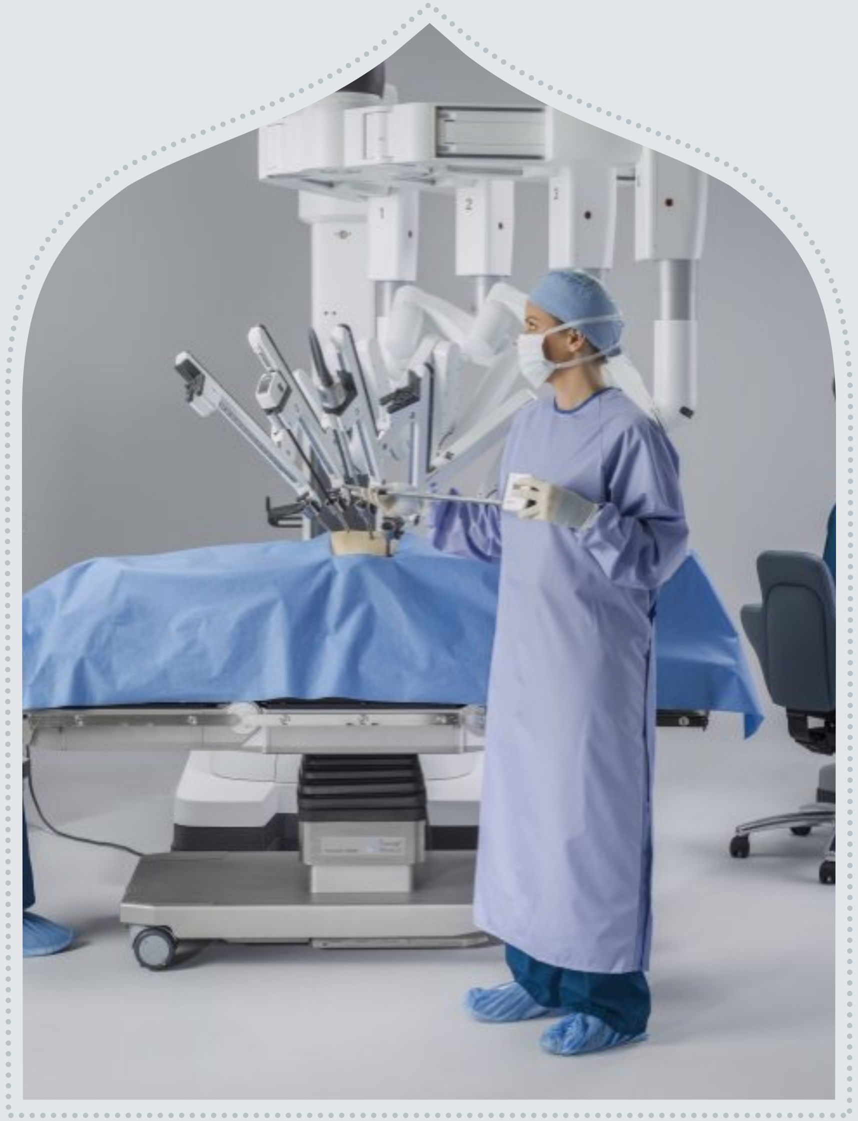
Robot Autonomy

DR. RISMAN ADNAN MATTOTORANG

TELKOM UNIVERSITY

Introduction

- Robotics is the science of perceiving and manipulating the physical world through computer-controlled devices
- Robotics systems are situated in the physical world, perceive information on their environments through sensors, and manipulate through physical forces.
- While much of robotics is still in its infancy, the idea of “intelligent” manipulating devices has an enormous potential to change society.



From automation ...

… to autonomy

Waymo Self-Driving Car

Intuitive DaVinci Surgical Robot

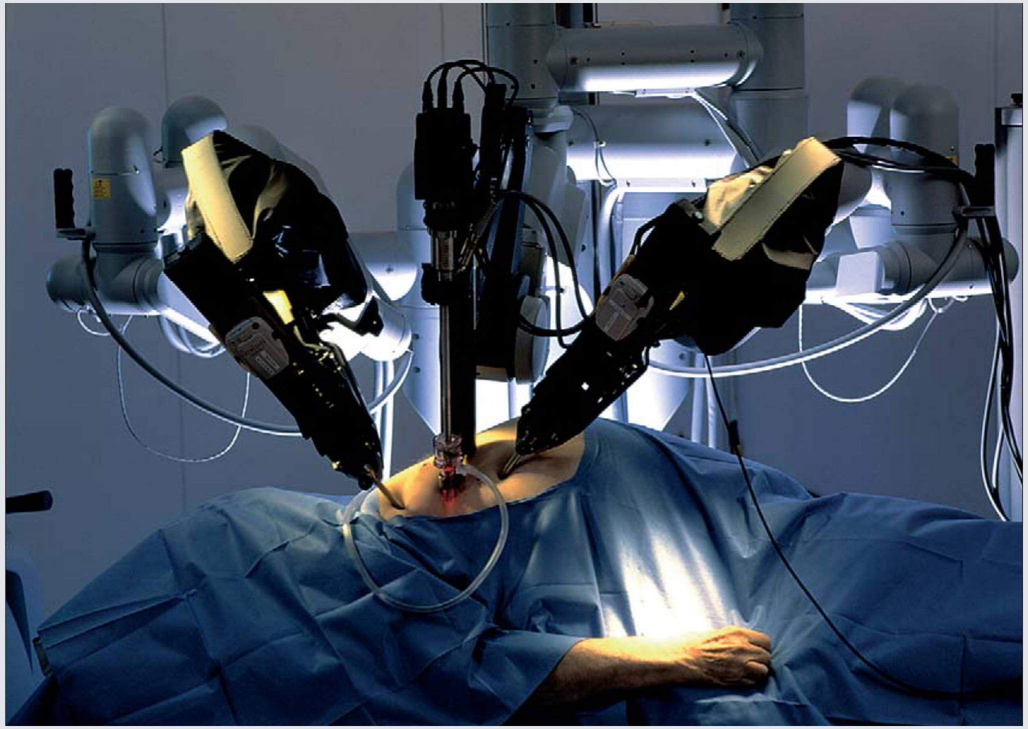
Apollo Robot at MPI for Intelligent Systems

