Robotics

An Introduction to the Field

Ahad Harati

A Glance at Contents

- History
- Manipulators as Industrial Robots
- Mobile Robots
- Means of Locomotion
- Wheeled Mobile Robots
- Legged Mechanisms
- Applications
- Research Directions

History of Robots

- 322 BC
  - Aristotle wrote “If every tool, when ordered, or even of its own accord, could do the work that befits it... then there would be no need either of apprentices for the master workers or of slaves for the lords.”

- 1921
  - Czech writer Karel Capek used the word ‘Robot’ derived from *robota* meaning boring and low level labor
    - Actually the word was invented by his brother, Josef.
    - In his play (Rossum's Universal Robots) he has a negative view point toward robots.

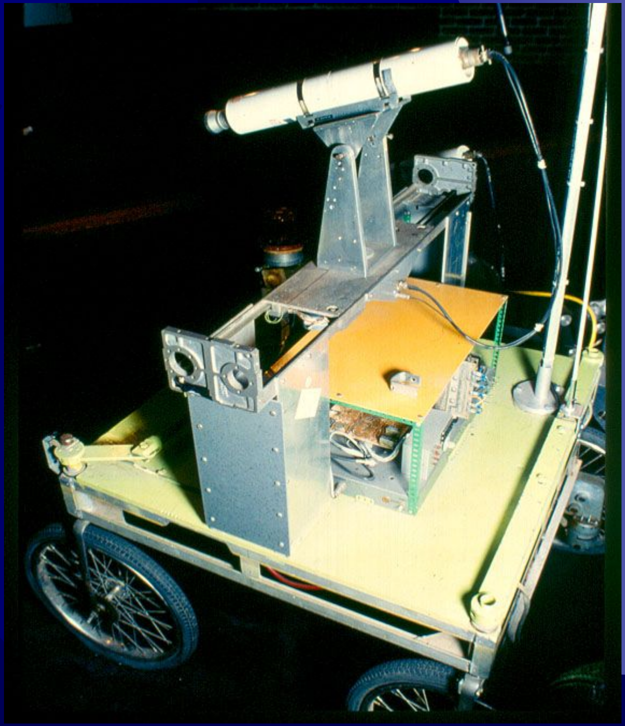
History of Robots

- In 1940s, Isaac Asimov presented a friendly picture of robots with three rules:
  - A robot never hurts a human being.
  - Robots obey human orders while they comply with the first rule.
  - Each robot should try to survive unless its efforts violates one of the two previous rules.
- Advent of real robots was in 1954 by George C. Devol in USA
  - He patented a programmable material handling machine which was able to record the desired motions and then repeat them (concept of teach-playback control)
  - In 1956, he established Unimation, the first robotic company.
- Finally in 1959 the first industrial robot was sold by Planet company.

More Recent Robotic History

- 1969
  - Victor Scheinman, a mechanical engineering student in the Stanford Artificial Intelligence Lab, created the Stanford Arm. The arm's design became a standard and is still influencing the design of robot arms today.

- 1970
  - Stanford University produced the Stanford Cart. It is designed to be a line follower but can also be controlled from a computer via radio link.



More Recent Robotic History

- 1972
  - Shigeo Hirose built the first snake robot; this became the first of many great Hirose robots
- 1974
  - Victor Scheinman formed his own company and started marketing the Silver Arm, which was capable of assembling small parts together using touch sensors – this leads to Adept Robots forming
- 1977
  - Deep space explorers Voyagers 1 and 2 were launched.
- 1978
  - PUMA robot is designed by Unimation.
- 1979
  - The Robotics Institute at Carnegie Mellon University was established.

What is a Robot?

- No commonly accepted definition
  - Sample definition: a robot is a software-controllable mechanical device that uses sensors to guide one or more end-effectors through programmed motions in a workspace in order to manipulate physical objects.

Three Main Stages of Robot Evolution

hi nose حارسی

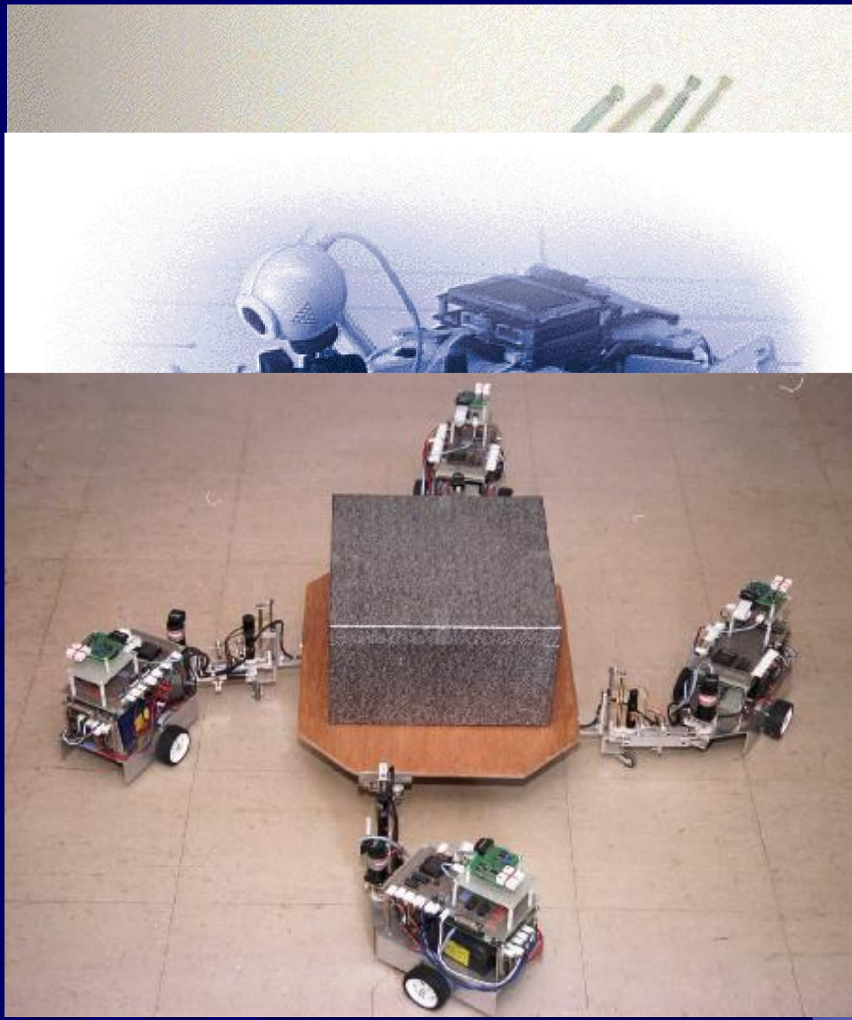
- Mechanism

electronic nose

- Sensing and Control

After 2000

- Intelligence

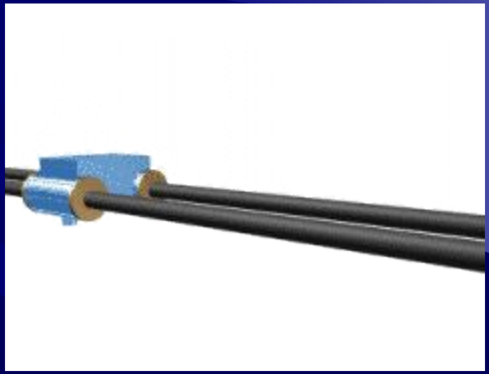


Why Robots?

- Continual production طور کلی دیده شد
- Higher production rate
- Lower costs
- Higher quality
  در کنار هم جوری سازماندهی شده که تلفیق تمام پروسه‌ها را جبران کنند یا تمام را کنترل کنند.
- Possibility of having integrated intelligent factory
- Flexible production lines — حفظ تولید برنامه‌ریزی گوشی موبایل
- Complementing human role and eliminating his weaknesses عدم جایگزینی انسان در کارهای خطرناک

Industrial Robots or manipulators

- Small working space
- Fixed base
- Structured environments
- Consist of Links and Joints
  - Revolute
  - Prismatic



Main Challenges of Manipulators

- Forward Kinematics → معادله جواب دارد → تعیین زاویه‌ها
- Inverse Kinematics → ممکن است جواب نداشته باشد ↔ redundancy
- Motion Planning → کنترل لینک‌ها ـ تعیین بهترین مسیر از نظر انرژی در حین حرکت مفاصل
  - Minimizing Energy
  - Obstacle Avoidance

نیروی استاتیک و کنترل (مثلا تماس با ...)

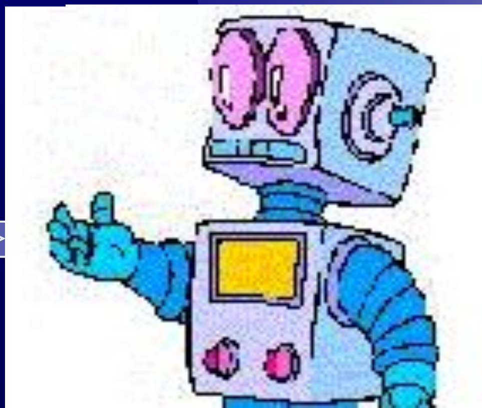
(مثل آچار یا مایع) نیروی در حین حرکت

- Force Control and Dynamics
- Multi Fingers and Grasp ناخن و انگشت نرم

How do I put my hand here?

Inverse Kinematics: Del

Forward Kinema



Why Mobile Robots?

- Achieving larger working area
  - As needed in many service applications
- Transportation اگر هدف حمل و نقل باشد
  - Possibility of using mobile systems by disabled people
- Exploration
- As a mean of interaction در جهت ارتباطات درست عمل شد
- Entertainment and Toys

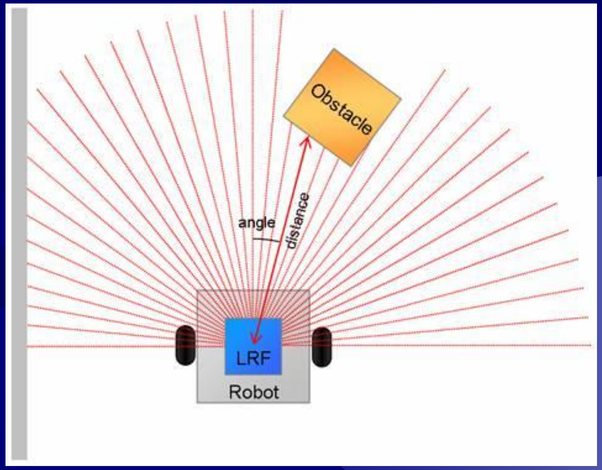
How to Move?

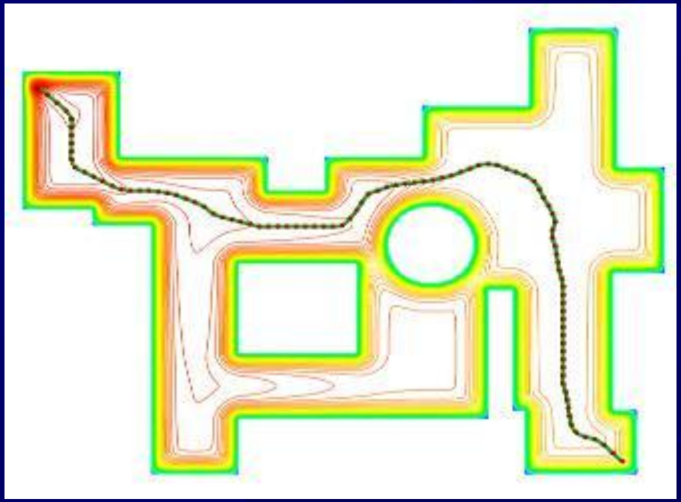
- از همه رایج تر
- Wheeled Robots
- Walking Robots
- Jumping Robots
- drone - دوربین رایج
- Flying Robots
- Swimming Robots
- Crawling Robots

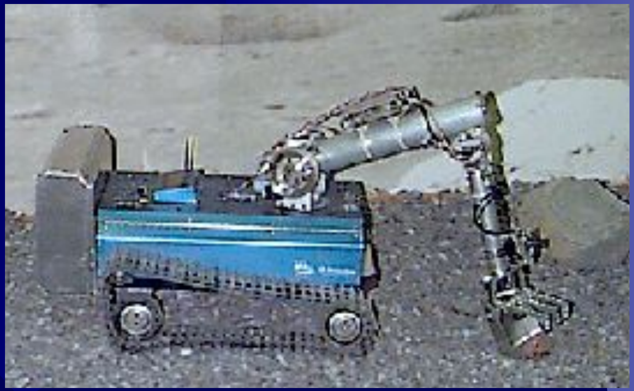
Common Challenges of Mobile Robots

دستورات نحوه حرکت

- Positioning and Localization
- Environment Detection تفسیر دور اطراف (مرحله تشخیص)
- Path Planning ربات مسیری نمایند و سرعت مقدار تحلیل
  با حرکت درست (از کجا به کجا)
- Dynamics and Control Boston dynamics
- Man-Machine Interface
  ارتباط با انسان







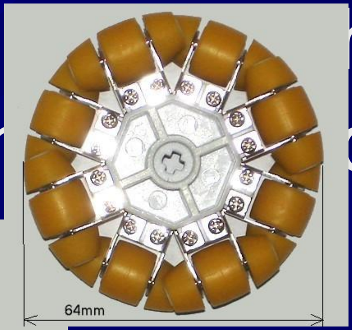
Wheeled Mobile Robots

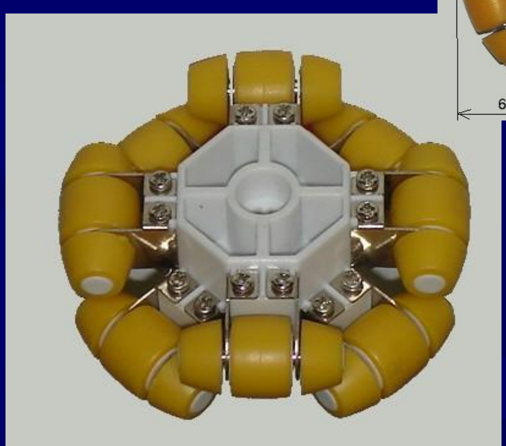
- High mobility حرکت دیار و راست
- Natural stability ایستایی و تعادل
- High load capacity نیرو به زمین منتقل می‌شود
  پس می‌تواند بار بیشتری تحمل کند.
- High speed

- Non-holonomic nature (mostly) مشکلات حرکت چرخ
- Restriction on roughness of the path هرجایی نمی‌رود و در محیط‌ها و اشکال
  متفاوت است.
- Need wide spaces (low maneuver)

Special Challenges of WMRs

- Mechanical Mechanism
  - Less restrictions on moving directions
  - Special mechanism for moving on rough terrains
- Require... hi...
  environm... ct...





تا مجبور نباشیم سراغش نمی‌رویم. در مرحله research قرار دارد.

Legged Mobile Robots

- High maneuvering
- Possibility of using on rough terrains
- Low efficiency
- Low payload
- Low speed
- Complicated mechanism and Control

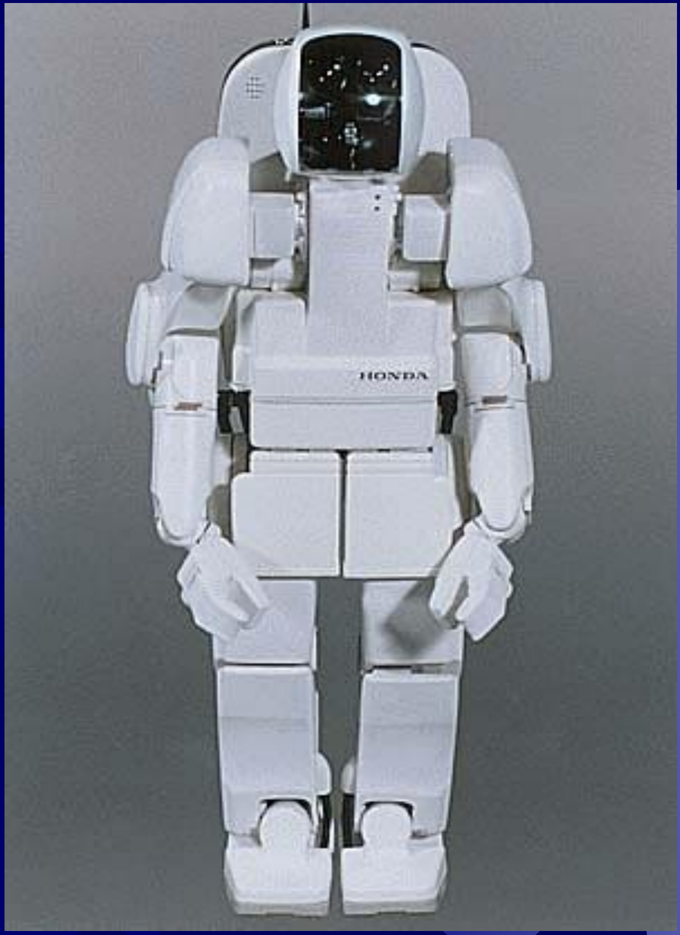
در حرکت هنوز چالش داریم.

Legged Mobile Robots

- Possibility of being used as Humanoid
  - Our world is engineered according to physics of humans
- Compact
- Inspiration from the nature
- Gait and stability

شکل تفاوت بین راه رفتن و دویدن فشار در ← نقطه حالت برای داشتن
سرعت بیشتر ذخیره گام برای داشتن نیاز دارد.

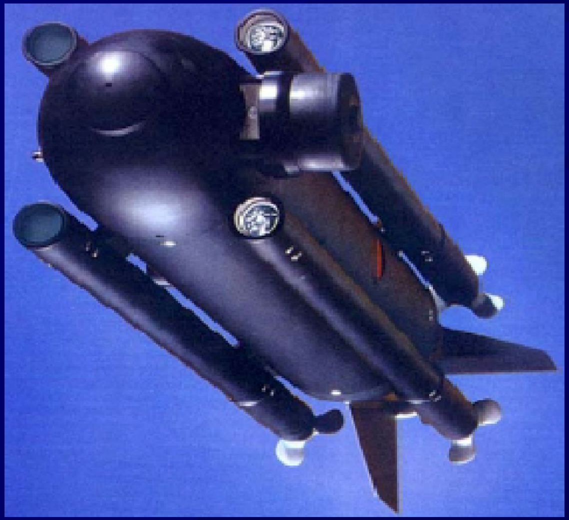
* مهمترین مبحث بخش انرژی است و هدف و مشکل اصلی بهینه سازی انرژی است.



۹ ماه و ۹ روز (بچه) ← تعادل انرژی و سازی

Jumping,Flying, and Swimming Robots

- High mobility
- Stability پایداری
- Control
- Special and complicated mechanism



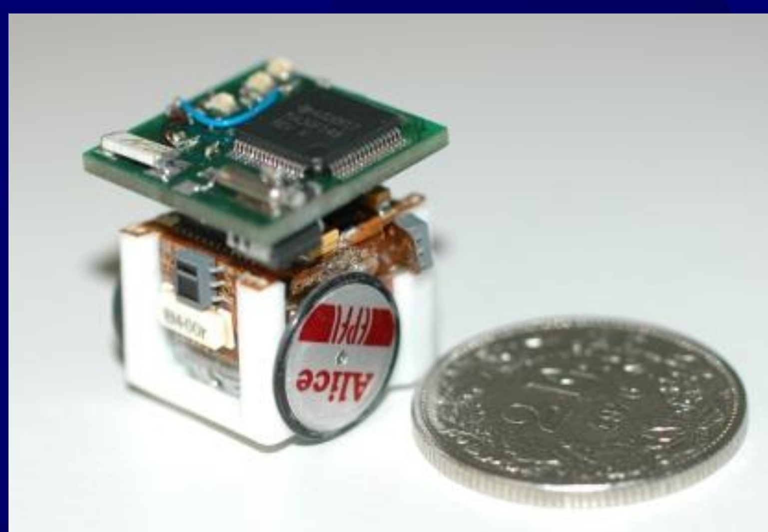
Growing Applications

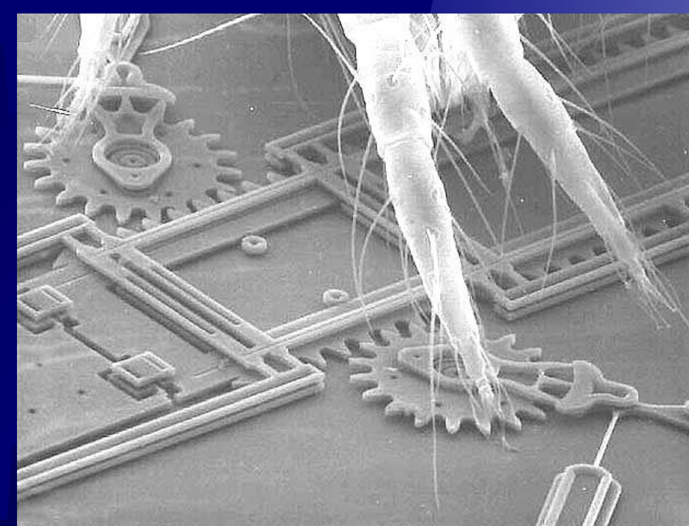
- Industrial → در بررسی ماشین‌ها، ربات‌ها و وسایل دیگر
  در محیط صنعتی - بخش تجارت
- Remote exploration
  مکان‌هایی که انسان‌ها نمی‌توانند بروند.
- Medical Tools
  کنترل از راه دور vs خود کنترل (مستقل) - پهپاد ROV
- Military Missions
- Home applications
  - ✔ Entertainment
  - ✔ Support disabled people and rehabilitation
  - ✔ Service applications

چالش مهم رباتیک: scale که در ارتباط با جنس انرژی و باتری مطرح است.

Research Directions

- Simple control
- Advanced sensor system (esp. vision)
- Intelligent behavior
- Interface
- Mini, Micro and Nano-robotics
  - Micro Electro Mechanical Systems (MEMS)





Common Manipulator Designs

Ahad Harati

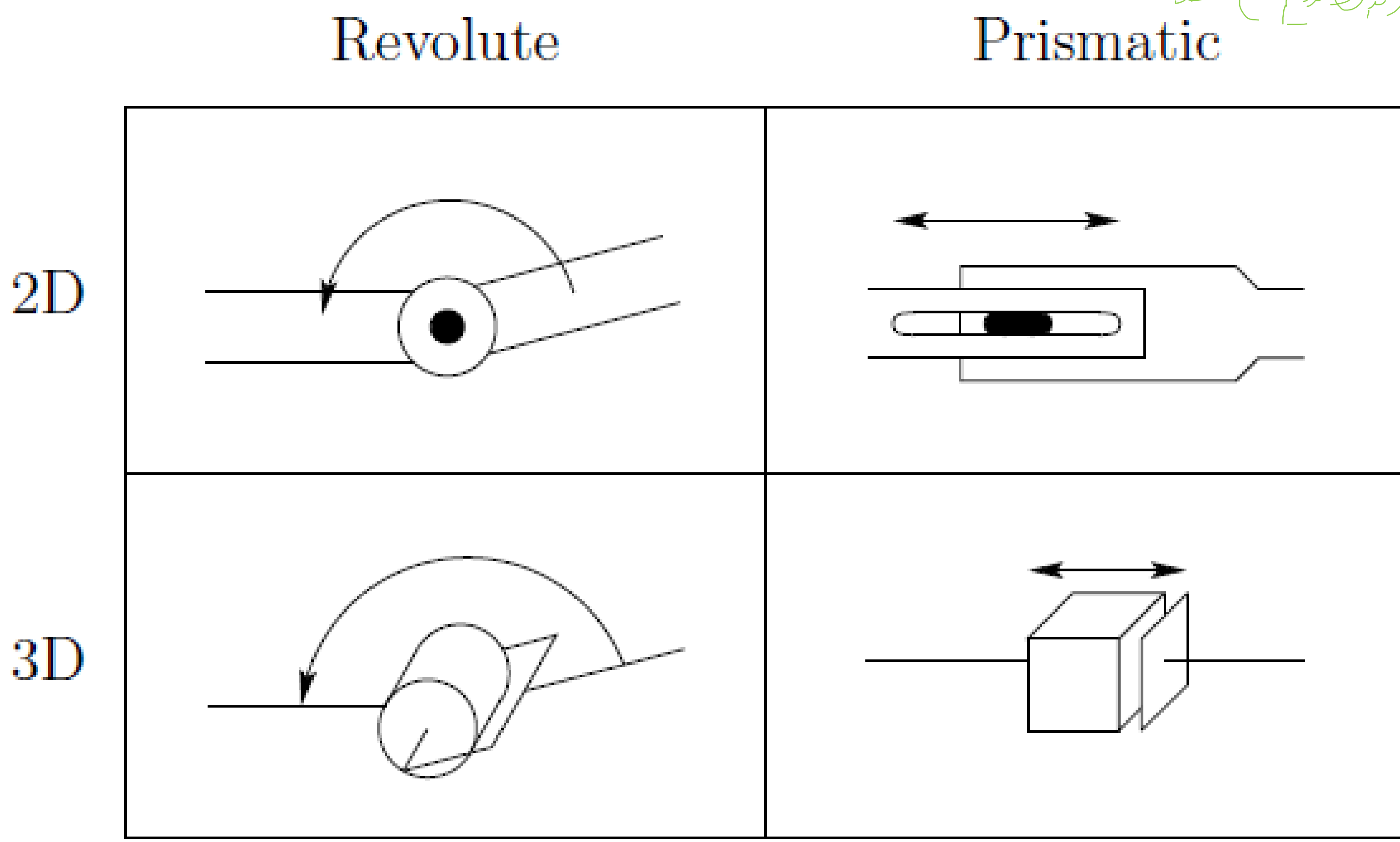
مشکل در حرکت‌ها دست یافتن در سمت است و جهتی نیست.
(عدم قابلیت دسترسی به مختصات توسط ربات)

Review

serial linkage

- Symbolic Representation of Robot Joints

حرکت خطی - کشویی



Some Terminology

محدودیت موانع و محیط را در آن لحاظ
همه را لحاظ می‌کنیم

فضای بدون محدودیت

- Configuration and Configuration Space
  - Any set of parameter values which uniquely specifies the robot kinematic state is a robot configuration (represented as a vector like q)

بردار حرکت

دست جیگر رو در زیر

- State and State Space

نوع حرکات را توسط configuration داریم ولی باید مشخص کنیم جهت و سرعت حالت حرکتی را. باید دو حالت ثبت کنیم
جهت را مشخص می‌کند حرکت

مشتق q

  - The robot state includes its configuration vector besides the configuration differentials: $s = \begin{bmatrix} q & \dot{q} \end{bmatrix}^T$

فضای با محدودیت

فضای حاصل

- Work Space and Dexterous Work Space
  - The set of all possible working points. Each work point is a vector containing target parameters of the manipulator.
  - Parts of workspace with necessary agility is called dexterous workspace.

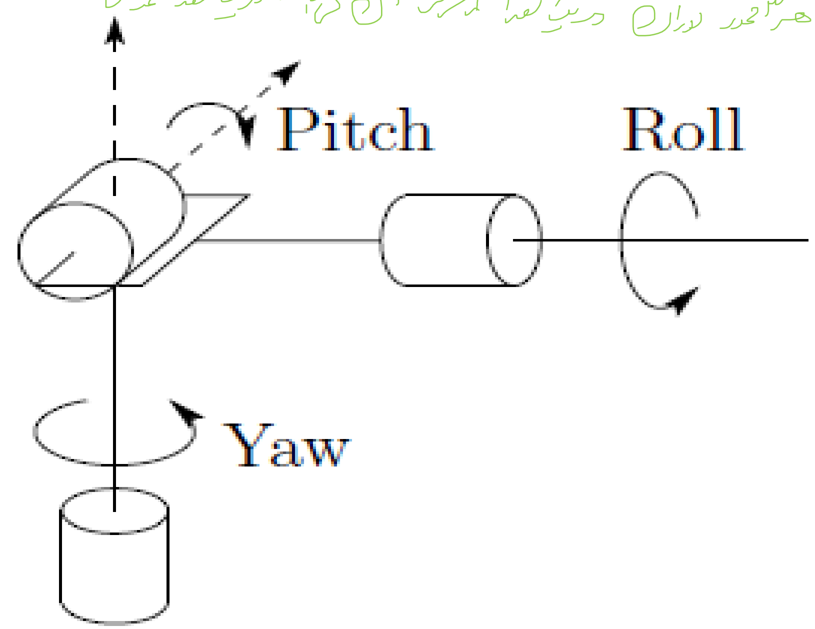
چه قدر می‌تونیم در شکل ایراد اش نیرو وارد کنیم؟ (شرط دست کشیدن) - می‌توانیم برسیم و چرخ جهت را داریم.

Terminology Continued

- Degree Of Freedom (DOF)
  - The minimum number of parameters required to specify configuration of an object is its degrees of freedom. A totally free object in the space has 6 DOFs.
- Constraint
  - The concept which takes away one or more degrees of freedom.
    - Holonomic Constraint: Partitions the space into two sub spaces and limit the position. Example: mechanical limits on the joint parameters.
    - Non-holonomic Constraint: Imposes limitation on the velocities or higher degree derivatives but leaves all points in the vector space acceptable. Example: wheel contacts with the ground.

Terminology Continued

- Arm and Wrist
  - The first three DOF of the manipulator is called the arm and the rest of them is denoted as the wrist.
  - The wrist joints are normally revolute and a common configuration is spherical wrist. The spherical wrist joints have axes which intersect at a common point in space.
  - The special geometry of spherical wrist greatly simplifies the kinematics of the manipulator.



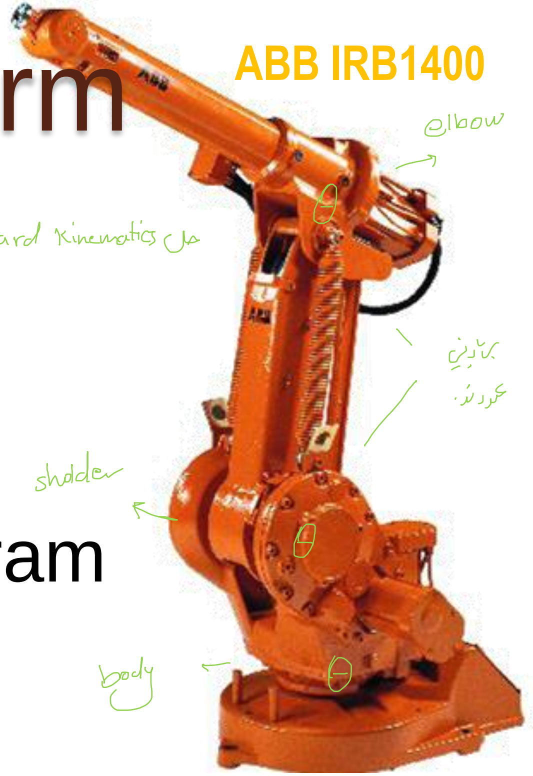
Categorization of the arm

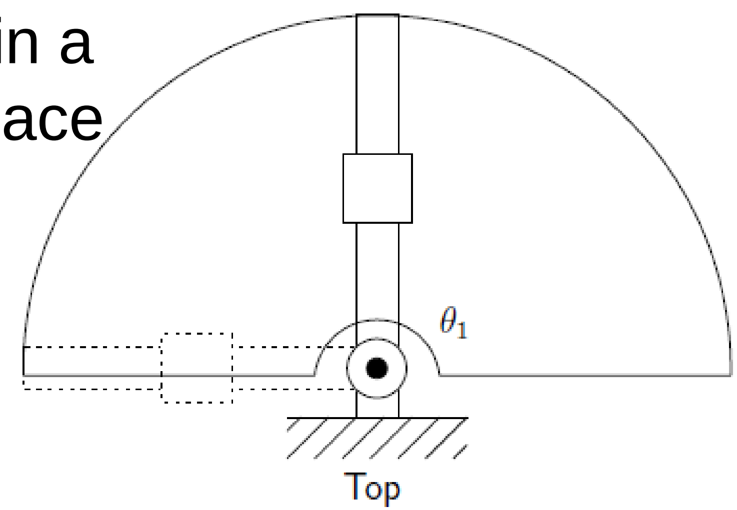
Arm

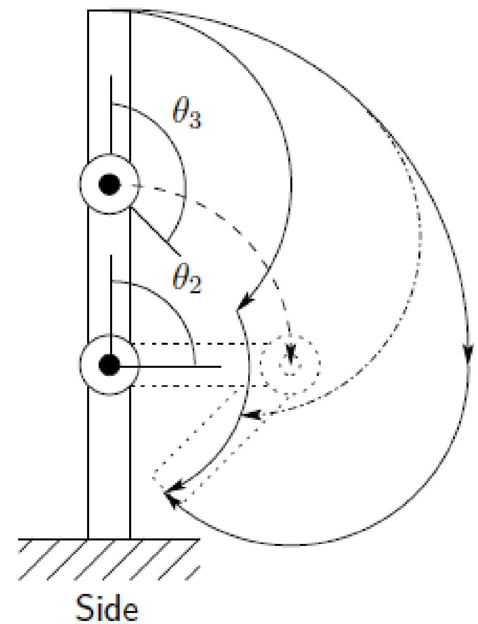
- Articulated : RRR

  rotation — forward kinematics

  - Also called a revolute or anthropomorphic manipulator.
  - Includes Elbow and Parallelogram configurations
- Workspace
  - Provides relatively large freedom of movement in a compact space







Elbow Manipulator

- A dexterous articulated manipulator which consists of body, shoulder, elbow and forearm:

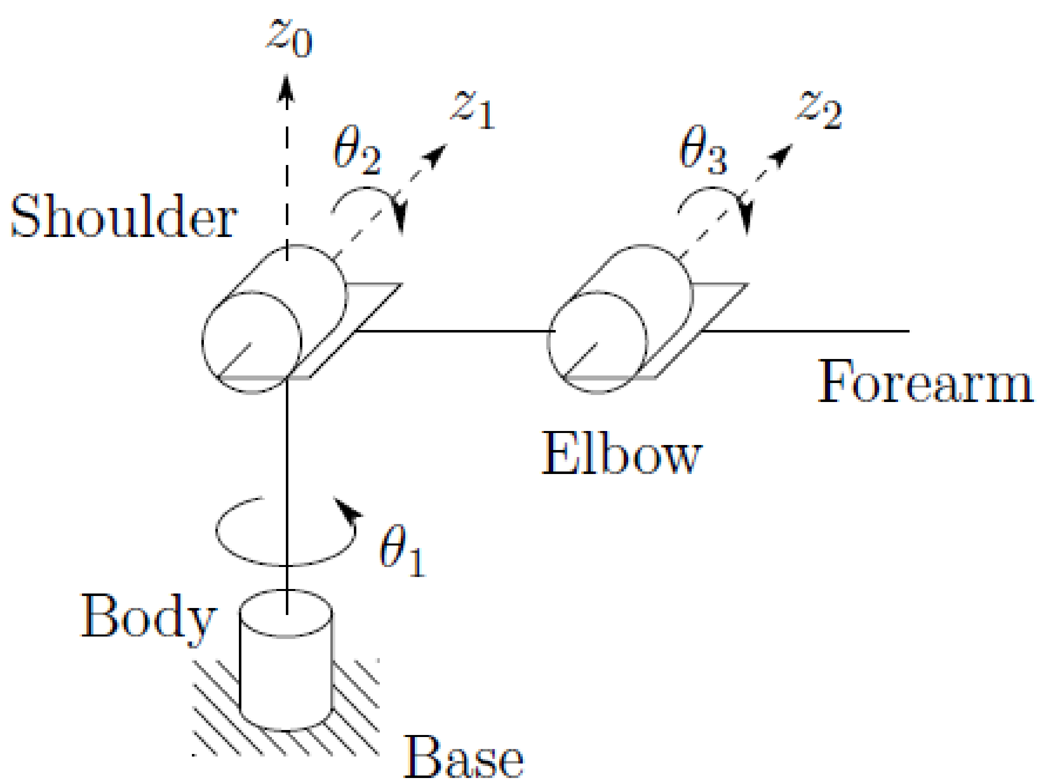


ABB IRB1400

Parallelogram Linkage

(RRR)

- Special case of Articulated Manipulator
  - Typically less dexterous than the elbow manipulator, but has several advantages
    - The actuator of the third joint is positioned on the first link:
      - Lighter links
      - Smaller motors
      - Faster dynamics
      - Easier control
    - The extra parallel linkage transfers the power

برای تعادل هم نیازی نیست با استفاده از ... مرکز موتور را به المان های پایین تری انتقال داده که حرکت بر لینک ها تاثیر نداشته باشد و ابتدای لینک در آن است. (ربات موازی) - یعنی بار تحمل شده توسط ربات را به زمین ترجیح کرده و منتقل می کنیم.

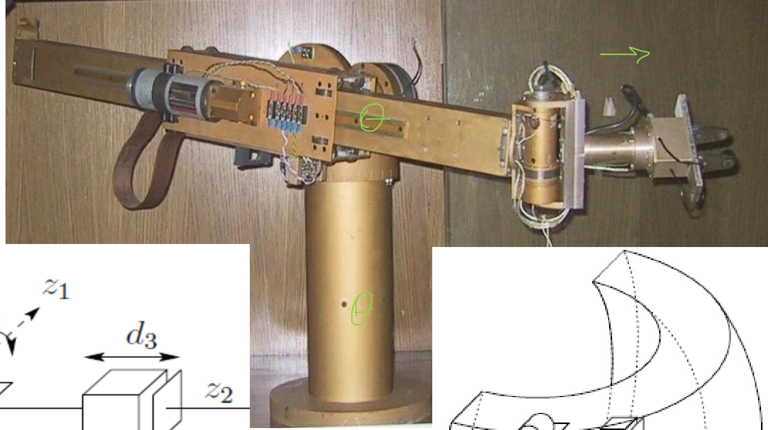
باعث ایجاد محدودیت هایی می شود مثلاً joint ۲ نمی تواند بیشتر از ۱۸۰° حرکت کند. (دامنه حرکت کمتر و فضای ماهرانه کمتر) به علت این محدودیت ممانعت ایجاد شده است.

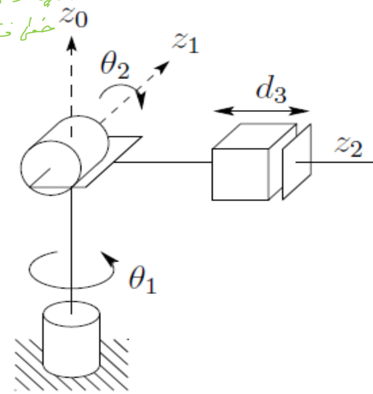
ولی چون هزینه مناسب تر می شود می توان موتورهای کوچک تر استفاده کرد که دقت را بالاتر برده و ظرافت کار بیشتر می شود.

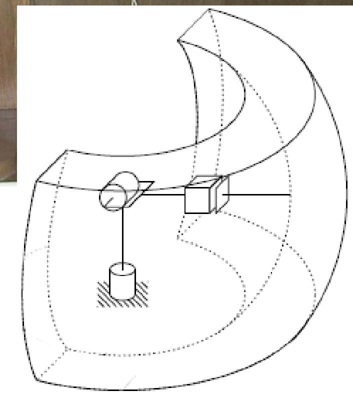
در هر forward ساده و inverse سخت است ولی در موازی برعکس است و inverse ساده است.

Categorization of the arm

به جای x,y,z می‌توان مولفه‌ها را افزایش داد. (مرکز تحرک)
کاربرد در جاهایی که حرکت خطی مورد استفاده است. به علت وجود حرکت
خطی فضای بزرگتری را ربات مورد نیاز است. بنابراین می‌توانند محدودیت ایجاد کنند.





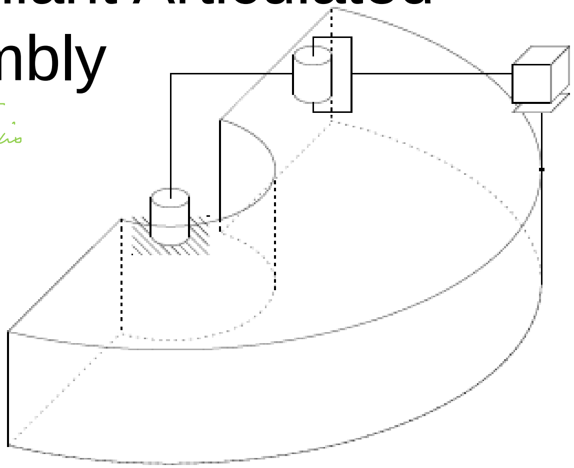


- Spherical : RRP
  - Its parameters match spherical coordinates
  - A famous example is the Stanford arm — scheinman arm

در جهت z prismatic است.

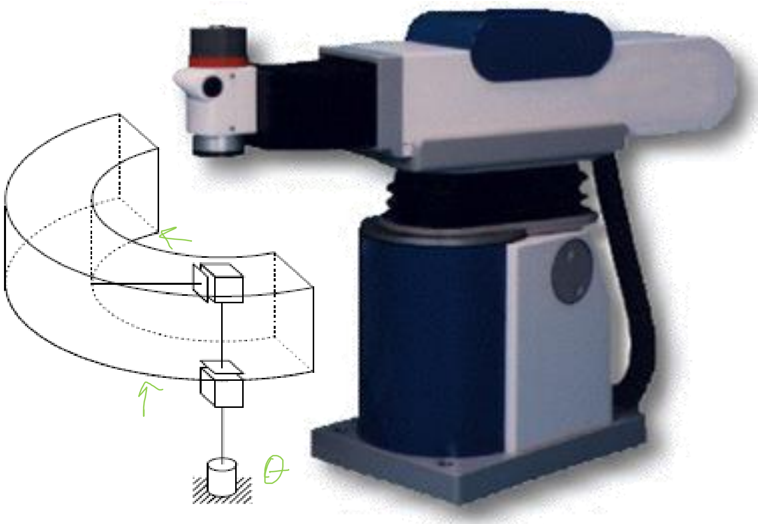
شبیه استوانه‌ای است ولی به عمق محیط
در جهت cartezian

- SCARA : RRP
  صرفه‌جویی
  - Selective Compliant Articulated Robot for Assembly
    Assembly ؛ سرهم‌سازی



Categorization of the arm

- Cylindrical : RPP
  - Its parameters match cylindrical coordinates

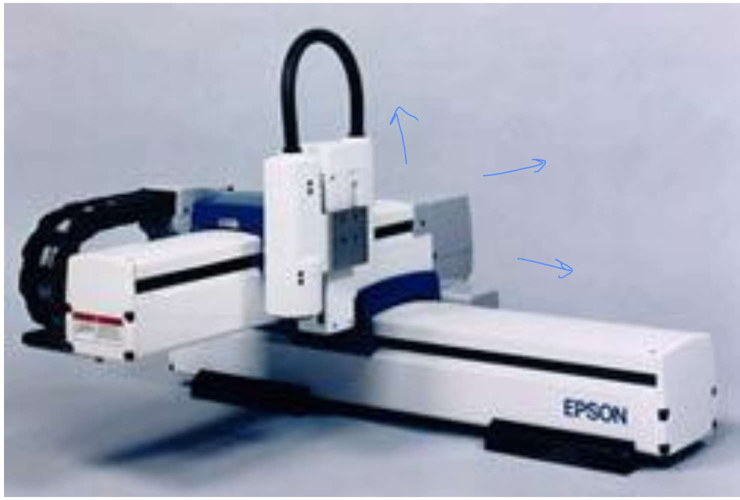


Seiko RT3300

- Cartesian : PPP
  - The simplest kinematic
  - Cubic workspace
  - Useful in assembly, storage and gantry facilities

اینجا دوران و حرکت چرخشی نداریم.

در شرکت‌ها و کارخانه‌ها
برای ساخت پل‌ها



شبیه به plotter که برای خطوط مورد نظر به printer تبدیل است و به همین
دلیل دقت بیشتری دارد

Parallel Manipulator

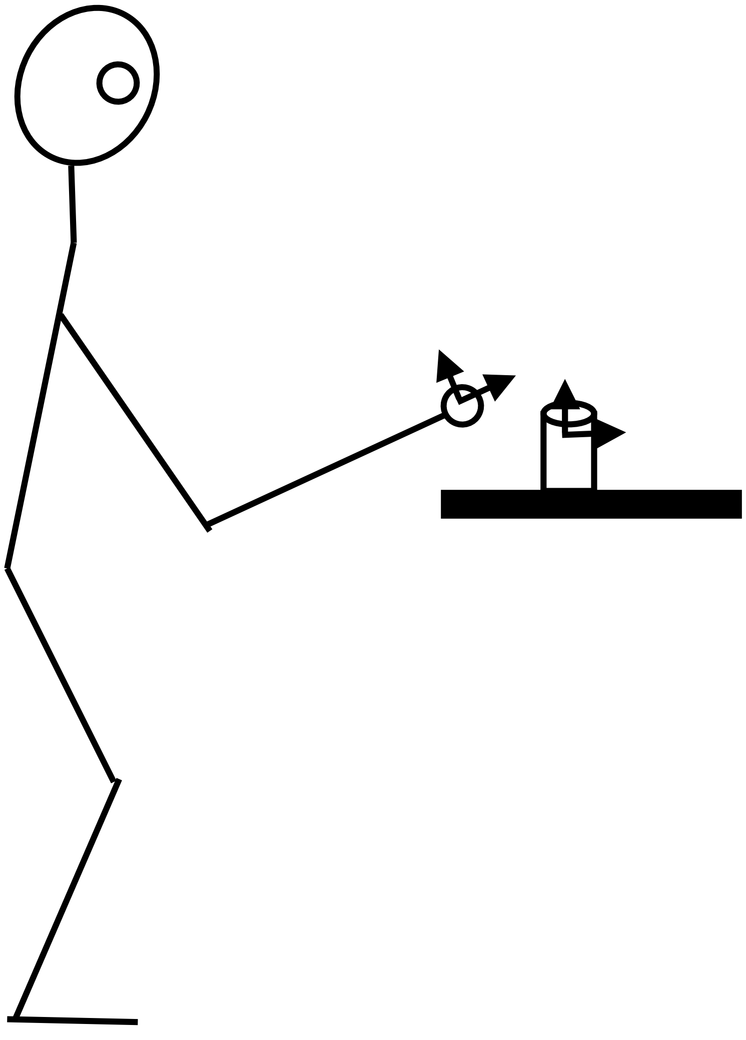
- When there are parallel linkages in the manipulator design, that is two or more independent kinematic chains connecting the base to the end-effector, the manipulator is called parallel.
- Greater structural rigidity and higher accuracy are the resultant advantages, but the solution to the kinematics of such robots are more complicated.

Kinematics:
Pose (position and orientation)
of a Rigid Body

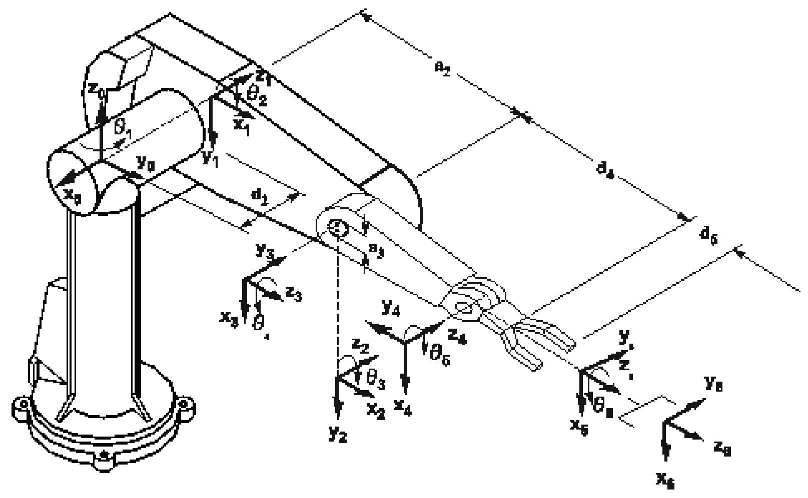
Why are we studying pose?

position + orientation

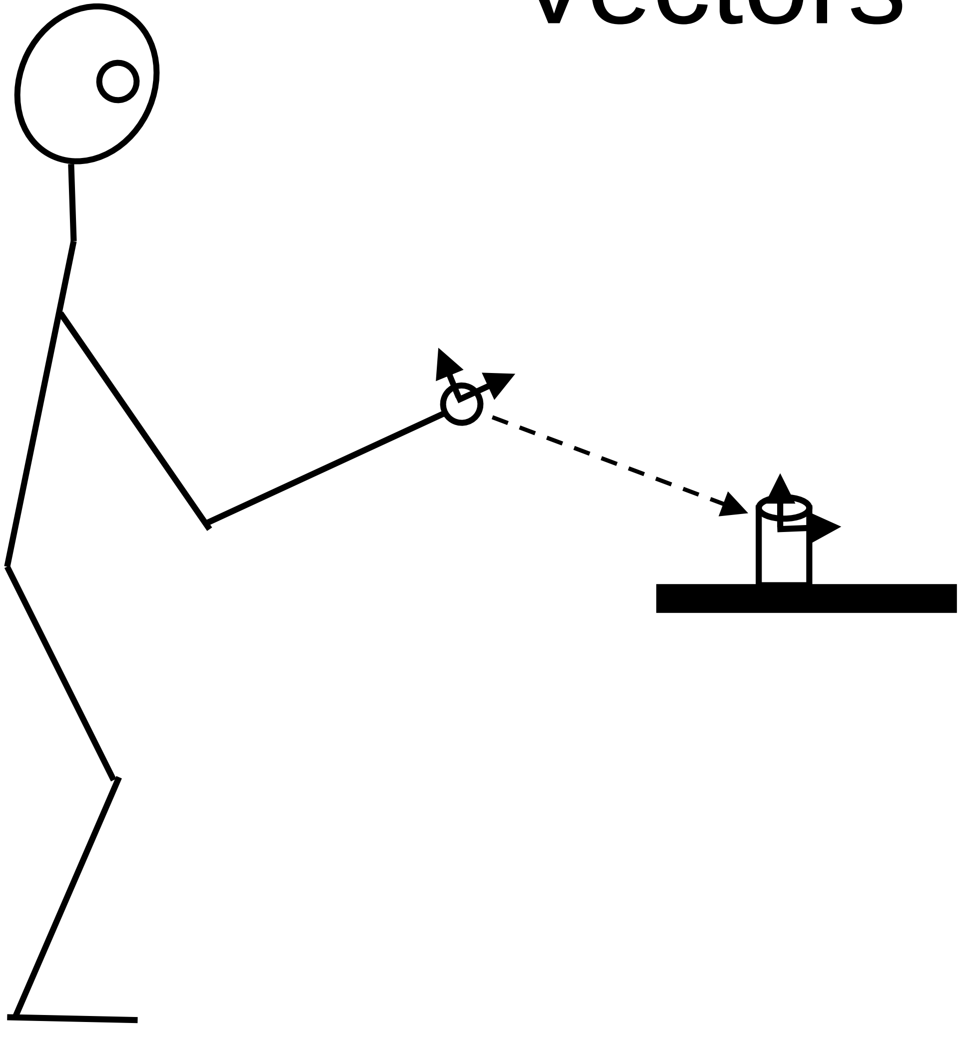
You want to put your hand on the cup…



- Suppose your eyes tell you where the mug is and its orientation in the robot *base frame* (big assumption)
- In order to put your hand on the object, you want to align the coordinate frame of your hand with that of the object
- This kind of problem makes representation of pose important...

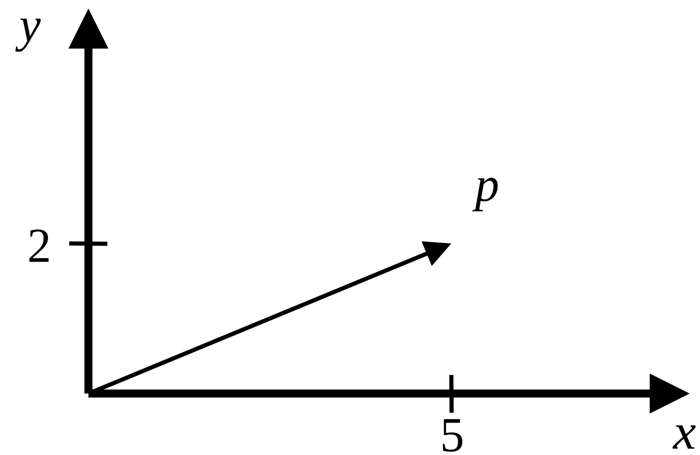


Representing Position: Vectors



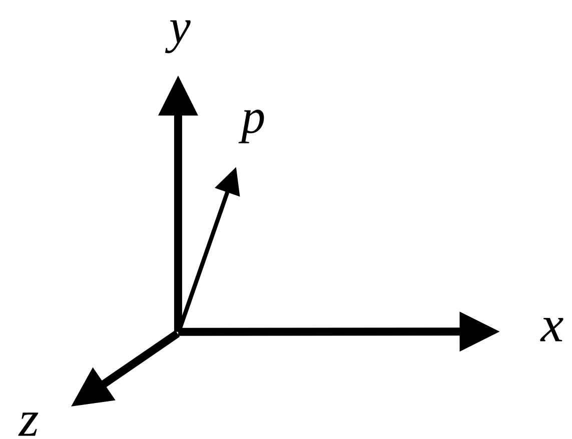
Representing Position: vectors

$$p = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$ ("column" vector)



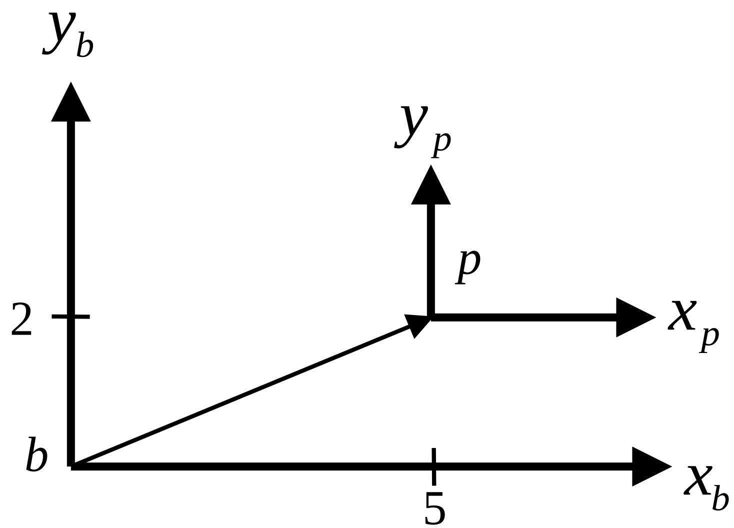
$$p = \begin{bmatrix} 2 & 5 \end{bmatrix}$$ ("row" vector)

$$p = \begin{bmatrix} 2 \\ 5 \\ 2 \end{bmatrix}$$



Representing Position: vectors

- Vectors are a way to transform between two different reference frames with the same orientation
- The prefix superscript denotes the reference frame in which the vector should be understood

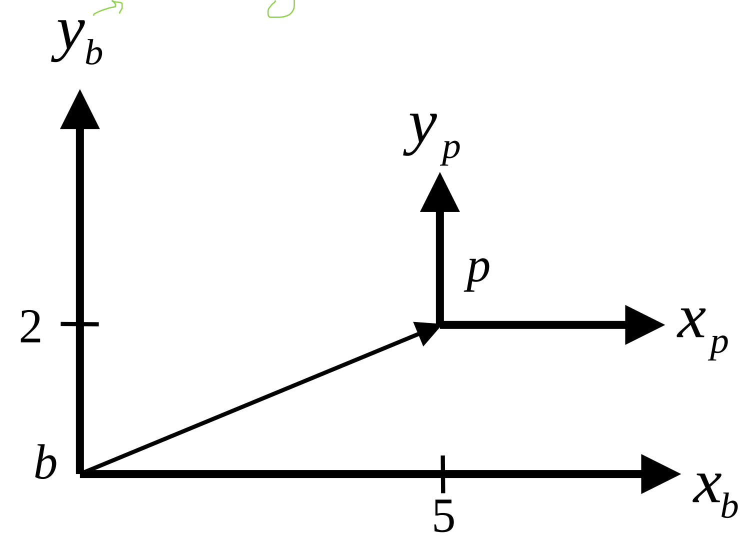


بردار p در فضای b

$$ {}^{b}p = \begin{bmatrix} 5 \\ 2 \end{bmatrix} \qquad {}^{p}p = \begin{bmatrix} 0 \\ 0 \end{bmatrix} $$

Same point, two different reference frames

Representing Position: vectors



- Note that the axes are *orthogonal* unit basis vectors

This means “perpendicular”

y_b ← *y* axis of the base frame

y_p ← *y* axis of *p* frame

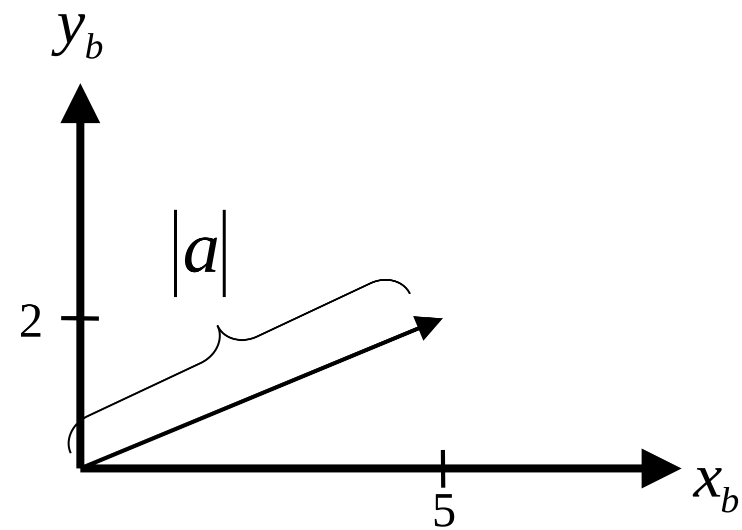
What is this unit vector you speak of?

These are the elements of a:

$$a = \begin{bmatrix} a_x \\ a_y \end{bmatrix}$$

Vector length/magnitude:

$$|a| = \sqrt{a_x^2 + a_y^2}$$



بردار راستا

Definition of unit vector: $|\hat{a}| = 1$

You can turn a into a unit vector of the same direction this way:

$$\hat{a} = \frac{a}{\sqrt{a_x^2 + a_y^2}}$$

And what does orthogonal mean?

First, define the dot product: $$a \cdot b = a_x b_x + a_y b_y$$

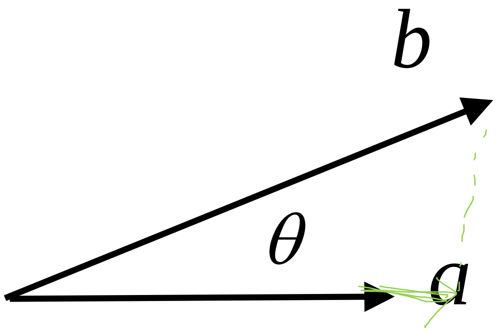
$$= |a||b|\cos(\theta)$$

تصویر

$a \cdot b = 0$ when: $a = 0$

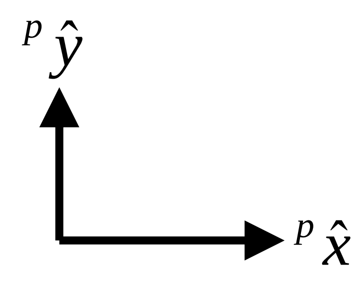
or, $b = 0$

or, $\cos(\theta) = 0$



Two vectors are orthogonal *iff* (if and only if) the dot product is zero:

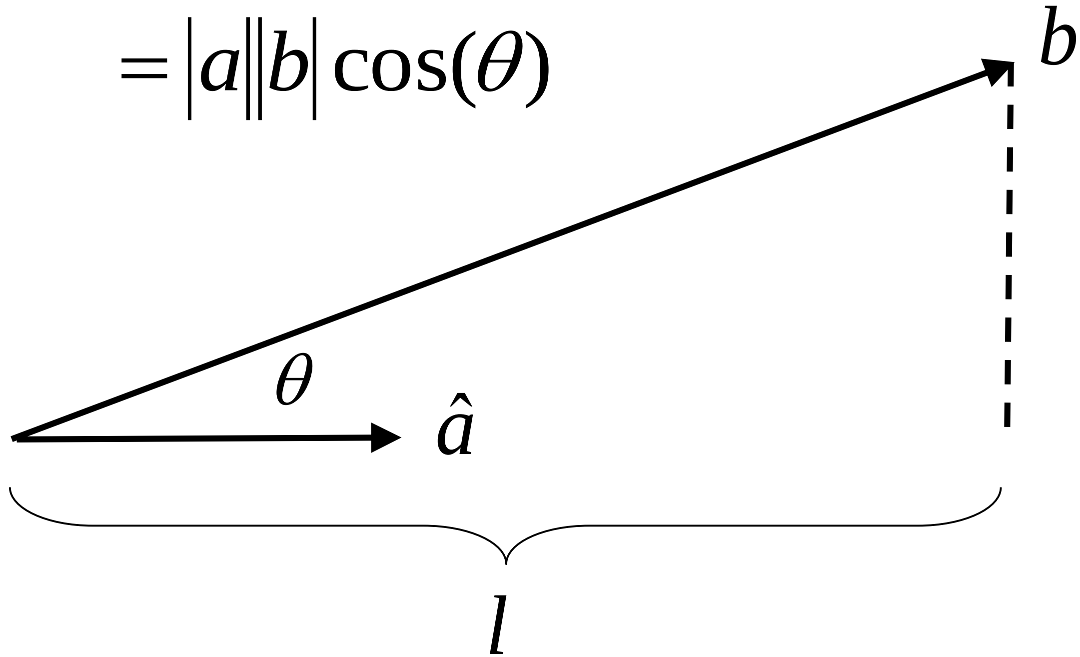
${}^{p}\hat{x}$ is orthogonal to ${}^{p}\hat{y}$ iff ${}^{p}\hat{x} \cdot {}^{p}\hat{y} = 0$



Another important use of the dot product: projection

$$a \cdot b = a_x b_x + a_y b_y$$

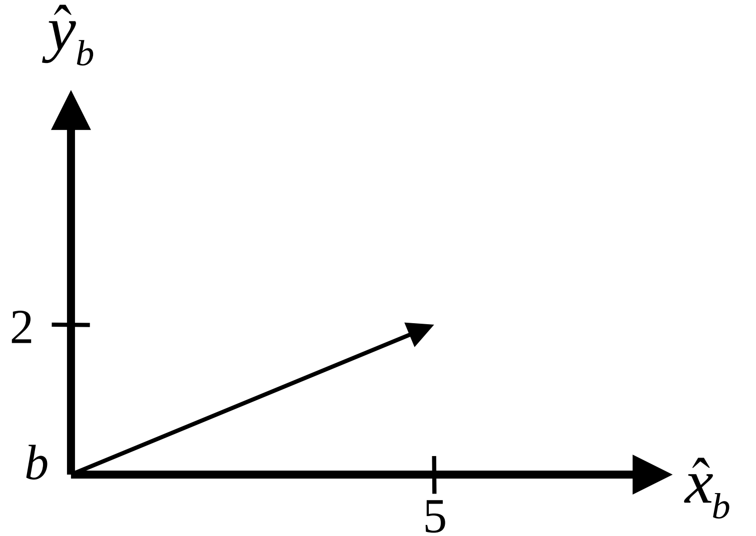
$$= |a||b|\cos(\theta)$$



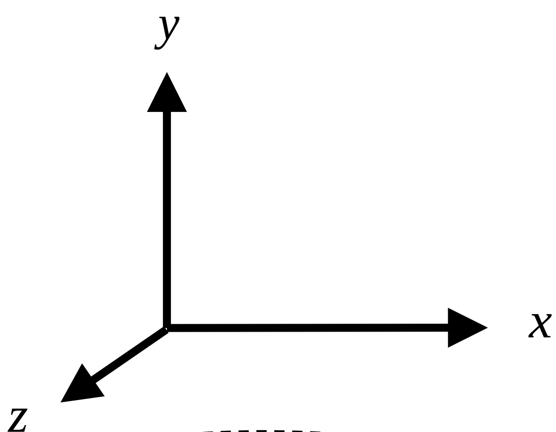
$$l = \hat{a} \cdot b = |\hat{a}||b|\cos(\theta) = |b|\cos(\theta)$$

A couple of other random things

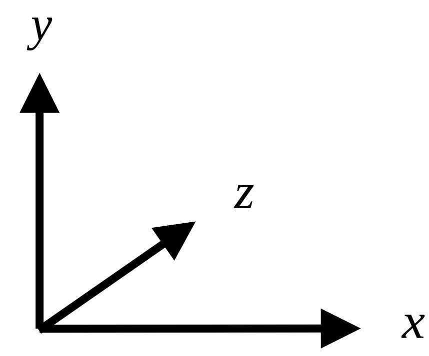
$$p_b = 5\hat{x}_b + 2\hat{y}_b$$



Vectors are elements of R^n



right-handed coordinate frame

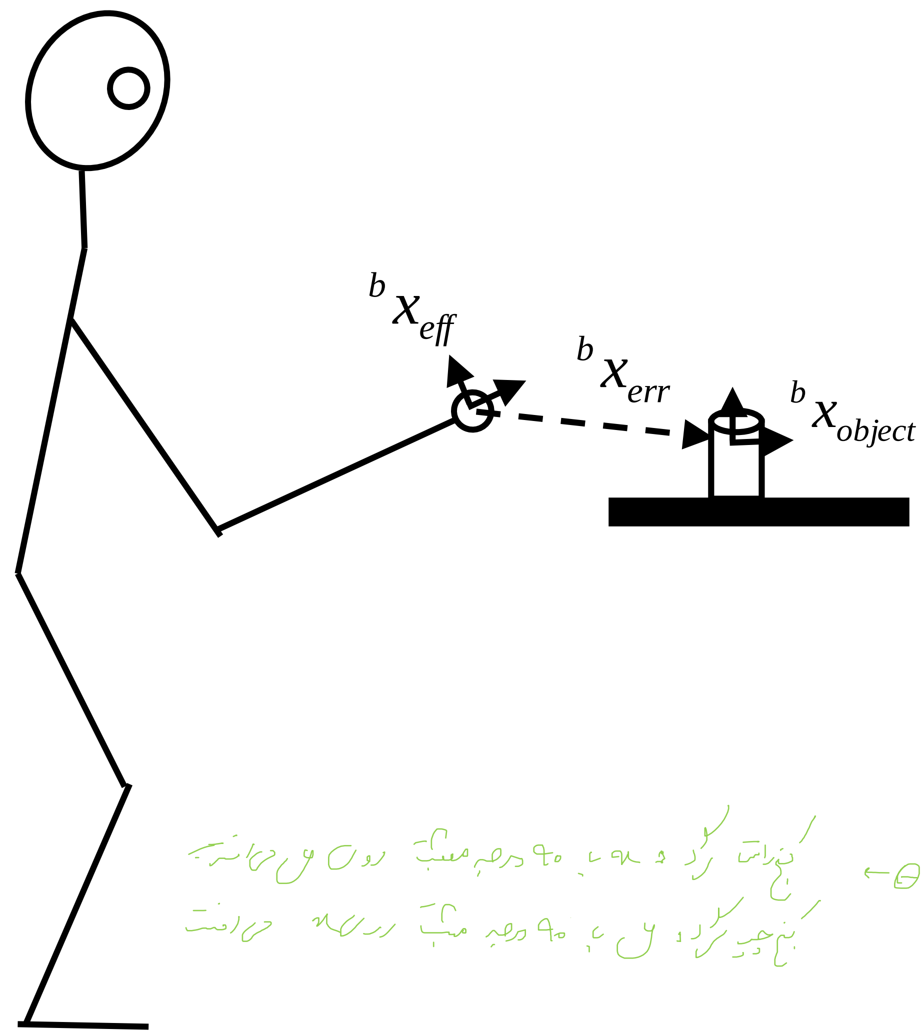


left-handed coordinate frame

The importance of differencing two vectors

$$ {}^{b}x_{object} - {}^{b}x_{eff} = {}^{b}x_{err} $$

The *eff* needs to make a Cartesian displacement of this much to reach the object

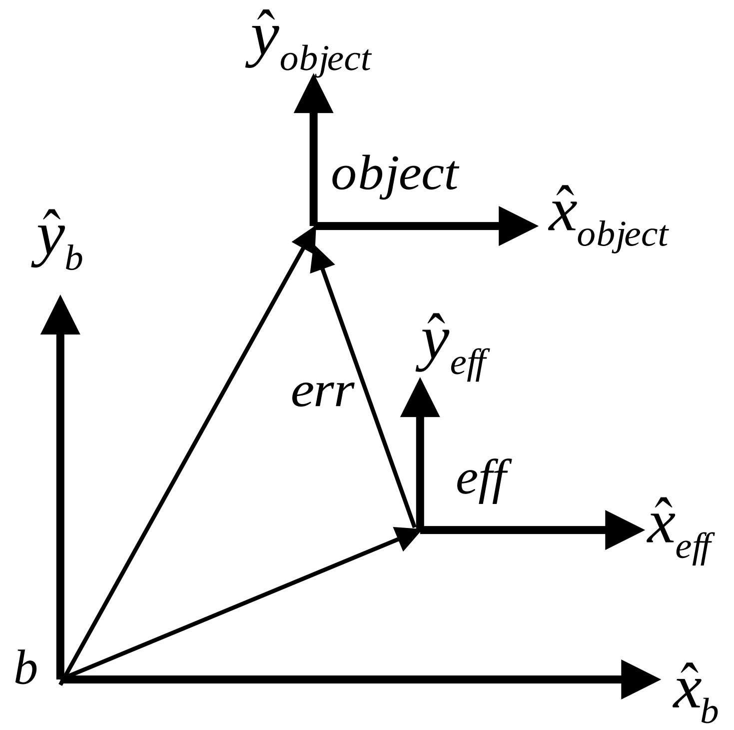


$\theta \leftarrow$ [illegible] ۹۰ درجه مثبت [illegible]
[illegible] ۹۰ درجه مثبت [illegible]

The importance of differencing two vectors

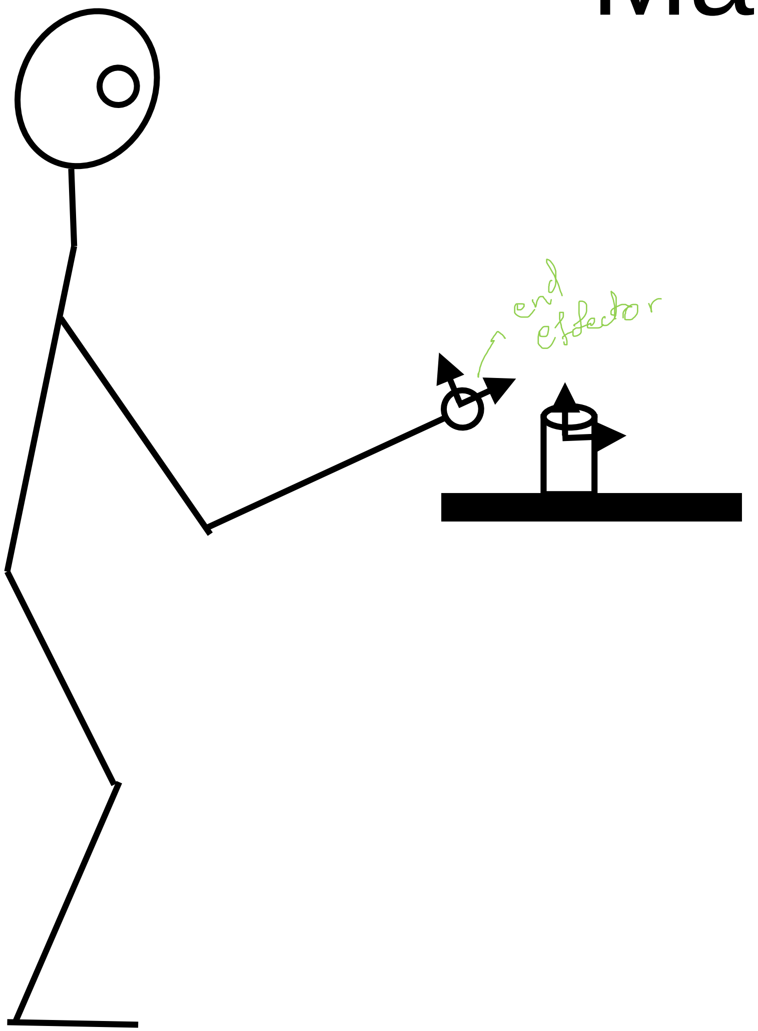
$$ {}^{b}x_{object} - {}^{b}x_{eff} = {}^{b}x_{err} $$

اختلاف بین
حالت فعلی و
حالت هدف



The *eff* needs to make a Cartesian displacement of this much to reach the object

Representing Orientation: Rotation Matrices



- The reference frame of the hand and the object have different orientations
- We want to represent and difference orientations just like we did for positions…

Before we go there – review of matrix transpose

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$\mathbf{A}^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix}$$

$$p = \begin{bmatrix} 5 \\ 2 \end{bmatrix} \longrightarrow p^T = \begin{bmatrix} 5 & 2 \end{bmatrix}$$

Important property: $\mathbf{A}^T\mathbf{B}^T = (\mathbf{BA})^T$

and matrix multiplication…

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$\mathbf{AB} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

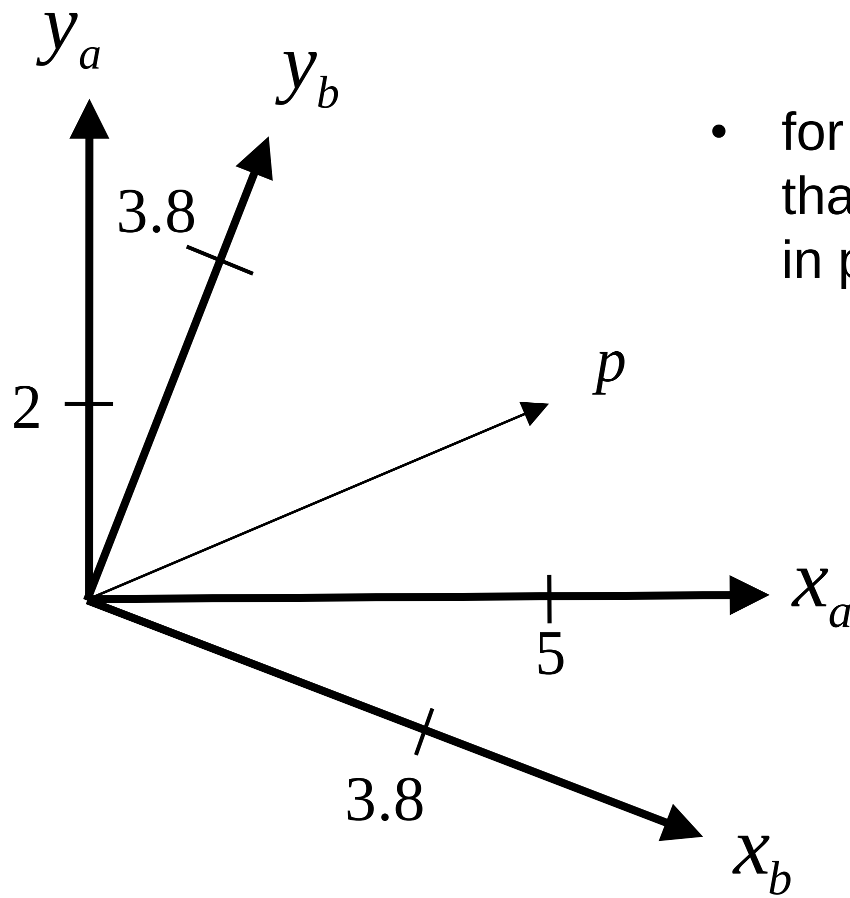
Dot product can be represented as multiplication of two matrices:

ارتباط ضرب داخلی و ضرب ماتریسی

$$a \cdot b = a_x b_x + a_y b_y = \begin{bmatrix} a_x & a_y \end{bmatrix} \begin{bmatrix} b_x \\ b_y \end{bmatrix} = a^T b$$

matrix
Product

Same point - different reference frames



- for the moment, assume that there is no difference in position…

$$^{a}p = \begin{bmatrix} 5 \\ 2 \end{bmatrix} \qquad ^{b}p = \begin{bmatrix} 3.8 \\ 3.8 \end{bmatrix}$$

Same point - different reference frames

یعنی

$$^{b}p_y = {}^{a}\hat{y}_b \cdot {}^{a}p$$

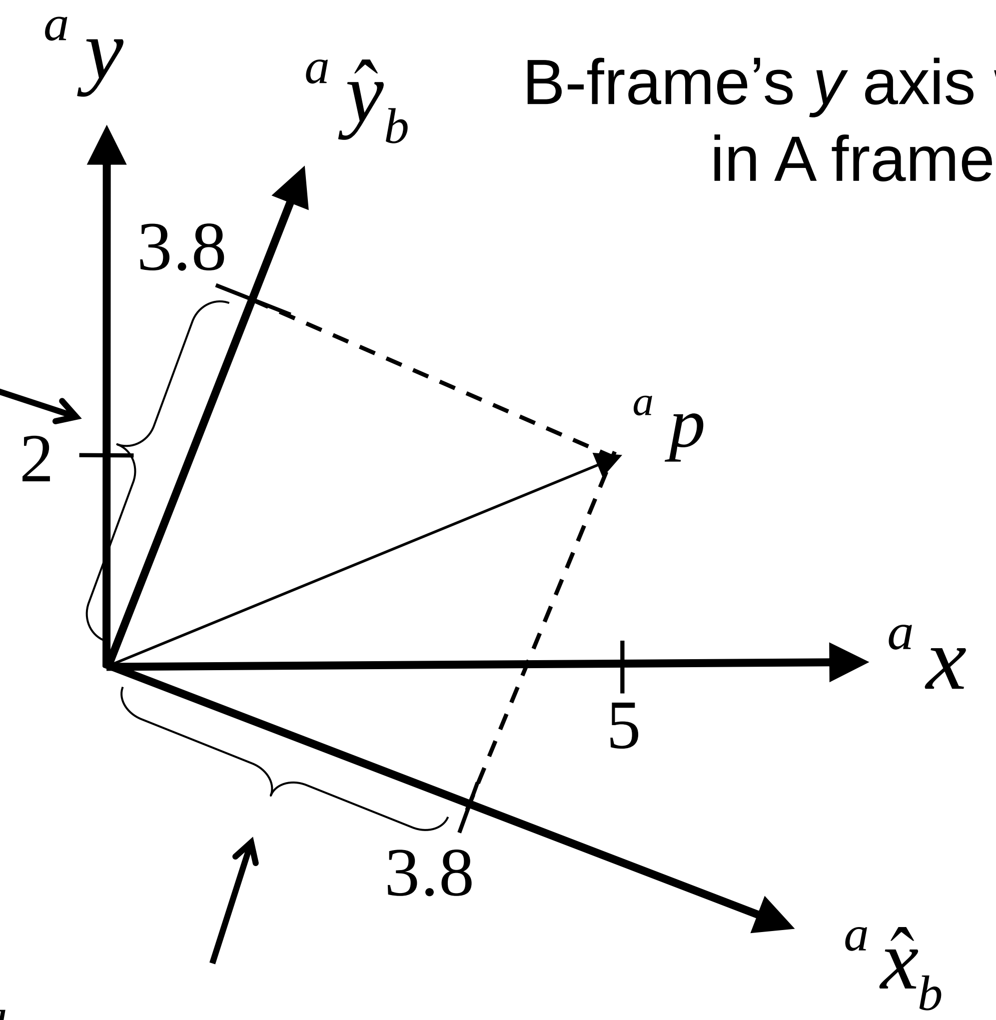
تصویر p روی ŷ در دستگاه b

در دستگاه a

چون ŷ بردار یکه است پس با ضرب داخلی

اگر بخواهیم مقدار y بردار p را در دستگاه

b بدست می‌آوریم.



B-frame's *y* axis written in A frame

$$^{b}p_x = {}^{a}\hat{x}_b \cdot {}^{a}p$$

B-frame's *x* axis written in A frame

Same point - different reference frames

$$ {}^{B}p = \begin{pmatrix} {}^{A}\hat{x}_{B} \cdot {}^{A}p \\ {}^{A}\hat{y}_{B} \cdot {}^{A}p \end{pmatrix} = \begin{pmatrix} {}^{A}\hat{x}_{B}{}^{T}\, {}^{A}p \\ {}^{A}\hat{y}_{B}{}^{T}\, {}^{A}p \end{pmatrix} = \begin{pmatrix} {}^{A}\hat{x}_{B}{}^{T} \\ {}^{A}\hat{y}_{B}{}^{T} \end{pmatrix} {}^{A}p $$

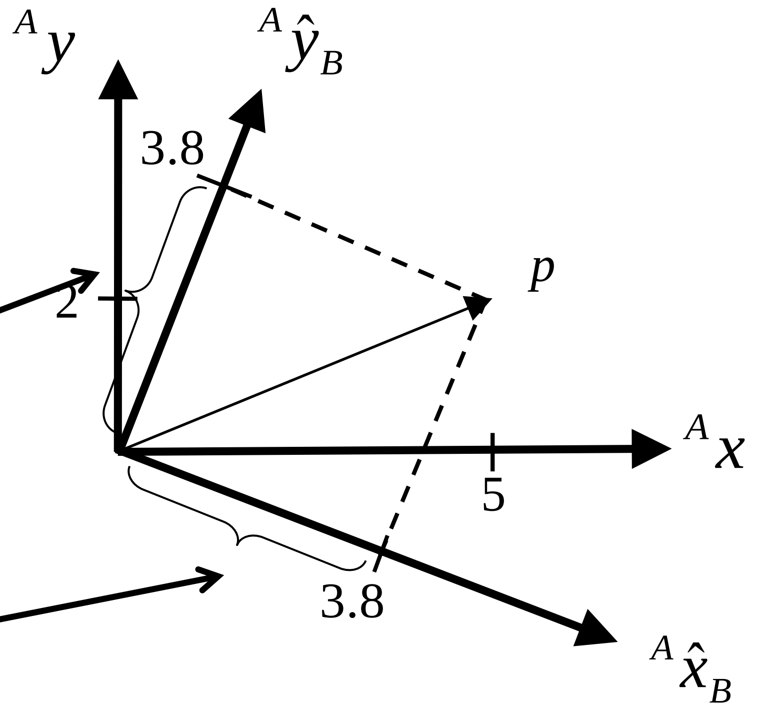
$$ {}^{B}p = \begin{pmatrix} {}^{A}\hat{x}_{B}{}^{T} \\ {}^{A}\hat{y}_{B}{}^{T} \end{pmatrix} {}^{A}p $$

$$ {}^{B}p = {}^{A}R_{B}{}^{T}\, {}^{A}p $$

Rotation matrix

$$ {}^{B}p_{y} = {}^{A}\hat{y}_{B} \cdot {}^{A}p $$

$$ {}^{B}p_{x} = {}^{A}\hat{x}_{B} \cdot {}^{A}p $$



The rotation matrix

From last page: $${}^{B}p = \begin{pmatrix} {}^{A}\hat{x}_{B}{}^{T} \\ {}^{A}\hat{y}_{B}{}^{T} \end{pmatrix} {}^{A}p \longrightarrow {}^{B}p = {}^{A}R_{B}{}^{T}\,{}^{A}p$$

By the same reasoning: $${}^{A}p = \begin{pmatrix} {}^{B}\hat{x}_{A}{}^{T} \\ {}^{B}\hat{y}_{A}{}^{T} \end{pmatrix} {}^{B}p \longrightarrow {}^{A}p = {}^{B}R_{A}{}^{T}\,{}^{B}p$$

The rotation matrix

به علت عدم توانایی انتقال بردارها در فضاهای مختلف، برای اینکه از بردار در frame جدید آن را در frame قدیم بازسازی کنیم.

$$^{A}R_{B} = \left({}^{A}\hat{x}_{B} \quad {}^{A}\hat{y}_{B} \right) \quad \text{and} \quad \boxed{{}^{A}R_{B} = {}^{B}R_{A}^{\;T}} * = \begin{pmatrix} {}^{B}\hat{x}_{A}^{\;T} \\ {}^{B}\hat{y}_{A}^{\;T} \end{pmatrix}$$

$$^{A}R_{B} = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \qquad \begin{matrix} {}^{A}\hat{x}_{B} & {}^{A}\hat{y}_{B} \end{matrix}$$

ستون‌ها و سطرها بر هم عمودند

orthonormal matrix

$A^{-1} = A^{T}$

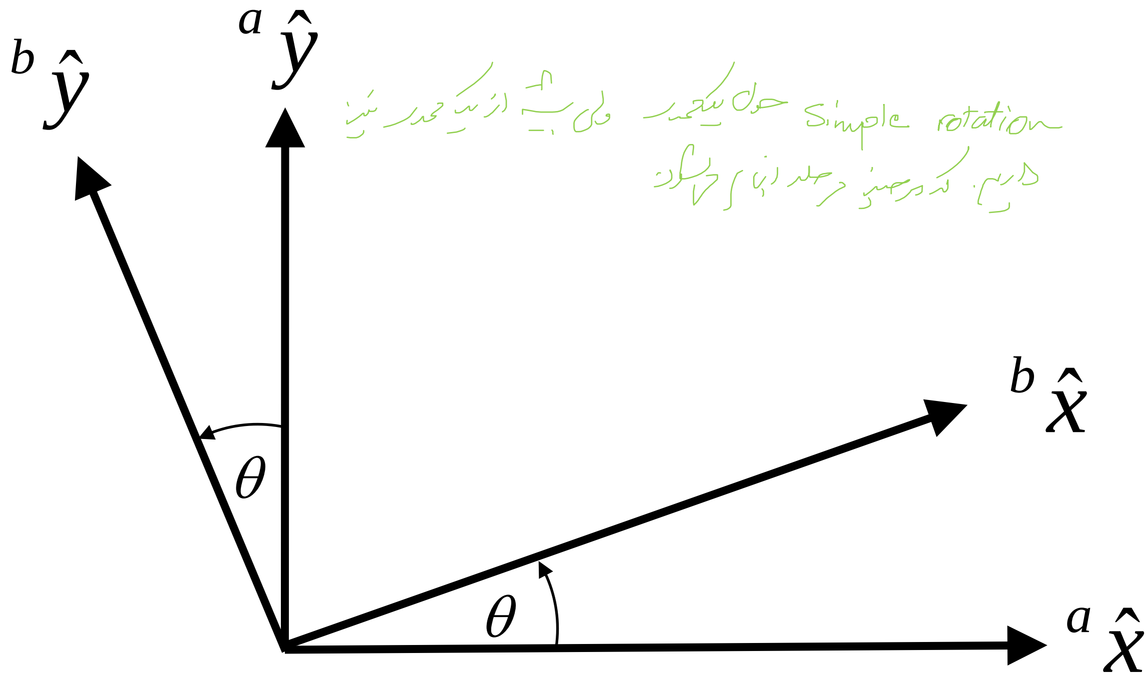
$$^{A}R_{B} = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \qquad \begin{matrix} {}^{B}\hat{x}_{A}^{\;T} \\ {}^{B}\hat{y}_{A}^{\;T} \end{matrix}$$

تغییر پایه بردار، اشاره دارد به بردارهای یکه و بر هم عمود. نتیجه: ترانهاده با معکوس برابر است.

The rotation matrix can be understood as:

1. Columns of vectors of *B* in *A* reference frame, OR
2. Rows of column vectors *A* in *B* reference frame

Example 1: rotation matrix



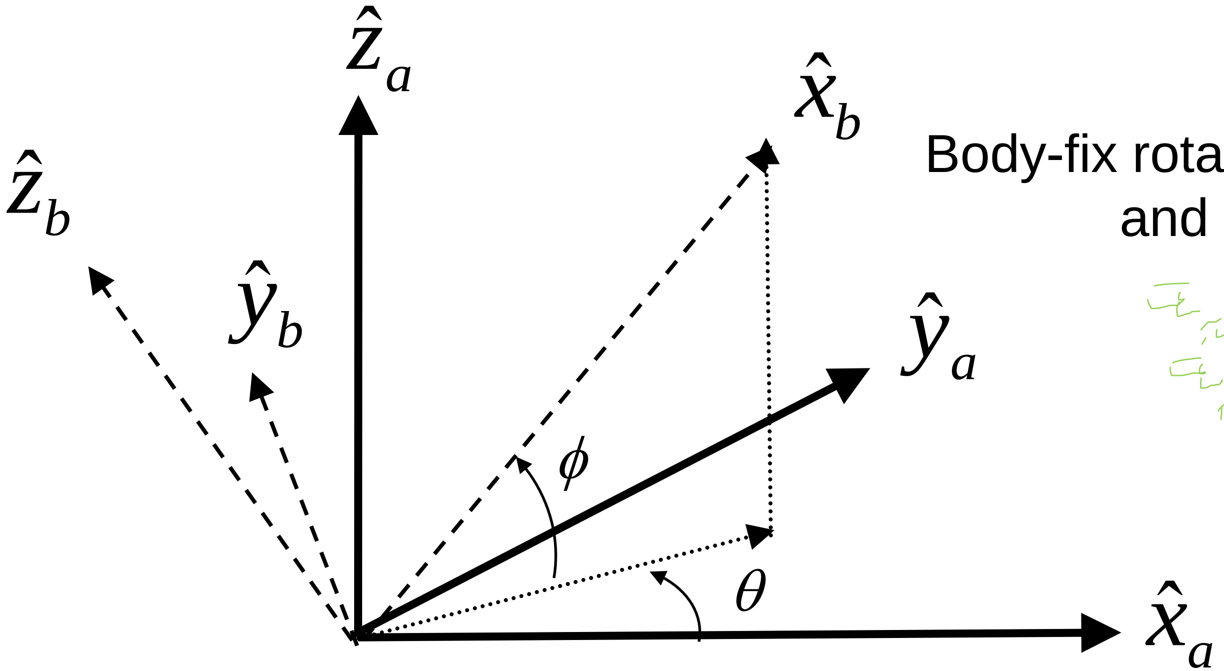
Simple rotation حول یک محور ولی بیشتر از یک محور نیز
داریم که در حین حرکت ایجاد میکند

$${}^{a}\hat{x}_b = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \qquad {}^{a}R_b = \begin{pmatrix} {}^{a}\hat{x}_b & {}^{a}\hat{y}_b \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

$${}^{a}\hat{y}_b = \begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \end{pmatrix} \qquad {}^{b}R_a = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

عمود بودن e سه بردار دستگاه
a و b در ترتیب e ا
وابسته به θ نیست.

Example 2: rotation matrix



Body-fix rotations: first around Z and then around Y

در جهت مثلثاتی مثبت و در جهت خلاف عقربه ساعت

خلاف جهت مثلثاتی منفی در جهت عقربه ساعت

$$
{}^{a}R_{b} = \begin{pmatrix} c_\theta c_\phi & -s_\theta & c_\theta c_{\phi+\frac{\pi}{2}} \\ s_\theta c_\phi & c_\theta & s_\theta c_{\phi+\frac{\pi}{2}} \\ s_\phi & 0 & s_{\phi+\frac{\pi}{2}} \end{pmatrix} = \begin{pmatrix} c_\theta c_\phi & -s_\theta & -c_\theta s_\phi \\ s_\theta c_\phi & c_\theta & -s_\theta s_\phi \\ s_\phi & 0 & c_\phi \end{pmatrix}
$$

Rotations about x, y, z

$$R_z(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

مثبت

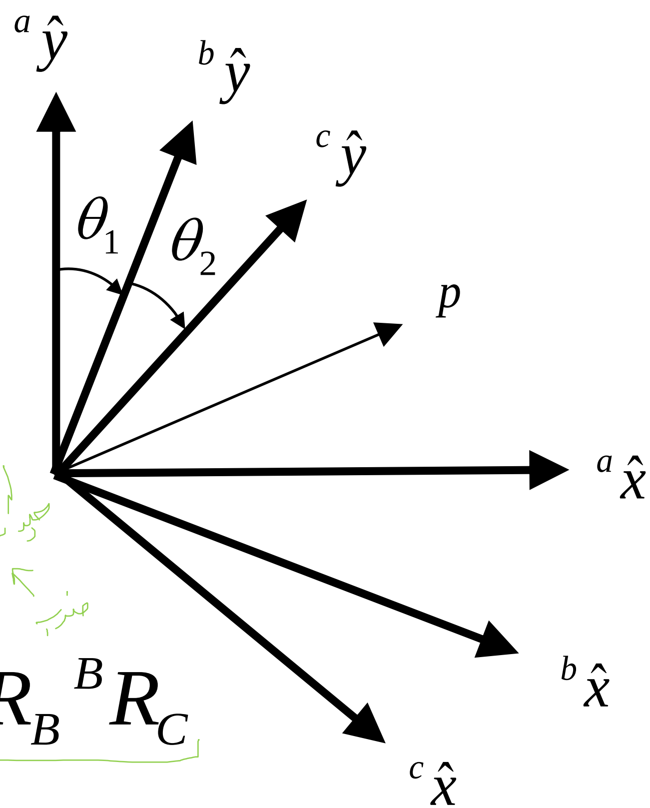
$$R_y(\beta) = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix}$$

منفی

$$R_x(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{pmatrix}$$

These rotation matrices encode the basis vectors of the after-rotation reference frame in terms of the before-rotation reference frame

Example 3: composition of rotation matrices



space fix

body fix

$$\sin(\theta \pm \phi) = \sin(\theta)\cos(\phi) \pm \cos(\theta)\sin(\phi)$$

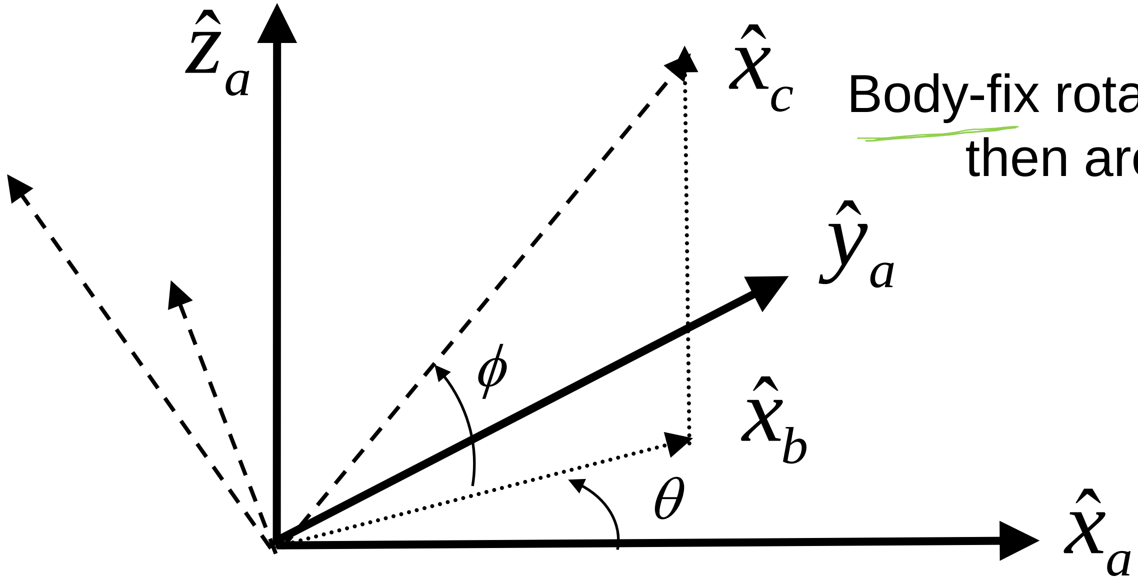
$$\cos(\theta \pm \phi) = \cos(\theta)\cos(\phi) \mp \sin(\theta)\sin(\phi)$$

$$^A R_C = \underbrace{^A R_B \, ^B R_C}$$

$$^a R_c = \begin{pmatrix} \cos(\theta_1) & -\sin(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) \end{pmatrix} \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) \end{pmatrix} = \begin{pmatrix} c_1 c_2 - s_1 s_2 & -c_1 s_2 - s_1 c_2 \\ s_1 c_2 + c_1 s_2 & c_1 c_2 - s_1 s_2 \end{pmatrix}$$

$$= \begin{pmatrix} c_{12} & -s_{12} \\ s_{12} & c_{12} \end{pmatrix}$$

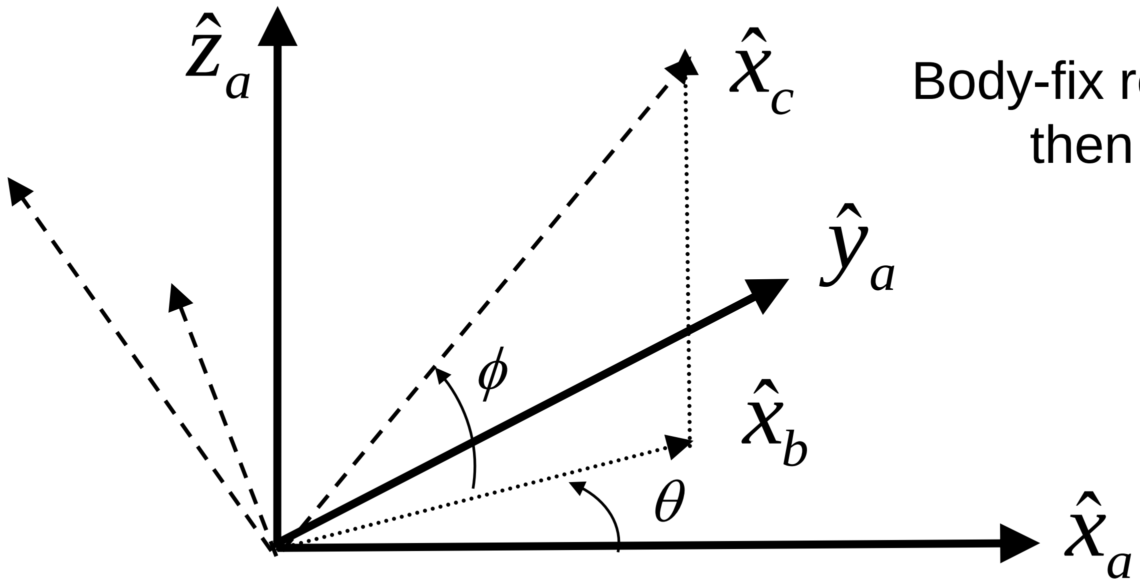
Example 4: composition of rotation matrices



Body-fix rotations: first around Z and then around Y, same as Ex. 2

$$ {}^{a}R_{b} = \begin{pmatrix} c_\theta & -s_\theta & 0 \\ s_\theta & c_\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad {}^{b}R_{c} = \begin{pmatrix} c_{-\phi} & 0 & s_{-\phi} \\ 0 & 1 & 0 \\ -s_{-\phi} & 0 & c_{-\phi} \end{pmatrix} = \begin{pmatrix} c_\phi & 0 & -s_\phi \\ 0 & 1 & 0 \\ s_\phi & 0 & c_\phi \end{pmatrix} $$

Example 4: composition of rotation matrices



Body-fix rotations: first around Z and then around Y, same as Ex. 2

$$
{}^aR_c = {}^aR_b\,{}^bR_c = \begin{pmatrix} c_\theta & -s_\theta & 0 \\ s_\theta & c_\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_\phi & 0 & -s_\phi \\ 0 & 1 & 0 \\ s_\phi & 0 & c_\phi \end{pmatrix} = \begin{pmatrix} c_\theta c_\phi & -s_\theta & -c_\theta s_\phi \\ s_\theta c_\phi & c_\theta & -s_\theta s_\phi \\ s_\phi & 0 & c_\phi \end{pmatrix}
$$

Recap of rotation matrices

$$
{}^{A}R_{B} = \begin{pmatrix} {}^{A}\hat{x}_{B} & {}^{A}\hat{y}_{B} \end{pmatrix} = \begin{pmatrix} {}^{b}\hat{x}_{a}{}^{T} \\ {}^{b}\hat{y}_{a}{}^{T} \end{pmatrix}
$$

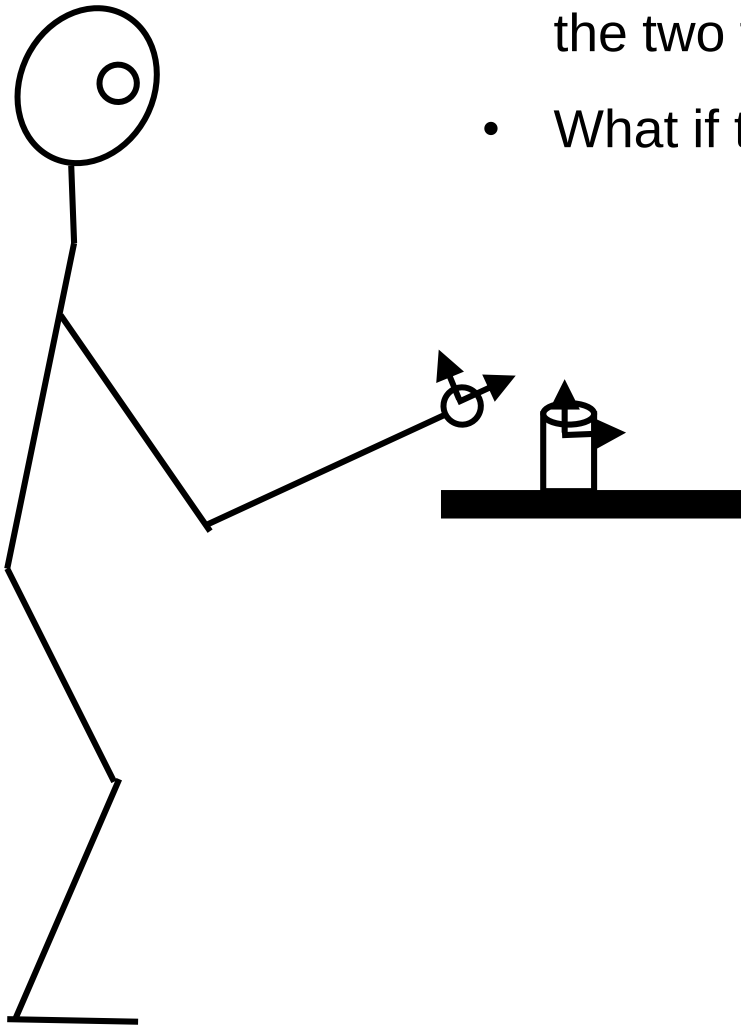
$$
{}^{b}R_{a}{}^{-1} = {}^{b}R_{a}{}^{T} = {}^{a}R_{b}
$$

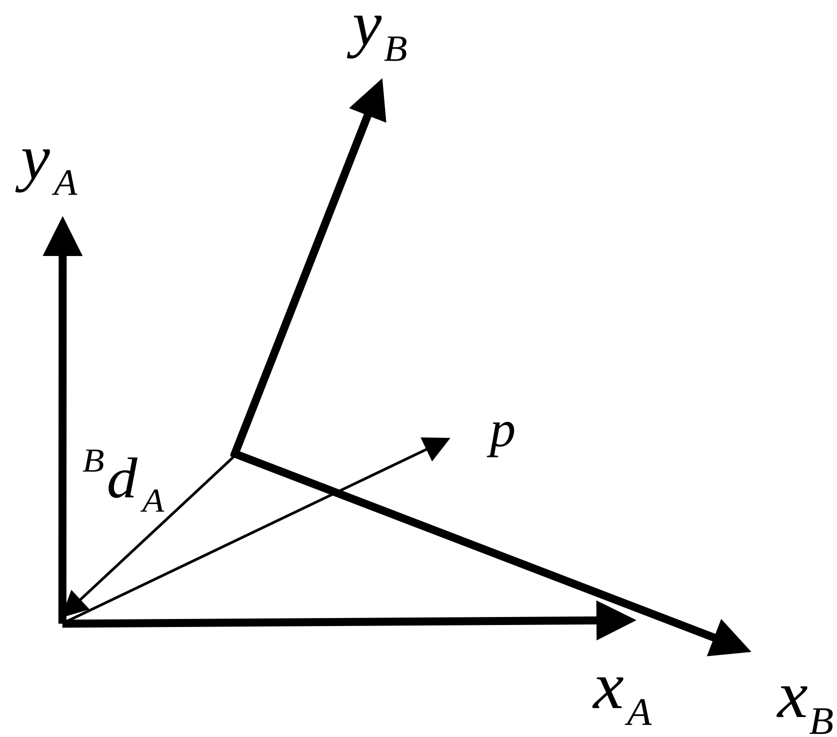
$$
{}^{A}R_{C} = {}^{A}R_{B}\,{}^{B}R_{C}
$$

Homogeneous transforms

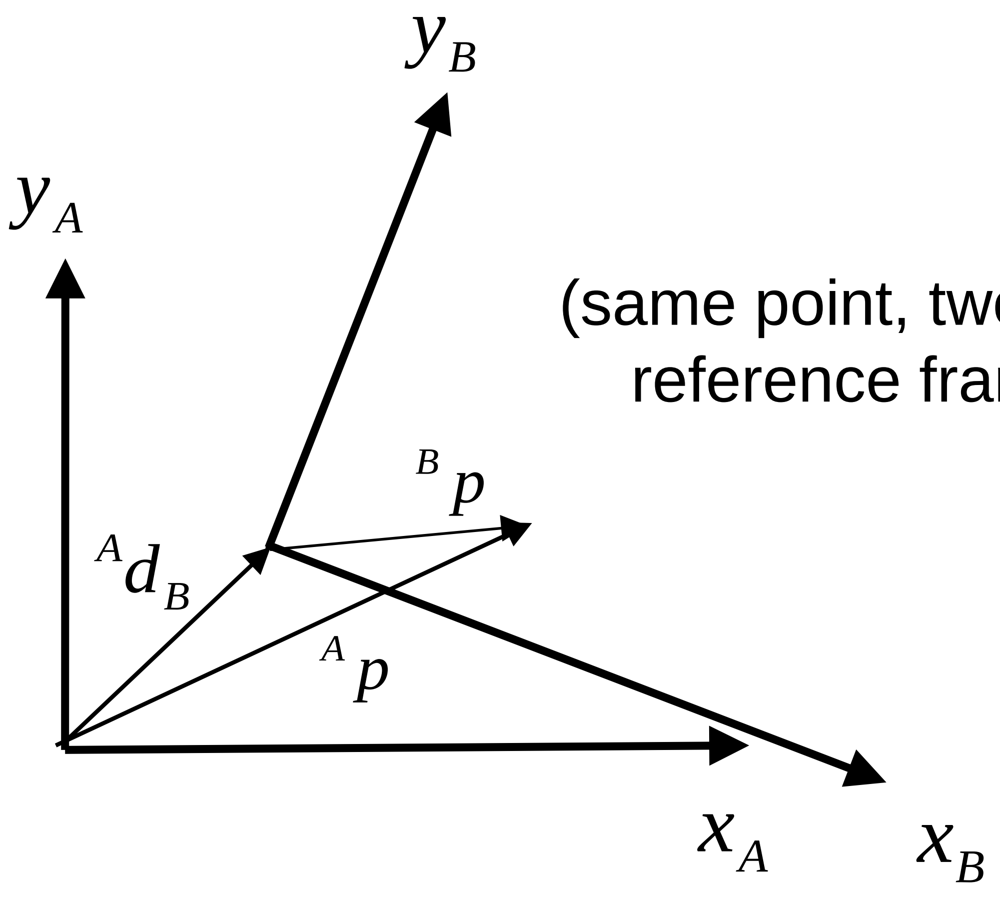
Rotation matrices assume that the origins of the two frames are co-located.

- What if they’re separated by a translation?





Homogeneous transform

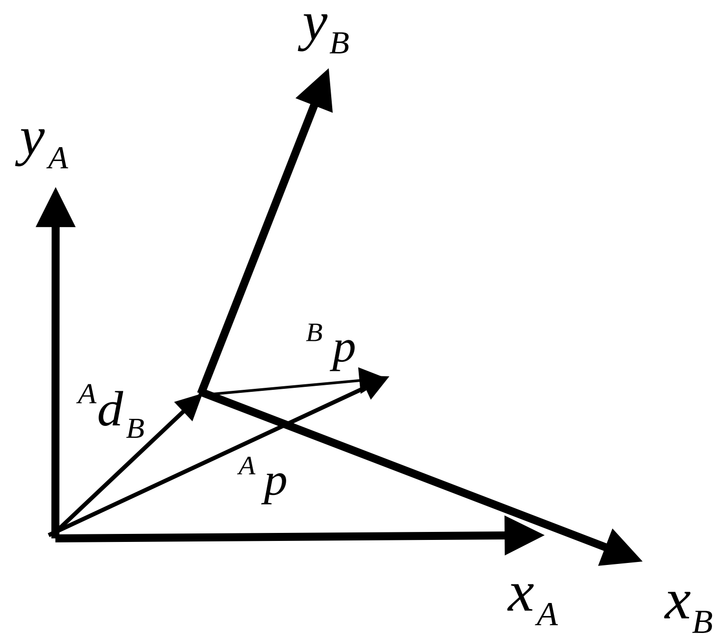


(same point, two reference frames)

$$^{A}p = {}^{A}R_B\, {}^{B}p + {}^{A}d_B$$

Homogeneous transform

$$ {}^{A}p = {}^{A}R_{B}\,{}^{B}p + {}^{A}d_{B} $$



$$ = \begin{pmatrix} {}^{A}R_{B} & {}^{A}d_{B} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{B}p \\ 1 \end{pmatrix} $$

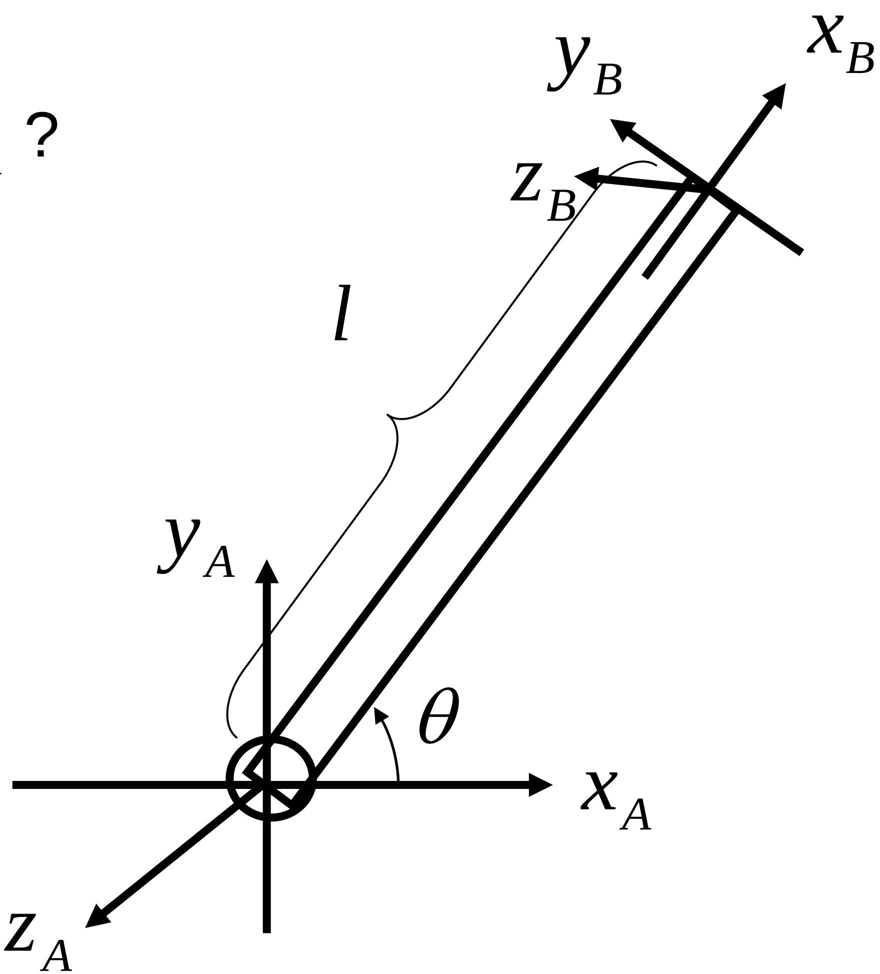
$$ = \begin{pmatrix} r_{11} & r_{12} & r_{13} & {}^{A}d_{x} \\ r_{21} & r_{22} & r_{23} & {}^{A}d_{y} \\ r_{31} & r_{32} & r_{33} & {}^{A}d_{z} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{B}p \\ 1 \end{pmatrix} = {}^{A}T_{B} \begin{pmatrix} {}^{B}p \\ 1 \end{pmatrix} $$

always zeros

always one

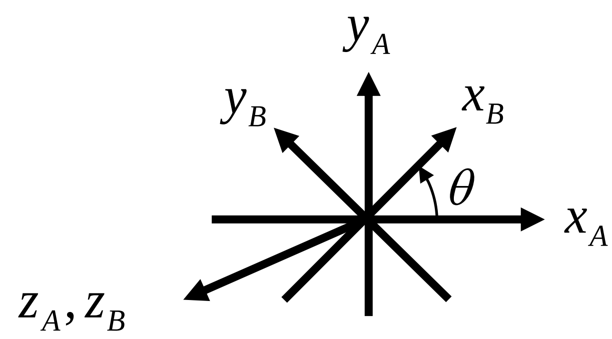
Example 1: homogeneous transforms

What's ${}^{B}T_{A}$?



Example 1: homogeneous transforms

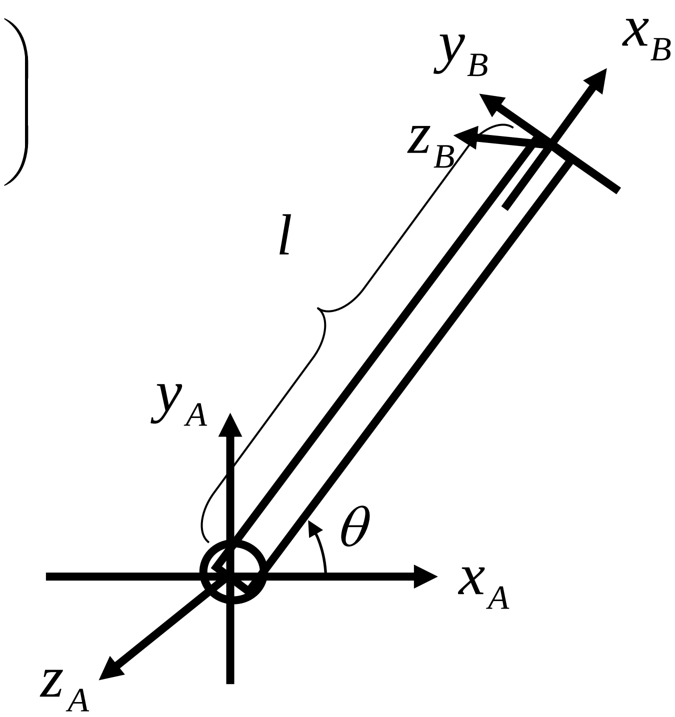
What's ${}^{B}T_{A}$?



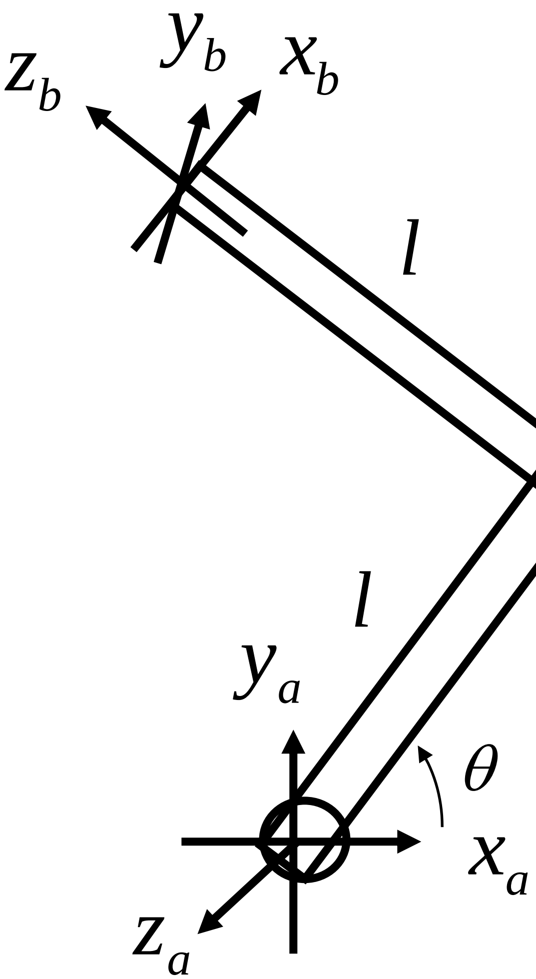
$$ {}^{A}R_{B} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{B}d_{A} = \begin{pmatrix} -l \\ 0 \\ 0 \end{pmatrix} \qquad {}^{B}T_{A} = \begin{pmatrix} {}^{B}R_{A} & {}^{B}d_{A} \\ 0 & 1 \end{pmatrix} $$

$$ {}^{B}T_{A} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 & -l \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$



Example 2: homogeneous transforms

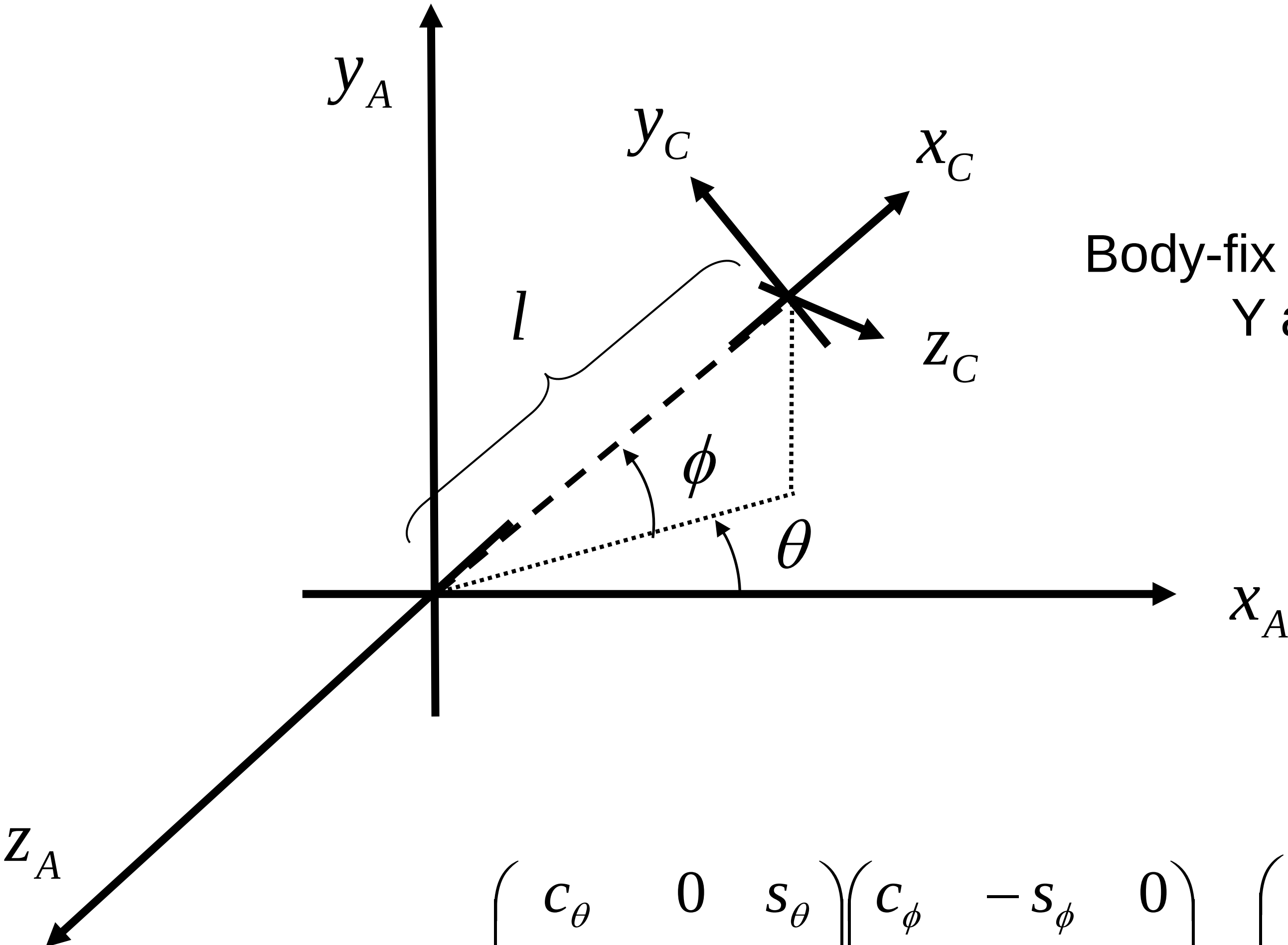


This arm rotates about the z_a axis.

What's ${}^aT_b$?

$$
{}^aT_b = \begin{bmatrix} c\theta & 0 & -s\theta & l\sqrt{2}c\left(\theta + \frac{\pi}{4}\right) \\ s\theta & 0 & c\theta & l\sqrt{2}s\left(\theta + \frac{\pi}{4}\right) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

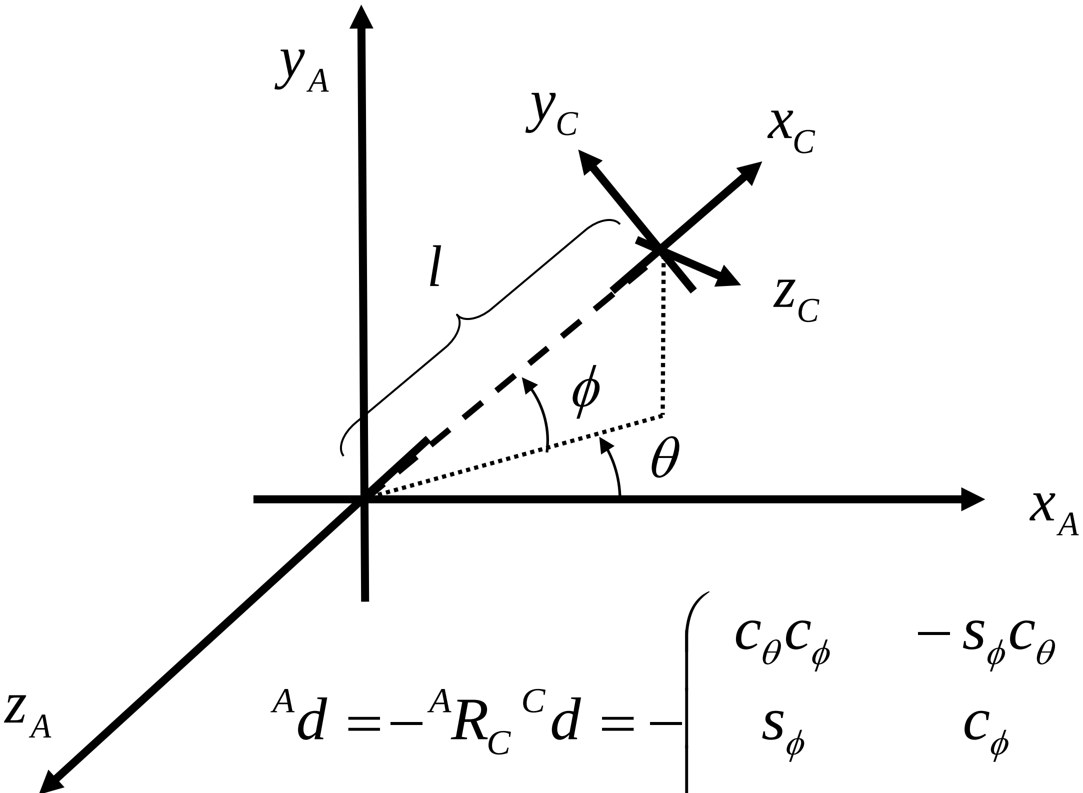
Example 3: homogeneous transforms



Body-fix rotations: first around Y and then around Z

$$ {}^{A}R_{C} = {}^{A}R_{B}\,{}^{B}R_{C} = \begin{pmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{pmatrix} \begin{pmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c_\theta c_\phi & -s_\phi c_\theta & s_\theta \\ s_\phi & c_\phi & 0 \\ -s_\theta c_\phi & s_\theta s_\phi & c_\theta \end{pmatrix} $$

Example 3: homogeneous transforms



$$^{C}d = \begin{pmatrix} -l \\ 0 \\ 0 \end{pmatrix}$$

$$^{A}d = -^{A}R_{C}\,^{C}d = -\begin{pmatrix} c_\theta c_\phi & -s_\phi c_\theta & s_\theta \\ s_\phi & c_\phi & 0 \\ -s_\theta c_\phi & s_\theta s_\phi & c_\theta \end{pmatrix}\begin{pmatrix} -l \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} lc_\theta c_\phi \\ ls_\phi \\ -ls_\theta c_\phi \end{pmatrix}$$

$$^{A}T_{C} = \begin{pmatrix} ^{A}R_{C} & ^{A}d \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} c_\theta c_\phi & -s_\phi c_\theta & s_\theta & lc_\theta c_\phi \\ s_\phi & c_\phi & 0 & ls_\phi \\ -s_\theta c_\phi & s_\theta s_\phi & c_\theta & -ls_\theta c_\phi \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Inverse of the Homogeneous transform

First, we derive it from the forward Homogeneous transform:

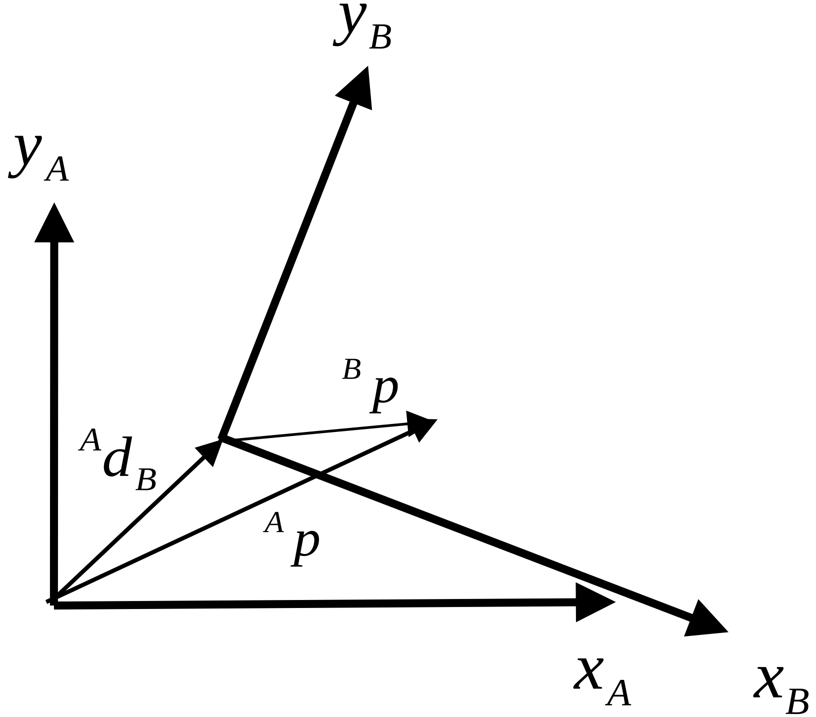
Inverse of the Homogeneous transform

$$ {}^{B}p = {}^{B}R_{A}\,{}^{A}p - {}^{A}d_{B} $$

$$ {}^{B}p = \begin{pmatrix} {}^{B}R_{A} & -{}^{A}d_{B} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A}p \\ 1 \end{pmatrix} $$

$$ {}^{B}p = {}^{B}T_{A} \begin{pmatrix} {}^{A}p \\ 1 \end{pmatrix} $$

$$ {}^{A}p = {}^{B}T_{A}^{-1} \begin{pmatrix} {}^{B}p \\ 1 \end{pmatrix} $$

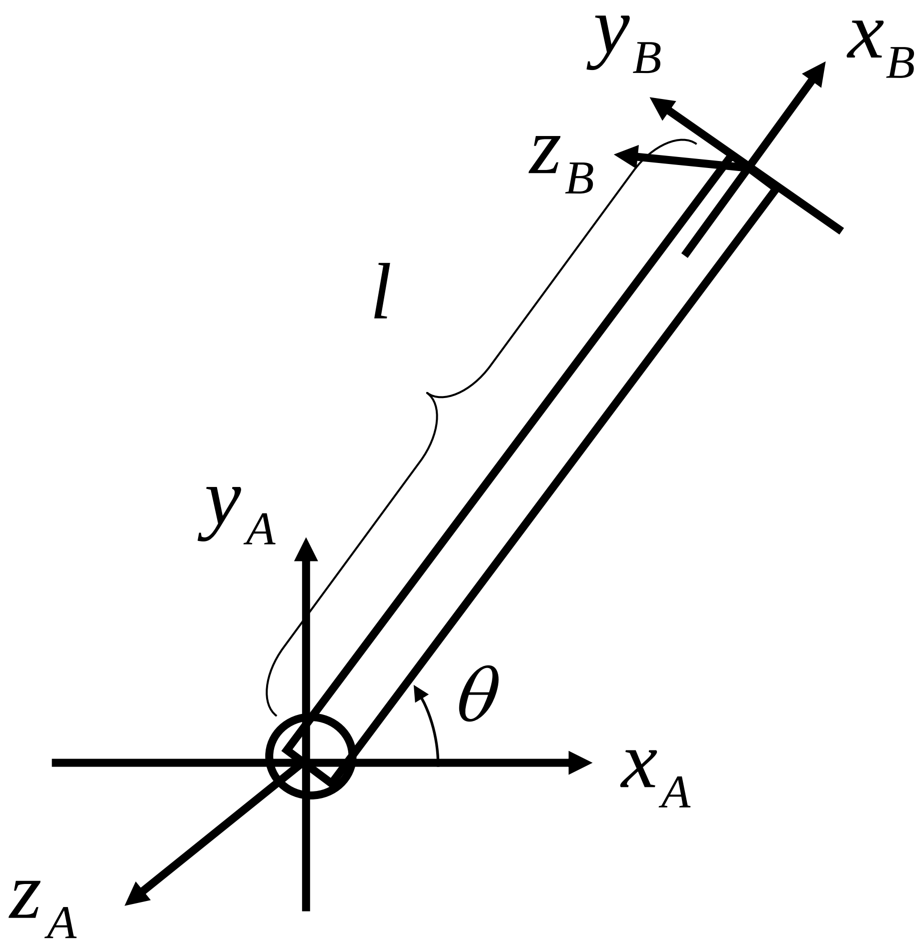


$$ {}^{B}p = {}^{B}R_{A}\,{}^{A}p + {}^{B}d_{A} $$

$$ {}^{A}p = {}^{B}R_{A}^{T}\left({}^{B}p - {}^{B}d_{A}\right) $$

$$ {}^{A}p = \begin{pmatrix} {}^{B}R_{A}^{T} & -{}^{B}R_{A}^{T}\,{}^{B}d_{A} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{B}p \\ 1 \end{pmatrix} $$

Example 1: homogeneous transform inverse



$$
{}^{B}T_{A} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 & -l \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
$$

What's ${}^{A}T_{B}$?

Example 1: homogeneous transform inverse

$$ {}^{B}T_{A}^{-1} = \begin{pmatrix} {}^{B}R_{A}^{T} & -{}^{B}R_{A}^{T}\,{}^{B}d_{A} \\ 0 & 1 \end{pmatrix} $$

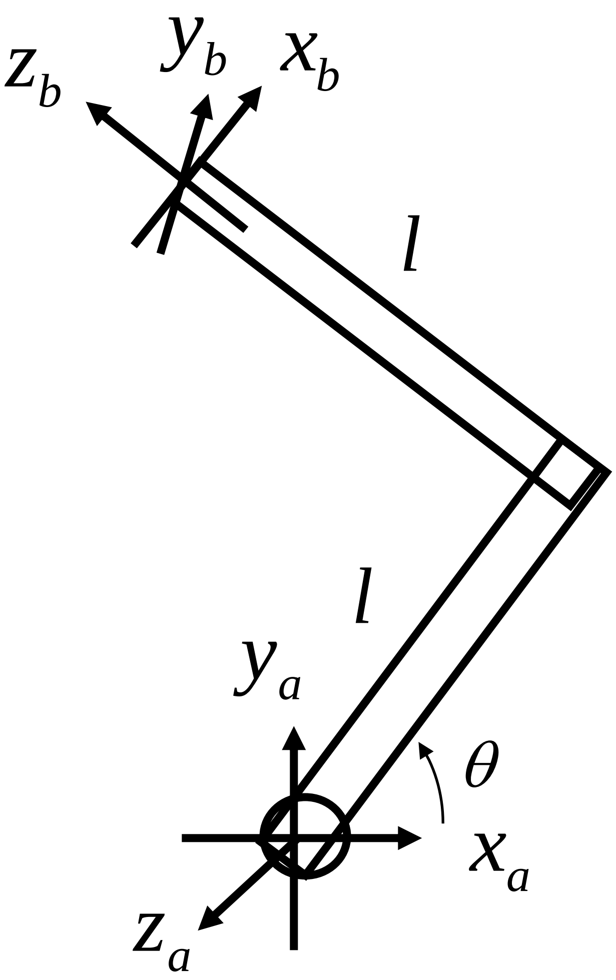
$$ {}^{A}R_{B} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{B}d_{A} = \begin{pmatrix} -l \\ 0 \\ 0 \end{pmatrix} $$

$$ -{}^{A}R_{B}\,{}^{B}d_{A} = -\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} -l \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} l\cos(\theta) \\ l\sin(\theta) \\ 0 \end{pmatrix} $$

$$ {}^{B}T_{A}^{-1} = {}^{A}T_{B} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & l\cos(\theta) \\ \sin(\theta) & \cos(\theta) & 0 & l\sin(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Example 2: homogeneous transform inverse



$$ {}^{a}T_b = \begin{bmatrix} c\theta & 0 & -s\theta & l\sqrt{2}c\left(\theta+\frac{\pi}{4}\right) \\ s\theta & 0 & c\theta & l\sqrt{2}s\left(\theta+\frac{\pi}{4}\right) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

What's ${}^{b}T_a$?

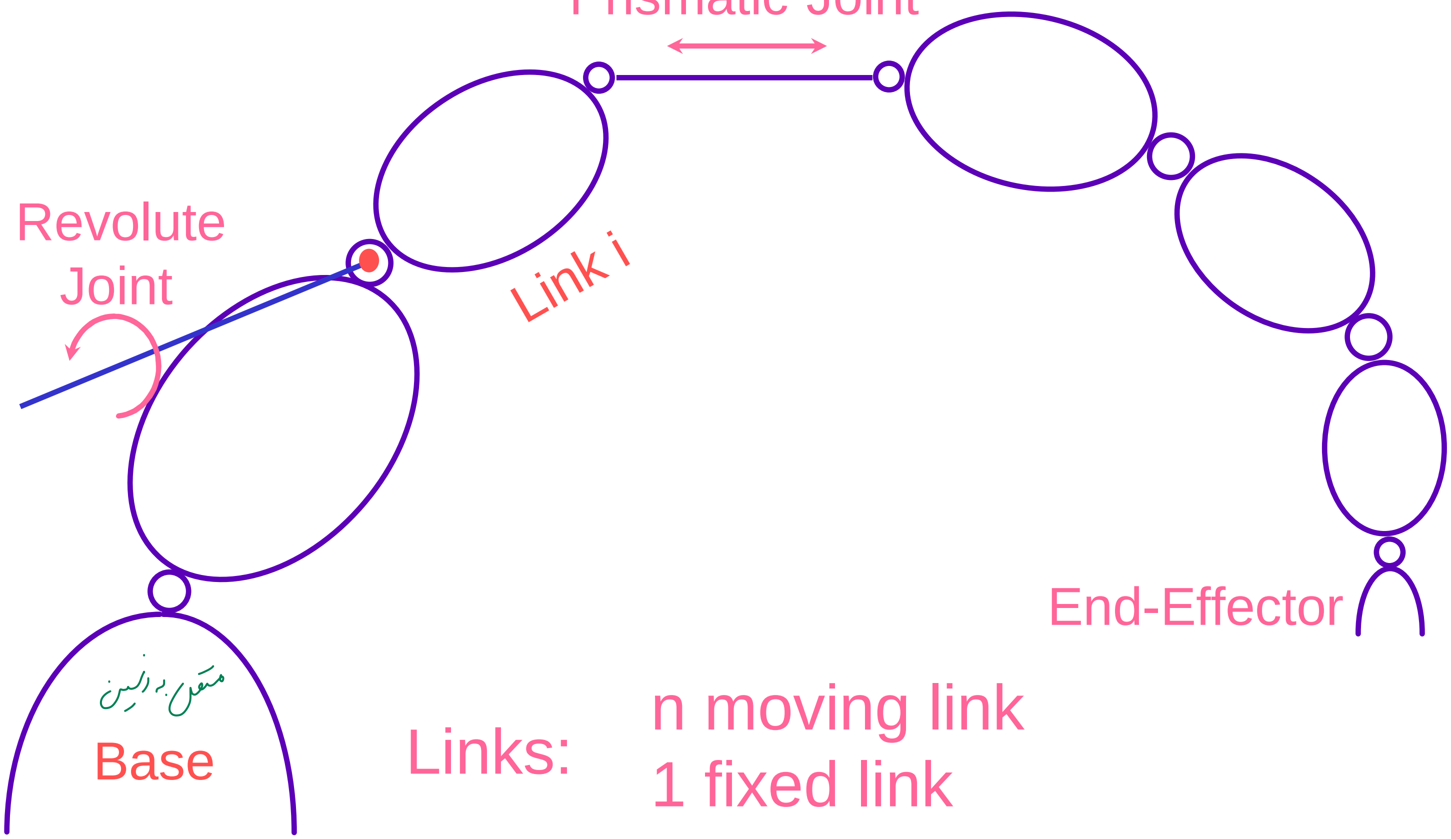
$$ {}^{b}T_a = \begin{bmatrix} c\theta & s\theta & 0 & -l\sqrt{2}\left(c_\theta c_{\theta+\frac{\pi}{4}} + s_\theta s_{\theta+\frac{\pi}{4}}\right) \\ 0 & 0 & -1 & 0 \\ -s\theta & c\theta & 0 & l\sqrt{2}\left(s_\theta c_{\theta+\frac{\pi}{4}} - c_\theta s_{\theta+\frac{\pi}{4}}\right) \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Kinematics

Spatial Descriptions

- Task Description توصیف کار
- Transformations تبدیل ها
- Representations

Manipulator



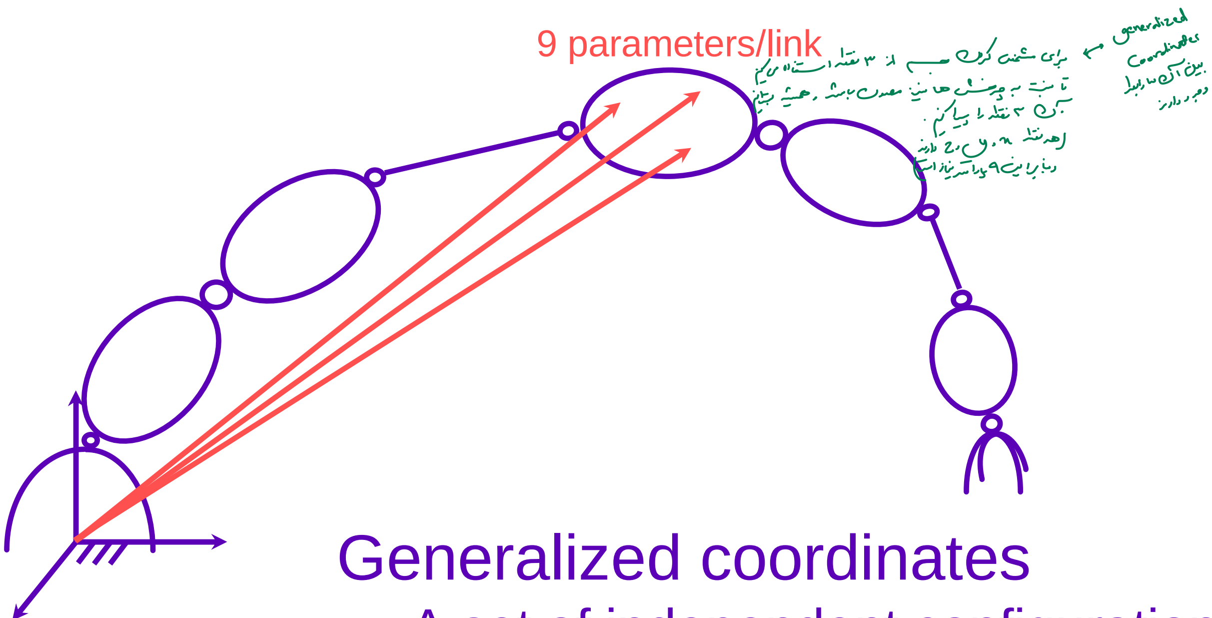
Links: n moving link
1 fixed link

Joints: Revolute (1 DOF)

Prismatic (1 DOF)

Configuration Parameters

A set of position parameters that describes the full configuration of the system.



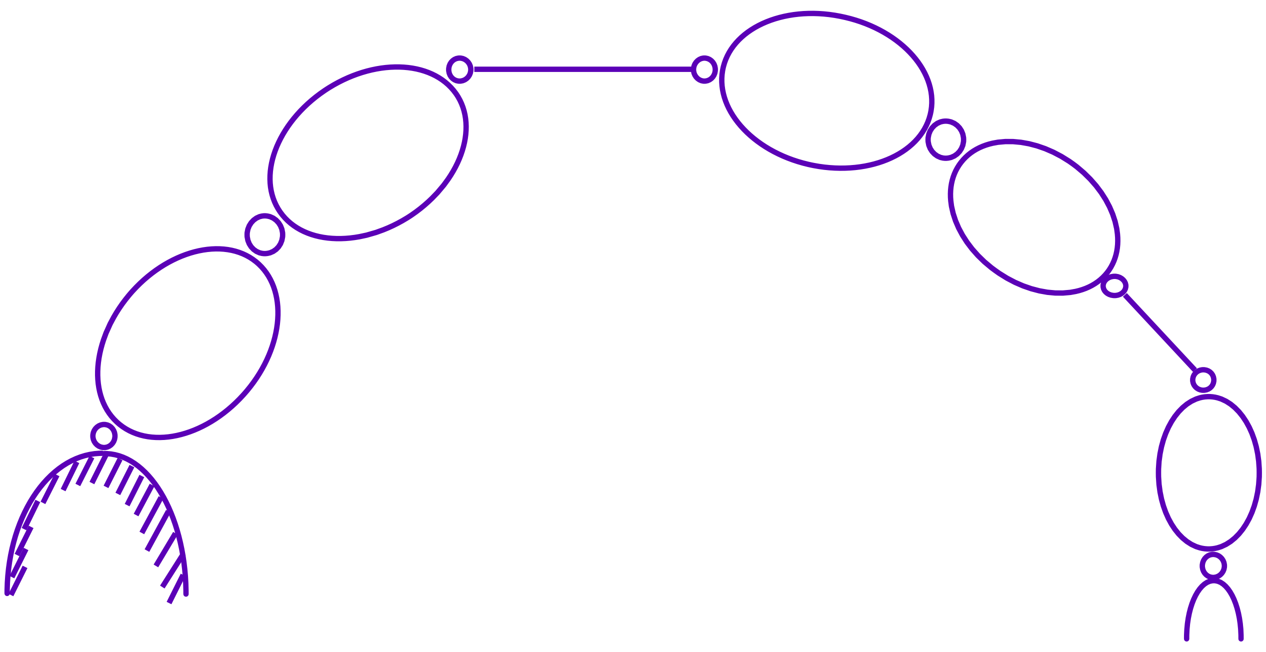
Generalized coordinates

A set of independent configuration parameters

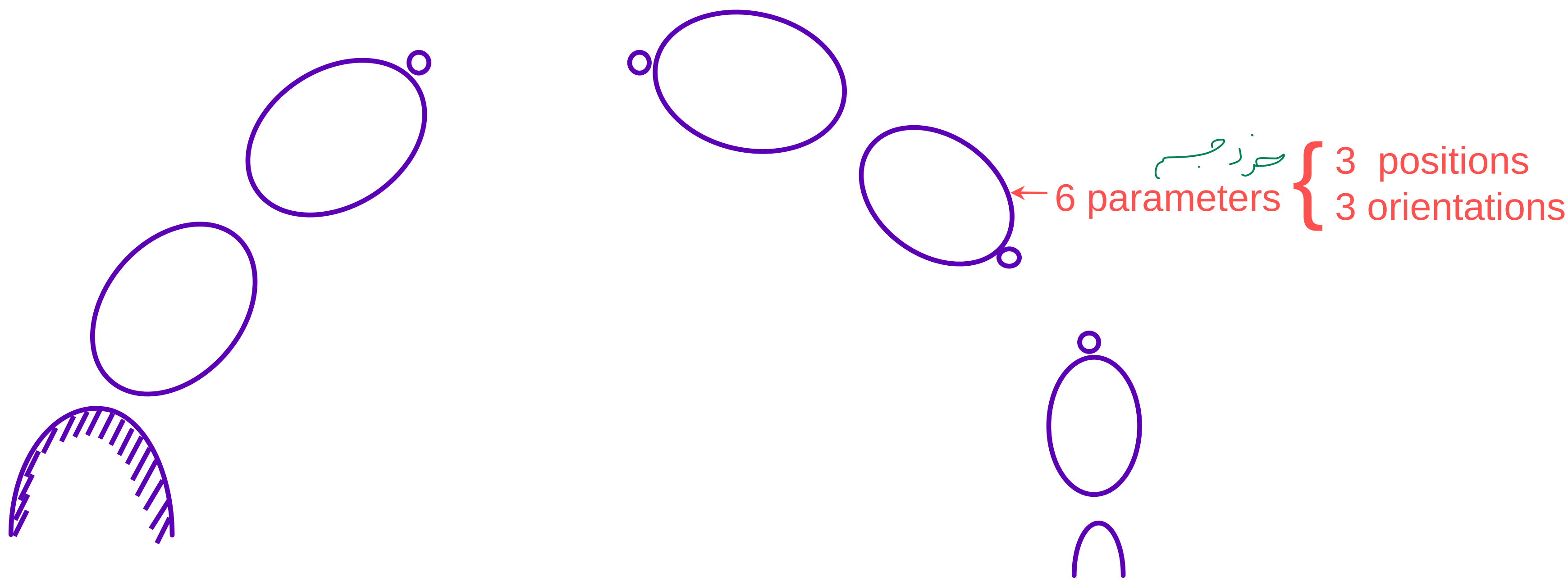
Degrees of Freedom

Number of generalized coordinates

Generalized Coordinates

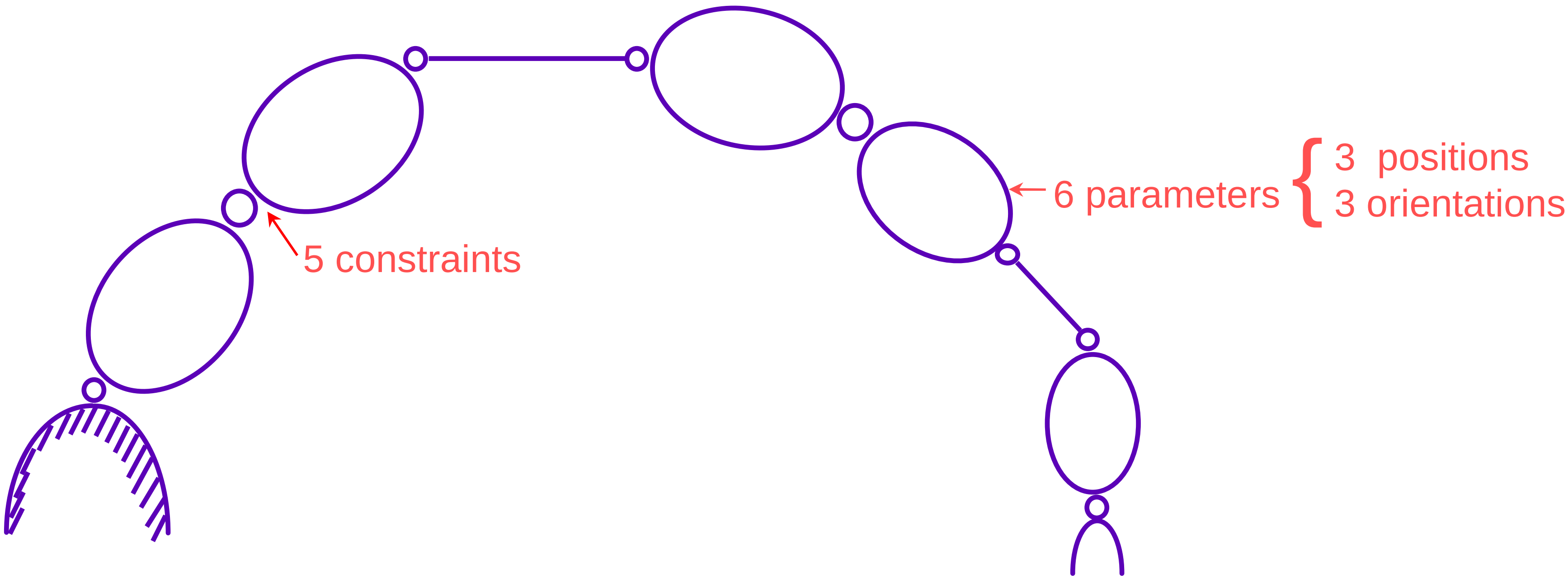


Generalized Coordinates



n moving links: 6n parameters

Generalized Coordinates



n moving links: 6n parameters

n 1 d.o.f. joints: 5n constraints

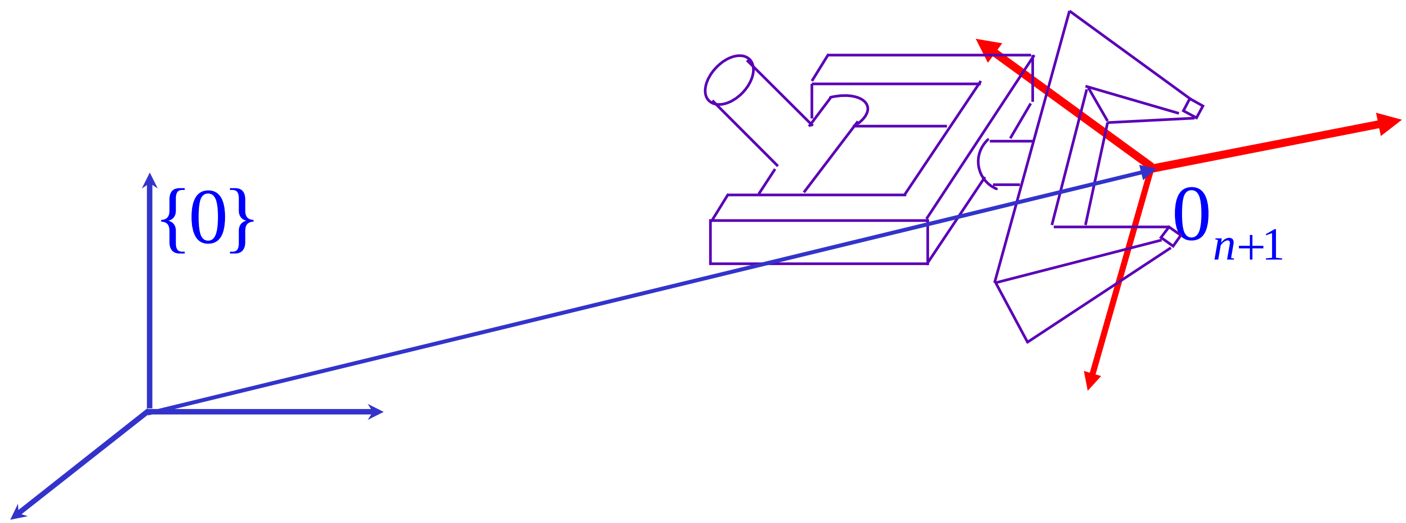
$\boxed{\text{d.o.f. (system): } 6n - 5n = n}$

هر عضو
۵ حرکتش حذف می‌شود
و تنها ۱ درجه آزادی
می‌ماند.

هر اتصال پس از اتصال های قبلی به همان تعداد درجه آزادی بدست آورده است و بنابراین اگر ۶ link داشته باشیم به ۶ درجه آزادی احتیاج داریم.

بیشترین آزادی

End-Efector Configuration Parameters



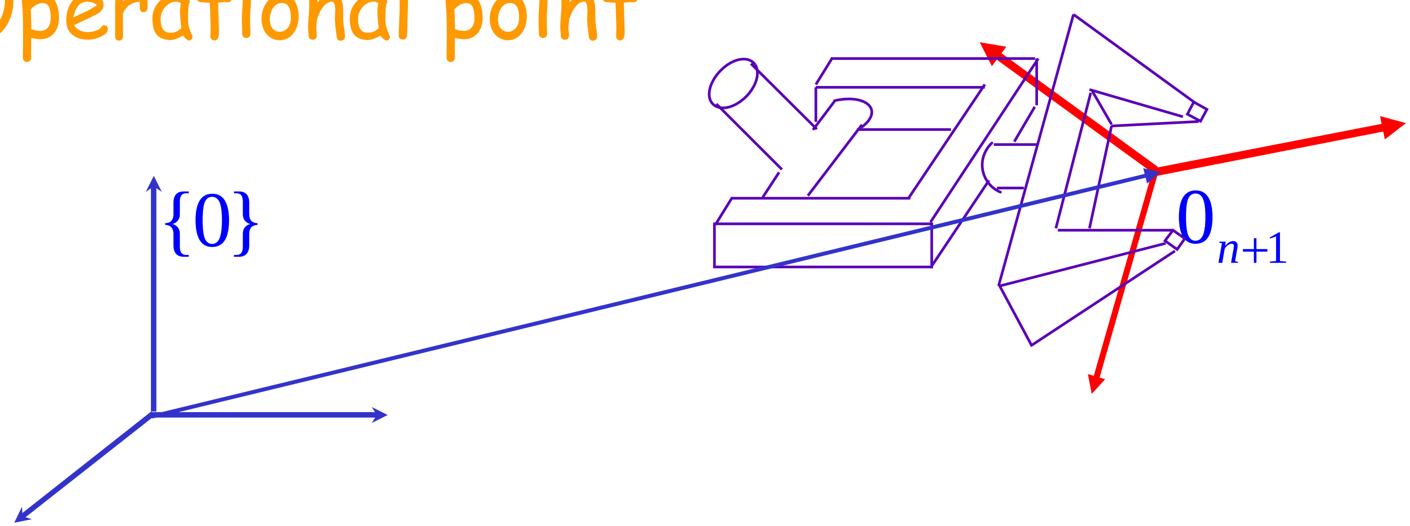
A set of m parameters:

$$(x_1, x_2, x_3, \ldots, x_m)$$

that completely specifies the end-effector $\underbrace{\text{position and orientation}}_{\text{pos}}$ with respect to $\{0\}$

Operational Coordinates

O_{n+1} : Operational point

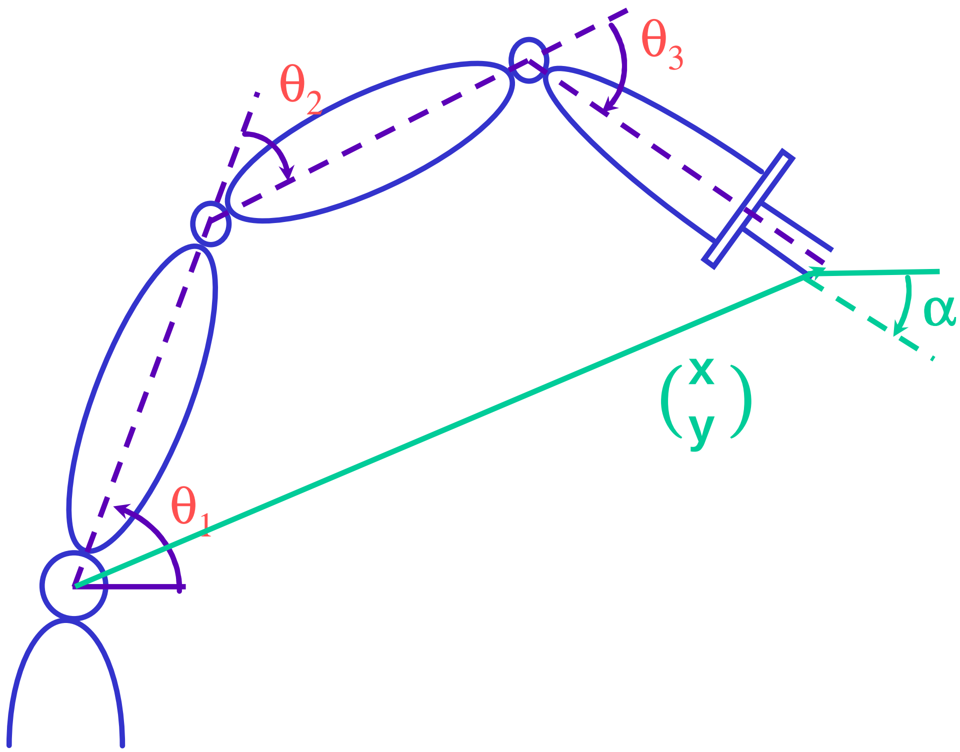


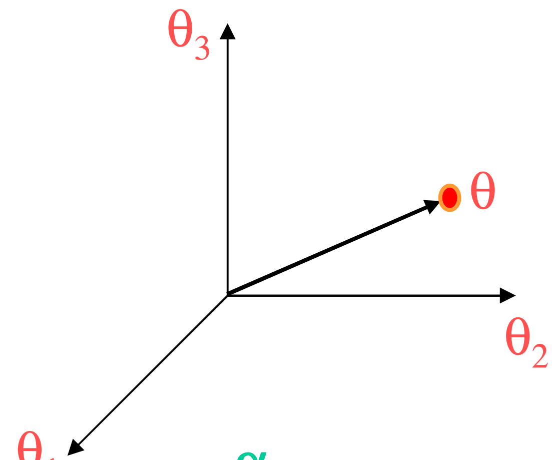
A set $x_1, x_2, \ldots, x_{m_0}$ [illegible] این نقاط بهم وابسته نباشند

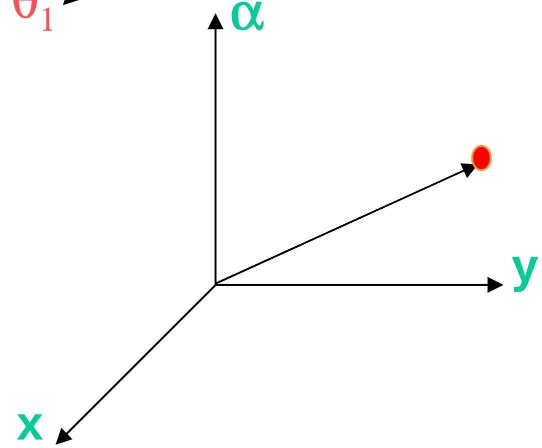
of m_0 independent configuration parameters

m_0 : number of degrees of freedom of the end-effector.

