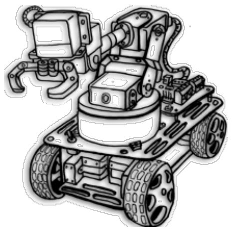
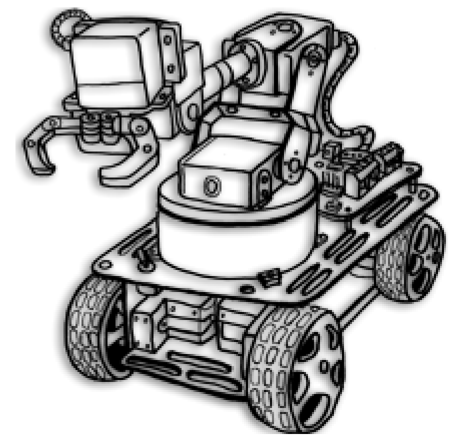


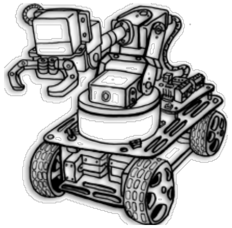
Robotics: Science & Systems
[Introduction]



Prof. Sethu Vijayakumar

Course webpage: http://wcms.inf.ed.ac.uk/ipab/rss

THE UNIVERSITY of EDINBURGH
informatics



Lectures

- **Professor Sethu Vijayakumar**
  - Kinematics, Dynamics, Control, Learning
- **Dr. Zhibin (Alex) Li**
  - Planning, Localisation, Decision Making

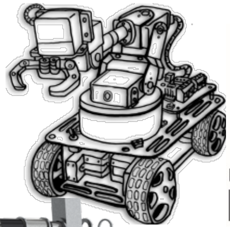
- 09:00-10:50 **[Lecture attendance is essential]**
  - Mondays and Thursday [AT 6.06]

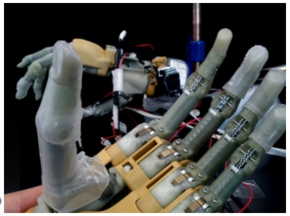
Practicals

- Two Groups
- Teams of 2-3 people
- Group 1 Lab: Monday 11.00 - 13.00
- Group 2 Lab: Thursdays 11.00 - 13.00
- Venue: (AT 3.04 Robotics Lab)
- **Demonstrators:** Dr. Vladimir Ivan & Henrique Ferrolho

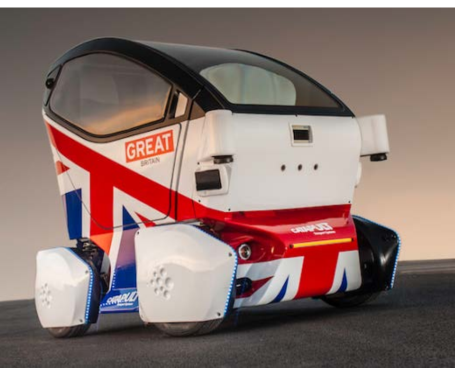
EDINBURGH CENTRE FOR
ROBOTICS

Cutting Edge of Robotics



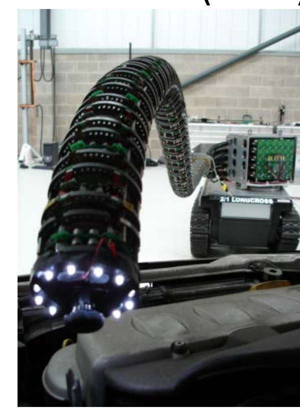


Prosthetics, Exoskeletons



Self Driving Cars

Field Robots (Land)



Nuclear
Decommissioning

Key challenges due to

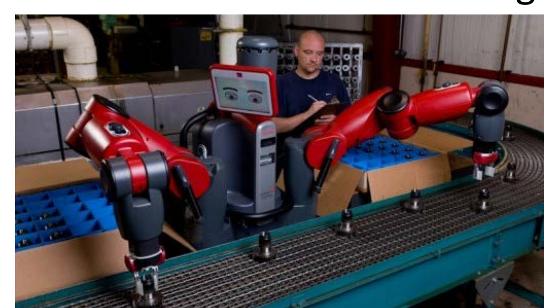
1. Close interaction with multiple objects
2. Multiple contacts
3. Hard to model non-linear dynamics
4. Guarantees for safe operations
5. Highly constrained environment
6. Under significant autonomy
7. Noisy sensing with occlusions

...classical methods do not scale!

