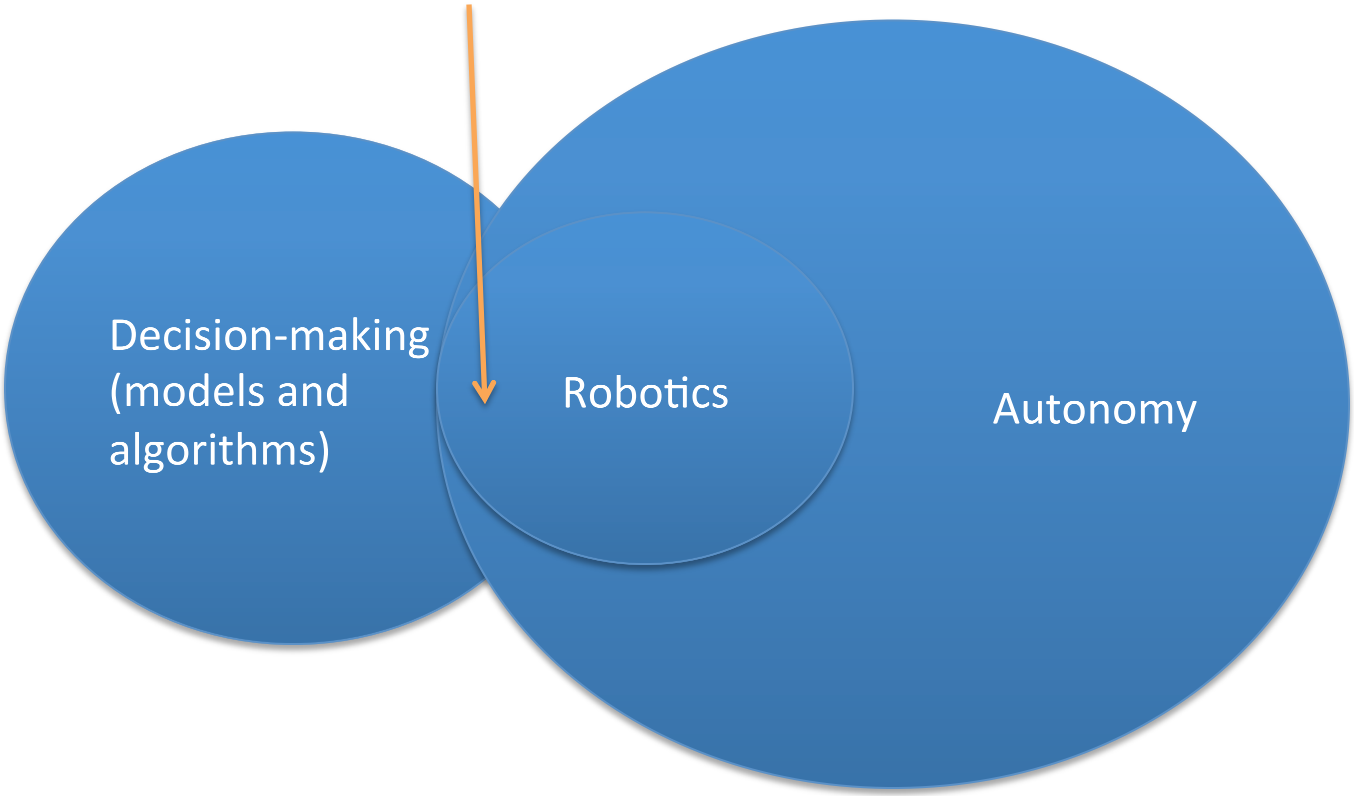
Decision Making
in Robots and Autonomous Agents

Introduction

Subramanian Ramamoorthy
School of Informatics

18 January, 2019

What do the Terms Mean?



What is the Goal of Robotics?

Programming Machines That Work

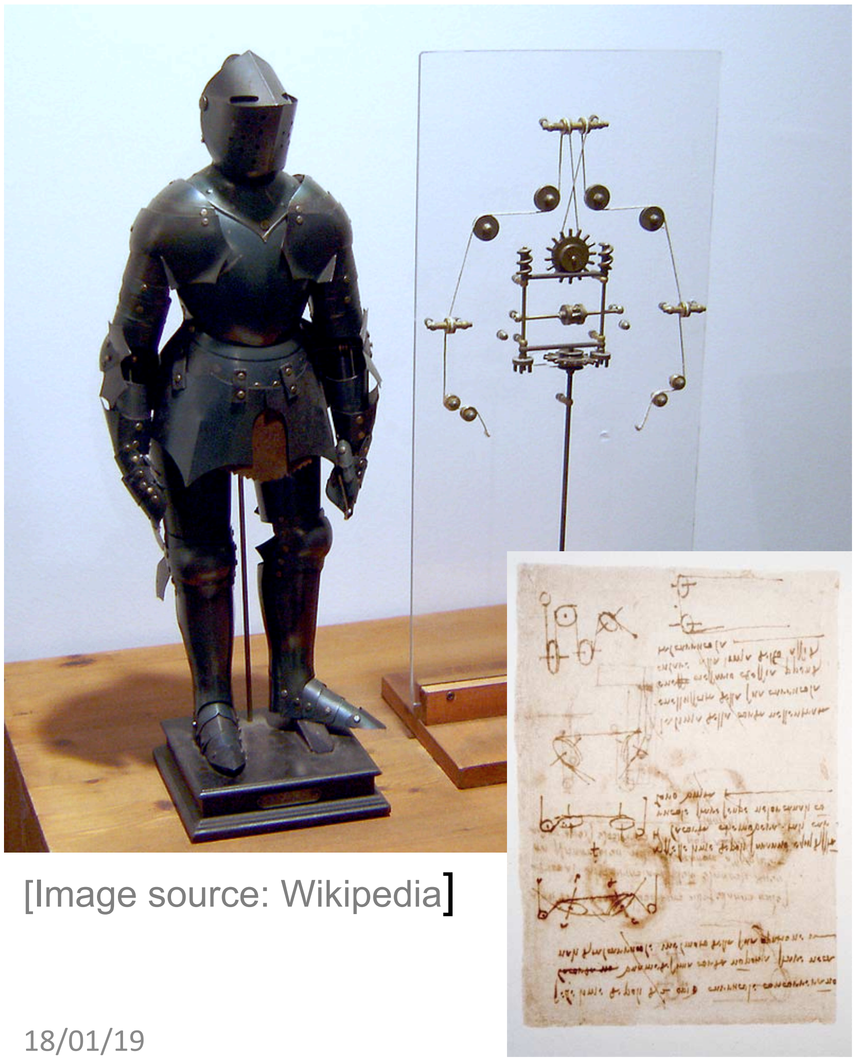
Daniel Koditschek
University of Michigan

Abstract

Robotics is a fledgling discipline concerned with programming work: that is, specifying and controlling the exchange of energy between a machine and its environment. Because our understanding of how to do this is still quite rudimentary, the best progress in the field has come from a mix of inspired building and formal analysis. For more than a decade, my students and I have pursued such an agenda, building robots whose controllers drive the coupled robot-environment state toward a goal set and away from obstacles. The talk reviews our progress to date: what sort of "programs" do we know to build, with what theoretical guarantees, and with what empirical success?

- What do we want the program to do?
- What principles might guide the design of such programs?
- Are there generic structures that can be utilized across domains?

Early Examples of Autonomy?
da Vinci’s Mechanical Knight

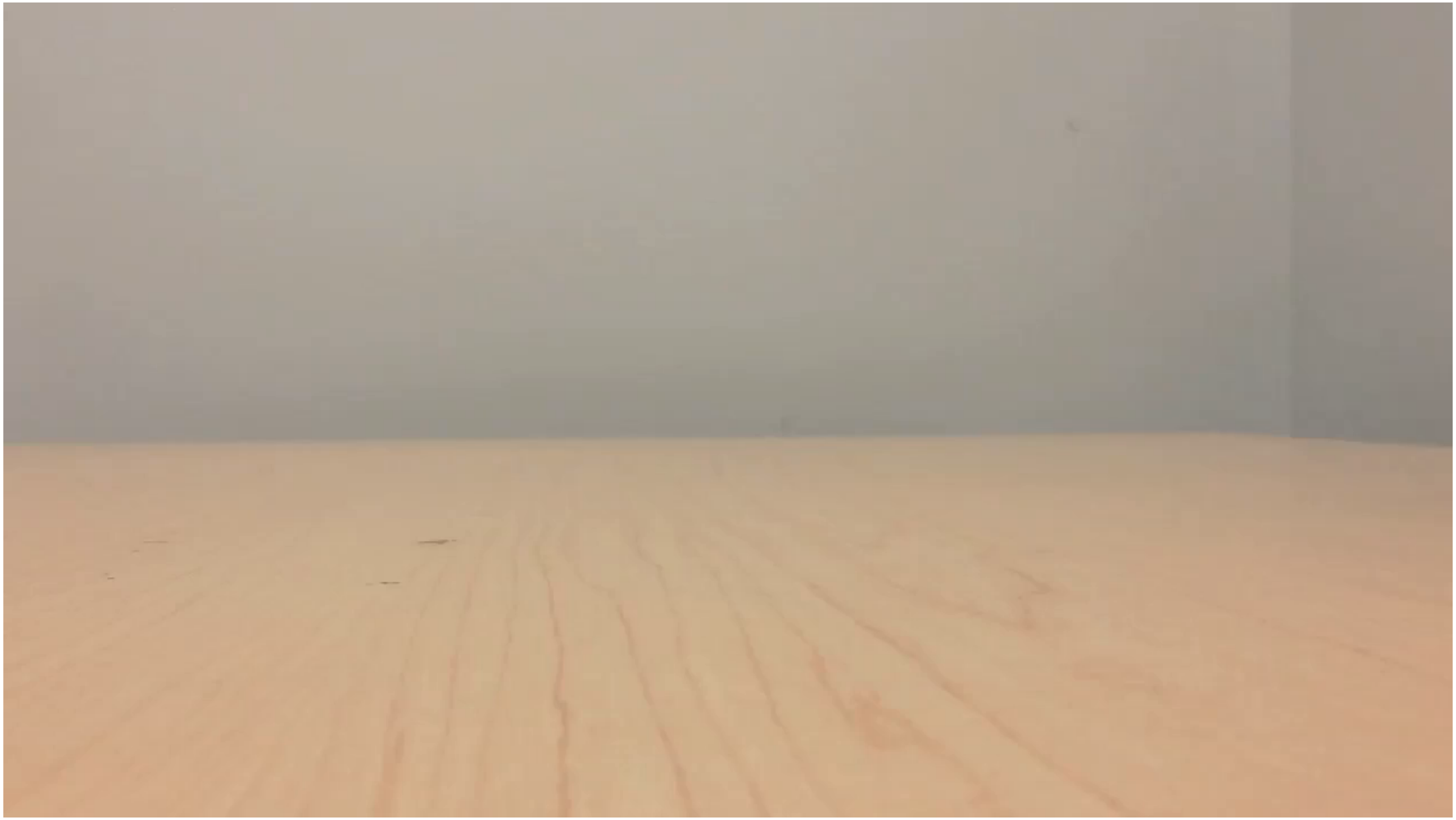


[Image source: Wikipedia]

- Is it a “robot”?
- (What) decisions does this automaton make?

How about this?

How about the Tippe Top?



How about the Tippe Top?

[Image source: Physics Stack Exchange]

- Is this a “robot”?
- This is not a trivial question…

  e.g., passive walkers

[Source: https://www.youtube.com/watch?v=e2Q2Lx8O6Cg]

How about a Marionette?

[Source: https://www.youtube.com/watch?v=bXFPWZSIOs0]

Teleoperation: "Invisible" Puppet Strings?