Joint Coordinates → Joint Space







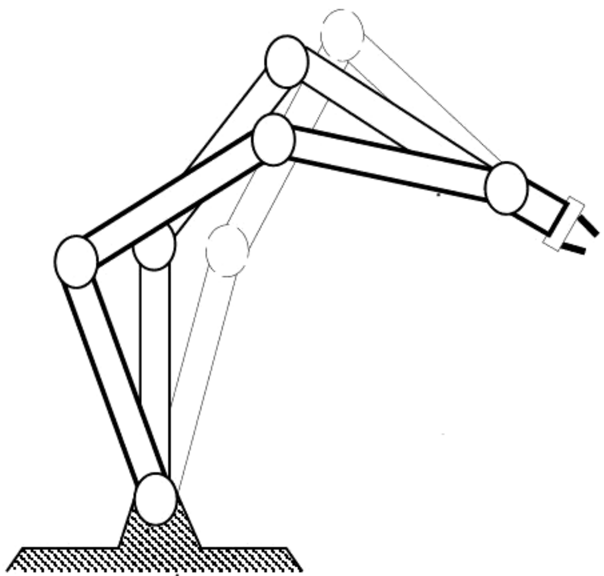
Operational Coordinates → Operational Space

می‌گیم در فضای اصلی کجا باشه داشتار دارم ببینم در فضای مال ربات کجاست.

این مفهوم همان forward kinematic است.

نقطه انتهایی

Redundancy



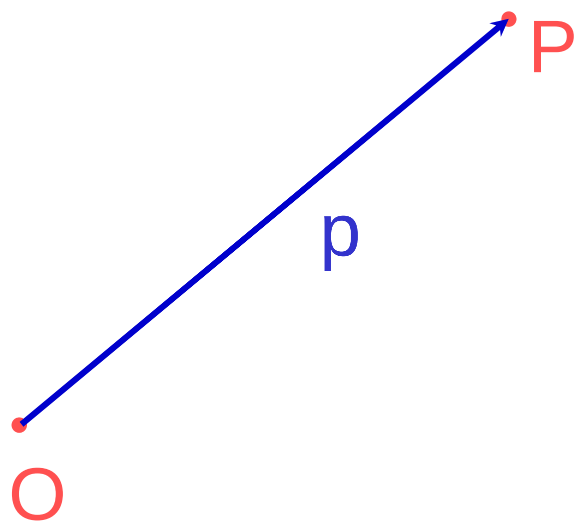
A robot is said to be redundant if

$$\boxed{n > m_0}$$

Degrees of redundancy: $n - m_0$

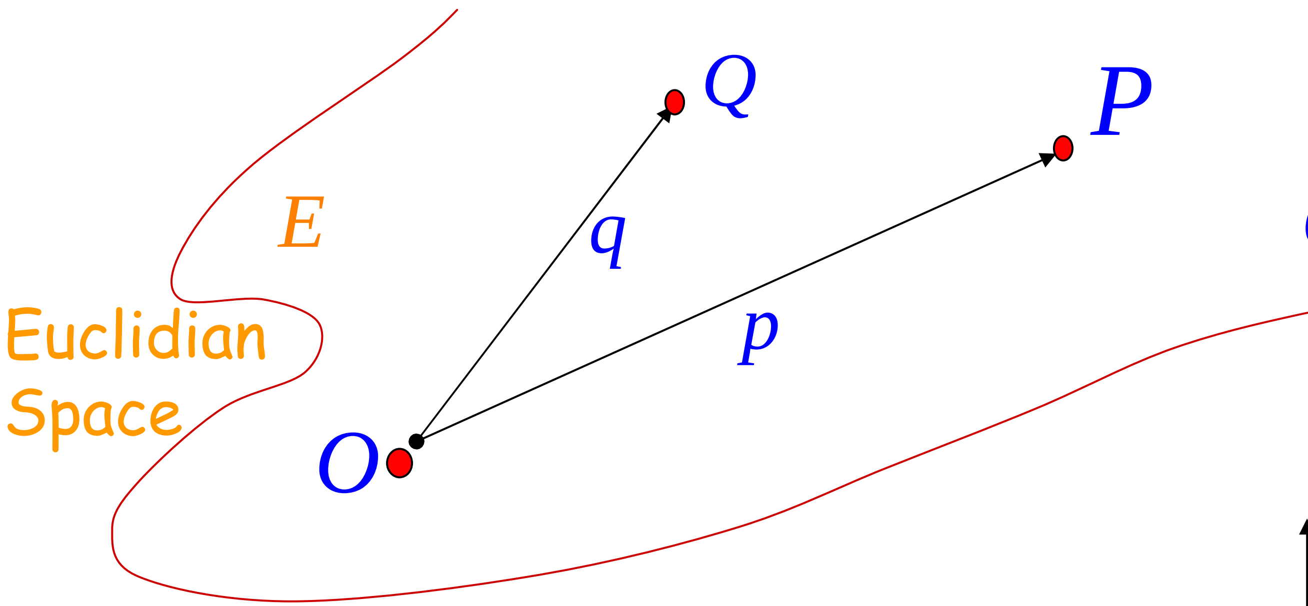
بیشتر از ۶ درجه آزادی و ۷ تا link
باعث آزادی عمل افزایش پیدا کند
برای دست انسان به ازای یک نقطه بی‌نهایت حالت داریم
به دور زدن موانع هم خود را کمک می‌کند درست هست و این عدد را می‌توانیم به
نقطه مورد نظر برسیم.

Position of a Point

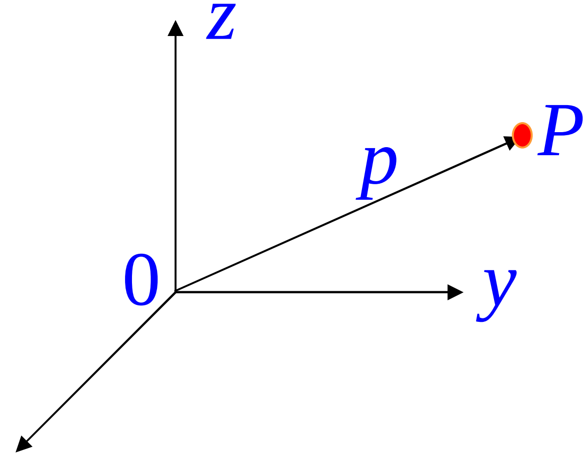


With respect to a fixed origin O, the position of a point P is described by the vector OP or simply by p.

Rigid Body Configuration

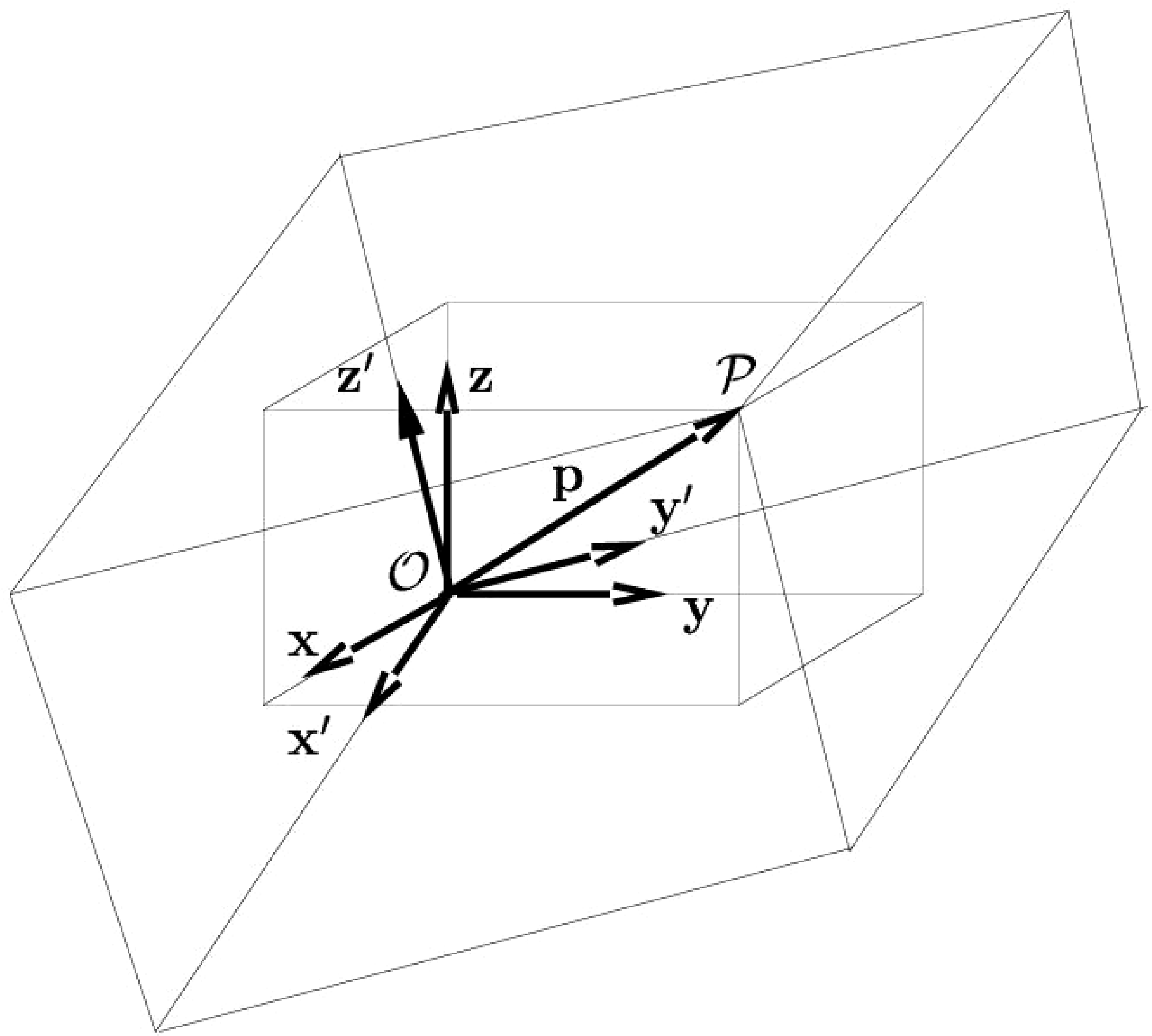


$$p = \overrightarrow{OP}$$
$$q = \overrightarrow{OQ}$$



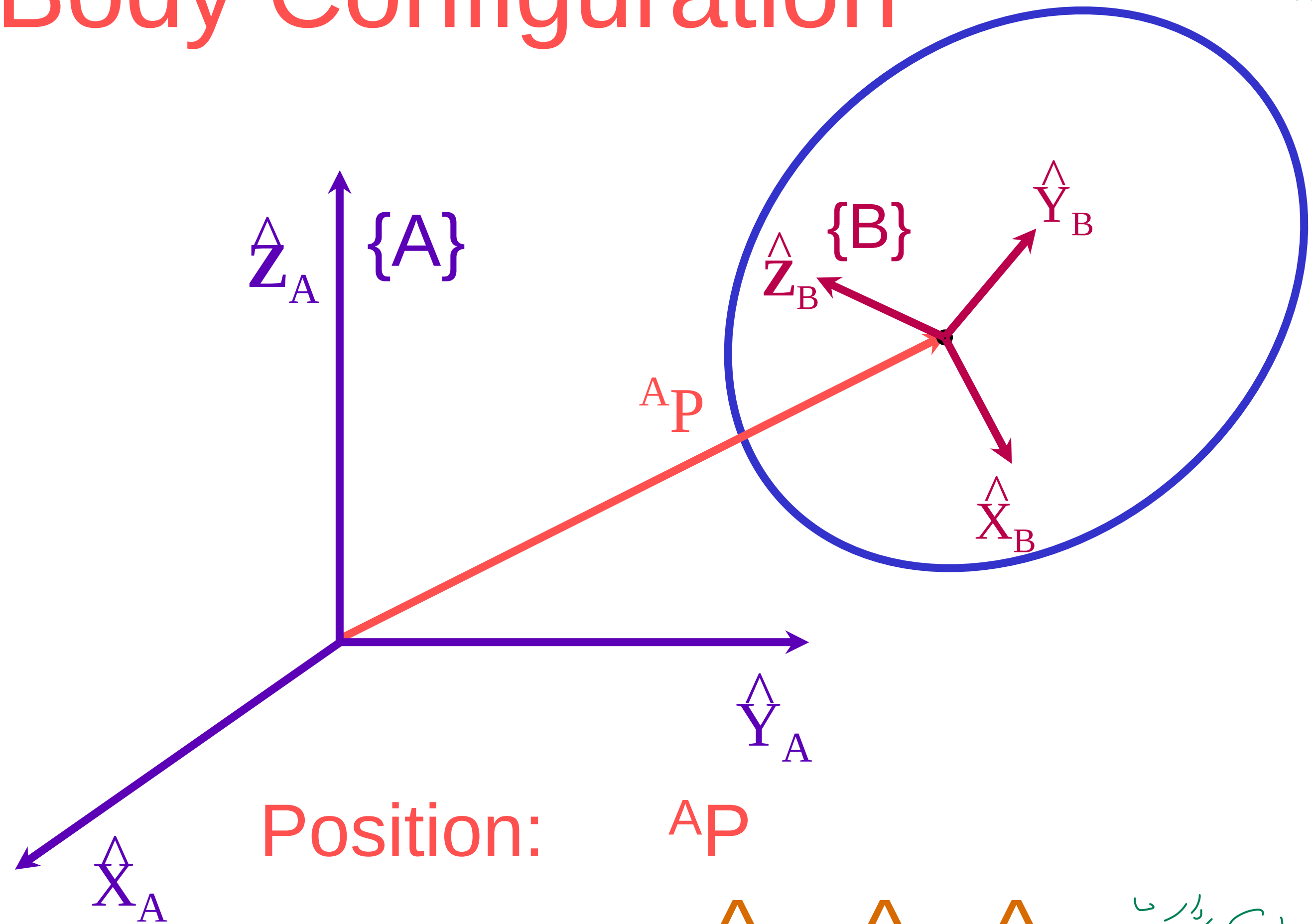
Cartesian Frame

Coordinate Frames



Rigid Body Configuration

که استاندارد مختصات ثابت هستند را توصیف کنم



Position: ^{A}P

Orientation: $\{{}^{A}\hat{X}_B, {}^{A}\hat{Y}_B, {}^{A}\hat{Z}_B\}$

بیان بردارها در A → ماتریس ۳×۳

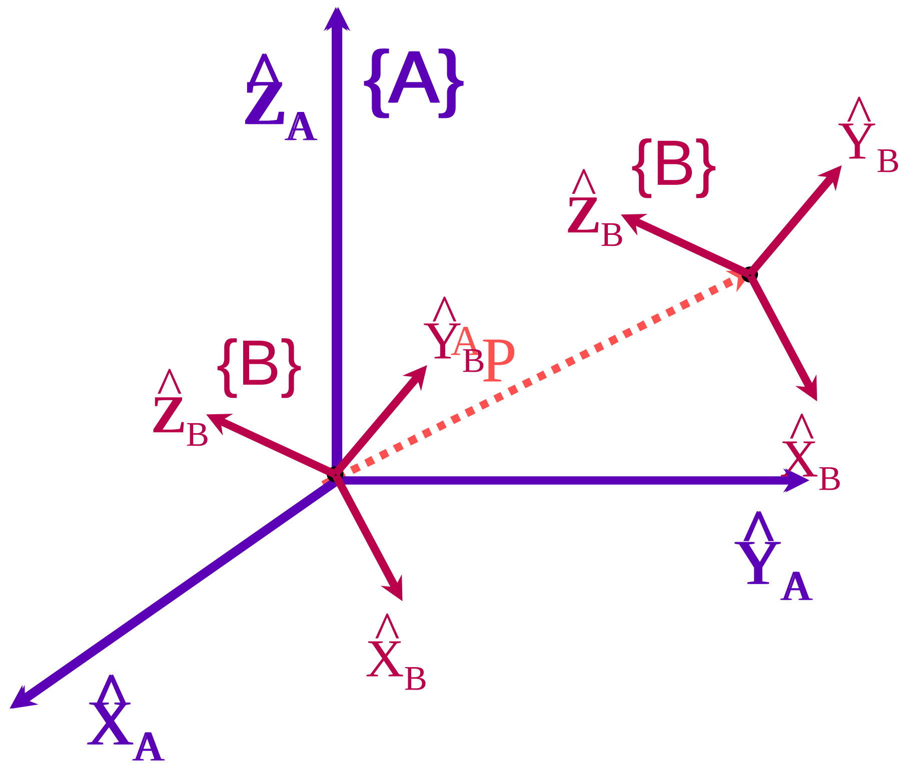
Describes rotations of {B} with respect to {A}

➡ **Rotation Matrix**

Rotation Matrix

$$ {}^{A}_{B}R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} $$

$$ \boxed{{}^{A}\hat{X}_B = {}^{A}_{B}R \, {}^{B}\hat{X}_B} $$

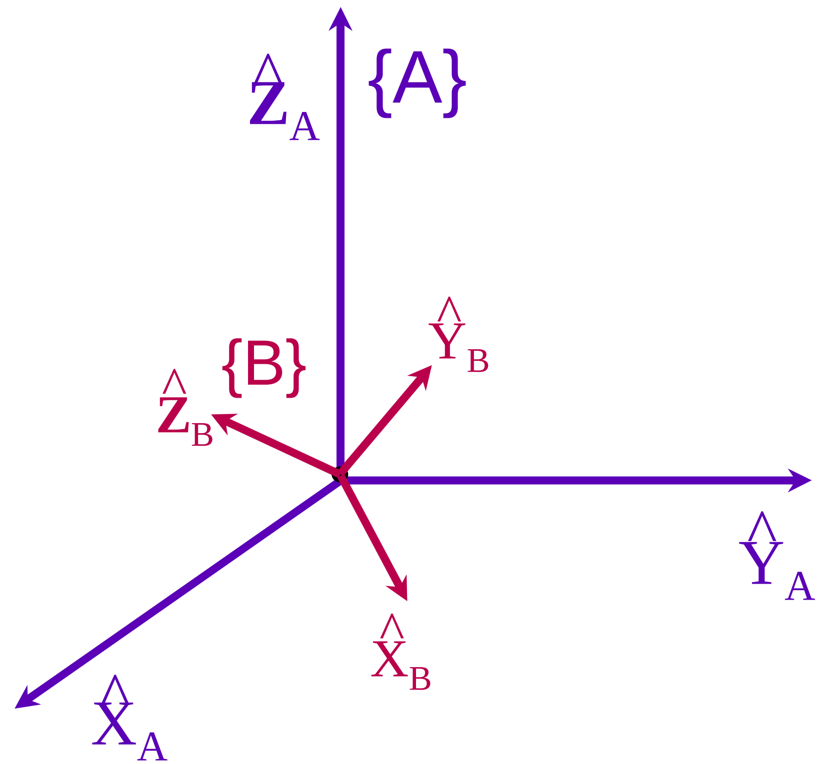


$$ {}^{A}\hat{X}_B = {}^{A}_{B}R \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad {}^{A}\hat{Y}_B = {}^{A}_{B}R \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad {}^{A}\hat{Z}_B = {}^{A}_{B}R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow $$

$$ {}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_B & {}^{A}\hat{Y}_B & {}^{A}\hat{Z}_B \end{bmatrix} $$

Rotation Matrix

$$ {}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_{B} & {}^{A}\hat{Y}_{B} & {}^{A}\hat{Z}_{B} \end{bmatrix} $$



Dot Product

$$ {}^{A}\hat{X}_{B} = \begin{bmatrix} \hat{X}_{B}\cdot\hat{X}_{A} \\ \hat{X}_{B}\cdot\hat{Y}_{A} \\ \hat{X}_{B}\cdot\hat{Z}_{A} \end{bmatrix} $$

تصویرها

$$ \boxed{{}^{B}X_{A}^{T}} $$

$$ {}^{A}_{B}R = \begin{bmatrix} \hat{X}_{B}\cdot\hat{X}_{A} & \hat{Y}_{B}\cdot\hat{X}_{A} & \hat{Z}_{B}\cdot\hat{X}_{A} \\ \hat{X}_{B}\cdot\hat{Y}_{A} & \hat{Y}_{B}\cdot\hat{Y}_{A} & \hat{Z}_{B}\cdot\hat{Y}_{A} \\ \hat{X}_{B}\cdot\hat{Z}_{A} & \hat{Y}_{B}\cdot\hat{Z}_{A} & \hat{Z}_{B}\cdot\hat{Z}_{A} \end{bmatrix} $$

Rotation Matrix

$$ {}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_{B} & {}^{A}\hat{Y}_{B} & {}^{A}\hat{Z}_{B} \end{bmatrix} = \begin{bmatrix} {}^{B}\hat{X}_{A}^{T} \\ {}^{B}\hat{Y}_{A}^{T} \\ {}^{B}\hat{Z}_{A}^{T} \end{bmatrix} = \begin{bmatrix} {}^{B}\hat{X}_{A} & {}^{B}\hat{Y}_{A} & {}^{B}\hat{Z}_{A} \end{bmatrix}^{T} = {}^{B}_{A}R^{T} $$

$$ {}^{A}_{B}R = {}^{B}_{A}R^{T} $$

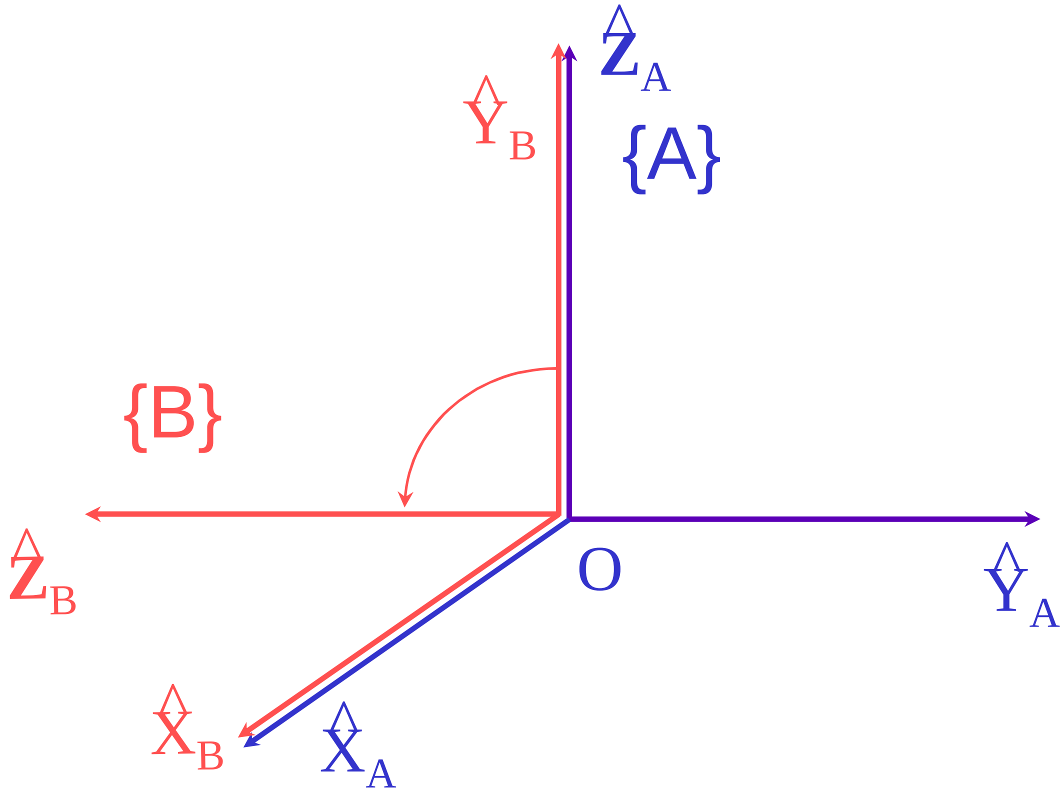
Inverse of Rotation Matrices

$$ {}^{A}_{B}R^{-1} = {}^{B}_{A}R = {}^{A}_{B}R^{T} $$

$$ \boxed{{}^{A}_{B}R^{-1} = {}^{A}_{B}R^{T}} $$

Orthonormal Matrix

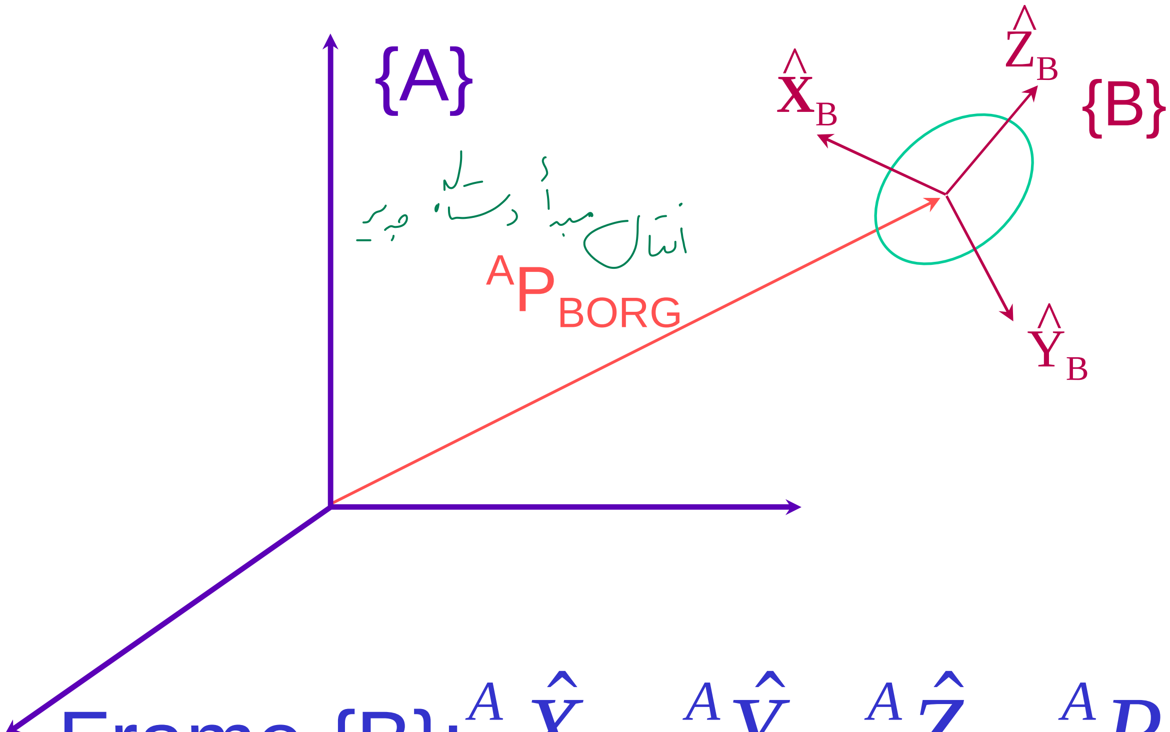
Example



$$
{}^A_B R = \begin{array}{ccc}
\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} & \begin{array}{l} \leftarrow {}^B\hat{X}_A^T \\ \leftarrow {}^B\hat{Y}_A^T \\ \leftarrow {}^B\hat{Z}_A^T \end{array} \\
\begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ {}^A\hat{X}_B & {}^A\hat{Y}_B & {}^A\hat{Z}_B \end{array} &
\end{array}
$$

Description of a Frame

with respect to reference frame



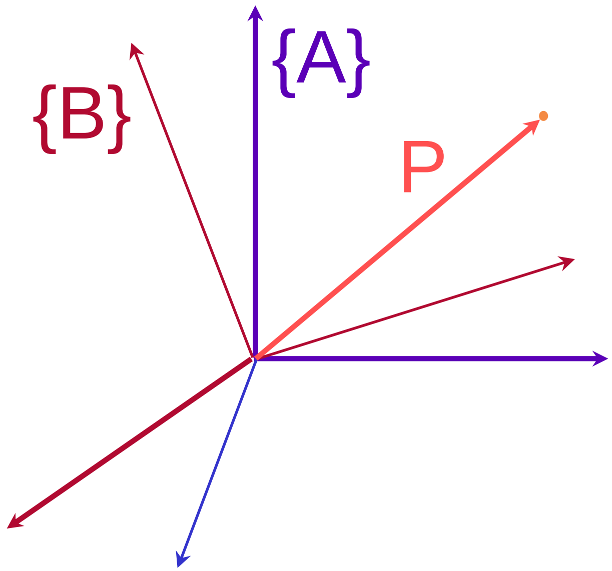
Frame {B}: $^{A}\hat{X}_{B},\ ^{A}\hat{Y}_{B},\ ^{A}\hat{Z}_{B},\ ^{A}P_{Borg}$

$$\{B\} = \left\{ {}^{A}_{B}R \quad {}^{A}P_{Borg} \right\}$$

Mapping

changing descriptions from frame to frame

Rotations



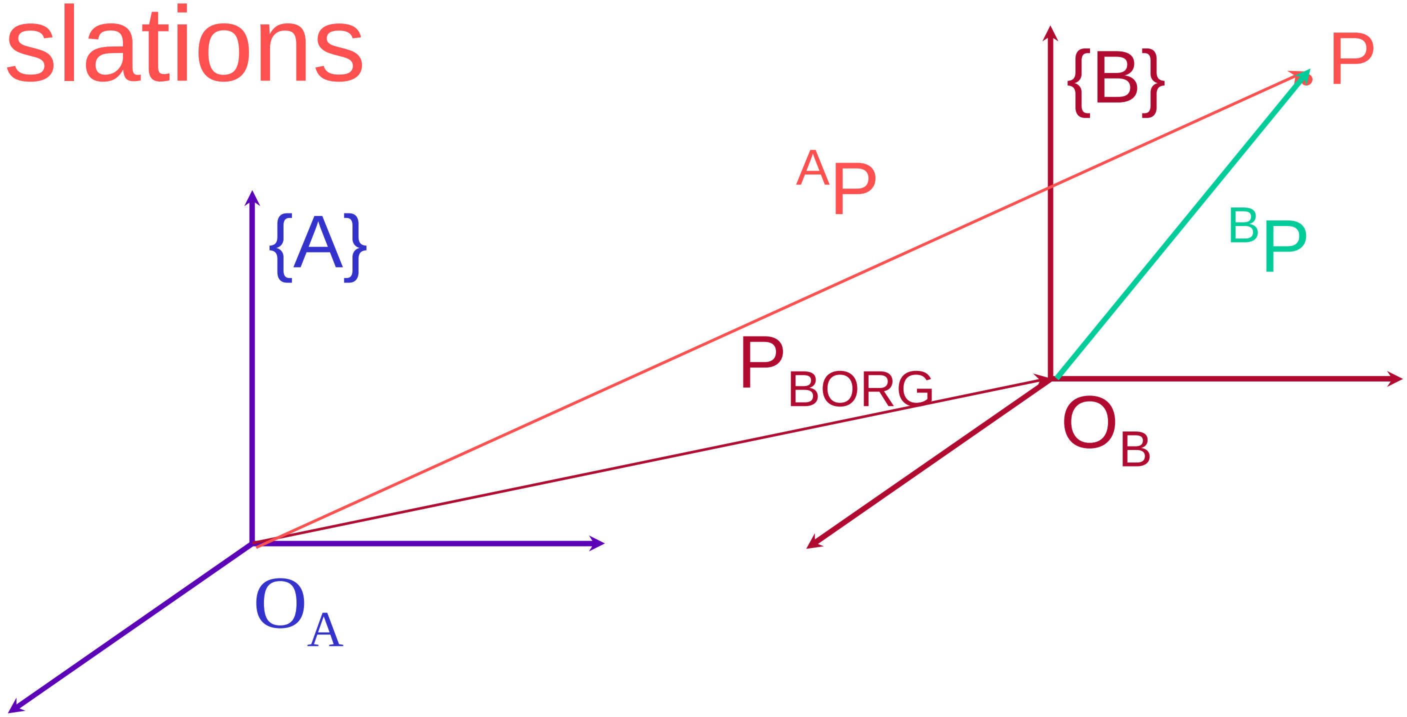
If P is given in {B}: ^{B}P

$$^{A}P = \begin{pmatrix} {}^{B}\hat{X}_A \cdot {}^{B}P \\ {}^{B}\hat{Y}_A \cdot {}^{B}P \\ {}^{B}\hat{Z}_A \cdot {}^{B}P \end{pmatrix} = \begin{pmatrix} {}^{B}\hat{X}_A^T \\ {}^{B}\hat{Y}_A^T \\ {}^{B}\hat{Z}_A^T \end{pmatrix} \cdot {}^{B}P$$

$$\Downarrow$$

$$^{A}P = {}^{A}_{B}R \; {}^{B}P$$

Translations



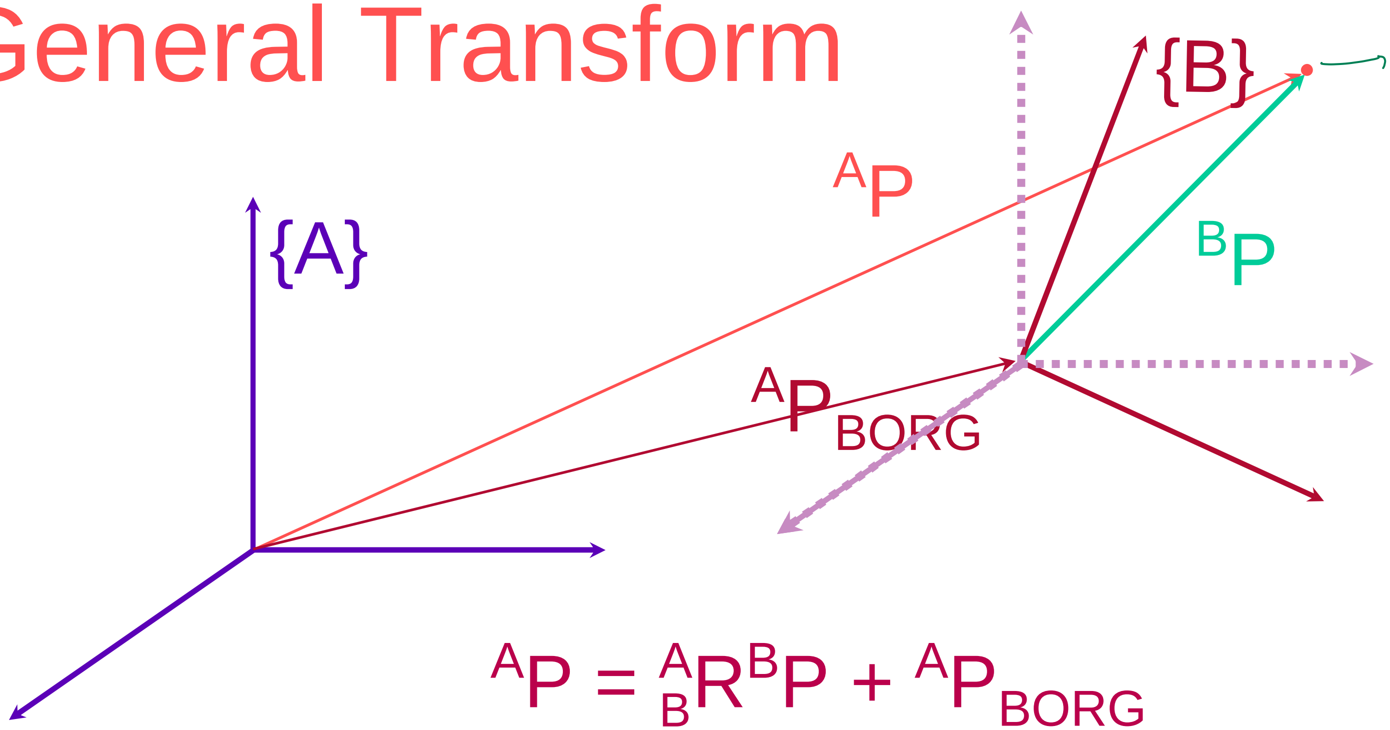
changing the position description of a point P

$\overrightarrow{O_BP} \Rightarrow \overrightarrow{O_AP}$ **(Two different vectors)**

$P_{BORG}: \quad P_{O_B} \Rightarrow P_{O_A}$

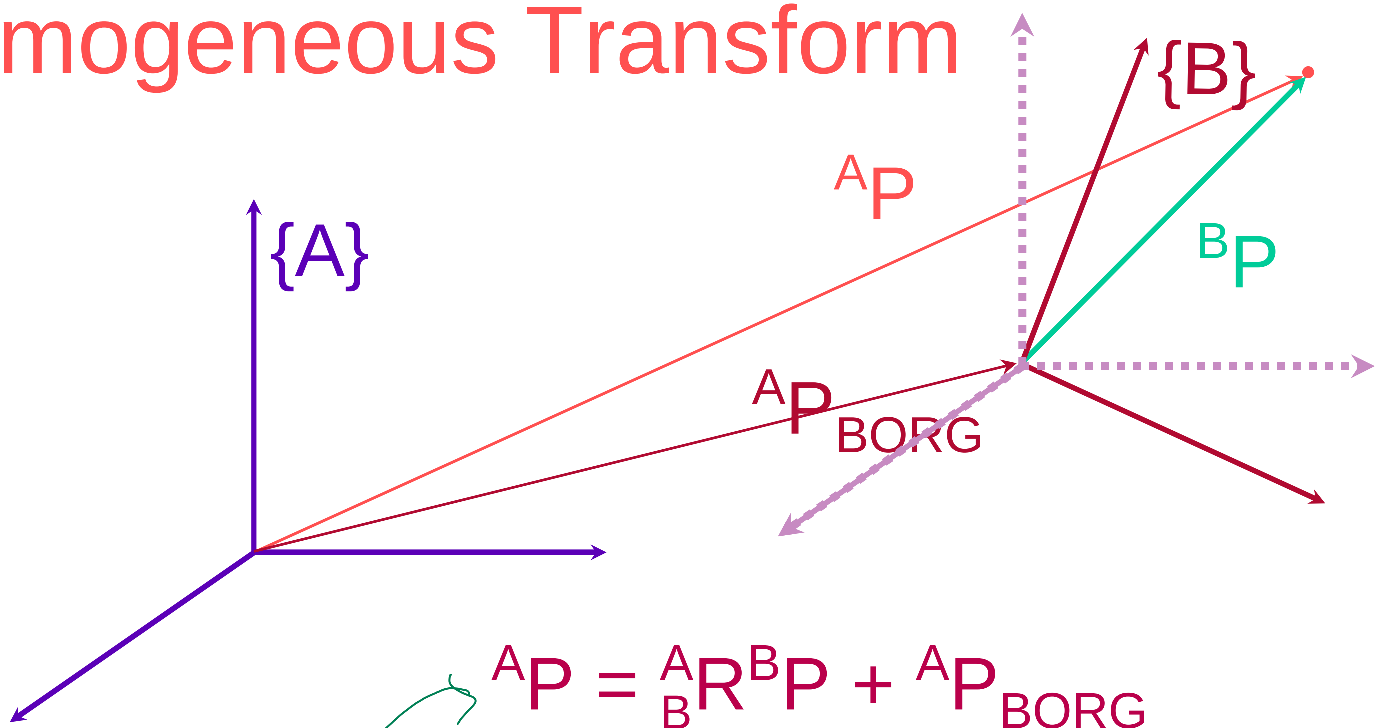
$$\boxed{{}^AP_{O_A} = {}^A\underset{\overset{\parallel}{B}}{P_{O_B}} + {}^AP_{BORG}}$$

General Transform



$$^{A}P = {}^{A}_{B}R\,{}^{B}P + {}^{A}P_{BORG}$$

Homogeneous Transform



$$^{A}P = {}^{A}_{B}R\,{}^{B}P + {}^{A}P_{BORG}$$

$$\begin{bmatrix} ^{A}P \\ \hline 1 \end{bmatrix} = \left[\begin{array}{c|c} ^{A}_{B}R & ^{A}P_{Borg} \\ \hline 0 \quad 0 \quad 0 & 1 \end{array}\right] \begin{bmatrix} ^{B}P \\ \hline 1 \end{bmatrix}$$

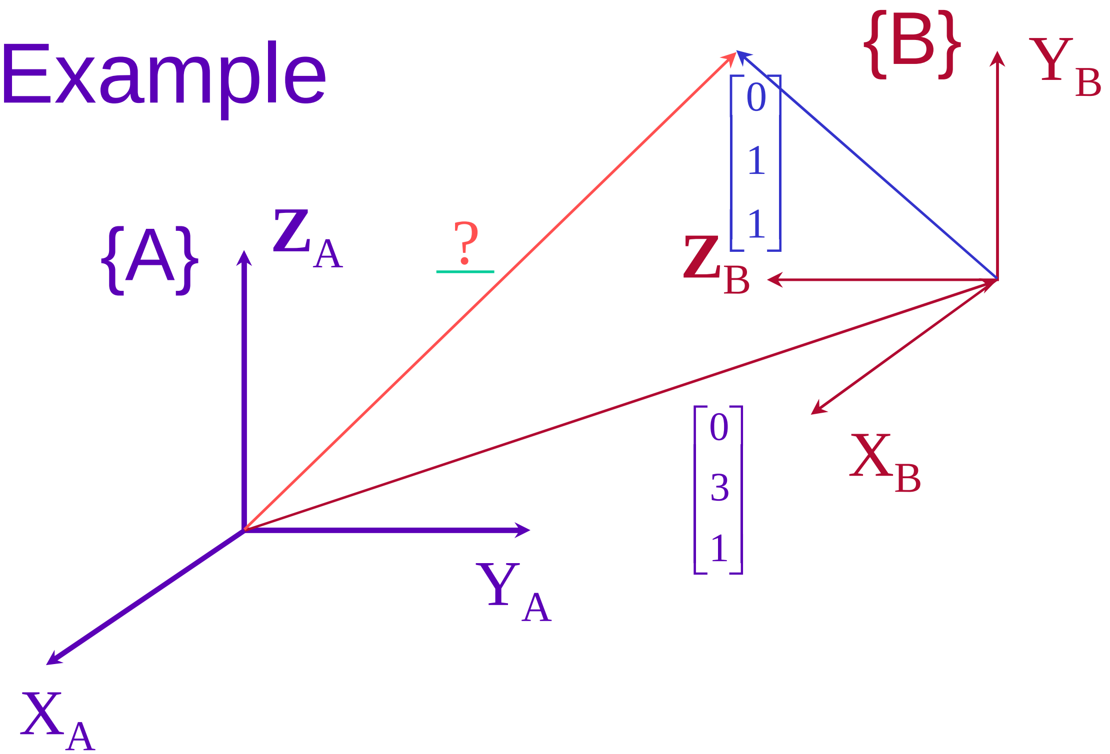
هم بخش ضرب
در هم بخش جمع

$$\underset{(4x1)}{^{A}P} = \underset{(4x4)}{^{A}_{B}T}\ \underset{(4x1)}{^{B}P}$$

همه تبدیل به ضرب

homogeneous

Example



Homogeneous Transform

$$ {}^{A}_{B}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 3 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{B}P = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} $$

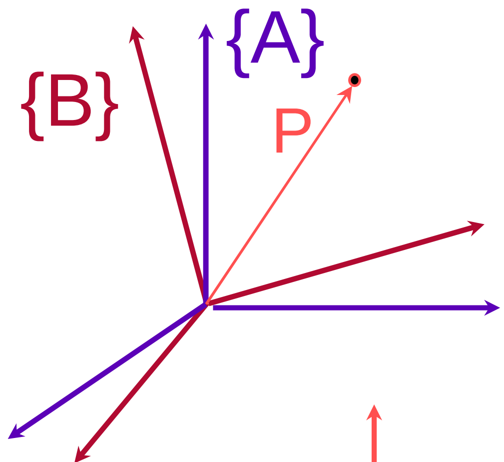
$$ {}^{A}P = {}^{A}_{B}T \cdot {}^{B}P \Rightarrow {}^{A}P = \begin{bmatrix} 0 \\ 2 \\ 2 \\ 1 \end{bmatrix} $$

Operators فرض میکنیم سیستم ثابت است ولی نقطه جابجا شده است.

Mapping: changing descriptions from frame to frame

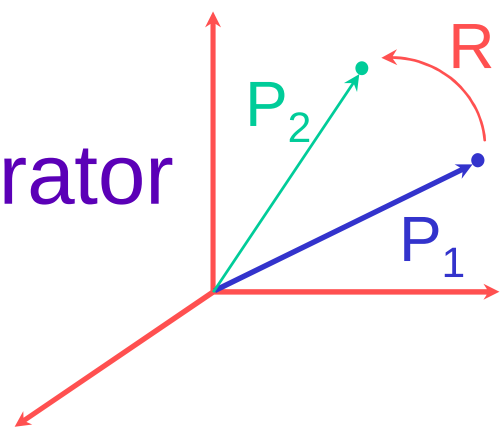
Operators: moving points (within the same frame)

Mapping



$$ {}^{A}P = {}^{A}_{B}R \; {}^{B}P $$

Rotational Operator



$$ R:\; P_1 \longrightarrow P_2 $$

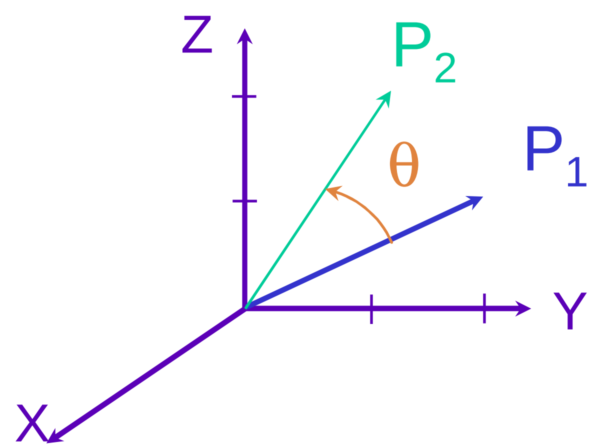
$$ \boxed{P_2 = R \; P_1} $$

Rotational Operators

$$R_K(\theta): \quad P_1 \longrightarrow P_2$$

$$P_2 = R_K(\theta)\, P_1$$

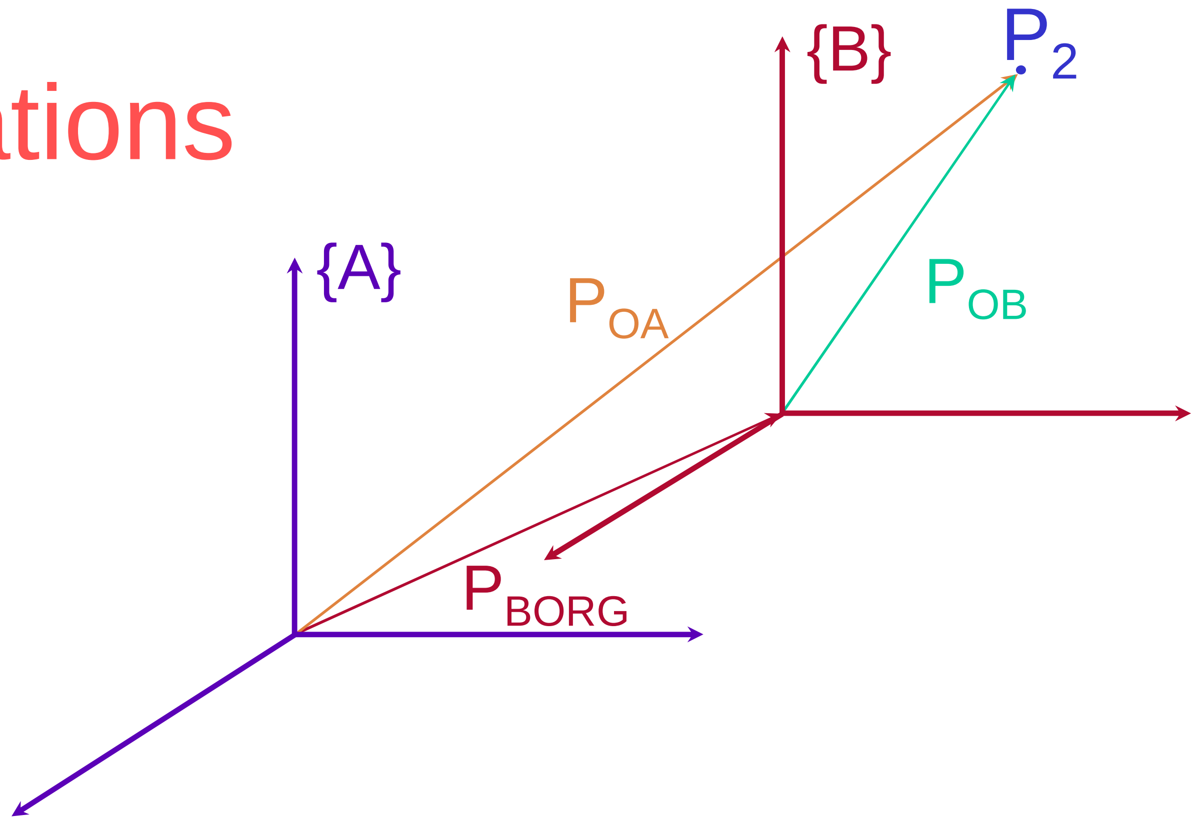
Example



$$R_X(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$P_2 = R_X(\theta) P_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.8 & -0.6 \\ 0 & 0.6 & 0.8 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

Translations

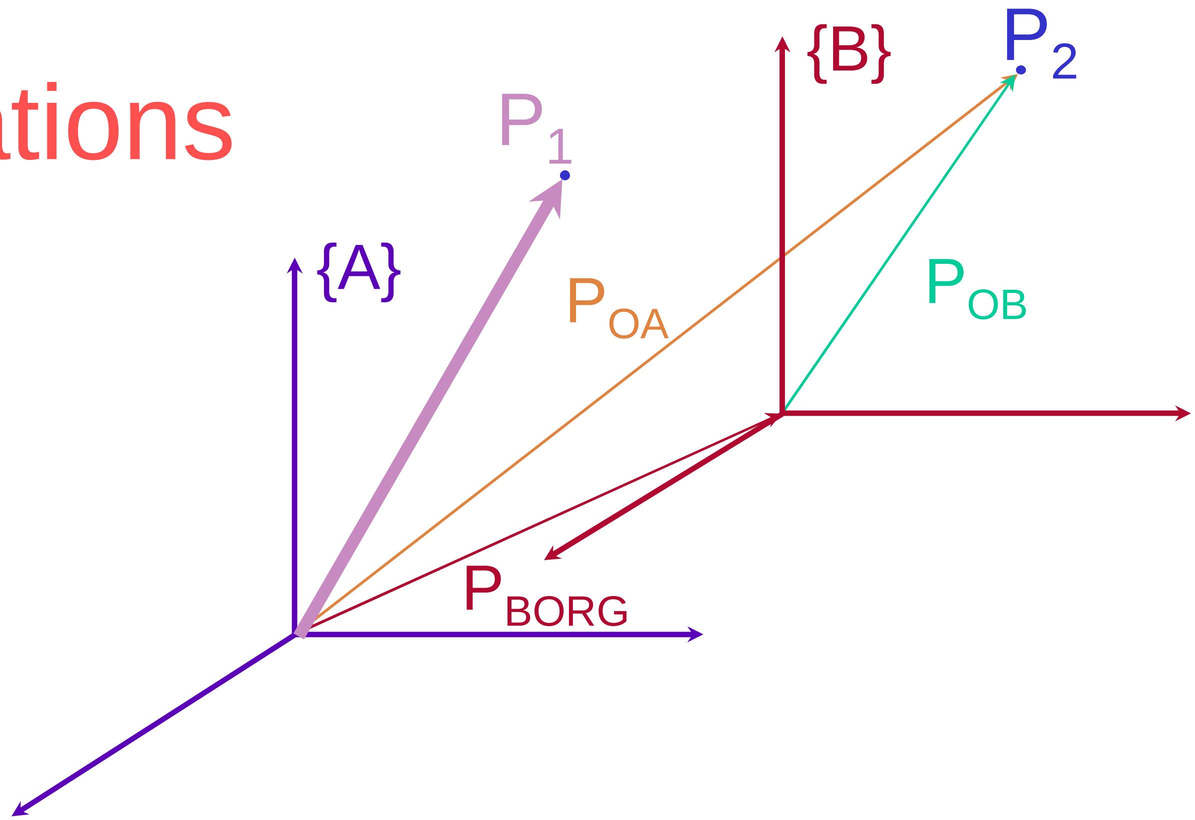


Mapping: $P_{BORG}: \quad P_{OB} \longrightarrow P_{OA}$ (same point)

2 diff. vectors

$$P_{OA} = P_{OB} + P_{BORG}$$

Translations



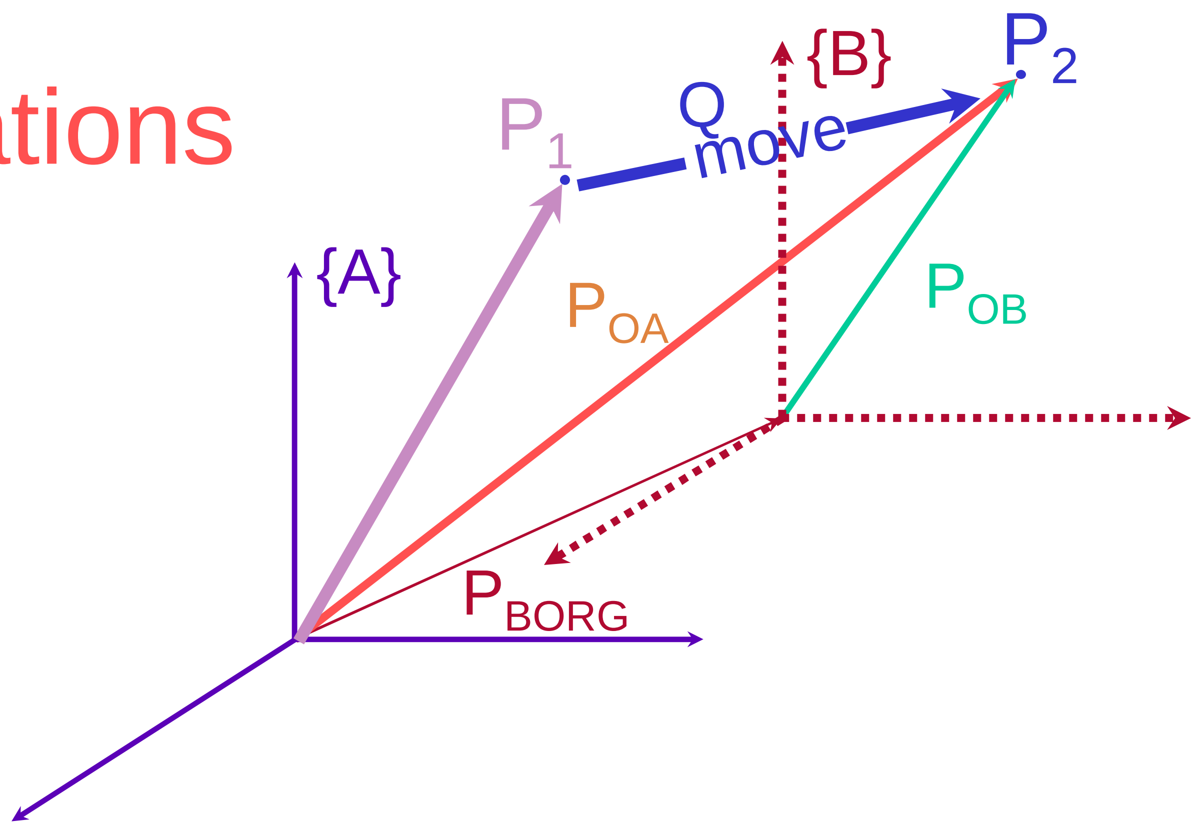
Mapping: P_{BORG} : $P_{OB} \longrightarrow P_{OA}$ (same point)

2 diff. vectors

$$P_{OA} = P_{OB} + P_{BORG}$$

Translational Operator:

Translations



Mapping: P_{BORG} : $P_{OB} \longrightarrow P_{OA}$ (same point)

2 diff. vectors

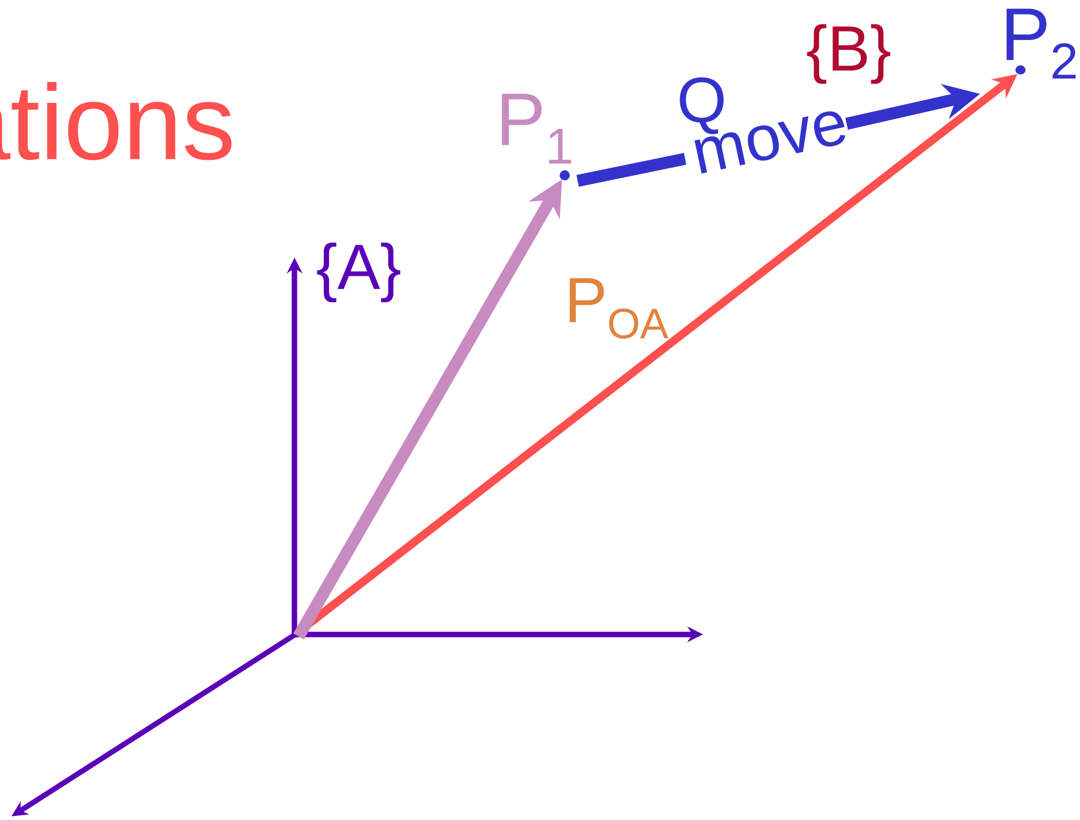
$$P_{OA} = P_{OB} + P_{BORG}$$

Translational Operator:

$$Q : \quad P_1 \longrightarrow P_2 \text{ (2 points, 2 diff vectors)}$$

$$P_2 = P_1 + Q$$

Translations

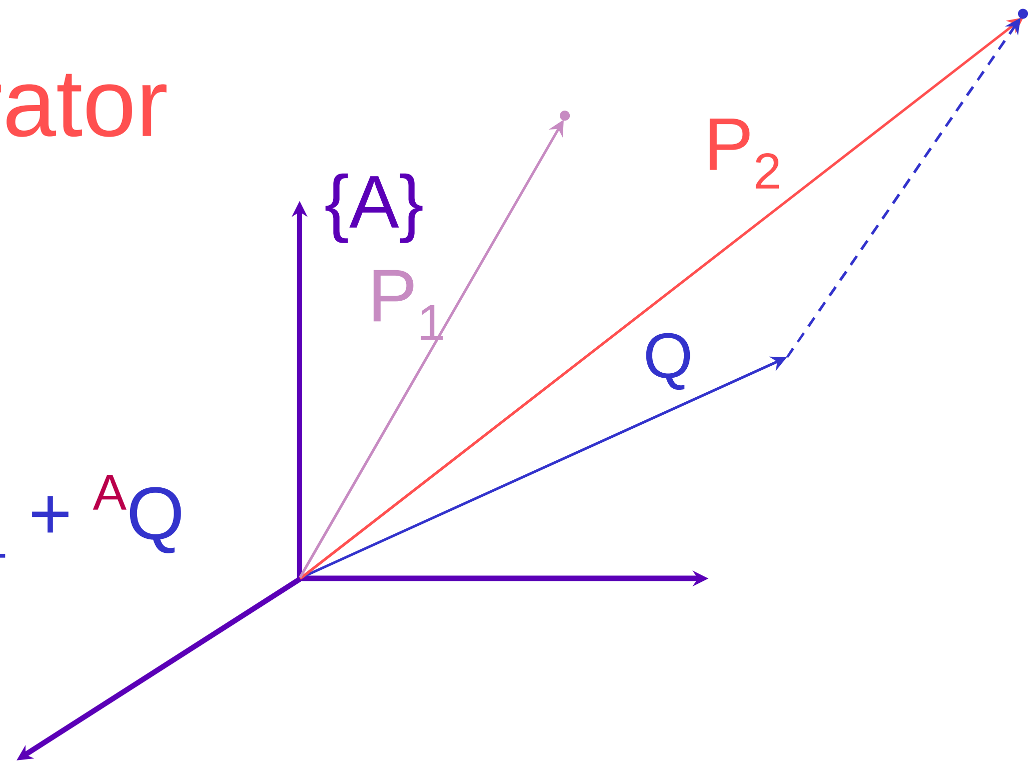


Translational Operator:

$Q: \quad P_1 \longrightarrow P_2$ (2 points, 2 diff vectors)

$P_2 = P_1 + Q$

Translation Operator



Operator: ${}^{A}P_2 = {}^{A}P_1 + {}^{A}Q$

Homogeneous Transform:

$$D_Q = \begin{bmatrix} 1 & 0 & 0 & q_x \\ 0 & 1 & 0 & q_y \\ 0 & 0 & 1 & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow {}^{A}P_2 = {}^{A}D_Q \, {}^{A}P_1$$

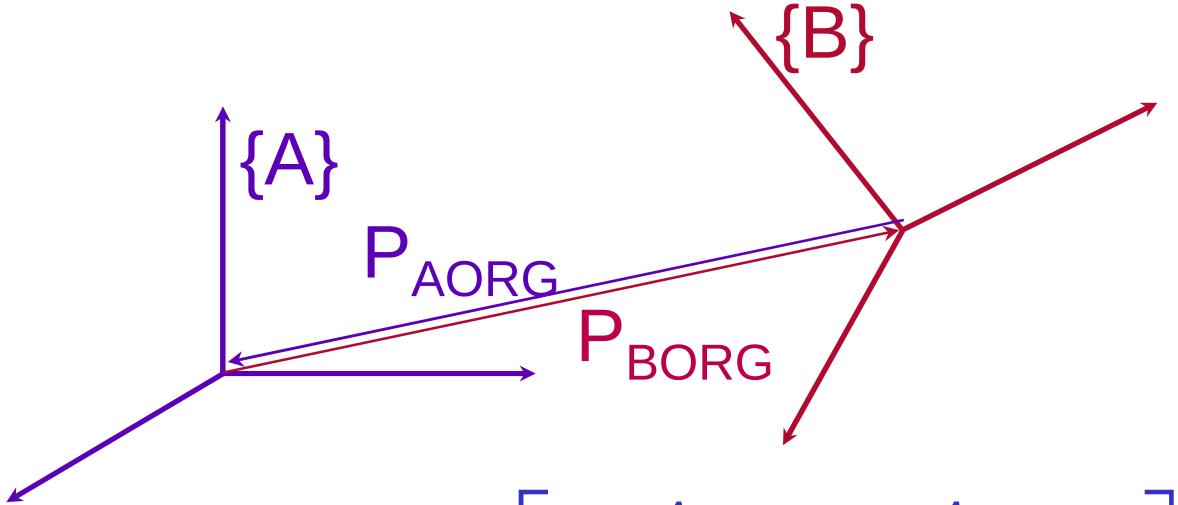
General Operators

Rotation

$$P_2 = \left(\begin{array}{ccc|c} & R_K(\theta) & & Q \\ \hline 0 & 0 & 0 & 1 \end{array}\right) P_1$$

$$\boxed{P_2 = T\,P_1}$$

Inverse Transform



$${}^A_B T = \begin{bmatrix} {}^A_B R & {}^A P_{Borg} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

$$R^{-1} = R^T \qquad (T^{-1} \neq T^T)$$

$${}^A_B T^{-1} = {}^B_A T = \begin{bmatrix} {}^A_B R^T & -{}^A_B R^T \cdot {}^A P_{Borg} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

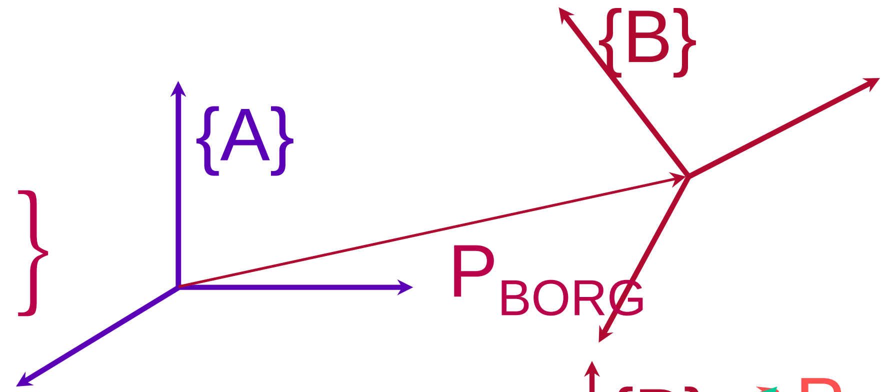
$${}^B P_{AORG}$$

Homogeneous Transform Interpretations

Description of a frame

توصیف دستگاه

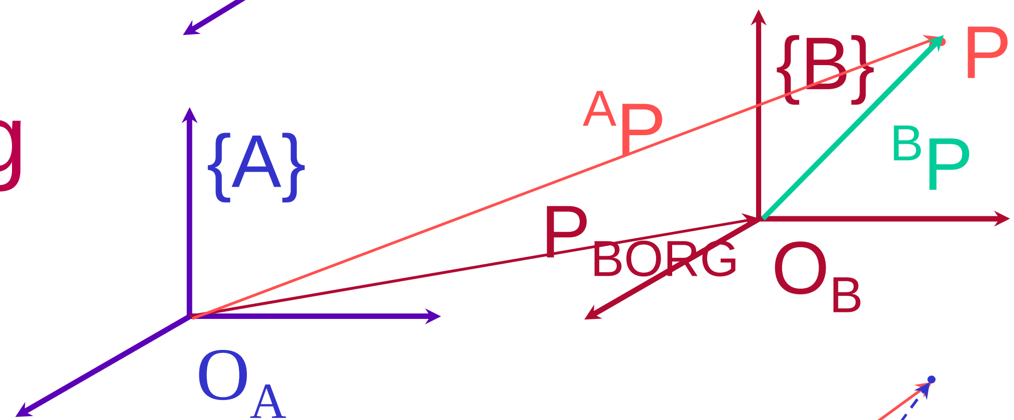
$${}^{A}_{B}T: \quad \{B\} = \left\{ {}^{A}_{B}R \quad {}^{A}P_{Borg} \right\}$$



Transform mapping

توصیف بردار در دو دستگاه

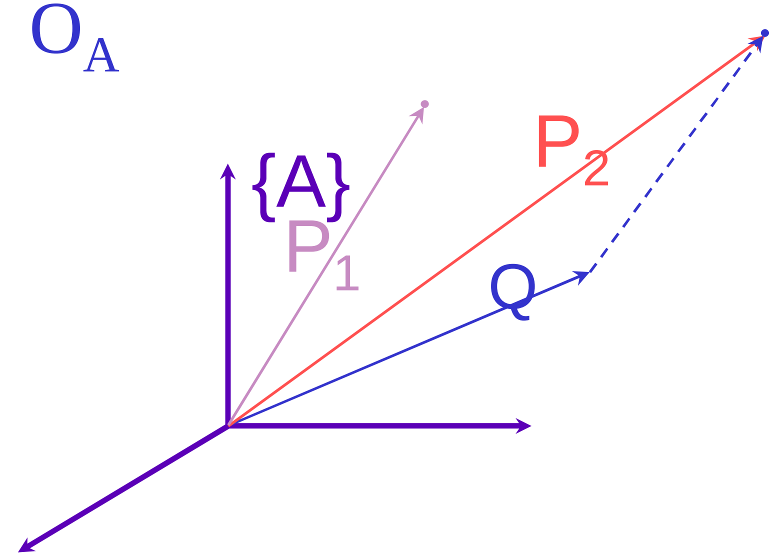
$${}^{A}_{B}T: \quad {}^{B}P \rightarrow {}^{A}P$$



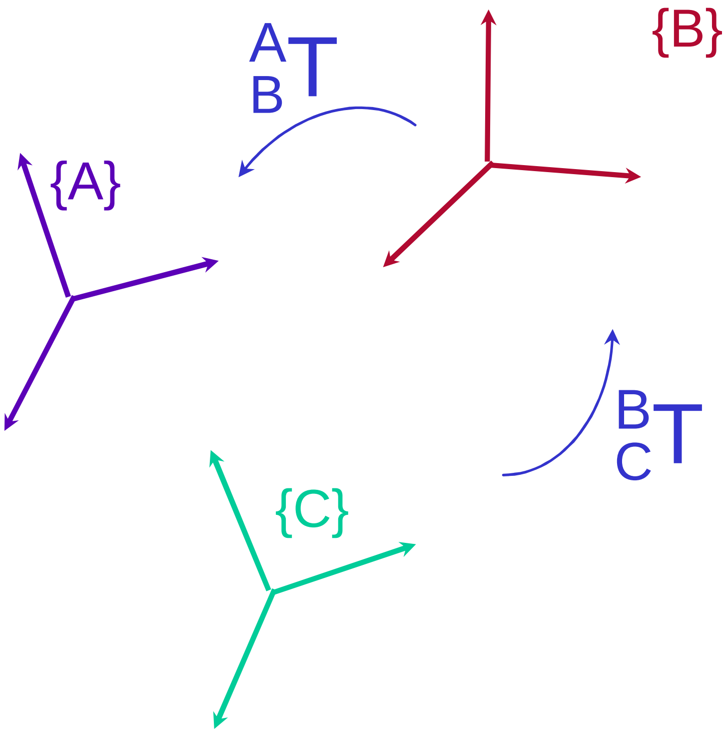
Transform operator

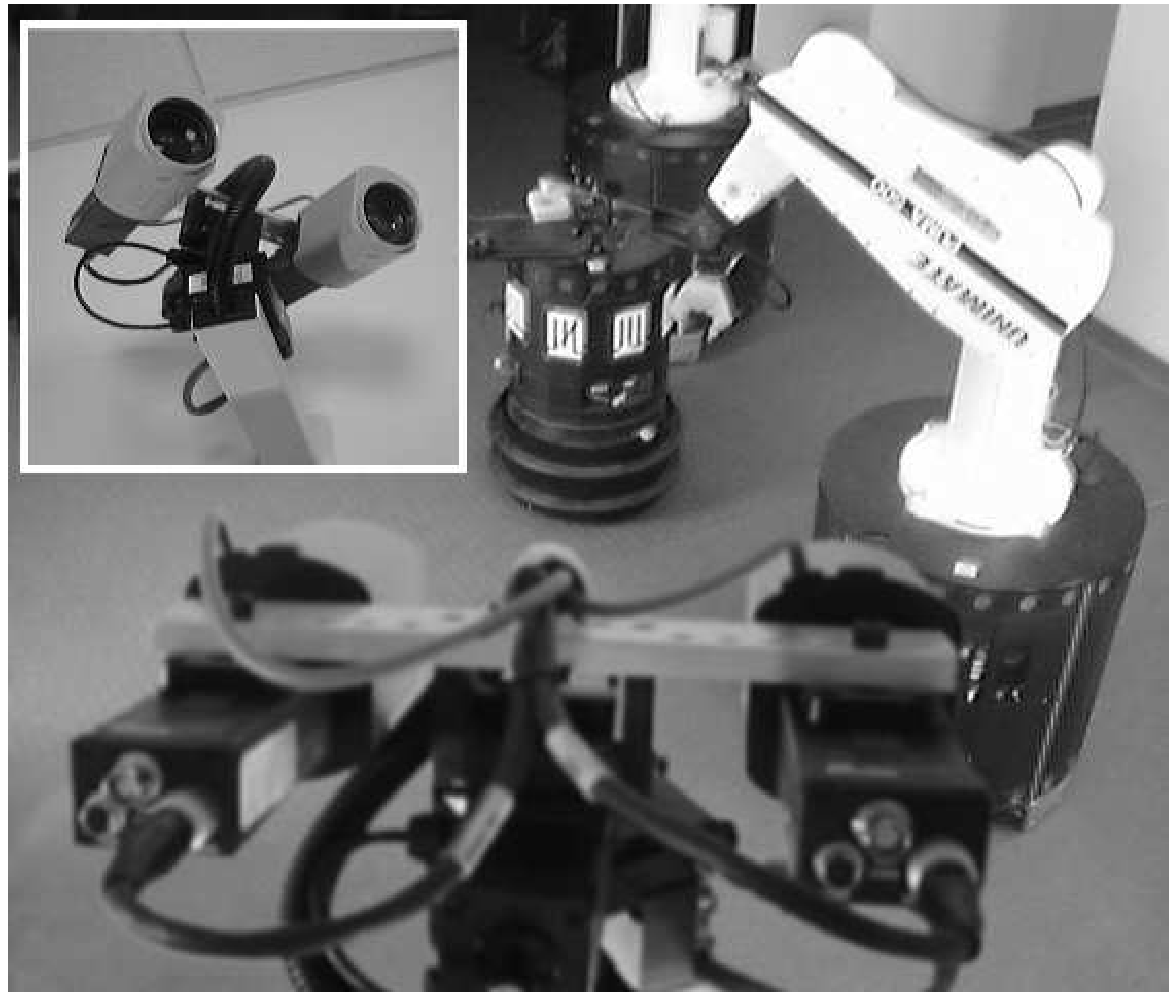
Operator

$$T: \quad P_1 \rightarrow P_2$$

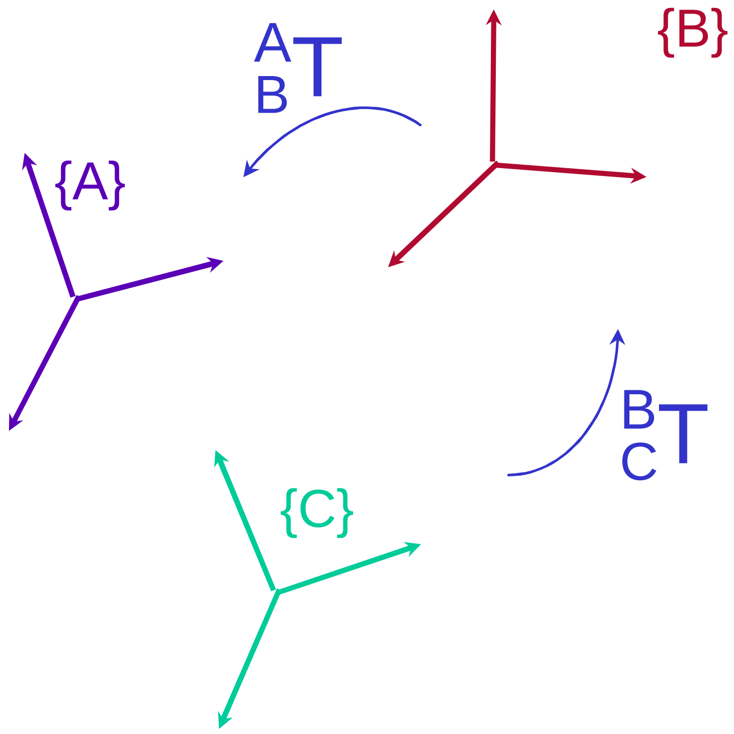


Transform Equation

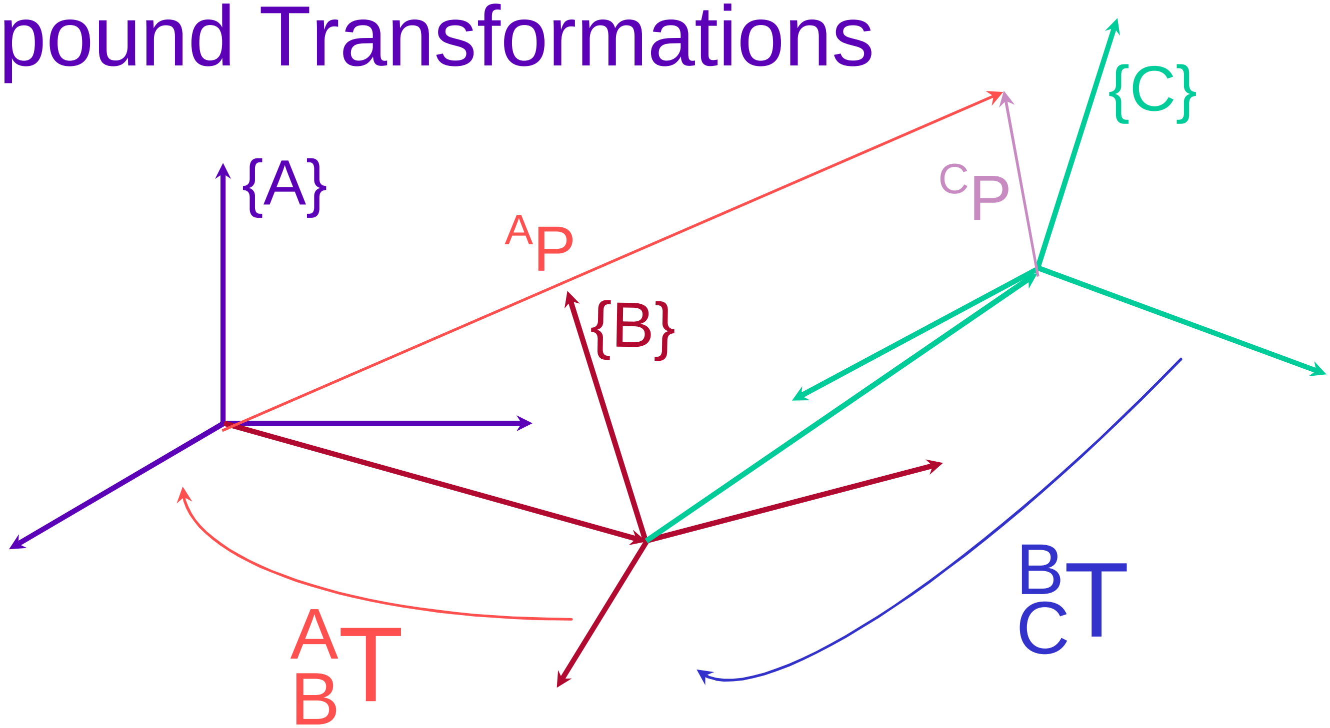




Transform Equation



Compound Transformations



$$^{B}P = {}^{B}_{C}T \, {}^{C}P$$

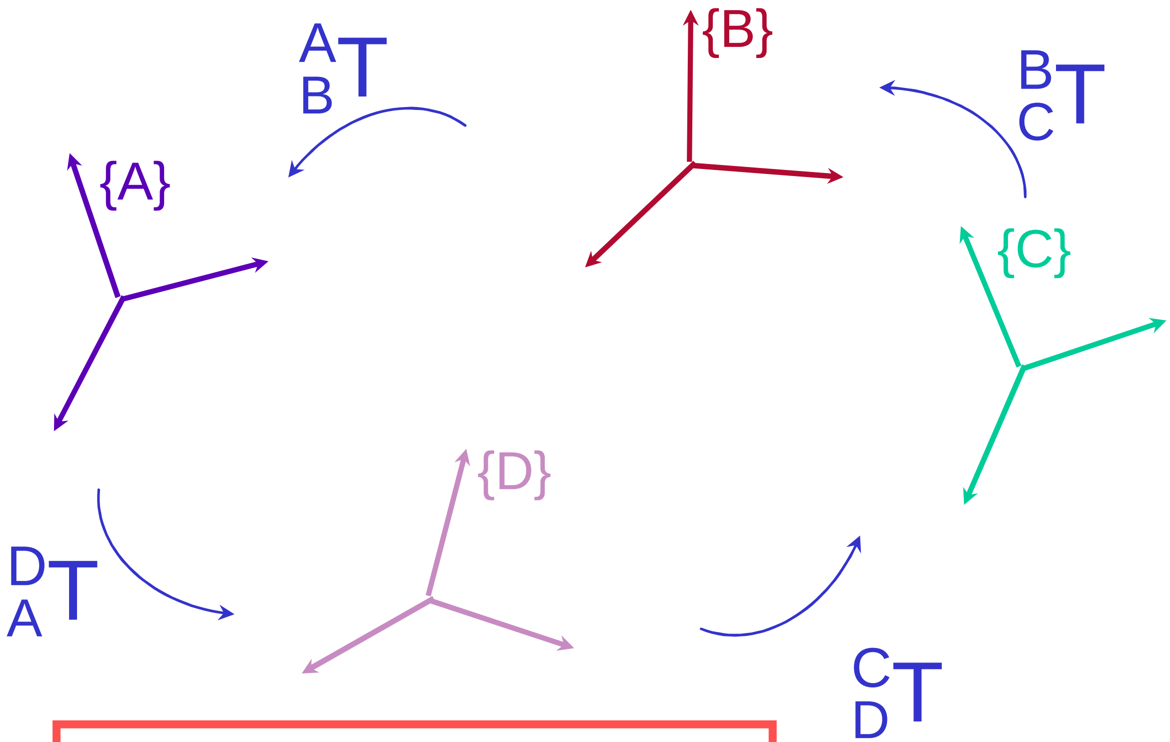
$$^{A}P = {}^{A}_{B}T \, {}^{B}P$$

$$^{A}P = {}^{A}_{B}T \, {}^{B}_{C}T \, {}^{C}P \Rightarrow {}^{A}_{C}T = {}^{A}_{B}T \, {}^{B}_{C}T$$

$$ {}^{A}_{C}T = {}^{A}_{B}T \, {}^{B}_{C}T $$

$$ {}^{A}_{C}T = \begin{bmatrix} {}^{A}_{B}R \, {}^{B}_{C}R & {}^{A}_{B}R \, {}^{B}P_{Corg} + {}^{A}P_{Borg} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} $$

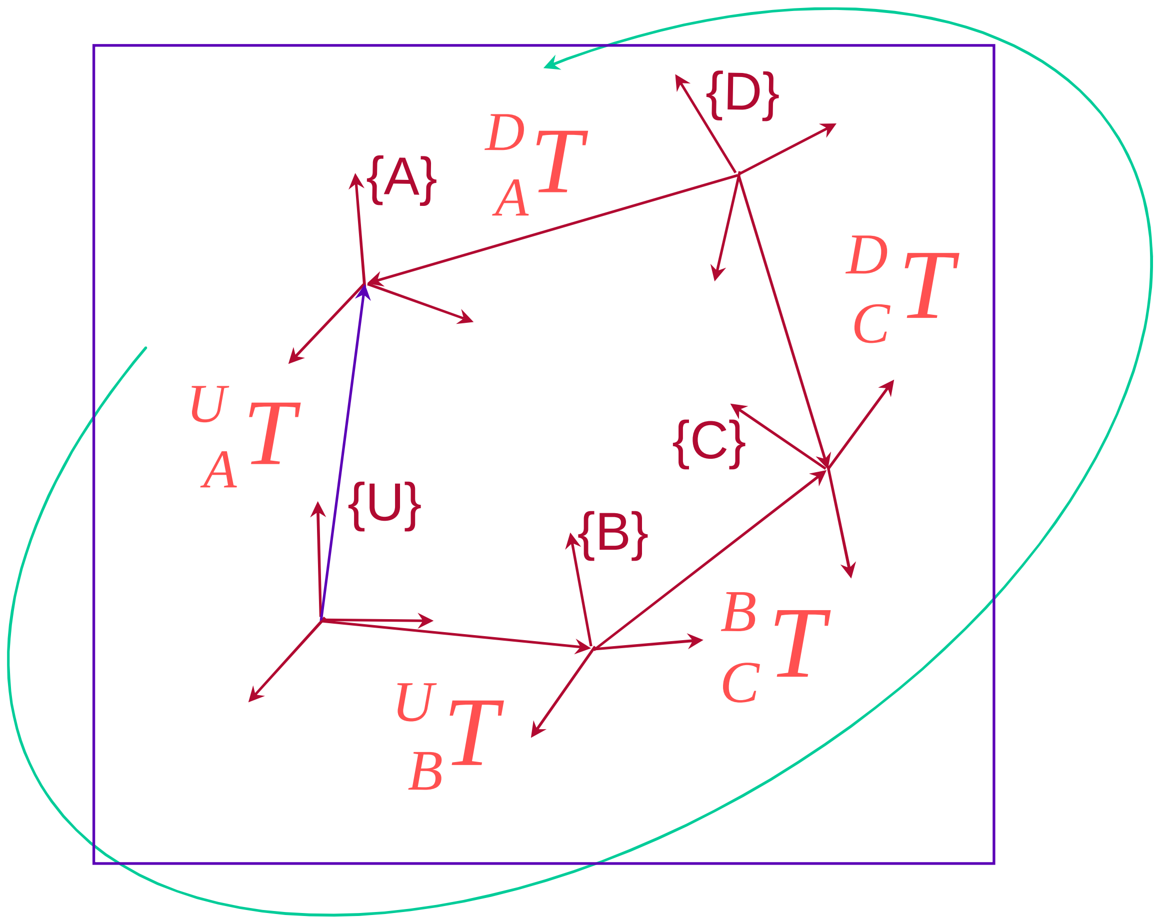
Transform Equation



به تعداد n برابر لازم

$$\boxed{{}^{A}_{B}T\;{}^{B}_{C}T\;{}^{C}_{D}T\;{}^{D}_{A}T \equiv I}$$

$$\Rightarrow \quad {}^{B}_{A}T = {}^{B}_{C}T\;{}^{C}_{D}T\;{}^{D}_{A}T$$



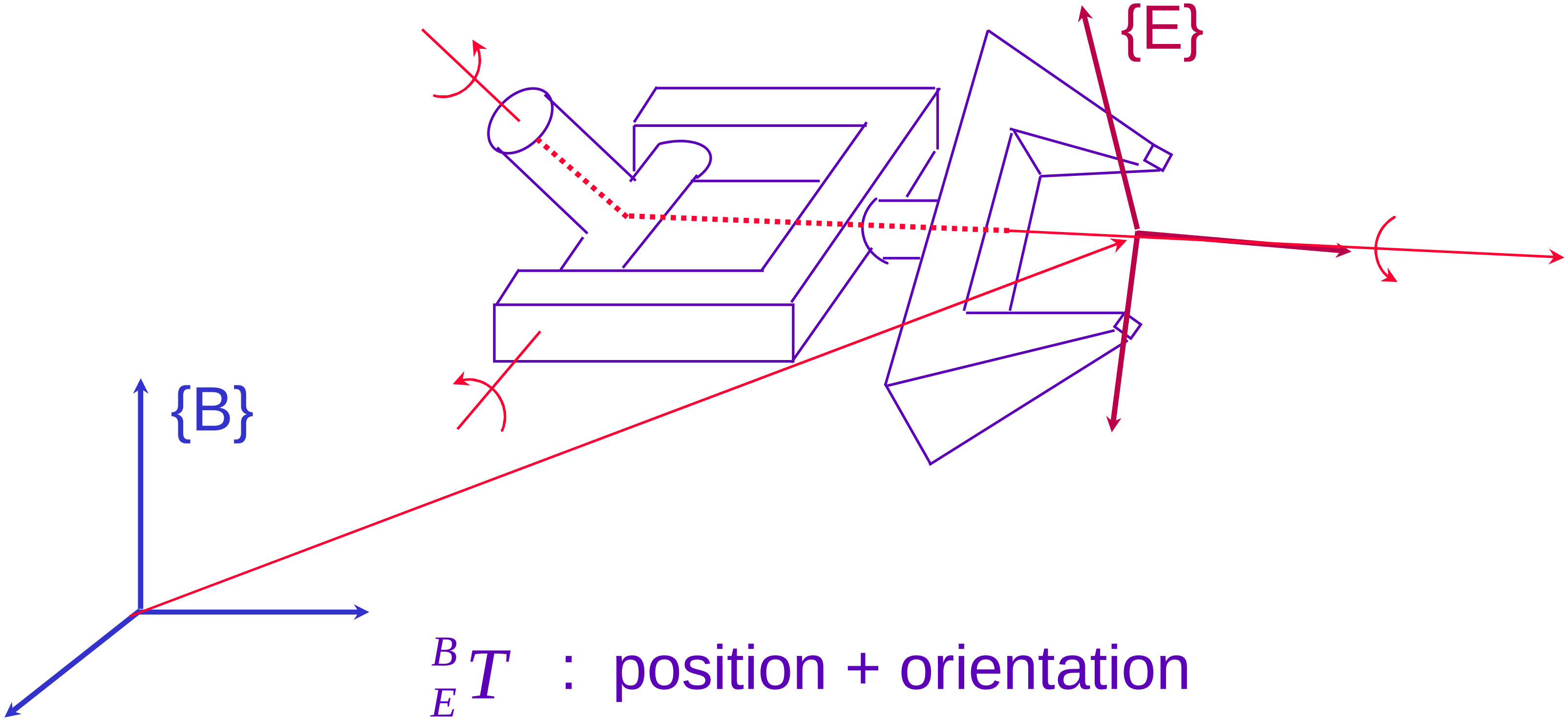
$$^{D}_{A}T^{-1}\cdot{}^{D}_{C}T\cdot{}^{B}_{C}T^{-1}\cdot{}^{U}_{B}T^{-1}\cdot{}^{U}_{A}T\equiv I$$

$$^{U}_{A}T={}^{U}_{B}T\cdot{}^{B}_{C}T\cdot{}^{D}_{C}T^{-1}\cdot{}^{D}_{A}T$$

Spatial Descriptions

- Task Description
- Transformations
- Representations

End-Effector Configuration

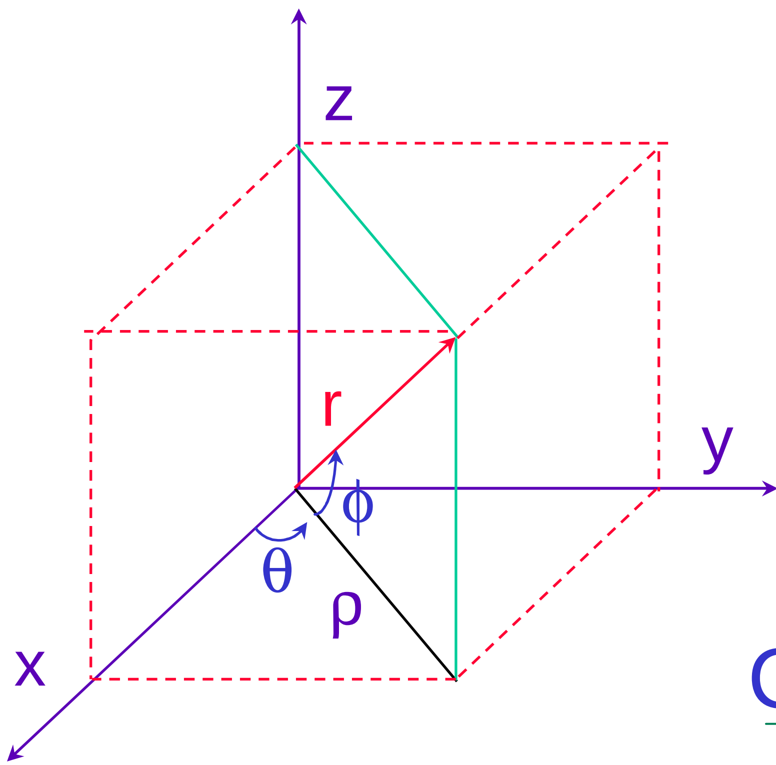


${}^{B}_{E}T$: position + orientation

End-Effector Configuration Parameters

$$X = \begin{bmatrix} X_P \\ X_R \end{bmatrix} \begin{matrix} \leftarrow \text{position} \\ \leftarrow \text{orientation} \end{matrix}$$

Position Representations



Cartesian: (x, y, z)

Cylindrical: (ρ, θ, z)

Spherical: (r, θ, ϕ)

Rotation Representations

Rotation Matrix

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix}$$

۹ عدد داریم ولی فقط ۳ درجه
آزادی را داریم.

Direction Cosines

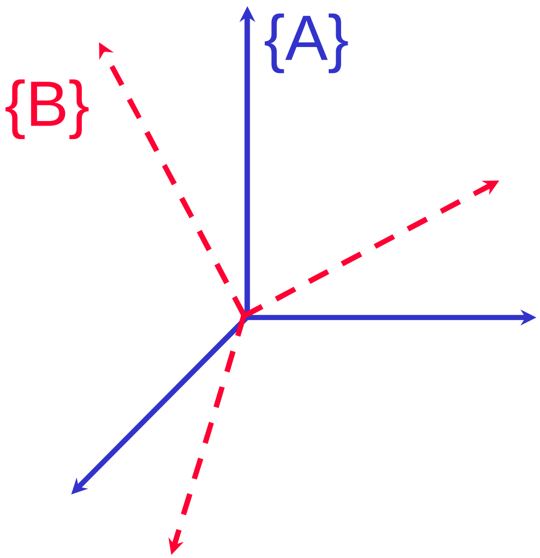
$$x_r = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \end{bmatrix}_{(9x1)}$$

Constraints

$$|\mathbf{r}_1| = |\mathbf{r}_2| = |\mathbf{r}_3| = 1$$ orthonormal

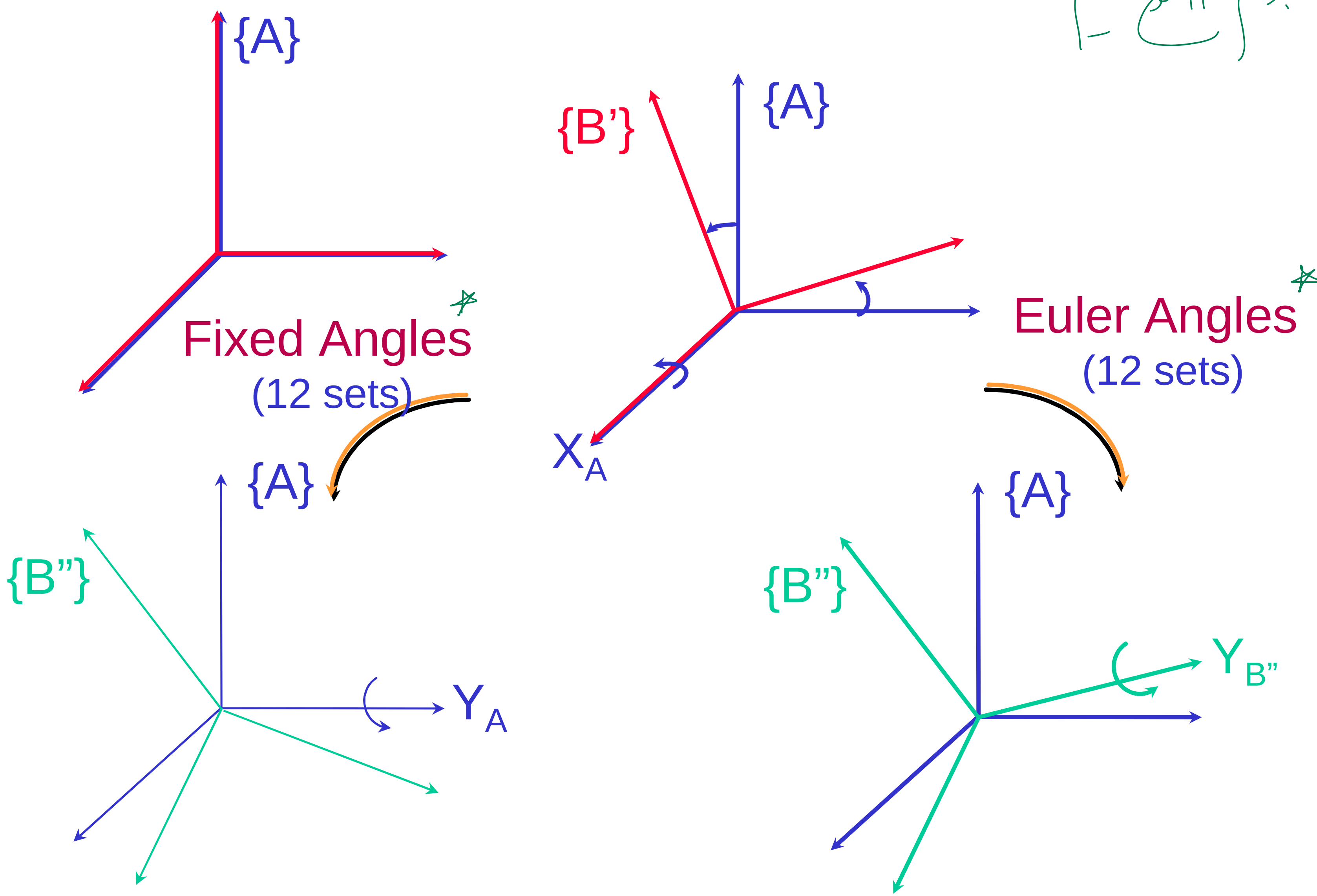
$$\mathbf{r}_1 . \mathbf{r}_2 = \mathbf{r}_1 . \mathbf{r}_3 = \mathbf{r}_2 . \mathbf{r}_3 = 0$$

Three Angle Representations



Three Angle Representations

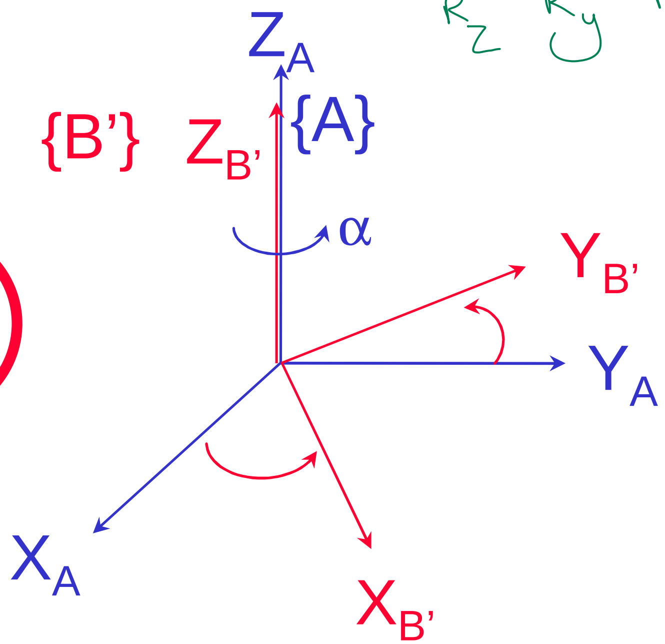
مجدداً باز هم ۱۲ نوع داریم

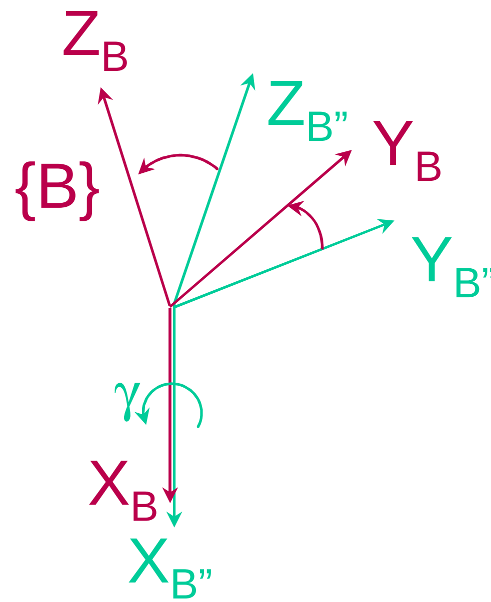


Euler Angles (Z-Y-X)

R_z R_y R_x

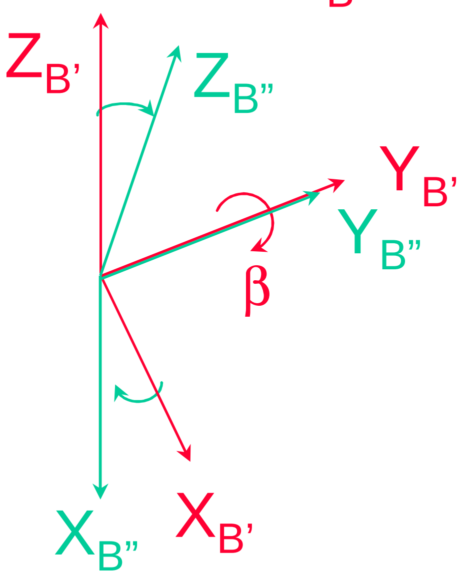
${}^{A}_{B'}R$





${}^{B''}_{B}R$

${}^{B'}_{B''}R$

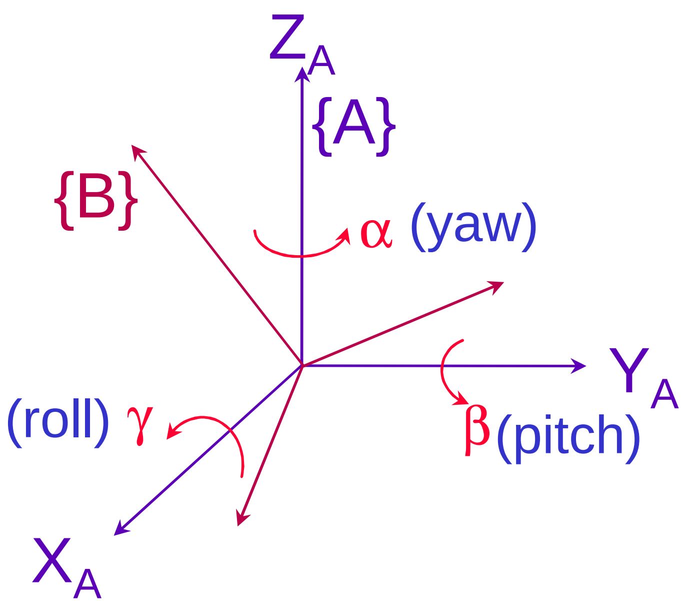


$${}^{A}_{B}R = {}^{A}_{B'}R \cdot {}^{B'}_{B''}R \cdot {}^{B''}_{B}R$$

به ترتیب اعمال

$$\boxed{{}^{A}_{B}R = R_Z(\alpha) \cdot R_Y(\beta) \cdot R_X(\gamma)}$$

X-Y-Z Fixed Angles



$$R_X(\gamma):\ v \rightarrow R_X(\gamma).v$$

$$R_Y(\beta):\ (R_X(\gamma).v) \rightarrow R_Y(\beta).(R_X(\gamma).v)$$

$$R_Z(\alpha):\ (R_Y(\beta).R_X(\gamma).v) \rightarrow R_Z(\alpha).(R_Y(\beta).R_X(\gamma).v)$$

$$\boxed{{}^A_B R = {}^A_B R_{XYZ}(\gamma,\beta,\alpha) = R_Z(\alpha).R_Y(\beta).R_X(\gamma)}$$

$\leftrightarrow$ برعکس ترتیب اجرا

Z-Y-X Euler Angles

$$ {}^{A}_{B}R = R_{Z'}(\alpha).R_{Y'}(\beta).R_{X'}(\gamma) $$

$$ \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix} $$

$$ {}^{A}_{B}R = {}^{A}_{B}R_{Z'Y'X'}(\alpha,\beta,\gamma) = \begin{bmatrix} c\alpha.c\beta & X & X \\ s\alpha.c\beta & X & X \\ -s\beta & c\beta.s\gamma & c\beta.c\gamma \end{bmatrix} $$

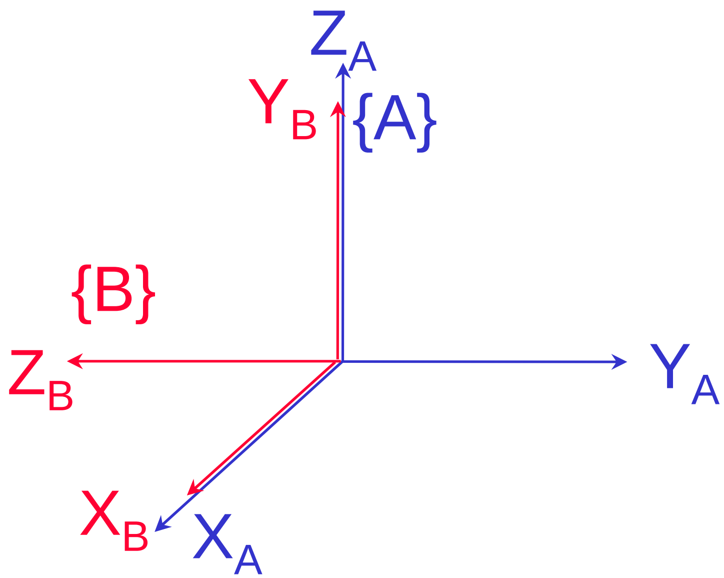
بسته به ترتیب اعمال ، از این ماتریس می‌توان زوایا را بدست آورد.

Z-Y-Z Euler Angles

$$ {}^{A}_{B}R = R_{Z'}(\alpha).R_{Y'}(\beta).R_{Z'}(\gamma) $$

$$ {}^{A}_{B}R = {}^{A}_{B}R_{Z'Y'Z'}(\alpha,\beta,\gamma) = \begin{bmatrix} X & X & c\alpha.s\beta \\ X & X & s\alpha.s\beta \\ -s\beta.c\gamma & s\beta.s\gamma & c\beta \end{bmatrix} $$

Example



$$R_{Z'Y'X'}(\alpha, \beta, \gamma):$$

$$\alpha = 0$$

$$\beta = 0$$

$$\gamma = 90^o$$

Fixed & Euler Angles

X-Y-Z Fixed Angles

$$R_{XYZ}(\gamma,\beta,\alpha) = R_Z(\alpha).R_Y(\beta).R_X(\gamma)$$

Z-Y-X Euler Angles

$$R_{Z'Y'X'}(\alpha,\beta,\gamma) = R_Z(\alpha).R_Y(\beta).R_X(\gamma)$$

$$\boxed{R_{Z'Y'X'}(\alpha,\beta,\gamma) = R_{XYZ}(\gamma,\beta,\alpha)}$$

معادل هستند

Inverse Problem

Given ${}^{A}_{B}R$ find (α, β, γ)

$R_{Z'Y'X'}$

$$
{}^{A}_{B}R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} c\alpha.c\beta & c\alpha.s\beta.s\gamma - s\alpha.c\gamma & c\alpha.s\beta.c\gamma + s\alpha.s\gamma \\ s\alpha.c\beta & s\alpha.s\beta.s\gamma + c\alpha.c\gamma & s\alpha.s\beta.c\gamma - c\alpha.s\gamma \\ -s\beta & c\beta.s\gamma & c\beta.c\gamma \end{bmatrix}
$$

$$
\left.\begin{aligned} \cos\beta = c\beta = \sqrt{r_{11}^2 + r_{21}^2} \\ \sin\beta = s\beta = -r_{31} \end{aligned}\right\} \rightarrow \beta = A\tan 2(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2})
$$

به محدودیت سن را باید توجه کرد.

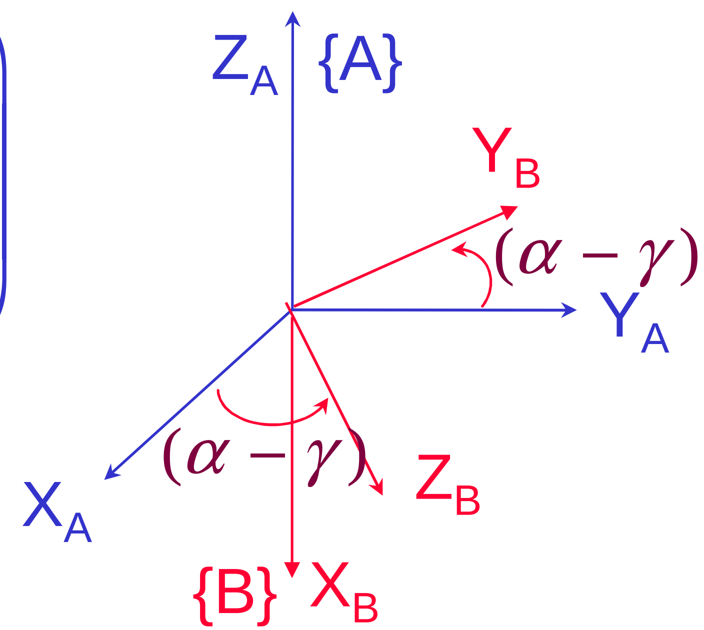
if $c\beta = 0$ $(\beta = \pm 90^{\circ})$ ⇨ Singularity of the representation

⇨ Only $(\alpha + \gamma)$ or $(\alpha - \gamma)$ is defined

Singularities - Example $\left(R_{Z'Y'X'} \right)$

$$\underline{c\beta = 0,\ s\beta = +1}$$

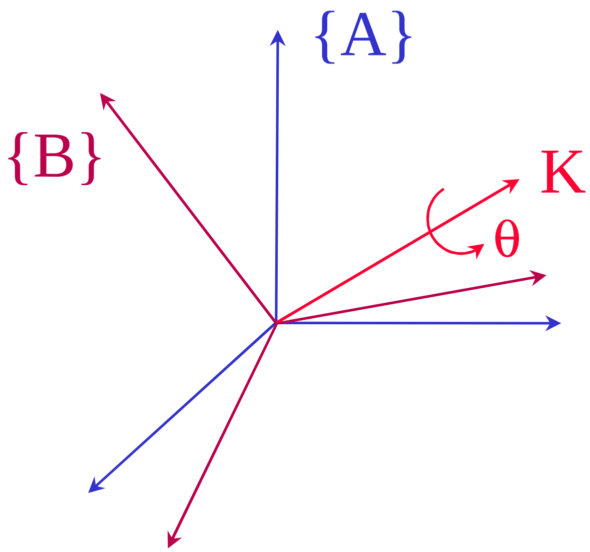
$${}^{A}_{B}R = \begin{pmatrix} 0 & -s(\alpha-\gamma) & c(\alpha-\gamma) \\ 0 & c(\alpha-\gamma) & s(\alpha-\gamma) \\ -1 & 0 & 0 \end{pmatrix}$$



$$\underline{c\beta = 0,\ s\beta = -1}$$

$${}^{A}_{B}R = \begin{pmatrix} 0 & -s(\alpha+\gamma) & -c(\alpha+\gamma) \\ 0 & c(\alpha+\gamma) & -s(\alpha+\gamma) \\ 1 & 0 & 0 \end{pmatrix}$$

Equivalent angle-axis representation, $R_K(\theta)$



هم‌جهت با بردار، اندازه زاویه چرخش

$$X_r = \theta.K = \begin{bmatrix} \theta.k_X \\ \theta.k_Y \\ \theta.k_Z \end{bmatrix}$$

$$R_K(\theta) = \begin{bmatrix} k_x.k_x.v\theta + c\theta & k_x.k_y.v\theta - k_z.s\theta & k_x.k_z.v\theta + k_y.s\theta \\ k_x.k_y.v\theta + k_z.s\theta & k_y.k_y.v\theta + c\theta & k_y.k_z.v\theta - k_x.s\theta \\ k_x.k_z.v\theta - k_y.s\theta & k_y.k_z.v\theta + k_x.s\theta & k_z.k_z.v\theta + c\theta \end{bmatrix}$$

with $v\theta = 1 - c\theta$

$$R_K(\theta) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$\theta = Ar\cos\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right)$$

$${}^{A}K = \frac{1}{2.\sin\theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix},$$

singularity for $\sin\theta = 0$

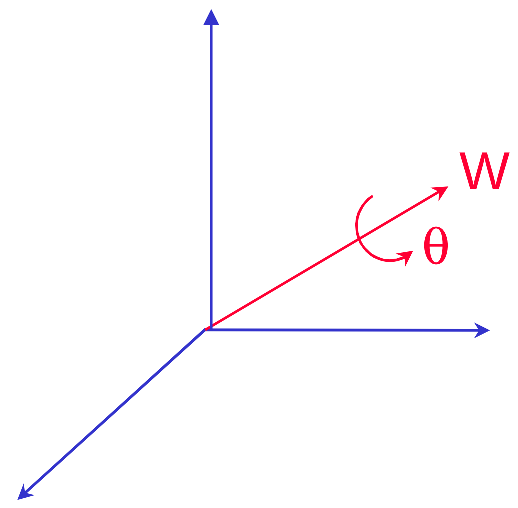
Euler Parameters

$$\varepsilon_1 = W_x . \sin\frac{\theta}{2}$$

$$\varepsilon_2 = W_y . \sin\frac{\theta}{2}$$

$$\varepsilon_3 = W_z . \sin\frac{\theta}{2}$$

$$\varepsilon_4 = \cos\frac{\theta}{2}$$



Normality Condition

$$|W| = 1, \quad \varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 + \varepsilon_4^2 = 1$$

ε : point on a unit hypershpere
in four-dimensional space

Inverse Problem

Given ${}^A_B R$ find ε

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \equiv \begin{bmatrix} 1-2\varepsilon_2^2-2\varepsilon_3^2 & 2(\varepsilon_1\varepsilon_2-\varepsilon_3\varepsilon_4) & 2(\varepsilon_1\varepsilon_3+\varepsilon_2\varepsilon_4) \\ 2(\varepsilon_1\varepsilon_2+\varepsilon_3\varepsilon_4) & 1-2\varepsilon_1^2-2\varepsilon_3^2 & 2(\varepsilon_2\varepsilon_3-\varepsilon_1\varepsilon_4) \\ 2(\varepsilon_1\varepsilon_3-\varepsilon_2\varepsilon_4) & 2(\varepsilon_2\varepsilon_3+\varepsilon_1\varepsilon_4) & 1-2\varepsilon_1^2-2\varepsilon_2^2 \end{bmatrix}$$

$$r_{11}+r_{22}+r_{33} = 3-4\underbrace{(\varepsilon_1^2+\varepsilon_2^2+\varepsilon_3^2)}_{(1-\varepsilon_4^2)}$$

$$\varepsilon_4 = \frac{1}{2}\sqrt{1+r_{11}+r_{22}+r_{33}}$$

$$\varepsilon_1 = \frac{r_{32}-r_{23}}{4\varepsilon_4}, \quad \varepsilon_2 = \frac{r_{13}-r_{31}}{4\varepsilon_4}, \quad \varepsilon_3 = \frac{r_{21}-r_{12}}{4\varepsilon_4}$$

$$\varepsilon_4 = 0?$$

Lemma

For all rotations one of the Euler Parameters is greater than or equal to 1/2

$$\left(\sum_{1}^{4} \varepsilon_i^2 = 1\right)$$

چون جمعشان یک است همه نمی‌توانند صفر باشند.
از آن که صفر نیست در مخرج استفاده می‌کنیم.

توجه شود اگر بهترین → کار شود با بزرگترین مقدار است.

Algorithm

Solve with respect to $\max_i \{\varepsilon_i\}$

- $\varepsilon_1 = \max_i \{\varepsilon_i\}$

$$\varepsilon_1 = \frac{1}{2}\sqrt{r_{11} - r_{22} - r_{33} + 1}$$

$$\varepsilon_2 = \frac{(r_{21} + r_{12})}{4\varepsilon_1}, \quad \varepsilon_3 = \frac{(r_{31} + r_{13})}{4\varepsilon_1}, \quad \varepsilon_4 = \frac{(r_{32} - r_{23})}{4\varepsilon_1}$$

- $\varepsilon_1 = \max_i \{\varepsilon_i\}$

$$\varepsilon_1 = \frac{1}{2}\sqrt{r_{11} - r_{22} - r_{33} + 1}$$

- $\varepsilon_2 = \max_i \{\varepsilon_i\}$

$$\varepsilon_2 = \frac{1}{2}\sqrt{-r_{11} + r_{22} - r_{33} + 1}$$

- $\varepsilon_3 = \max_i \{\varepsilon_i\}$

$$\varepsilon_3 = \frac{1}{2}\sqrt{-r_{11} - r_{22} + r_{33} + 1}$$

- $\varepsilon_4 = \max_i \{\varepsilon_i\}$

$$\varepsilon_4 = \frac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}}$$

Euler Parameters / Euler Angles

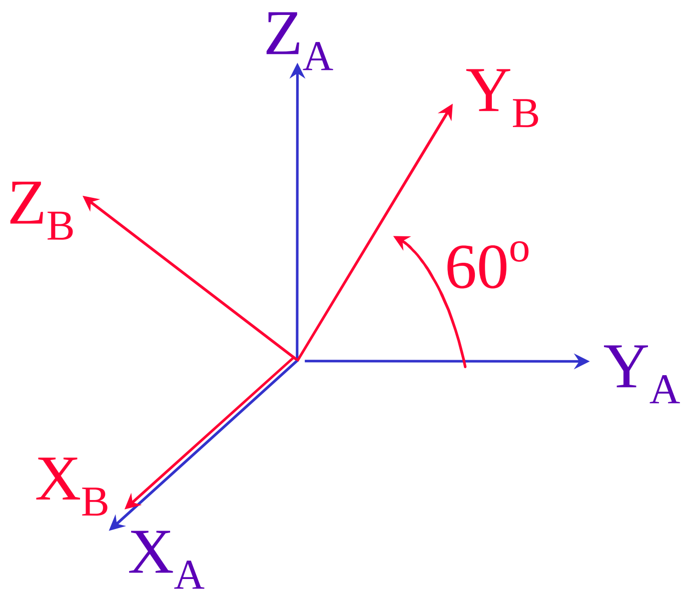
$$\varepsilon_1 = \sin\frac{\beta}{2}\cos\frac{\alpha-\gamma}{2}$$

$$\varepsilon_2 = \sin\frac{\beta}{2}\sin\frac{\alpha-\gamma}{2}$$

$$\varepsilon_3 = \cos\frac{\beta}{2}\sin\frac{\alpha+\gamma}{2}$$

$$\varepsilon_4 = \cos\frac{\beta}{2}\cos\frac{\alpha+\gamma}{2}$$

Quiz



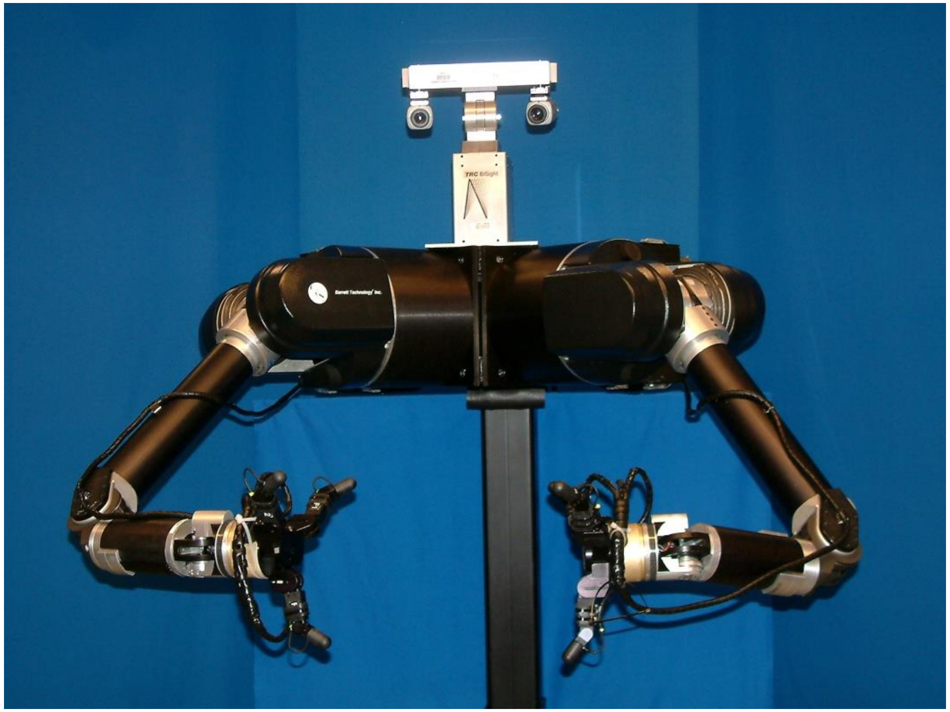
Direction Cosines

$$x_r = \begin{array}{c} \left[\begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 1/2 \\ \sqrt{3}/2 \\ 0 \\ -\sqrt{3}/2 \\ 1/2 \end{array}\right] \end{array} \begin{array}{c} \\ r_1 \\ \\ \\ r_2 \\ \\ \\ r_3 \\ \\ \end{array}$$

Euler Parameters

$$x_r = \left[\begin{array}{c} 1/2 \\ 0 \\ 0 \\ \sqrt{3}/2 \end{array}\right]$$

Forward Kinematics



- Where is the end effector w.r.t. the “base” frame?

Composition of homogeneous transforms

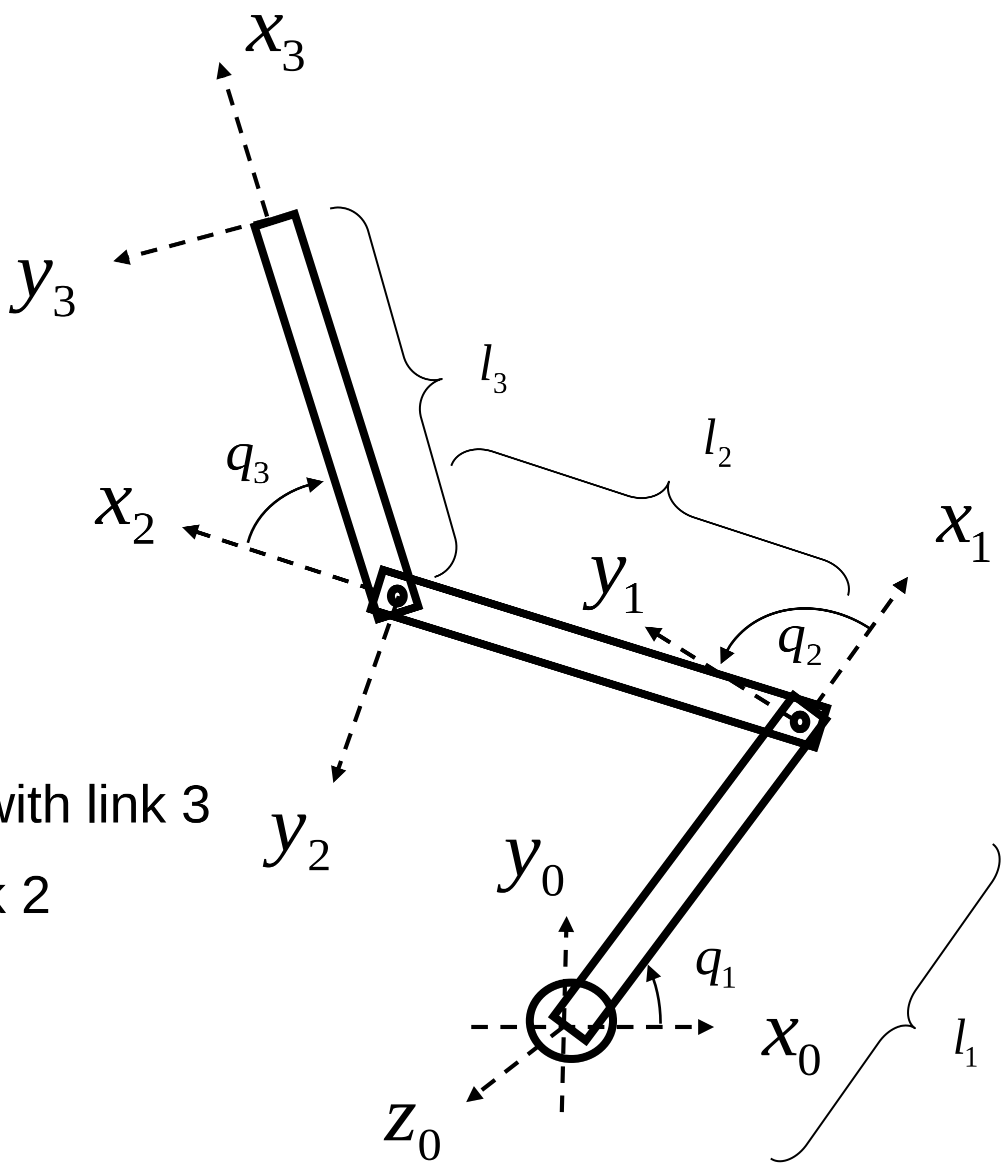
Base to *eff* transform

$$^{0}T_{3} = {}^{0}T_{1}\,{}^{1}T_{2}\,{}^{2}T_{3}$$

Transform associated with link 3

Transform associated with link 2

Transform associated with link 1

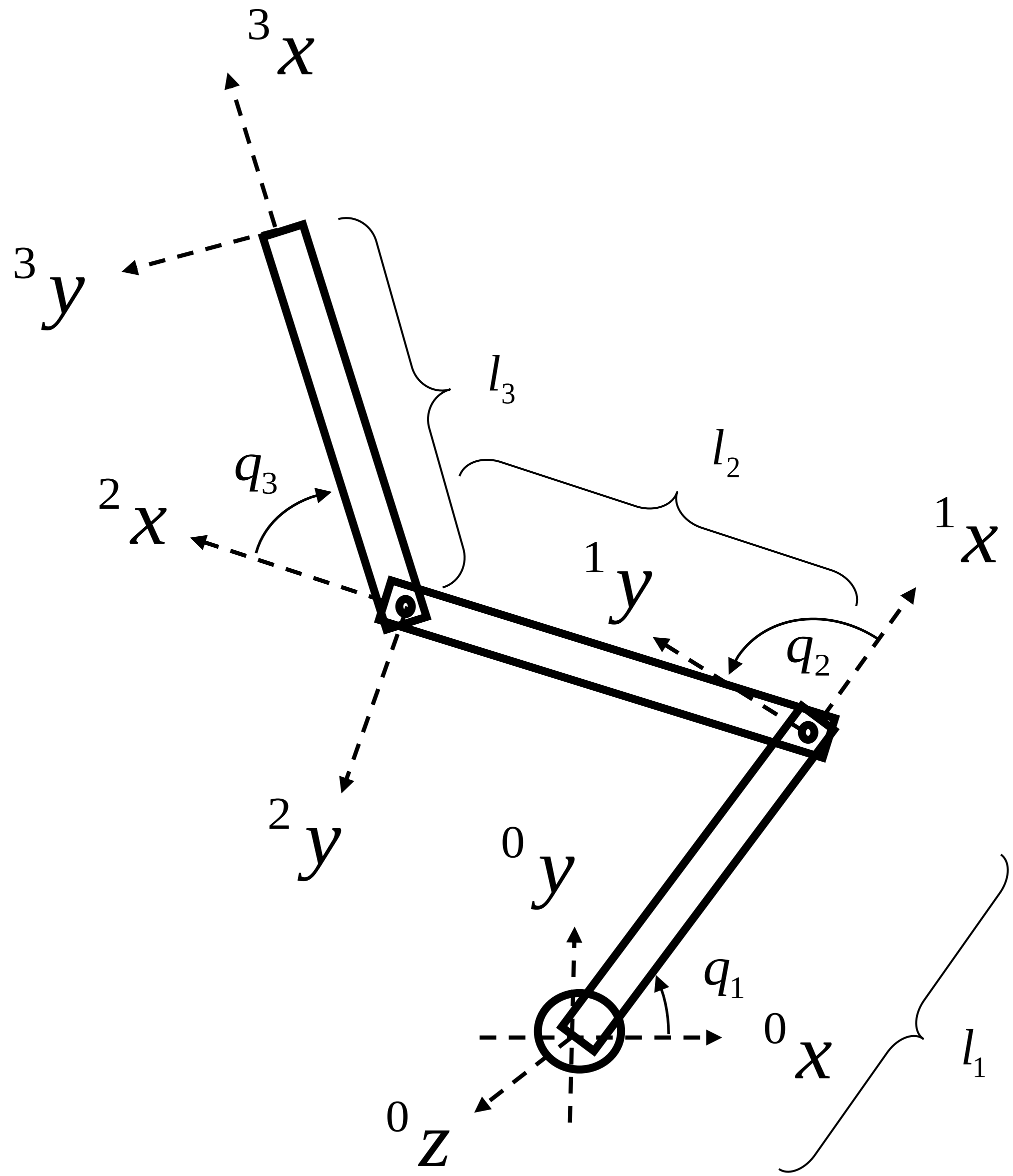


Forward kinematics: composition of homogeneous transforms

$$ {}^{0}T_{3} = {}^{0}T_{1}\,{}^{1}T_{2}\,{}^{2}T_{3} $$

$$ {}^{0}T_{1} = \begin{pmatrix} c_1 & -s_1 & 0 & l_1 c_1 \\ s_1 & c_1 & 0 & l_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

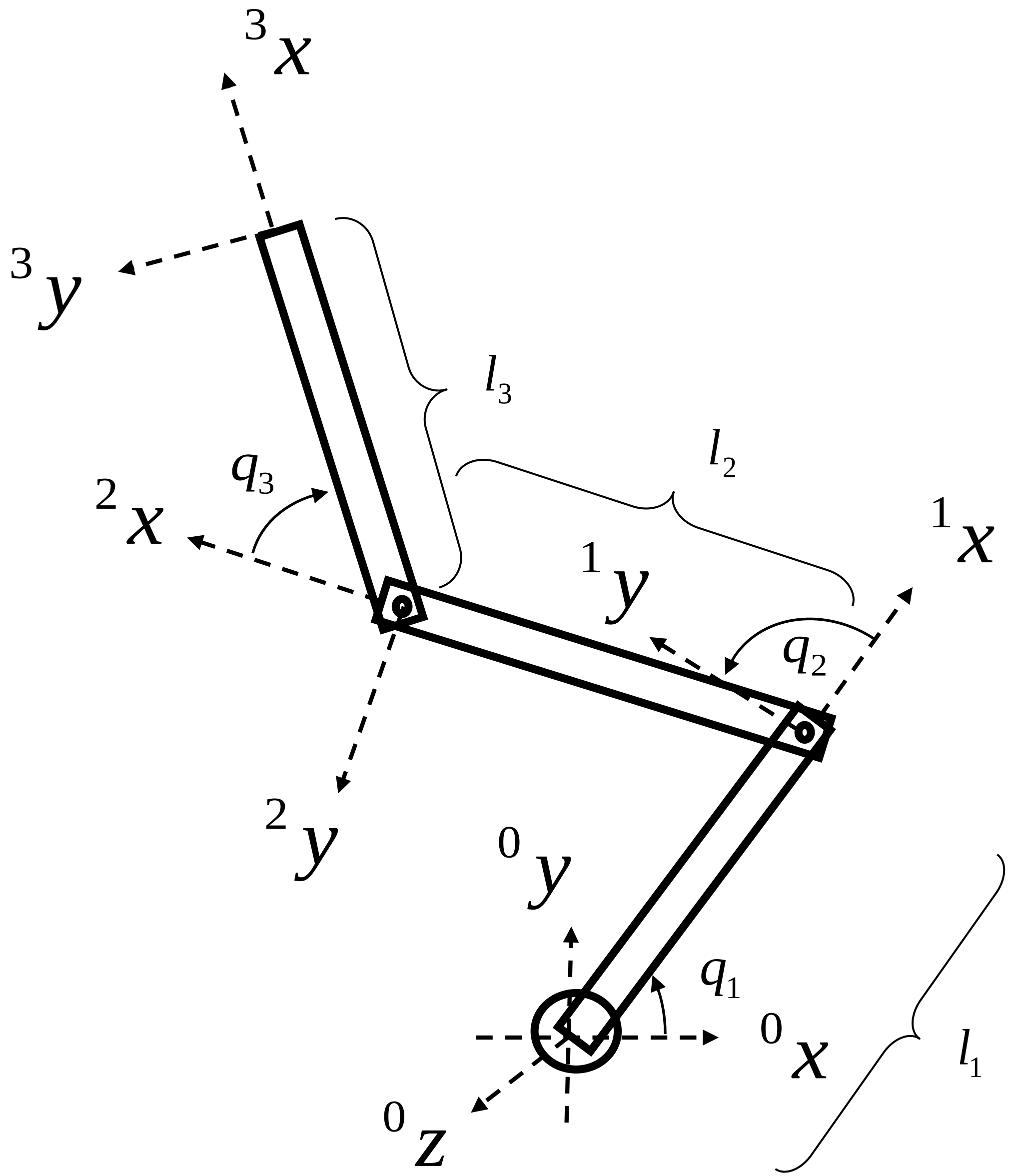
$$ {}^{1}T_{2} = \begin{pmatrix} c_2 & -s_2 & 0 & l_2 c_2 \\ s_2 & c_2 & 0 & l_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$



Forward kinematics: composition of homogeneous transforms

$$ {}^0T_3 = {}^0T_1\, {}^1T_2\, {}^2T_3 $$

$$ {}^2T_3 = \begin{pmatrix} c_3 & -s_3 & 0 & l_3 c_3 \\ s_3 & c_3 & 0 & l_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$



Remember those double-angle formulas…

$$\sin(\theta \pm \phi) = \sin(\theta)\cos(\phi) \pm \cos(\theta)\sin(\phi)$$

$$\cos(\theta \pm \phi) = \cos(\theta)\cos(\phi) \mp \sin(\theta)\sin(\phi)$$

Forward kinematics: composition of homogeneous transforms

$$ {}^{0}T_{3} = {}^{0}T_{1}\,{}^{1}T_{2}\,{}^{2}T_{3} $$

$$ {}^{0}T_{3} = \begin{pmatrix} c_1 & -s_1 & 0 & l_1c_1 \\ s_1 & c_1 & 0 & l_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_2 & -s_2 & 0 & l_2c_2 \\ s_2 & c_2 & 0 & l_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_3 & -s_3 & 0 & l_3c_3 \\ s_3 & c_3 & 0 & l_3s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

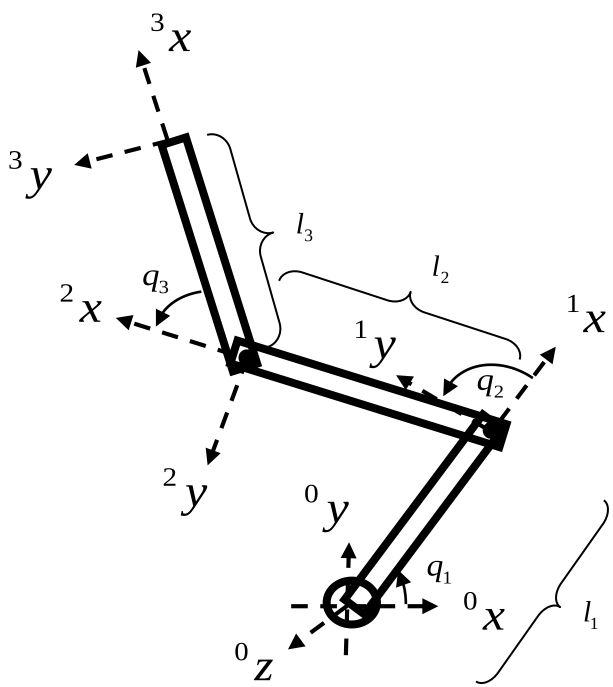
$$ {}^{0}T_{3} = \begin{pmatrix} \overbrace{\begin{matrix} c_{123} & -s_{123} & 0 \\ s_{123} & c_{123} & 0 \\ 0 & 0 & 1 \end{matrix}}^{R} & \overbrace{\begin{matrix} l_1c_1 + l_2c_{12} + l_3c_{123} \\ l_1s_1 + l_2s_{12} + l_3s_{123} \\ 0 \end{matrix}}^{P_{B\,org}} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{pmatrix} $$

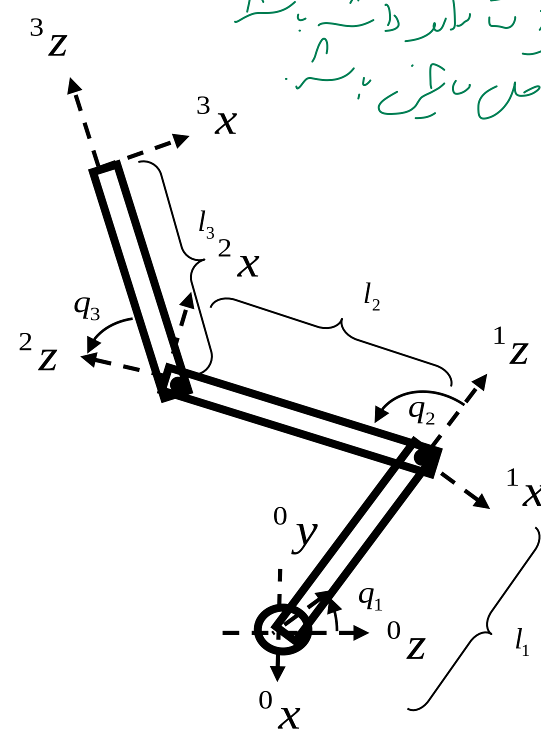
۳×۳

DH parameters

- There are a large number of ways that homogeneous transforms can encode the kinematics of a manipulator
- We will sacrifice some of this flexibility for a more systematic approach: DH (Denavit-Hartenberg) parameters.
- DH is a standard for describing a series of transforms for arbitrary mechanisms.

یک حل مسئله به جهت انتخاب دستگاه، ما به هر شکل، هر کدام از سیستم‌ها
تکمیل باشد ولی ایده را نداریم داشته باشیم
می‌توانیم این مراحل را طی کنیم باشیم.





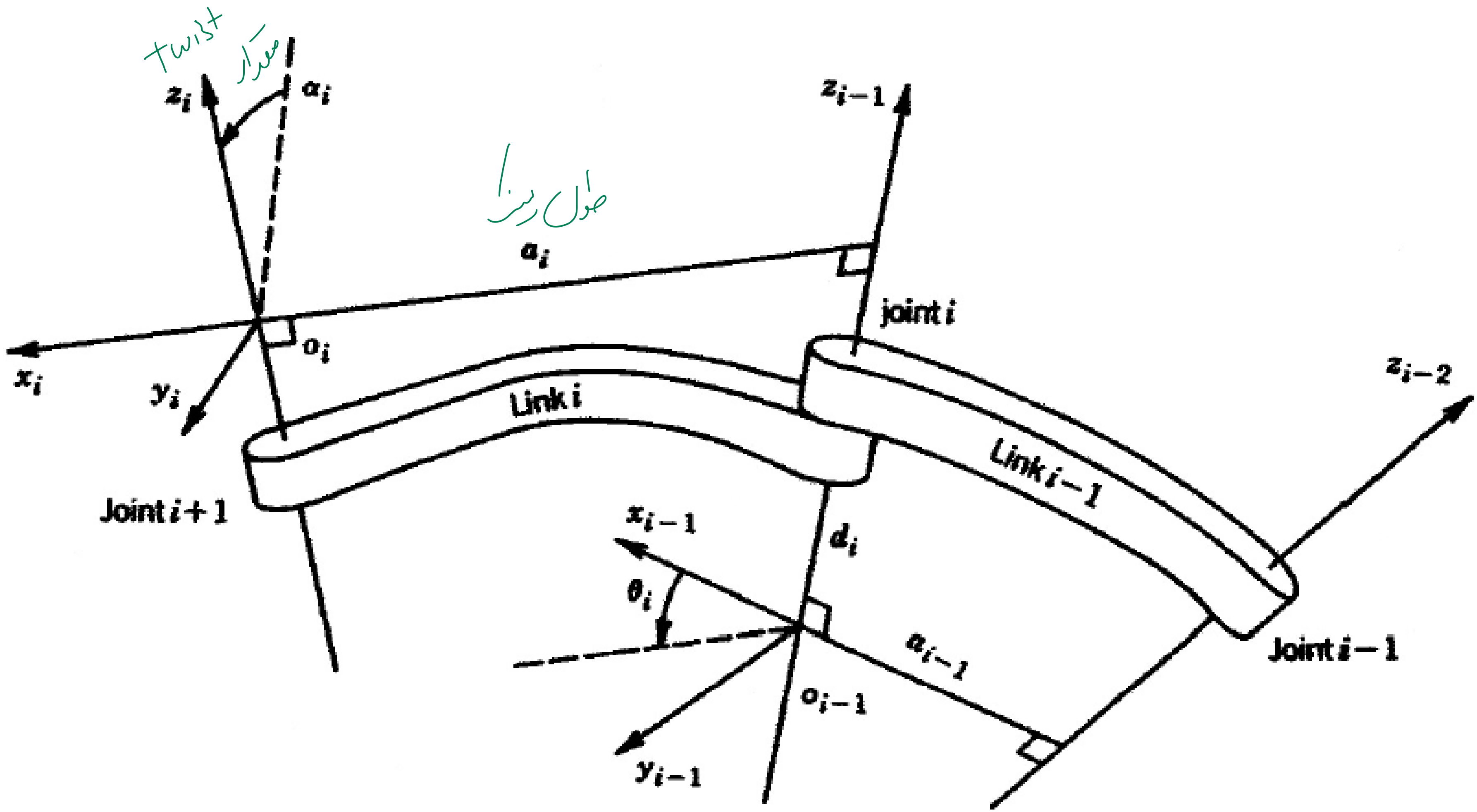
DH Parameterization

- Assumptions
  - Axis x_i intersects axis z_{i-1}
  - Axis x_i is perpendicular to axis z_{i-1}
- Therefore four DOFs remain as parameters
  - Link Parameters
    - length and twist (a_i, α_i)
  - Joint Parameters
    - offset and angle(d_i, θ_i)

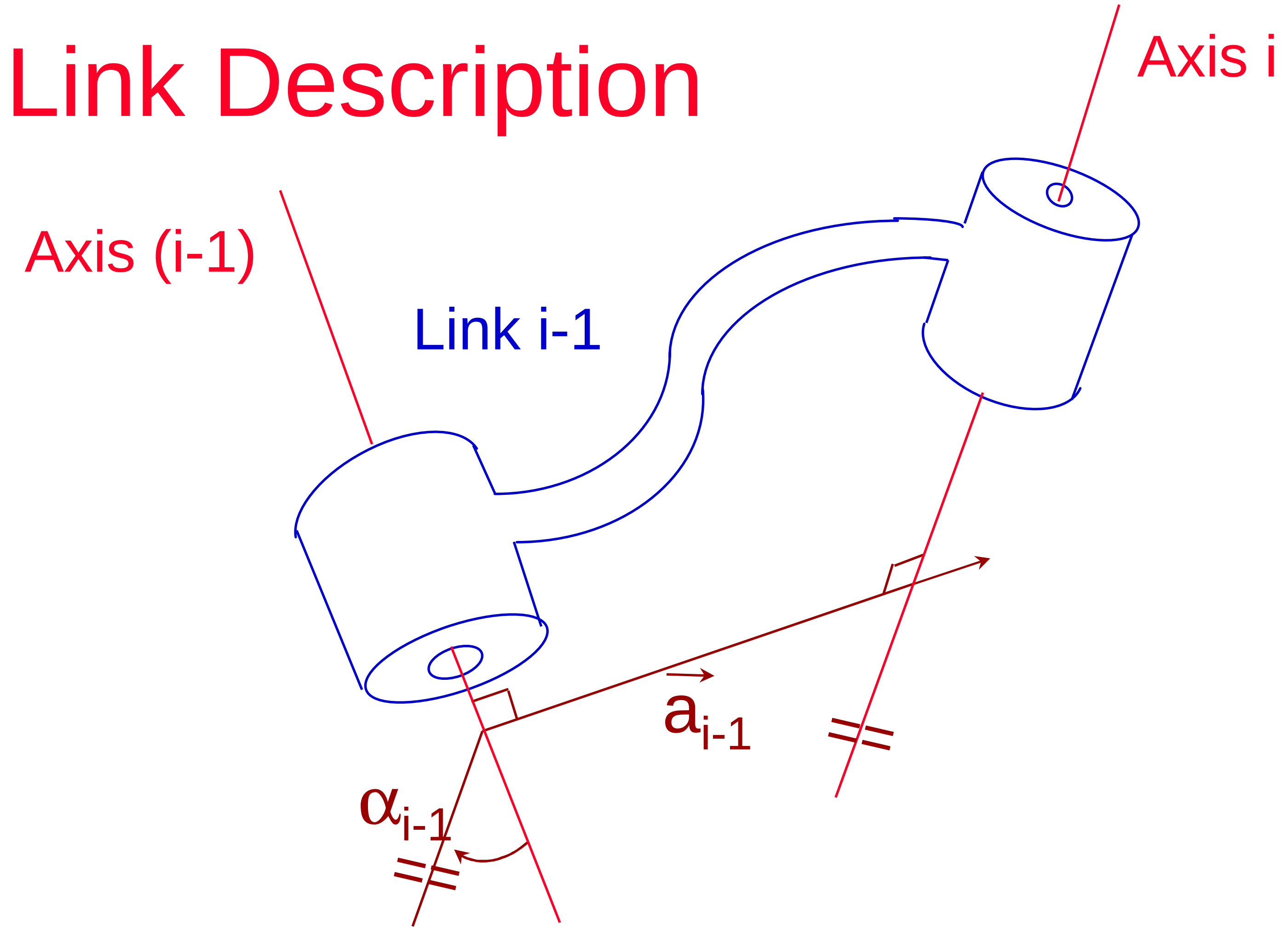
Joint Parameters

- Prismatic: variable d_i
- Revolute: variable θ_i

حداقل یکی باید متغیر باشد



Link Description

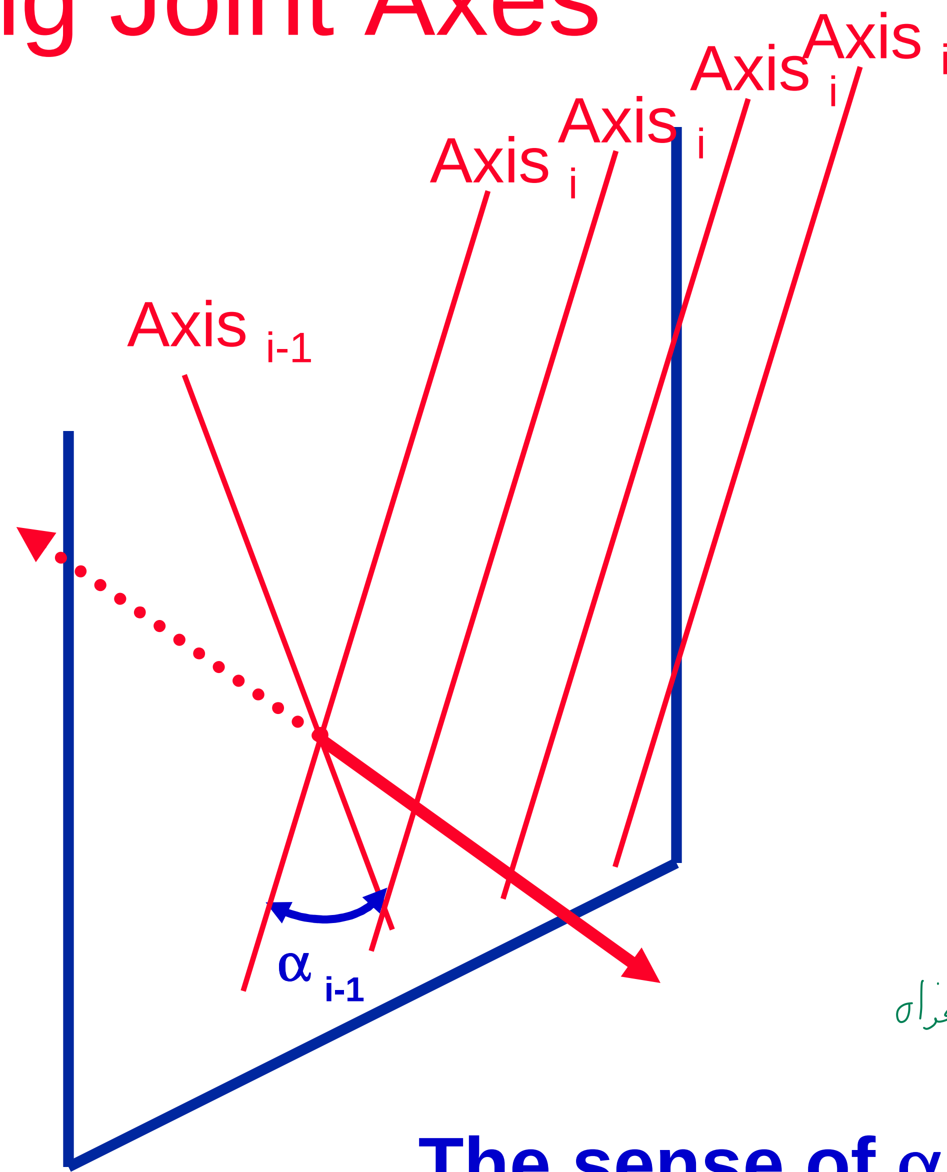


a_{i-1}: Link Length - mutual perpendicular

unique except for parallel axis

α_{i-1}: Link Twist - measured in the right-hand sense about $\vec{a}_{i-1}$

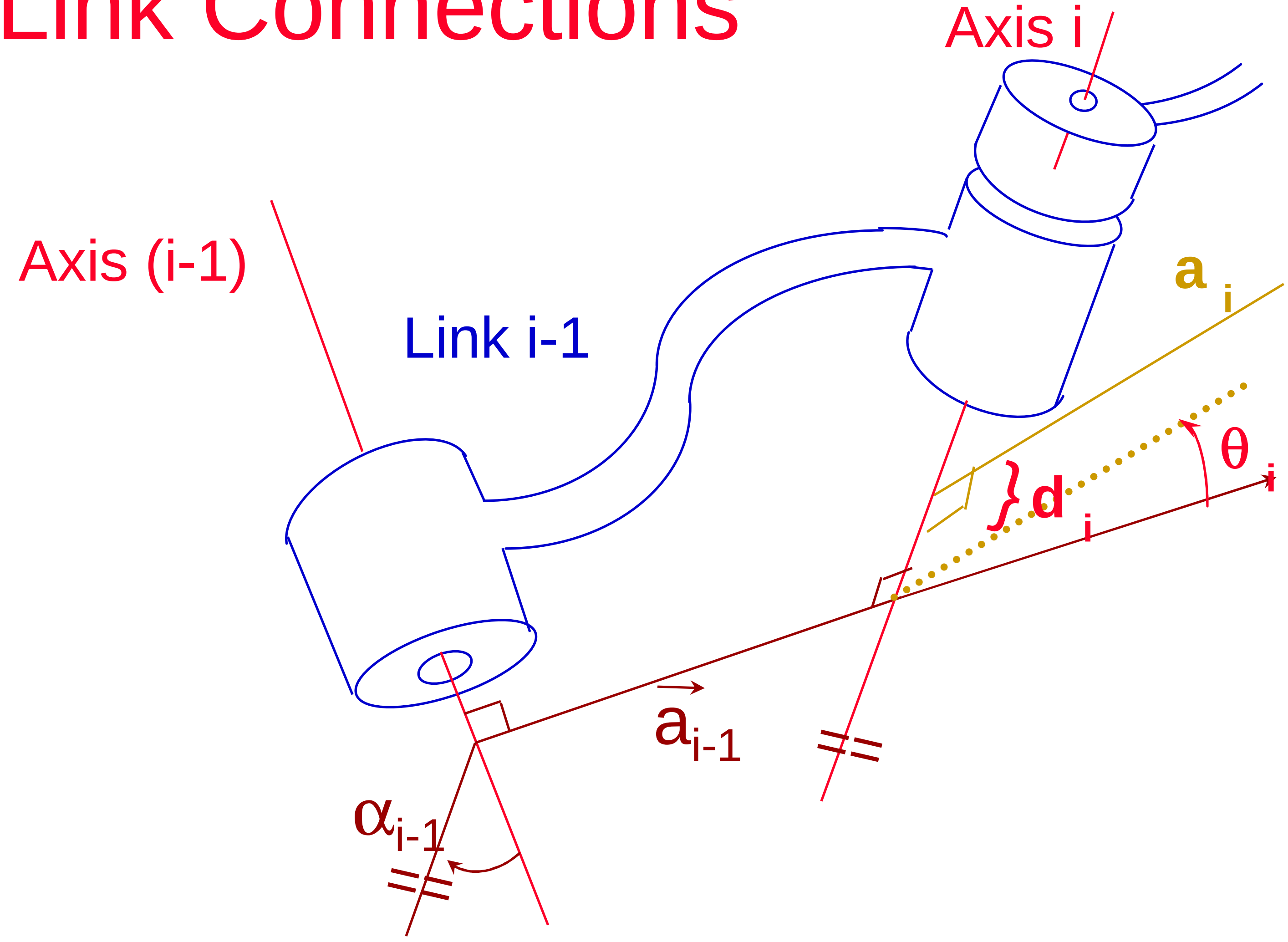
Intersecting Joint Axes



جهت زاویه دلخواه

The sense of α_{i-1} is free

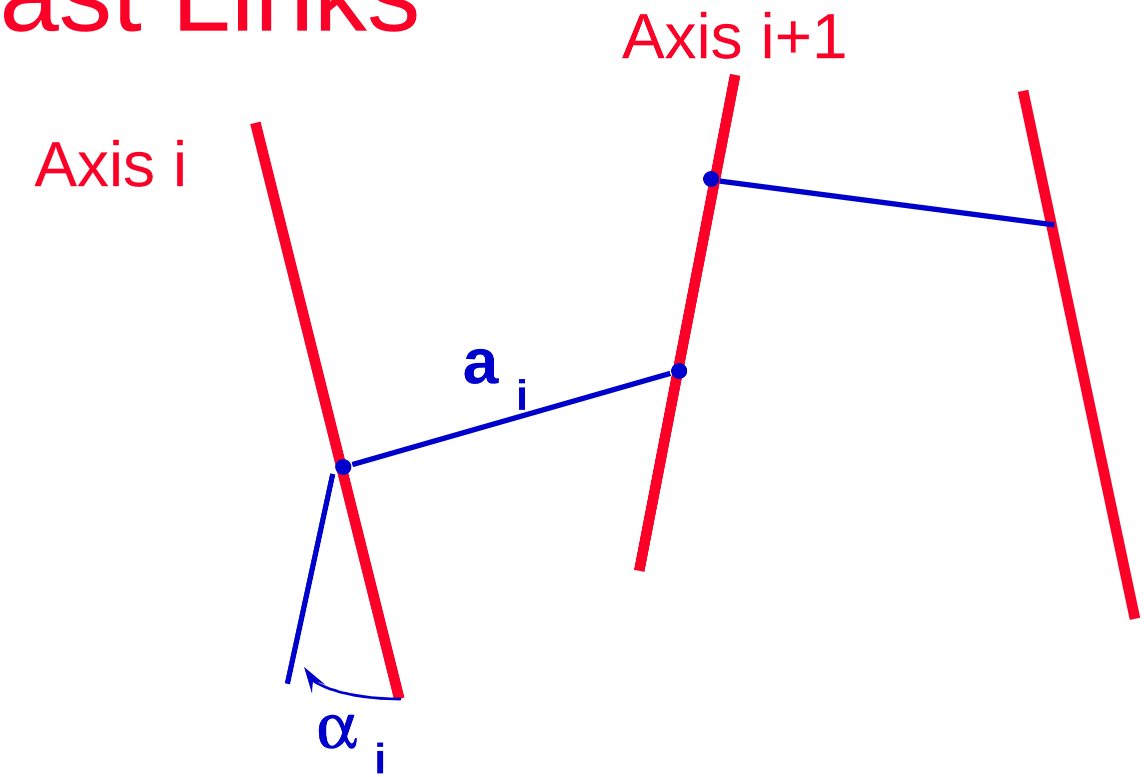
Link Connections



d $_i$: Link Offset -- variable if joint i is *prismatic*

θ $_i$: Joint Angle -- variable if joint i is *revolute*

First & Last Links



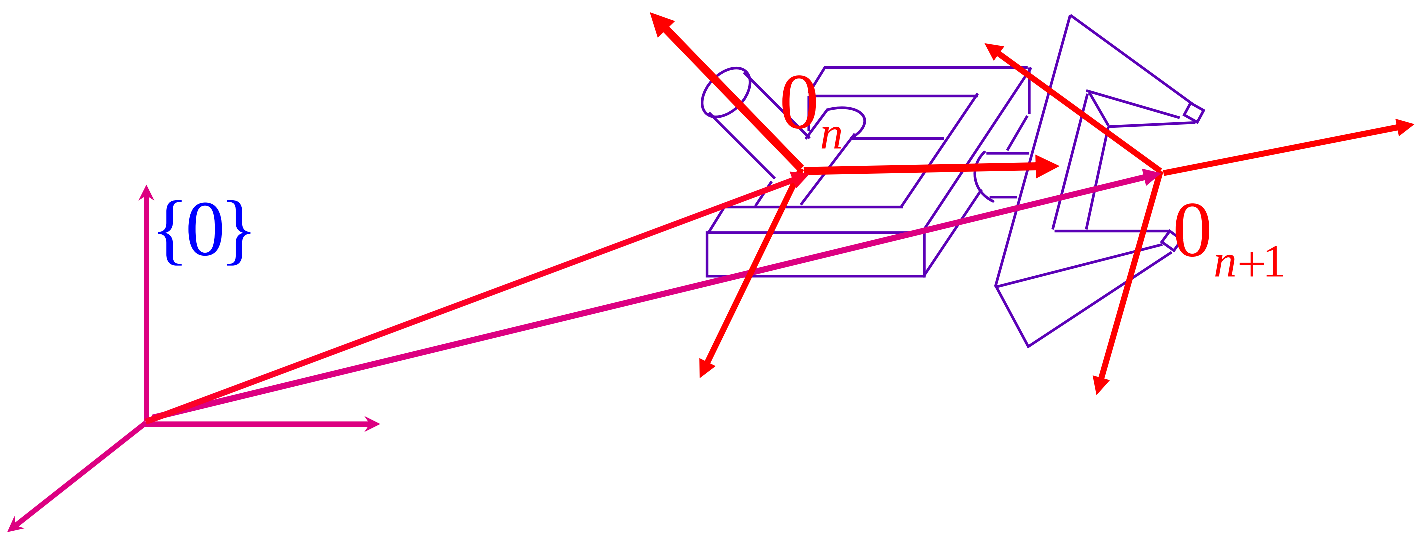
a_i and α_i depend on joint axes i and i+1

Axes 1 to n: determined

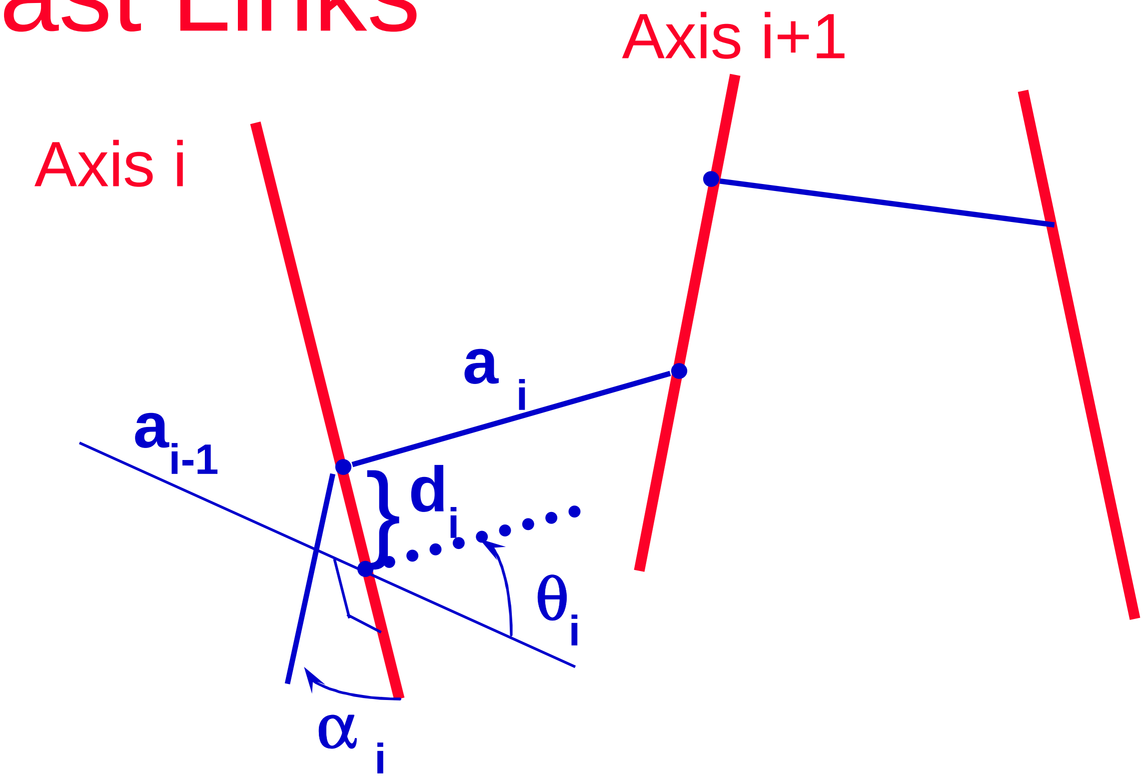
$\Rightarrow$ a_1 , a_2 ... a_{n-1} and α_1, α_2 ... α_{n-1}

Convention: $\boxed{a_0 = a_n = 0 \text{ and } \alpha_0 = \alpha_n = 0}$

End-Efector Frame



First & Last Links



θ_i and d_i depend on links i-1 and i

⇒ $\theta_2, \theta_3, \ldots, \theta_{n-1}$ and $d_2, d_3, \ldots, d_{n-1}$

Convention: set the constant parameters to zero

Following joint type: d_1 or $\theta_1 = 0$ and d_n or $\theta_n = 0$

Denavit-Hartenberg Parameters

4 D-H parameters $(\alpha_i\,,\ a_i\,,\ d_i\,,\ \theta_i)$

نیاز به دستگاه مختصات داریم.

