CS3027/5059: Introduction

N. Oren
n.oren@abdn.ac.uk

University of Aberdeen

Today

- Course practicalities
- Learning Outcomes:
  - What is a robot?
  - Why are robots difficult?
  - How do robots operate — Conceptual control architecture.
  - Robot motion I

Practicalities

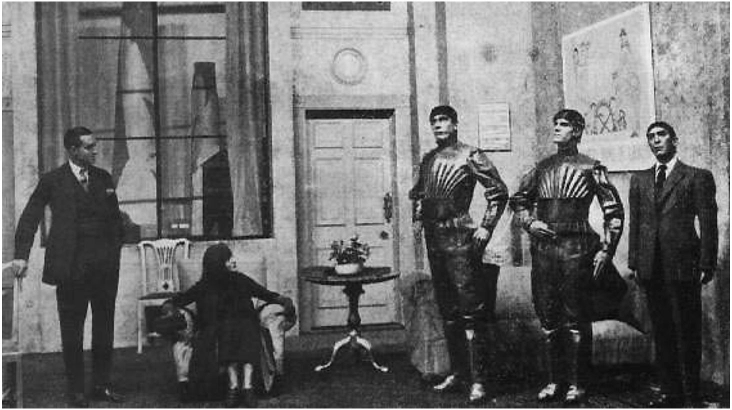
- Lecturer: Nir Oren
- Lectures: Tue-9 Meston 2, Thu-9 Meston 311
- Practicals: Mon 9-11, Tue 3-5 Meston 204
- Marks: 100% coursework (2 assignments).
- You should
    - Be able to program.
    - Not be afraid of maths (we'll mostly use linear algebra, with a bit of calculus thrown in occasionally).

What is a Robot

"The Encyclopedia Galactica defines a robot as a mechanical apparatus designed to do the work of a man. The marketing division of the Sirius Cybernetics Corporation defines a robot as "Your Plastic Pal Who's Fun to Be With. The Hitchhiker's Guide to the Galaxy defines the marketing devision of the Sirius Cybernetic Corporation as "a bunch of mindless jerks who'll be the first against the wall when the revolution comes."

— *Douglas Adams, The Hitchhiker's Guide to the Galaxy.*

Robots



- Term originated in Karel Ĉapek's R.U.R play (1920).

- What's a robot?

Robots

- What's a robot?
  - Moves around
  - Interacts with environment
  - behaves "intelligently" (semi or fully autonomous)

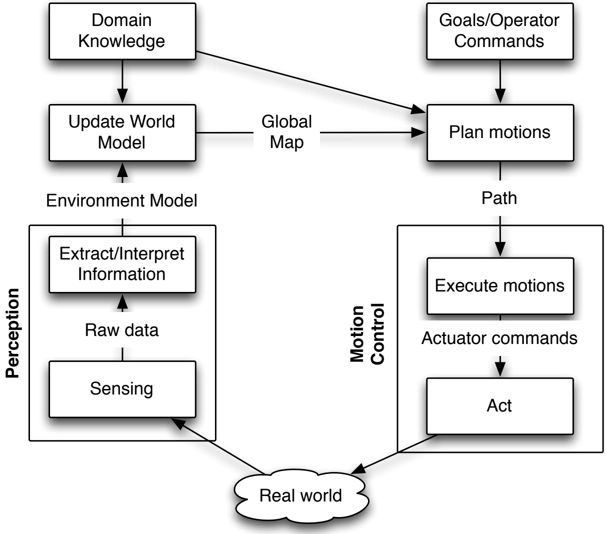
Why do we care?

- http://www.youtube.com/watch?v=u6KW8fIBjr8
- http://www.youtube.com/watch?v=4z-aT8Wp2YQ
- http://www.youtube.com/watch?v=b2bExqhhWRI
- http://www.youtube.com/watch?v=e_lGPRIRG3Y

Why Is Robotics Hard?

How Do We Build A Robot?

What do robots do?



Plan Motions

- Given a set of goals and a view of the domain, we must generate a plan to achieve these goals.
- Both manipulators and mobile robots move through the environment to achieve their goals.
- How should we move through the environment?
- How can we avoid obstacles?
- The resultant path can consist of high level actions (e.g. “close grasp”, “shine laser”, etc.) or positions/trajectories within the environment.

Execute Motions

- Given a path, what instructions should we pass the hardware in order to follow the path?
- The difficulty here is in translating high level paths into low level instructions (e.g. apply a force of 5N for 3 seconds to the left wheel).
- Output is a set of actuator commands which cause the robot to act within the real world.

Sensing

- Robots typically have sensors with which they can sample their environment.
    - Cameras
    - IR/Ultrasonic sensors
    - Noise, radiation, force
    - GPS
- Sensors provide the robot with a set of sensor readings.

Information Extraction

- The raw data obtained from sensor readings must be interpreted.
  - It may be incomplete
  - It could contain noise
- Data from different sensors might need to be fused to obtain a representation of the environment.
- Interpreted sensor readings provide the robot with a snapshot of its environment at that point in time.

World Model

- Information extracted from sensors is noisy, and must be fused to obtain a coherent world model.
- This world model is then used to plan further, select goals for execution, etc., starting the cycle afresh.