Field Robots (Marine)

Medical Robotics

Service Robots



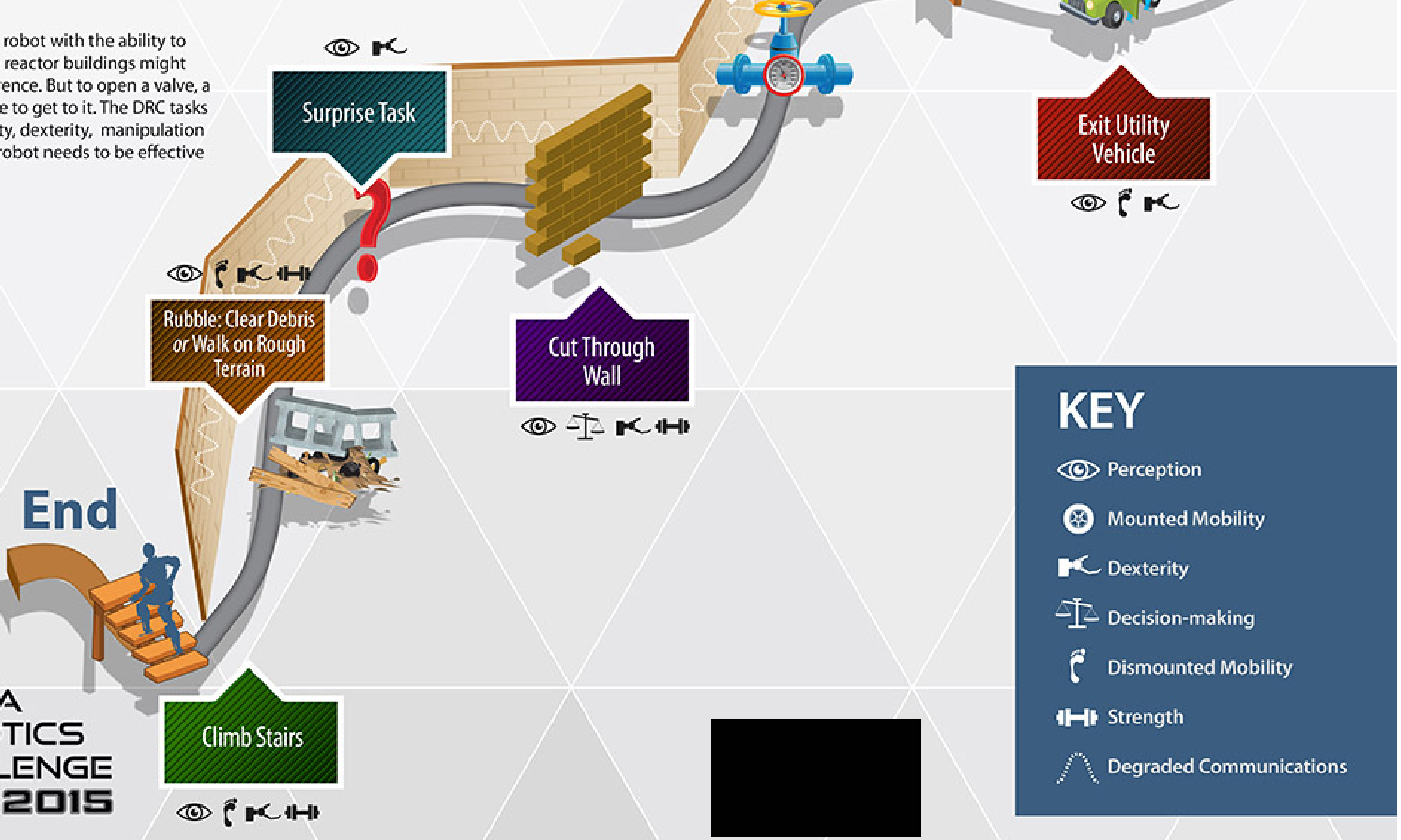
Industrial/ Manufacturing

DARPA Robotics Challenge

WHY THE DARPA ROBOTICS CHALLENGE TASKS?

The story of the DARPA Robotics Challenge (DRC) begins on March 12, 2011, the day after the Tohoku, Japan earthquake and tsunami struck the Fukushima-Daiichi nuclear power plant. On that day, a team of plant workers set out to enter the darkened reactor buildings and manually vent accumulated hydrogen to the atmosphere. Unfortunately, the vent team soon encountered the maximum level of radiation allowed for humans and had to turn back. In the days that followed, with the vents still closed, hydrogen built up in each of three reactor buildings, fueling explosions that extensively damaged the facility, contaminated the environment and drastically complicated stabilization and remediation of the site.

At Fukushima, having a robot with the ability to open valves to vent the reactor buildings might have made all the difference. But to open a valve, a robot first has to be able to get to it. The DRC tasks test some of the mobility, dexterity, manipulation and perception skills a robot needs to be effective in disaster response.