Robot's Uncertainty

Think of a medical robot. The robot has to be able to accommodate the enormous uncertainty that exists in "the physical world". Factors that contribute robot's uncertainty:

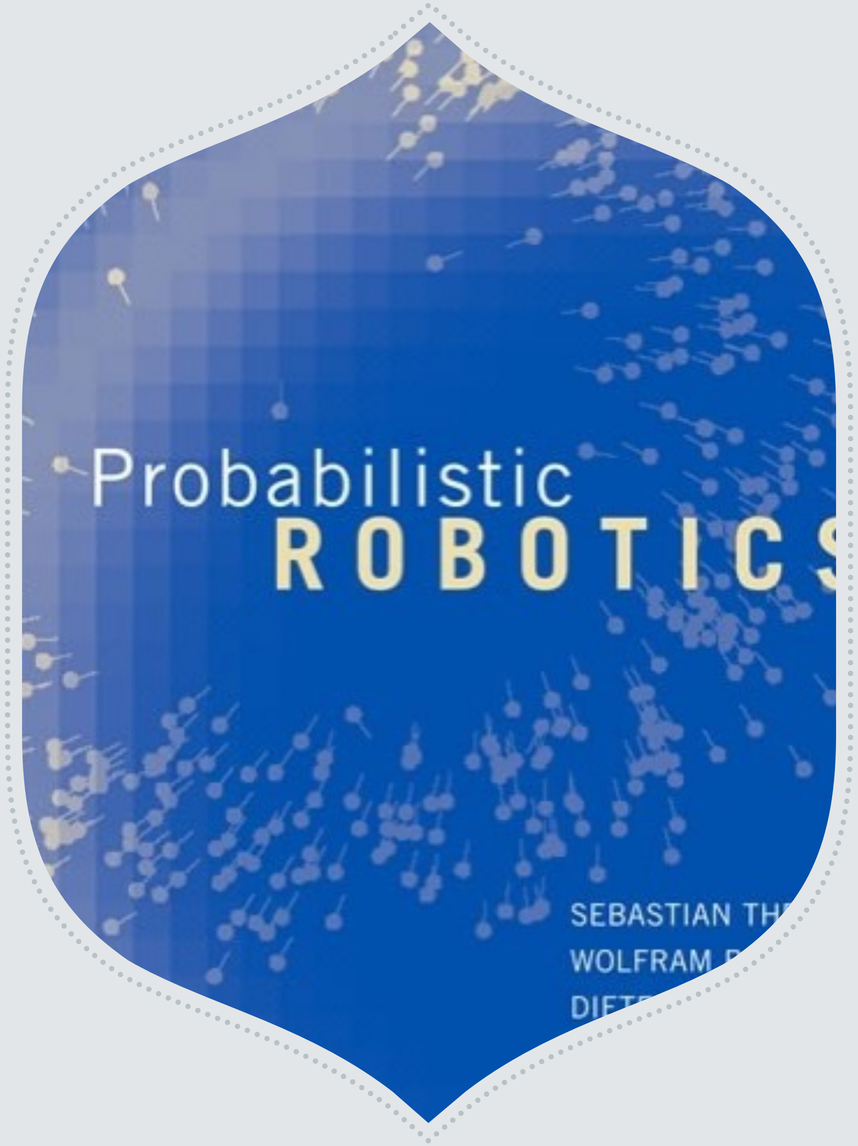
- ***Robot environment is*** inherently unpredictable especially in the proximity of people.
- ***Sensors*** are limited in what they can perceive, sensors are also subject to noise and can break.
- ***Robot actuation*** involves motors that are, at least to some extent, unpredictable.
- ***Robot software*** uses *internal models* of the world as abstractions of the real world.
- ***Robots are real-time systems*** with algorithmic approximations.

The level of uncertainty depends on the application domain. Managing uncertainty is possibly the most important step towards robust real-world robot systems (eq. environment, sensor, action, software, realtime, etc)



Probabilistic Robotics

- The key idea in probabilistic robotics is to represent uncertainty explicitly using **the calculus of probability theory**.
- Probabilistic algorithms represent information by probability distributions over a whole space of guesses. Hence, they can represent ambiguity and degree of belief in a mathematically sound way.



Implications

Advantages:

- Probabilistic robotics seamlessly integrates models with sensor data, overcoming the limitations of both at the same time.
- Probabilistic approaches tend to be more robust in the face of sensor limitations and model limitations.
- Probabilistic algorithms have weaker requirements on the accuracy of the robot's models, thereby relieving the programmer from the insurmountable burden to come up with accurate models.

