上海科技大学
ShanghaiTech University

CS283: **Robotics Fall 2020: Sensors**

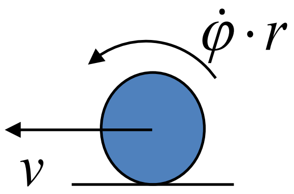
Sören Schwertfeger

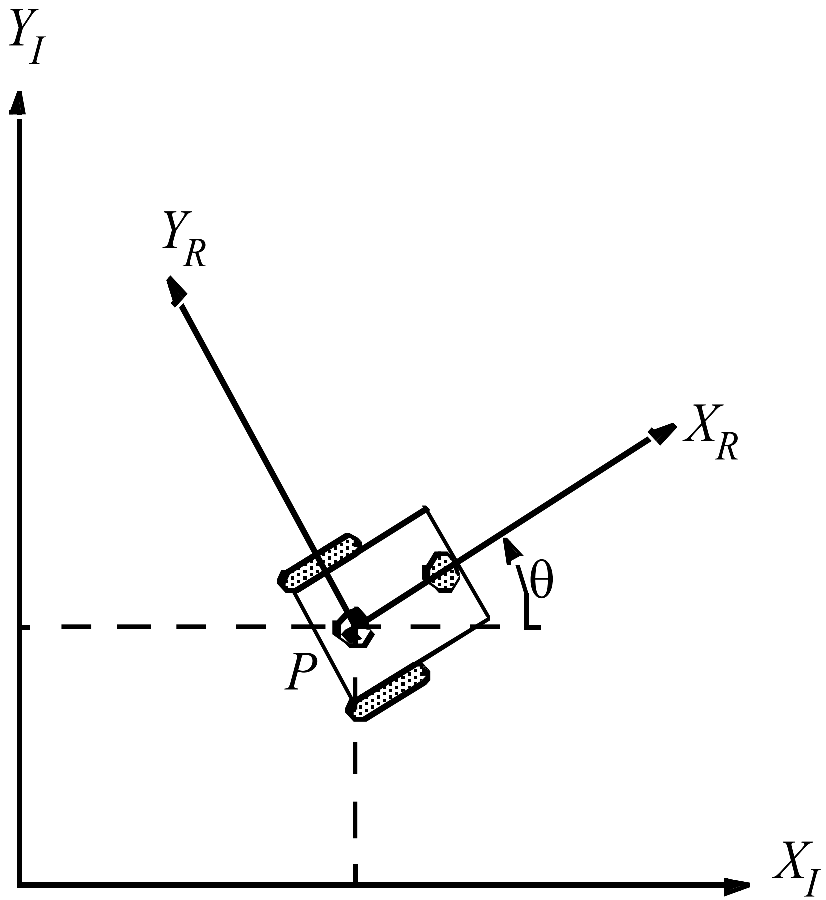
ShanghaiTech University

KINEMATICS CONTINUED

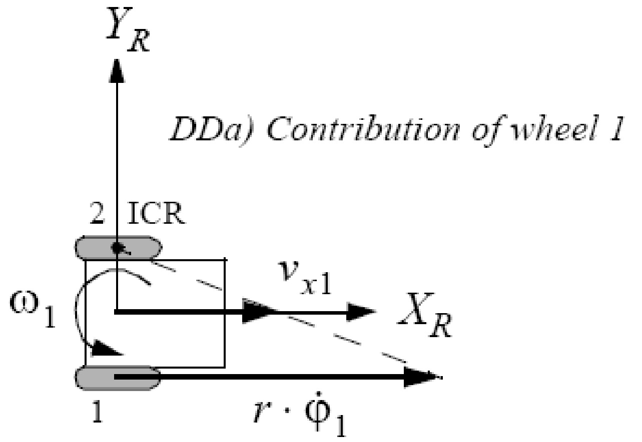
Wheel Kinematic Constraints: Assumptions

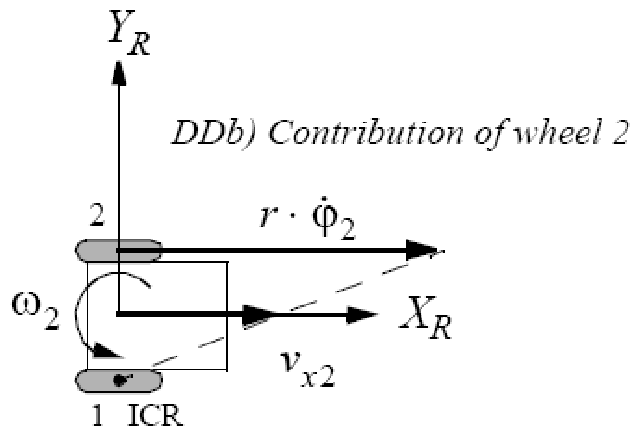
- Movement on a horizontal plane
- Point contact of the wheels
- Wheels not deformable
- Pure rolling
  - $v_c = 0$ at contact point
- No slipping, skidding or sliding
- No friction for rotation around contact point
- Steering axes orthogonal to the surface
- Wheels connected by rigid frame (chassis)





Forward Kinematic Model: Geometric Approach





Differential-Drive:

$$\text{DDa)}\ v_{x1} = \frac{1}{2}r\dot{\varphi}_1 \quad ; \quad v_{y1} = 0 \quad ; \quad \omega_1 = \frac{1}{2l}r\dot{\varphi}_1$$

$$\text{DDb)}\ v_{x2} = \frac{1}{2}r\dot{\varphi}_2 \quad ; \quad v_{y2} = 0 \quad ; \quad \omega_2 = -\frac{1}{2l}r\dot{\varphi}_2$$

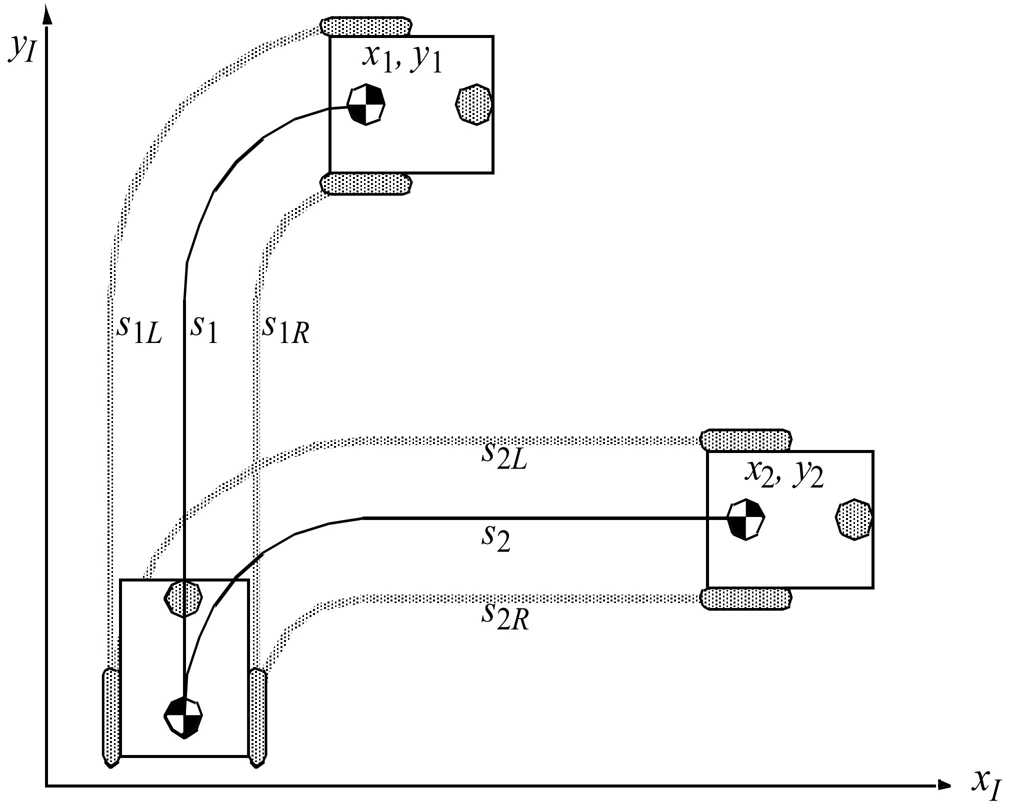
$$\text{->}\ \xi_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}_I = R(\theta)^{-1} \begin{bmatrix} v_{x1} + v_{x2} \\ v_{y1} + v_{y2} \\ \omega_1 + \omega_2 \end{bmatrix} = R(\theta)^{-1} \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ \frac{r}{2l} & -\frac{r}{2l} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix}$$

