Why is Building Robust and Reliable Robotics Software So Hard?

Prof. Jonathan Kelly
University of Toronto Institute for Aerospace Studies

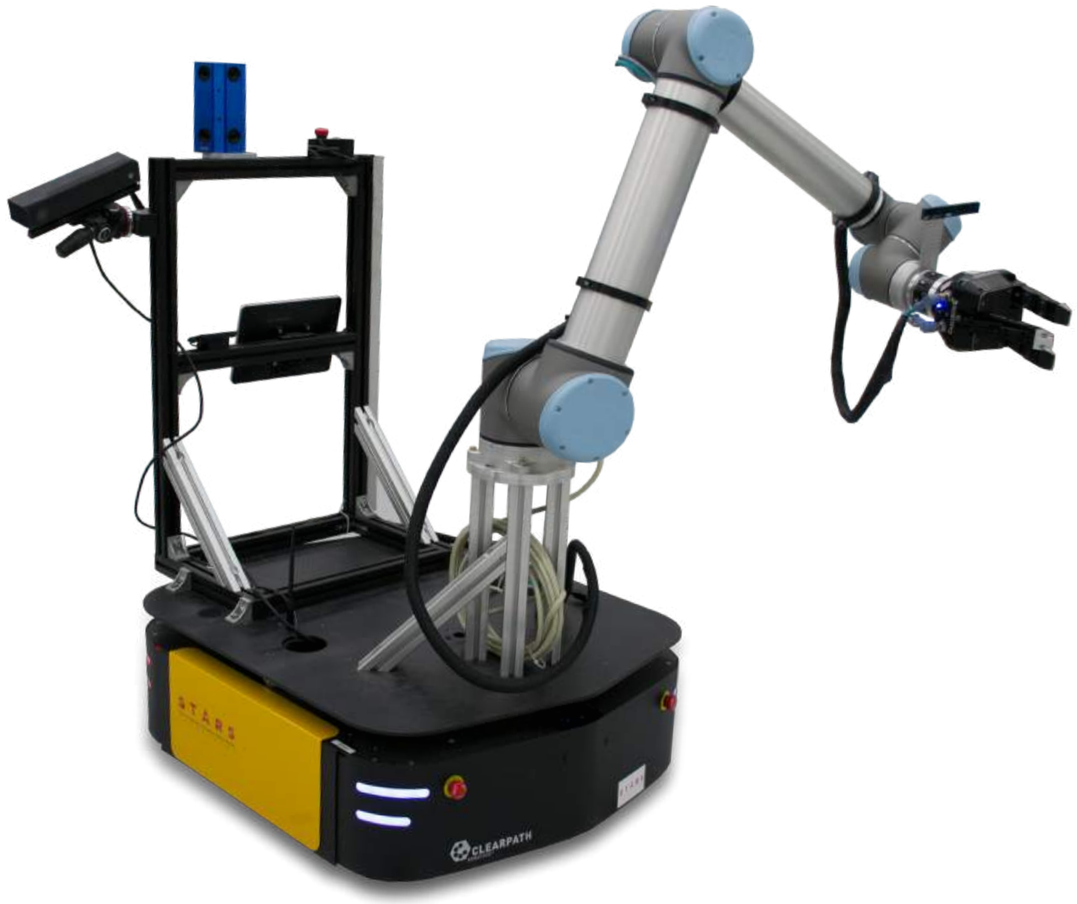
CANARIE Research Software Conference Talk

