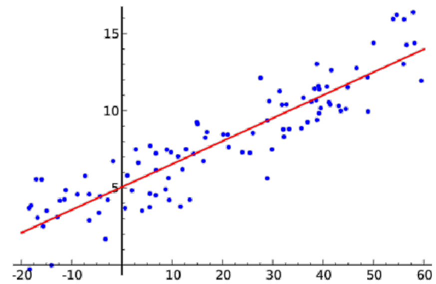
CMPT 419/983

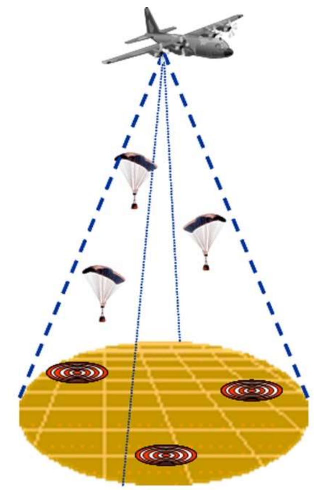
Robotic Autonomy: Algorithms and Computation

Mo Chen

https://www.sfu.ca/~mochen

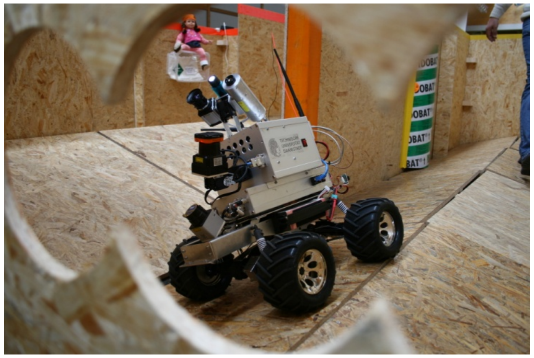
Course Outline





- Overview of algorithms used for robotic decision making
  - Theory-focused
  - Fundamentals for doing many areas of robotics research

- Dynamical systems
- Optimization and optimal control
- Machine learning in robotics
- Localization and mapping



Logistics

- Academic Quadrangle 5601,
  - Mondays 10:30-12:20
  - Wednesdays 10:30-11:20
- Office hour: Wednesdays 13:00-14:30, TASC 1 8225
- Course website: https://coursys.sfu.ca/2019fa-cmpt-419-x1/pages/
- Contact:
  - mochen@cs.sfu.ca
  - shubams@sfu.ca

Caveats

- This class is in “experimental mode”
- Slight changes are expected
- Some things may not be super polished
- Please provide feedback and comments

Grading

- 40% Homework
  - 3 assignments

- 60% Project

Project suggestions

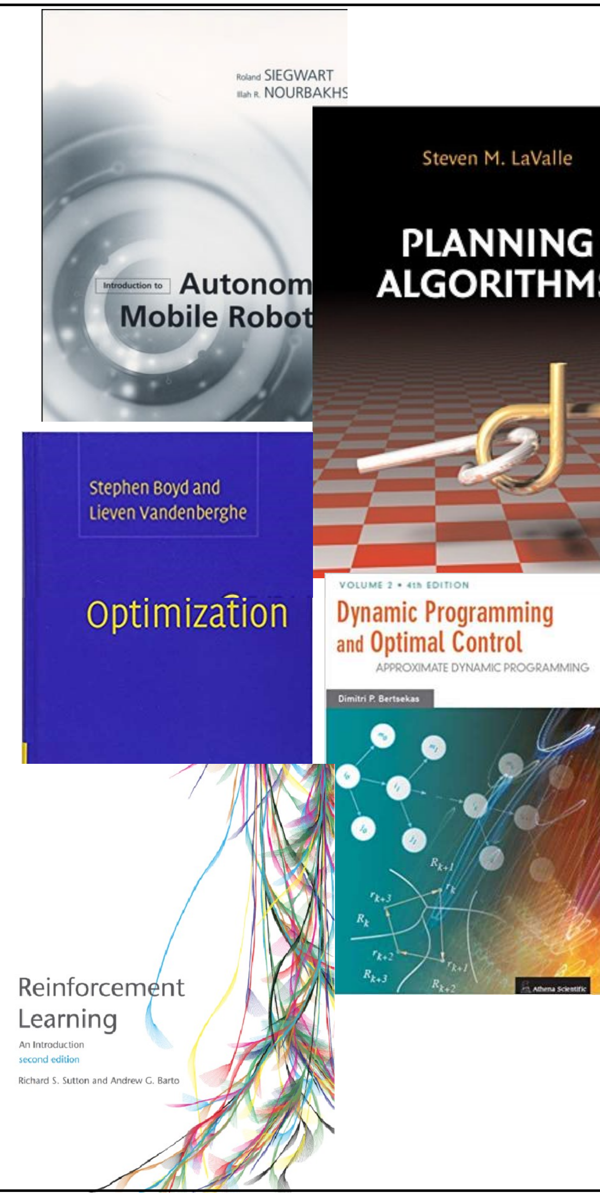
- Thoroughly understand and critically evaluate 3 to 5 papers in an area covered in this course
- Reproduce the results of 1 to 2 papers in an area covered in this course, and suggest or make improvements
- Mini Research project related to an area covered in this course
- Other: Please consult instructor

Project timeline

- Proposal (1-2 paragraphs)
  - Due Oct. 7
- Poster session
  - Last lecture of the term, Dec. 2
- Report (6 pages maximum)
  - Due Dec. 2

Recommended textbooks

- R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. The MIT Press, 2011, 9780262015356.
- S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006, 9780521862059.
- S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2008, 9780521833783.
- D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2017, 1886529434.
- R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 1998, 9780262257053.



Dynamical systems

- Mathematical models of robotic systems
  - Deterministic vs. stochastic
  - Continuous vs. discrete time

- Configuration of system described by the state, often denoted x
  - State changes, or evolves, according to the model

- Deterministic, continuous time
  - $\frac{dx}{dt} = \dot{x}(t) = f(x, u)$
- Stochastic, discrete time
  - x_{k+1} obtained from the probability distribution $p(x_{k+1}|x_k, u_k)$

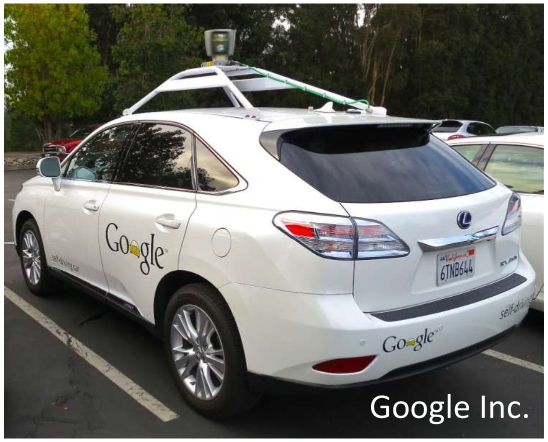
System State

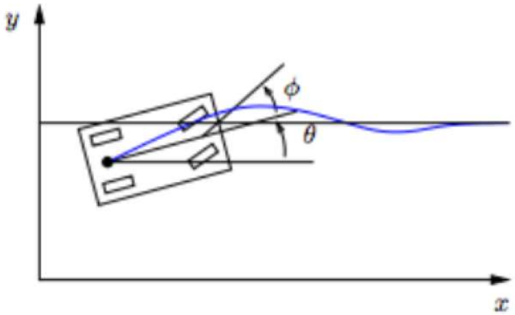
- Defined in terms of any variables of interest
  - Often denoted $x(t)$ or x_k

