Spatial Reasoning about Human-Robot Interactions

Aris Valtazanos
University of Edinburgh

Overview

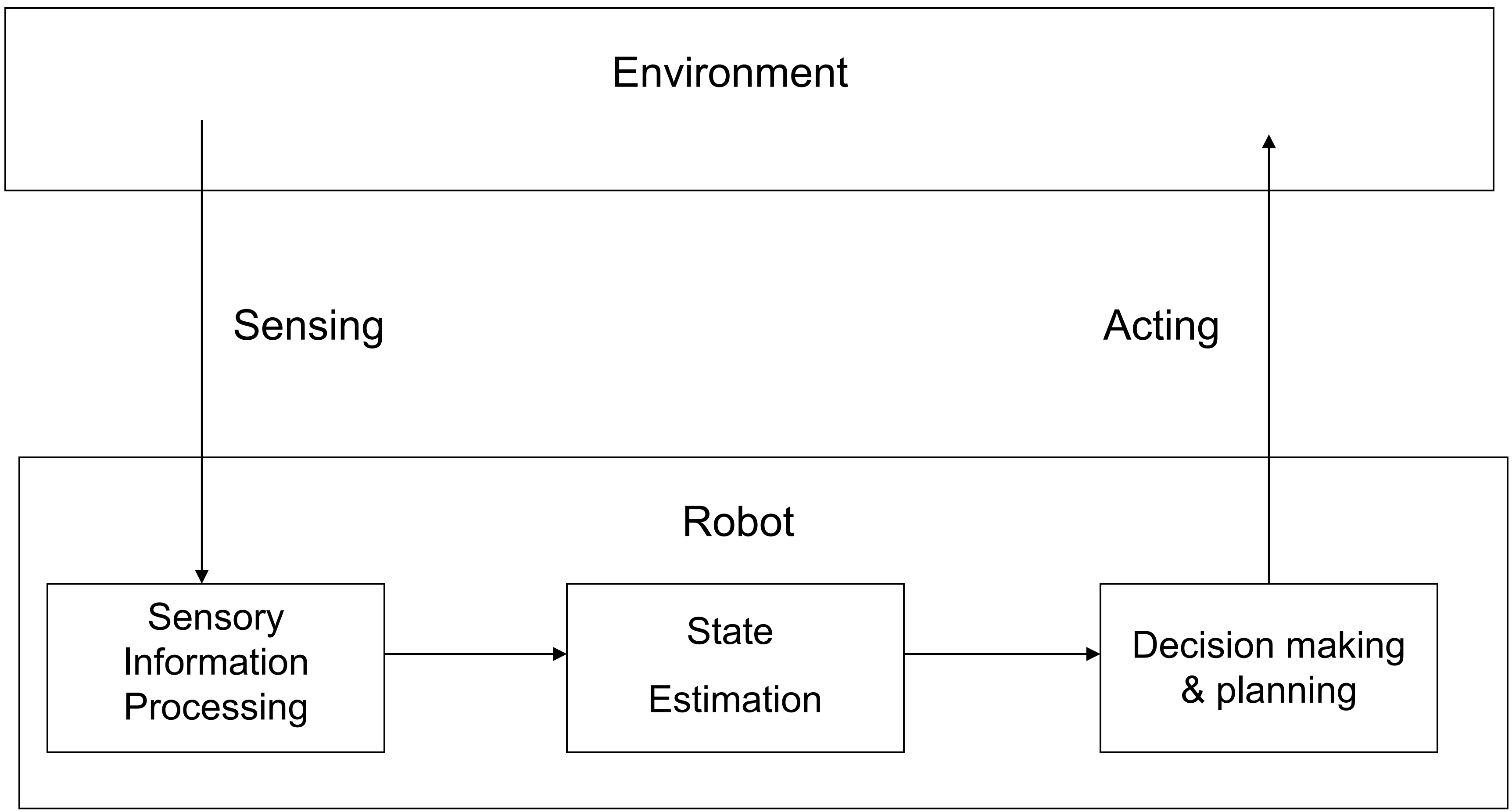
- Robotics, its applications, and its challenges
- Robotic soccer and robust autonomy
- Multi-robot interaction
- Human-robot interaction
  - Speckled Computing and human behaviour modeling

Why robotics?