Disadvantages:

- **Computation complexity:** Probabilistic algorithms are inherently less efficient than their non-probabilistic counterparts.
- **Need to approximate: In some cases approximations are too crude to be of use, and more complicated representations must be employed.**

Goals of This Course

To learn the “Fundamental, Conceptual and Practical” aspects of Robot Autonomy covering basic non-probabilistic ideas:

- Motion control and planning
- Robotic perception
- Localization and SLAM
- State machines and system architecture

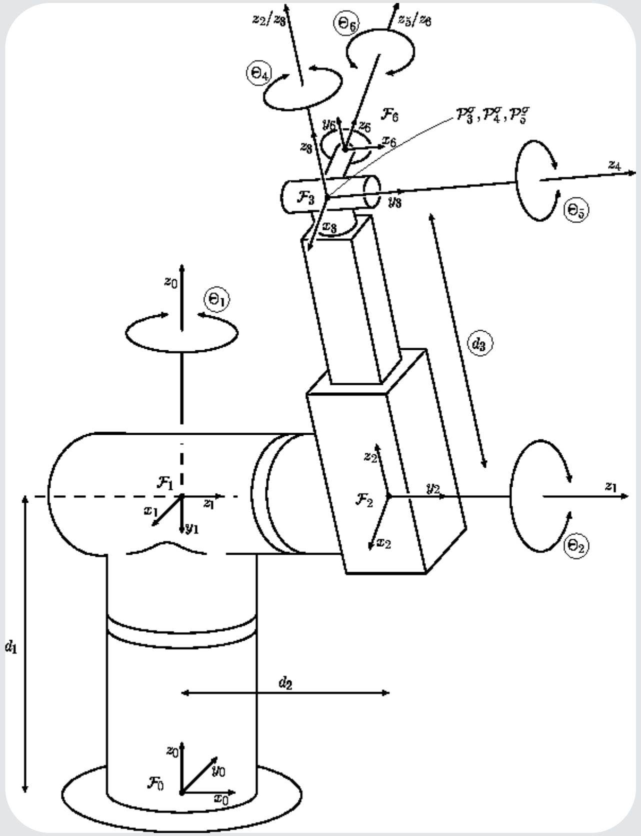
Specifically, you will:

1. Gain a fundamental knowledge of the “autonomy stack”
2. Be able to apply such knowledge in applications / research by using ROS
3. Devise novel methods and algorithms for robot autonomy

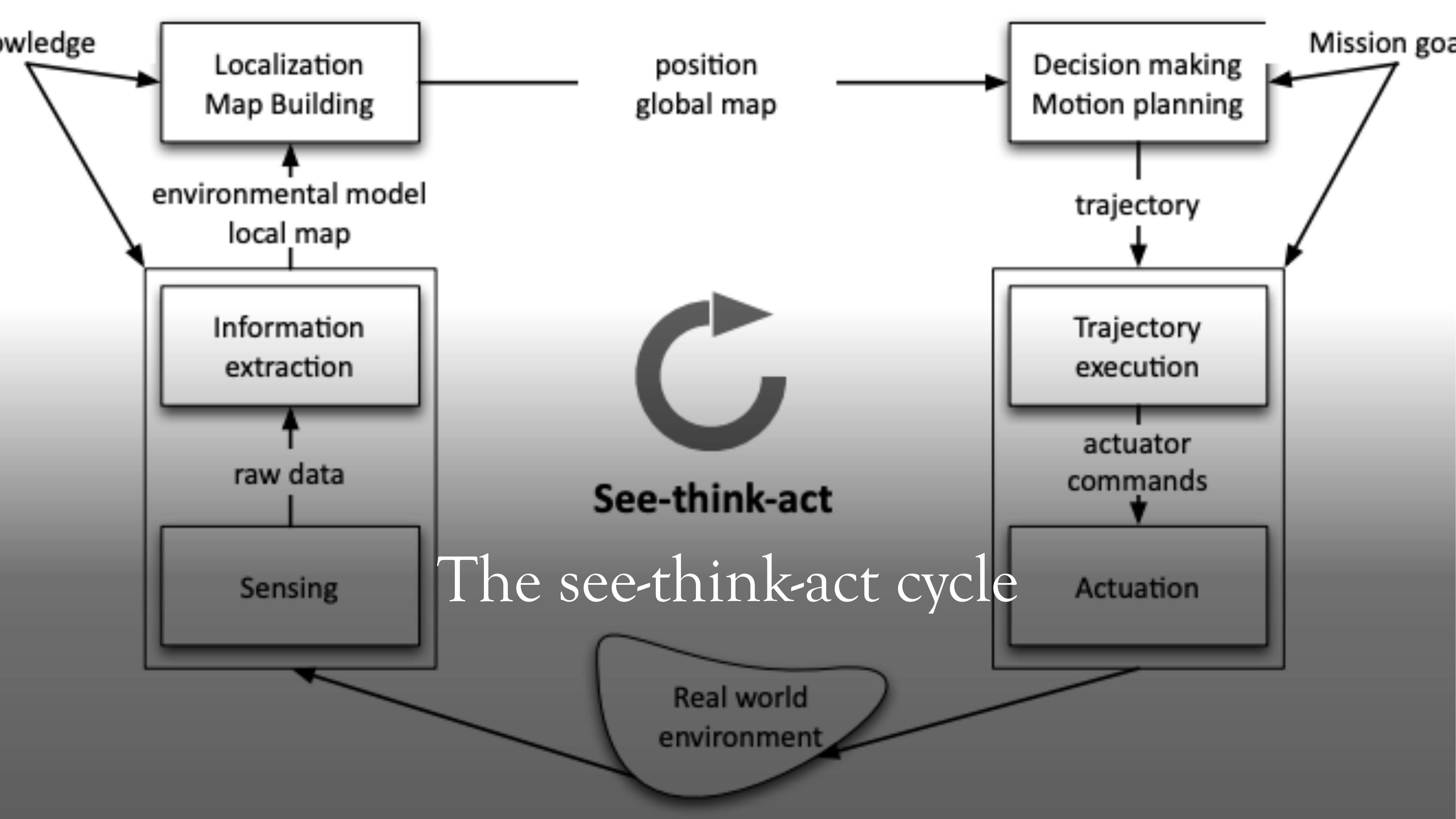
Principles of Robot Autonomy I

W1: Introduction to Robot Kinematics

- Mobile Kinematics Robot
- Generalized Coordinates
- Kinematic Constraint
- Holonomic and Nonholonomic Constraints
- Kinematic Model
- Kinematic Model of Wheeled Robots
- Dynamic Models