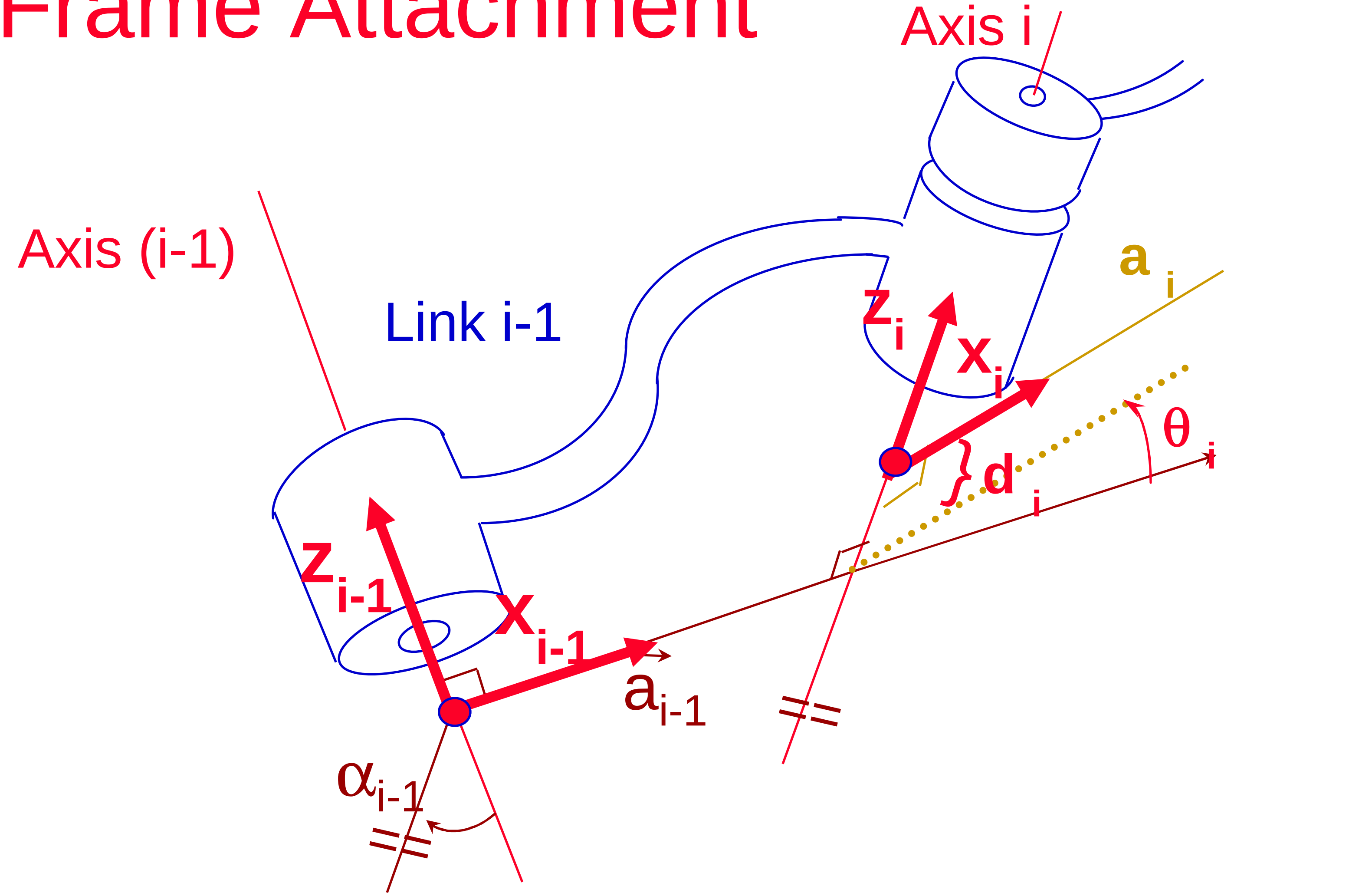
3 fixed link parameters

1 joint variable $\begin{cases} \theta_i & \text{revolute joint} \\ d_i & \text{prismatic joint} \end{cases}$

α_i and a_i : describe the Link i

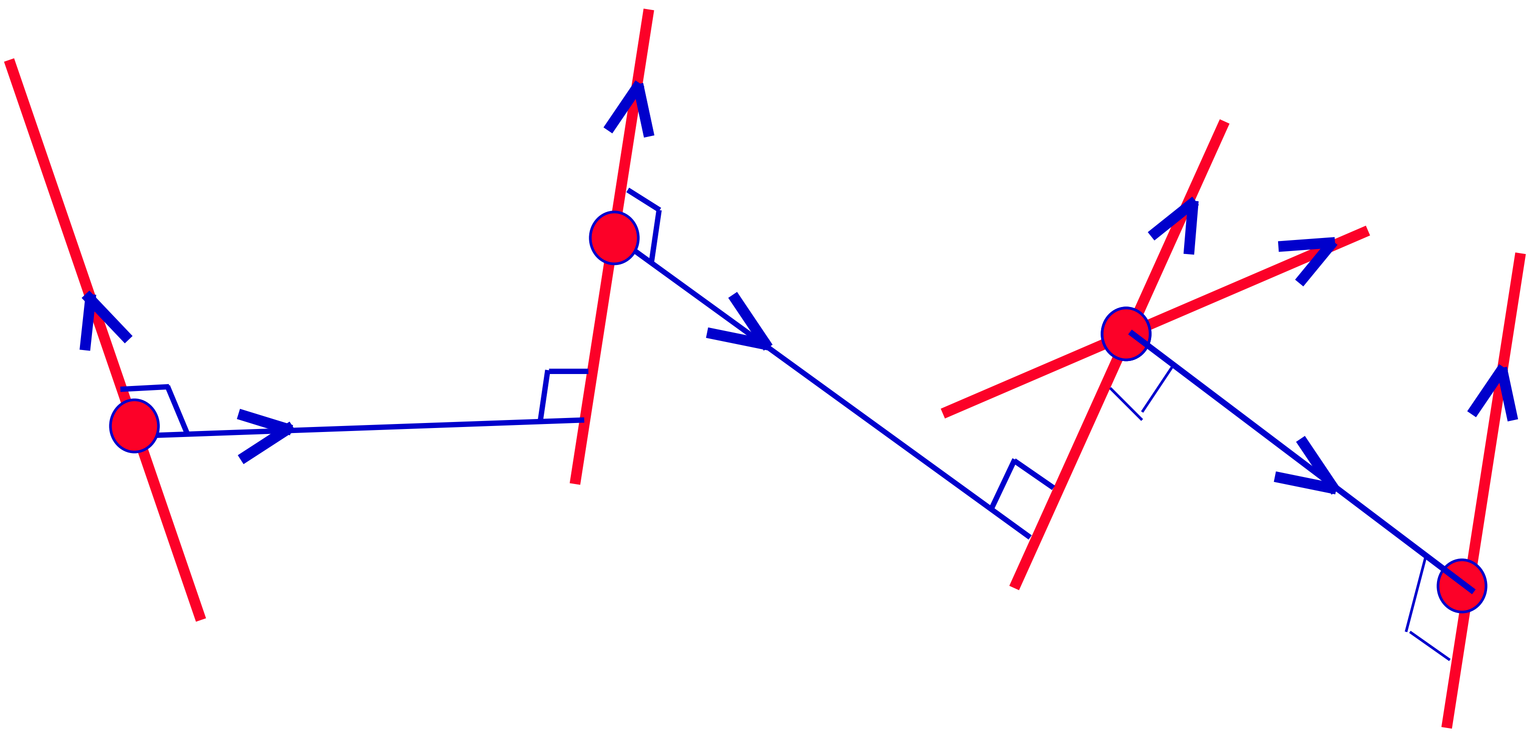
d_i and θ_i : describe the Link's connection

Frame Attachment



y-vectors: complete right-hand frames

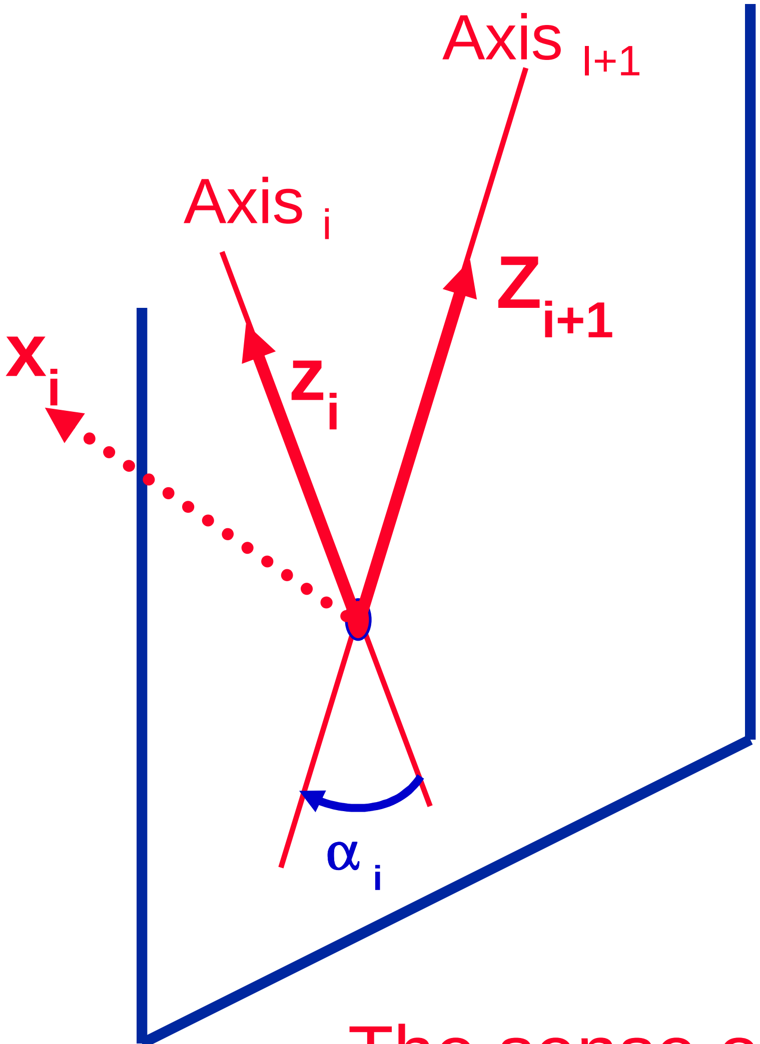
Summary – Frame Attachment



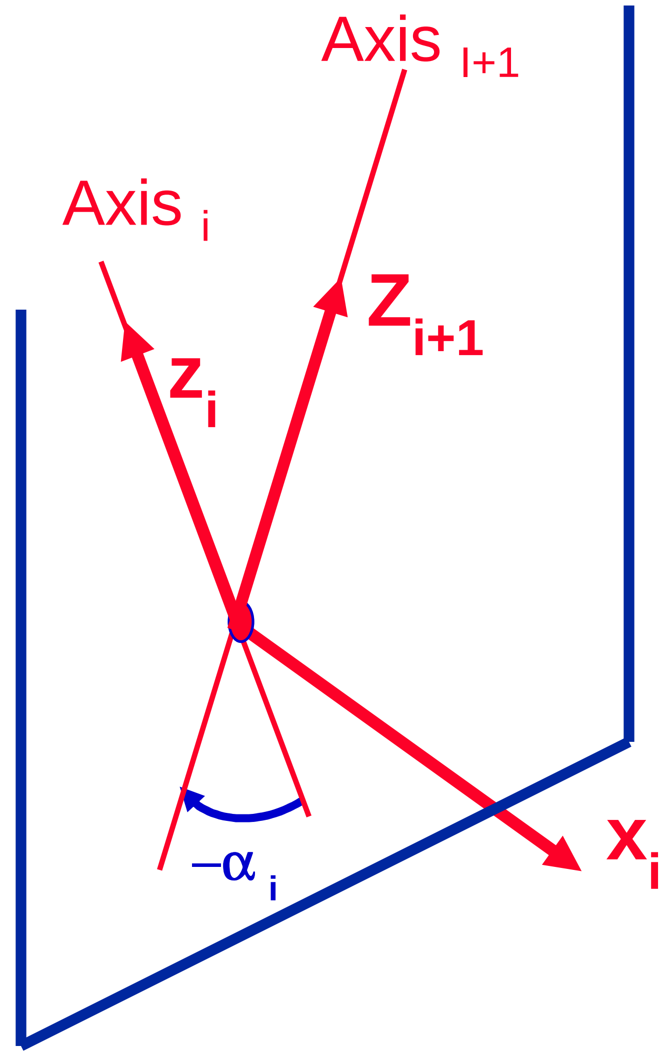
1. Normals
2. Origins

3. Z-axes
4. X-axes

Intersecting Joint Axes



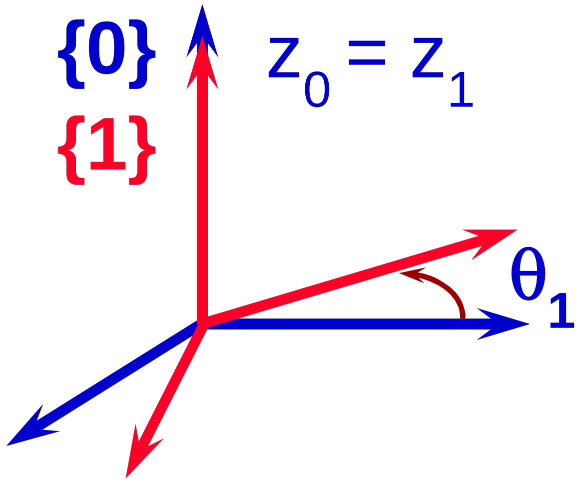
جهت x
دحرکه



The sense of α_i is determined by the direction of x

First Link

Revolute



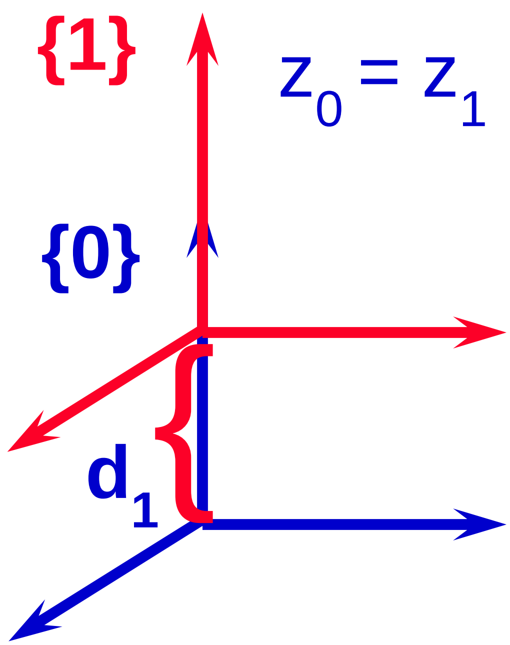
$a_0 = 0$

$\alpha_0 = 0$

$d_1 = 0$

$\theta_1 = 0 \longrightarrow \{0\} \equiv \{1\}$

Prismatic



$a_0 = 0$

$\alpha_0 = 0$

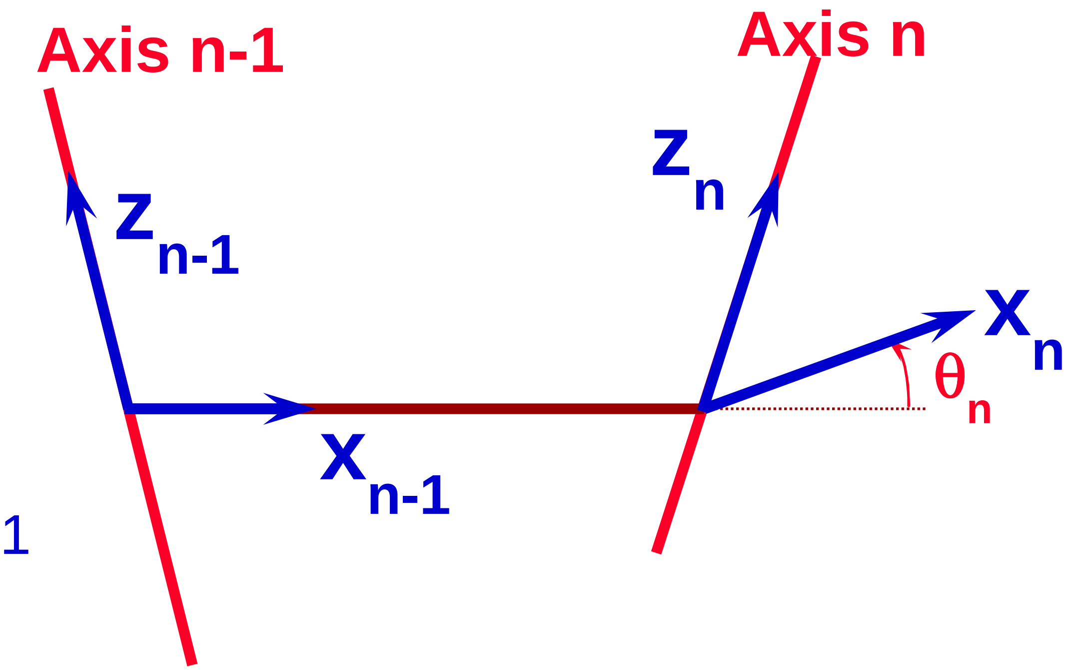
$\theta_1 = 0$

$\mathbf{d_1 = 0} \longrightarrow \mathbf{\{0\} \equiv \{1\}}$

Last Link

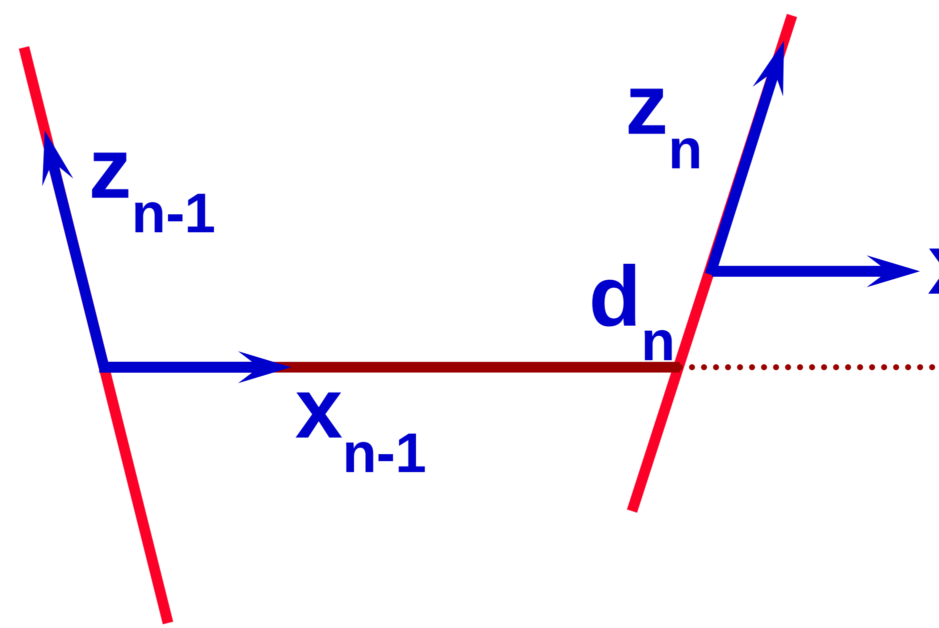
Revolute $\boxed{d_n = 0}$

$\theta_n = 0 \longrightarrow x_n = x_{n-1}$

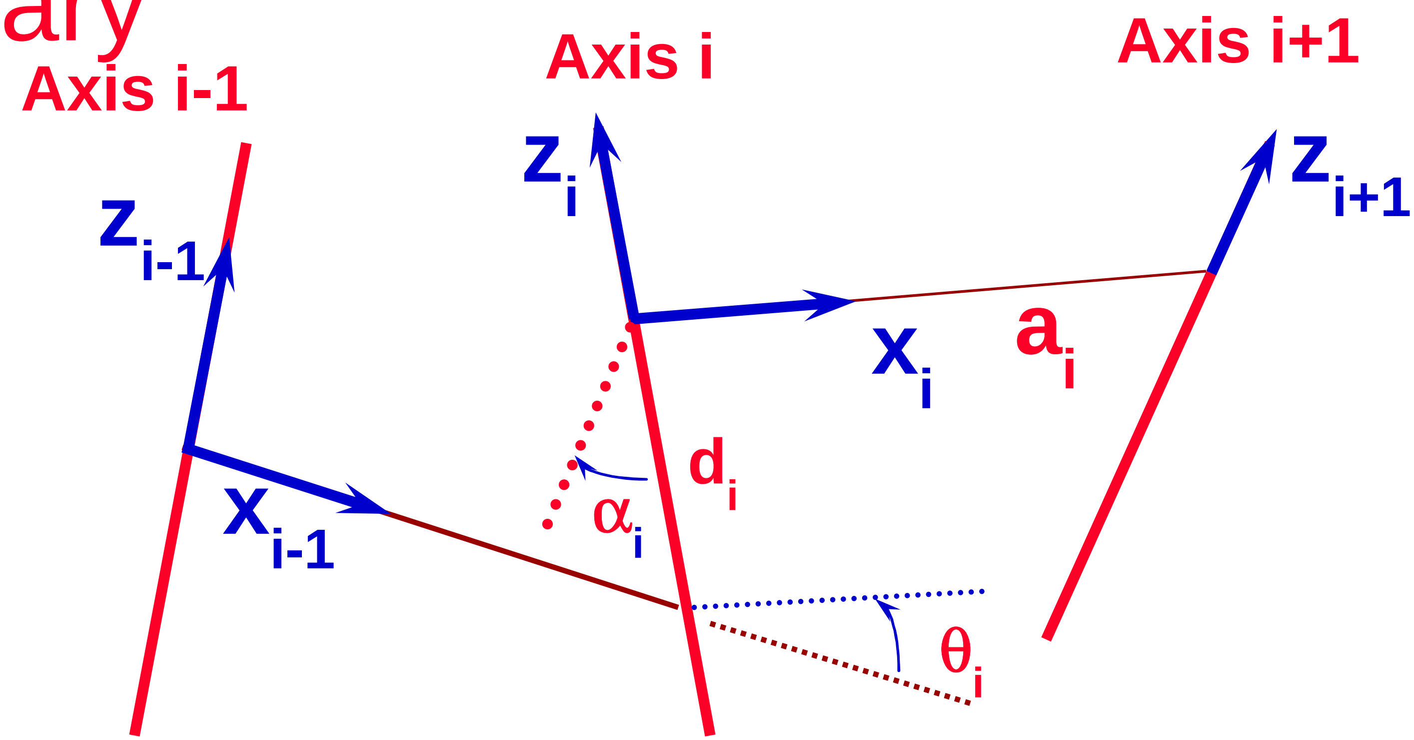


Prismatic $\boxed{\theta_n = 0}$

$d_n = 0 \longrightarrow x_n = x_{n-1}$



Summary



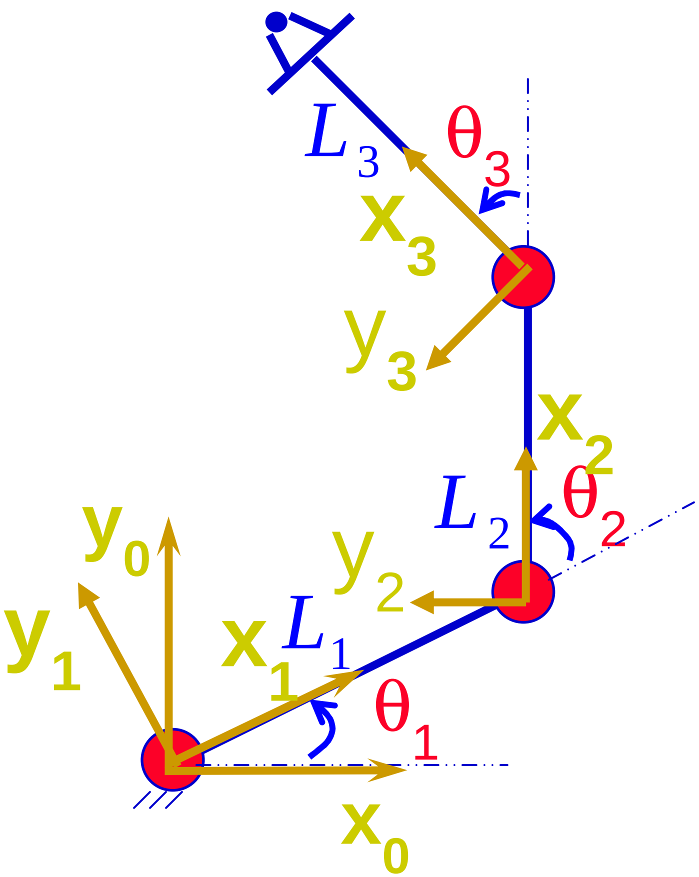
a_i : distance (z_i , z_{i+1}) along x_i

α_i : angle (z_i , z_{i+1}) about x_i

d_i : distance (x_{i-1} , x_i) along z_i

θ_i : angle (x_{i-1} , x_i) about z_i

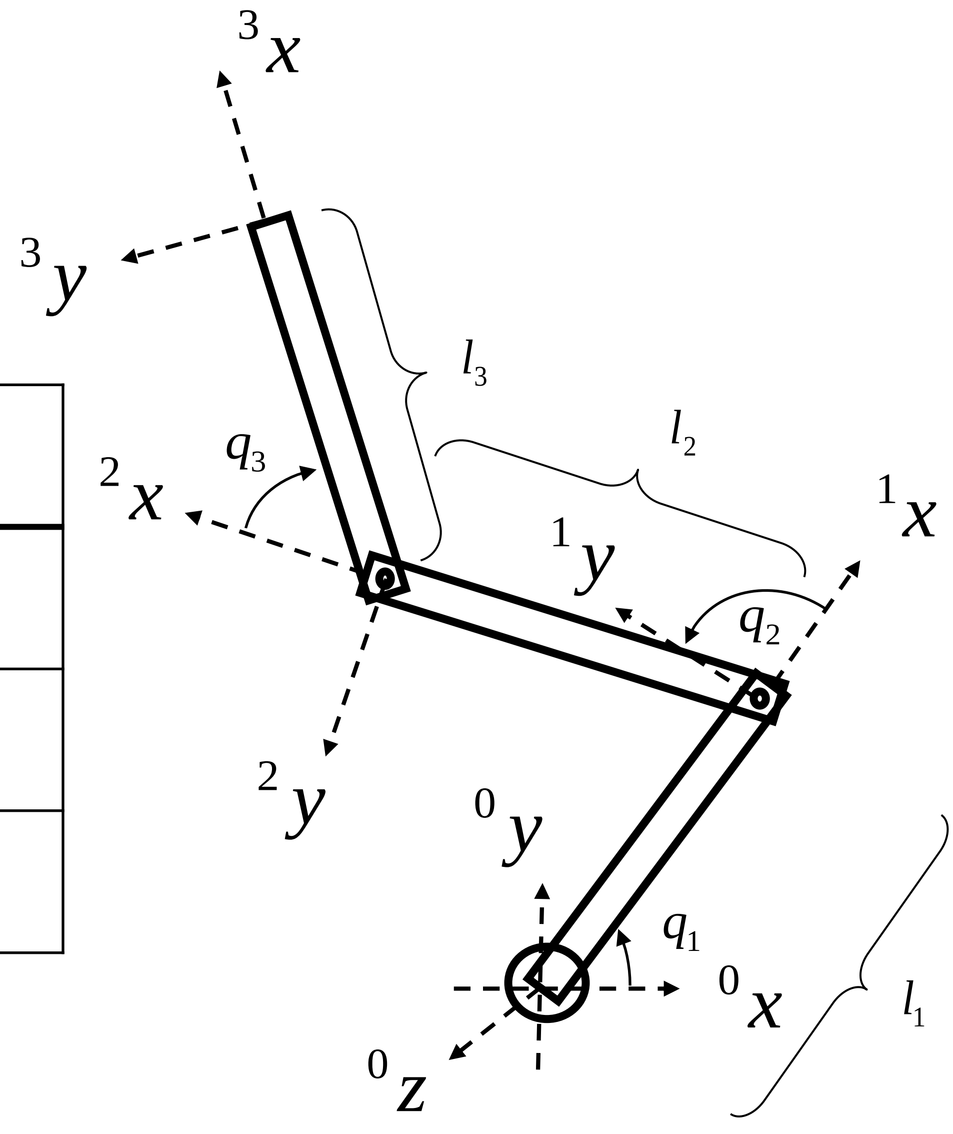
Example – RRR Arm



i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	0	L_2	0	θ_3

Example 1: DH parameters

	a_i	α_i	d_i	θ_i
1	l_1	0	0	q_1
2	l_2	0	0	q_2
3	l_3	0	0	q_3



Forward kinematics: DH parameters

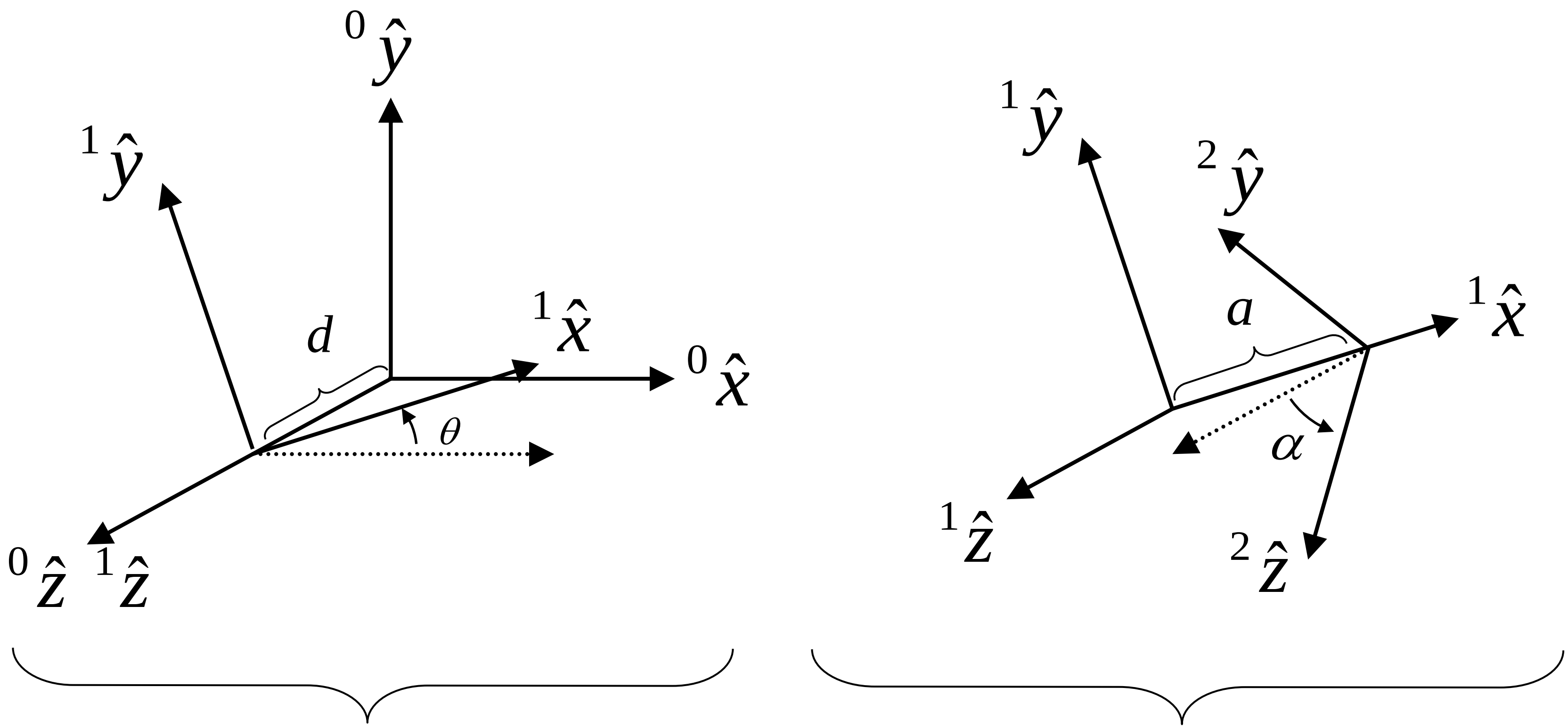
- A series of transforms is written as a table:

xform	a_i	α_i	d_i	θ_i
1	a_1	α_1	d_1	θ_1
2	a_2	α_2	d_2	θ_2

Forward kinematics: DH parameters

Four DH parameters: $\left(a_i \quad \alpha_i \quad d_i \quad \theta_i\right)$

$$T = T_{rot(z,\theta_i)} T_{trans(z,d_i)} T_{rot(x,\alpha_i)} T_{trans(x,a_i)}$$



First, translate by d_i along *z* axis

and rotate by θ_i about *z* axis

Then, translate by a_i along *x* axis

and rotate by α_i about *x* axis

Forward kinematics: DH parameters

These four DH parameters,

$$\begin{pmatrix} a_i & \alpha_i & d_i & \theta_i \end{pmatrix}$$

represent the following homogeneous matrix:

$$T = \underbrace{\begin{pmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}} \underbrace{\begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}$$

Then, translate by d_i along z_i axis and rotate by θ_i about z_i axis

First, rotate by α_i about x_i axis and translate by a_i along x_i axis

Forward kinematics: DH parameters

$$T = \begin{pmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

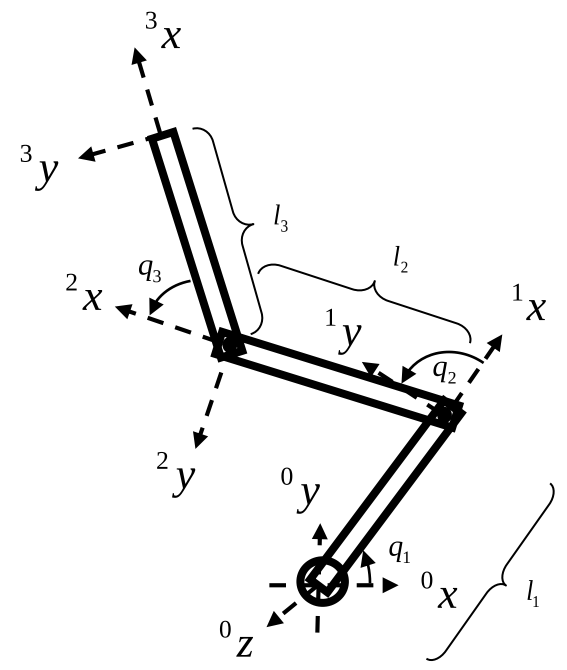
Example 1: DH parameters

$$
{}^{0}T_1 = \begin{pmatrix} c_{q_1} & -s_{q_1} & 0 & l_1 c_{q_1} \\ s_{q_1} & c_{q_1} & 0 & l_1 s_{q_1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
$$

$$
{}^{1}T_2 = \begin{pmatrix} c_{q_2} & -s_{q_2} & 0 & l_2 c_{q_2} \\ s_{q_2} & c_{q_2} & 0 & l_2 s_{q_2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
$$

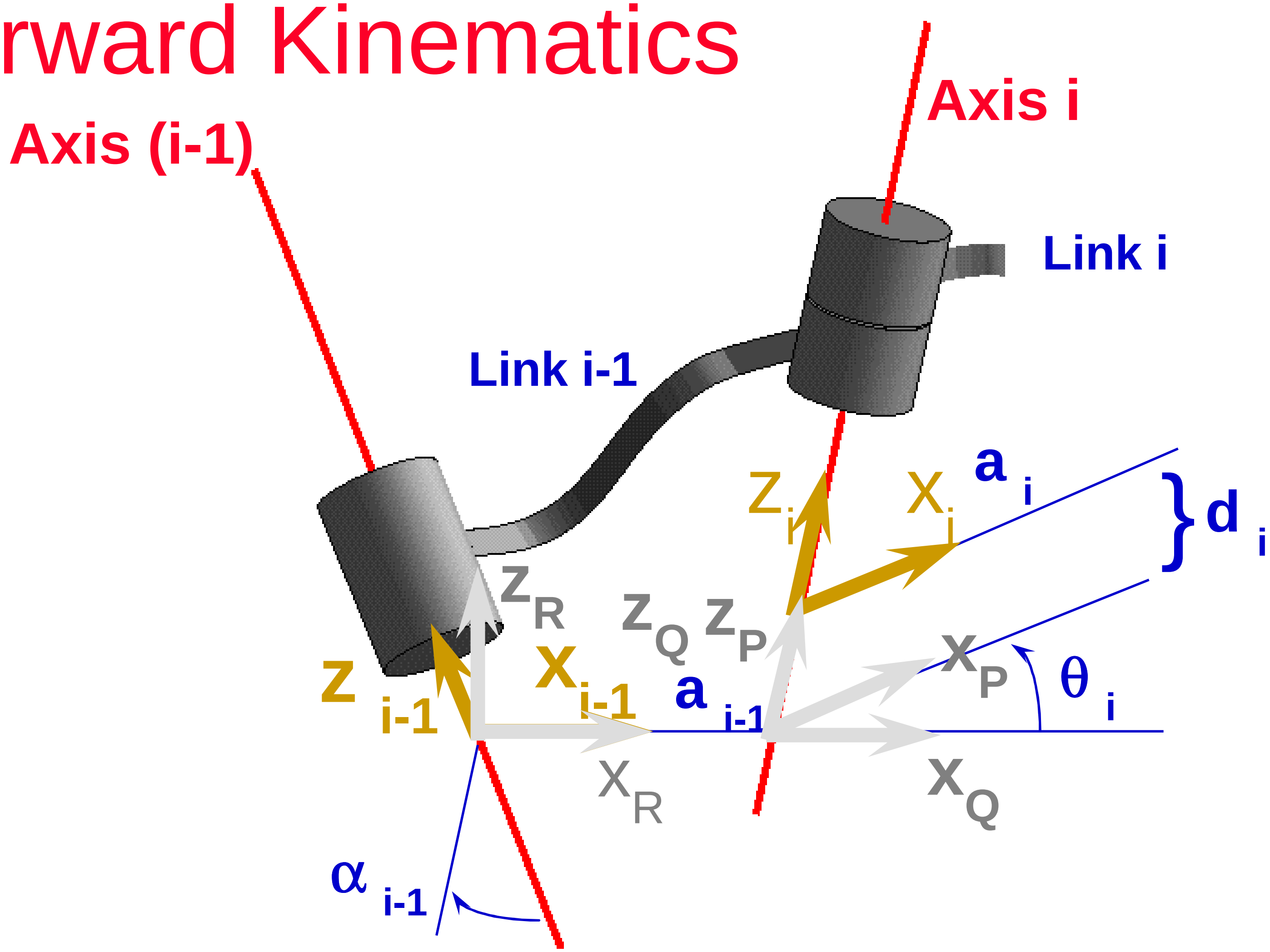
$$
{}^{2}T_3 = \begin{pmatrix} c_{q_3} & -s_{q_3} & 0 & l_3 c_{q_3} \\ s_{q_3} & c_{q_3} & 0 & l_3 s_{q_3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
$$

$$
{}^{0}T_3 = {}^{0}T_1 \, {}^{1}T_2 \, {}^{2}T_3
$$



	a_i	α_i	d_i	θ_i
1	l_1	0	0	q_1
2	l_2	0	0	q_2
3	l_3	0	0	q_3

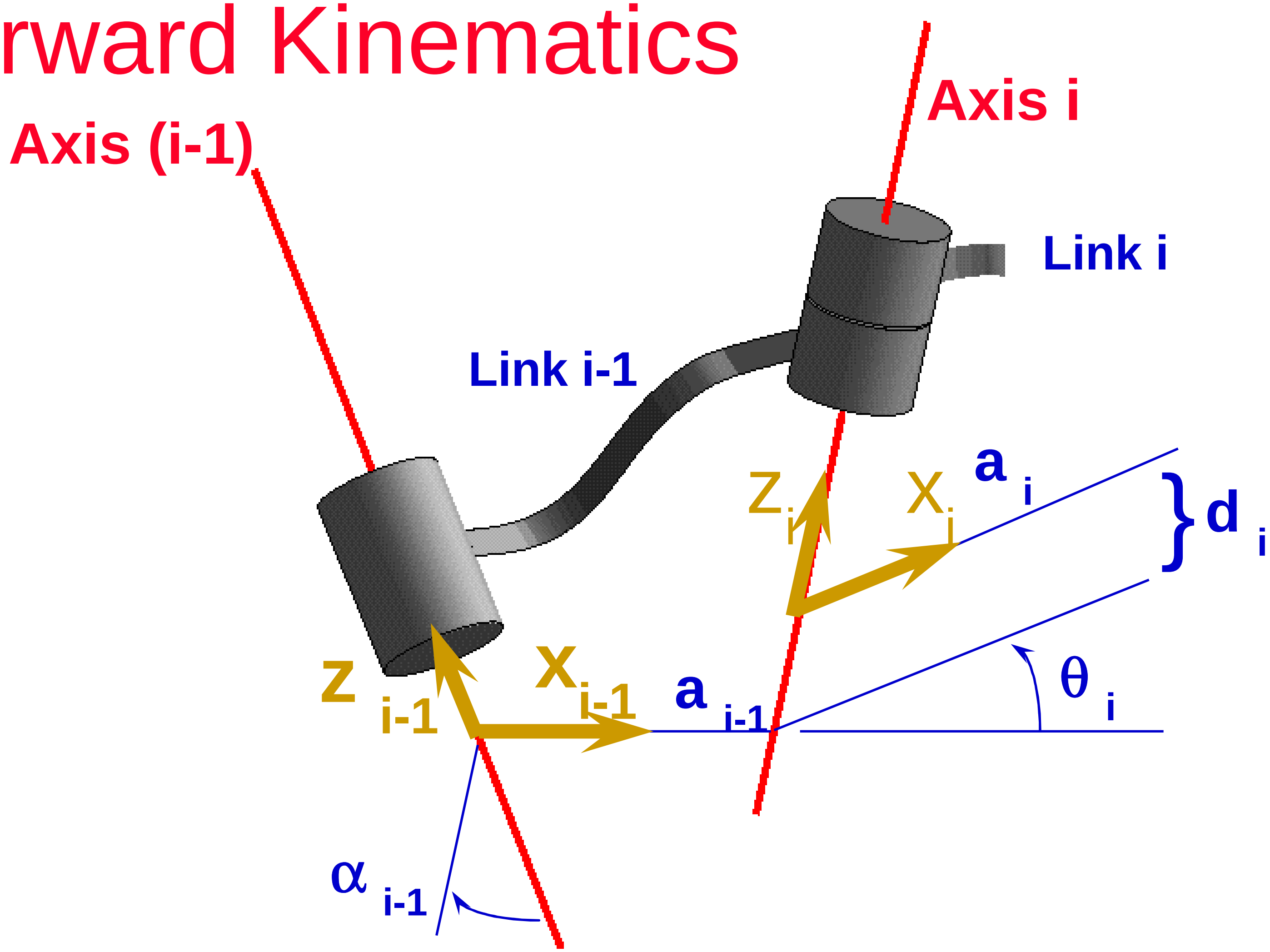
Forward Kinematics



$$ {}^{i-1}_{\ \ i}T = {}^{i-1}_{\ \ R}T \ {}^{R}_{Q}T \ {}^{Q}_{P}T \ {}^{P}_{i}T $$

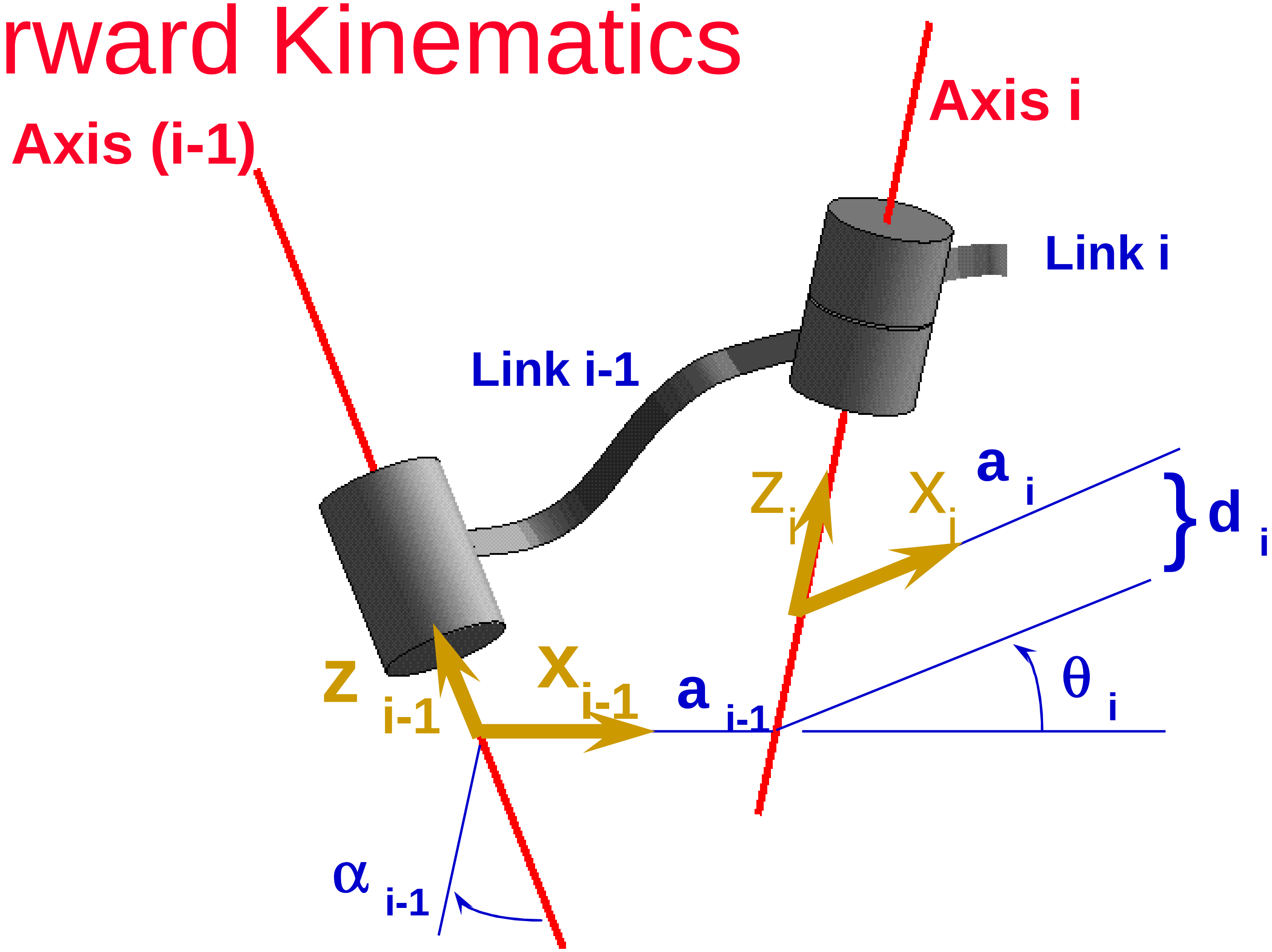
$$ {}^{i-1}_{\ \ i}T_{(\alpha_{i-1},\, a_{i-1},\, \theta_i,\, d_i)} = R_x(\alpha_{i-1})\, D_x(a_{i-1})\, R_z(\theta_i)\, D_z(d_i) $$

Forward Kinematics



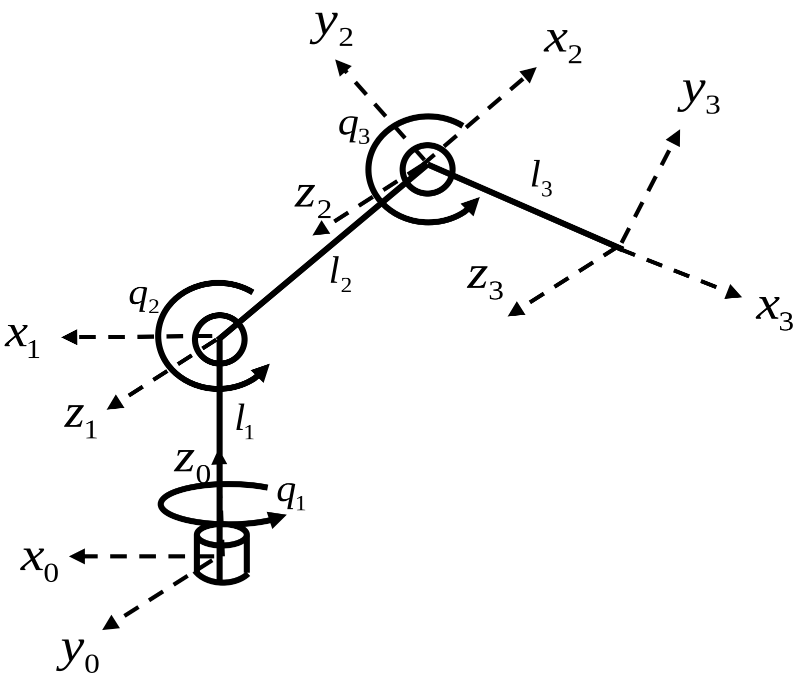
$$
{}^{i-1}_{1}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i\, c\alpha_{i-1} & c\theta_i\, c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}\, d_i \\ s\theta_i\, s\alpha_{i-1} & c\theta_i\, s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}\, d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Forward Kinematics



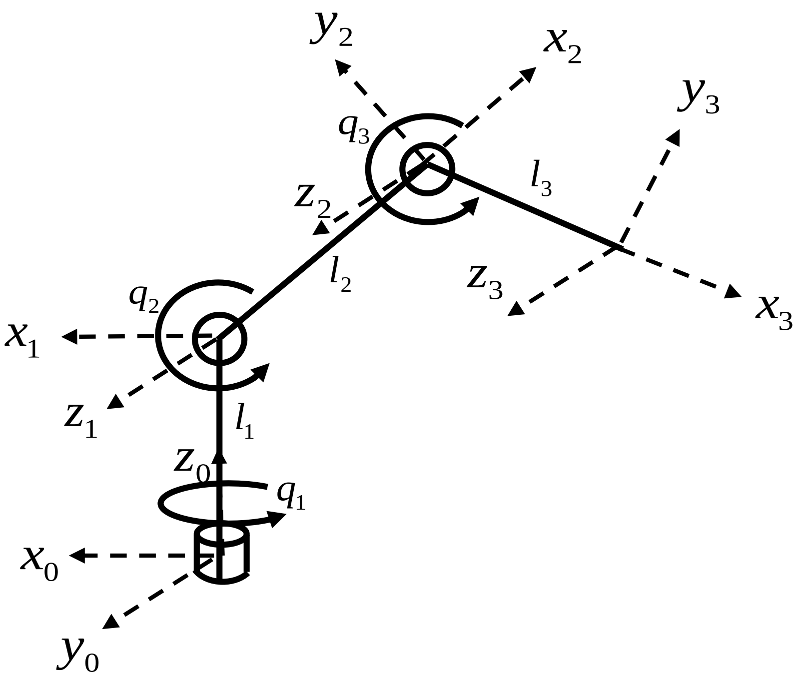
Forward Kinematics: $^{0}_{N}T = \, ^{0}_{1}T \;\; ^{1}_{2}T \; \ldots \; ^{N-1}_{N}T$

Example 2: DH parameters

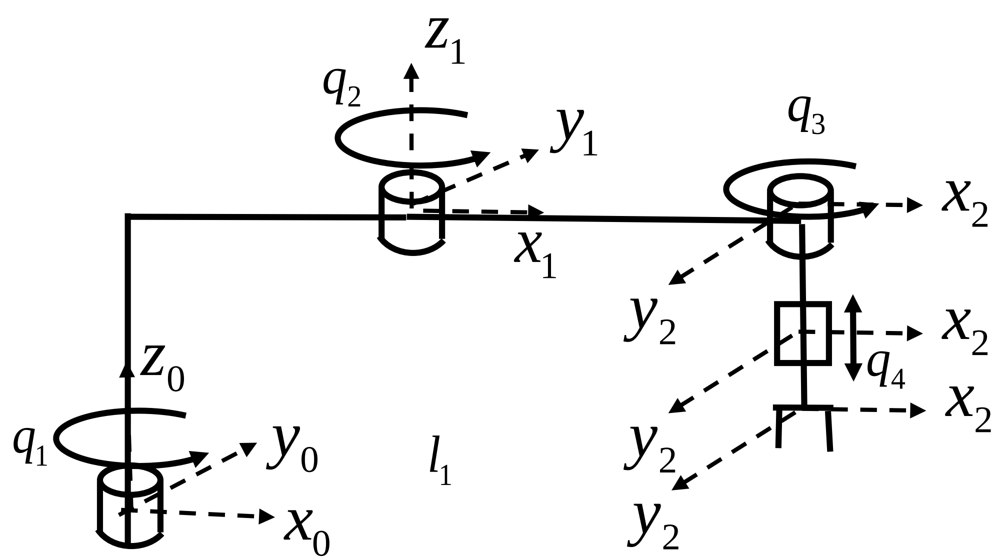


Example 2: DH parameters

	a_i	α_i	d_i	θ_i
1	0	$-\frac{\pi}{2}$	l_1	q_1
2	l_2	0	0	$q_2-\frac{\pi}{2}$
3	l_3	0	0	q_3

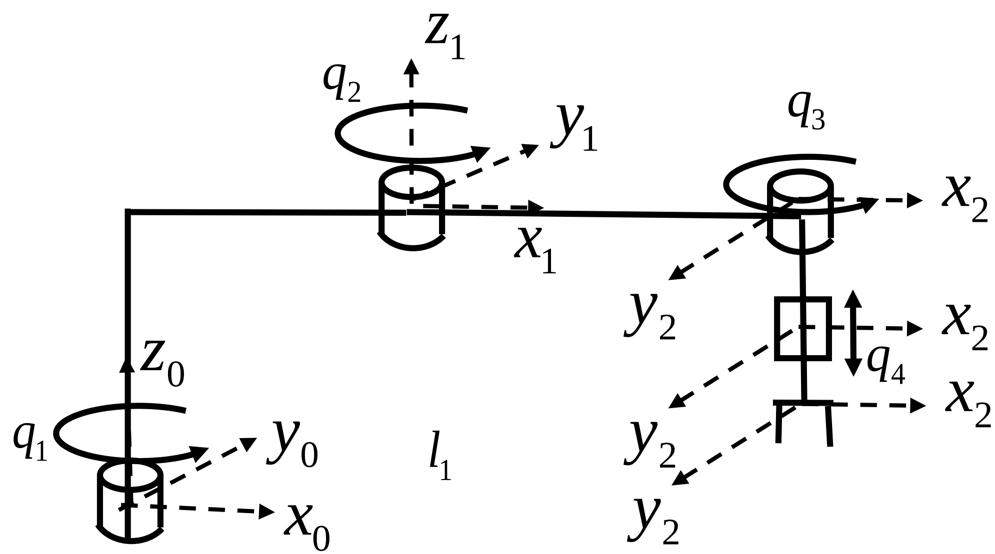


Example 3: DH parameters

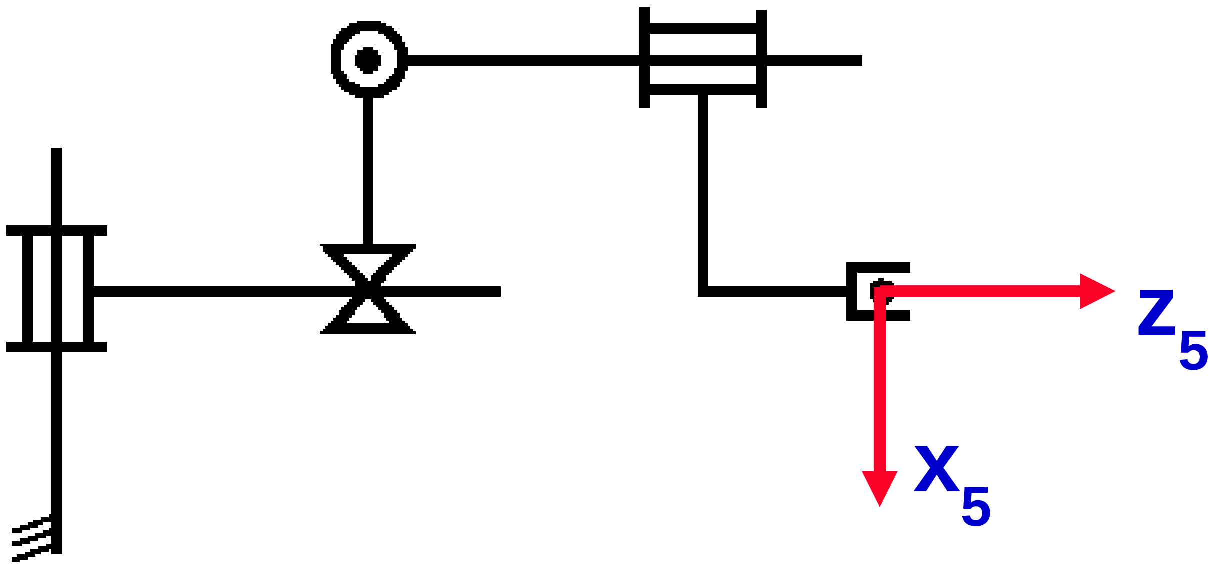


Example 3: DH parameters

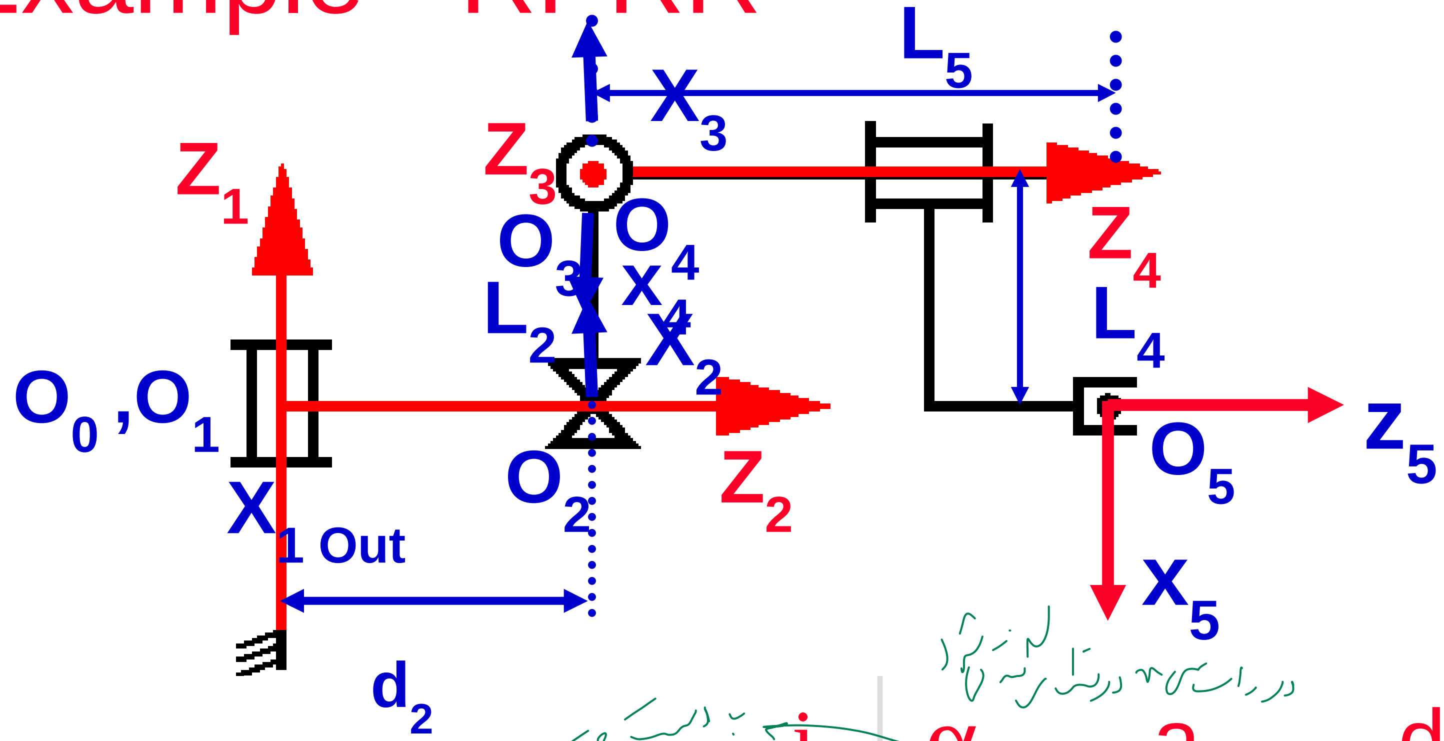
	a_i	α_i	d_i	θ_i
1	l_2	0	l_1	q_1
2	l_3	π	0	q_2
3	0	0	l_4	q_3
4	0	0	q_4	0



Example - RPRR



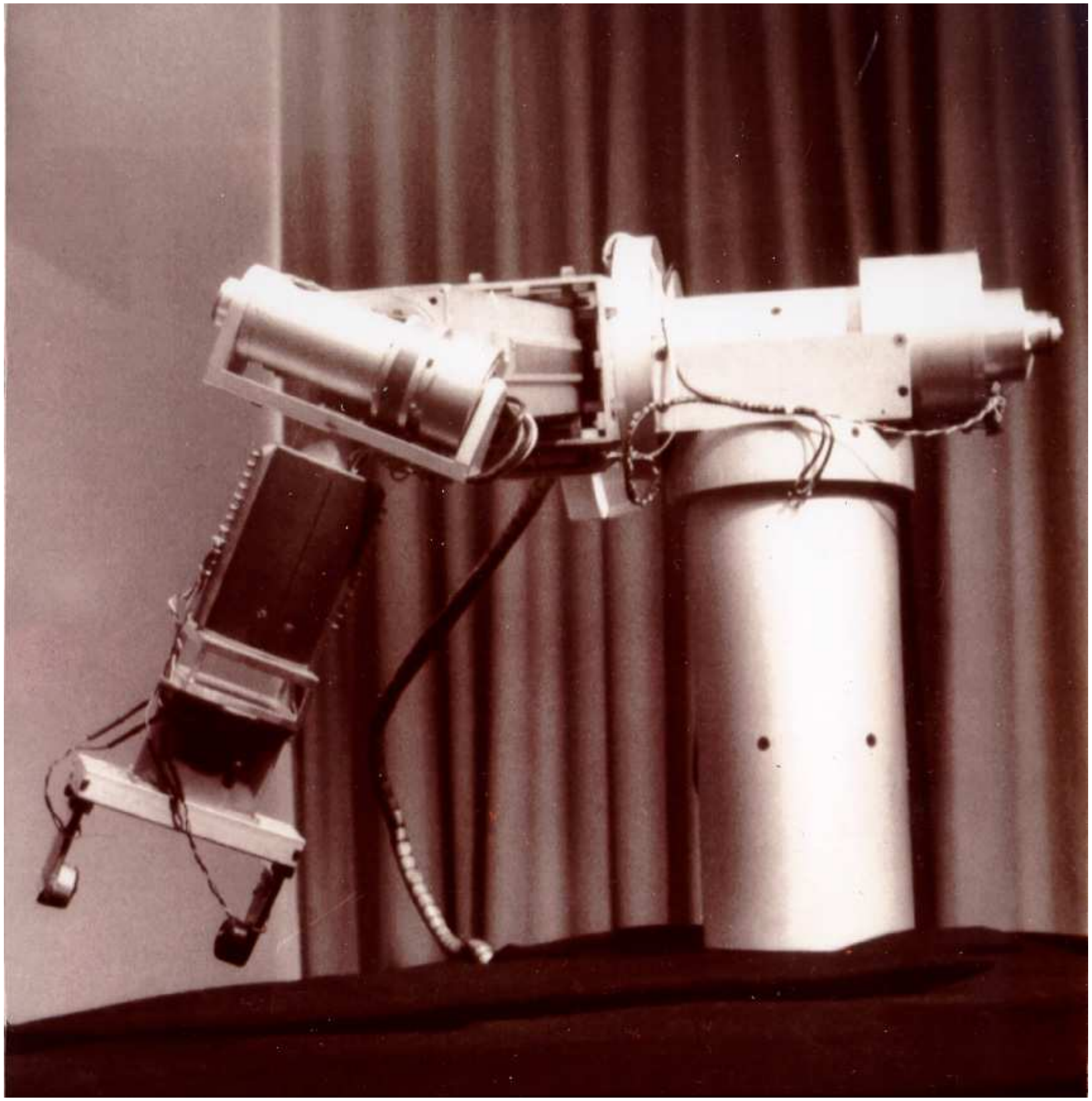
Example - RPRR



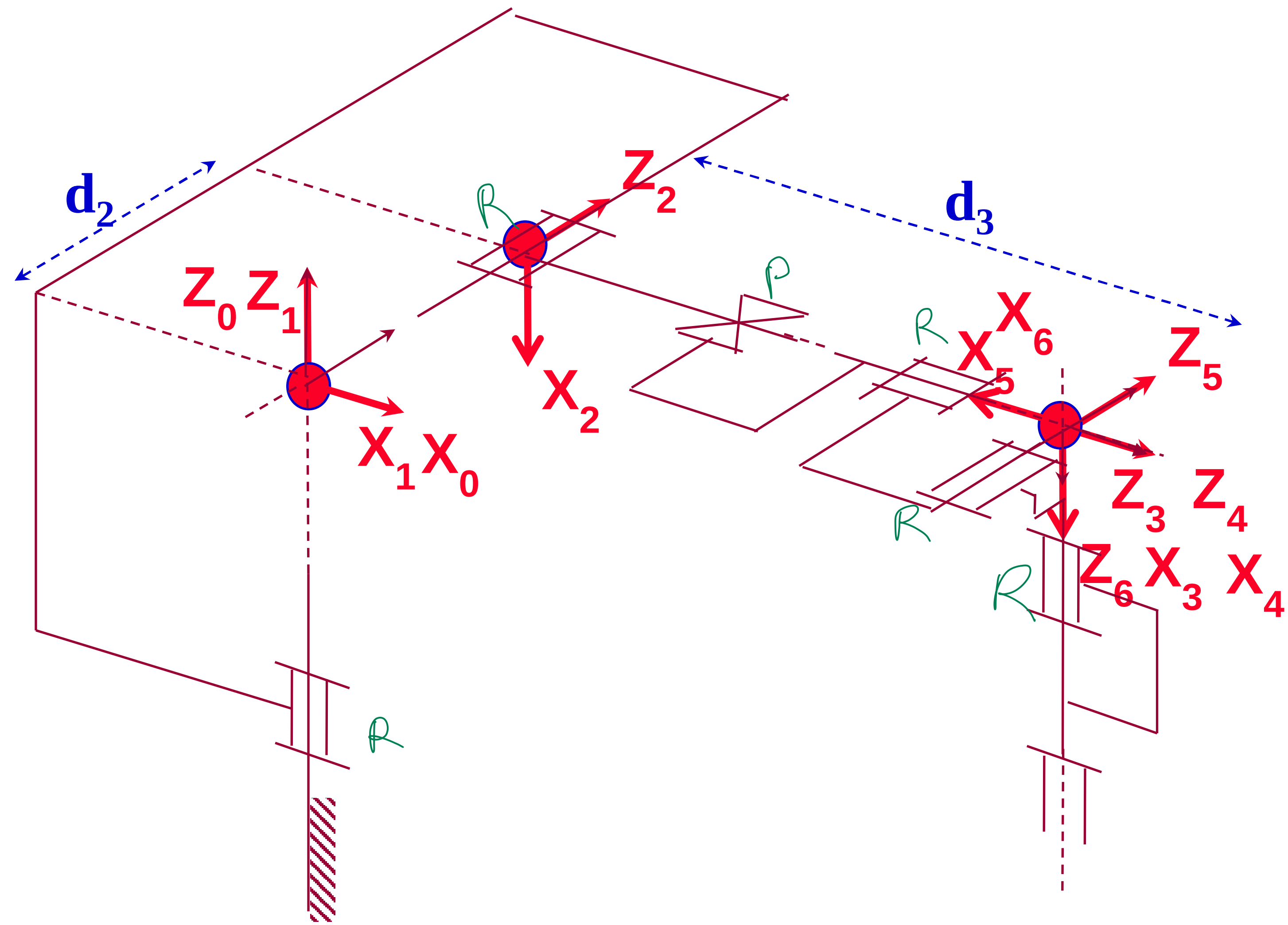
i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90	0	d_2	-90
3	-90	L_2	0	θ_3
4	90	0	L_5	θ_4
5	0	L_4	0	0

a_i : distance (z_i , z_{i+1}) along x_i
α_i : angle (z_i , z_{i+1}) about x_i
d_i : distance (x_{i-1} , x_i) along z_i
θ_i : angle (x_{i-1} , x_i) about z_i

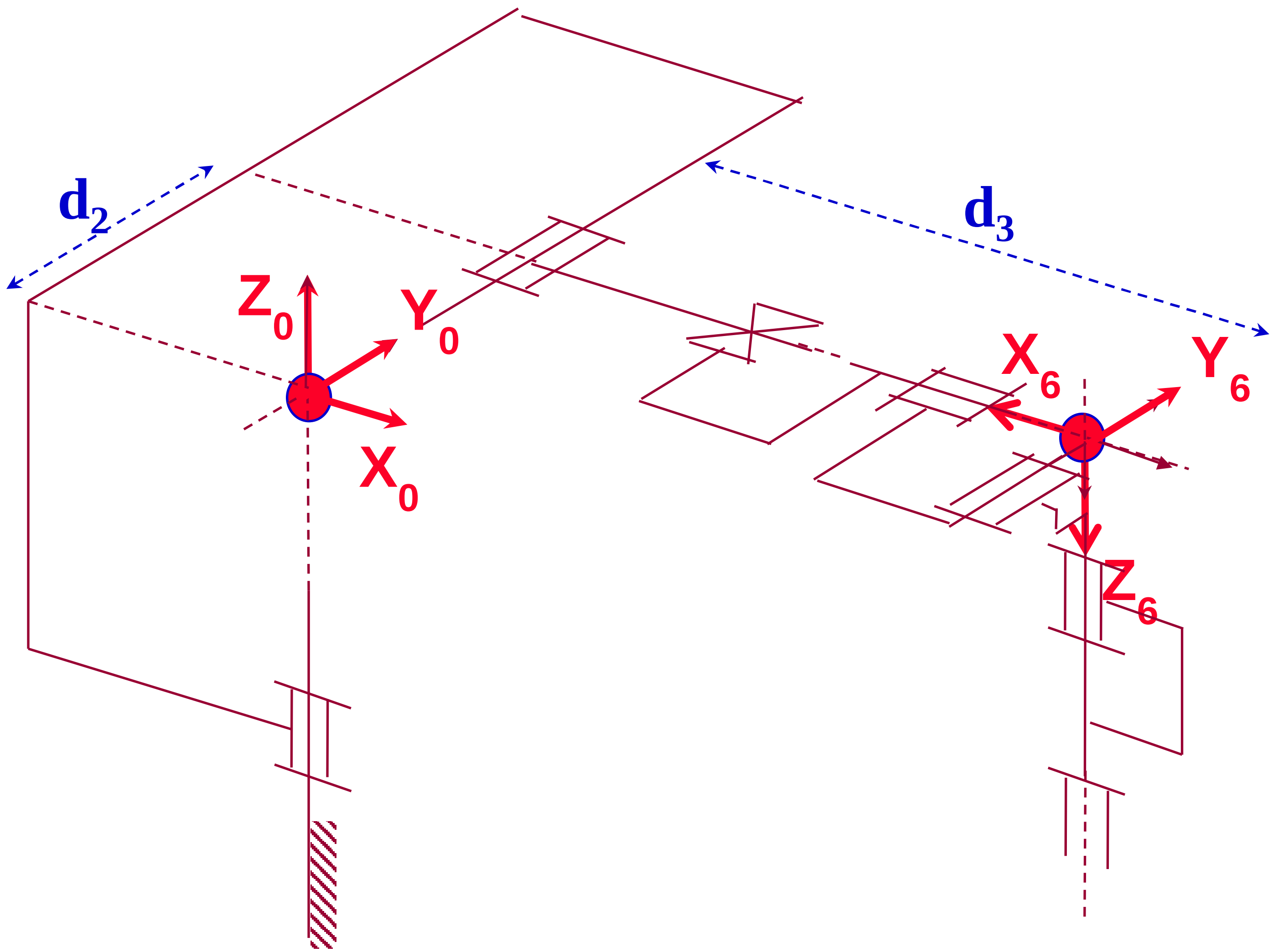
Stanford Scheinman Arm

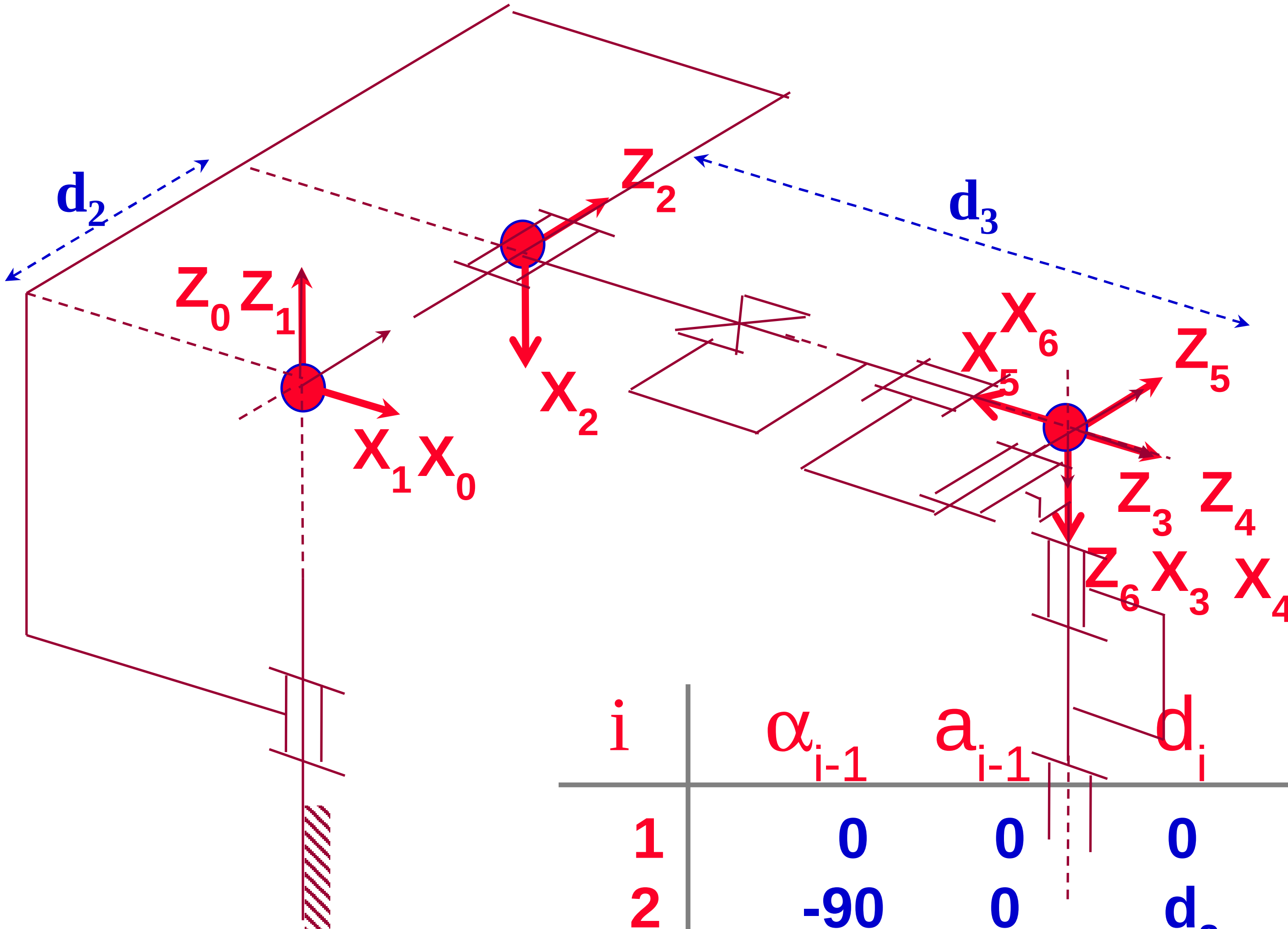


Stanford Scheinman Arm



Stanford Scheinman Arm





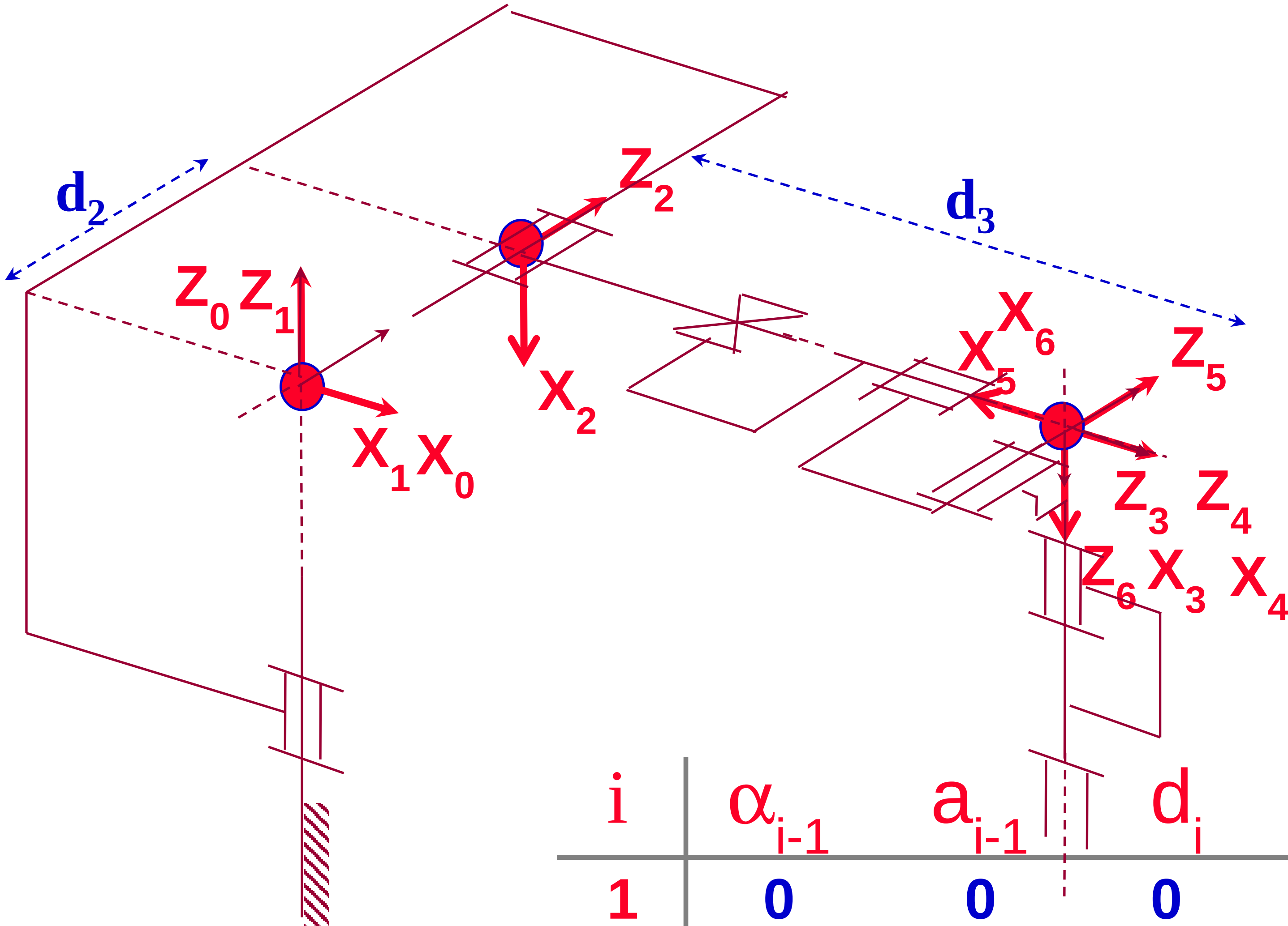
i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90	0	d_2	θ_2
3	90	0	d_3	0
4	0	0	0	θ_4
5	-90	0	0	θ_5
6	90	0	0	θ_6

a_i : distance (z_i , z_{i+1}) along x_i

α_i : angle (z_i , z_{i+1}) about x_i

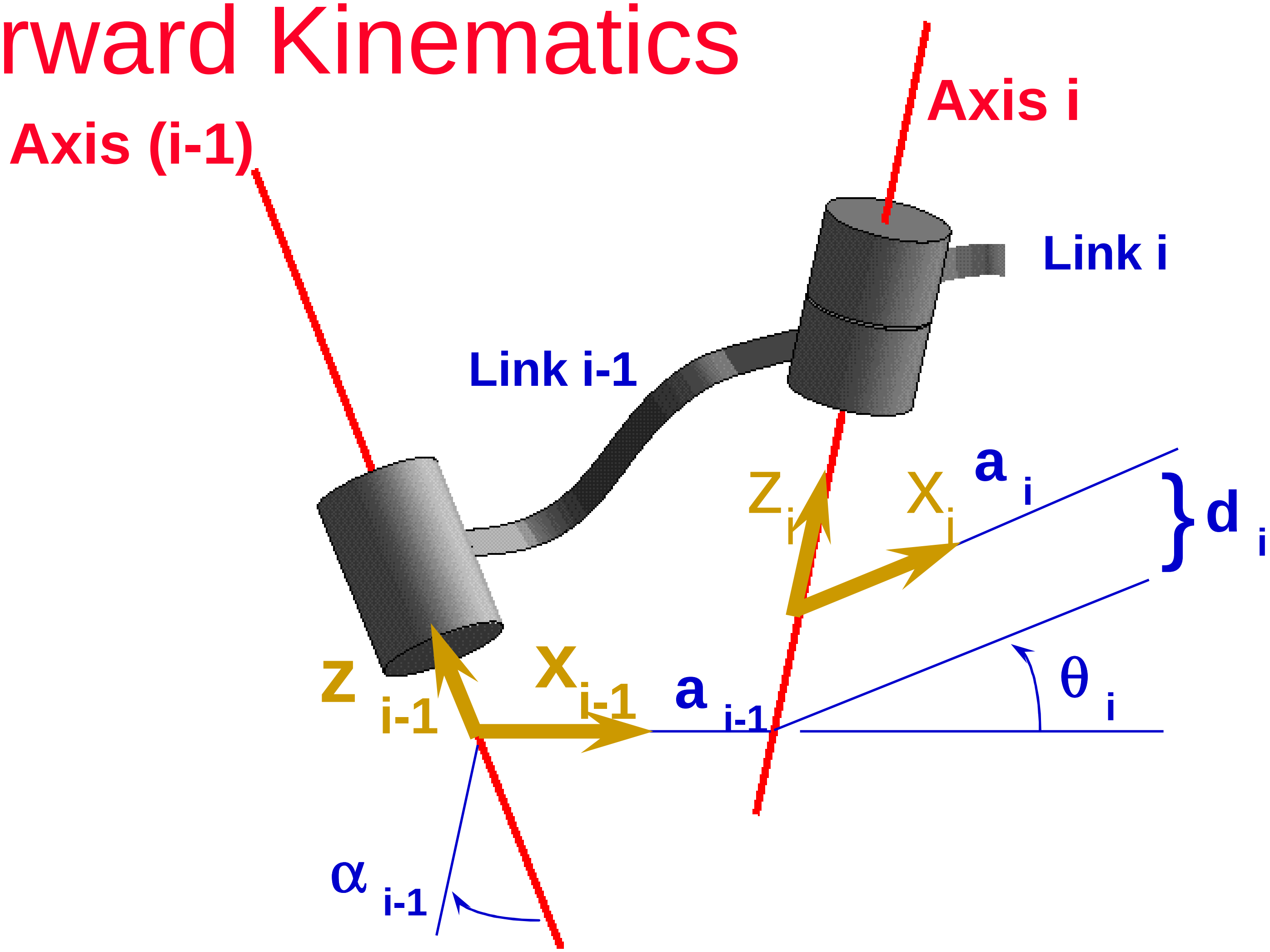
d_i : distance (x_{i-1} , x_i) along z_i

θ_i : angle (x_{i-1} , x_i) about z_i



i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90	0	d_2	θ_2
3	90	0	d_3	0
4	0	0	0	θ_4
5	-90	0	0	θ_5
6	90	0	0	θ_6

Forward Kinematics



$$
{}^{i-1}_{\;\;i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i\, c\alpha_{i-1} & c\theta_i\, c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}\, d_i \\ s\theta_i\, s\alpha_{i-1} & c\theta_i\, s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}\, d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Stanford Scheinman Arm

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90	0	d_2	θ_2
3	90	0	d_3	0
4	0	0	0	θ_4
5	-90	0	0	θ_5
6	90	0	0	θ_6

$$\begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i\, c\alpha_{i-1} & c\theta_i\, c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}\, d_i \\ s\theta_i\, s\alpha_{i-1} & c\theta_i\, s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}\, d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0_1T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1_2T = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ -s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

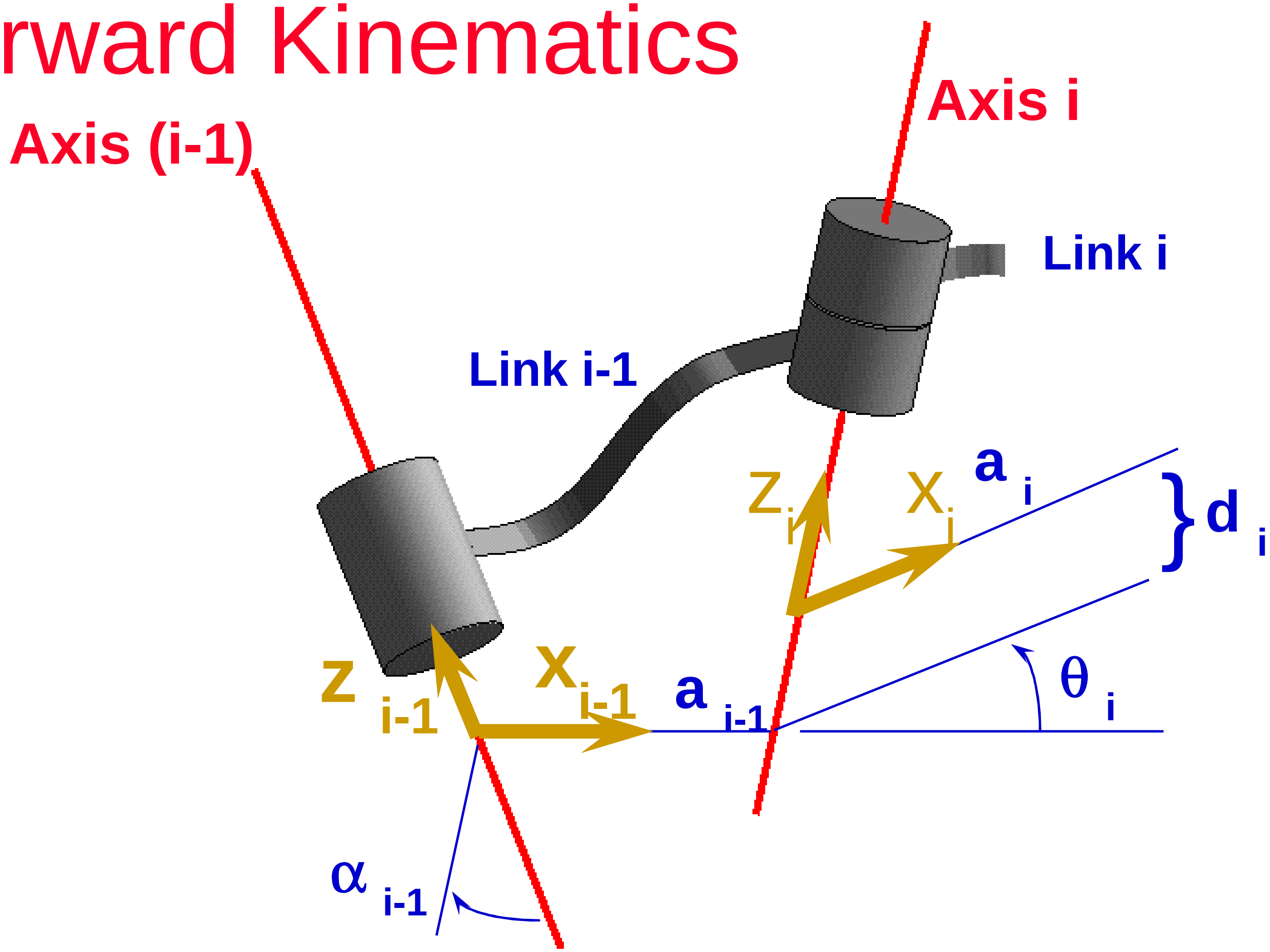
$${}^2_3T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -d_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$
{}^{3}_{4}T = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{4}_{5}T = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_5 & -c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{5}_{6}T = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Forward Kinematics



Forward Kinematics: $\,^{0}_{N}T = \,^{0}_{1}T \;\; ^{1}_{2}T \; \ldots \; ^{N-1}_{N}T$

$$
{}^{0}_{1}T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{0}_{2}T = \begin{bmatrix} c_1 c_2 & -c_1 s_2 & -s_1 & -s_1 d_2 \\ s_1 c_2 & -s_1 s_2 & c_1 & c_1 d_2 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{0}_{3}T = \begin{bmatrix} c_1 c_2 & -s_1 & c_1 s_2 & c_1 d_3 s_2 - s_1 d_2 \\ s_1 c_2 & c_1 & s_1 s_2 & s_1 d_3 s_2 + c_1 d_2 \\ -s_2 & 0 & c_2 & d_3 c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

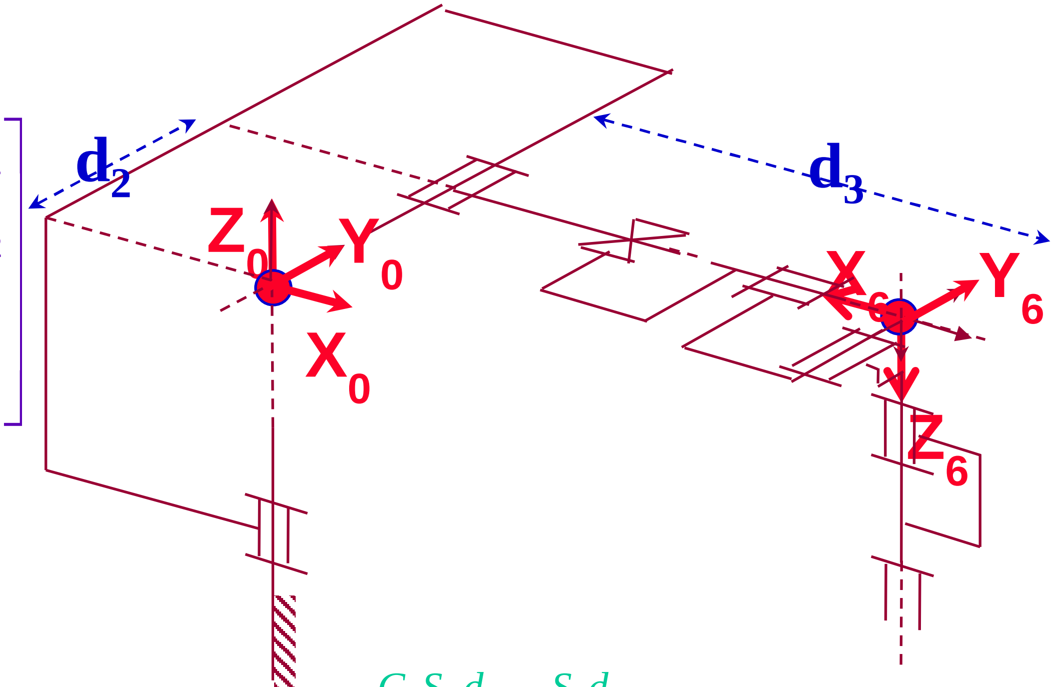
$$
{}^{0}_{4}T = \begin{bmatrix} c_1c_2c_4 - s_1s_4 & -c_1c_2s_4 - s_1c_4 & c_1s_2 & c_1d_3s_2 - s_1d_2 \\ s_1c_2c_4 + c_1s_4 & -s_1c_2s_4 + c_1c_4 & s_1s_2 & s_1d_3s_2 + c_1d_2 \\ -s_2c_4 & s_2s_4 & c_2 & d_3c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{0}_{5}T = \begin{bmatrix} X & X & -c_1c_2s_4 - s_1c_4 & c_1d_3s_2 - s_1d_2 \\ X & X & -s_1c_2s_4 + c_1c_4 & s_1d_3s_2 + c_1d_2 \\ X & X & s_2s_4 & d_3c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

هرچه جلوتر بریم پیچیده
[illegible]

$$
{}^{0}_{6}T = \begin{bmatrix} X & X & c_1c_2c_4s_5 - s_1s_4s_5 + c_1s_2s_5 & c_1d_3s_2 - s_1d_2 \\ X & X & s_1c_2c_4s_5 + c_1s_4s_5 + s_1s_2c_5 & s_1d_3s_2 + c_1d_2 \\ X & X & -s_2c_4s_5 + c_5c_2 & d_3c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

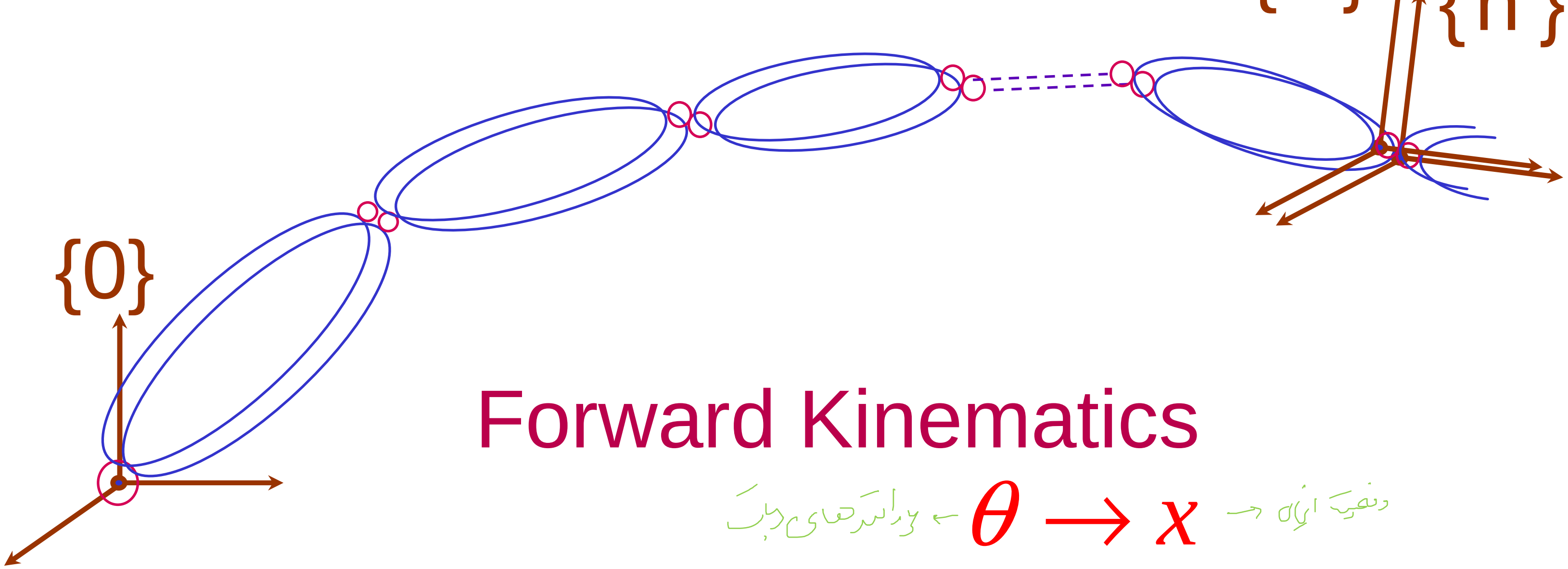
$$
{}^{0}_{6}T=\begin{bmatrix} X & X & c_1c_2c_4s_5 - s_1s_4s_5 + c_1s_2s_5 & c_1d_3s_2 - s_1d_2 \\ X & X & s_1c_2c_4s_5 + c_1s_4s_5 + s_1s_2c_5 & s_1d_3s_2 + c_1d_2 \\ X & X & -s_2c_4s_5 + c_5c_2 & d_3c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$



$$
x = \begin{pmatrix} x_P \\ r_1 \\ r_2 \\ r_3 \end{pmatrix} =
\begin{array}{c} p \\ \\ \\ x \\ \\ \\ y \\ \\ \\ z \\ \\ \\ \end{array}
\left|\begin{array}{c}
C_1S_2d_3 - S_1d_2 \\
S_1S_2d_3 + C_1d_2 \\
C_2d_3 \\
C_1[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] - S_1(S_4C_5C_6 + C_4S_6) \\
S_1[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] + C_1(S_4C_5C_6 + C_4S_6) \\
-S_2(C_4C_5C_6 - S_4S_6) - C_2S_5C_6 \\
C_1[-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] - S_1(-S_4C_5S_6 + C_4C_6) \\
S_1[-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] + C_1(-S_4C_5S_6 + C_4C_6) \\
S_2(C_4C_5S_6 + S_4C_6) + C_2S_5S_6 \\
C_1(C_2C_4S_5 + S_2C_5) - S_1S_4S_5 \\
S_1(C_2C_4S_5 + S_2C_5) + C_1S_4S_5 \\
-S_2C_4S_5 + C_2C_5
\end{array}\right|
$$

Instantaneous Kinematics

Differential Motion



Forward Kinematics

$$\theta \rightarrow x$$

Instantaneous Kinematics

$$\theta + \delta\theta \rightarrow x + \delta x$$

Relationship: $\boxed{\delta\theta \leftrightarrow \delta x}$

$$\boxed{\dot{\theta} \leftrightarrow \dot{x}}$$

Linear Velocity

Angular Velocity

Jacobian ← ماتریسی که تبدیل‌ها را انجام می‌کند

- Differential Motion
- Linear & Angular Motion
- Velocity Propagation
- Explicit Form
- Static Forces

Joint Coordinates

$$coordinate - i : \begin{cases} \theta_{\mathrm{i}} & \text{revolute} \\ \mathrm{d}_{\mathrm{i}} & \text{prismatic} \end{cases}$$

مستقل از نوع مفصل

مفصل i ام

Joint coordinate-i: $\boxed{q_i = \bar{\varepsilon}_i \theta_i + \varepsilon_i d_i}$

with $$\varepsilon_i = \begin{cases} 0 & \text{revolute} \\ 1 & \text{prismatic} \end{cases}$$

and $$\bar{\varepsilon}_i = 1 - \varepsilon_i$$

Joint Coordinate Vector: $\boxed{q = (q_1 q_2 \ldots . q_n)^T}$

Jacobians: Direct Differentiation

Position

تابعی بر حسب مفاصل

$$x = f(q); \quad \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} f_1(q) \\ f_2(q) \\ \vdots \\ f_m(q) \end{pmatrix}$$

برحسب مدل ناحیه ← تعداد پارامترهای خروجی

$$\delta x_1 = \frac{\partial f_1}{\partial q_1}\delta q_1 + \cdots + \frac{\partial f_1}{\partial q_n}\delta q_n$$

تعداد درجات آزادی

redundancy

$$\vdots$$

$$\delta x_m = \frac{\partial f_m}{\partial q_1}\delta q_1 + \cdots + \frac{\partial f_m}{\partial q_n}\delta q_n$$

$$\delta x = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \cdots & \frac{\partial f_m}{\partial q_n} \end{bmatrix} . \delta q$$

$$\boxed{\delta x_{(m\times 1)} = J_{(m\times n)}(q)\,\delta q_{(n\times 1)}}$$

Jacobian

$$\delta x_{(m\times 1)} = J_{(m\times n)}(q)\,\delta q_{(n\times 1)}$$

تابعی از q

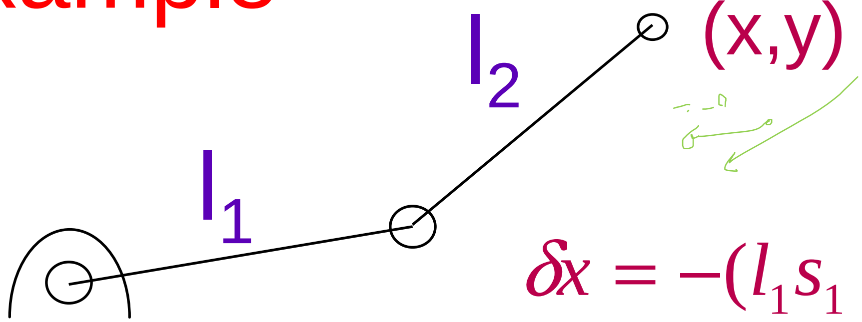
$$\dot{x}_{(m\times 1)} = J_{(m\times n)}(q)\;\dot{q}_{(n\times 1)}$$

خطی نیست و در هر لحظه مقدار خاصی دارد.　　　سرعت مفاصل ربات را

where

$$J_{ij}(q) = \frac{\partial}{\partial q_j} f_i(q)$$

Example



forward حل

$$x = l_1 c_1 + l_2 c_{12}$$

$$y = l_1 s_1 + l_2 s_{12}$$

$$\delta x = -(l_1 s_1 + l_2 s_{12})\delta\theta_1 - l_2 s_{12}\delta\theta_2$$

$$\delta y = (l_1 c_1 + l_2 c_{12})\delta\theta_1 + l_2 c_{12}\delta\theta_2$$

ماتریس ژاکوبین ماتریس تغییرات است رفتار زوایا

همین تغییر را تعریف می‌کند

$$\delta X = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix} = \begin{bmatrix} -y & -l_2 s_{12} \\ x & l_2 c_{12} \end{bmatrix} \begin{pmatrix} \delta\theta_1 \\ \delta\theta_2 \end{pmatrix}$$

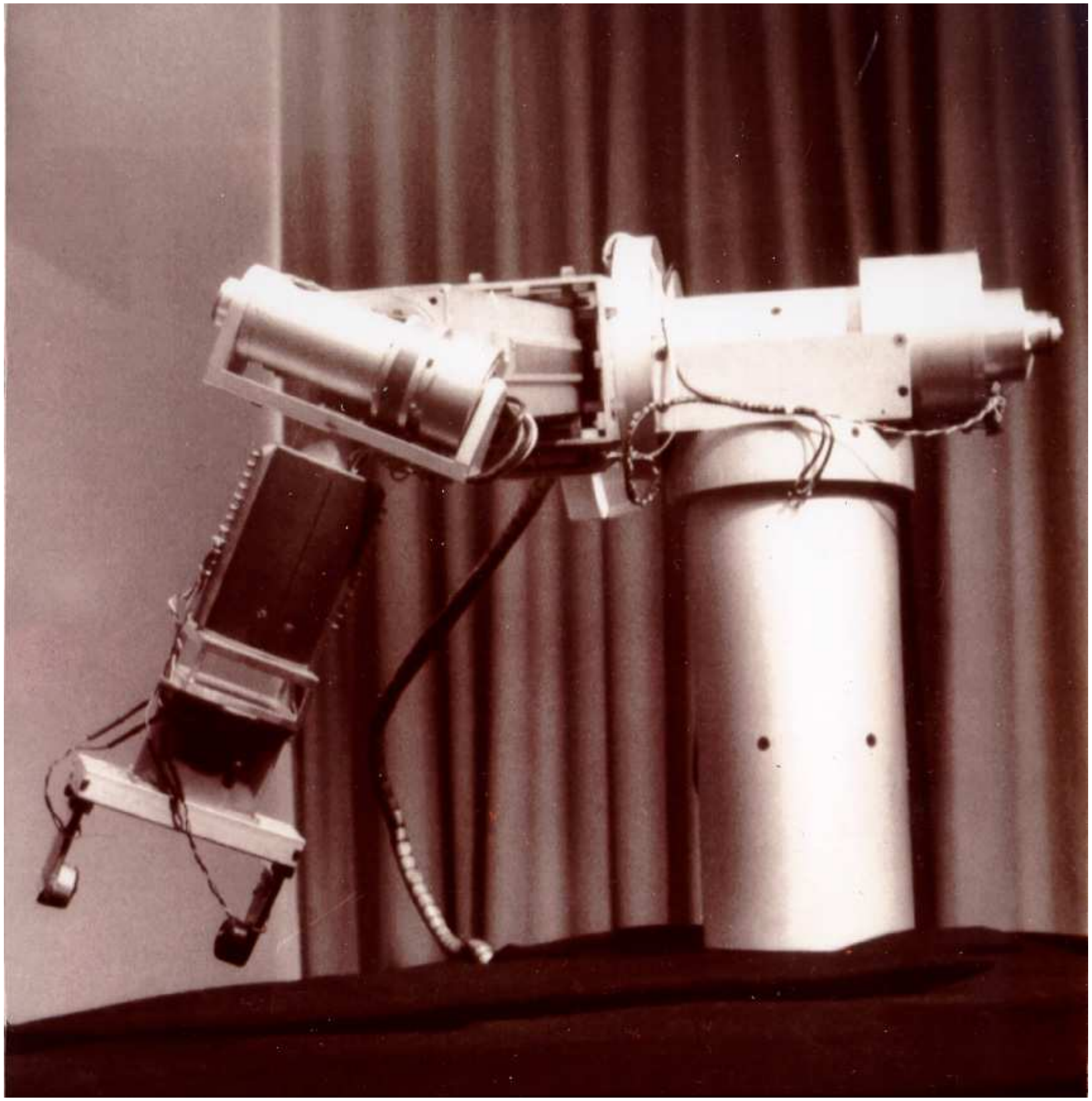
دیفرانسیل به دلیل یک بار تغییر کوچک نقطه اتفاق می‌افتد

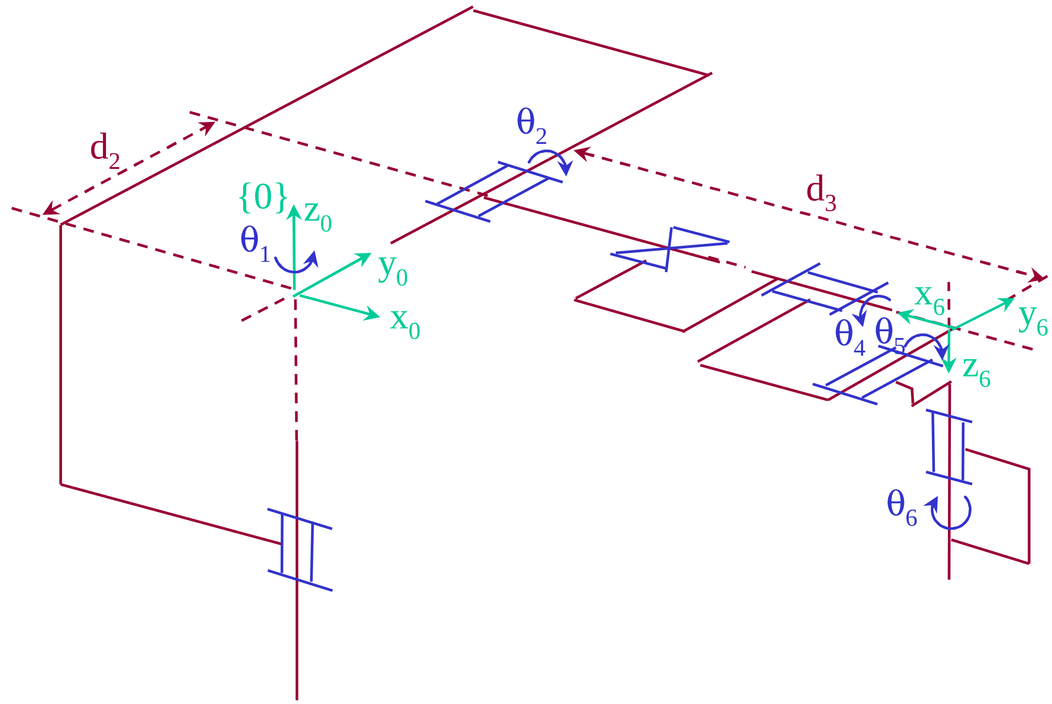
$$\boxed{\delta x = J(\theta)\delta\theta}$$

$$\dot{x} = J(\theta)\dot{\theta}$$

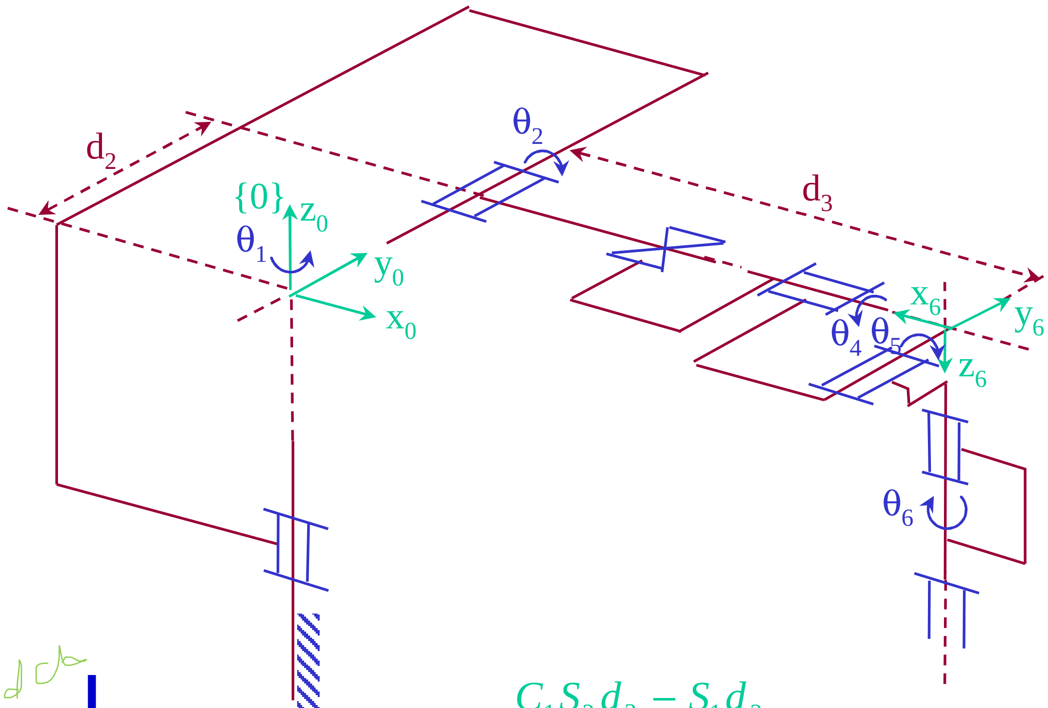
$$J \equiv \begin{pmatrix} \dfrac{\partial x}{\partial\theta_1} & \dfrac{\partial x}{\partial\theta_2} \\ \dfrac{\partial y}{\partial\theta_1} & \dfrac{\partial y}{\partial\theta_2} \end{pmatrix} = \begin{bmatrix} -y & -l_2 s_{12} \\ x & l_2 c_{12} \end{bmatrix}$$

Stanford Scheinman Arm





i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90	0	d_2	θ_2
3	90	0	d_3	0
4	0	0	0	θ_4
5	-90	0	0	θ_5
6	90	0	0	θ_6



حل forward به صورت مختصات از
آن ET مسیر برمی‌داریم

$$
x = \begin{pmatrix} x_P \\ r_1 \\ r_2 \\ r_3 \\ \\ \end{pmatrix} = \left[\begin{array}{c}
C_1 S_2 d_3 - S_1 d_2 \\
S_1 S_2 d_3 + C_1 d_2 \\
C_2 d_3 \\
C_1[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] - S_1(S_4C_5C_6 + C_4S_6) \\
S_1[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] + C_1(S_4C_5C_6 + C_4S_6) \\
-S_2(C_4C_5C_6 - S_4S_6) - C_2S_5C_6 \\
C_1[-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] - S_1(-S_4C_5S_6 + C_4C_6) \\
S_1[-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] + C_1(-S_4C_5S_6 + C_4C_6) \\
S_2(C_4C_5S_6 + S_4C_6) + C_2S_5S_6 \\
C_1(C_2C_4S_5 + S_2C_5) - S_1S_4S_5 \\
S_1(C_2C_4S_5 + S_2C_5) + C_1S_4S_5 \\
-S_2C_4S_5 + C_2C_5
\end{array} \right]
$$

Stanford Scheinman Arm

Position

$$x_p = \begin{bmatrix} c_1 s_2 d_3 - s_1 d_2 \\ s_1 s_2 d_3 + c_1 d_2 \\ c_2 d_3 \end{bmatrix}$$

مشتق → $\dot{q}_1$ → درمرحله بر حسب متغیر → y

$$\dot{x}_p = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{bmatrix} -y & c_1 c_2 d_3 & c_1 s_2 & 0 & 0 & 0 \\ x & s_1 c_2 d_3 & s_1 s_2 & 0 & 0 & 0 \\ 0 & -s_2 d_3 & c_2 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix}$$

مشتق ها صفر هستند چون revolute است و متغیر نگیری عوضش نمی‌کرد.

$$\dot{x}_{p(3\times1)} = J_{x_p(3\times6)}(q)\,\dot{q}_{(6x1)}$$

Linear Velocity **V**

Orientation: Direction Cosines

$$x_R = \begin{bmatrix} r_1(q) \\ r_2(q) \\ r_3(q) \end{bmatrix}$$

تابع برداری q

$$\dot{x}_R = J_{X_R}(q)\dot{q}$$

$$\dot{x}_R = \begin{pmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \dot{r}_3 \end{pmatrix}_{(9x1)} = \begin{pmatrix} \dfrac{\partial r_1}{\partial q_1} & \cdots & \dfrac{\partial r_1}{\partial q_6} \\ \dfrac{\partial r_2}{\partial q_1} & \cdots & \dfrac{\partial r_2}{\partial q_6} \\ \dfrac{\partial r_3}{\partial q_1} & \cdots & \dfrac{\partial r_3}{\partial q_6} \end{pmatrix}_{(9x6)} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_6 \end{pmatrix}_{(6x1)}$$

$$
X_R = \begin{vmatrix}
C_1[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] - S_1(S_4C_5C_6 + C_4S_6) \\
S_1[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] + C_1(S_4C_5C_6 + C_4S_6) \\
-S_2(C_4C_5C_6 - S_4S_6) - C_2S_5C_6 \\
C_1[-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] - S_1(-S_4C_5S_6 + C_4C_6) \\
S_1[-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] + C_1(-S_4C_5S_6 + C_4C_6) \\
S_2(C_4C_5S_6 + S_4C_6) + C_2S_5S_6 \\
C_1(C_2C_4S_5 + S_2C_5) - S_1S_4S_5 \\
S_1(C_2C_4S_5 + S_2C_5) + C_1S_4S_5 \\
-S_2C_4S_5 + C_2C_5
\end{vmatrix}
$$

$$
\dot{x}_R = \begin{pmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \dot{r}_3 \end{pmatrix}_{(9x1)} = \begin{pmatrix}
\frac{\partial r_1}{\partial q_1} & \cdots & \frac{\partial r_1}{\partial q_6} \\
\frac{\partial r_2}{\partial q_1} & \cdots & \frac{\partial r_2}{\partial q_6} \\
\frac{\partial r_3}{\partial q_1} & \cdots & \frac{\partial r_3}{\partial q_6}
\end{pmatrix}_{(9x6)} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_6 \end{pmatrix}_{(6x1)}
$$

rotation Jaccobian ↩

Representations

براساس نیاز کاربرد است تعداد و جنس پارامترها متفاوت

position + orientation

pose — linear velocity

$$X = \begin{bmatrix} x_P \\ x_R \end{bmatrix}$$

rotation

angular velocity

- Cartesian — ۳ عدد از جنس طول
- Spherical → ۲ عدد از جنس طول / یک عدد از جنس زاویه
- Cylindrical
- ....

- Euler Angles → redundant نیست
- Direction Cosines → ۹ پارامتر (مشکل تعداد زیاد)
- Euler Parameters → ۴ پارامتر ([illegible] عدد موهومی)

[illegible] حل مشکل singularity [illegible]

Jacobian for X

$$\dot{x}_P = J_{X_P}(q)\dot{q}$$

$$\dot{x}_R = J_{X_R}(q)\dot{q}$$

$$\begin{pmatrix} \dot{x}_P \\ \dot{x}_R \end{pmatrix} = \begin{pmatrix} J_{X_P}(q) \\ J_{X_R}(q) \end{pmatrix}\dot{q}$$

3x6 9x6 → 12x6

Cartesian & Direction Cosines

$$\dot{X}_{(12x1)} = J_X(q)_{(12x6)}\,\dot{q}_{(6x1)}$$

The Jacobian is dependent on the representation

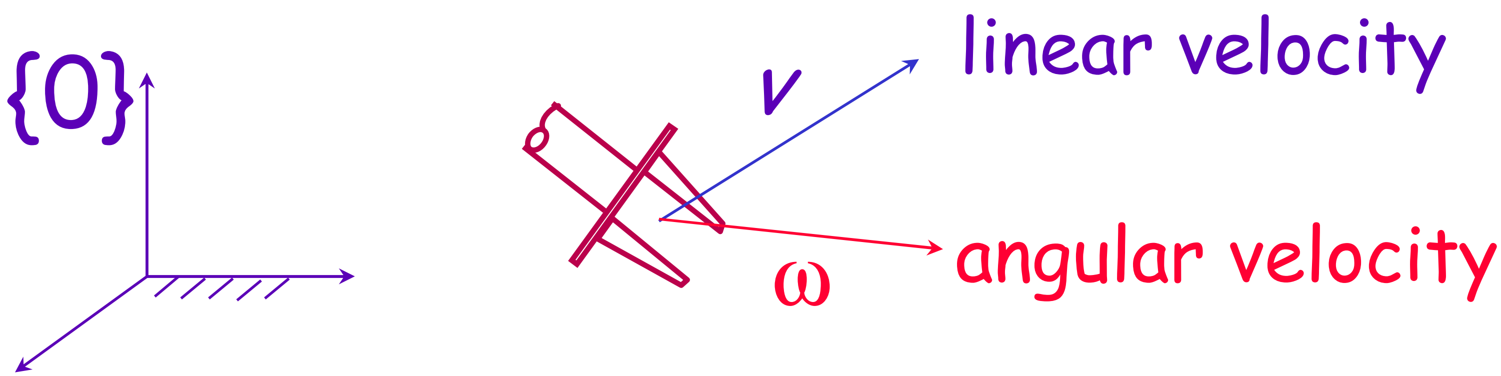
برای این عمل خوب نیست چون می‌خواهیم بازو را نسبت به موقعیت کینماتیک دیدگاه در حال حاضر بسته لذا صرفاً برای کلی در این حالت دو مبنا اطلاعاتی

که به جدا است که سینماتیک داریم بر ترکیب شده است و نیاز به جدا شده که سه داریم. ممکن است حرکت نسبت بازکی از اتاق باشد ۱ ۲ ۳ تا مشترک ۲ ۳ شده

بازو در حال رفتن به شده که حرکت را تا پس قابل را را نیست

Basic Jacobian

پس jacobian ی که بدست می‌آوریم در حقیقت با هم رفتن تعلق بازوی پایه و ابزار
به عنوان basic یا analytical شناخته می‌شود (geometric هم می‌گویند)



بردار استاندارد

$$\begin{pmatrix} v \\ \omega \end{pmatrix}_{(6x1)} = J_0(q)_{(6xn)}\,\dot{q}_{(nx1)}$$

تبدیل بازنمایی‌ها به بازنمایی استاندارد

$$\dot{x}_P = E_P(x_P)v$$

$$\dot{x}_R = E_R(x_R)\omega$$

Examples

Euler [illegible]

* $$x_R = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}; E_R(x_R) = \begin{pmatrix} -\dfrac{s\alpha.c\beta}{s\beta} & \dfrac{c\alpha.c\beta}{s\beta} & 1 \\ c\alpha & s\alpha & 0 \\ \dfrac{s\alpha}{s\beta} & -\dfrac{c\alpha}{s\beta} & 0 \end{pmatrix}$$

* $$x_P = \begin{pmatrix} x \\ y \\ z \end{pmatrix}; E_P(x_P) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Jacobian for X

Given a representation $x = \begin{bmatrix} x_P \\ x_R \end{bmatrix}$

$$\dot{x} = J_x(q)\,\dot{q}$$

$$J_x(q) = E(x)\,J_0(q)$$

Basic Jacobian $\begin{pmatrix} v \\ w \end{pmatrix} = J_0(q)\,\dot{q}$

Jacobian and Basic Jacobian

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = J_0(q)\dot{q}$$

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} J_v \\ J_\omega \end{pmatrix} \dot{q}$$

$$\begin{cases} v = J_v \dot{q} \\ \omega = J_\omega \dot{q} \end{cases}$$

$$\dot{x}_P = E_P . v \Rightarrow \dot{x}_P = (E_P . J_v)\dot{q}$$

$$\dot{x}_R = E_R . \omega \Rightarrow \dot{x}_R = (E_R . J_\omega)\dot{q}$$

$$\begin{cases} J_{X_P} = E_P . J_v \\ J_{X_R} = E_R . J_\omega \end{cases}$$

$$J = \begin{pmatrix} J_{XP} \\ \hline J_{XR} \end{pmatrix} = \left(\begin{array}{c|c} E_P & 0 \\ \hline 0 & E_R \end{array}\right) \begin{pmatrix} J_v \\ \hline J_w \end{pmatrix}$$

basic

۳ تا پارامتر

6x6

6x6

$$\underline{\underline{J(q) \qquad = \qquad E(X) \qquad J_0(q)}}$$

$$\underline{\underline{\begin{pmatrix} v \\ w \end{pmatrix} \qquad = \qquad J_0(q)\ \dot{q}}}$$

With Cartesian Coordinates

$E_P = I_3$; $J_{XP} = J_v$; and $E = \left(\begin{array}{c|c} I & 0 \\ \hline 0 & E_R \end{array}\right)$

Position Representations

Cartesian Coordinates (x, y, z)

$$E_P(X) = I_3$$

Cylindrical Coordinates (ρ, θ, z)

Using $(x \quad y \quad z)^T = (\rho\cos\theta \quad \rho\sin\theta \quad z)^T$

$$E_P(X) = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta/\rho & \cos\theta/\rho & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Spherical Coordinates (ρ, θ, ϕ)

مختصات کروی

Using

$$(x \quad y \quad z)^T = (\rho\cos\theta\sin\phi \quad \rho\sin\theta\sin\phi \quad \rho\cos\theta)^T$$

$$E_P(X) = \begin{pmatrix} \cos\theta\sin\phi & \sin\theta\sin\phi & \cos\phi \\ -\sin\theta/(\rho\sin\phi) & \cos\theta/(\rho\sin\phi) & 0 \\ \cos\theta\cos\phi/\rho & \sin\theta\cos\phi/\rho & -\sin\phi/\rho \end{pmatrix}$$

Euler Angles

$$x_R = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}; E_R(x_R) = \begin{pmatrix} -\dfrac{s\alpha.c\beta}{s\beta} & \dfrac{c\alpha.c\beta}{s\beta} & 1 \\ c\alpha & s\alpha & 0 \\ \dfrac{s\alpha}{s\beta} & -\dfrac{c\alpha}{s\beta} & 0 \end{pmatrix}$$

Singularity of the representation for $\beta = k\pi$

Jacobian for X

Given a representation $x = \begin{bmatrix} x_P \\ x_R \end{bmatrix}$

$$\dot{x} = J_x(q)\,\dot{q}$$

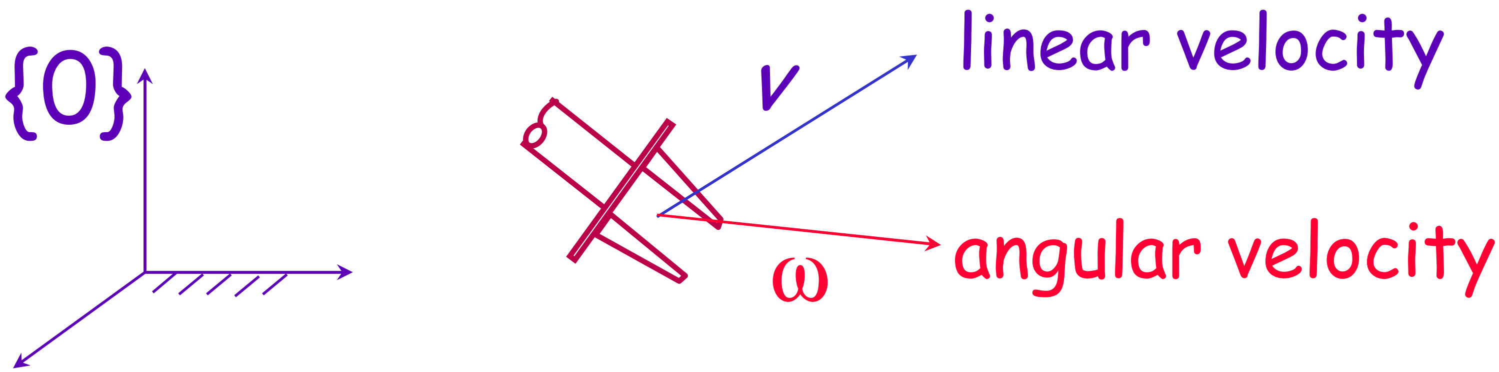
position

$$J_x(q) = E(x)\,J_0(q)$$

Basic Jacobian $\begin{pmatrix} v \\ w \end{pmatrix} = J_0(q)\,\dot{q}$

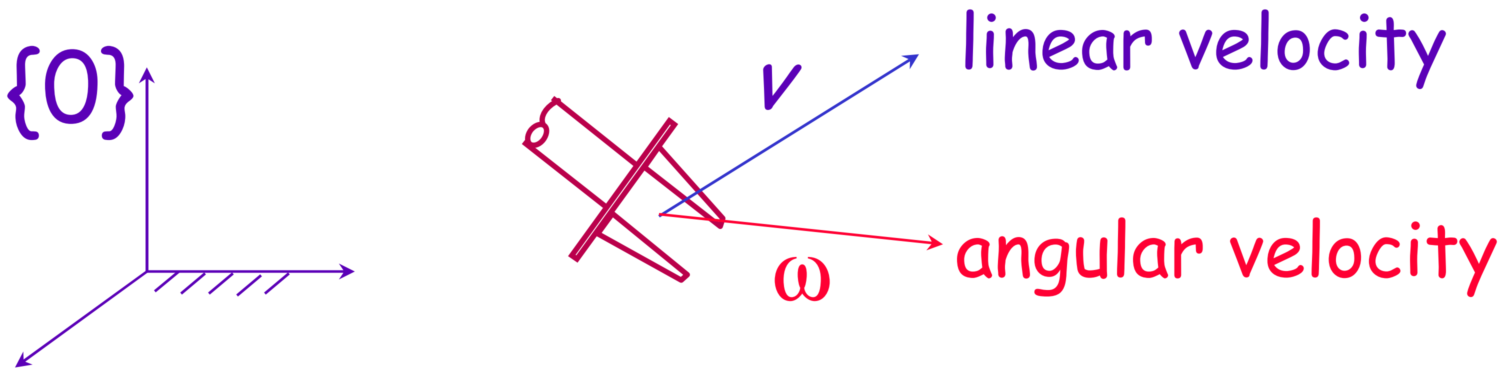
J_a

Jacobian

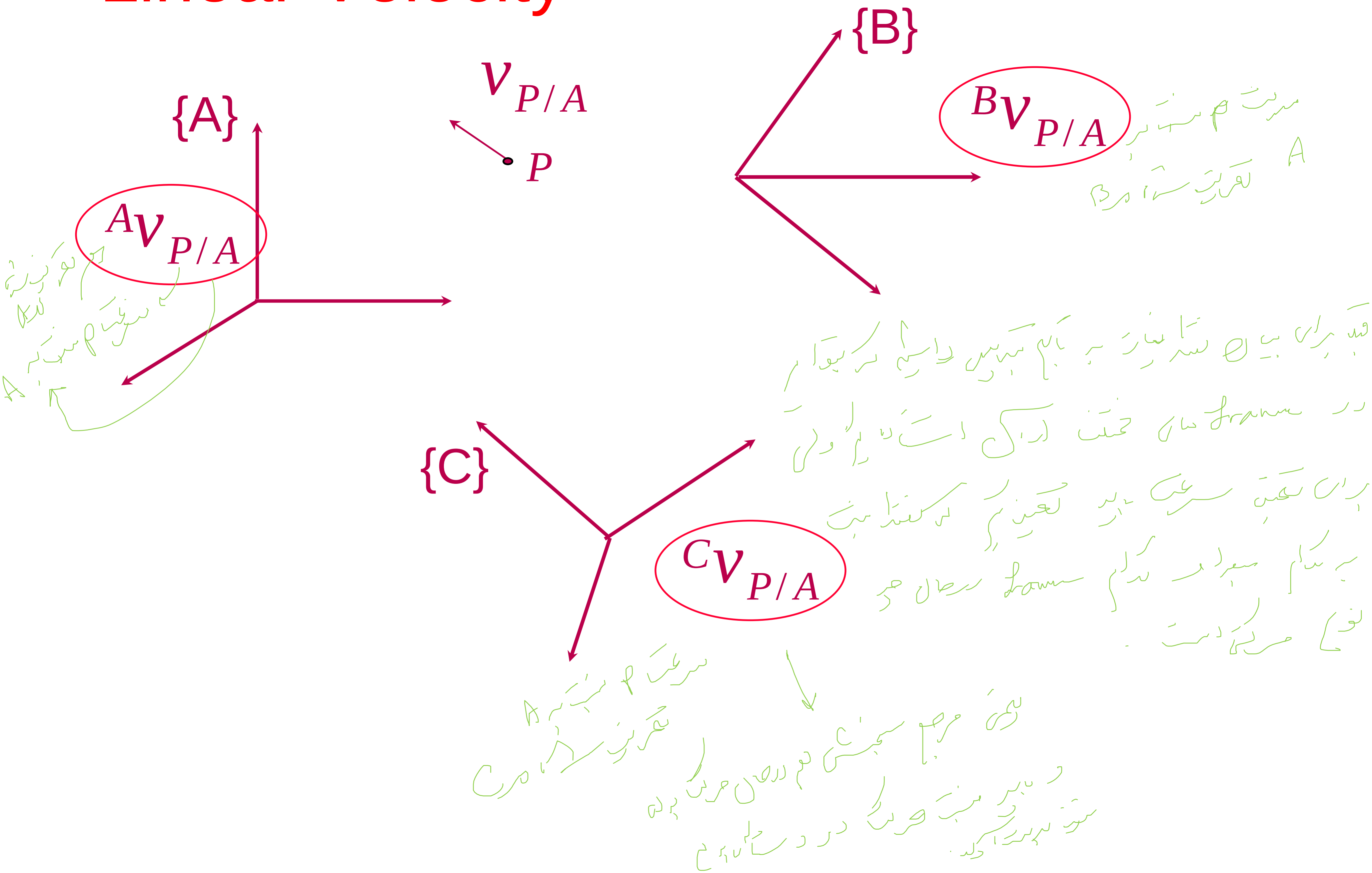


$$\begin{pmatrix} v \\ \omega \end{pmatrix}_{(6x1)} = J(q)_{(6xn)}\,\dot{q}_{(nx1)}$$

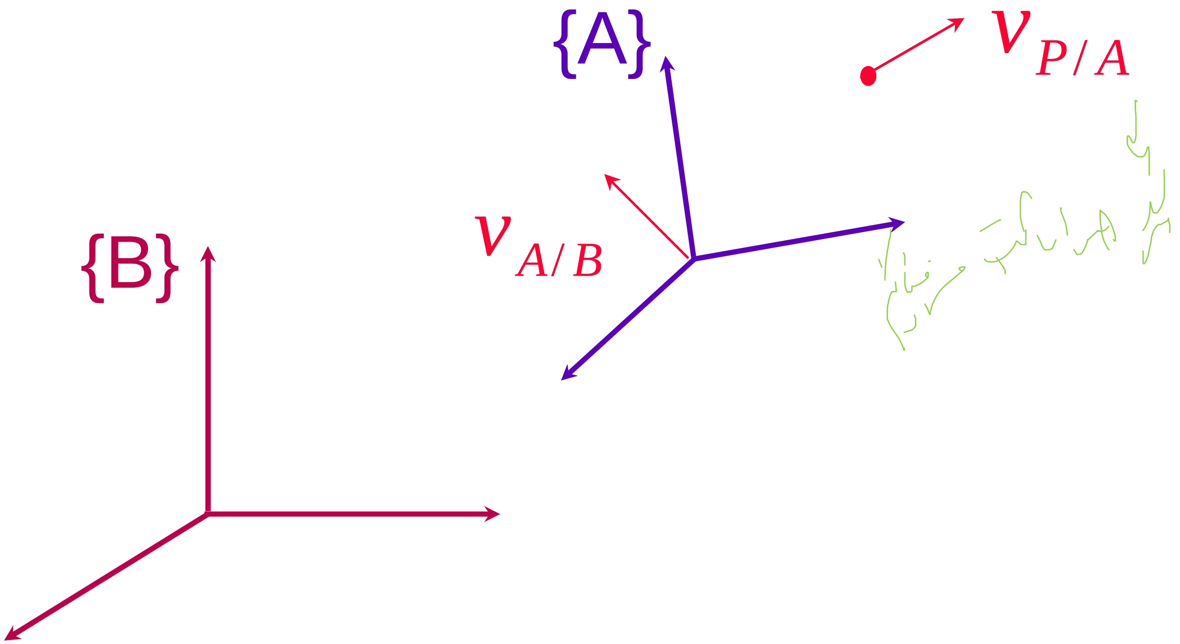
Linear & Angular Velocities



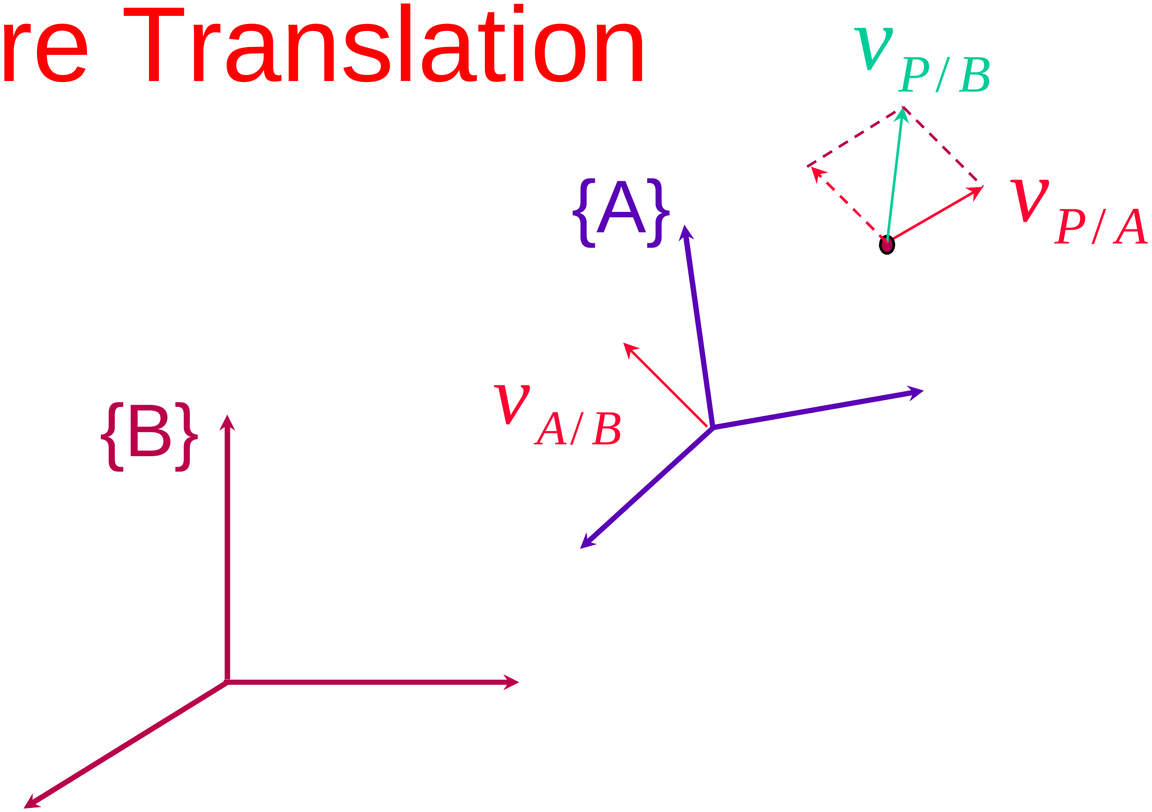
Linear Velocity



Pure Translation



Pure Translation

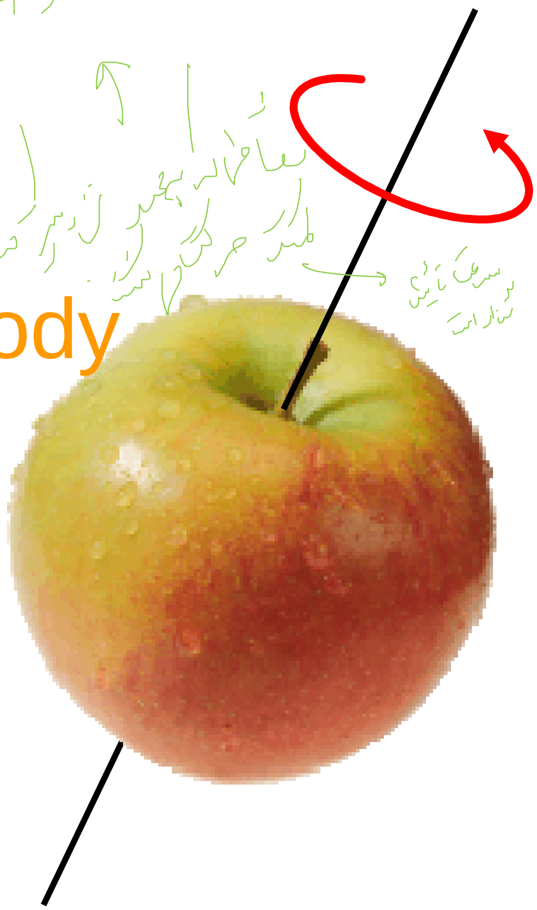


[illegible]

$$v_{P/B} = v_{A/B} + v_{P/A}$$

[illegible]

Rotational Motion



Axis of rotation

fixed points on the rigid body

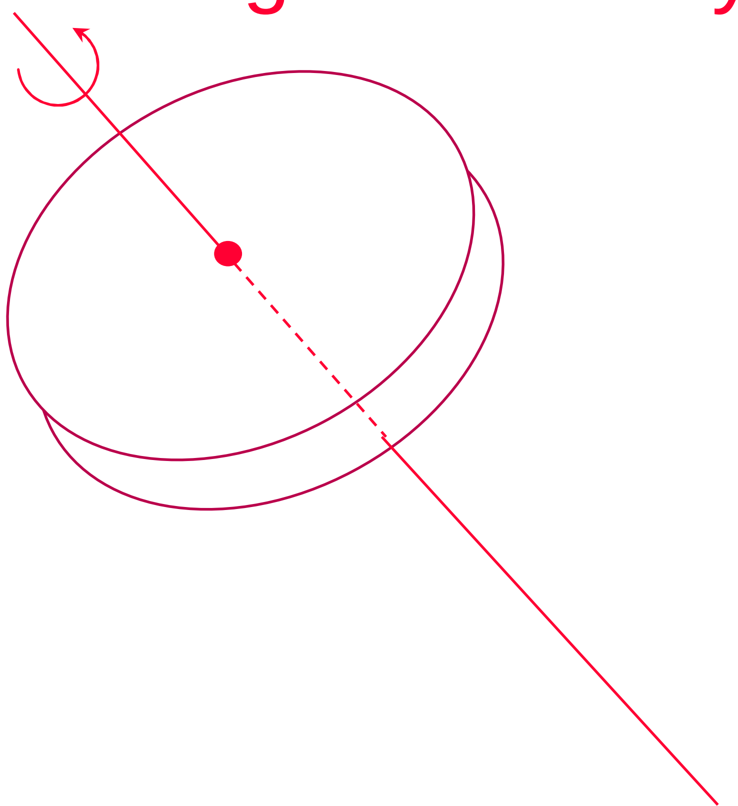
rigid body

Rotational Motion

Ω Angular Velocity

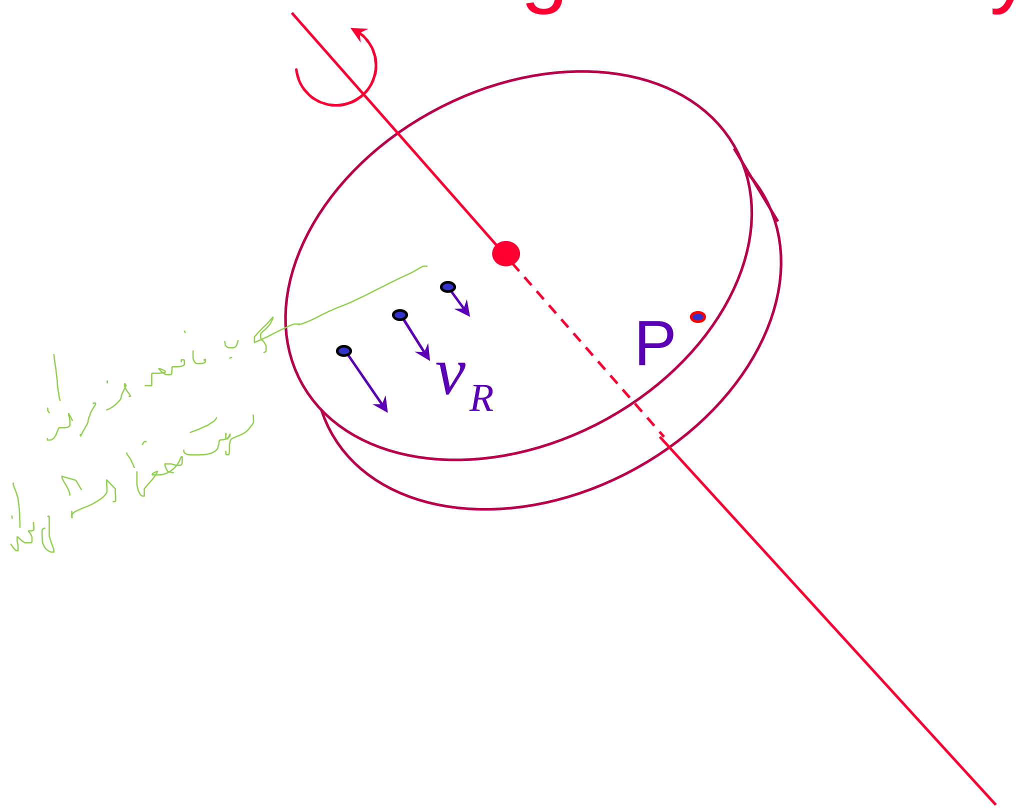
→ علامت رو مثبت کن در جهت کلاکوایز

منفی هم است .