May 28, 2019

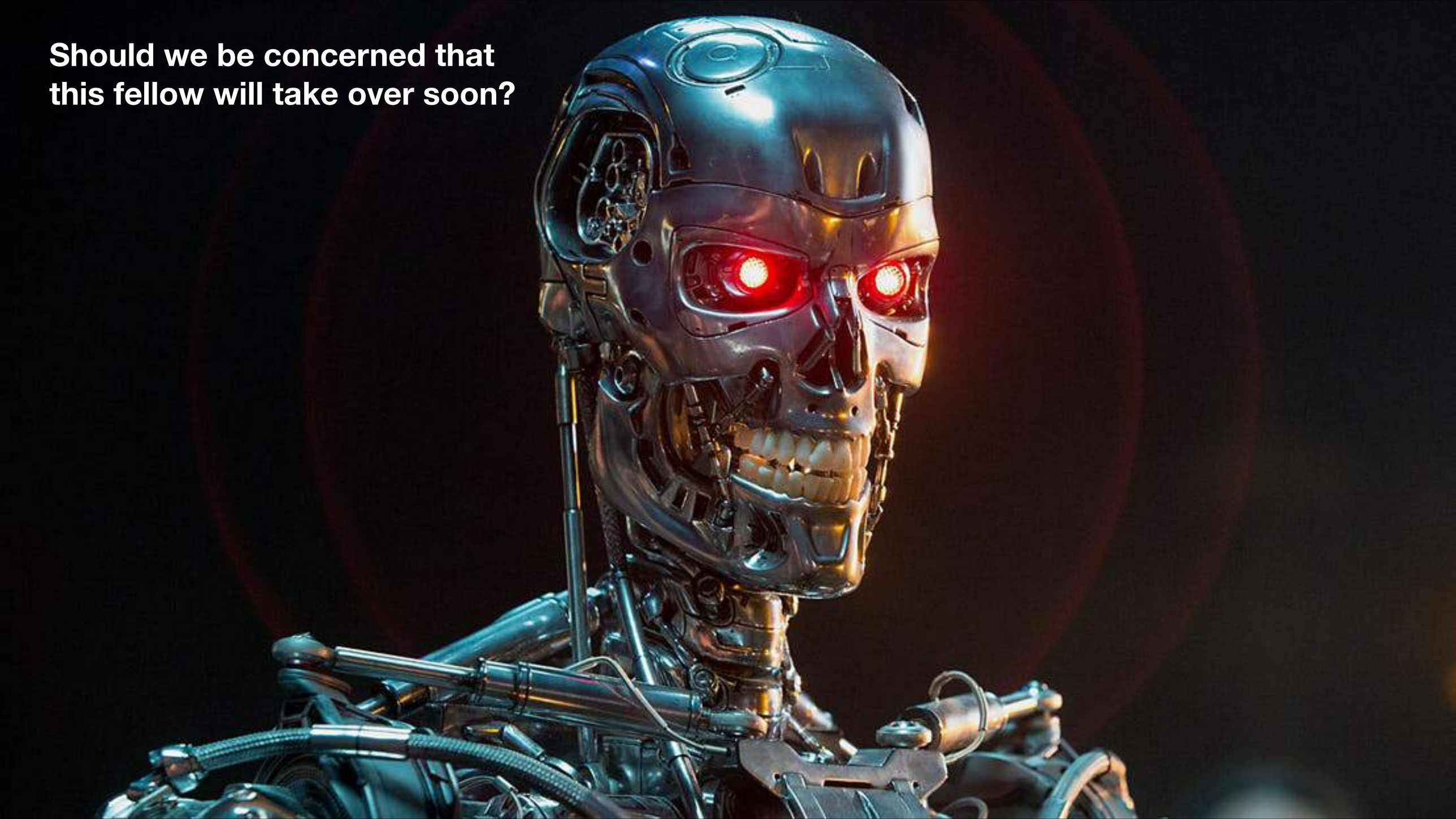
Institute for Aerospace Studies
UNIVERSITY OF TORONTO