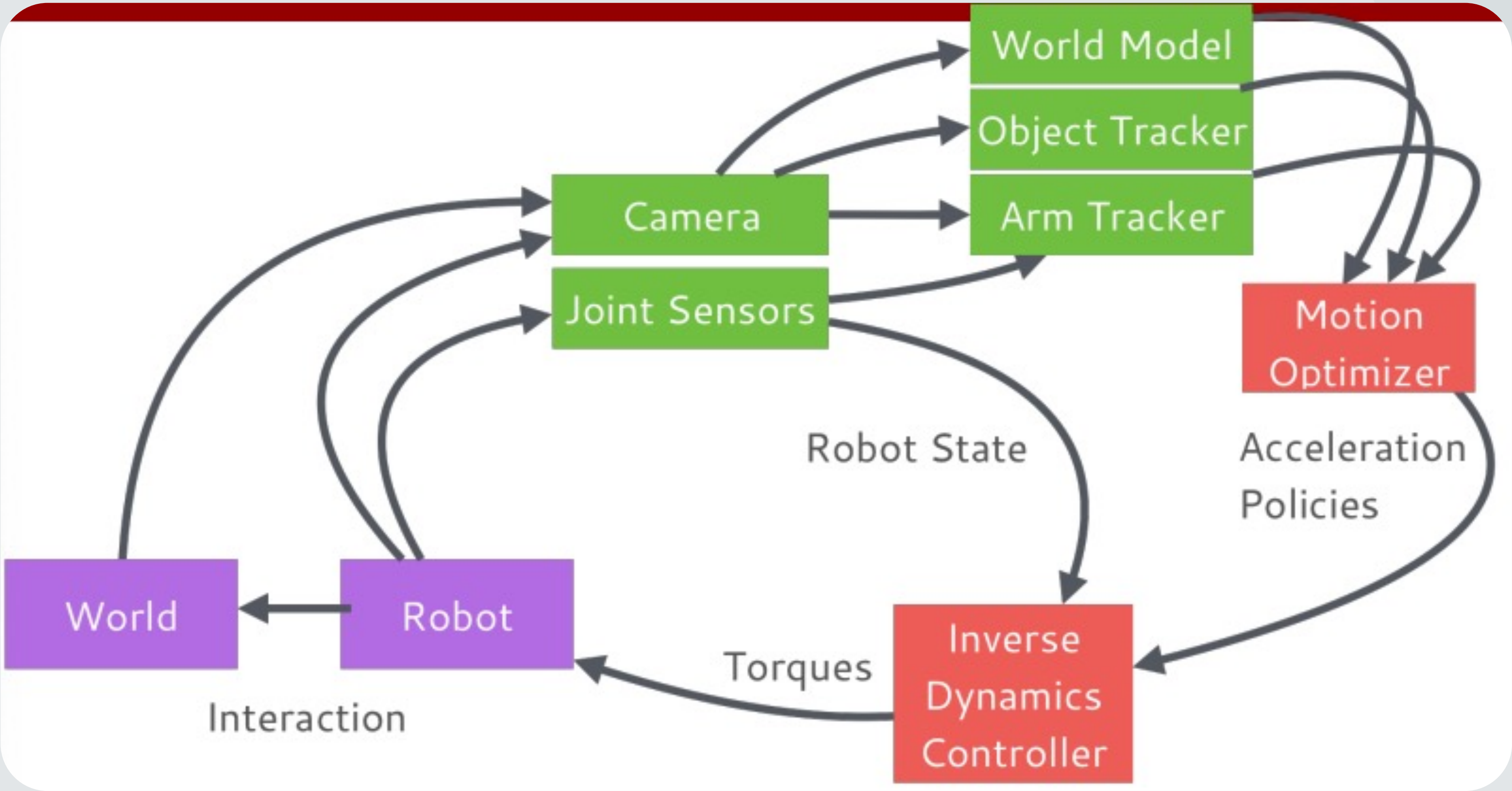


Stanford Arm Robot

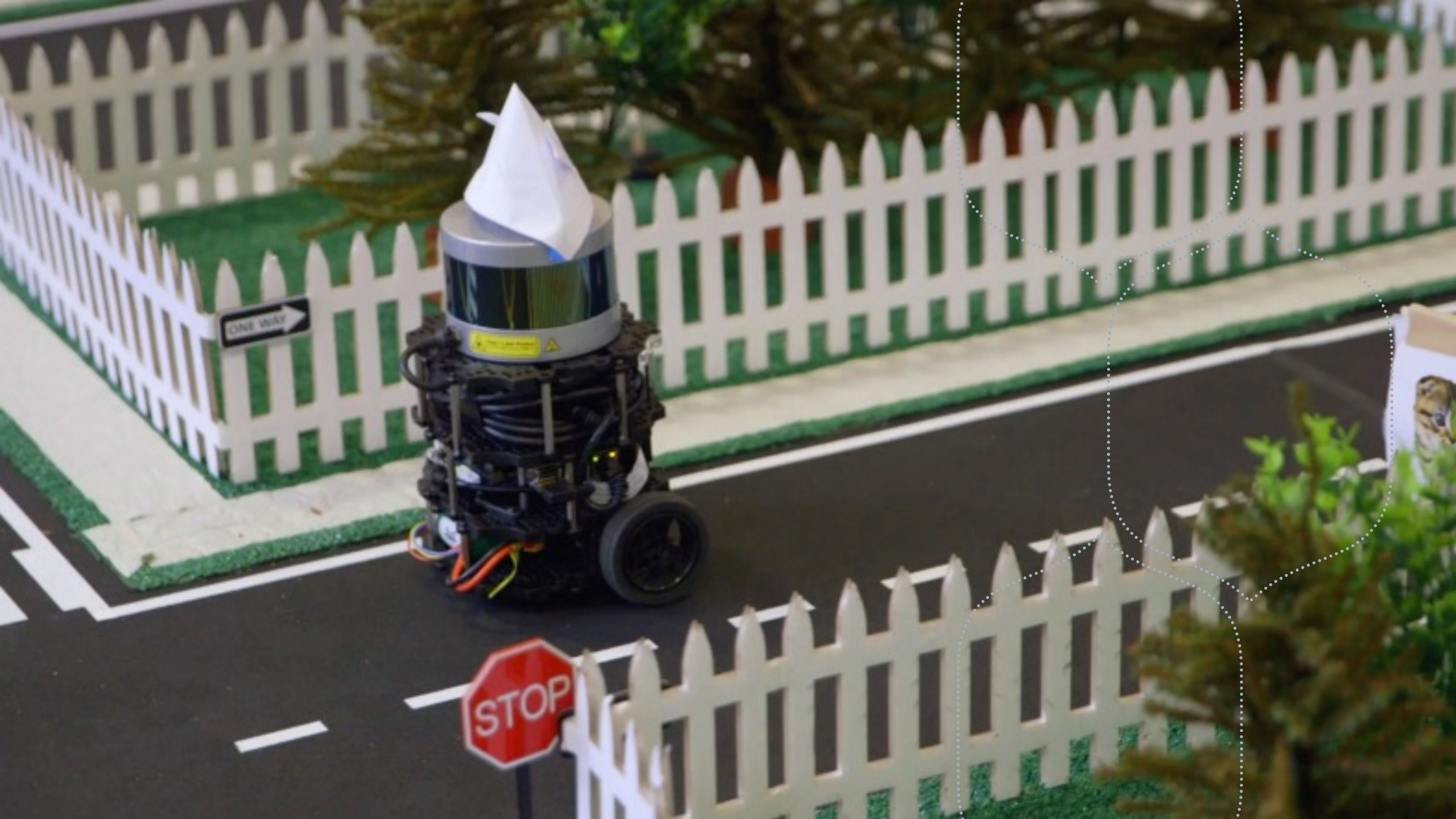


The see-think-act cycle

Real-time Perception meets Reactive Motion Generation

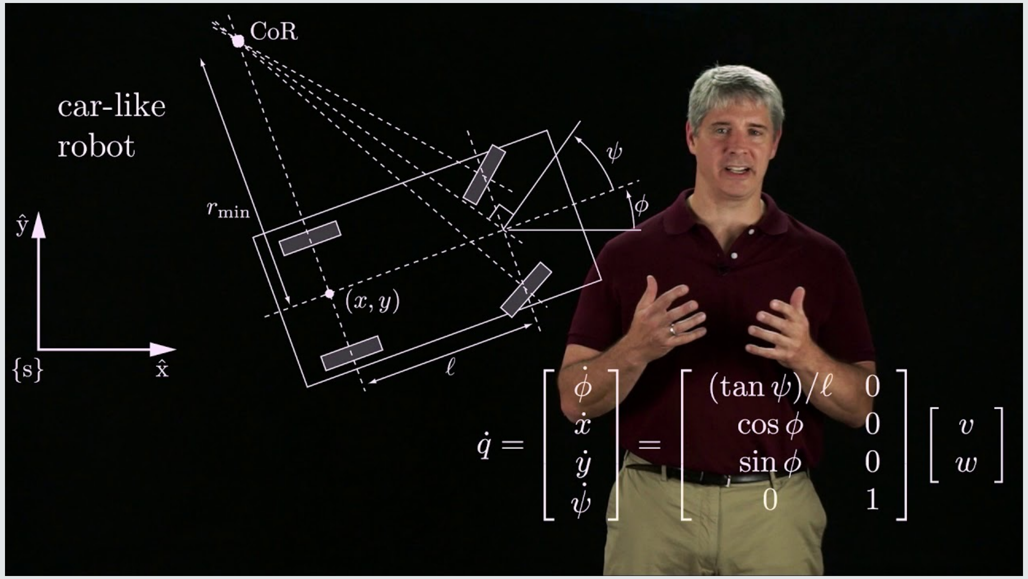


Kappler et al. *Real-time Perception meets Reactive Motion