Inverse of R => Active and Passive Transform:
http://en.wikipedia.org/wiki/Active_and_passive_transformation

Mobile Robot Kinematics: Non-Holonomic Systems

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

but: $x_1 \neq x_2$; $y_1 \neq y_2$



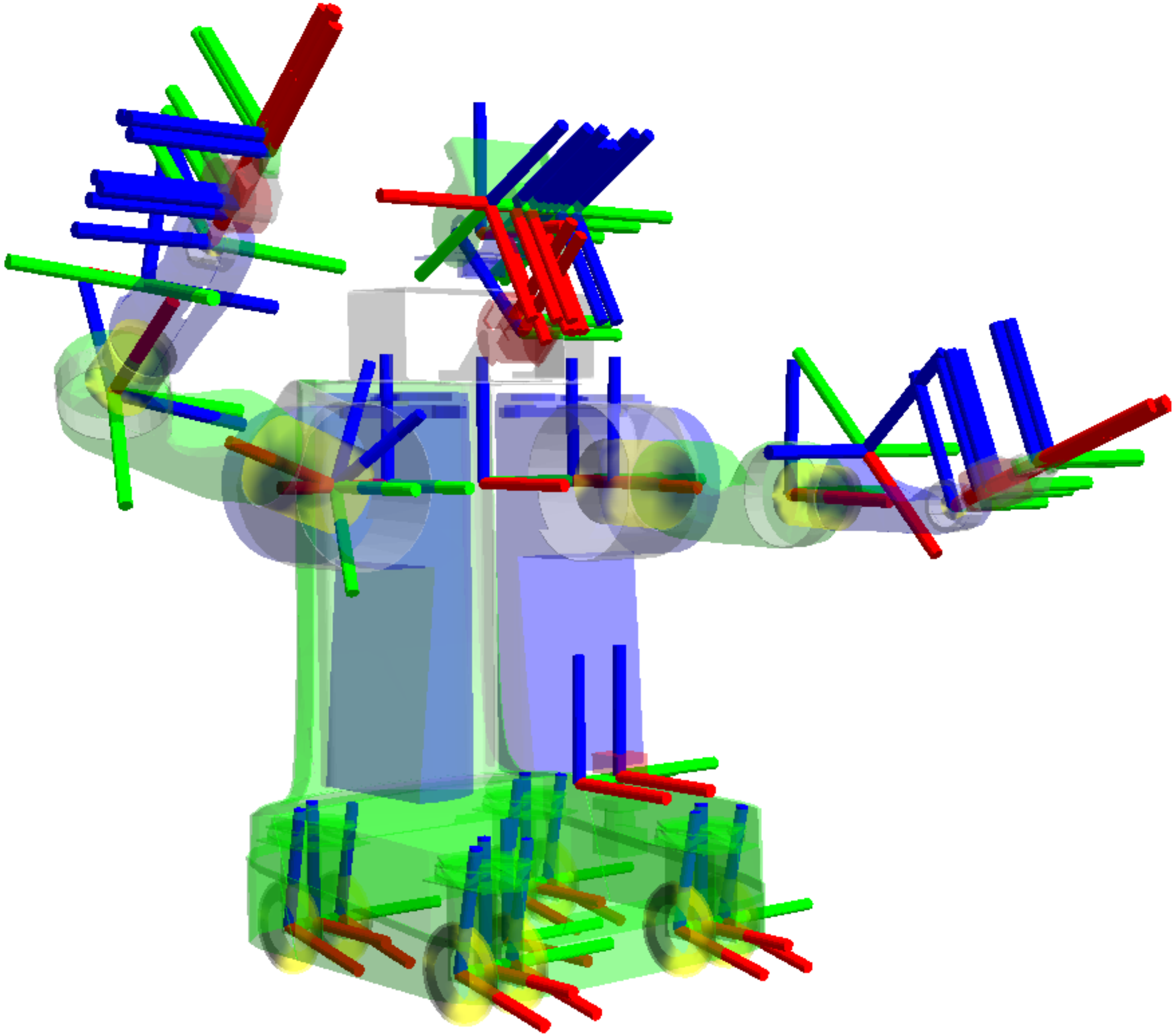
- Non-holonomic systems
  - differential equations are not integrable to the final pose.
  - the measure of the traveled distance of each wheel is not sufficient to calculate the final position of the robot. One has also to know how this movement was executed as a function of time.

Holonomic examples

Uranus, CMU

ROS: 3D Transforms : TF

- http://wiki.ros.org/tf
- http://wiki.ros.org/tf/Tutorials



ROS geometry_msgs/TransformStamped

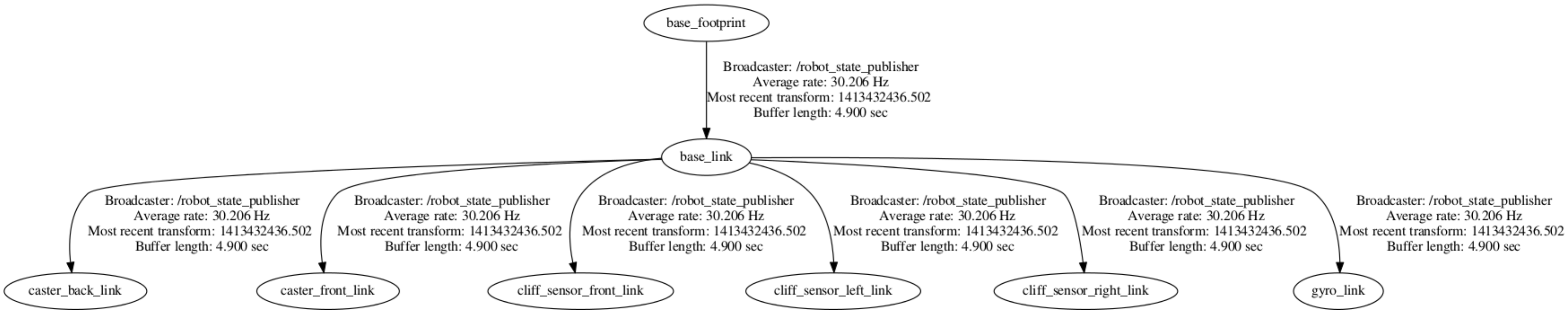
- $^{\text{header.frame\_id[header.stamp]}}_{\text{child\_frame\_id[header.stamp]}}\mathbf{T}$
- Transform between header (time and reference frame) and child_frame
- 3D Transform representation:
  - geometry_msgs/Transform:
    - Vector3 for translation (position)
    - Quaternion for rotation (orientation)