My Background and Research Areas

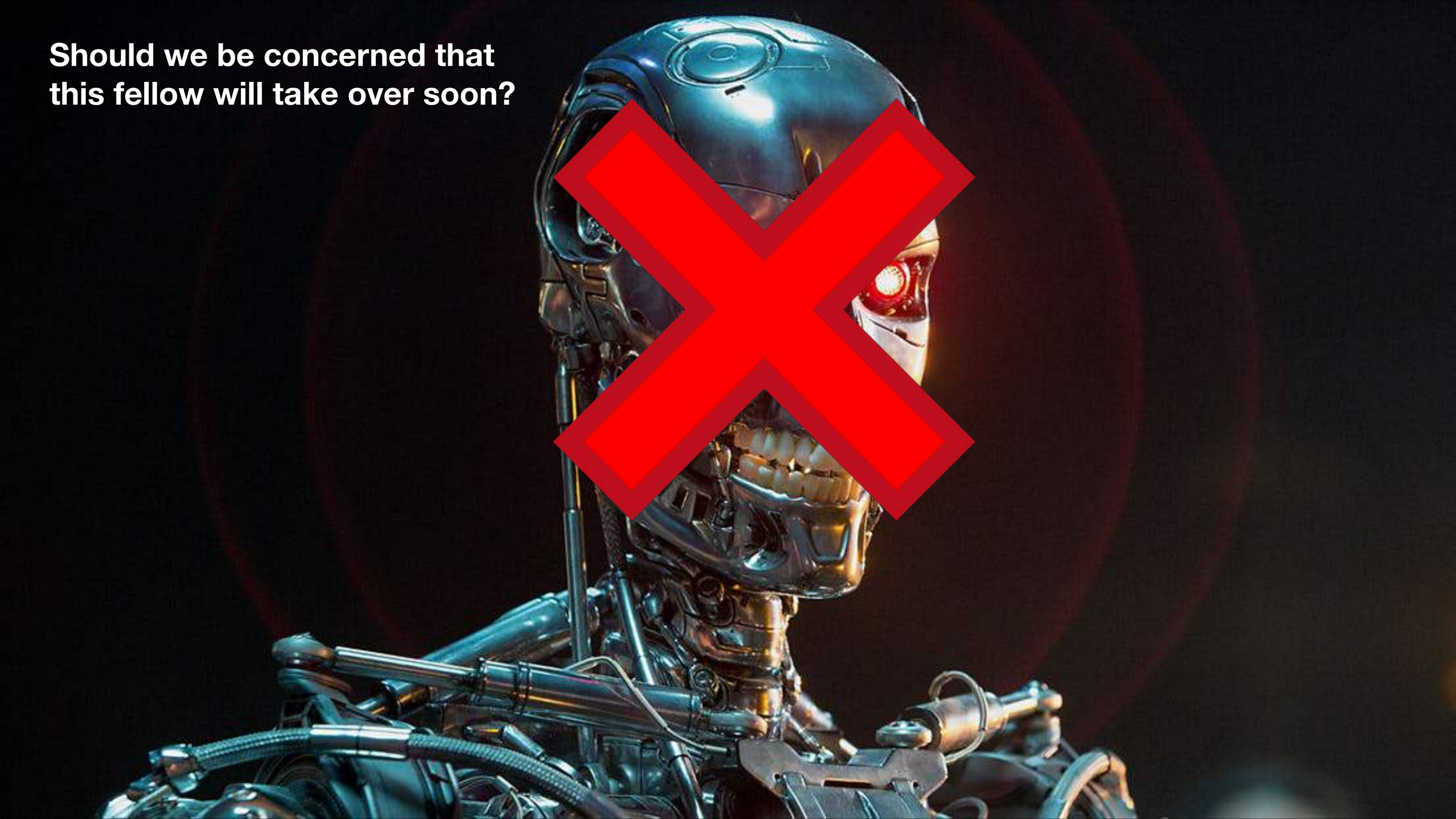
- My degrees are all in Computer Engineering or Computer Science
  - Began in Canada, rotated through the US, back to Canada
- Direct the Space & Terrestrial Autonomous Robotic Systems Laboratory at U of T
- Research focus is on sensing, perception, navigation, mapping, and manipulation
  - Spend a lot of time working on **collaborative robots**, or '**cobots**'



Should we be concerned that this fellow will take over soon?