Generation. RA-L + ICRA'18. Finalist 2018 Amazon Best Systems Paper*

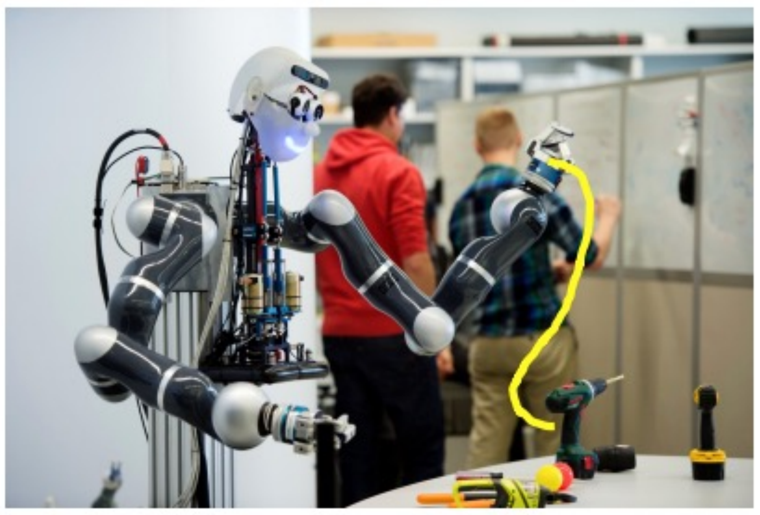

ONE WAY
STOP

Mobile Robot Kinematics

- Aim
  - Understand motion constraints
  - Learn about basic motion models for wheeled vehicles
  - Gain insights for motion control



