Bootstrapping robotic sensorimotor cascades

Andrea Censi (G4)   Richard M. Murray

Award # 0931746

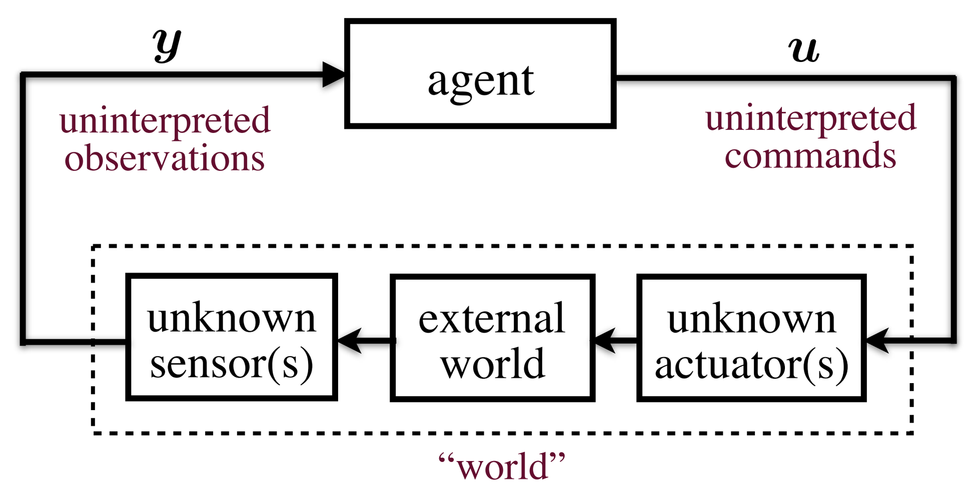
Caltech

The problem

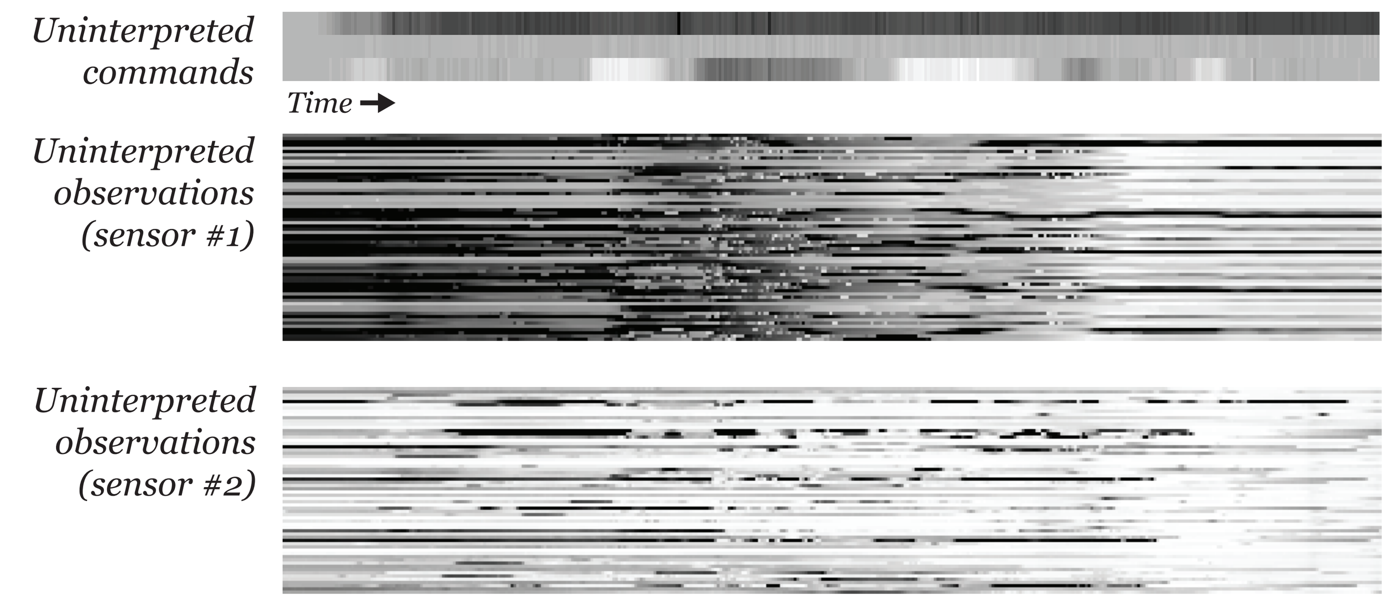
- **Can an agent learn to use any sensorimotor cascade (set of sensors and actuators) from scratch, with no prior information about them?**
We call this problem "bootstrapping".

- The agent "wakes up" connected to **streams of uninterpreted observations and uninterpreted commands**.

- The agent must obtain a model of its own body, and use the model to perform a useful task.



We start from uninterpreted streams of observations and commands.
Which one is the camera and which one is the range-finder?



Motivation

In robotics, like all engineering, there is a constant tension between:

Getting the **best results** possible with the **maximum effort** available.