[Source: https://www.youtube.com/watch?v=fSskyIaWkMk]

Teleoperation in Real Applications

[Source: Intuitive Surgical, Inc. 2008]

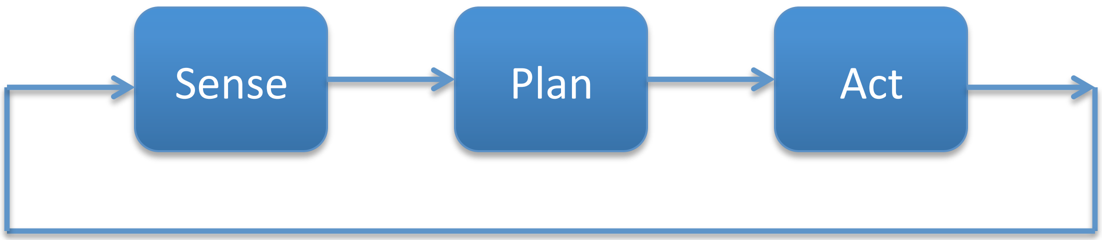
On Robotic Paradigms

- Questions so far may seem like pedantic nitpicking, but they have been at the heart of discussions regarding *paradigms*
- Paradigm: Philosophy or set of assumptions and/or techniques which characterize an approach to a class or problems
  - Rarely is any one paradigm uniquely best for all problems (bit like Cartesian vs. polar coordinates in calculus)
- Robotic paradigms can be described in terms of:
  - Relationship between commonly accepted primitives: SENSE, PLAN and ACT
  - Ways in which sensory data is processed and distributed throughout the system

Robot Primitives in terms of I/O

ROBOT PRIMITIVES	INPUT	OUTPUT
SENSE	Sensor data	Sensed information
PLAN	Information(sensed or cognitive)	Directives
ACT	Sensed information or directives	Actuator commands

The *Hierarchical* Paradigm

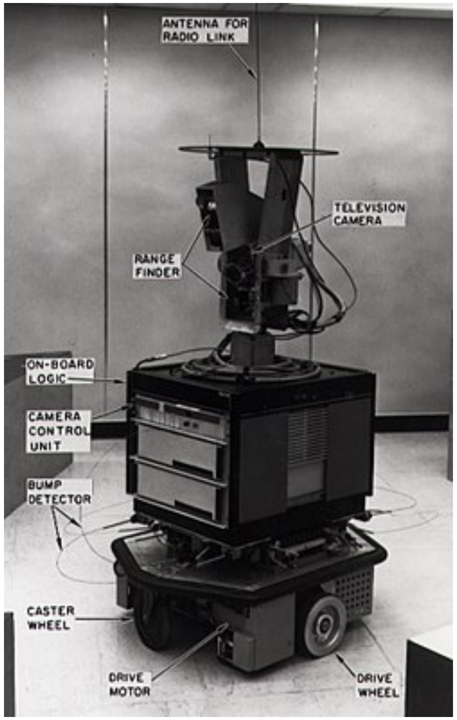


- One of the oldest approaches (1967 – 1990)
- Top down, sensed data is compiled into world model and planner operates on this global model
- Can be hard and brittle due to *closed world assumption* and the so-called *frame problem* (how to define a sufficient set of axioms for the world?)

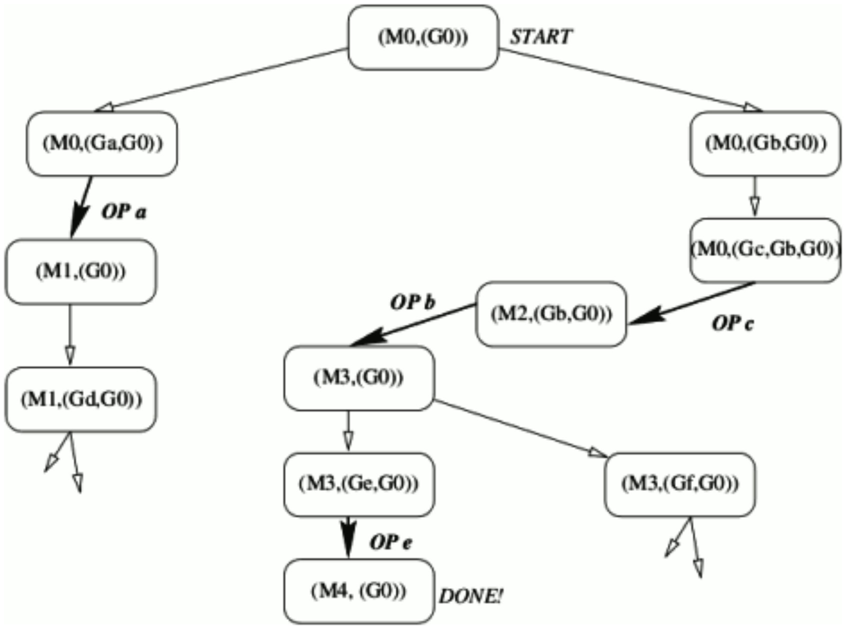
The Hierarchical Paradigm

ROBOT PRIMITIVES	INPUT	OUTPUT
SENSE	Sensor data	Sensed information
PLAN	Information(sensed or cognitive)	Directives
ACT	~~Sensed information or~~ directives	Actuator commands

Example: Shakey and STRIPS



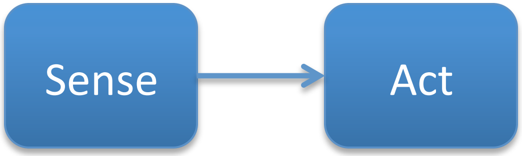
[Source: Wikipedia]



[R.E. Fikes, N.J. Nilsson, STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving, Artificial Intelligence. 2 (3–4): 189–208, 1971.]

Freddy and Shakey: Video

The *Reactive* Paradigm



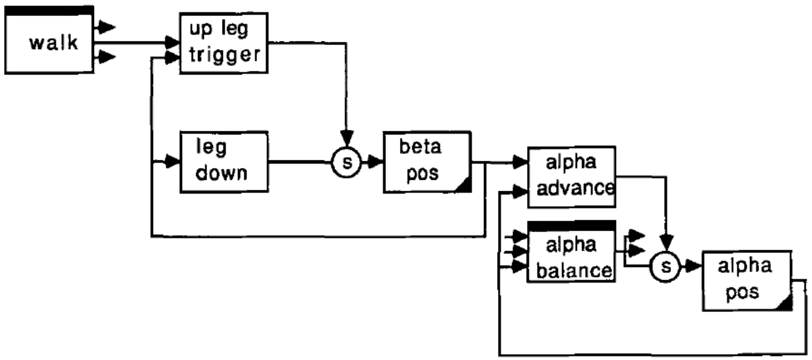
- Started due to disappointment with features of the hierarchical paradigm (1988 – 1992, but older roots in biology and cognitive science)
- Threw out planning altogether! Leveraged availability of low-cost hardware and computing resources
- Several clever robot insect demonstrations, but not sufficiently general purpose for robotics

The *Reactive* Paradigm

ROBOT PRIMITIVES	INPUT	OUTPUT
SENSE	Sensor data	Sensed information
~~PLAN~~	~~Information(sensed or cognitive)~~	~~Directives~~
ACT	Sensed information ~~or directives~~	Actuator commands

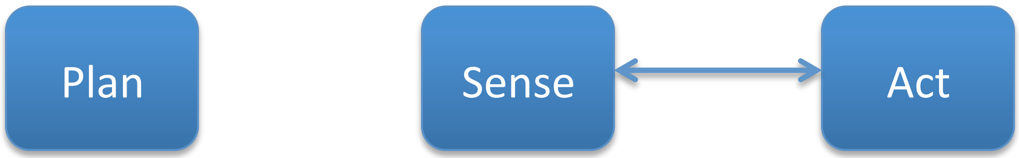
Example: Brooks' Insect Robots

[Source: ai.mit.edu]



[Brooks, R.A., A robot that walks; emergent behaviors from a carefully evolved network. Neural computation, 1(2), pp.253-262, 1989]

The *Hybrid Deliberative/Reactive* Paradigm

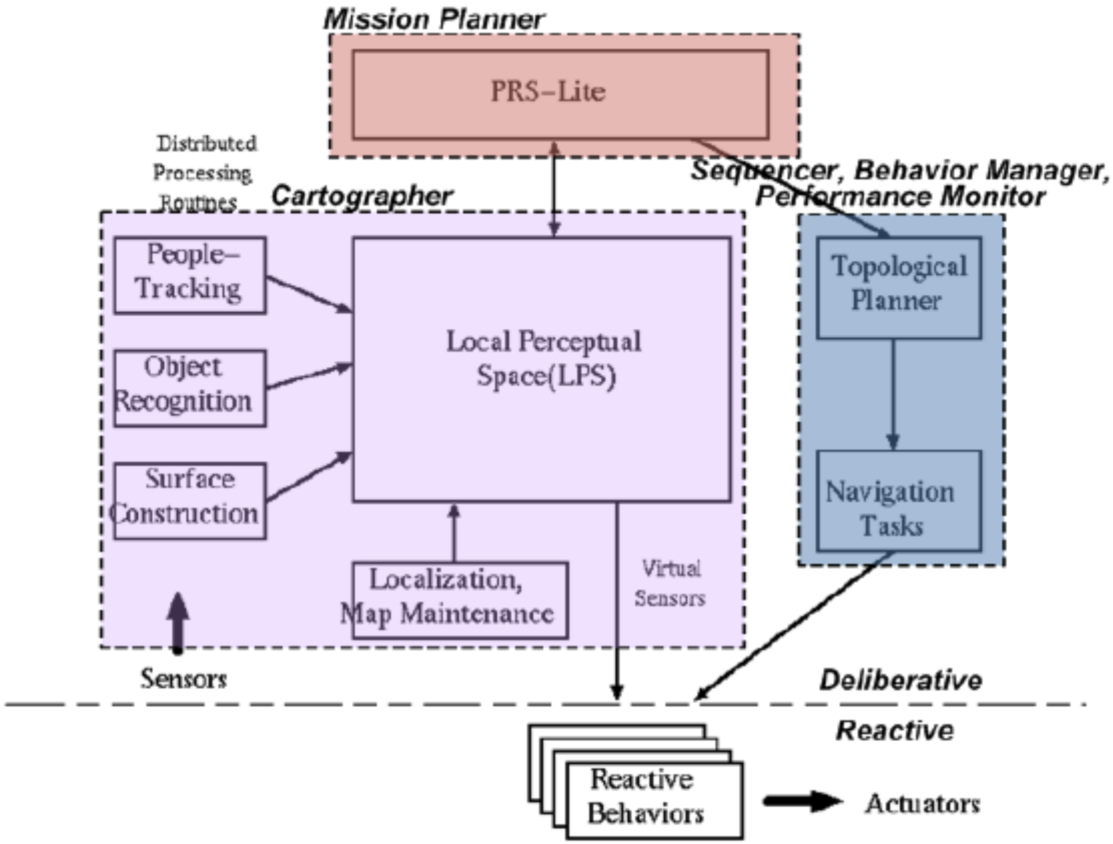


- Many current robots use this approach (1990s onwards)
- First, the robot deliberates how to break down task into subtasks (mission planning)
- Then the individual behaviours are executed as per a fast reactive paradigm
- PLAN, SENSE-ACT (P, S-A)