Summary of Challenges

- Reasoning/Planning:
  - Given a set of possible goals, which should we pursue?
  - Given a goal, what plan should we use to achieve it?
- Motion Control:
  - How can we navigate robustly?
  - What instructions should we send our actuators to operate in the environment?
- Perception and Update
  - How can we combine different sensor data to obtain an accurate view of the world?

How Do We Solve These Problems?

- Create a formal (mathematical) representation of the system.
- Describe formal techniques that solve the problem.
  - Ideally we can prove that the problem is solved.
  - Or perhaps show some statistically desirable properties of our solution.
  - We seek computationally efficient solutions — formalisation allows us to analyse complexity.
- Implement the solution (formalisation allows us to move between implementations easily).
- Evaluate it.

Practical Difficulties

- No one can solve all these problems, multiple techniques need to be combined.
- This is also a software engineering problem.

ROS: the Robot Operating System

- Open source set of libraries and tools to help build cross-(robotic) platform applications.
- Originally developed at Stanford (2007), then Willow Garage. In 2013 development moved to the Open Source Robotics Foundation.

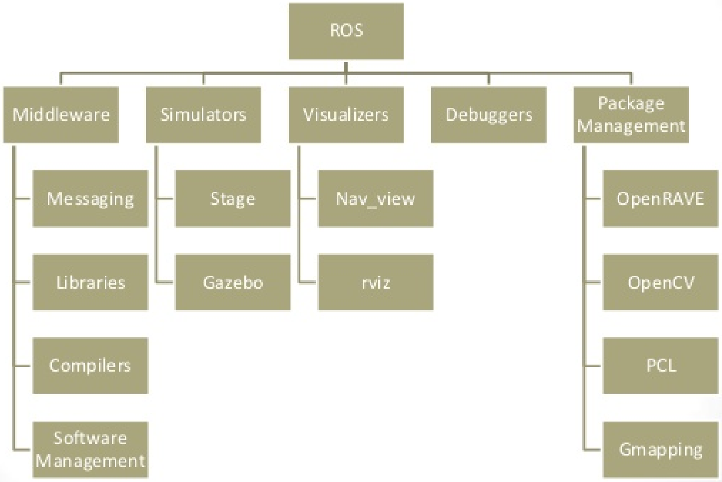
`http://www.ros.org`

- Easiest to run on Ubuntu (14.04 for ROS Indigo, which we'll use).
- Interfaces with Python and C++ (some support for languages such as MATLAB).

ROS

- Two elements: the "operating system"
  - Hardware abstraction
  - low level device control
  - common functionality
  - package management
  - message passing
- User contributed packages (stacks) which implement common functionality (e.g., planning).

Components of ROS



ROS Resources

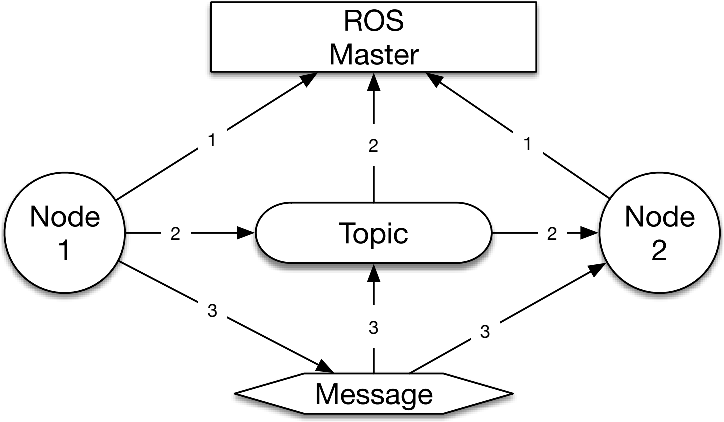
- Tutorials: `http://wiki.ros.org/ROS/Tutorials`
- Tutorial videos: `http://www.youtube.com/playlist?list=PLDC89965A56E6A8D6`
- Cheat sheet: `https://github.com/ros/cheatsheet/releases/download/0.0.1/ROScheatsheet_catkin.pdf`
- A gentle introduction to ROS: `https://cse.sc.edu/~jokane/agitr/`

ROS Preliminaries

- Installing ROS
  - You can either install ROS on your machine (instructions here: `http://wiki.ros.org/indigo/Installation/Ubuntu`);
  - Use a PC in Meston 204; or
  - Install Virtualbox (`https://www.virtualbox.org/`) and use a virtual machine from here: `http://nootrix.com/software/ros-indigo-virtual-machine/`
- You should know/learn
  - How to get around in Linux (see `http://linuxcommand.org`)
  - How to program in C/C++ or Python (I'll use the latter in this course, and therefore recommend it). `http://www.learnpython.org/`

ROS Basic Concepts

- Nodes are (running) single purpose programs. They communicate via
- Topics which are channels. Nodes publish
- Messages to topics, and receive messages by subscribing to topics.
- Services allow for synchronous communication.
- Nodes find each other by using the ROS Master, which also provides a
- Parameter Server — a dictionary accessible via network APIs.
- Packages contain one or more nodes (and message and service specifications) and provide a ROS interface.
- Packages in turn are grouped into Stacks.