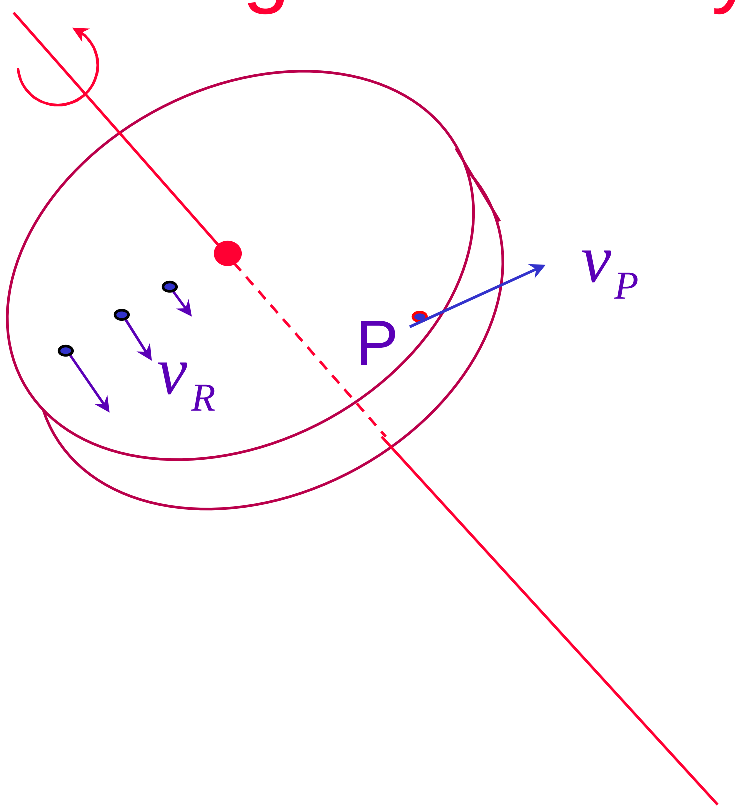
Rotational Motion

Ω Angular Velocity



Rotational Motion

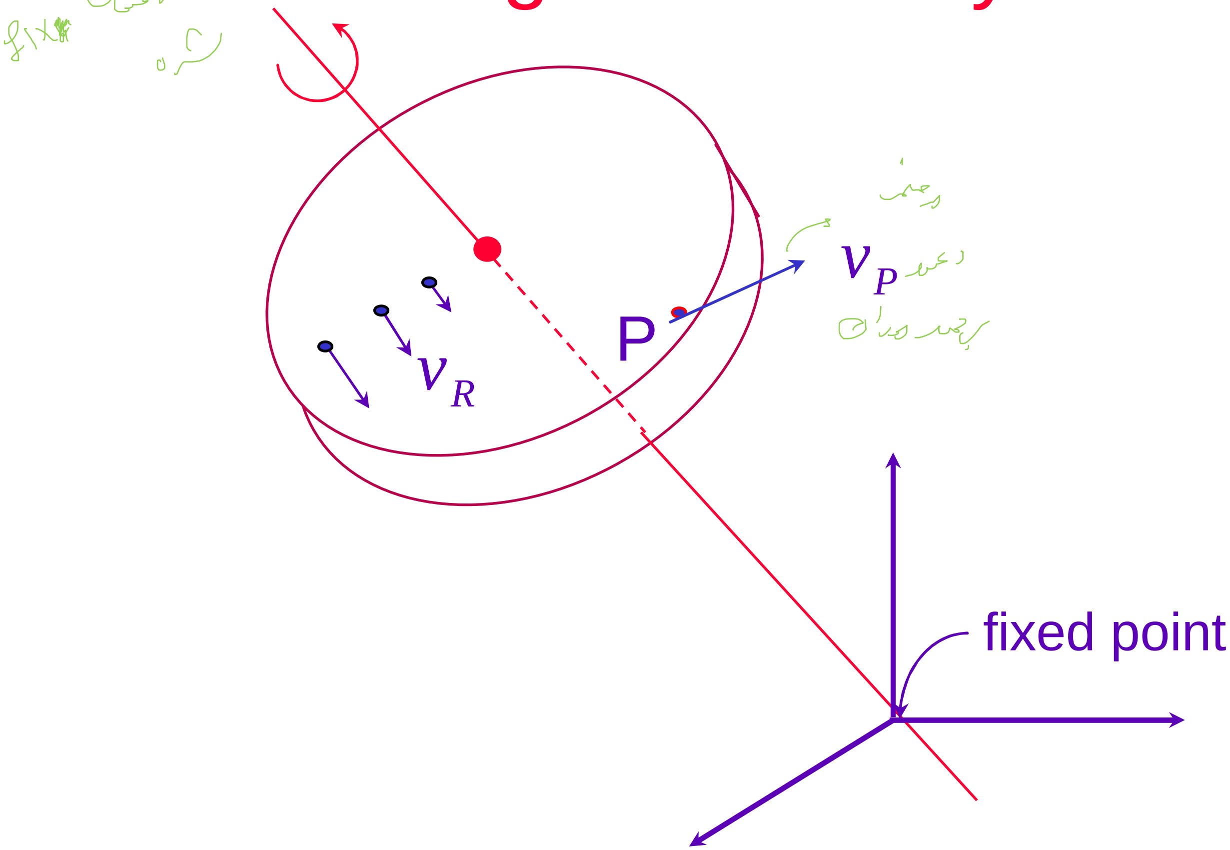
Ω Angular Velocity



$$v_P = ?$$

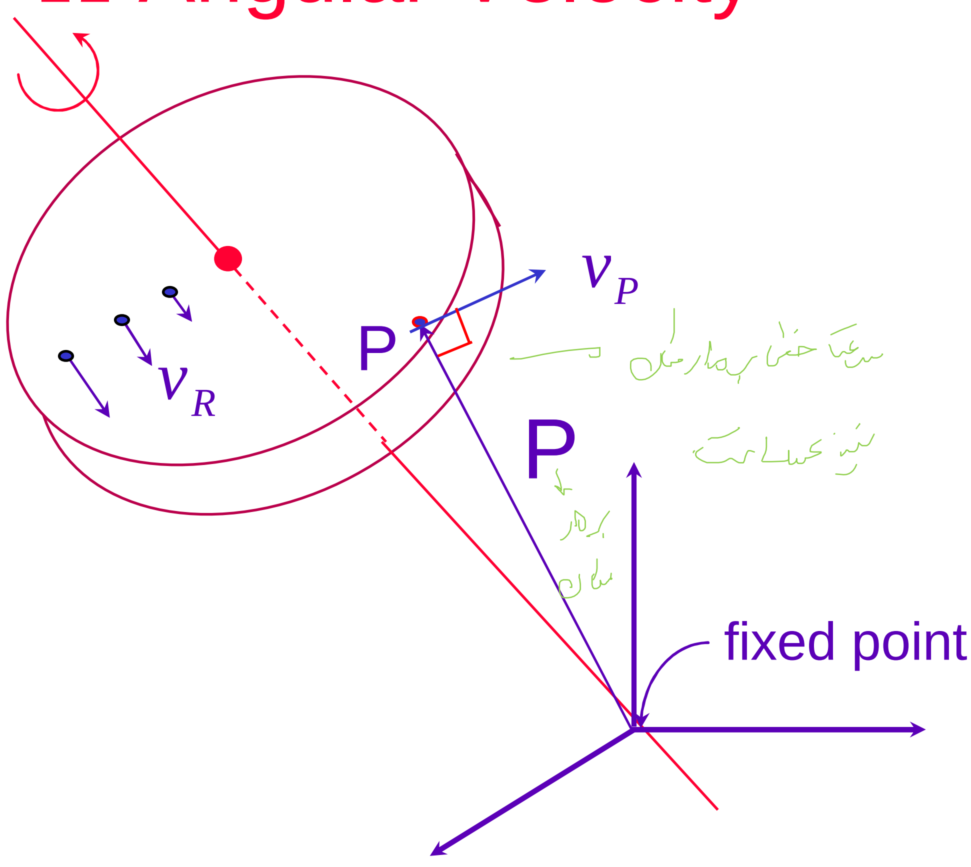
Rotational Motion

Ω Angular Velocity

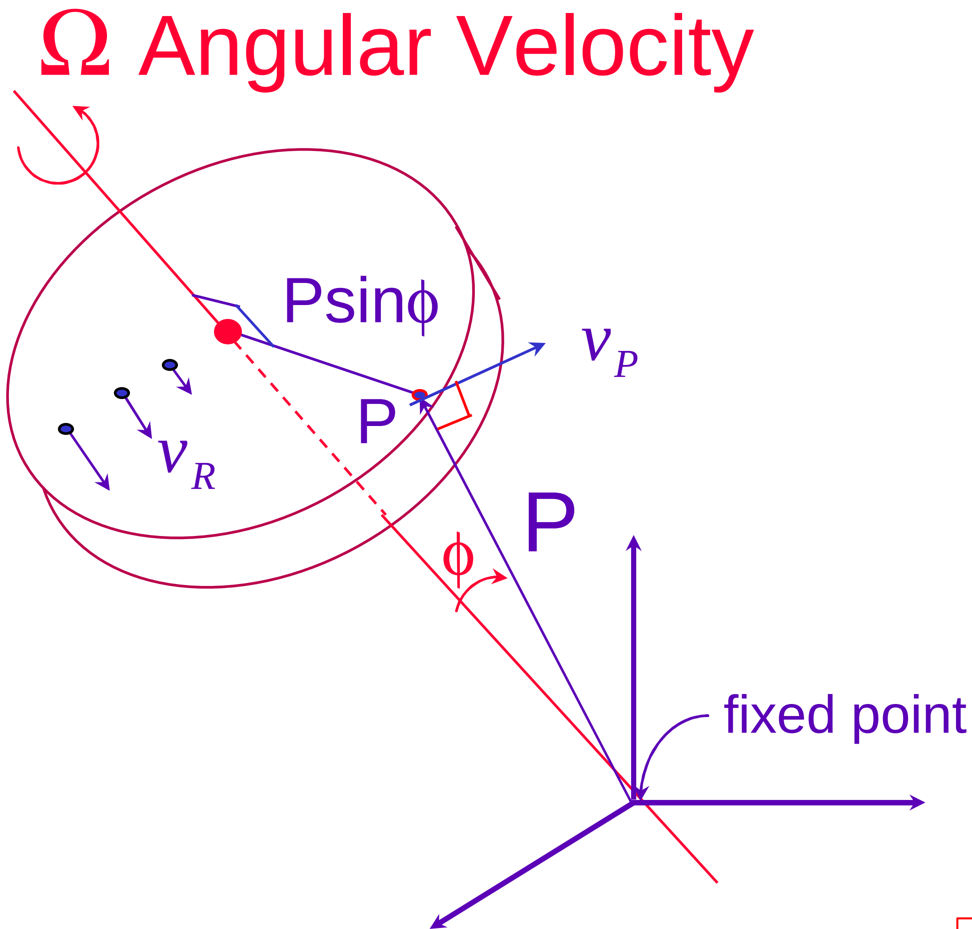


Rotational Motion

Ω Angular Velocity



Rotational Motion



v_P is proportional to:

- $\|\Omega\|$
- $\|P\sin\phi\|$

and

- $v_P \perp \Omega$
- $v_P \perp P$

$$\boxed{v_P = \Omega \times P}$$

Cross Product Operator

$$a=\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}, b=\begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}$$

$$c = a \times b \Rightarrow c = \hat{a} b$$

vectors ⇨ matrices

$$a \times \Rightarrow \hat{a}$$: a skew-symmetric matrix

$$c=\hat{a}b=\begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}\begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}$$

$$\boxed{c=\hat{a}b}$$

Cross Product Operator

$$v_P = \Omega \times P \Rightarrow v_P = \hat{\Omega} P$$

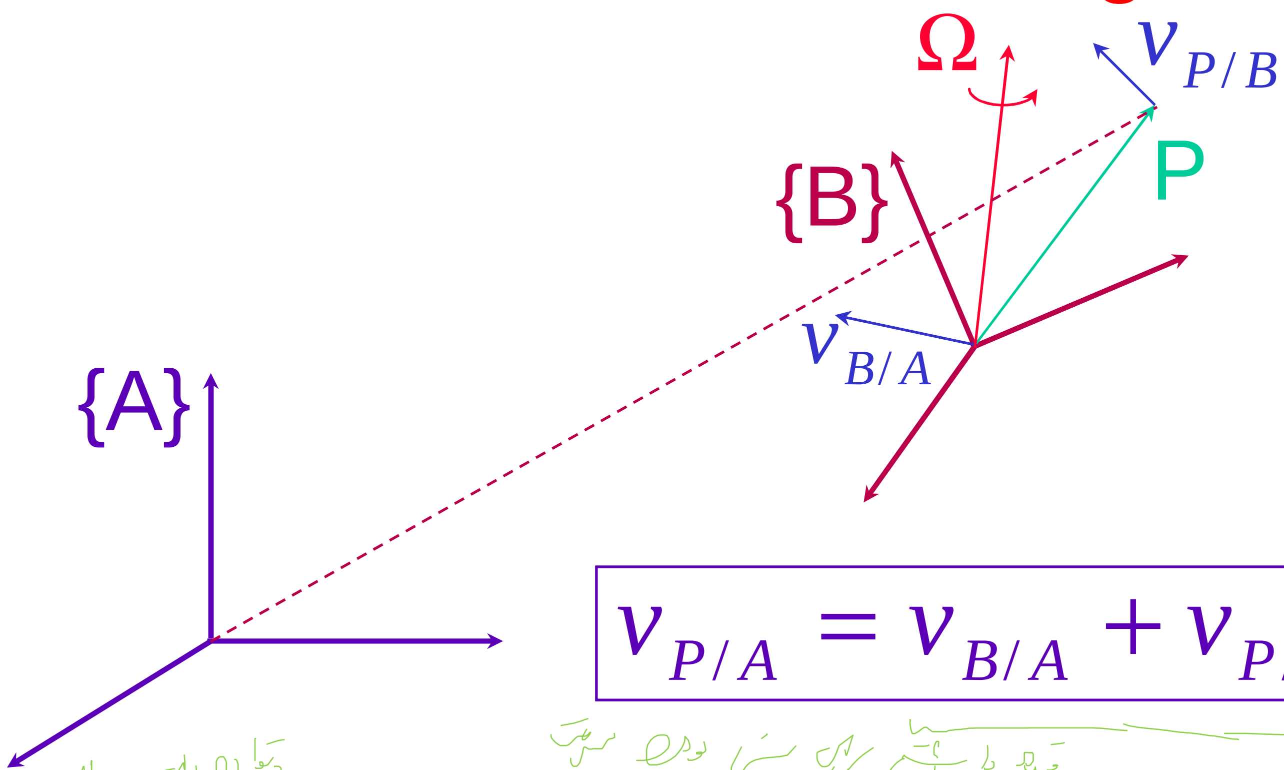
$\Omega \times \Rightarrow \hat{\Omega}$: a skew-symmetric matrix

$$\Omega = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}; P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}$$

$$v_P = \hat{\Omega} P = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix} \cdot \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}$$

$$\boxed{v_P = \hat{\Omega} P}$$

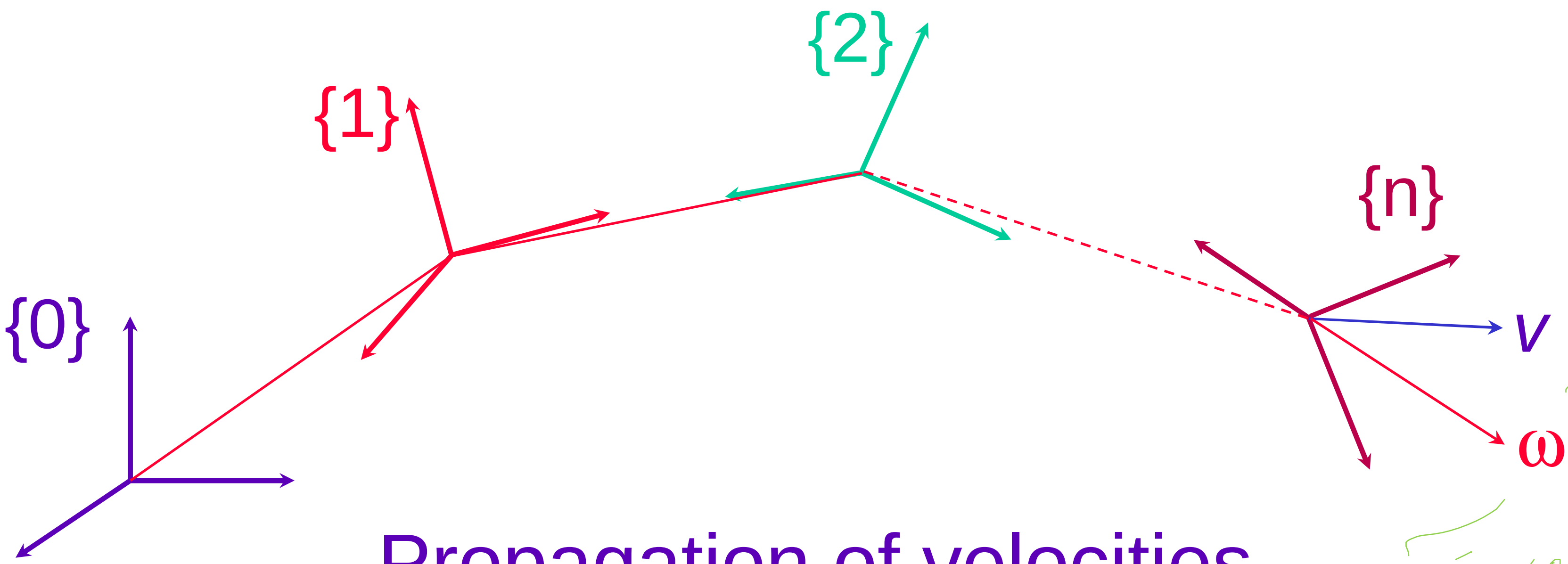
Simultaneous linear and angular motion



$$\boxed{v_{P/A} = v_{B/A} + v_{P/B} + \Omega \times P_B}$$

$$^{A}v_{P/A} = {}^{A}v_{B/A} + {}^{A}_{B}R \cdot {}^{B}v_{P/B} + {}^{A}\Omega_{B} \times {}^{A}_{B}R \cdot {}^{B}P_{B}$$

Spatial Mechanisms



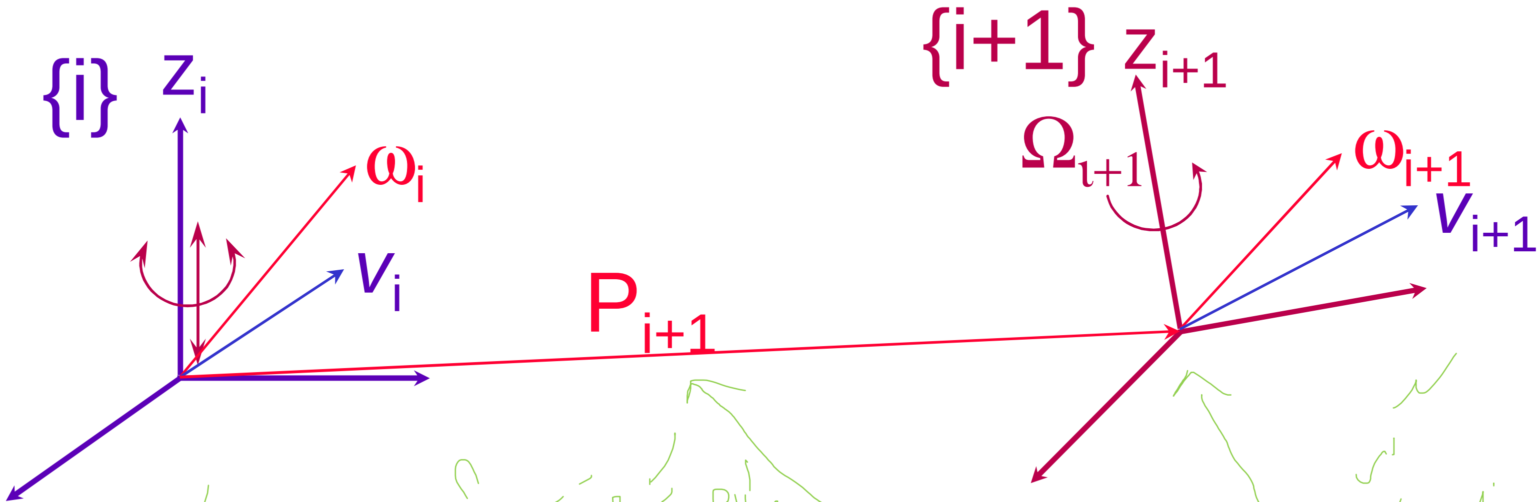
می‌خواهیم نسبت به زمین (0) و در دستگاه زمین (0) بدست آوریم.

Propagation of velocities

$\dot{x}$ → v : linear velocity
$\dot{x}$ → ω : angular velocity

$$\dot{x} = J(\theta).\dot{\theta}$$

Velocity propagation



prismatic

Linear

$$\boxed{v_{i+1} = v_i + \omega_i \times P_{i+1} + \dot{d}_{i+1} . Z_{i+1}}$$

revolute

Angular

$$\omega_{i+1} = \omega_i + \Omega_{i+1}$$

$$\Omega_{i+1} = \dot{\theta}_{i+1} . Z_{i+1}$$

Velocity propagation

Joint 1

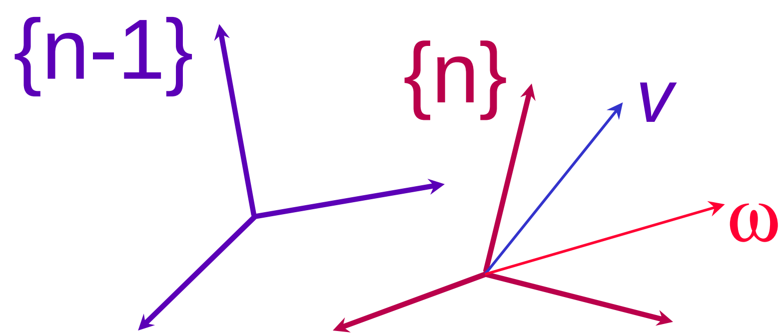
v_1 and ω_1 in frame {1}

Joint i+1

$$^{i+1}\omega_{i+1} = {}^{i+1}_{\;\;i}R.{}^{i}\omega_i + \dot{\theta}_{i+1}.{}^{i+1}Z_{i+1}$$

$$^{i+1}v_{i+1} = {}^{i+1}_{\;\;i}R.({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{i+1}) + \dot{d}_{i+1}.{}^{i+1}Z_{i+1}$$

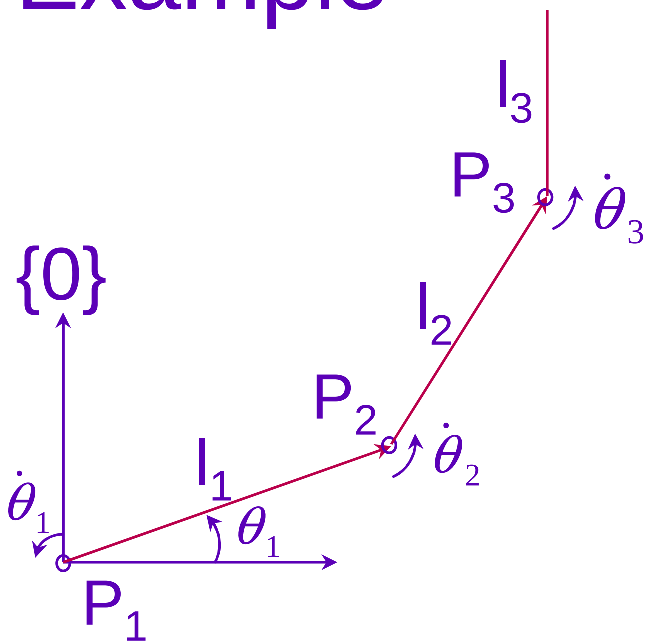
$$\Rightarrow \; {}^{n}\omega_n \text{ and } {}^{n}v_n$$



$$\begin{pmatrix} {}^{0}v_n \\ {}^{0}\omega_n \end{pmatrix} = \begin{pmatrix} {}^{0}_{n}R & 0 \\ 0 & {}^{0}_{n}R \end{pmatrix} . \begin{pmatrix} {}^{n}v_n \\ {}^{n}\omega_n \end{pmatrix}$$

6 x 6

Example



$$v_{i+1} = v_i + \omega_i \times P_{i+1}$$

- $v_{P_1} = 0$ $\qquad {}^0\omega_1 = \dot{\theta}_1 . {}^0Z_1$
- $v_{P_2} = v_{P_1} + \omega_1 \times P_2$
- $v_{P_3} = v_{P_2} + \omega_2 \times P_3$

$$ {}^0v_{P_2} = 0 + \begin{bmatrix} 0 & -\dot{\theta}_1 & 0 \\ \dot{\theta}_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} . \begin{bmatrix} l_1 . c_1 \\ l_1 . s_1 \\ 0 \end{bmatrix} = \begin{bmatrix} -l_1 . s_1 \\ l_1 . c_1 \\ 0 \end{bmatrix} . \dot{\theta}_1 $$

$$ {}^{0}v_{P_3} = {}^{0}v_{P_2} + {}^{0}\omega_2 \times {}^{0}P_3 $$

$$ {}^{0}v_{P_3} = \begin{bmatrix} -l_1.s_1 \\ l_1.c_1 \\ 0 \end{bmatrix}.\dot{\theta}_1 + \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.(\dot{\theta}_1 + \dot{\theta}_2).{}^{0}P_3 $$

$$ \begin{bmatrix} l_2.c_{12} \\ l_2.s_{12} \\ 0 \end{bmatrix} $$

$$ = \begin{bmatrix} -l_1.s_1 \\ l_1.c_1 \\ 0 \end{bmatrix}.\dot{\theta}_1 + \begin{bmatrix} -l_2.s_{12} \\ l_2.c_{12} \\ 0 \end{bmatrix}.(\dot{\theta}_1 + \dot{\theta}_2) $$

$$ {}^{0}\omega_3 = (\underbrace{\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3}).{}^{0}Z_0 $$

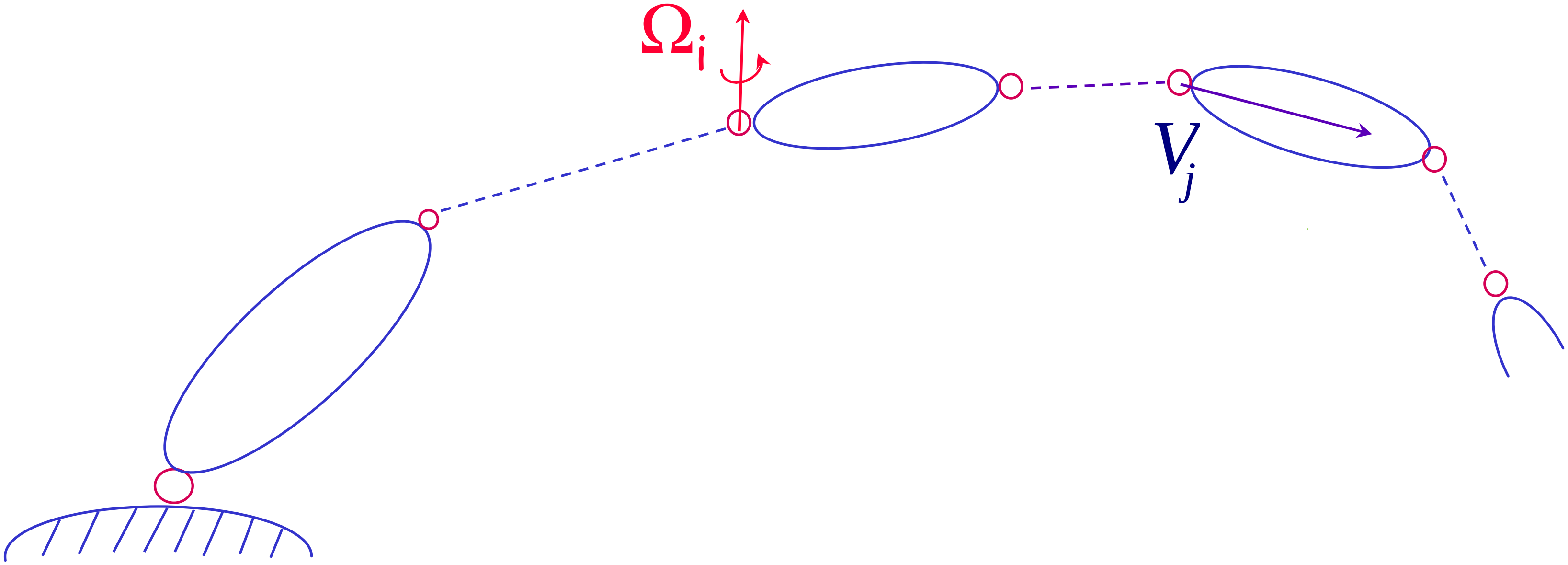
$$
{}^{0}v_{P_3} = \underbrace{\begin{bmatrix} -(l_1 s_1 + l_2 s_{12}) & -l_2 s_{12} & 0 \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{J_v} . \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}
$$

$$
{}^{0}\omega_3 = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}}_{J_\omega} . \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}
$$

${}^{0}Z_{0}$

$$
\begin{pmatrix} v \\ \omega \end{pmatrix} = J . \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix}
$$

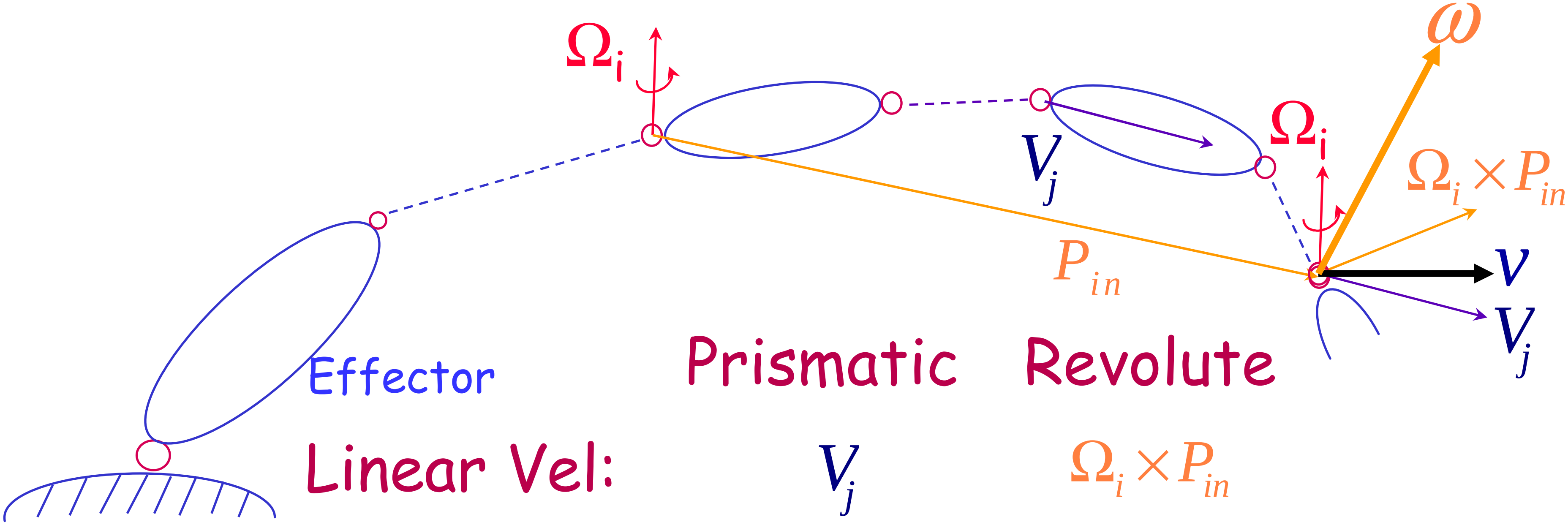
The Jacobian (EXPLICIT FORM)



Revolute Joint $\quad \Omega_i = Z_i \dot{q}_i$

Prismatic Joint $\quad V_i = Z_i \dot{q}_i$

The Jacobian (EXPLICIT FORM)



	Prismatic	Revolute
Linear Vel:	V_j	$\Omega_i \times P_{in}$
Angular Vel:	none	Ω_i

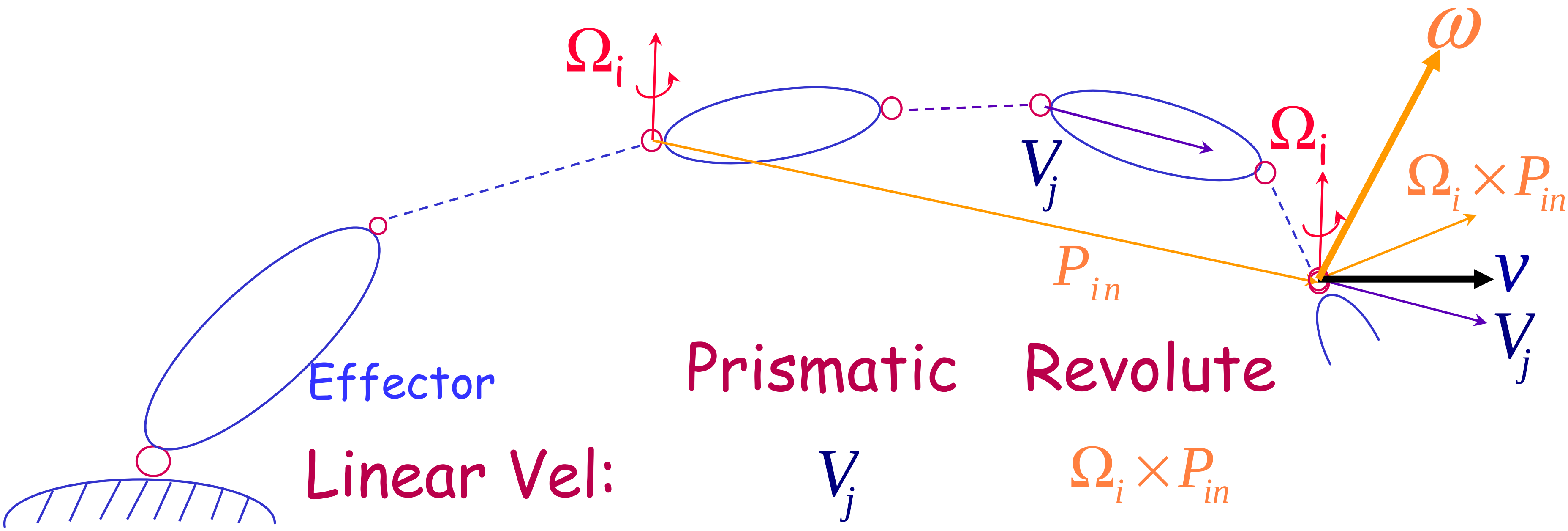
Effector Linear Velocity

$$v = \sum_{i=1}^{n} [\in_i V_i + \overline{\in}_i (\Omega_i \times P_{in})] \quad \Leftarrow \quad V_i = Z_i \dot{q}_i$$

Effector Angular Velocity

$$\omega = \sum_{i=1}^{n} \overline{\in}_i \, \Omega_i \quad \Leftarrow \quad \Omega_i = Z_i \dot{q}_i$$

The Jacobian (EXPLICIT FORM)



	Prismatic	Revolute
Linear Vel:	V_j	$\Omega_i \times P_{in}$
Angular Vel:	none	Ω_i

Effector Linear Velocity

$$v = \sum_{i=1}^{n} [\in_i Z_i + \bar{\in}_i (Z_i \times P_{in})] \dot{q}_i \quad \Leftarrow \quad V_i = Z_i \dot{q}_i$$

Effector Angular Velocity

$$\omega = \sum_{i=1}^{n} (\bar{\in}_i Z_i) \dot{q}_i \quad \Leftarrow \quad \Omega_i = Z_i \dot{q}_i$$

$$v = [\in_1 Z_1 + \bar{\in}_1 (Z_1 \times P_{1n})]\dot{q}_1 + \cdots$$
$$+ [\in_{n-1} Z_{n-1} + \bar{\in}_{n-1} (Z_{n-1} \times P_{(n-1)n})]\dot{q}_{n-1} + \in_n Z_n \dot{q}_n$$

$$v = \begin{bmatrix} \in_1 Z_1 + \bar{\in}_1 (Z_1 \times P_{1n}) & \in_2 Z_2 + \bar{\in}_2 (Z_2 \times P_{2n}) & \cdots \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

$$\boxed{v = J_v \dot{q}}$$

$$\omega = \bar{\in}_1 Z_1 \dot{q}_1 + \bar{\in}_2 Z_2 \dot{q}_2 + \cdots + \bar{\in}_n Z_n \dot{q}_n$$

$$\omega = \begin{bmatrix} \bar{\in}_1 Z_1 & \bar{\in}_2 Z_2 & \cdots & \bar{\in}_n Z_n \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

$$\boxed{\omega = J_\omega \dot{q}}$$

The Jacobian

$$J = \begin{pmatrix} J_v \\ \hline J_w \end{pmatrix}$$

Matrix J_v (direct differentiation)

$$v = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \dot{x}_P = \frac{\partial x_P}{\partial q_1} . \dot{q}_1 + \frac{\partial x_P}{\partial q_2} . \dot{q}_2 + \cdots + \frac{\partial x_P}{\partial q_n} . \dot{q}_n$$

رابطه برحسب ماتریس

۱

مشتق

مستقیم

$$J_v = \begin{pmatrix} \dfrac{\partial x_P}{\partial q_1} & \dfrac{\partial x_P}{\partial q_2} & \cdots & \dfrac{\partial x_P}{\partial q_n} \end{pmatrix}$$

Jacobian in a Frame

Vector Representation

با توجه O در P حساب
تعریف را
تعریف در base

$$J = \begin{pmatrix} \dfrac{\partial x_P}{\partial q_1} & \dfrac{\partial x_P}{\partial q_2} & \cdots & \dfrac{\partial x_P}{\partial q_n} \\ \bar{\epsilon}_1 . Z_1 & \bar{\epsilon}_2 . Z_2 & \cdots & \bar{\epsilon}_n . Z_n \end{pmatrix}$$

In {0}

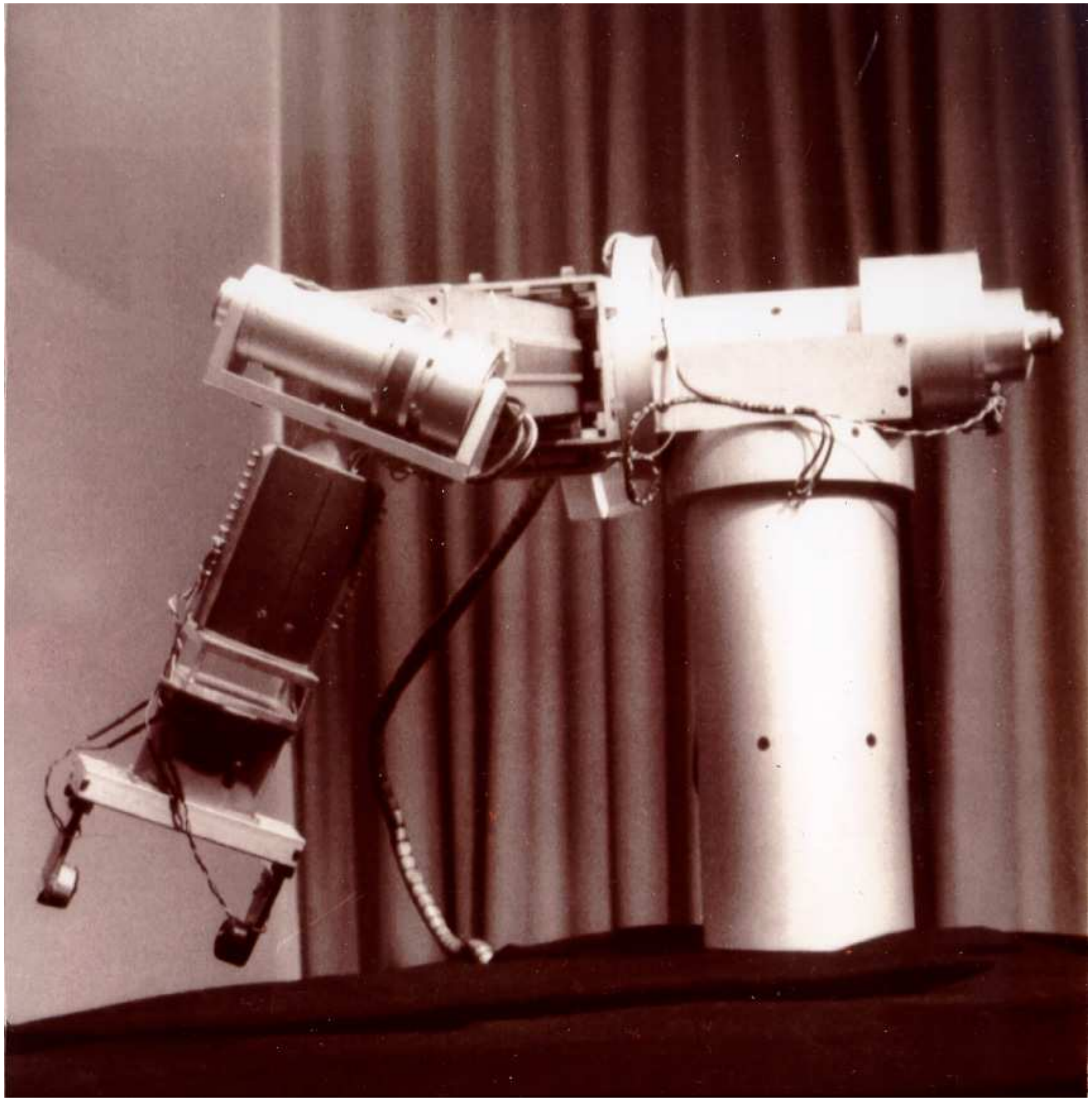
$$^0J = \begin{pmatrix} \dfrac{\partial\, {}^0x_P}{\partial q_1} & \dfrac{\partial\, {}^0x_P}{\partial q_2} & \cdots & \dfrac{\partial\, {}^0x_P}{\partial q_n} \\ \bar{\epsilon}_1 . {}^0Z_1 & \bar{\epsilon}_2 . {}^0Z_2 & \cdots & \bar{\epsilon}_n . {}^0Z_n \end{pmatrix}$$

J in Frame {0}

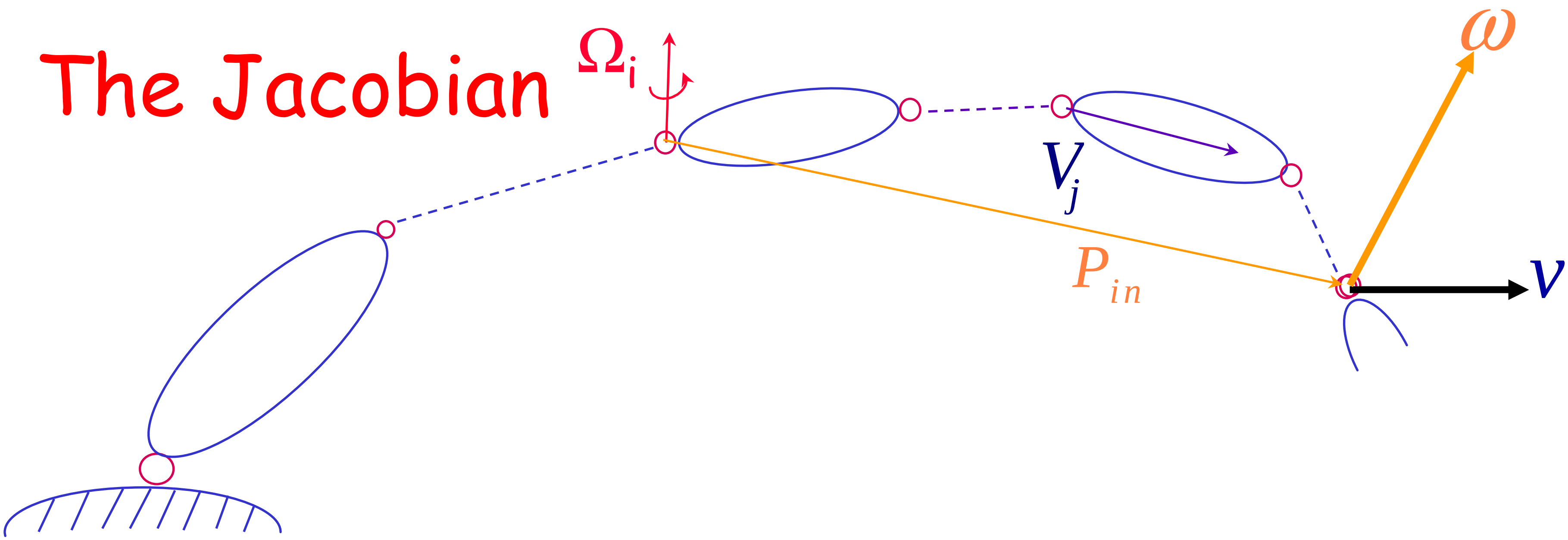
$$
{}^{0}Z_i = {}^{0}_{i}R \; {}^{i}Z_i ; \quad {}^{i}Z_i = Z = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
$$

$$
{}^{0}J = \begin{pmatrix} \dfrac{\partial}{\partial q_1}({}^{0}x_P) & \dfrac{\partial}{\partial q_2}({}^{0}x_P) & \cdots & \dfrac{\partial}{\partial q_n}({}^{0}x_P) \\ \overline{\epsilon}_1 . ({}^{0}_{1}R . Z) & \overline{\epsilon}_2 . ({}^{0}_{2}R . Z) & \cdots & \overline{\epsilon}_n . ({}^{0}_{n}R . Z) \end{pmatrix}
$$

Stanford Scheinman Arm



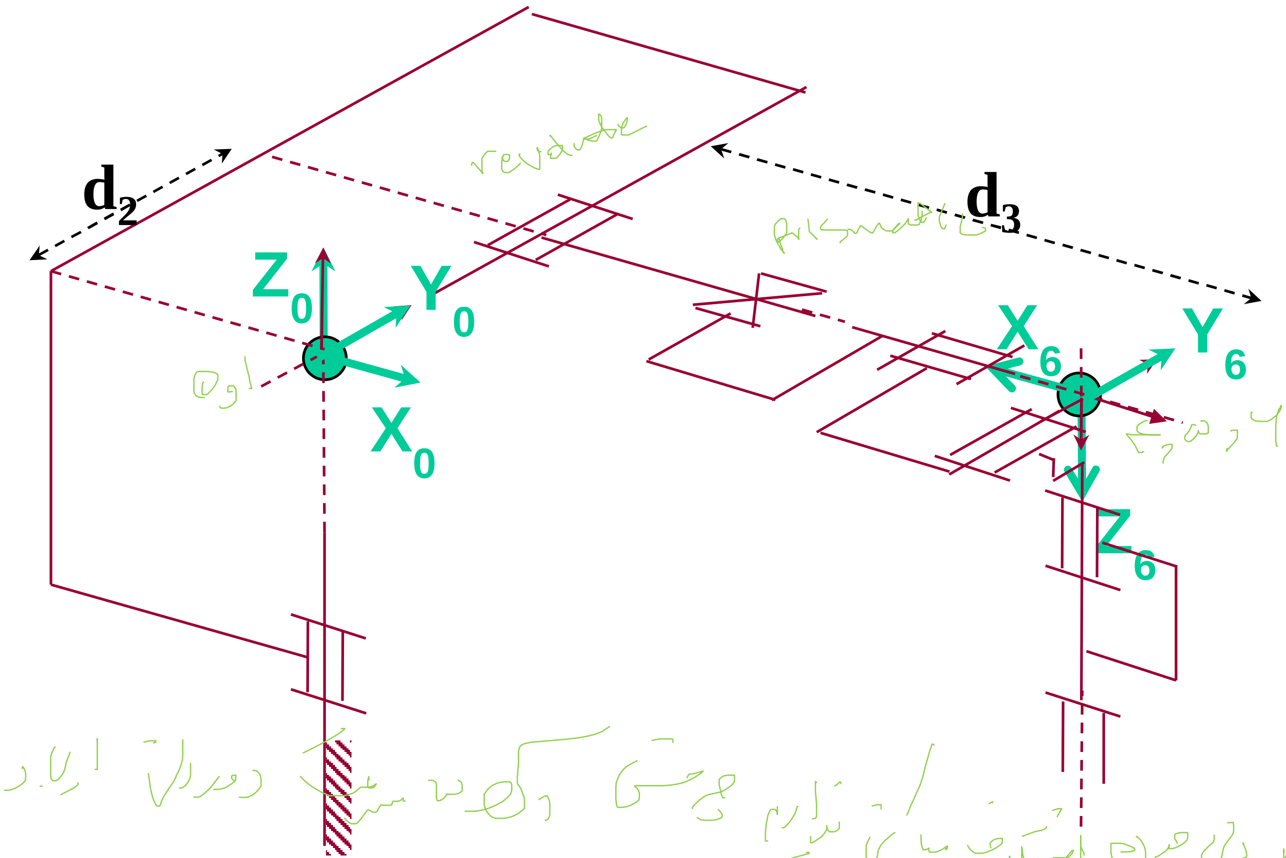
The Jacobian



$$J = \begin{pmatrix} J_{v\,3\times 6} \\ \hline J_{w\,3\times 6} \end{pmatrix}_{6\times 6} \qquad v = J_v \dot{q} \qquad \omega = J_\omega \dot{q}$$

$$J_v = \left[\in_1 Z_1 + \bar{\in}_1 (Z_1 \times P_{1n}) \quad \in_2 Z_2 + \bar{\in}_2 (Z_2 \times P_{2n}) \quad \cdots \right]$$

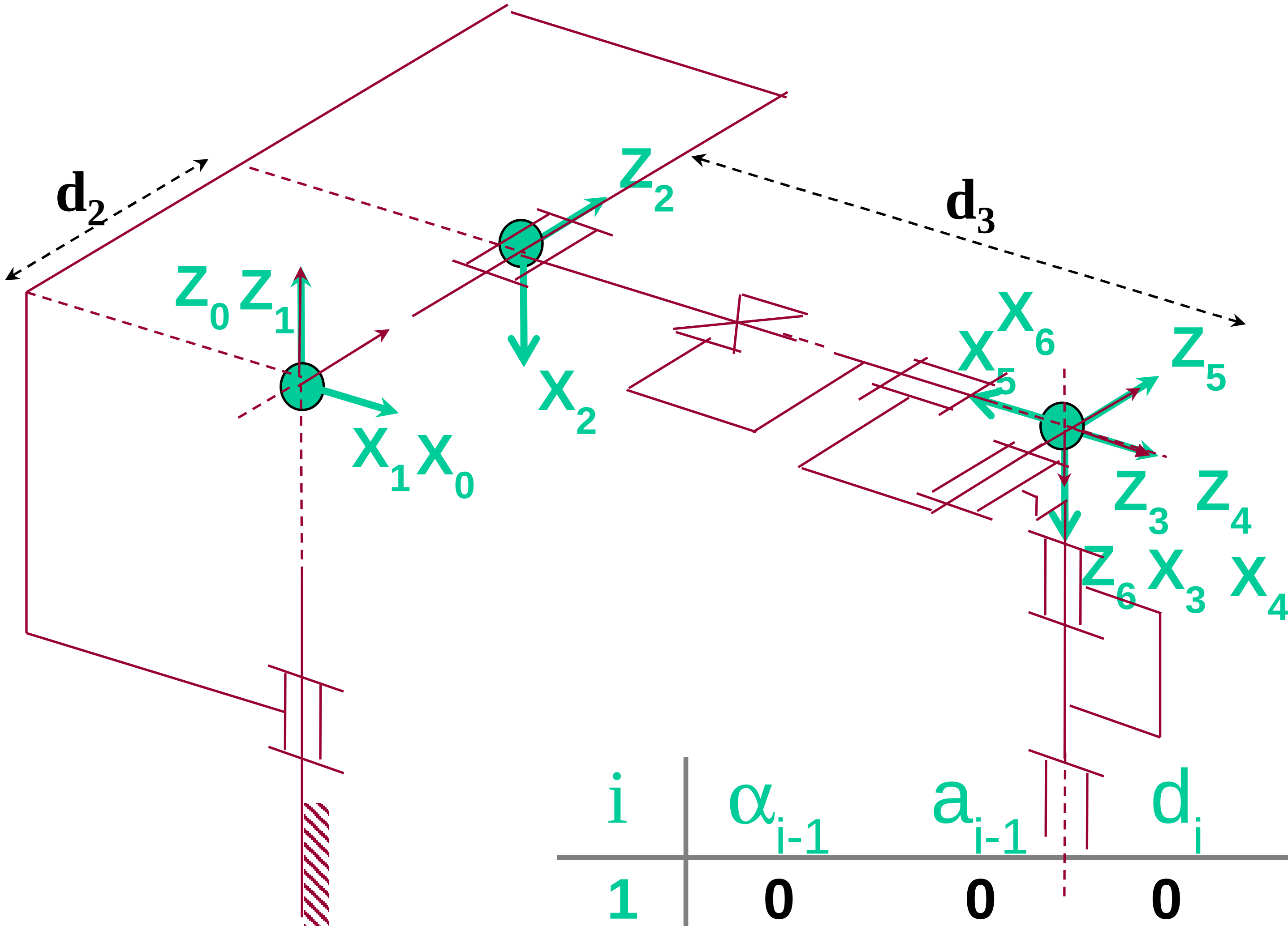
$$J_\omega = \left[\bar{\in}_1 Z_1 \quad \bar{\in}_2 Z_2 \quad \cdots \quad \bar{\in}_n Z_n \right]$$



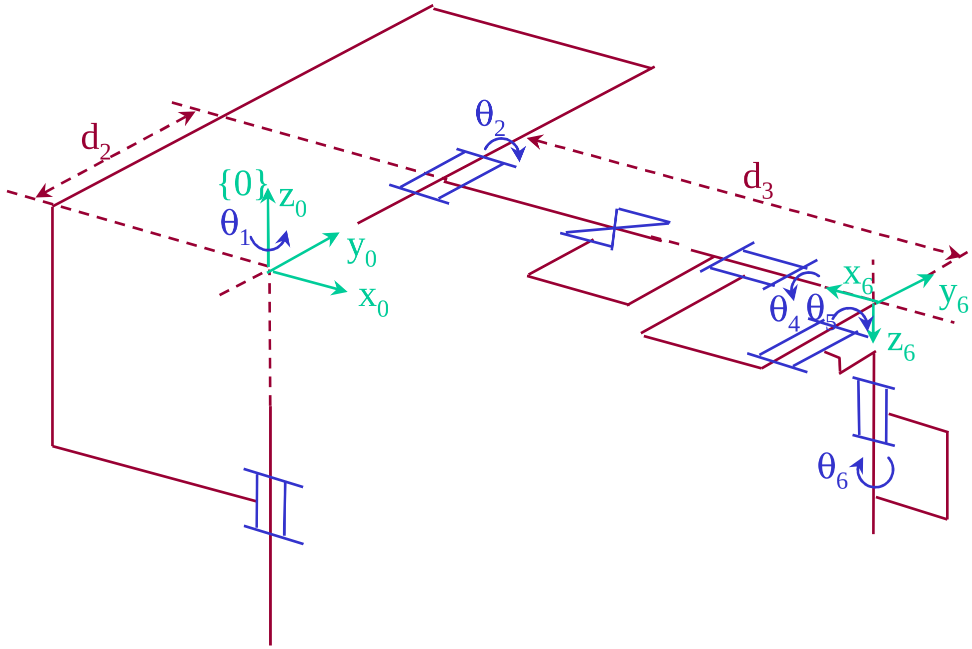
revolute prismatic

$$ {}^0J = \begin{pmatrix} Z_1 \times P_{13} & Z_2 \times P_{23} & Z_3 & 0 & 0 & 0 \\ {}^0Z_1 & {}^0Z_2 & 0 & {}^0Z_4 & {}^0Z_5 & {}^0Z_6 \end{pmatrix} $$

prismatic



i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90	0	d_2	θ_2
3	90	0	d_3	0
4	0	0	0	θ_4
5	-90	0	0	θ_5
6	90	0	0	θ_6



i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90	0	d_2	θ_2
3	90	0	d_3	0
4	0	0	0	θ_4
5	-90	0	0	θ_5
6	90	0	0	θ_6

فرمول از کریگ اومده

$$ {}^{i-1}_{\ \ i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i\, c\alpha_{i-1} & c\theta_i\, c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}\, d_i \\ s\theta_i\, s\alpha_{i-1} & c\theta_i\, s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}\, d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Forward Kinematics: $ {}^{0}_{N}T = {}^{0}_{1}T\ {}^{1}_{2}T \cdots {}^{N-1}_{\ \ N}T $

Stanford Scheinman Arm

$$ {}^{0}_{1}T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{1}_{2}T = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ -s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}_{3}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -d_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$
{}^{3}_{4}T = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{4}_{5}T = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_5 & -c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{5}_{6}T = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

${}^0z_1$

$$
{}^0_1T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

${}^0z_2$

$$
{}^0_2T = \begin{bmatrix} c_1c_2 & -c_1s_2 & -s_1 & -s_1d_2 \\ s_1c_2 & -s_1s_2 & c_1 & c_1d_2 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

${}^0z_3$

$$
{}^0_3T = \begin{bmatrix} c_1c_2 & -s_1 & c_1s_2 & c_1d_3s_2 - s_1d_2 \\ s_1c_2 & c_1 & s_1s_2 & s_1d_3s_2 + c_1d_2 \\ -s_2 & 0 & c_2 & d_3c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

${}^0z_4$

$$
{}^0_4T = \begin{bmatrix} c_1c_2c_4 - s_1s_4 & -c_1c_2s_4 - s_1c_4 & \boxed{c_1s_2} & c_1d_3s_2 - s_1d_2 \\ s_1c_2c_4 + c_1s_4 & -s_1c_2s_4 + c_1c_4 & s_1s_2 & s_1d_3s_2 + c_1d_2 \\ -s_2c_4 & s_2s_4 & c_2 & d_3c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

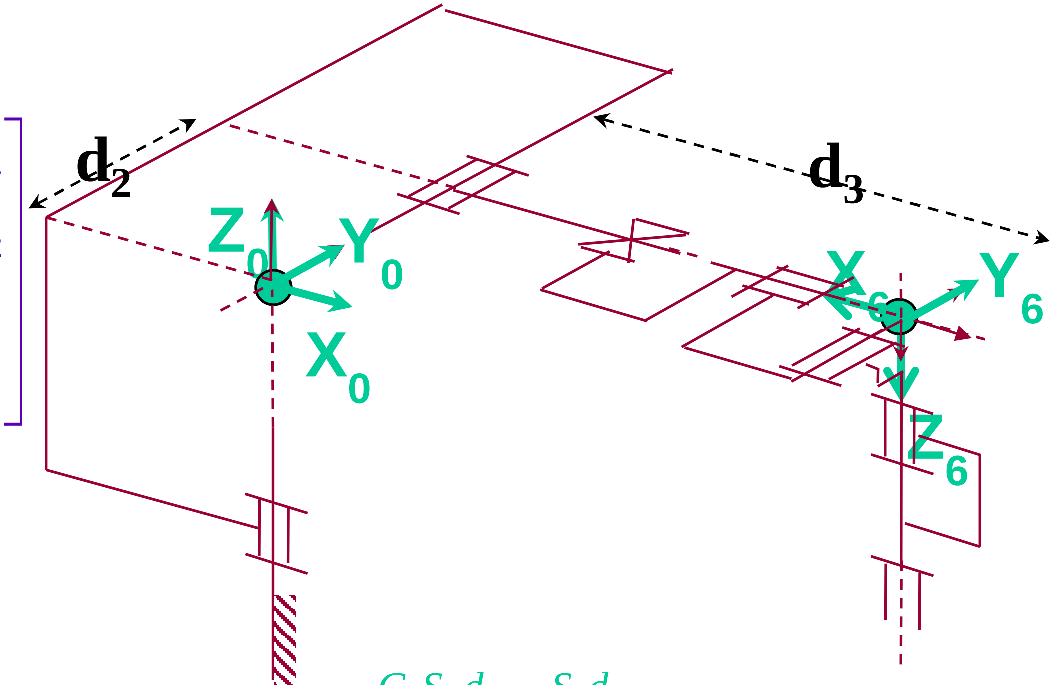
${}^0z_5$

$$
{}^0_5T = \begin{bmatrix} X & X & -c_1c_2s_4 - s_1c_4 & c_1d_3s_2 - s_1d_2 \\ X & X & -s_1c_2s_4 + c_1c_4 & s_1d_3s_2 + c_1d_2 \\ X & X & s_2s_4 & d_3c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

${}^0z_4$

$$
{}^0_6T = \begin{bmatrix} X & X & c_1c_2c_4s_5 - s_1s_4s_5 + c_1s_2s_5 & c_1d_3s_2 - s_1d_2 \\ X & X & s_1c_2c_4s_5 + c_1s_4s_5 + s_1s_2c_5 & s_1d_3s_2 + c_1d_2 \\ X & X & -s_2c_4s_5 + c_5c_2 & d_3c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{0}_{6}T=\begin{bmatrix} X & X & c_1c_2c_4s_5-s_1s_4s_5+c_1s_2s_5 & c_1d_3s_2-s_1d_2 \\ X & X & s_1c_2c_4s_5+c_1s_4s_5+s_1s_2c_5 & s_1d_3s_2+c_1d_2 \\ X & X & -s_2c_4s_5+c_5c_2 & d_3c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$



$$
x=\begin{pmatrix} x_P \\ r_1 \\ r_2 \\ r_3 \end{pmatrix}=\left|\begin{array}{c}
C_1S_2d_3-S_1d_2 \\
S_1S_2d_3+C_1d_2 \\
C_2d_3 \\
C_1[C_2(C_4C_5C_6-S_4S_6)-S_2S_5C_6]-S_1(S_4C_5C_6+C_4S_6) \\
S_1[C_2(C_4C_5C_6-S_4S_6)-S_2S_5C_6]+C_1(S_4C_5C_6+C_4S_6) \\
-S_2(C_4C_5C_6-S_4S_6)-C_2S_5C_6 \\
C_1[-C_2(C_4C_5S_6+S_4C_6)+S_2S_5S_6]-S_1(-S_4C_5S_6+C_4C_6) \\
S_1[-C_2(C_4C_5S_6+S_4C_6)+S_2S_5S_6]+C_1(-S_4C_5S_6+C_4C_6) \\
S_2(C_4C_5S_6+S_4C_6)+C_2S_5S_6 \\
C_1(C_2C_4S_5+S_2C_5)-S_1S_4S_5 \\
S_1(C_2C_4S_5+S_2C_5)+C_1S_4S_5 \\
-S_2C_4S_5+C_2C_5
\end{array}\right|
$$

Stanford Scheinman Arm Jacobian

$$
{}^0J = \begin{pmatrix} \dfrac{\partial\, {}^0x_P}{\partial q_1} & \dfrac{\partial\, {}^0x_P}{\partial q_2} & \dfrac{\partial\, {}^0x_P}{\partial q_3} & 0 & 0 & 0 \\ {}^0Z_1 & {}^0Z_2 & 0 & {}^0Z_4 & {}^0Z_5 & {}^0Z_6 \end{pmatrix}
$$

$$
\begin{bmatrix}
-c_1 d_2 - s_1 s_2 d_3 & c_1 c_2 d_3 & c_1 s_2 & 0 & 0 & 0 \\
-s_1 d_2 + c_1 s_2 d_3 & s_1 c_2 d_3 & s_1 s_2 & 0 & 0 & 0 \\
0 & -s_2 d_3 & c_2 & 0 & 0 & 0 \\
0 & -s_1 & 0 & c_1 s_2 & -c_1 c_2 s_4 - s_1 c_4 & c_1 c_2 c_4 s_5 - s_1 s_4 s_5 + c_1 s_2 c_5 \\
0 & c_1 & 0 & s_1 s_2 & -s_1 c_2 s_4 + c_1 c_4 & s_1 c_2 c_4 s_5 + c_1 s_4 s_5 + s_1 s_2 c_5 \\
1 & 0 & 0 & c_2 & s_2 s_4 & -s_2 c_4 s_5 + c_5 c_2
\end{bmatrix}
$$

Kinematic Singularity

The Effector Locality loses the ability to move in a direction or to rotate about a direction - **singular direction**

حتماً با یک basis عضو مشترک دارد

$$J = \left(J_1 \;\; J_2 \;\; \cdots \;\; J_n \right)$$

J_0

$$\det\left(J_a \right) = 0$$

اگر دترمینان J برابر با صفر بشه rank (J) از n کمتر

خود را از دست داده باشد (کم) اگر سیستمی کامل درجه آزادی داشت را کامل می‌سازیم.

$$\det\left({}^{i}J \right) = \det\left({}^{j}J \right)$$

Kinematic Singularity

ortho normal
full rank

$$ {}^{B}J = \begin{pmatrix} {}^{B}_{A}R & 0 \\ 0 & {}^{B}_{A}R \end{pmatrix} {}^{A}J $$

position

6x6

$$ \det[{}^{B}J] \equiv \det[{}^{A}J] $$

$$ \boxed{\det\left({}^{i}J\right) = \det\left({}^{j}J\right)} $$

Singular Configurations

در هر دستگاه همه متغیرها [illegible] و وابسته [illegible]

[illegible]

$$\det[J(q)] = 0$$

$\Longrightarrow$ Singular Configurations

[illegible]

$$\det[J(q)] = S_1(q)S_2(q)\ldots S_s(q) = 0$$

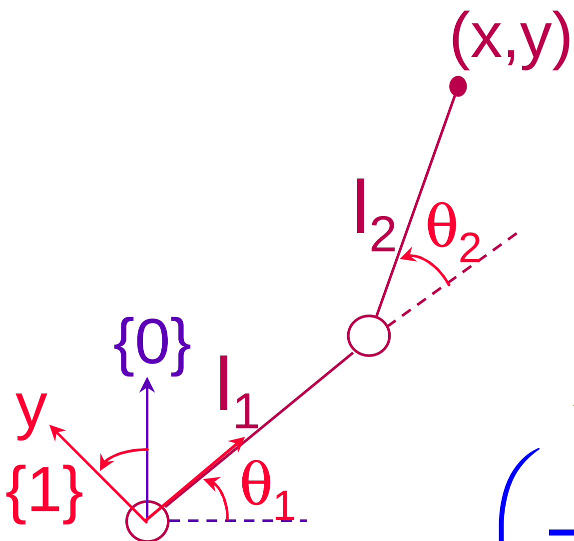
$$\Rightarrow \boxed{\begin{array}{l} S_1(q) = 0 \\ S_2(q) = 0 \\ \quad\vdots \\ S_s(q) = 0 \end{array}}$$

اگر هر کدام صفر شد

(یا بیشتر از یکی)

[illegible] rank از [illegible] یکی بیشتر [illegible] کم شده است

Example (Kinematic Singularities)



$$x = l_1C1 + l_2C12$$

حل forward

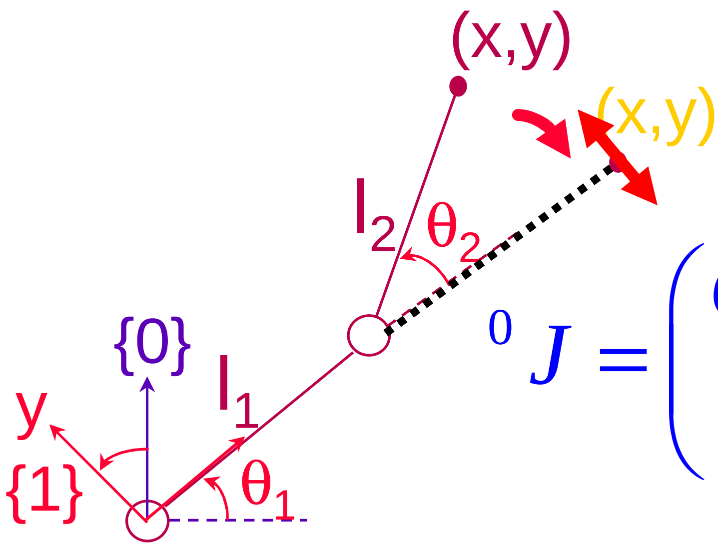
$$y = l_1S1 + l_2S12$$

$$J = \begin{pmatrix} -(l_1S1 + l_2S12) & -l_2S12 \\ l_1C1 + l_2C12 & l_2C12 \end{pmatrix}$$

$$\det(J) = l_1 l_2 S2$$

Singularity at $q_2 = k\pi$

Example (Kinematic Singularities)



$$^{1}J = {}^{1}_{0}R\ {}^{0}J$$

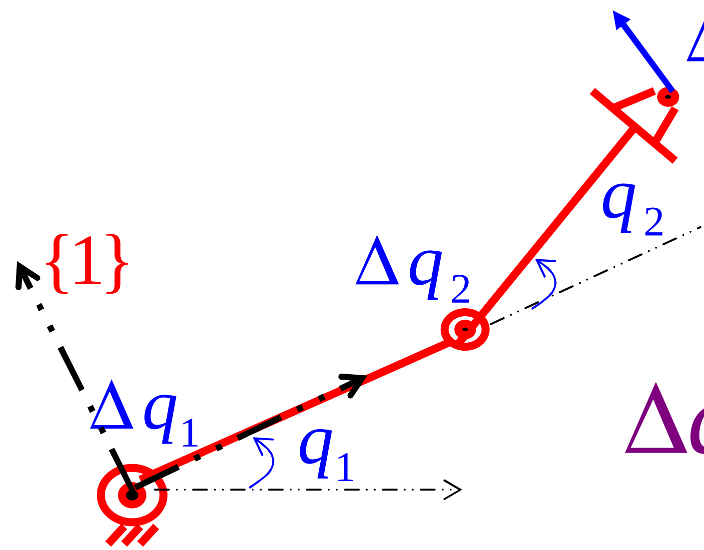
$$^{0}J = \begin{pmatrix} C1 & -S1 \\ S1 & C1 \end{pmatrix} \begin{pmatrix} -l_2 S2 & -l_2 S2 \\ l_1 + l_2 C2 & l_2 C2 \end{pmatrix}$$

At Singularity

$$^{1}J = \begin{pmatrix} 0 & 0 \\ l_1 + l_2 & l_2 \end{pmatrix}$$

$$\begin{cases} {}^{1}\delta x = 0 \\ {}^{1}\delta y = (l_1 + l_2)\delta\theta_1 + l_2 \delta\theta_2 \end{cases}$$

Small Displacements $\Delta q, \Delta X$



$$\Delta X = \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

$$\Delta q = J^{-1} \Delta X$$

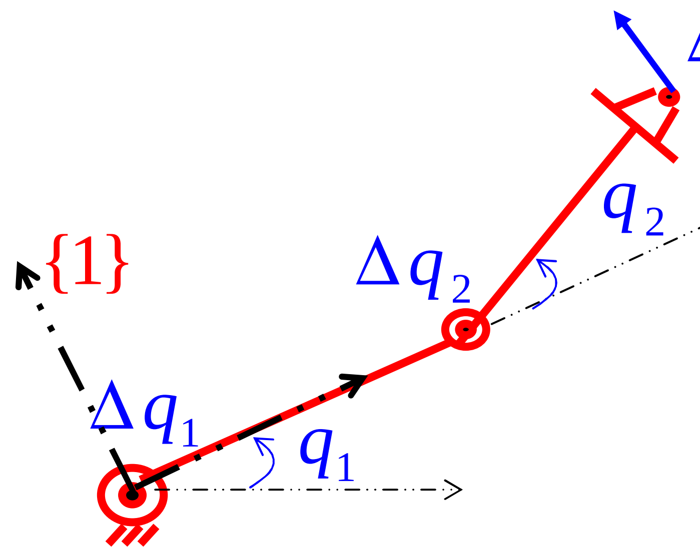
small θ_2

اینجا θ_2 را با صفر تقریب زده است و
ولی در آن کسرها اصلا تقریب نزده است

$$J_{(1)}^{-1} \cong \begin{pmatrix} \dfrac{1}{l_1 \theta_2} & \dfrac{1}{l_1} \\ -\dfrac{l_1 + l_2}{l_1 l_2 \theta_2} & -\dfrac{1}{l_1} \end{pmatrix}$$

Small Displacements $\Delta q, \Delta X$

$$\Delta X = \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

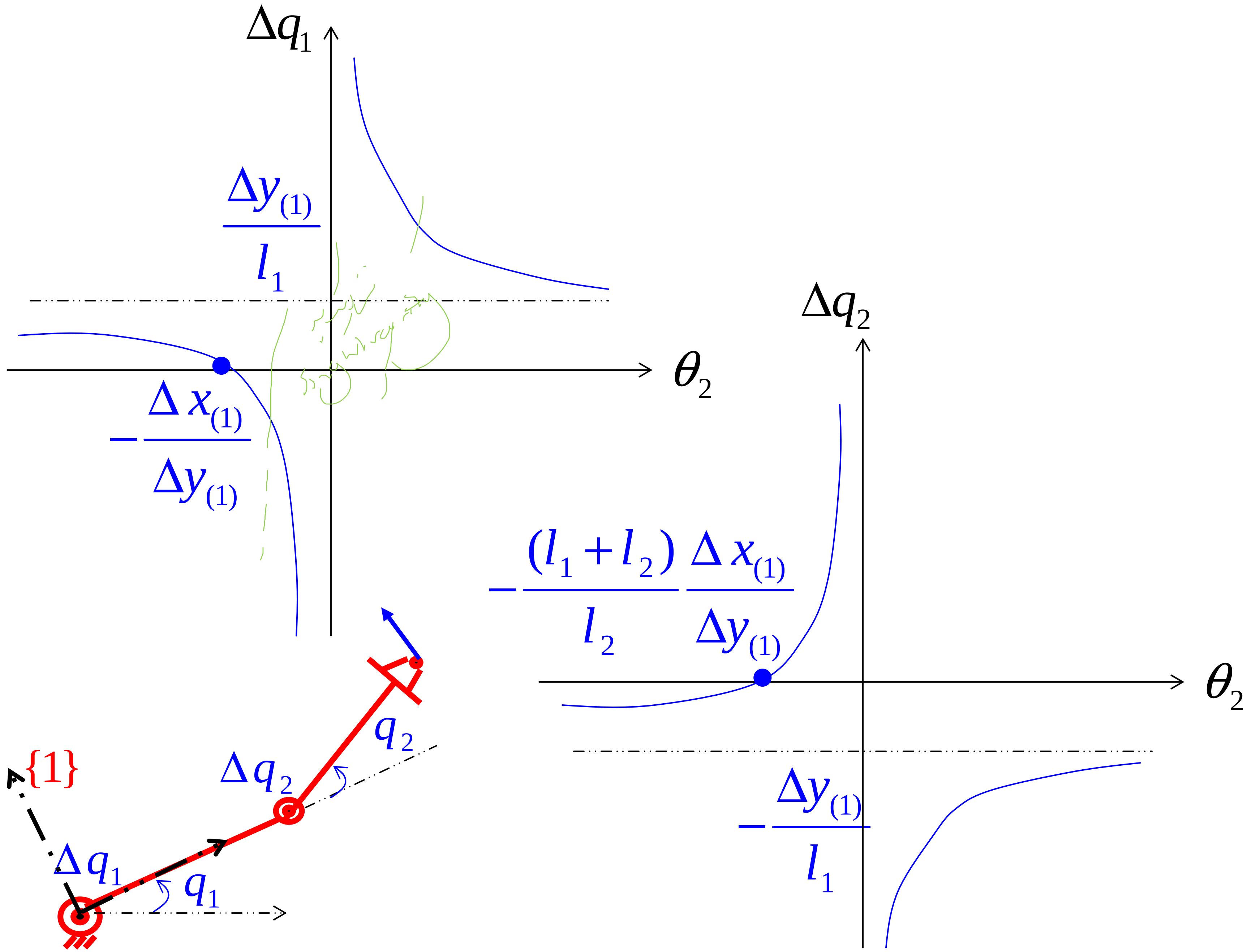


پس [illegible]
[illegible]

در حالتی که نزدیک به حالت singular است
مقادیر Δq_1 و Δq_2 بسیار بزرگ
[illegible]

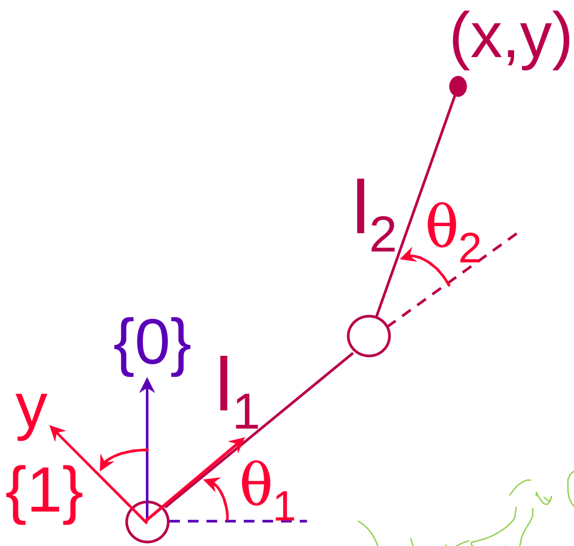
$$\Delta q_1 = \frac{\Delta x_{(1)}}{l_1} \cdot \frac{1}{\theta_2} + \frac{\Delta y_{(1)}}{l_1}$$

$$\Delta q_2 = \frac{(l_1 + l_2)\Delta x_{(1)}}{l_1 l_2} \cdot \frac{1}{\theta_2} + \frac{\Delta y_{(1)}}{l_1}$$



Δq_1
$\dfrac{\Delta y_{(1)}}{l_1}$
θ_2
$-\dfrac{\Delta x_{(1)}}{\Delta y_{(1)}}$
Δq_2
$-\dfrac{(l_1+l_2)}{l_2}\dfrac{\Delta x_{(1)}}{\Delta y_{(1)}}$
θ_2
$-\dfrac{\Delta y_{(1)}}{l_1}$
{1}
Δq_2
q_2
Δq_1
q_1

Kinematic Singularities (reduced matrix)



$$J = \begin{pmatrix} -\left(l_1 S1 + l_2 S12\right) & -l_2 S12 \\ l_1 C1 + l_2 C12 & l_2 C12 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix}$$

$$\det(J) = l_1 l_2 S2$$

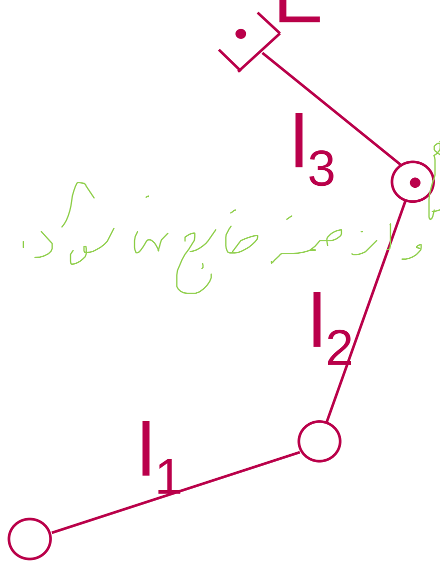
$$J = \begin{pmatrix} -\left(l_1 S1 + l_2 S12\right) & -l_2 S12 \\ l_1 C1 + l_2 C12 & l_2 C12 \end{pmatrix}$$

Singularity at $q_2 = k\pi$

$$
{}^{0}J_E = \begin{pmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \\ 1 & 1 & 1 \end{pmatrix}
$$

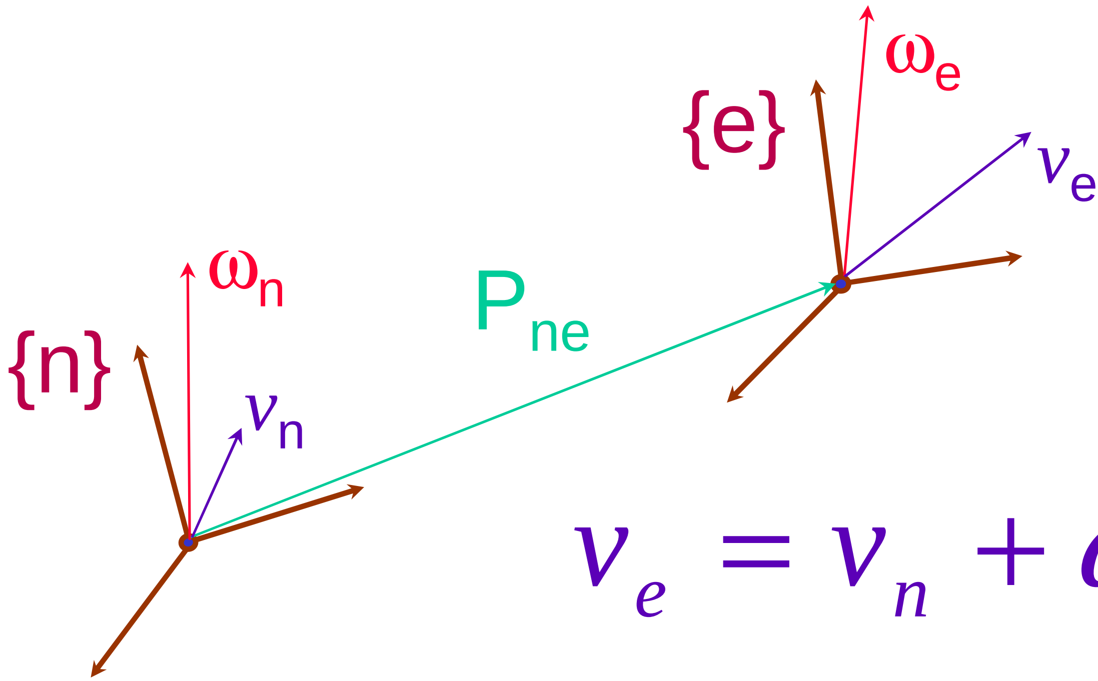
بعد Z هم
اگر نداشته و از صفحه خارج نشود

[illegible]



$$
{}^{0}J_E = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}
$$

Jacobian at the End-Effector



$$v_e = v_n + \omega_n \times P_{ne}$$

$$\begin{cases} v_e = v_n - P_{ne} \times \omega_n \\ \omega_e = \omega_n \end{cases}$$

$$
\begin{cases}
v_e = v_n - P_{ne} \times \omega_n \\
\omega_e = \omega_n
\end{cases}
$$

$$
\begin{pmatrix} v_e \\ \omega_e \end{pmatrix} = \left(\begin{array}{c|c} I & -\hat{P}_{ne} \\ \hline O & I \end{array}\right) \begin{pmatrix} v_n \\ \omega_n \end{pmatrix}
$$

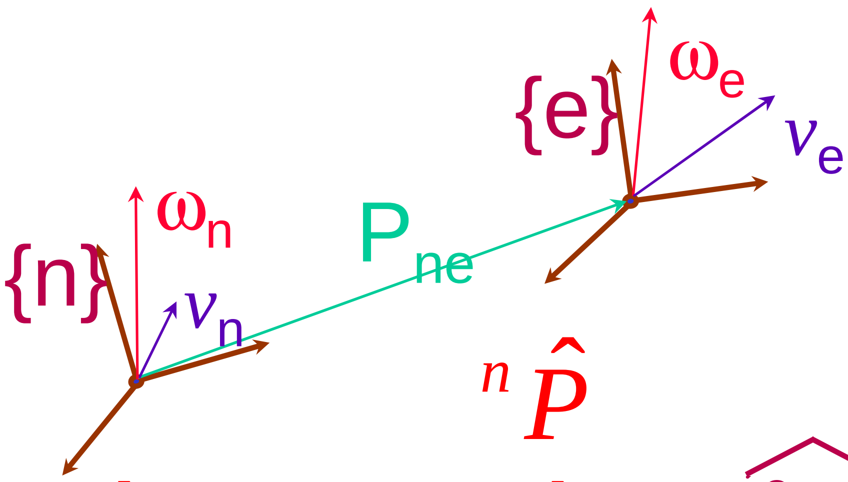
$$
J_e \dot{q} = \left(\begin{array}{c|c} I & -\hat{P}_{ne} \\ \hline O & I \end{array}\right) J_n \dot{q}
$$

[illegible]

و ثابت

$$
J_e = \left(\begin{array}{c|c} I & -\hat{P}_{ne} \\ \hline O & I \end{array}\right) J_n
$$

Cross Product Operator (in diff. frames)



$${}^{n}\hat{P}$$

$${}^{0}J_e = \begin{pmatrix} I & -{}^{0}\hat{P}_{ne} \\ 0 & I \end{pmatrix} {}^{0}J_n$$

$${}^{0}\hat{P} \neq {}^{0}_{n}R \; {}^{n}\hat{P};$$

$$\widehat{{}^{0}P} = \widehat{({}^{0}_{n}R.{}^{n}P)} \neq {}^{0}_{n}R.\widehat{{}^{n}P}$$

$${}^{0}P \times {}^{0}\omega = {}^{0}_{n}R.({}^{n}P \times {}^{n}\omega)$$

$${}^{0}\hat{P}.{}^{0}\omega = {}^{0}_{n}R.({}^{n}\hat{P}.{}^{n}\omega) = {}^{0}_{n}R.({}^{n}\hat{P}.{}^{0}_{n}R^{T}.{}^{0}\omega)$$

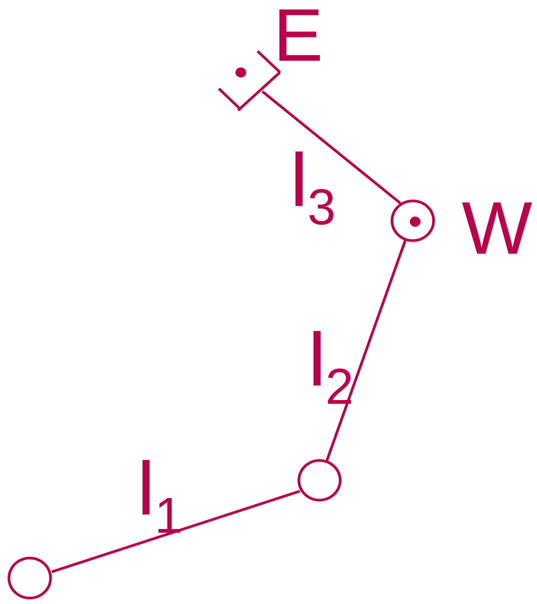
حاصل ضرب

$$\boxed{{}^{0}\hat{P} = {}^{0}_{n}R \; {}^{n}\hat{P} \; {}^{0}_{n}R^{T}}$$

$$
{}^{i}J = \begin{pmatrix} {}^{i}_{j}R & 0 \\ 0 & {}^{i}_{j}R \end{pmatrix} {}^{j}J
$$

اعمال نیروی خارجی — تبدیل به دستگاه صفر

$$
{}^{0}J_{e} = \begin{pmatrix} {}^{0}_{n}R & -{}^{0}_{n}R\,{}^{n}\hat{P}_{ne}\,{}^{0}_{n}R^{T} \\ 0 & {}^{0}_{n}R \end{pmatrix} {}^{n}J_{n}
$$



Wrist Point

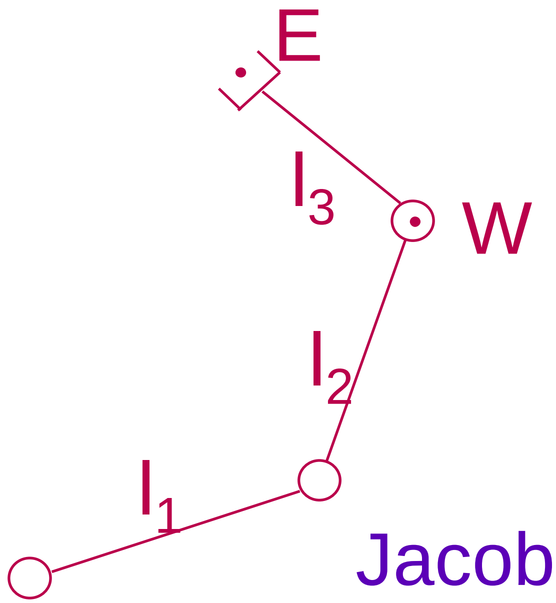
$$x = l_1 c_1 + l_2 c_{12}$$

$$y = l_1 s_1 + l_2 s_{12}$$

End-Effector Point

$$x = l_1 c_1 + l_2 c_{12} + l_3 c_{123}$$

$$y = l_1 s_1 + l_2 s_{12} + l_3 s_{123}$$



Wrist Point

$$x = l_1 c_1 + l_2 c_{12}$$

$$y = l_1 s_1 + l_2 s_{12}$$

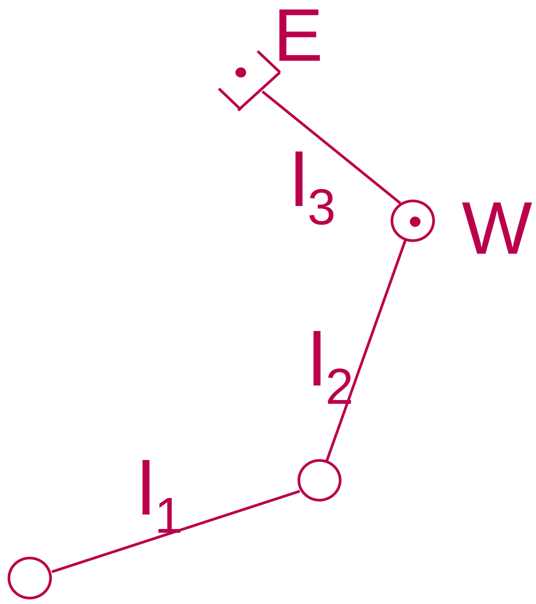
End-Effector Point

$$x = l_1 c_1 + l_2 c_{12} + l_3 c_{123}$$

$$y = l_1 s_1 + l_2 s_{12} + l_3 s_{123}$$

Jacobian (W)

$$J_W = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} & 0 \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \; ; \; {}^0 J_E = \begin{pmatrix} I & -{}^0\hat{P}_{WE} \\ 0 & I \end{pmatrix} {}^0 J_W$$



Wrist Point

$$x = l_1 c_1 + l_2 c_{12}$$

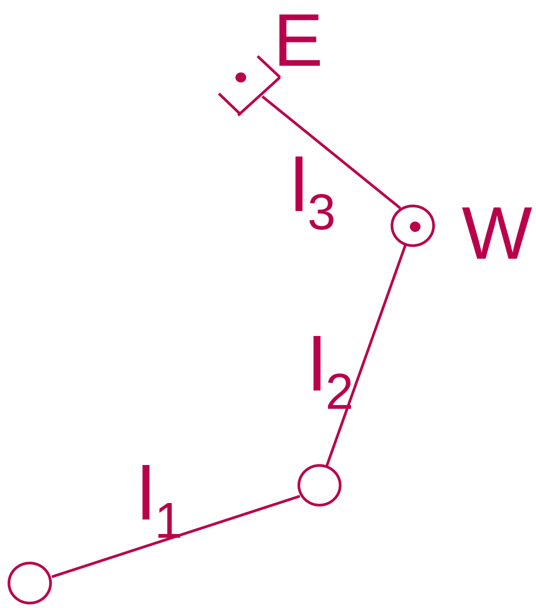
$$y = l_1 s_1 + l_2 s_{12}$$

End-Effector Point

$$x = l_1 c_1 + l_2 c_{12} + l_3 c_{123}$$

$$y = l_1 s_1 + l_2 s_{12} + l_3 s_{123}$$

$$J_W = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} & 0 \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad {}^0 J_E = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$



Wrist Point

$$x = l_1 c_1 + l_2 c_{12}$$

$$y = l_1 s_1 + l_2 s_{12}$$

End-Effector Point

$$x = l_1 c_1 + l_2 c_{12} + l_3 c_{123}$$

$$y = l_1 s_1 + l_2 s_{12} + l_3 s_{123}$$

$$J_W = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} & 0 \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

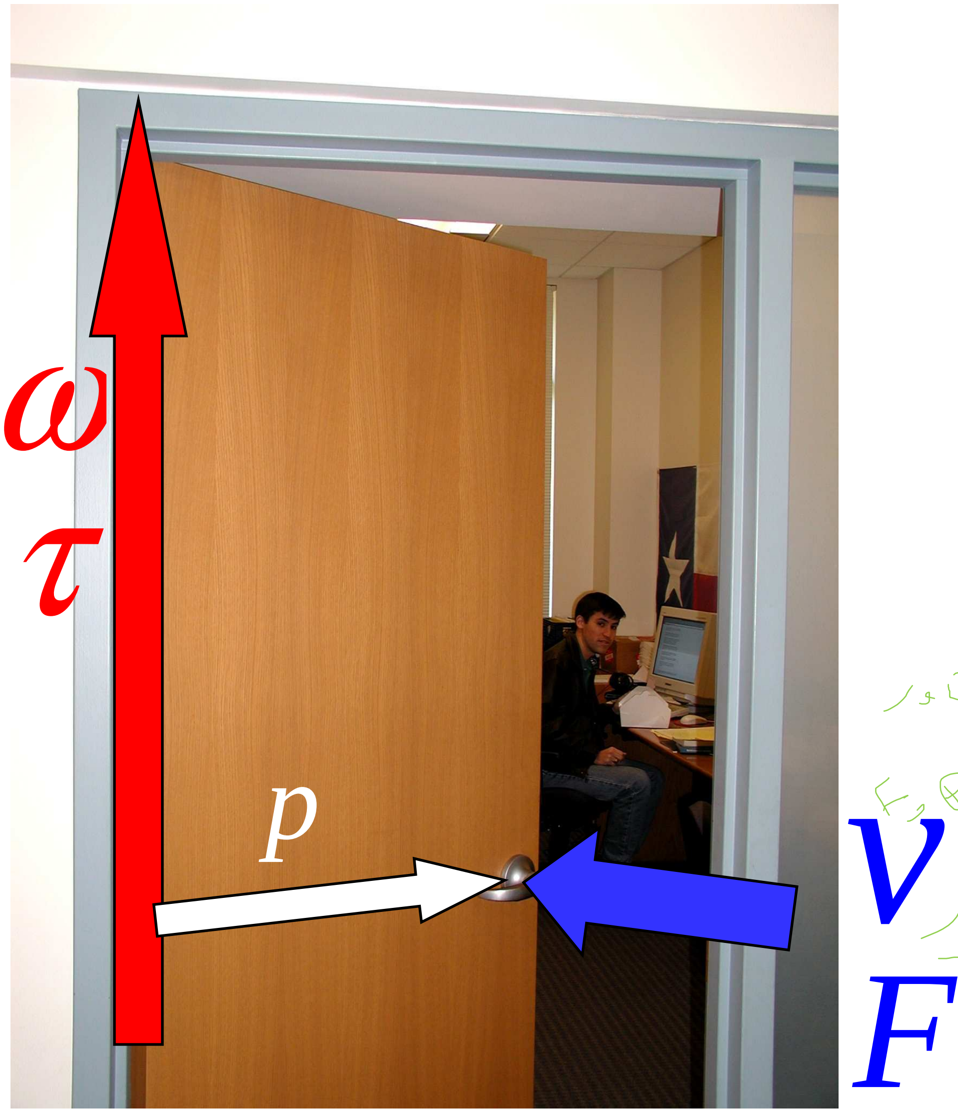
$$^0J_E = \begin{pmatrix} I & -{}^0\hat{P}_{WE} \\ 0 & I \end{pmatrix} {}^0J_W$$

$$^0P_{WE} = \begin{bmatrix} l_3 c_{123} \\ l_3 s_{123} \\ 0 \end{bmatrix} \Rightarrow {}^0\hat{P}_{WE} = \begin{pmatrix} 0 & 0 & l_3 s_{123} \\ 0 & 0 & -l_3 c_{123} \\ -l_3 s_{123} & l_3 c_{123} & 0 \end{pmatrix}$$

Jacobian

- Differential Motion
- Linear & Angular Motion
- Velocity Propagation
- Explicit Form
- Static Forces

Angular/Linear – Velocities/Forces



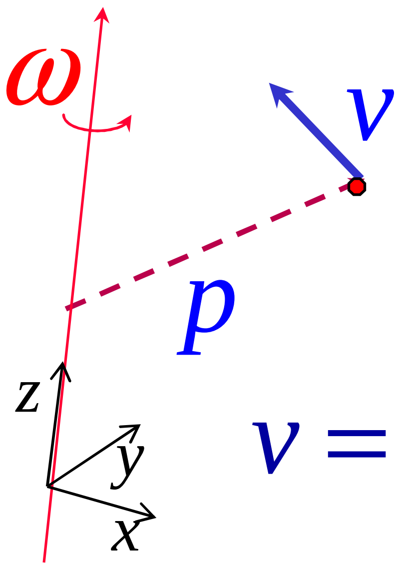
post

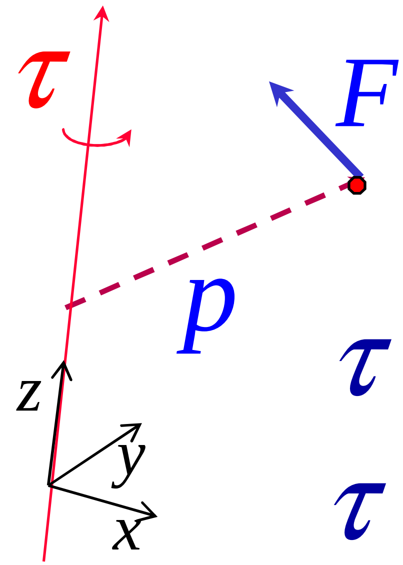
$$v = \omega \times p$$

pre

$$\tau = p \times F$$

Angular/Linear – Velocities/Forces





$$v = \omega \times p$$

$$v = -\hat{p}\,\omega$$

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} -p_y \\ p_x \end{pmatrix} \dot{\theta}$$

$$\boxed{v = J\,\dot{\theta}}$$

$$\tau = p \times F$$

$$\tau = \hat{p}\,F$$

$$\tau = (-\hat{p})^T\,F$$

$$\tau = \begin{pmatrix} -p_y & p_x \end{pmatrix} \begin{pmatrix} F_x \\ F_y \end{pmatrix}$$

$$\boxed{\tau = J^T F}$$

Velocity/Force Duality

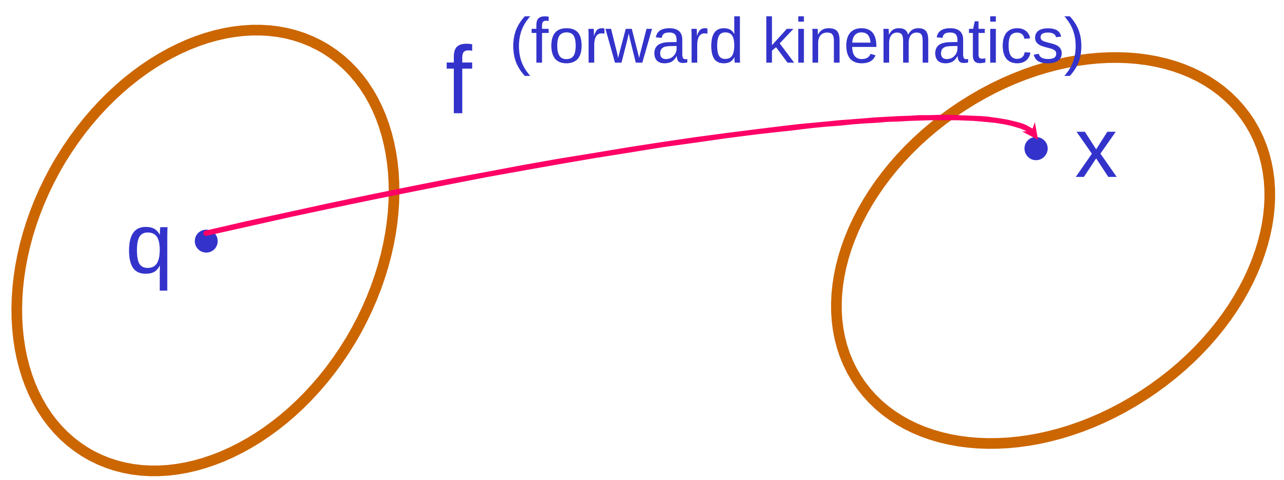
$$\boxed{\dot{x} = J\,\dot{\theta}}$$

$$\boxed{\tau = J^T F}$$

Inverse Kinematics

Direct Kinematics

forward همواره جواب دارد ولی inverse
ممکنه جواب نداشته باشه



Joint Space
(dimensions n)

Task Space
(dimensions m)

[illegible]
نیاز به اطلاعات در راستای ابعاد
نیز دارد.

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ . \\ . \\ . \\ q_n \end{bmatrix}$$

مفصل

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_m \end{bmatrix}$$

$$\mathbf{x} = f(\mathbf{q})$$

Joint Coordinates

Revolute Joints θ_i

Prismatic Joints d_i

$$q_i = \bar{\varepsilon}_i \theta_i + \varepsilon_i d_i$$

$$\varepsilon_i = \begin{cases} 0 & \text{revolute joint} \\ 1 & \text{prismatic joint} \end{cases}$$

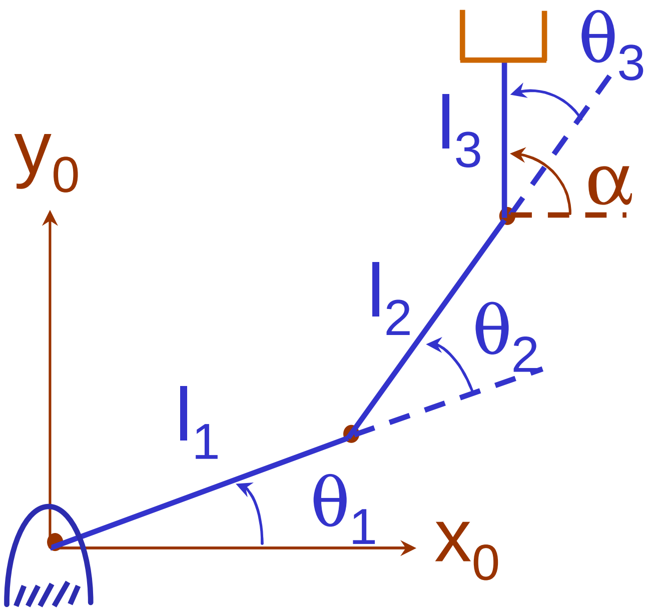
$$\bar{\varepsilon}_i \equiv 1 - \varepsilon_i$$

<u>Direct Kinematics</u>

Given $\mathbf{q} = (q_1 \quad q_2 \quad \ldots \quad q_n)^T$

$${}^0_nT = {}^0_nT(\mathbf{q}) \text{ or } \mathbf{x} = f(\mathbf{q}) \quad \text{(Geometric Model)}$$

Inverse Kinematics



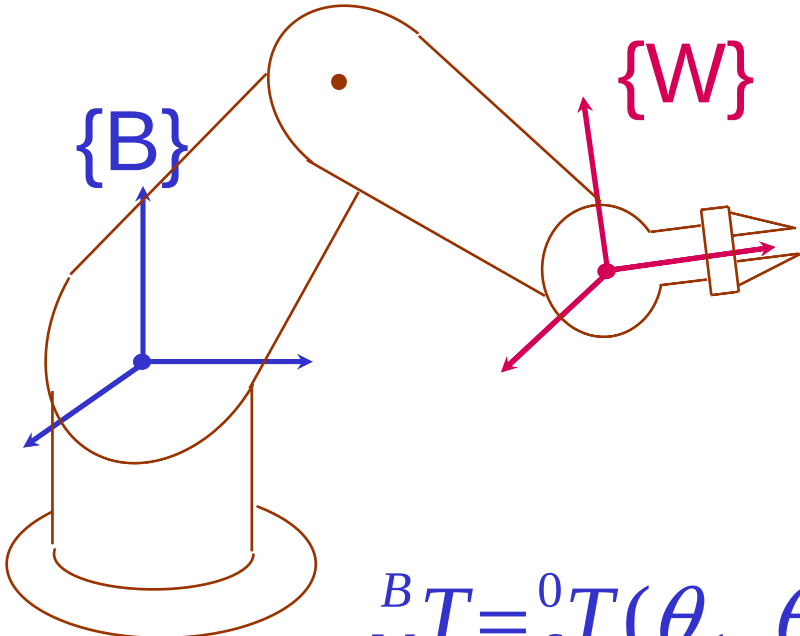
$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \alpha \end{bmatrix} = f(\mathbf{q})$$

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \alpha \end{bmatrix} = \begin{bmatrix} l_1 c_1 + l_2 c_{12} \\ l_1 s_1 + l_2 s_{12} \\ \theta_1 + \theta_2 + \theta_3 \end{bmatrix}$$

$$\cos(\theta_1 + \theta_2) = c_{12}$$

Given **q** $\longrightarrow$ a unique **x**

Inverse Kinematics



$$ {}^{B}_{W}T = {}^{0}_{6}T(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) $$

or $$ X = \begin{bmatrix} X_P \\ X_R \end{bmatrix} = f(\Theta) $$

Inverse Problem

Given $({}^{B}_{W}T$ or X) find Θ

Inverse Kinematics

Finding

$$\Theta = f^{-1}(X)$$

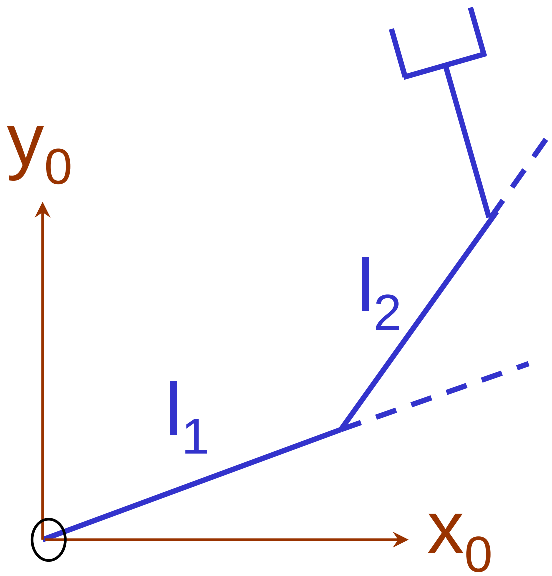
or

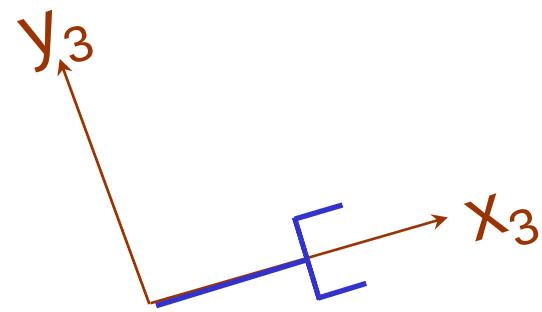
Solving

$${}^{0}_{6}T(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = {}^{B}_{W}T$$

$$\left(\begin{array}{c} \text{12 equations} \\ \text{6 unknowns} \end{array} \right)$$

Existence of Solutions



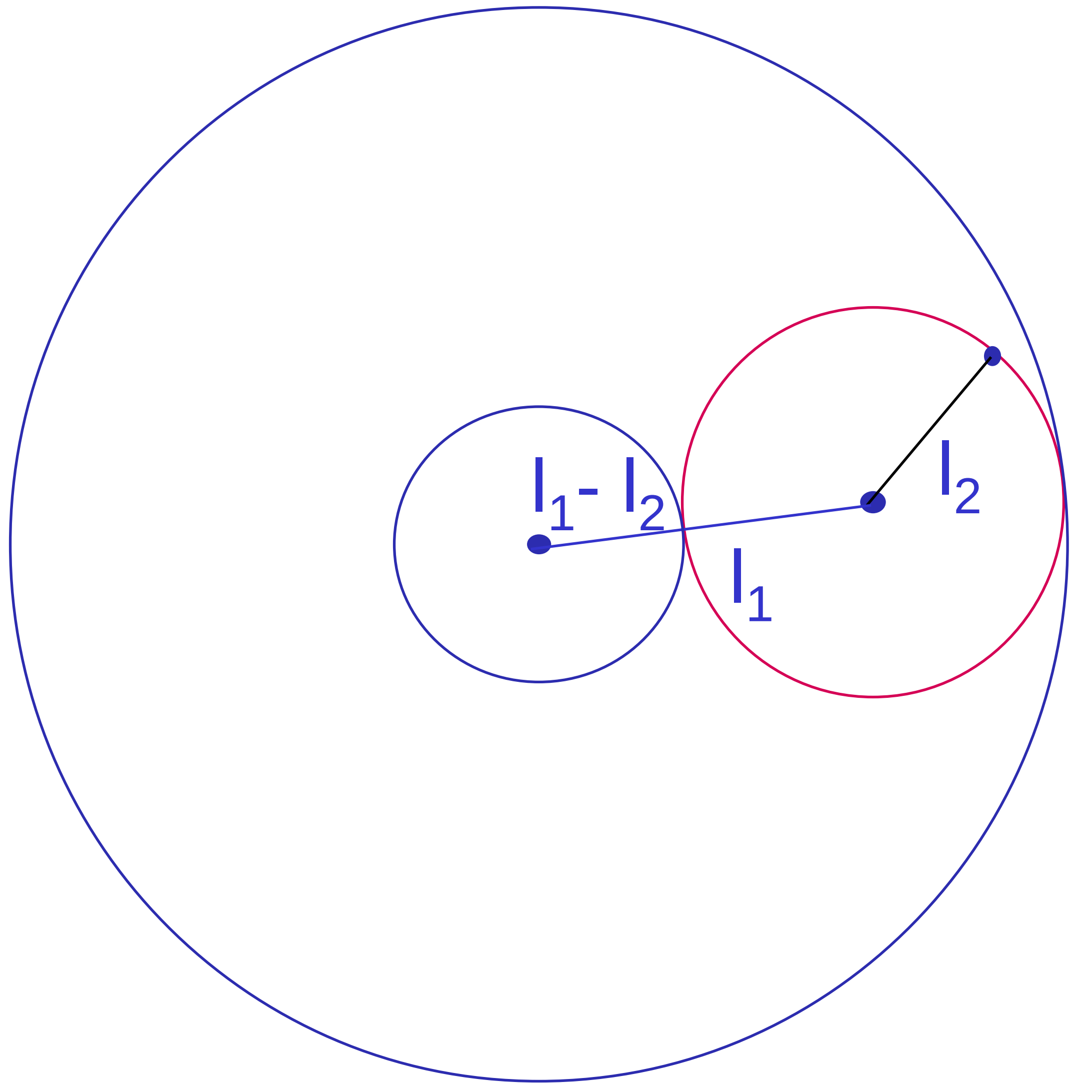


$(x_0, y_0);\ \alpha_0$

$$
{}^0_3T = \begin{pmatrix} c_{123} & -s_{123} & 0 & l_1c_1 + l_2c_{12} \\ s_{123} & c_{123} & 0 & l_1s_1 + l_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c\alpha_0 & -s\alpha_0 & 0 & x_0 \\ s\alpha_0 & c\alpha_0 & 0 & y_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
$$

solution if

$$(l_1 - l_2)^2 \le x_0^2 + y_0^2 \le (l_1 + l_2)^2$$

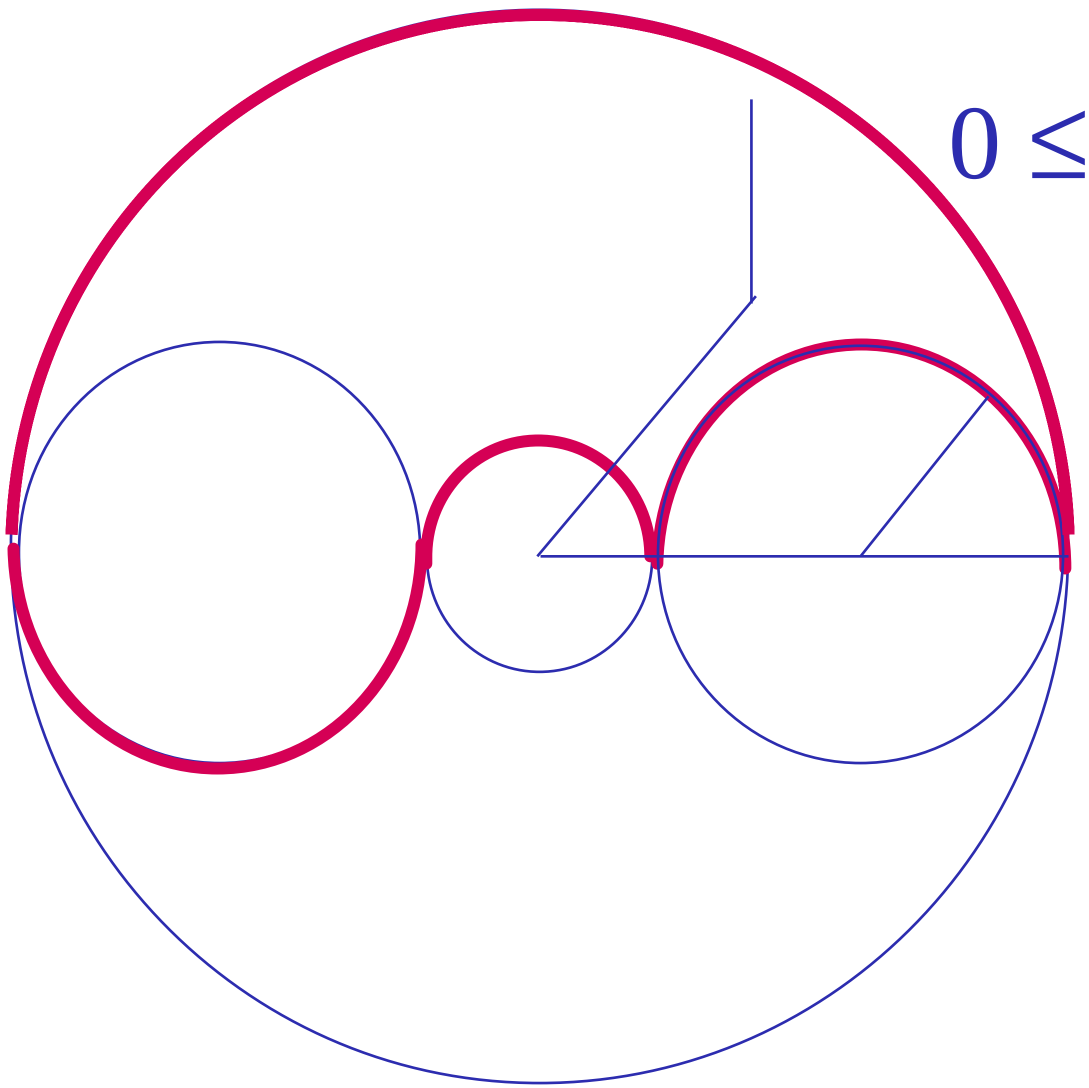


$l_1 - l_2$
l_2
l_1

Joint Limits

$$0 \leq \theta_1 \leq 180^o$$

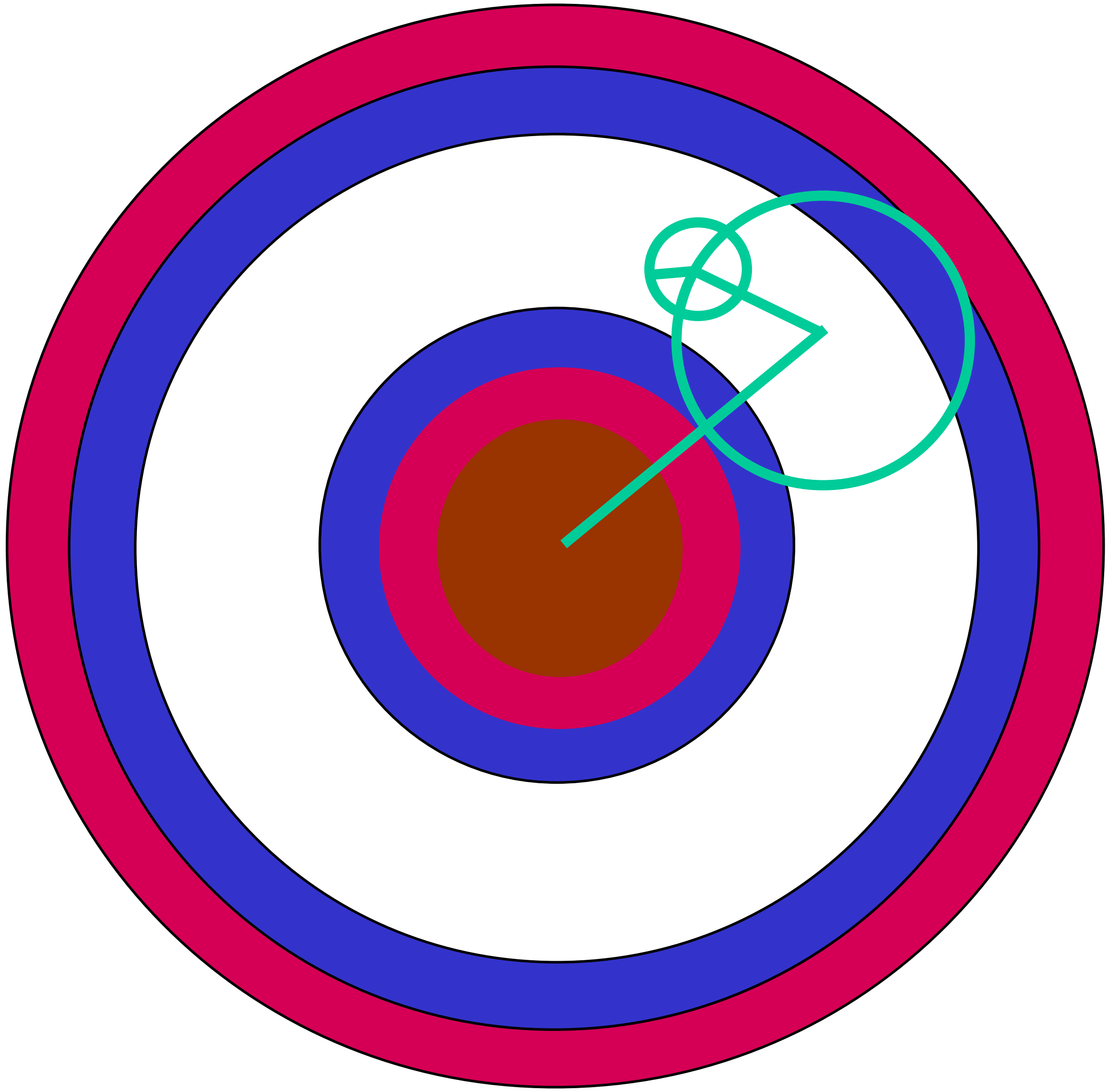
$$0 \leq \theta_2 \leq 180^o$$



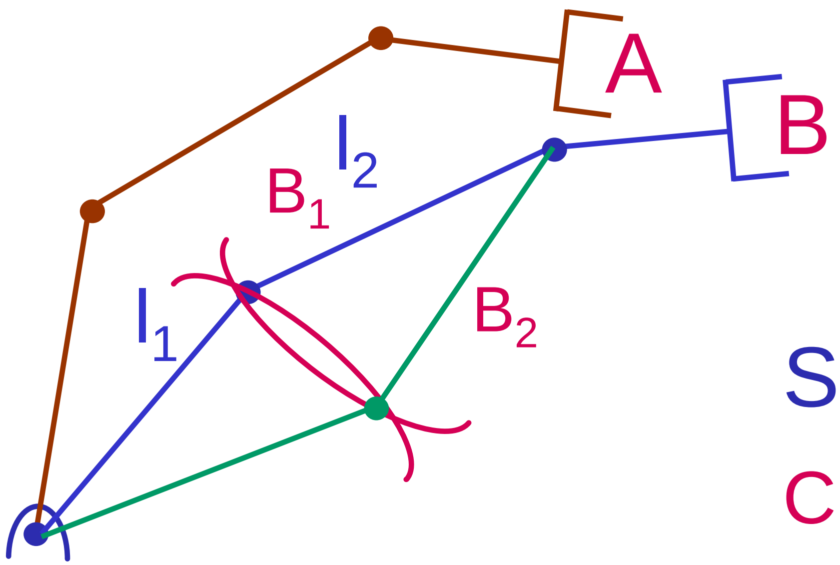
Workspace

- Reachable Workspace
- Dextrous Workspace

Dextrous Workspace



Multiplicity of Solutions



Selection of a solution

Criterion: Joint distance

$$C_1 = \left\| \Theta_{(B1)} - \Theta_{(A)} \right\|$$

$$C_2 = \left\| \Theta_{(B2)} - \Theta_{(A)} \right\|$$

Weighted Joint distance

moving smaller joints

Number of Solutions

It depends on

- Number of Joints
- Link Parameters

e.g. 6-revolute-joint manipulator

if all $a_i \neq 0$ Number solutions ≤ 16

if $a_1 = a_3 = a_5 = 0$ Number solutions ≤ 4

- Range of Motion

General Mechanism with 6 d.o.f.

Number of solutions ≤ 16

Main Results

General 6R open-chain 16 solutions
General 5RP open-chain 16 solutions
General 4R2P open-chain 8 solutions
General 3R3P open-chain 2 solutions

Special conditions in the structure [such as intersecting or parallel axes] cause the general number of solutions to reduce. There exist open-chain manipulators with 16, 14, 12, 10, 8, 6, 4, 2 solutions.

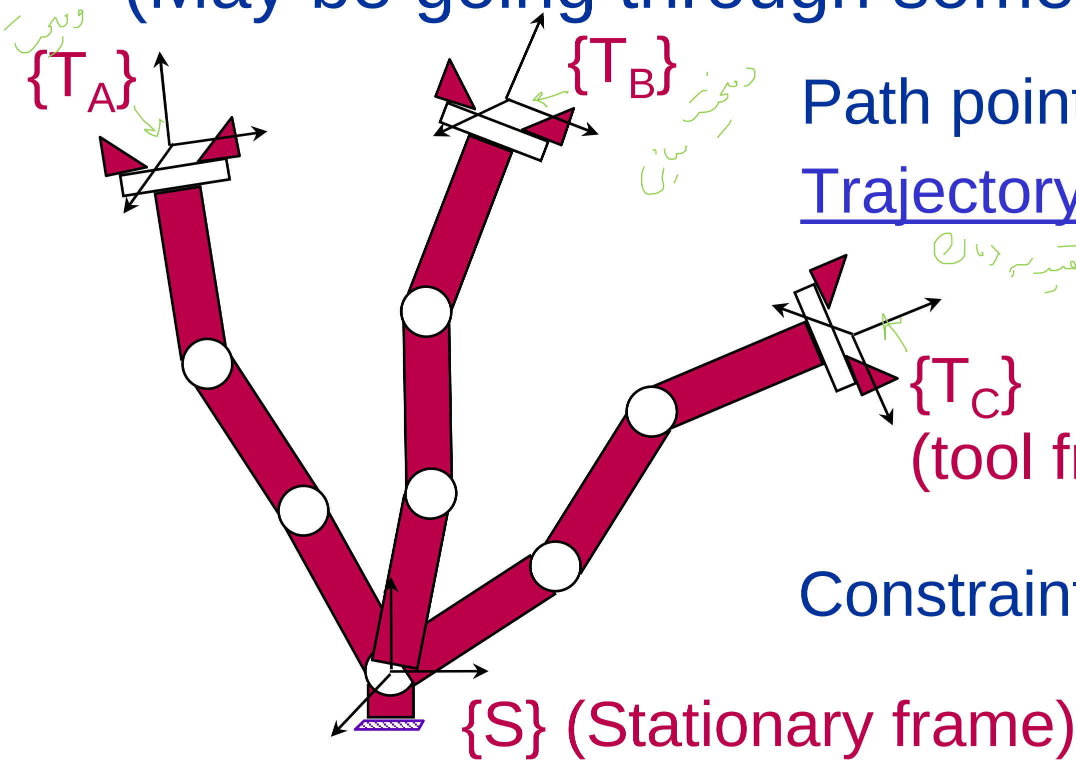
Trajectory Generation

Trajectory Generation

Basic Problem:

Move the manipulator arm from some initial position $\{T_A\}$ to some desired final position $\{T_C\}$.

(May be going through some via point $\{T_B\}$)



Path points : Initial, final and via points

Trajectory : Time history of position, velocity and acceleration for each DOF

Constraints: Spatial, time, smoothness

Solution Spaces :

Joint space

- Easy to go through via points
  (Solve inverse kinematics at all path points and plan)
- No problems with singularities
- Less calculations
- Can not follow straight line

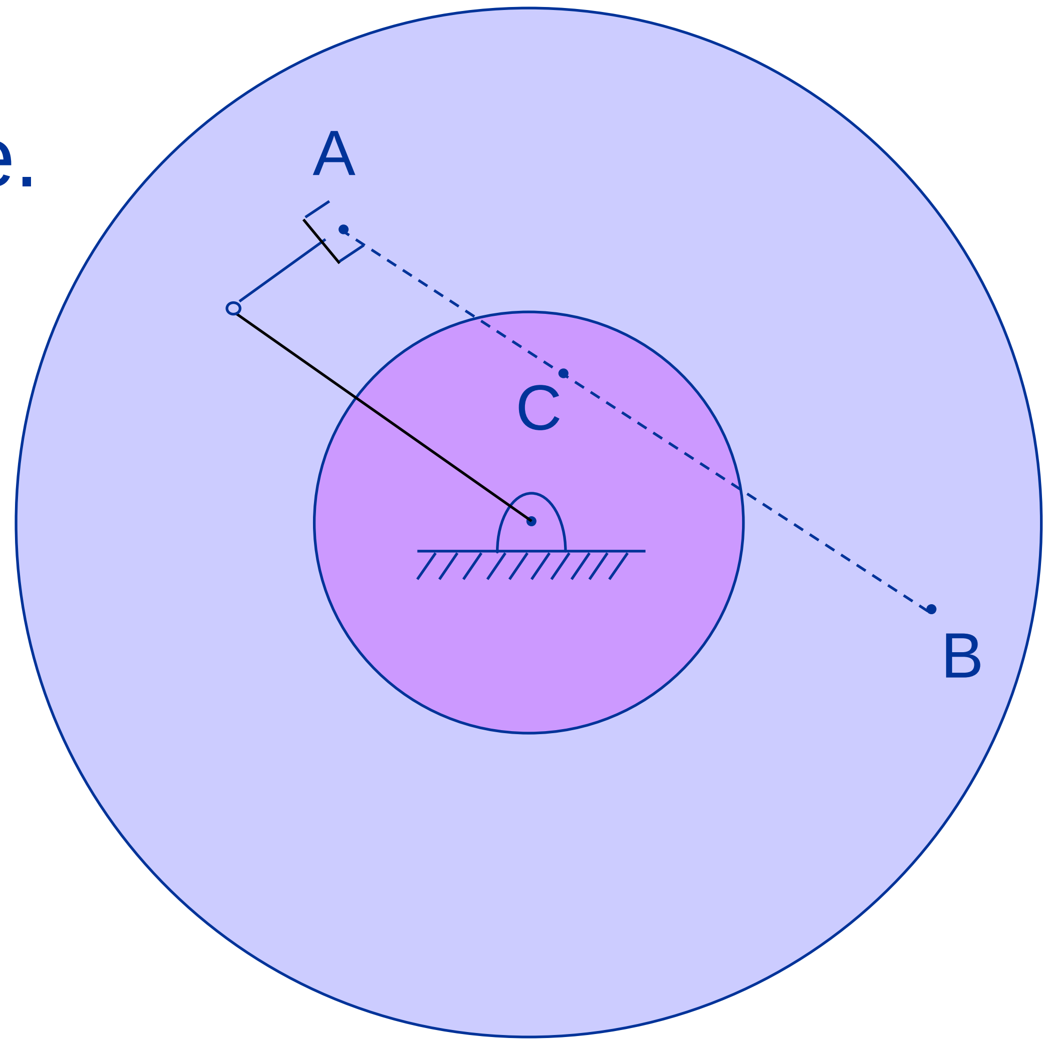
Cartesian space

- We can track a shape
  (for orientation : equivalent axes, Euler angles,…
- More expensive at run time
  (after the path is calculated need joint angles
  in a lot of points)
- Discontinuity problems

Cartesian planning difficulties :

Initial and Goal Points are reachable.

Intermediate points (C) unreachable.

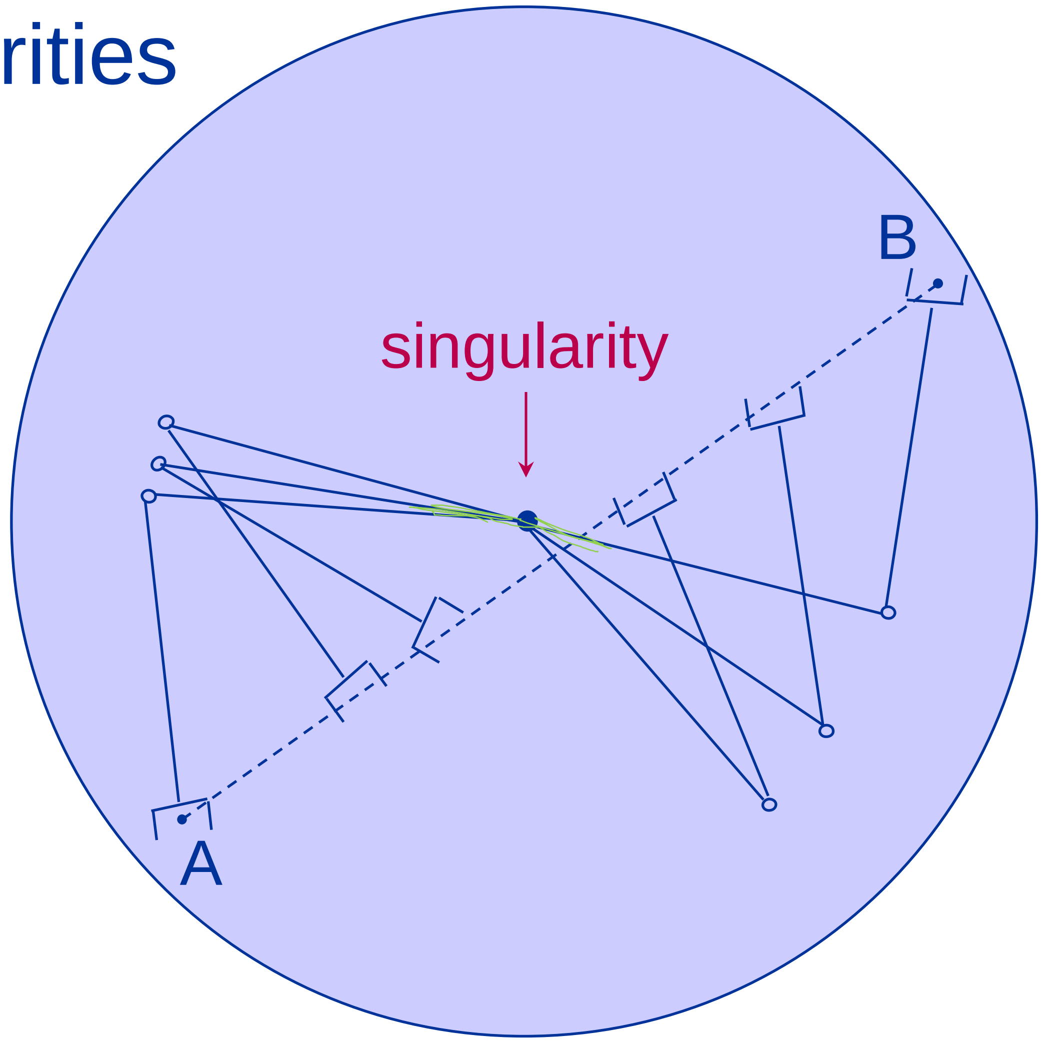


Cartesian planning difficulties :

2

Approaching singularities
some joint velocities
go to ∞
causing deviation
from the path

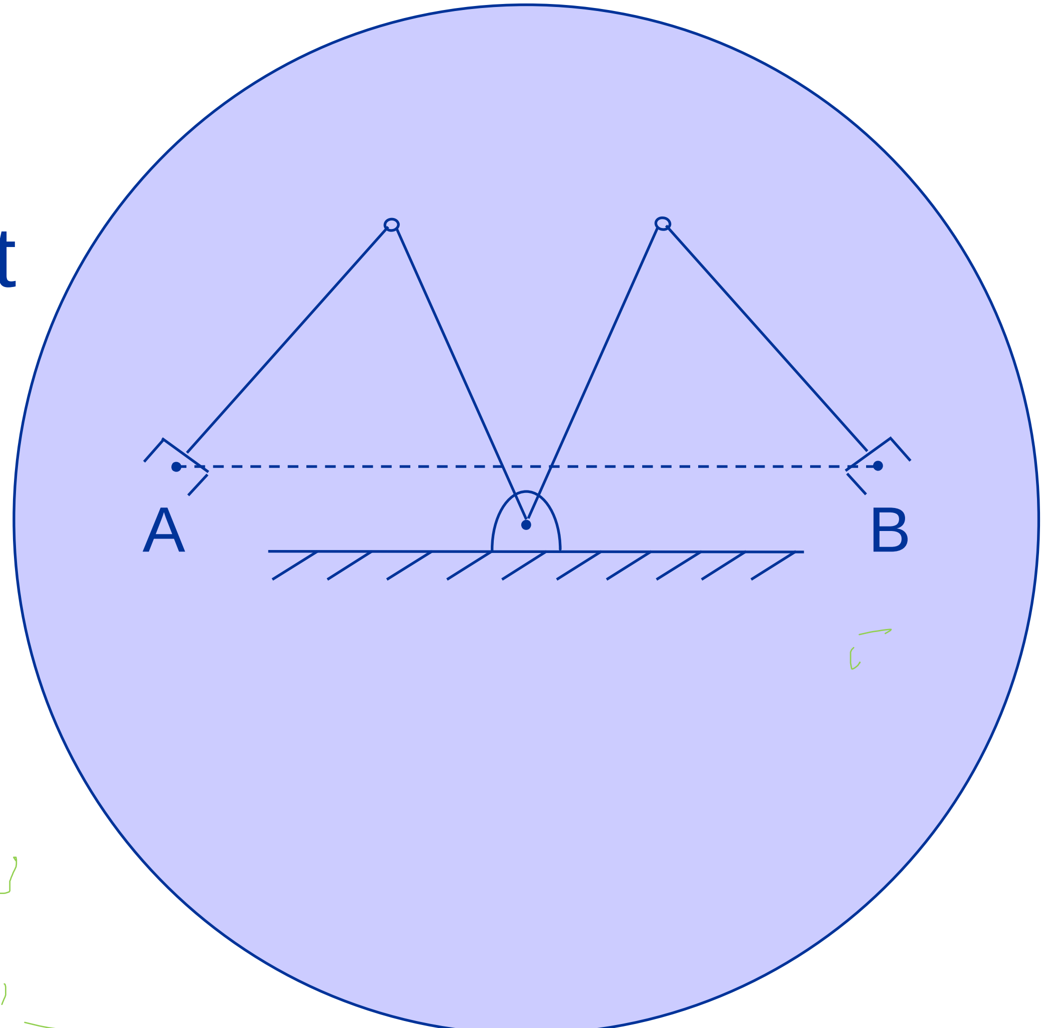
وسط راه به singularity برمی‌خوریم



Cartesian planning difficulties :

3)

Start point (A) and goal point (B) are reachable in different joint space solutions (The middle points are reachable from below.)



[illegible]

Actual planning in <u>any</u> space:

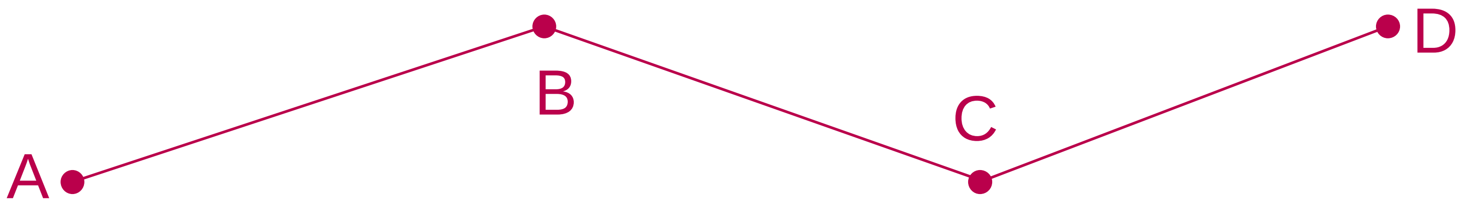
Assume one <u>generic</u> variable <u>u</u>

(can be x, y, z, orientation - α, β, γ)

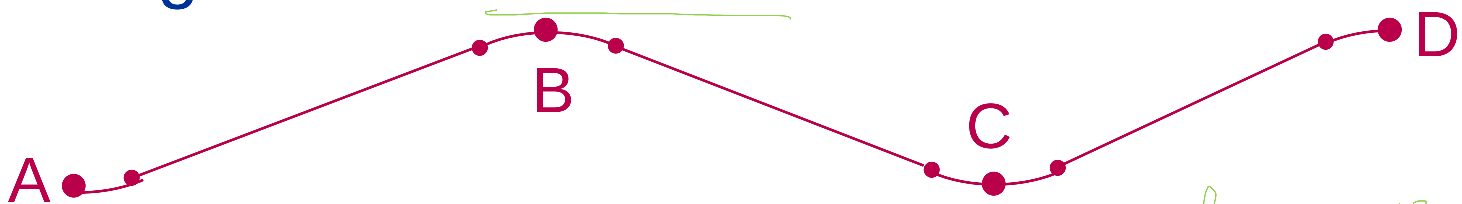
joint variables direction cosines

<u>Candidate curves</u> :

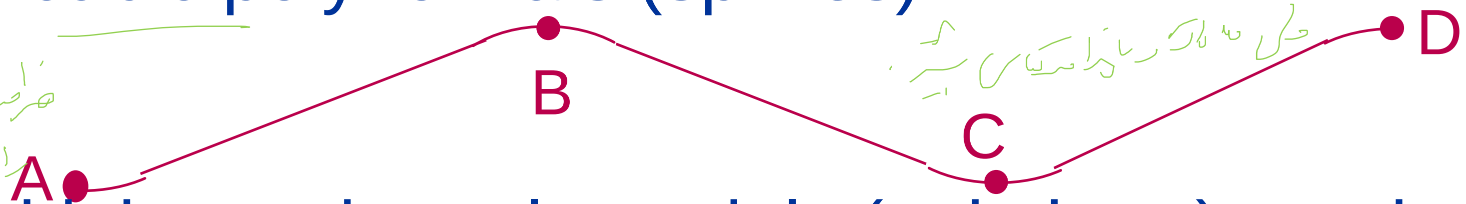
straight line (discontinuous velocity at path points)



straight line with blends

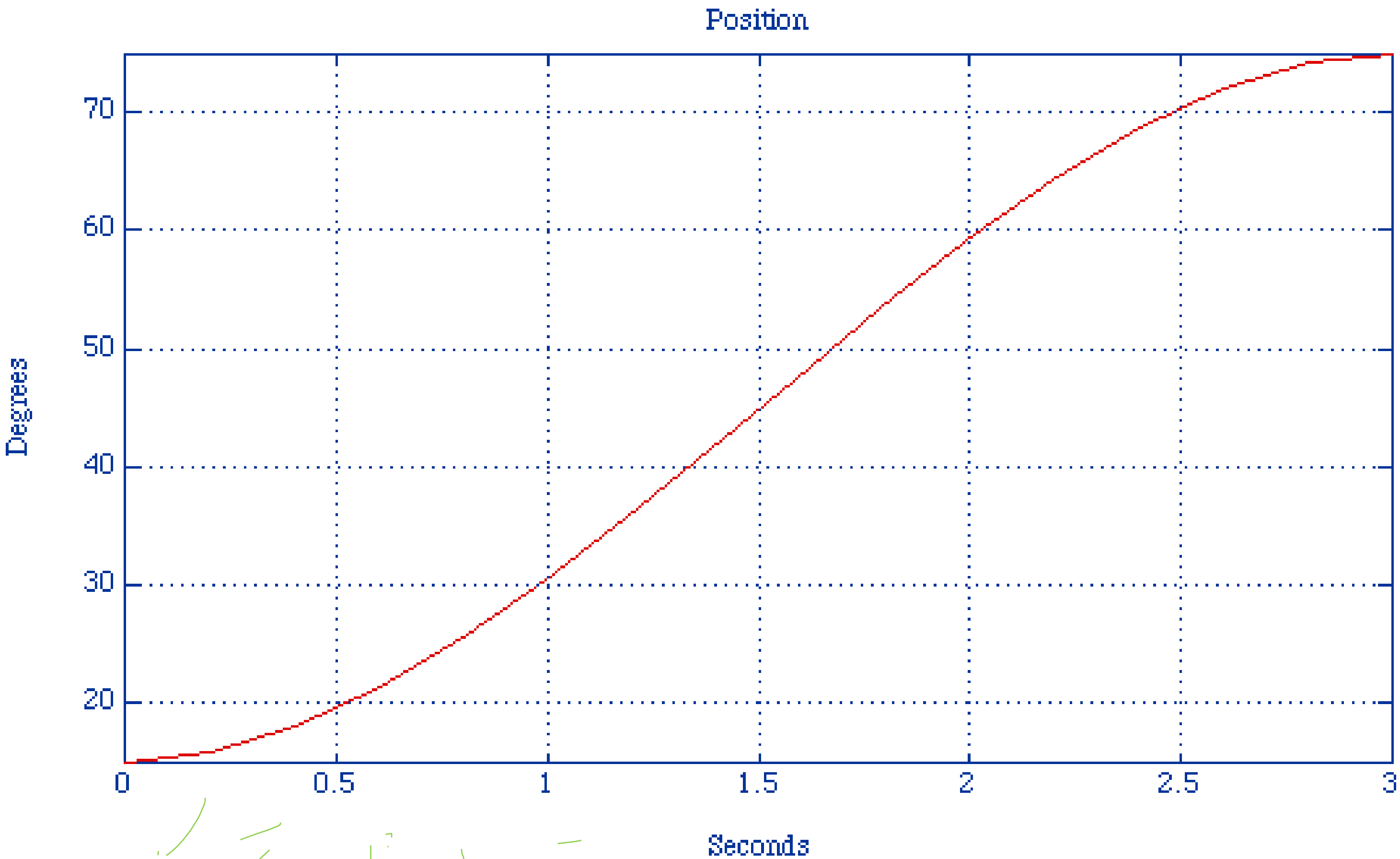


cubic polynomials (splines)



higher order polynomials (quintic,…) or other curves

Single Cubic Polynomial



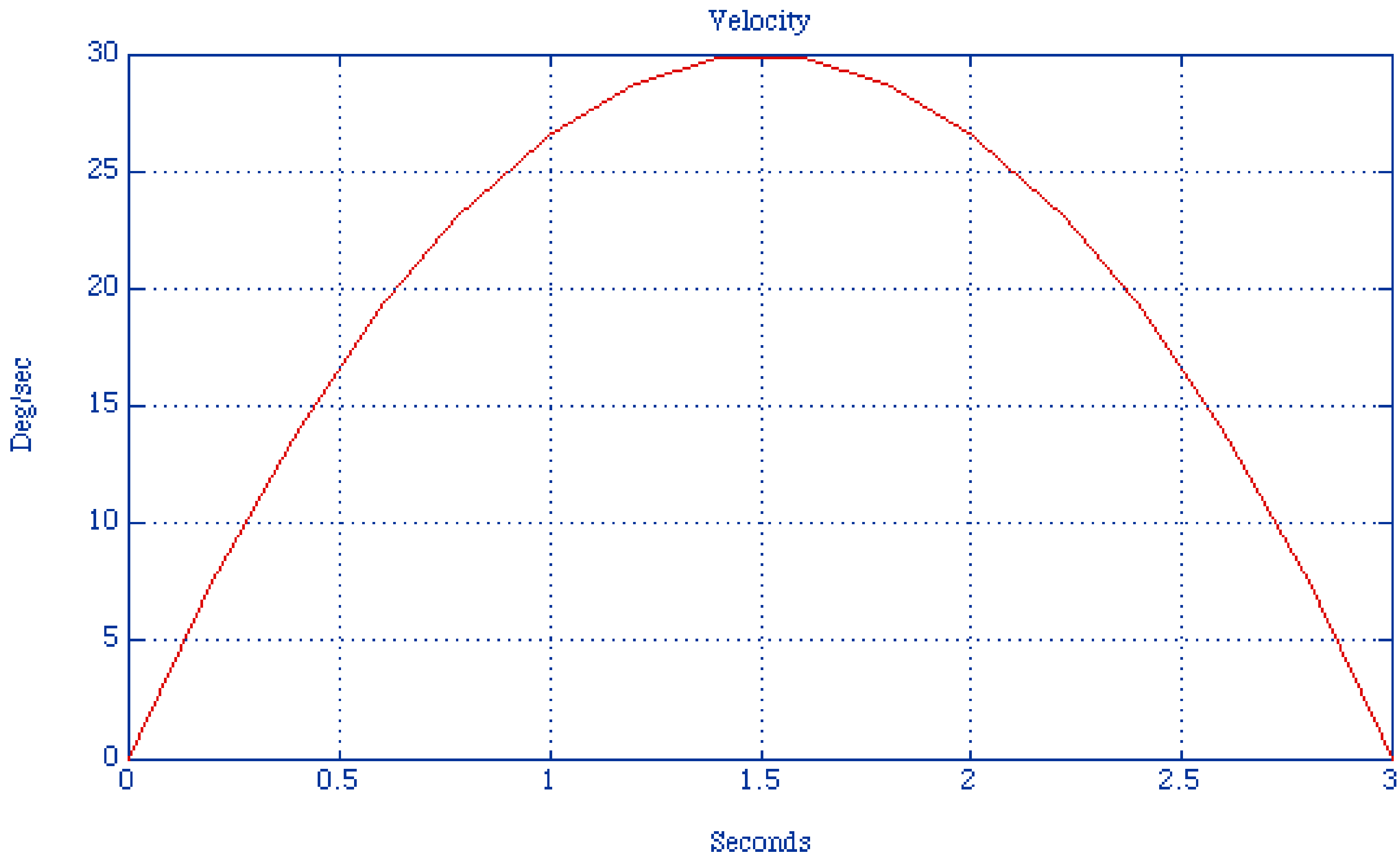
ضرایب را تحمیل کنیم

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

Initial Conditions:

$$\theta(0) = \theta_0 \; ; \quad \theta(t_f) = \theta_f$$

Single Cubic Polynomial



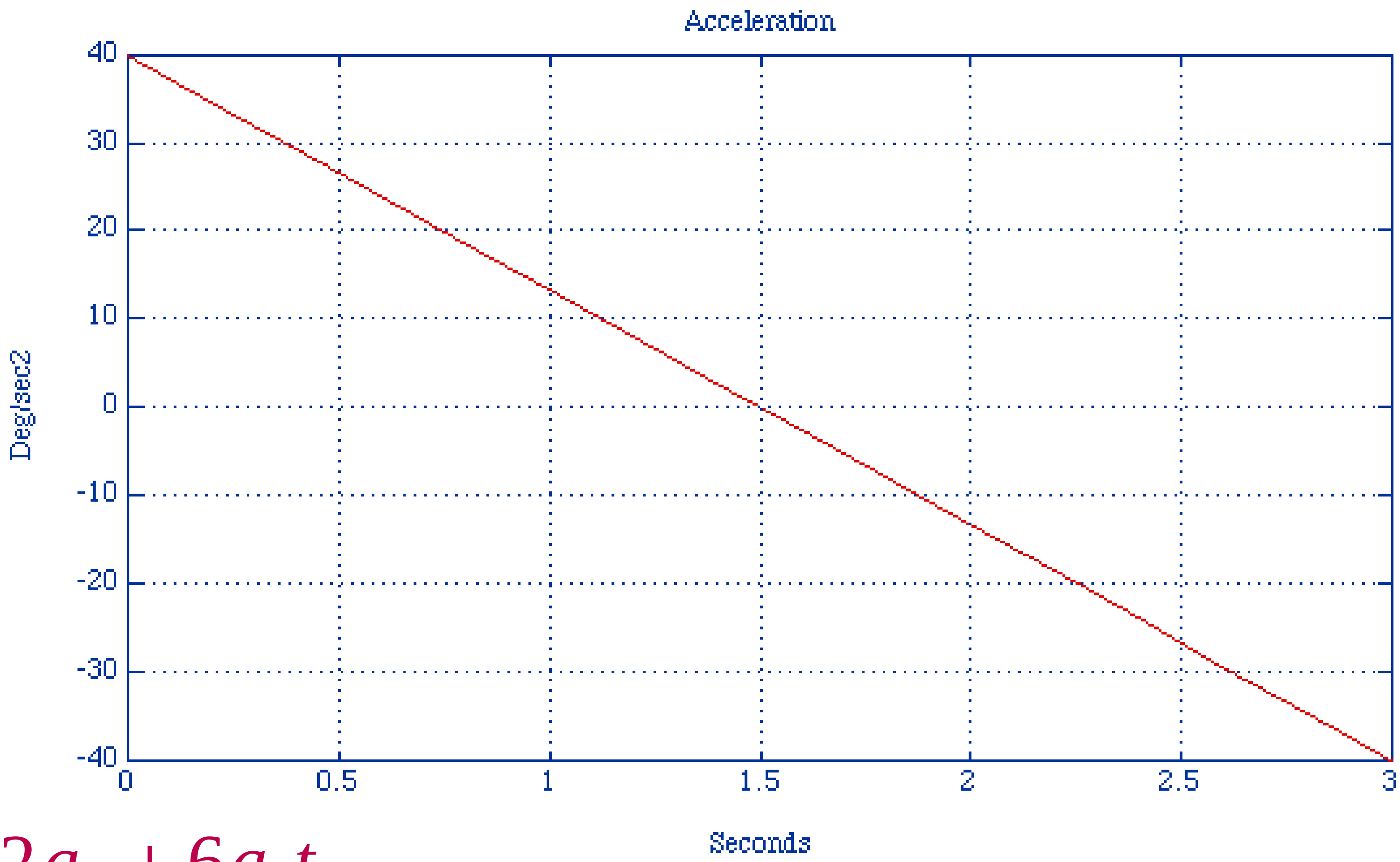
$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2$$

Initial Conditions:

$$\dot{\theta}(0) = 0 \;;\quad \dot{\theta}(t_f) = 0$$

Starts and ends at rest

Single Cubic Polynomial



$$\ddot{\theta}(t) = 2a_2 + 6a_3 t$$

$$\dddot{\theta}(t) = 6a_3 \text{ (constant)}$$

Solution : $\theta(t) = \theta_0 + \frac{3}{t_f^2}(\theta_f - \theta_0)t^2 + \left(-\frac{2}{t_f^3}\right)(\theta_f - \theta_0)t^3$

Cubic Polynomials with via points

- If we come to rest at each point
  use formula from previous slide
- For continuous motion (no stops)
  need velocities at intermediate points:

$$\dot{\theta}(0) = \dot{\theta}_0$$

$$\dot{\theta}(t_f) = \dot{\theta}_f$$

Initial Conditions

Solution : $a_0 = \theta_0$

$$a_1 = \dot{\theta}_0$$

$$a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) - \frac{2}{t_f}\dot{\theta}_0 - \frac{1}{t_f}\dot{\theta}_f$$

$$a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0) + \frac{1}{t_f^2}(\dot{\theta}_f + \dot{\theta}_0)$$

Run Time Path Generation

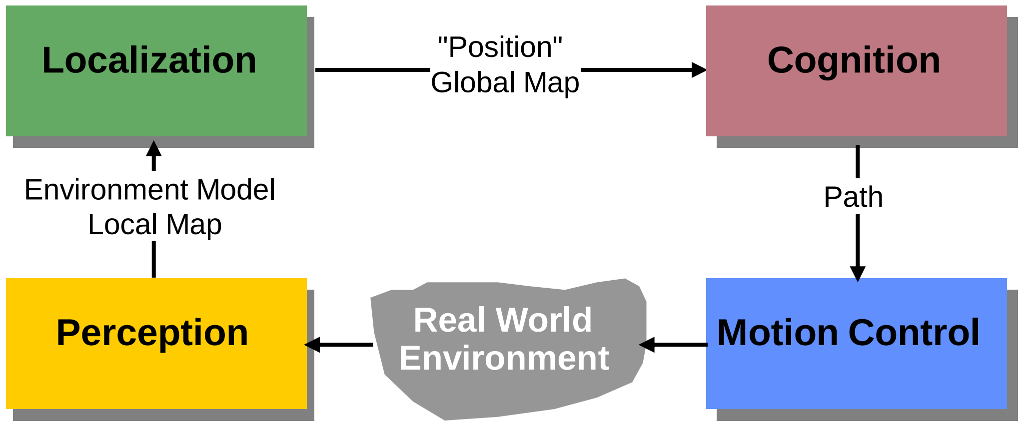
- trajectory in terms of $\Theta, \dot{\Theta}, \ddot{\Theta}$ fed to the control system
- Path generator computes at path update rate
- In joint space directly:
  - cubic splines -- change set of coefficients at the end of each segment
  - linear with parabolic blends -- check on each update if you are in linear or blend portion and use appropriate formulas for **u**
- In Cartesian space:
  - calculate Cartesian position and orientation at each update point using same formulas
  - convert into joint space using inverse Jacobian and derivatives

  or

  find equivalent frame representation and use inverse kinematics function to find $\Theta, \dot{\Theta}, \ddot{\Theta}$

Trajectory Planning with Obstacles

- Path planning for the whole manipulator
  - Local vs. Global Motion Planning
    - Gross motion planning for relatively uncluttered environments end effector
    - Fine motion planning for the end-effector frame
  - Configuration space (C-space) approach
- Planning for a point robot
  - graph representation of the free space, quadtree
  - Artificial Potential Field method
- Multiple robots, moving robots and/or obstacles



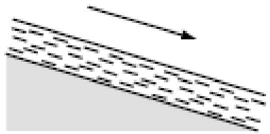
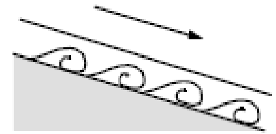
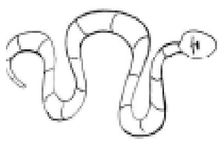
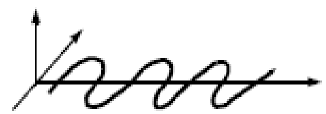
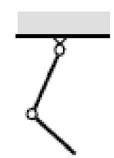
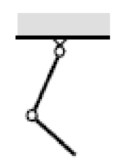
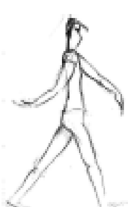
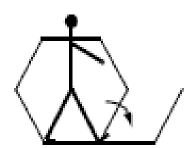
Locomotion Concepts

Concepts

Legged Locomotion

Wheeled Locomotion

2 Locomotion Concepts: Principles Found in Nature

Type of motion	Resistance to motion	Basic kinematics of motion
Flow in a Channel 	Hydrodynamic forces	Eddies 
Crawl 	Friction forces	 Longitudinal vibration
Sliding 	Friction forces	 Transverse vibration
Running 	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum 
Jumping 	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum 
Walking 	Gravitational forces	Rolling of a polygon (see figure 2.2) 

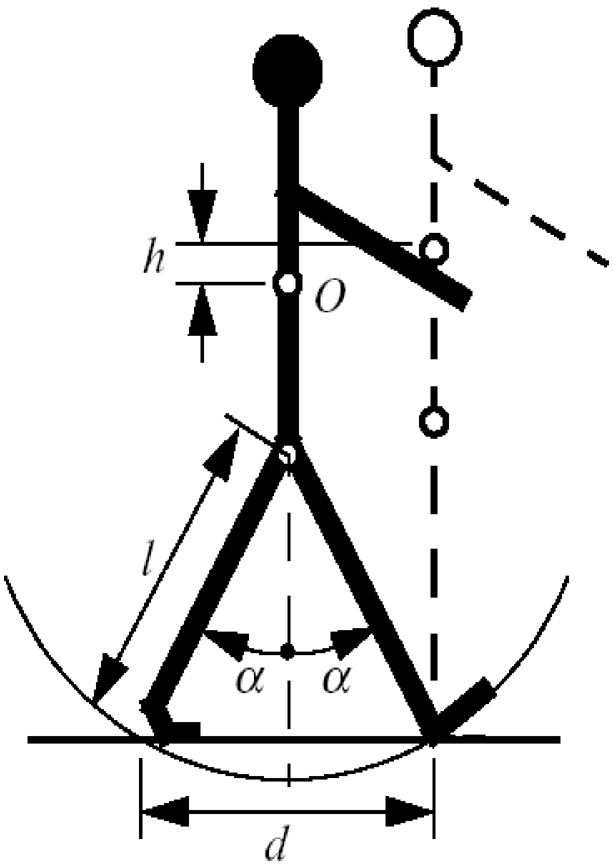
Locomotion Concepts

- Nature came up with a multitude of locomotion concepts
  - Adaptation to environmental characteristics
  - Adaptation to the perceived environment (e.g. size)

- Concepts found in nature
  - Difficult to imitate technically
  - Do not employ wheels
  - Sometimes imitate wheels (bipedal walking)

- Most technical systems today use wheels or caterpillars
  - Legged locomotion is still mostly a research topic

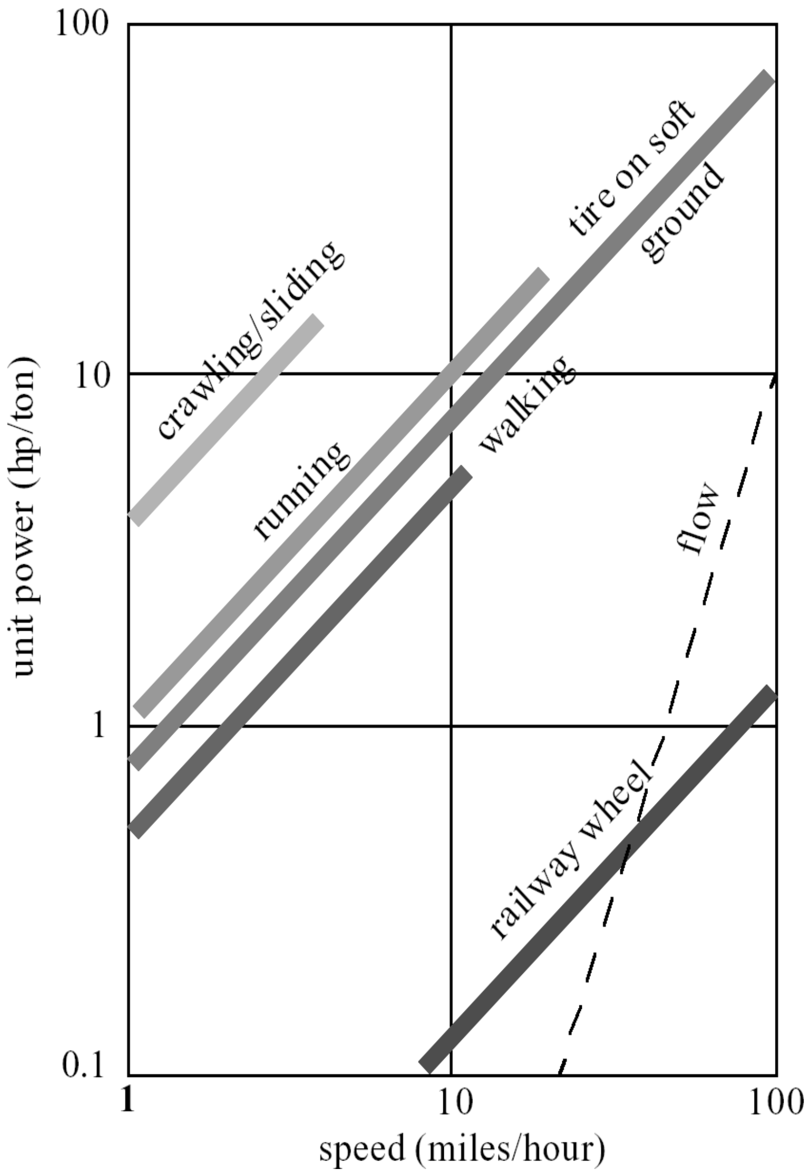
Biped Walking



- Biped walking mechanism
  - not too far from real rolling
  - rolling of a polygon with side length equal to the length of the step
  - the smaller the step gets, the more the polygon tends to a circle (wheel)
- But…
  - rotating joint was not invented by nature
  - Work against gravity is required
  - More detailled analysis follows later in this presentation

Walking or rolling?

