DQNs)





Universiteit Leiden. Bij ons leer je de wereld kennen

Organization and Overview

Period: February 4th – April 28th 2020
Time: Tuesday 14.15 – 16.00
Place: LIACS, Room 407-409 (Workshops Room 302-304)
Lecturer: Dr Erwin M. Bakker (erwin@liacs.nl)
Assistant: Laduona Dai

NB E-mail your name and student number to erwin@liacs.nl

Schedule:

4-2	Introduction and Overview
17-2	No class
18-2	Locomotion and Inverse Kinematics & *Yetiborg Introduction*
25-2	*Robotics Sensors and Image Processing*
3-3	***Project Proposals** (presentation by students)*
10-3	**Yetiborg Qualification Challenge**
17-3	Robotics Image Processing and Understanding
24-3	***Yetiborg Race***
31-3	**Project Progress Report** (by students)
7-4	Robotics Reinforcement Learning
14-4	**Robotics Reinforcement Learning Workshop II**
21-4	TBA
28-4	***Project Demos** (by students)*

Website: http://liacs.leidenuniv.nl/~bakkerem2/robotics/

Grading (6 ECTS):
- Presentations and Robotics Project (60% of grade).
- Class discussions, attendance, workshops and assignments (40% of grade).
- It is necessary to be at every class and to complete every workshop.

Universiteit Leiden. Bij ons leer je de wereld kennen

Robotics Homework I

Assignment I:

Give a link to the coolest, strangest, most impressive, most novel, or technologically inspirational robot you could find.

Assignment II:

Visit http://modernrobotics.org and obtain the pdf of the book.

Read Chapters 1 and 2 and answer the following exercises:

- 2.2
- 2.9 for Figures 2.18 c, d, and f
- 2.18.

Due: Monday 24-2 at 14.00 PM.

Email your answers to erwin@liacs.nl with subject 'Robotics HW1'.

YetiBorg Racing Teams

1) Form YetiBorg Racing Teams of 4 people

Appoint one person who will be responsible for the robot.

Email your teams to erwin@liacs.nl with subject 'Robotics YetiBorg Racing Team'.

Due: Monday 24-2 at 14.00 PM.

References

1. K.M. Lynch, F.C. Park, Modern Robotics: Mechanics, Planning and Control, Cambridge University Press, 2017. (DOI: 10.1017/9781316661239)
2. https://pybullet.org/wordpress/