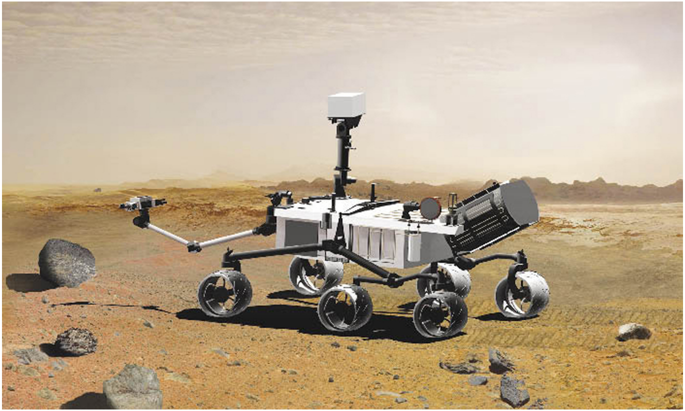
1: Nodes register with master
2: Node 1 registers as a *publisher* with master on a topic
Node 2 registers as *listener* with master on a topic
3: Node 1 *publishes* a message to the topic
Node 2 is informed that a message has been sent on the topic

NB: multiple publishers and subscribers can exist for a topic

Types of Robots

Robot Arms/Manipulators

Types of Robots



Mobile Robots

Movement

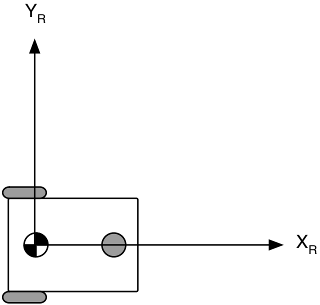
- There are many different ways of moving
  - Joints: rotate, move in/out
  - Legs: walk, run, jump, skip
  - Body: crawl, slither
  - Wings: fly, glide
  - Wheels: roll
- Fundamentally, we can apply a force to a motor or joint, and it reacts, moving the robot.
- We will focus on wheeled locomotion.
- Core question: <u>If we apply a force to wheel(s), where will we end up?</u>
- We need a <u>kinematic model</u>

Kinematics

- Kinematics is the study of motion, without consideration of the causes of motion.
- Important in both mobile robotics and manipulator robotics.
- We adopt a bottom up approach
  - Describe contribution of a wheel to motion
  - Consider configuration of wheels on robot body to affect, and constrain, behaviour of entire robot.
- Ultimate goal: to work out a robot's motion due to wheel's rolling.
- How do we describe this motion?

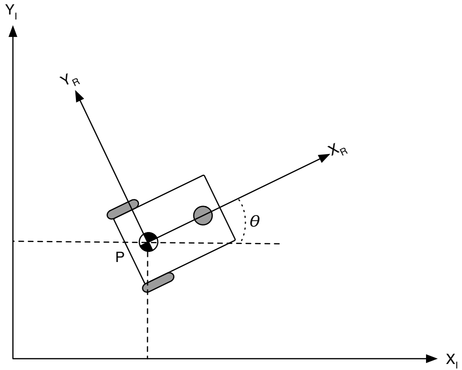
Frames of Reference

- We need a coordinate system to describe a robot's position.
- The robot has a local coordinate system (also called a reference frame)



- Useful for reporting from sensors etc, but not for describing robot position. For this we use a global reference frame.
- Multiple frames of reference are possible, the ROS TF package can be used to relate them to each other.
- How many parameters do we need to describe the robot in the global frame?

The Global Frame



- This robot chassis has 3 degrees of freedom — 2 for position in the plane, 1 for orientation along the axis.
- We ignore internal degrees of freedom (e.g. due to wheel rotation, steering joint angles etc).

Intro

- https://www.youtube.com/watch?v=uWLjgs2CEyE

Useful links

- http://www.khanacademy.org/math/linear-algebra

Position

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

- The vector ξ_I specifies the pose of the robot in the I reference frame.
- x, y specify the position of P, θ the angular difference between the I and R reference frames.
- Individual component motion is defined in terms of the I frame, and we will therefore need to map between I and R.
- Such a mapping is a function of the current angle of the robot, and depends on the orthogonal rotation matrix

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

Rotation Matrix

- We use R to map between the global reference frame and local frame: $\xi_R = R(\theta)\xi_I$.
- Consider $\theta = \pi/2$, and a velocity $(\dot{x}, \dot{y}, \dot{\theta})$

$$\dot{\xi_R} = R(\pi/2)\dot{\xi_I} = R(\pi/2) = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{y} \\ -\dot{x} \\ \dot{\theta} \end{bmatrix}$$

- The robot knows (in its local frame) where it will be given its velocity in the global frame.
- Why do we care?

Forward Kinematics

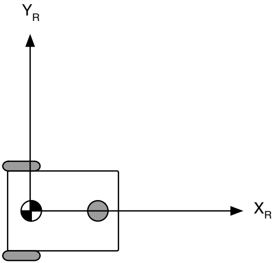
- Robot spins left wheel for 5 revolutions, right wheel for 3 revolutions at same time. Then accelerates right wheel to left wheel speed, taking 2 revolutions to get up to the same speed. Where is the robot?
- Relatively easy for a differential drive robot, more involved for other robot types.