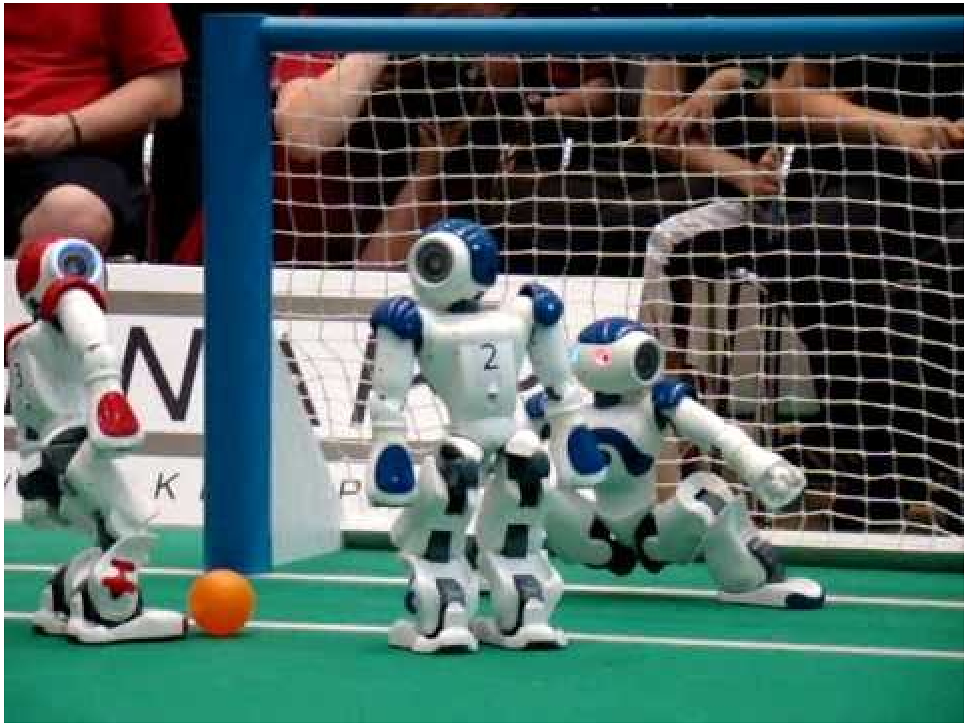
- Robots are needed to perform tasks that humans find:
  - Too difficult
  - Too dangerous
  - Too tedious
  - Or any combination of the above.
- They must do so:
  - Autonomously
  - Robustly
  - Reliably

Why robotics?

Typical problem structure



A slightly different example



Why is robotic soccer challenging?

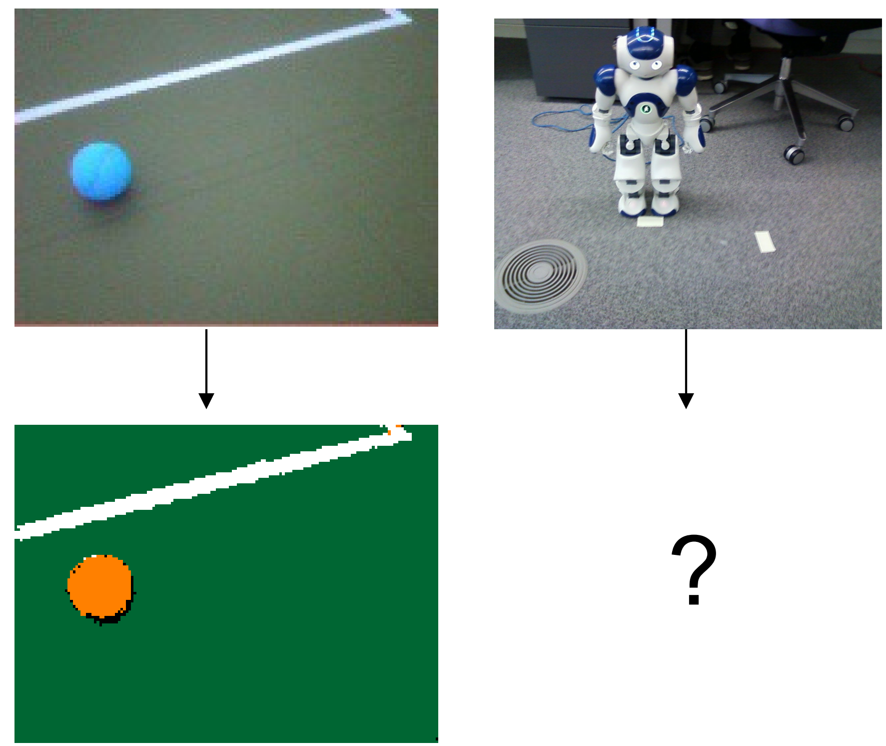
- Uncertainty in sensing
- Uncertainty in actuation
- Partial observability
- Manipulation of physical objects (e.g. ball)
- Game against strategic adversaries, whose exact behaviour is unknown