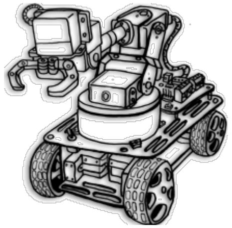


KEY

- Perception
- Mounted Mobility
- Dexterity
- Decision-making
- Dismounted Mobility
- Strength
- Degraded Communications

DARPA ROBOTICS CHALLENGE FINALS 2015

[redacted]



Why Robotics?

- Robotics as a scientific tool for Fundamental Research

  (Machine Intelligence, AI, Computer Science, Computer Vision, Language)

  Why do plants have no *brains*? Because they do not *move* …

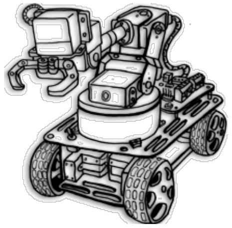
  – motion needs control and decision making

  ↔ Fast information processing

  – motion needs anticipation and planning

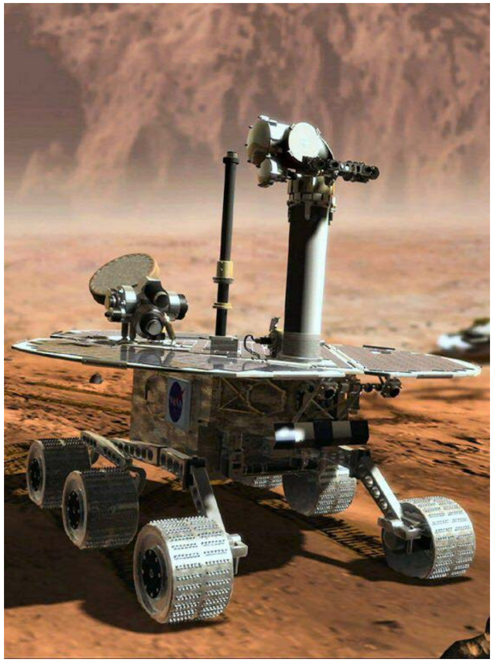
  – motion needs perception

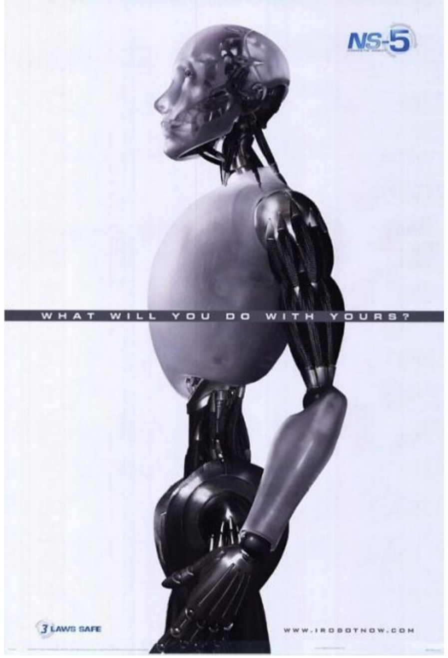
  – motion needs spatial representation

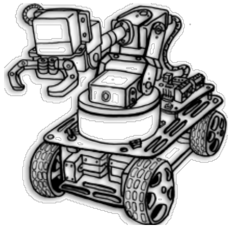


Aim: The Bigger Picture

- Machines that autonomously perform intelligent tasks in the real world







What does ‘autonomous’ mean?

No human in the control loop (automatic – “self-moving”)

Not attached to anything for power or processing (self-contained in operation)

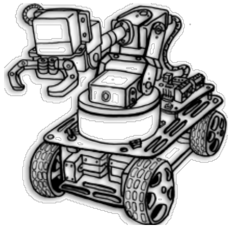
Capable of maintaining behaviour against disturbance (autopilot – “self-regulating” – cybernetic)

Generates own capabilities (self-organising)

Not dependent on human intervention to survive (self-sufficient)

Generates own goals (self-governing - autonomous)

Generates own existence (autopoietic – “self-producing”)



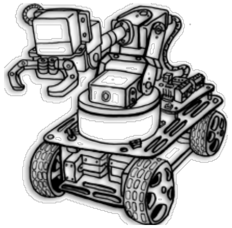
What does 'autonomous' mean?