Aiming for **satisfactory results** with the **minimum effort**.

- Currently, robotics research is quite unbalanced towards the left; we design methods that rely on very precise models of sensors and actuators.

- However, it is unrealistic to rely on accurate modeling of complex systems; models change and fail, and this makes systems unreliable.

- Eventually, we will realize that to obtain reliable robots we should focus on methods that obtain satisfactory results minimizing the prior information requirements.

- Bootstrapping is what we obtain when we take this principle to the extreme: can we design behaviors with absolutely zero prior information?

Major challenges

- **High dimensionality** of the data; we aim for processing raw sensory streams.

- Highly **nonlinear nuisances** corrupting the data (e.g., diffeomorphism, permutations).

This makes classical techniques of system identification not applicable.

- **Behaviorally-relevant state/features** often **not directly observable.**

This makes most techniques from machine learning not applicable.
Creating stateful representations is a major challenge of bootstrapping.

- Nature gives a proof of existence that **a solution exists which uses simple and slow computation.**

Our approach

- Obtain **formal results** in the style of control theory.

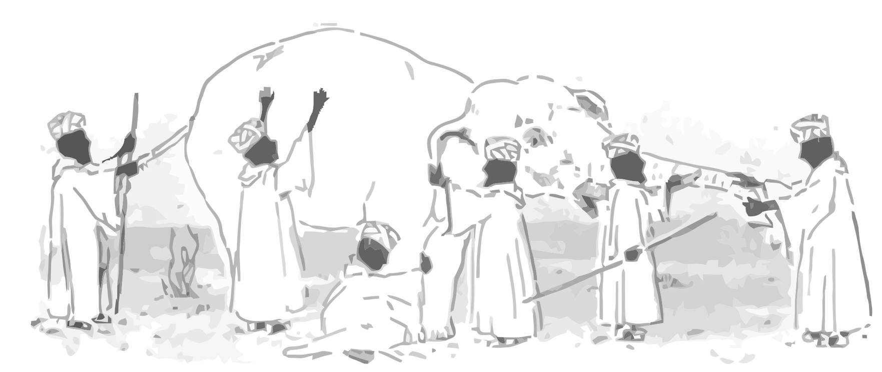
- Study **intrinsic tasks** that are independent of the sensors and actuators (e.g., "servoing").

- Design methods implementable with **slow computing and bio-plausible computation.**

- Design methods that work for a wide range of sensorimotor cascades, parameter-free.

Is bootstrapping equivalent to the full AI problem?

Hopefully not!
While an agent that could solve the bootstrapping problem does display some of the attributes of intelligence, this would be closer to "animal intelligence" rather than human-level AI.



But isn't bootstrapping also...?

Yes! It is a huge problem with many different aspects to it, and it is at the intersection of many scientific communities.

We are focusing on a few aspects related to low-level sensorimotor interaction, that we believe ready for a rigorous formalization and solution.

Necessary invariance properties of bootstrapping agents

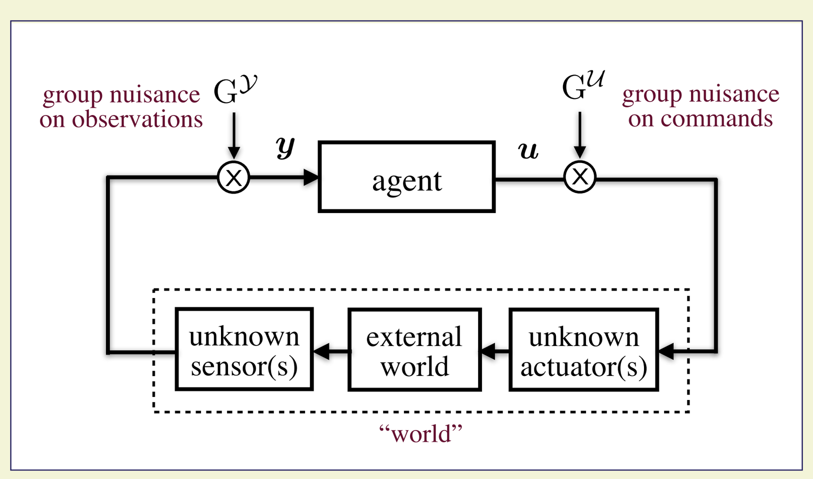
What does it mean that an agent uses "uninterpreted" observations and commands?

- We show that this can be formalized by positing the existence of **representation nuisances** that act on the data, and which must be tolerated by an agent.

- **The classes of nuisances tolerated indirectly encode the assumptions of the agents.**

- The **behavior of an optimal agent must be invariant** to the representation nuisances.

The uncertain semantics of observations and commands is represented by group actions acting on the signals (diffeomorphisms, permutations, etc.) that change the representation but preserve the information.



An optimal agent must compensate those nuisances, as they do not change observability and controllability of the system.

To appear in ICDL 2011.