- Position
- Heading

- Velocity
- Angular velocity

- Voltages, concentrations of chemicals
- Human comfort, degree of trust

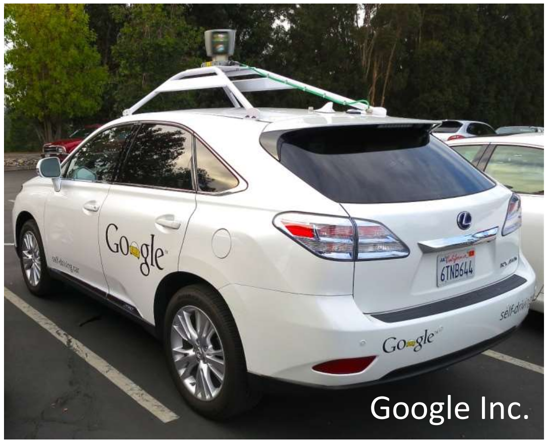




Control and disturbance

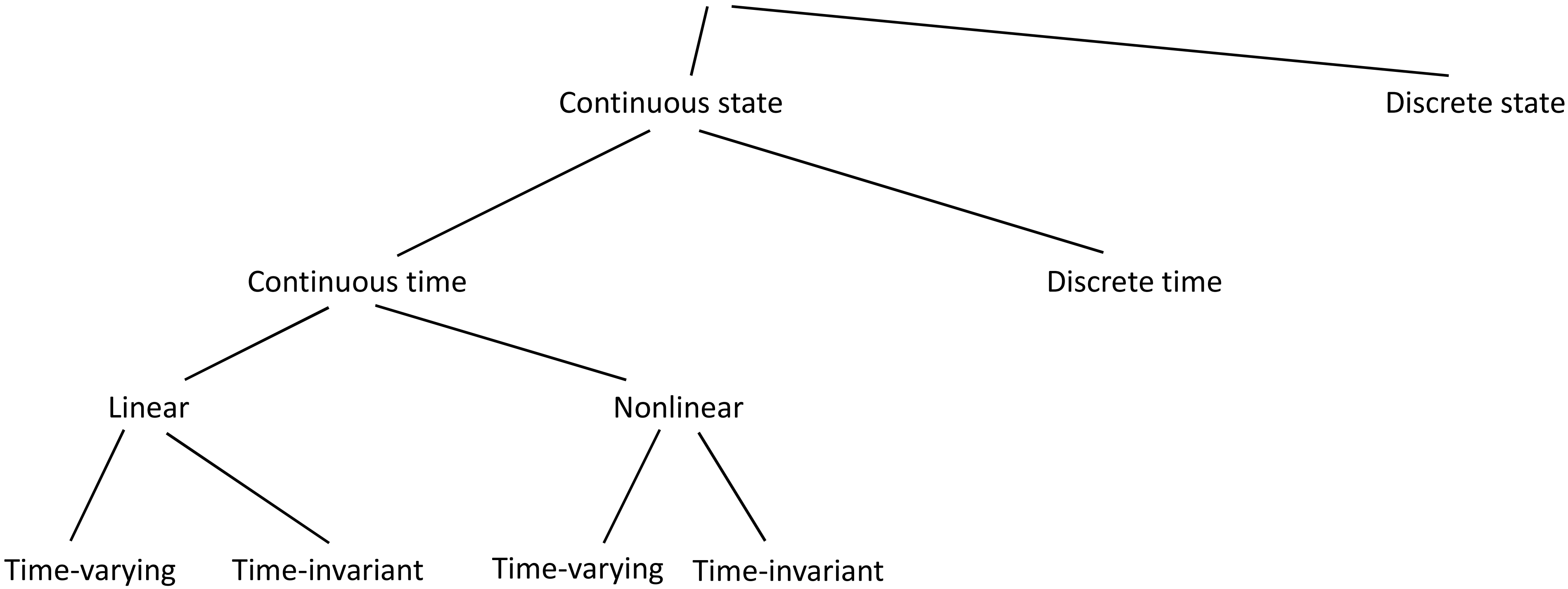
- Control/action : usually used to achieve a desired goal
  - Usually denoted $u(t)$ or u_k
  - Acceleration
  - Turn rate