- number of actuators
- structural complexity
- control expense
- energy efficient
  - terrain (flat ground, soft ground, climbing..)
- movement of the involved masses
  - walking / running includes up and down movement of COG
  - some extra losses



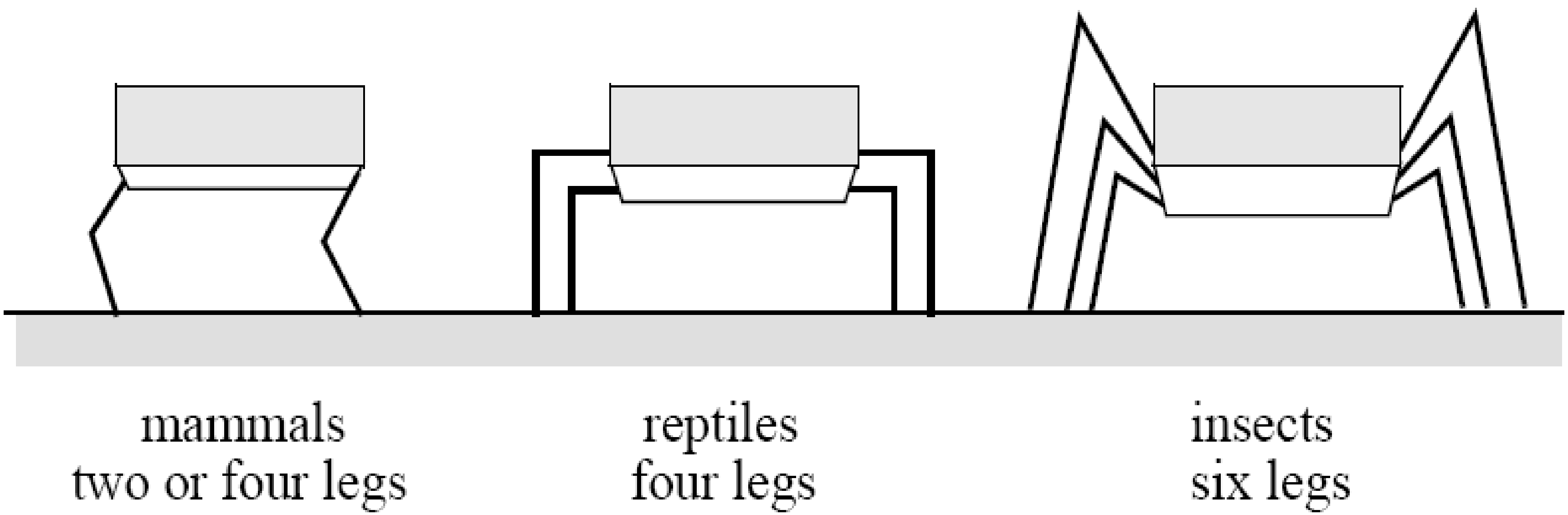
Characterization of locomotion concept

- Locomotion
  - physical interaction between the vehicle and its environment.
- Locomotion is concerned with **interaction forces**, and the **mechanisms** and **actuators** that generate them.

- The most important issues in locomotion are:
  - **stability**
    - number of contact points
    - center of gravity
    - static/dynamic stabilization
    - inclination of terrain
  - **characteristics of contact**
    - contact point or contact area
    - angle of contact
    - friction
  - **type of environment**
    - structure
    - medium (water, air, soft or hard ground)

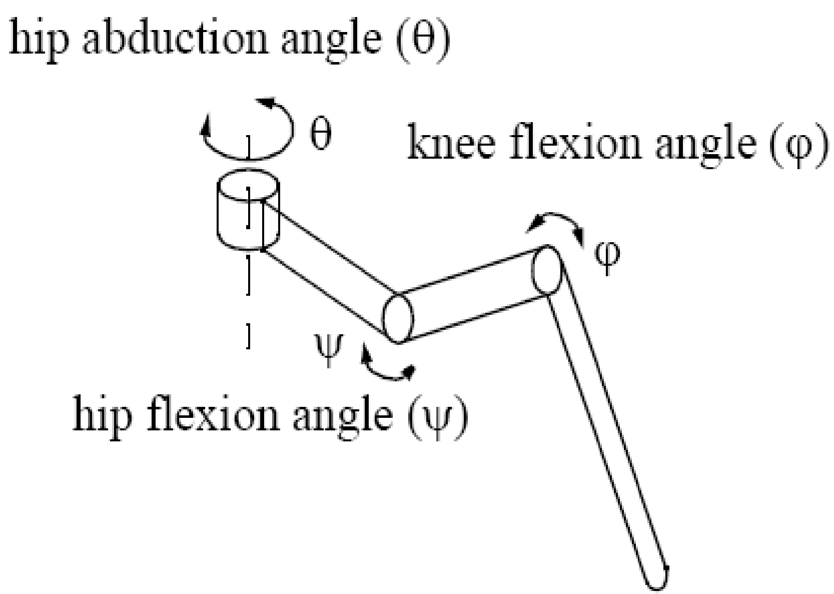
Mobile Robots with legs (walking machines)

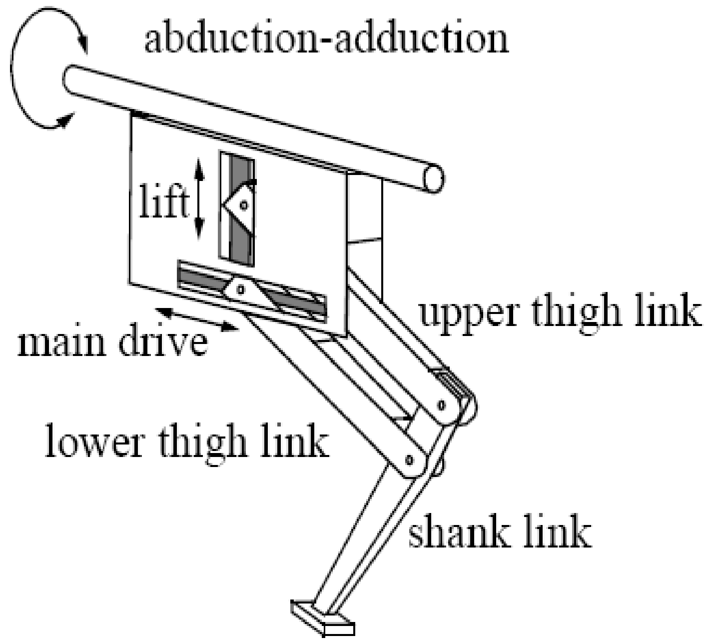
- The fewer legs the more complicated becomes locomotion
  - Stability with point contact- at least three legs are required for static stability
  - Stability with surface contact – at least one leg is required
- During walking some (usually half) of the legs are lifted
  - thus loosing stability?
- For static walking at least 4 (or 6) legs are required
  - Animals usually move two legs at a time
  - Humans require more than a year to stand and then walk on two legs.



Number of Joints of Each Leg (DOF: degrees of freedom)

- A minimum of two DOF is required to move a leg forward
  - a *lift* and a *swing* motion.
  - Sliding-free motion in more than one direction not possible
- Three DOF for each leg in most cases (as pictured below)
- 4th DOF for the ankle joint
  - might improve walking and stability
  - additional joint (DOF) increases the complexity of the design and especially of the locomotion control.





The number of distinct event sequences (gaits)

- The gait is characterized as the distinct sequence of **lift and release events** of the individual legs
  - it depends on the number of legs.
  - the number of possible events N for a walking machine with k legs is:

$$N = (2k-1)!$$

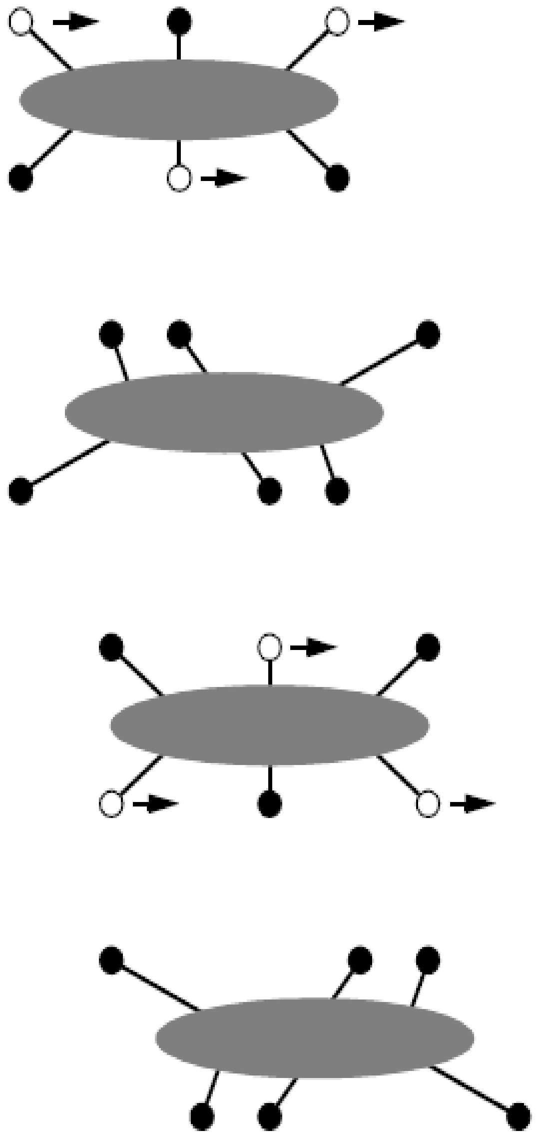
- For a biped walker (k=2) the number of possible events N is:

$$N = (2k-1)! = 3! = 3 \cdot 2 \cdot 1 = 6$$

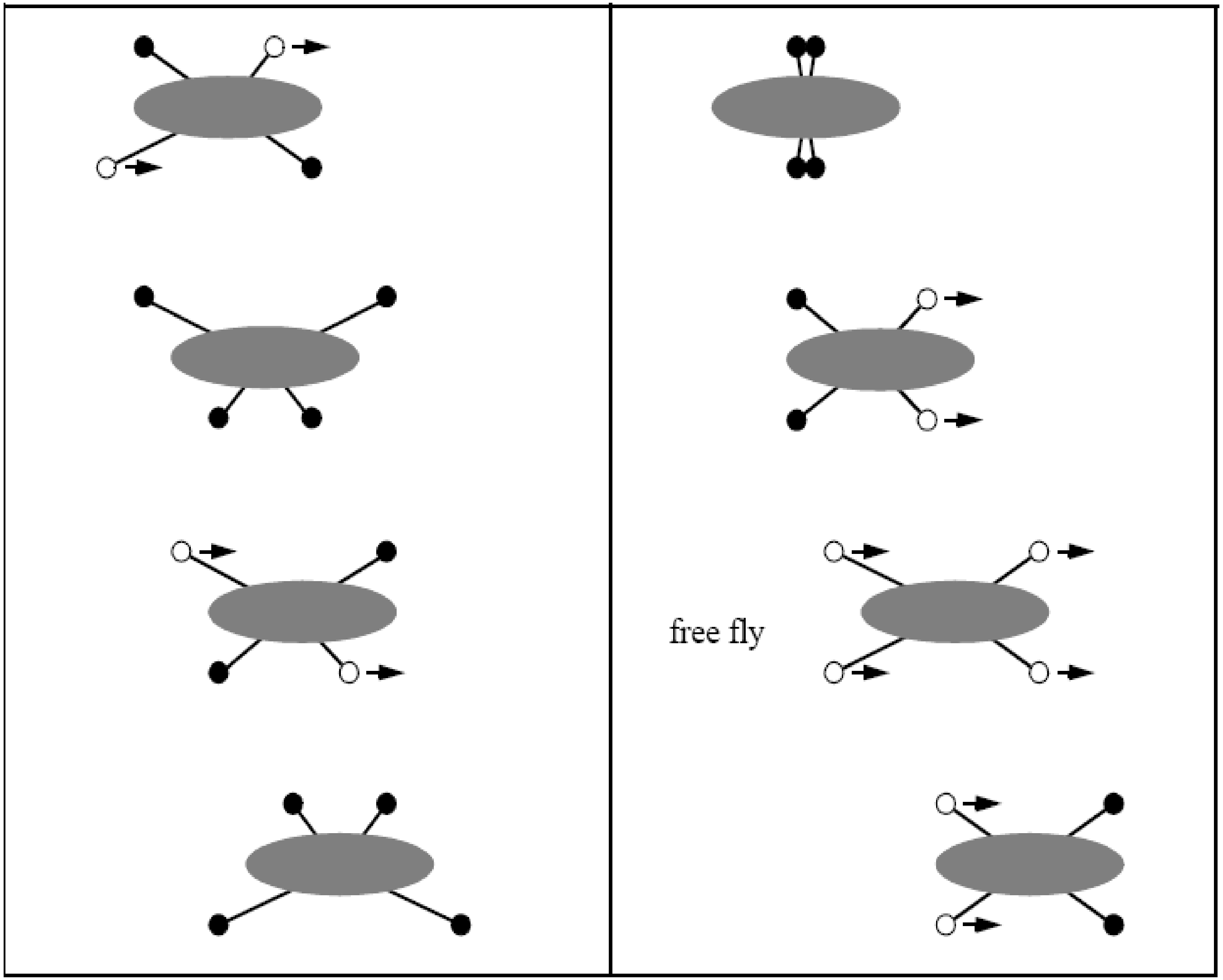
- For a robot with 6 legs (hexapod) N is already

$$N = 11! = 39'916'800$$

Most Obvious Gait with 6 Legs is Static



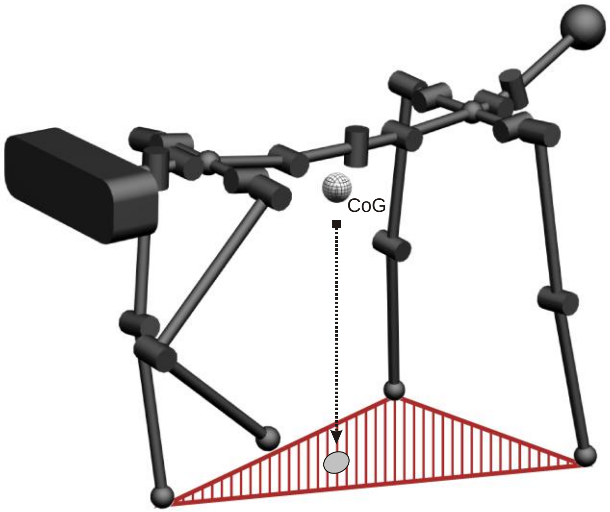
Most Obvious Natural Gaits with 4 Legs are Dynamic

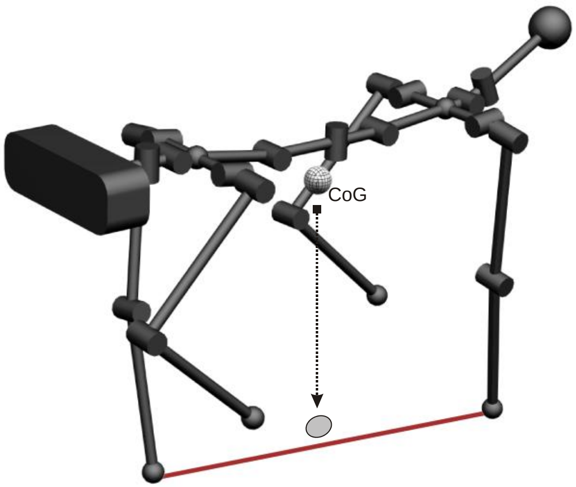


Changeover Walking Galloping

Dynamic Walking vs. Static Walking

- **Statically stable**
- **Dynamic walking**



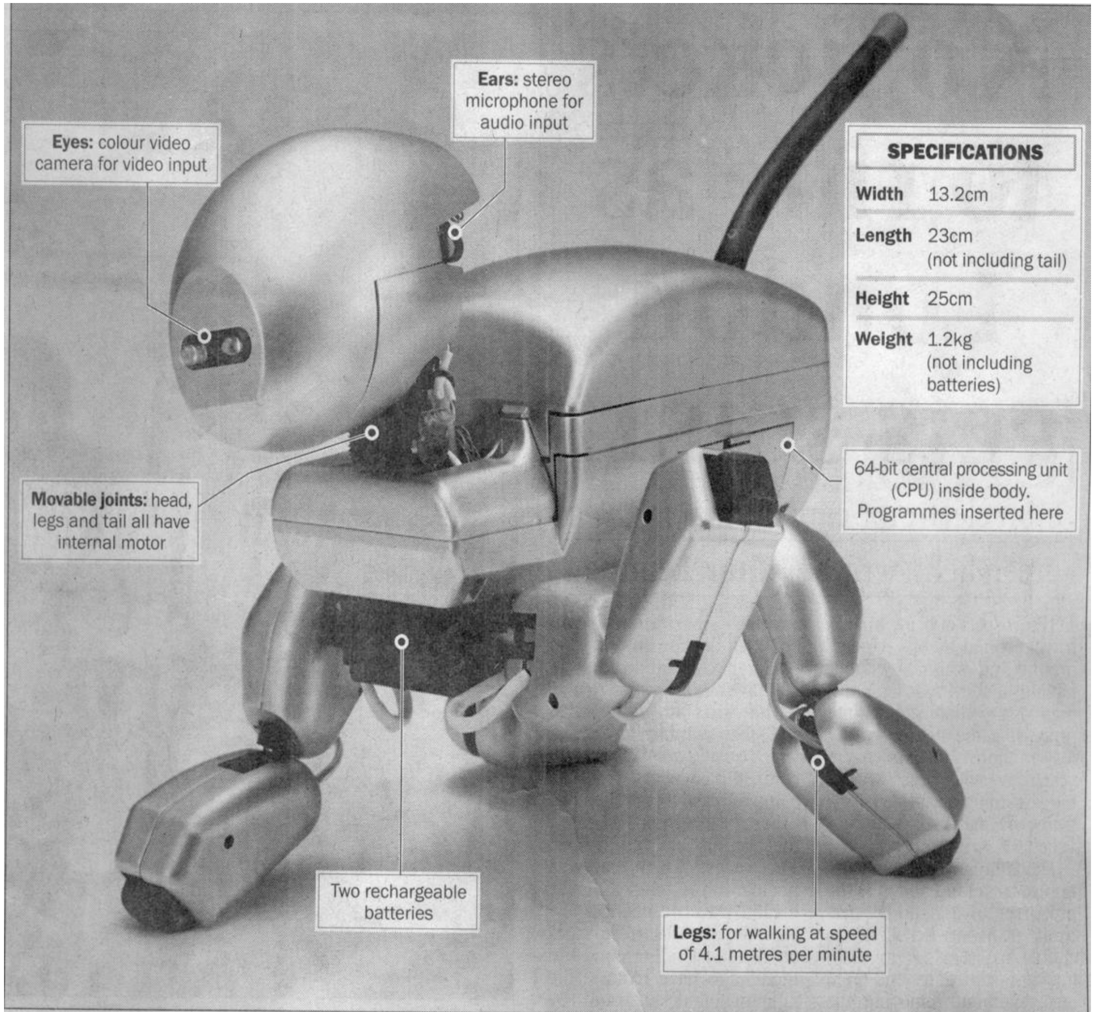


- Bodyweight supported by at least three legs
- Even if all joints ‘freeze’ instantaneously, the robot will not fall
- safe ↔ slow and inefficient

- The robot will fall if not continuously moving
- Less than three legs can be in ground contact
- fast, efficient ↔ demanding for actuation and control

Walking Robots with Four Legs (Quadruped)

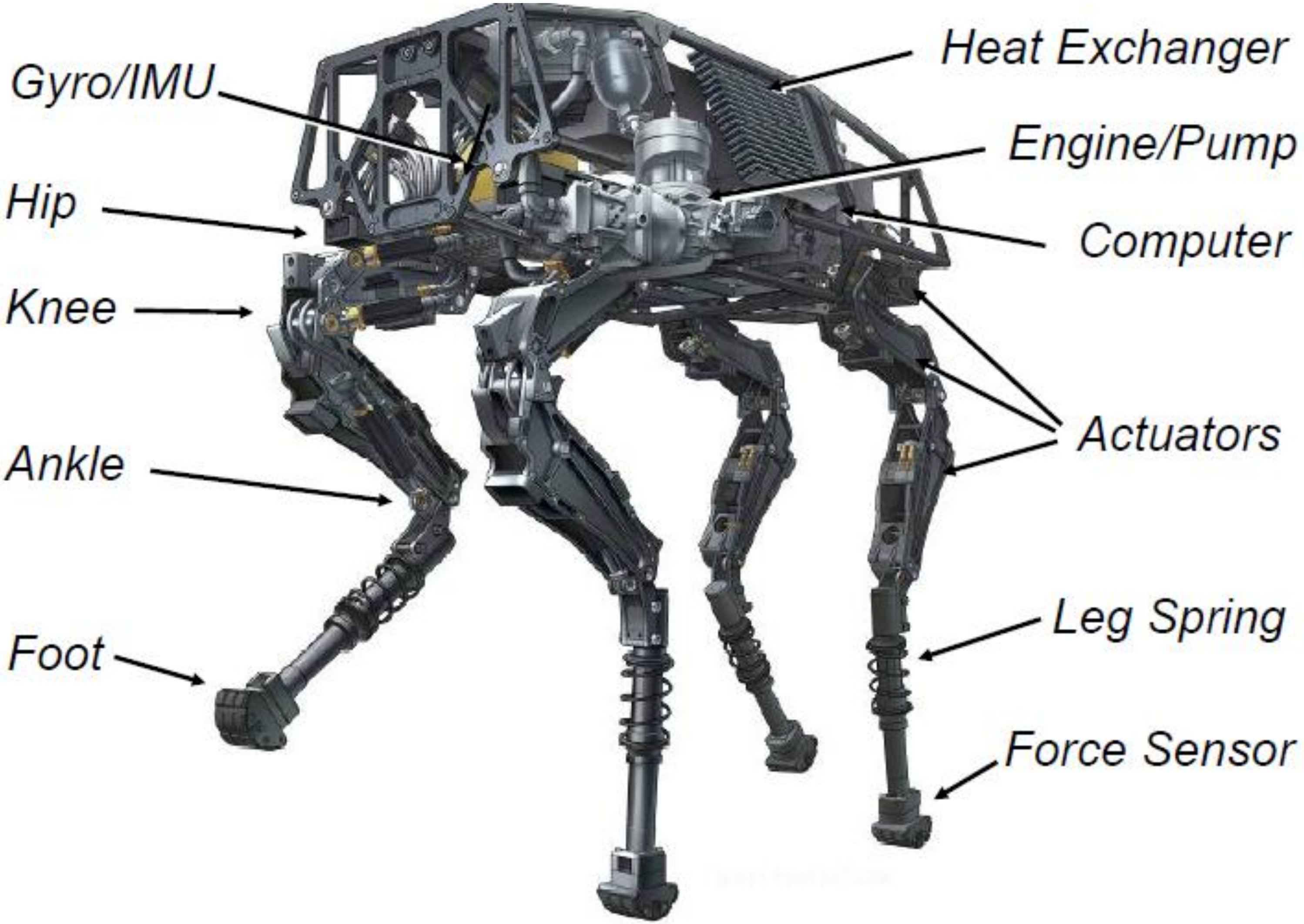
- Artificial Dog Aibo from Sony, Japan



Sony's new computer canine will chase a ball or sit when asked, but cleaning up will be a thing of the past

Dynamic Walking Robots with Four Legs (Quadruped)

- Boston Dynamics Big Dog



C Boston Dynamics

The number of distinct event sequences for biped:

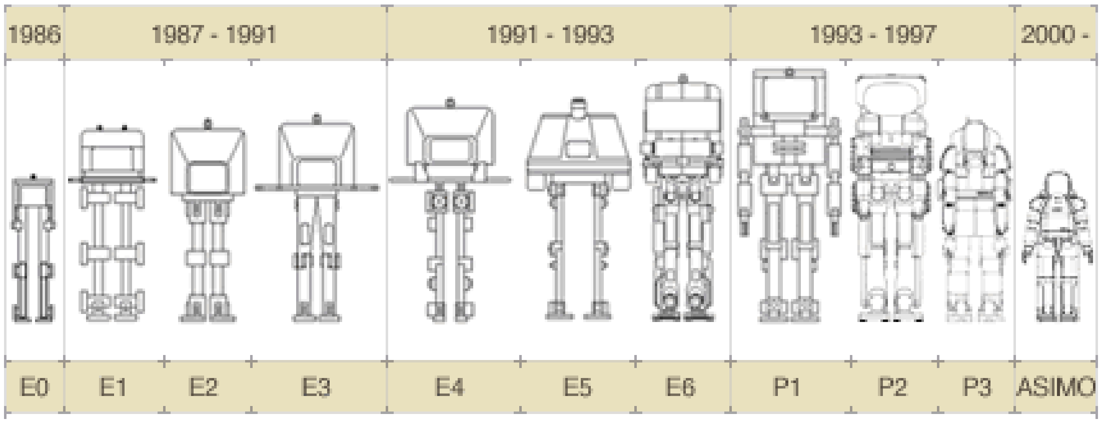
● *Leg down*
○ *Leg up*

- With two legs (biped) one can have four different states
  - 1) Both legs down ●/●
  - 2) Right leg down, left leg up ○/●
  - 3) Right leg up, left leg down ●/○
  - 4) Both leg up ○/○
- A distinct event sequence can be considered as a change from one state to another and back.
- So we have the following $N = (2k-1)! = 6$ distinct event sequences (change of states) for a biped:

1 -> 2 -> 1	●/● ○/● ●/● →	*turning on right leg*
1 -> 3 -> 1	●/● ●/○ ●/● →	*turning on left leg*
1 -> 4 -> 1	●/● ○/○ ●/● →	*hopping with two legs*
2 -> 3 -> 2	○/● ●/○ ○/● →	*walking running*
2 -> 4 -> 2	○/● ○/○ ○/● →	*hopping right leg*
3 -> 4 -> 3	●/○ ○/○ ●/○ →	*hopping left leg*

Case Study: Stiff 2 Legged Walking

- P2, P3 and Asimo from Honda, Japan
- P2
  - Maximum Speed: 2 km/h
  - Autonomy: 15 min
  - Weight: 210 kg
  - Height: 1.82 m
  - Leg DOF: 2x6
  - Arm DOF: 2x7



C Honda corp.

Efficiency Comparison

- Efficiency = c_{mt} = |mech. energy| / (weight x dist. traveled)

$c_{mt}^{est.} \approx 1.6$

Collins *et al.* 2005

$c_{mt} \approx 0.31$

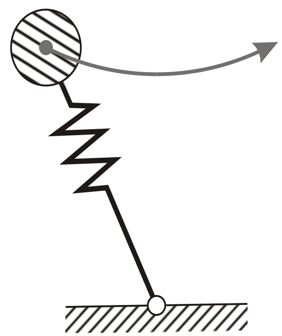
$c_{mt} \approx 0.055$

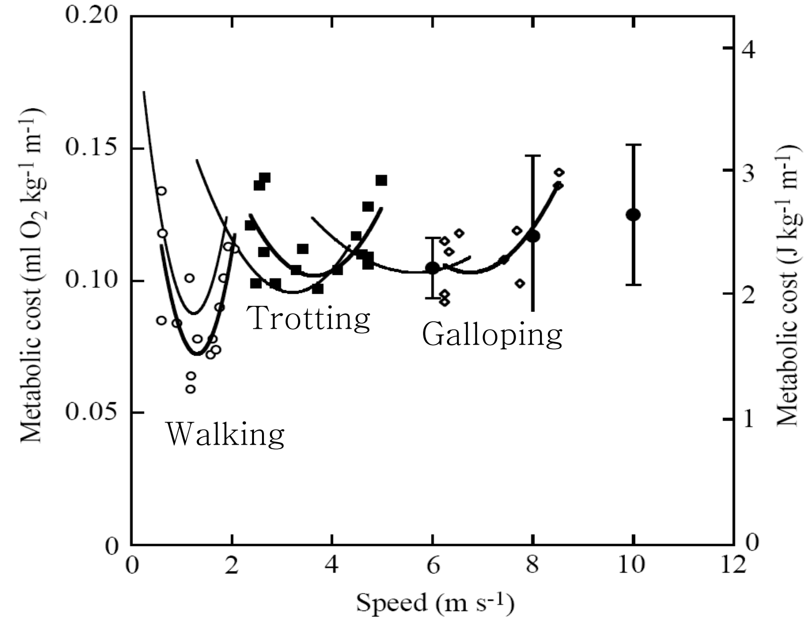
Collins *et al.* 2005

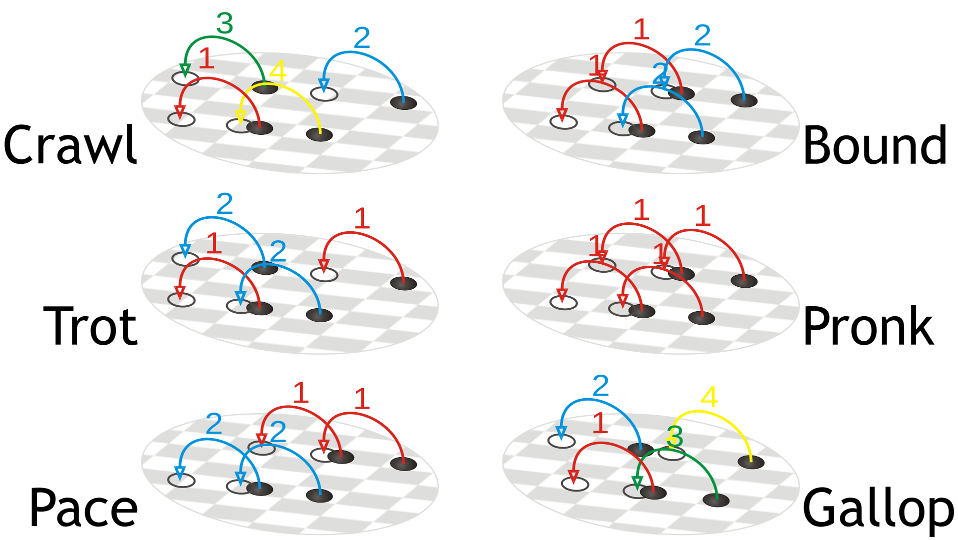
C J. Braun, University of Edinburgh, UK

Towards Efficient Dynamic Walking: Optimizing Gaits

- Nature optimizes its gaits
- Storage of “elastic” energy
- To allow locomotion at varying frequencies and speeds, different gaits have to utilize these elements differently

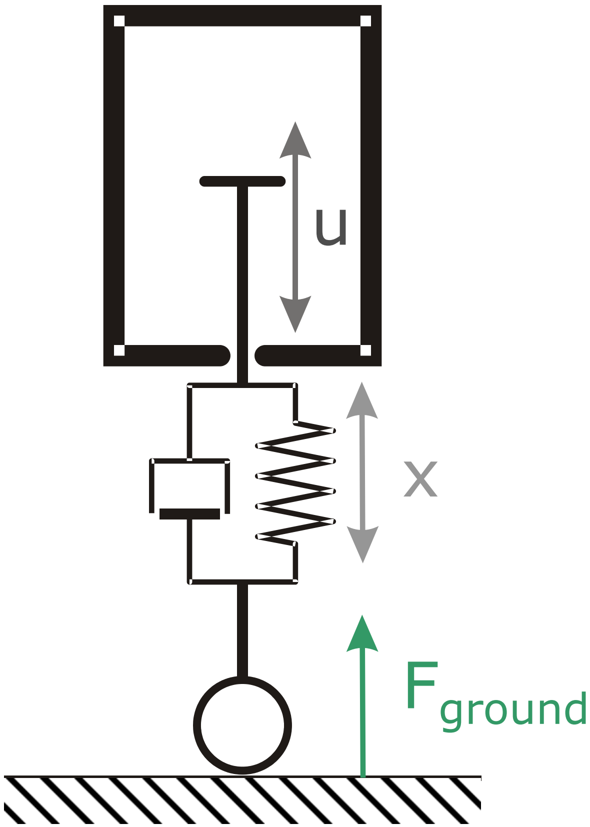






- The energetically most economic gait is a function of desired speed.
(Figure [Minetti et al. 2002])

Towards Efficient Dynamic Walking: Series Elastic Actuation

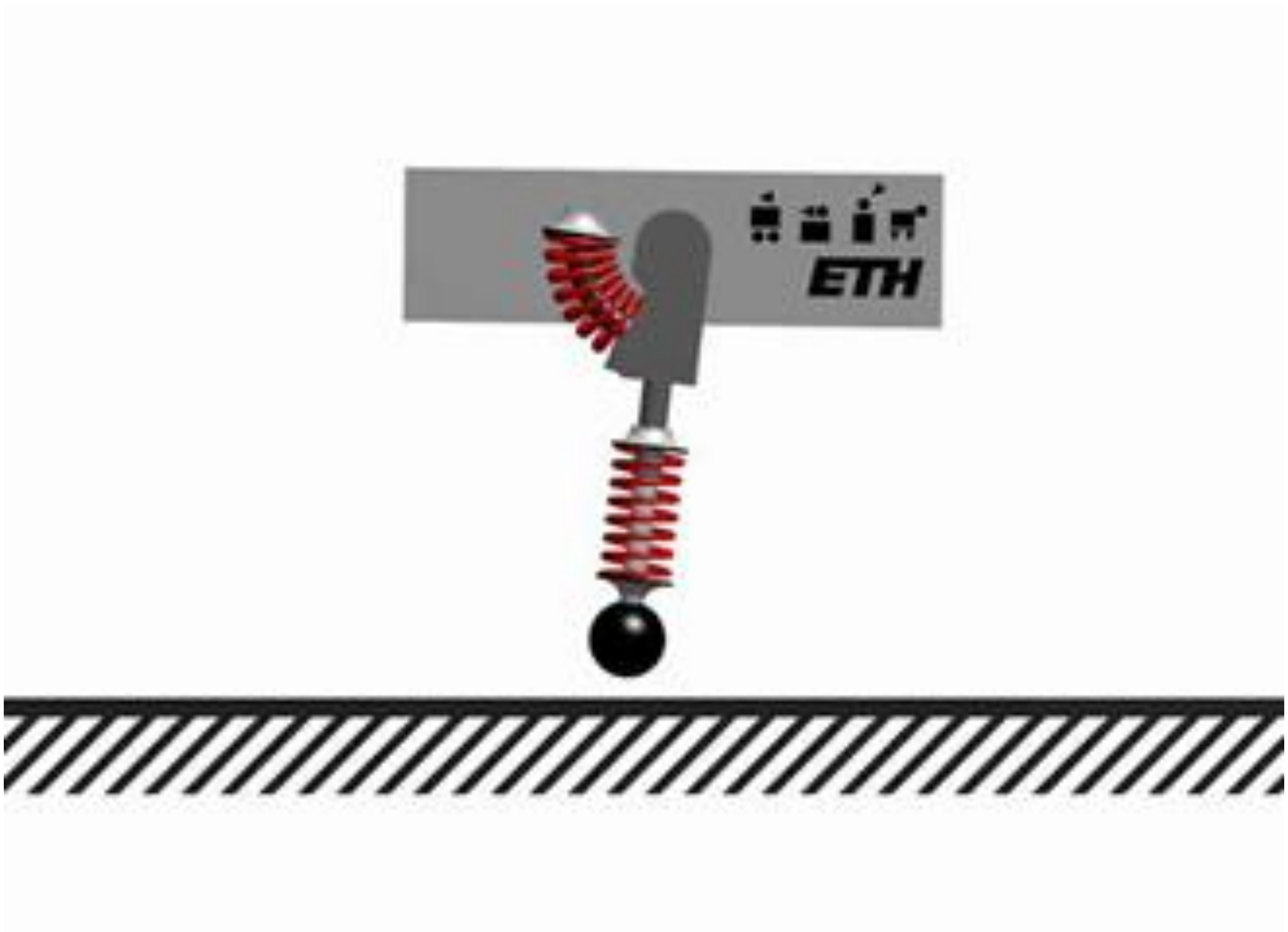


Series Elastic Actuator

- The optimal actuator for such a purpose should
  - be **backdrivable** to allow unimpeded natural dynamics
  - be able to **perform negative work**
  - have a low inertia and gear ratio to keep the reflected inertia small
  - have an adjustable actuator compliance
  - be **highly efficient**
- Series Elastic Actuators can emulate some of these properties
  - However, they come at the cost of active energy consumption
  - Some of the efficiency-benefits of passive dynamic locomotion are only shown as 'proof of concept'

:Case Study: Efficient Walking with Springs

- ETH-ASL Hopping Leg

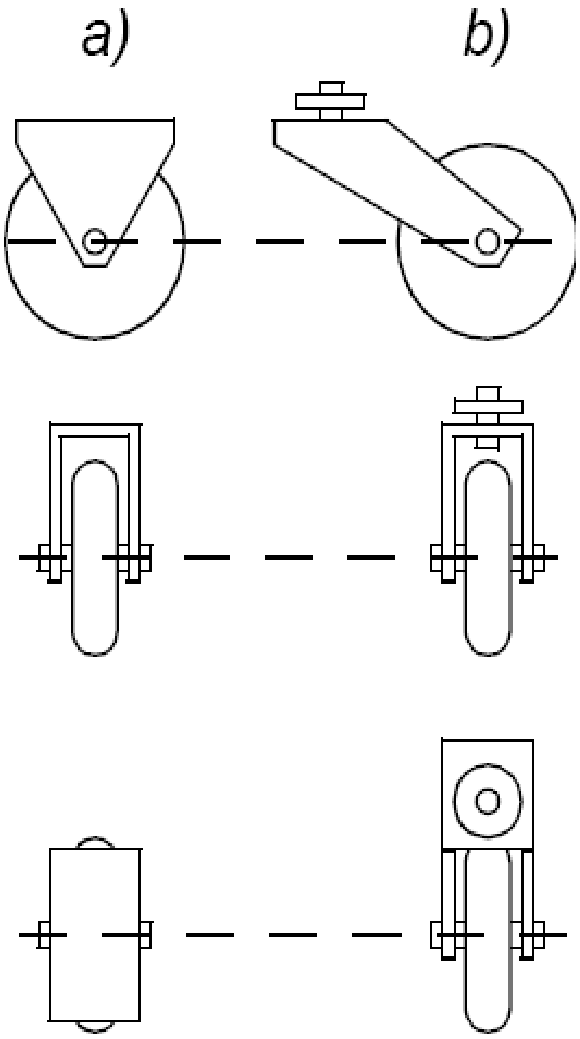


Mobile Robots with Wheels

- Wheels are the most appropriate solution for most applications
- Three wheels are sufficient to guarantee stability
- With more than three wheels an appropriate suspension is required
- Selection of wheels depends on the application

The Four Basic Wheels Types

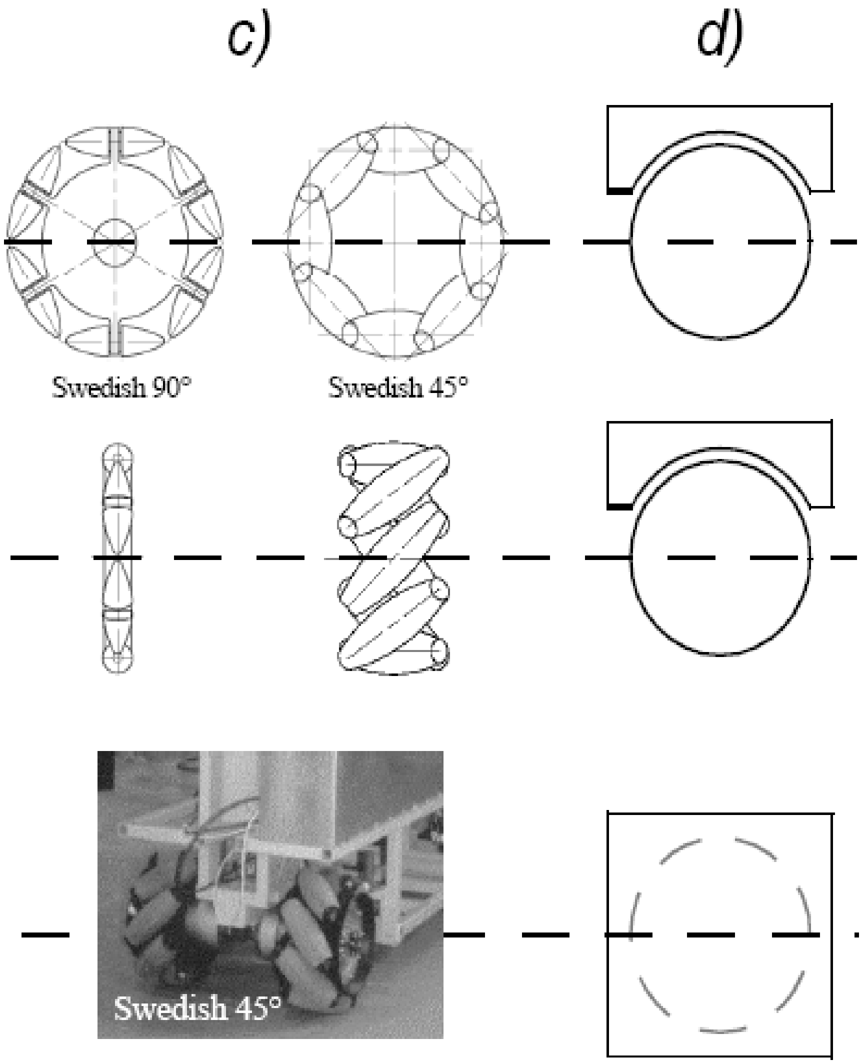
- a) Standard wheel: Two degrees of freedom; rotation around the (motorized) wheel axle and the contact point
- b) Castor wheel: Three degrees of freedom; rotation around the wheel axle, the contact point and the castor axle



The Four Basic Wheels Types

- c) Swedish wheel: Three degrees of freedom; rotation around the (motorized) wheel axle, around the rollers and around the contact point

- d) Ball or spherical wheel: Suspension technically not solved

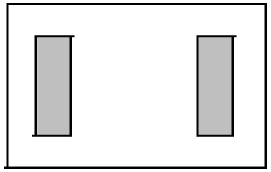


Characteristics of Wheeled Robots and Vehicles

- **Stability** of a vehicle is be guaranteed with **3 wheels**
  - If center of gravity is within the triangle which is formed by the ground contact point of the wheels.
- Stability is improved by 4 and more wheel
  - however, this arrangements are hyper static and require a flexible suspension system.
- **Bigger wheels** allow to overcome **higher obstac**les
  - but they require higher torque or reductions in the gear box.
- Most arrangements are **non-holonomic** (see chapter 3)
  - require high control effort
- Combining actuation and steering on one wheel makes the design complex and adds additional errors for odometry.

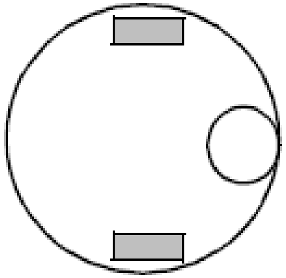
Different Arrangements of Wheels I

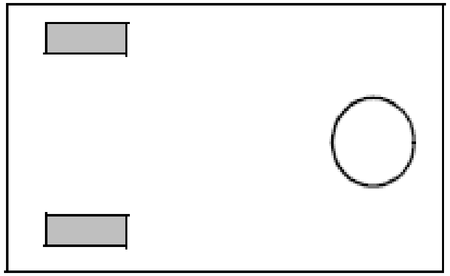
- Two wheels

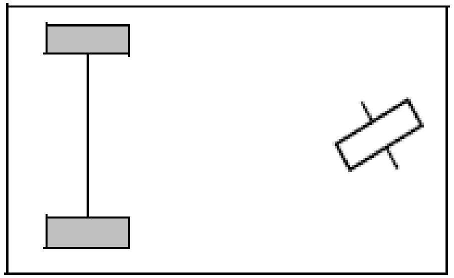


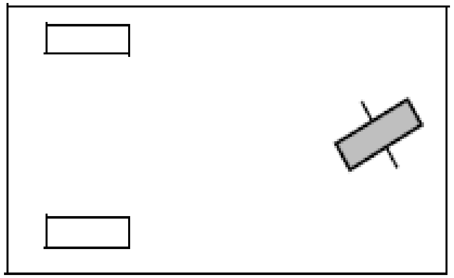
COG below axle

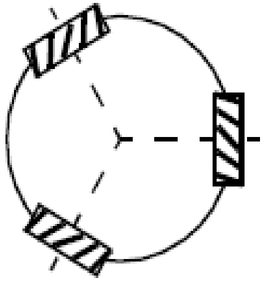
- Three wheels



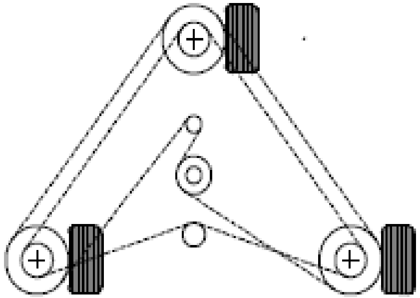








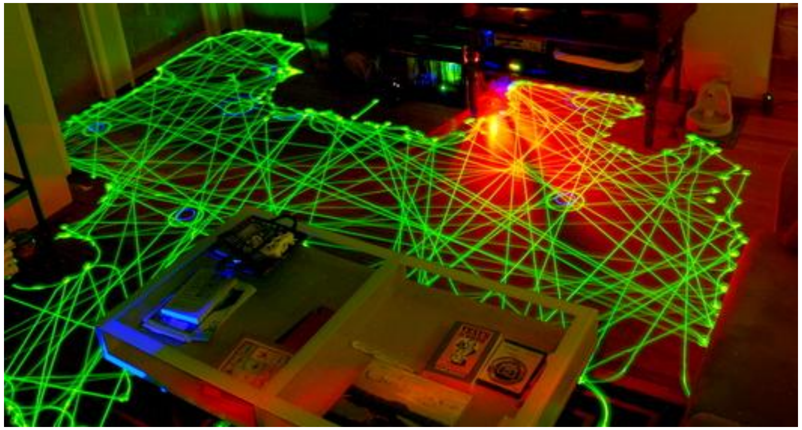
Omnidirectional Drive

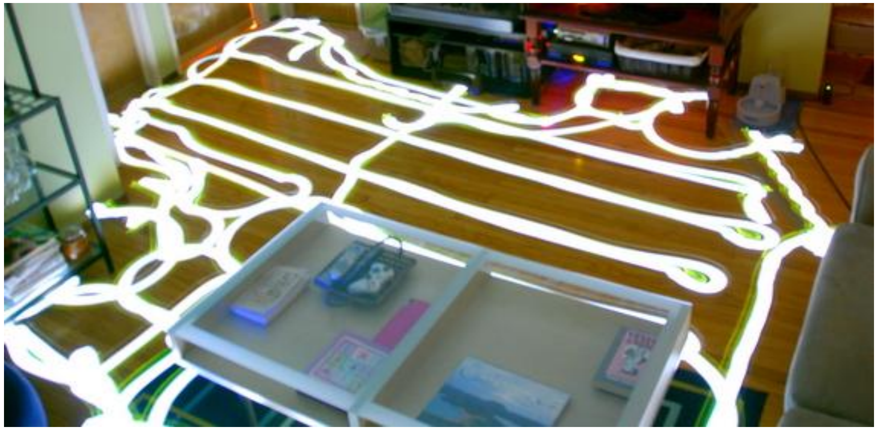


Synchro Drive

Case Study: Vacuum Cleaning Robots

- iRobot Roomba vs.
- Neato XV-11



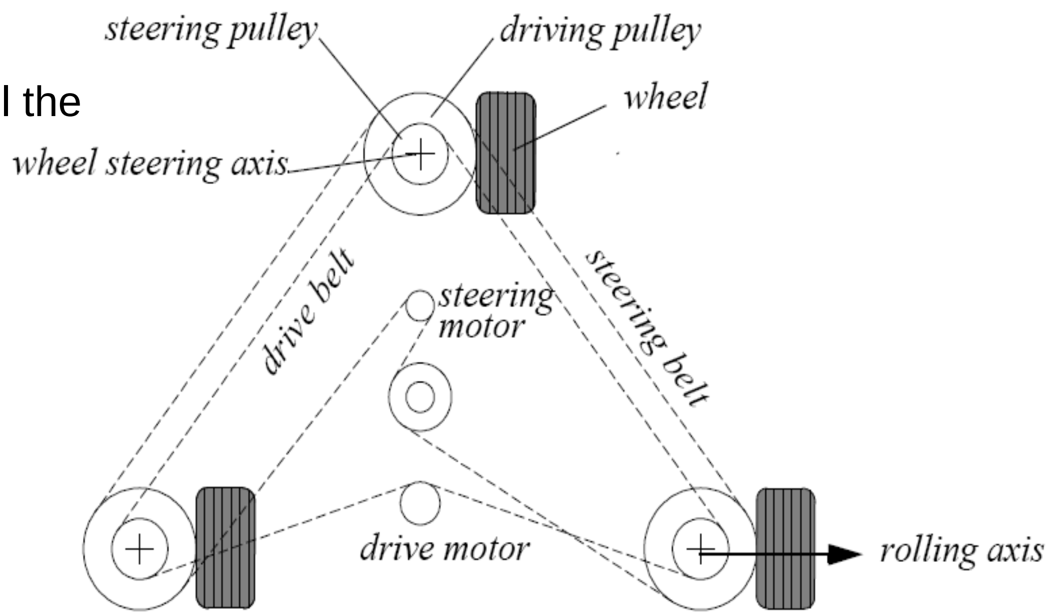


Images courtesy http://www.botjunkie.com

Synchro Drive

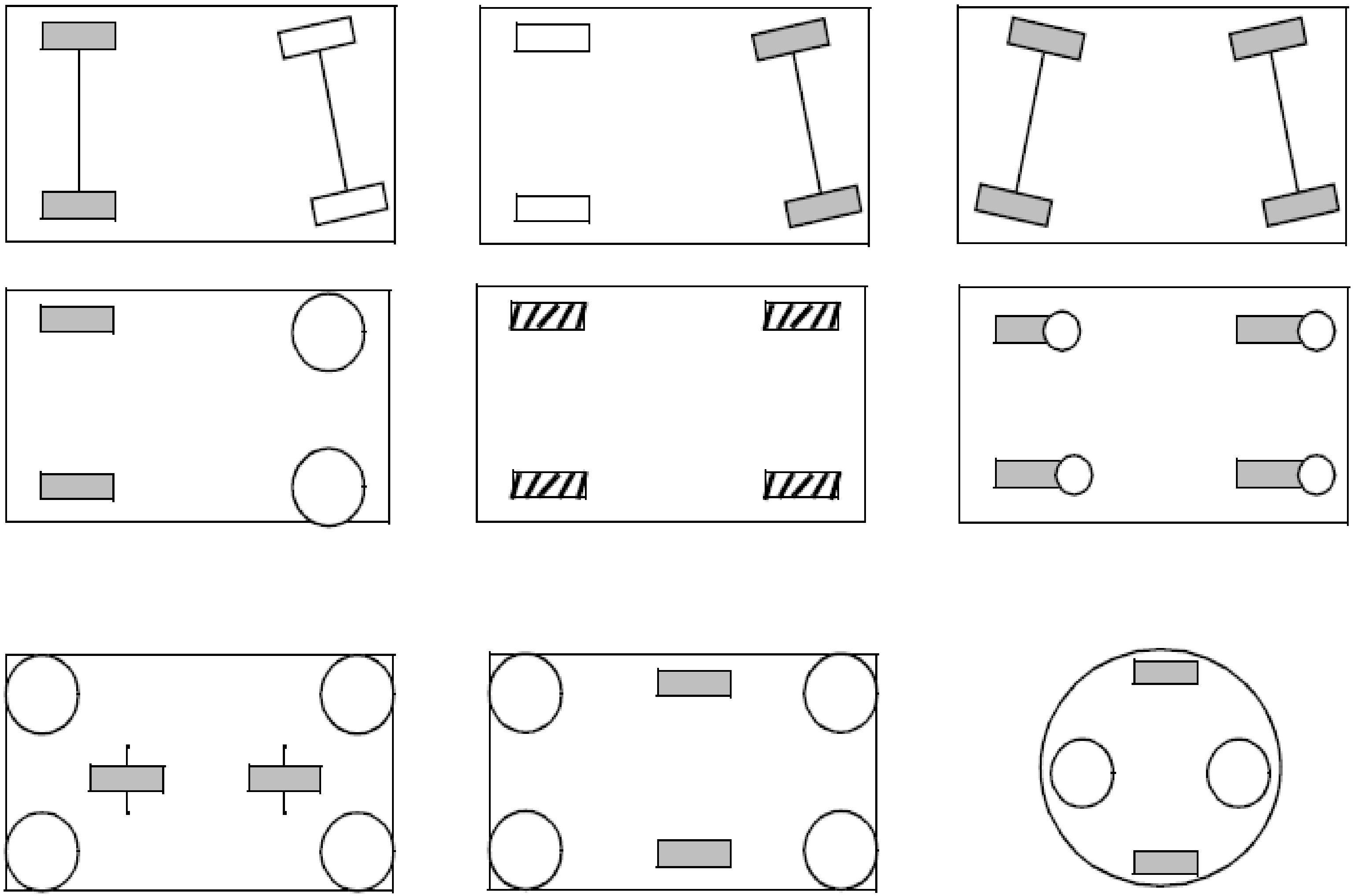
- All wheels are actuated synchronously by one motor
  - defines the speed of the vehicle
- All wheels steered synchronously by a second motor
  - sets the heading of the vehicle

- The orientation in space of the robot frame will always remain the same
  - It is therefore not possible to control the orientation of the robot frame.



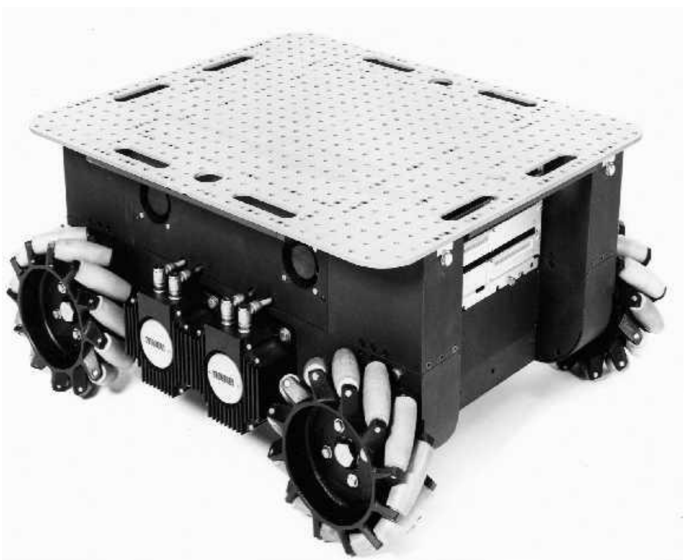
Different Arrangements of Wheels II

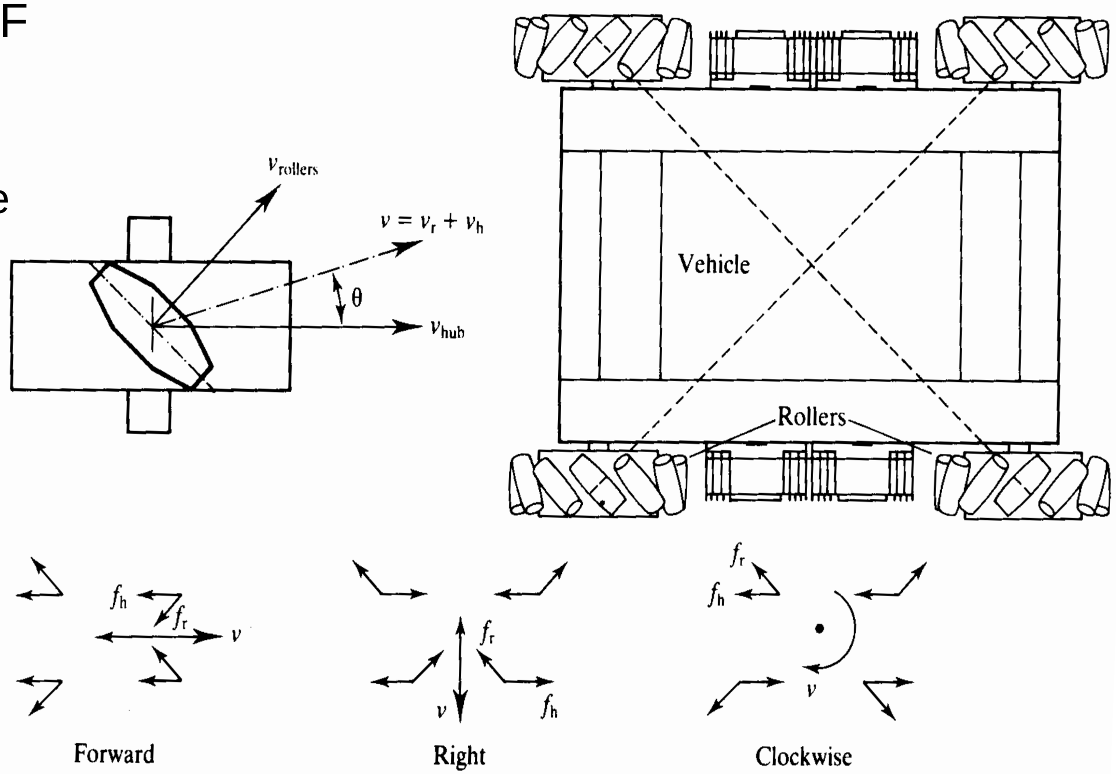
- Four wheels



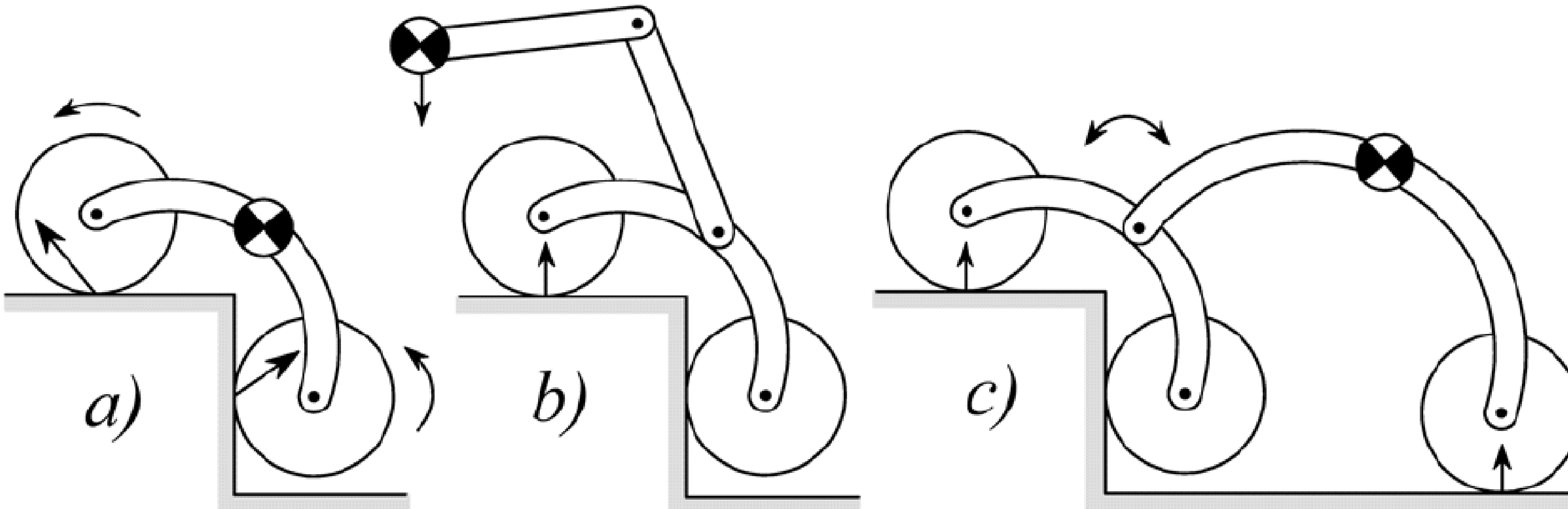
CMU Uranus: Omnidirectional Drive with 4 Wheels

- Movement in the plane has 3 DOF
  - thus only three wheels can be independently controlled
  - It might be better to arrange three swedish wheels in a triangle





Wheeled Rovers: Concepts for Object Climbing



Purely friction based

Change of center of gravity (CoG)

Adapted suspension mechanism with passive or active joints

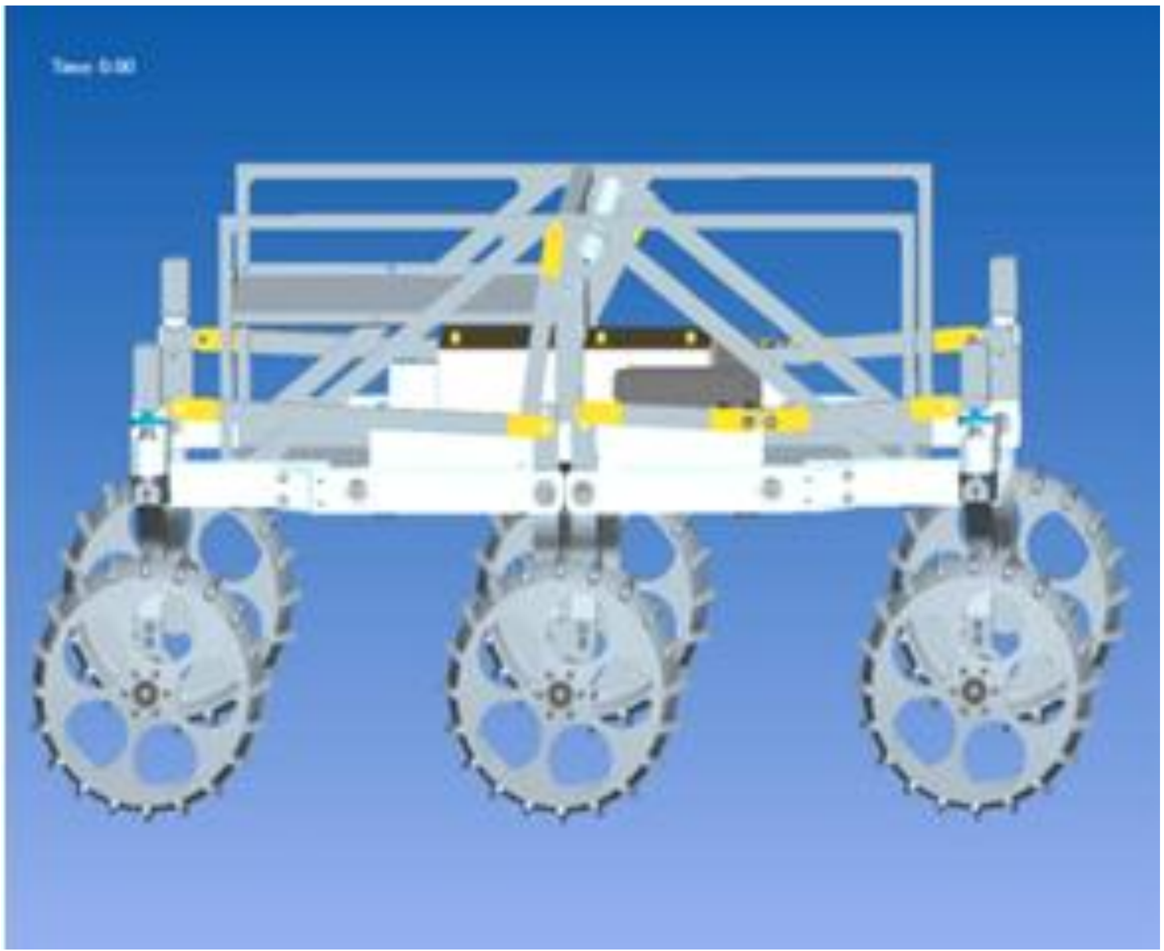
The Personal Rover

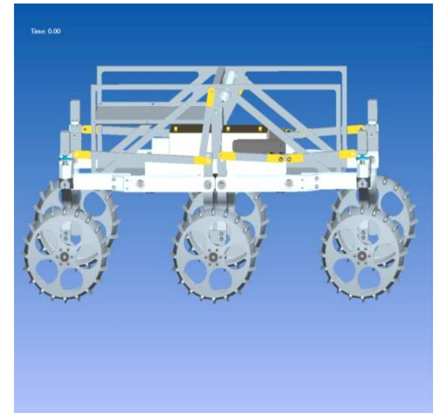
Climbing with Legs: EPFL Shrimp

- Passive locomotion concept
- 6 wheels
  - two boogies on each side
  - fixed wheel in the rear
  - front wheel with spring suspension
- Dimensions
  - length: 60 cm
  - height: 20 cm
- Characteristics
  - highly stable in rough terrain
  - **overcomes obstacles up to 2 times its wheel diameter**

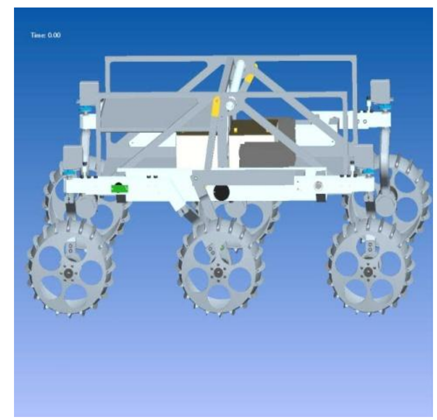
Rover Concepts for Planetary Exploration

- ExoMars: ESA Mission to Mars in ~~2013, 2015~~, 2018
  - Six wheels
  - Symmetric chassis
  - No front fork → intstrument placement

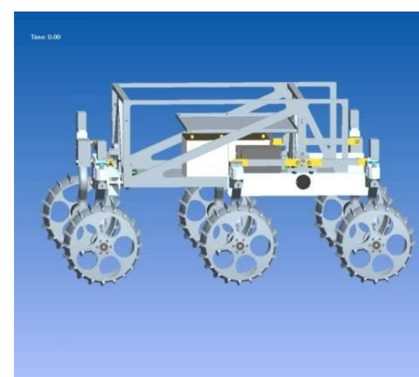




Crab ETH



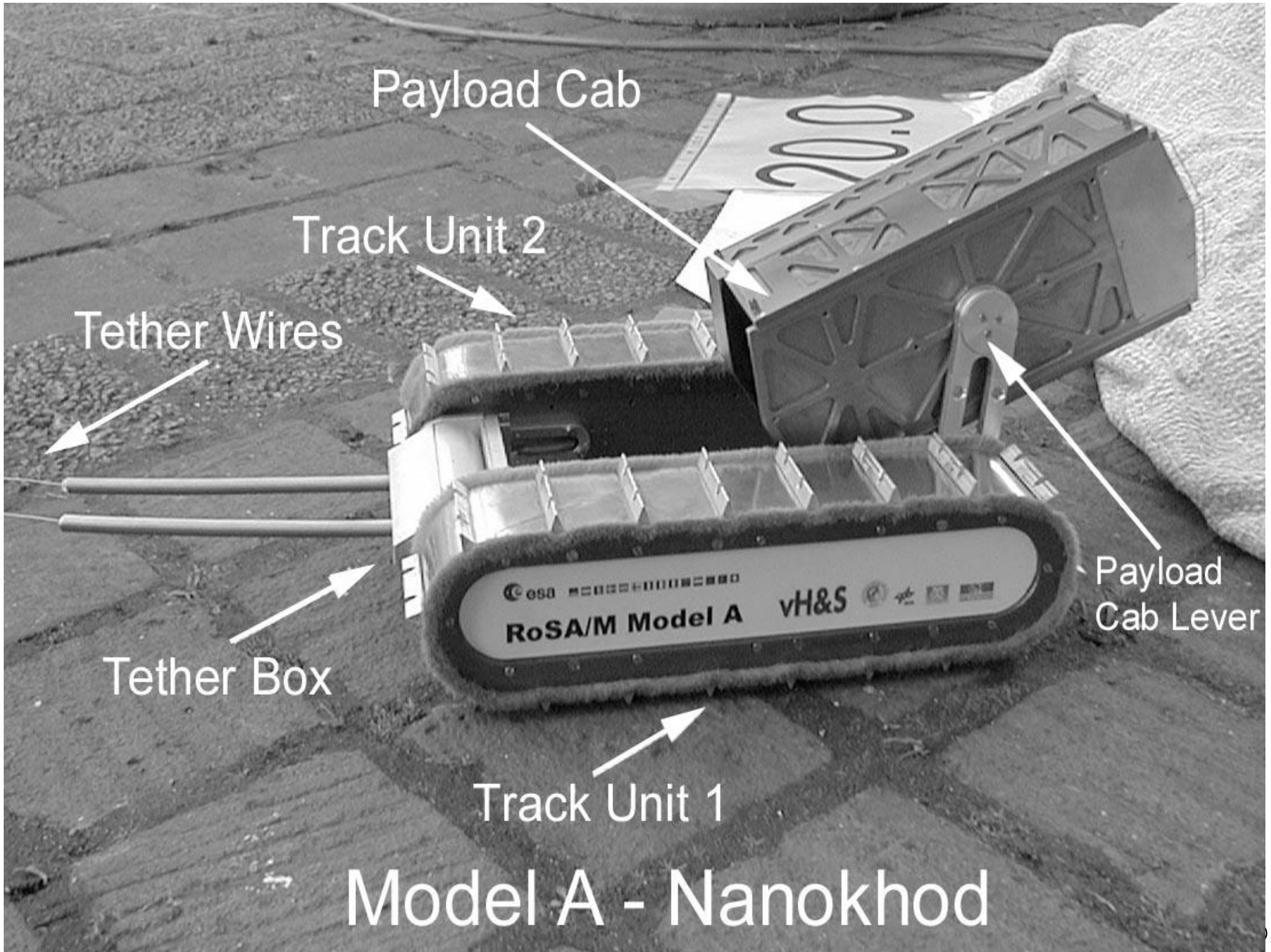
Concept C
RCL Russia



Concept E

Caterpillar

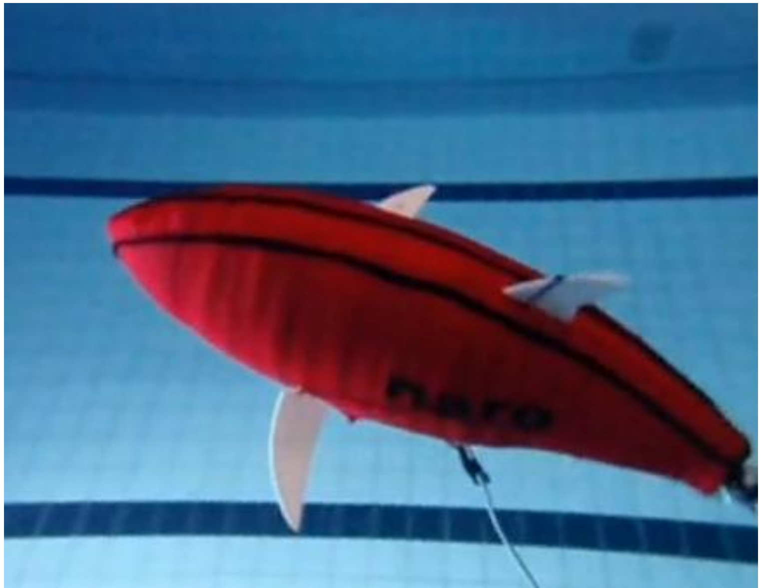
- The NANOKHOD II,
  - developed by von Hoerner & Sulger GmbH and Max Planck Institute, Mainz
  - will probably go to Mars



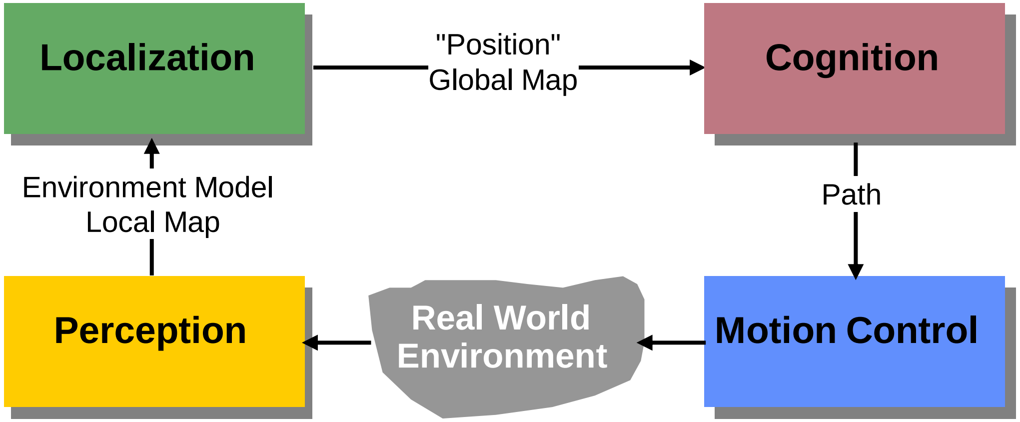
Other Forms of „Locomotion“: Traditional and Emerging

- Flying

- Swimming



C Essex Univ.



Mobile Robot Kinematics

Mobile Robot Kinematics: Overview

- Mobile robot and manipulator arm characteristics
  - Arm is fixed to the ground and usually comprised of a single chain of actuated links
  - Mobile robot motion is defined through rolling and sliding constra-ints taking effect at the wheel-ground contact points

C Willow Garage C dexter12322222222222, youtube.com

3 Mobile Robot Kinematics: Overview

- Definition and Origin
  - From *kinein* (Greek); to move
  - Kinematics is the subfield of Mechanics which deals with motions of bodies

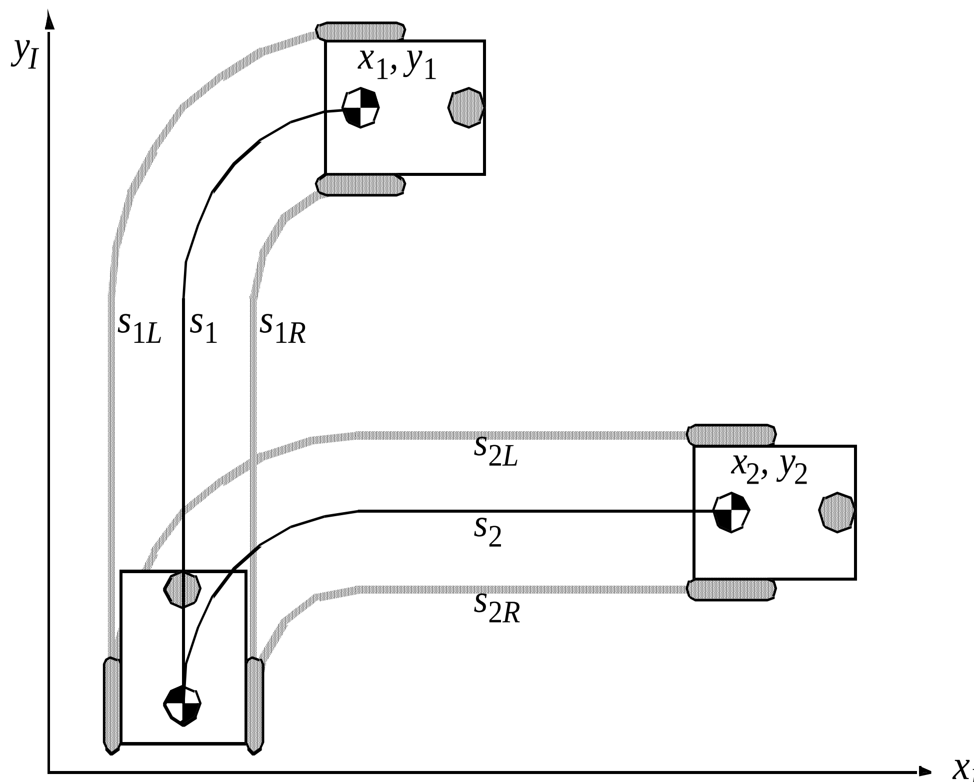
- Manipulator- vs. Mobile Robot Kinematics
  - Both are concerned **with forward and inverse kinematics**
  - However, for mobile robots, encoder values don‘t map to unique robot poses
  - However, **mobile robots** can move unbound with respect to their environment
    - There is **no direct** (=instantaneous) **way to measure the robot’s position**
    - **Position must be integrated over time**, depends on path taken
    - Leads to inaccuracies of the position (motion) estimate
  - Understanding mobile robot motion starts with **understanding wheel con-straints** placed on the robot’s mobility

Non-Holonomic Systems

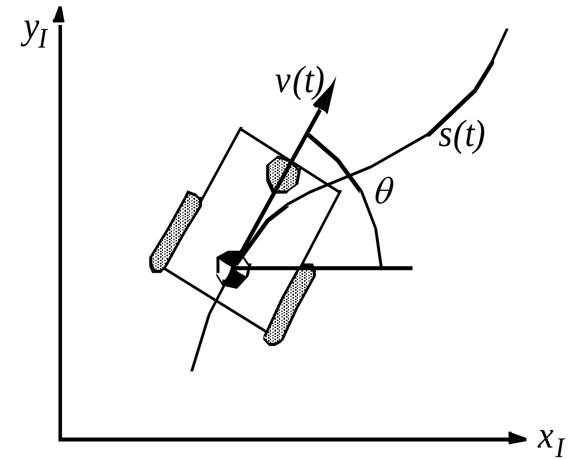
- Non-holonomic systems
  - differential equations are not integrable to the final position.
  - the measure of the traveled distance of each wheel is not sufficient to calculate the final position of the robot. One has also to know how this movement was executed as a function of time.
  - This is in stark contrast to actuator arms

$$s_1 = s_2, s_{1R} = s_{2R}, s_{1L} = s_{2L}$$

$$x_1 \neq x_2, y_1 \neq y_2$$



Non-Holonomic Systems



- A mobile robot is running along a trajectory *s(t)*.
  At every instant of the movement its velocity *v(t)* is:

$$v(t)=\frac{\partial s}{\partial t}=\frac{\partial x}{\partial t}\cos\theta+\frac{\partial y}{\partial t}\sin\theta$$

$$ds=dx\cos\theta+dy\sin\theta$$

- Function *v(t)* is said to be integrable (holonomic) if there exists a trajectory function *s(t)* that can be described by the values *x*, *y*, and θ only.

$$s=s(x,y,\theta)$$

- This is the case if

$$\boxed{\frac{\partial^2 s}{\partial x\partial y}=\frac{\partial^2 s}{\partial y\partial x}\ ;\ \frac{\partial^2 s}{\partial x\partial\theta}=\frac{\partial^2 s}{\partial\theta\partial x}\ ;\ \frac{\partial^2 s}{\partial y\partial\theta}=\frac{\partial^2 s}{\partial\theta\partial y}}$$

Condition for s to be integrable function

Non-Holonomic Systems

- With *s* = *s(x,y,θ)* we get for *ds*

$$ds = \frac{\partial s}{\partial x} dx + \frac{\partial s}{\partial y} dy + \frac{\partial s}{\partial \theta} d\theta$$

- and by comparing the equation above with $ds = dx \cos\theta + dy \sin\theta$

- we find $\frac{\partial s}{\partial x} = \cos\theta \;\; ; \;\; \frac{\partial s}{\partial y} = \sin\theta \;\; ; \;\; \frac{\partial s}{\partial \theta} = 0$

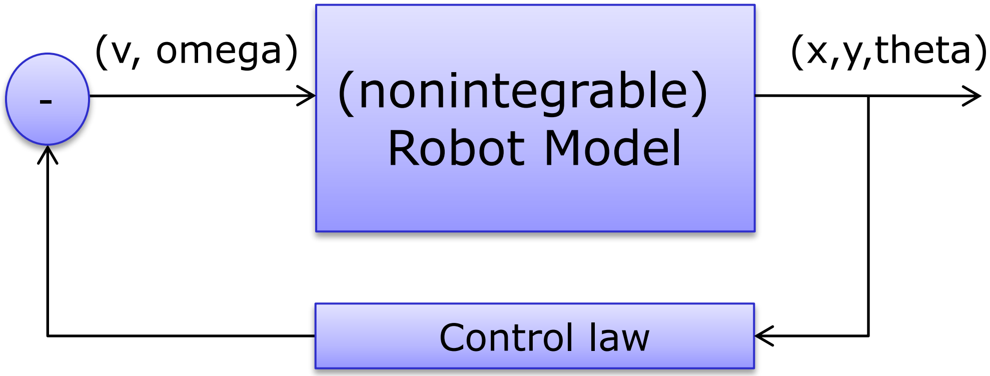
- Condition for an integrable (holonomic) function:
  - the second (-sinθ=0) and third (cosθ=0) term in the equation do not hold!

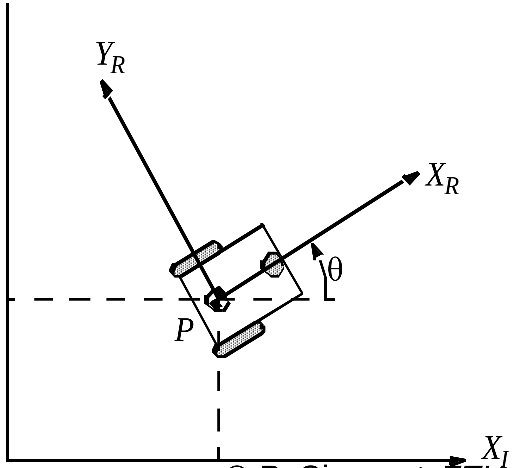
$$\frac{\partial^2 s}{\partial x \partial y} = \frac{\partial^2 s}{\partial y \partial x} \;\; ; \;\; \frac{\partial^2 s}{\partial x \partial \theta} = \frac{\partial^2 s}{\partial \theta \partial x} \;\; ; \;\; \frac{\partial^2 s}{\partial y \partial \theta} = \frac{\partial^2 s}{\partial \theta \partial y}$$

Forward and Inverse Kinematics

- Forward kinematics:
  - Transformation from joint- to physical space
- Inverse kinematics
  - Transformation from physical- to joint space
  - Required for motion control

- Due to nonholonomic constraints in mobile robotics, we deal with **differential** (inverse) kinematics
  - Transformation between velocities instead of positions
  - Such a differential kinematic model of a robot has the following form:

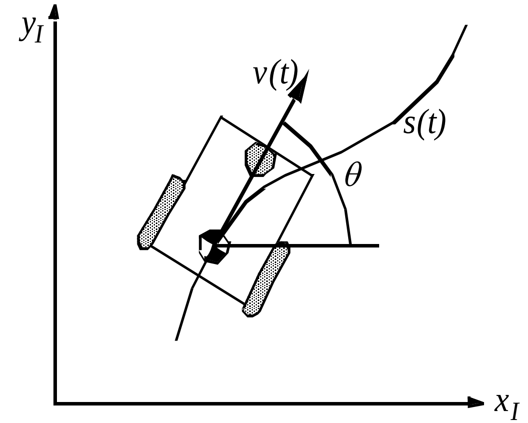




Differential Kinematics Model

- Due to a lack of alternatives:
  - establish the robot speed $\dot{\xi} = \begin{bmatrix}\dot{x} & \dot{y} & \dot{\theta}\end{bmatrix}^T$ as a function of the wheel speeds $\dot{\varphi}_i$, steering angles β_i , steering speeds $\dot{\beta}_i$ and the geometric parameters of the robot (*configuration coordinates*).
  - forward kinematics

$$\dot{\xi} = \begin{bmatrix}\dot{x} \\ \dot{y} \\ \dot{\theta}\end{bmatrix} = f(\dot{\varphi}_1, \ldots \dot{\varphi}_n, \beta_1, \ldots \beta_m, \dot{\beta}_1, \ldots \dot{\beta}_m)$$



  - Inverse kinematics

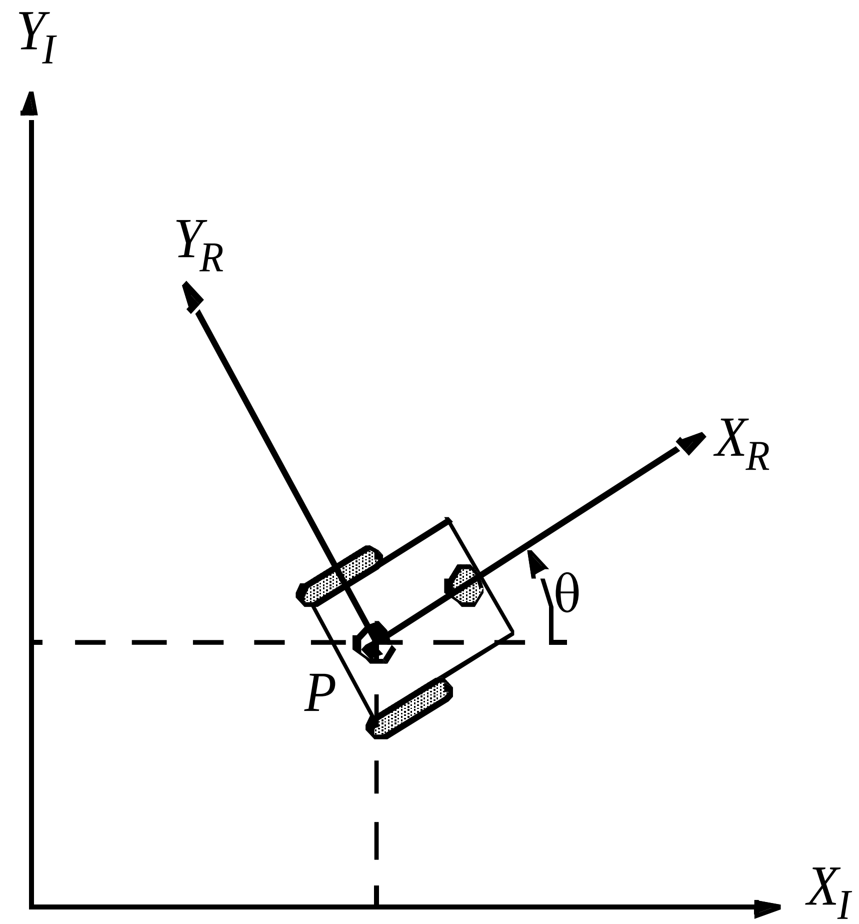
$$\begin{bmatrix}\dot{\varphi}_1 & \cdots & \dot{\varphi}_n & \beta_1 & \ldots & \beta_m & \dot{\beta}_1 & \ldots & \dot{\beta}_m\end{bmatrix}^T = f(\dot{x}, \dot{y}, \dot{\theta})$$

  - But generally not integrable into

$$\begin{bmatrix}x \\ y \\ \theta\end{bmatrix} = f(\varphi_1, \ldots \varphi_n, \beta_1, \ldots \beta_m)$$

Representing Robot Pose

- Representing the robot within an arbitrary initial frame
  - Inertial frame: $\{X_I, Y_I\}$
  - Robot frame: $\{X_R, Y_R\}$
  - Robot pose: $\xi_I = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$
  - Mapping between the two frames



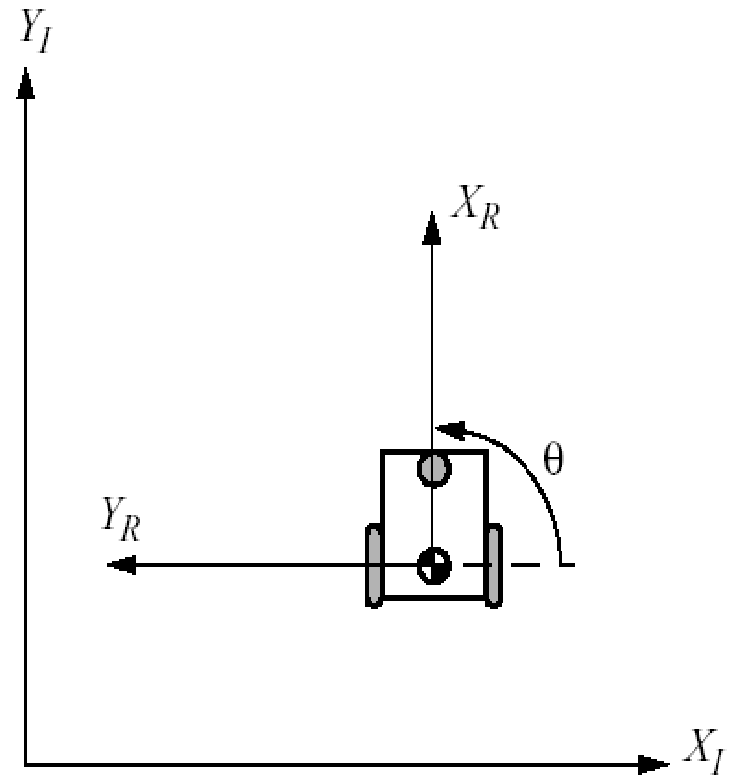
$$\dot{\xi}_R = R(\theta)\dot{\xi}_I = R(\theta)\cdot\begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T$$

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Example: Robot aligned with Y_I

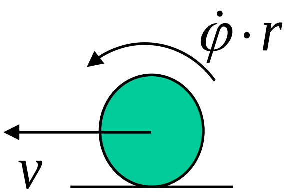
$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

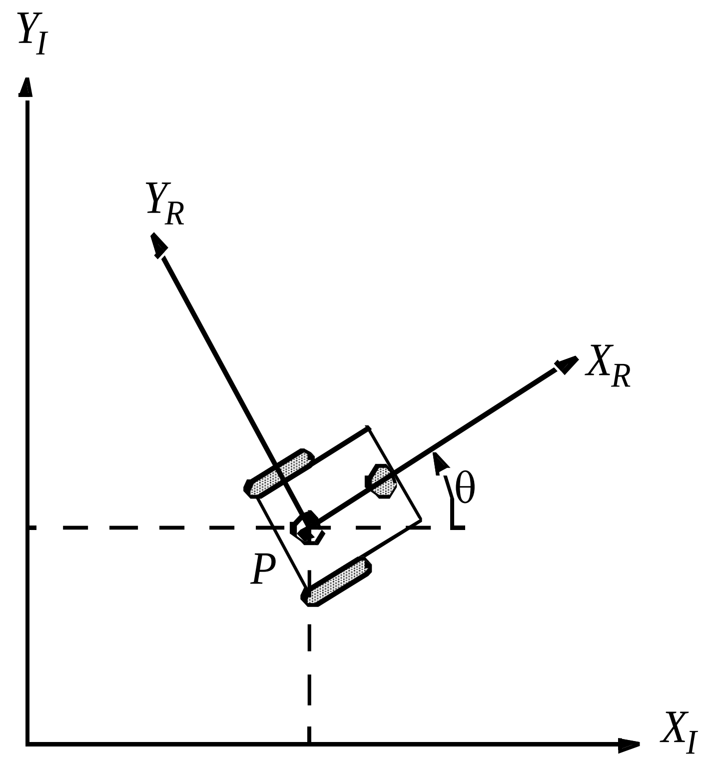
$$\dot{\xi}_R = R(\frac{\pi}{2})\dot{\xi}_I = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{y} \\ -\dot{x} \\ \dot{\theta} \end{bmatrix}$$



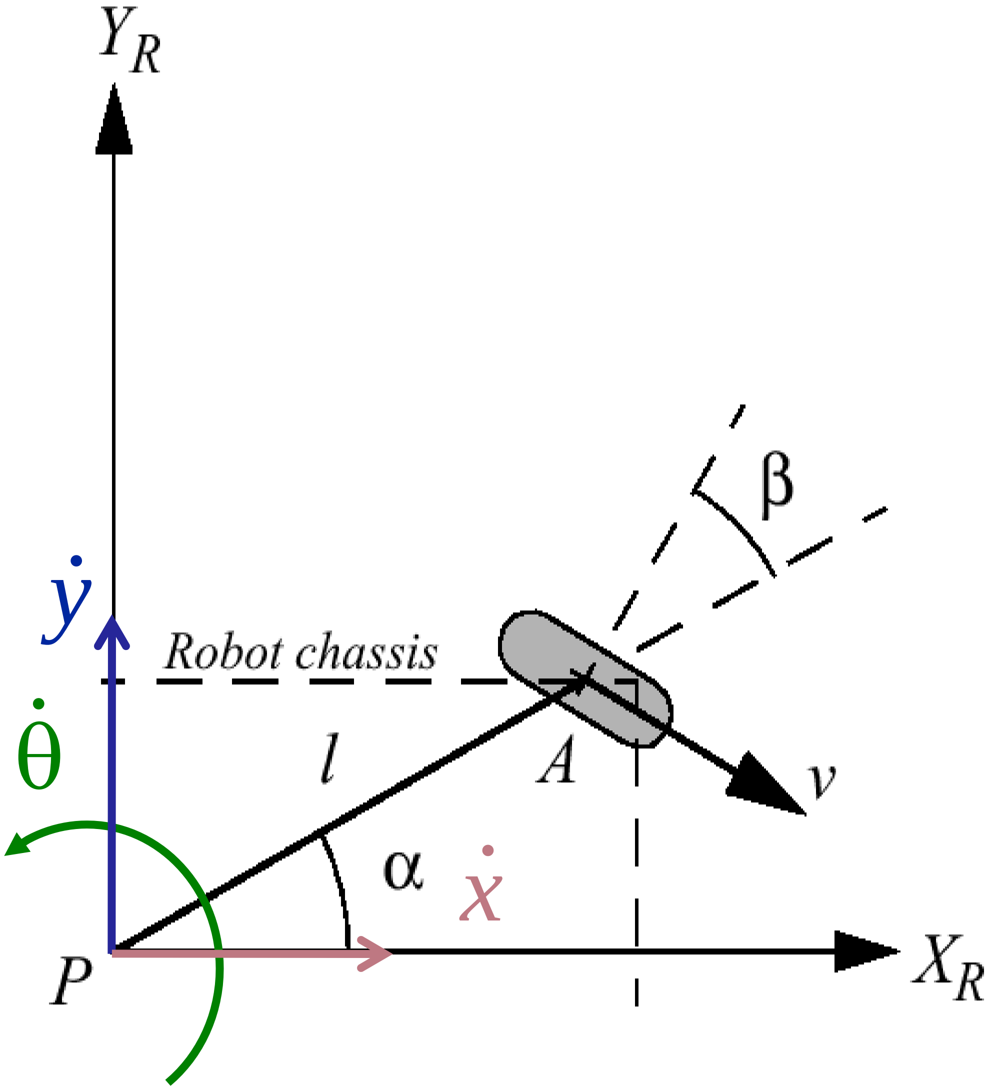
Wheel Kinematic Constraints

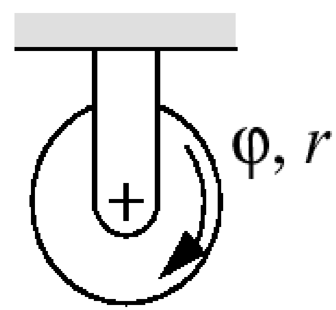
- Assumptions
  - Movement on a horizontal plane
  - Point contact of the wheels
  - Wheels not deformable
  - Pure rolling ($v_c = 0$ at contact point)
  - No slipping, skidding or sliding
  - No friction for rotation around contact point
  - Steering axes orthogonal to the surface
  - Wheels connected by rigid frame (chassis)

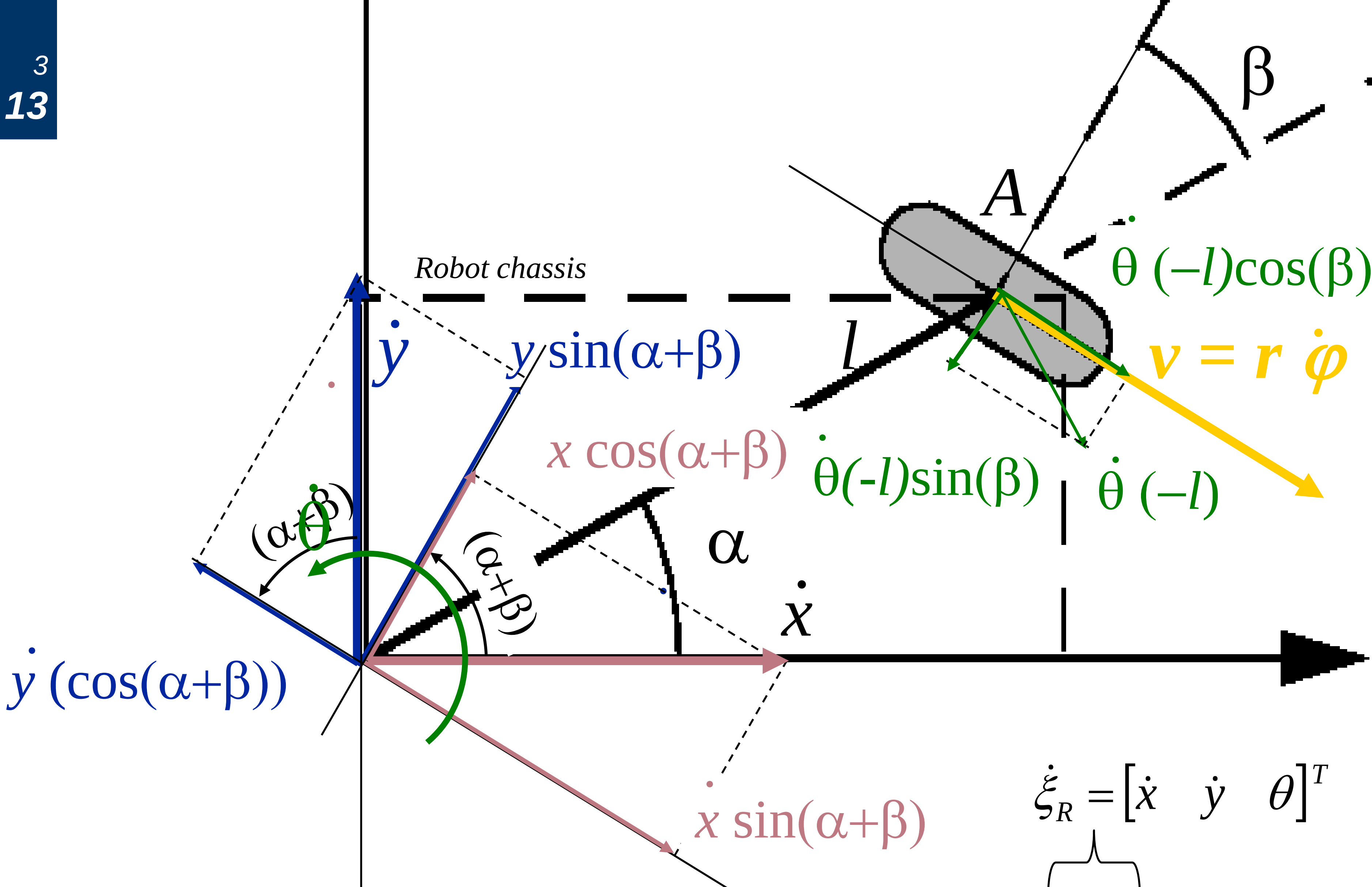




Kinematic Constraints: Fixed Standard Wheel







$$\dot{\xi}_R = \begin{bmatrix}\dot x & \dot y & \theta\end{bmatrix}^T$$

$$\begin{bmatrix}\sin(\alpha+\beta) & -\cos(\alpha+\beta) & (-l)\cos\beta\end{bmatrix}\overbrace{R(\theta)\dot{\xi}_I}^{\dot\xi_R} - r\dot{\varphi} = 0$$

$$\begin{bmatrix}\cos(\alpha+\beta) & \sin(\alpha+\beta) & l\sin\beta\end{bmatrix}R(\theta)\dot{\xi}_I = 0$$

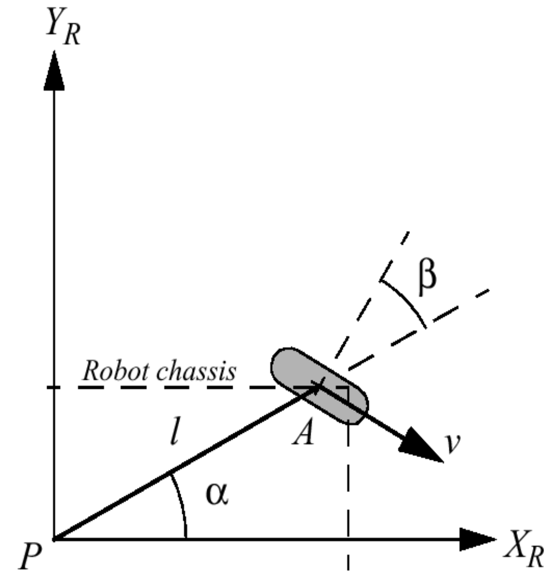
Example

$$\left[\sin(\alpha+\beta)\ \ -\cos(\alpha+\beta)\ \ (-l)\cos\beta\right]R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

$$\left[\cos(\alpha+\beta)\ \ \sin(\alpha+\beta)\ \ l\sin\beta\right]R(\theta)\dot{\xi}_I = 0$$

- Suppose that the wheel A is in position such that **α = 0 and β = 0**
- This would place the contact point of the wheel on X_I with the plane of the wheel oriented parallel to Y_I. If $\theta = 0$, then the **sliding constraint** reduces to:

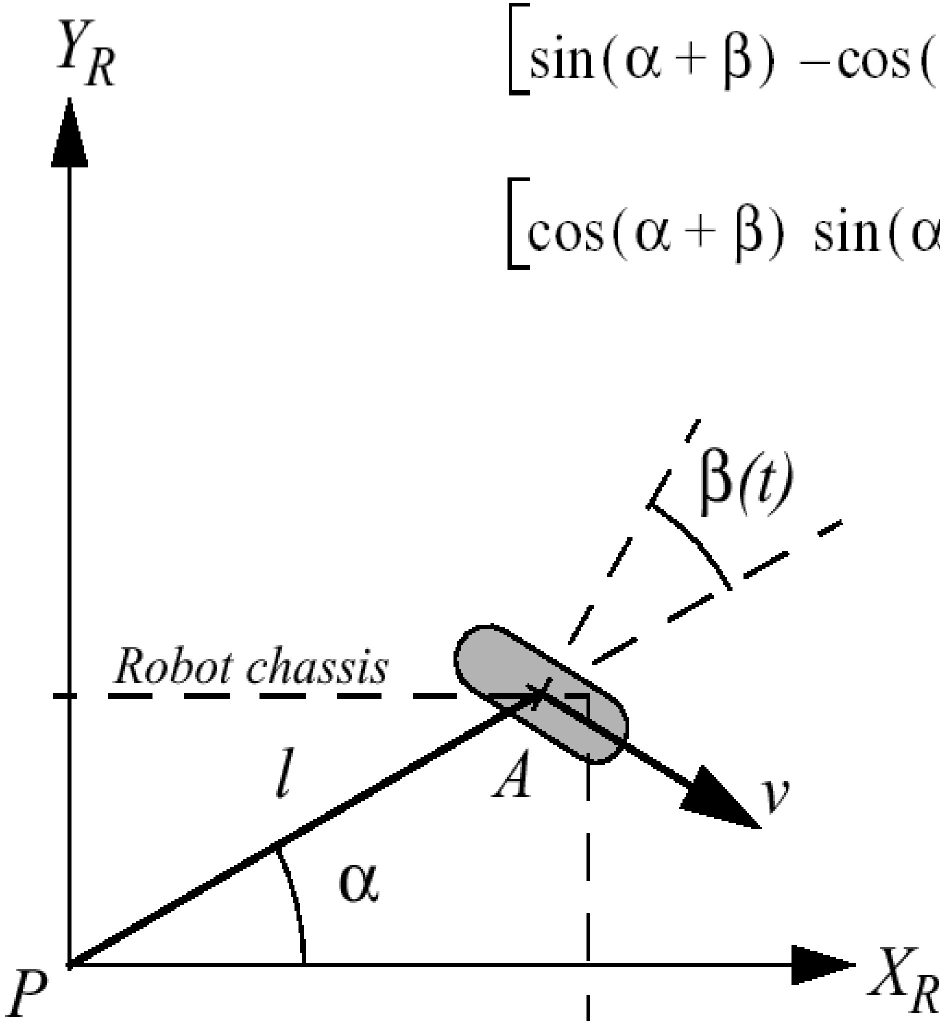
$$\begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}\dot{x}\\\dot{y}\\\dot{\theta}\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}\dot{x}\\\dot{y}\\\dot{\theta}\end{bmatrix} = 0$$

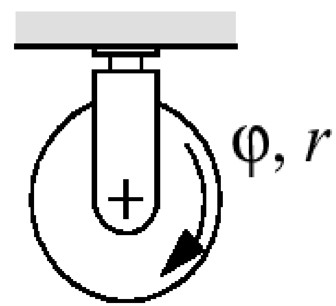


Kinematic Constraints: Steered Standard Wheel

$$\left[\sin(\alpha+\beta) \;\; -\cos(\alpha+\beta) \;\; (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

$$\left[\cos(\alpha+\beta) \;\; \sin(\alpha+\beta) \;\; l\sin\beta\right] R(\theta)\dot{\xi}_I = 0$$

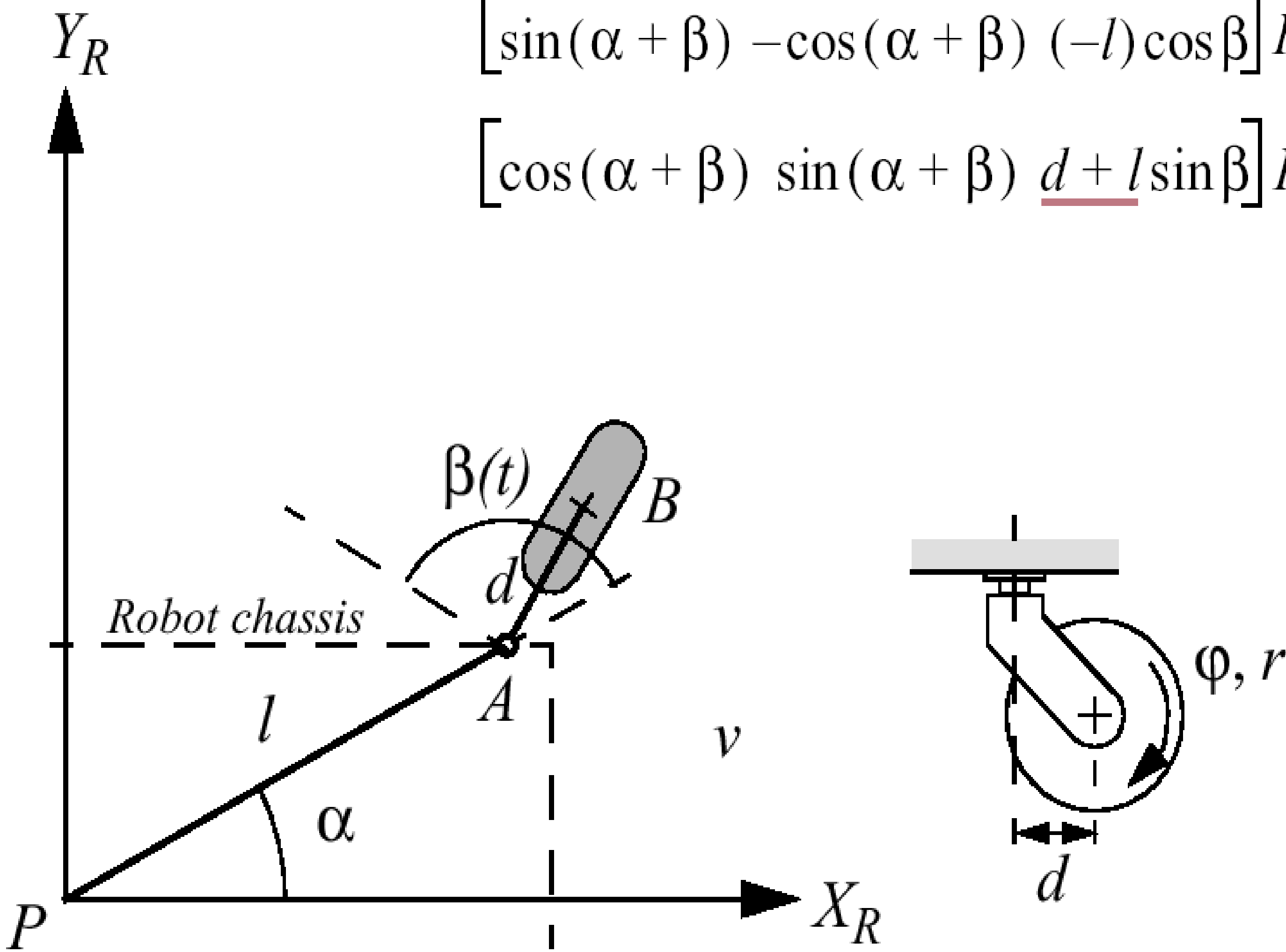




Kinematic Constraints: Castor Wheel

$$\left[\sin(\alpha+\beta) \;\; -\cos(\alpha+\beta) \;\; (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

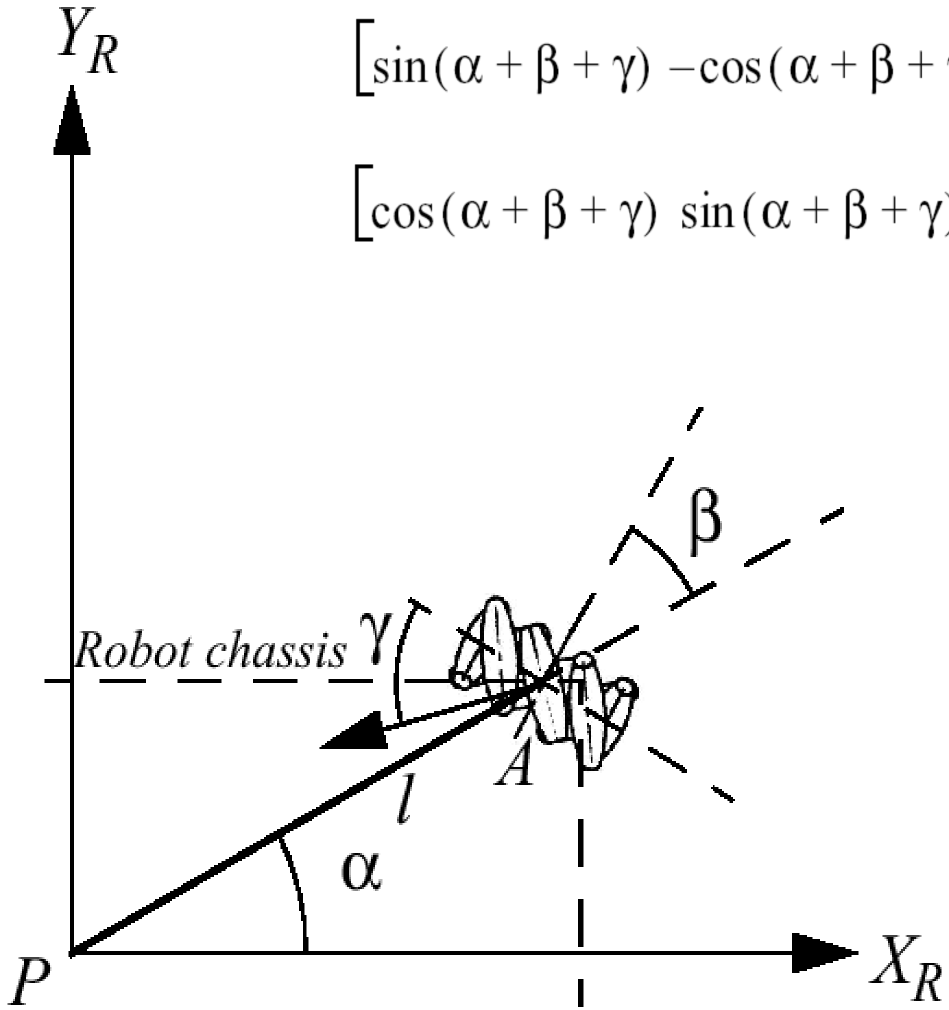
$$\left[\cos(\alpha+\beta) \;\; \sin(\alpha+\beta) \;\; d + l\sin\beta\right] R(\theta)\dot{\xi}_I + d\dot{\beta} = 0$$

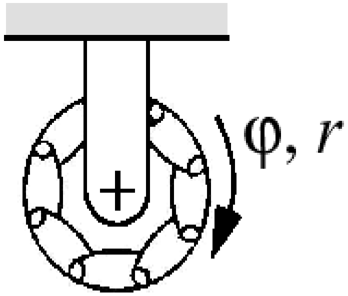


Kinematic Constraints: Swedish Wheel

$$\begin{bmatrix}\sin(\alpha+\beta+\gamma) & -\cos(\alpha+\beta+\gamma) & (-l)\cos(\beta+\gamma)\end{bmatrix} R(\theta)\dot{\xi}_I - r\dot{\varphi}\cos\gamma = 0$$

$$\begin{bmatrix}\cos(\alpha+\beta+\gamma) & \sin(\alpha+\beta+\gamma) & l\sin(\beta+\gamma)\end{bmatrix} R(\theta)\dot{\xi}_I - r\dot{\varphi}\sin\gamma - r_{sw}\dot{\varphi}_{sw} = 0$$

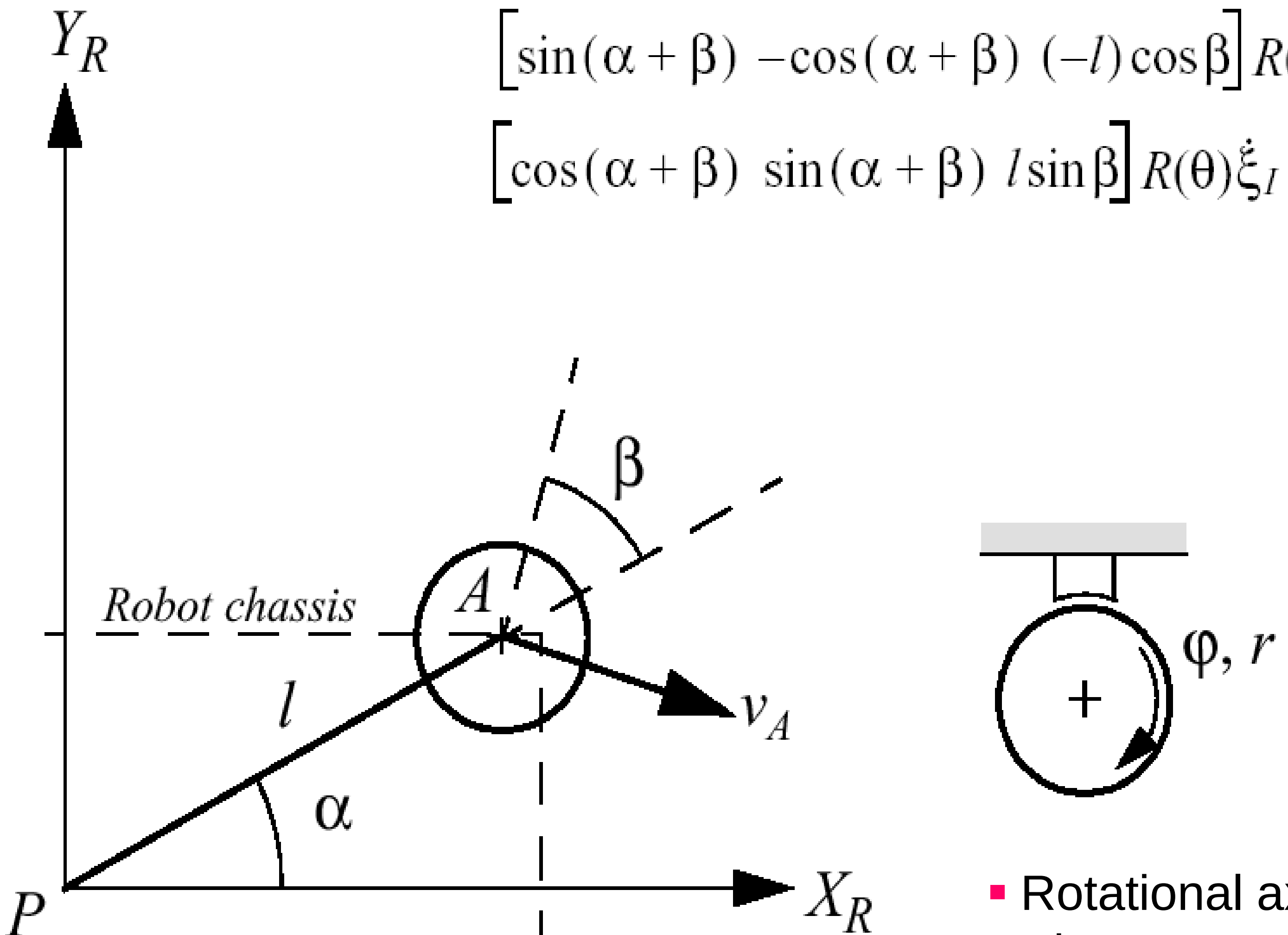




Kinematic Constraints: Spherical Wheel

$$\begin{bmatrix} \sin(\alpha+\beta) & -\cos(\alpha+\beta) & (-l)\cos\beta \end{bmatrix} R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

$$\begin{bmatrix} \cos(\alpha+\beta) & \sin(\alpha+\beta) & l\sin\beta \end{bmatrix} R(\theta)\dot{\xi}_I = 0$$



- Rotational axis of the wheel can have an arbitrary direction

Kinematic Constraints: Complete Robot

- Given a robot with *M* wheels
  - each wheel imposes zero or more constraints on the robot motion
  - **only fixed and steerable standard wheels impose constraints**
- What is the maneuverability of a robot considering a combination of different wheels?

چرخ fix در جهت عمود نمی‌تواند حرکت کند.
و در شرایط خاص می‌تواند در جهت عادی حرکت کند.

چرخ fix   چرخ فرمان

- Suppose we have a total of $N = N_f + N_s$ standard wheels
  - We can develop the equations for the constraints in matrix forms:
  - Rolling

$$J_1(\beta_s)R(\theta)\dot{\xi}_I + J_2\dot{\varphi} = 0 \qquad \varphi(t) = \underbrace{\begin{bmatrix} \varphi_f(t) \\ \varphi_s(t) \end{bmatrix}}_{(N_f+N_s)\times 1} \qquad J_1(\beta_s) = \underbrace{\begin{bmatrix} J_{1f} \\ J_{1s}(\beta_s) \end{bmatrix}}_{(N_f+N_s)\times 3} \qquad J_2 = diag(r_1 \cdots r_N)$$

  - Lateral movement

$$C_1(\beta_s)R(\theta)\dot{\xi}_I = 0 \qquad C_1(\beta_s) = \underbrace{\begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s) \end{bmatrix}}_{(N_f+N_s)\times 3}$$

Mobile Robot Maneuverability

- The maneuverability of a mobile robot is the combination
  - of the mobility available based on the sliding constraints
  - plus additional freedom contributed by the steering

- Three wheels is sufficient for static stability
  - additional wheels need to be synchronized
  - this is also the case for some arrangements with three wheels

- It can be derived using the equation seen before
  - Degree of mobility δ_m حرکت
  - Degree of steerability δ_s فرمان
  - Robots maneuverability $\delta_M = \delta_m + \delta_s$

مانور → هرچه بیشتر باشد
محدودیت حرکتی کمتر

Mobile Robot Maneuverability: Degree of Mobility

- To avoid any lateral slip the motion vector $R(\theta)\dot{\xi}_I$ has to satisfy the following constraints:

$$C_{1f} R(\theta)\dot{\xi}_I = 0 \qquad C_1(\beta_s) = \begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s) \end{bmatrix}$$

$$C_{1s}(\beta_s) R(\theta)\dot{\xi}_I = 0$$

- Mathematically:
  - $R(\theta)\dot{\xi}_I$ must belong to the *null space* of the projection matrix $C_1(\beta_s)$
  - *Null space* of $C_1(\beta_s)$ is the space N such that for any vector n in N

$$C_1(\beta_s) \cdot n = 0$$

نقطه‌ای فرض کن که دور آن حرکت می‌کنیم. در لحظه تعریف شده و لحظه بعد فرق دارد

  - Geometrically this can be shown by the *Instantaneous Center of Rotation (ICR)*

نقطه را در فضا در نظر نگرفتیم و نقطه روی صفحه task space در نظر می‌گیریم.

Mobile Robot Maneuverability: ICR

- Instantaneous center of rotation (ICR)

به زاویه فرمان خطا داده و متناظر روی یک محور حرکت می‌کنند اراسی
چرخ های عقب (اگر چرخ های عقب بچرخند) می‌توانند به
به ترتیب برسند.

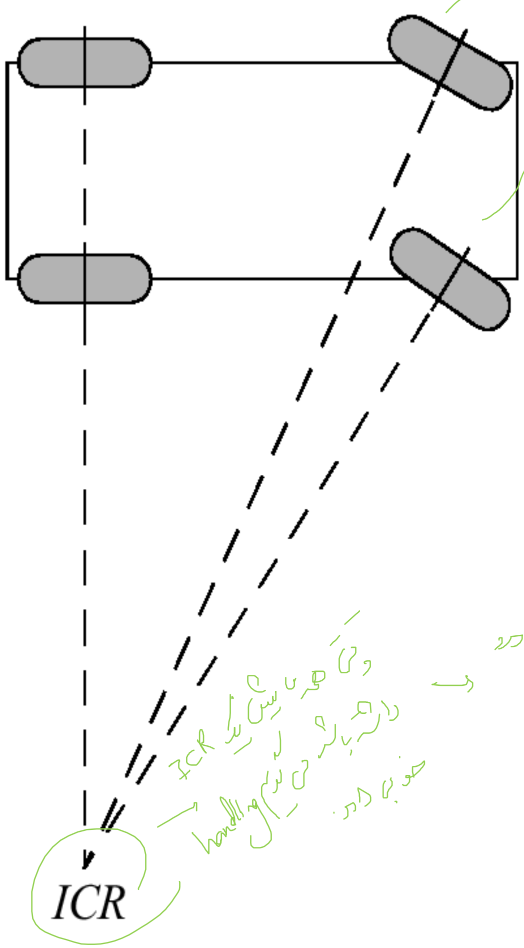
اگر چرخ های عقب هم بتوانند حرکت کنند در مرحله
حرکت کنند درجه آزادی داشته باشند، می‌توانند در
جابه جا ها و دور زدن ها حرکت انسان ها را بهتر انجام
دهند.

- Ackermann Steering

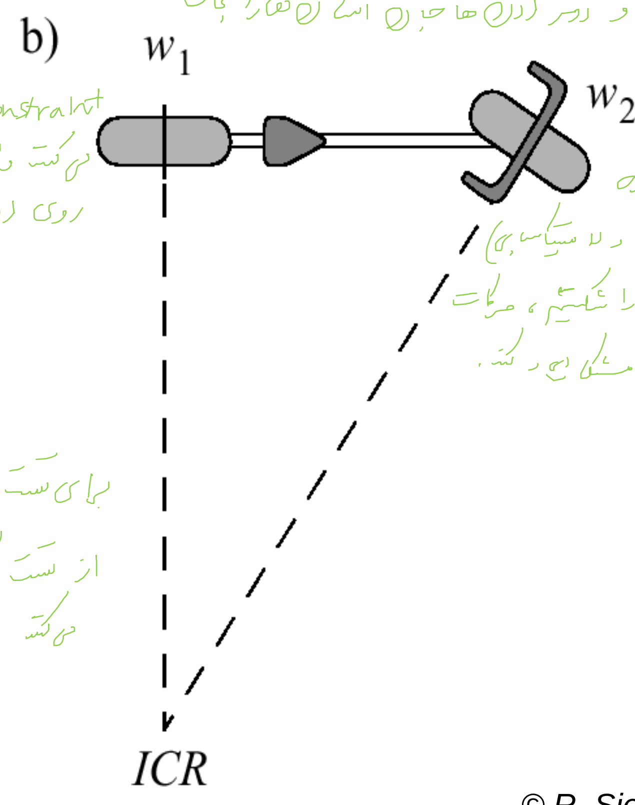
هر دو چرخ نباید به یک اندازه
بچرخند، (به آن handling
می گویند)

سعی می کنند در
خودروهای
جدید بهتر
کنند.

Bicycle



constraint را حذف
می کنند ولی مثلث بین
روی زمین ایجاد می کنند.



همچنین هم نمی‌توانند هر دو چرخ
حرکت کنند چرا که در مصرف انرژی
بهینه نیست. (در مصرف سوخت و لاستیک ها)
همچنین به علت اینکه برخی قسمت ها را نشکستیم، حرکات
پیچیده در سرعت است مشکل ایجاد می کند.

ICR برای هر چرخ و
handling خوب دارد.

برای تست handling خودرو
از تست کوردن استفاده
می کنند.

- ترمز می‌تواند چرخ ها را به مقدار متفاوتی آرام کند و نیرو به آن ها وارد کند. علت آن این است که توزیع نیرو روی چرخ ها یکسان
یکی را شیب و دسته در به تعبیر ما شکل دارد و چرخ در آن باعث سر خوردن می‌شود. علت آن سطح آب و اصطکاک متفاوت سطح زیر
هر چرخ می‌باشد. در سیستم های با یک ترمز کوچک هم می‌تواند حتی به یک چرخ دو یا چهار برابر نیروی بیشتری را اعمال کند.
محور گاردان تا حدی این اختلاف سرعت ها را تحمل می‌کند.

- مزیت معادلی به اهدافی که برای ترکیب نیرو توسط یک سیستم مشابه با دستگاه انجام می‌شود گفته می‌شود. اگر فرض کنیم هر مقدار انرژی که اعمال کنیم باعث مصرف
می‌شود، با تغییر مقدار انرژی با سطح تأثیر انرژی می‌توان روی آن تأثیر گذاشت و با هر کدام تأثیر بیشتری می‌بینیم. مثل قسمتی که با هم ترکیب شده‌اند
شده و روی هم به نسبت تأثیر بیشتری نسبت به هر کدام به تنهایی می‌گذارند.
در انجام ها که تمام انرژی تبدیل به حرکت شود ما چیزی می‌دهیم و چیزی می‌گیریم. مثل جایی که همه انرژی تبدیل به حرکت نمی‌شود و مثلاً پوست دست کنده می‌شود
مثال دیگر شبیه به مقاومت است.
اما در ترمز مزیت معادلی به فرصت نوع رها یا پخش شکل می‌گیرد و از خلاء موقت استفاده می‌شود و باعث حرکت شده که با فشار کاری حرکت می‌کنند.
در سیستم هیدرولیک شبیه به روغن ترمز باعث هل دادن و کمک به حرکت ماشین می‌شود.

- فرمان هیدرولیک یا powered steering. دیگر فرمان را هیدرولیک در نظر نمی‌گیرند نیاز داریم. مشکل این فرمان ها مشکلات بیشتر از اندازه است
که در صورت وجود مشکل با کمترین حرکت و کمترین واکنش باعث حرکت خیلی زیادی می‌شود (over steering) یا under steering
برای حل مشکل از فرمان برقی یا drive by wire استفاده می‌شود که با استفاده از encoder تشخیص حرکت انجام شده و به میزان درست توسط موتور
فرمان مورد نیاز را اجرا می‌کند. فرمان برقی حرکت را کنترل کرده و در صورت مشکل مقاومت می‌کند.

Mobile Robot Maneuverability: More on Degree of Mobility

- Robot chassis kinematics is a function of the set of independent constraints

$$rank\left[C_1(\beta_s)\right] \qquad C_1(\beta_s)=\begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s) \end{bmatrix} \qquad \begin{matrix} C_{1f}R(\theta)\dot{\xi}_I = 0 \\ C_{1s}(\beta_s)R(\theta)\dot{\xi}_I = 0 \end{matrix}$$

  - the greater the rank of $C_1(\beta_s)$ the more constrained is the mobility

- Mathematically

$$\delta_m = \dim N\left[C_1(\beta_s)\right] = 3 - rank\left[C_1(\beta_s)\right] \qquad 0 \le rank\left[C_1(\beta_s)\right] \le 3$$

  - no standard wheels $rank\left[C_1(\beta_s)\right]=0$
  - all direction constrained $rank\left[C_1(\beta_s)\right]=3$

- Examples:
  - Unicycle: One single fixed standard wheel
  - Differential drive: Two fixed standard wheels
    - wheels on same axle
    - wheels on different axle

Mobile Robot Maneuverability: Degree of Steerability

- Indirect degree of motion

$$\delta_s = rank\left[C_{1s}(\beta_s)\right]$$

  - The particular orientation at any instant imposes a kinematic constraint
  - However, the ability to change that orientation can lead additional degree of maneuverability
- Range of δ_s: $0 \le \delta_s \le 2$

- Examples:
  - one steered wheel: Tricycle
  - two steered wheels: No fixed standard wheel
  - car (Ackermann steering): $N_f = 2$, $N_s = 2$ -> common axle

Mobile Robot Maneuverability: Robot Maneuverability

- Degree of Maneuverability

$$\delta_M = \delta_m + \delta_s$$

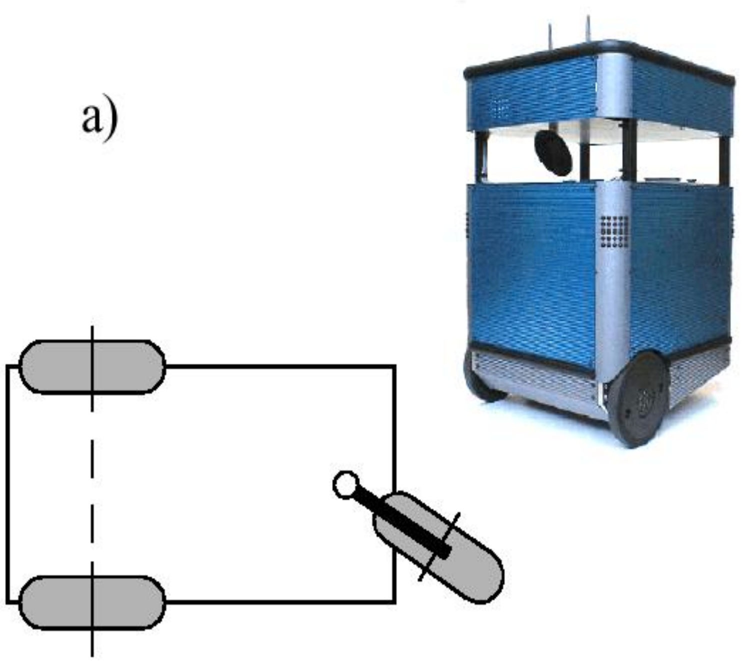
  - Two robots with same δ_M are not necessary equal
  - Example: Differential drive and Tricycle (next slide)

  - For any robot with $\delta_M = 2$ the ICR is always constrained to *lie on a line*
  - For any robot with $\delta_M = 3$ the ICR is not constrained and can *be set to any point on the plane*

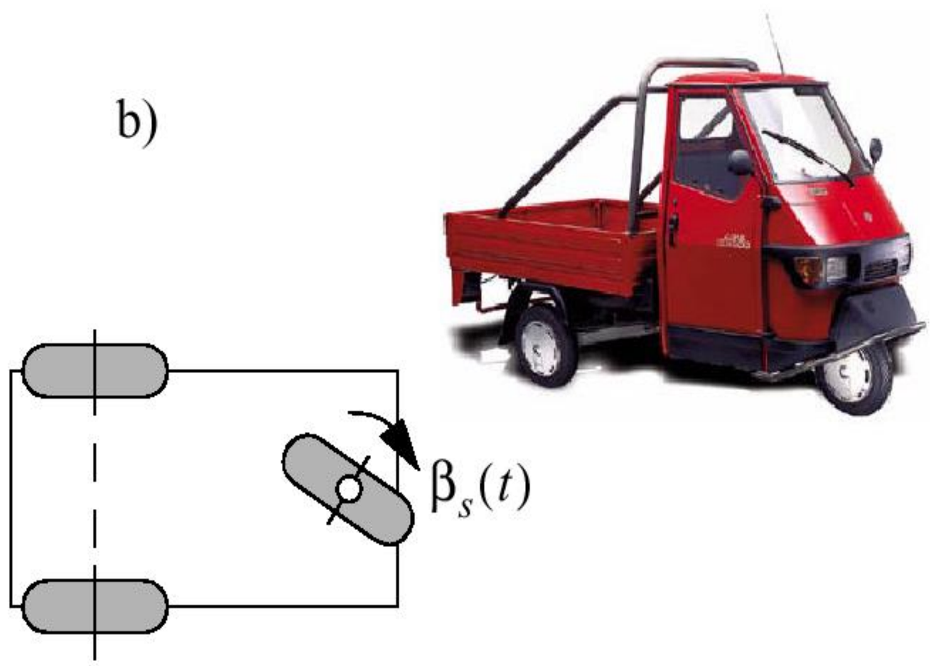
- The Synchro Drive example: $\delta_M = \delta_m + \delta_s = 1 + 1 = 2$

Mobile Robot Maneuverability: Wheel Configurations

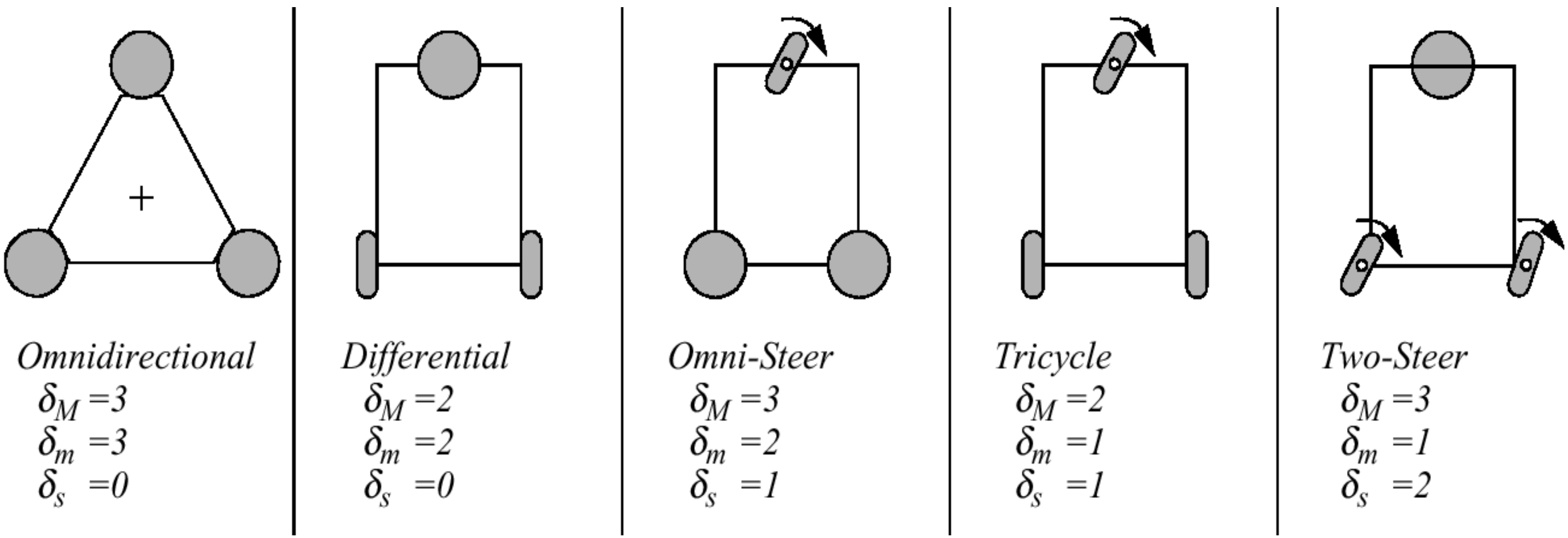
- Differential Drive



Tricycle

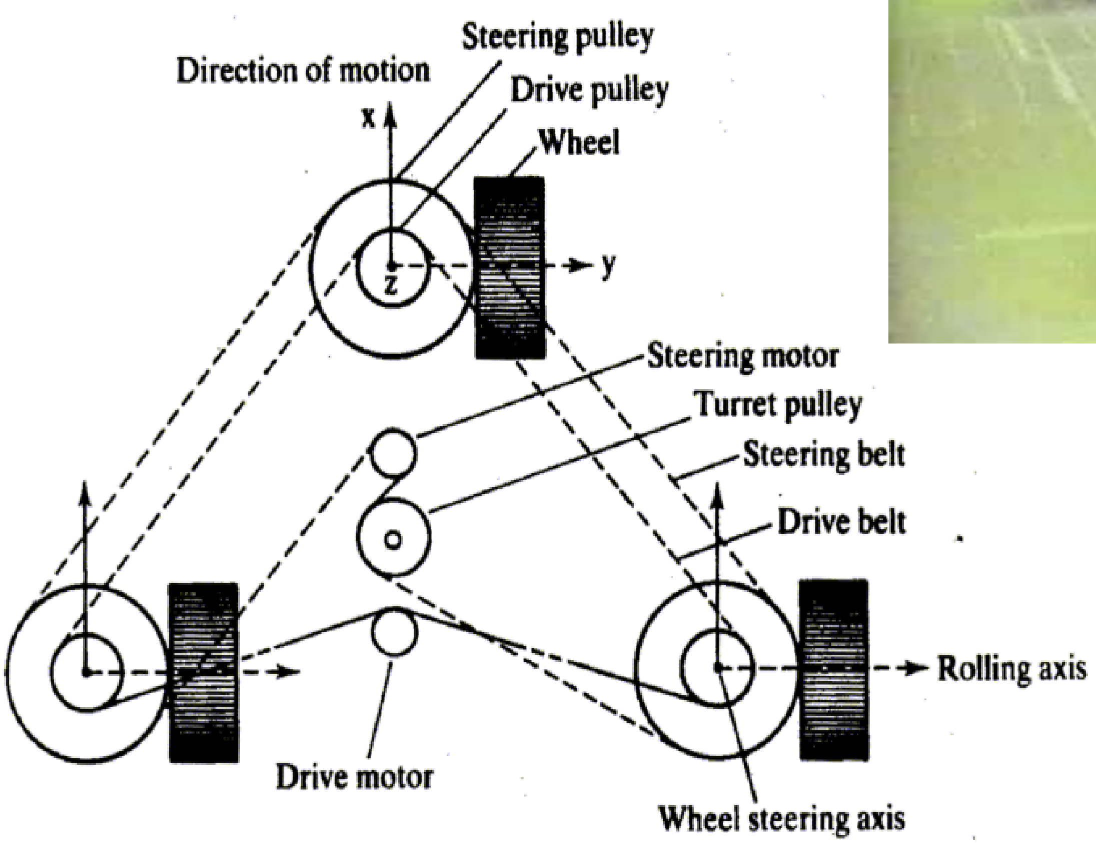


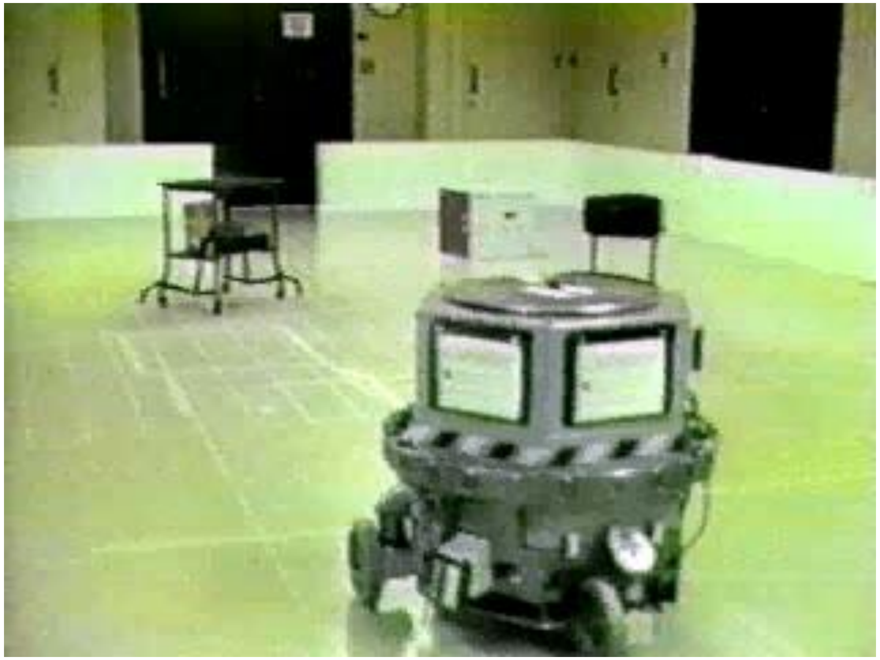
Five Basic Types of Three-Wheel Configurations



Synchro Drive

$$\delta_M = \delta_m + \delta_s = 1 + 1 = 2$$





C J. Borenstein

29 Mobile Robot Workspace: Degrees of Freedom

- The Degree of Freedom (DOF) is the robot's ability to achieve various poses.

- But what is the degree of vehicle's freedom in its environment?
  - Car example

- Workspace
  - how the vehicle is able to move between different configuration in its workspace?

- The robot's independently achievable velocities
  - = *differentiable degrees of freedom (DDOF) =* δ_m
  - Bicycle: $\delta_M = \delta_m + \delta_s = 1 + 1$ DDOF = 1; DOF=3
  - Omni Drive: $\delta_M = \delta_m + \delta_s = 3 + 0$ DDOF=3; DOF=3

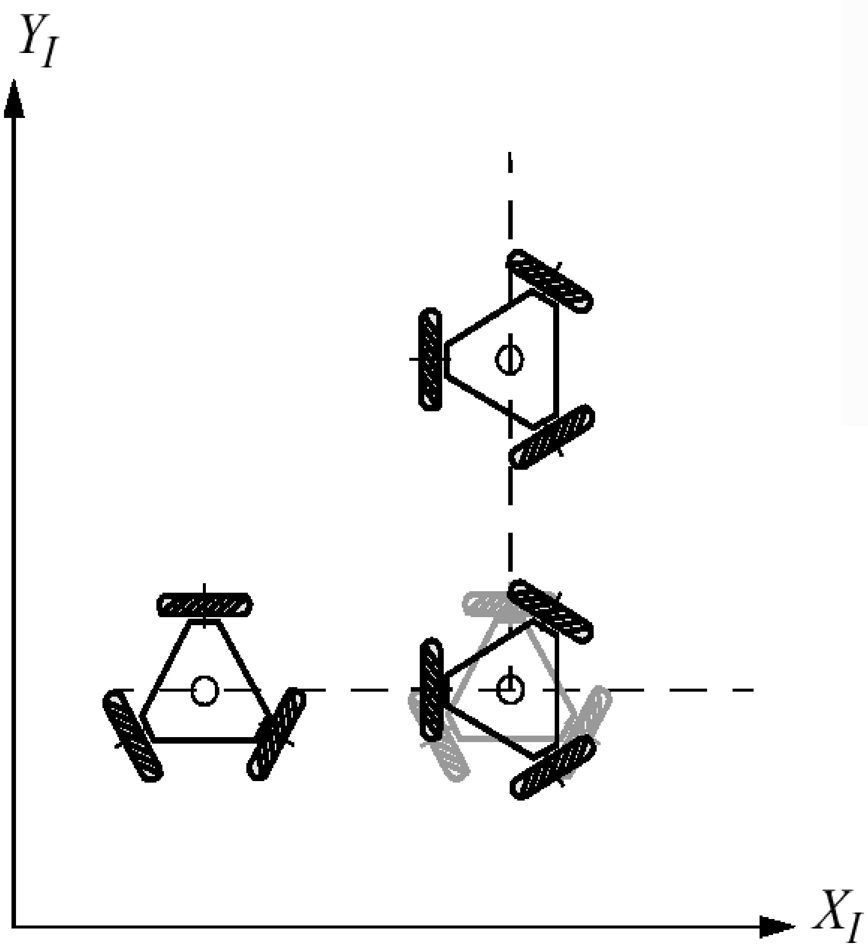
Mobile Robot Workspace: Degrees of Freedom, Holonomy

- DOF *degrees of freedom*:
  - Robots ability to achieve various poses
- DDOF *differentiable degrees of freedom*:
  - Robots ability to achieve various trajectories

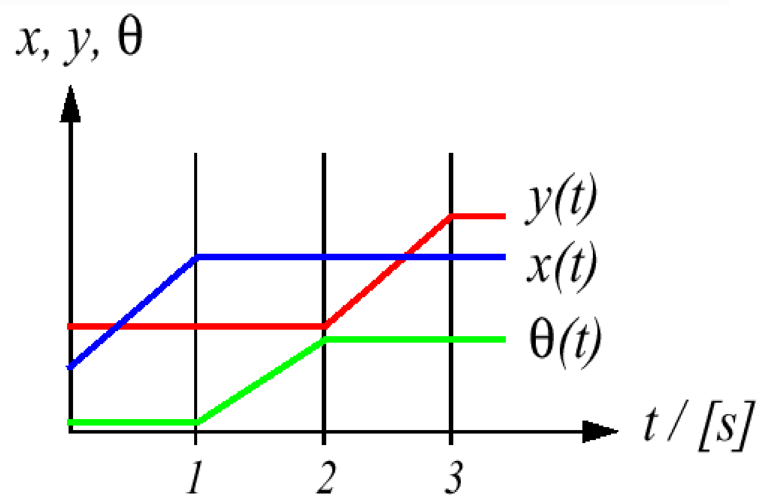
$$DDOF \leq \delta_M \leq DOF$$

- Holonomic Robots
  - A holonomic kinematic constraint can be expressed as an explicit function of position variables only
  - A non-holonomic constraint requires a different relationship, such as the derivative of a position variable
  - *Fixed and steered standard wheels impose non-holonomic constraints*

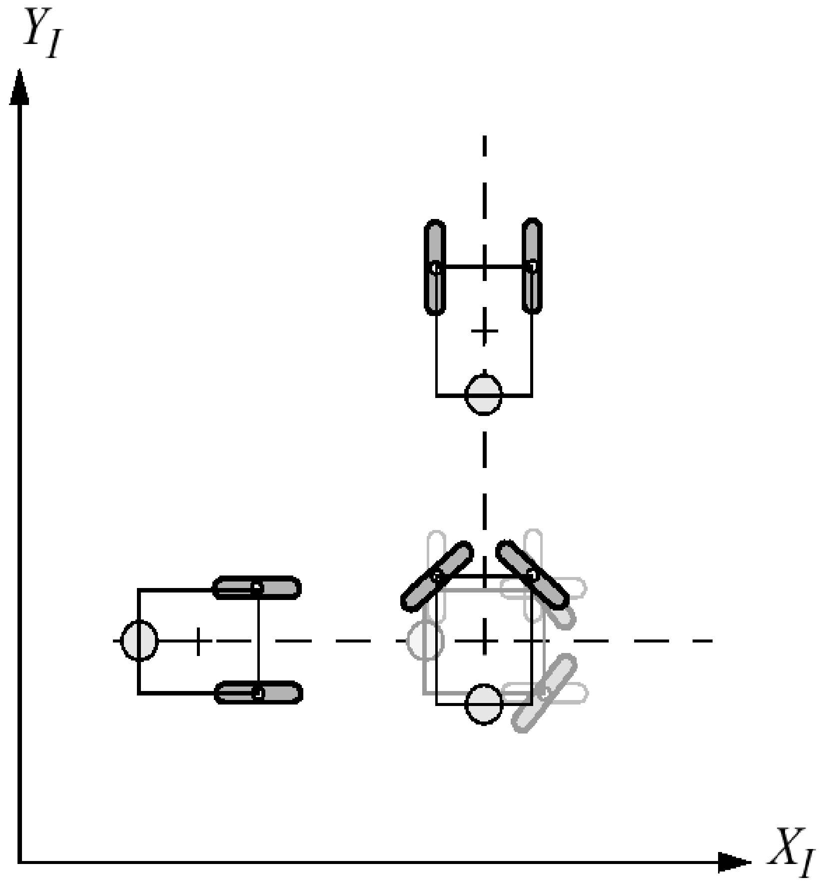
Path / Trajectory Considerations: Omnidirectional Drive

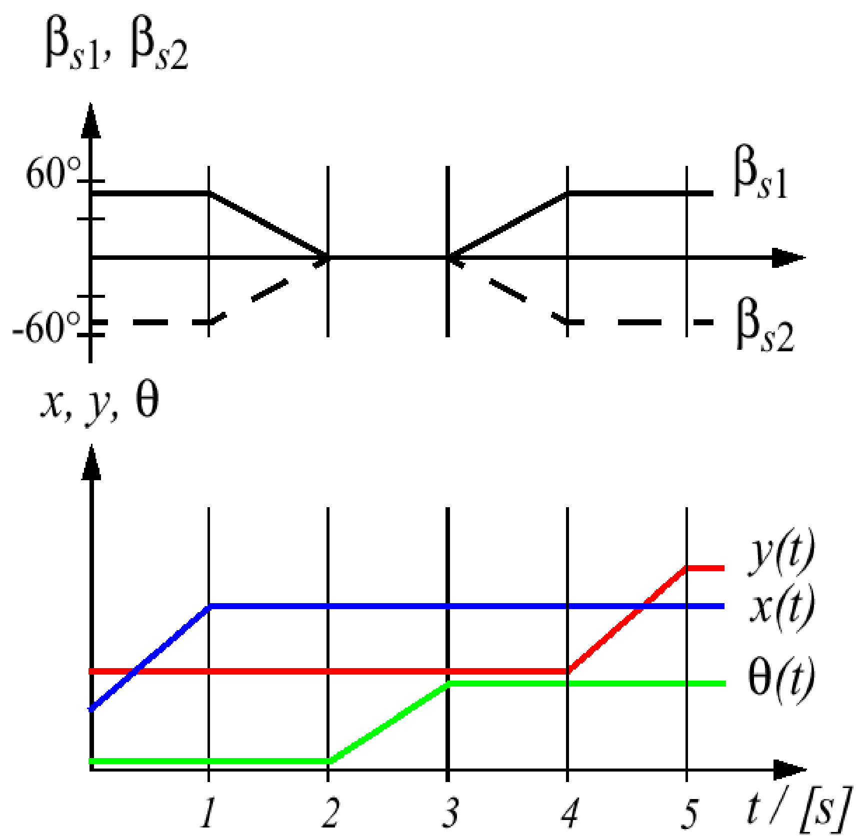


Tribolo ETH/EPFL



Path / Trajectory Considerations: Two-Steer



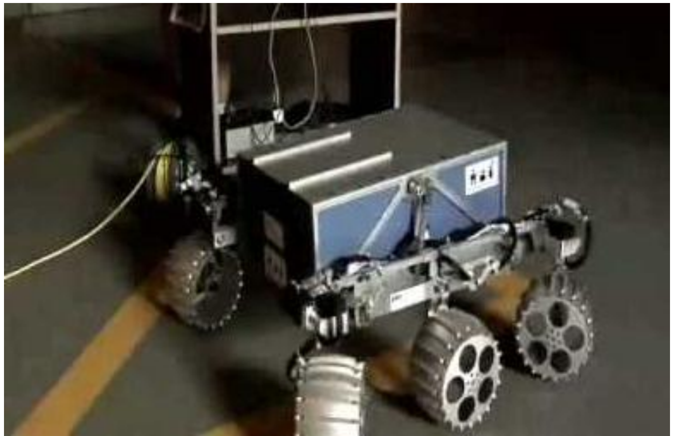


Beyond Basic Kinematics

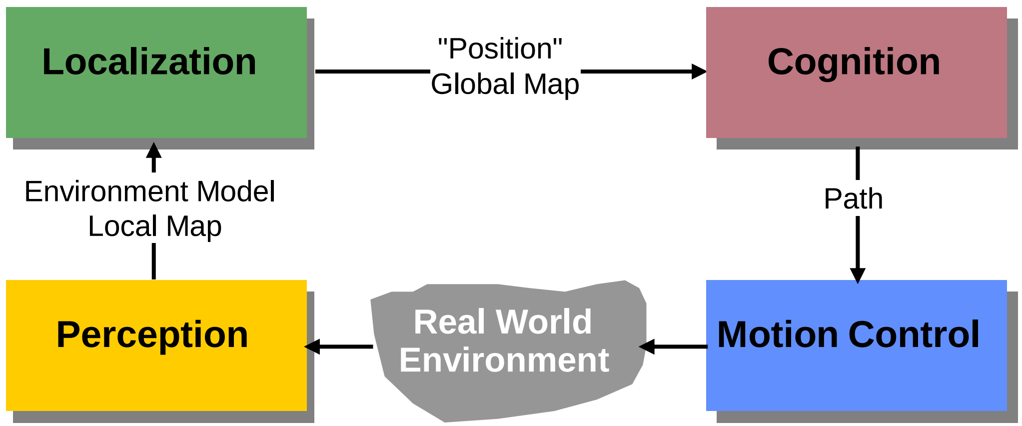
- At higher speeds, and in difficult terrain, dynamics become important

C Stanford University

- For other vehicles, the no-sliding constraints, and simple kinematics presented in this lecture do not hold



C ito-germany.de



Motion Control

wheeled robots

Wheeled Mobile Robot Motion Control: Overview

- The objective of a kinematic controller is to follow a trajectory described by its position and/or velocity profiles as function of time.
- Motion control is not straight forward because mobile robots are typically non-holonomic and MIMO systems.
- Most controllers (including the one presented here) are not considering the dynamics of the system

path planned → motion planned → motion control

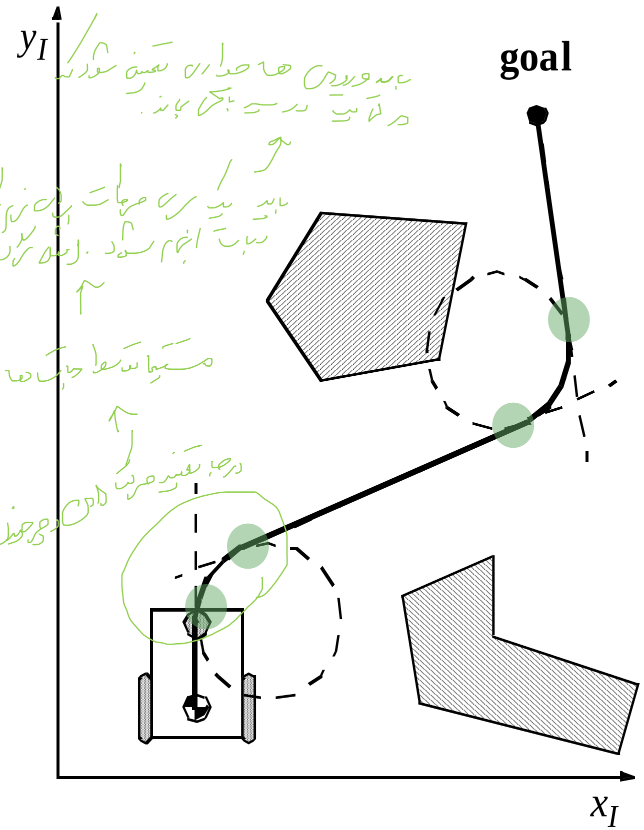
Motion Control: Open Loop Control

- trajectory (path) divided in motion segments of clearly defined shape:
  - straight lines and segments of a circle
  - Dubins car, and Reeds-Shepp car
- control problem:
  - pre-compute a smooth trajectory based on line, circle (and clothoid) segments
- Disadvantages:
  - It is not at all an easy task to pre-compute a feasible trajectory
  - limitations and constraints of the robots velocities and accelerations
  - does not adapt or correct the trajectory if dynamical changes of the environment occur.
  - The resulting trajectories are usually not smooth (in acceleration, jerk, etc.)