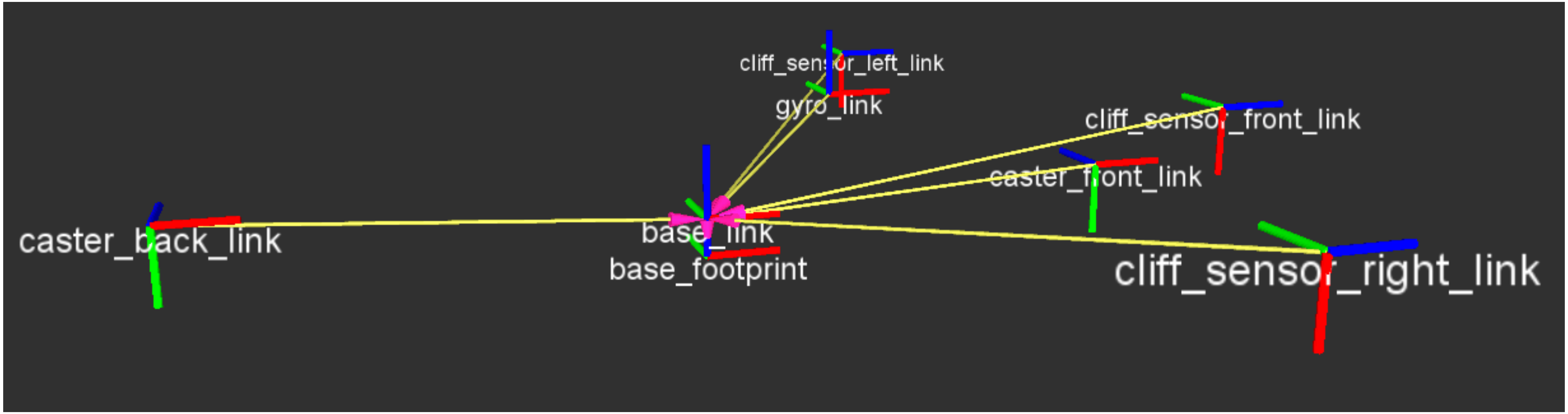
```
rosmsg show geometry_msgs/TransformStamped

std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
string child_frame_id
geometry_msgs/Transform transform
  geometry_msgs/Vector3 translation
    float64 x
    float64 y
    float64 z
  geometry_msgs/Quaternion rotation
    float64 x
    float64 y
    float64 z
    float64 w
```

ROS tf2_msgs/TFMessage

- An array of TransformStamped
- Transforms form a tree
- Transform listener: traverse the tree
  - tf::TransformListener listener;
- Get transform:
  - tf::StampedTransform transform;
  - listener.lookupTransform("/base_link", "/camera1", ros::Time(0), transform);
  - ros::Time(0): get the latest transform
  - Will calculate transform by chaining intermediate transforms, if needed

```
rosmsg show tf2_msgs/TFMessage

geometry_msgs/TransformStamped[] transforms
  std_msgs/Header header
    uint32 seq
    time stamp
    string frame_id
  string child_frame_id
  geometry_msgs/Transform transform
    geometry_msgs/Vector3 translation
      float64 x
      float64 y
      float64 z
    geometry_msgs/Quaternion rotation
      float64 x
      float64 y
      float64 z
      float64 w
```


base_footprint
Broadcaster: /robot_state_publisher
Average rate: 30.206 Hz
Most recent transform: 1413432436.502
Buffer length: 4.900 sec
base_link
Broadcaster: /robot_state_publisher
Average rate: 30.206 Hz
Most recent transform: 1413432436.502
Buffer length: 4.900 sec
caster_back_link
caster_front_link
cliff_sensor_front_link
cliff_sensor_left_link
cliff_sensor_right_link
gyro_link

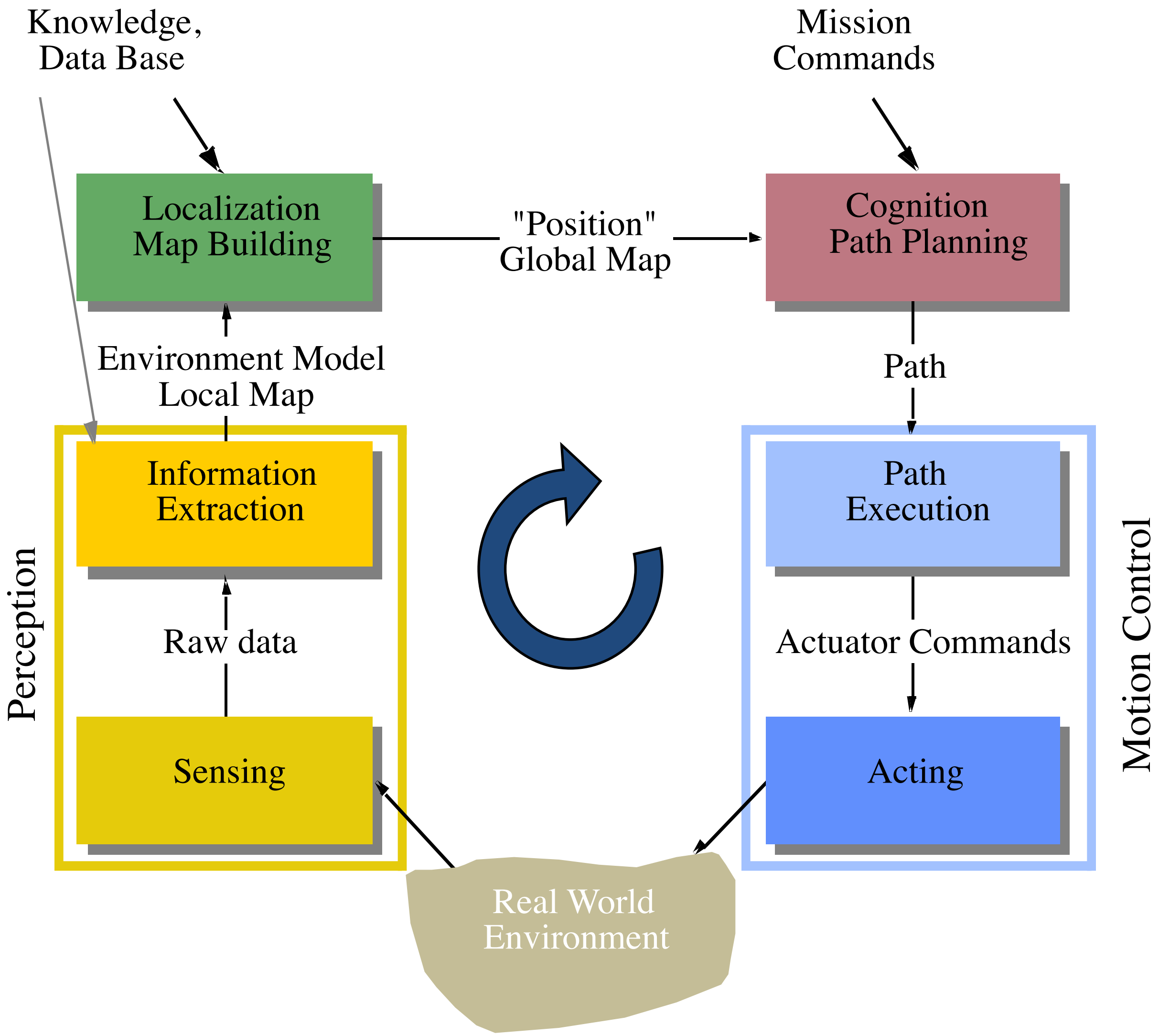

cliff_sensor_left_link
gyro_link
cliff_sensor_front_link
caster_front_link
caster_back_link
base_link
base_footprint
cliff_sensor_right_link