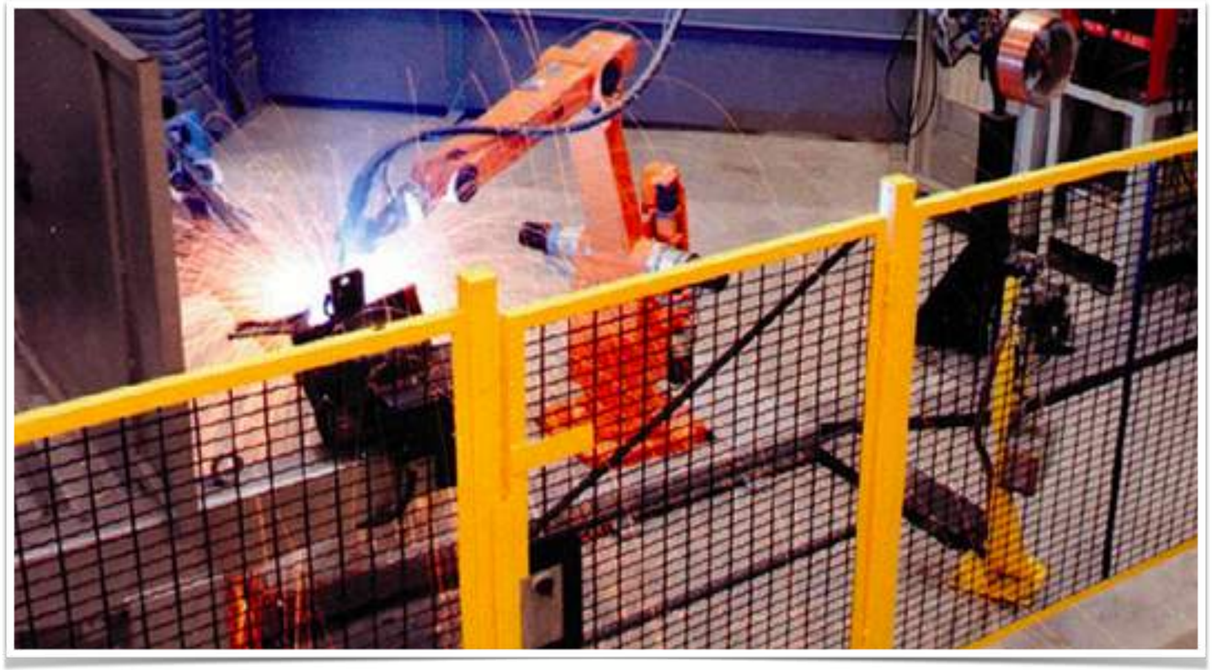


Should we be concerned that this fellow will take over soon?



A (Very) Brief History of (Bad) Robots

- The first industrial robot, the **Unimate**, created by American inventor George Devol, was deployed at a GM automative assembly plant in New Jersey in 1961
  - This (4,000 lb) machine operated behind a safety cage, far away from workers, just as most factory robots do today
- Despite the cages, and sophisticated (relatively) safety systems, accidents still happen…
  - At least 7 factory deaths have been attributed to robots since 1979
  - In all cases, the cause was *not* failure of the mechanical system