Crucial aspects of autonomy for this course are:

- The system can achieve a task on its own
- The system is affected by and affects the real world around it *directly*, with no intervention (at least for the duration of its task)

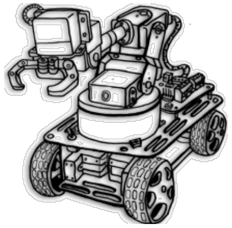
As a consequence we have a closed loop:

- Output affects subsequent input (and task achievement) in ways governed by real world physics (e.g. time, forces, materials…)

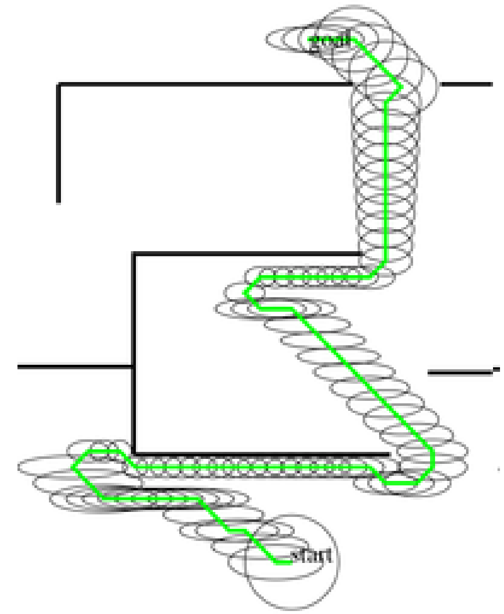


What does ‘intelligent’ mean?

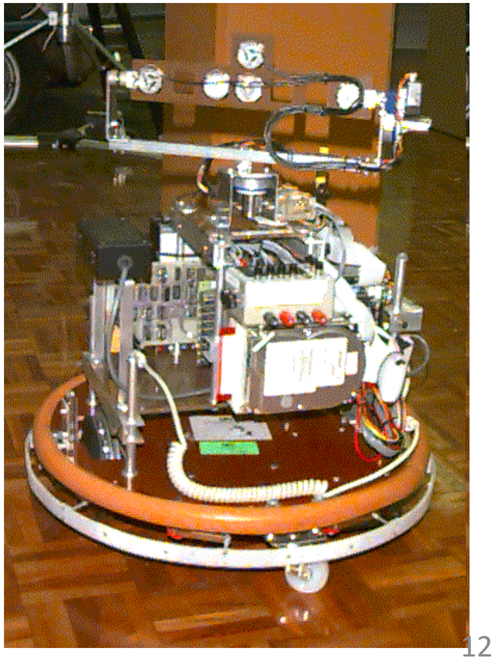
- Can carry out a task that requires more than a pre-programmed sequence, e.g., with decision points depending on the real state of the world
- Adapts to dynamic environments
- Can plan (and re-plan) appropriate actions given high-level goals
- Learns to improve performance from experience



What is hard?

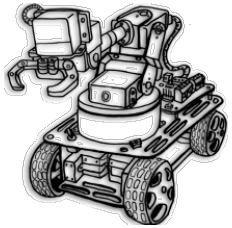


- Intrinsic uncertainty is inherent to robotics
- A robot's knowledge of the problem is limited to what it has been told and what its sensors can tell it
  - Typically high level prior info
  - Typically limited sensor range
- The actual effects of a robot's actions is usually uncertain
  - And the world might change



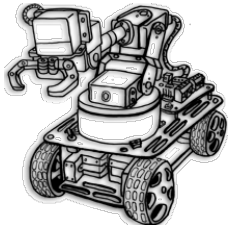
Different Approaches to the Problem

Model-based Principled but brittle	Assume everything is known, or engineer robot or situation so this is approximately true	sense→plan→act
Reactive Robust and cheap but unprincipled	Assume nothing is known, use immediate input for control in multiple tight feedback loops	sense→act sense→act
Hybrid Best and worst of both ?	Plan for ideal world, react to deal with run-time error	plan ⇩ sense→act
Probabilistic Principled, robust but computationally expensive	Explicitly model what is not known	sense→ plan → act with uncertainty



What is this course intended for?

- Give you sufficient exposure to fundamental topics relevant to robotics
  - Planning
  - Dynamics, Kinematics and Control
- Give you hands on (practical) experience in conceptualising a robotic solution to a problem
  - Build a robot (by making design decision)
  - Program it
  - Compete in a real-world environment



Lectures: Four Key Themes

- **Generating Motion**

  goal: orchestrate joint movements for desired endeffector movements

  (kinematics chains, Jacobian, inverse kinematics, multiple tasks, collision, dynamics and control, operational space control, singularities)

- **Planning and Optimization**

  goal: planning around obstacles, optimizing trajectories

  (path finding, sampling based methods, configuration space, RRTs, differential constraints, metrics, trajectory optimization, cost function, Dynamic Prog.)