  - Gas throttle
  - Steering wheel angle

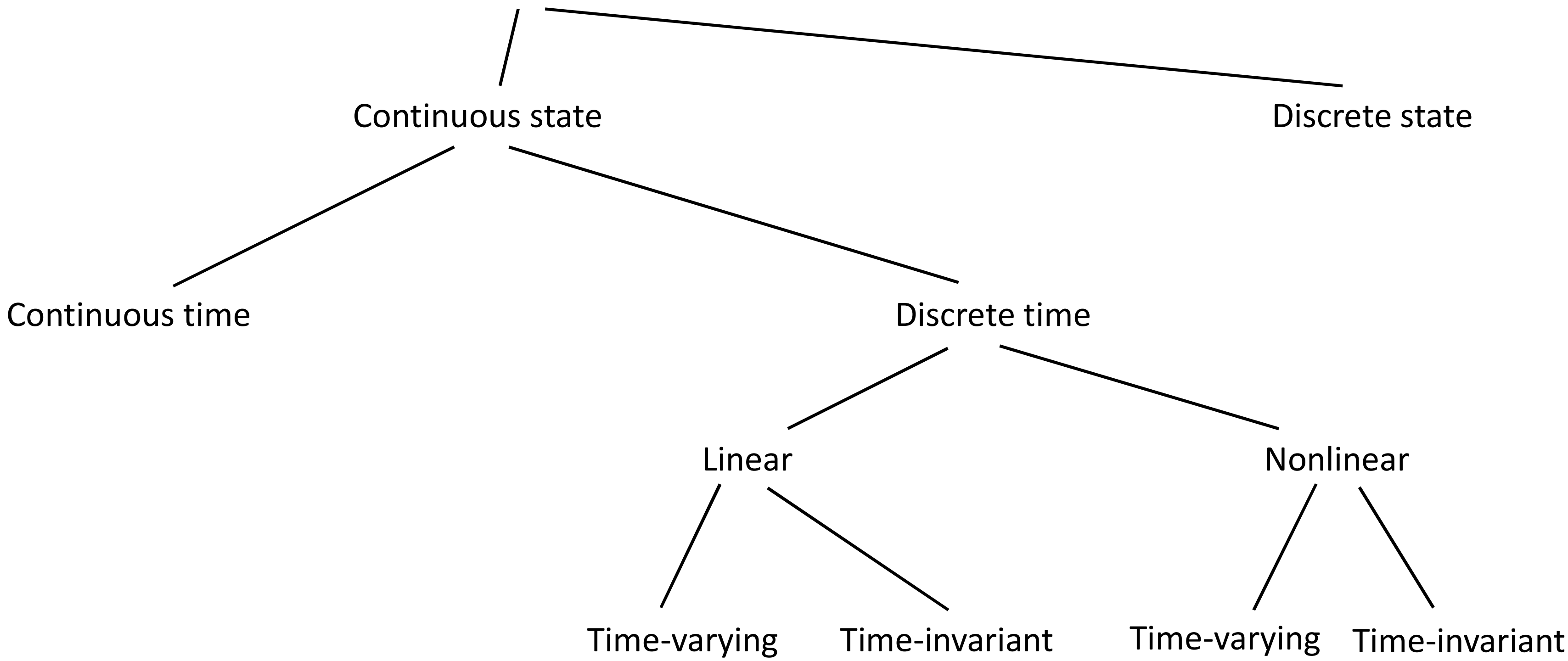


- Disturbance
  - Usually denoted $d(t)$ or d_k
  - Bumps on the road
  - Input noise

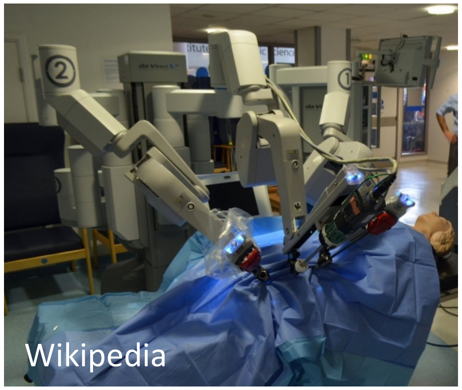
Mathematical models of robotic systems

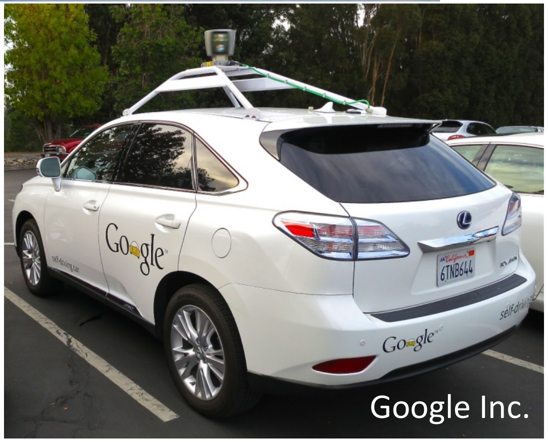


Mathematical models of robotic systems

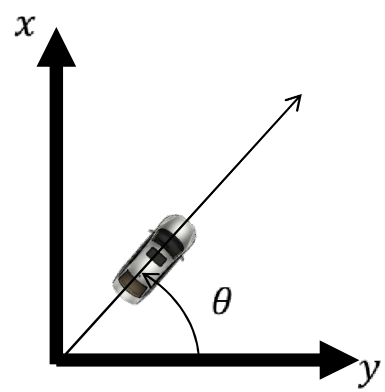


Examples of Robotic Systems





Car models



①

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

States: (x, y, θ); position and heading
Control: ω; turn rate (angular speed)

②

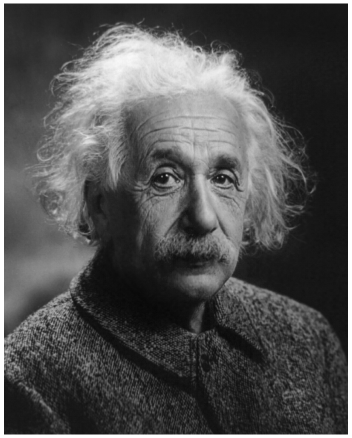
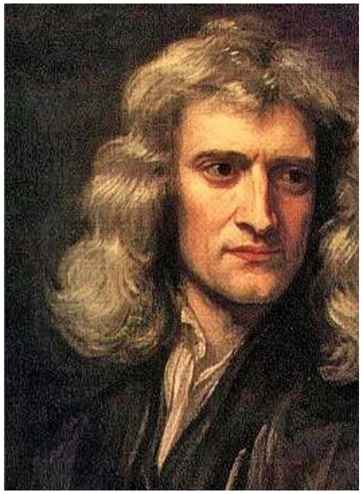
$$\dot{x} = v\cos\theta$$
$$\dot{y} = v\sin\theta$$
$$\dot{\theta} = \omega$$
$$\dot{v} = a$$
$$\dot{\omega} = \alpha$$

States: $(x, y, \theta, v, \omega)$;
position, heading, speed, turn rate
Control: (a, α);
acceleration, angular acceleration

③ Bicycle model

$$\dot{x} = v_x$$
$$\dot{y} = v_y$$
$$\dot{v}_x = \omega v_y + a_x$$
$$\dot{v}_y = -\omega v_x + \frac{2}{m}\left(F_{c,f}\cos\delta_f + F_{c,r}\right)$$
$$\dot{\psi} = \omega$$
$$\dot{\omega} = \frac{2}{I_z}\left(l_f F_{c,f} - l_r F_{c,r}\right)$$
$$\dot{X} = v_x\cos\psi - v_y\sin\psi$$
$$\dot{Y} = v_x\sin\psi + v_y\cos\psi$$

Models

- All models are wrong; some are useful
- Definition of “useful” depends on situation
  - Simulation
  - Analysis and control
  - Verification
- Considerations
  - Does the model capture the desired system behaviours
  - Is the model amenable to tractable computation

Nonlinear Optimization

- Choose x to minimize some cost, subject to constraints

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, i = 1, \dots, n \\ & h_j(x) = 0, j = 1, \dots, m \end{aligned}$$

Fuel cost, distance to obstacles, distance from goal, prediction error in machine learning

System dynamics, obstacle avoidance, goal reaching

  - Equivalently, $\text{maximize} \; -f(x)$: Maximize reward, maximize profit

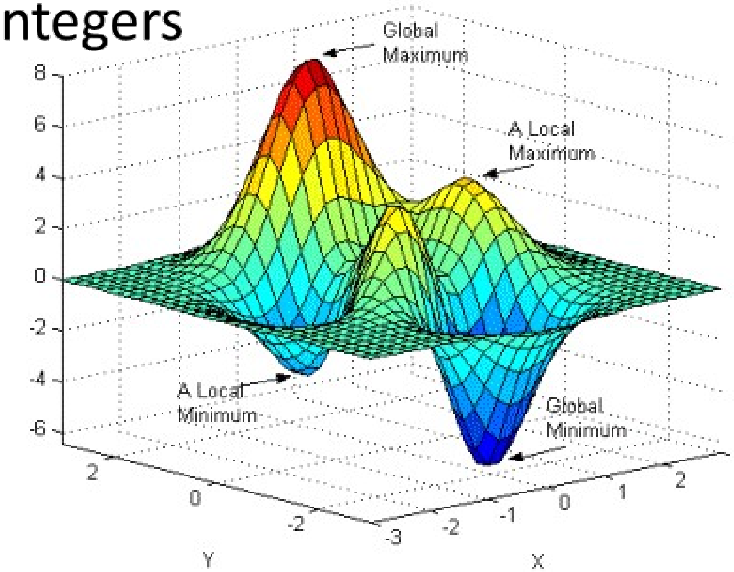
- Robotics spans many fields
  - Many conventions
  - Many notations clashes

Nonlinear Optimization

- A very difficult problem in general for $x \in \mathbb{R}^n$ where n is large
  - Calculus facts: necessary and sufficient conditions
  - Rely on gradients (if possible)

- Sometimes, some components of x may be integers
  - Can we do better than brute force?