Transforms in ROS

- Imagine: Object recognition took 3 seconds – it found an object with:
  - tf::Transform object_transform_camera; // ${}^{\mathrm{Cam}[X]}_{Obj}\mathbf{T}$ (has tf::Vector3 and tf::Quaternion)
  - and header with: ros::Time stamp; // Timestamp of the camera image (== X)
    - and std::string frame_id; // Name of the frame (“Cam”)
- Where is the object in the global frame (= odom frame) “odom” ${}^{\mathrm{G}}_{Obj}\mathbf{T}$?
  - tf::StampedTransform object_transform_global; // the resulting frame
  - listener.lookupTransform(child_frame_id, “/odom", header.stamp, object_transform_global);
- tf::TransformListener keeps a history of transforms – by default 10 seconds

HIGH-LEVEL CONTROL SCHEMES

General Control Scheme for Mobile Robot Systems



SENSORS

Introduction to Autonomous Mobile Robots page 102 ff

Sensors for Mobile Robots

- Why should a robotics engineer know about sensors?
  - Is the **key technology** for perceiving the environment
  - **Understanding the physical principle** 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**

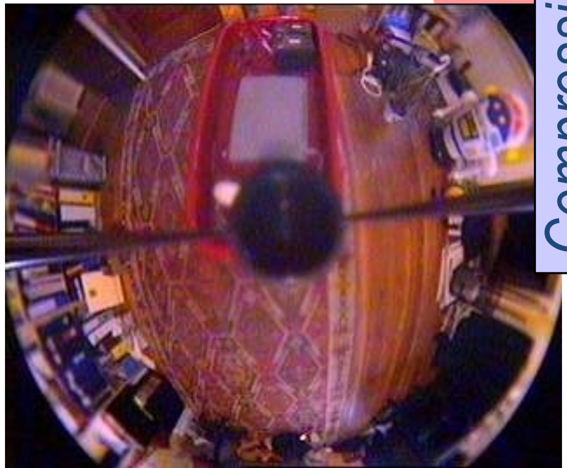
Dealing with Real World Situations

- Reasoning about a situation

- Cognitive systems have to interpret situations based on uncertain and only partially available information
- The need ways to learn functional and contextual information (semantics / understanding)

Probabilistic Reasoning

Perception for Mobile Robots



Compressing Information

Places / Situations
A specific room, a meeting situation, …

Servicing / Reasoning

Objects
Doors, Humans, Coke bottle, car , …

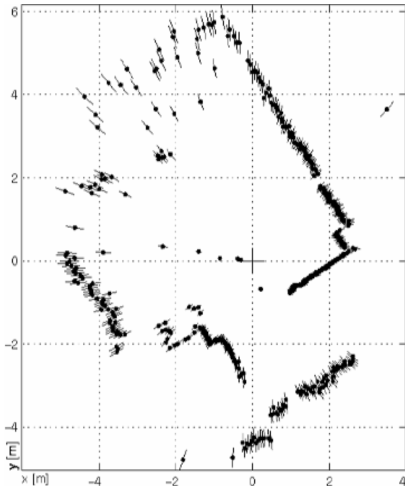
Interaction

Features
Lines, Contours, Colors, Phonemes, …

Navigation

Raw Data
Vision, Laser, Sound, Smell, …

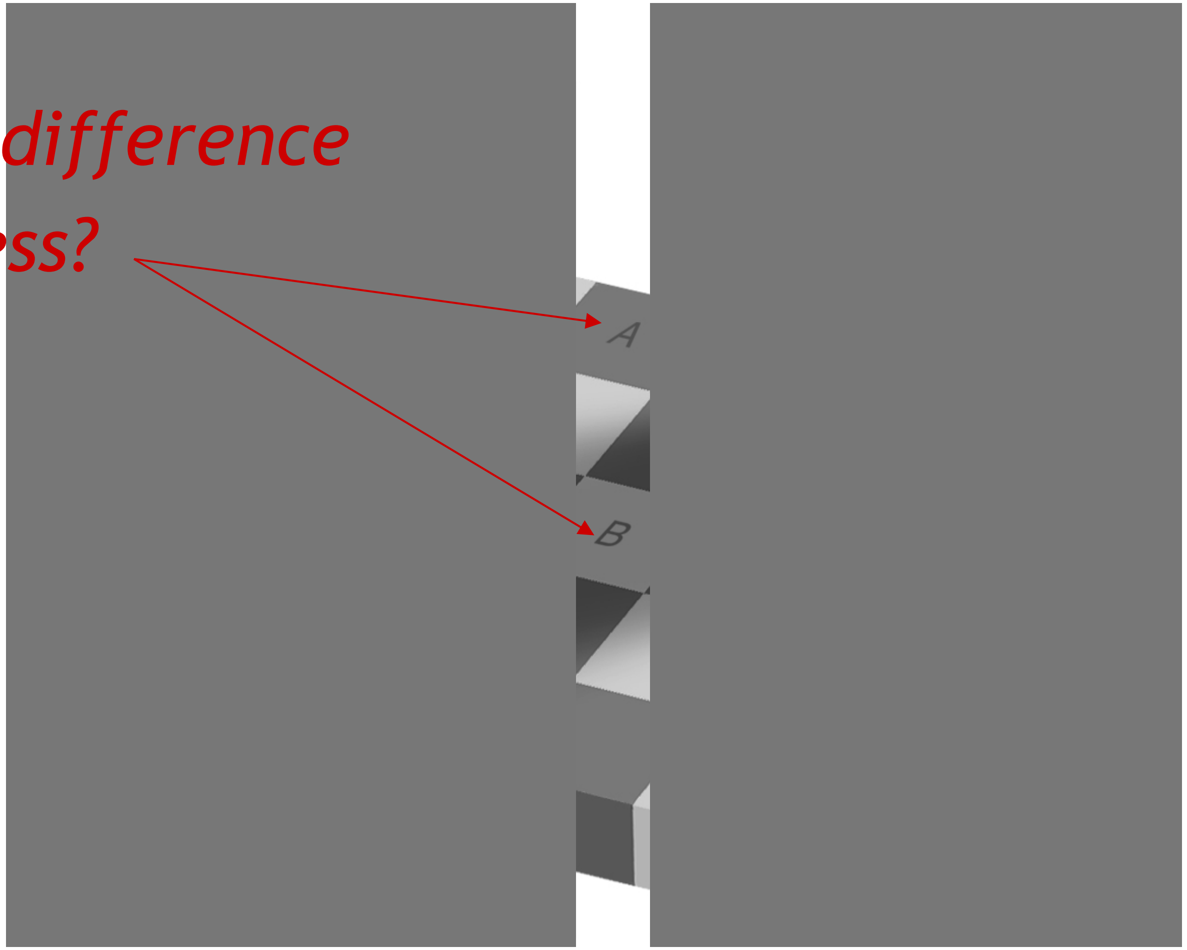
- ***Functional / Contextual Relationships of Objects***
  - *imposed*
  - *learned*
  - *spatial / temporal/semantic*
- ***Models / Semantics***
  - *imposed*
  - *learned*
- ***Models***
  - *imposed*
  - *learned*



The Challenge

- Perception and models are strongly linked

What is the difference in brightness?