What makes interaction difficult?

- Uncertainty in sensing
- Uncertainty in actuation
- Partial observability
- Manipulation of physical objects (e.g. ball)
- Game against strategic adversaries, whose exact behaviour is unknown

Sensing and partial observability

- Limited field of view
- Perspective viewpoint – no external information
- Large parts of the environment obstructed or invisible most of the time
- Image processing is computationally complex



Strategic behaviours

- Soccer robots must devise strategies that allow them to outperform their opponents
- Adversarial strategies not known a priori
- Need to develop robust adversarial modeling techniques…
- …in the face of limited processing power, partial observability, etc.

Modeling strategic interactions

- We need a state estimation tool that behaves robustly in partially observable environments…
- …and an adversarial modeling mechanism that ties in with our noisy sensing and actuation models

Modeling strategic interactions

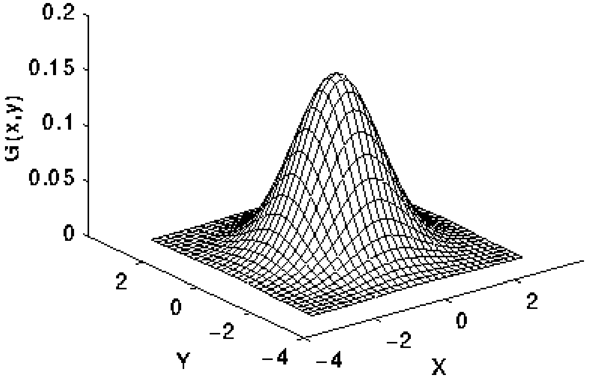
- **Particle filtering:** probabilistic state estimation based on sensor and motion models
- **Reachable set composition:** synthesis of multiple hypotheses on the adversaries' physical *capabilities*

Filtering and state estimation

- **Kalman filtering:** Linear dynamics, system variables drawn from Gaussian distributions
- State model: $\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k$
- Measurement model: $\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k$
- *Prediction* step used to generate a priori state estimate
- *Update* step introduces measurement innovation

Filtering and state estimation

- Problems with Kalman Filters:
  - Dynamics may not be linear
  - Underlying distributions may not be Gaussian
  - System may have multiple *modes*



Particle filters

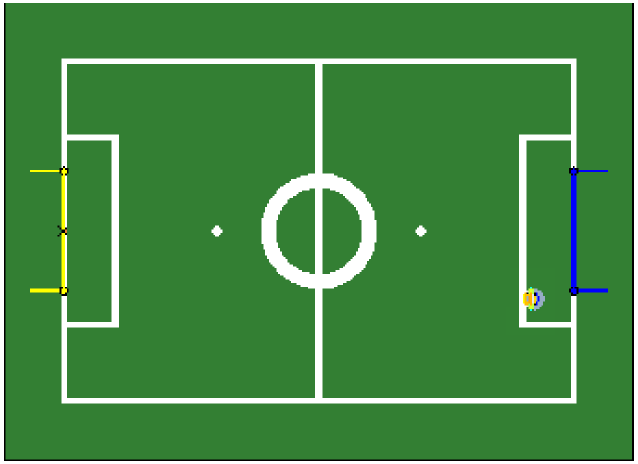
- Each particle corresponds to a distinct state hypothesis, which is updated over time
- Particles also carry a weight representing their likelihood
- Can represent arbitrary distributions
- Filter complexity depends on the number of particles used