The Hybrid Paradigm

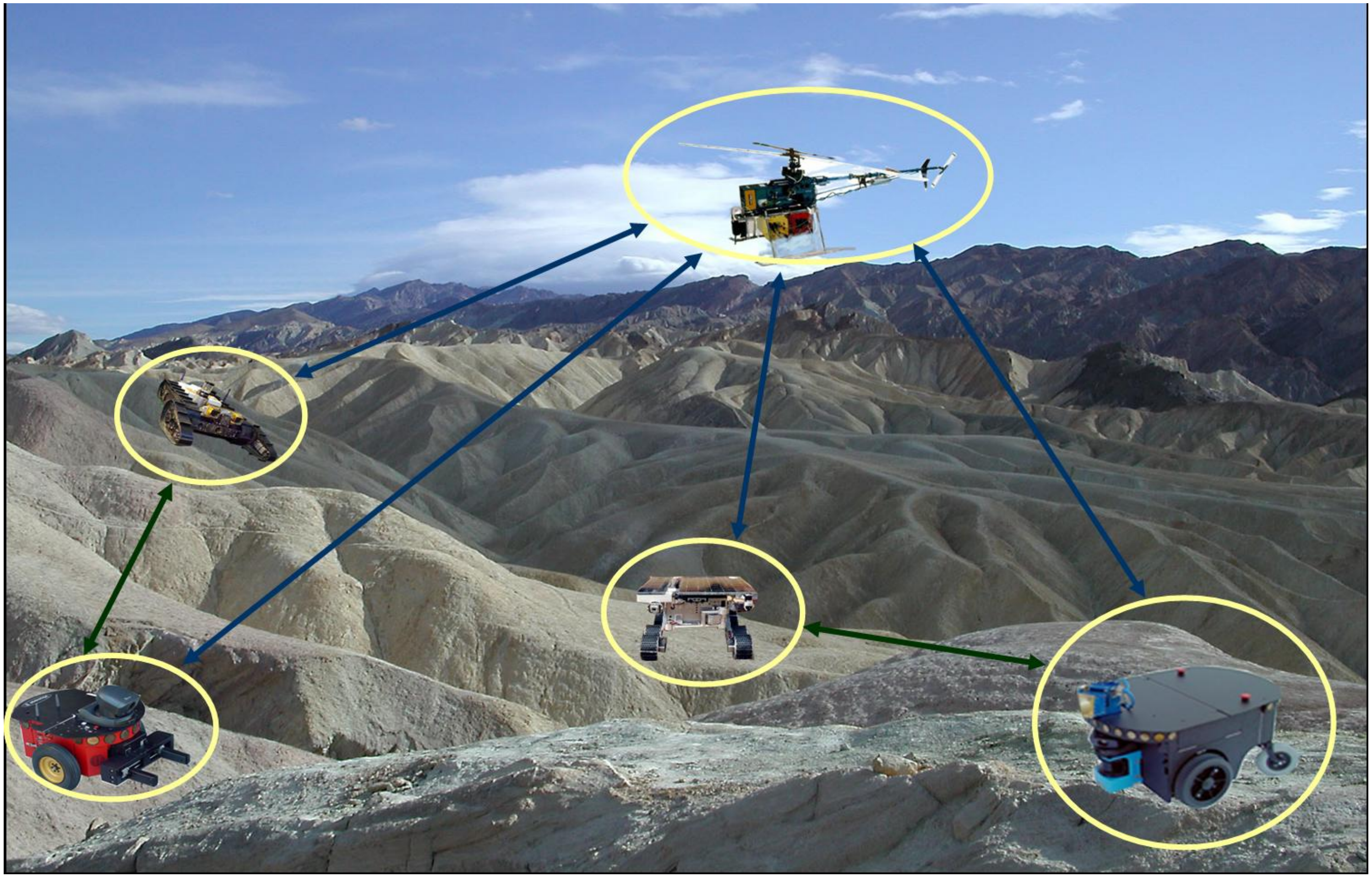
ROBOT PRIMITIVES	INPUT	OUTPUT
PLAN	Information(sensed or cognitive)	Directives
SENSE-ACT (behaviours)	Sensor data	Actuator commands

Example: “Modern” Mobile Robots



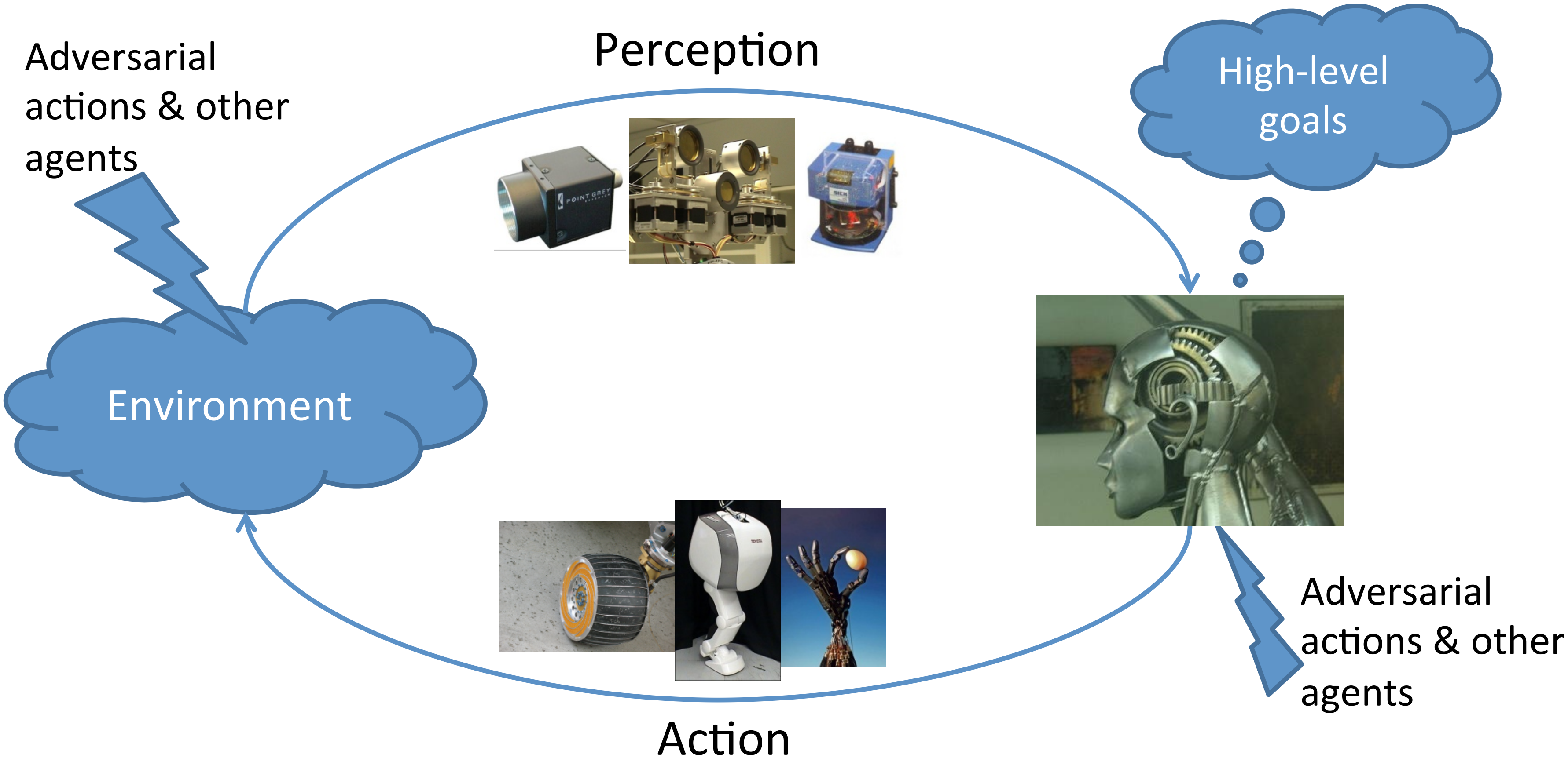
[Konolige, K., Myers, K., Ruspini, E., & Saffiotti, A. The Saphira architecture: A design for autonomy. Journal of experimental & theoretical artificial intelligence, 9(2-3), 215-235, 1997.]

Another "Modern" Issue: Interaction



[Source: http://www.ee.ucr.edu/~mourikis/project_pages/images/multi.jpg]

So, What **is** a Robot?



<u>Problem</u>: How to generate actions, to achieve high-level goals, using limited perception and incomplete knowledge of environment & adversarial actions?

Example Application: Autonomous Vehicles

- http://www.youtube.com/watch?gl=GB&v=1W27Q6YvTXc

- What are the various decisions involved?
- What paradigm(s) would you adopt?

Example Application: Rescue Robots

- http://www.youtube.com/watch?v=F7lqriYKsX4

- What are the various decisions involved?
- What paradigm(s) would you adopt?

Example Application: Automated Warehouses

- https://www.youtube.com/watch?v=6KRjuuEVEZs

- What are the various decisions involved?
- What paradigm(s) would you adopt?

Example Application: Humanoid Robots at Work!

- https://www.youtube.com/watch?v=DpTSXeei9zo

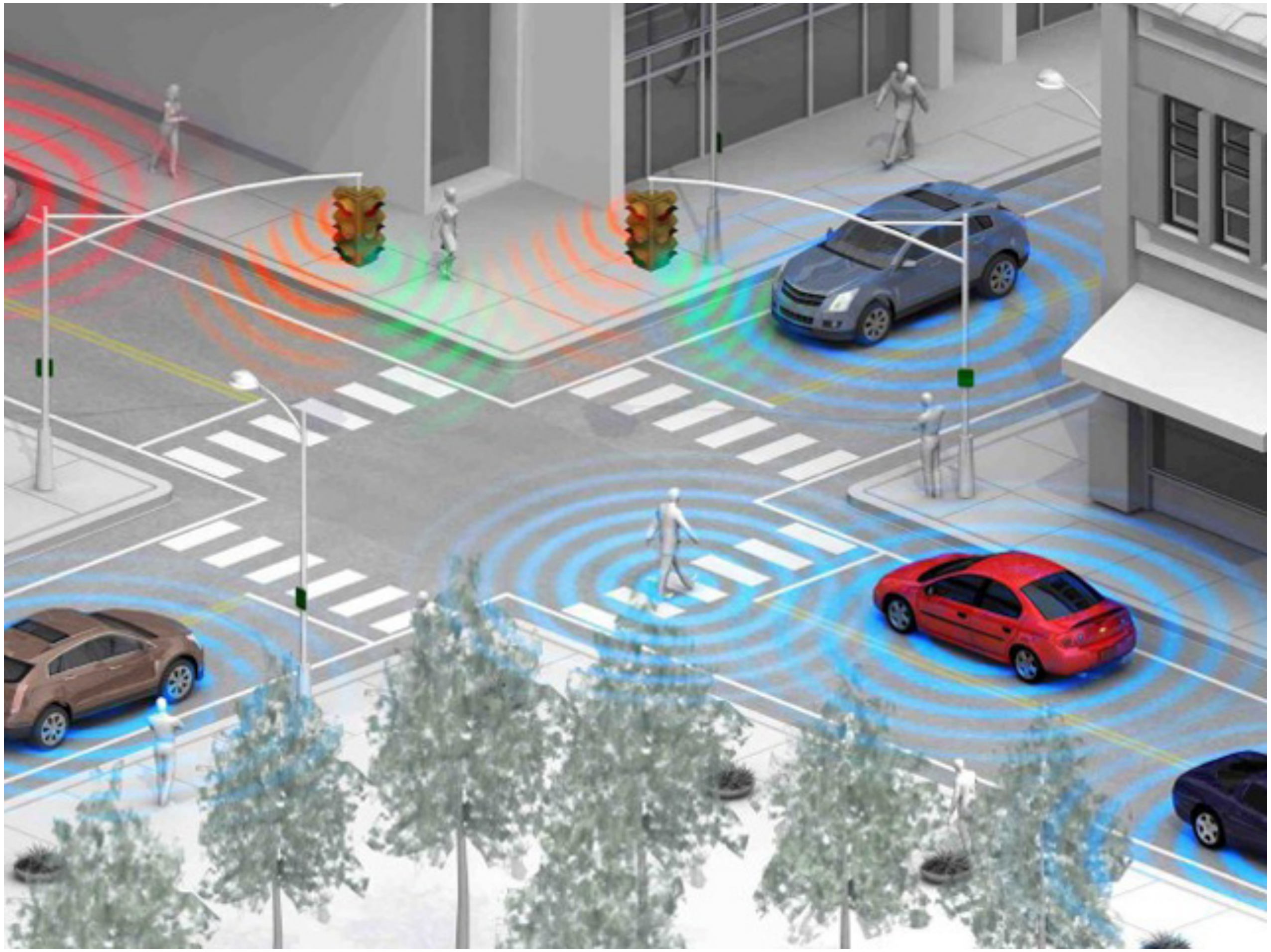
- What are the various decisions involved?
- What paradigm(s) would you adopt?

The Designer's Task: Components of the Problem

In each case,

- what are the components? how do you delineate?
- what does one (i.e., your robot) need to know?
- what does a motion strategy consist of?
  - what properties must the strategy satisfy?

What changes?
Who else is around?