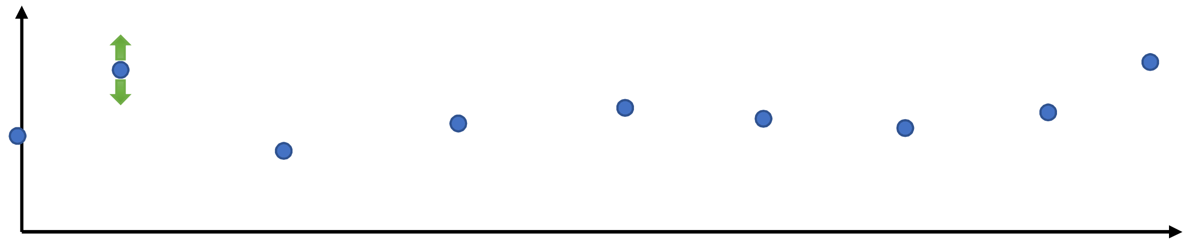
- Simpler cases
  - Differentiable functions
  - Linear, convex, quasiconvex
  - Unconstrained problems



Nonlinear Optimization

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, i = 1, \dots, n \\ & h_j(x) = 0, j = 1, \dots, m \end{aligned}$$

- Nonlinear optimization:
  - Decision variable is $x \in \mathbb{R}^n$



Optimal Control

$$\underset{u(\cdot)}{\text{minimize}}\ \underbrace{l\big(x(t_f),t_f\big)}_{\text{Final cost}} + \underbrace{\int_0^{t_f} c(x(t),u(t),t)dt}_{\text{Running cost}}$$

Cost functional, $J\big(x(\cdot),u(\cdot)\big)$

$$\text{subject to } \dot{x}(t) = f\big(x(t),u(t)\big)$$

Dynamic model

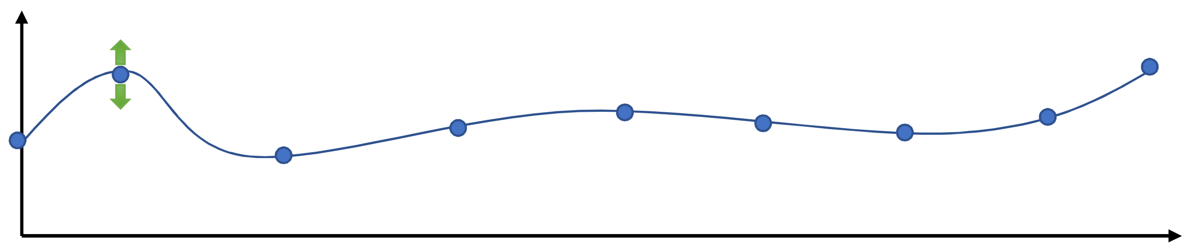
$$g\big(x(t),u(t)\big) \geq 0$$

Additional constraints

- Eg. actuation limits

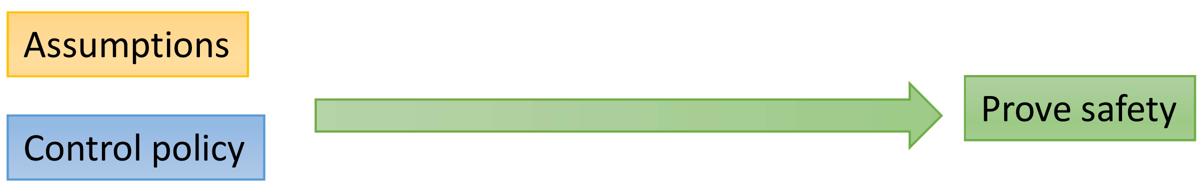
$$x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m, x(0) = x_0$$

- Nonlinear optimization:
  - Decision variable is $x \in \mathbb{R}^n$
- Optimal control:
  - Decision variable is a function $u(\cdot)$