- http://web.mit.edu/persci/people/adelson/checkershadow_downloads.html

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

In Situ Sensor Performance

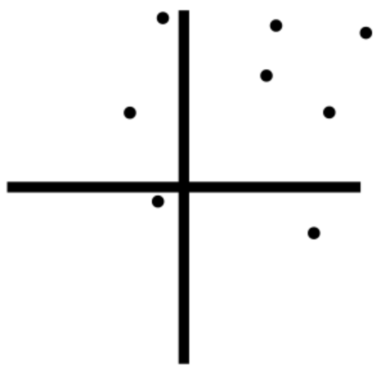
- In Situ: Latin for “in place”

- Error / Accuracy
  - How close to true value
- Precision
  - Reproducibility

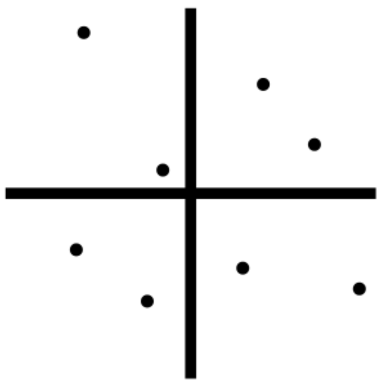
error

$$\left(accuracy = 1 - \frac{|m - v|}{v} \right)$$

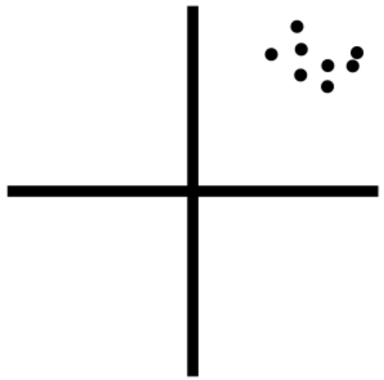
m = measured value
v = true value



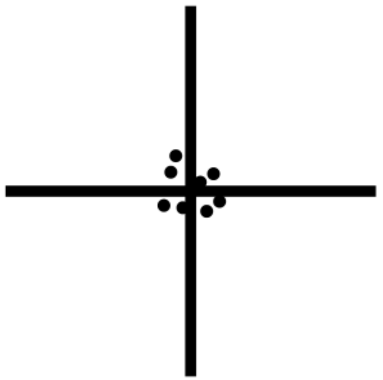
(a) Low precision and low accuracy



(b) Low precision and high accuracy



(c) High precision and low accuracy



(d) High precision and high accuracy

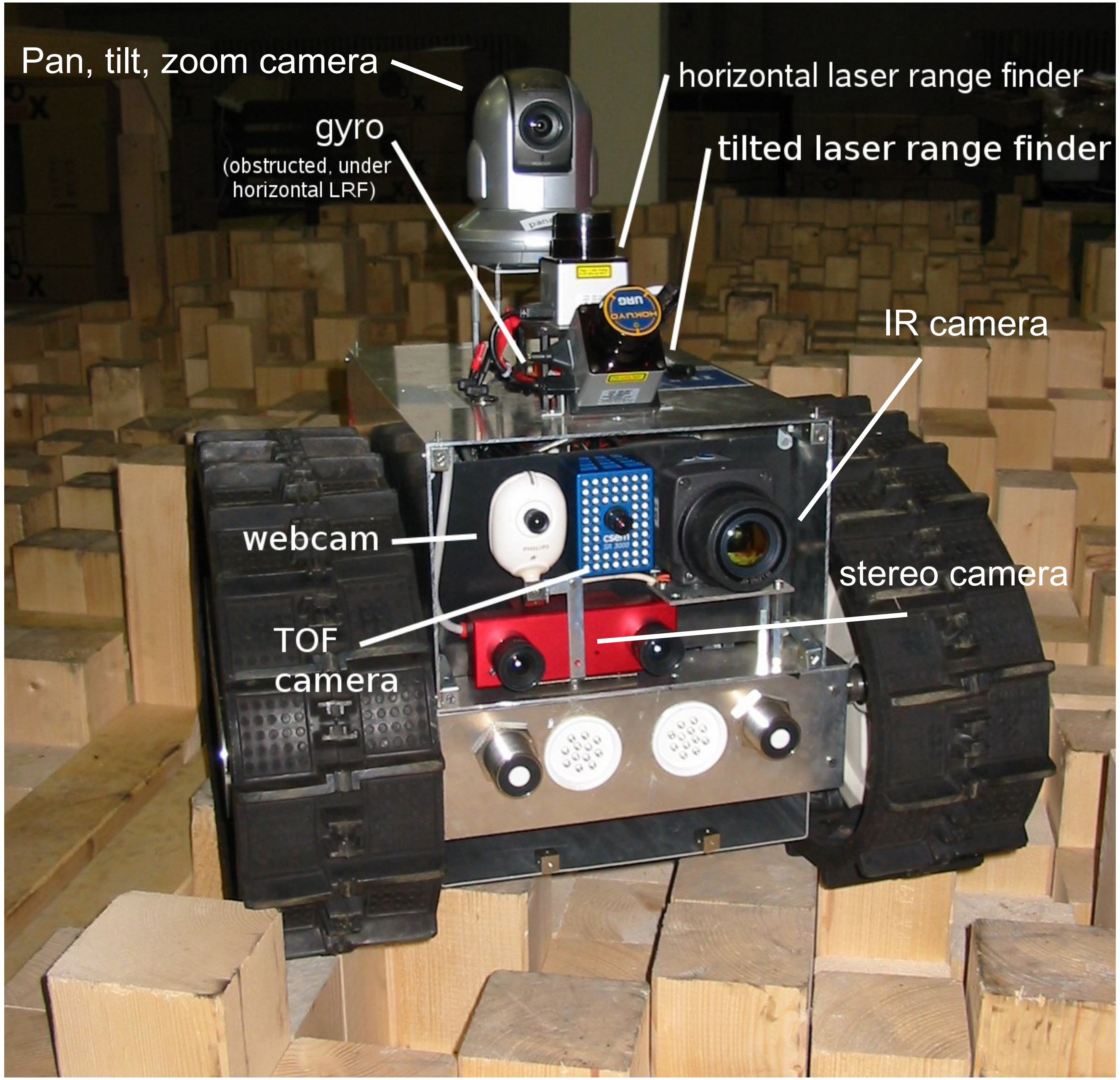
Types of error

- 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 optic of a camera

- Random error -> non-deterministic
  - no prediction possible
  - however, they can be described probabilistically
  - e.g. Hue instability of camera, black level noise of camera ..

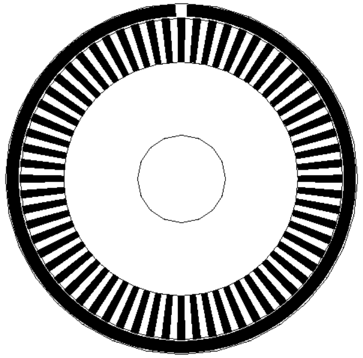
Sensors: outline

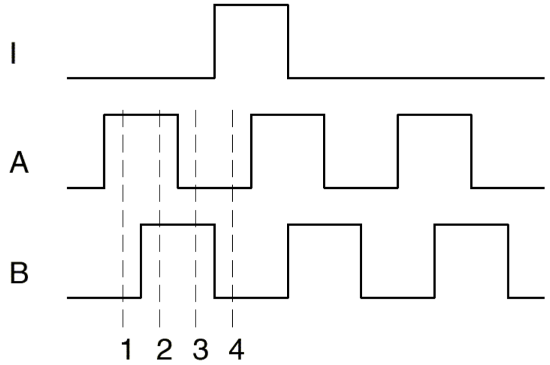
- Optical encoders
- Heading sensors
  - Compass
  - Gyroscopes
  - Accelerometer
  - IMU
- GPS
- Range sensors
  - Sonar
  - Laser
  - Structured light
- Vision