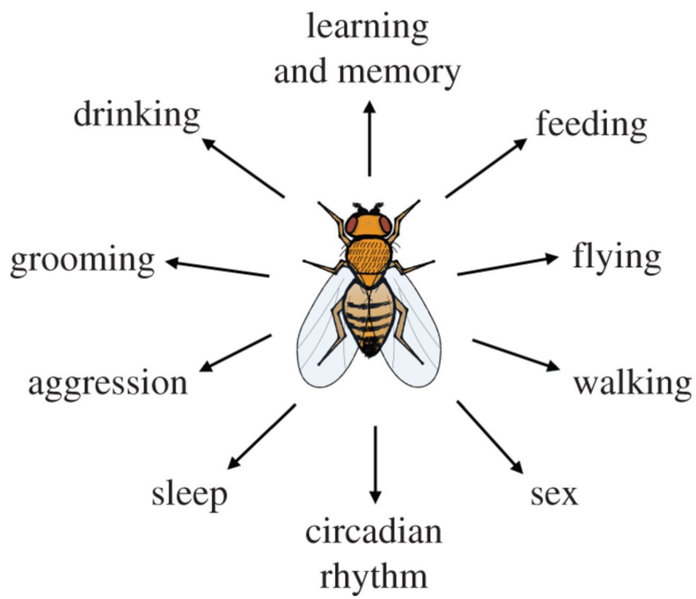


How does the car move?
- <u>Kinematics, Dynamics</u>

Where does the car move?
- <u>World models</u>

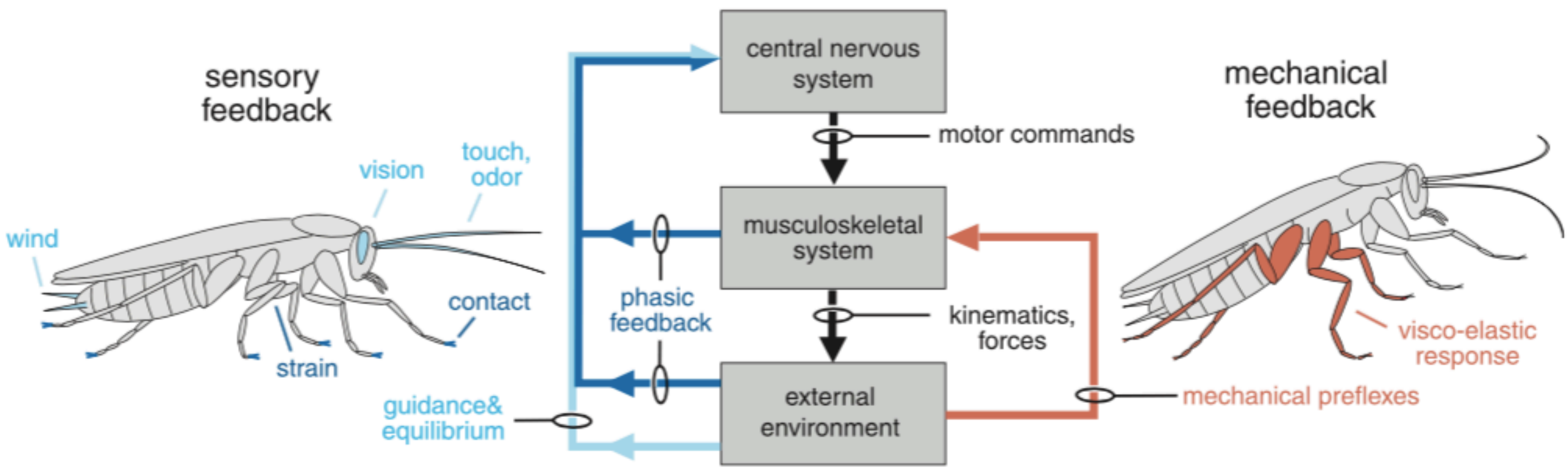
Is this AI perspective the only way? No – Consider Neuroscience/Biology

- Organisms like flies have been very extensively studied
  - Could they form useful inspiration for UAVs, for instance
- They outperform engineered systems on both counts:
  - Robustnss
  - Energy efficiency



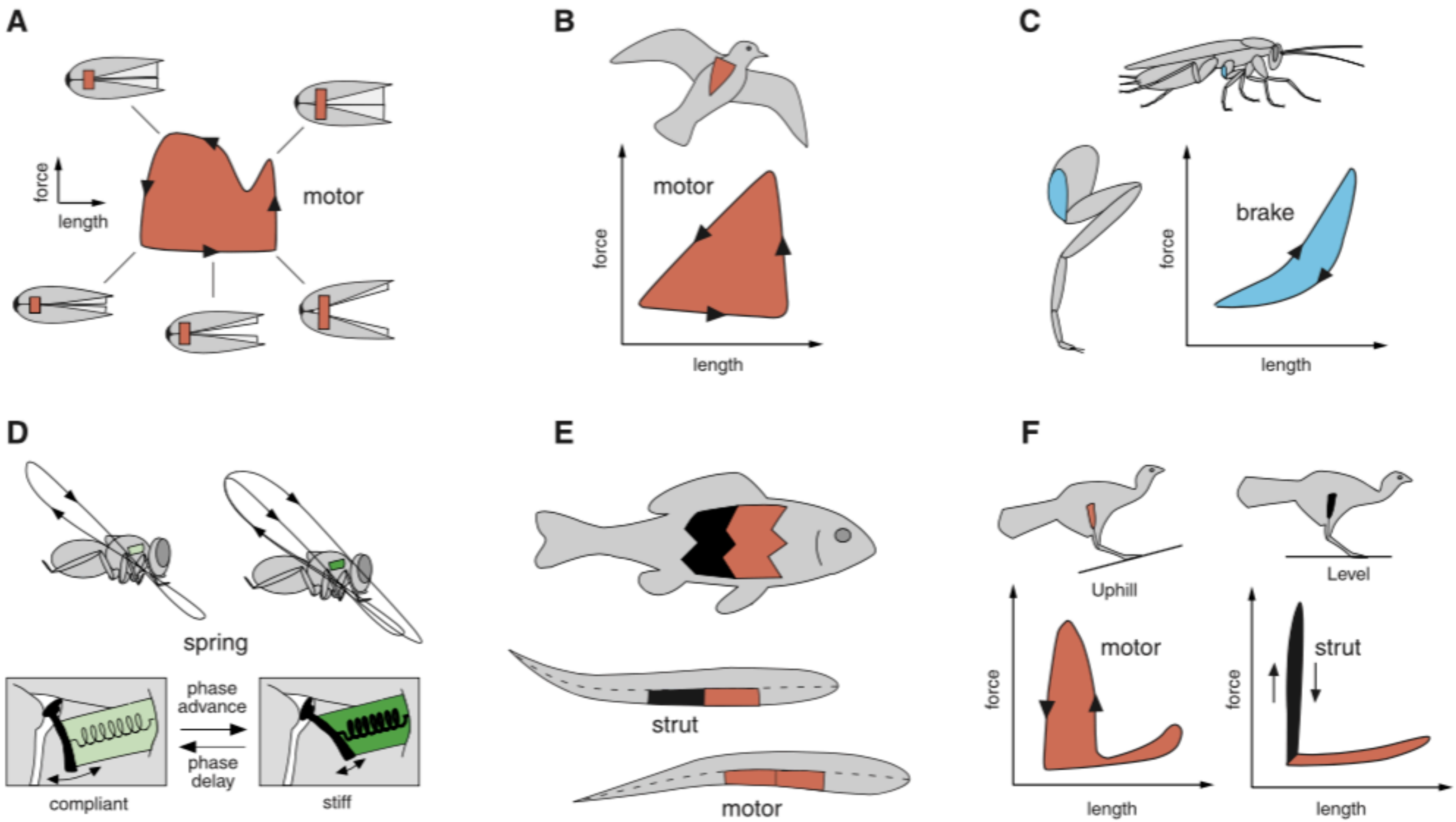
[Owald et al., Phil Trans Royal Soc B 2015: https://doi.org/10.1098/rstb.2014.0211]

Feedback Loops in Biology Could Inform Robotics



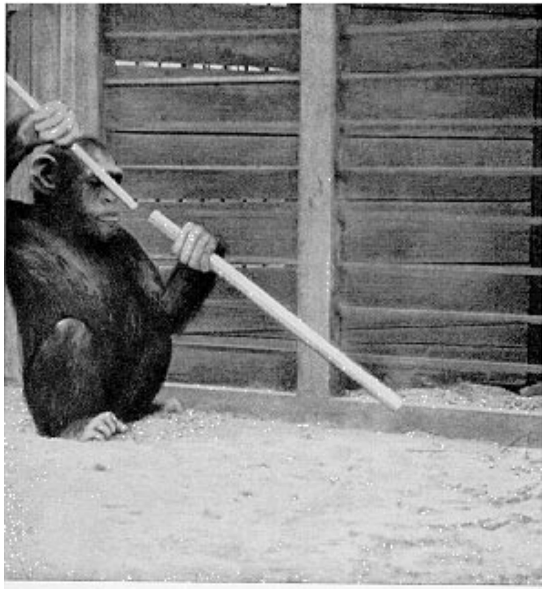
[Dickinson et al., Science Vol 288, 2000]

Example Insight: Muscles Can Act as Motors, Brakes, Springs, Struts



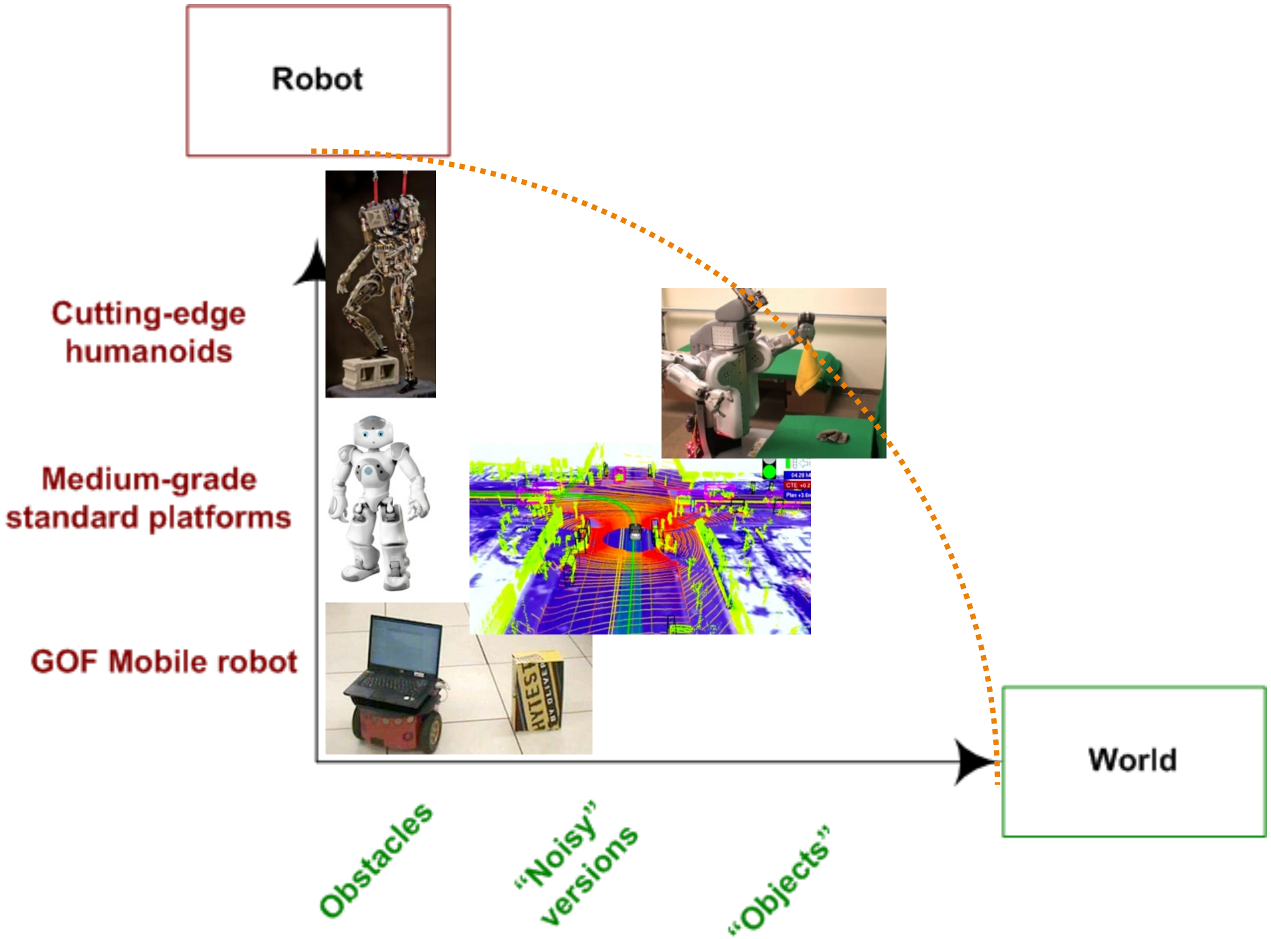
[Dickinson et al., Science Vol 288, 2000]

How to Understand Non-trivial Decisions Bottom-up? Can be Difficult!