Robotic Safety

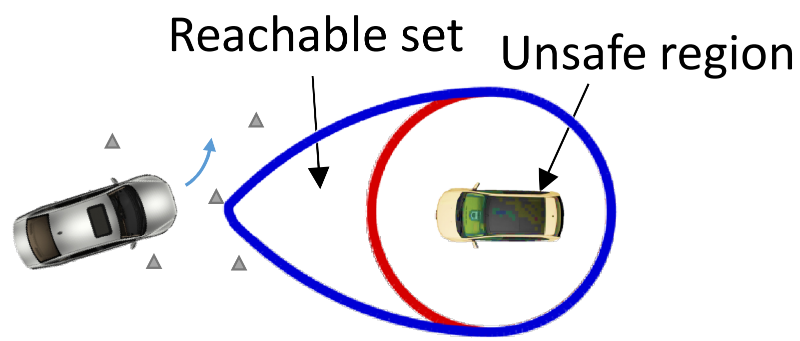
- Verification methods



  - Considers all possible system behaviours, given assumptions

- Can be written as an optimal control problem

Reachability Analysis



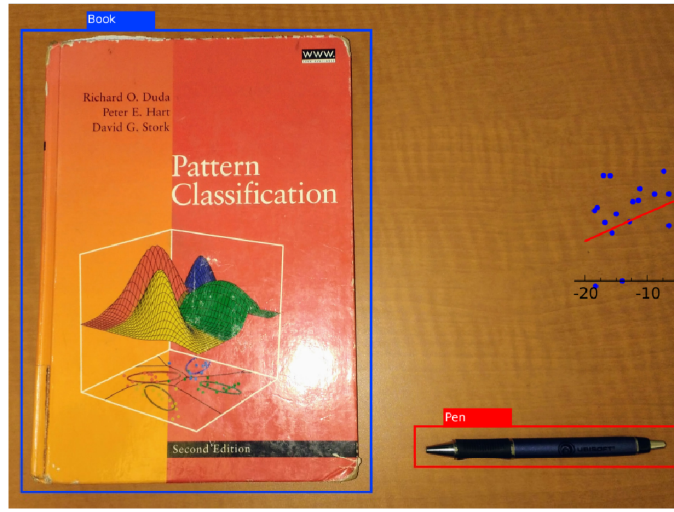
- Model of robot
- Unsafe region

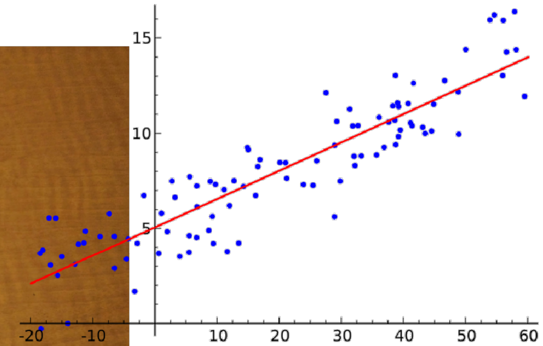
Optimal control policy to avoid danger

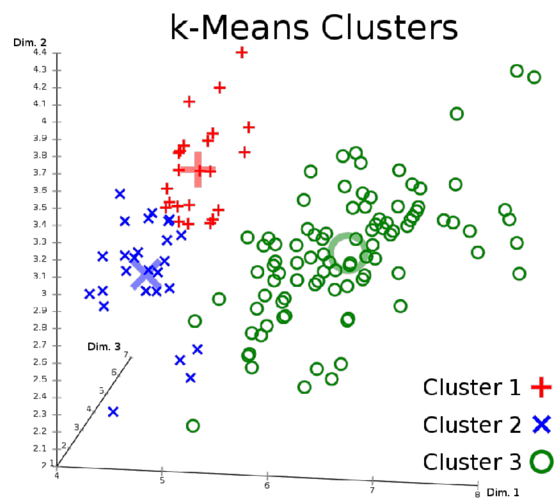
Reachable set (States leading to danger)

Machine Learning

- Application of nonlinear optimization
  - Takes advantage of available data
- Supervised learning
  - Regression
  - Classification
- Unsupervised learning
  - Clustering
  - Reinforcement learning







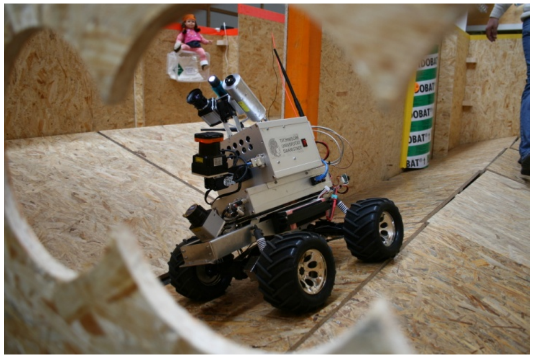
Machine Learning

- Very scalable with additional data
- Requires a lot of data

- Computer vision
- Natural language processing
- Game playing
- Simulated robotics

- Physical robotics?

Localization and Mapping



- Localization
  - Given a map, figure out where the robot is (with respect to the map) using sensor information
  - Continuously do this while moving around in the environment

- Simultaneous localization and mapping
  - Figure out the map and localize at the same time

- Probabilistic models
  - of how the robot moves
  - of how the robot senses the environment

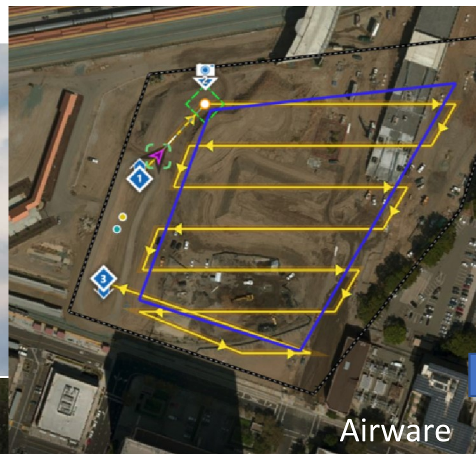
Sample of MARS Research

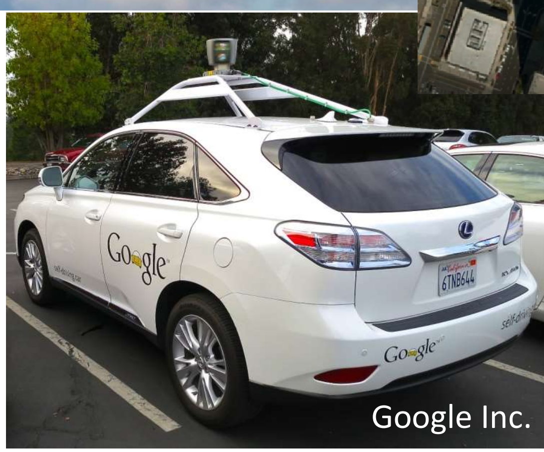
- https://sfumars.com

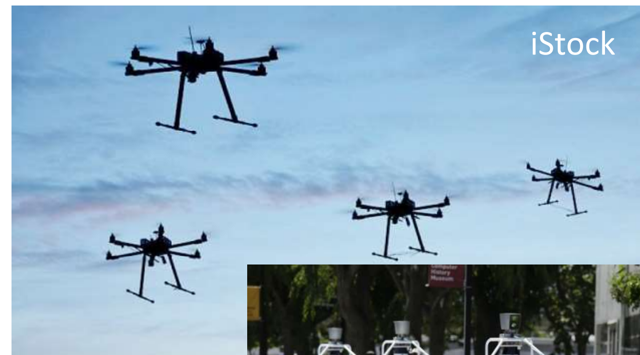
- Control algorithms
- Computational complexity
- Reinforcement learning
- Human intent inference

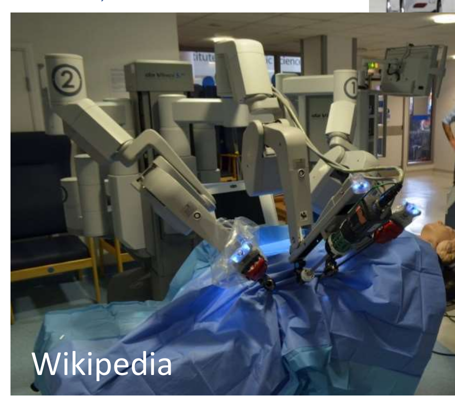
- Theory
- Computation
- Experiments

Safety: A Crucial Perspective in Automation





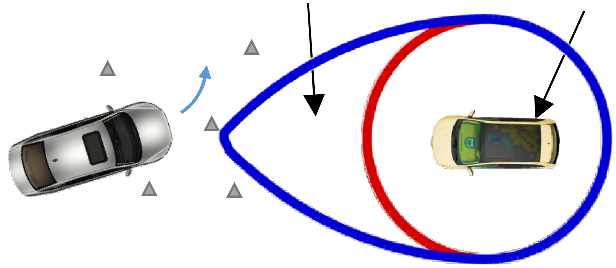




Challenges in Safety-Critical Systems

- Account for all possible system behaviours
- Complex systems
- Complex environment
  - Weather conditions
  - Other robots