Wheel / Motor Encoders

- 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
- 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

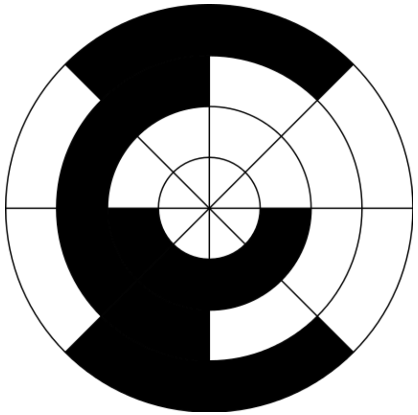




State	Ch A	Ch B
S_1	High	Low
S_2	High	High
S_3	Low	High
S_4	Low	Low

Gray Encoder

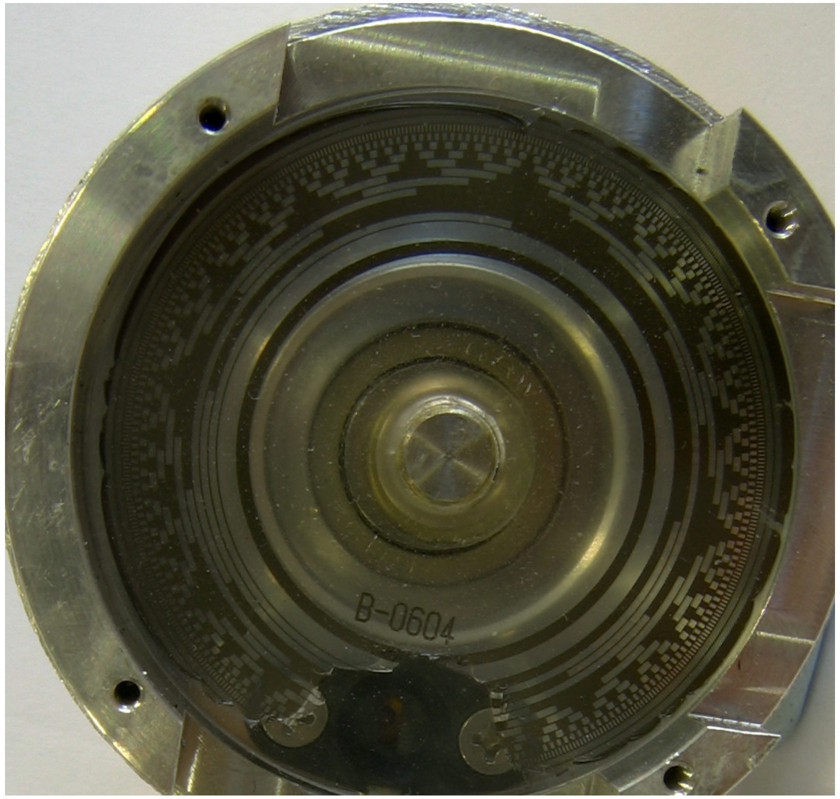
http://en.wikipedia.org/wiki/Gray_code



- Aka: reflected binary code, Gray Code
  - Binary numeral system where two successive values differ in only one bit
  - Also used for error correction in digital communications

- Absolute position encoder
  - Normal binary => change from 011 to 100
  - 2 bits change – NEVER simultaneously =>
  - 011 -> 111 -> 101 -> 100 or
  - 011 -> 010 -> 110 -> 100 ….
  - => wrong encoder positions might be read
  - Gray encoding: only one bit change!

Dec	Gray	Binary
0	000	000
1	001	001
2	011	010
3	010	011
4	110	100
5	111	101
6	101	110
7	100	111

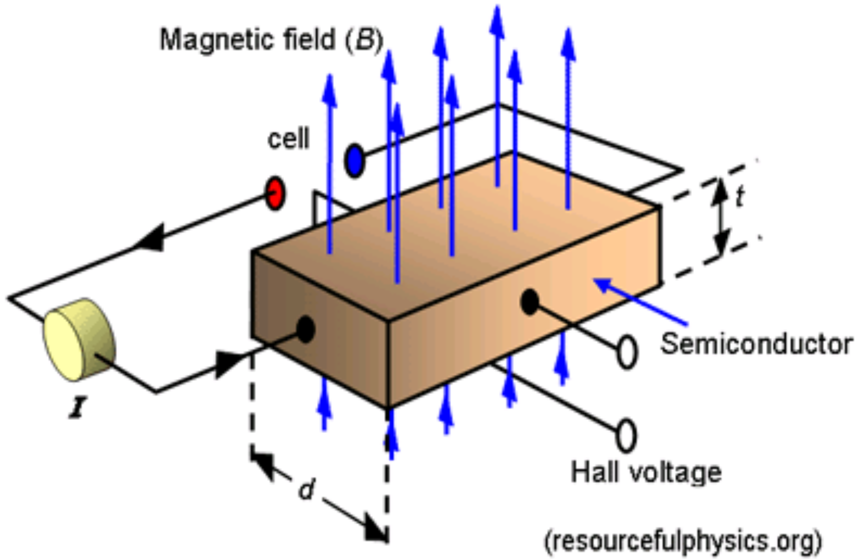


Heading Sensors

- Heading sensors can be proprioceptive (gyroscope, acceleration) or exteroceptive (compass, inclinometer).
- Used to determine the robots orientation and inclination.
- Allow, together with an appropriate velocity information, to integrate the movement to a position estimate.
  - This procedure is called **deduced reckoning** (ship navigation)

Compass

- Magnetic field on earth
  - absolute measure for orientation
- Large variety of solutions to measure the earth magnetic field
  - mechanical magnetic compass
  - direct measure of the magnetic field (Hall-effect, magneto-resistive sensors)
- Major drawback
  - 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 feasible for indoor environments for absolute orientation
  - useful indoor (only locally)



(resourcefulphysics.org)

Gyroscope

- Heading sensors that preserve their orientation in relation to a fixed reference frame
  - absolute measure for the heading of a mobile system.

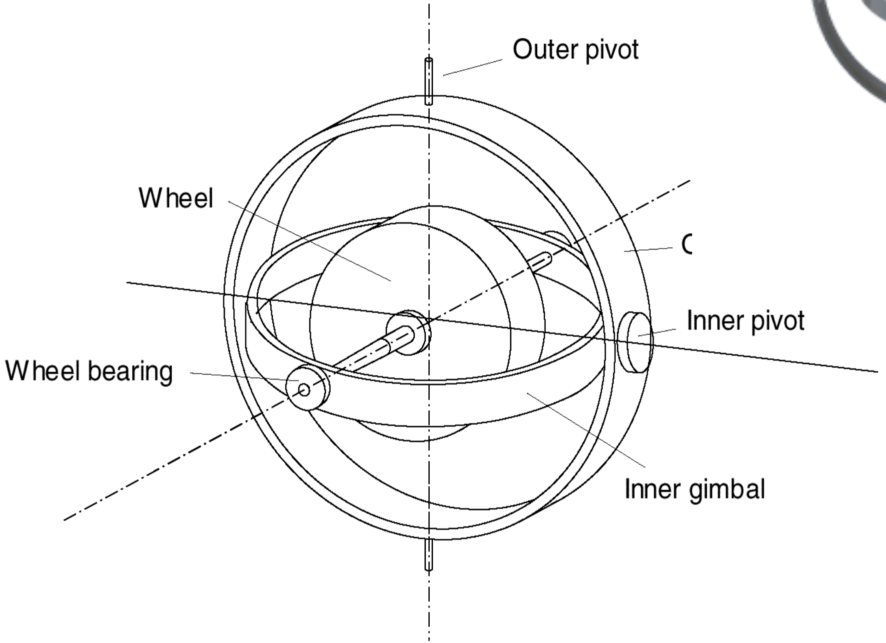
- Two categories, the mechanical and the optical gyroscopes
  - Mechanical Gyroscopes
    - Standard gyro (angle)
    - Rate gyro (speed)