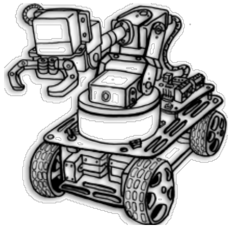
- **Robot Vision**

  goal: identifying, localizing and tracking objects

- **Mobile Robots**

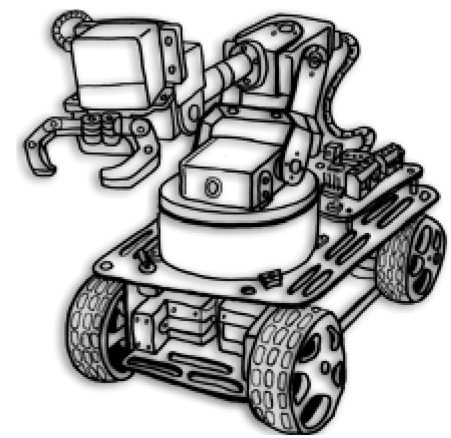
  goal: localize and map yourself; walk, navigate

  (state estimation problem, Bayesian filter, Odometry, Particle & Kalman filter, simultaneous localization and mapping (SLAM)



Robotics: Science & Systems

[Topic 1: 3D Geometry]

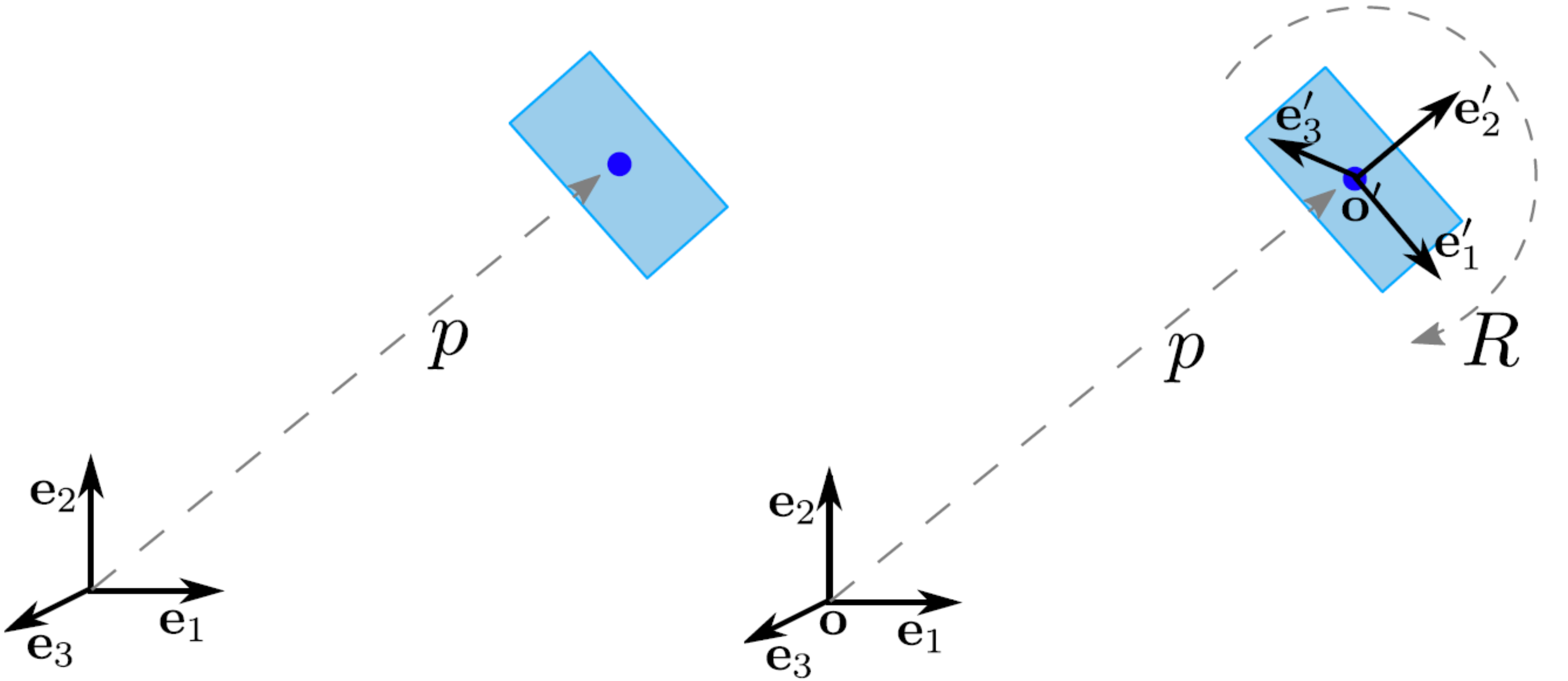


Prof. Sethu Vijayakumar

Course webpage: http://wcms.inf.ed.ac.uk/ipab/rss

THE UNIVERSITY of EDINBURGH informatics

Rigid Body Position & Pose



Position

Pose = Position + Orientation

Rotation Matrices

- Properties

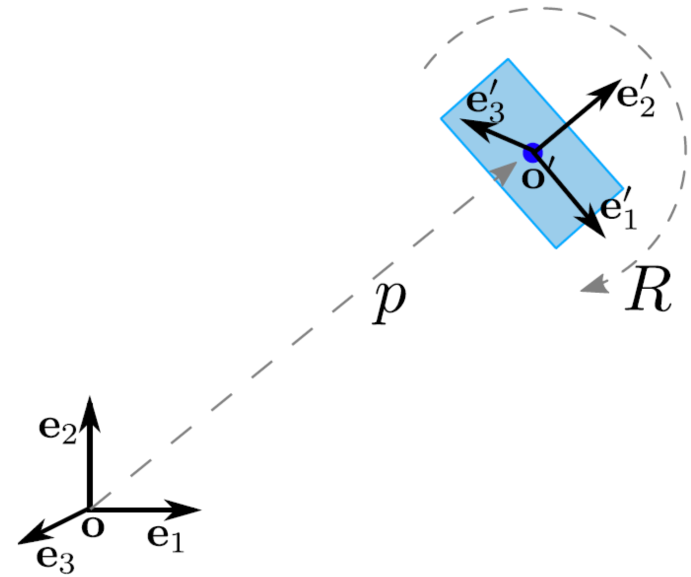
$$R \in \mathbb{R}^{n \times n}$$

orthonormal matrix (orthogonal vectors stay orthogonal, normal vectors stay normal)

$$R^{-1} = R^{\top}$$

columns and rows are orthogonal unit vectors

$$\det(R) = 1$$



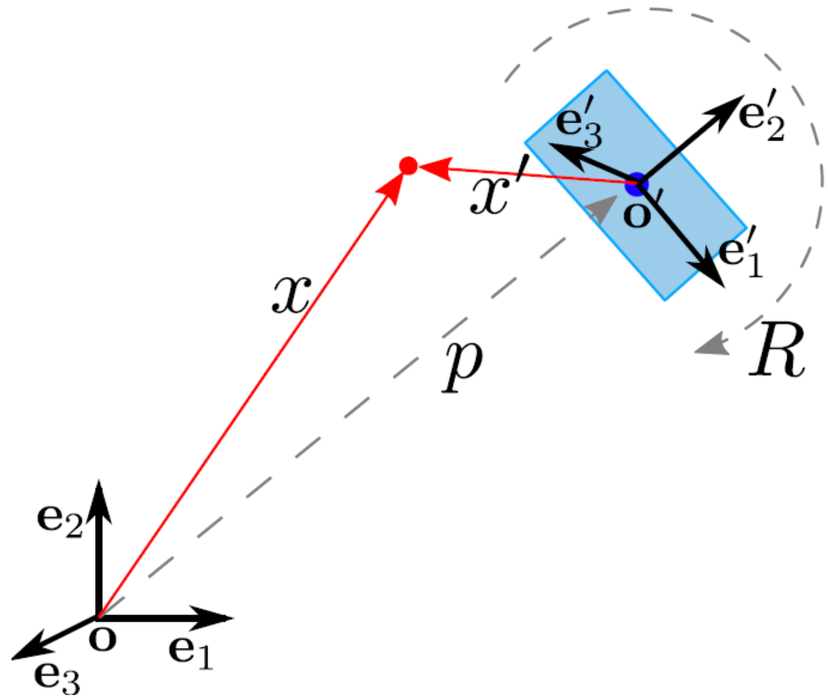
Let the new basis vectors be (for e.g.)

$$e'_1 = R_{11}e_1 + R_{21}e_2 + R_{31}e_3$$

Then, the ***coordinate transformation*** from frame $e'_{1:3}$ to $e_{1:3}$ is:

$$R = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix}$$

Coordinate Transform



x = coordinates of red point in world coordinate frame $(\boldsymbol{o}, \boldsymbol{e}_{1:3})$

x' = coordinates of red point in coordinate frame $(\boldsymbol{o}', \boldsymbol{e}'_{1:3})$

p = coordinates of $\boldsymbol{o}'$ in world coordinate frame $(\boldsymbol{o}, \boldsymbol{e}_{1:3})$

$$x = p + Rx'$$

Simple Rotation Matrices

- 2D

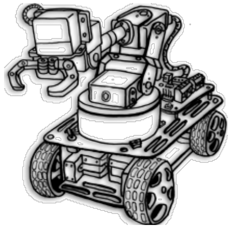
$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

- 3D

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

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

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



Rotation Matrix: Good & Bad

- Pros
  - Rotates vectors directly
  - Easy composition
- Cons
  - 9 numbers
  - Difficult to enforce constraints