Particle filters

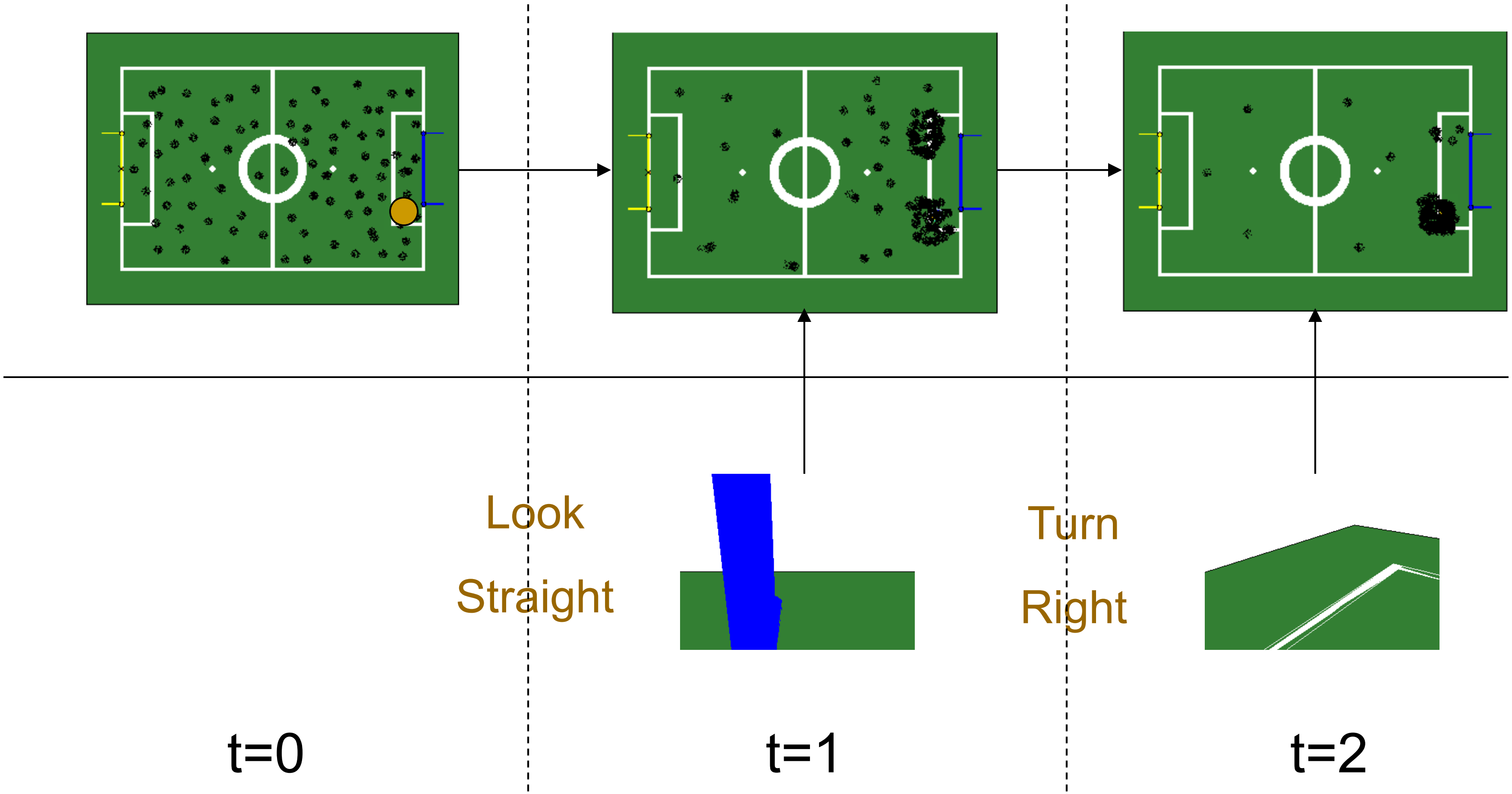
- Initially all particles are uniformly distributed
- **Prediction step:** compute the likely change in the state of each particle based on a *motion model*
- **Update step:** compute the likelihood of each updated particle based on the current *measurement* and *sensor model*
- **Resampling step:** an additional prediction step biased towards the more likely particles