  - 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
- Quality: 0.1° in 6 hours (a high quality mech. gyro costs up to 100,000 $)

- If the spinning axis is aligned with the north-south meridian, the earth's rotation has no effect on the gyro's horizontal axis
- If it points east-west, the horizontal axis reads the earth rotation

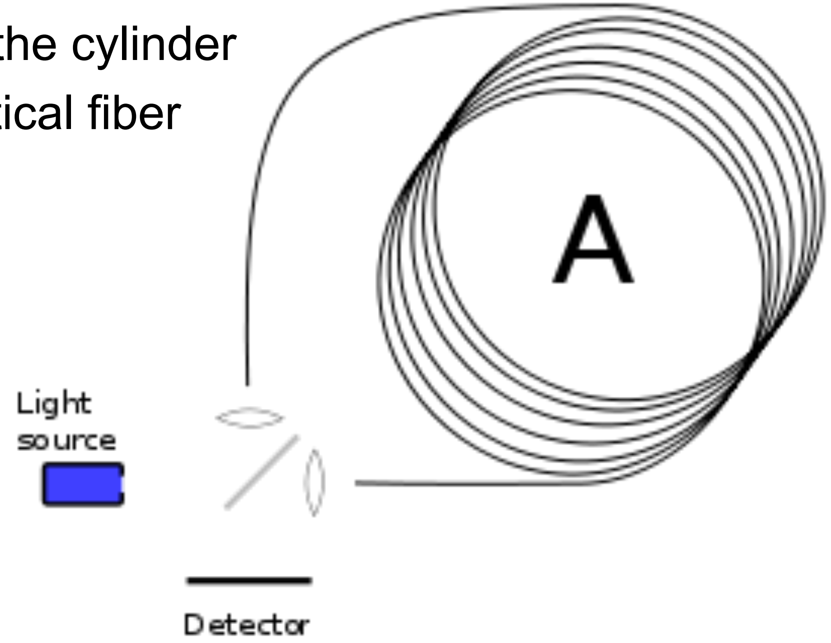


Optical Gyroscopes

- First commercial use: early 1980 in airplanes
- 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 micro-fabrication technology.

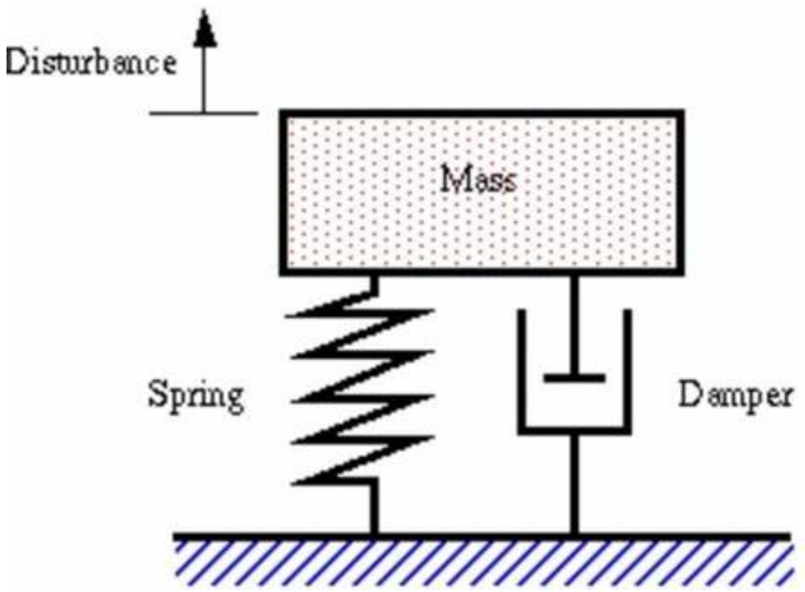

Single axis optical gyro


3-axis optical gyro

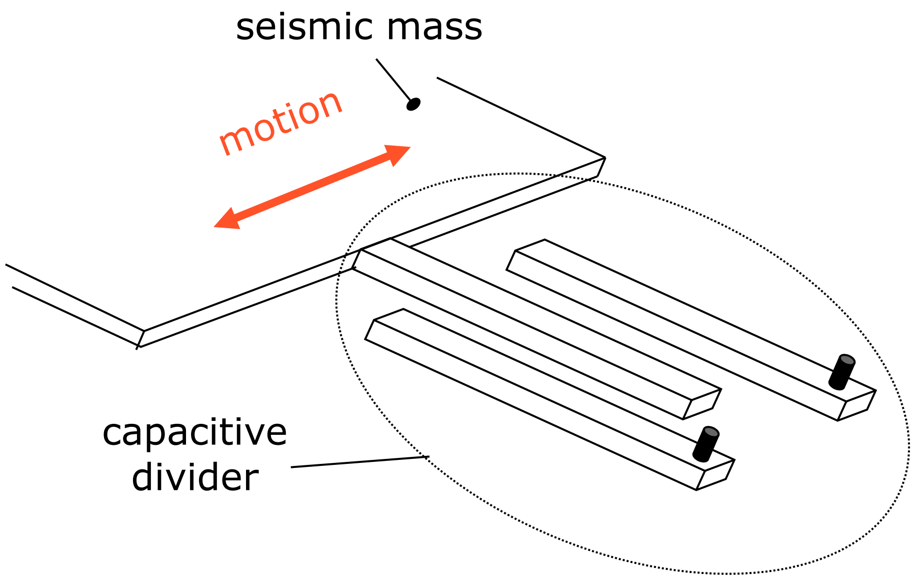


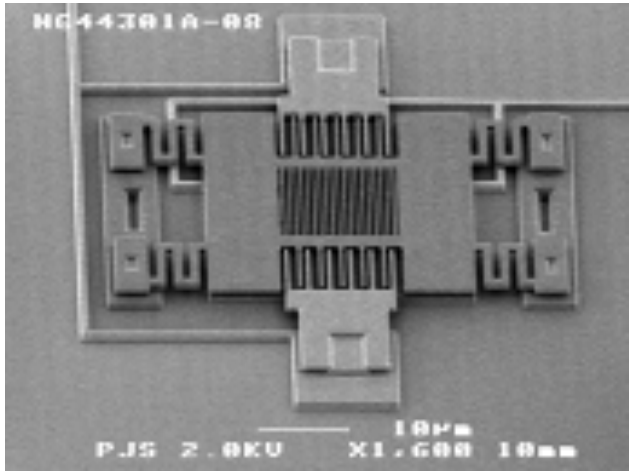
Mechanical Accelerometer

- Accelerometers measure all external forces acting upon them, including gravity
- Accelerometer acts like a spring–mass–damper system
- 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.
- Bandwidth up to 50 KHz
- An accelerometer measures acceleration only along a single axis
- => mount 3 accelerometers orthogonally => three-axis accelerometer