Differential Drive Robot

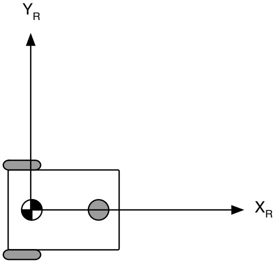
$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(l, r, \theta, \dot{\varphi}_1, \dot{\varphi}_2)$$

- l wheel distance from a point P centred between the two wheels.
- r wheel diameter
- $\dot{\varphi}_x$ spinning speed of wheel x
- These parameters allow us to compute the robot's motion in the global frame from its local frame: $\dot{\xi}_I = R(\theta)^{-1}\dot{\xi}_R$
- We compute the contribution of each wheel's spinning speed.

Differential Drive Robot

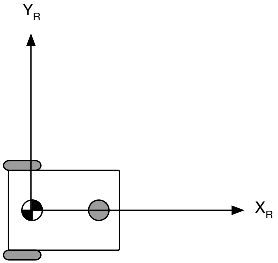


- If the top wheel spins and the bottom is stationary, how will P move with respect to X_R?



- If the top wheel spins and the bottom is stationary, how will P move with respect to X_R?
- It will move at half the speed $\dot{x}_{r1} = (1/2)r\dot{\varphi}_1$.
- Same for the second wheel $\dot{x}_{r2} = (1/2)r\dot{\varphi}_2$.
- If top wheel moves forward, second backwards, P is stationary.
- Total X contribution is simply $\dot{x}_{r1} + \dot{x}_{r2}$

Differential Drive Robot



- The robot's wheels do not allow the robot to move sideways, $\dot{y_R} = 0$
- For $\dot{\theta_R}$ remember that if the "bottom" wheel spins forward, the robot will spin counter-clockwise.
- If the "top" wheel is stationary, the robot will then spin around it.

$$\omega_1 = \frac{r\dot{\varphi}_1}{2l} \qquad \omega_2 = -\frac{r\dot{\varphi}_2}{2l}$$

- We can simply sum these contributions.

Differential Drive Robot

$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} \frac{r\dot{\varphi}_1 + r\dot{\varphi}_2}{2} \\ 0 \\ \frac{r\dot{\varphi}_1 - r\dot{\varphi}_2}{2l} \end{bmatrix}$$

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

Example

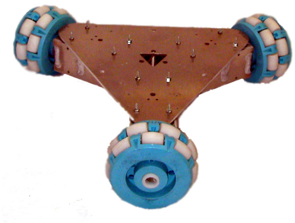
- Consider robot with $r = 1, l = 1$, positioned at $\theta = \pi/2$.
- Robot engages wheel 1 with speed 4, wheel 2 with speed 2.
- What is velocity in the global reference frame?

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix}$$

- Robot will move (instantaneously) along the y axis with speed 3 while rotating with speed 1.

Where are we?

- We can describe the motion of a differential drive robot by examining the contribution of motion of the chassis due to each of its wheels.
- Can we say more? What motion can a robot make/not make?
- What about the motion of more complex robots?
  - Bicycles
  - "Cars"
  - Omniwheels



Complex Robots

http://www.youtube.com/watch?v=mNy09kuIdzs&feature=related

http://www.youtube.com/watch?v=E7X0_6o9J10

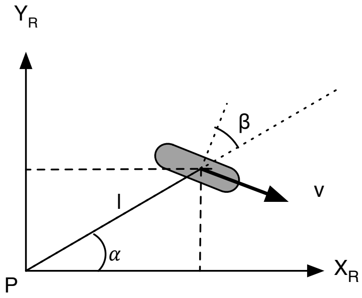
http://www.youtube.com/watch?v=-cZv5oKABPQ&feature=fvwrel

- Basic approach:
  - Express constraints on how individual wheels can move.
  - Combine motion of wheels to describe robot movement.
- Assumptions:
  - Wheels always remain vertical.
  - Wheels never slip or slide.
  - Assume there is a single point of contact between wheel and ground.

Constraints

- Given our assumption, we can identify two constraints on wheel motion.
    - The wheel must roll when motion takes place.
    - The wheel does not slide orthogonally to the plane.
- We will consider several wheel types:
    - Fixed standard wheel
    - Steered standard wheel
    - Castor wheel
    - Swedish wheel
    - Spherical wheel

Fixed Wheel



- Using polar coordinates, the wheel is at position l, α from the origin of the robot's local frame.
- The angle of the wheel, relative to the robot chassis, is β
- If the wheel has a radius r, it's rotational position around the horizontal axis is a function of time, $\varphi(t)$.

Rolling Constraint

- All motion along the direction of the wheel plane must ensure that rolling takes place.

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

- Note the transformation to the robot reference frame which is needed as $\alpha, \beta, \ldots$ are specified in this frame.
- Sliding constraint means that motion orthogonal to the wheel plane is zero

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

- For a steered wheel, the only difference is that β is a function rather than a constant.
- However, instantaneously, this makes no difference.
- Other wheel types (e.g. castor wheels) have slightly different constraints.