Example: robot localisation

- Robots must infer their own position in the field through a set of fixed features
- Goals, lines, cross spots, etc.
- These features are very ambiguous – need multiple hypotheses



Particle filter example

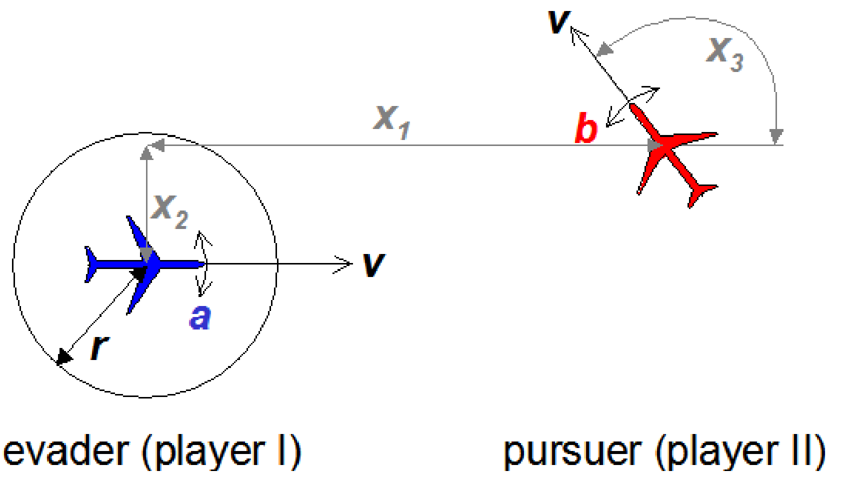


Particle filters

- Can extend to estimating the state of the adversaries (position, velocity etc.)
- Further advantages: partial observability fits nicely into framework
- When no measurements are available, probabilities updated based on previous estimates or additional heuristics

Reachable sets

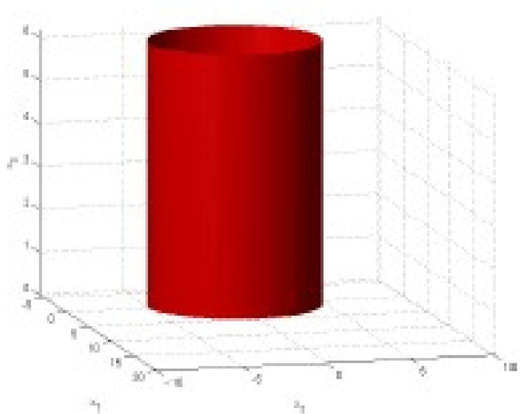
- Inspired from hybrid systems literature
- Classic example: aircraft collision avoidance (Tomlin et al.)



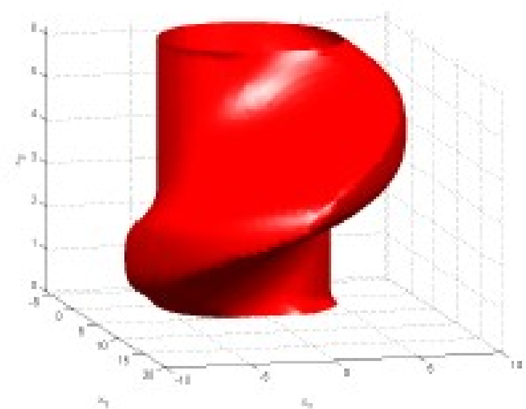
$$\dot{x} = \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -v + v\cos x_3 + ux_2 \\ v\sin x_3 - ux_1 \\ d - u \end{bmatrix} = f(x, u, d)$$

Reachable sets

- Given the system dynamics and the robots' capabilities, compute the control inputs that are likely to avoid/lead to collision in the future
- Solution: Hamilton-Jacobi-Isaacs PDE, solved for a fixed time horizon [0, *t*]



