Remote Inspection:
Some Problems with Autonomy
(and Regulation/Verification)

Michael Fisher
Department of Computer Science, University of Manchester
January 2021

The University of Manchester

https://web.cs.manchester.ac.uk/~michael

Royal Academy of Engineering

Who am I?

Standards:

- BSI AMT/10 Robotics (part of ISO/TC 299)
  `https://standardsdevelopment.bsigroup.com`
- IEEE P7009 Failsafe Design of Autonomous Systems
  `https://standards.ieee.org`

Verification:

- UKRI Trustworthy Autonomous Systems Verifiability Node
  `http://verifiability.org`
- IEEE Technical Committee on Verification of Autonomous Systems
  `https://www.ieee-ras.org/verification-of-autonomous-systems/`

RAI Hubs:

- FAIR-SPACE - `https://www.fairspacehub.org`
- ORCA - `https://orcahub.org`
- RAIN. - `https://rainhub.org.uk`

What I’m Going To Say….

1. “My area is special”
2. Autonomy can be the answer
3. Autonomy is scary!
4. Nothing new here?

“My area is special”

Obviously there are different environments - nuclear, weather, temperature, etc.

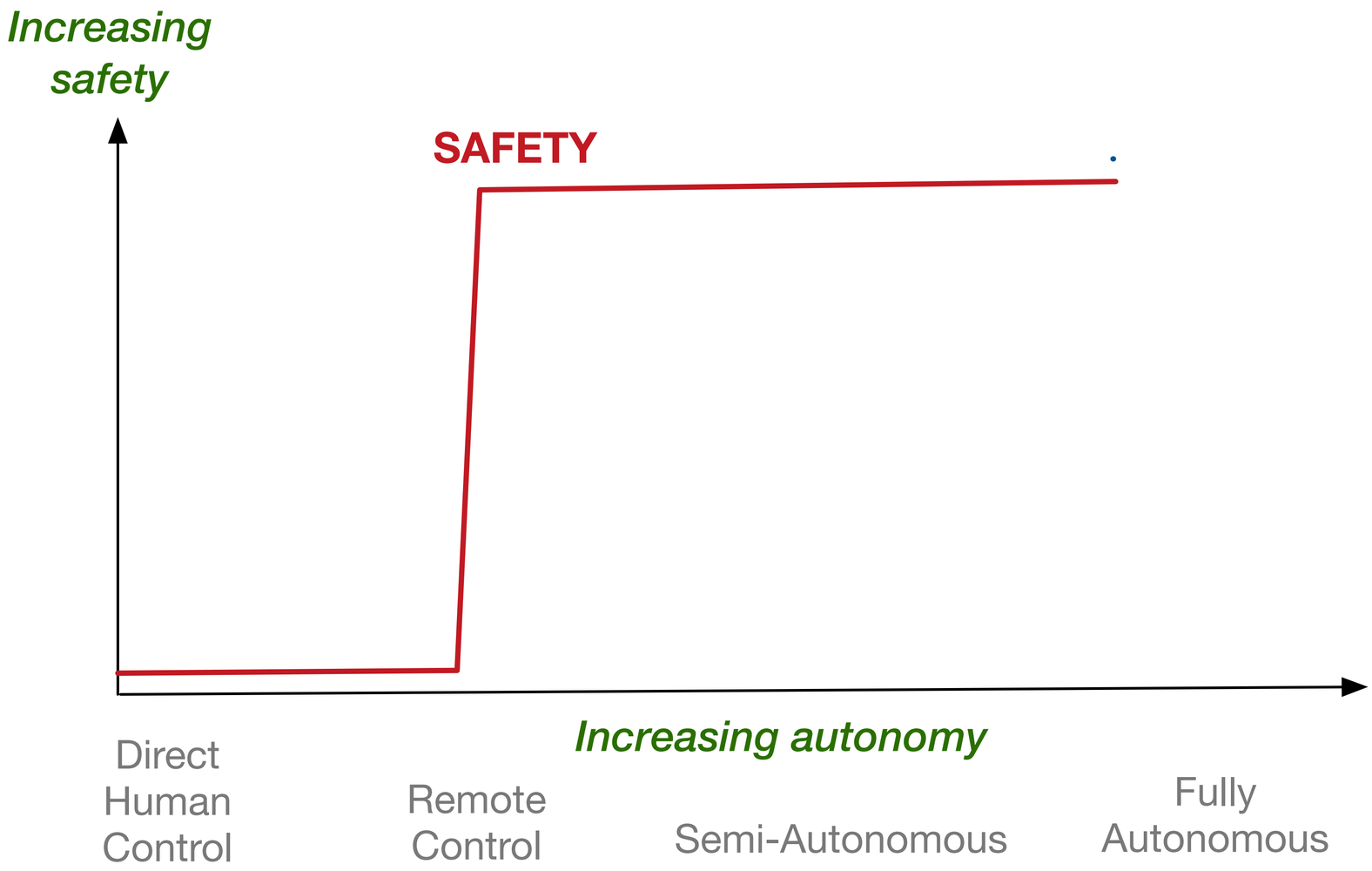
But many elements are similar even across these

- Navigation mechanisms
- Object recognition
- Environmental visualisation
- Generic HRI aspects
- Decision-making
- General-purpose planning
- ….