Motion Planning and Control



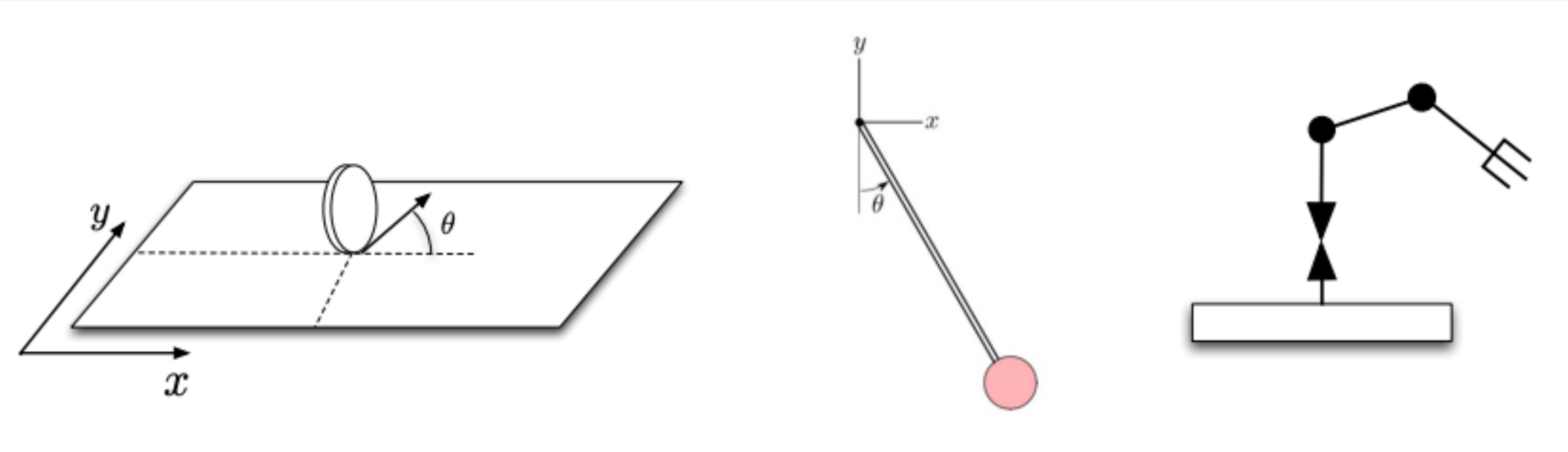
Constraint in Motion Planning and Control

https://tenor.com/view/parallel-park-parking-proper-gif-13789379

Futurama – Put Your Head on My Shoulders [S02E10] : https://getyarn.io/tv-series/6e5aa01e-9637-11e7-a34c-42010af00cf6

Generalized Coordinates

- Let $\xi = [\xi_1, \ldots, \xi_n]^T$ denote the configuration of a robot (e.g., $\xi = [x, y, \theta]^T$ for a wheeled mobile robot)

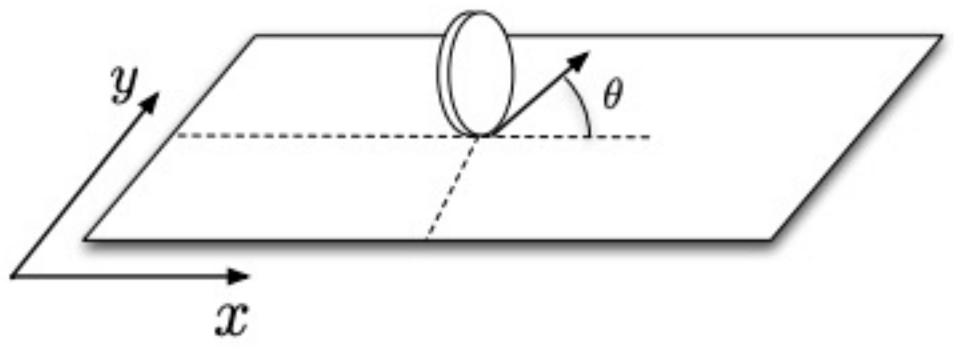


Kinematics Constraints

$$a_i(\xi, \dot{\xi}) = 0, \quad i = 1, \ldots, k < n$$

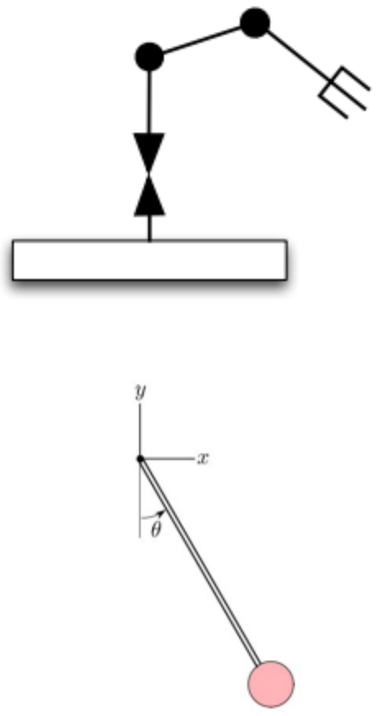
- Constrain the instantaneous admissible motion of the mechanical system
- Generally expressed in Pfaffian form, i.e., linear in the generalized velocities

$$a_i^T(\xi)\dot{\xi} = 0, \quad i = 1, \ldots, k < n$$



Holonomic Constraints

- $h_i(\xi) = 0$, for $i = 1, \ldots, k < n$
- Reduce space of accessible configurations to an $n - k$ dimensional subset
- If all constraints are holonomic, the mechanical system is called holonomic
- Generally, the result of mechanical interconnections



Examples of Holonomic Constraints

Xiang, Qin, Mo et al., “SAPIEN: A SimulAted Part-based Interactive ENvironment”, CVPR 2020