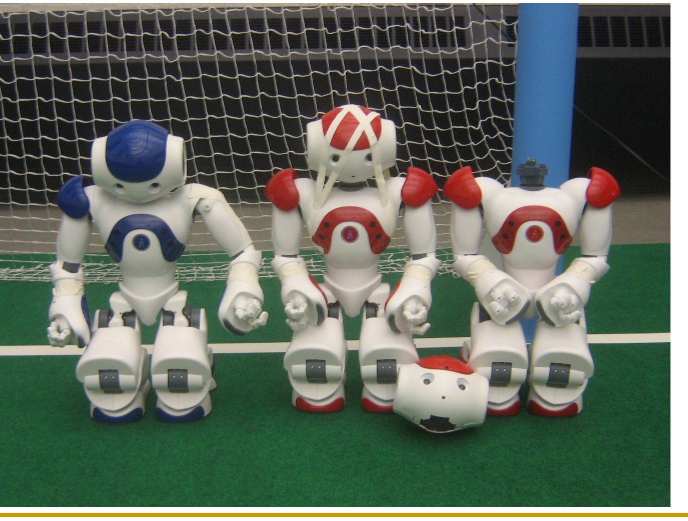
Initial Conditions



Reachable set at time *t*

Adversarial modeling

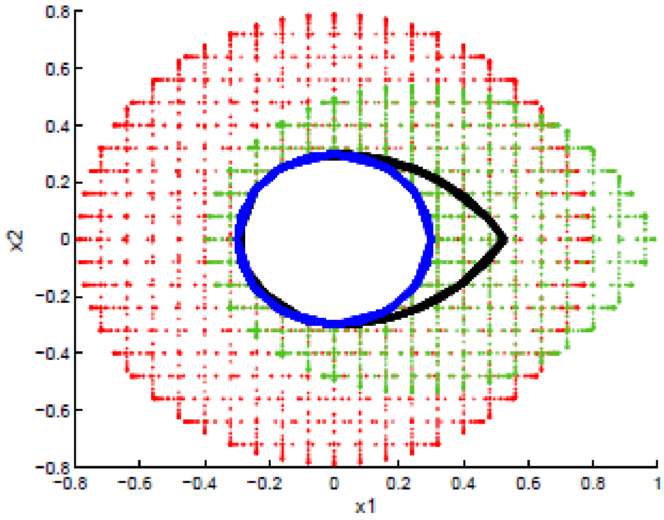
- Each reachable set corresponds to one hypothesis on the adversary's capabilities
- In (robotic) soccer, is it sensible to assume that all (robots) humans are equally agile?

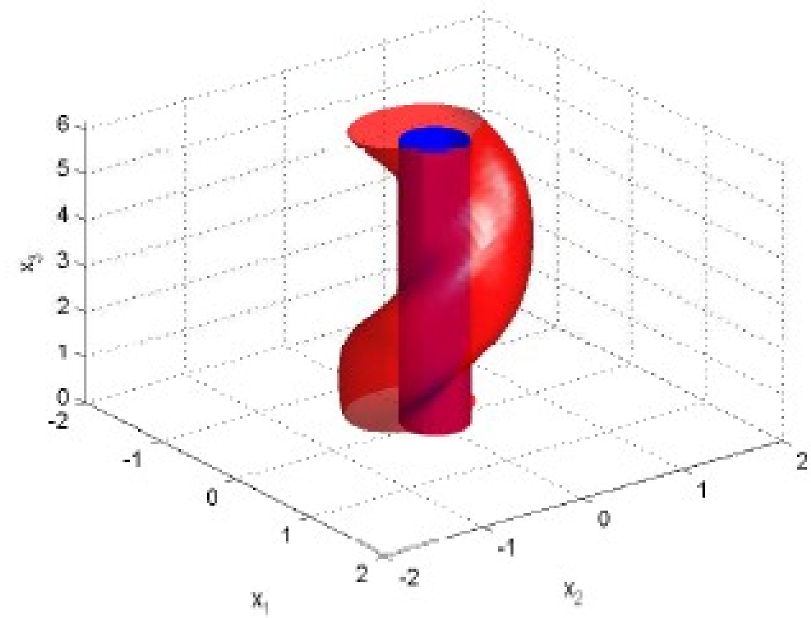


Reachable set composition

- **Offline:** Compute a collection of reachable sets, for different capability hypotheses
- **Online:** Use these sets as templates for adversarial modeling and planning
- At each step, select the set that most closely matches the perceived capabilities of the adversary