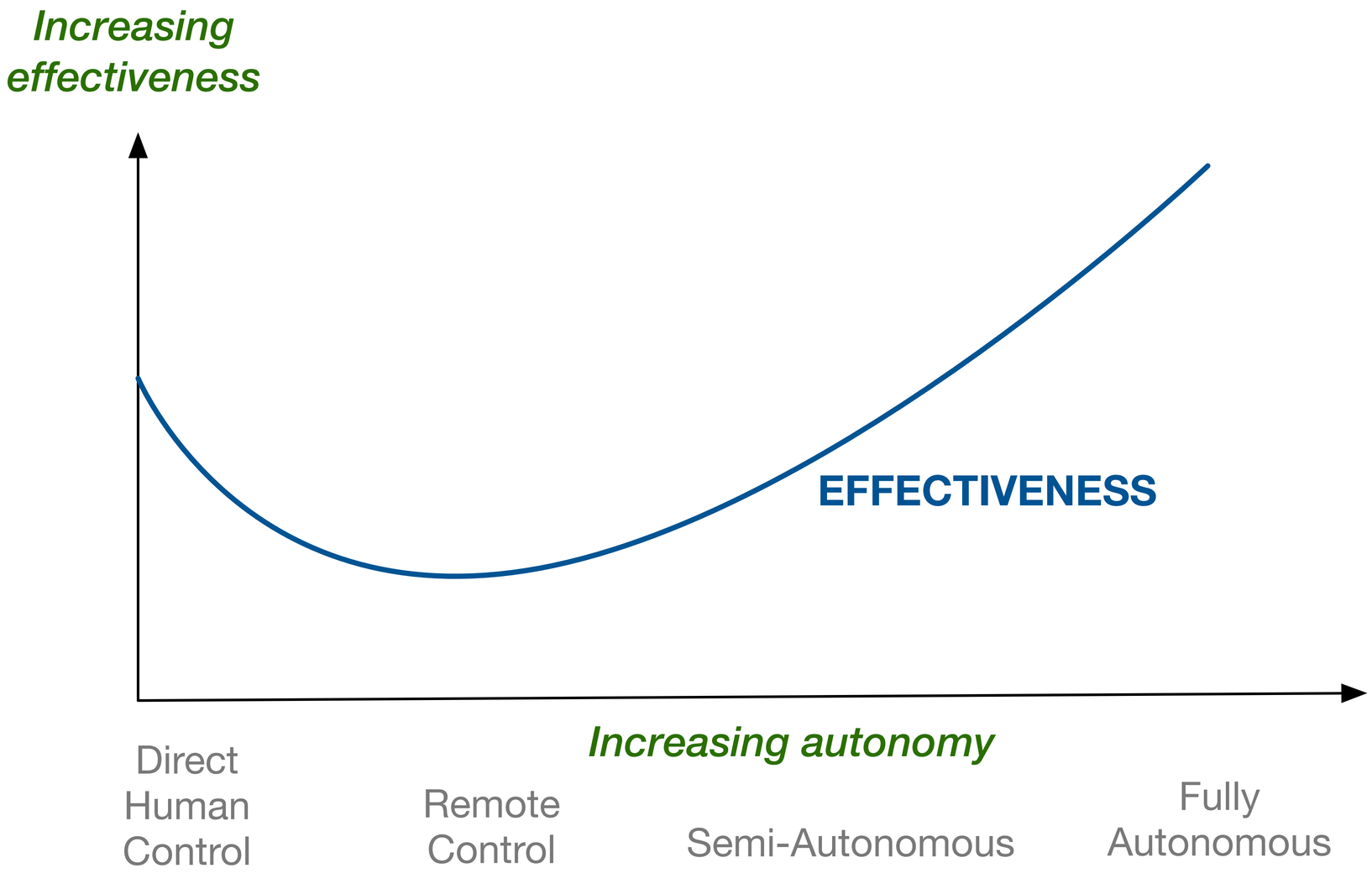
Robot inspection is not so different across sectors

1. Should make use of modularity (ROS, ISO standards, etc) and re-use ***much more***
2. Architectures are ***vital*** - transparency, compositionally, verification, responsibility, …

Autonomy is the Answer



Autonomy is the Answer



Autonomy can be the Answer

Remote control is difficult:

- Responsiveness
- Awareness
- ....

Semi-autonomous systems are better:

- Select an area to inspect and let robot get there itself
- Robot is rarely better/quicker than (distant) human operator
- ....

Autonomy has further advantages:

- Just tell robot what issues to look for and let it explore
- Control/comms links will always fail so must not rely on human control
- ....

Autonomy is scary!

We've seen those Science Fiction films and letting a robot make decisions is dangerous!

- Not if you know how it makes decisions and can provide strong/formal verification.

If a robot uses Machine Learning, etc, there's little we can do about verification

- Maybe, but that's why ***architectures*** are important
- We should ensure that key decisions are only made by strongly verifiable components
- Hence hybrid architectures are crucial:
  - symbolic components for decisions/verifiability/explainability;
  - sub-symbolic components for efficiency/flexibility.

Nothing New Here?

Do you know enough about the environment to describe everything that can go wrong?

- If so, you probably don't need autonomy as all decisions/situations can be pre-scripted
- In this case there's nothing new here - current verification/regulation techniques suffice

But if you are deploying into an unknown environment

- Can't identify every fault/problem/scenario that will occur
- …. and so many traditional approaches aren't sufficient

What can we do?

- Want systems that will make decisions for the `right' reasons in unanticipated situations
- Don't verify/regulate the *decisions*, verify/regulate the *decision-making process*