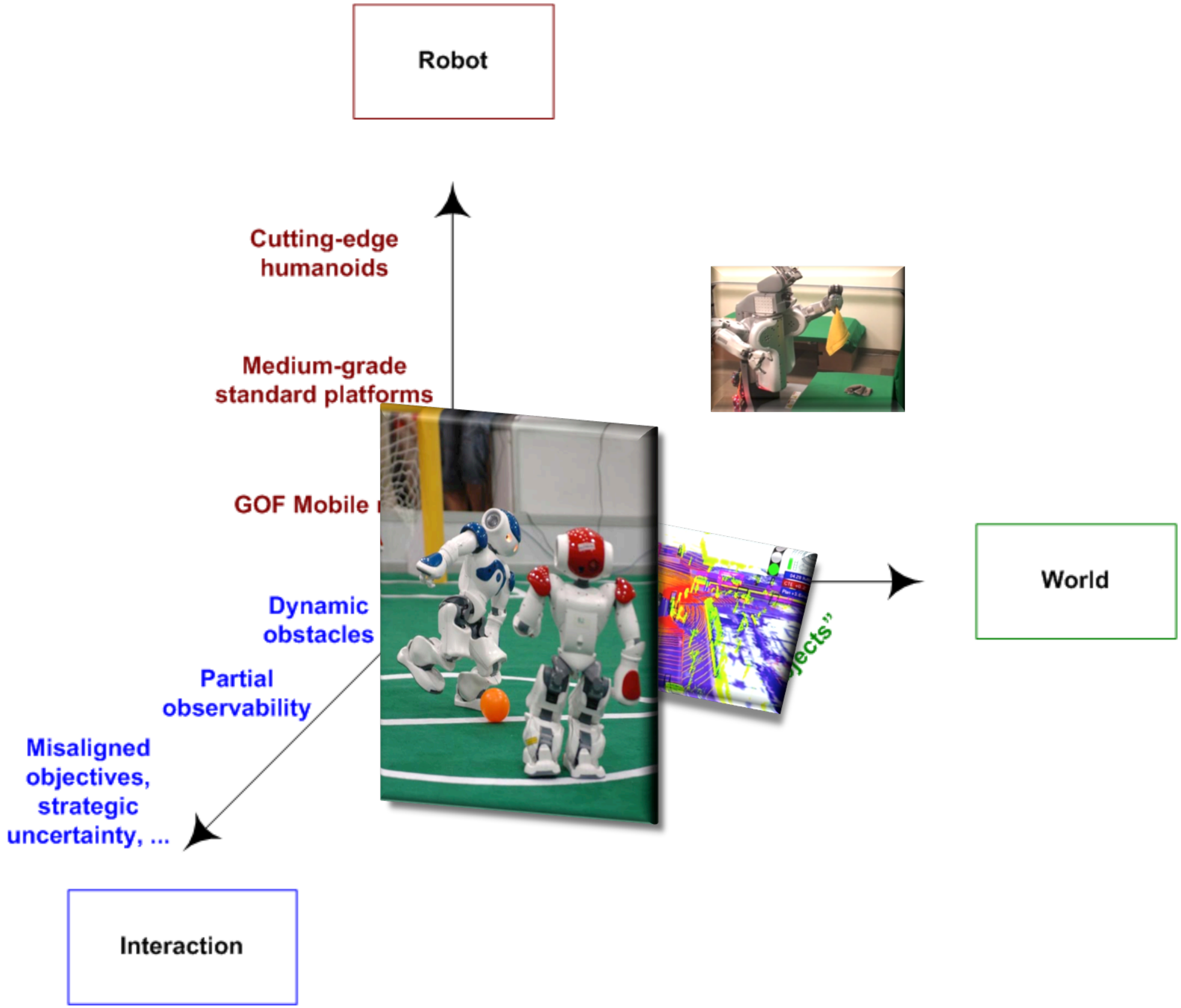
[Kohler's Mentality of Apes: Kohler suspended fruit from the ceiling, and the apes found creative strategies to get at it!]

What Makes Robotics Problems *Hard*?

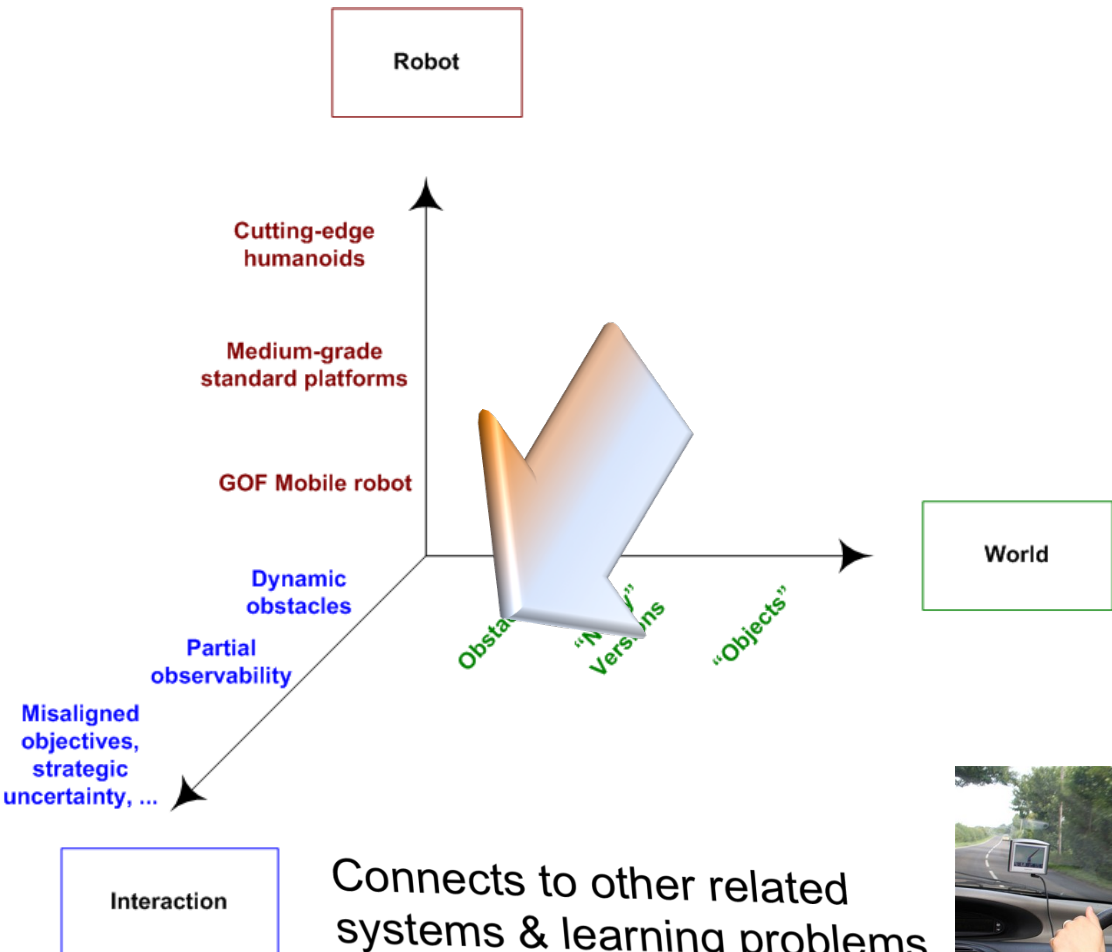


One more thing…

What Makes Robotics Problems Hard?

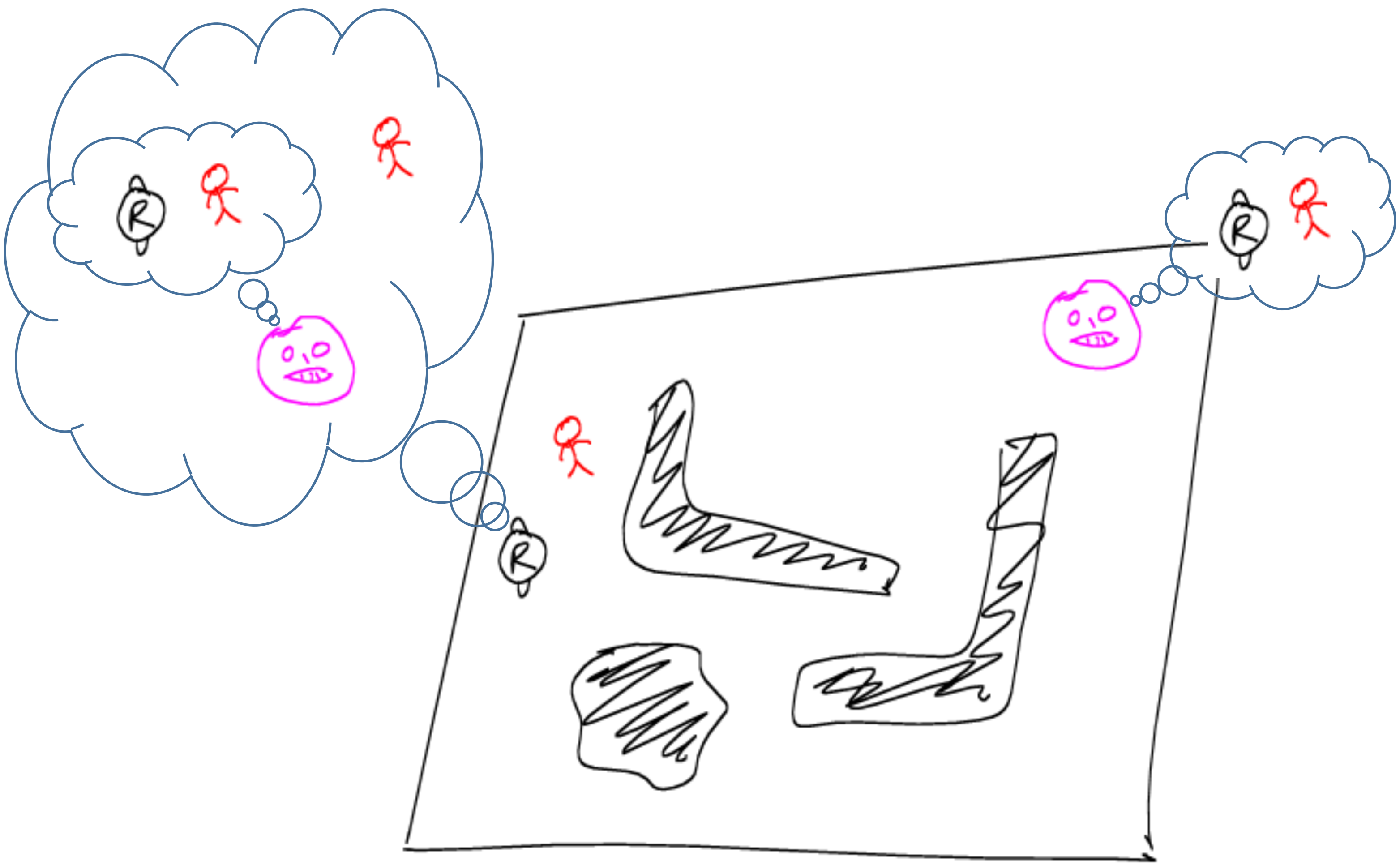


What Happens if You Plug in *Real* People?