Reachable set composition





Putting it all together

- Use particle filters to estimate the state of the other robots
  - Define additional constraints based on the robot's field of view
- Use the precomputed reachable sets to fit the adversary's state
- Demo…

Human-robot interaction

- Many modern robots must interact closely with humans
- We require that these interactions are safe and robust

Human robot interaction

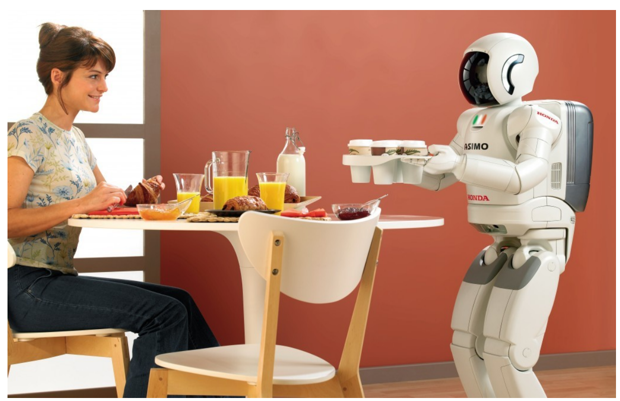
- **Example:** human and robot sharing a kitchen bench
- What happens if they both reach for the same knife?
- Need a negotiation protocol that facilitates this interaction

Human-robot interaction

- From a robot’s viewpoint, humans may behave and react unpredictably
- Need appropriate tools for describing behaviour
- Important for both cooperative and adversarial tasks

Framework

- Can the particle filter/reachable set paradigm be extended?
- In principle, yes!
- Like robots, just treat humans as agents sharing the same workspace



Challenges

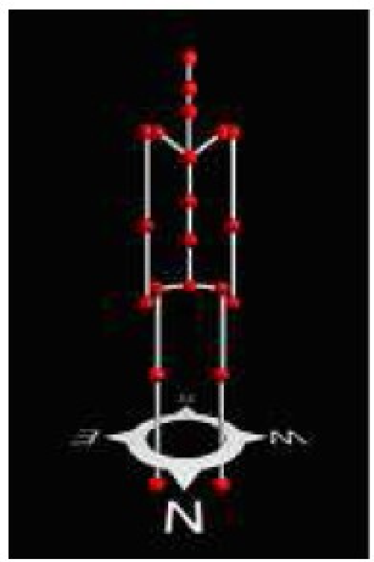
- **Behavioural templates:** more difficult to describe than with robots - humans have more varied and versatile behaviours
- Solution: extend the definition and scope of reachable sets beyond velocity constraints
- Data-driven learning of reachability of unsafe configurations

Challenges

- **State estimation:** which is the current pose of the human? And what other configurations can be reached from that pose?
- Vision: partial observability constraints
- Difficult enough to estimate position from a single cue
- Pose estimation from multiple features significantly harder!

State estimation

- Instead: use an inertial sensing technology like Speckled Computing
- Use multiple sensors to get continuous updates on the human's pose



Example revisited - comparison

- Human and robot sharing a workbench reaching for the same object
- **Vision:** would need to keep track of both the object and the human reaching for it – real-time multi-object tracking hard
- **Speckled Computing:** decouple the two tasks – let the robot focus on the object

Advantages

- High frame rate
- No occlusion
- **Continuous** stream of observations
  - Particle filters work better, no reliance on past observations
- Pose provides much richer information than simple position estimates

Discussion

- Would still need to use both specks and vision
- Problem with frame rate discrepancy – wireless sensing much faster
- Particle filter would require different motion model for various body parts – multimodal reachable sets?

Conclusions

- Multi-robot and human-robot interaction are very challenging open problems
- Robust sensing-planning-acting required
- Speckled Computing could reduce the computational load
- New behavioural modeling and state estimation algorithms for real-time interactions