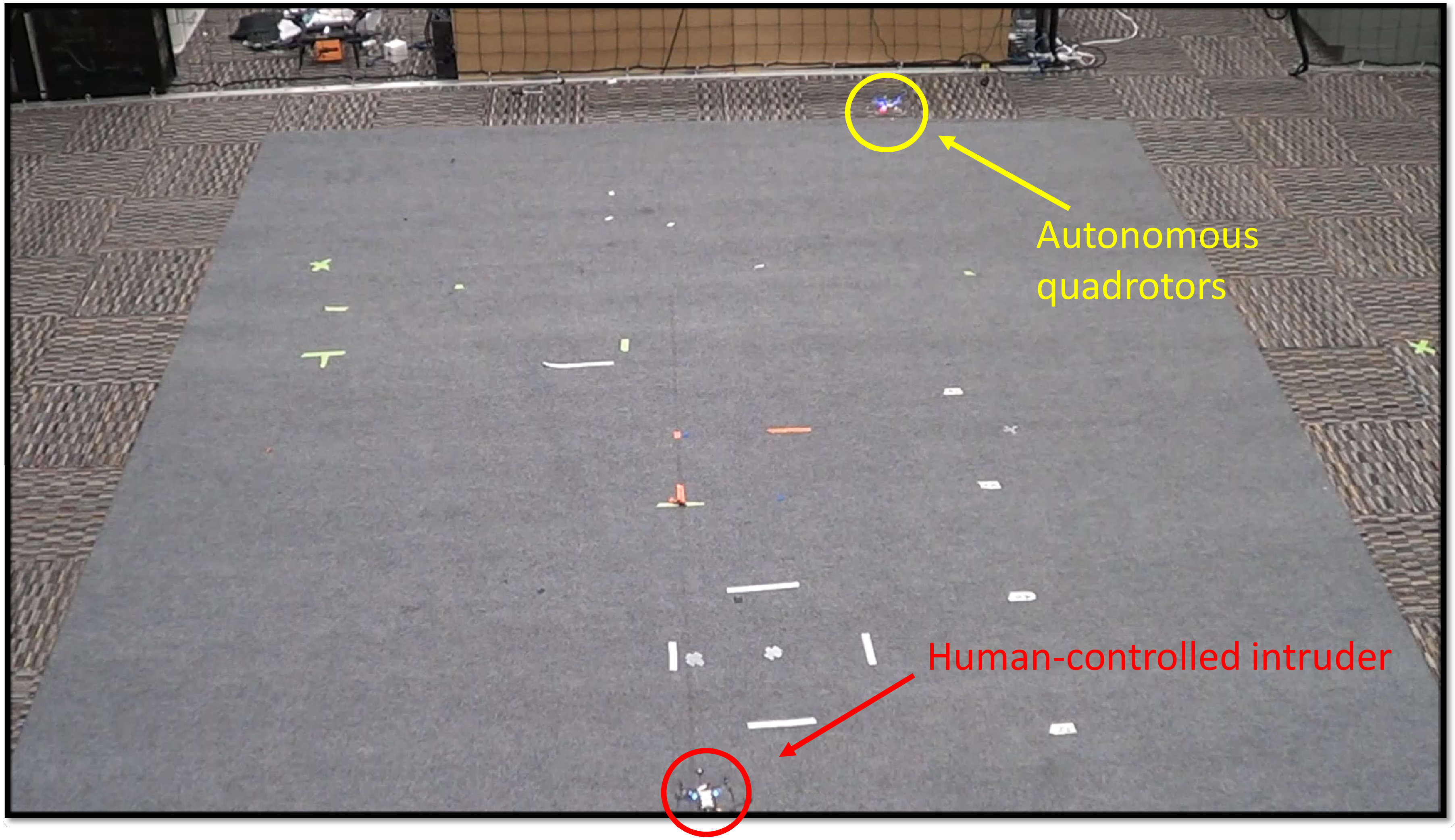
Reachability Analysis



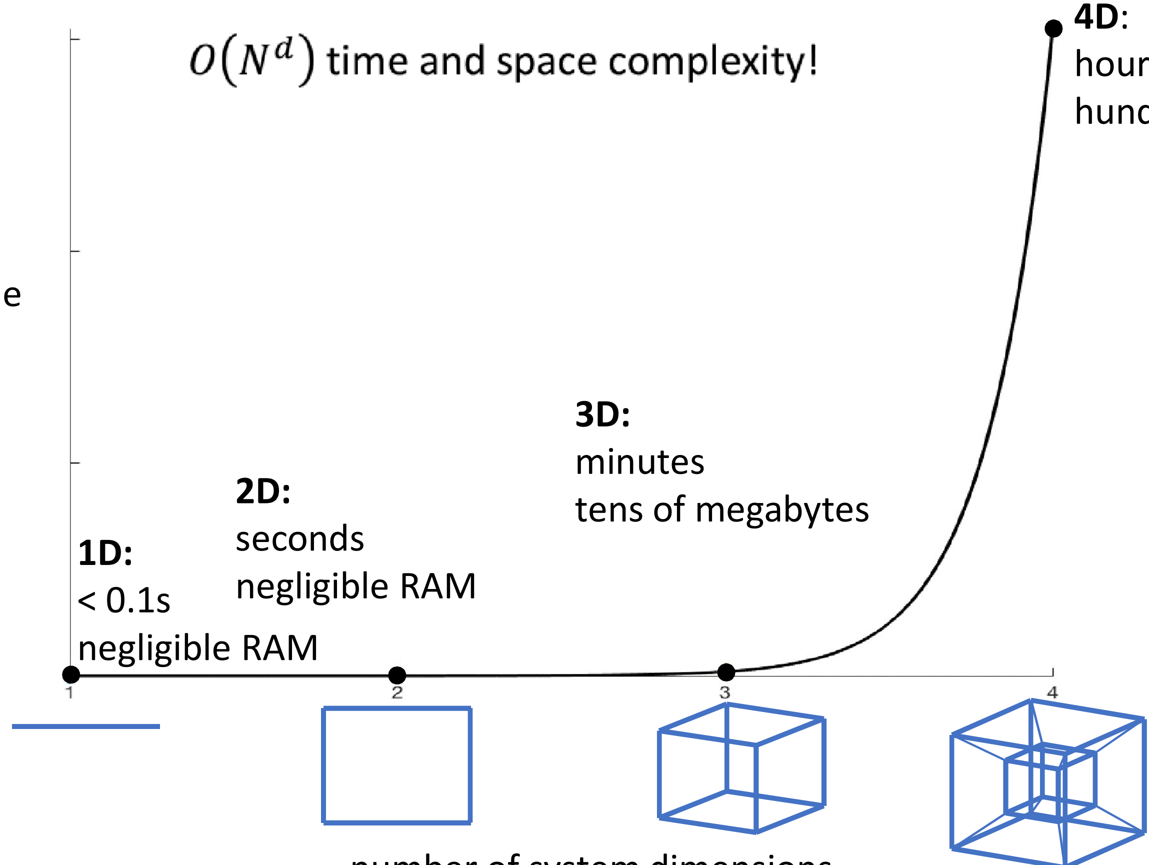
- Model of robot
- Unsafe region

Reachable set (States leading to danger)