Connects to other related systems & learning problems

Computational Issues: Toy Example



Non-stationarity, plan recognition, personalisation, incentives, strategic coordination

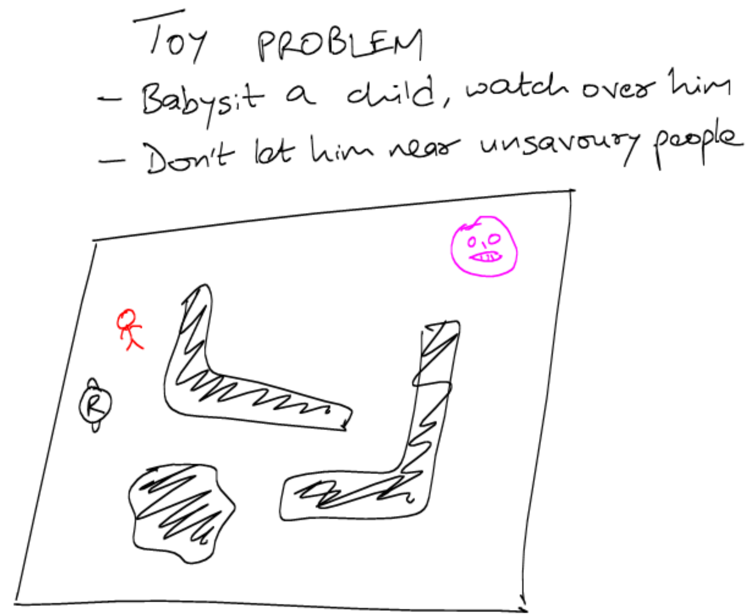
Levels of Difficulty in Interaction

Consequences for *hardness of learning*:

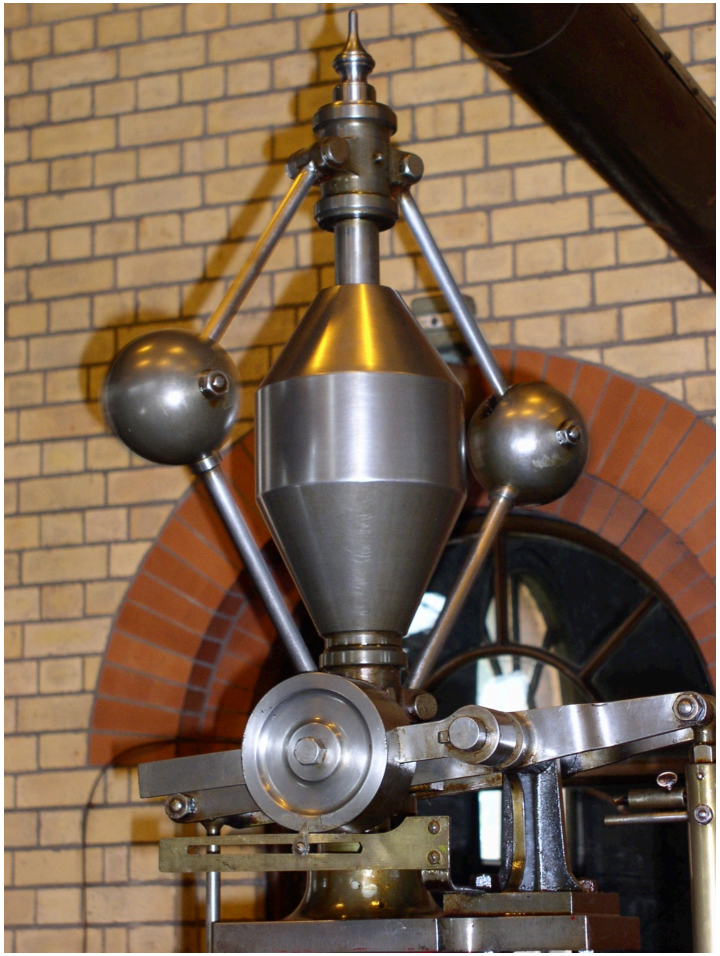
1. Base case: spatial asymmetry
   – Learn a vector field

2. Next level: deal with reactive behaviour
   – 'Inverse' planning, plan recognition

3. Harder case: recursive exchange of beliefs (e.g., signaling, implicit coordination, trust, persuasion)
   – Need to model as a game?



Centrifugal Governor (James Watt, 1788)



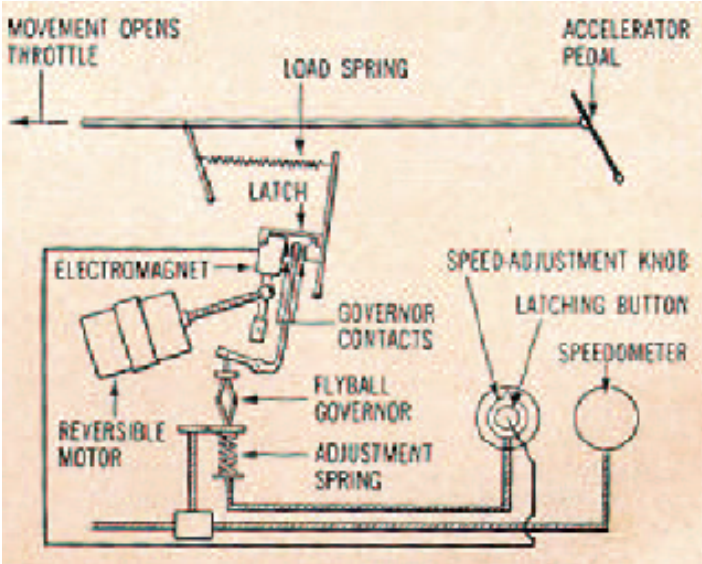
Not Only of Historical Interest…

. . . with AUTO PILOT
for *CHRYSLER and IMPERIAL*

Just set the convenient instrument panel dial to your desired speed. Then drive in your usual manner. When you reach the pre-set speed you feel a gentle nudge of the accelerator on your foot telling you you've reached your desired speed.

For completely automatic control, pull the control knob when you feel the nudge of the pedal and remove your foot from the accelerator. Then, drive relaxed with your eyes on the road.

A touch of your brake pedal instantly returns the control to manual. To return to automatic control, just accelerate until you feel the nudge and remove your foot from the accelerator.



How does a Governor Work? Proportional Control

- A feedback system that controls the speed of an engine by regulating the amount of fuel (or working fluid) admitted
- Goal is to maintain a near-constant speed, irrespective of the load or fuel-supply conditions.

A sequence of operations:

1) Power is supplied to the governor from the engine's output shaft. The governor is connected to a throttle valve that regulates the flow of working fluid (steam) supplying the prime mover.

How does a Governor Work? Proportional Control

2) As the speed of the prime mover increases, the central spindle of the governor rotates at a faster rate and the kinetic energy of the balls increases.

3) This allows the two masses on lever arms to move outwards and upwards against gravity.

4) If the motion goes far enough, this motion causes the lever arms to pull down on a thrust bearing, which moves a beam linkage, which reduces the aperture of a throttle valve.

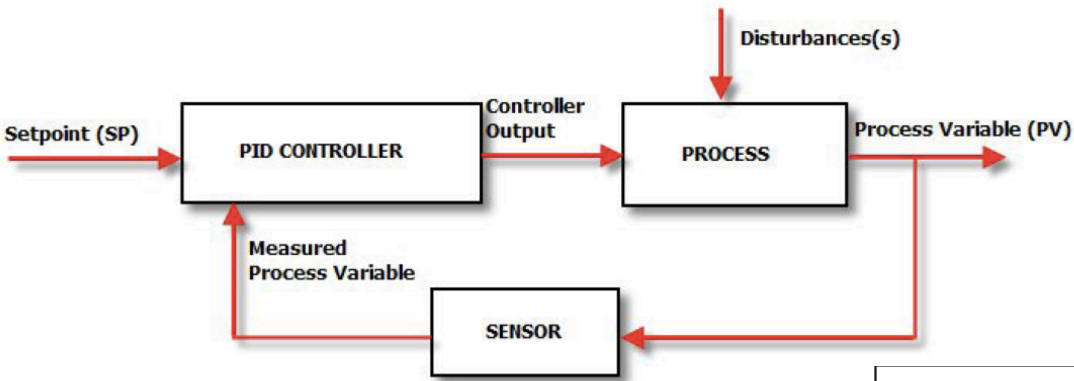
5) The rate of working-fluid entering the cylinder is thus reduced and the speed of the prime mover is controlled, preventing over-speeding.

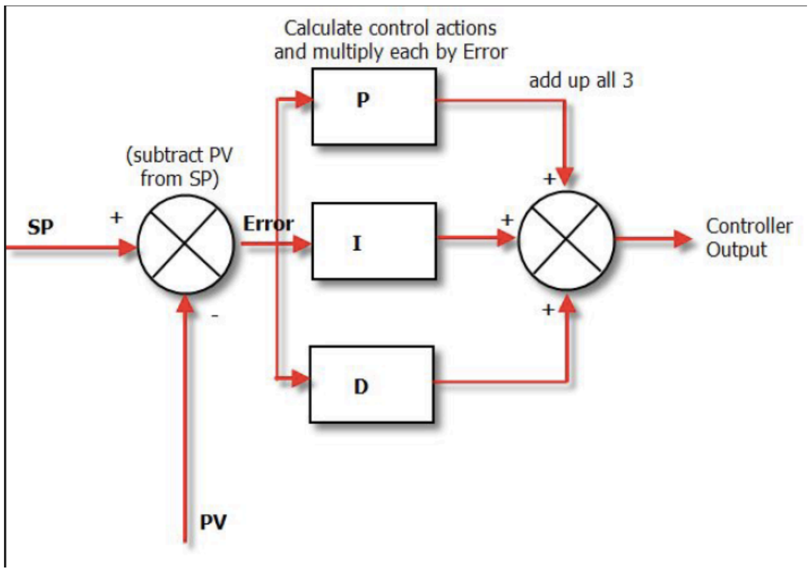
Proportional Control

- We want to hold system "in place" – in this case, at a certain rate of flow
- When flow exceeds desired value, the mechanism applies a correction which is proportional to the excess

- This idea of regulation is quite valuable in all engineered systems
- However, the quantity being regulated is not always flow
- How to write down the principle mathematically?
  – We also need to say how to describe the system

PID Controllers





Proportional-Integral-Derivative Control

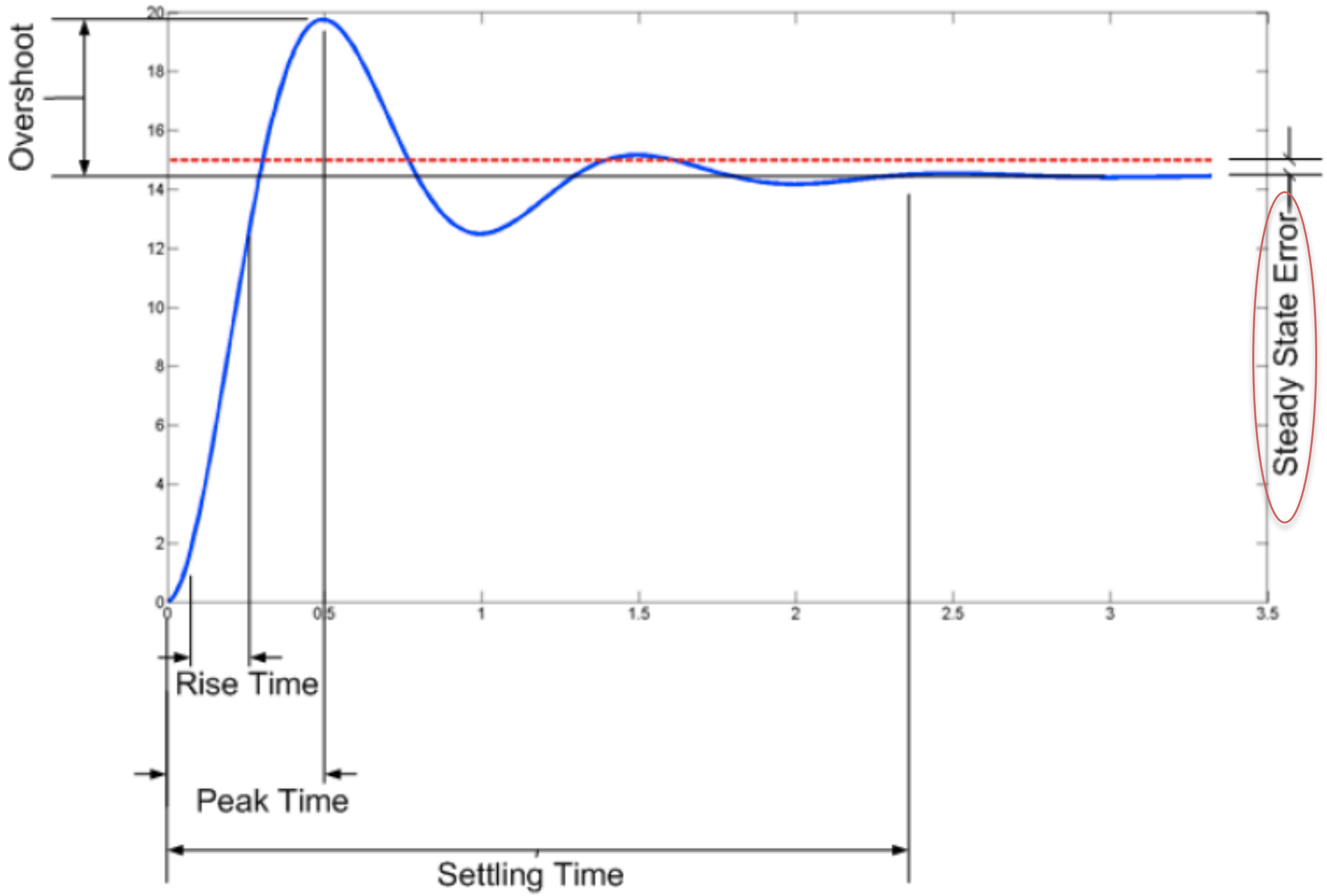
- The control signal, $u(t)$, is given in terms of the error $e(t)$ as,