Kinematics Constraints

$$a_i(\xi,\dot{\xi}) = 0, \quad i = 1, \dots, k < n$$

- Constrain the instantaneous admissible motion of the mechanical system
- Generally expressed in Pfaffian form, i.e., linear in the generalized velocities

$$a_i^T(\xi)\dot{\xi} = 0, \quad i = 1, \dots, k < n$$

- k holonomic constraints imply the existence of an equal number of kinematic constraints

$$\frac{d\,hi(\xi)}{dt} = \frac{\partial\,hi(\xi)}{\partial \xi}\dot{\xi} = \; = 0, \quad i = 1, \dots, k < n$$

- However, the converse is not true in general…

Nonholonomic Constraints

- If a kinematic constraint is not integrable in the form $h_i(\xi) = 0$, then it is said *nonholonomic* -> nonholonomic mechanical system
- Nonholonomic constraints reduce mobility in a completely different way. Consider a single Pfaffian constraint

$$a^T(\xi)\dot{\xi} = 0$$

- Holonomic
  - Can be integrated to $h(\xi) = 0$
  - Loss of accessibility, motion constrained to a level surface of dimension $n - 1$
- Nonholonomic
  - Velocities constrained to belong to a subspace of dimension n - 1, the null space of $a^T(\xi)$
  - No loss of accessibility

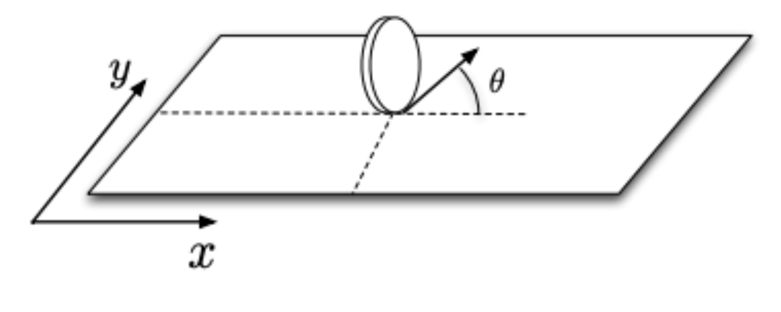
Example of Nonholonomic Constraints

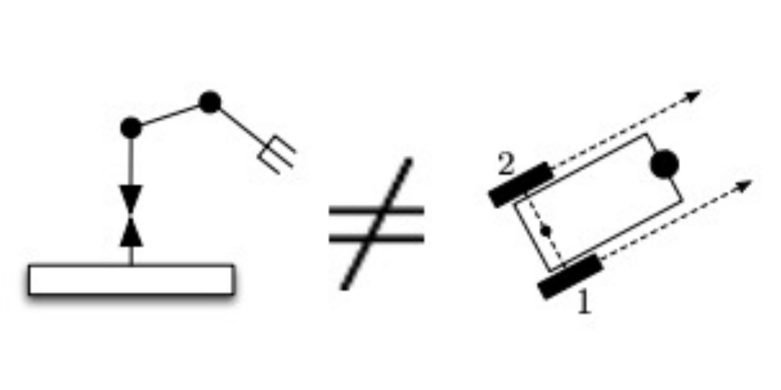
- **System:** disk that rolls without slipping
  $\xi = [x, y, \theta]^T$

- No side slip constraints

$$[\dot{x}, \dot{y}] \cdot \begin{bmatrix} \sin\theta \\ -\cos\theta \end{bmatrix} = \dot{x}\sin\theta - \dot{y}\cos\theta = [\sin\theta, -\cos\theta, 0]\,\dot{\xi} = 0$$

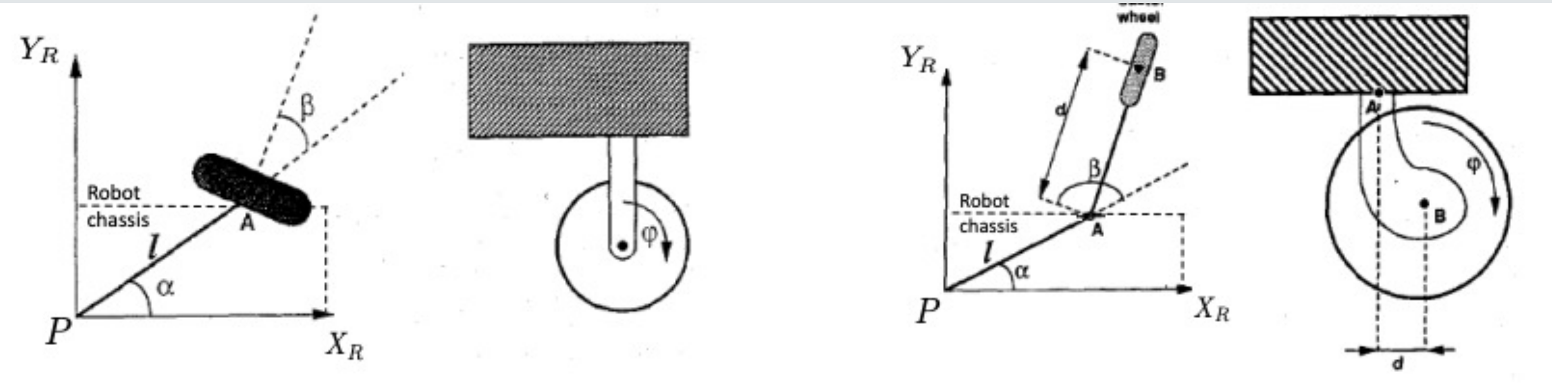
- **Facts:**
  - No loss of accessibility
  - Wheeled vehicles are generally nonholonomic





Types of Wheels

- Standard wheels (four types)