Bootstrapping bilinear models of robotic sensorimotor cascades

Can we find a unified representation of sensorimotor cascades?

- We consider sensorimotor cascades composed by omnidirectional kinematics and **three "canonical" exteroceptive sensors: field samplers, range-finders, and cameras.**

- We study **bilinear dynamics systems** as generic approximators for such cascades.

- We design an agent that can learn such models and use them for solving the same task (servoing) for any sensor.

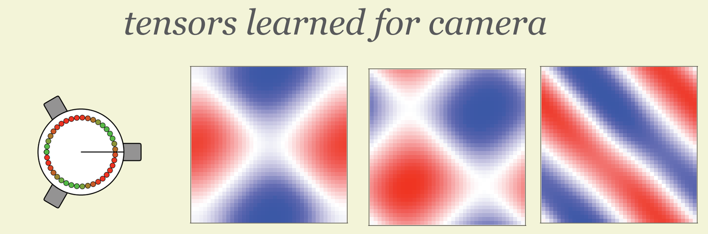
- We approach the problem from a control theory perspective, with the aim of obtaining strong theoretical results.

We use bilinear dynamics sensors (BDS) as a general representation for sensorimotor cascades.

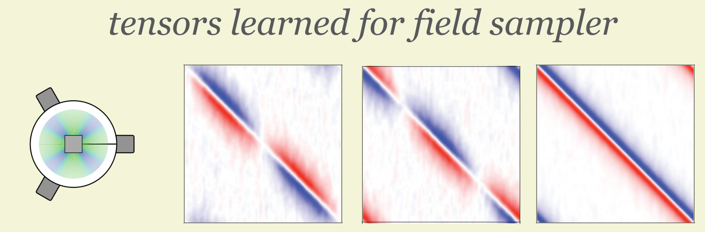
$$\dot{y}^s = \mathsf{M}^s_{vi} y^v u^i$$

A system with n observations and k commands is represented by k tensors of size (n,n).

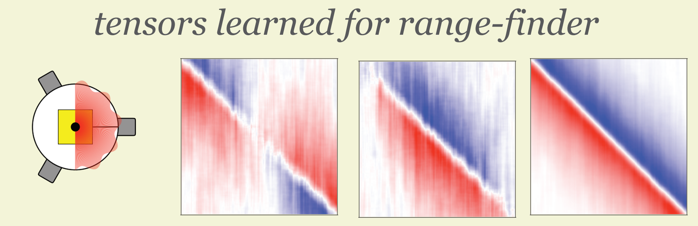
tensors learned for camera



tensors learned for field sampler



tensors learned for range-finder



Appeared in ICRA 2011.

Towards more efficient models (bilinear gradient dynamics)

- BDS models (studied in the ICRA'11 paper) have quadratic complexity in the number of sensels.

- BGDS models are a subclass in which **the dynamics depend on the gradient** of the observations.

- Assumptions verified for the three "canonical" sensors (camera, range-finder, field sampler).

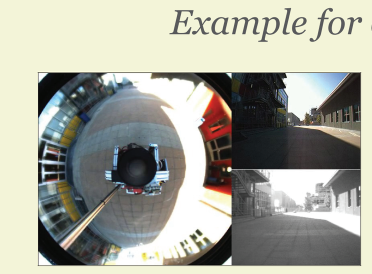
- This makes it possible to feed the model the **raw sensory streams** from real robotic platforms and use **massively parallel and slow computing.**

We assume that observations are functions on a manifold (e.g. visual sphere), and the dynamics, bilinear in the gradient and commands, is parametrized by two tensors G and B:

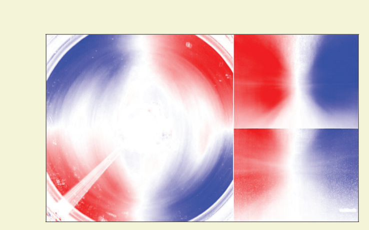
$$\dot{y}(s,t) = (\mathsf{G}^d_i(s)\nabla_d y(s,t) + \mathsf{B}_i(s))u^i(t)$$

This model allows learning with a computational costs linear in the number of sensels.

Example for camera data

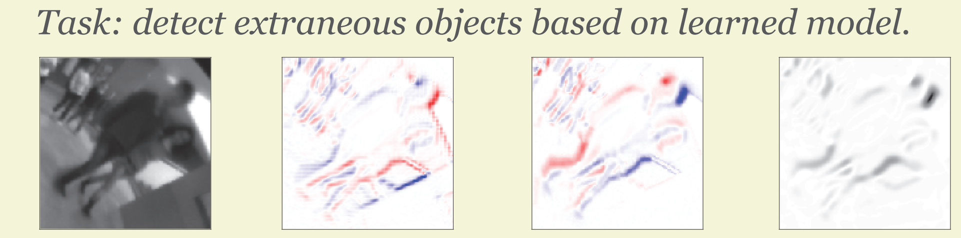


camera frame



a slice of the tensor G

Task: detect extraneous objects based on learned model.



To appear in IROS 2011.