Example

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

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

- If $\alpha = \beta = 0$ then, if $\theta = 0$

Example

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

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

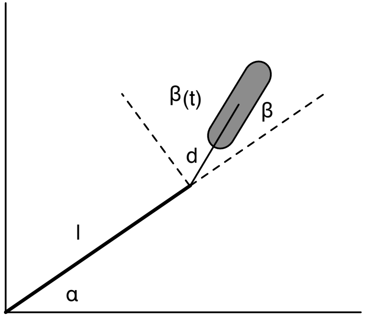
- If $\alpha = \beta = 0$ then, if $\theta = 0$ the sliding constraint reduces to

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

- For a castor wheel, the force orthogonal to the wheel causes the castor to turn.

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

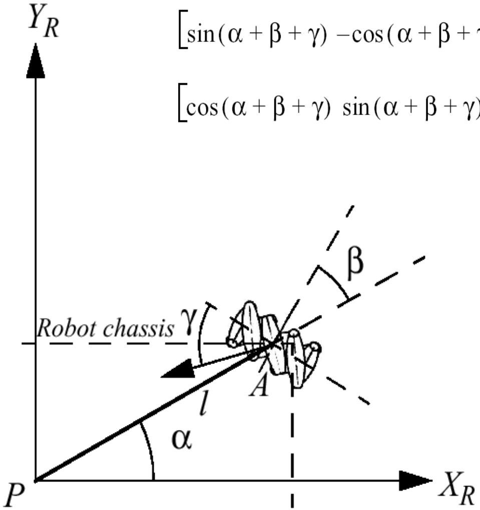
$$\left[\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad d+l\sin(\beta)\right] R(\theta)\dot{(\xi_I)} + d\dot{\beta} = 0$$

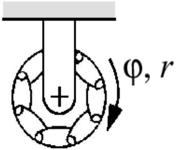


Swedish Wheel

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

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

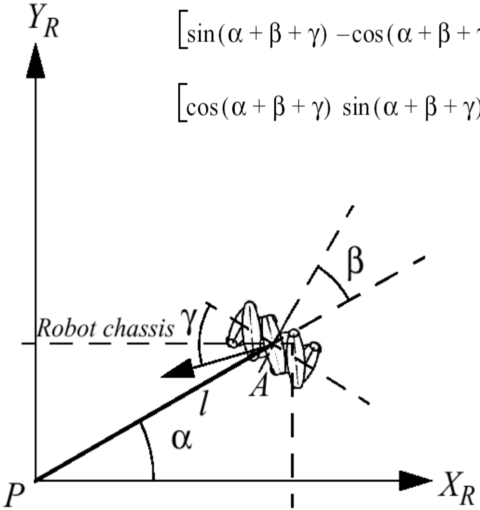


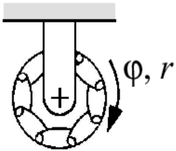


Spherical Wheel

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

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





Moving up to the robot

- Castor wheels, swedish wheels and spherical wheels impose no kinematic constraints on the robot chassis - $\dot{\xi}_I$ can range freely due to internal wheel degrees of freedom.
- Only fixed and steerable wheels have an impact on robot chassis kinematics.
- Assume robot has N standard wheels - N_f fixed and N_s steerable.
- $\beta_s(t)$ is the steering angle of the steerable wheel at time t.
- β_f is the orientation of the fixed wheels.
- $\varphi(t)$ represents the rotational position of the wheels.

Moving up to the robot

- We can represent all the wheel constraints in a matrix
- Rolling:

$$J_1(\beta_s)R(\theta)\dot{\xi}_I + J_2\dot{\varphi} = 0 \qquad \varphi(t) = \begin{bmatrix} \varphi_f(t) \\ \varphi_s(t) \end{bmatrix} \qquad J_1(\beta_s) = \begin{bmatrix} J_{1f} \\ J_{1s}(\beta_s) \end{bmatrix}$$

$$J_2 = diag(r_1 \dots r_N)$$

  - $\varphi(t)$ is of size $(N_f + N_s) \times 1$
  - $J_1(\beta_s)$ is of size $(N_f + N_s) \times 3$
  - This represents the constraint that the wheels must roll an appropriate amount.
- Sliding:

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

  - C_{1f} is of size $N_f \times 3$, C_{1s} is of size $N_s \times 3$
  - Captures the standard wheels prohibition on motion orthogonal to the wheel plane.

Example

- Combining the rolling and sliding constraints yields

$$\begin{bmatrix} J_1(\beta_s) \\ C_1(\beta_s) \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} J_2\varphi \\ 0 \end{bmatrix}$$

- For a differential drive robot we ignore the castor/spherical wheel.
- Remaining wheels are not steerable, so $J_1(\beta_s) = J_{1f}$ and $C_1(\beta_s) = C_{1f}$.

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

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

- What are α and β?

Example

- Combining the rolling and sliding constraints yields

$$\begin{bmatrix} J_1(\beta_s) \\ C_1(\beta_s) \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} J_2\varphi \\ 0 \end{bmatrix}$$

- For a differential drive robot we ignore the castor/spherical wheel.
- Remaining wheels are not steerable, so $J_1(\beta_s) = J_{1f}$ and $C_1(\beta_s) = C_{1f}$.

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

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

- What are α and β?
- $\alpha = -\pi/2$, $\beta = \pi$ for one wheel, $\alpha = \pi/2$, $\beta = 0$ for the other (note positive spin).