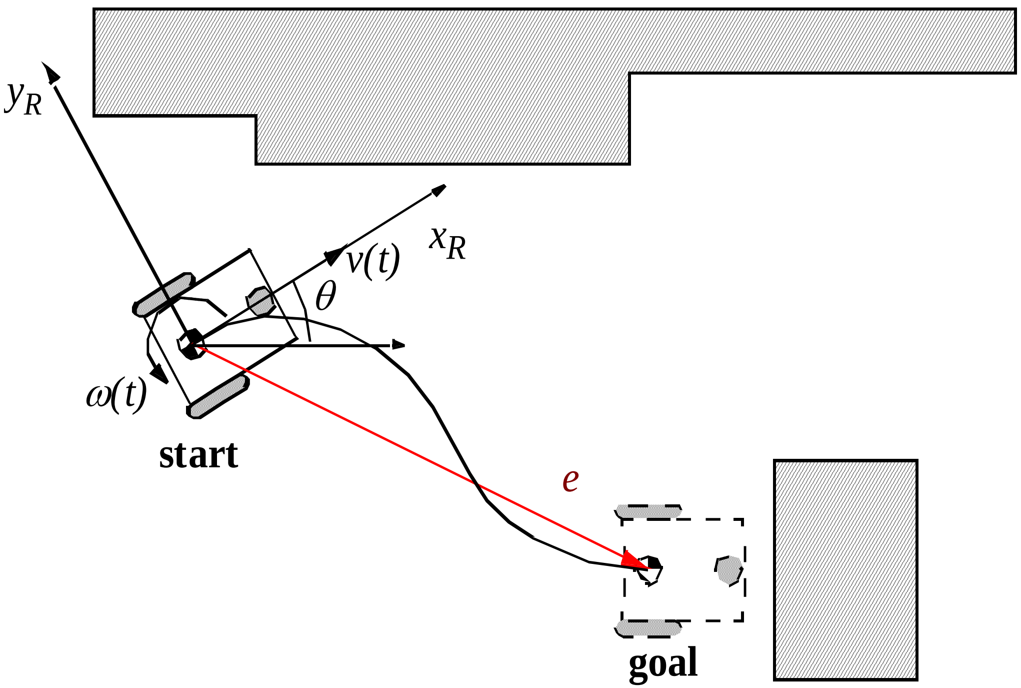
باید یک سری محاسبات از قبل ... نسبت به ... انجام شود. (مسیر کلی ...)

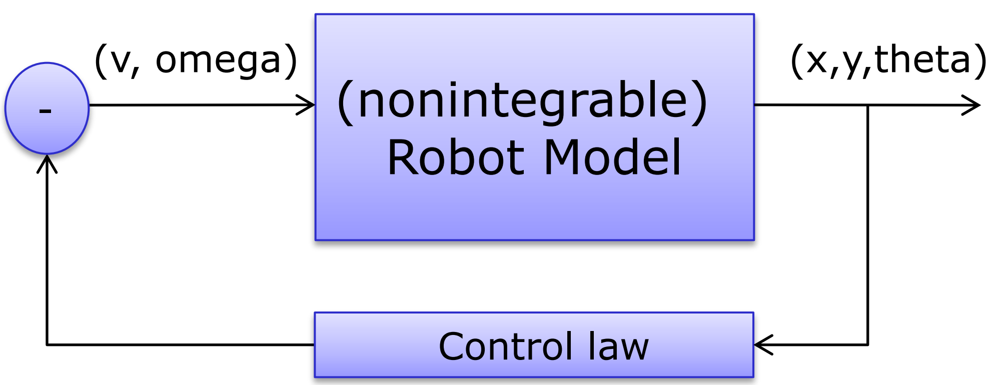
باید ورودی ها جوری تعیین شود که در نهایت درست ... باشد.

مستقیما ... قابل انجام نیست



Motion Control: Feedback Control





- Find a control matrix *K*, if exists

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \end{bmatrix}$$

with $k_{ij}=k(t,e)$

- such that the control of *v(t)* and $\omega(t)$

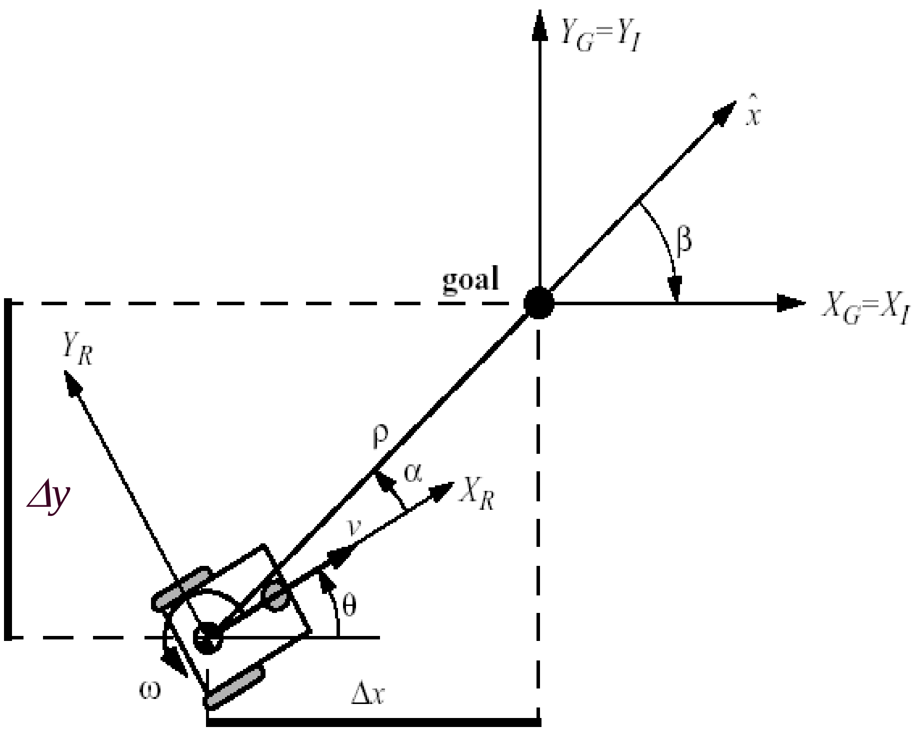
$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = K \cdot e = K \cdot {}^{R}\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

- drives the error e to zero

$$\lim_{t \to \infty} e(t) = 0$$

- MIMO state feedback control

Motion Control: Kinematic Position Control



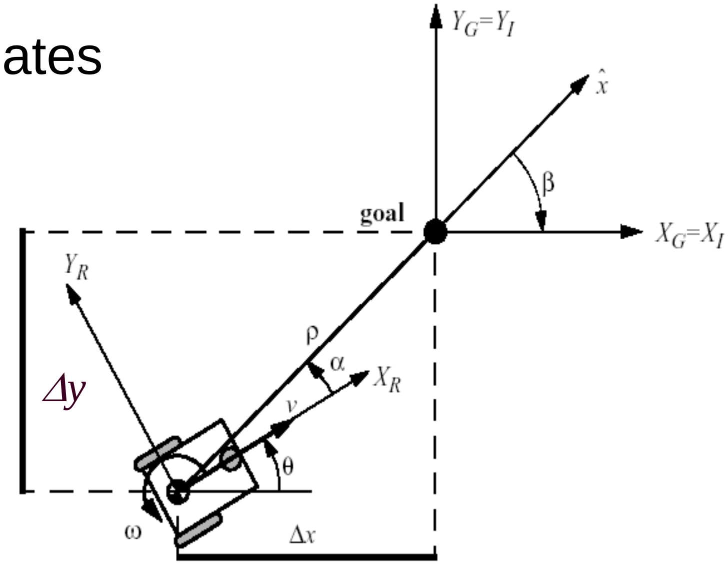
- The kinematics of a differential drive mobile robot described in the inertial frame $\{x_I, y_I, \theta\}$ is given by,

$$^{I}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

- where $\dot{x}$ and $\dot{y}$ are the linear velocities in the direction of the x_I and y_I of the inertial frame.
- Let alpha denote the angle between the x_R axis of the robots reference frame and the vector connecting the center of the axle of the wheels with the final position.

Kinematic Position Control: Coordinates Transformation

- Coordinates transformation into polar coordinates with its origin at goal position:



$$\rho = \sqrt{\Delta x^2 + \Delta y^2}$$

$$\alpha = -\theta + \operatorname{atan2}(\Delta y, \Delta x)$$

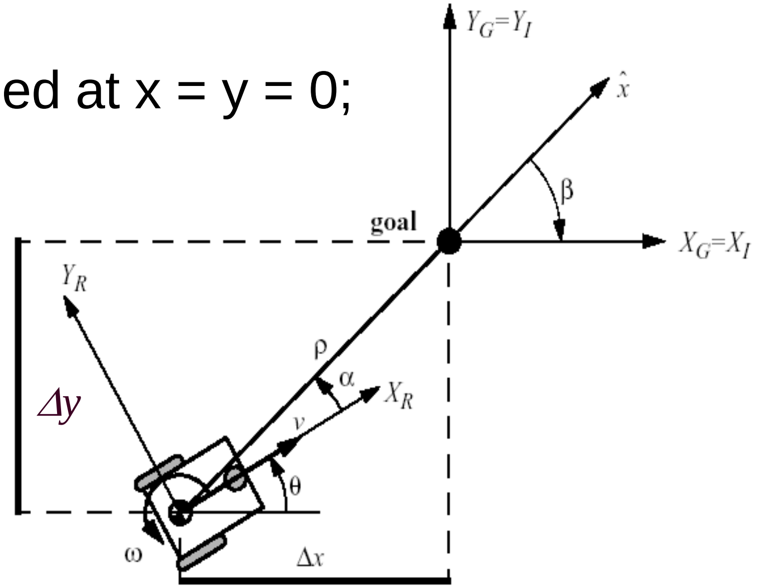
$$\beta = -\theta - \alpha$$

- System description, in the new polar coordinates

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos\alpha & 0 \\ \dfrac{\sin\alpha}{\rho} & -1 \\ -\dfrac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \qquad \begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 \\ -\dfrac{\sin\alpha}{\rho} & -1 \\ \dfrac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

$$\textit{for} \quad I_1 = \left(-\frac{\pi}{2}, \frac{\pi}{2}\right] \qquad \textit{for} \quad I_2 = (-\pi, -\pi/2] \cup (\pi/2, \pi]$$

Kinematic Position Control: Remarks



- The coordinates transformation is not defined at x = y = 0;
- For $\alpha \in I_1$ the forward direction of the robot points toward the goal, for $\alpha \in I_2$ it is the backward direction.

$$\alpha \in I_1 = \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$$

- By properly defining the forward direction of the robot at its initial configuration, it is always possible to have $\alpha \in I_1$ at t=0. However this does not mean that a remains in I_1 for all time t.

Kinematic Position Control: The Control Law

- It can be shown, that with

$$v = k_\rho \rho \qquad \omega = k_\alpha \alpha + k_\beta \beta$$

the feedback controlled system

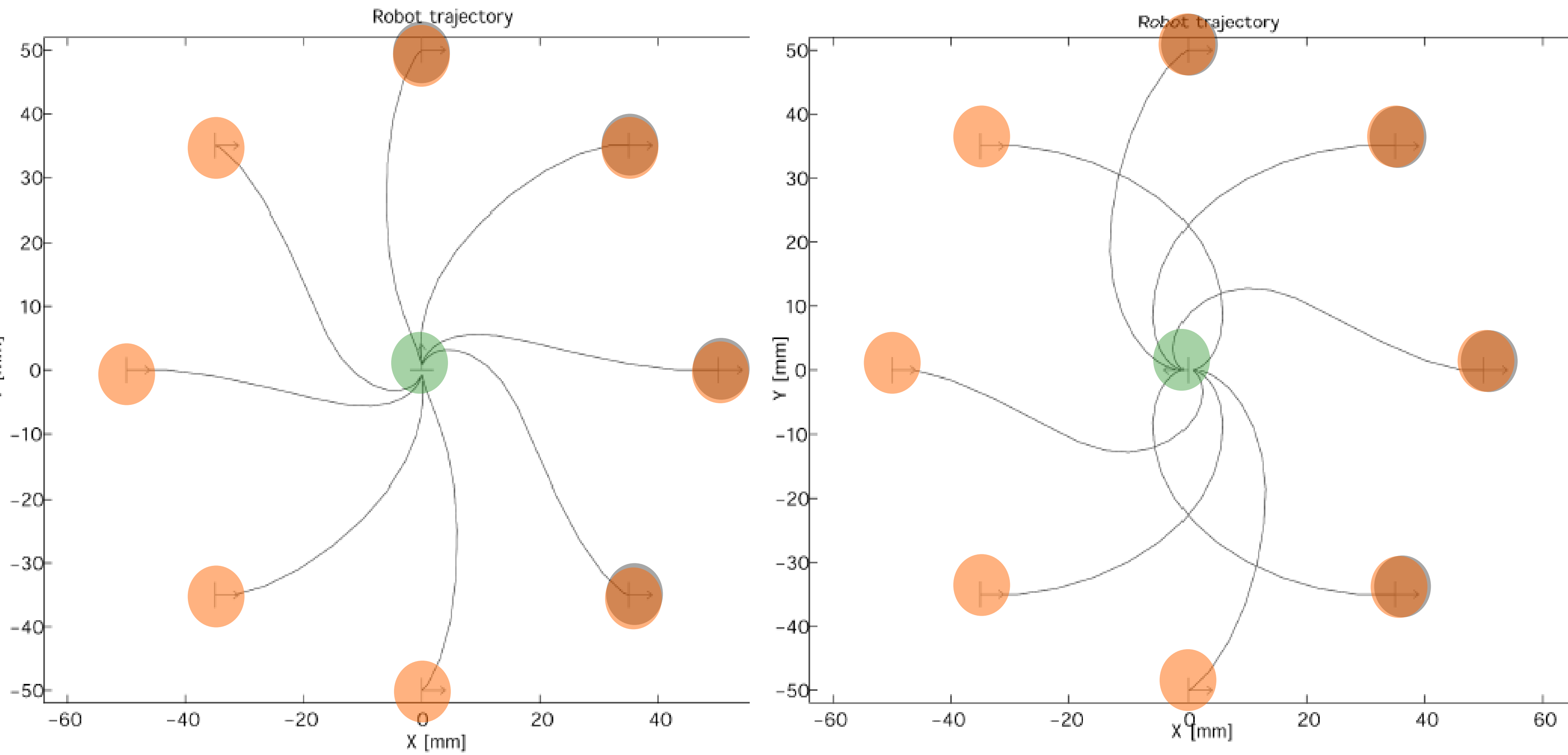
$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho \rho \cos\alpha \\ k_\rho \sin\alpha - k_\alpha \alpha - k_\beta \beta \\ -k_\rho \sin\alpha \end{bmatrix}$$

will drive the robot to $(\rho,\alpha,\beta)=(0,0,0)$

- The control signal v has always constant sign,
  - the direction of movement is kept positive or negative during movement
  - parking maneuver is performed always in the most natural way and without ever inverting its motion.

Kinematic Position Control: Resulting Path

- The goal is in the center and the initial position on the circle.



$$k = (k_\rho, k_\alpha, k_\beta) = (3,8,-1.5)$$

Kinematic Position Control: Stability Issue

- It can further be shown, that the closed loop control system is locally exponentially stable if

$$k_\rho > 0 \;\; ; \;\; k_\beta < 0 \;\; ; \;\; k_\alpha - k_\rho > 0$$

$$\mathrm{k} = (\mathrm{k}_\rho, \mathrm{k}_\alpha, \mathrm{k}_\beta) = (3, 8, -1.5)$$

- Proof:
  for small x –> $\cos x = 1$, $\sin x = x$

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho & 0 & 0 \\ 0 & -(k_\alpha - k_\rho) & -k_\beta \\ 0 & -k_\rho & 0 \end{bmatrix} \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix} \qquad A = \begin{bmatrix} -k_\rho & 0 & 0 \\ 0 & -(k_\alpha - k_\rho) & -k_\beta \\ 0 & -k_\rho & 0 \end{bmatrix}$$

  and the characteristic polynomial of the matrix A of all roots

$$(\lambda + k_\rho)(\lambda^2 + \lambda(k_\alpha - k_\rho) - k_\rho k_\beta)$$

  have negative real parts.

Sensors

Ahad Harati

What is a sensor?

- anything that measures some quantity or detects the state of the environment (including the robot)
  - Oxford Dic.: a device giving a signal for the detection or measurement of a physical property to which it responds.
  - Another Def.: a device that receives a signal or stimulus and response with an electrical signal.

Classification of sensors

- Stimulus چه پدیده فیزیکی؟
  - Electrical, Magnetic, Mechanical, Optical, Thermal, Chemical, Radiation
- Working Principle برچه اساسی؟ (اساس کاری) ـ فتورزیستورها ـ تغییر مقاومت برای تابش نور ـ
  مثل ترموکوپل یا دماسنج

  تحت چه پدیده‌ای
  محاسبه کرد
  - Eg. strain is measured as change of resistance (voltage) resulted from small displacements
- Properties ویژگی‌ها و بازه‌ها ـ مشخصات
  - Like working range or precision
- Application کاربرد

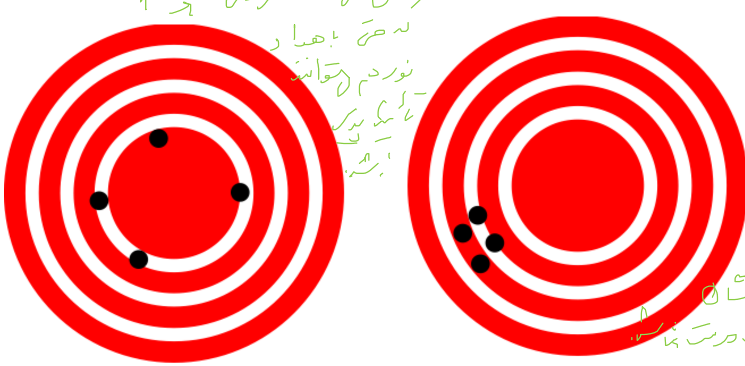
Measurement

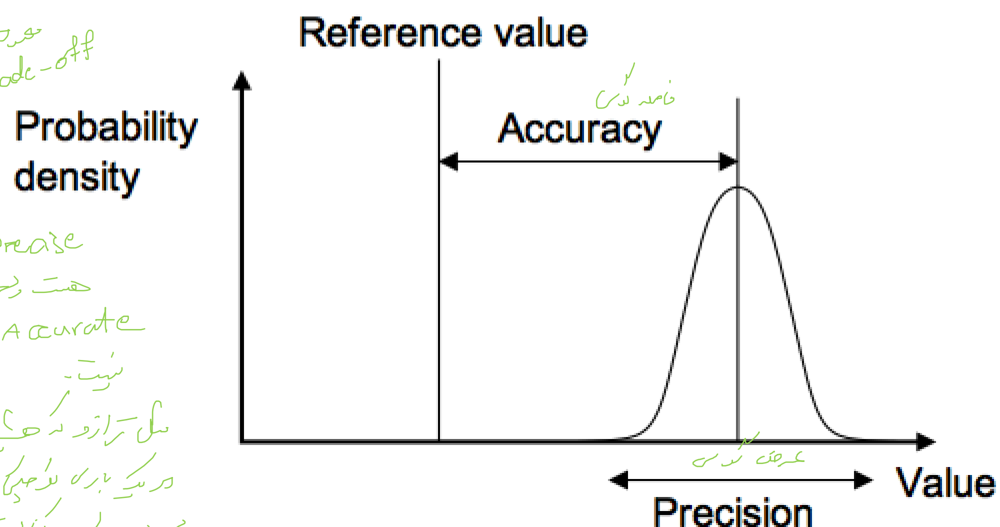
- Bias (deterministic errors)
- Noise (random errors)
- Signal to noise ratio (SNR)
- Precision
- Accuracy

Accurate

precise

trade-off





precise

Accurate

Sensor Characteristics

- **Sensitivity:** change in output for unit change of input
- **Resolution:** smallest detectable change of input
- **Range:** acceptable input interval
- **Linearity:** closeness of characteristic curve to a straight line
- **Drift:** deviation from null reading when input is kept constant long enough (affects accuracy)
- **Hysteresis:** dependence of output to the direction of input change (increase/decrease)
- **Repeatability:** maximum deviation from the average in repeated measurements (relates to precision)
- **Dynamic Characteristics:** transient response to a step input
  - Delay, Rise time, Fall time, Peak time, Settling time, Overshoot percentage
- **Other:** packaging and physical characteristics, power consumption, different limitations or compensations …

- sensitivity:

- resolution پاسخ دقیق

- range

- linearity ← به معنای خطی بودن سنسورهاست که در این بازه‌ها بنابراین بازه عملیاتی سنسورها از قسمت خطی نمودار صد درصد برداشت می‌شود. (یعنی بازه اصلی قابل استفاده سنسورها معمولا بخشی از بازه تعریف شده است) - مانند سنسورهای مثل دما محدود در هر چیزی تا بتوان کالیبره استفاده کرد در صورت تحمل است محدود تا بتوان تاثیر کار را پیدا کرد را حذف کرد.

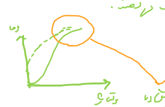
- drift ← مثل با گذشت زمان map که حالت استهلاک در خود سنسور به وجود می‌آید. به علت پارامترهای محیطی که در زمان زیاد عدد آن شاید بهم ریخته شود -

- **hysteresis** ← درصورت ایده‌آل صفر است. معمولا در نمودار روبه‌رو برای نشان دادن آن استفاده می‌شود.

← اگر نمودار متقارن باشه یعنی hysteresis نداریم.

مثال‌ها:

- مثل ترازوهای فنری که با گذاشتن وزنه زود میره پایین ولی دیر بالا میاد.
- مثل سیم پیچ که اگر آهنی [illegible] میدان زیاد می‌شه تا حد اشباع. مغناطیسی [illegible] از دست می‌دهد [illegible] به علت [illegible] از دست [illegible] (مثلا [illegible] خاصیت مغناطیسی [illegible].)
  اما اگر به جای آهن از چیزی مثل فولاد استفاده کنیم خاصیت مغناطیسی خیلی دیرتر از دست می‌دهند.
- مثل اتو که دماش بالا رفته دیرتر انجام می‌شود تا پایین آمدن آن.



هنگام بالا رفتن دما
متفاوت هم با هم دارند

علت اینکه لامپ‌های LED که نیاز به مقاومت دارند [illegible]
مستقیم به برق زد [illegible] نیاز به مقاومت دارند که دما خیلی
بالا نرفت، و از آن [illegible] نگه داشته شود

- **repeatability** ← در رباتیک خیلی مهم است مثل کار ربات در pick-and-place.

- **dynamic characteristics**

- **others** ← تابع تبدیل - [illegible] - سنسور در جایی که می‌خواهیم [illegible] - [illegible] در جایی که هوا [illegible].

Sensing as Perception

اطلاعات داخلی ربات را می‌دهد. مثلاً خودش را درک می‌کند مثل انسان که می‌داند دستش مشت است یا نه ولی این دانش برای سیستم‌ها [illegible] لازم نیست.

- Proprioception
  - Perception of self-movement and internal states

    encoder موجود در چرخ — دور سنج

    [illegible]

  - Wheel encoders, tachometers, Inertial Measurement Units (IMUs) …

    مجموعه‌ای از چند سنسور برای کنترل و تعادل ربات. مثل سیستم گوش میانی که تعادل را حفظ می‌کند.

    مثل اینکه موبایل‌ها حالت افقی و عمودی را دارند اگر یک رفتیم به حالت Hybernate

- Exteroception
  - Perception of external stimuli or objects
  - Range finders, contact switches, cameras …

    که مثل موبایل که برای تنظیم میزان نور صفحه باید نور محیط را بداند.

Typical Robot Sensors

- Manipulators
  - Joint encoders (proprioception used for forward kinematics)
  - Strain gages or other force sensors
  - Tactile sensors (usually in hand)
- Mobile Robots
  - Wheel encoders and IMUs (dead reckoning used for localization)
  - Proximity sensors (to avoid collisions)
    limit switch – contact switch
  - Range sensors, Cameras (range Camera – event Camera – thermal Camera)
    → IR – laser → scanner

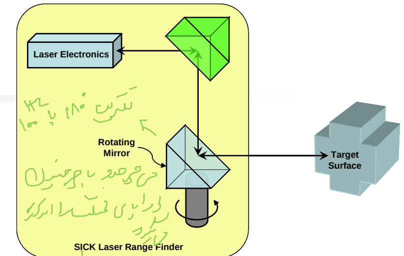
Sensing for Localization and Mapping

- Dead Reconing → Internal Configuration state حالت را اندازه گیری می‌کنند مثل چرخ‌ها
  - Wheel Encoders
  - IMUs
- Measuring Range → فاصله سنجی
  - 2D/3D Laser Scanner
  - Stereo Vision → مثل چشم انسان یا حیوانات شکاری که چشم‌ها کنار هم هستند. مثل عقاب (برخلاف حیوانات شکار شونده که چشم‌ها دو طرف سر هست ۳۶۰ درجه (مثل زرافه)) — از دو دوربین کار می‌گیرد برای تشخیص فاصله استفاده می‌شود. متناسب با فاصله عمق مشخص می‌شود (فاصله، disparity به این صورت فاصله را حساب می‌کند) — با یک دوربین هم می‌توان به راحتی فاصله را حساب کرد. راه‌های زیادی وجود دارد ولی سخت است ۱ از حرکت سنسور تغییر را دیده فاصله را می‌گیرد ۲ depth-from-defocuse
  - Range Camera
- Measuring the bearing → زاویه سنجی مثل قطب‌نما یا camera
  - Gyro, Inclinometer, etc.
  - Camera
  - Compass
- GPS
- RFID → Identification — مثل کارت دانشجویی — برای ورود به سالن‌ها و درب‌ها وسیله را بفهمد

Range Sensors

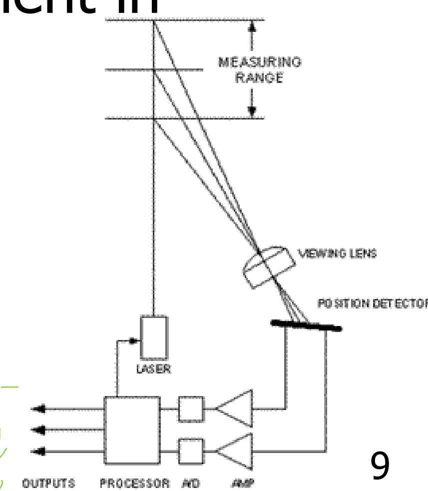
- RADAR (Radio Detection and Ranging)
  - TOF, Long range, usually metallic targets
- LIDAR (Light Detection and Ranging)
  - TOF, Medium range, usually based on Laser, pulse and phase modulation

time of flight



- Sonar
  - TOF, Short range (acoustic waves scatter fast) or simple environment (depth measurement in ocean)

- Modulated Infra red
  - TOF
  - PSD



- Range Camera
  - TOF
  - Structured Light

CCD → Intensity

Kinect

Swiss ranger

unambiguous range

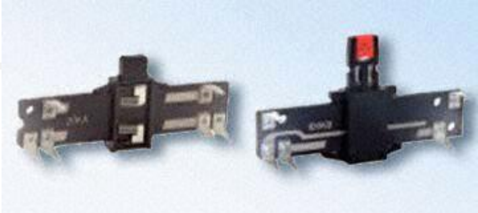
unique

stereo

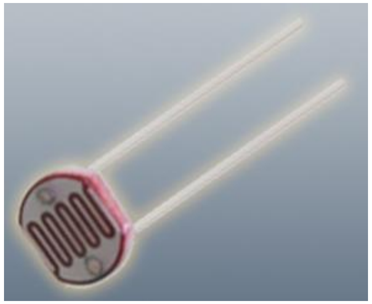
Some Other Types of Sensors

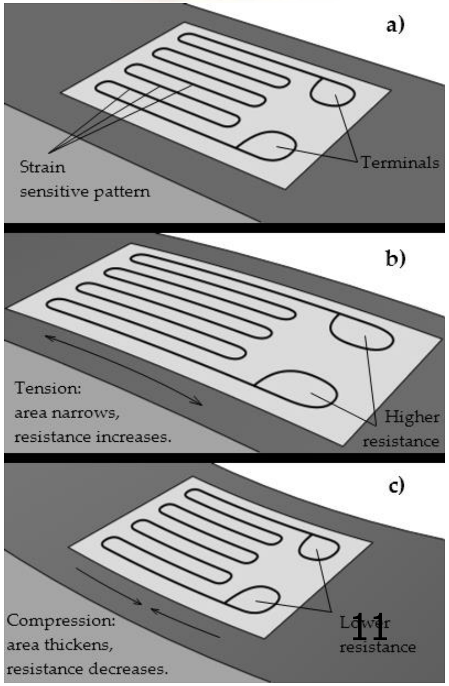
- Light Sensors
- Thermocouple
- Tilt sensors (eg. Mercury switch)
- Limit switches

Resistive Sensors



- Potentiometers
  - Cheap, easy to mount but limited span
  - Simple rotation/sliding measurement
  - Friction disturbs the measurement
- Strain gages and bend sensors
  - Suitable for force sensing
- Light sensors (photo cells)
  - Slow response
  - Nonlinear resistance, but cheap
  - Suitable for detection of light or its direction

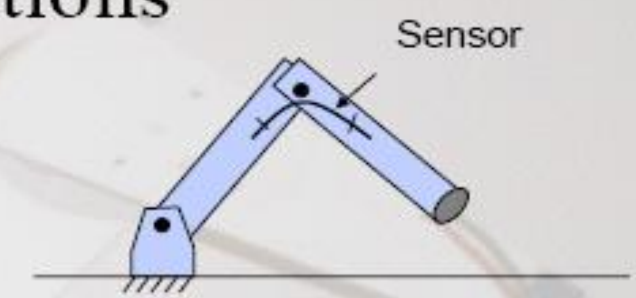




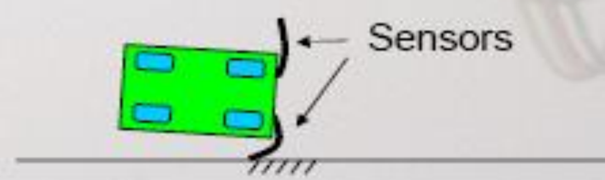
Strain Gages and Potentiometers

Applications

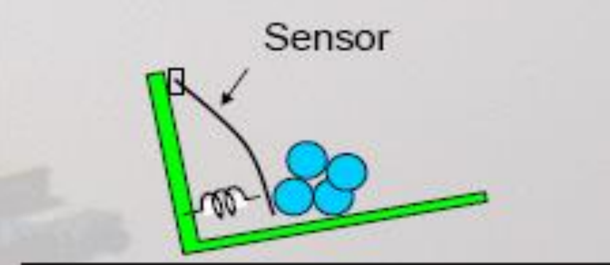
- Measure bend of a joint

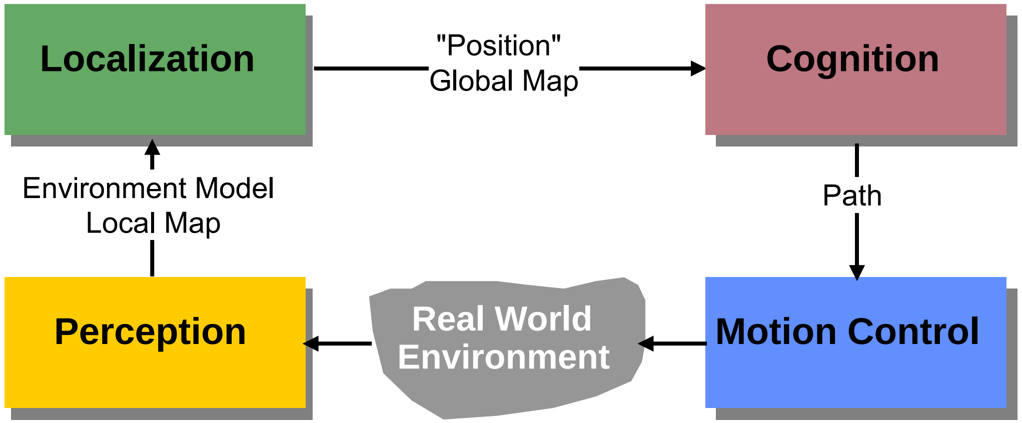


- Wall Following/Collision Detection



- Weight Sensor





Perception

Sensors

Vision

Uncertainties, Fusion

Features

Sensors for Mobile Robots

- Why should a robotics engineer know about sensors?
  - They are the **key components** for perceiving the environment
  - **Understanding the physical principles** enables appropriate use

- Understanding the physical principle behind sensors enables us:
  - To **properly select** the sensors for a given application
  - To **properly model** the sensor system, e.g. resolution, bandwidth, **uncertainties**

Example of “Simple” Real World Situations

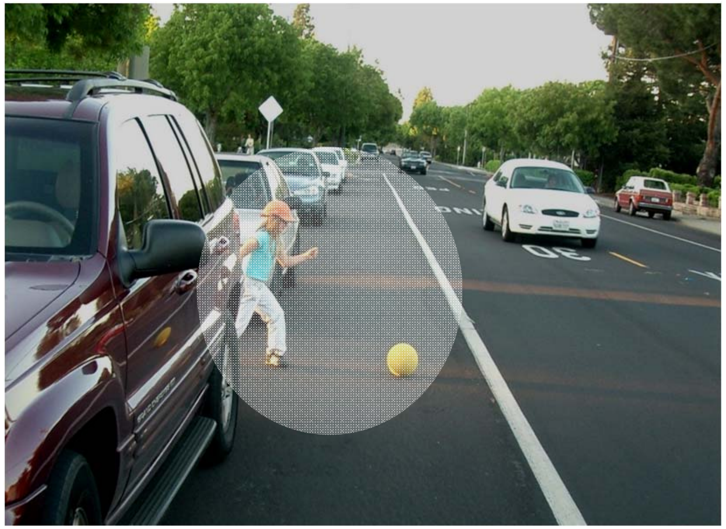
- A typical play in robocup

- What the robot sees

Curtesy of Manuela Veloso <veloso@cs.cmu.edu >
Carnegie Mellon University, veloso@cs.cmu.edu

Dealing with Real World Situations

- Reasoning about a situation



- Cognitive systems have to interpret situations based on uncertain and only partially available information
- They need ways to learn functional and contextual information (semantics / understanding)

Probabilistic Reasoning

4a 6 Perception for Mobile Robots

Places / Situations

A specific room, a meeting situation, …

Servicing / Reasoning

Objects

Doors, Humans, Coke bottle, car , …

Interaction

Features

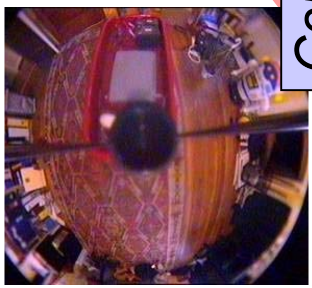
Lines, Contours, Colors, Phonemes, …

Navigation

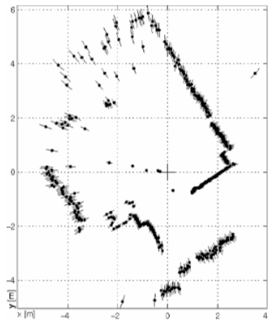
Raw Data

Vision, Laser, Sound, Smell, …

Compressing Information



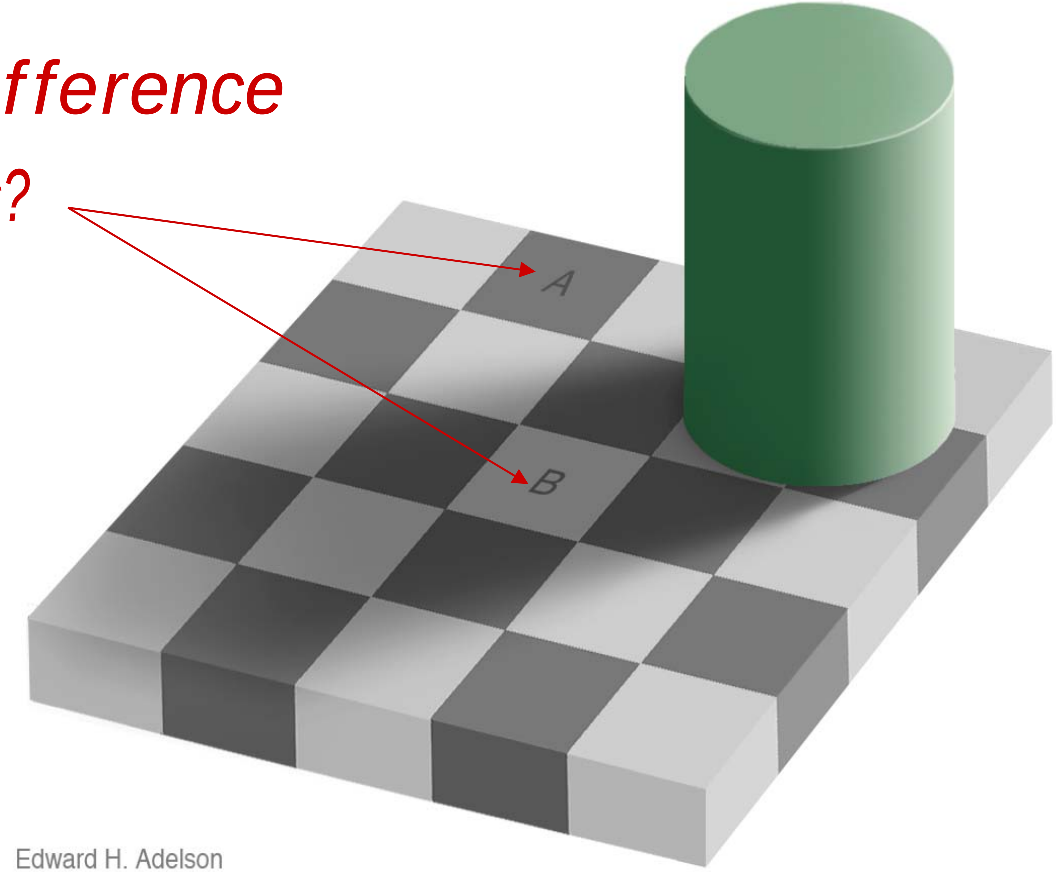
- *Functional / Contextual Relationships of Objects*
  - *imposed*
  - *learned*
  - *spatial / temporal/semantic*
- *Models / Semantics*
  - *imposed*
  - *learned*
- *Models*
  - *imposed*
  - *learned*



The Challenge

- Perception and models are strongly linked

What is the difference in brightness?



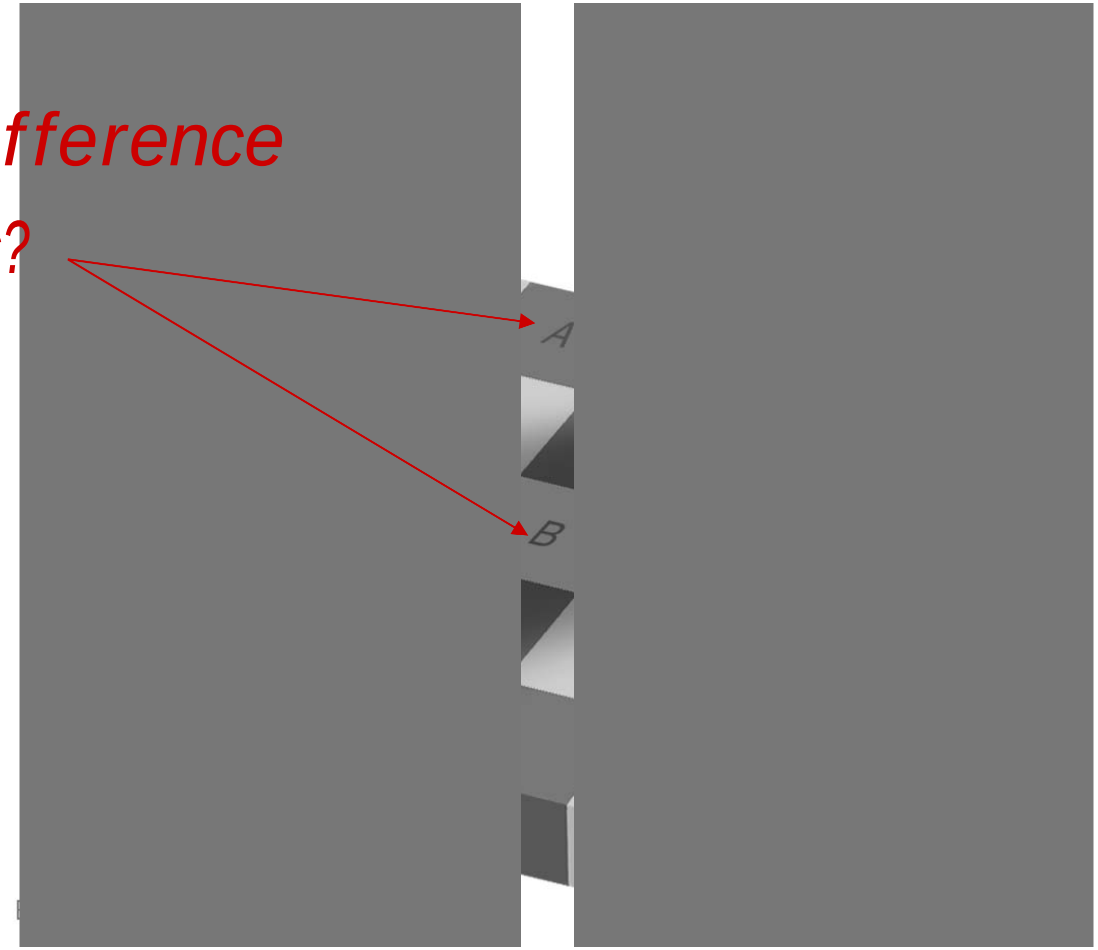
Courtesy E. Adelson

http://web.mit.edu/persci/people/adelson/checkershadow_downloads.html

The Challenge

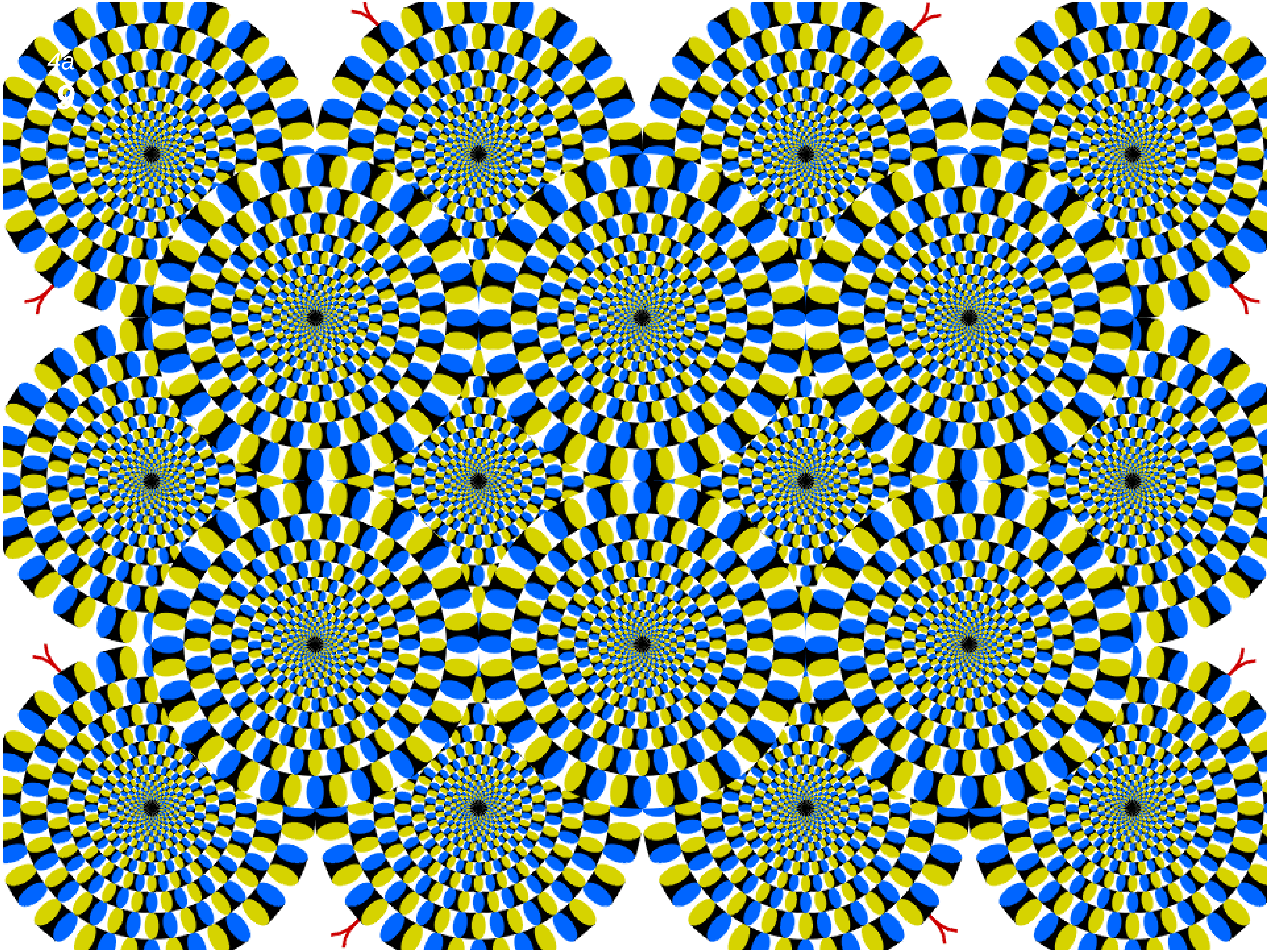
- Perception and models are strongly linked

What is the difference in brightness?



Courtesy E. Adelson

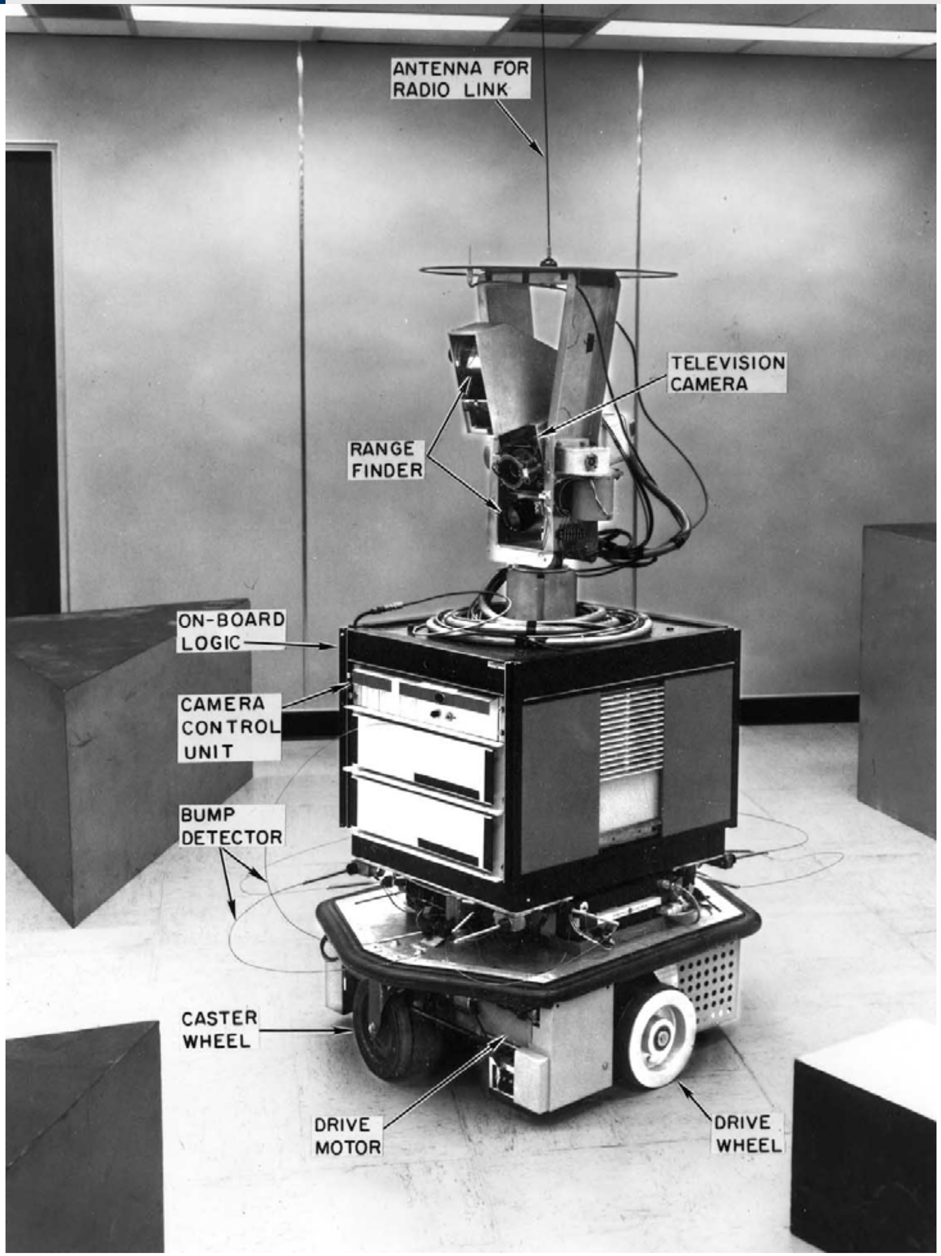
http://web.mit.edu/persci/people/adelson/checkershadow_downloads.html


4a
9

Case Studies

- Lets look at a couple of case studies before we begin
  - What sensors are commonly employed on robots
  - How has the choice of sensors evolved over time
    - Shakey
    - Tourguide robots late 90ies
    - Willow Garage PR2
    - SmartTer – the autonomous car

Shakey the Robot (1966-1972), SRI International



- Operating environment
  - Indoors
  - Engineered
- Sensors
  - Wheel encoders
  - Bumb detector
  - Range finder
  - Camera

Rhino Tourguide Robot (1995-1998), University of Bonn

- Operating environment
  - Indoors (Museum: unstructured and dynamic)

- Sensors
  - Wheel encoders
  - Ring of sonar sensors
  - Pan-tilt camera

C University of Bonn

PR2 (2010-), Willow Garage

C Willow Garage

- Operating environment
  - Indoors and outdoors
  - Onroad only
- Sensors
  - Wheel encoders
  - Bumper
  - IR sensors
  - Laser range finder
  - 3D nodding laser range finder
  - Inertial measurement unit
  - Pan-tilt stereo camera with texture projector (active)
  - Pressure sensor and accelerometer inside hands
  - ...

The SmartTer Platform

- Three navigation SICK laser scanners
  - Obstacle avoidance and local navigation
- Two rotating laser scanners (3D SICK)
  - 3D mapping of the environment
  - Scene interpretation
- Omnidirectional camera
  - Texture information for the 3D terrain maps
  - Scene interpretation
- Monocular camera
  - Scene interpretation

Motion Estimation / Localization

- Differential GPS system *(Omnistar 8300HP)*
- Inertial measurement unit *(Crossbow NAV420)*
- **Optical Gyro**
- Odometry (wheel speed, steering angel)
  - Motion estimation
  - Localization

Internal car state sensors

- Vehicle state flags (engine, door, etc.)
- Engine data, gas pedal value

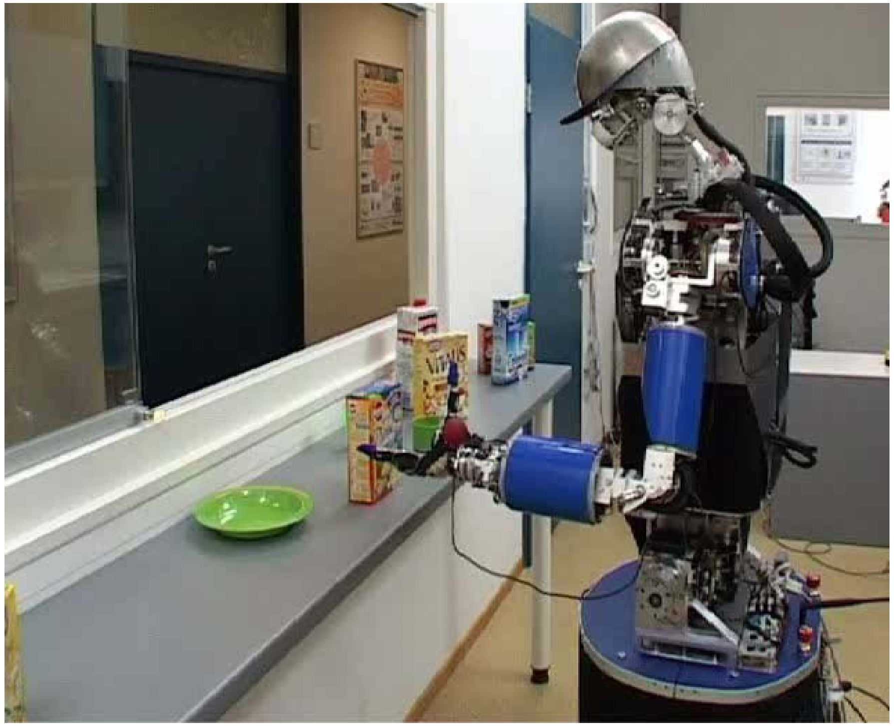
Camera for live video streaming

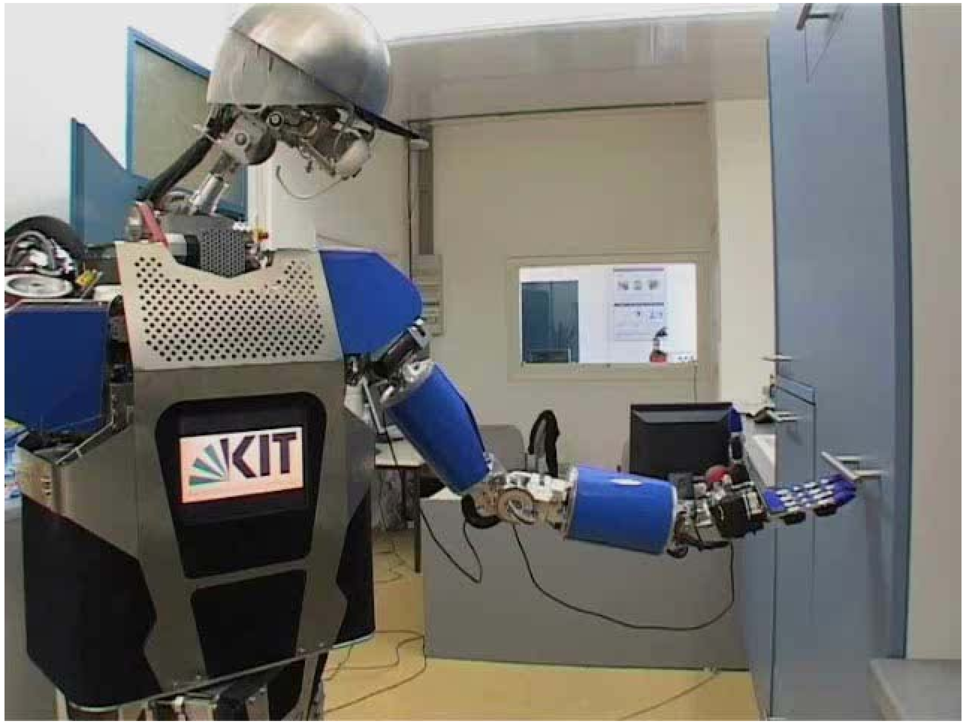
- Transmission range up to 2 km

Autonomous Navigation and 3D Mapping

AMAR - Demonstration Scenario

- Interactive service tasks in a kitchen environment





Courtesy of Dillmann & Asfour,
Karlsruhe Instititue of Technology, Germany

Multimodal detection and tracking

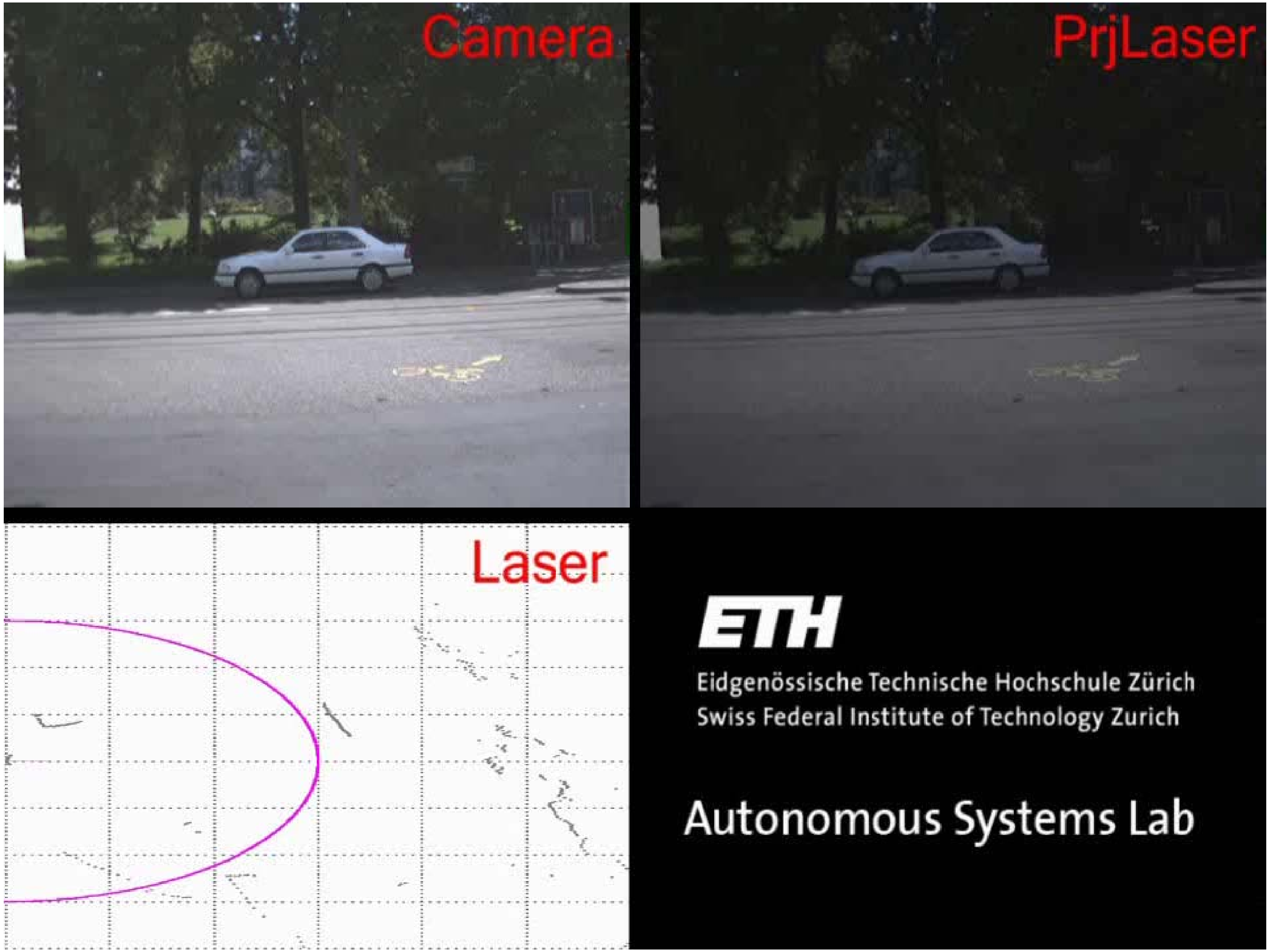
Pedestrians

Cars

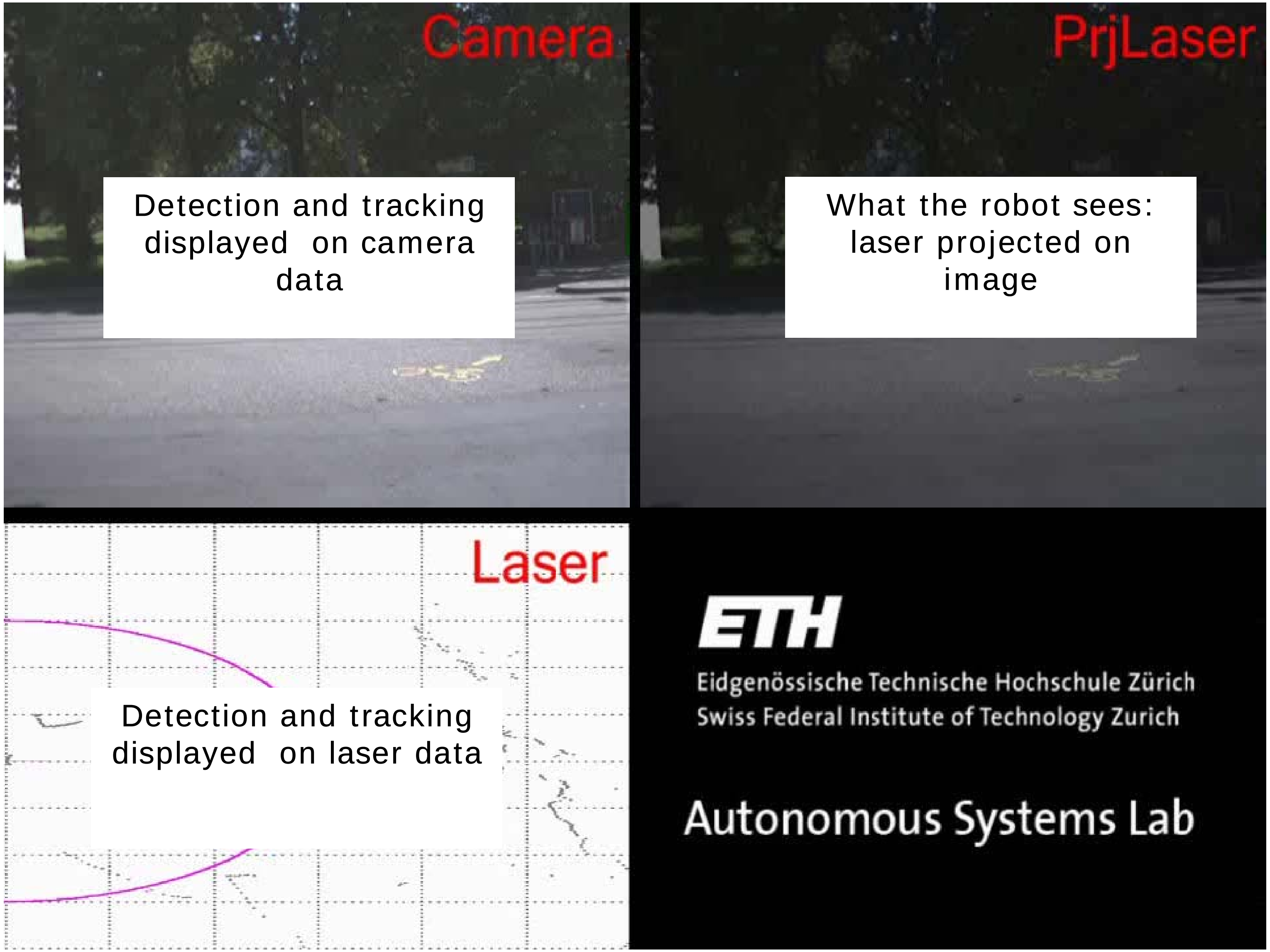
Rich info
Inexpensive
Noise
No distance

High prec
Light indep
Low info

Video


Camera
PrjLaser
Laser
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
Autonomous Systems Lab

Video


Camera
Detection and tracking displayed on camera data
PrjLaser
What the robot sees: laser projected on image
Laser
Detection and tracking displayed on laser data
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
Autonomous Systems Lab

Classification of Sensors

- What:
  - Proprioceptive sensors
    - measure values internally to the system (robot),
    - e.g. motor speed, wheel load, heading of the robot, battery status
  - Exteroceptive sensors
    - information from the robots environment
    - distances to objects, intensity of the ambient light, unique features.
- How:
  - Passive sensors
    - Measure energy coming from the environment
  - Active sensors
    - emit their proper energy and measure the reaction
    - better performance, but some influence on environment

General Classification (1)

General classification (typical use)	Sensor Sensor System	PC or EC	A or P
Tactile sensors (detection of physical contact or closeness; security switches)	Contact switches, bumpers Optical barriers Noncontact proximity sensors	EC EC EC	P A A
Wheel/motor sensors (wheel/motor speed and position)	Brush encoders Potentiometers Synchros, resolvers Optical encoders Magnetic encoders Inductive encoders Capacitive encoders	PC PC PC PC PC PC PC	P P A A A A A
Heading sensors (orientation of the robot in relation to a fixed reference frame)	Compass Gyroscopes Inclinometers	EC PC EC	P P A/P

A, active; P, passive; P/A, passive/active; PC, proprioceptive; EC, exteroceptive.

General Classification (2)

General classification (typical use)	Sensor Sensor System	PC or EC	A or P
Ground-based beacons (localization in a fixed reference frame)	GPS Active optical or RF beacons Active ultrasonic beacons Reflective beacons	EC EC EC EC	A A A A
Active ranging (reflectivity, time-of-flight, and geometric triangulation)	Reflectivity sensors Ultrasonic sensor Laser rangefinder Optical triangulation (1D) Structured light (2D)	EC EC EC EC EC	A A A A A
Motion/speed sensors (speed relative to fixed or moving objects)	Doppler radar Doppler sound	EC EC	A A
Vision-based sensors (visual ranging, whole-image analysis, segmentation, object recognition)	CCD/CMOS camera(s) Visual ranging packages Object tracking packages	EC	P

Characterizing Sensor Performance (2)

- Basic sensor response ratings (cont.)

  - Range
    - **upper limit - lower limit**

  - Resolution
    - **minimum difference between two values**
    - usually: lower limit of dynamic range = resolution
    - for digital sensors it is usually the A/D resolution.
      - e.g. 5V / 255 (8 bit)

  - Linearity
    - variation of output signal as function of the input signal
    - linearity is less important when signal is treated with a computer

$$x \to f(x)$$
$$y \to f(y)$$

$$\alpha \cdot x + \beta \cdot y \to f(\alpha \cdot x + \beta \cdot y) = \alpha \cdot f(x) + \beta \cdot f(y)$$

Characterizing Sensor Performance (3)

- Basic sensor response ratings (cont.)

  - Bandwidth or Frequency
    - **the speed with which a sensor can provide a stream of readings**
    - usually there is an upper limit depending on the sensor and the sampling rate
    - lower limit is also possible, e.g. acceleration sensor
    - one has also to consider phase (delay) of the signal

In Situ Sensor Performance (1)

Characteristics that are especially relevant for real world environments

- Sensitivity
  - ratio of output change to input change $\frac{dy}{dx}$
  - however, in real world environment, the sensor has very often high sensitivity to other environmental changes, e.g. illumination
- Cross-sensitivity (and cross-talk)
  - sensitivity to other environmental parameters (e.g. temperature, magnetic field)
  - influence of other active sensors
- Error / Accuracy
  - difference between the sensor's output and the true value

error

$$\left(accuracy = 1 - \frac{|m - v|}{v} \right)$$

m = measured value

v = true value

In Situ Sensor Performance (2)

Characteristics that are especially relevant for real world environments

- Systematic error -> deterministic errors
  - **caused by factors that can (in theory) be modeled -> prediction**
  - e.g. calibration of a laser sensor or of the distortion caused by the optics of a camera

- Random error -> non-deterministic
  - **no prediction possible with given sensors**
  - however, they can be described probabilistically

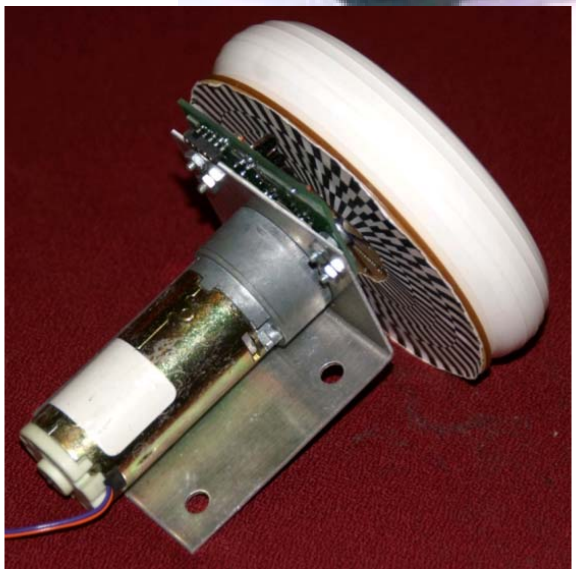
- Precision
  - ***reproducibility* of sensor results:** $precision = \frac{range}{\sigma}$

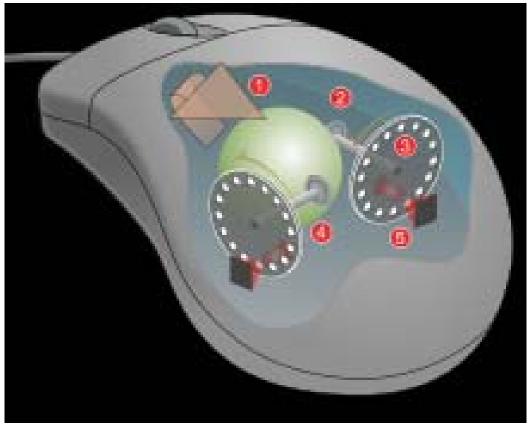
Sensors: outline

- Optical encoders
- Heading sensors
  - Compass
  - Gyroscopes
- Accelerometer
- IMU
- GPS
- Range sensors
  - Sonar
  - Laser
  - Structured light
- Vision (next lecture)

Encoders

- Definition:
  - **electro-mechanical device** that converts linear or angular position of a shaft to an analog or digital signal, making it an linear/anglular transducer

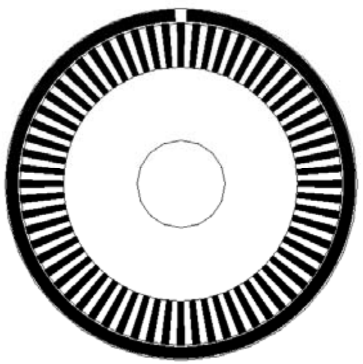


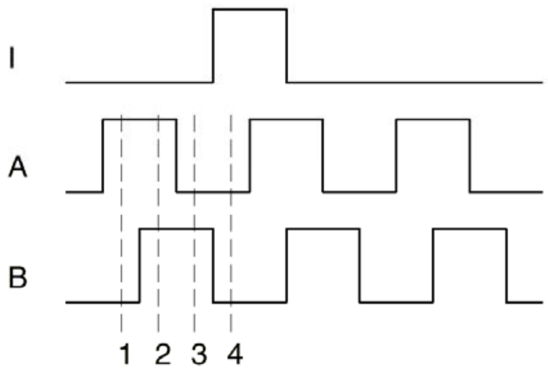


Wheel / Motor Encoders

- Use cases
  - **measure position** or speed of the wheels or steering
  - **integrate wheel movements** to get an estimate of the position -> odometry
  - optical encoders are proprioceptive sensors
  - typical resolutions: 64 - 2048 increments per revolution.
  - for high resolution: interpolation

- Working principle of optical encoders
  - regular: counts the number of transitions but cannot tell the direction of motion
  - quadrature: uses two sensors in quadrature-phase shift. The ordering of which wave produces a rising edge first tells the direction of motion. Additionally, resolution is 4 times bigger
  - a single slot in the outer track generates a reference pulse per revolution

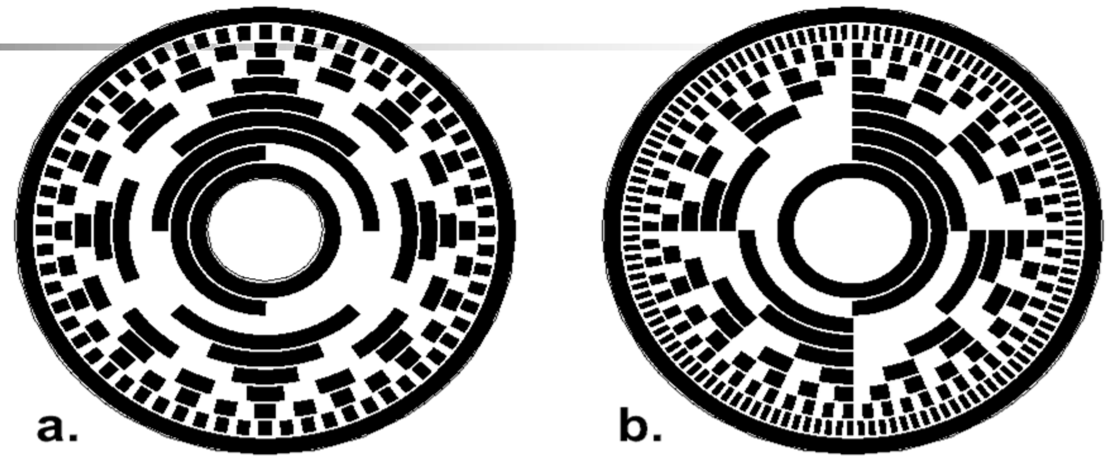




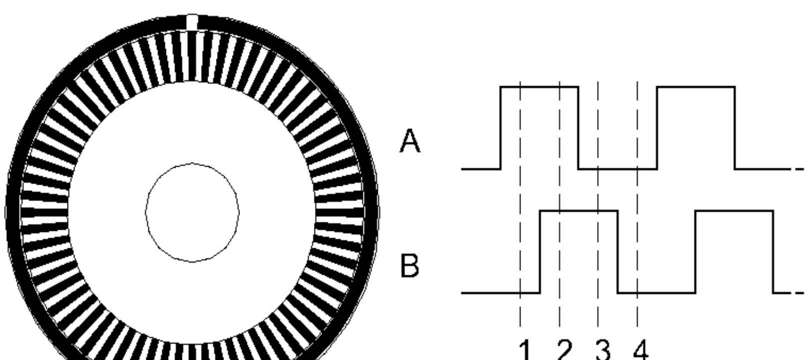
State	Ch A	Ch B
S_1	High	Low
S_2	High	High
S_3	Low	High
S_4	Low	Low

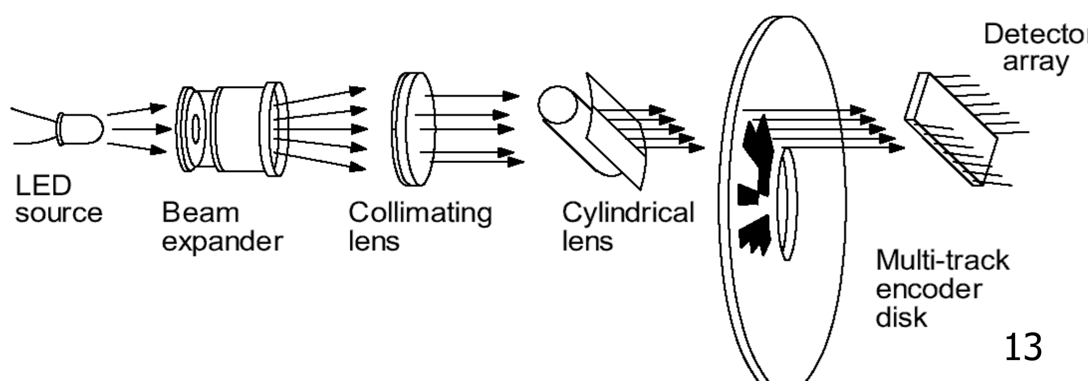
Optical Encoders

- Absolute
  - Binary Coded
  - Grey Coded
- Incremental

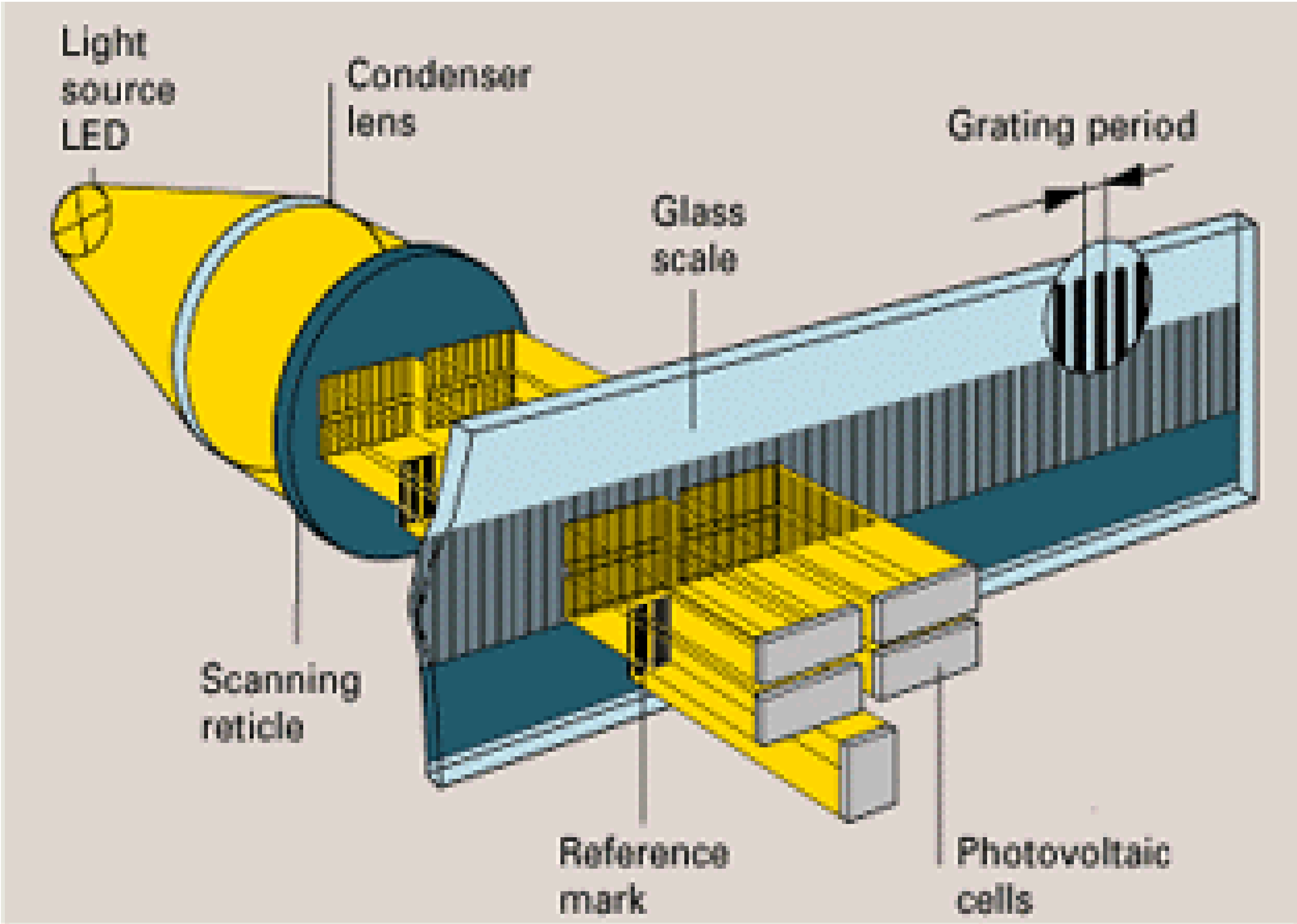


a. Counterclockwise rotation by one position increment will cause only one bit to change.
b. The same rotation of a binary-coded disk will cause all bits to change in the particular case (255 to 0) illustrated by the reference line at 12 o'clock.

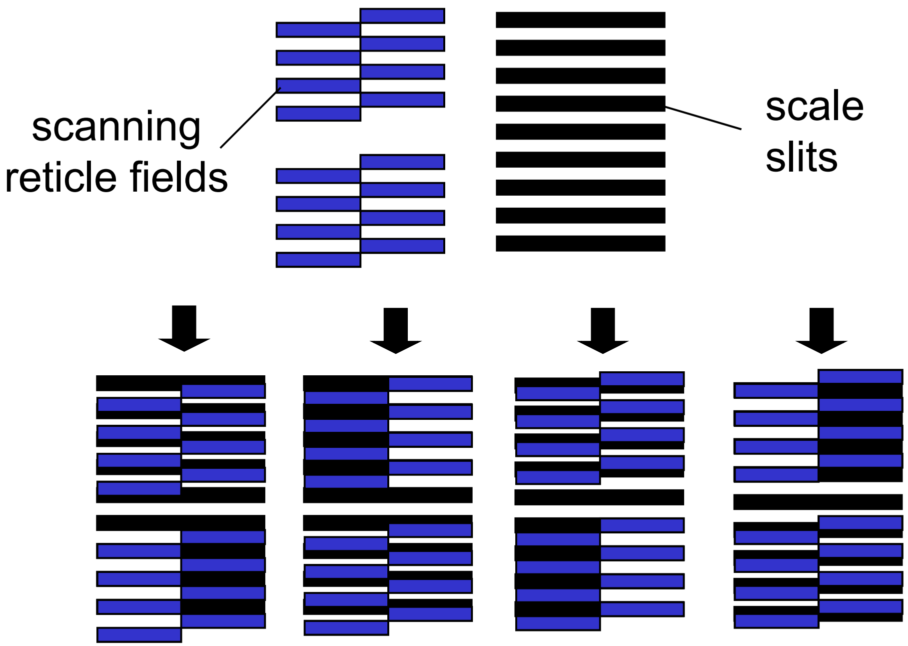




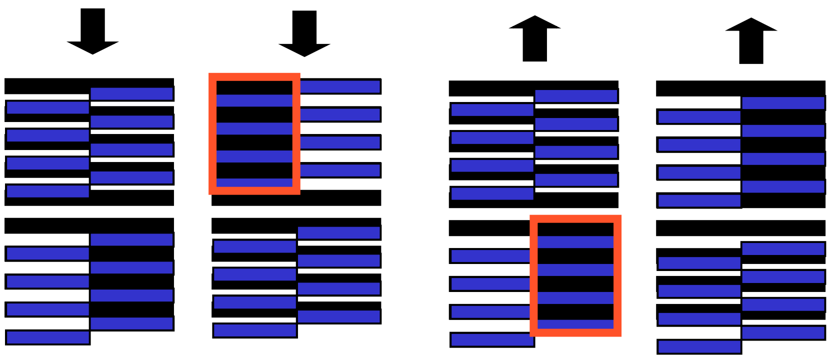
Wheel / Motor Encoders (2)



Wheel / Motor Encoders (3)



Notice what happens when the direction changes:



2. Main Characteristics

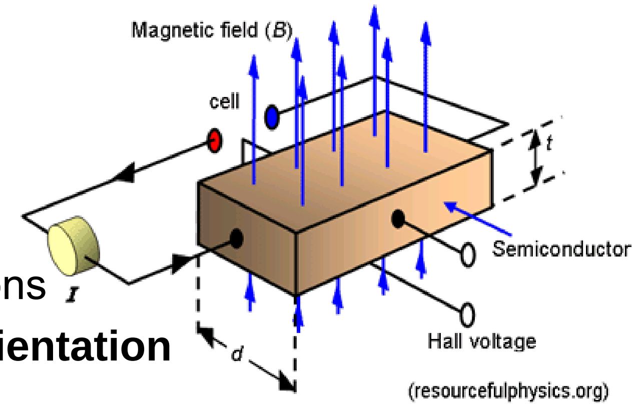
- The four fields on the scanning reticle are shifted in phase relative to each other by one quarter of the grating period, which equals 360°/(number of lines)
- This configuration allows the detection of a change in direction
- Easy to interface with a micro-controller

Heading Sensors

- Definition:
  - Heading sensors are sensors that determine the robot's orientation and inclination.
- Heading sensors can be proprioceptive (gyroscope, accelerometer) or exteroceptive (compass, inclinometer).
- Allows, together with an appropriate velocity information, to integrate the movement to a position estimate.
  - This procedure is called **deduced reckoning** (ship navigation)

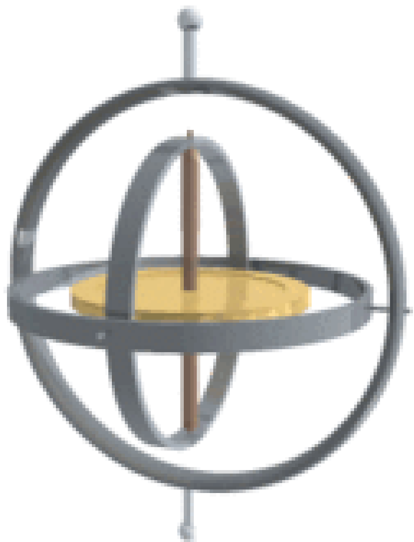
Compass

- Used since before 2000 B.C.
  - when Chinese suspended a piece of natural magnetite from a silk thread and used it to guide a chariot over land.
- Magnetic field on earth
  - absolute measure for orientation (even birds use it for migrations (2001 discovery))
- Large variety of solutions to measure magnetic/true north
  - mechanical magnetic compass
  - Gyrocompass
  - direct measure of the magnetic field (Hall-effect, magneto-resistive sensors)
- Major drawback of magnetic solutions
  - weakness of the earth field (30 μTesla)
  - easily disturbed by magnetic objects or other sources
  - bandwidth limitations (0.5 Hz) and susceptible to vibrations
  - **not suitable for indoor environments for absolute orientation**
  - useful indoor (only locally)



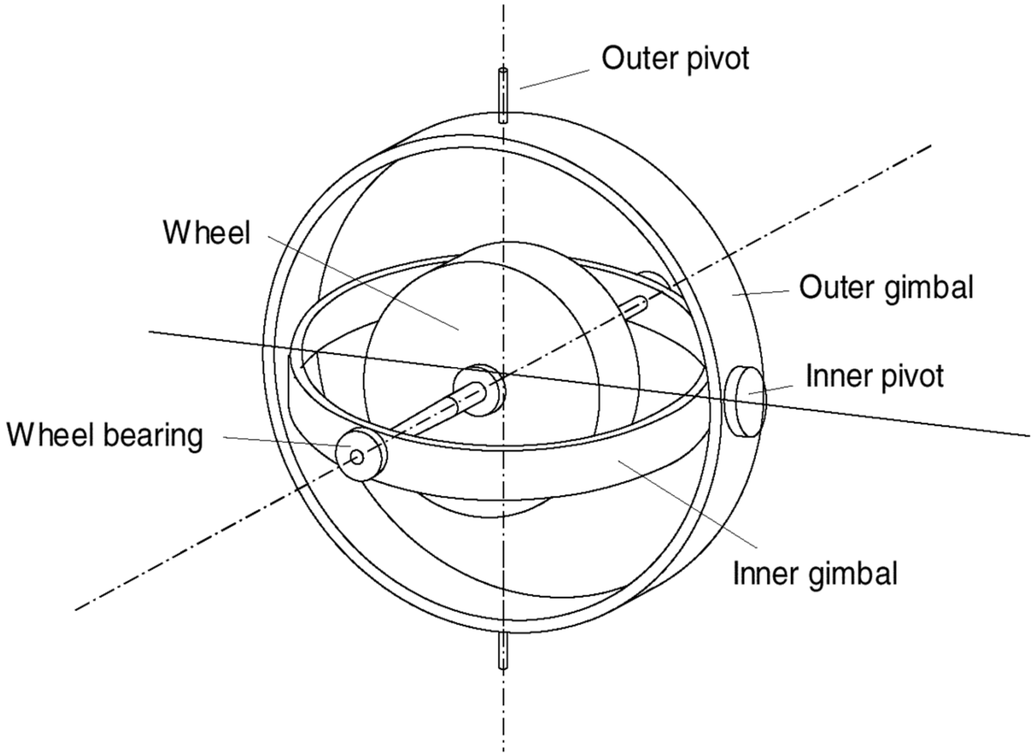
Gyroscope

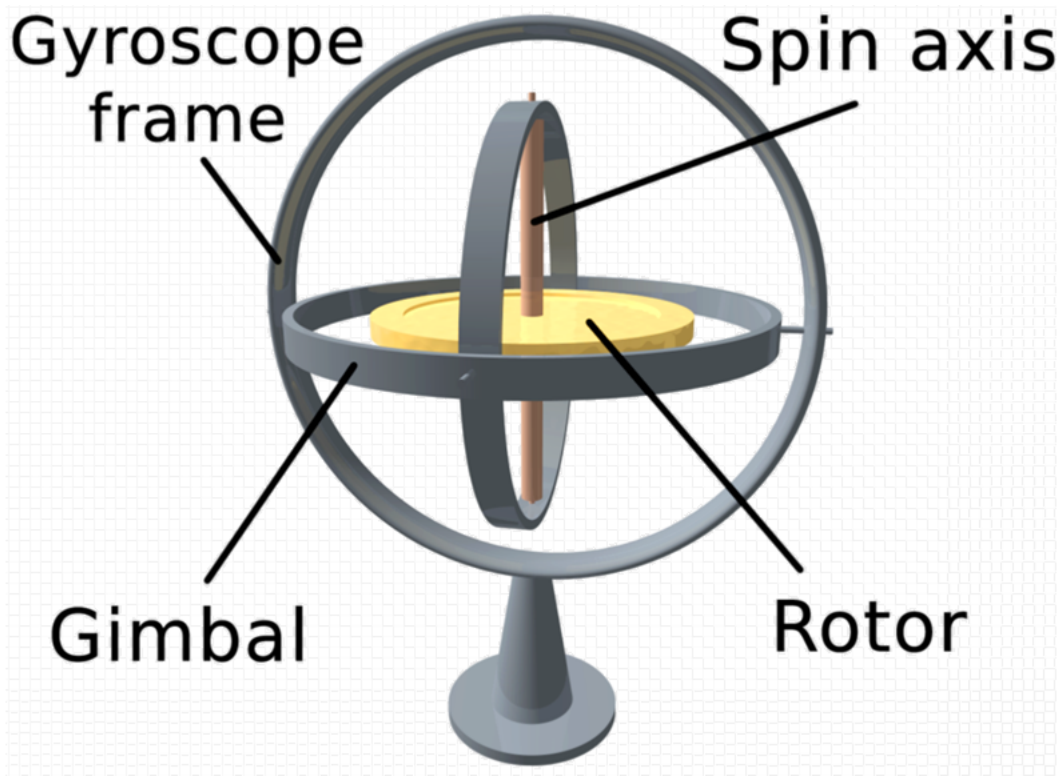
- Definition:
  - Heading sensors that preserve their orientation in relation to a fixed reference frame
  - They provide an absolute measure for the heading of a mobile system.
- Two categories, the mechanical and the optical gyroscopes
  - Mechanical Gyroscopes
    - Standard gyro (angle)
    - Rate gyro (speed)
  - Optical Gyroscopes
    - Rate gyro (speed)



Mechanical Gyroscopes

- Concept:
  - Inertial properties of a fast spinning rotor
  - Angular momentum associated with a spinning wheel keeps the axis of the gyroscope inertially stable.
- **No torque can be transmitted from the outer pivot to the wheel axis**
  - spinning axis will therefore be space-stable
  - however friction in the axes bearings will introduce torque and so drift ->precession
- Quality: 0.1° in 6 hours (a high quality mech. gyro costs up to 100,000 $)

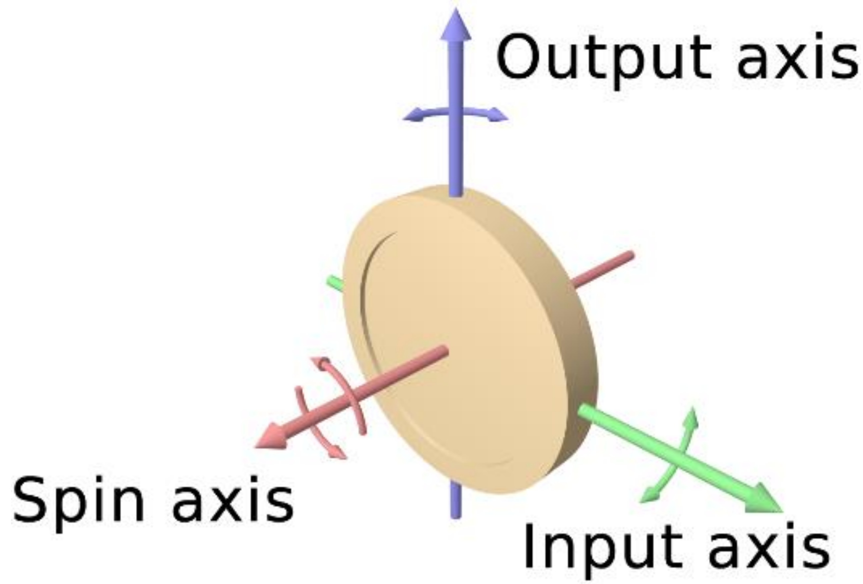


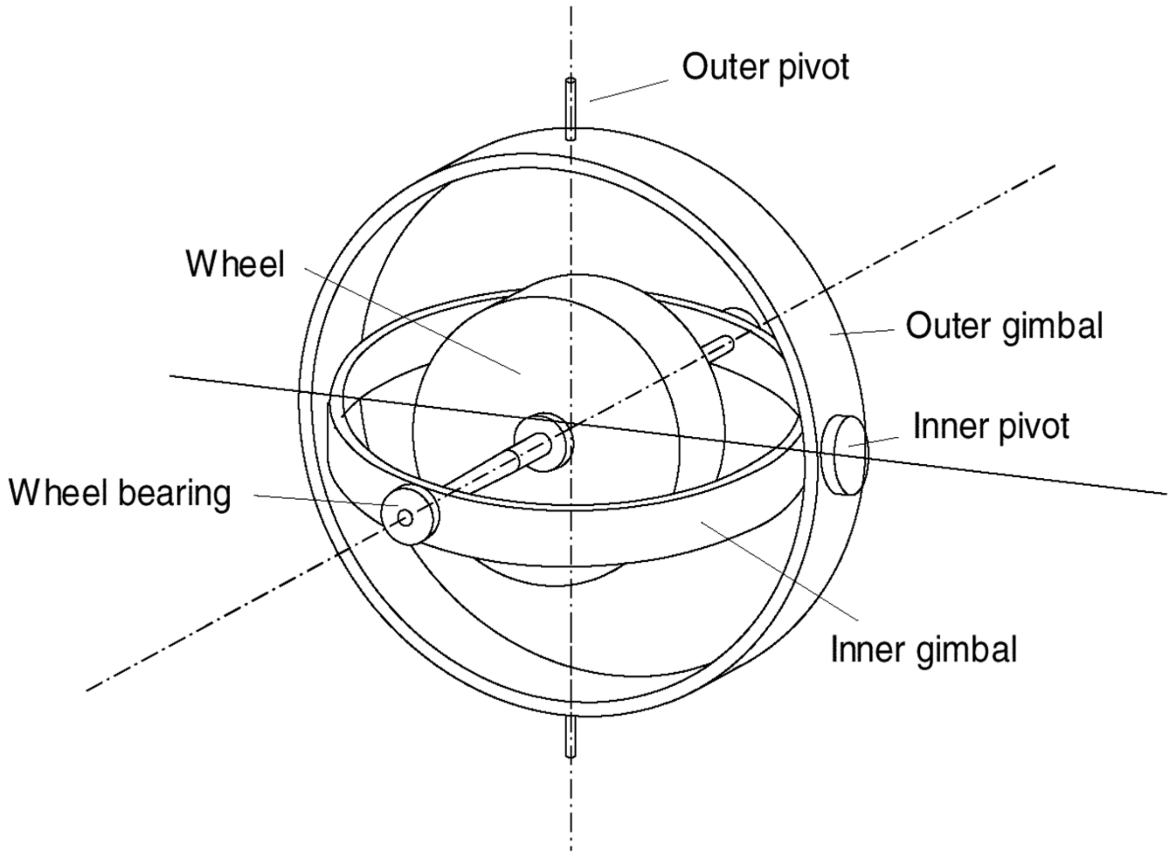


Rate gyros

- Same basic arrangement shown as regular mechanical gyros

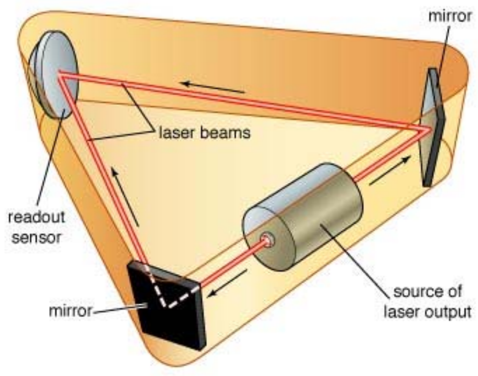
- But: gimbals are restrained by torsional springs
  - enables to measure angular speeds instead of the orientation.

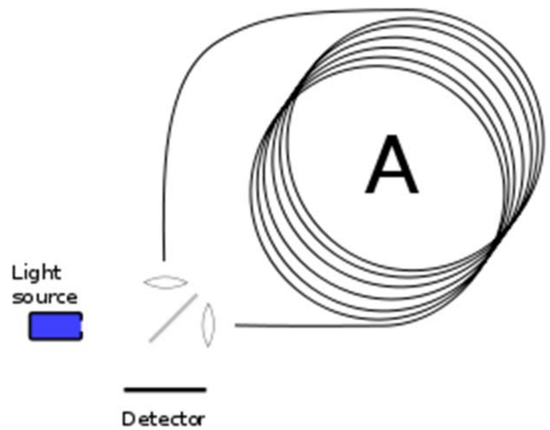




4a
41
Optical Gyroscopes

- Optical gyroscopes
  - angular speed (heading) sensors
    using two monochromic light (or laser) beams
    from the same source.
  - One is traveling in a fiber clockwise,
    the other counterclockwise around a cylinder
- Laser beam traveling in direction opposite to the rotation
  - slightly shorter path
  - phase shift of the two beams is proportional to the angular velocity Ω of the cylinder
  - In order to measure the phase shift, coil consists of as much as 5Km optical fiber
- New solid-state optical gyroscopes based on the same principle are build using microfabrication technology.





Single axis optical gyro

3-axis optical gyro

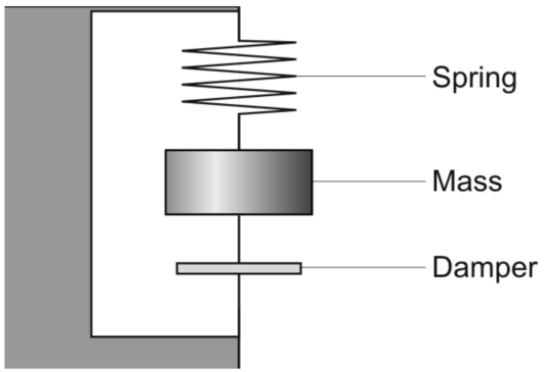
Mechanical Accelerometer

- Accelerometers measure all external forces acting upon them, including gravity
- accelerometer acts like a spring–mass–damper system

$$F_{applied} = F_{inertial} + F_{damping} + F_{spring} = m\ddot{x} + c\dot{x} + kx$$

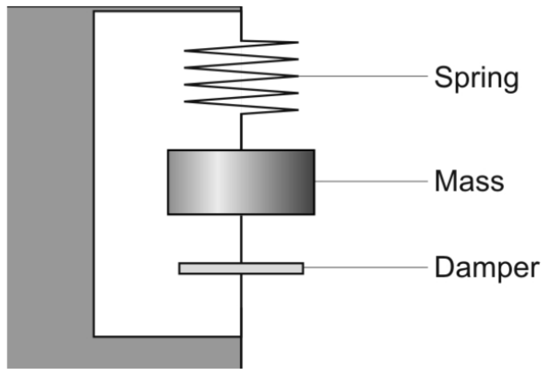
Where *m* is the proof mass, *c* the damping coefficient, *k* the spring constant

- at steady-state: $a_{applied} = \frac{kx}{m}$



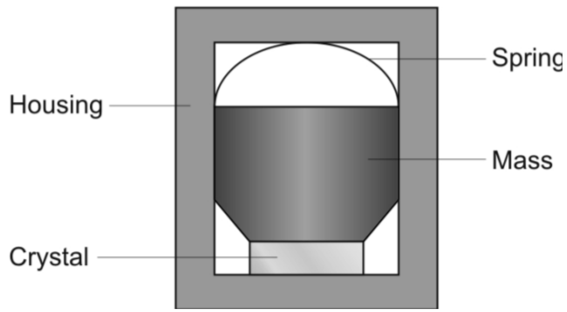
Mechanical Accelerometer

- On the Earth's surface, the accelerometer always indicates 1g along the vertical axis
- To obtain the inertial acceleration (due to motion alone), the gravity must be subtracted. Conversely, the device's output will be zero during free fall
- Bandwidth up to 50 KHz
- An accelerometer measures acceleration only along a single axis. By mounting three accelerometers orthogonally to one another, a three-axis accelerometer can be obtained

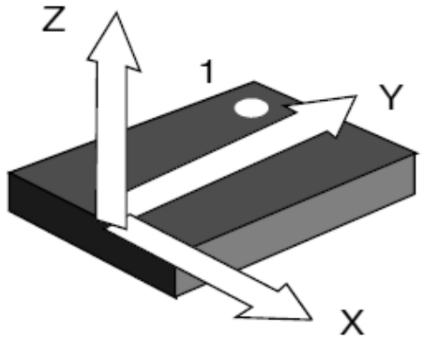


Other Accelerometers

- Modern accelerometers use Micro Electro-Mechanical Systems (MEMS) consisting of a spring-like structure with a proof mass. Damping results from the residual gas sealed in the device.

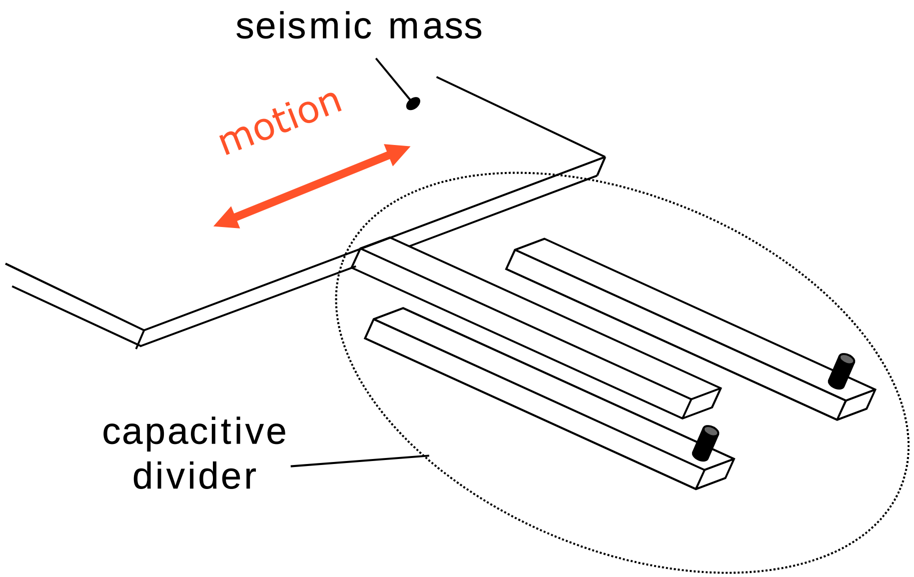


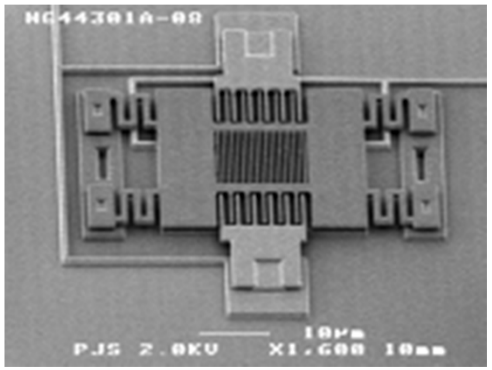
- In **capacitive accelerometers** the capacitance between a fixed structure and the proof mass is measured



- **Piezoelectric accelerometers** are based on the property exhibited by certain crystals to generate a voltage when a mechanical stress is applied to them

Factsheet: MEMS Accelerometer (1)





<http://www.mems.sandia.gov/>

1. Operational Principle

The primary transducer is a vibrating mass that relates acceleration to displacement. The secondary transducer (a capacitive divider) converts the displacement of the seismic mass into an electric signal.

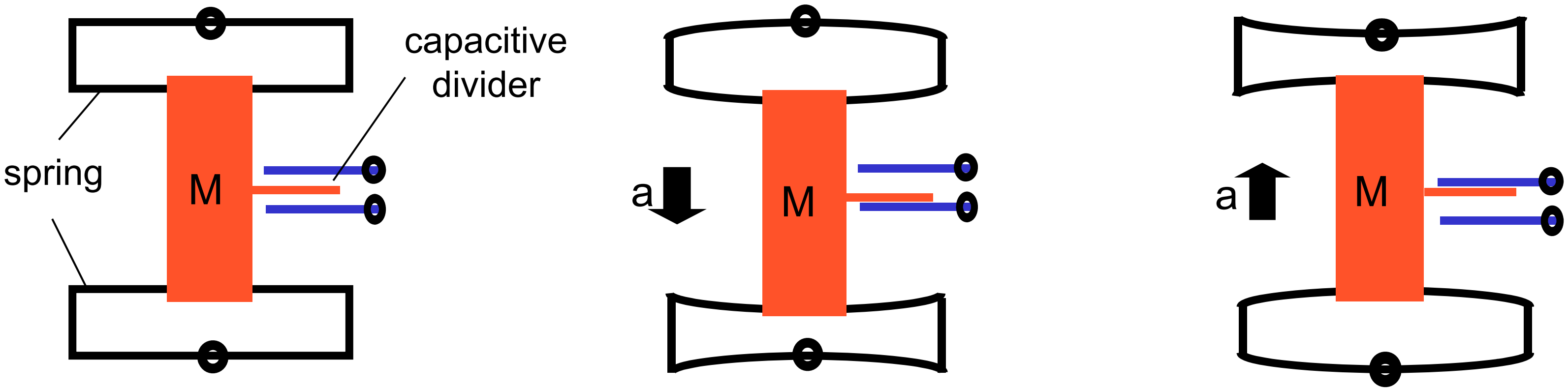
2. Main Characteristics

- Can be multi-directional
- Various sensing ranges from 1 to 50 g

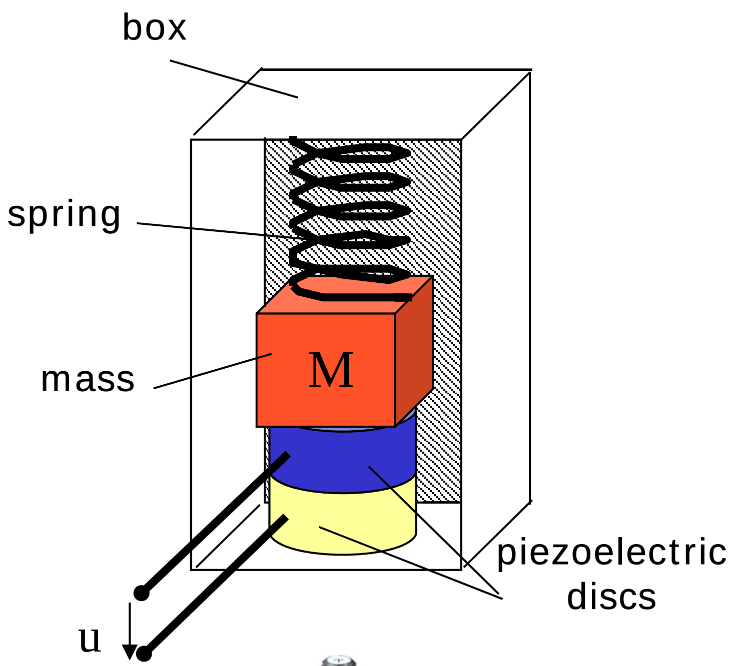
3. Applications

- Dynamic acceleration
- Static acceleration (inclinometer)
- Airbag sensors (+- 35 g)
- Control of video games (Wii)

Factsheet: MEMS Accelerometer (2)



Factsheet: Piezoelectric Accelerometer



<http://www.pcb.com/>

1. Operational Principle

Primary transducer is typically a single-degree-of-freedom spring-mass system that relates acceleration to displacement. Secondary transducer (piezoelectric discs) converts displacement of the seismic mass into an electrical signal (voltage).

2. Main Characteristics

- Piezoelectric elements cannot produce a signal under constant acceleration (i.e., static) conditions
- 2-D and 3-D accelerometers can be created by combining 2 or 3 1-D modules

3. Applications

- Vibration analysis
- Machine diagnostics
- Active vehicle suspension
- Autonomously guided vehicles
- Earthquake sensors
- Modal analysis

Inertial Measurement Unit (IMU)

- **Definition**
  - **An inertial measurement unit (IMU) is a device that uses** measurement systems such as **gyroscopes and accelerometers** to estimate the relative position (x, y, z), orientation (roll, pitch, yaw), velocity, and acceleration of a moving vehicle.

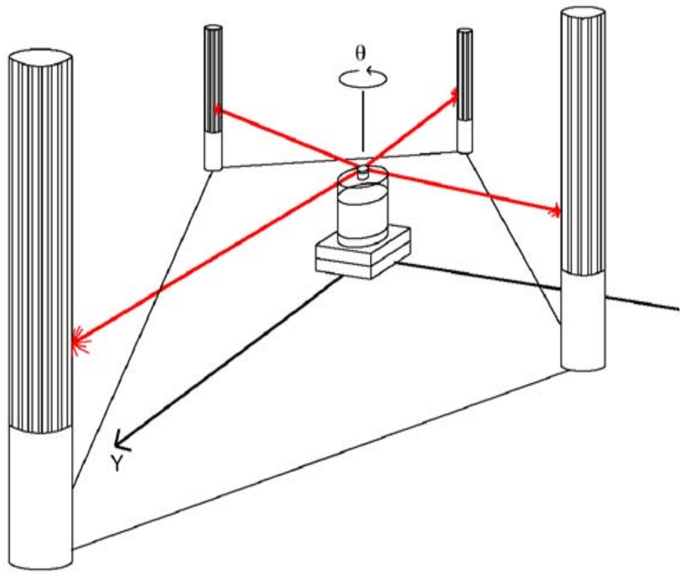
- In order to estimate motion, the gravity vector must be subtracted. Furthermore, initial velocity has to be known.

- **IMUs are extremely sensitive to measurement errors** in gyroscopes and accelerometers: **drift in the gyroscope** unavoidably undermines the estimation of the vehicle orientation relative to gravity, which results in incorrect cancellation of the gravity vector. Additionally observe that, because the **accelerometer data is integrated twice** to obtain the position, any residual gravity vector results in a quadratic error in position.

- After long period of operation, **all IMUs drift.** To cancel it, some external reference like GPS or cameras has to be used.

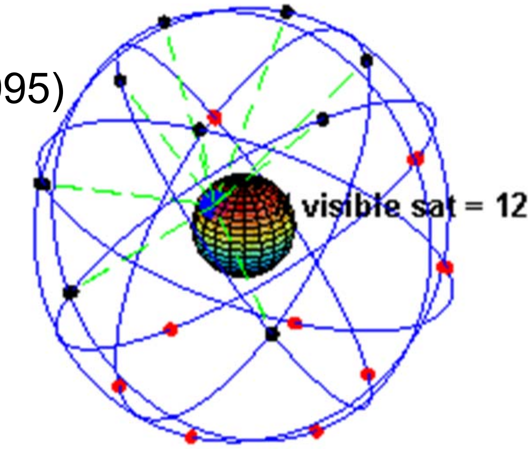
Ground-Based Active and Passive Beacons

- "Elegant" way to solve the localization problem in mobile robotics
- **Beacons are signaling guiding devices with a precisely known position**
- Beacon base navigation is used since the humans started to travel
  - Natural beacons (landmarks) like **stars, mountains or the sun**
  - Artificial beacons like **lighthouses**
- The recently introduced Global Positioning System (GPS) revolutionized modern navigation technology
  - Already one of the key sensors for outdoor mobile robotics
  - For indoor robots GPS is not applicable,
- Major drawback with the use of beacons in indoor:
  - Beacons require changes in the environment -> costly.
  - Limit flexibility and adaptability to changing environments.



Global Positioning System (GPS) (1)

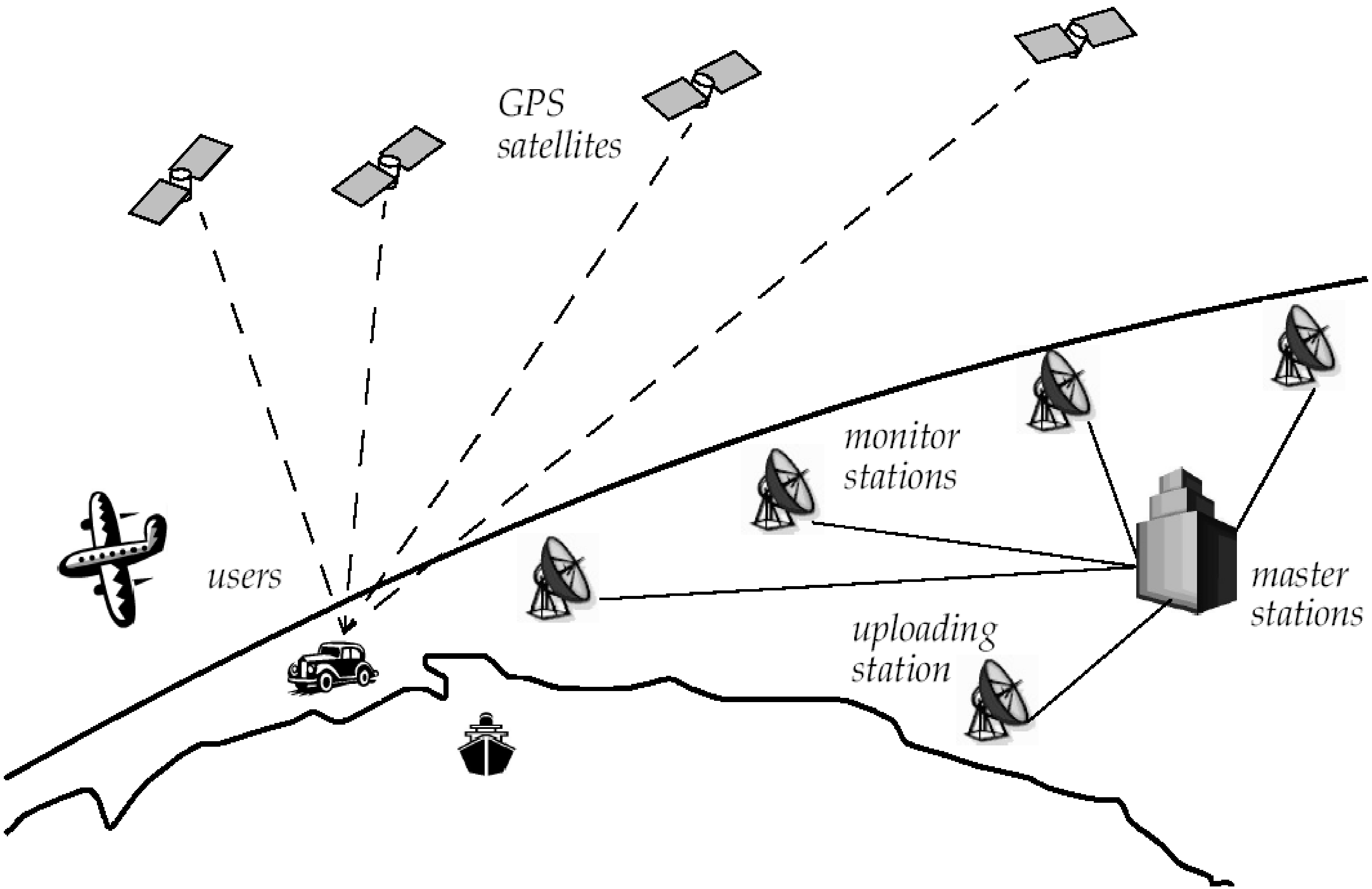
- Facts
  - Recently it became accessible for commercial applications (1995)
  - 24+ satellites orbiting the earth every 12 hours at a height of 20.190 km.
  - 4 satellites are located in each of 6 orbits with 60 degrees orientation between each other.



- Working Principle
  - Location of any GPS receiver is determined through a time of flight measurement (satellites send orbital location (*ephemeris*) plus time; the receiver computes its location through **trilateration** and **time correction**)

- Technical challenges:
  - **Time synchronization** between the individual satellites and the GPS receiver
  - Real time update of the exact location of the satellites
  - Precise measurement of the time of flight
  - **Interferences** with other signals

Global Positioning System (GPS) (2)

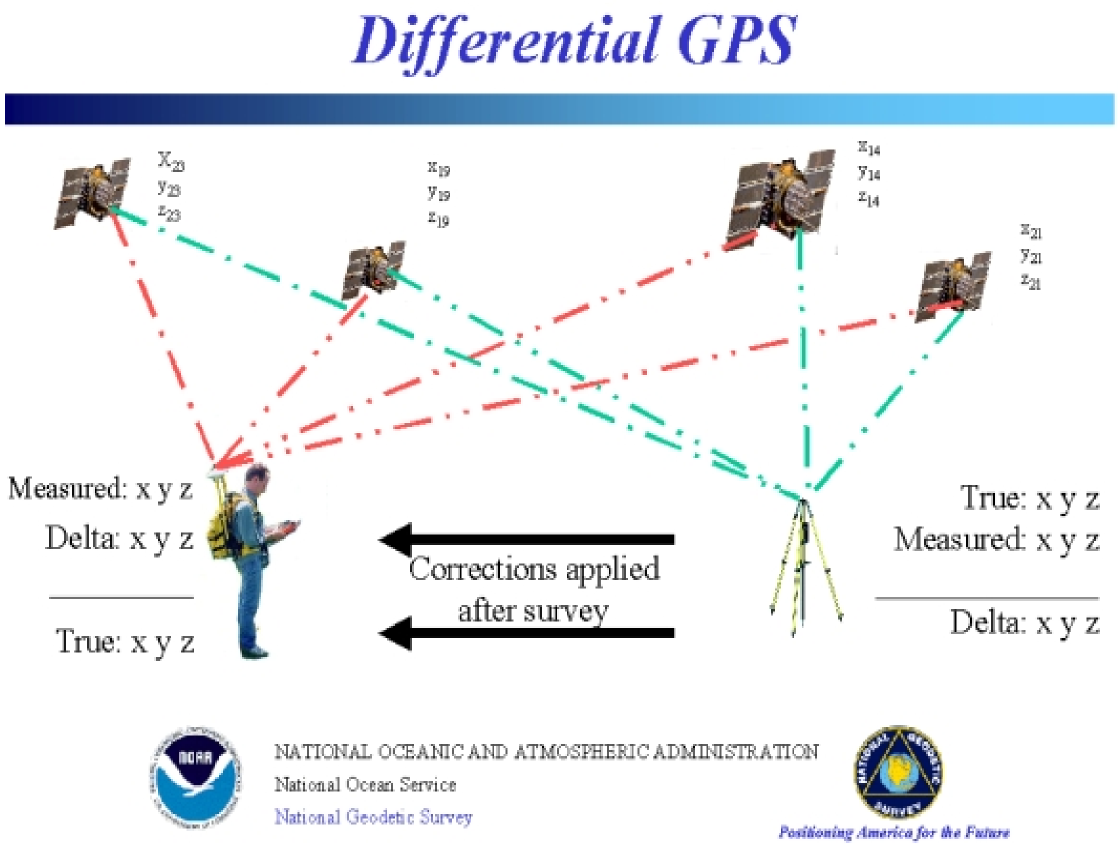


Global Positioning System (GPS) (3)

- **Time synchronization**:
  - **atomic clocks on each satellite**
  - monitoring them from different ground stations.
- Ultra-precision time synchronization is extremely important
  - electromagnetic radiation propagates at light speed
- **Light travels roughly 0.3 m per nanosecond**
  - position accuracy proportional to precision of time measurement
- **Real time update of the exact location of the satellites:**
  - monitoring the satellites from a number of widely distributed ground stations
  - master station analyses all the measurements and transmits the actual position to each of the satellites
- **Exact measurement of the time of flight**
  - the receiver correlates a pseudocode with the same code coming from the satellite
  - The delay time for best correlation represents the time of flight.
  - **quartz clock on the GPS receivers are not very precise**
  - the range measurement with four satellite allows to identify the three values (x, y, z) for the position and the clock correction ΔT
- Recent commercial GPS receiver devices allows position accuracies down to a couple meters.

Differential Global Positioning System (dGPS) (4)

- DGPS requires that a GPS receiver, known as the **base station**, be set up on a **precisely known location.** The base station receiver calculates its position based on satellite signals and compares this location to the known location. The difference is applied to the GPS data recorded by the roving GPS receiver
- **position accuracies in sub-meter to cm range**



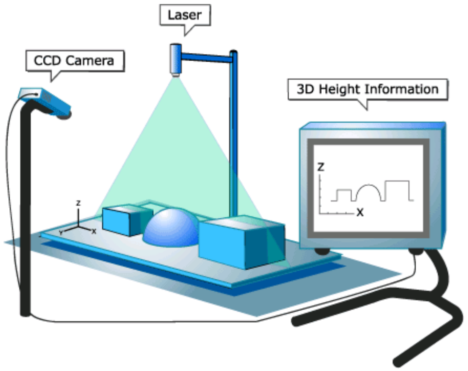
Range sensors

- Sonar

- Laser range finder

- Time of Flight Camera

- Structured light



Range Sensors (time of flight) (1)

- Large range distance measurement → thus called range sensors
- Range information:
  - key element for localization and environment modeling
- Ultrasonic sensors as well as laser range sensors make use of propagation speed of sound or electromagnetic waves respectively.
- The traveled distance of a sound or electromagnetic wave is given by

$$d = c \cdot t$$

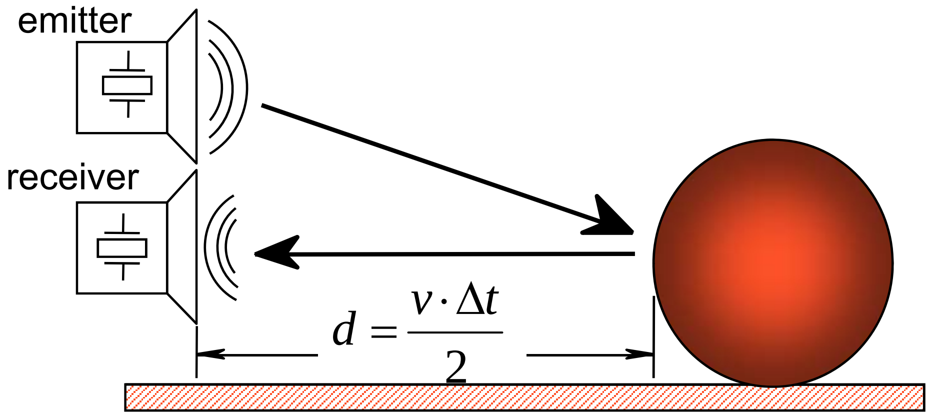
  - d = distance traveled (usually round-trip)
  - c = speed of wave propagation
  - t = time of flight.

Range Sensors (time of flight) (2)

- It is important to point out
  - **Propagation speed v of sound: 0.3 m/ms**
  - **Propagation speed v of of electromagnetic signals: 0.3 m/ns,**
  - **Electromagnetic signals travel one million times faster.**
  - 3 meters
    - Equivalent to **10 ms** for an ultrasonic system
    - Equivalent to only **10 ns** for a laser range sensor
    - Measuring time of flight with electromagnetic signals is not an easy task
    - laser range sensors expensive and delicate

- The quality of time of flight range sensors mainly depends on:
  - **Inaccuracies** in the time of fight **measurement** (laser range sensors)
  - **Opening angle** of transmitted beam (especially ultrasonic range sensors)
  - Interaction with the target (surface, specular reflections)
  - **Variation of propagation speed (sound)**
  - **Speed of mobile robot and target (if not at stand still)**

Factsheet: Ultrasonic Range Sensor



<http://www.robot-electronics.co.uk/shop/Ultrasonic_Rangers1999.htm>

1. Operational Principle

An ultrasonic pulse is generated by a piezo-electric emitter, reflected by an object in its path, and sensed by a piezo-electric receiver. Based on the speed of sound in air and the elapsed time from emission to reception, the distance between the sensor and the object is easily calculated.

2. Main Characteristics

- Precision influenced by angle to object (as illustrated on the next slide)
- Useful in ranges from several cm to several meters
- **Typically relatively inexpensive**

3. Applications

- Distance measurement (also for transparent surfaces)
- Collision detection

Ultrasonic Sensor (time of flight, sound) (1)

- transmit a packet of (ultrasonic) pressure waves
- distance *d* of the echoing object can be calculated based on the propagation speed of sound *c* and the time of flight *t*.

$$d = \frac{c \cdot t}{2}$$

- The speed of sound *c* (340 m/s) in air is given by

Where $c = \sqrt{\gamma \cdot R \cdot T}$

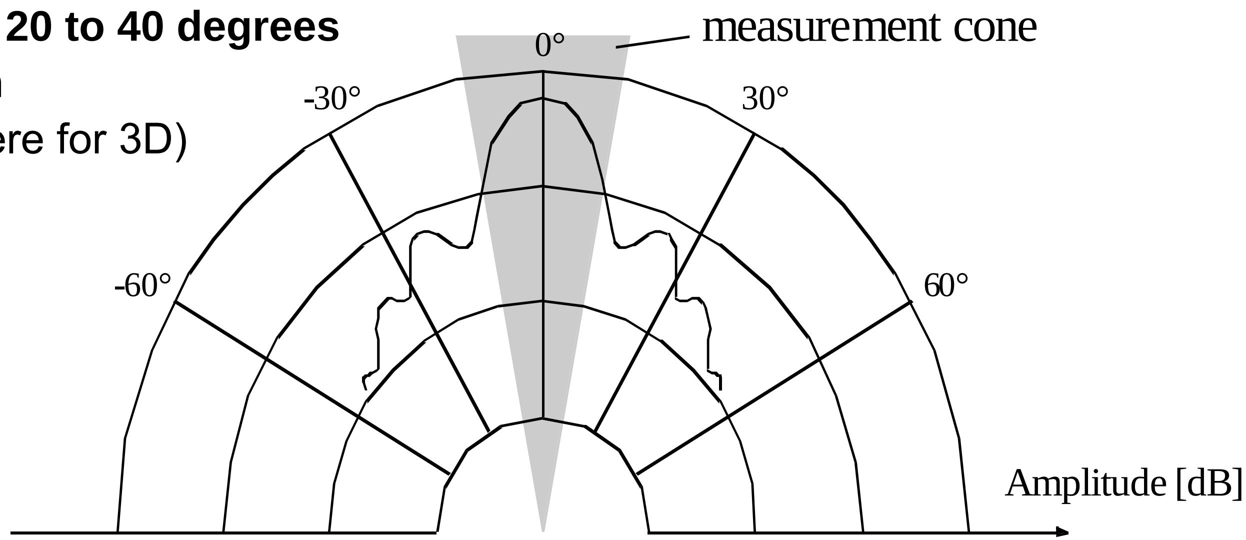
γ : adiabatic index (isentropic expansion factor) - ratio of specific heats of a gas

R: gas constant

T: temperature in degree Kelvin

Ultrasonic Sensor (time of flight, sound) (2)

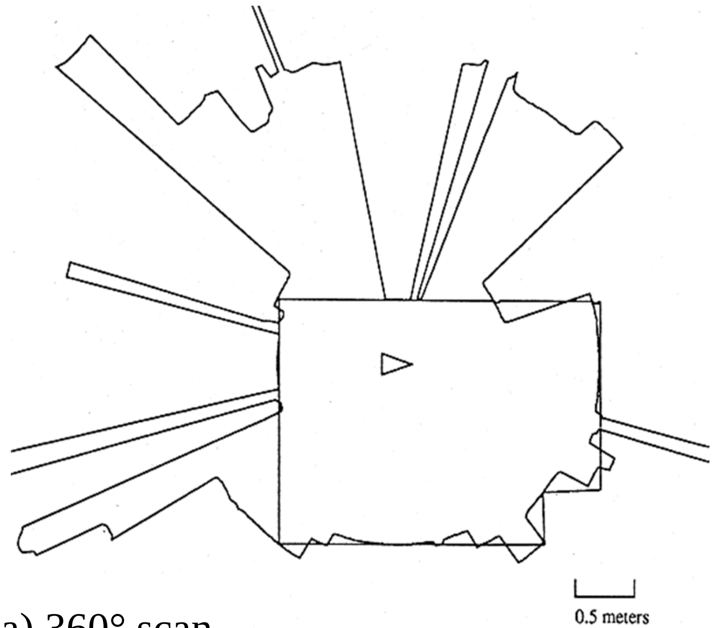
- typical frequency: 40kHz - 180 kHz
  - Lower frequencies correspond to longer maximal sensor range
- generation of sound wave via piezo transducer
  - transmitter and receiver can be separated or not separated
- **Range between 12 cm up to 5 m**
- **Resolution of ~ 2 cm**
- Accuracy 98% → relative error 2%
- sound beam propagates in a cone (*approx*.)
  - **opening angles around 20 to 40 degrees**
  - regions of constant depth
  - segments of an arc (sphere for 3D)



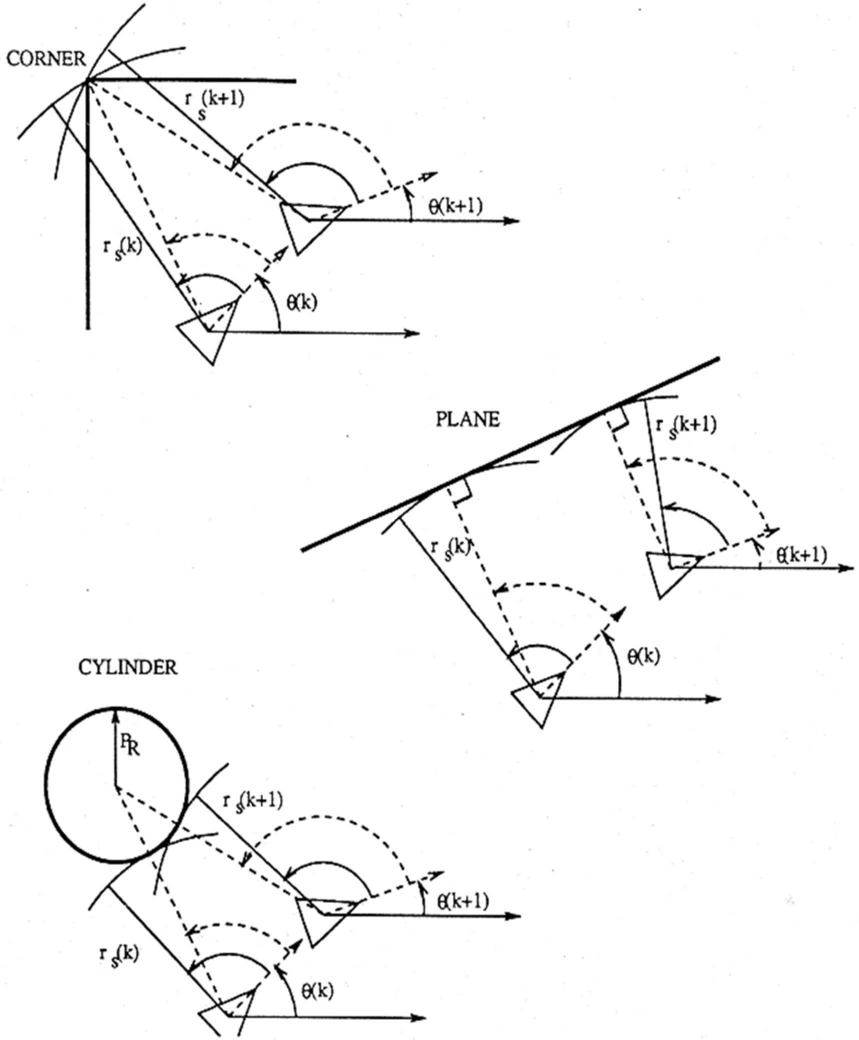
Typical intensity distribution of a ultrasonic sensor

Ultrasonic Sensor (time of flight, sound) (3)

- Other problems for ultrasonic sensors
  - soft surfaces that **absorb** most of the sound energy
  - surfaces that are fare from being perpendicular to the direction of the sound → **specular reflections**



a) 360° scan



b) results from different geometric primitives

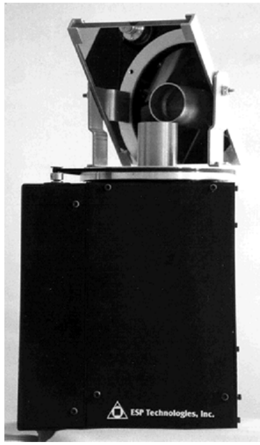
Ultrasonic Sensor (time of flight, sound) (4)

- Bandwidth
  - **measuring the distance to an object that is 3 m away will take such a sensor 20 ms, limiting its operating speed to 50 Hz**. But if the robot has a **ring of 20 ultrasonic sensors**, each firing sequentially and measuring to minimize interference between the sensors, then the ring's cycle time becomes 0.4 seconds => frequency of each one sensor = **2.5 Hz.**
  - This update rate can have a measurable impact on the maximum speed possible while still sensing and avoiding obstacles safely.

Laser Range Sensor (time of flight, electromagnetic) (1)

- Laser range finder are also known as Lidar (LIght Detection And Ranging)

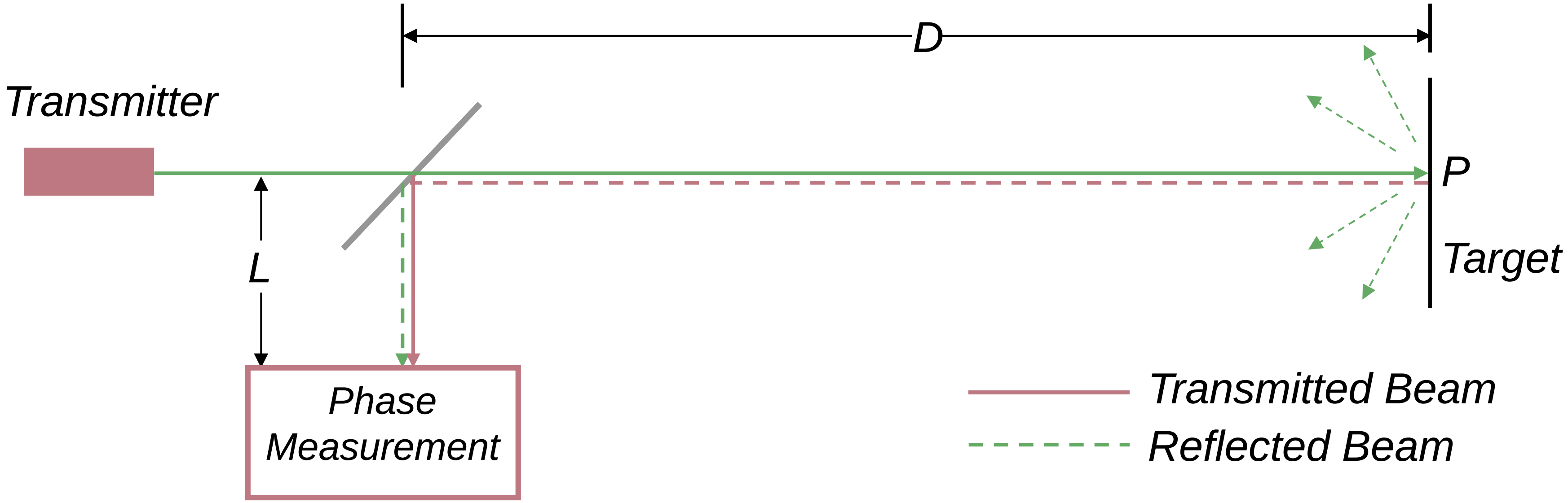
SICK



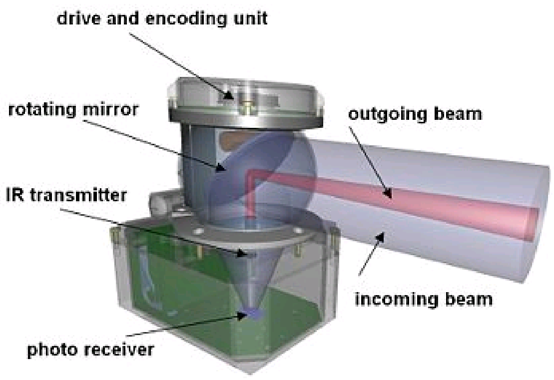
Alaska-IBEO

Hokuyo

4a 66 Laser Range Sensor (time of flight, electromagnetic) (1)



- Transmitted and received beams coaxial
- Transmitter illuminates a target with a collimated laser beam
- Receiver detects the time needed for round-trip
- A mechanical mechanism with a mirror sweeps
  - 2D or 3D measurement

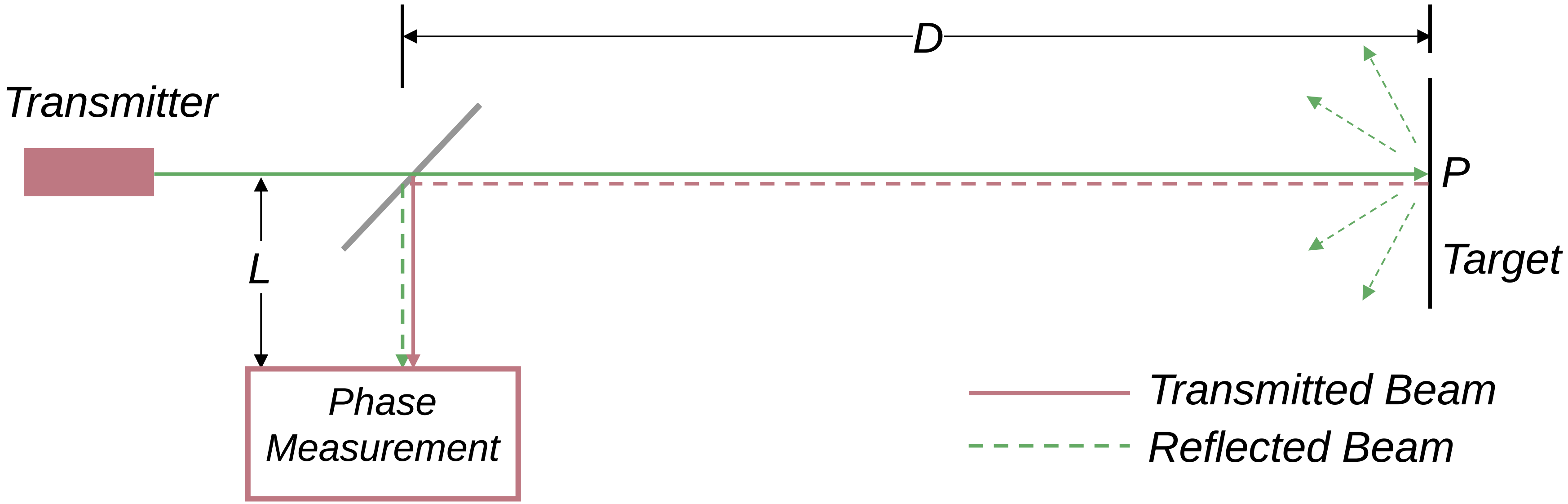


Laser Range Sensor (time of flight, electromagnetic) (2)

- Operating Principles:
  - Pulsed laser (today the standard)
    - measurement of elapsed time directly
    - resolving picoseconds
  - Phase shift measurement to produce range estimation
    - technically easier than the above method

Laser Range Sensor (time of flight, electromagnetic) (3)

- *Phase-Shift Measurement*



$$D' = L + 2D = L + \frac{\theta}{2\pi}\lambda \qquad \lambda = \frac{c}{f}$$

Where:

c: is the speed of light; *f* the modulating frequency; *D'* the distance covered by the emitted light is.

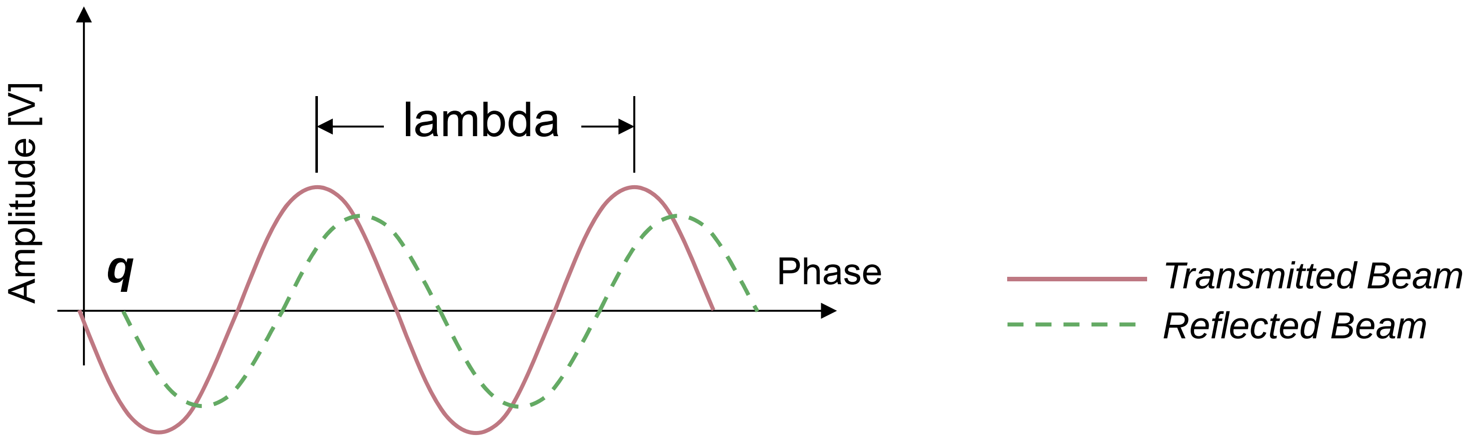
- for f = 5 MHz (as in the A.T&T. sensor), λ = 60 meters

Laser Range Sensor (time of flight, electromagnetic) (4)

- Distance D, between the beam splitter and the target

$$D = \frac{\lambda}{4\pi}\theta$$

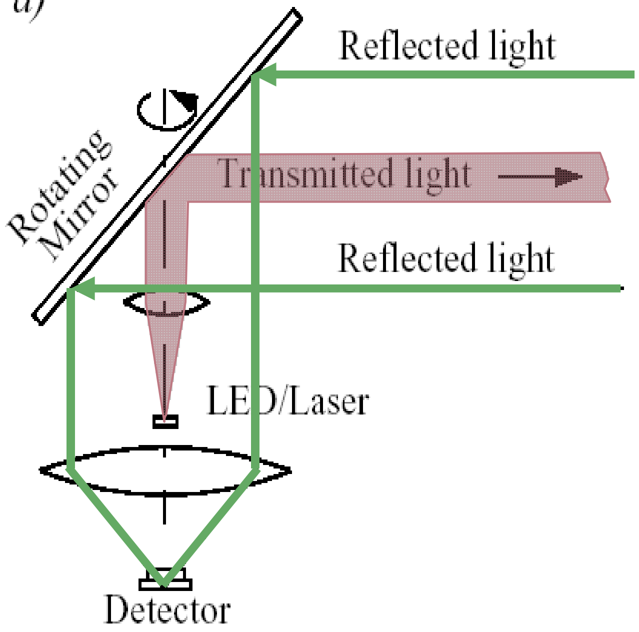
- where
  - θ: phase difference between transmitted and reflected beam
- Theoretically ambiguous range estimates
  - since for example if λ = 60 meters, a target at a range of 5 meters = target at 35 meters



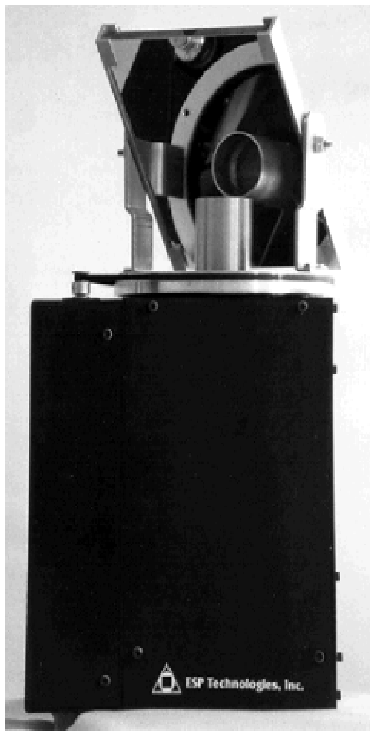
Laser Range Sensor (time of flight, electromagnetic) (5)

- Uncertainty of the range (phase/time estimate) is inversely proportional to the square of the received signal amplitude.
  - Hence dark, distant objects will not produce such good range estimated as closer brighter objects …

a)



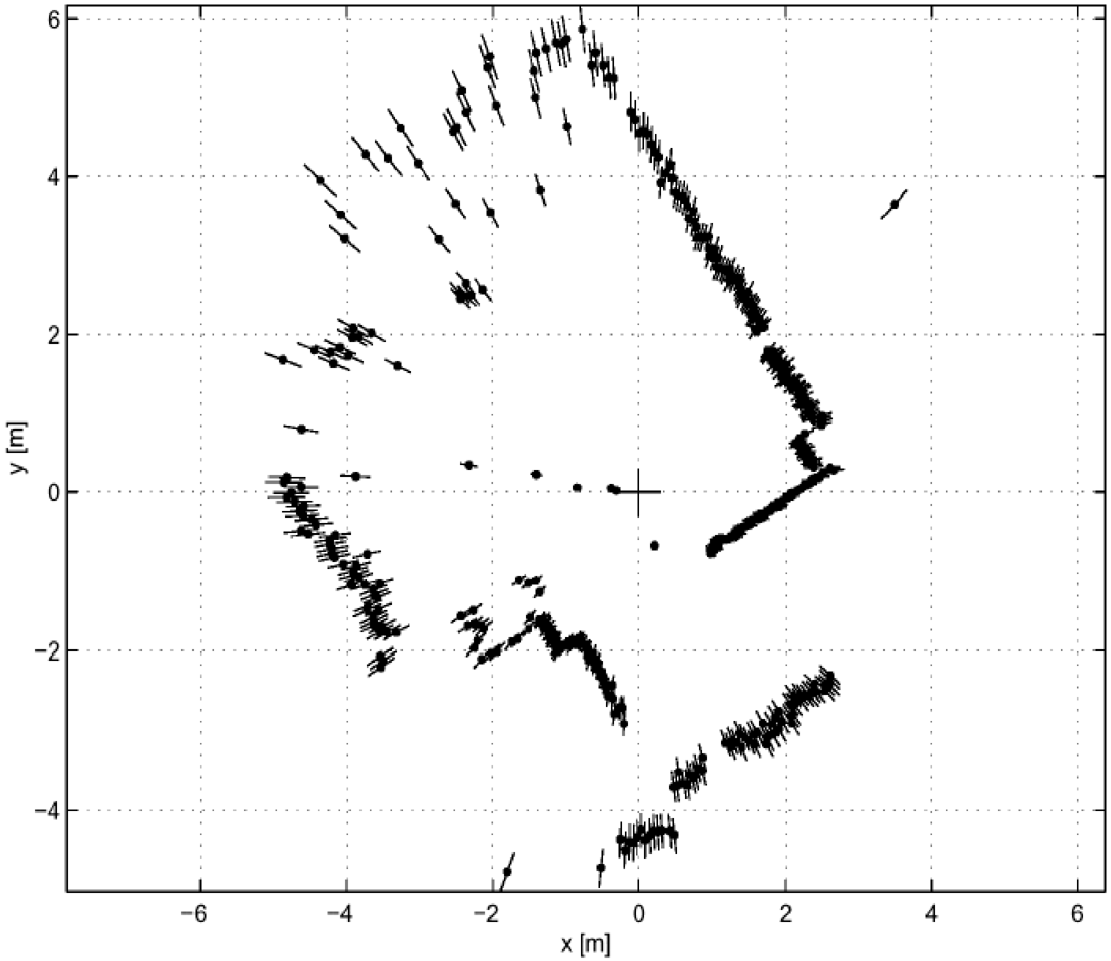
b)



c)

Laser Range Sensor (time of flight, electromagnetic)

- Typical range image of a 2D laser range sensor with a rotating mirror. The length of the lines through the measurement points indicate the uncertainties.

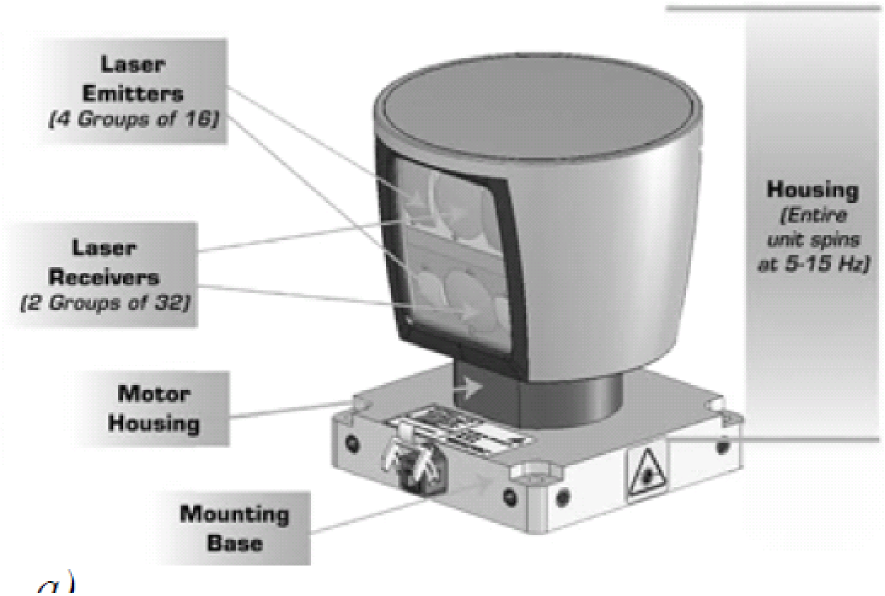


The SICK LMS 200 Laser Scanner

- Angular resolution 0.25 deg
- Depth resolution ranges between 10 and 15 mm and the typical accuracy is 35 mm, over a range from 5 cm up to 20 m or more (up to 80 m), depending on the reflectivity of the object being ranged.
- This device performs seventy five 180-degrees scans per second

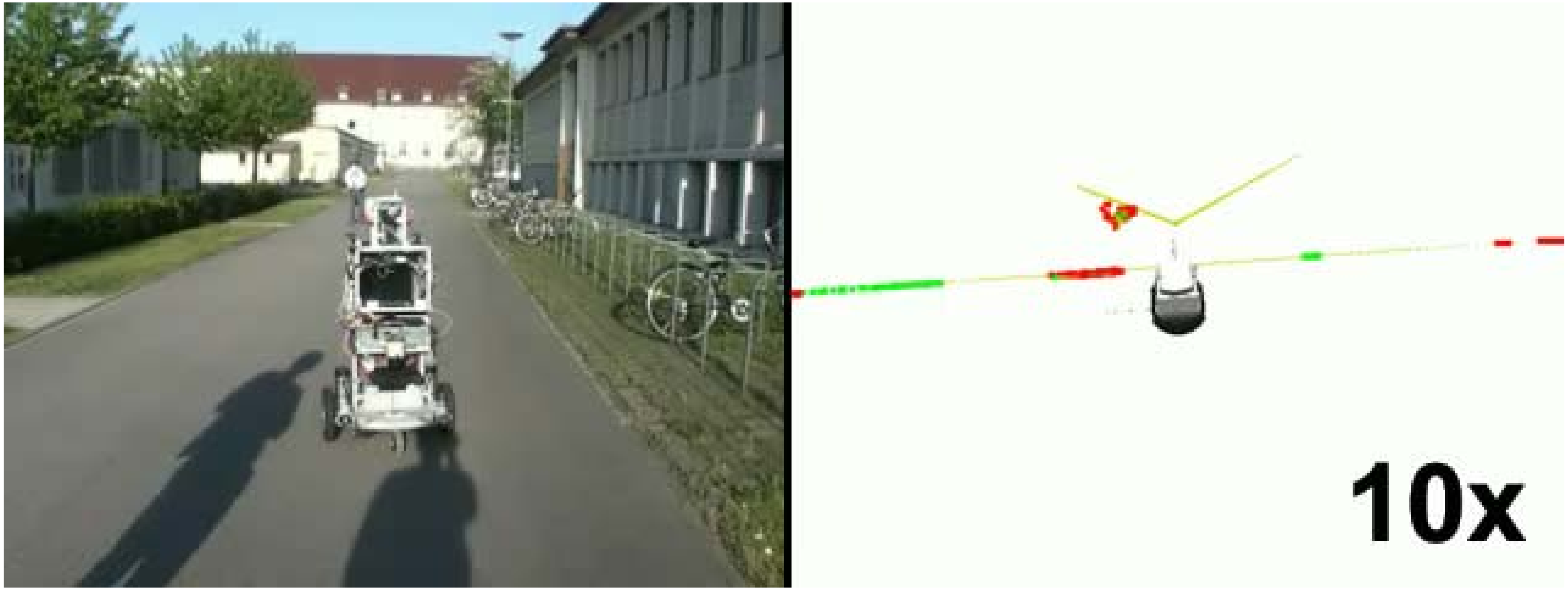
3D Laser Range Finder (2)

- The Velodyne HDL-64E uses 64 laser emitters.
  - Turn-rate up to 15 Hz
  - The field of view is 360° in azimuth and 26.8° in elevation
  - Angular resolution is 0.09° and 0.4° respectively
  - **Delivers over 1.3 million data points per second**
  - The distance accuracy is better than 2 cm and can measure depth up to 50 m
  - This sensor was the primary means of terrain map construction and obstacle detection for all the top DARPA 2007 Urban Challenge teams. However, the Velodyne iscurrently still much more expensive than Sick laser range finders (SICK ~ 5000 Euros, Velodyne ~50,000 Euros!)