Optimal control policy to avoid danger

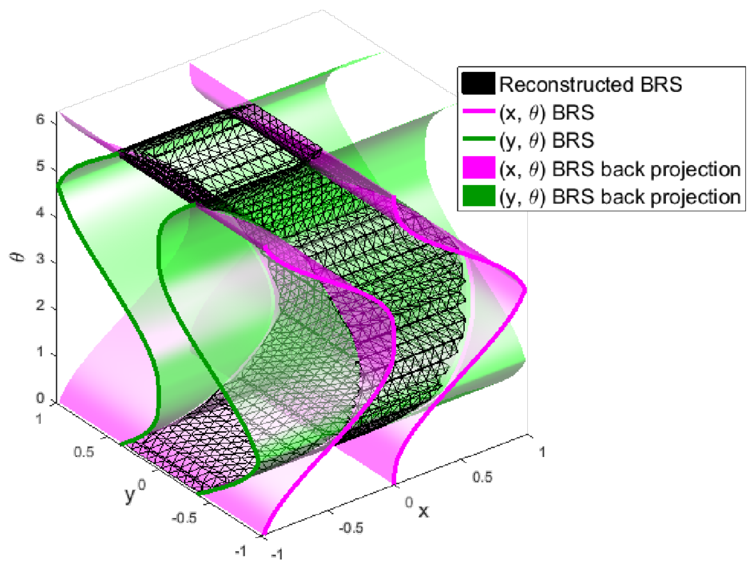

Autonomous
quadrotors
Human-controlled intruder

Main Challenge:
Exponential Computational Complexity with DP



Research Directions

Dimensionality reduction



Self-Contained Subsystems

- Motivating example: Dubins Car

Self-contained subsystem 1

$$\dot{x} = v\cos\theta$$
$$\boxed{\dot{y} = v\sin\theta}$$
$$\dot{\boldsymbol{\theta}} = \boldsymbol{\omega}$$

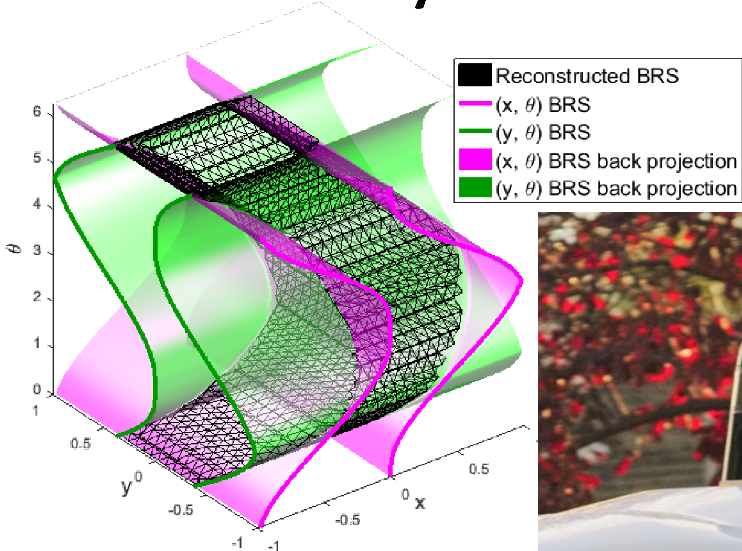
Self-contained subsystem 2

- Subsystems are coupled through state and control

- Many systems have states that are not directly coupled to each other
    - Most common in vehicle dynamics

Research Directions

Dimensionality reduction

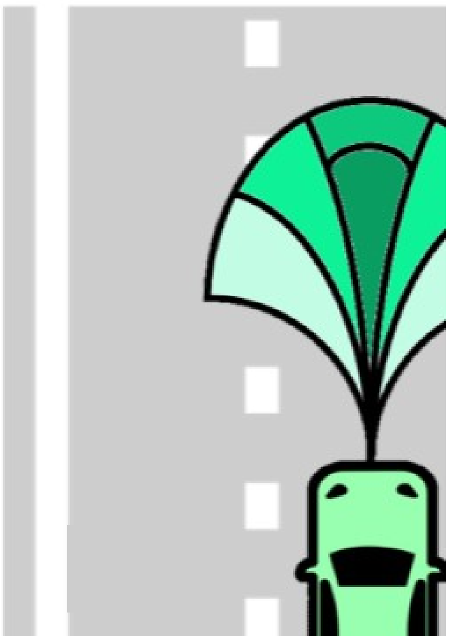


Parallel computing

Perception systems

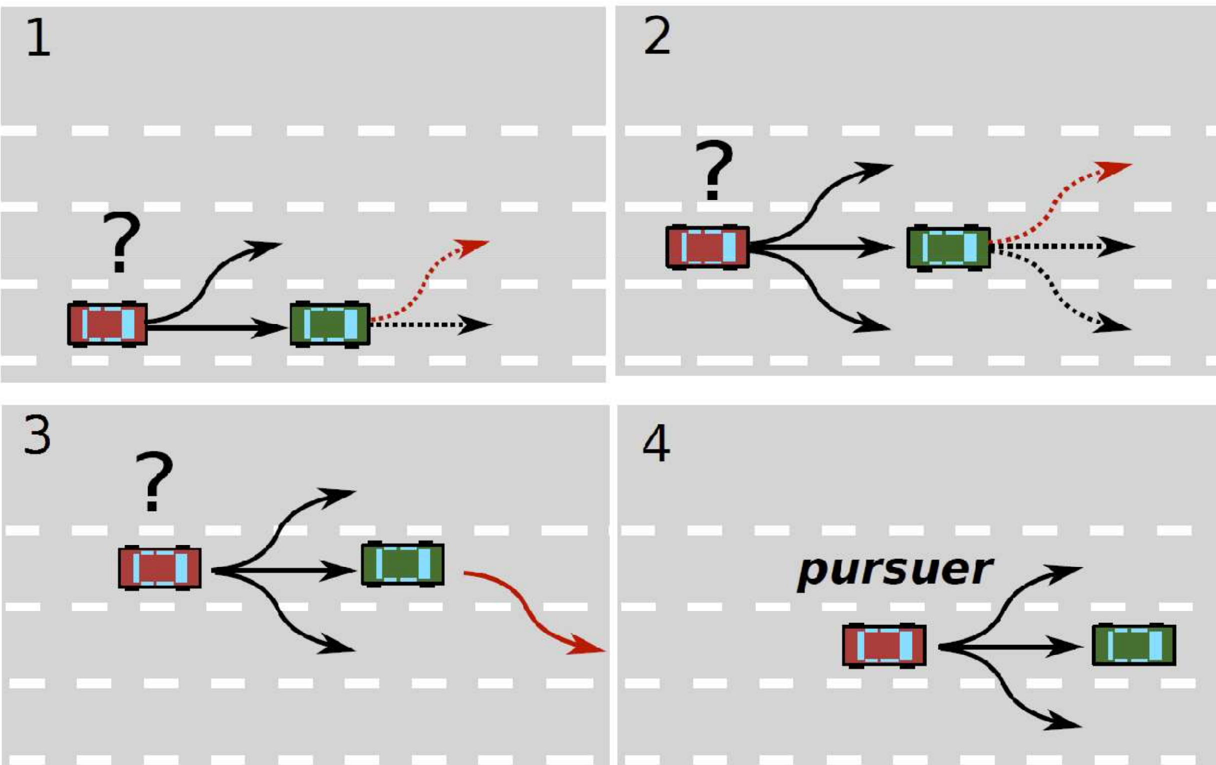
Research Directions

Human intent understanding



Campbell et al.