Example

$$\begin{bmatrix} \sin(\pi/2) & -\cos(\pi/2) & (-l)\cos(\pi) \\ \sin(\pi/2) & -\cos(\pi/2) & (-l)\cos(0) \\ \cos(\pi/2) & \sin(\pi/2) & l\sin(\pi) \\ \cos(\pi/2) & \sin(\pi/2) & l\sin(0) \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ 0 \\ 0 \end{bmatrix}$$

Example

$$\begin{bmatrix} \sin(\pi/2) & -\cos(\pi/2) & (-l)\cos(\pi) \\ \sin(\pi/2) & -\cos(\pi/2) & (-l)\cos(0) \\ \cos(\pi/2) & \sin(\pi/2) & l\sin(\pi) \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ 0 \end{bmatrix}$$

Example

$$\begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \\ 0 & 1 & 0 \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ 0 \end{bmatrix}$$

Example

Inverting we get

$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \\ 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ 0 \end{bmatrix}$$

Example

$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} 1/2 & 1/2 & 0 \\ 0 & 0 & 1 \\ 1/2l & -1/2l & 0 \end{bmatrix} \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ 0 \end{bmatrix}$$

Example

$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} \frac{r\dot{\varphi}_1 + r\dot{\varphi}_2}{2} \\ 0 \\ \frac{r\dot{\varphi}_1 - r\dot{\varphi}_2}{2l} \end{bmatrix}$$

Video

http://www.youtube.com/watch?v=Bd5iEke6UlE

http://www.youtube.com/watch?v=lWsMdN7HMuA

Where are we?

- We can describe the instantaneous velocity of a robot based on the angular velocity (and angle) of each of its wheels.
- So we can work out how a robot will move.
- But what wheel configurations give the robot the most freedom to manoeuvre?
- What is the space of possible poses for a given wheel configuration?

Manoeuvrability

- Basic constraint limiting a robot's mobility is the rule that every wheel must satisfy its sliding constraint.

$$C_1(\beta_s)R(\theta)\dot{\xi}_I = 0$$

- A robot can also adjust its position (but only over time) by steering.
- Overall manoeuvrability is a combination of mobility due to sliding constraints, together with steering.

$$C_1(\beta_s)R(\theta)\dot{\xi}_I = 0$$

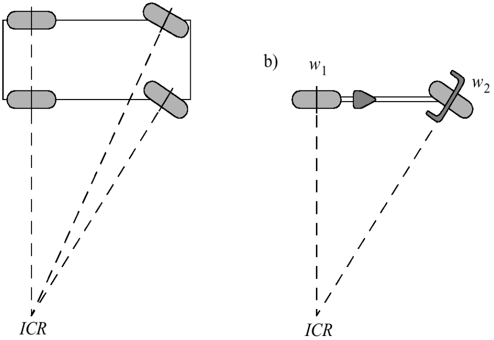
- We can separate out slippage constraints for fixed and steerable wheels.

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

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

- To satisfy these constraints, the motion vector $R(\theta)\dot{\xi}_I$ must belong to the null space of $C_1(\beta_s)$
- The null space N of a matrix X is such that for any vector $n \in N$ $Xn = 0$
- We can visualise it as the instantaneous centre of rotation (ICR).

ICR



- Each constraint prevents movement on zero motion line.
- Movement must occur around some circle with centre on the ZML.
- The centre of the circle is the instantaneous centre of rotation.
- Mobility depends on the number of independent constraints, not number of wheels.

More on Mobility

- Independence is related to the rank of a matrix — the largest number of linearly independent rows or columns.
- $rank\,[C_1(\beta_s)]$ is the number of independent constraints.
- Mobility is constrained with increasing rank.
- Maximum number of constraints is 3, minimum, 0.
- 0 is only possible if the vehicle has no fixed or steerable wheels.
- 3 completely constraints the robot — movement is impossible.
- $\delta_m = dimN[C_1(\beta_s)] = 3 - rank[C_1(\beta_s)]$
- This is a measure of the number of degrees of freedom that can be manipulated through wheel velocity changes.
- Compare unicycle to differential drive robot to bicycle (to spherical wheeled vehicle).

Degree of Steerability

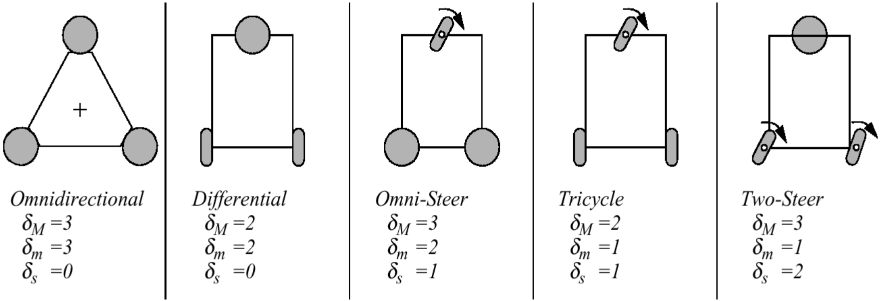
- Steering has an indirect impact on pose — robot must move for change in steering angle to have an effect.
- $\delta_s = rank[C_{1\mathbf{s}}(\beta_s)]$
- Increase in $C_{1s}(\beta_s)$ means more degrees of steering freedom.
- Robot can have decreased mobility and increased steer ability.
- $0 \leq \delta_s \leq 2$
- A two-steer vehicle can place ICR anywhere on the plane.