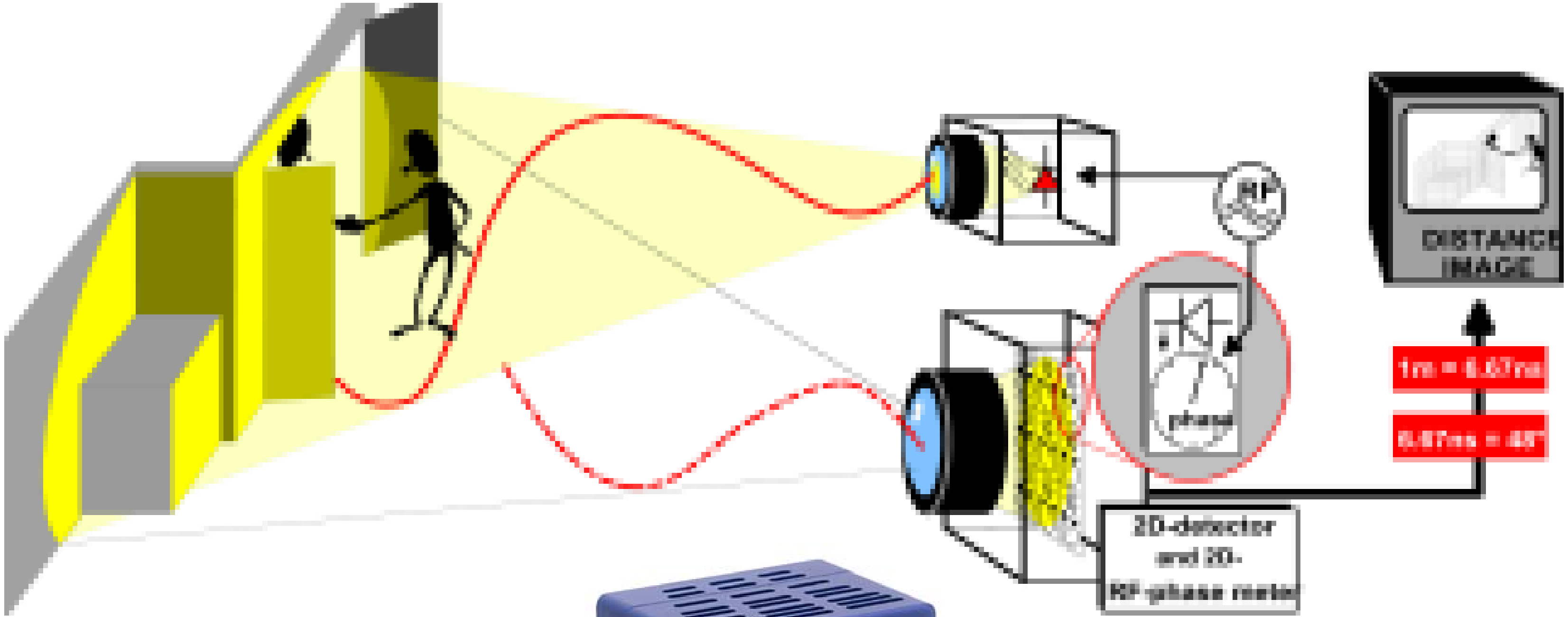
C Carnegie Mellon University

Europa Robot: Obstacle and Terrain Detection



3D Range Sensor (4): Time Of Flight (TOF) camera

- A Time-of-Flight camera (TOF camera, figure) works similarly to a lidar with the advantage that **the whole 3D scene is captured at the same time and that there are no moving parts.** This device uses an infrared lighting source to determine the distance for each pixel of a Photonic Mixer Device (PMD) sensor.



Swiss Ranger 3000
(produced by MESA)

Incremental Object Part Detection

- Range Camera
  - 3D information with high data rate (100 Hz)
  - Compact and easy to manage
  - High, non-uniform measurement noise
  - High outlier rate at jump edges
  - However very low resolution (174x144 pixels)

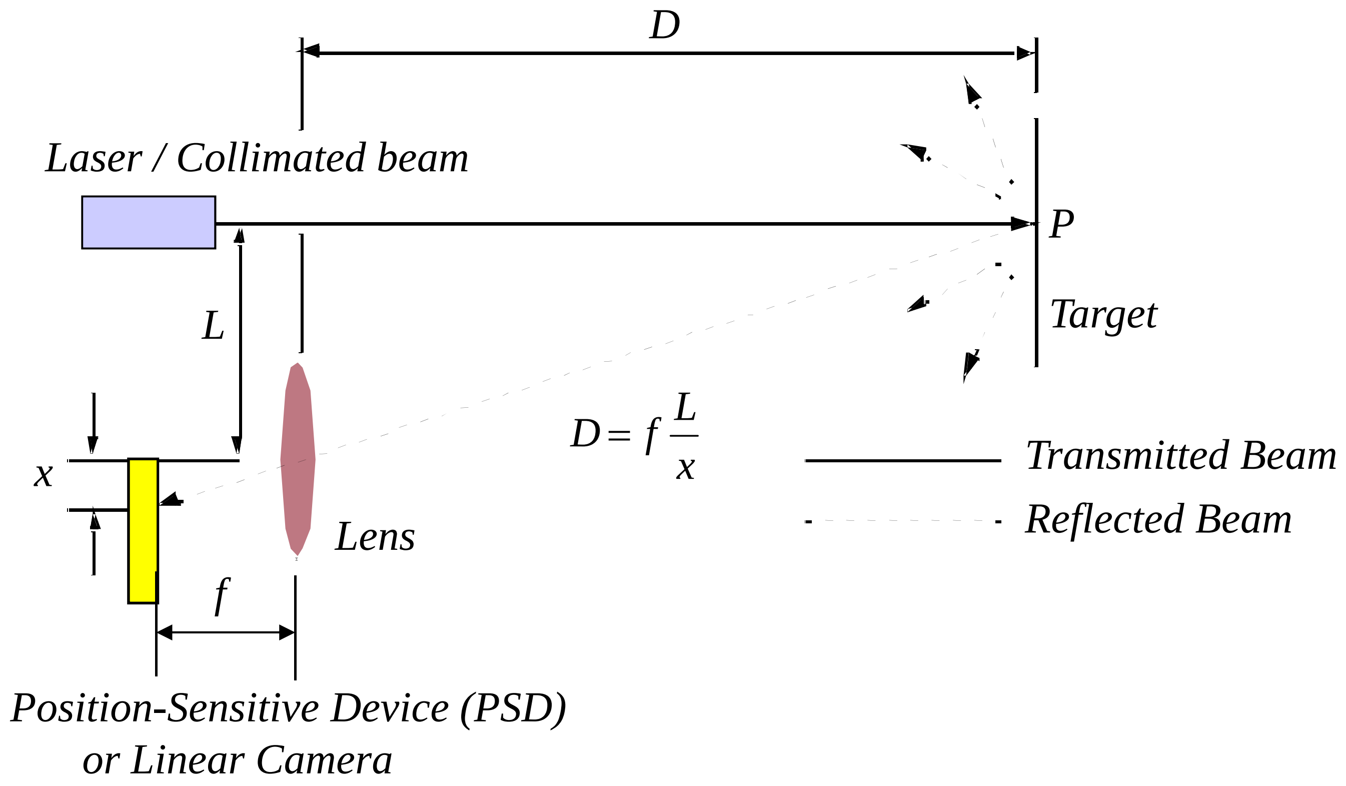
Range Camera SR-3000

C MESA Imaging AG

Triangulation Ranging

- Use of **geometrical properties** of the image to establish a **distance measurement**
- If a well defined light pattern (e.g. point, line) is projected onto the environment.
  - reflected light is then captured by a photo-sensitive line or matrix (camera) sensor device
  - simple triangulation allows to establish a distance.
- If size of a captured object is precisely known
  - triangulation without light projecting

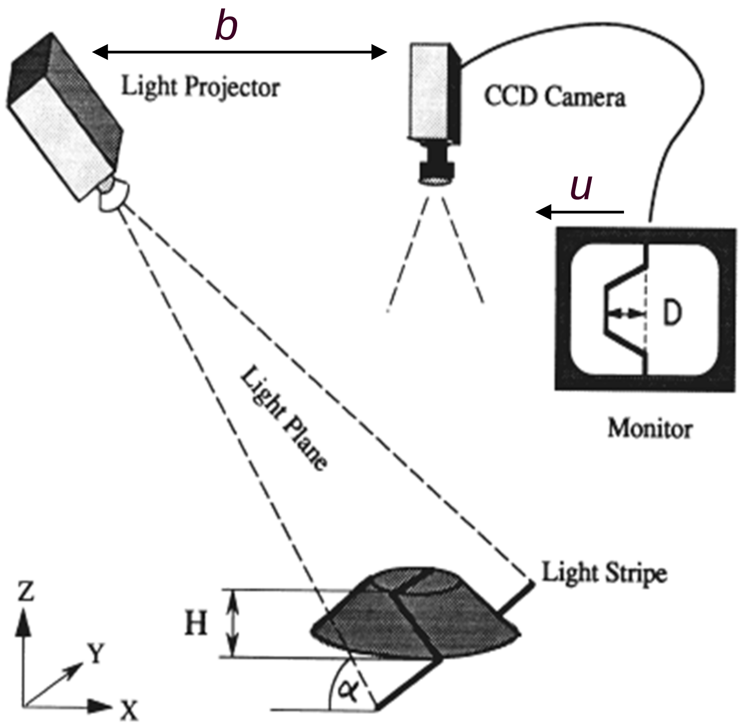
Laser Triangulation (1D)



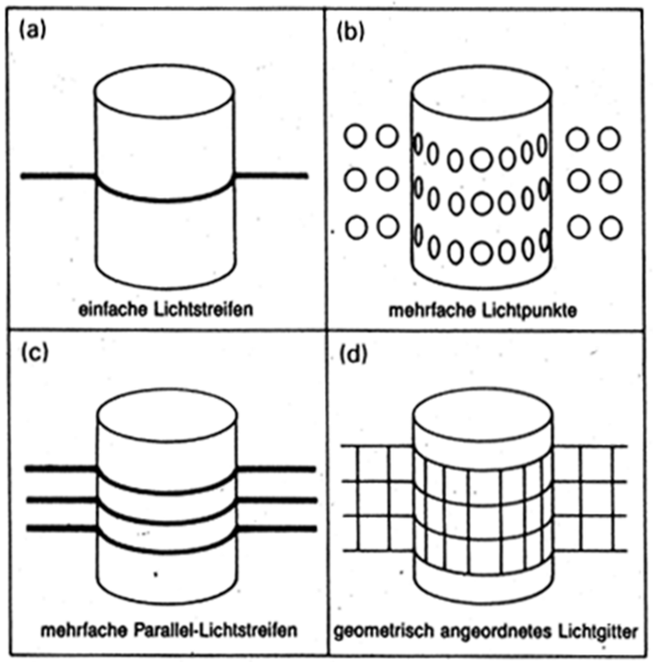
- Principle of 1D laser triangulation: $D = f\frac{L}{x}$

Structured Light (vision, 2D or 3D): Structured Light

a



b



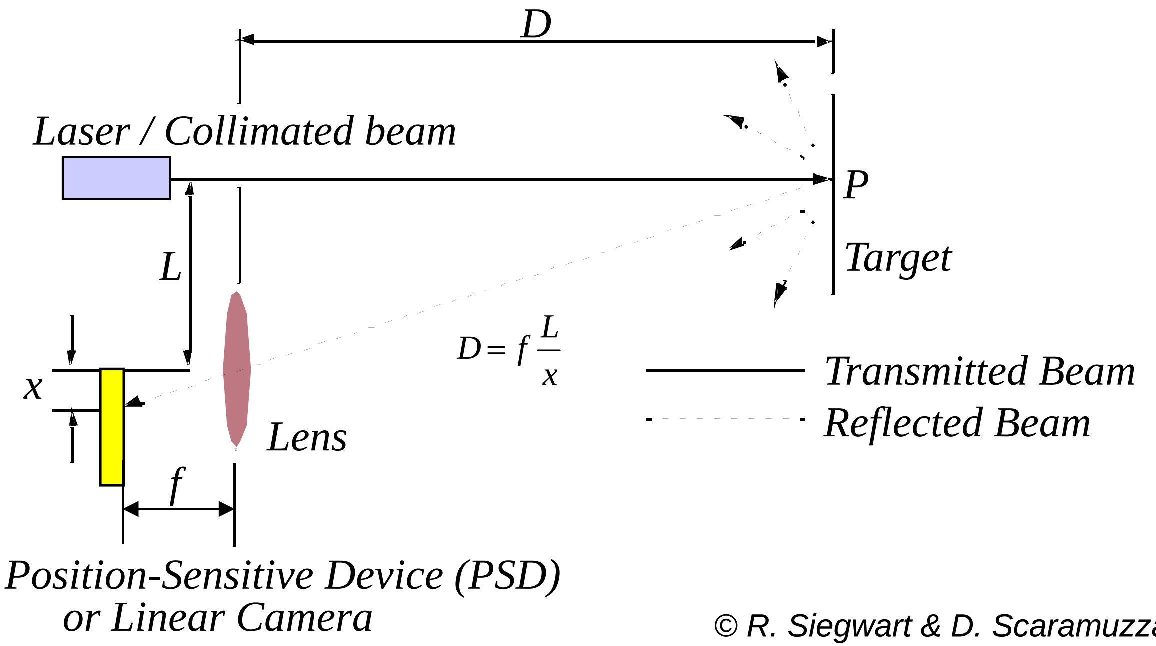
- Eliminate the correspondence problem by projecting structured light on the scene.
- Slits of light or emit collimated light (possibly laser) by means of a rotating mirror.
- Light perceived by camera
- Range to an illuminated point can then be determined from simple geometry.

Structured Light (vision, 2 or 3D)

- Baseline length *L*:
  - the smaller L is the more compact the sensor can be.
  - the larger L is the better the range resolution is.

Note: for large L, the chance that an illuminated point is not visible to the receiver increases.

- Focal length f:
  - larger focal length f can provide
    - either a larger field of view
    - or an improved range resolution
  - however, large focal length means a larger sensor head



Actuators

Ahad Harati

Actuators

- General Purpose
  - Electrical motors in joints (arms, legs, fingers, grippers …)
    - DC, brushless DC, stepper and AC (more on this later)
  - Pneumatic and hydraulics valves
    - A high pressure source produces the power, but usually electrical valves are used for control
- Special Purpose/Other Divers Cases
  - Welding guns, paint sprayers, speakers, lamps and LEDs, CRTs and LCDs

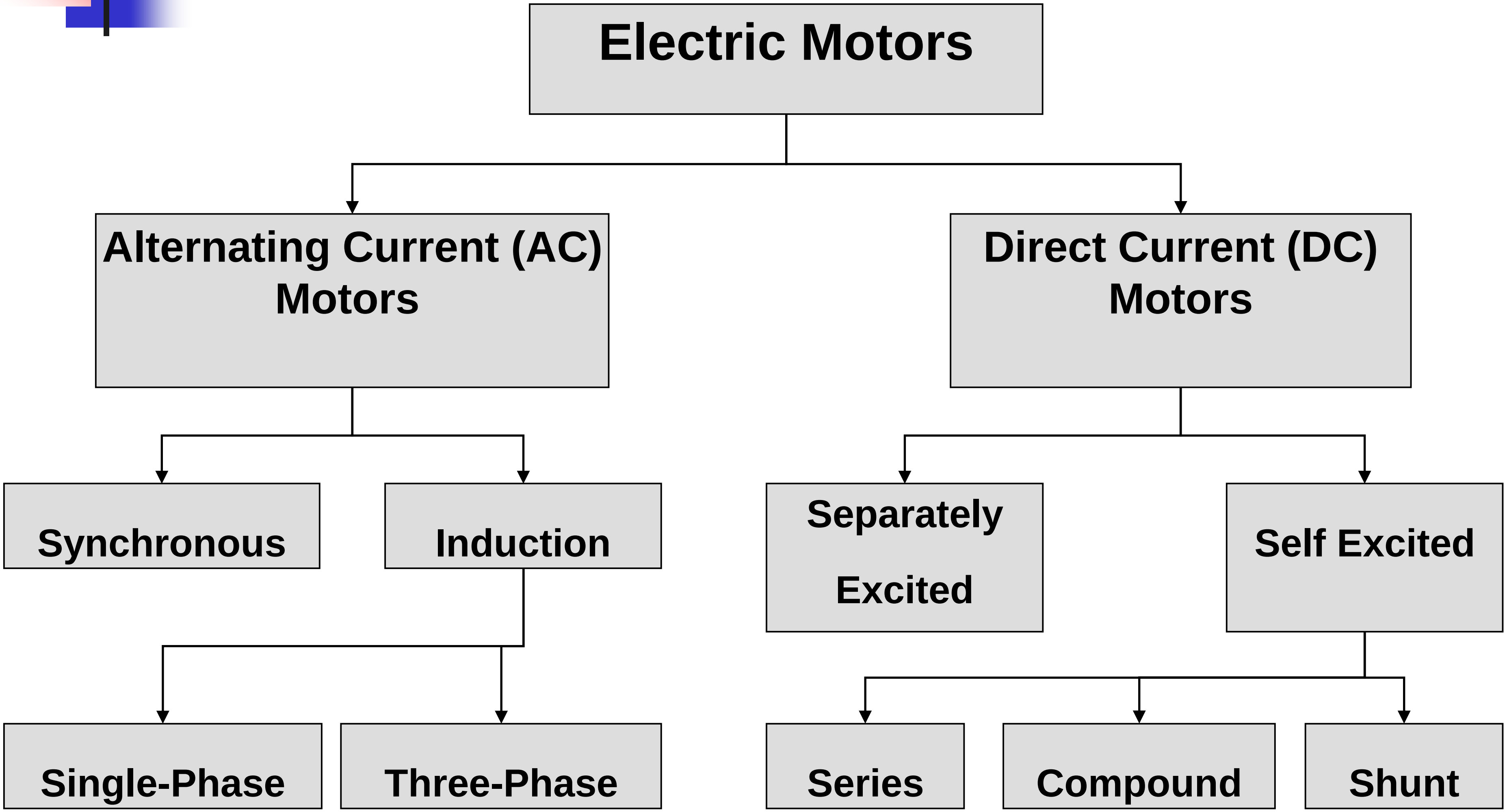
Electrical Motors

- An electric motor is an electromechanical device that converts electrical energy to mechanical energy.
  - The mechanical energy can be used to perform work such as rotating a pump impeller, fan, blower, driving a compressor, lifting materials etc.
  - Consists of two parts: stator and rotor
    - Stator: stationary electrical component
    - Rotor: coils and other rotary parts (may not be the shaft!)
- In Robotics, electrical motors are frequently used because of their compactness, fast response and ease of control.

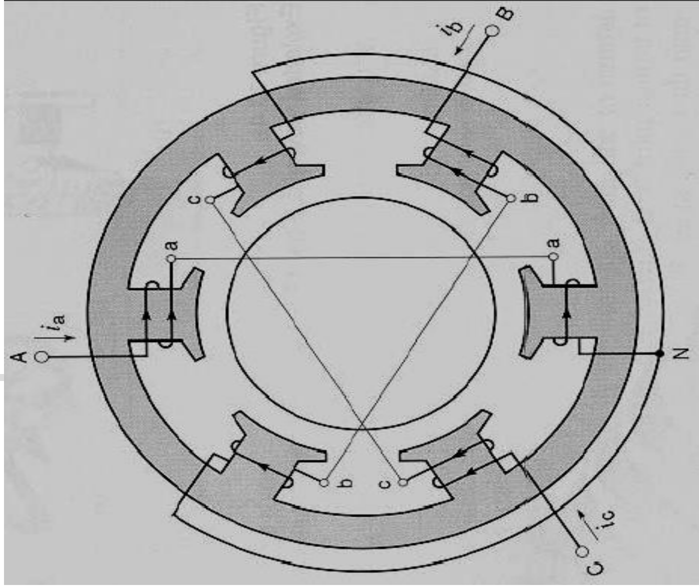
Electrical Motors

- Usually classified based on
  - Type of current (DC/AC)
  - Power class (the amount of power they can generate)
  - Application
- AC Motors
  - Mainly used in industry and home appliances
  - It’s rather difficult to precisely control their speed
- DC Motors
  - The main option in Robotics
  - Easy speed control and feedback based position control

Types of Electric Motors

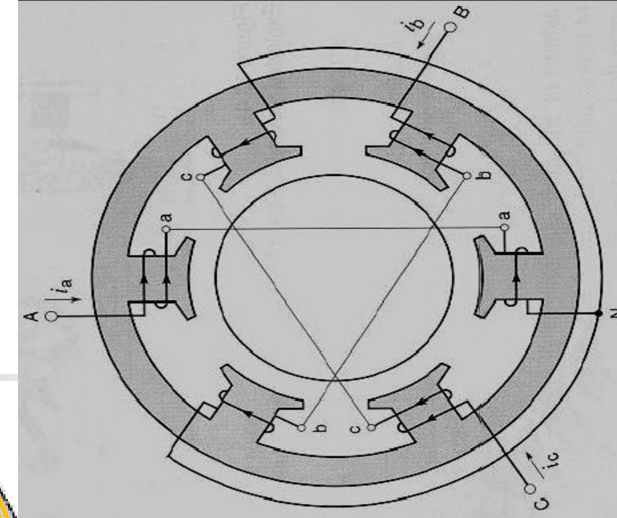


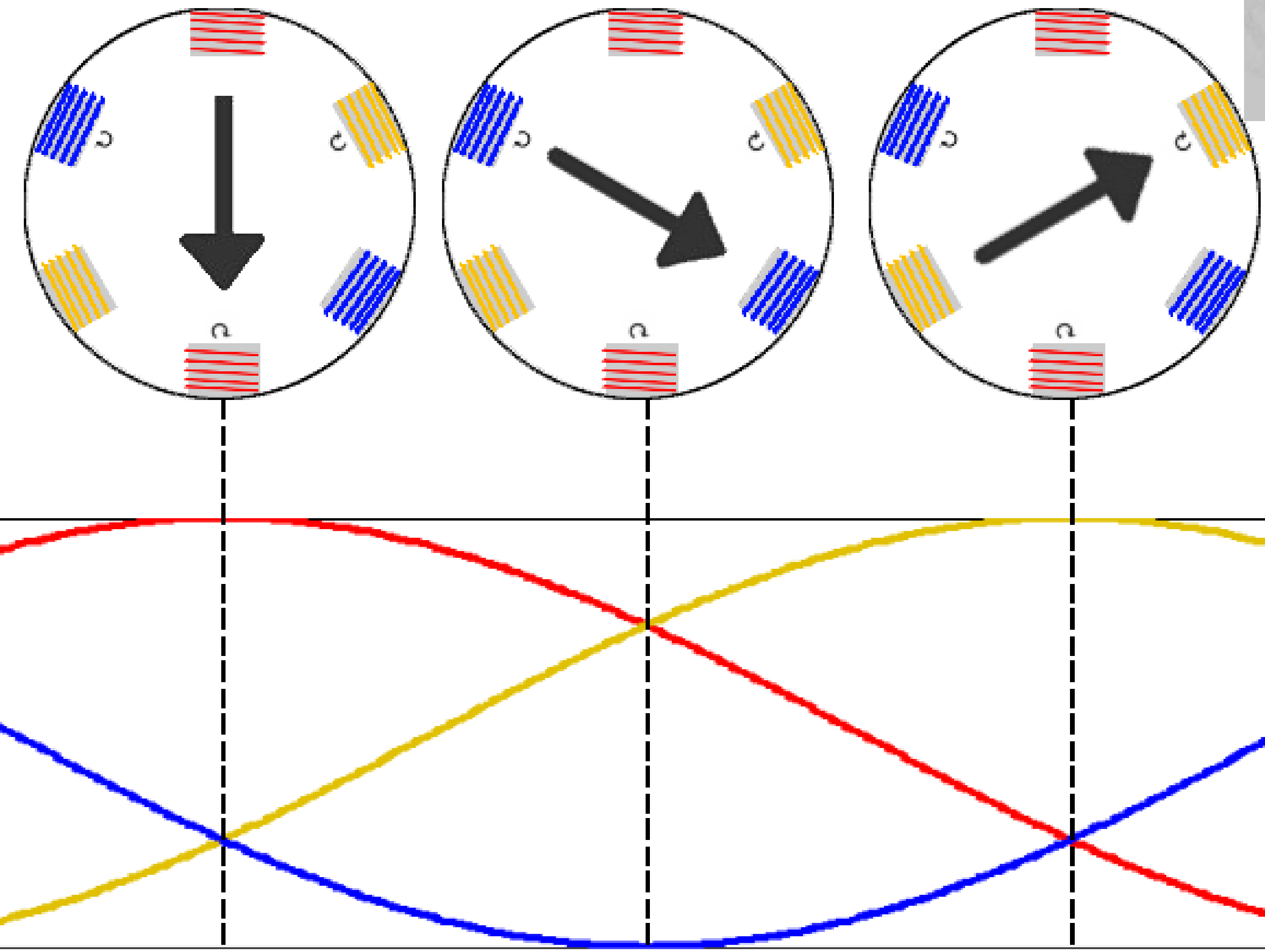
AC Motors



- Consider a simple stator with 3 windings on 6 salient poles
  - The windings are mechanically spaced at 120° from each other and are connected to a 3-phase AC source.
  - AC currents I_a, I_b and I_c will flow in the windings, but will be displaced in time by 120°.
  - Each winding produces its own MMF, which creates a flux across the hollow interior of the stator.
  - The 3 fluxes combine to produce a magnetic field that rotates at the same frequency as the supply.

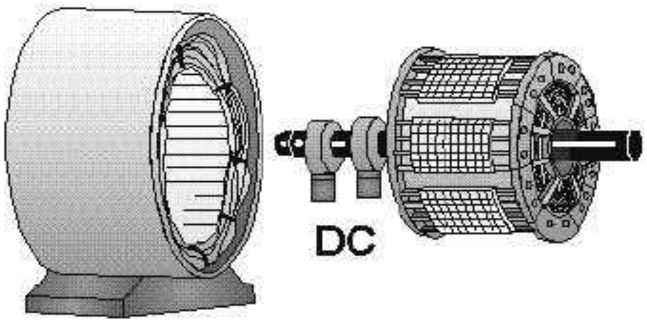
AC Motors: Rotating Flux





AC Motors: Working Principle

rotor may consist of permanent magnets or slip rings and coils



- Synchronous
  - A permanent magnet or DC-driven electromagnet (stationary flux) interacts with rotary flux and causes the rotor to rotate with the same frequency as the rotating flux.
- Induction
  - The magnetic field created by the inductive currents in the conductive parts of the rotor will force the rotor to rotate with a frequency less than the rotating flux.

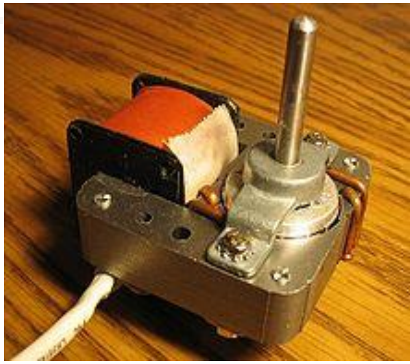
rotor types: squirrel cage, slip ring and solid

Applications of Synchronous Motors

- Low Power (fractional horsepower)
  - Where ever constant speed is necessary or precise speed control is needed: clocks, record players, measurement devices, etc.

- High Power
  - As a very efficient machines for conversion of AC energy to work: ball mills
  - For correction of power factor

Single Phase Induction Motors and Their Applications

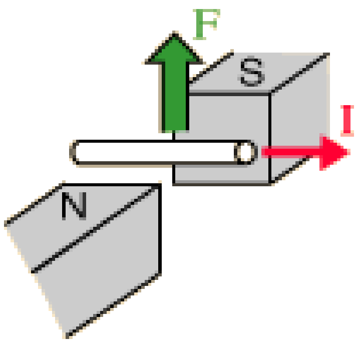


- Shaded-Pole motors
  - A ring of thick copper lags the local magnetic flux enough to create a rotary field.
  - Proper for low power motors with low start torque.
    - Fans and blowers
- Split-Phase motors
  - A separate stator coil is excited with out of phase current
    - The extra winding may remain active in normal operation of the motor to enhance torque
    - The extra winding may serve as a starter and is disconnected by some sort of timer or centrifugal switch, eg.

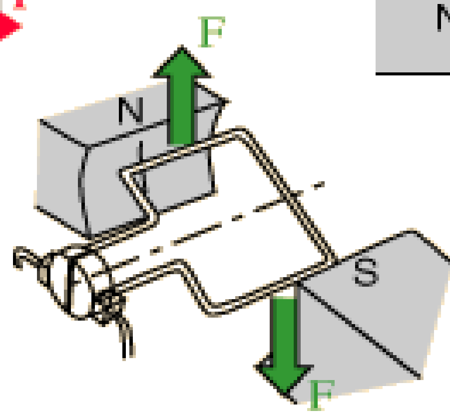
Categorization of Motor Loads

Motor loads	Description	Examples
Constant torque loads	Output power varies but torque is constant	Conveyors, rotary kilns, constant-displacement pumps
Variable torque loads	Torque varies with square of operation speed	Centrifugal pumps, fans
Constant power loads	Torque changes inversely with speed	Machine tools

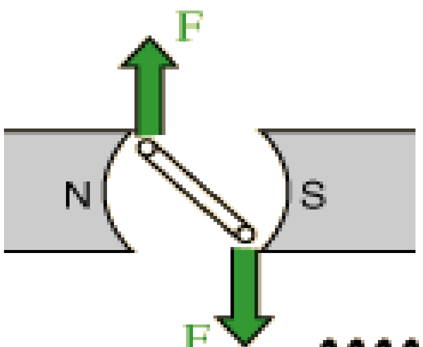
DC Motors: Basic Working Principle



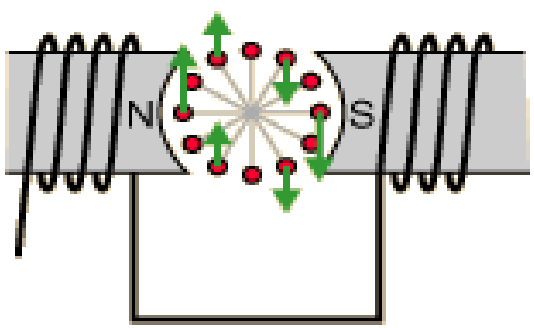
An electric current in a magnetic field will experience a force.



If the current-carrying wire is bent into a loop, then the two sides of the loop which are at right angles to the magnetic field will experience forces in opposite directions.

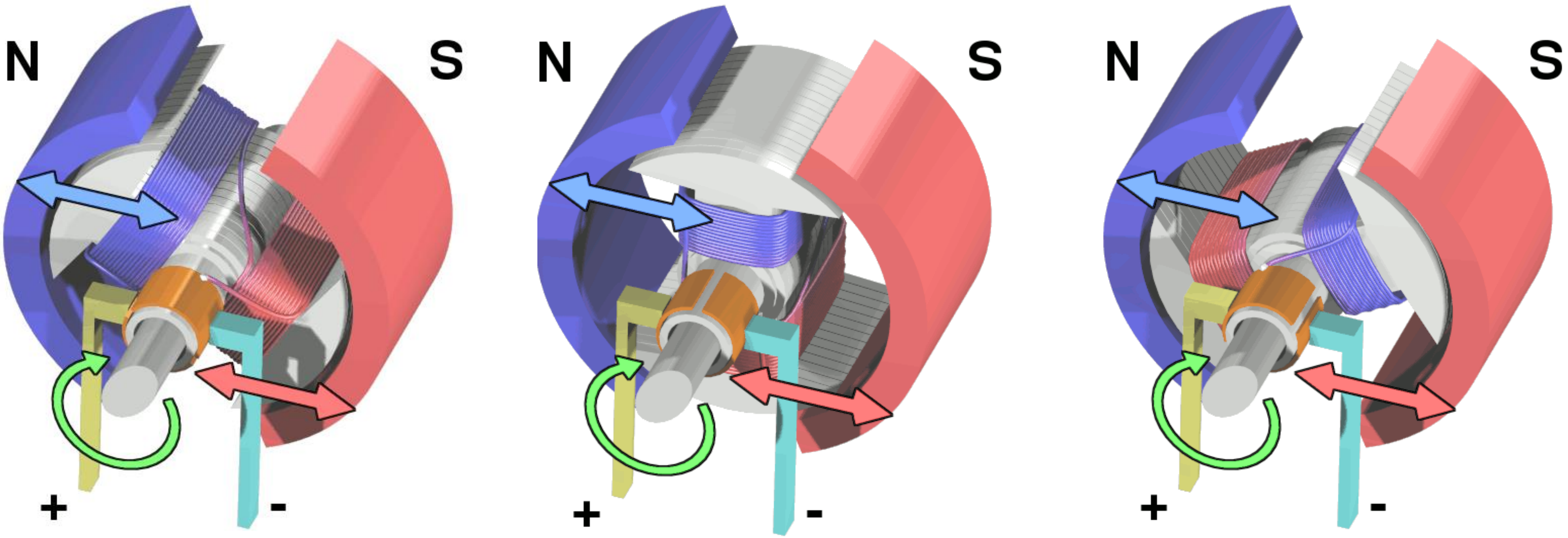


The pair of forces creates a turning influence or torque to rotate the coil.

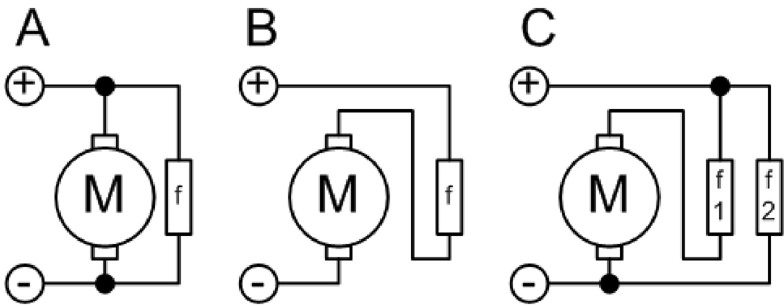


Practical motors have several loops on an armature to provide a more uniform torque and the magnetic field is produced by an electromagnet arrangement called the field coils.

DC Motors:
Basic Working Principle



DC motor: Characteristics

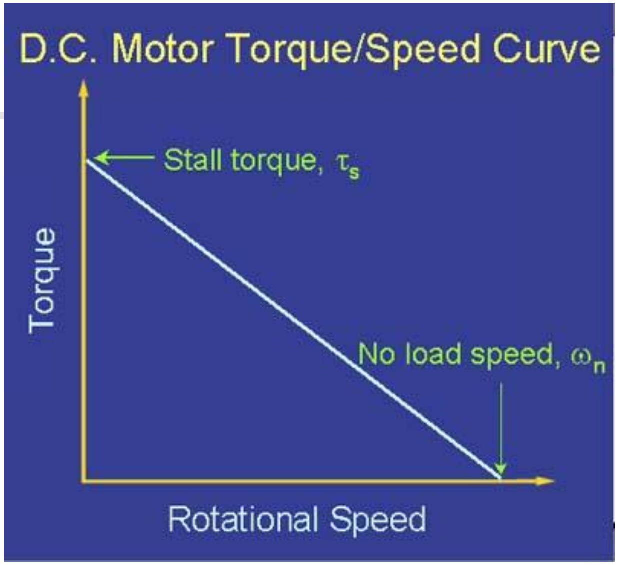


$$N = \frac{K(V - IR)}{\varphi}$$

- Shunt wound motor
  - High-resistance field winding is connected in parallel with the armature (Fig. A)
    - Increased load leads to decreased flux and increased armature current (may be unsuitable for widely-varying loads because of overheating). Hence, extra torque is generated. Decrease in load has inverse effect.
    - It responds to load change by trying to maintain its speed.
- Series wound motor
  - Low-resistance field winding connected in series with the armature (Fig. B)
    - Increased load leads to increased flux which produce larger back EMF in lower speeds. Resultant lower armature current stabilize the motor in lower speed.
    - It responds to increased load by slowing down and trying to maintain its output power.

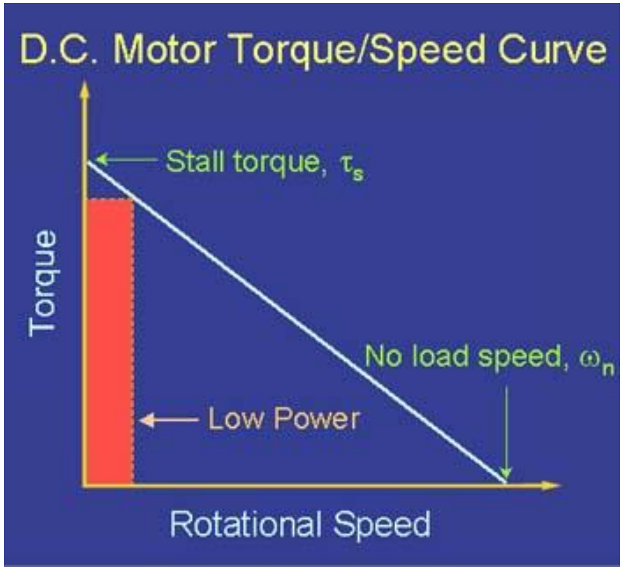
Permanent Magnet DC motor: Characteristics

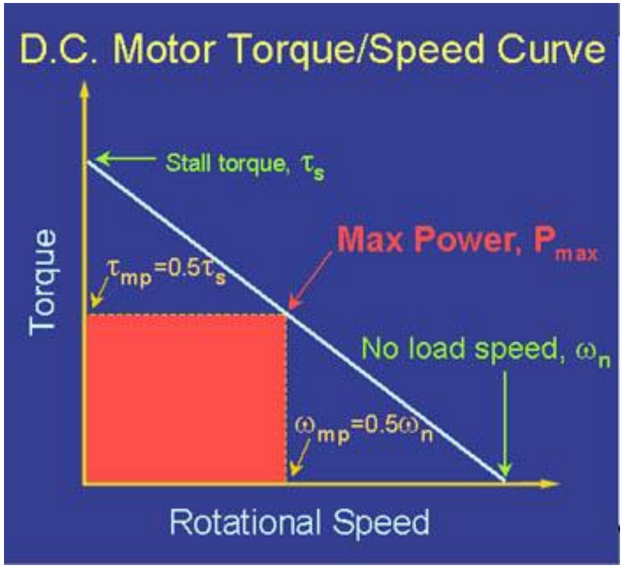
- Permanent magnet motor
  - Locked rotor (stall) torque
  - No-load angular velocity (speed).

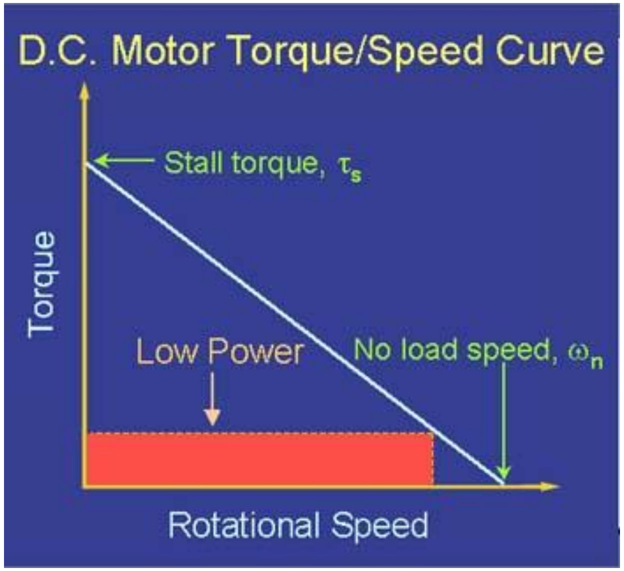


$$\tau_{motor} = \tau_s - \omega\tau_s/\omega_n$$

$$\omega_{motor} = (\tau_s - \tau)\omega_n/\tau_s$$



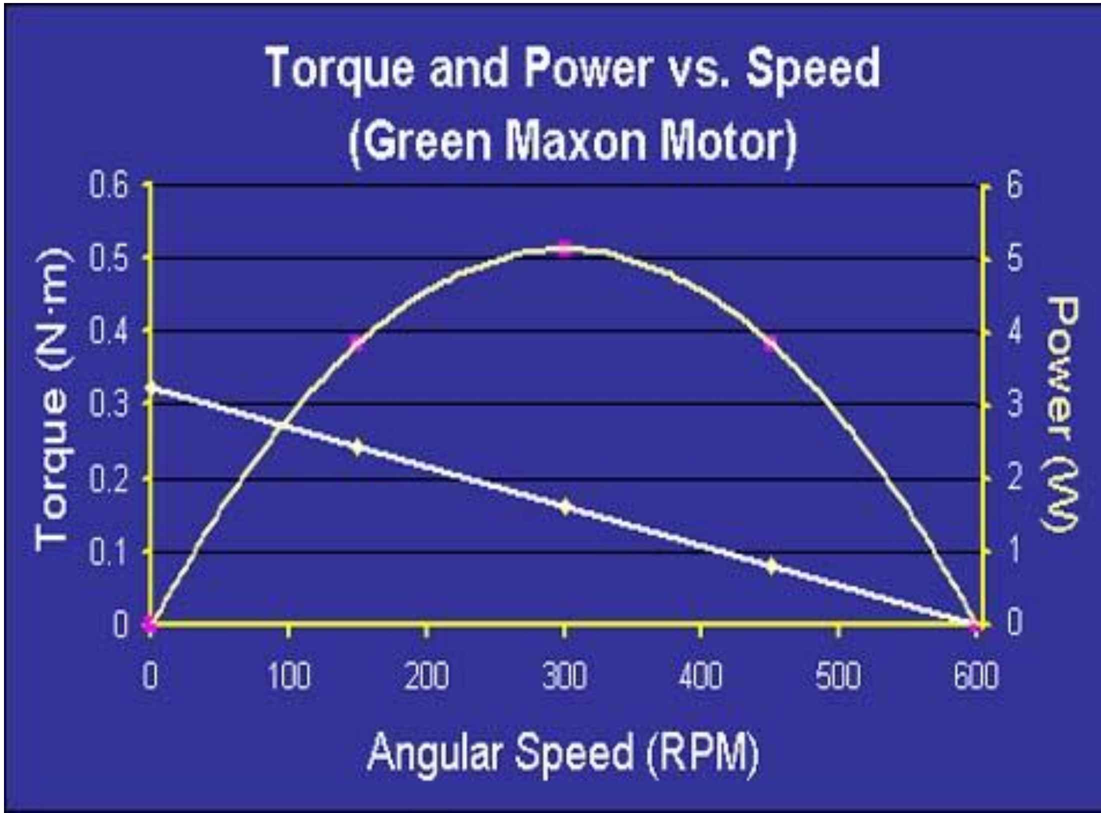




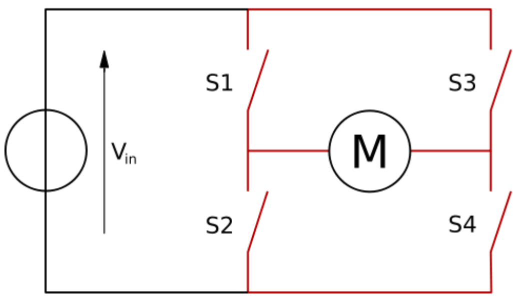
DC Motor: Ouput Power

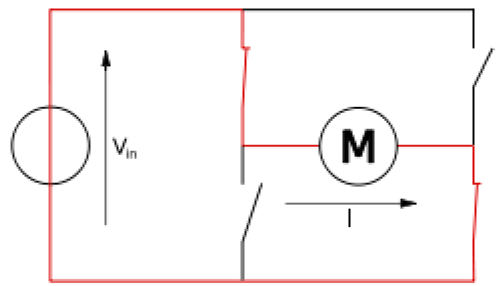
- Max power is reached at half stall torque
- Efficiency is decreased by increasing current (torque)

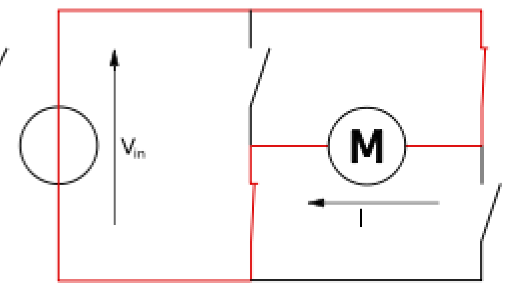
$$P_{motor}(\omega) = -(\tau_s/\omega_n)\,\omega^2 + \tau_s\omega$$
$$P_{motor}(\tau) = -(\omega_n/\tau_s)\,\tau^2 + \omega_n\tau$$



DC Motor: Driver Circuitry



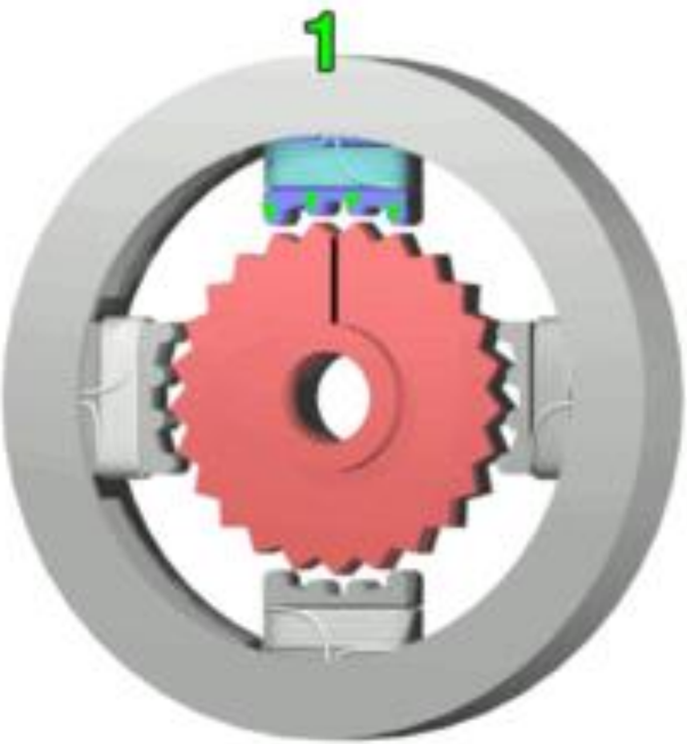


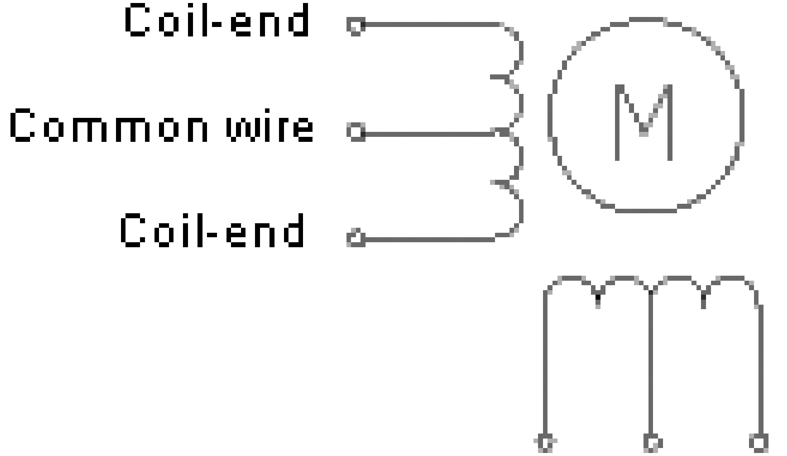


S1	S2	S3	S4	Result
1	0	0	1	Motor moves right
0	1	1	0	Motor moves left
0	0	0	0	Motor free runs
0	1	0	1	Motor brakes
1	0	1	0	Motor brakes

Brushless DC Motors

Stepper Motors

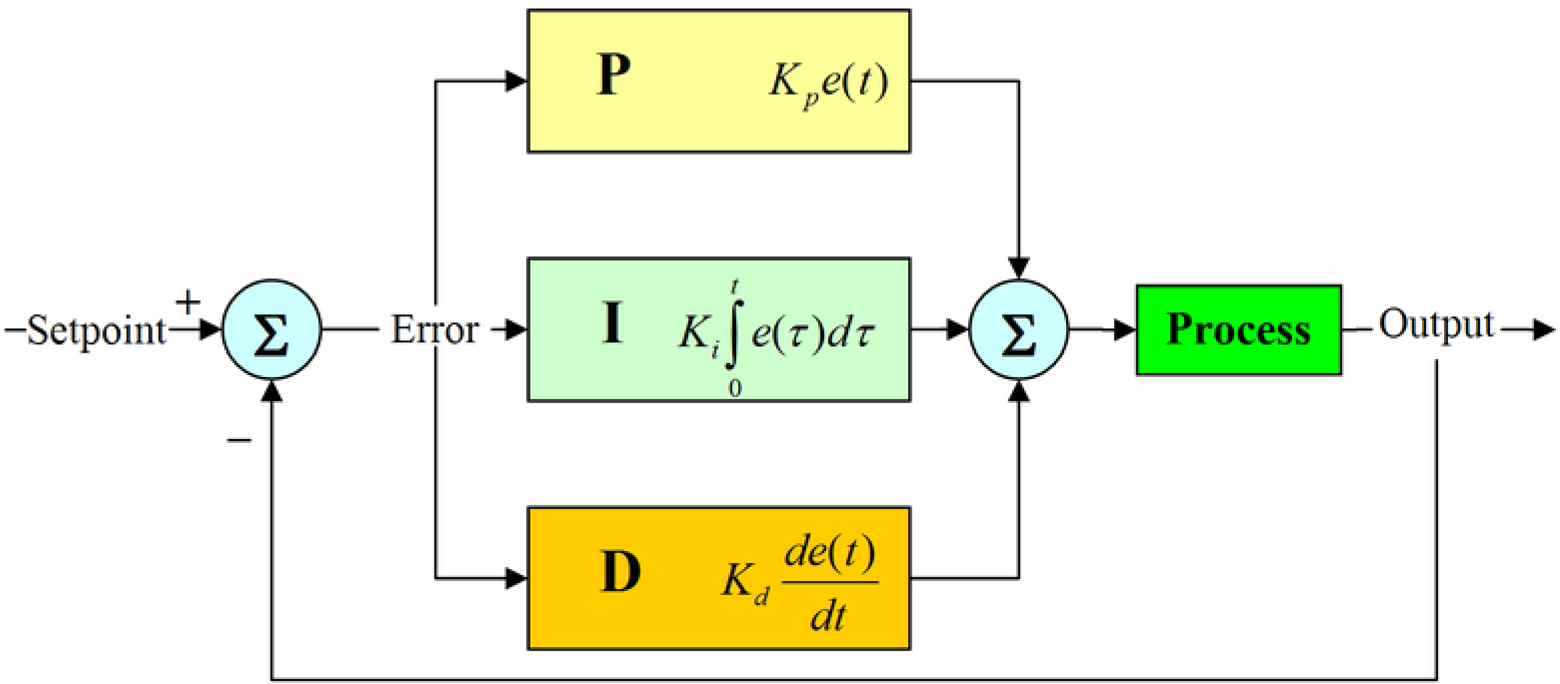




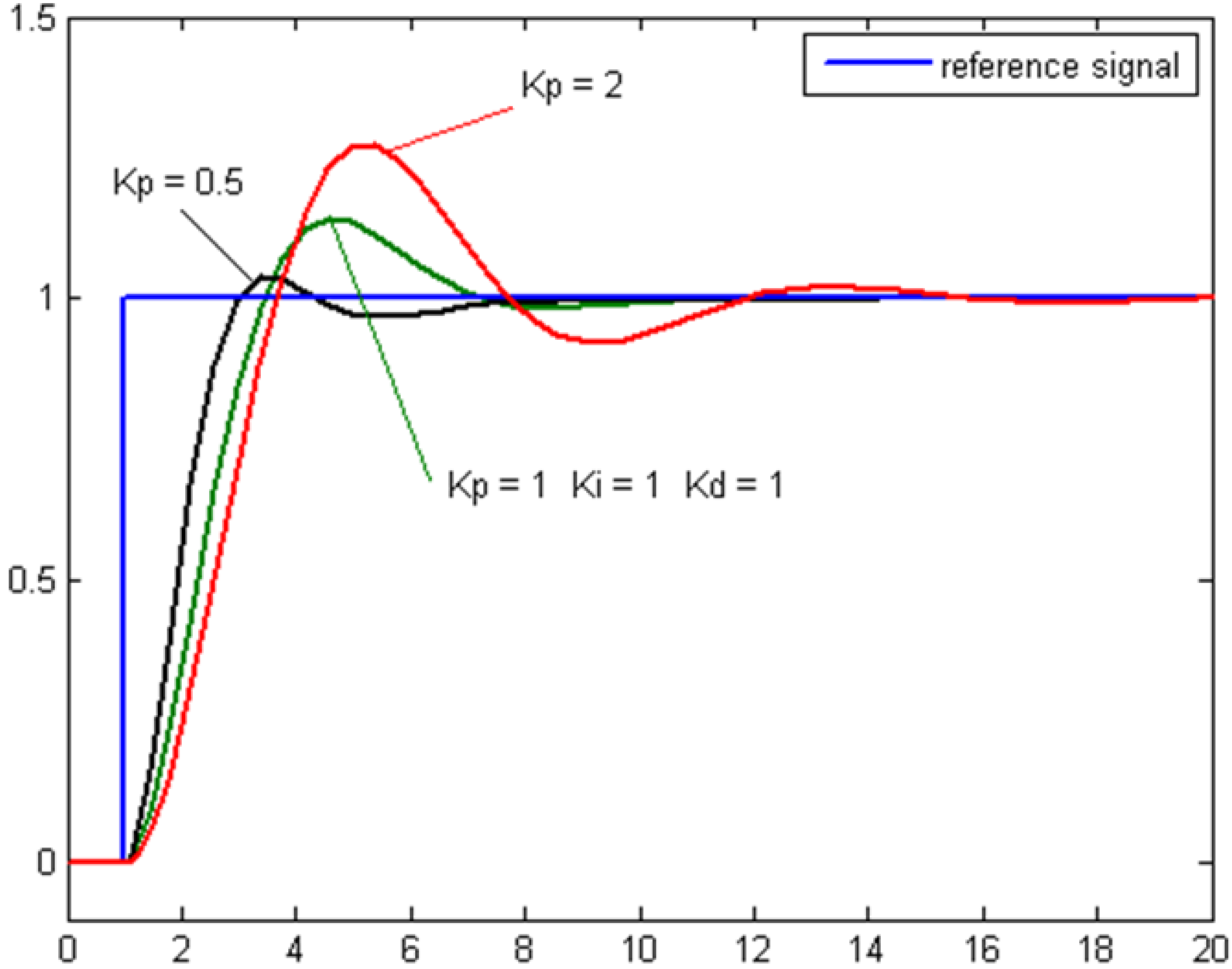
General Comparison

Type	Advantages	Disadvantages	Typical Application	Typical Drive
AC Induction (Shaded Pole)	Least expensive Long life	Rotation slips from frequency Low starting torque	Fans	Uni/Poly-phase AC
AC Induction (Split-Phase capacitor)	High power High starting torque	Rotation slips from frequency	Appliances	Uni/Poly-phase AC
AC Synchronous	Sync with freq Long-life (alternator)	More expensive	Industrial use Clocks, Tape drives Audio turntables	Uni/Poly-phase AC
Stepper DC	Precision positioning High holding torque	Requires a controller	Positioning in printers and floppy drives	DC
Brushless DC	Long lifespan Low maintenance High efficiency	High initial cost Requires a controller	Hard drives CD/DVD players electric vehicles	Controlled DC
Brushed DC	Low initial cost Simple speed control	High maintenance (brushes) Low lifespan	Treadmill exercisers automotive starters	DC or Switched DC (PWM)

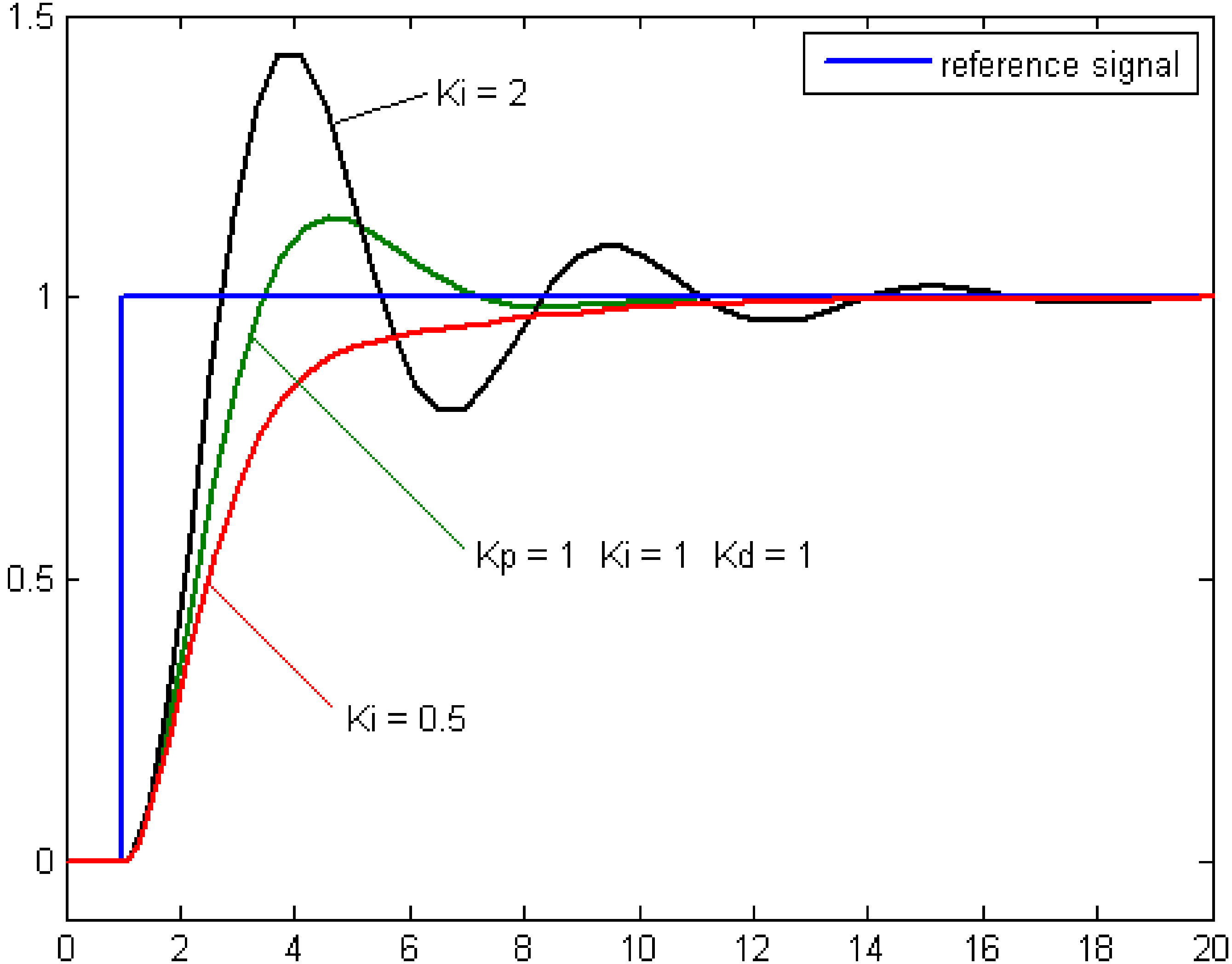
Closed Loop Control



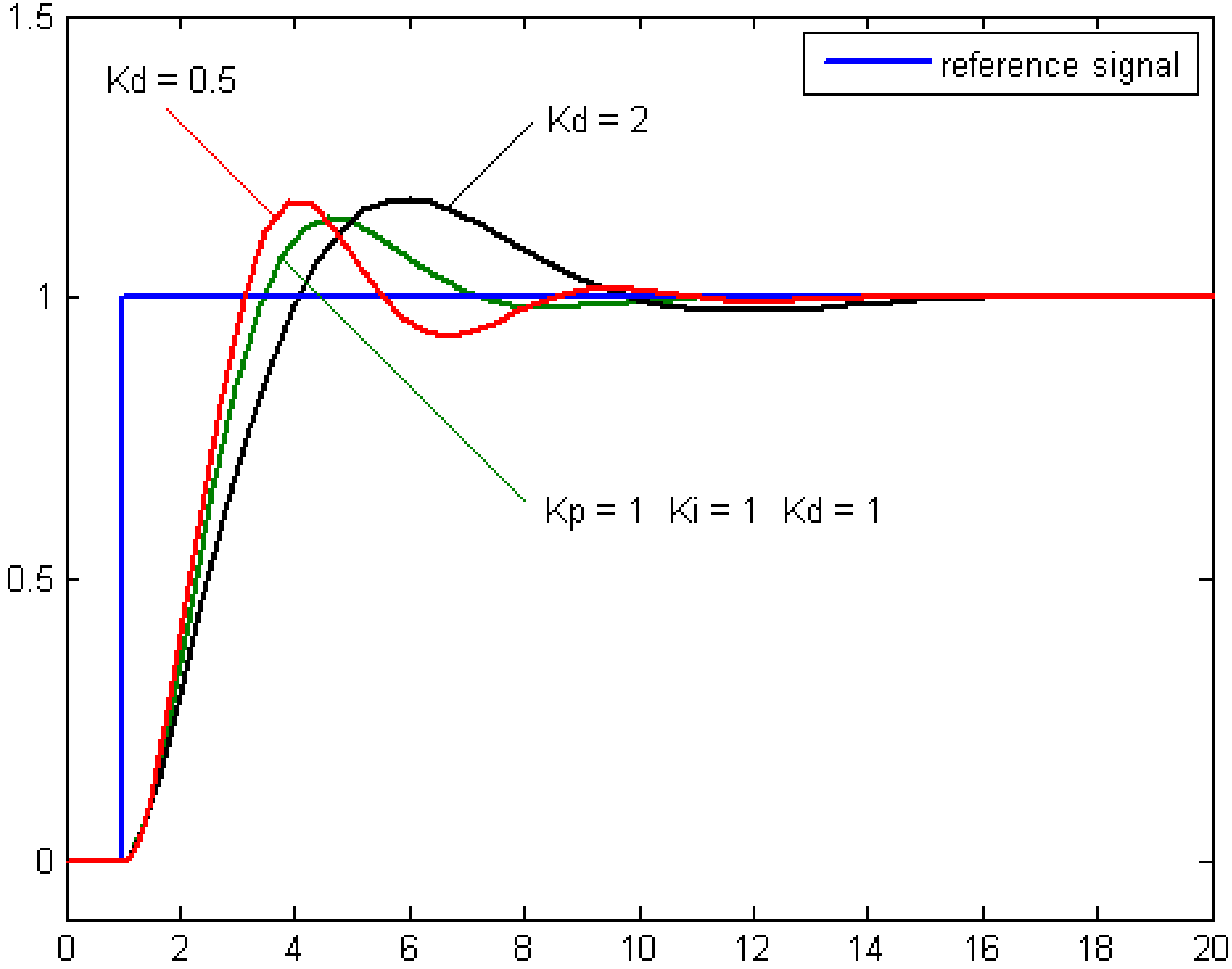
Proportional Controller



Integrative Controller



Derivative Controller



هوش مصنوعی پیشرفته

Advanced Artificial Intelligence

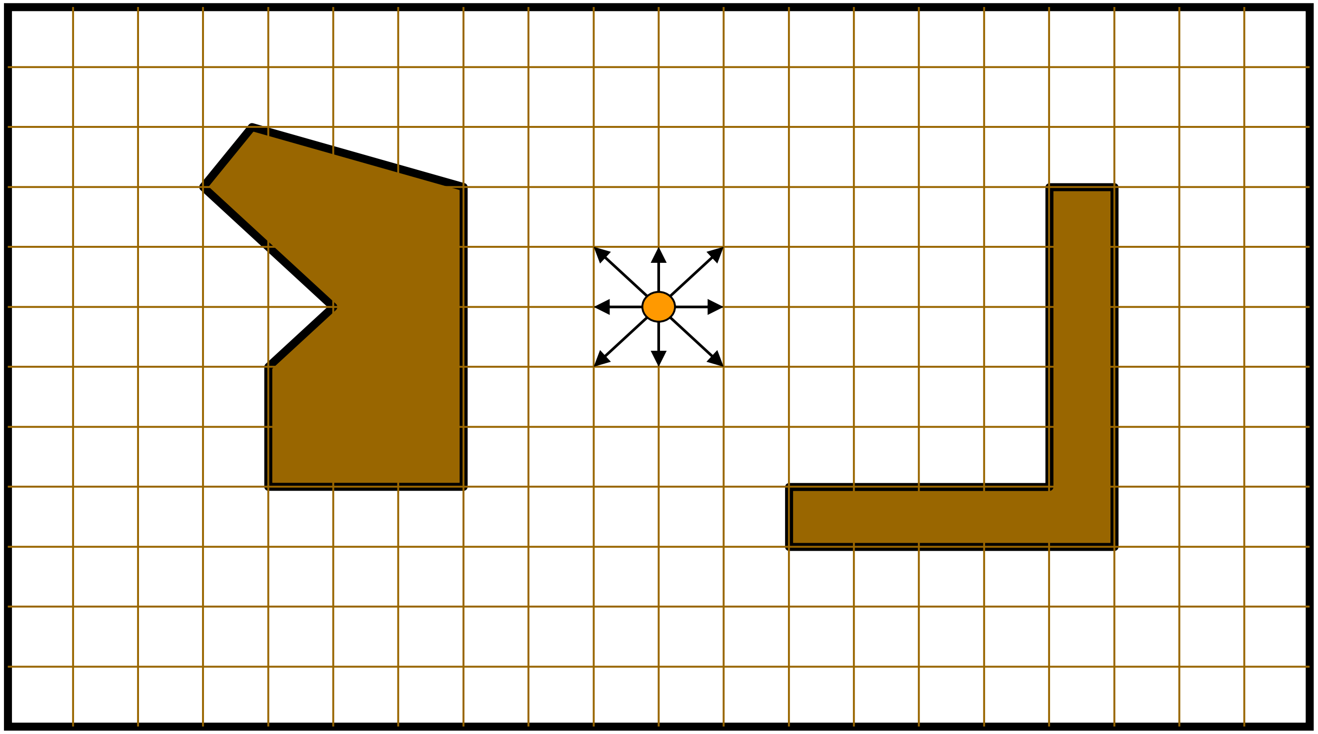
دانشگاه فردوسی مشهد
مرکز آموزش های الکترونیکی

Motion Planning and Discretization

برنامه ریزی حرکت و گسسته سازی

Presented by

Ahad Harati



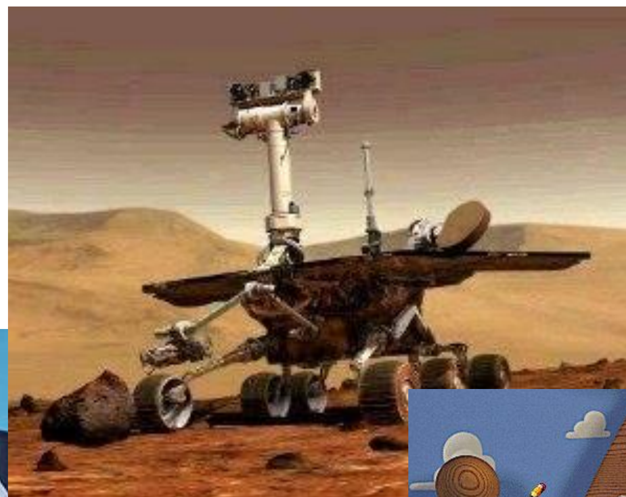
Examples and Techniques of

Discretization of State Space

Where intelligent grouping of states may make problems tractable

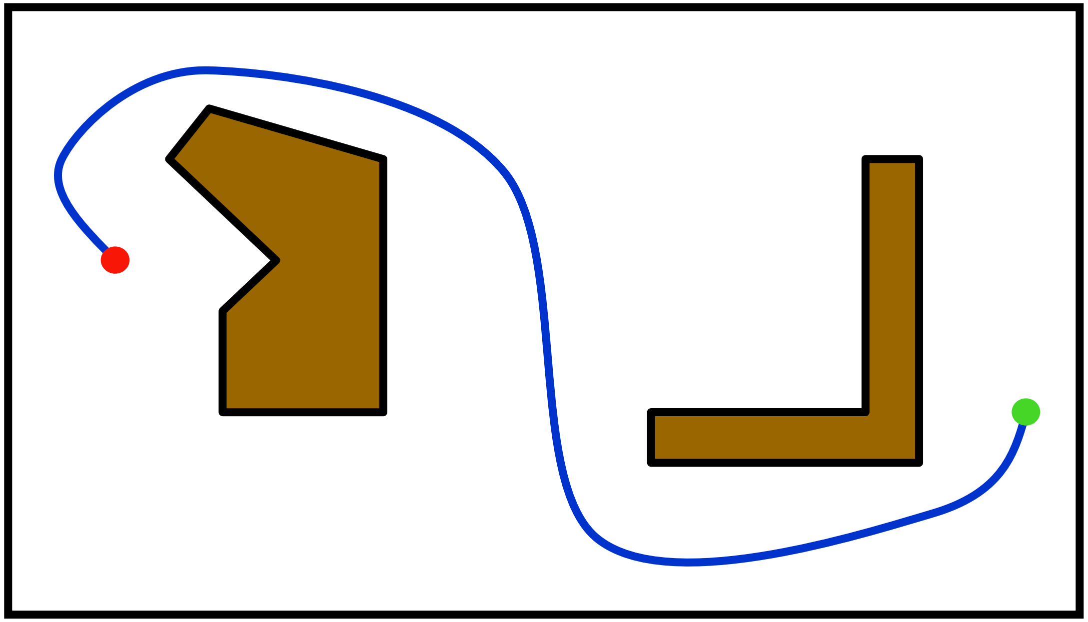
Motion planning

- is the ability of an agent to compute its own motions in order to achieve certain goals. All autonomous robots and digital actors should eventually have this ability



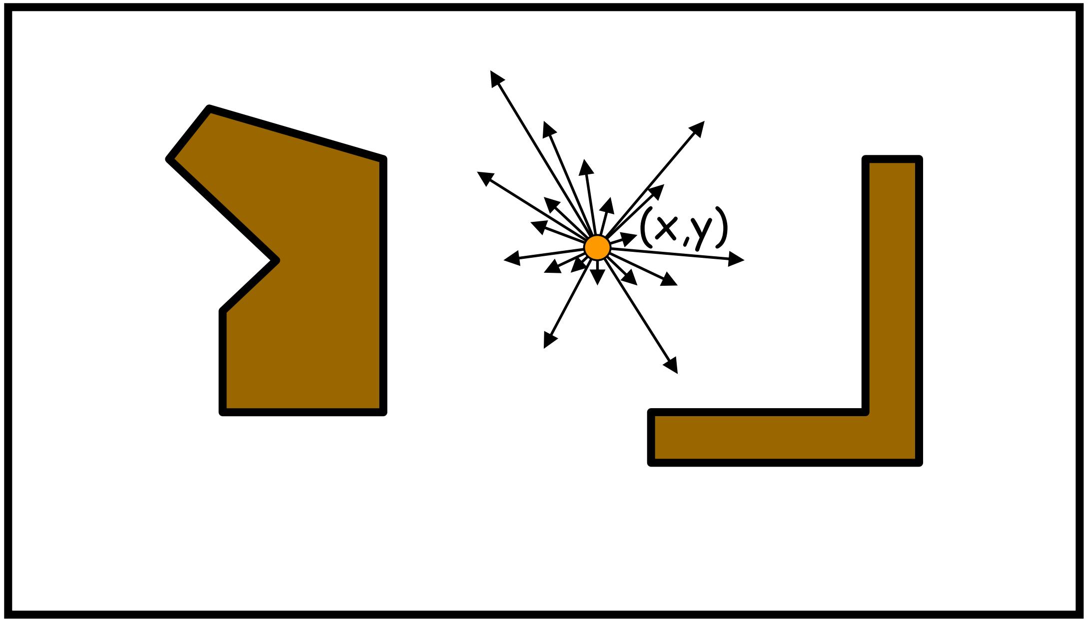
Basic problem

- Point robot in a 2-dimensional workspace with obstacles of known shape and position
- Find a collision-free path between a start and a goal position of the robot



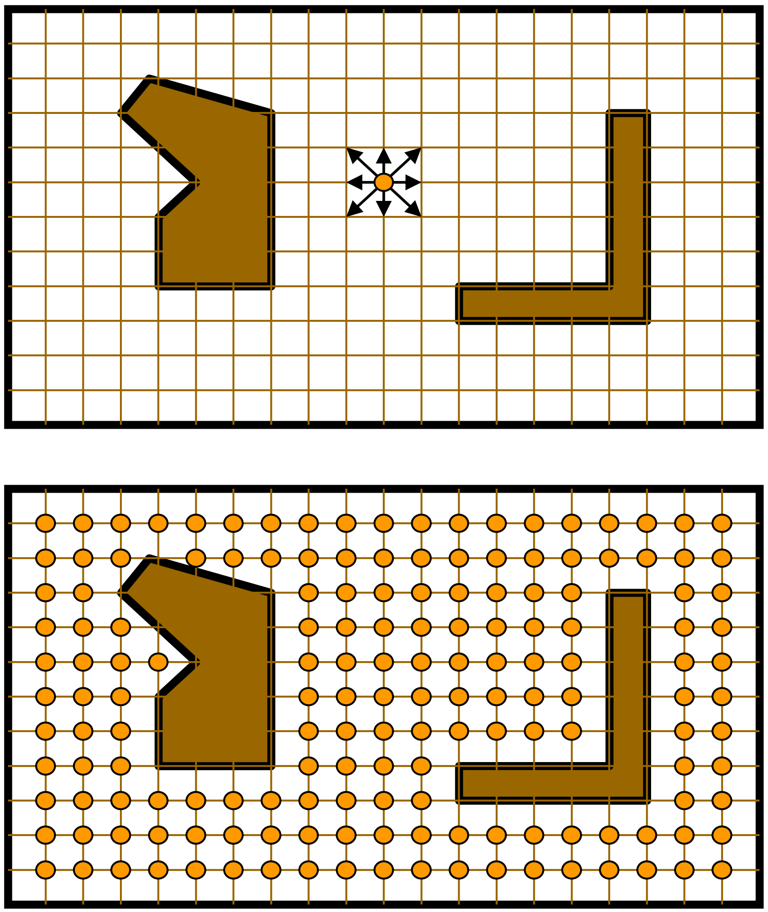
Continuity Problem

- Each robot position (x,y) can be seen as a state
  - → Continuous state space
- Each state has infinite successors (continuous actions)
- We need to discretize the state space



A Simple Discretization: Grid

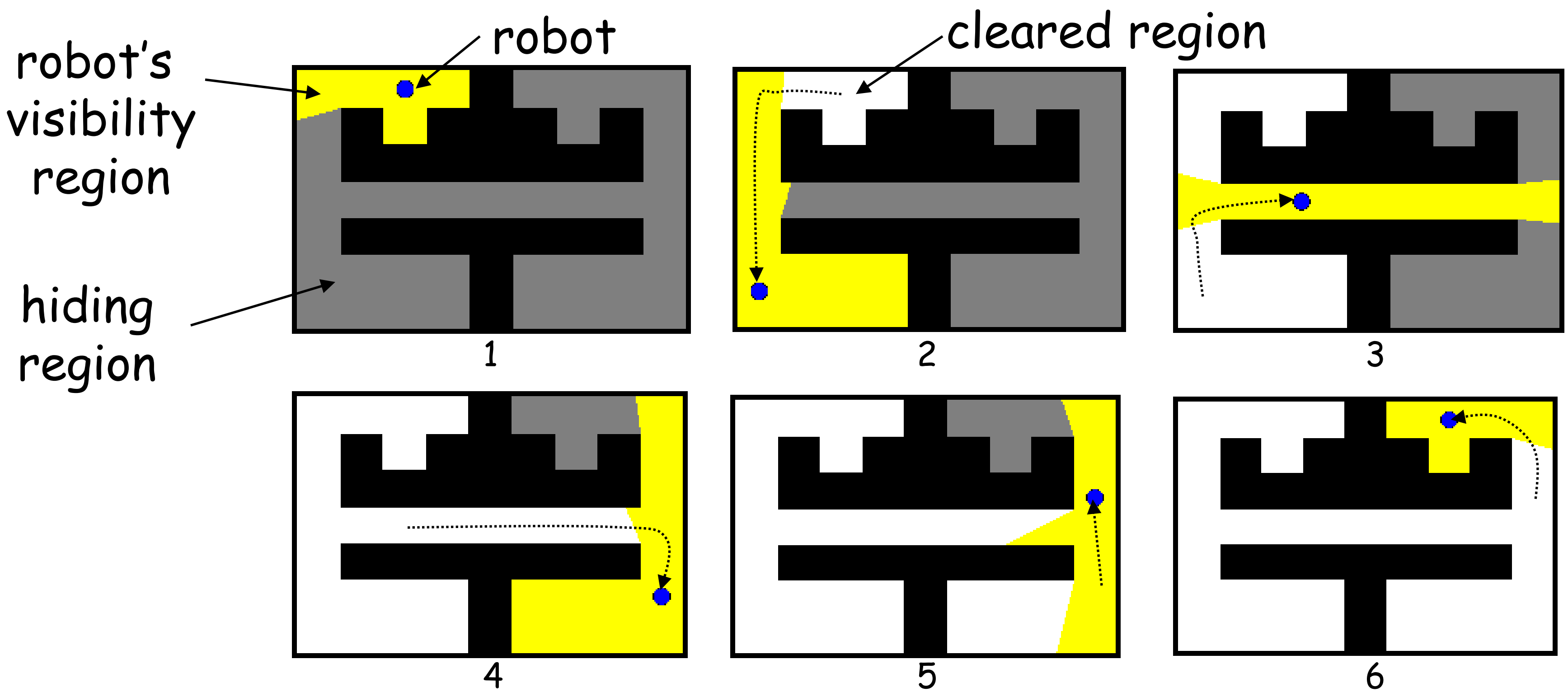
Grid-based



- Dimensionality Issue
  - Grid-based discretization leads to impractically large state spaces for dim(C-space) > 6
  - Each grid node has 3n-1 neighbors, n = dim(C-space)
- Does increasing grid size help?
  - Cannot overcome exponential growth caused by dimensionality and we loose on precision side

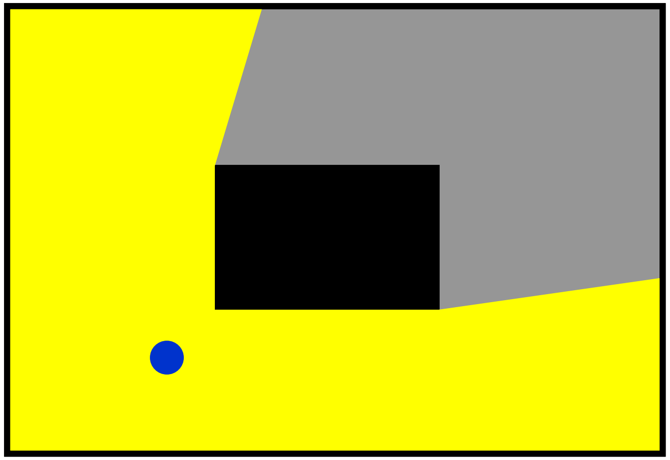
Intruder Finding Problem

- A moving intruder is hiding in a 2-D workspace
- The robot must “sweep” the workspace to find the intruder
- Both the robot and the intruder are points

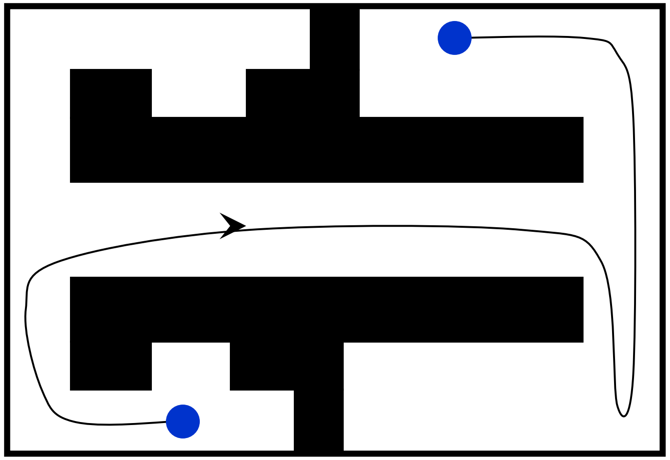


Does a solution always exist?

No !

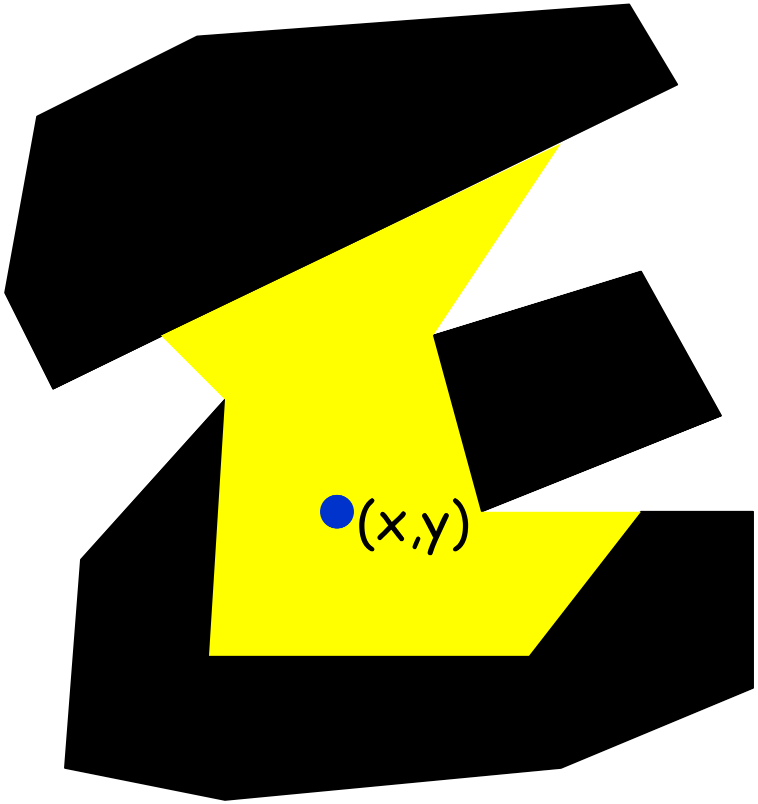


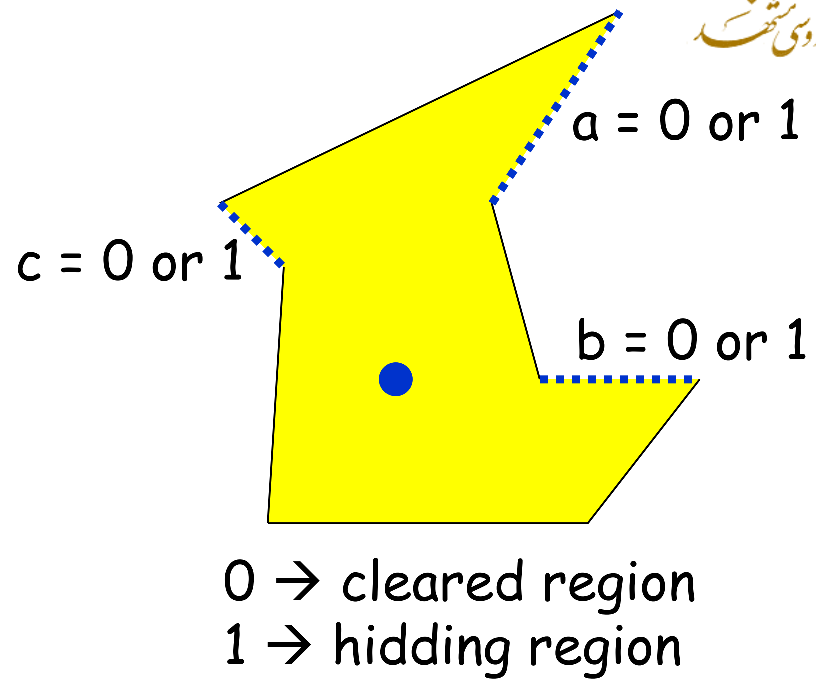
Easy to test:
"Hole" in the workspace



Hard to test:
No "hole" in the workspace

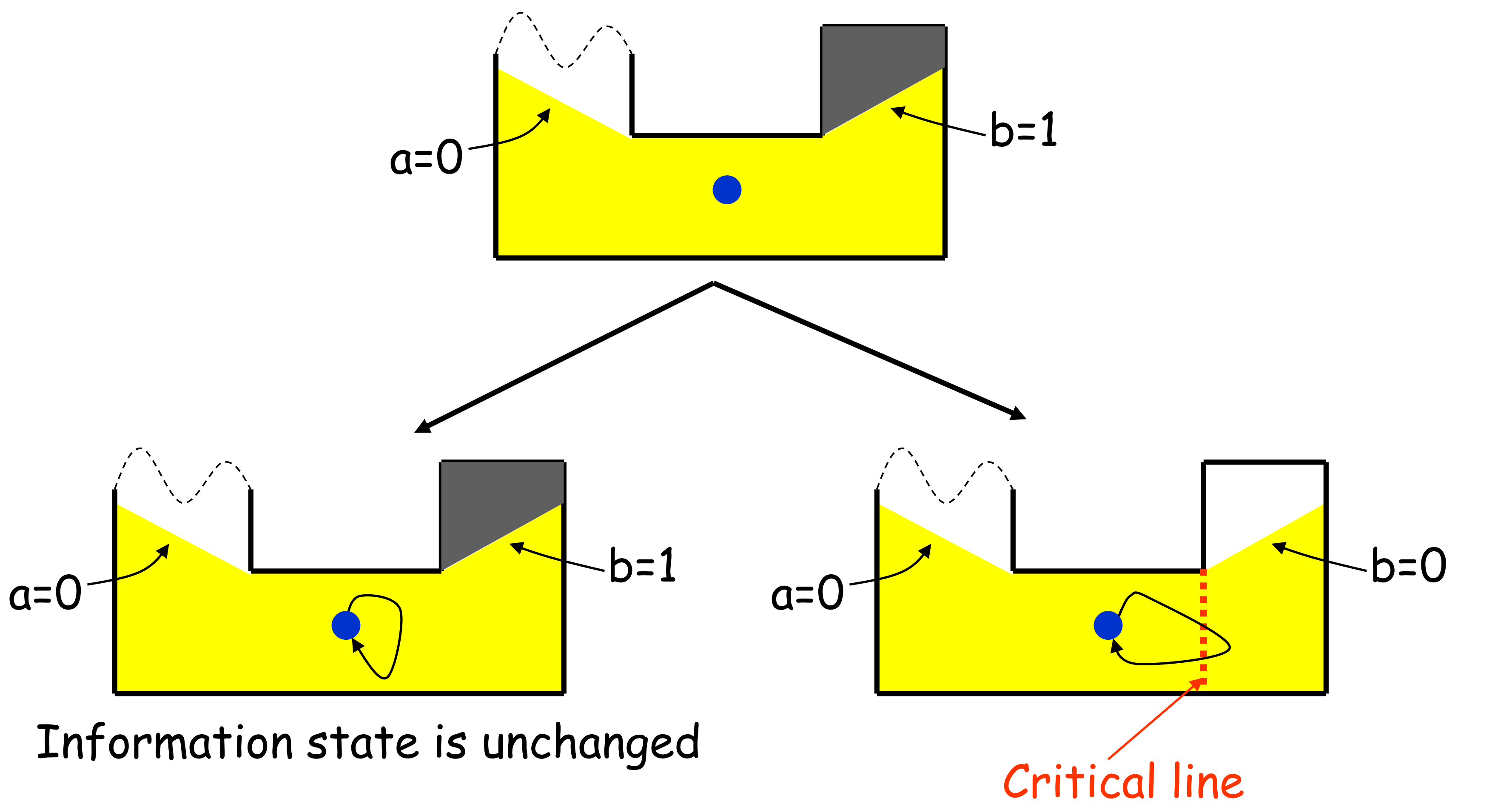
Information State



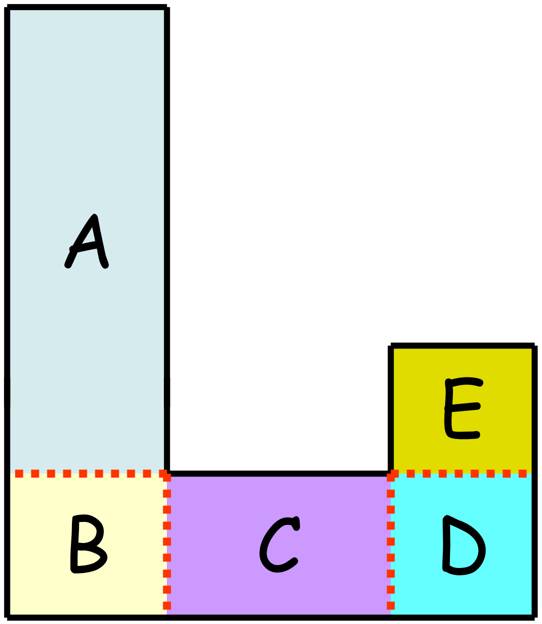


- Example of an information state: (x,y,a=1,b=1,c=0)
- An initial state is of the form (x,y,1, 1, ..., 1)
- A goal state is any state of the form (x,y,0,0, ..., 0)

Critical Line

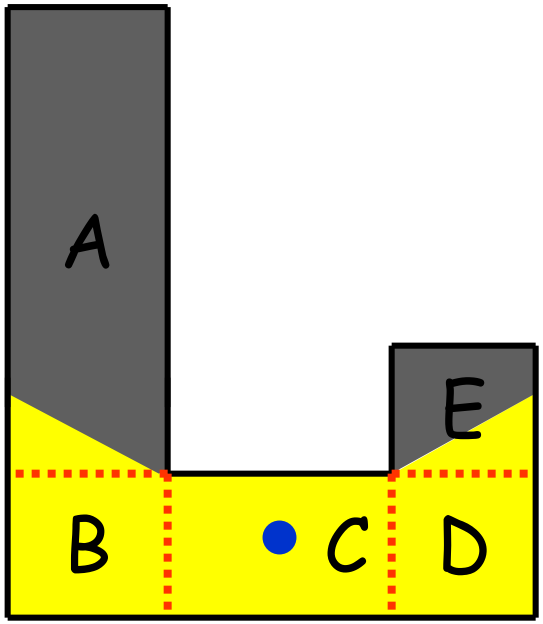


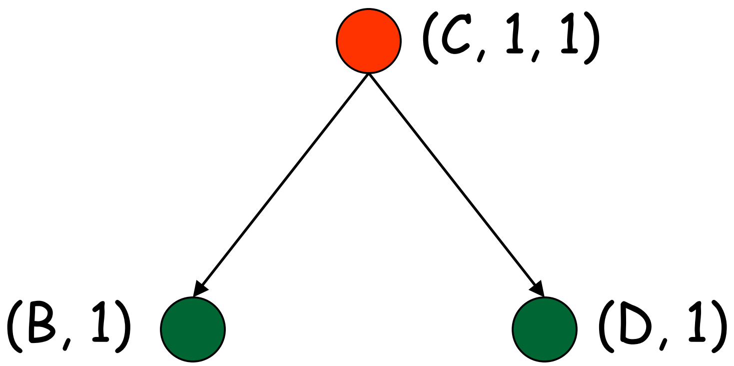
Criticality-Based Discretization



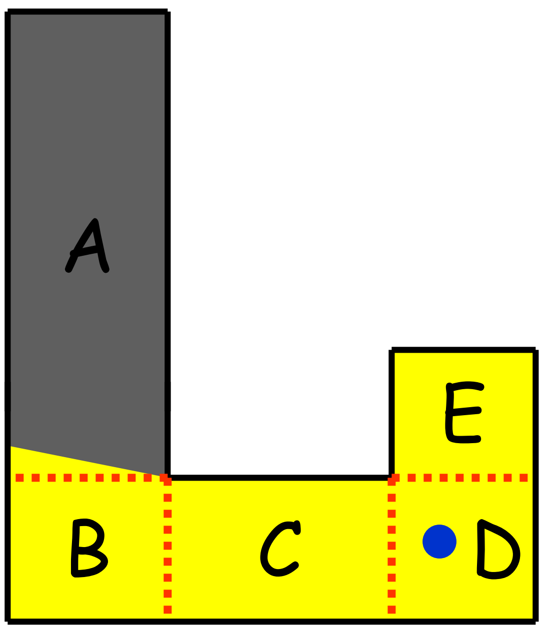
Each of the regions A, B, C, D, and E consists of “equivalent” positions of the robot,
so it’s sufficient to consider a single position per region

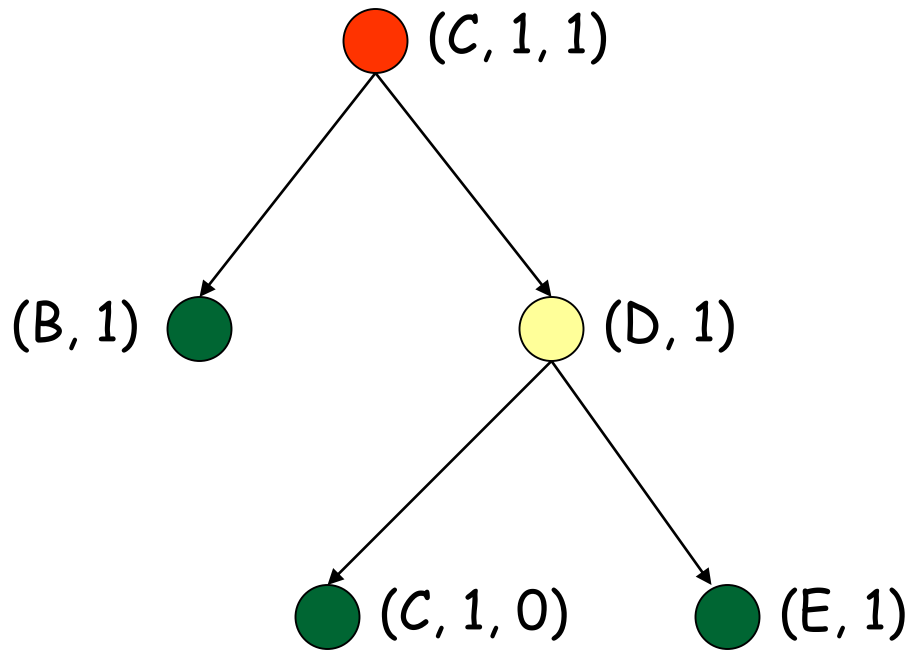
Criticality-Based Discretization



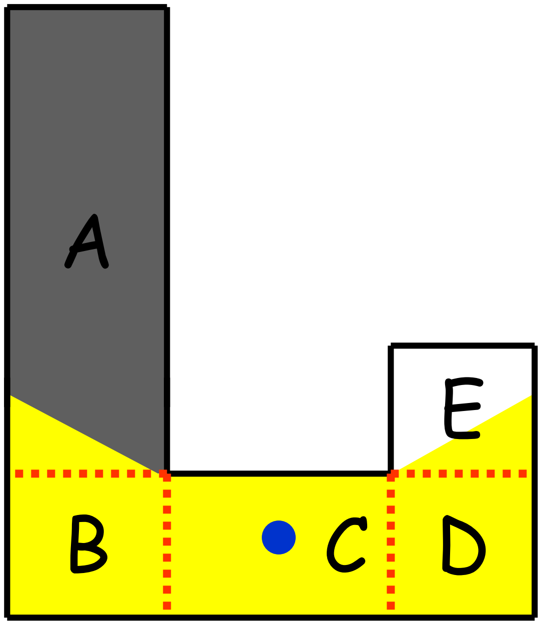


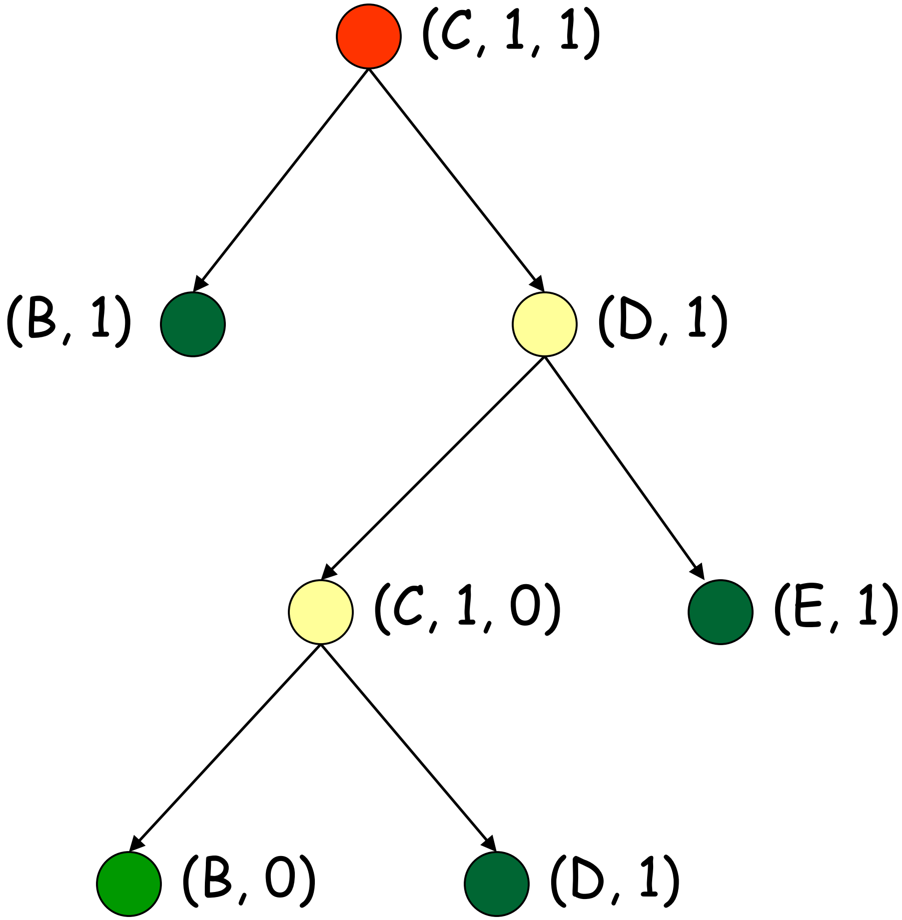
Criticality-Based Discretization



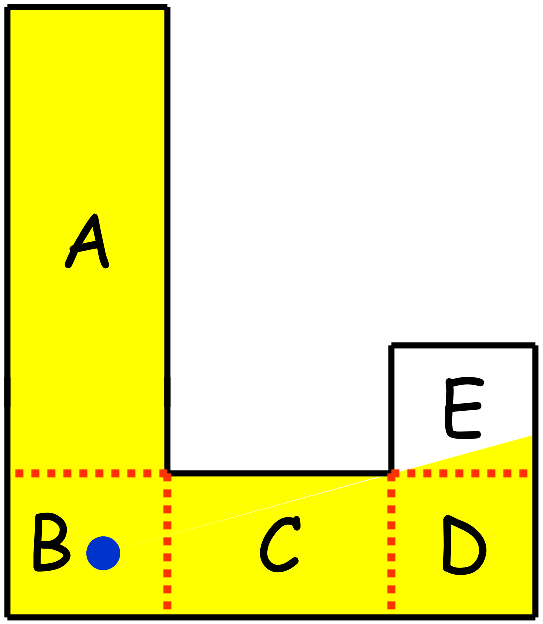


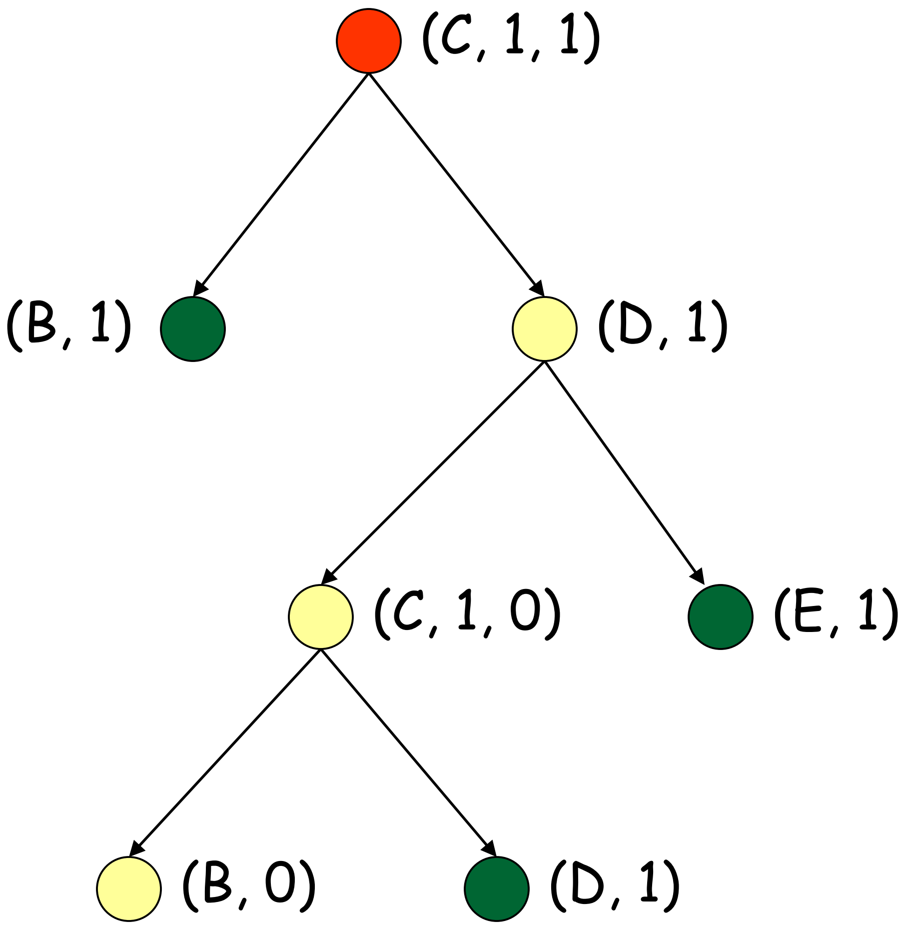
Criticality-Based Discretization





Criticality-Based Discretization

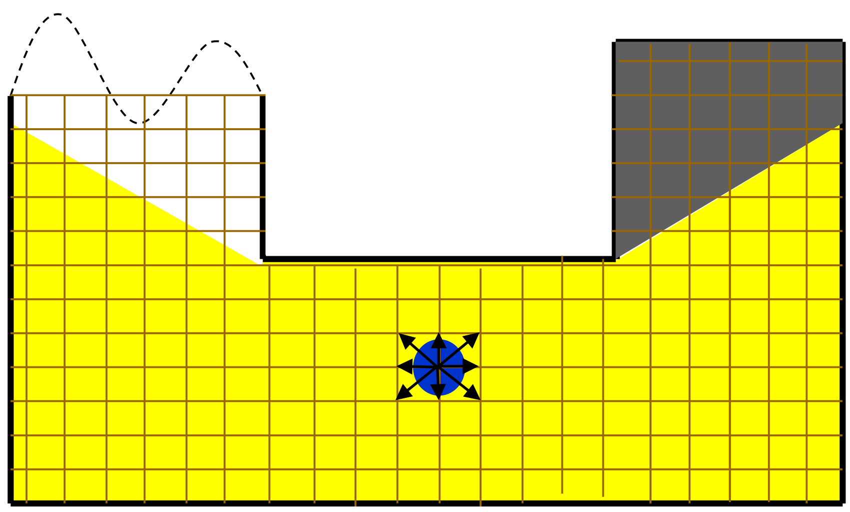




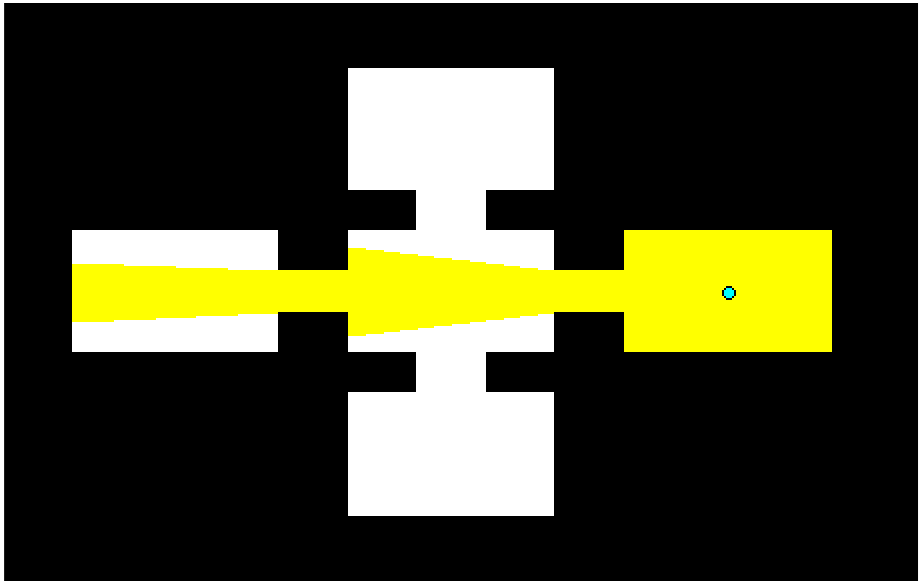
Much smaller search tree than with grid-based discretization !

Grid-Based Discretization

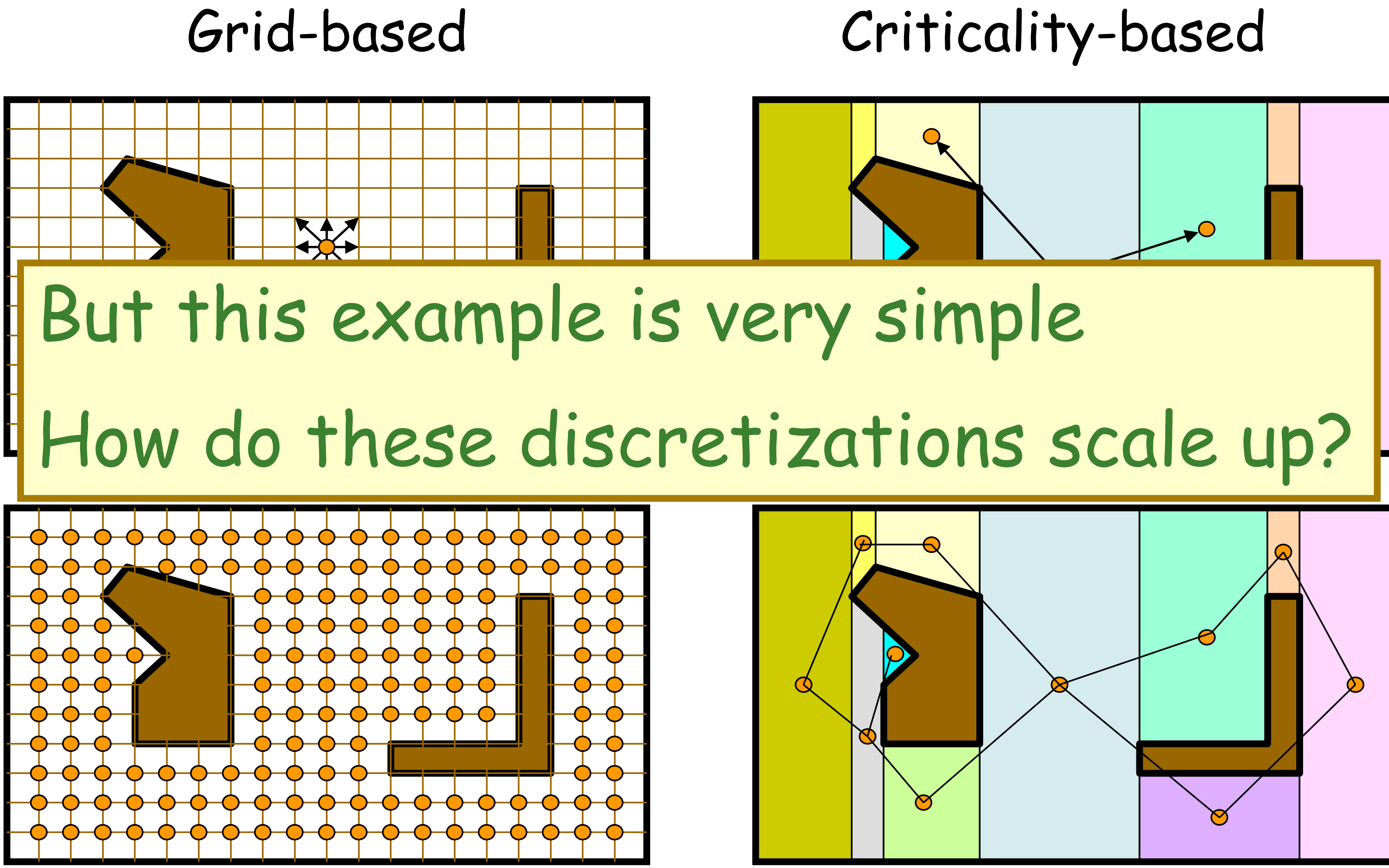
- Ignores critical lines → Visits many "equivalent" states
- Many information states per grid point
- Potentially very inefficient



Example of Solution



Two Possible Discretizations

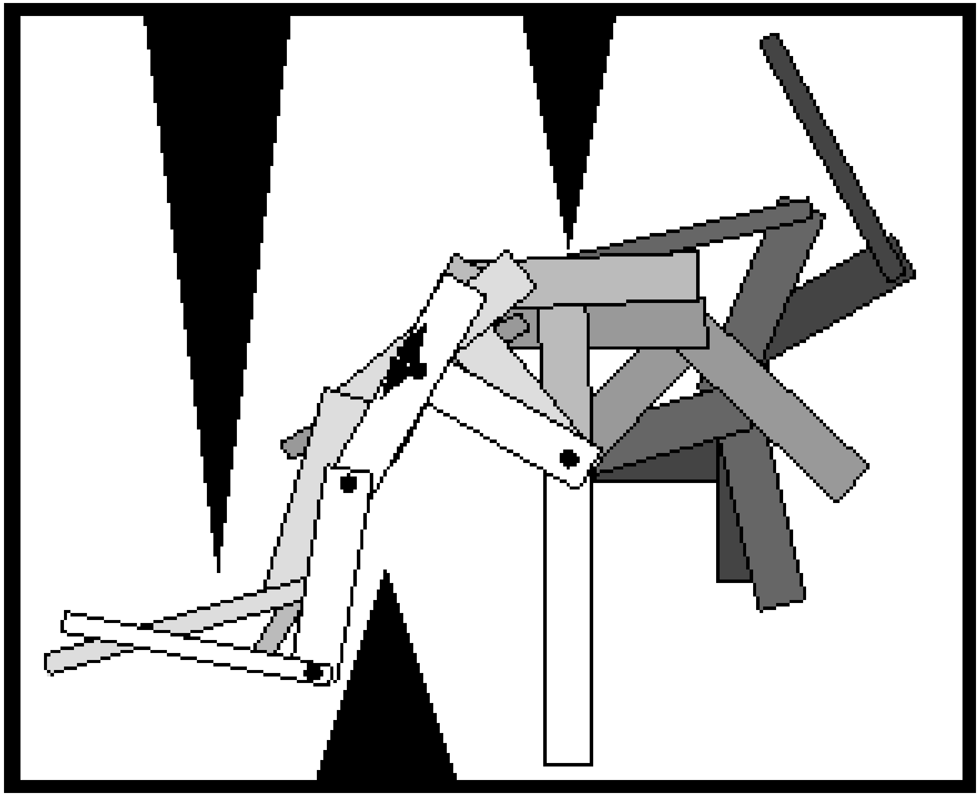


But this example is very simple

How do these discretizations scale up?

Unfortunately

- Grid-based discretization does not scale well as number of states grow exponentially regarding C-space dimensions even when the space is quite simple
- Criticality-based discretization does not scale well either in practice when the dimensionality of the continuous space increases
  - It becomes prohibitively complex to define and compute
- In next sessions we will see why and what we can do to overcome this problem



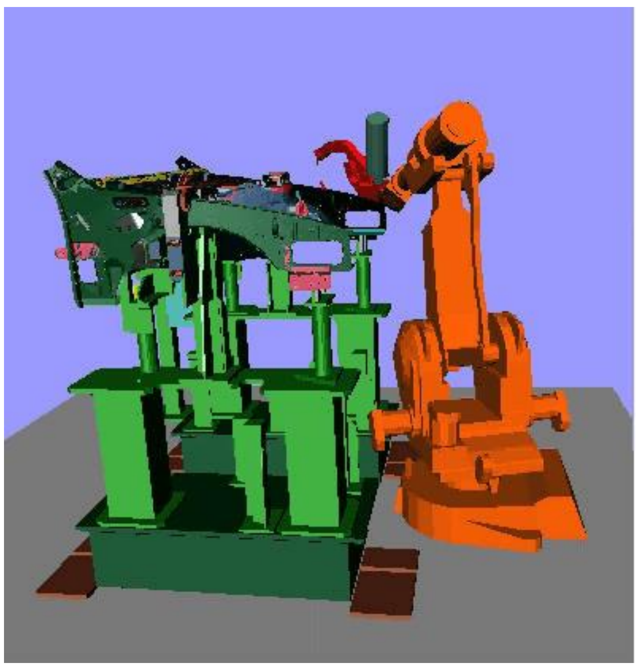
An Introduction to

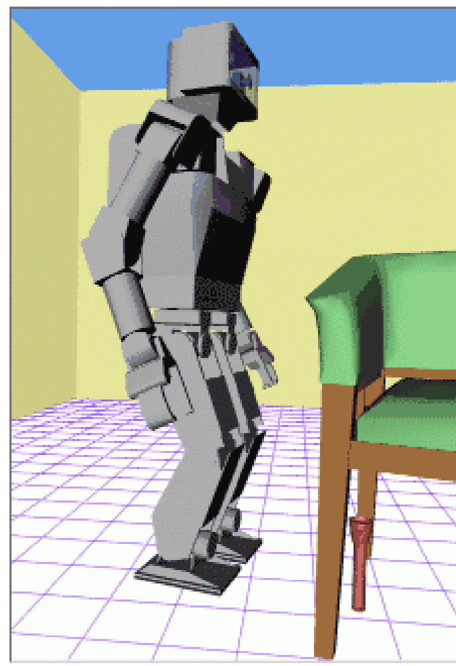
Motion Planning

Where geometrical and orientation constraints complicate planning

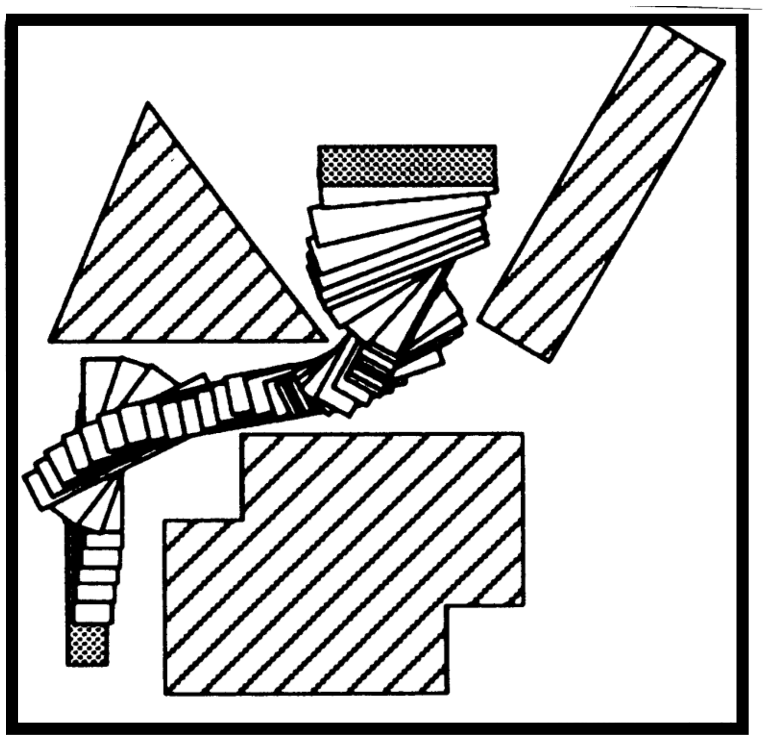
Motion Planning for an Articulated Robot

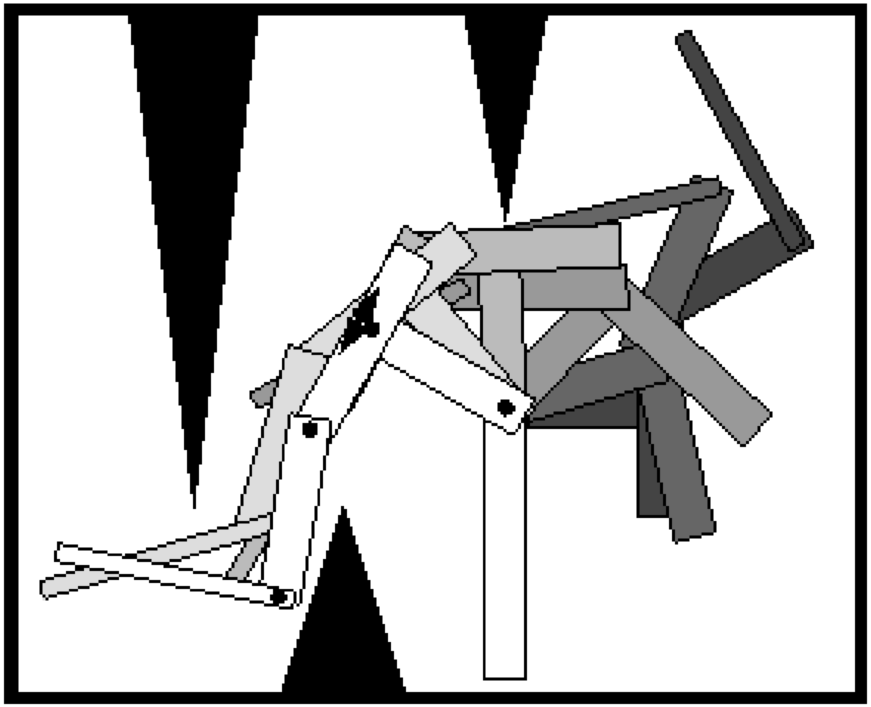
- Find a path to a goal configuration that satisfies various constraints: collision avoidance, equilibrium, etc...





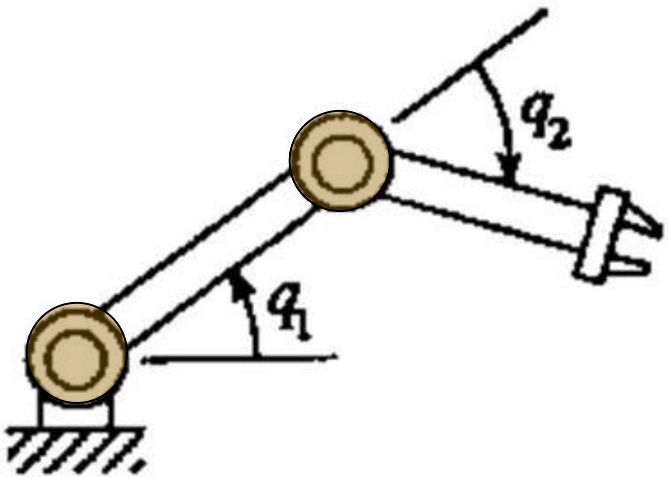
Robots have different shapes and kinematics! What is a path?

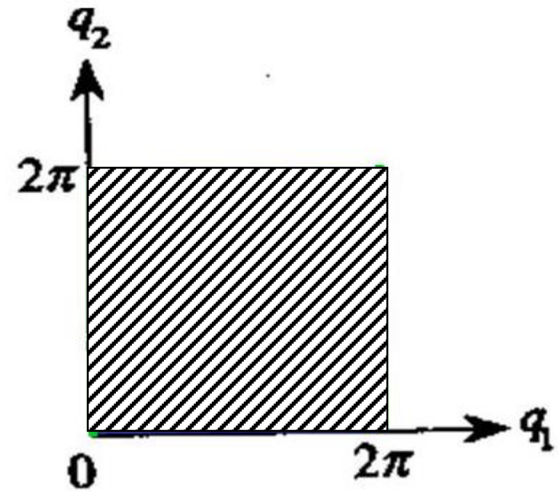




Configuration of an Articulated Robot

- Configuration of a robot is a list of non-redundant parameters that fully specify the position and orientation of each of its bodies
  - In this robot, one possible choice is: (q1, q2)
  - Here, there are two non-redundant parameters

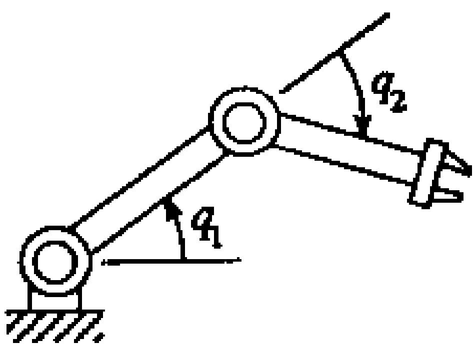


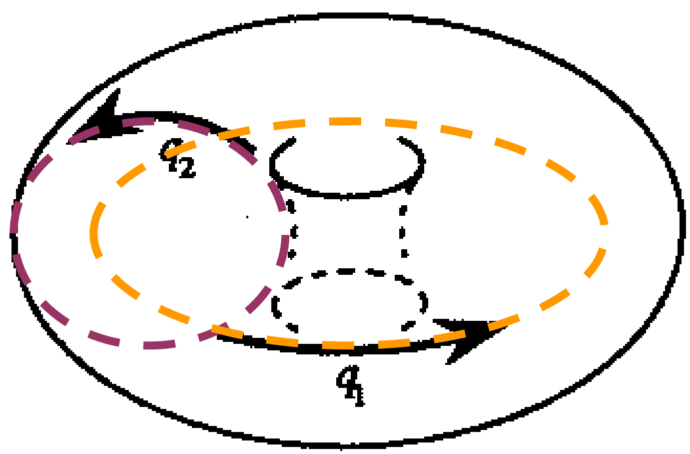


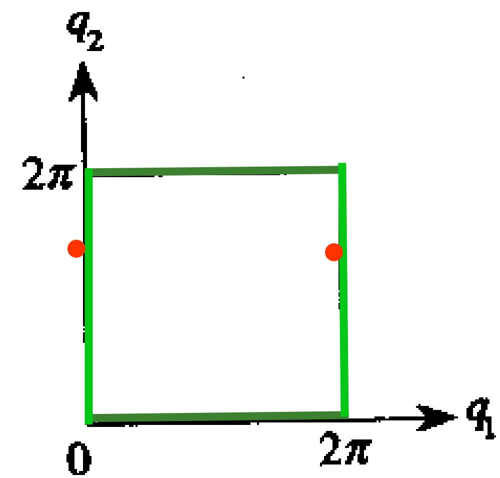
Configuration Space

- Space of all its possible configurations is called Configuration Space (C-Space)
  - For the depicted robot, it has 2 dimensions
- But the topology of this space is usually not that of a Cartesian space

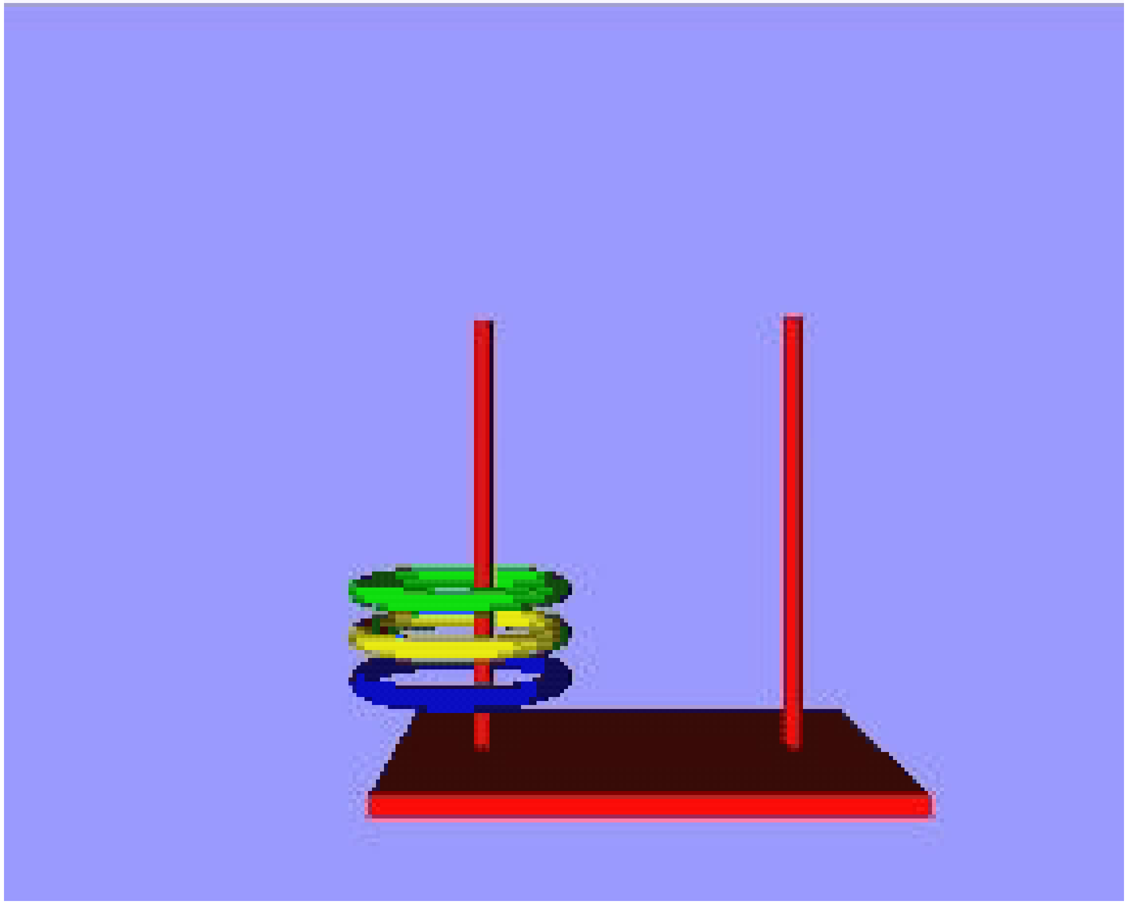
$C = S^1 \times S^1$





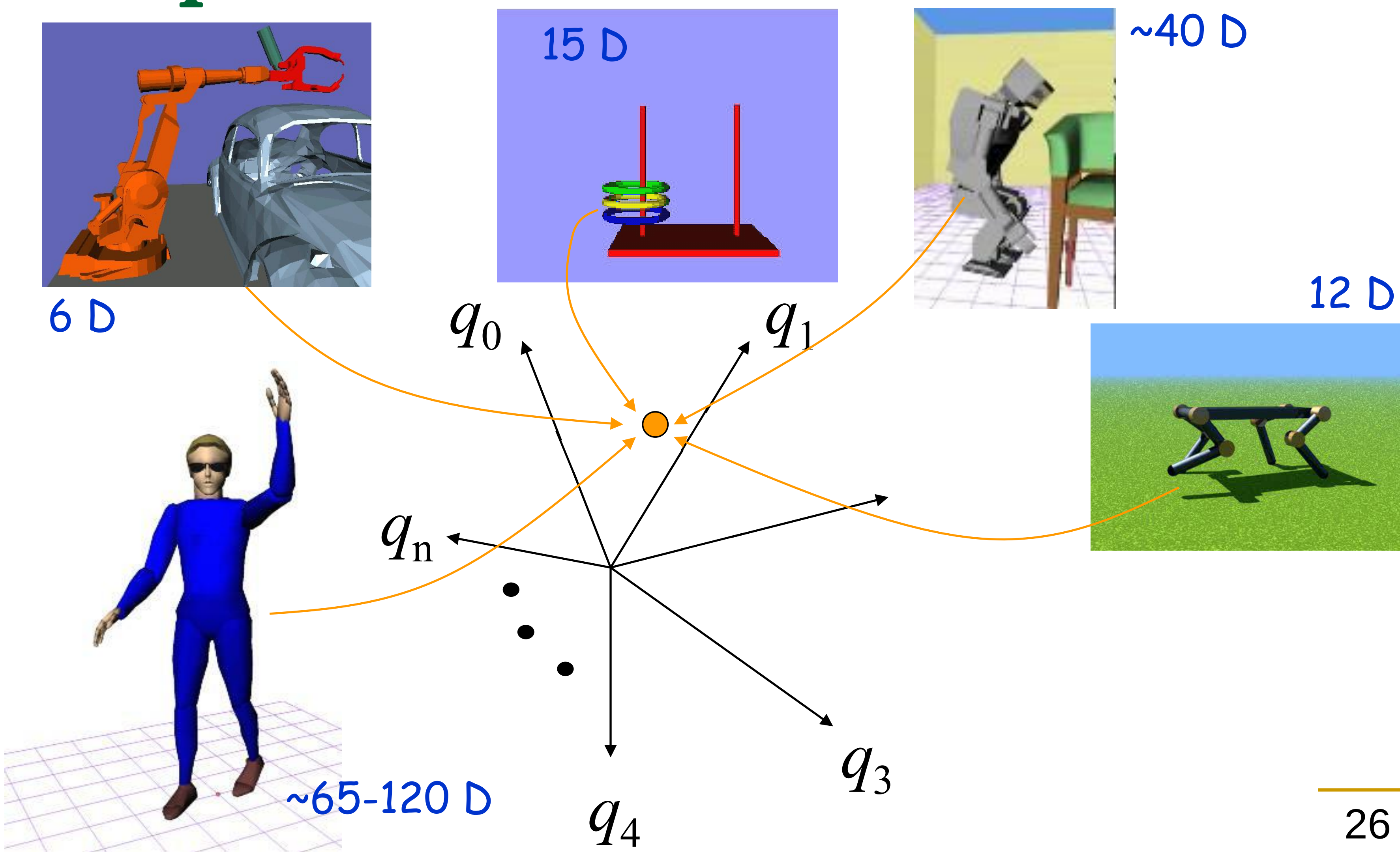


How many dimensions has the C-space of these 3 rings?

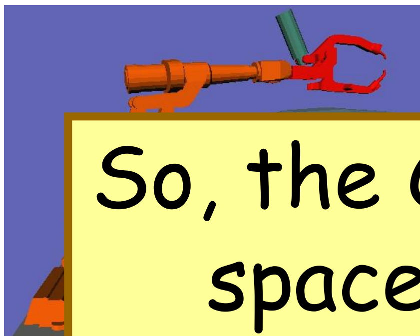


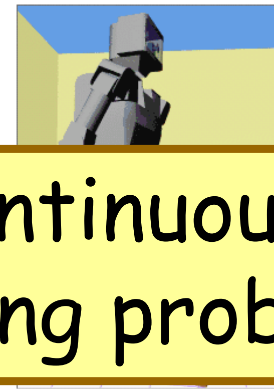
Answer:
$3 \times 5 = 15$

Every robot maps to a point in its C-space …

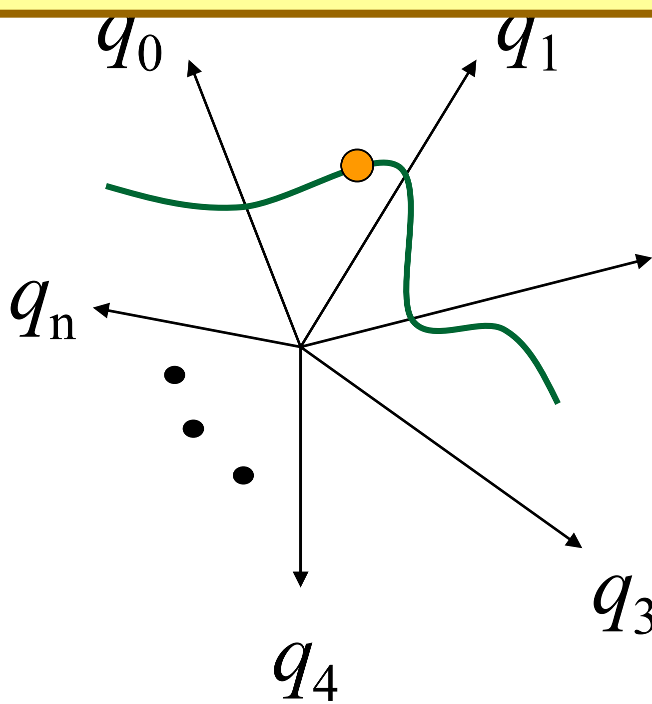


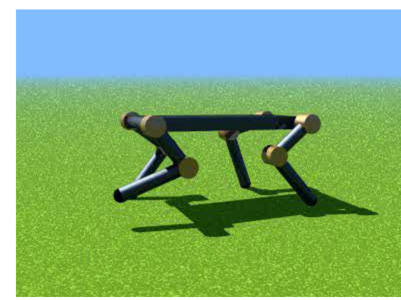
... and every robot path is a curve in C-space



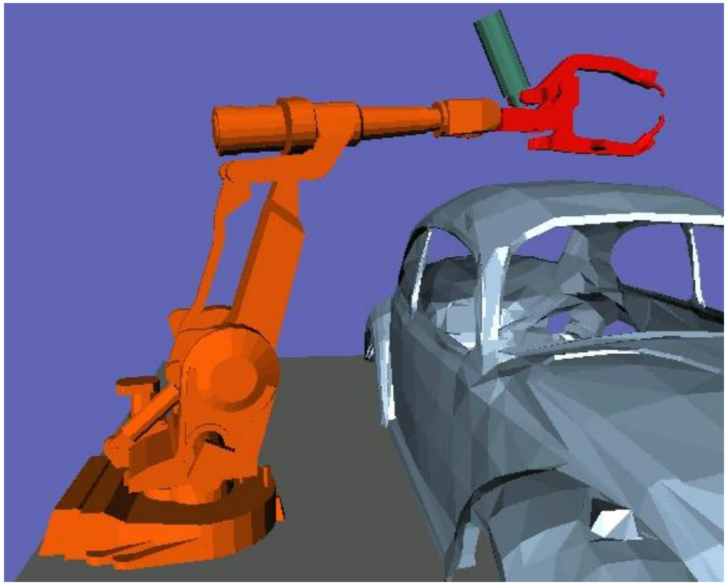


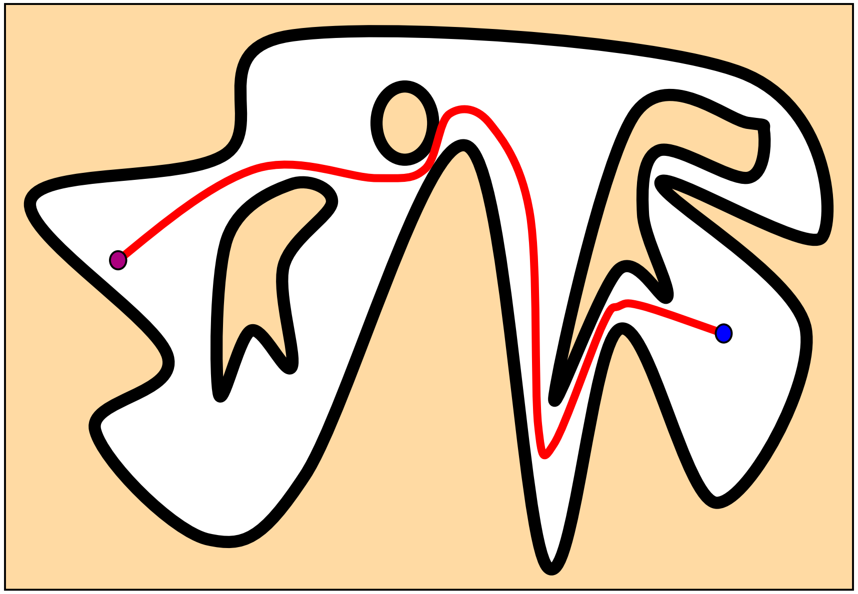
So, the C-space is the continuous state space of motion planning problems





C-space reduces motion planning to finding a path for a point



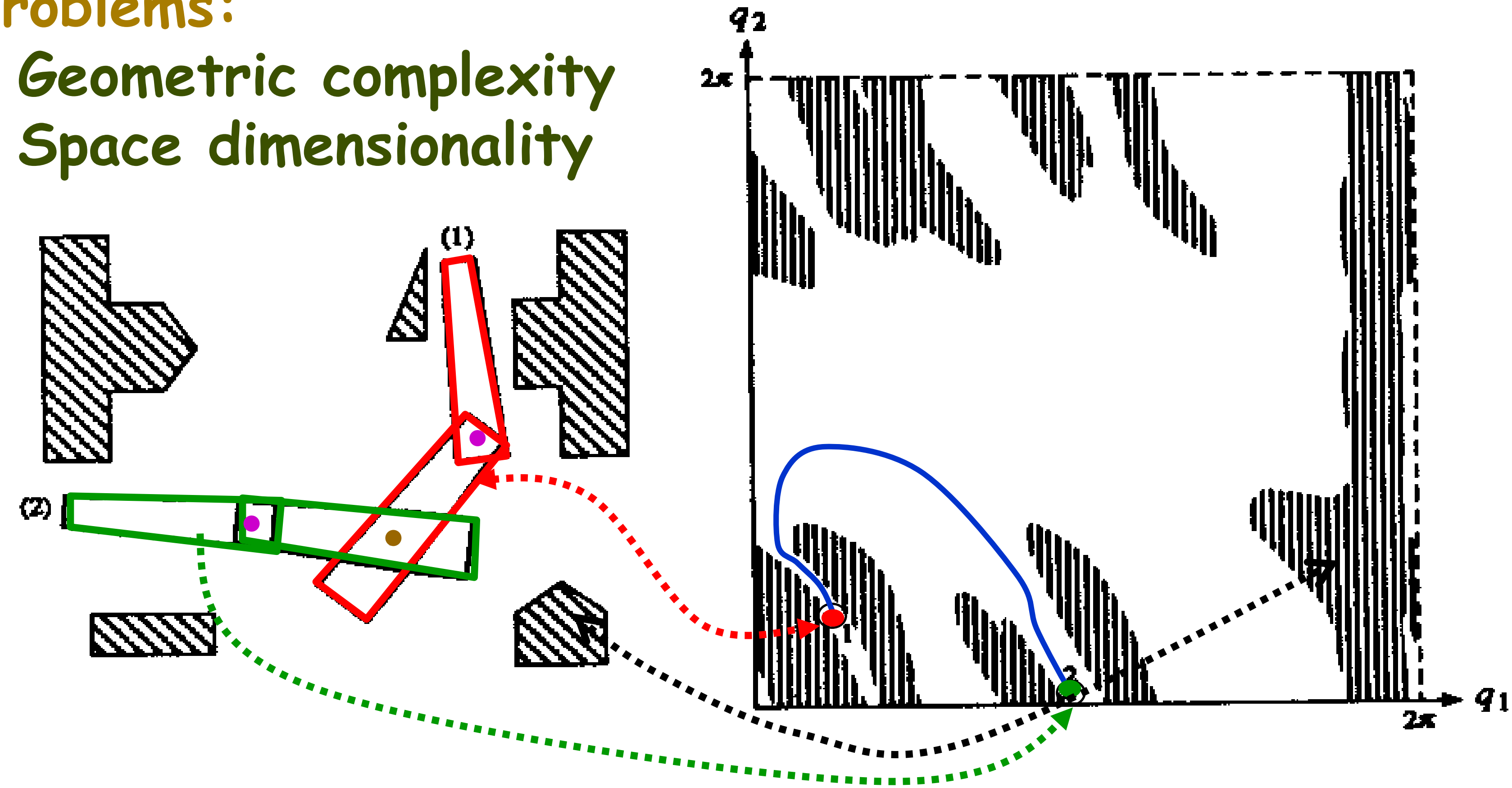


But how do the obstacle constraints map into *C*-space ?