Standard wheel -- fixed or steerable

Standard, off-centered wheel (caster)
-- passive or active

- **Special wheels:** achieve omnidirectional motion (e.g., Swedish or spherical wheels)

Kinematic models

- Assume the motion of a system is subject to *k* Pfaffian constraints

$$\begin{bmatrix} a_1^T(\xi) \\ \vdots \\ a_k^T(\xi) \end{bmatrix} \dot{\xi} := A^T(\xi)\,\dot{\xi} = 0$$

- Then, the admissible velocities at each configuration ξ belong to the (n − k)-dimensional null space of matrix $A^T(\xi)$
- Denoting by $\{g_1(\xi), \ldots, g_{n-k}(\xi)\}$ a basis of the null space of $A^T(\xi)$, admissible trajectories can be characterized as solutions to

$$\boxed{\dot{\xi} = \sum_{j=1}^{n-k} g_j(\xi) u_j = G(\xi)u}$$

Input vector

Example : Unicycle

- Consider pure rolling constraint for the wheel:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = [\sin\theta, -\ \cos\theta, 0]\,\dot{\xi} = a^T(\xi)\,\dot{\xi} = 0$$

- Consider the matrix

$$G(\xi) = [g_1(\xi), g_2(\xi)] = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}$$

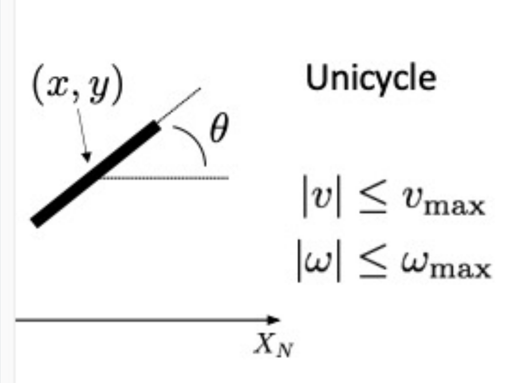
where $[g_1(\xi), g_2(\xi)]$ is a basis of the null space of $a^T(\xi)$

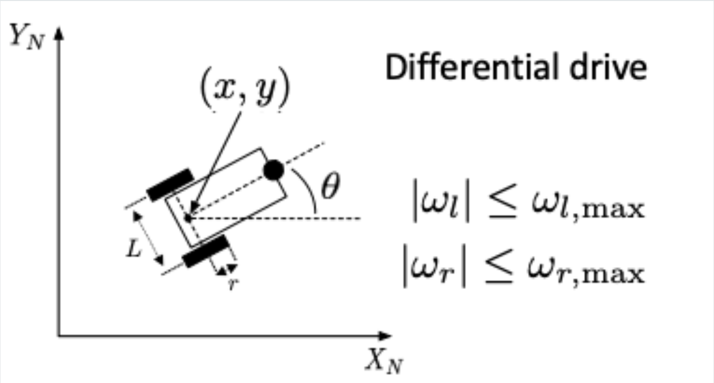
- All admissible velocities are therefore obtained as linear combination of $g_1(\xi)$, and $g_2(\xi)$

Unicycle and differential drive models

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} \nu + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega$$

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \frac{r}{2}(\omega_l + \omega_r)\cos\theta \\ \frac{r}{2}(\omega_l + \omega_r)\sin\theta \\ \frac{r}{L}(\omega_r - \omega_l) \end{pmatrix}$$

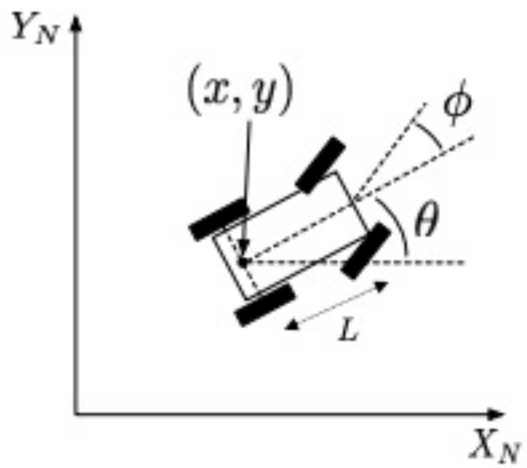




The kinematic model of the unicycle also applies to the differential drive vehicle, via the one-to-one input mappings: $\nu = \frac{r}{2}(\omega_\mathrm{r} + \omega_\mathrm{l})$ $\omega = \frac{r}{L}(\omega_\mathrm{r} - \omega_\mathrm{l})$

Simplified Car Model

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \upsilon \cos\theta \\ \upsilon \sin\theta \\ \frac{\upsilon}{L} \tan\phi \end{pmatrix}$$



Simple car

$$|v| \leq v_{\max}, \ |\phi| \leq \phi_{\max} < \frac{\pi}{2}$$ Simple car model

$$v \in \{-v_{\max}, v_{\max}\}, \ |\phi| \leq \phi_{\max} < \frac{\pi}{2}$$ Reeds&Shepp's car

$$v = v_{\max}, \ |\phi| \leq \phi_{\max} < \frac{\pi}{2}$$ Dubins' car

References: (1) J.-P. Laumond. Robot Motion Planning and Control. 1998. (2) S. LaValle. Planning algorithms, 2006.

From kinematic to dynamic models

- A kinematic state space model should be interpreted only as a subsystem of a more general dynamical model
- Improvements to the previous kinematic models can be made by placing **integrators** in front of action variables
- For example, for the unicycle model, one can set the speed as the integration of an action a representing acceleration, that is

$$\dot{x} = v \cos\theta, \quad \dot{y} = v \sin\theta, \quad \dot{\theta} = \omega, \quad \dot{v} = a$$

Aim

Understand motion constraints

Learn about basic motion models for wheeled vehicles

Gain insights for motion control

Readings

Please read the mandatory chapter and explore more in other books

Your Home Work