Degree of Maneuverability

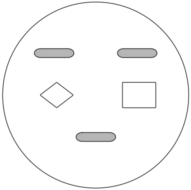
$$\delta_M = \delta_m + \delta_s$$

- Identifies the total degrees of freedom that the robot can manipulate.
- Two robots with identical δ_M are not identical
  - Differential drive $\delta_m = 2, \delta_s = 0$
  - Tricycle $\delta_m = 1, \delta_s = 1$
- If $\delta_M = 2$ ICR lies on a line
- If $\delta_M = 3$ ICR can be set anywhere on the plane

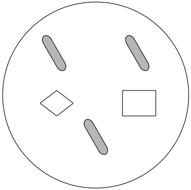
Five Basic Three Wheel Robots



Synchro-drive

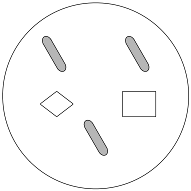


- The synchrodrive robot has one motor to drive all wheels, and another to rotate all wheels.
- $N_f = 0, \; N_s = 3$
- Two of the three wheels contribute independent sliding constraints, the third depends on these for motion to be possible.
- $rank[C_{1s}(\beta_s)] = 2, \; \delta_m = 1$
- What is δ_s?



- The synchrodrive robot has one motor to drive all wheels, and another to rotate all wheels.
- $N_f = 0,\ N_s = 3$
- Two of the three wheels contribute independent sliding constraints, the third depends on these for motion to be possible.
- $rank[C_{1s}(\beta_s)] = 2,\ \delta_m = 1$
- What is δ_s?

Synchro-drive



- The synchrodrive robot has one motor to drive all wheels, and another to rotate all wheels.
- $N_f = 0, \ N_s = 3$
- Two of the three wheels contribute independent sliding constraints, the third depends on these for motion to be possible.
- $rank[C_{1s}(\beta_s)] = 2, \ \delta_m = 1$
- What is δ_s?
- Normally $\delta_s = 2$, but wheels are not independently steerable; $\delta_s = 1$.
- $\delta_M = 2$
- Robot can only move in the plane, but cannot rotate.

Synchro-drive

http://www.youtube.com/watch?v=THdu6QD8Roc

- The synchrodrive robot has one motor to drive all wheels, and another to rotate all wheels.
- $N_f = 0$, $N_s = 3$
- Two of the three wheels contribute independent sliding constraints, the third depends on these for motion to be possible.
- $rank[C_{1s}(\beta_s)] = 2$, $\delta_m = 1$
- What is δ_s?
- Normally $\delta_s = 2$, but wheels are not independently steerable; $\delta_s = 1$.
- $\delta_M = 2$
- Robot can only move in the plane, but cannot rotate.

The Workspace

- We have identified the robot's control degrees of freedom.
- For example, an Ackerman vehicle has $\delta_M = 2$ through steering and the actuation of the drive wheels.
- But such a vehicle can place itself at any point in the plane, with any angle θ.
- The vehicle's possible workspace degree of freedom is therefore 3.
- But what paths can it follow? What are the possible trajectories through the configuration space?
- Ignore wheels, and focus on robot chassis pose.

DDOF

- The admissible velocity space describes the independent components of motion a robot can control.
- For a unicycle, the velocity space contains 2 axes, one for forward speed and the second for the instantaneous change in orientation.
- The dimensions of this space identify the number of independently achievable velocities, and are also called the differential degrees of freedom (DDOF).
- DDOF=δ_m
- For a bicycle $\delta_M = 2$, $\delta_m = DDOF = 1$. A bicycle can only (instantaneously) change its forward speed.
- An omnibot can set all three pose variables. DDOF=3

DDOF vs DOF

- An omnibot can clearly achieve any pose in its environment - workspace DOF=3
- So can a bicylce (e.g. parallel park) - workspace DOF=3

$$DDOF \leq \delta_M \leq DOF$$

- DOF governs the robot's ability to achieve various poses
- DDOF governs the robot's ability to achieve various paths.

Holonomic Robots

- A holonomic robot has zero nonholonomic constraints.
- A holonomic constraint can be expressed using only position variables.
- Non-holonomic constraint requires a differential relationship, e.g. taking into account $\dot{\varphi}$, and cannot be integrated to provide the constraint.

Example

- Consider a fixed wheel sliding constraint

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

- This constraint depends directly on robot motion.
- For a bicycle with a fixed and steered wheel, this constraint is in force; bicycles are non-holonomic.

Example

- Consider a bicycle with no steering. $\delta_M = 1$ for this, and it is non-holonomic.
- In X_R we can replace the sliding constraint with $y = 0, \theta = 0$, eliminating sliding non-holonomic constraints.
- Rolling constraint for wheels

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

- Given an initial rotational position, φ_0, we can replace the entire constraint with one that relates position on the line x with wheel rotation (and set $y = \theta = 0$).

$$\varphi = (x/r) + \varphi_0$$

- The locked bicycle is holonomic.