A Simple Example: Two-Joint Planar Robot Arm

Problems:

- *Geometric complexity*
- *Space dimensionality*



Continuous state space

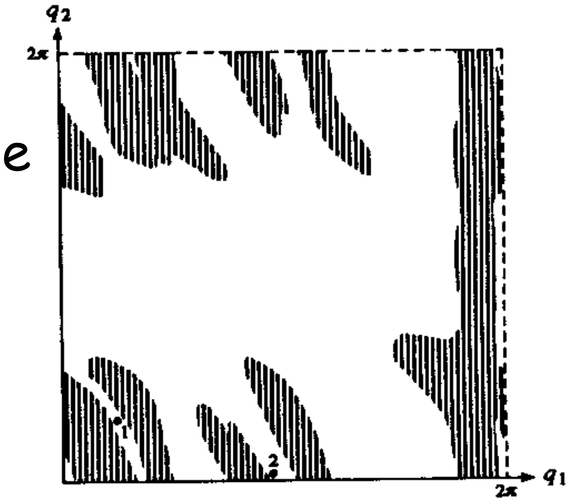
↓

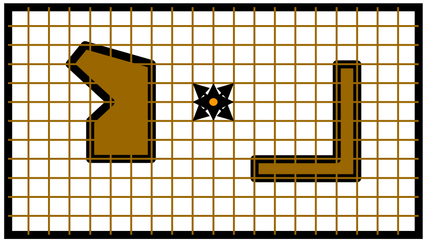
Discretization

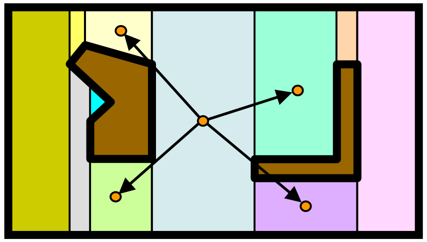
↓

Search

C-space







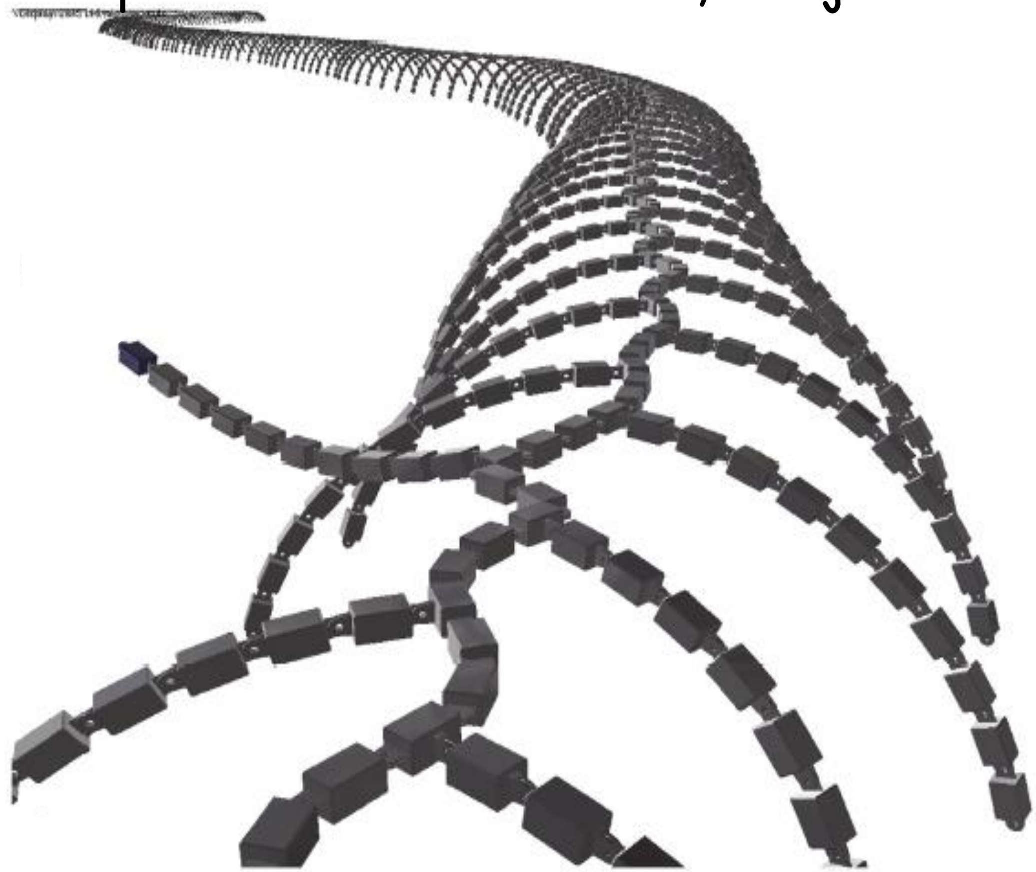
About Discretization

- Dimensionality Issue
  - Grid-based discretization leads to impractically large state spaces for dim(C-space) > 6
  - Each grid node has 3n-1 neighbors, n = dim(C-space)
- Dimensionality + geometric complexity
  - → Criticality-based discretization turns out to be prohibitively complex

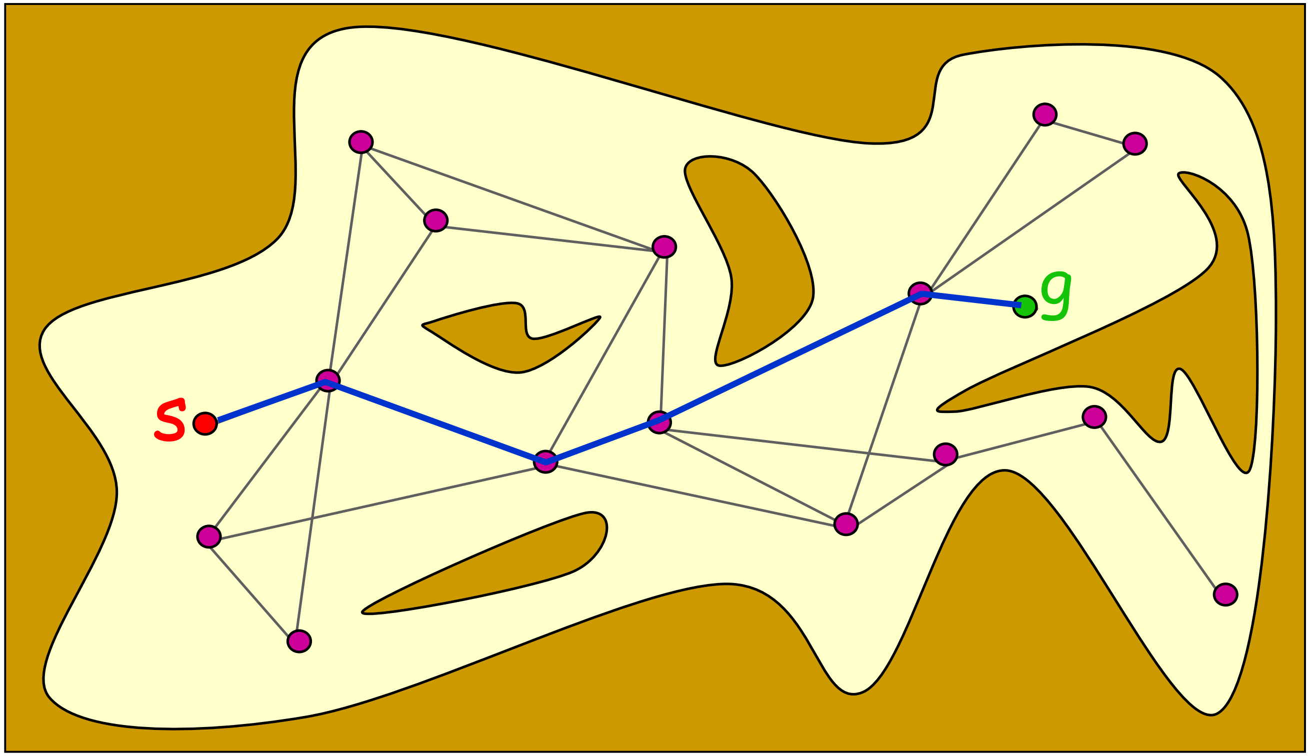
Robots with many joints: Modular Self-Reconfigurable Robots

Millipede-like robot with 13,000 joints

(M. Yim)



(S. Redon)

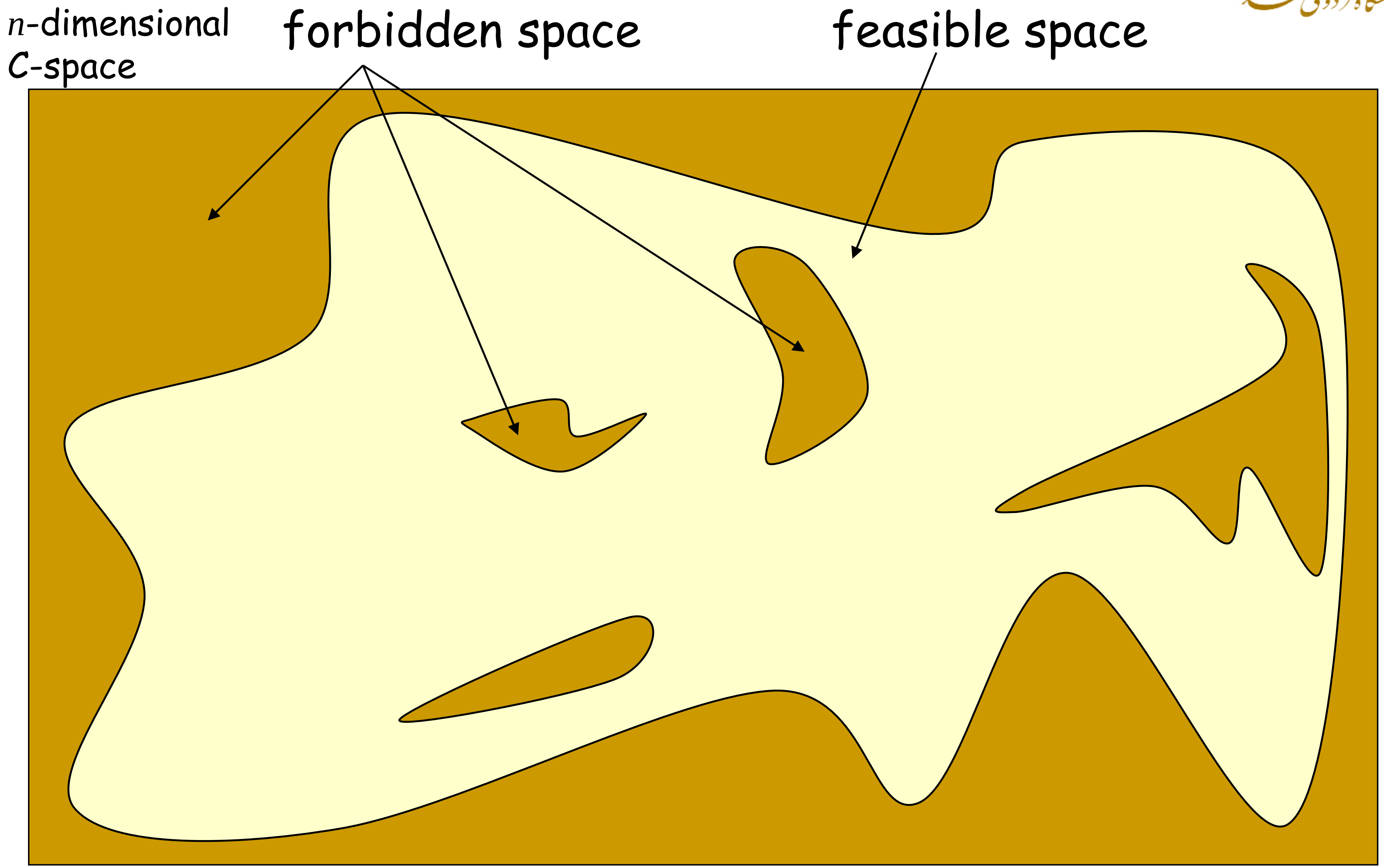


A Sampling based Planner:

Probabilistic Road Map

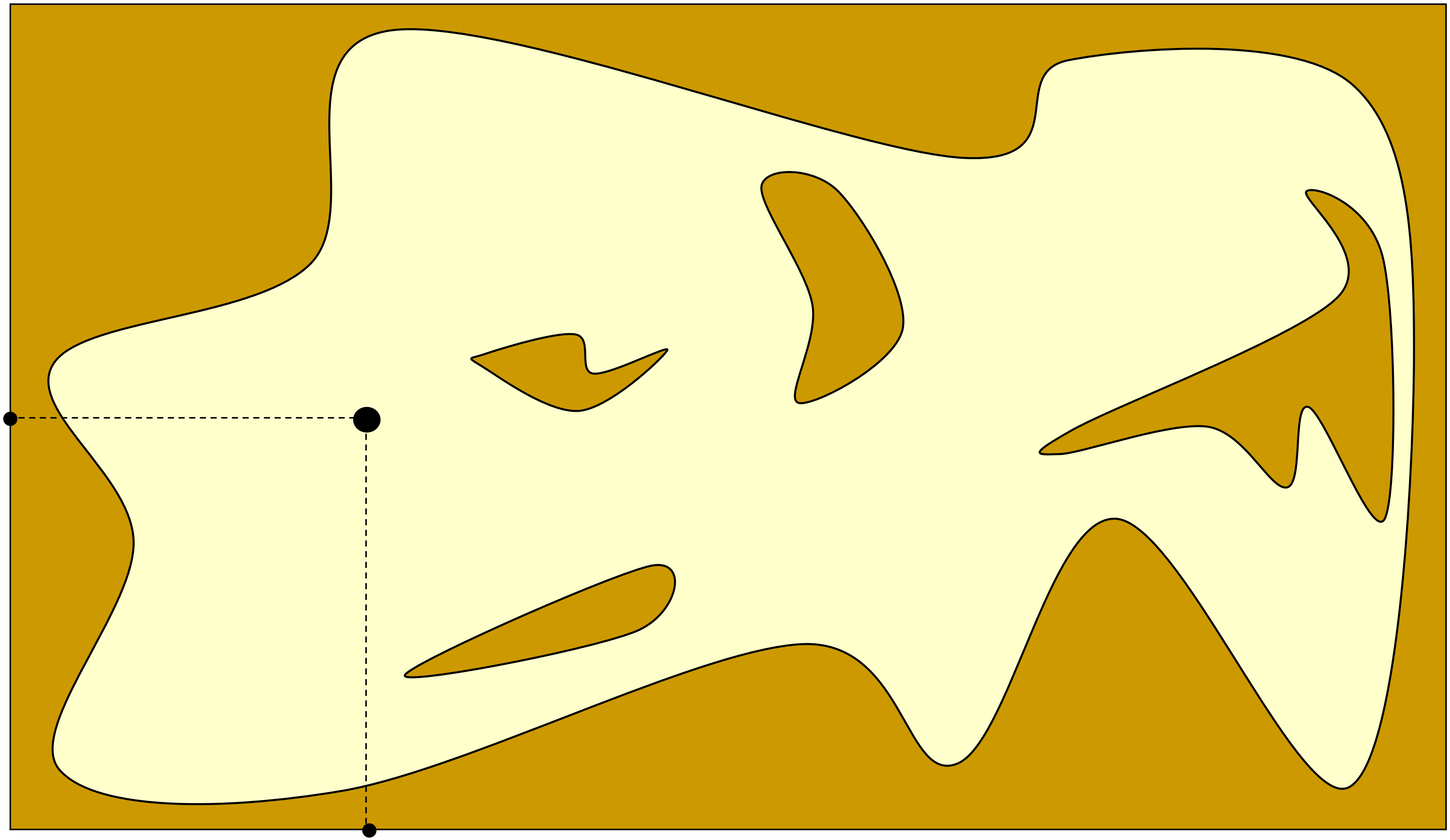
When sampling gives a better way to use our computational resources

Probabilistic Roadmap (PRM)



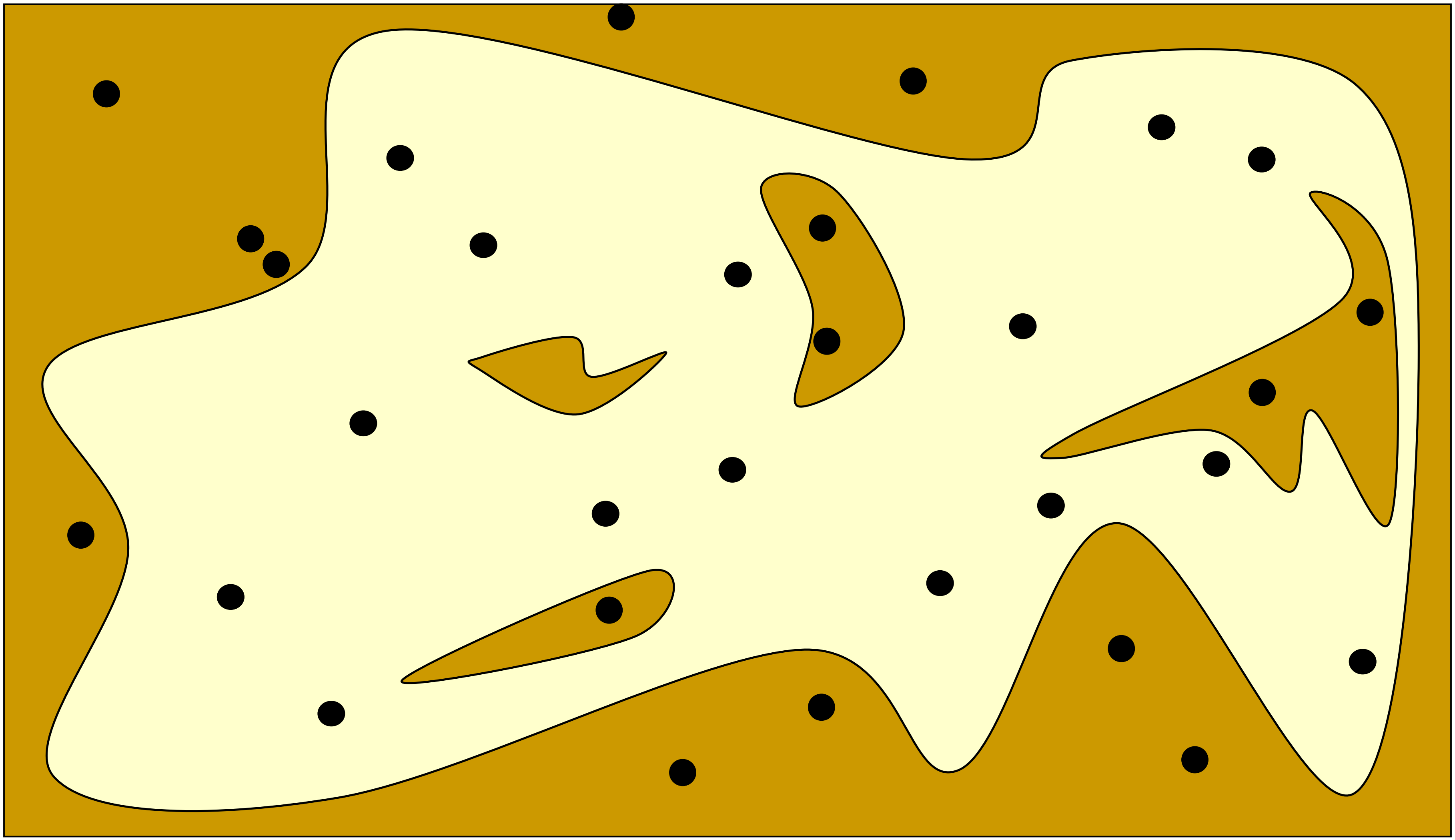
Probabilistic Roadmap (PRM)

Configurations are sampled by picking coordinates at random



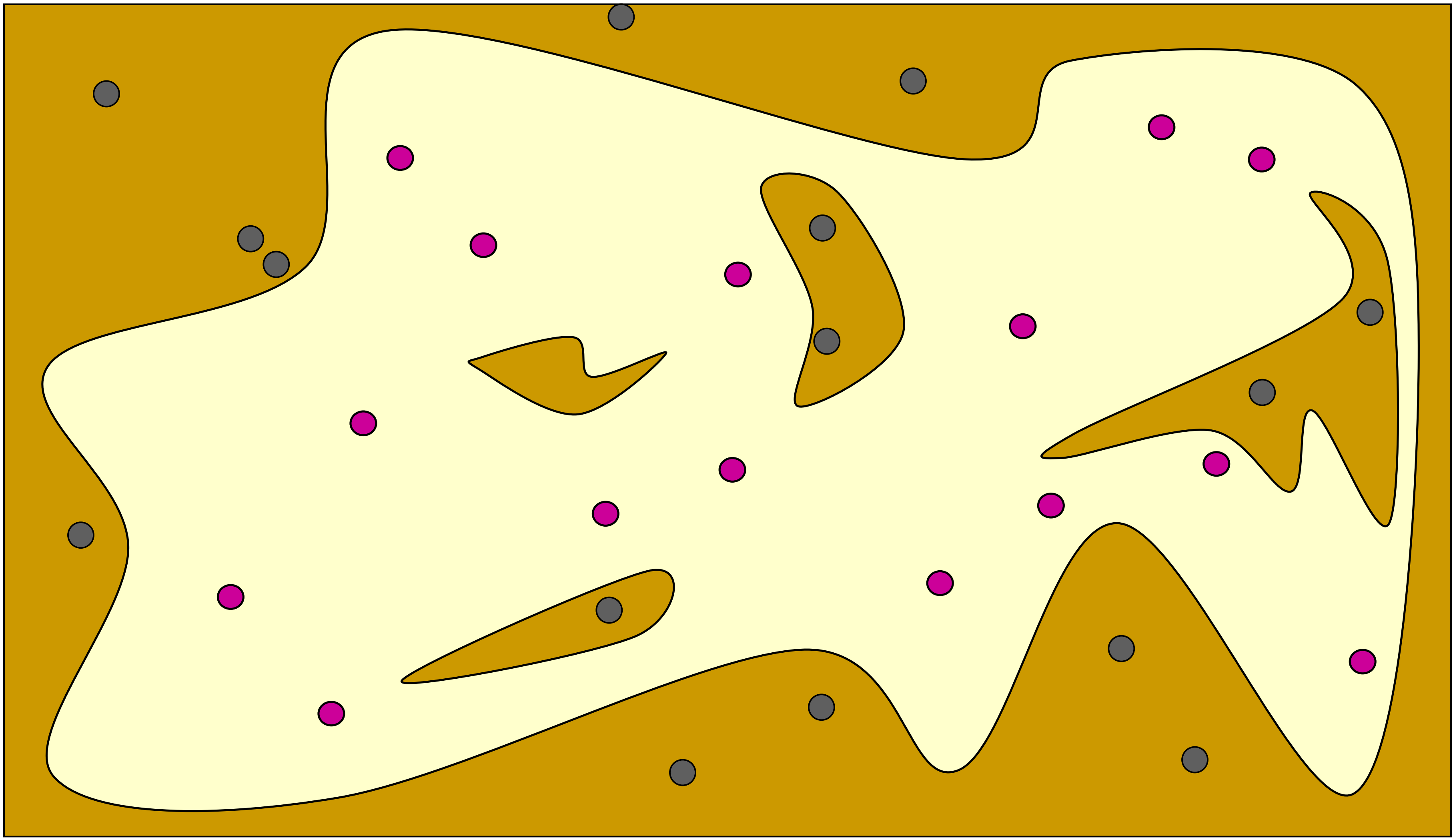
Probabilistic Roadmap (PRM)

Configurations are sampled by picking coordinates at random



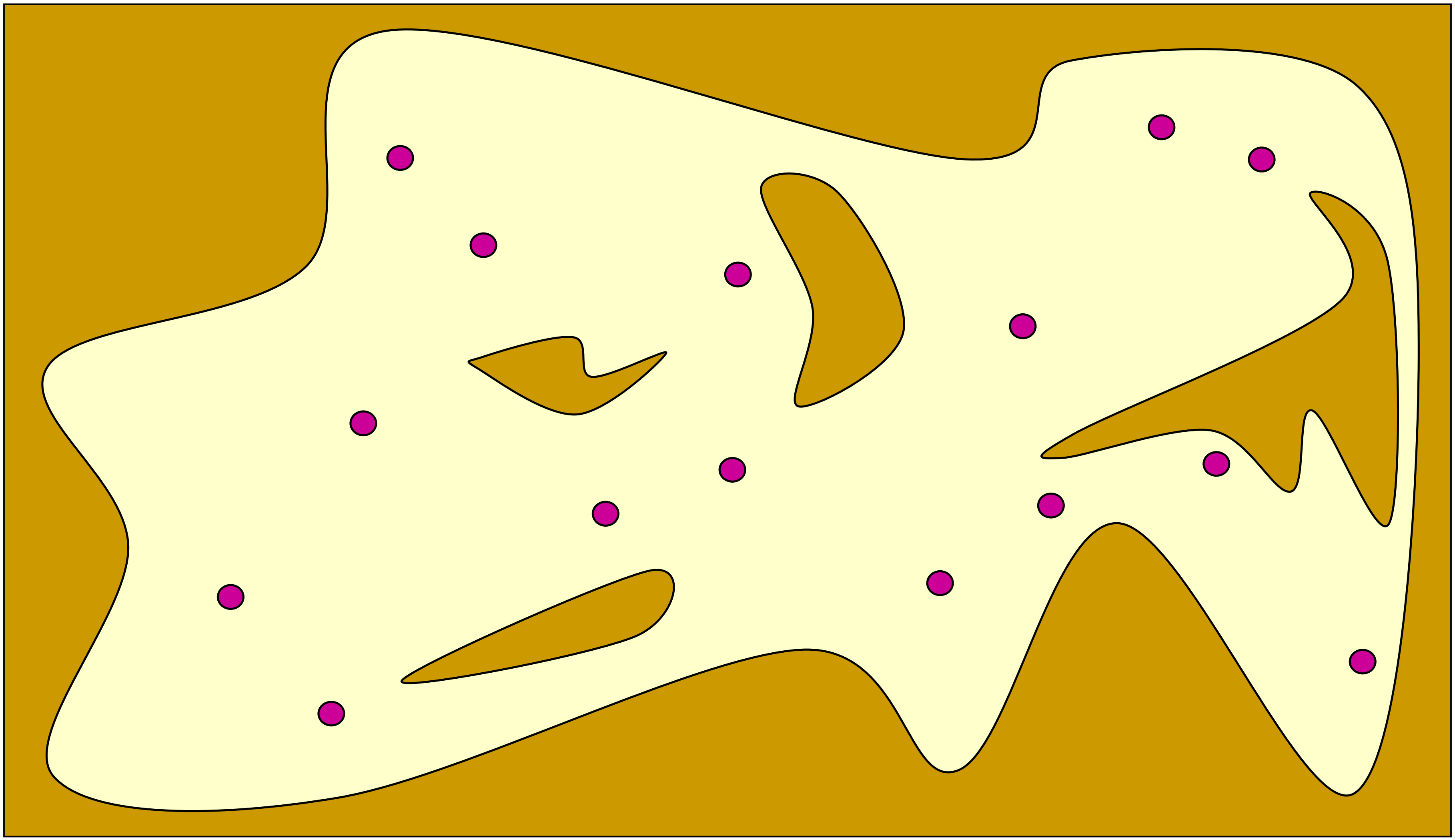
Probabilistic Roadmap (PRM)

Sampled configurations are tested for collision (feasibility)



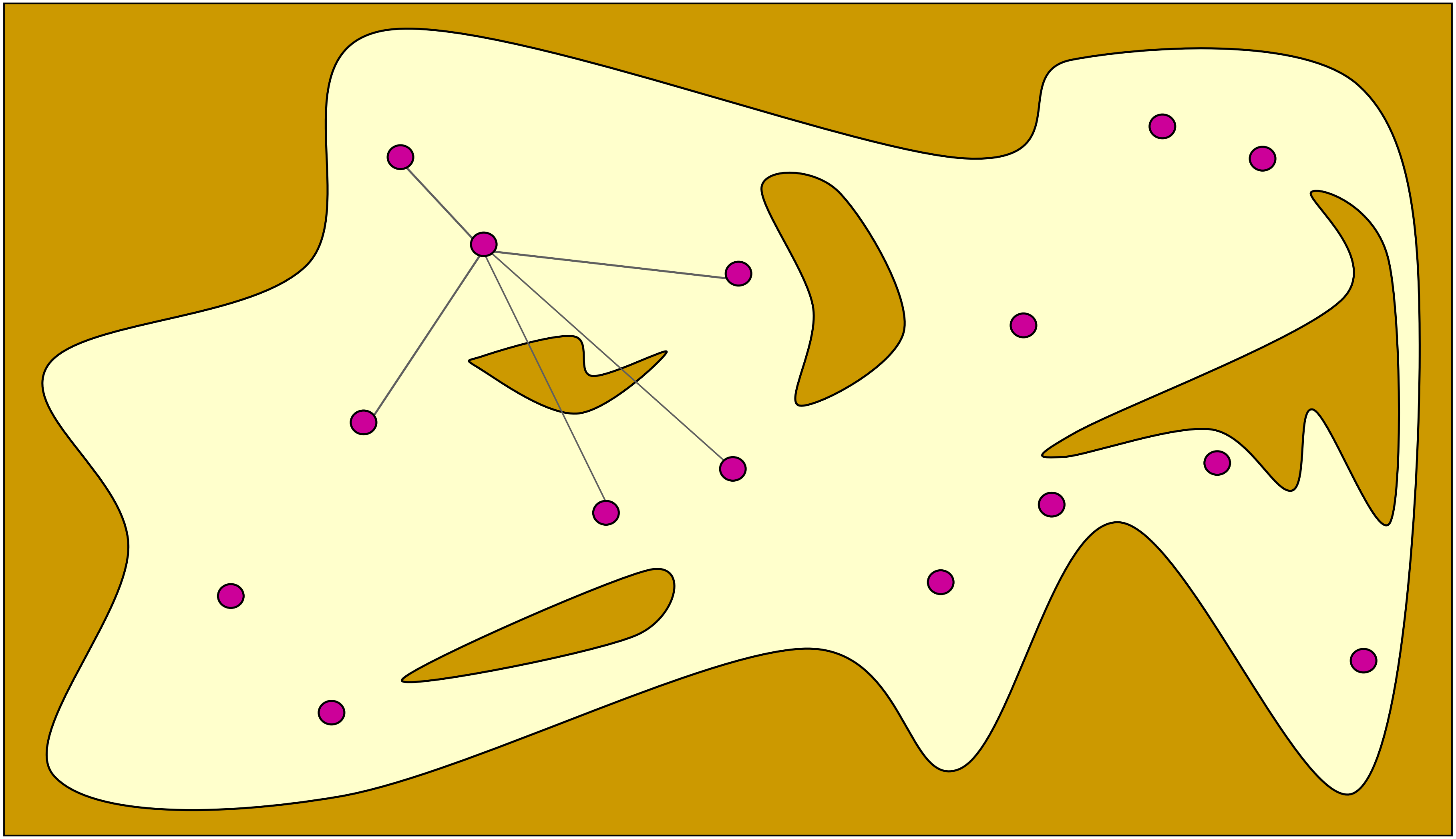
Probabilistic Roadmap (PRM)

The feasible configurations are retained as "milestones" (states)



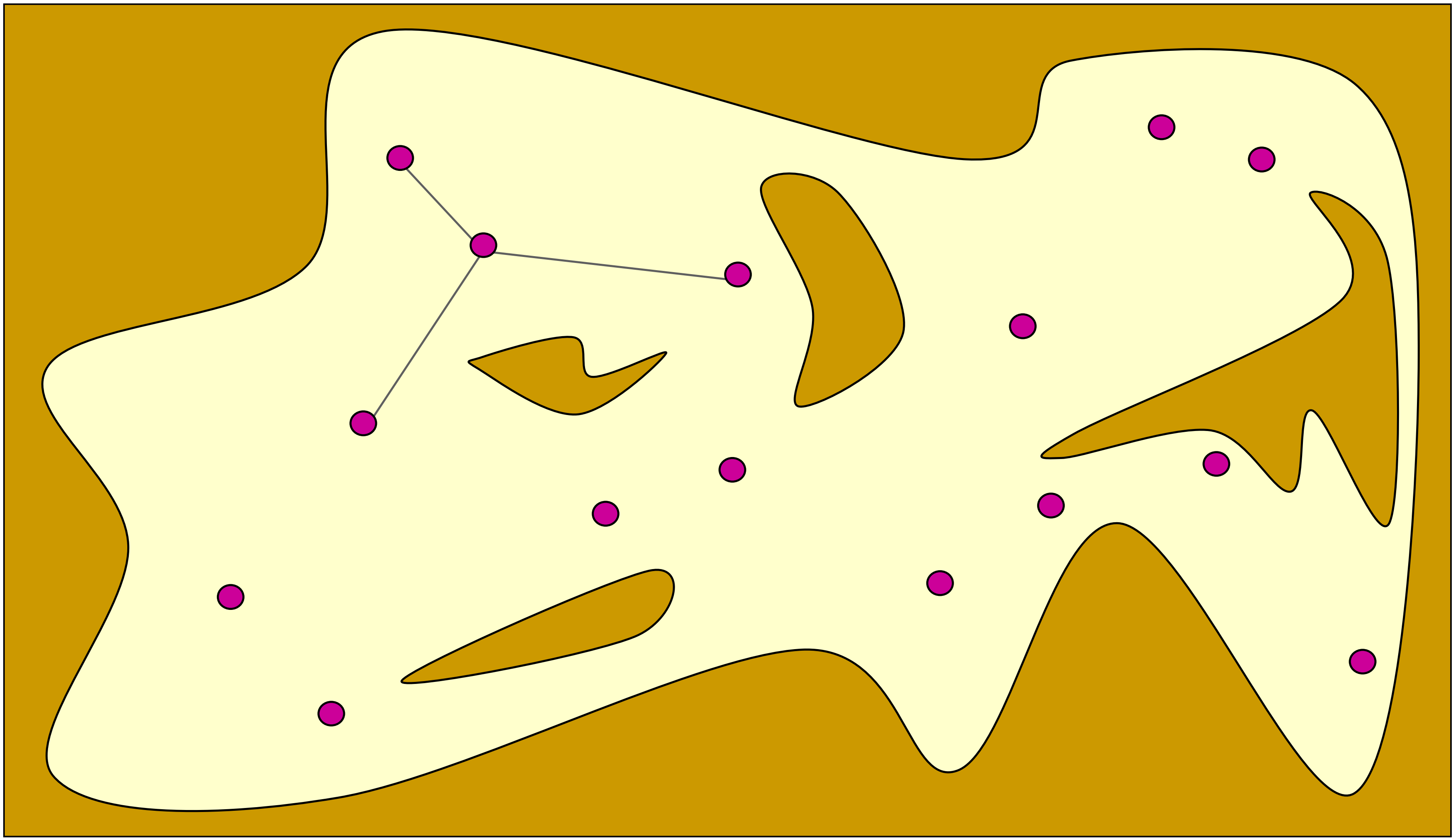
Probabilistic Roadmap (PRM)

Each milestone is linked by straight paths to its k-nearest neighbors



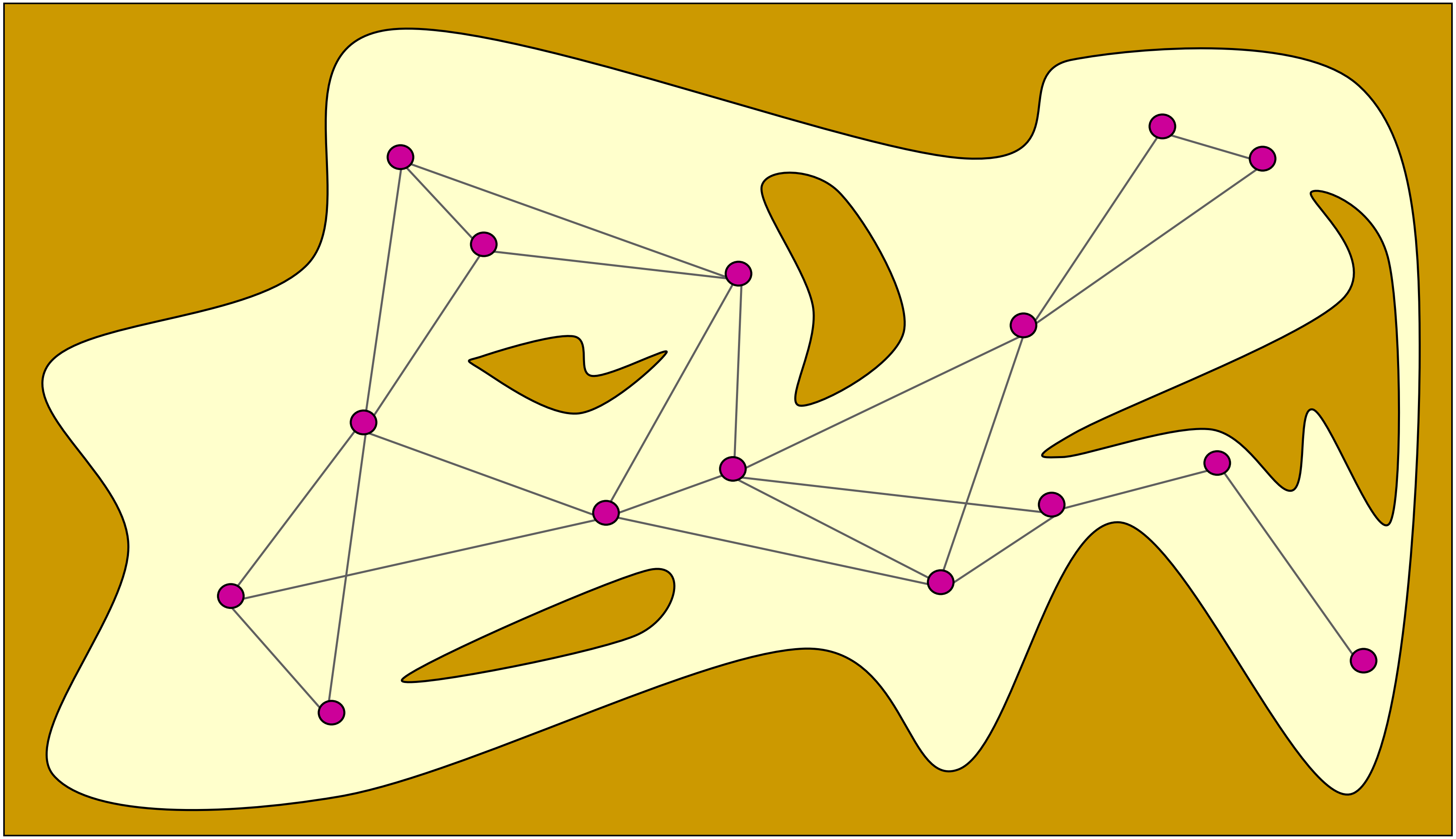
Probabilistic Roadmap (PRM)

Each milestone is linked by straight paths to its k-nearest neighbors



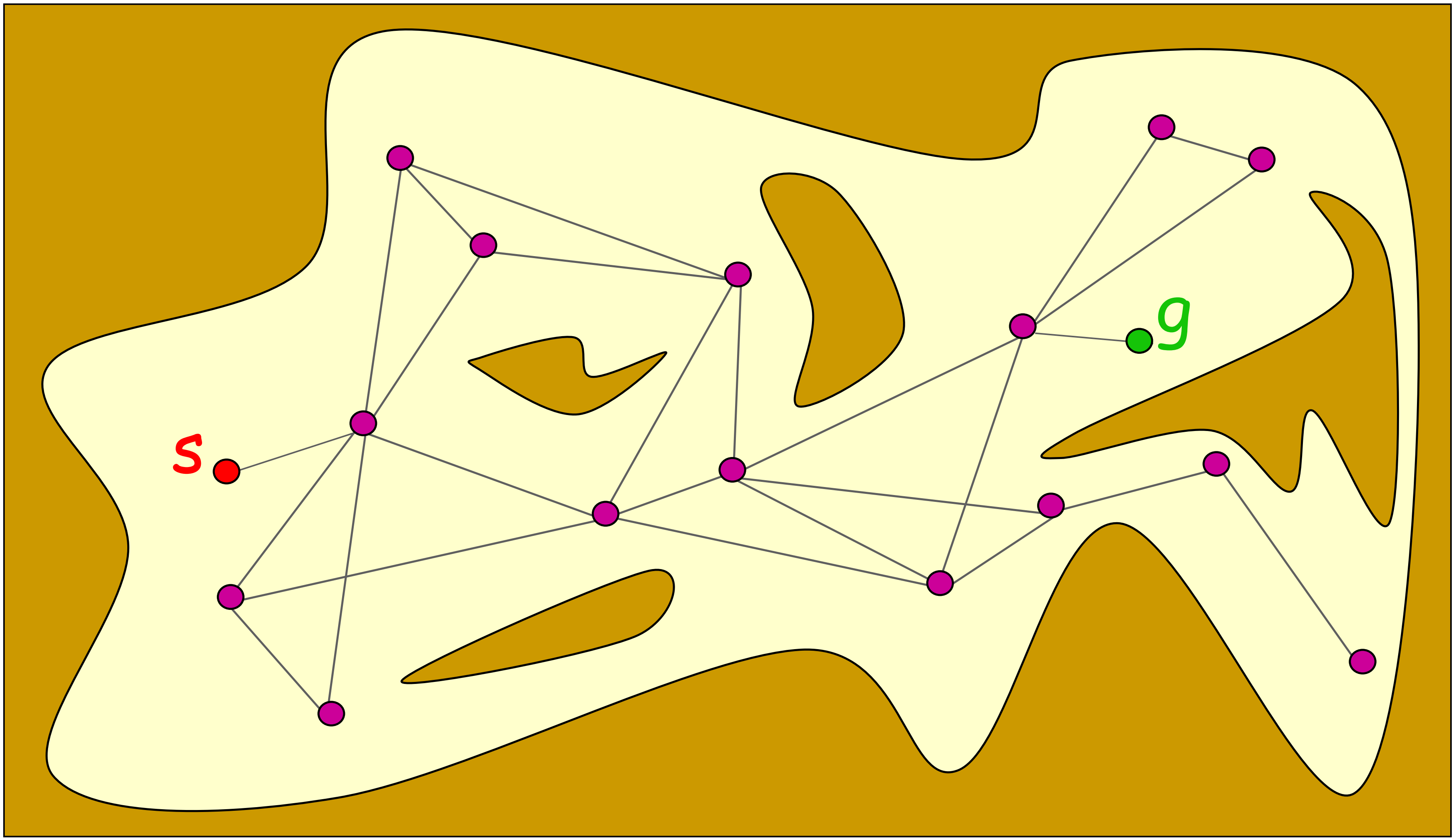
Probabilistic Roadmap (PRM)

The collision-free links are retained to form the PRM (state graph)



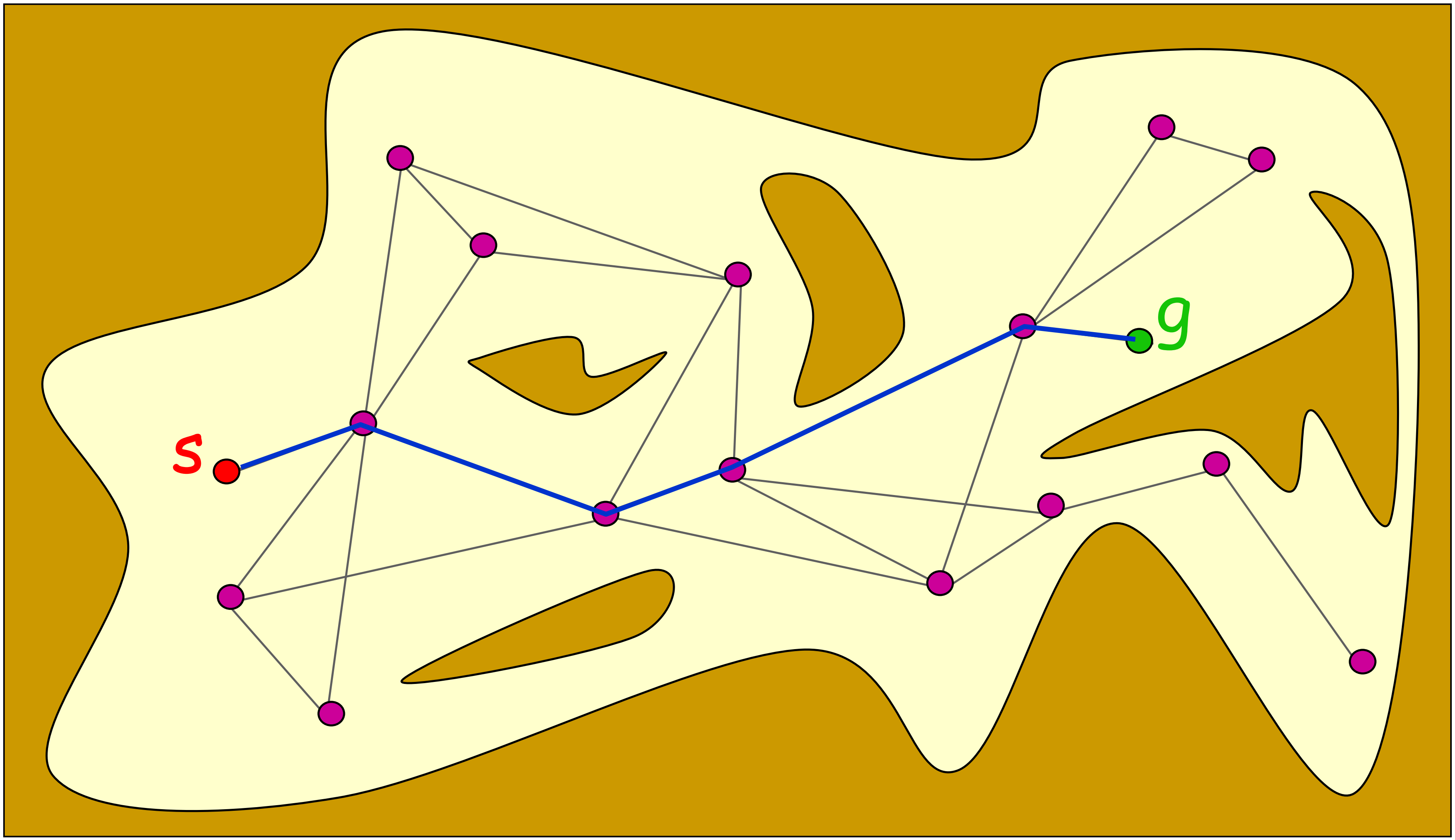
Probabilistic Roadmap (PRM)

The start and goal configurations are connected to nodes of the PRM



Probabilistic Roadmap (PRM)

The PRM is searched for a path from s to g

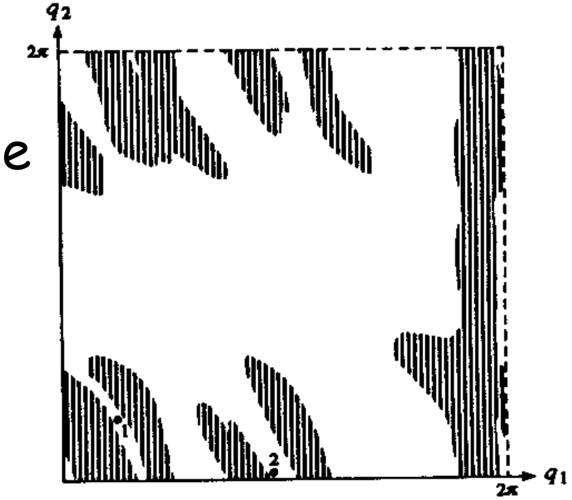


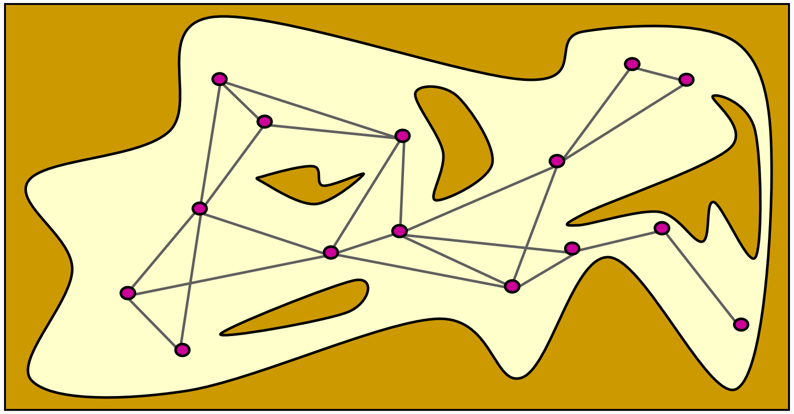
Continuous state space

Discretization

Search

C-space



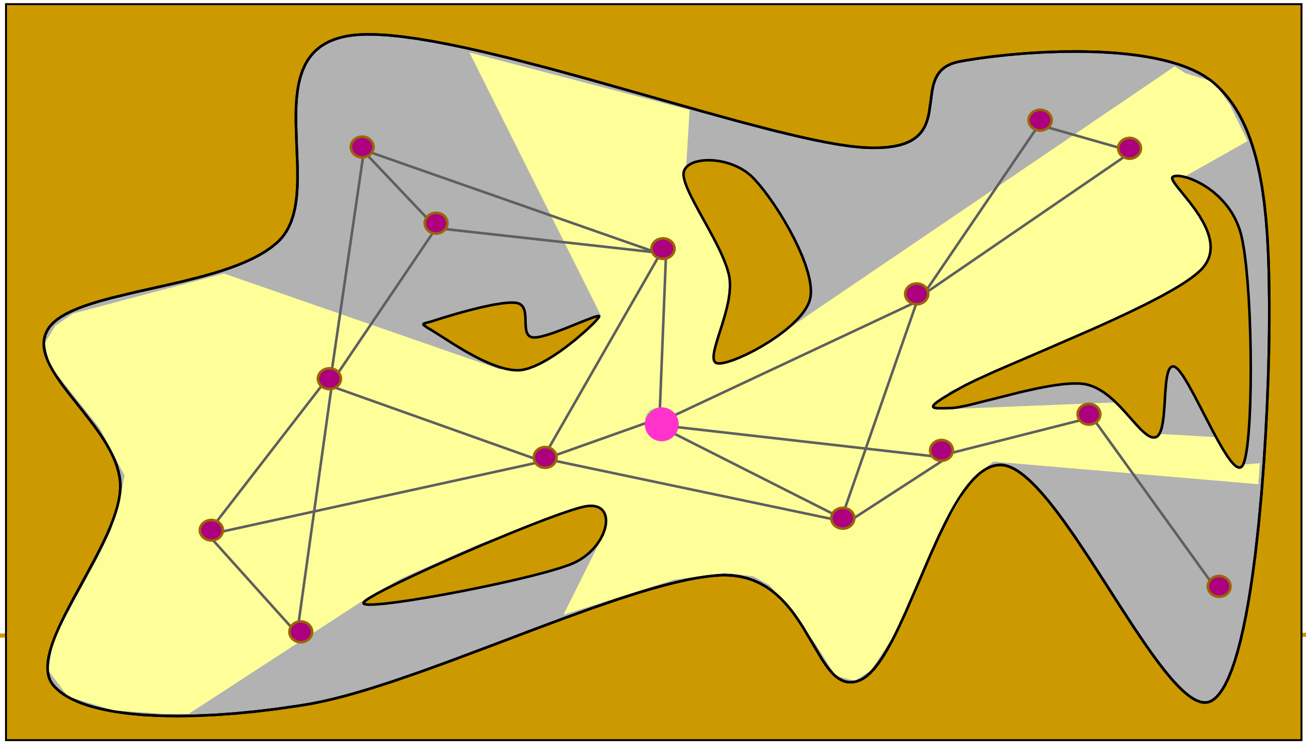


A*

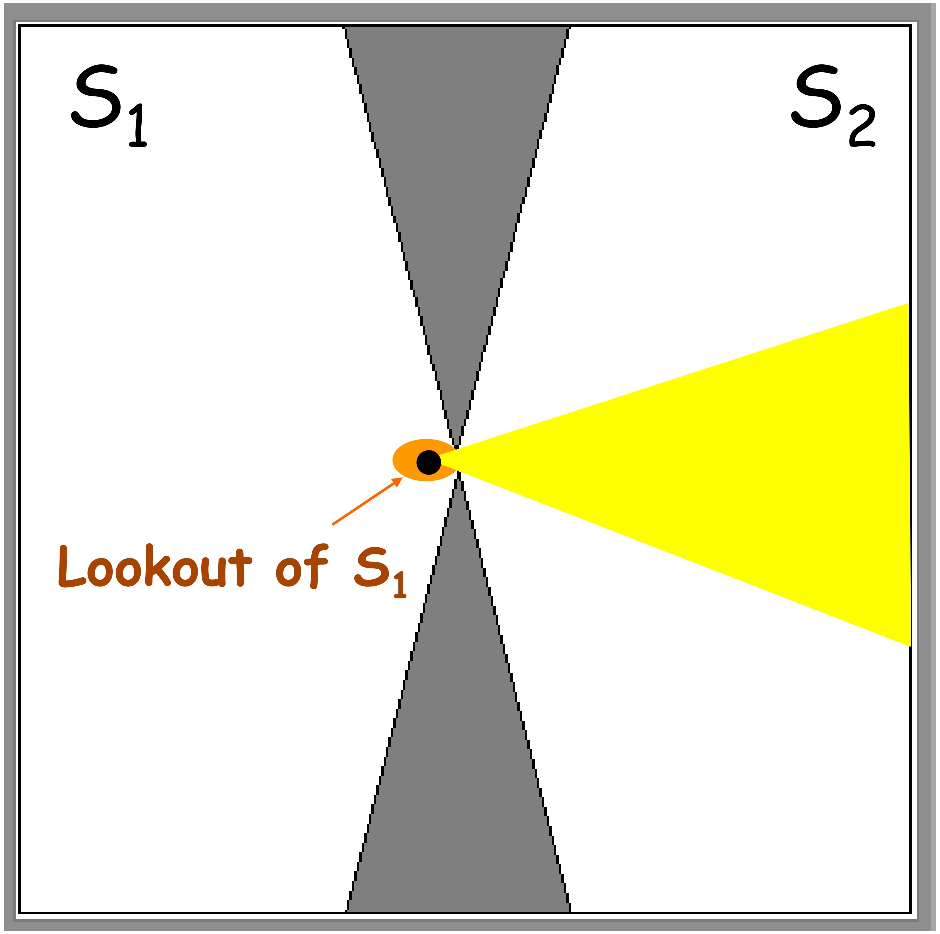
Why Does PRM Work?

- Because most feasible spaces verifies some good geometric (visibility) properties
  - Every configuration “sees” a significant fraction of the feasible space

→ A relatively small number of milestones and connections between them are sufficient to cover most feasible spaces with high probability



Narrow-Passage Issue



The lookout of a subset S of the feasible space is the set of all configurations in S from which it is possible to "see" a significant fraction of the feasible space outside S

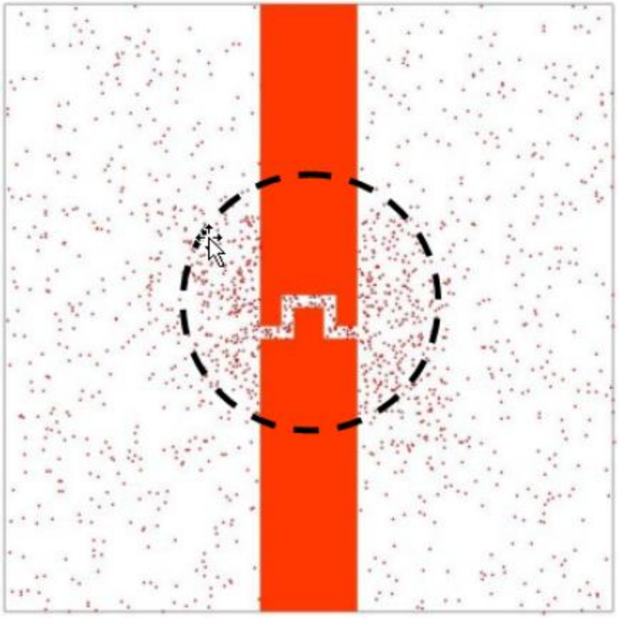
The feasible space is **expansive** if all of its subsets have a large lookout

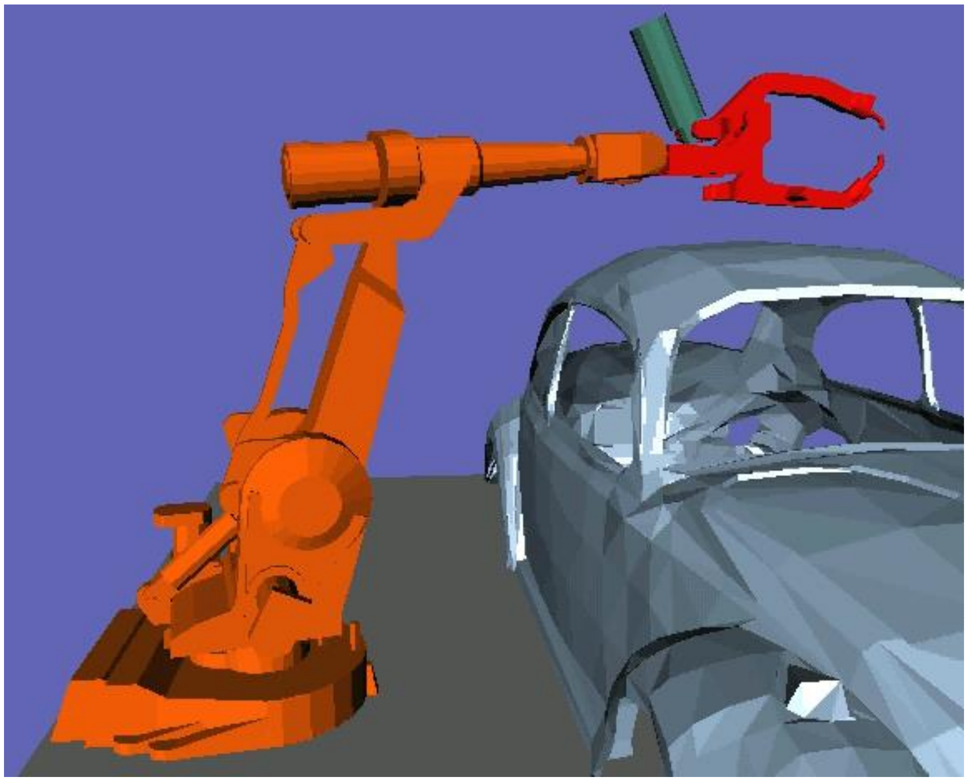
Probabilistic Completeness of a PRM Motion Planner

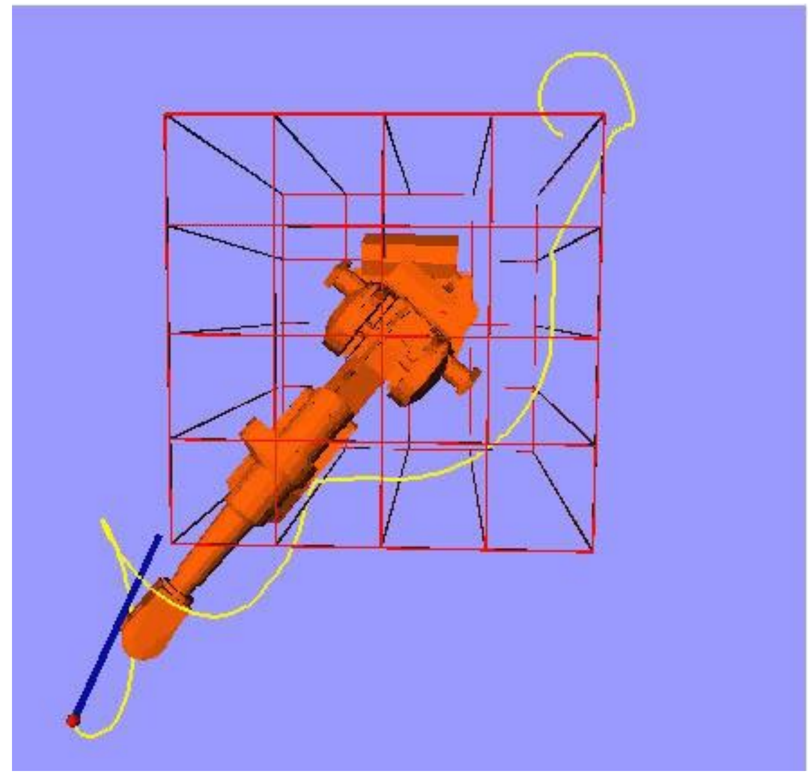
- In an expansive feasible space, the probability that a PRM planner with uniform sampling strategy finds a solution path, if one exists, goes to 1 exponentially with the number of milestones (~ running time)
- A PRM planner can't detect that no path exists. Like A*, it must be allocated a time limit beyond which it returns that no path exists. But this answer may be incorrect. Perhaps the planner needed more time to find one !

Sampling Strategies

- Issue: Where to sample configurations? That is, which probabilistic distribution to use?
- Example: Two-stage sampling strategy:
  - Construct initial PRM with uniform sampling
  - Identify milestones that have few connections to their close neighbors
  - Sample more configurations around them
  - → Greater density of milestones in “difficult” regions of the feasible space







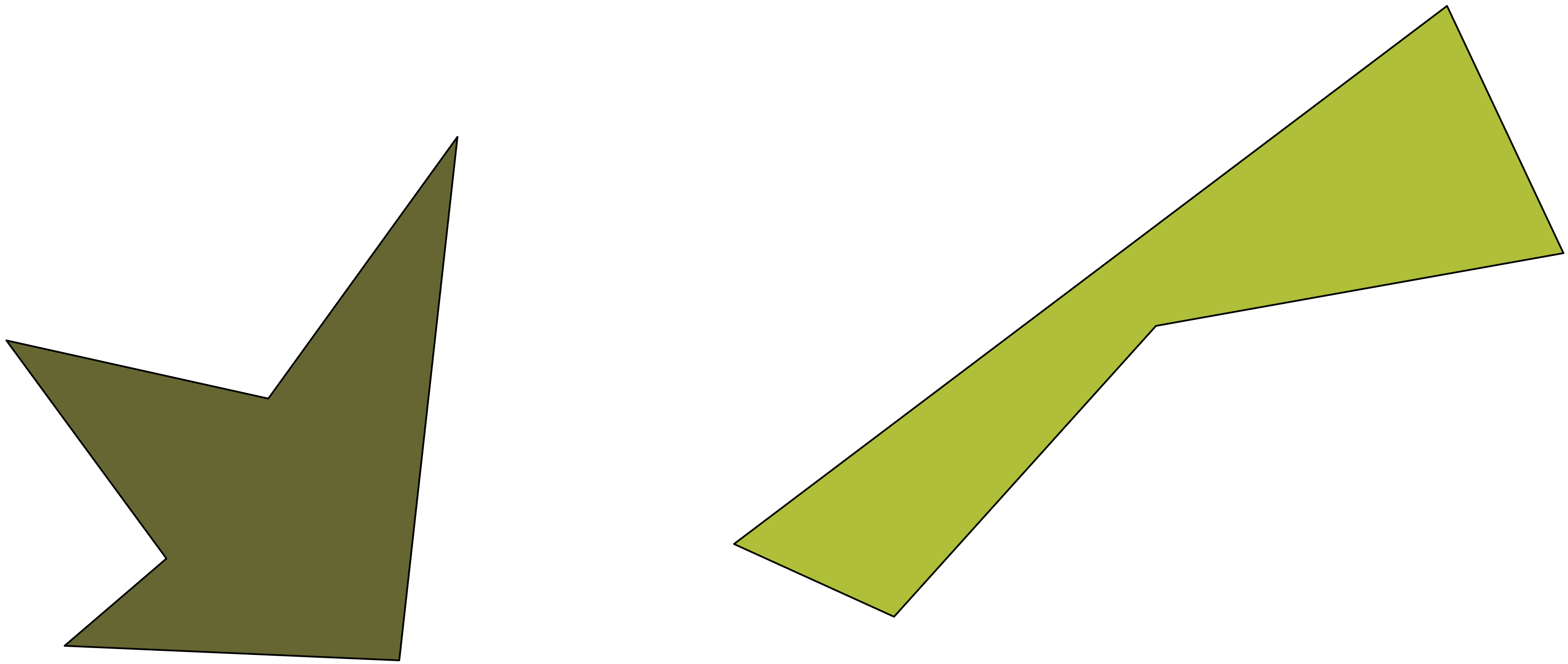
Checking for

Collisions

In which we see how to escape geometrical complexity via pruning

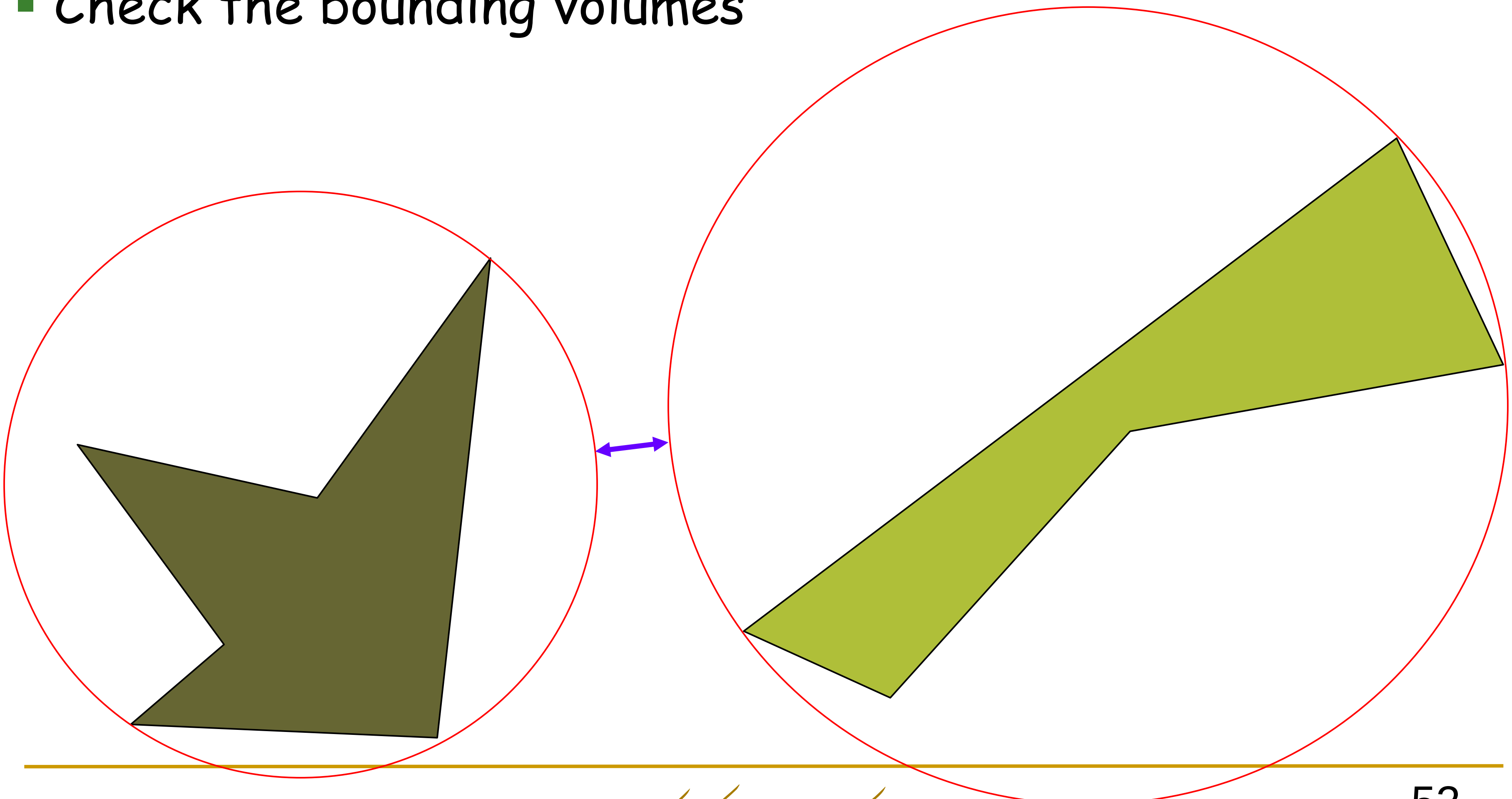
Collision Checking

- Check whether objects overlap



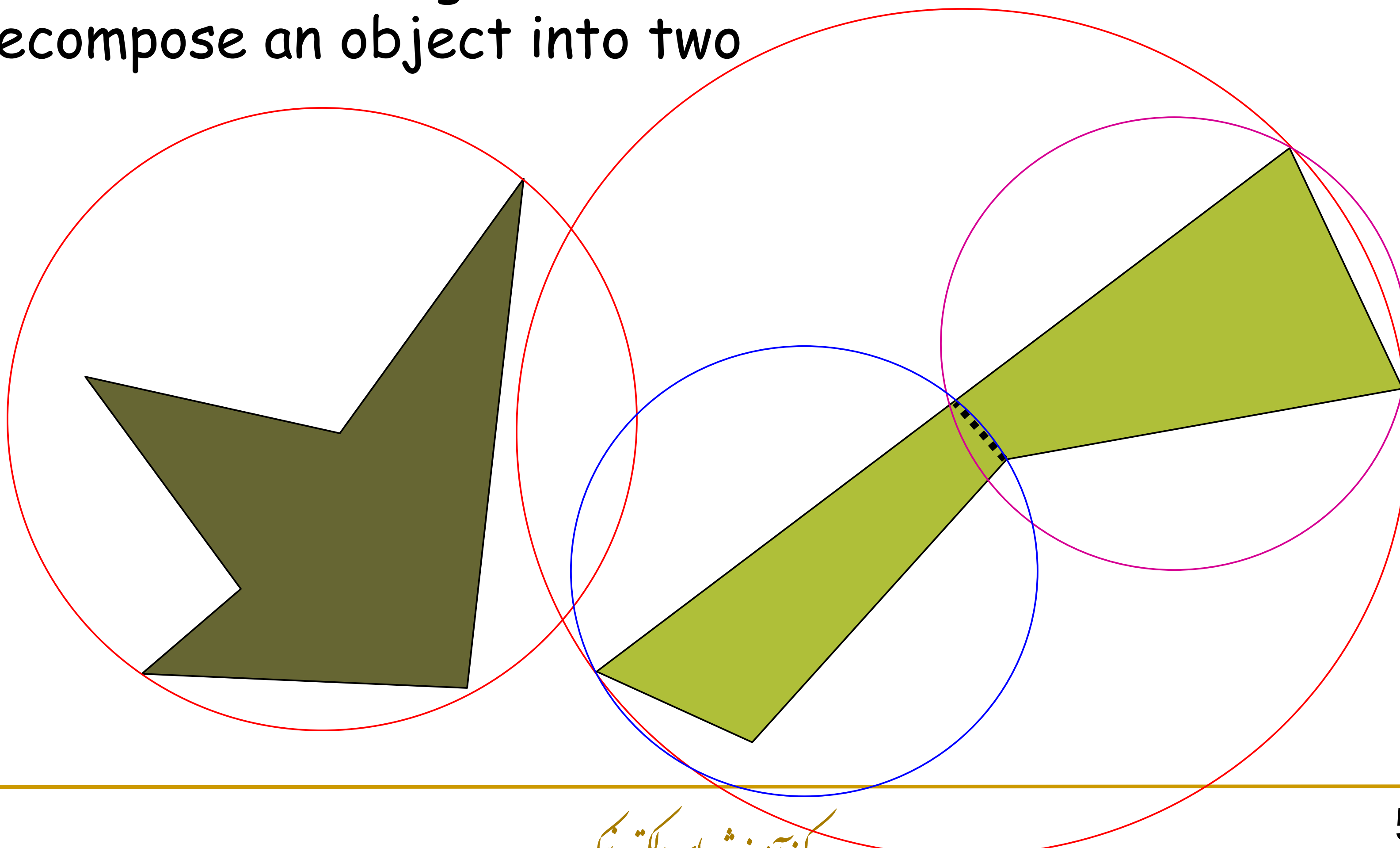
Hierarchical Collision Checking

- Enclose objects into bounding volumes (spheres or boxes)
- Check the bounding volumes



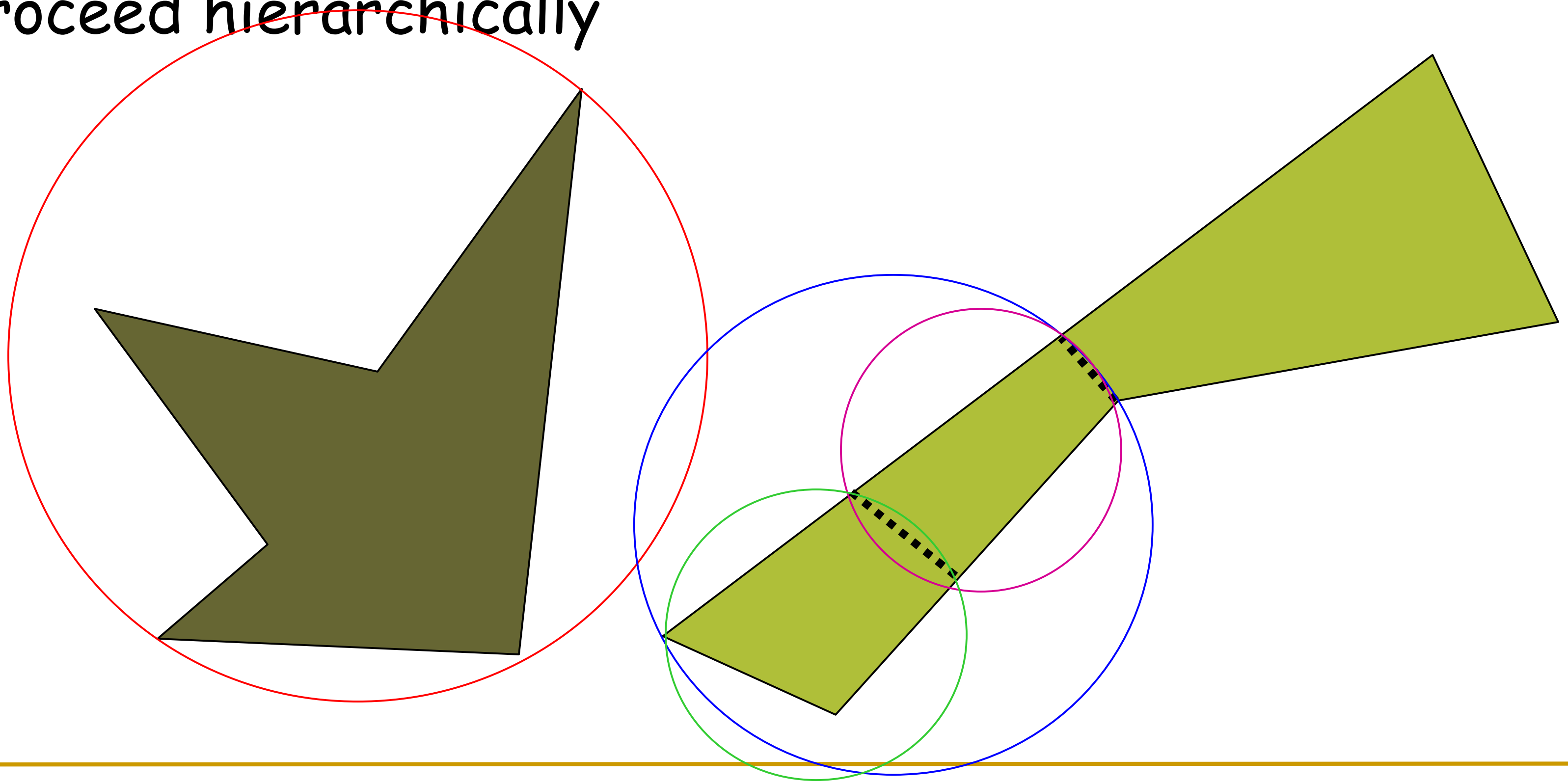
Hierarchical Collision Checking

- Enclose objects into bounding volumes (spheres or boxes)
- Check the bounding volumes first
- Decompose an object into two



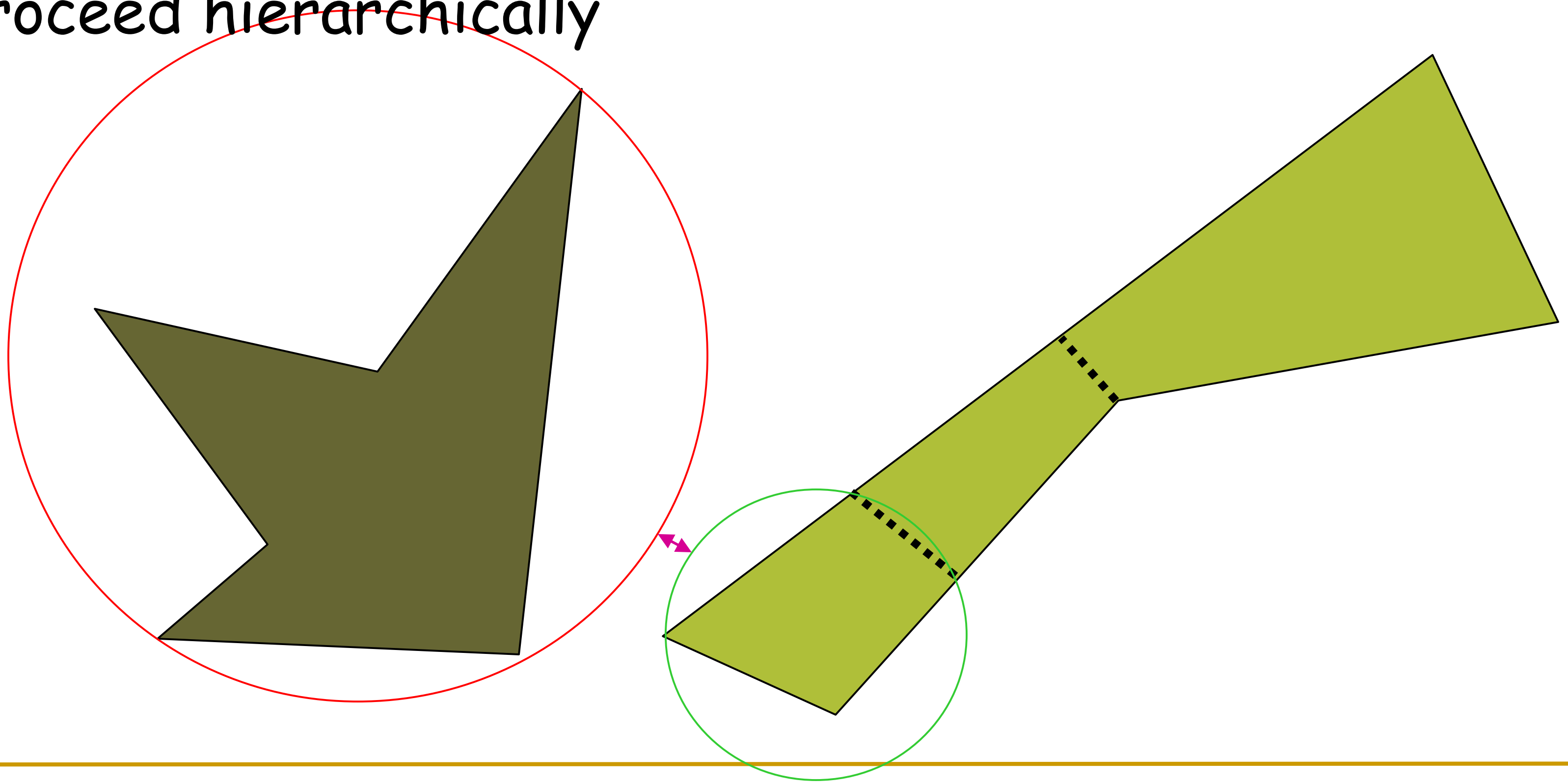
Hierarchical Collision Checking

- Enclose objects into bounding volumes (spheres or boxes)
- Check the bounding volumes first
- Decompose an object into two
- Proceed hierarchically



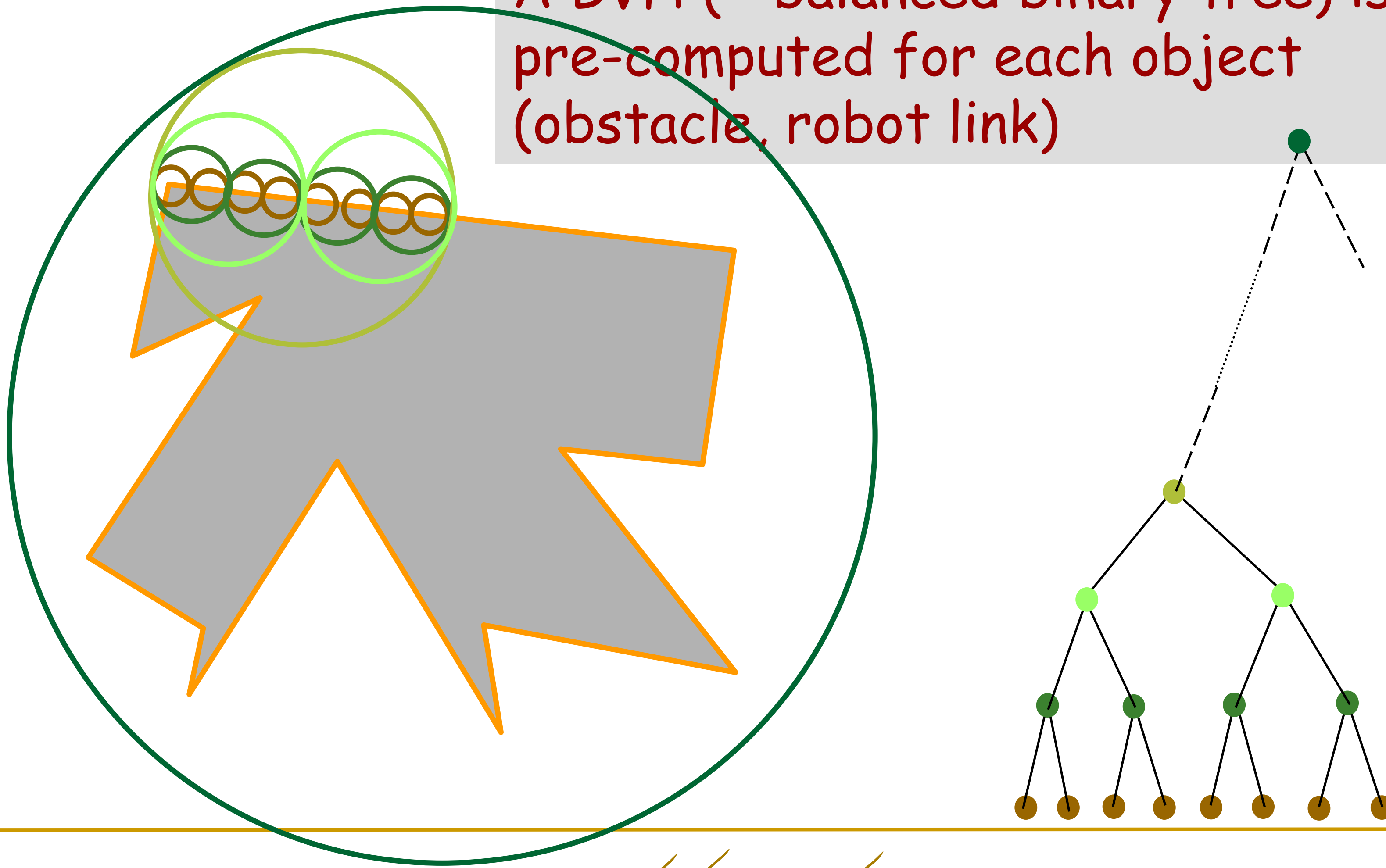
Hierarchical Collision Checking

- Enclose objects into bounding volumes (spheres or boxes)
- Check the bounding volumes first
- Decompose an object into two
- Proceed hierarchically



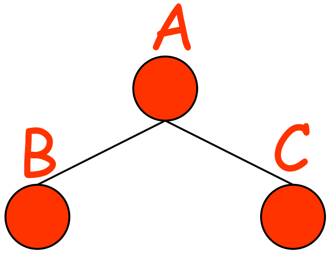
Bounding Volume Hierarchy (BVH)

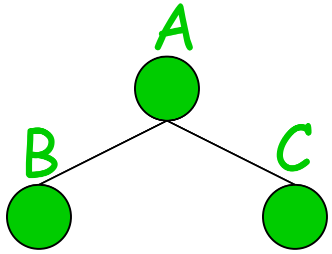
A BVH (~ balanced binary tree) is pre-computed for each object (obstacle, robot link)



BVH of a 3D Triangulated Cat

Collision Checking Between Two Objects





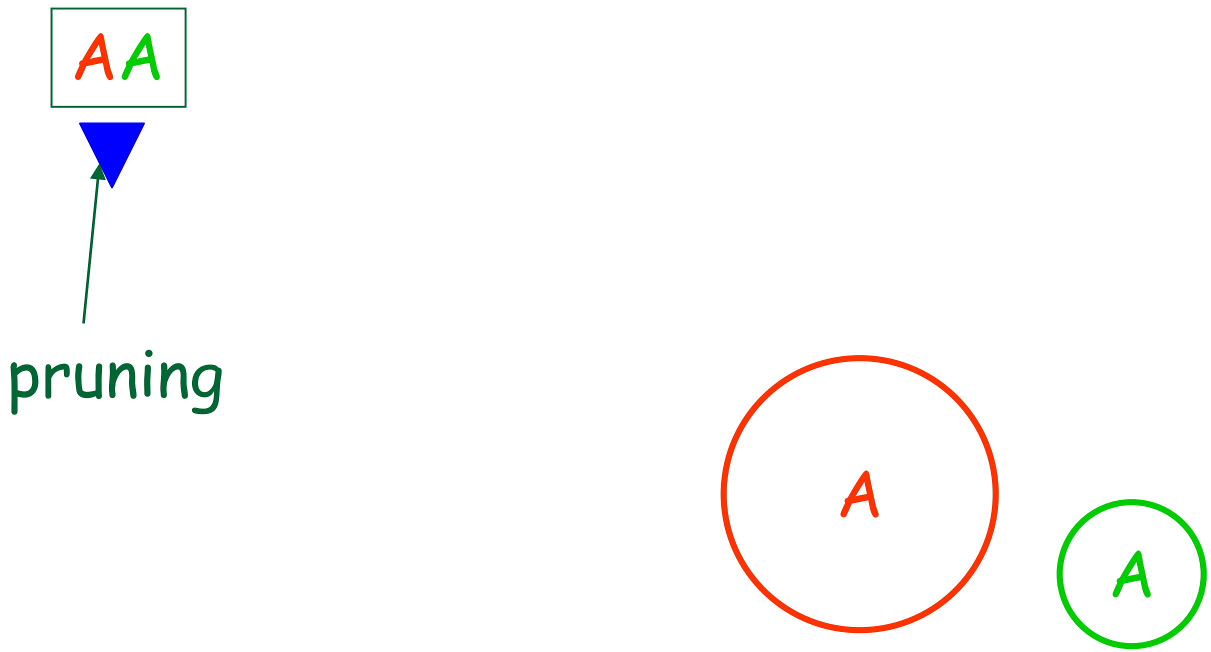
BVH of object 1 BVH of object 2

[Usually, the two trees have different sizes]

→ Search for a collision

Search for a Collision

Search tree

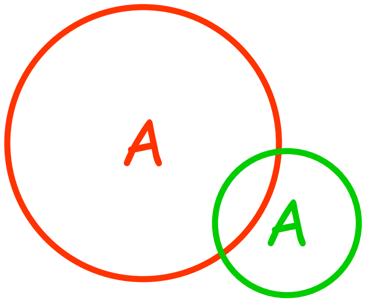


Search for a Collision

Search tree

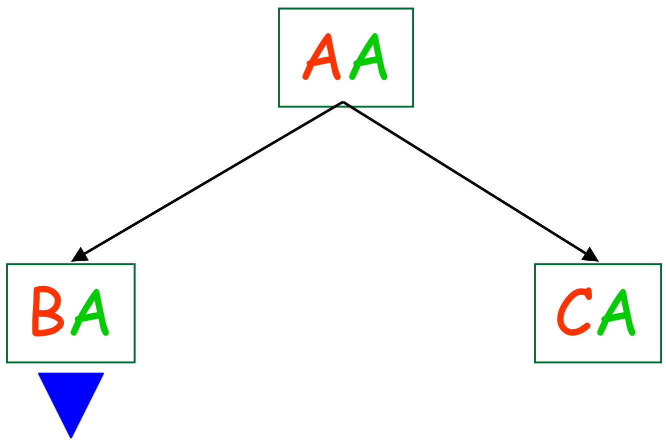
AA

Heuristic: Break the largest BV

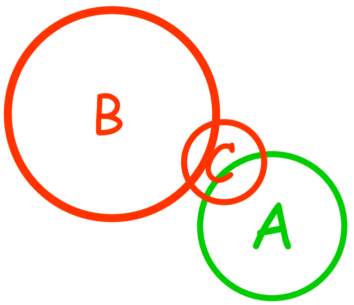


Search for a Collision

Search tree

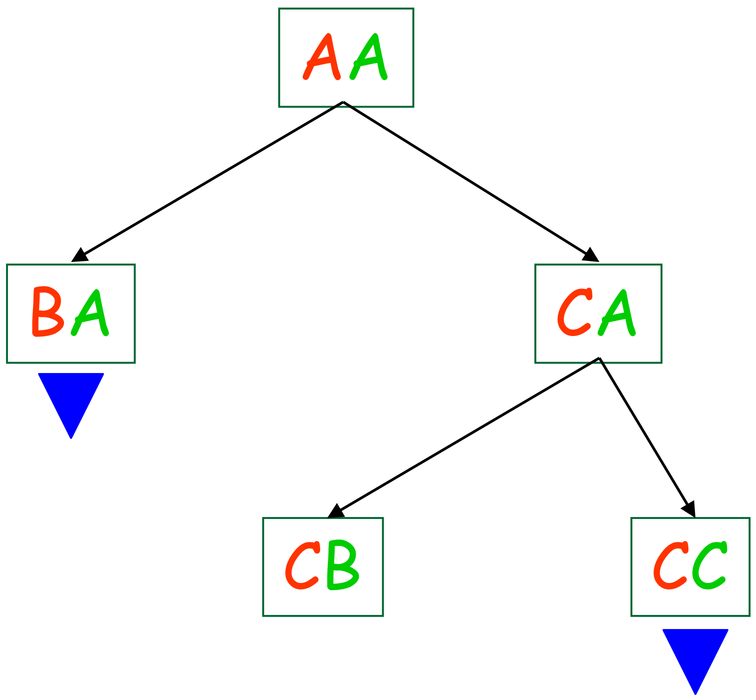


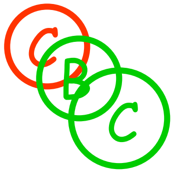
Heuristic: Break the largest BV



Search for a Collision

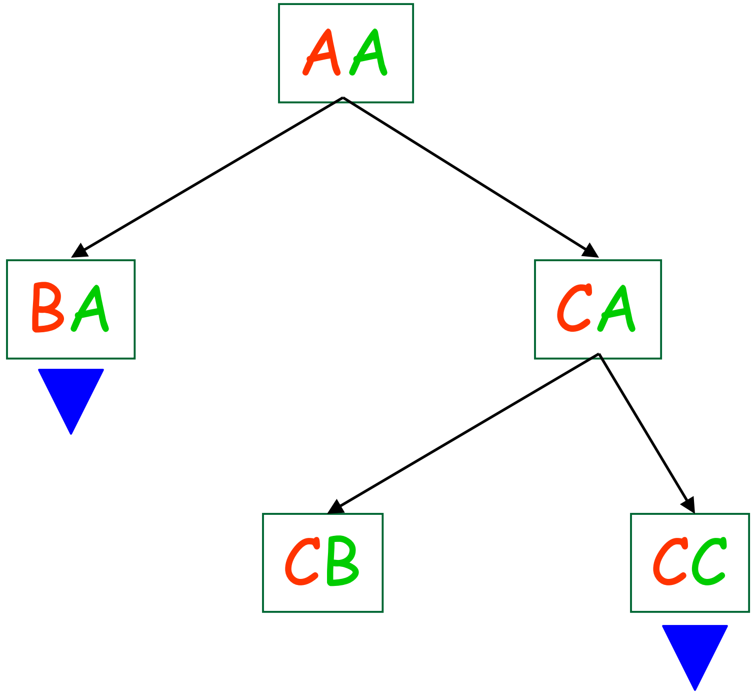
Search tree

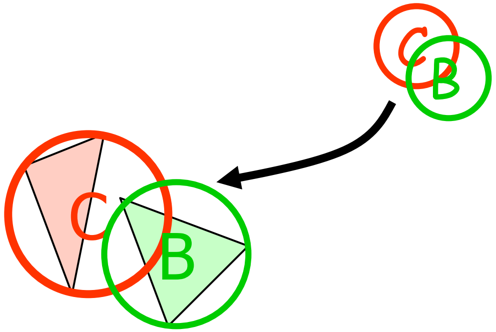




Search for a Collision

Search tree



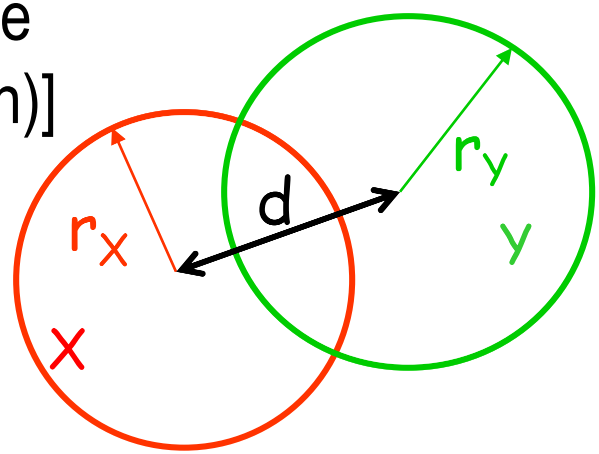


If two leaves of the BVH's overlap (here, C and B) check their content for collision

Search Strategy

- If there is no collision, all paths must eventually be followed down to pruning or a leaf node
- But if there is collision, one may try to detect it as quickly as possible
  - Greedy best-first search strategy with $f(N) = h(N) = d/(rX+rY)$

    [Expand the node XY with largest relative overlap (most likely to contain a collision)]



Auxiliary Searches

- So, to discretize the state space of a motion planning problem, a PRM planner performs thousands of auxiliary searches (sometimes even more) to detect collisions !
  - But from an outsider's point of view the search of the PRM looks like the main search
- Fortunately, hierarchical collision checkers are quite fast
  - On average, over 10,000 collision checks per second for two 3-D objects each described by 500,000 triangles, on a PC
- Checks are much faster when the objects are either neatly separated (early pruning) or neatly overlapping (quick detection of collision)

Creating Hierarchy in

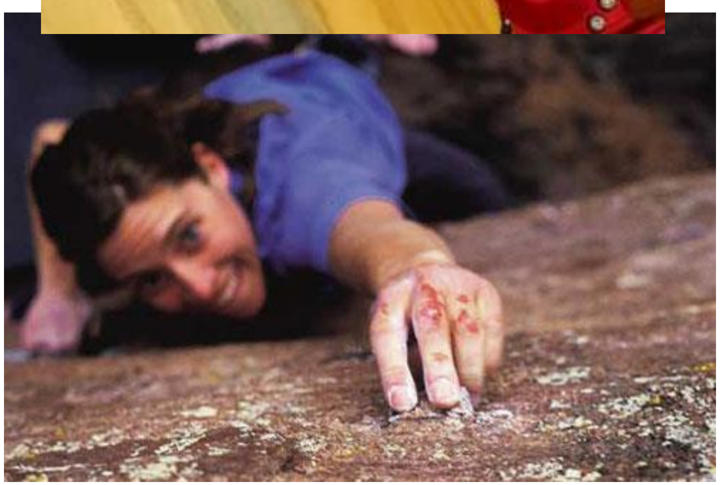
Multi-step Planning

Where complexity is reduced by ignoring lower level details

Free-Climbing Robot

LEMUR IIb robot (created by NASA/JPL)

Only friction and internal DOFs are used to achieve equilibrium

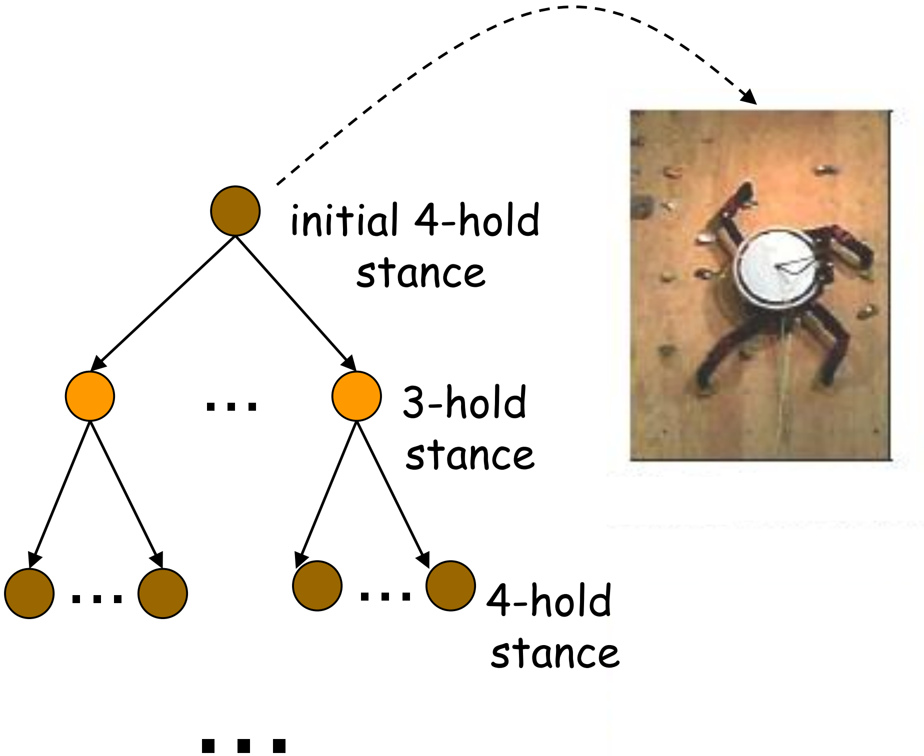


Two Levels of Planning

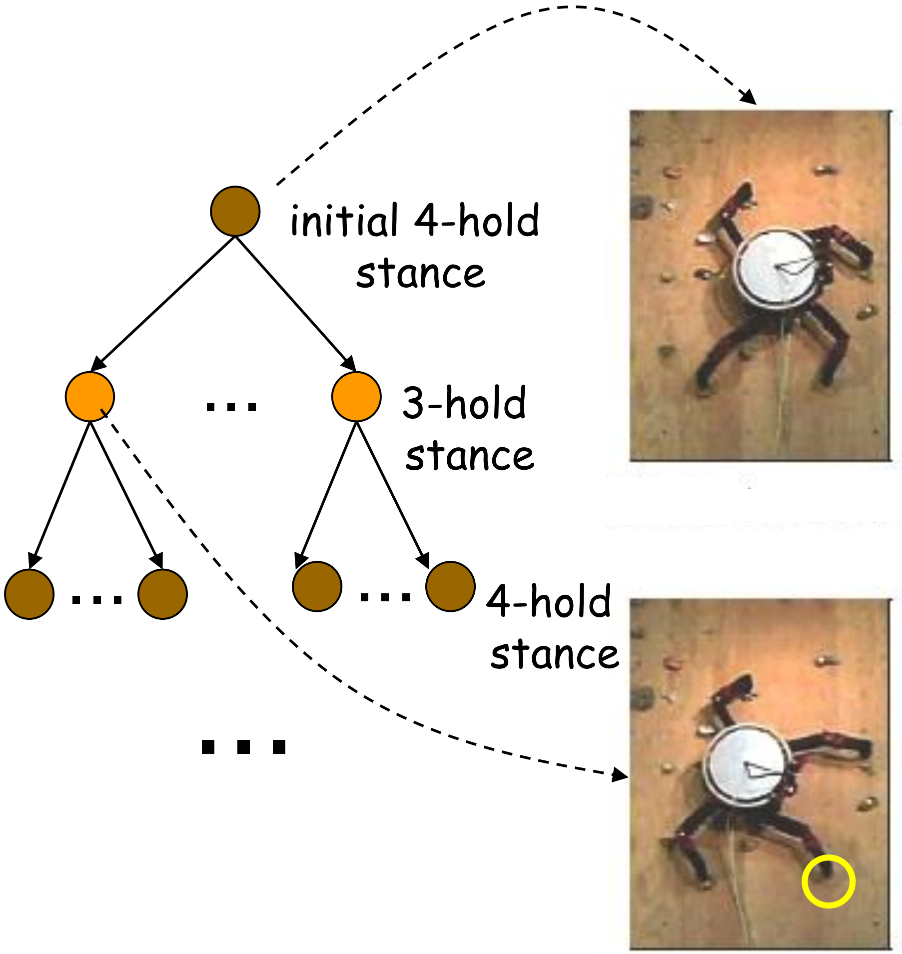
- One-step planning:
  - Plan a path for moving a foot/hand from one hold to another
  - Can be solved using a PRM planner

- Multi-step planning:
  - Plan a sequence of one-step paths
  - Can be solved by searching a stance space

Multi-Step Planning

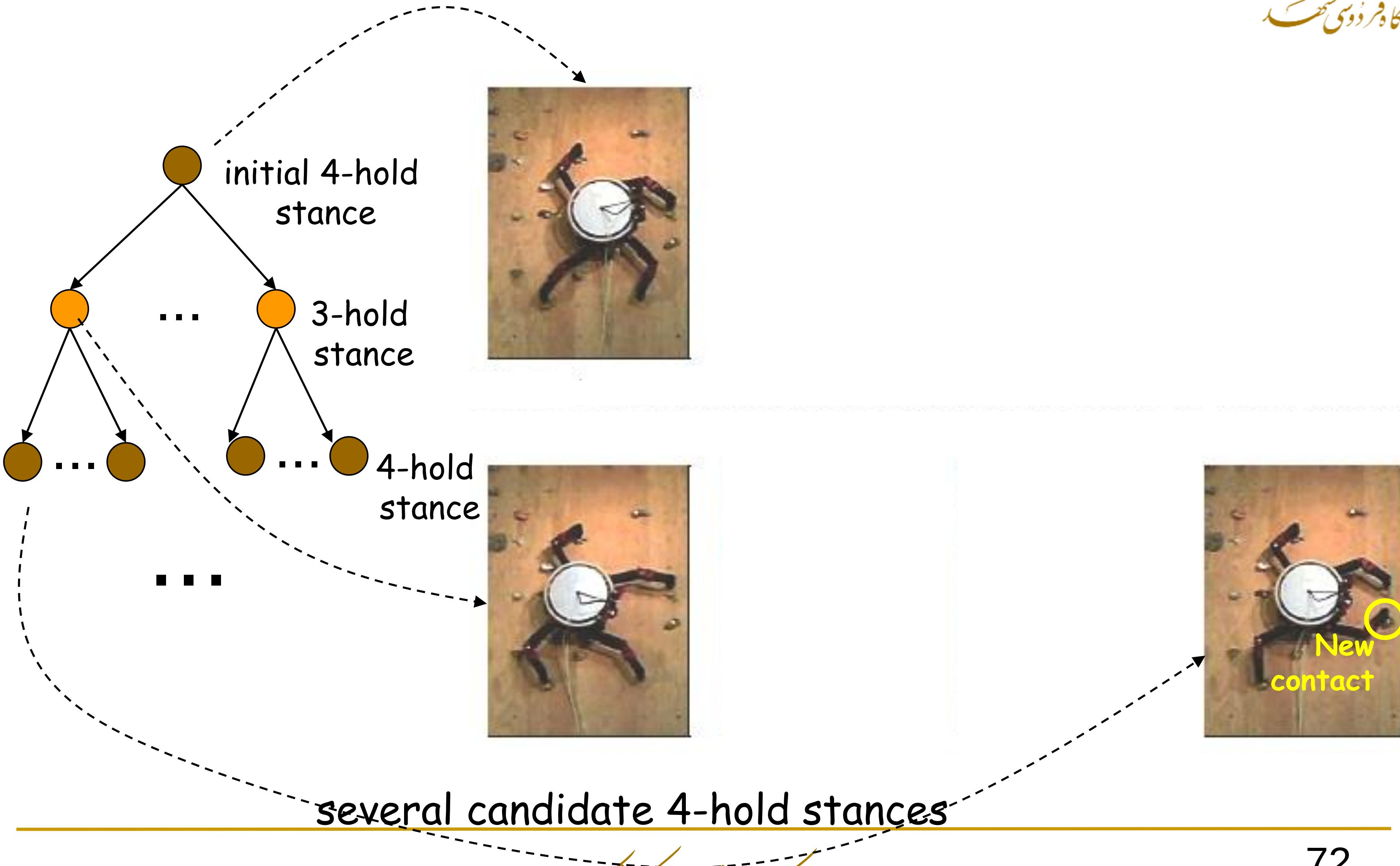


Multi-Step Planning

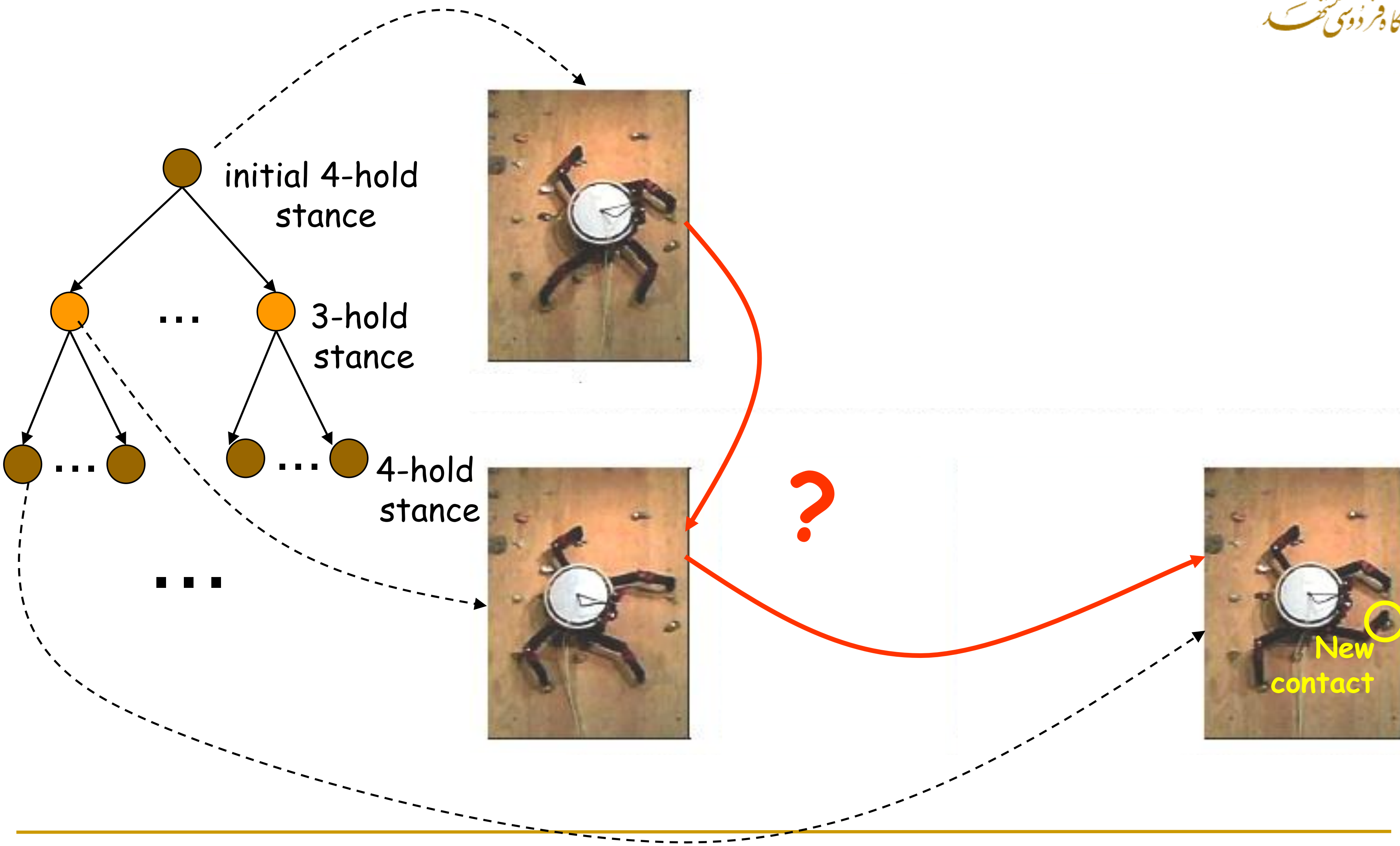


4 possible 3-hold stances

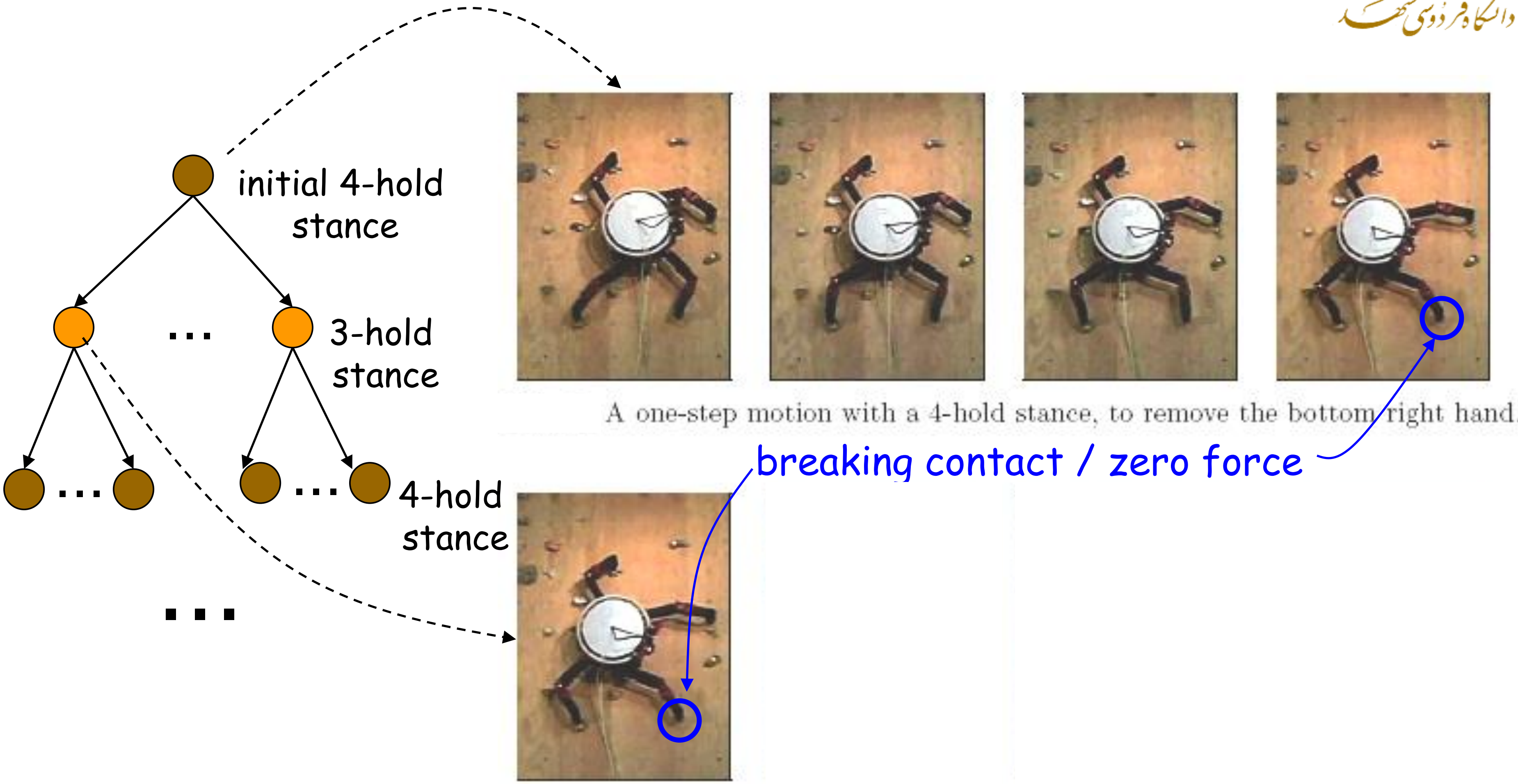
Multi-Step Planning



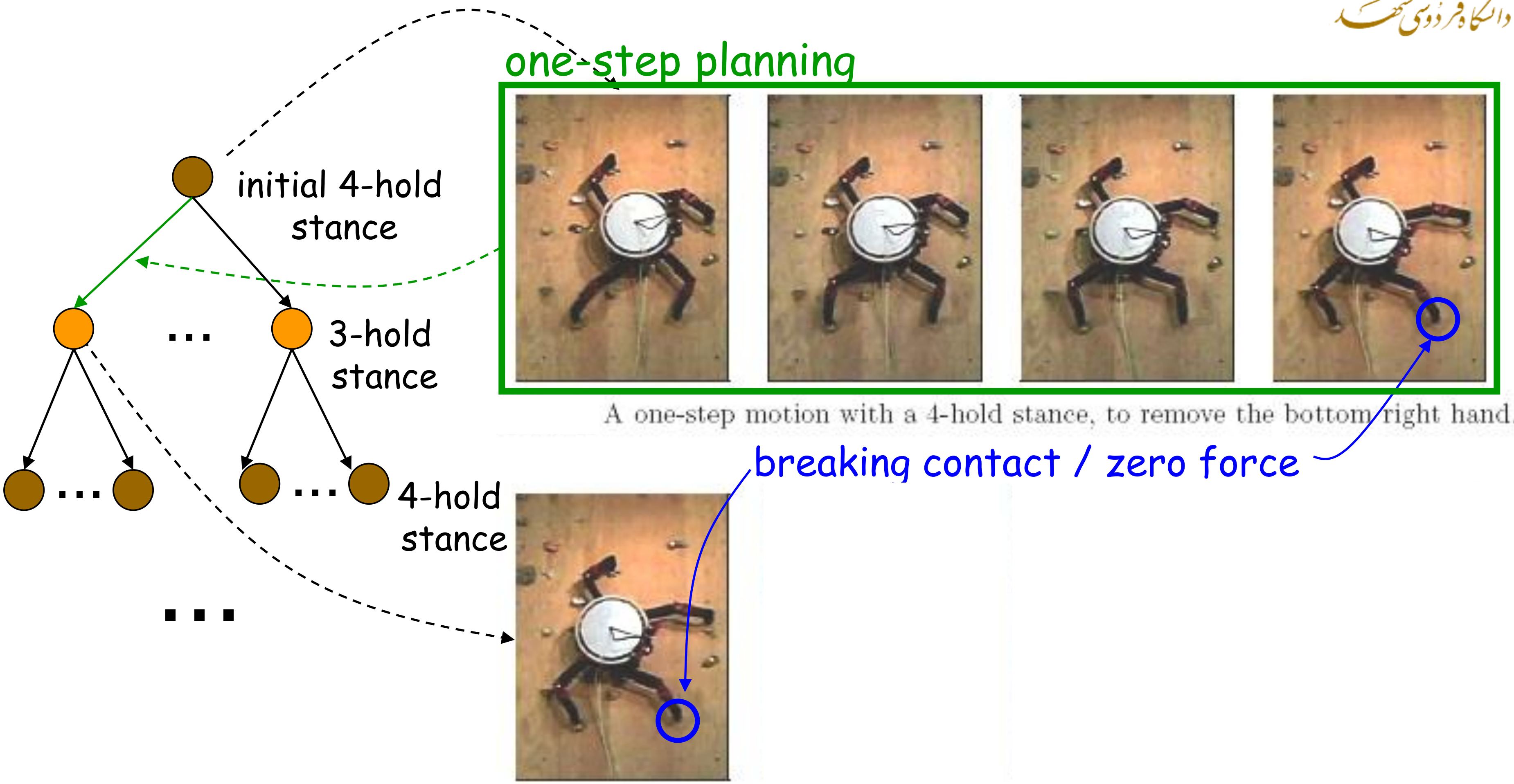
Multi-Step Planning



Multi-Step Planning

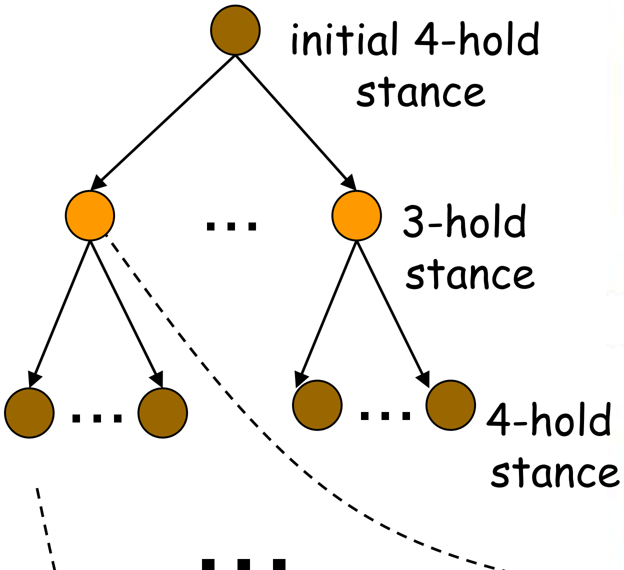


Multi-Step Planning



The one-step planner is needed to determine if a one-step path exists between two stances

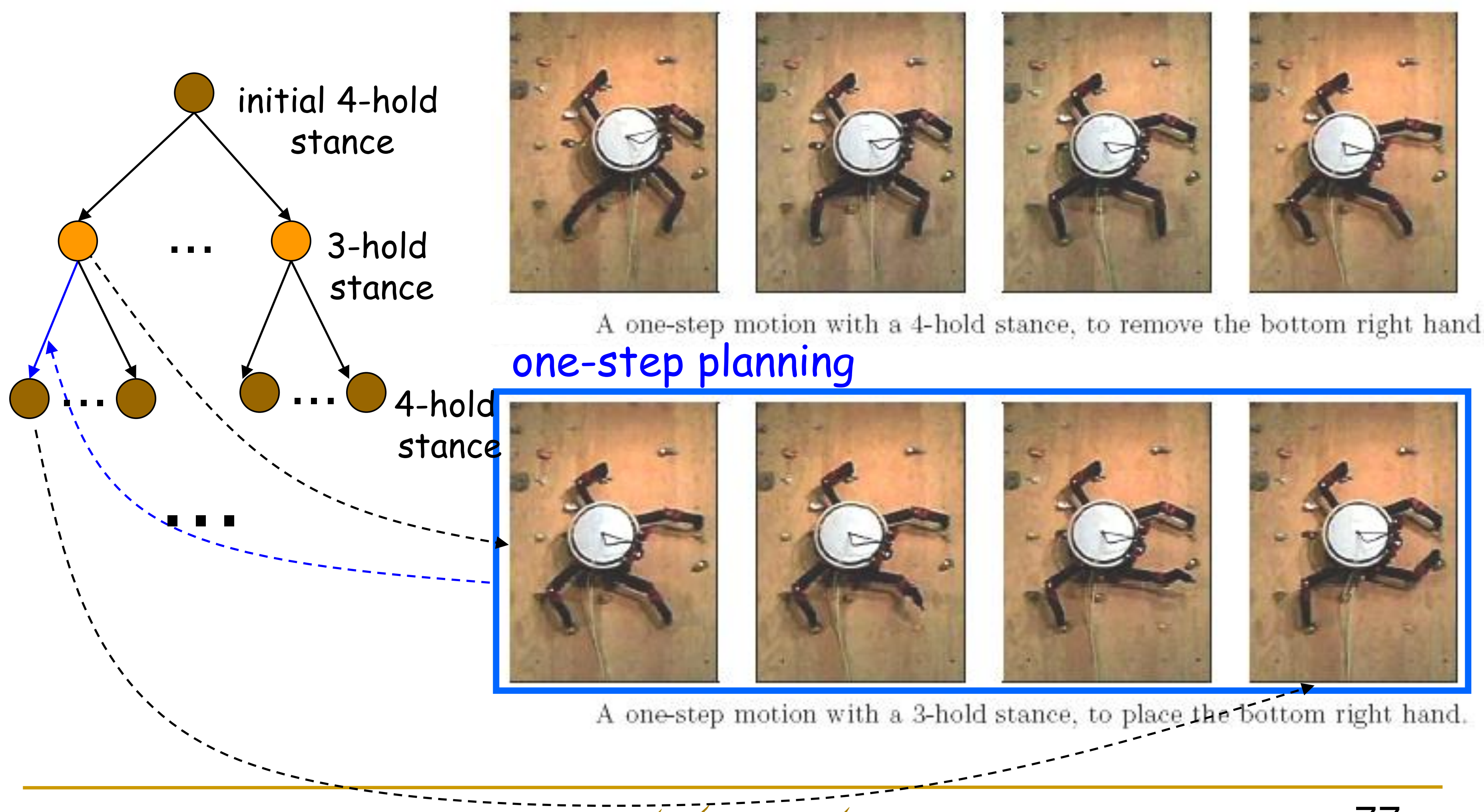
Multi-Step Planning



A one-step motion with a 4-hold stance, to remove the bottom right hand.

A one-step motion with a 3-hold stance, to place the bottom right hand.

Multi-Step Planning

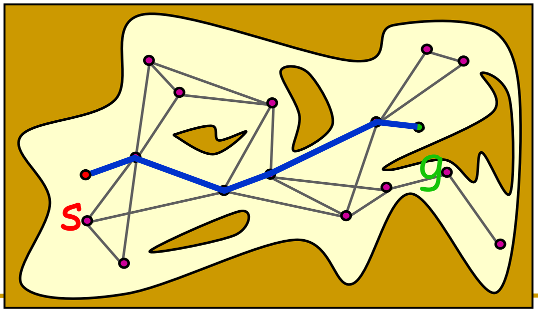


A one-step motion with a 4-hold stance, to remove the bottom right hand.

A one-step motion with a 3-hold stance, to place the bottom right hand.

One-Step Planning

- The contact constraints define specific C-space that is easy to sample at random
- It is also easy to test (self-)collision avoidance and equilibrium constraints at sampled configurations
  - → PRM planning



Planning as Search Problems

Multi-step planner

↓

One-step planner

↓

Hierarchical collision checking

Many searches !!

(+ searches for planning exploratory moves, for interpreting 3D sensor data, ...)

Dilemma

Multi-step planner

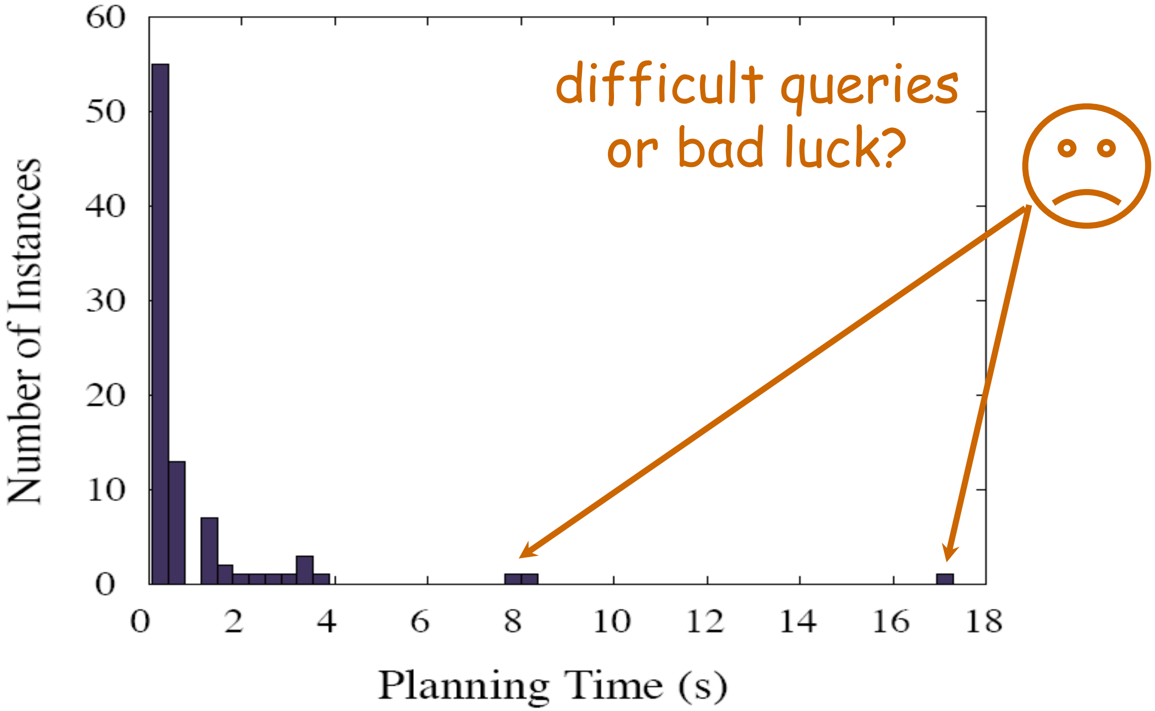
- Several 10,000 queries
- Many infeasible
- Some critical

One-step planner

- A PRM planner can't detect that a query is infeasible
- How much time should it spend on each query?
  - Too little, and it will often fail incorrectly
  - Too much, and it will waste time on infeasible queries

Dilemma

- More than 1,000,000 stances, only 20,000 feasible
- Several 1000 one-step planning queries
  - A large fraction of them have no solution
- The running times for (feasible) queries making up an 88-step path are highly variable



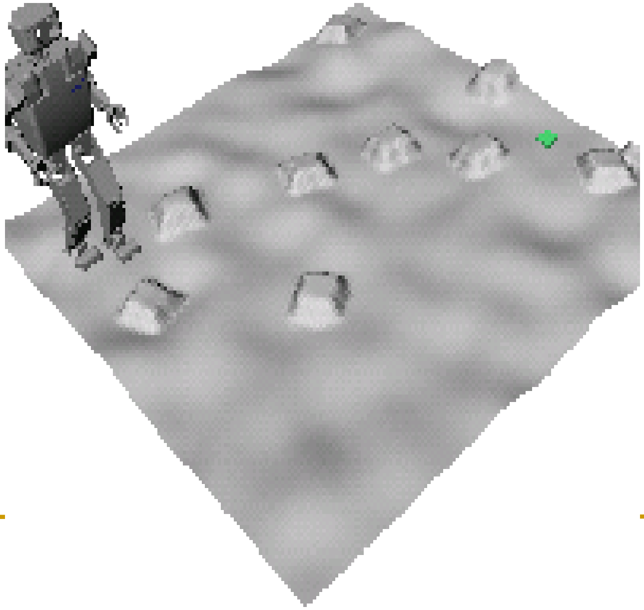
Possible Solution

- Use learning method to train a “feasibility” classifier
- Use this classifier to avoid infeasible one-step queries in the multi-step search tree
  - More on this later in a lecture on Learning (if there is enough time)
- This idea is similar to forward pruning in game trees
  - A risky approximation which occasionally may loose some answers but could save us lots of computations

Other Multi-Step Planning Problems

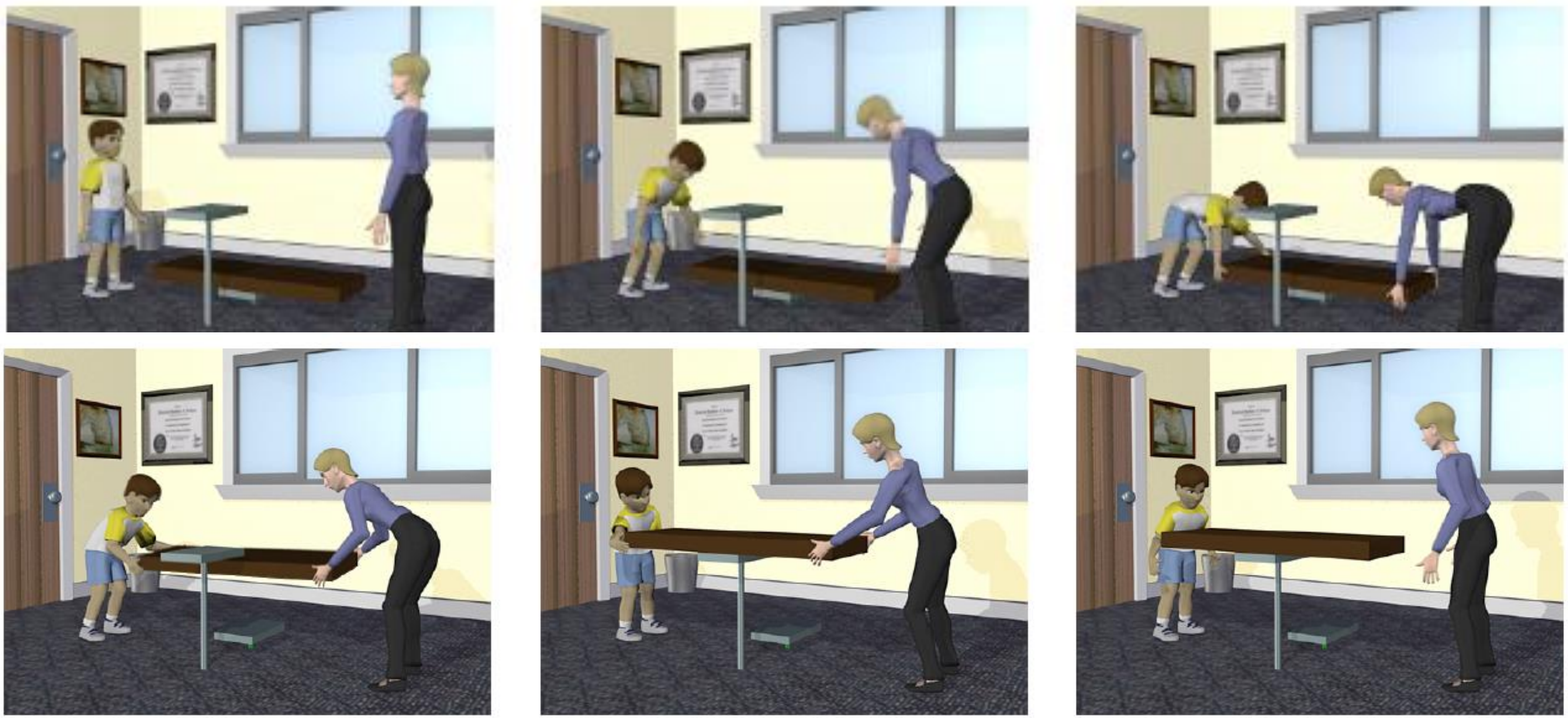
Navigation of legged robots on rough terrain

(stances → foot placements)

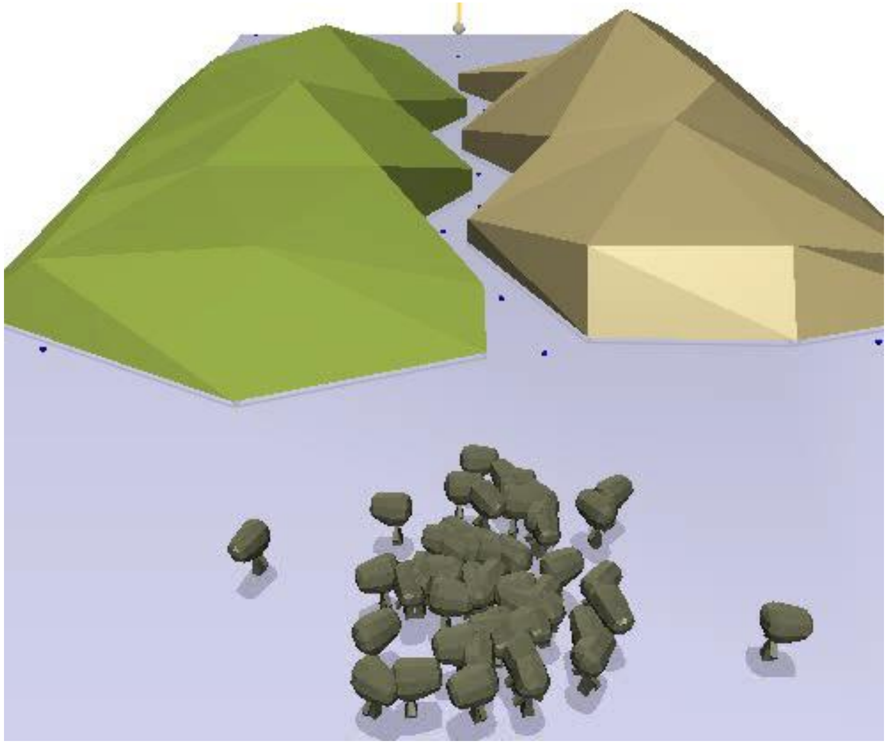


Other Multi-Step Planning Problems

Object manipulation (stances → grasps)



[Yamane, Kuffner and Hodgins]

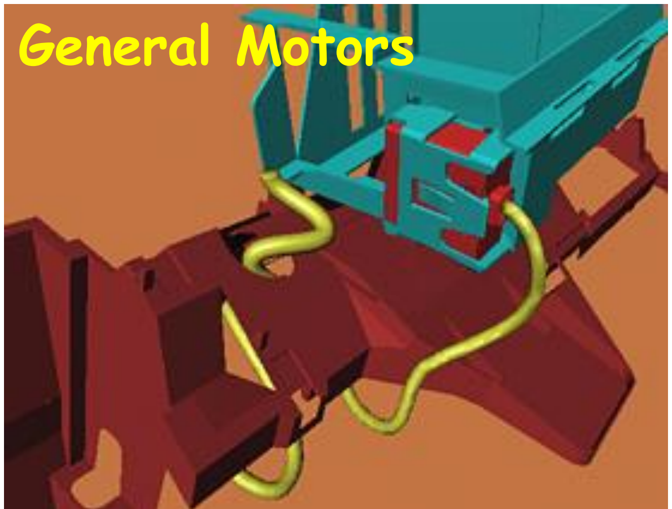


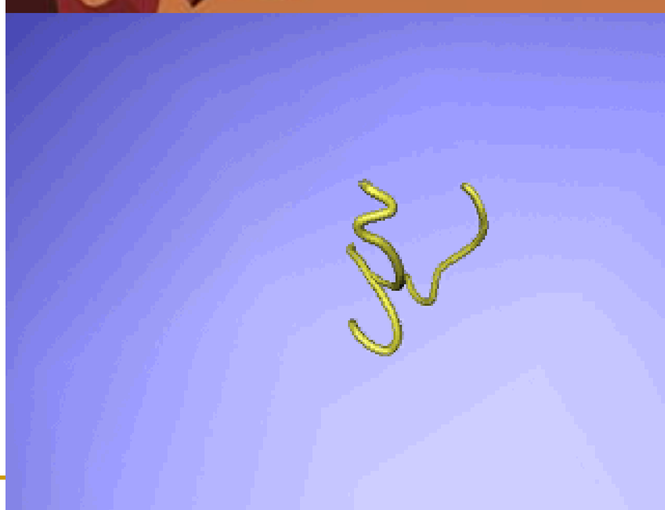
Sample Application of

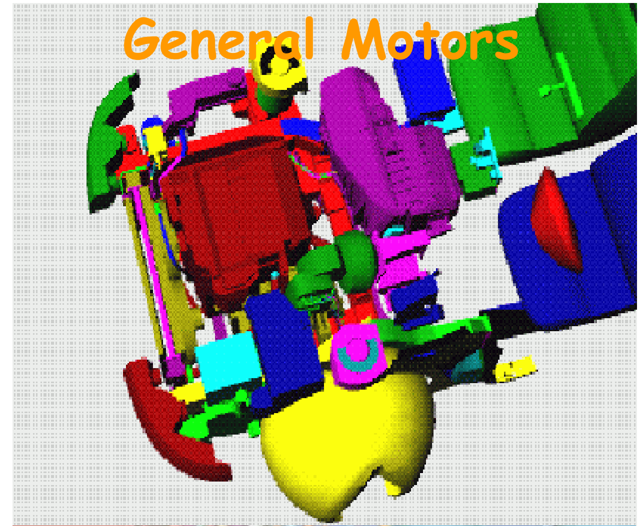
Motion Planning

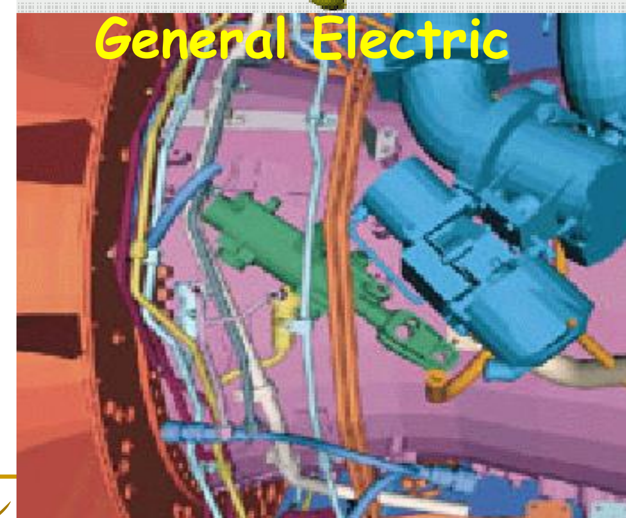
Where we name usages of motion planning in a few exemplary fields

Design for Manufacturing and Servicing

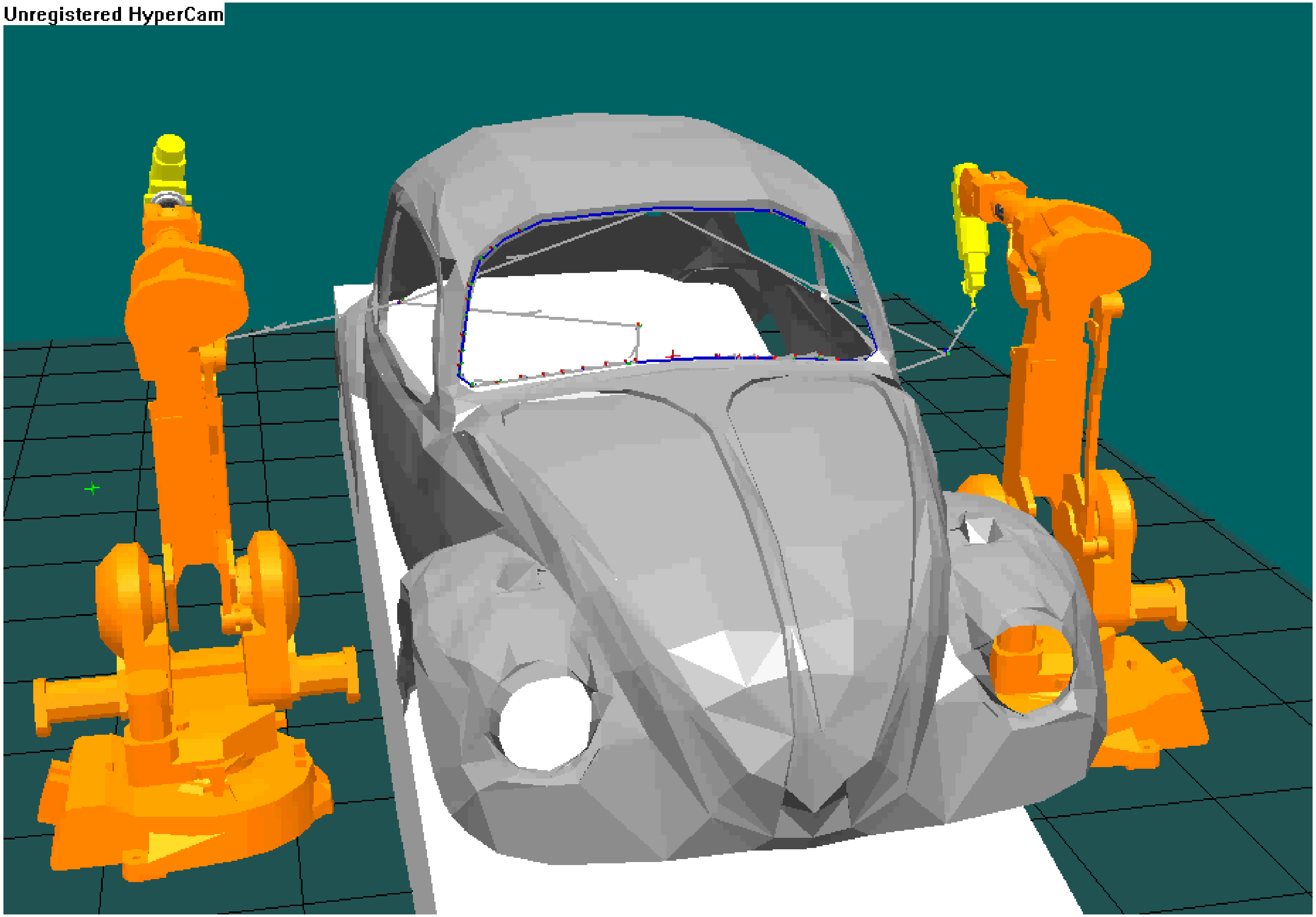








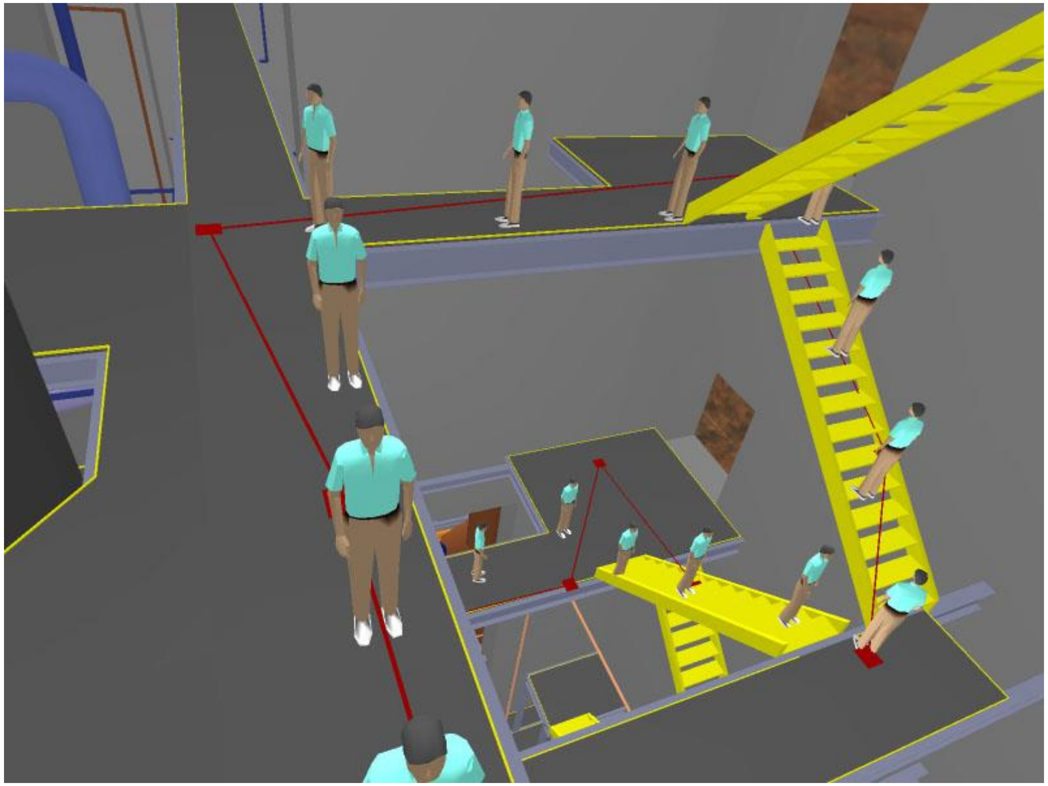
Automatic Robot Programming



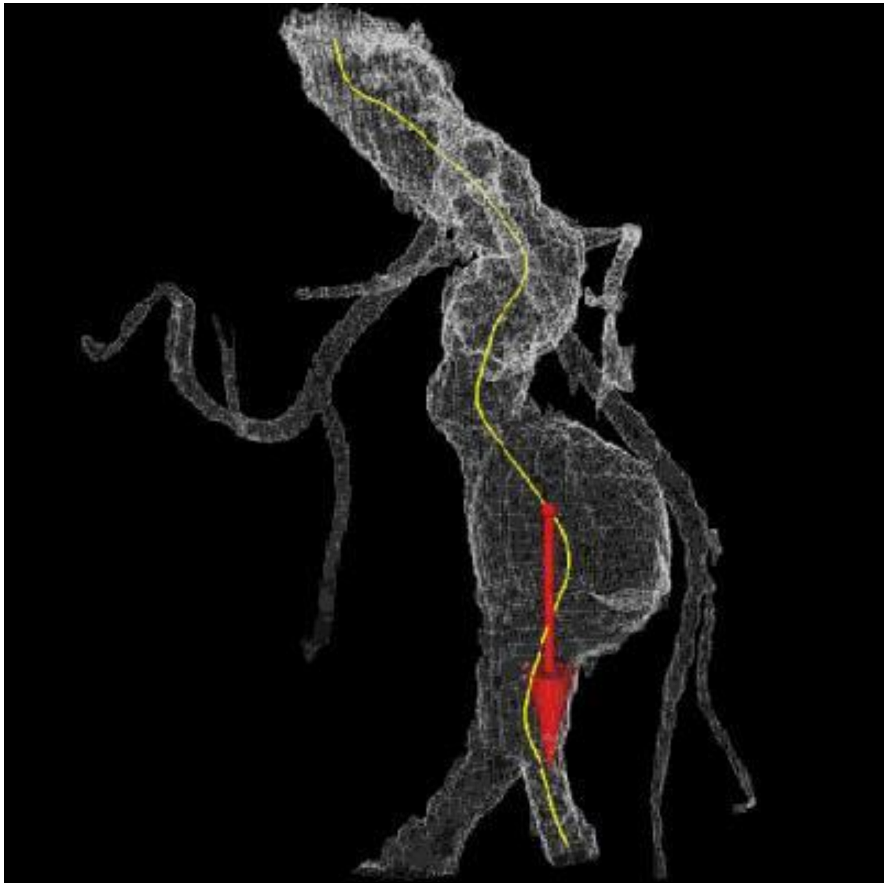
ABB

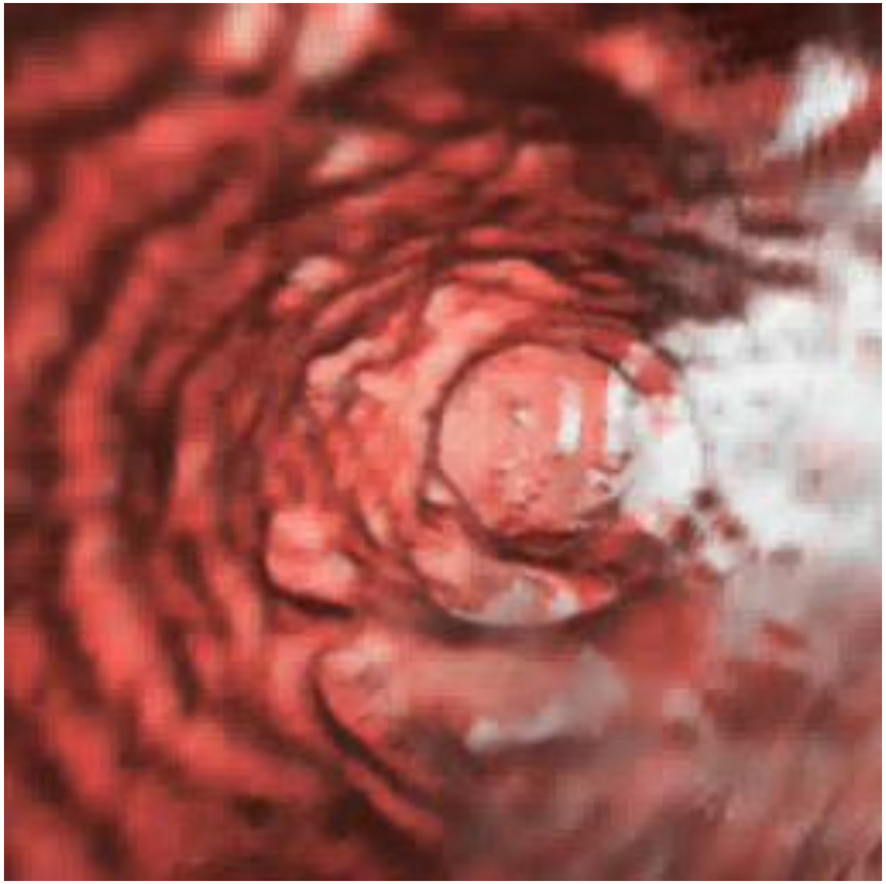
Navigation through Virtual Environments

M. Lin, UNC



Virtual Angiography

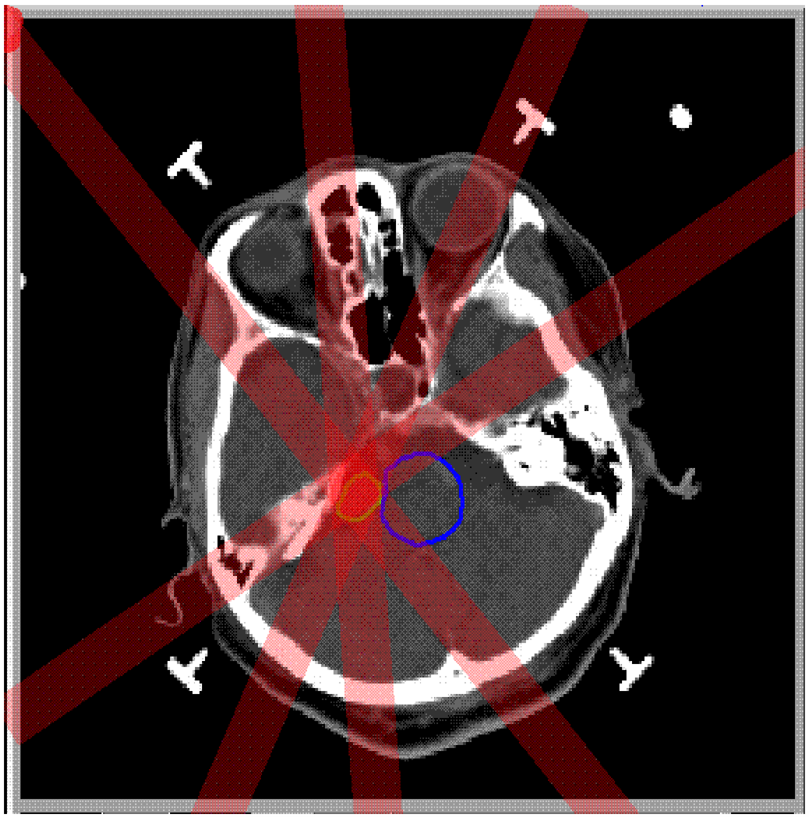


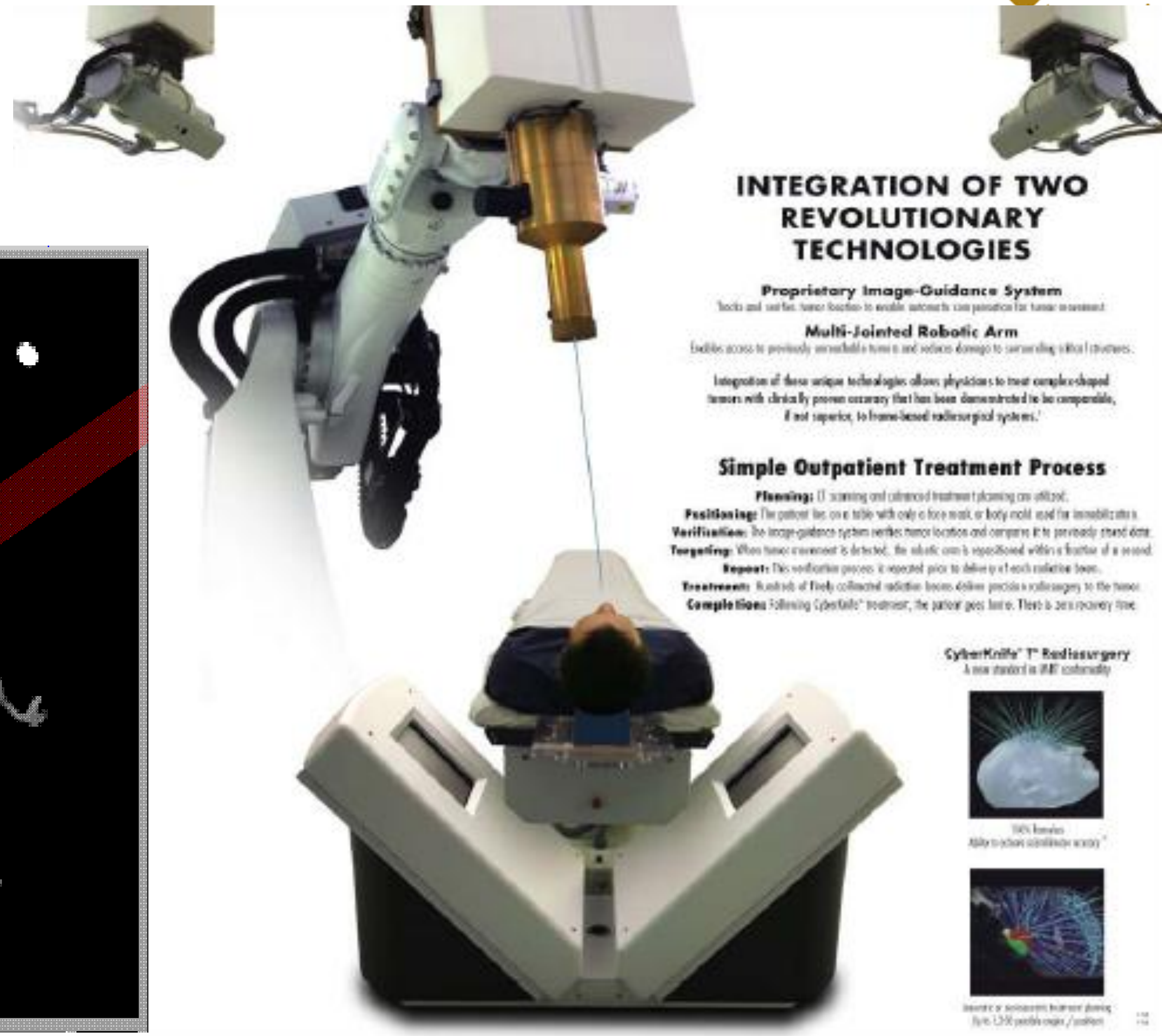


[S. Napel, 3D Medical Imaging Lab. Stanford]

Radiosurgery

CyberKnife (Accuray)

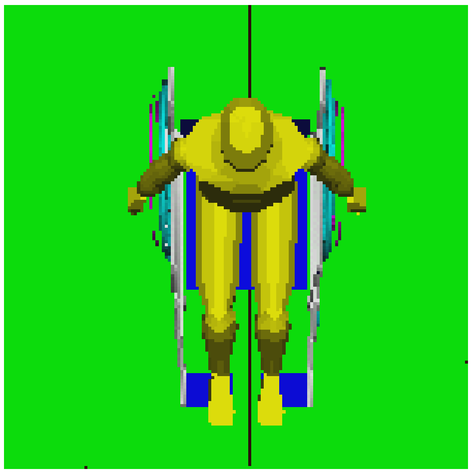


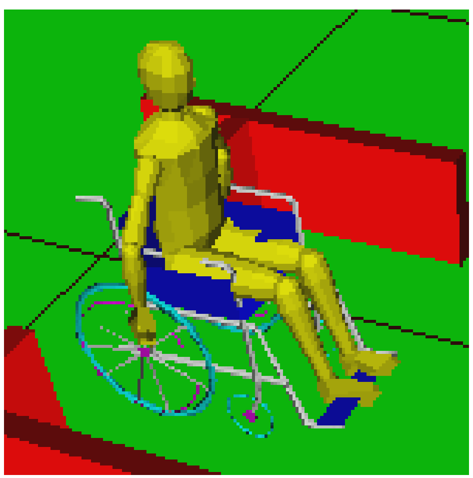


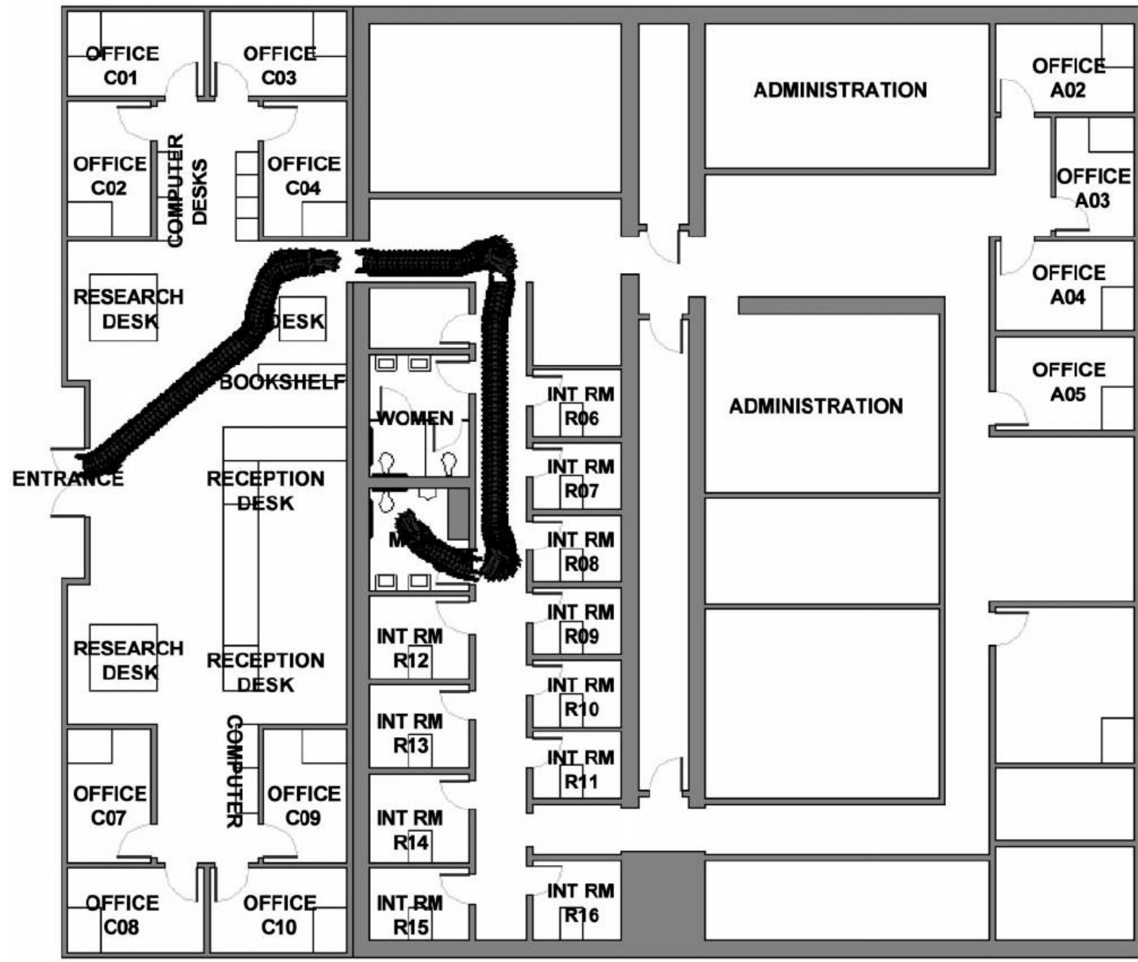
Transportation of A380 Fuselage through Small Villages

Kineo

Architectural Design: Verification of Building Code

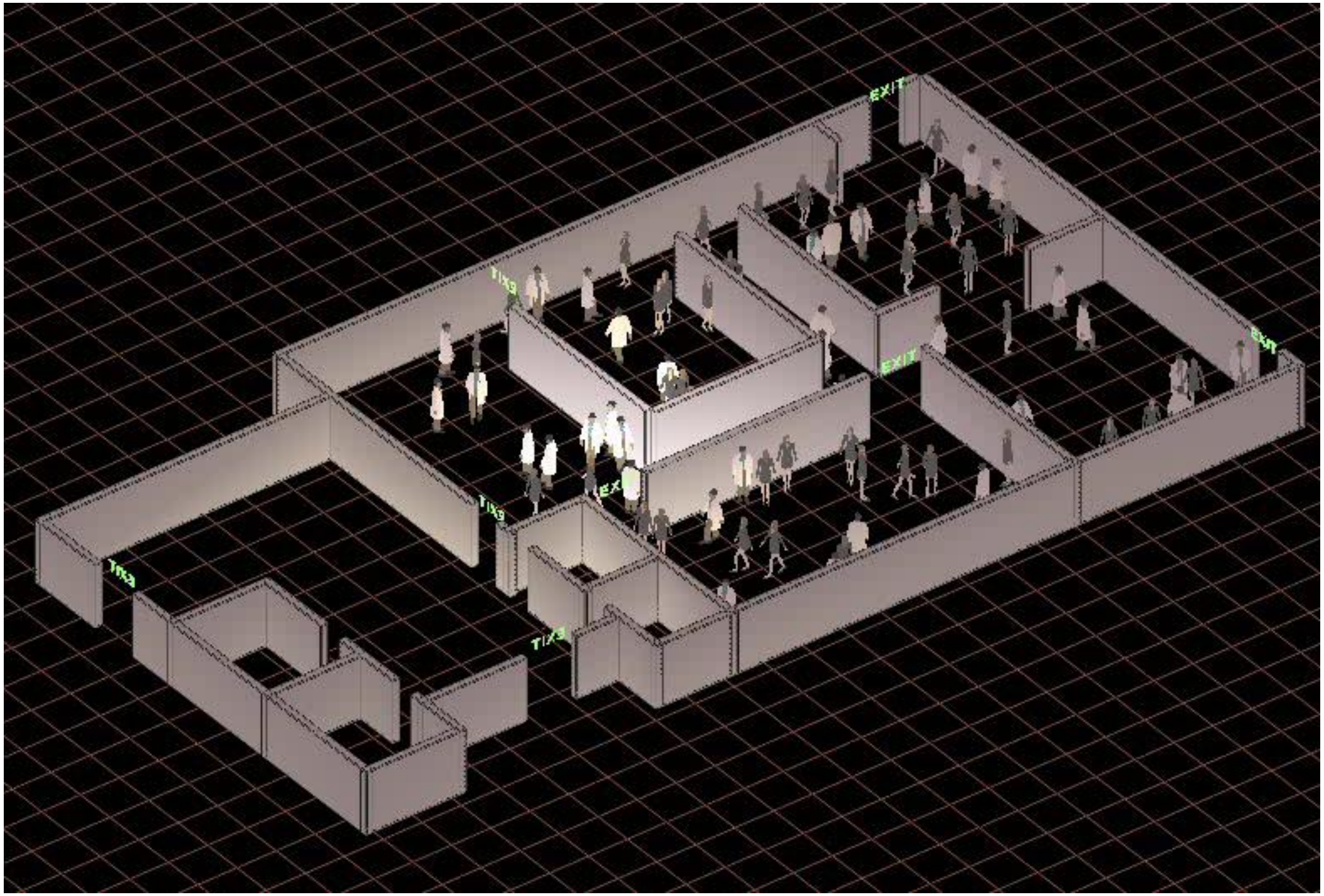




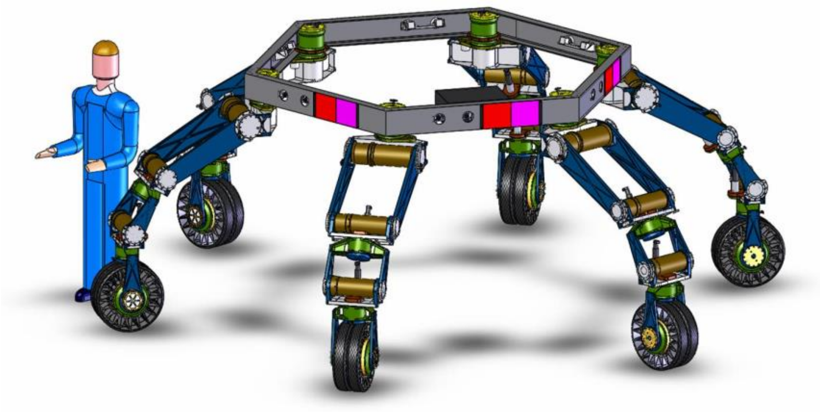


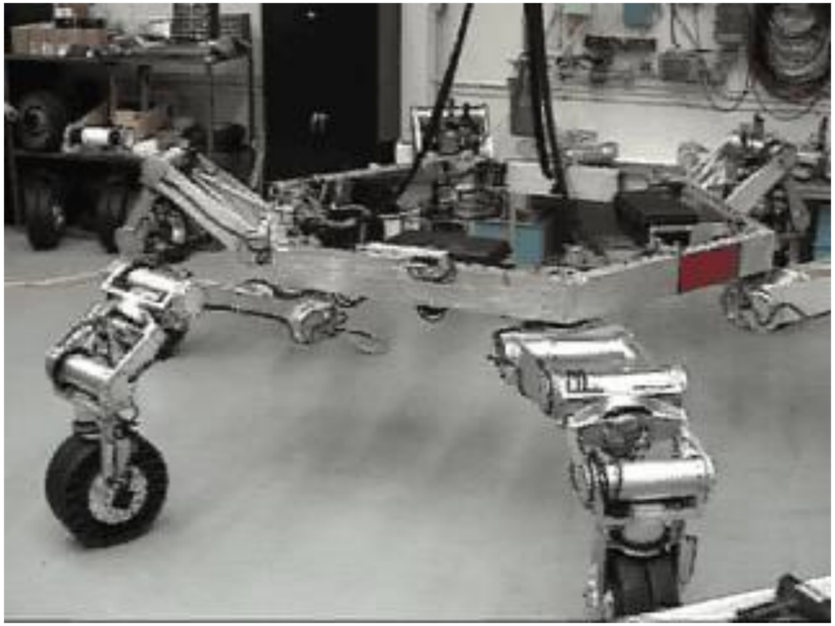
C. Han

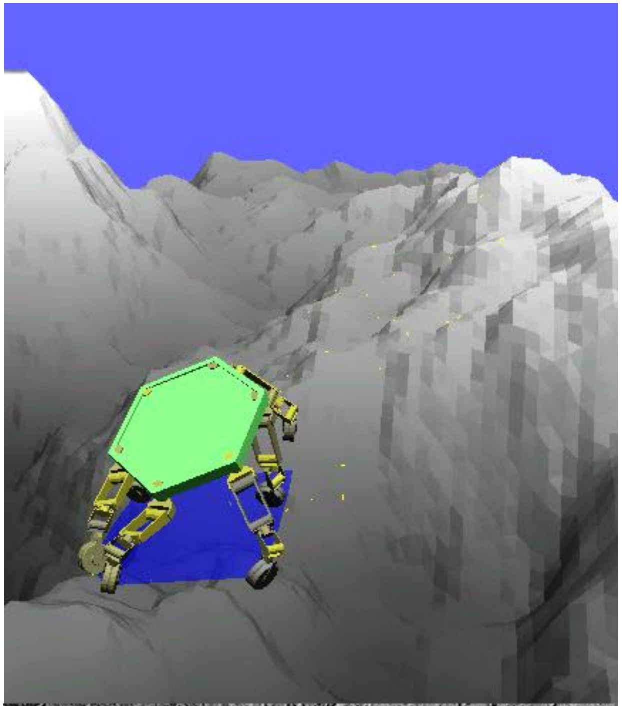
Architectural Design: Egress Analysis

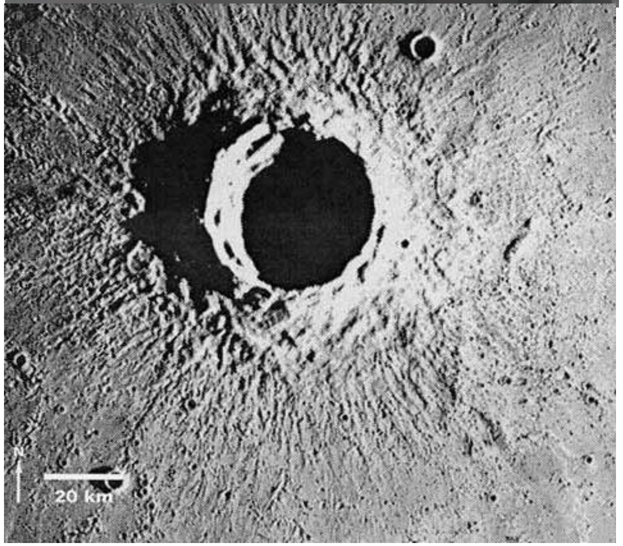


Planet Exploration

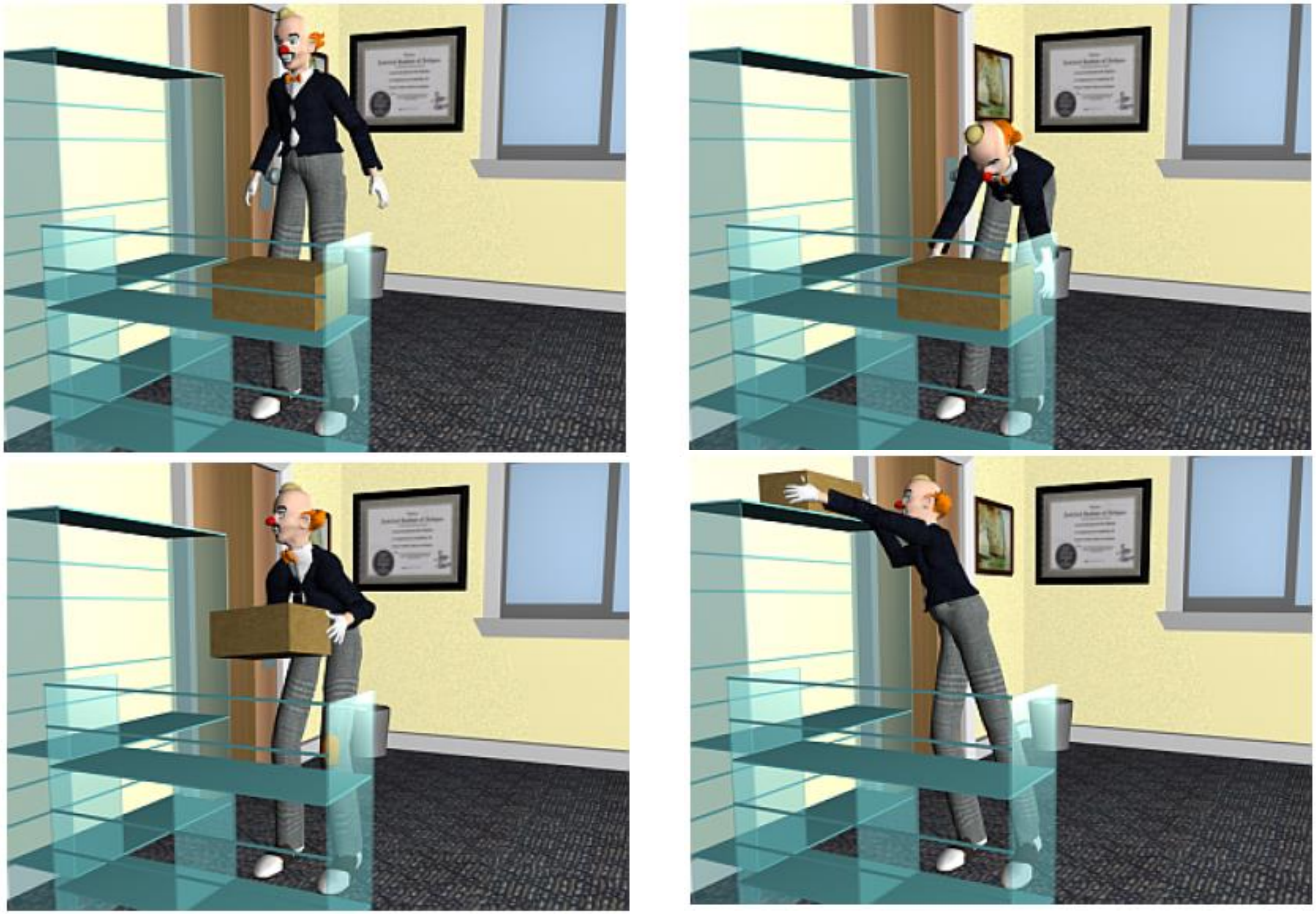








Autonomous Digital Actors



[Yamane, Kuffner and Hodgins]

Animation of Crowds