Factsheet: MEMS Accelerometer





<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

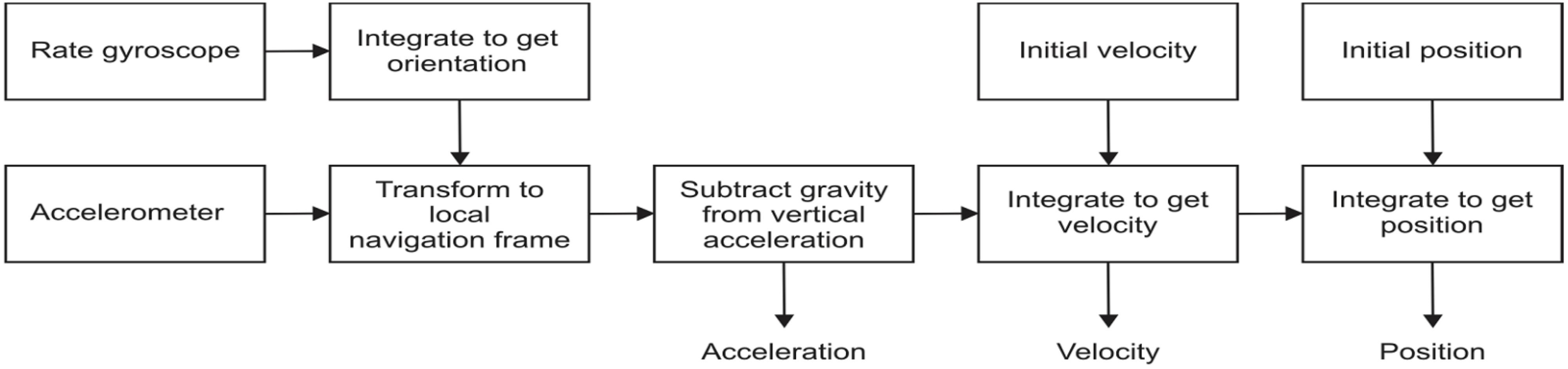
3. Applications

- Dynamic acceleration
- Static acceleration (inclinometer)
- Airbag sensors (+- 35 g)
- Control of video games (Wii)

Inertial Measurement Unit (IMU)

- Device combining different measurement systems:
  - Gyroscopes, Accelerometers, Compass
- Estimate relative position (x, y, z), orientation (roll, pitch, yaw), velocity, and acceleration
- Gravity vector is subtracted to estimate motion
  - Initial velocity has to be known

Xsens MTI



IMU Error and Drift

- Extremely sensitive to measurement errors in gyroscopes and accelerometers:
  - drift in the gyroscope unavoidably =>
  - error in orientation relative to gravity =>
  - incorrect cancellation of the gravity vector.

- Accelerometer data is integrated twice to obtain the position => gravity vector error leads to quadratic error in position.

- All IMUs drift after some time
  - Use of external reference for correction:
  - compass, GPS, cameras, localization

ADMIN

HW 1

- Read HW 1 carefully – do not forgot to include: “Your name (Chinese and Pinyin), your student id and your email. Also include the course name and the number of the homework.”

# of Groups	Name	Short Description	Advisor	Difficulty: Hardware	Software	Algorithm
1	Underwater Segmentation	Segment use DL to segment things like Coral, Divers, Sun, Open Water, Sand, Fish… in omni-directional UW images	Haofei	none	med	high
1	Mapping Robot Camera Collection	Collect, compress, store 60Hz image 5MP data (project taken already!)	Jiadi	med	high	low
1-2	Rover Manipulation	Continue last year's project to equipt the CAS planetary rover with manipulation (soil sampling)	Jiadi	med	med	low
1	Semantic Mapping/ OCR	Use camera to read door signs for mapping and localization	Jiadi	none	med	high
1	Elevator project	Enable mobile manipulation robot to ride the elevator. Continue last year's project	Jiawei	low	high	med
1	MoManTu: Extend to Speech	Add a module to MoManTu to enable speech interaction	Jiawei	low	high	med
1-2	MoManTu: Other Extensions	Propose to extend MoManTu with another capability (has no pdf description)	Jiawei	?	?	?
1	eFMT Extension	Use a drone to collect down-looking data. Use the novel eFMT algorithm to estimate the motion of the drone	Qingwen	low	high	med
1	FMT Star Gazer	Use an up-looking camera to localize and estimate motion of a robot using FMT	Qingwen	low	high	high
1	Multi-Turtlebots	Put a Control-Theory algorithm on real robots: Use many turtlebots to drive at a busy intersection	Qingwen	high	high	low
1	UW Camera	Build version 2 of the omni-directional underwater camera. Use LEDs and Arduino for synchronization. Collect IMU data.	Qingwen	high	med	low
1	5G Teleop	use 4G or 5G to teleoperate a robot. Transmit abritary ROS messages over 5G. Use bandwidth management.	Prof.	med	high	low
1	Rescue Omni Teleop	Use an omni-camera on the MARS Rescue Robot for teleoperation. Includes image transmission and rectification.	Prof.	med	high	low
1	RoboMasters: auto-aiming	Use computer vision to aim the gun	Prof.	low	high	high
1	RoboMasters: Engineer	automatic system to help the operator to align with and pick up supplies	Prof.	med	med	high

1	RoboMasters: Engineer	automatic system to help the operator to align with and pick up supplies	Prof.	med	med	high
1	Phone AR	Mobile Phone AR with scanned pointcloud	Xiting	none	med	high
1	Point Cloud Registration	Register Point Clouds based on geometry features (e.g. lines, planes)	Xiting	none	med	high
1	Point Cloud Colorization	Advanced Algorithm to assign colors to 3D points based on camera images	Xiting	none	med	high
1	Lidar Intrinsic Calibration	3D Lidars need to be calibrated. Develop an algorithm to automatically do this.	Xiting	none	med	high
1	Point Cloud Semantic Segmentation	Use Deep Learning for semantic segmentation of 3D point clouds	Yijun	none	med	high
1	Trinocular Stereo for Underwater	Stereo-vision with 3 cameras on (omni-directional) underwater data	Yijun	none	med	high
1	Car Project SLAM	Use data from last year's car project to create 3D map of ShanghaiTech parking garage	Zhijie	none	med	high
1	Mapping Robot: analog sensors	project (no PDF): Implement a sonar sensor array (I2C and Tinkerboard) and pressure sensor complete with ROS drivers. Do SLAM experiments with those.	Zhijie	med	med	low
1	Mapping Robot: Computing	We will use 12 or more Intel NUCs on the mapping robot. Explore how to effectively control this cluster (with ROS).	Zhijie	med	high	low
1 or 2	RoboCup Rescue Hardware	Upgrade the MARS rescue robot with new motors, a sensor head and updated software	Zhijie	high	high	low
1 or 2	Cotton Project	Join the ShanghaiTech Industry collaboration on an automatic cotton picking robot. Implement one or two advanced components.	Ziwen	med	high	high
1	Prof. Wang Yang quadcopters	Prof. Wang wants to resarch control with several quadcopters. Implement this in the MARS tracking system. (No PDF)	Prof.	high	high	med
?	Your own Project	(E.g. for students of Prof. Rosendo or Prof. Kneip, or others)				

Project

- 32 projects are offered. Max 25 are needed…
- Work in groups, min 2 students, max 3 students!
  - You can also do your own topic, but only after approval of Prof. Schwertfeger
    - Prepare a short, written proposal – **TODAY!**
- Topic selection: Till this Friday!
  - **One member writes an email for the whole group to Cui Jiadi: cuijd(at)shanghaitech.edu.cn ; Put the other group members on CC**
  - **Subject: [Robotics] Group Selection**
- One graduate student from my group will co-supervise your project
  - **Maybe request a meeting with the supervisor!**
- Weekly project meetings!
- Oral "exams" to evaluate the contributions of each member
- No work on project => bad grade of fail