Holonomic robot

- Type 1 holonomic robot: holonomic constraints exist. This is the case if $\delta_M < 3$
- Type 2: $N_f = N_s = 0$, i.e. $\delta_M = 3$.
- A robot is holonomic if DDOF=DOF
- For a nonholonomic robot, DOF>DDOF
- An omnidirectional robot is holonomic with DDOF=3.

- We typically want robots to be able to follow any path in the workspace and maintain any pose.
- Holonomic robots can achieve this.
- Why do we not always use holonomic robots?

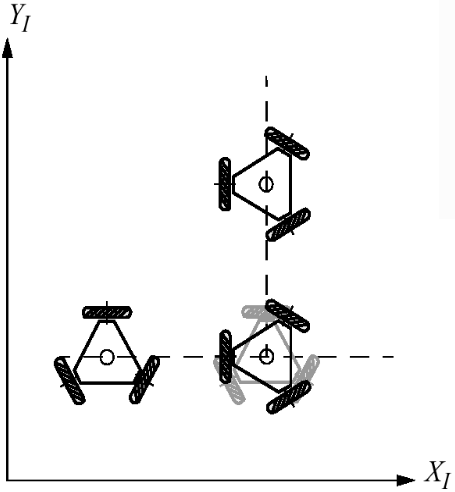
Paths and Trajectories

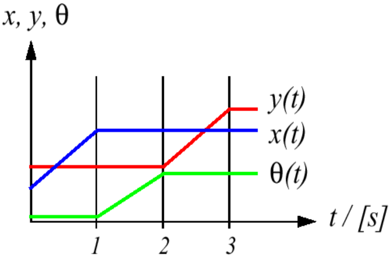
- We typically want robots to be able to follow any path in the workspace and maintain any pose.
- Holonomic robots can achieve this.
- Why do we not always use holonomic robots?
  - High ground clearance/suspension is hard with non-standard wheels.
  - Expensive/complex fabrication.
  - When moving around curves, such wheels must be steered by a motor to overcome centripetal force. Dangerous in case of motor failure.
- Standard wheels passively counteract such forces.

Paths and Trajectories

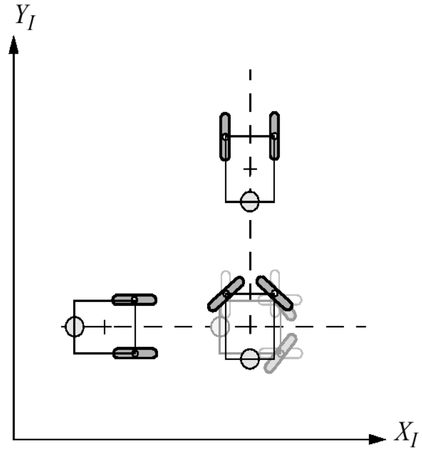
- Two-steer has $\delta_M = 3$ so can select any ICR and can thus follow any path in its workspace (as can any robot with $\delta_M = 3$ such as an omni-directional robot).
- So what is the difference between a two-steer and an omnidirectional robot?
- Wheel has to be steered into position - compare trajectories.

Omnidirectional drive

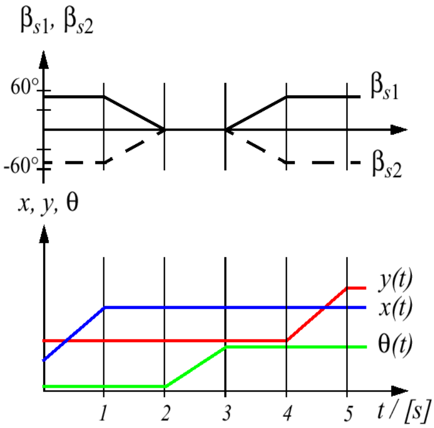




Two steer



Y_I
X_I



β_{s1}, β_{s2}
60°
-60°
β_{s1}
β_{s2}
x, y, θ
$y(t)$
$x(t)$
$\theta(t)$
1
2
3
4
5
t / [s]

Where are we?

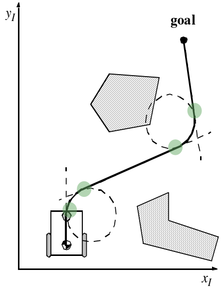
- We can analyse the range of motions available to a robot.
- We can describe a robot's velocity in terms of its wheel speeds.
- No consideration of the application of forces onto robot.
- No consideration of slippage.
- Given a robot, how do we control it - how should we change velocities to ensure that we get to a certain position and pose in the workspace?

Motion control

- So far, we have looked at forward kinematics - given wheel speeds, what motions result.
- Motion control requires consideration of inverse kinematics - given a desired motion, how should we move our wheels to achieve this motion?
- We will deal with this further when considering robotic actuators. For now, let us consider motion control.
- How can we smoothly and efficiently move from one place to another?

https://www.youtube.com/watch?v=wa6ZEG83eNM
https://www.youtube.com/watch?v=msxCG1_8ZfM

Open loop control



- Simple approach: build trajectory from lines and segments of circles.
- Problems:
  - Can't handle dynamic changes to the environment
  - Difficult to precompute desired trajectories
  - Constraints on robot velocity and acceleration are hard to take into consideration.
  - Trajectories not smooth