$$u(t) = K_p e(t) + K_i \int_{t_0}^{t} e(\tau) d\tau + K_d \dot{e}(t)$$

- This simple algorithm is most useful when processes are known to be stable and not very oscillatory
  - Parameters may not be well known, however

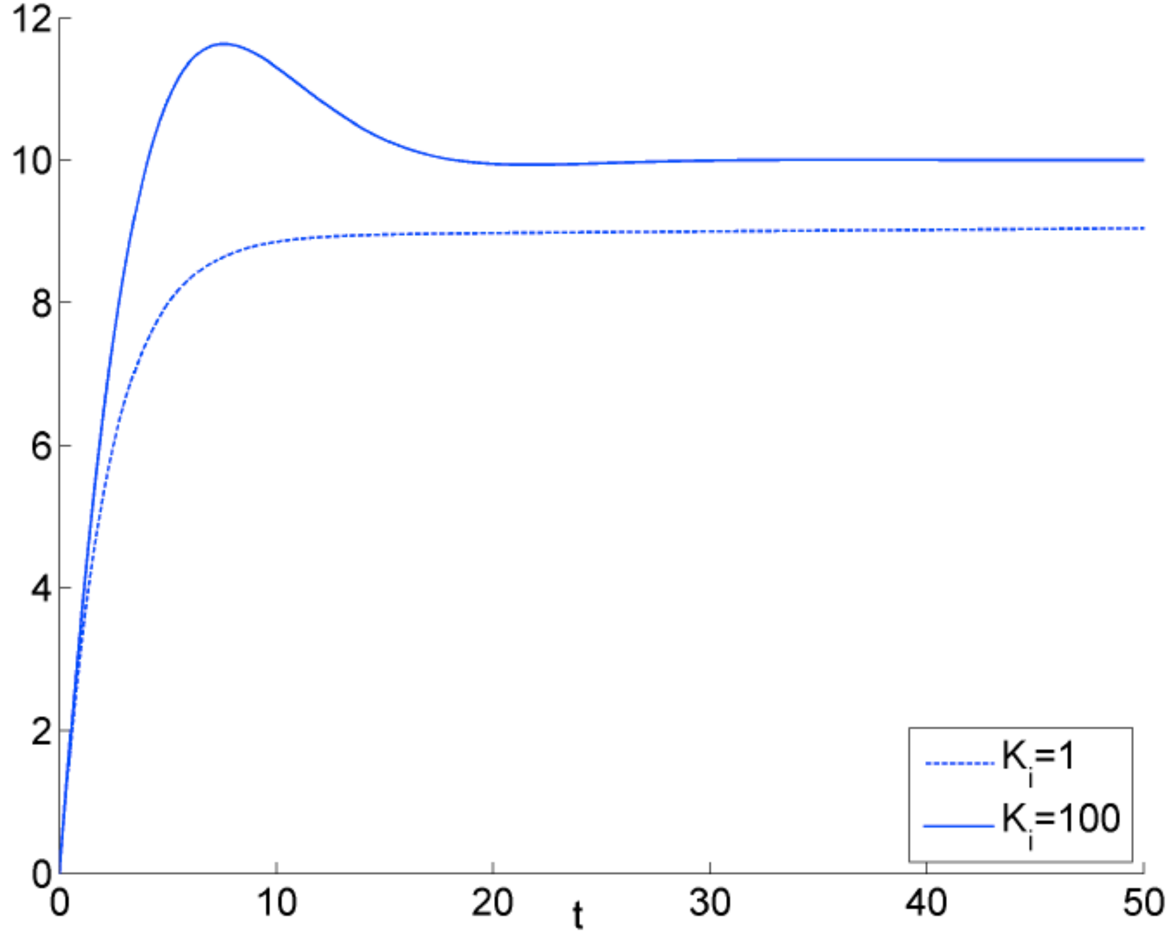
- Why is each term needed?
- How could we set the scale factors (the Ks)?

Typical Step Response of 2nd Order System with Proportional Control



Why?

Step Response with Different Levels of Integral Gain (Setpoint = 10)



Effects of Different Components

Control Action	Rise Time	Overshoot	Settling Time	Steady State Error
Increasing K_p	reduces	increases	small change	reduces
Increasing K_i	reduces	increases	increases	eliminates
Increasing K_d	small change	reduces	reduces	small change

Many Design Heuristics, e.g., Ziegler-Nichols Rules (1942)

- Trial and error procedure, entirely empirical
- Gradually increase proportional gain alone until the system begins to oscillate (with loop gain, K_u, and period, T_u)
- Then, set the gains to be:

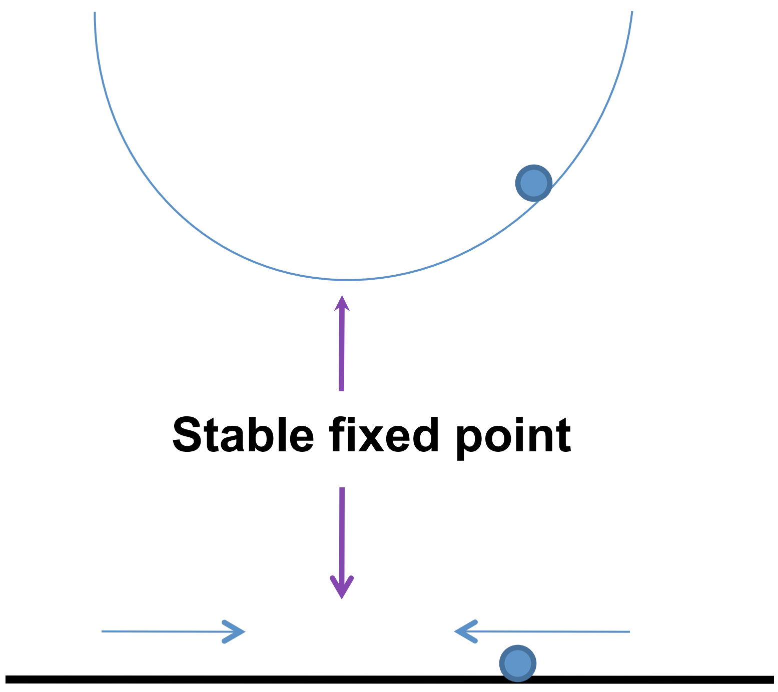
$$K_p = \frac{1}{2} K_u$$
$$K_i = \frac{2}{T_u} K_p$$
$$K_d = \frac{T_u}{8} K_p$$

- How to go from simple speed regulation to controlling movements?

Concept: Potential Function

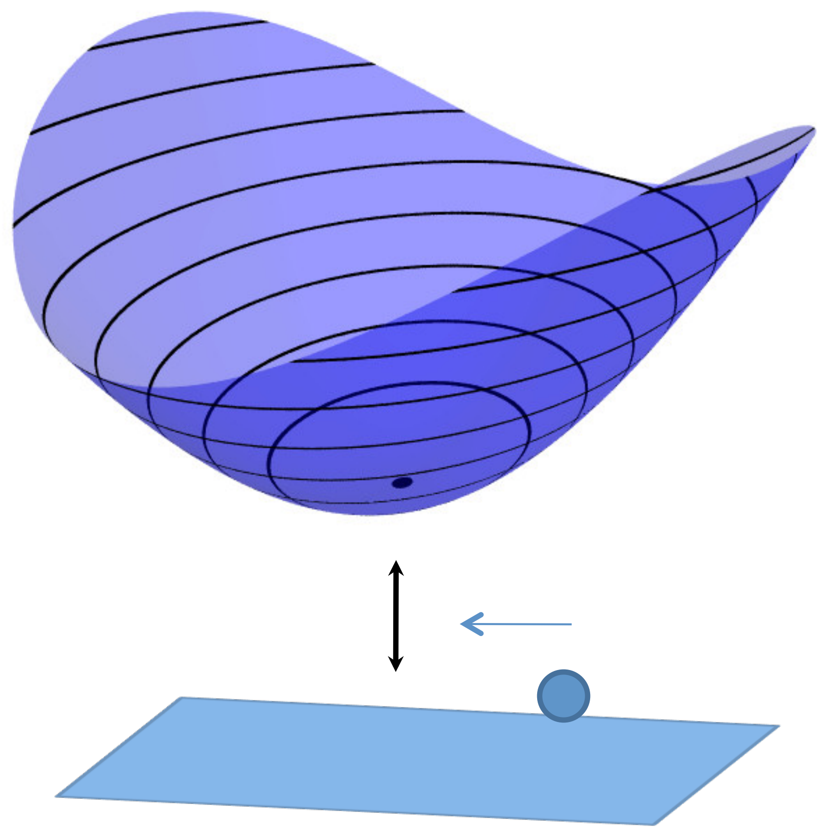
- Imagine a *1*-dim ball rolling within a flat bowl
- Where will it eventually end up after long time interval?
- What happens if you push the ball around with your finger?