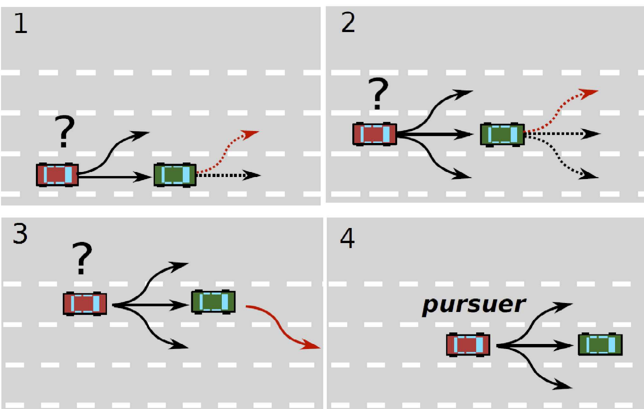
Proactive Human Intent Understanding



Is the red car
- A pursuer,
- Or a benign vehicle?

Robot car (green) *proactively* changes lanes to determine intent

Multi-Modal Human Intent Understanding



Motion

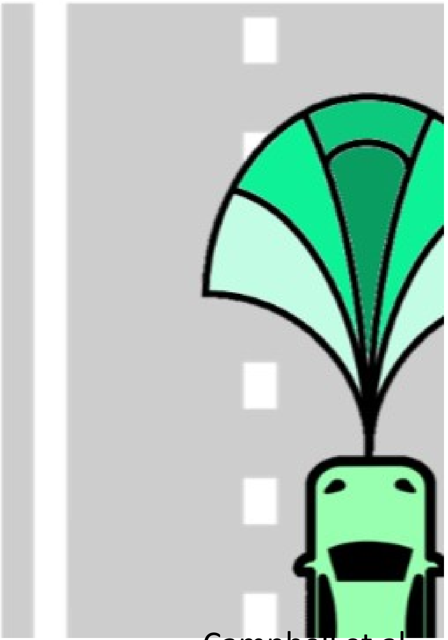
Emotion

Engagement

Audio

Research Directions

Human intent understanding

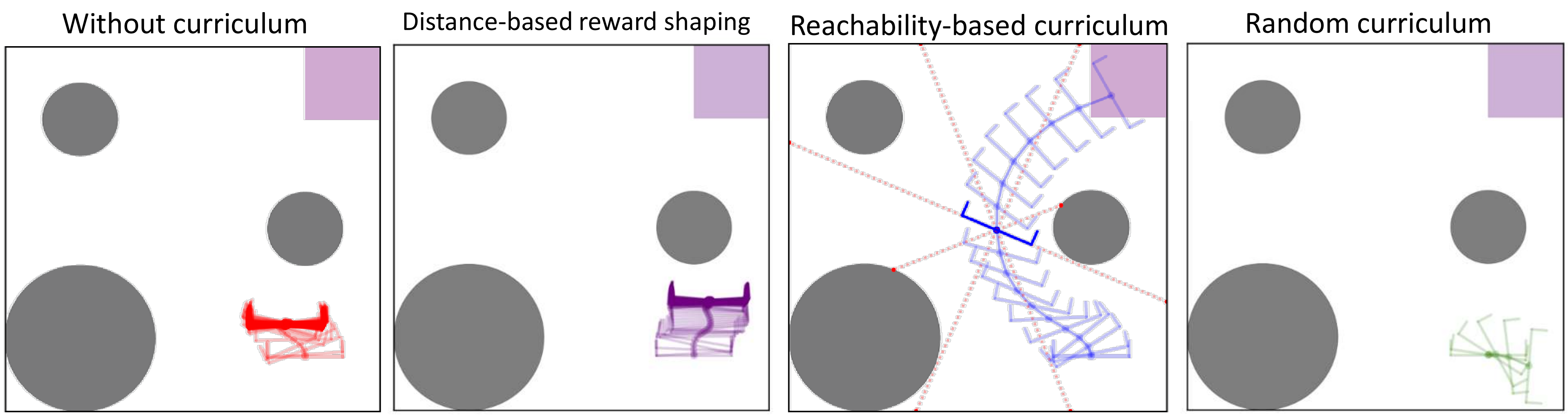


Campbell et al.

Robotic learning

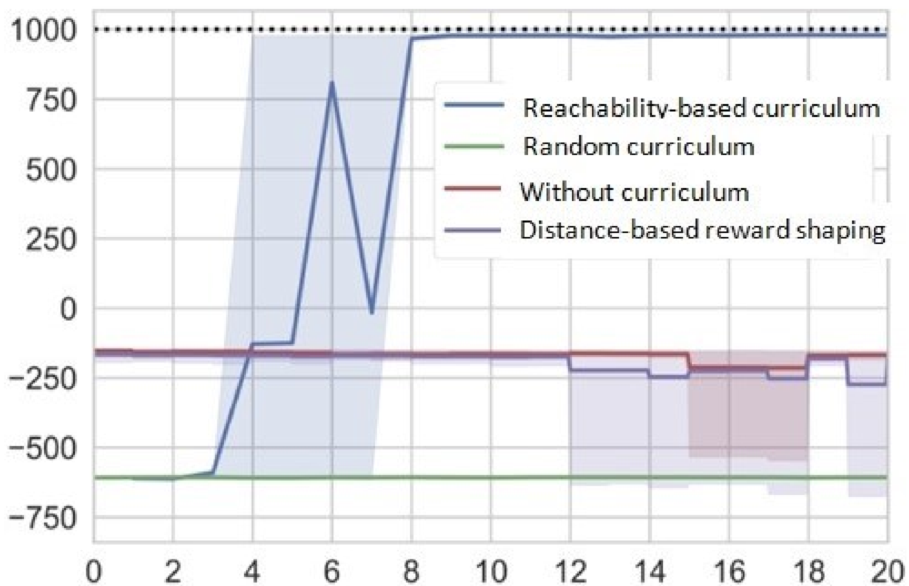
Global Robots Ltd.

Curriculum Reinforcement Learning



Curriculum Reinforcement Learning

Task performance



Curriculum performance