Degrees of Freedom (DOF) of a Rotation Matrix

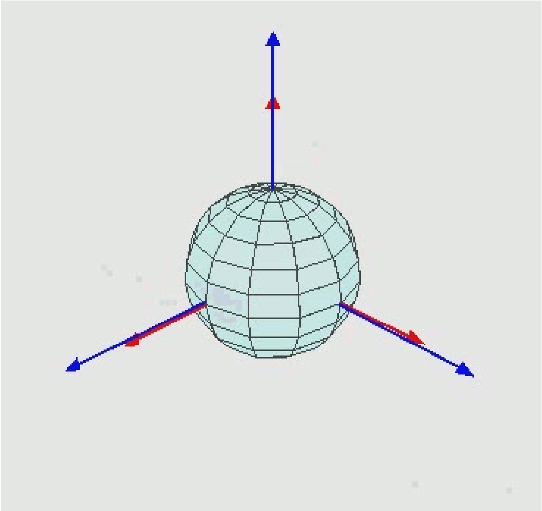
- R^{3x3} has 9 numbers
- 6 constraints (3 orthogonal, 3 normal)
- only 3 DOF

OK…then, can we represent with **minimal** (=3) independent parameters

Rotation: Euler Angles

- Describe rotations by consecutive rotations about different axes:

  "first rotate ϕ about $\hat{z}$, then θ about the *new* $\hat{x}'$, then ψ about the *new-new* $\hat{z}''$"

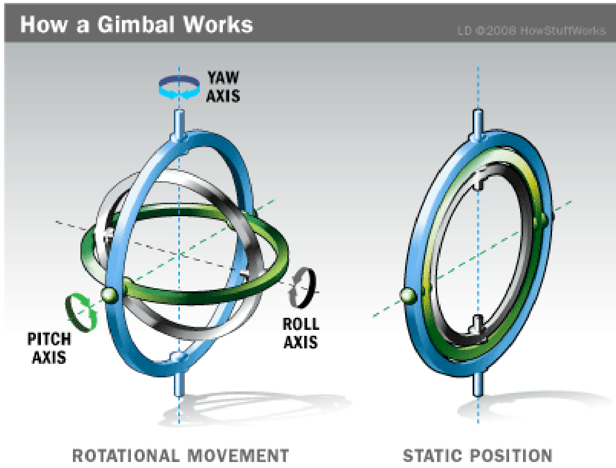


- Z-Y-Z (3-1-3) representation
- yaw-pitch-roll or Z-Y-X (3-2-1)

  ….used in flight!

Euler Angles and Gimbal Lock

- Euler angles have a severe problem:
  - If two axes align: blocks 1 DOF
  - `singularity’ of Euler angles



- Pros
  - minimal representation
  - human readable
- Cons
  - Gimbal lock
  - must convert to matrix to rotate vector
  - no easy composition

Rotation: Rotation Vector

- Using 3 numbers…

vector $w \in \mathbb{R}^3$

length $|w| = \theta$ is rotation angle (in radians)
direction of w = rotation axis ($\underline{w} = w/\theta$)

- Pros
  - minimal representation
  - human readable
- Cons
  - singularity for small rotations
  - must convert to matrix to rotate vector
  - no easy composition

Rotation: Quarternion

- Maths tells: all scheme with 3 numbers will have a singularity

A quaternion is $r \in \mathbb{R}^4$

$$r = \begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{pmatrix} = \begin{pmatrix} r_0 \\ \bar{r} \end{pmatrix}$$

$$r_0 = \cos(\theta/2)$$

$$\bar{r} = \sin(\theta/2)\ \underline{w}$$

with $\underline{w}$ = unit length rotation axis

Unit length constraint (to represent rotations)

$$r^\top r = r_0^2 + r_1^2 + r_2^2 + r_3^2 = 1$$

Quarternion: Composition

- Conversion to/from matrix

$$R(r) = \begin{pmatrix} 1 - r_{22} - r_{33} & r_{12} - r_{03} & r_{13} + r_{02} \\ r_{12} + r_{03} & 1 - r_{11} - r_{33} & r_{23} - r_{01} \\ r_{13} - r_{02} & r_{23} + r_{01} & 1 - r_{11} - r_{22} \end{pmatrix}$$

$$r_{ij} := 2 r_i r_j \; .$$

$$r_0 = \frac{1}{2}\sqrt{1 + \mathrm{tr} R}$$

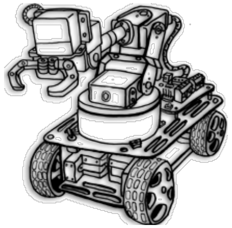
$$r_3 = (R_{21} - R_{12})/(4 r_0)$$

$$r_2 = (R_{13} - R_{31})/(4 r_0)$$

$$r_1 = (R_{32} - R_{23})/(4 r_0)$$

- Composition

$$r \circ r' = \begin{pmatrix} r_0 r'_0 - \bar{r}^\top \bar{r}' \\ r_0 \bar{r}' + r'_0 \bar{r} + \bar{r}' \times \bar{r} \end{pmatrix}$$



Quaternions: Pros and Cons

- Pros
  - no singularity
  - almost minimal representation
  - easy to enforce constraints
  - easy composition
  - easy interpolation
- Cons
  - somewhat confusing
  - not quite minimal
  - must convert to matrix to rotate vector

- **Summary** of Rotation representations
  - need rotation matrix to rotate vectors
  - Quarternions good for free rotations
  - Euler angles OK for small angular deviations
    - but beware singularities!