- If you "create" such a field on *1*-dim flat world, where will the ball go?
- This is what PID does…



Higher-dim bowl

- You could play the same game in higher dimensions
- With contours that shrink down to a point, the ball will move in the direction that decreases a measure of height



- The effect on a *2*-dim workspace is that the ball will converge to a fixed point

How to Encode a Bowl? Potential Function

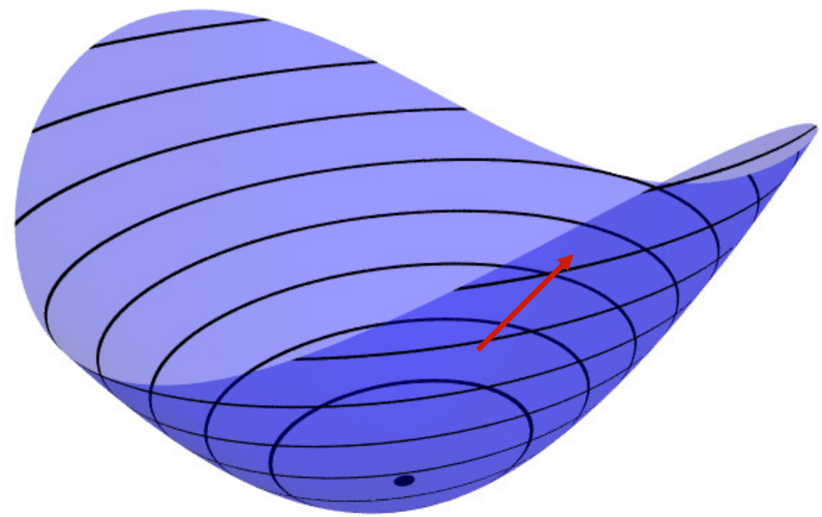
- Differentiable real-valued function,

$$U : \mathbb{R}^m \mapsto \mathbb{R}$$

- Treat the value as ‘energy’
- Then, gradient is the vector,

$$\nabla U(q) = DU(q)' = [\frac{\partial U}{\partial q_1}(q), \ldots, \frac{\partial U}{\partial q_m}(q)]'$$

- The gradient points in the direction that locally maximally increases U

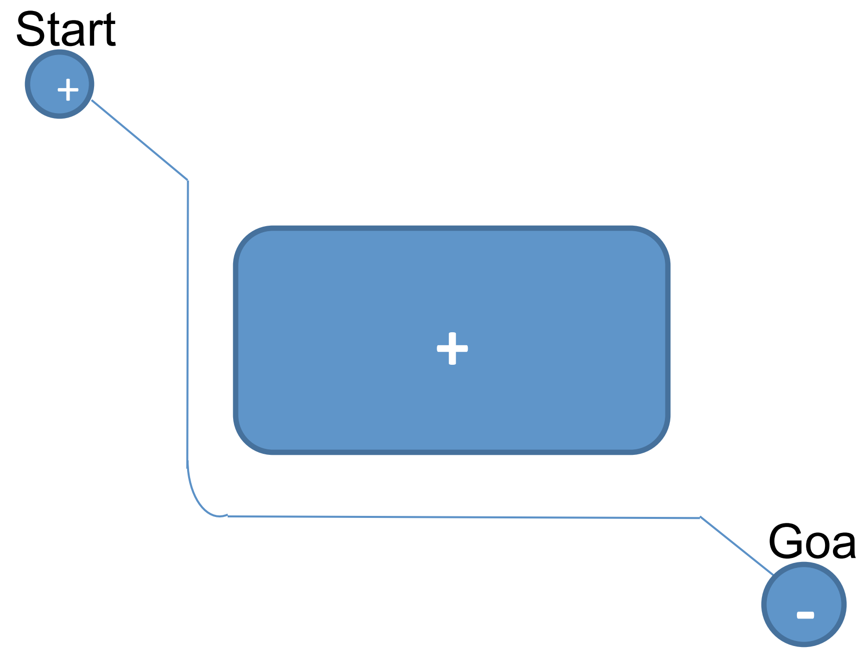


Property: Work done along a closed path is zero

Point Robot with Potential Function

- Moves along a vector field induced by gradients
- We can think of gradient as force on a charged particle attracted to opposite charge
- This way, one can have attraction/repulsion for navigation purposes

<u>Note</u>: We are currently ignoring "real dynamics"; but imposing an artificial dynamics for navigation



Path taken by a +ve charged particle

One Approach to Representing a Potential Function

Potential function can include attractive and repulsive terms,

$$U(q) = U_{att}(q) + U_{rep}(q)$$

Attractive Potential choices:

Conic: $U(q) = \zeta \|q - q_{goal}\|$

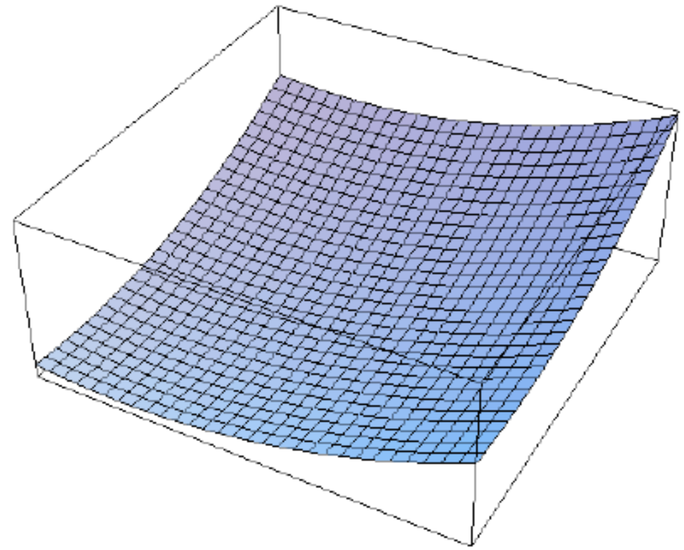
Quadratic: $U_{att}(q) = \frac{1}{2}\zeta d^2(q, q_{goal})$

Repulsive Potential choices:

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta(\frac{1}{D(q)} - \frac{1}{Q^*})^2, & D(q) \leq Q^* \\ 0, & D(q) > Q^* \end{cases}$$

Attractive Potential

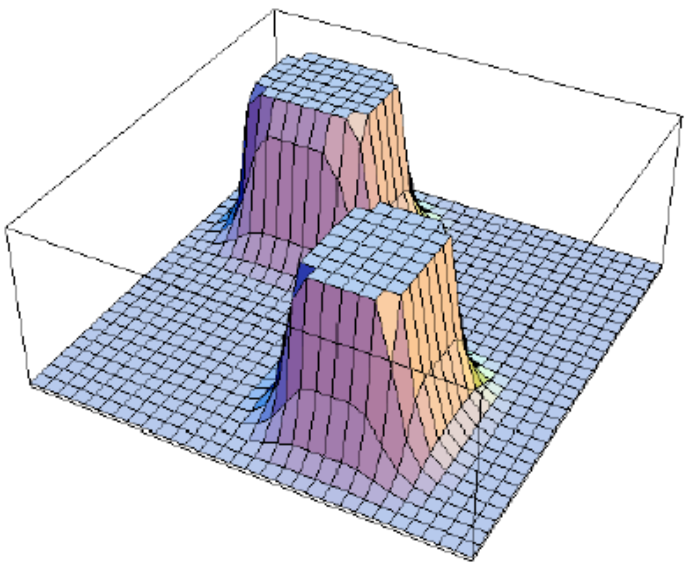
$$U_{\text{att}}(q) = \frac{1}{2} k \, \delta^2_{\text{goal}}(q)$$



$$F_{\text{att}}(q) = -\nabla U_{\text{att}}(q)$$
$$= -k \, \delta_{\text{goal}}(q)$$

Repulsive Potential

$U_{rep}(q) = 1/2\ \eta\ (1/D(q) - 1/Q^*)^2,\ D(q) \leq Q^*$, else 0

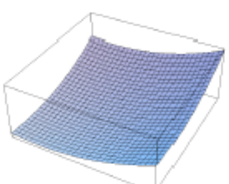


For convex obstacles:
Potentials can be superimposed (summed)

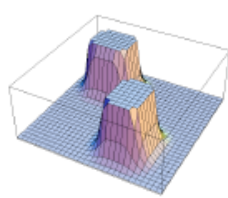
Combined Potential Field

$$U(q) = U_{\text{att}}(q) + U_{\text{rep}}(q)$$

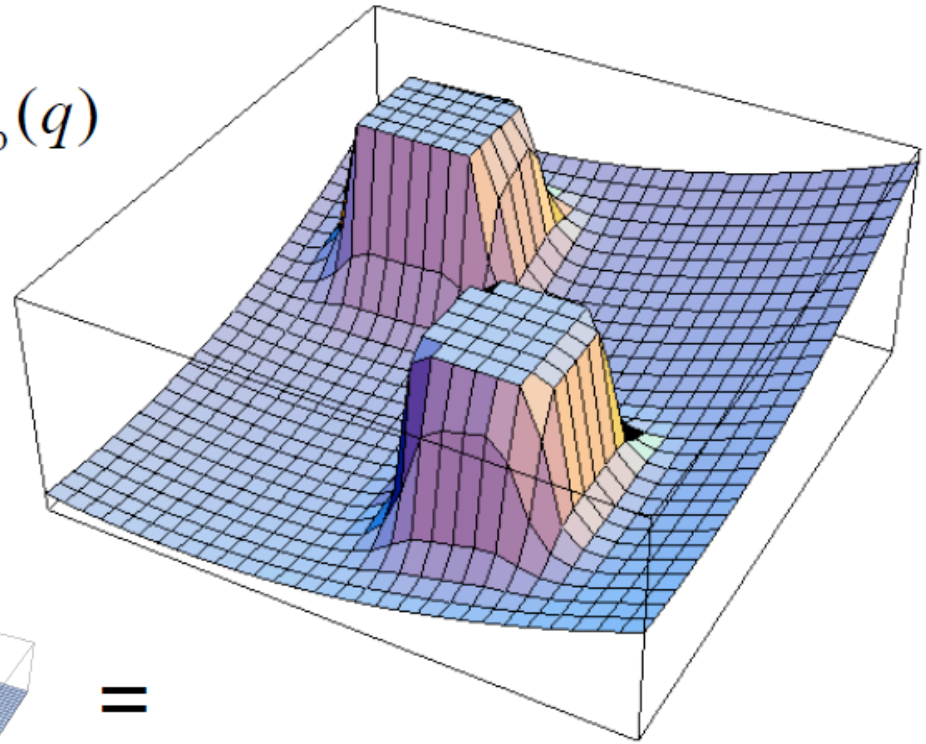
$$F(q) = -\nabla U(q)$$



+



=



On-line Motion Planning with PF

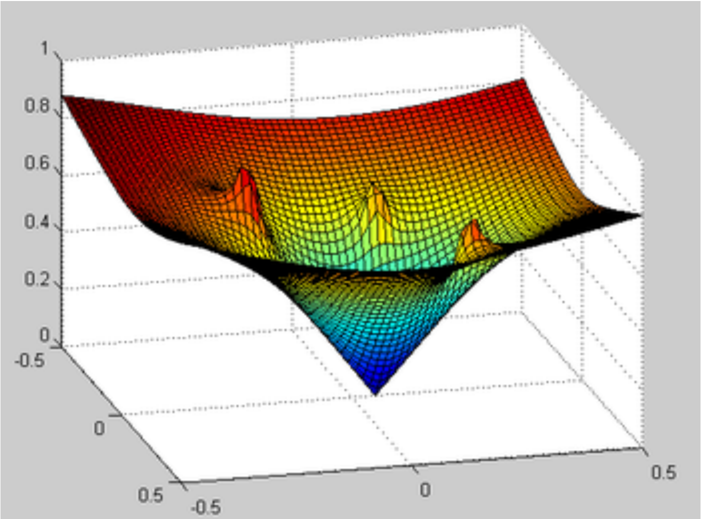
Input: Function, $\nabla U(q)$
Output: Sequence $[q(0), q(1), ...q(i)]$

1. $q(0) = q_{start}$
2. i = 0
3. **while** $\nabla U(q(i)) \neq 0$
4. $\quad q(i+1) = q(i) + \alpha(i)\nabla U(q(i))$
5. $\quad i = i + 1$
6. **end while**

Many possible improvements,
draw on numerical optimization methods

Possible Problems with Potential Function Approach

Local Minima!



As we create potential surfaces by superposition, it is not easy to ensure there are no spurious minima

- "complex" function shapes can be hard to engineer
- **Need systematic tools for task encoding or decision making**

In this course…

We will focus on how to model and compute decisions (*choices*),

- over <u>time</u>, under <u>uncertainty</u>, with <u>incompleteness</u> in models
- emphasizing difficulties involving hidden causality, interaction, etc.
- possibly needing methods for <u>learning</u> from experience and data.
- also, we'll think a bit about how ***real*** people make choices!

Major Themes:

1. Different models of decision making (control and planning)
2. Understand issues, sometimes through case studies
    - What to model, what/how to analyse?
3. Special issues: safety, security, explainability, human factors

Course Structure

- Schedule of lectures is available at the course web site

  http://www.inf.ed.ac.uk/teaching/courses/dmr/

  I will attempt to upload slides by day before (except in first week)
- Two homework assignments
  - Pen-and-paper exercise on models, concepts, methods (10%)
  - Practical programming exercise in a mock-up domain (20%)
- Term Paper
  - 4 page conference-style paper on your chosen topic
- Final Exam (60% of final mark)
- Resources:
  - No prescribed textbook
  - Suggested readings assigned with lecture slides

Ask Questions!

- During the lecture
- After class, if your questions are brief
- After hours, by prior appointment *only* (arranged via email)

- You could also approach your TA (arrange via email):
  - Dr Michael Burke, mburke33@exseed.ed.ac.uk
  - Dr Simon Smith, artificialsimon@ed.ac.uk

References

- The material regarding robotic paradigms is from R.R. Murphy, Introduction to AI Robotics, MIT Press, 2000.