Homogeneous Transformations

A compact way of representing **coordinate transformations** between two frames

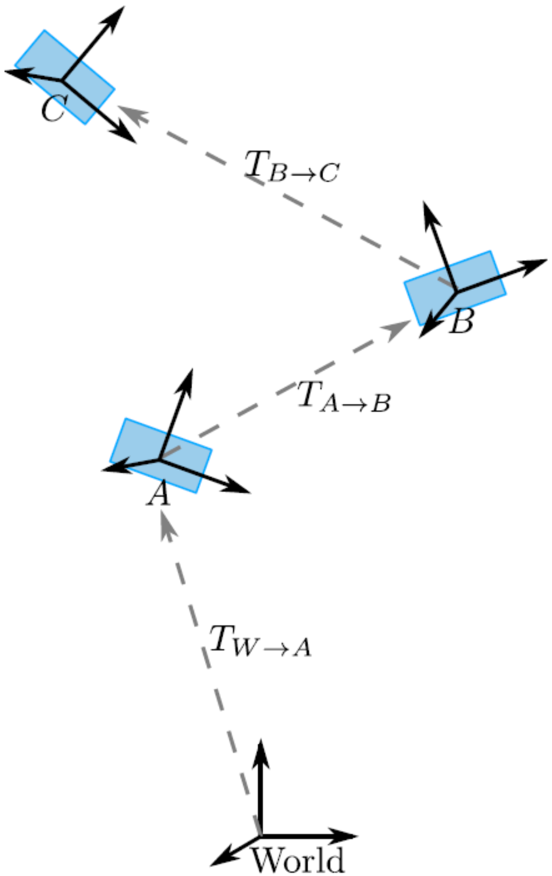
- x^A = coordinates of a point in frame A
  x^B = coordinates of a point in frame B
- Translation and rotation: $x^A = t + Rx^B$
- Homogeneous transform $T \in \mathbb{R}^{4\times 4}$:

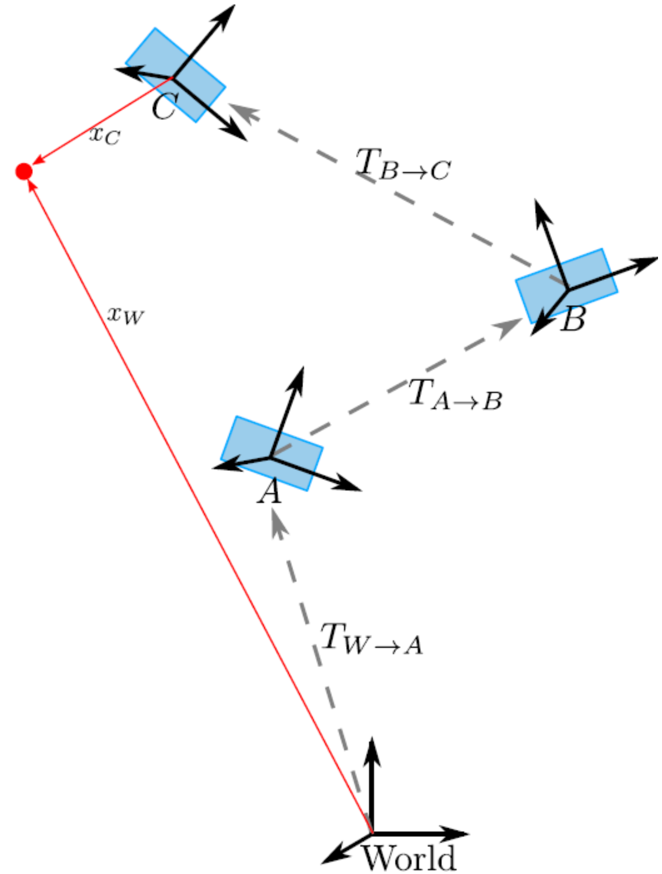
$$T_{A\to B} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$$

$$x^A = T_{A\to B}\, x^B = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x^B \\ 1 \end{pmatrix} = \begin{pmatrix} Rx^B + t \\ 1 \end{pmatrix}$$

in homogeneous transformations, we append 1 to all coordinate vectors

Composition of transforms





$$T_{W\to C} = T_{W\to A}\, T_{A\to B}\, T_{B\to C} \qquad x^W = T_{W\to A}\, T_{A\to B}\, T_{B\to C}\, x^C$$

x^W and x^C are the *coordinates* of the red dot in frames W and C