Hardware and Mechanism Design | Robust and Reliable

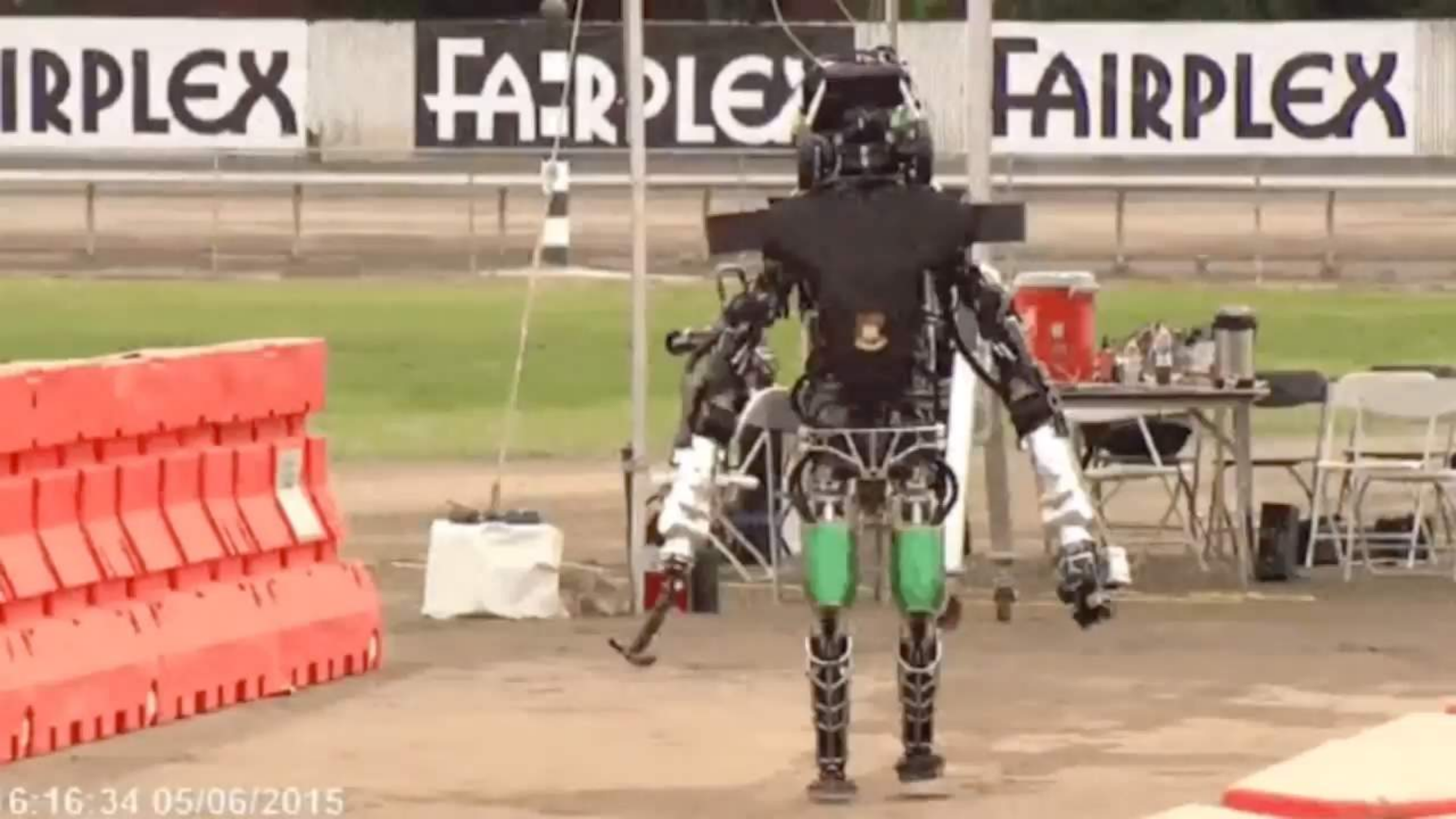
- Gears are one of the oldest inventions, going back 2,000+ years in various forms…
- The harmonic drive (strain wave gearing), which eliminates backlash, was developed in the 1950s…
- We have excellent models of mechanisms, including failure modes and statistics (e.g., MTBF)…
- The electric motor was introduced in the 1830s, and was in practical use by the 1860s…
- Series elastic actuators, which incorporate compliance (via a flexure), are now very popular…
- *These components and their interactions are well understood*

Image: Hebi Robotics

So Where is the Problem? The Devil's in the Software

- The design, development, and test of robotics software is inherently *very* difficult
  - Robotics software tends to be **1) concurrent, 2) distributed, 3) embedded, 4) real time,** and **5) data intensive**
    - These are all *scary* things when robustness, reliability, and safety are required…
- Article from 10 years ago noted "there is a pressing need to engineer the software development process to reduce the cost and time-to-market," while ensuring above requirements are met
- Testing, in particular, is non-trivial because robots are physical (interactive) systems and hence must be deployed in the world

Image: Boston Dynamics


FAIRPLEX
FAIRPLEX
FAIRPLEX
6:16:34 05/06/2015


NOT AN EXIT
CAUTION
Boston Dynamics

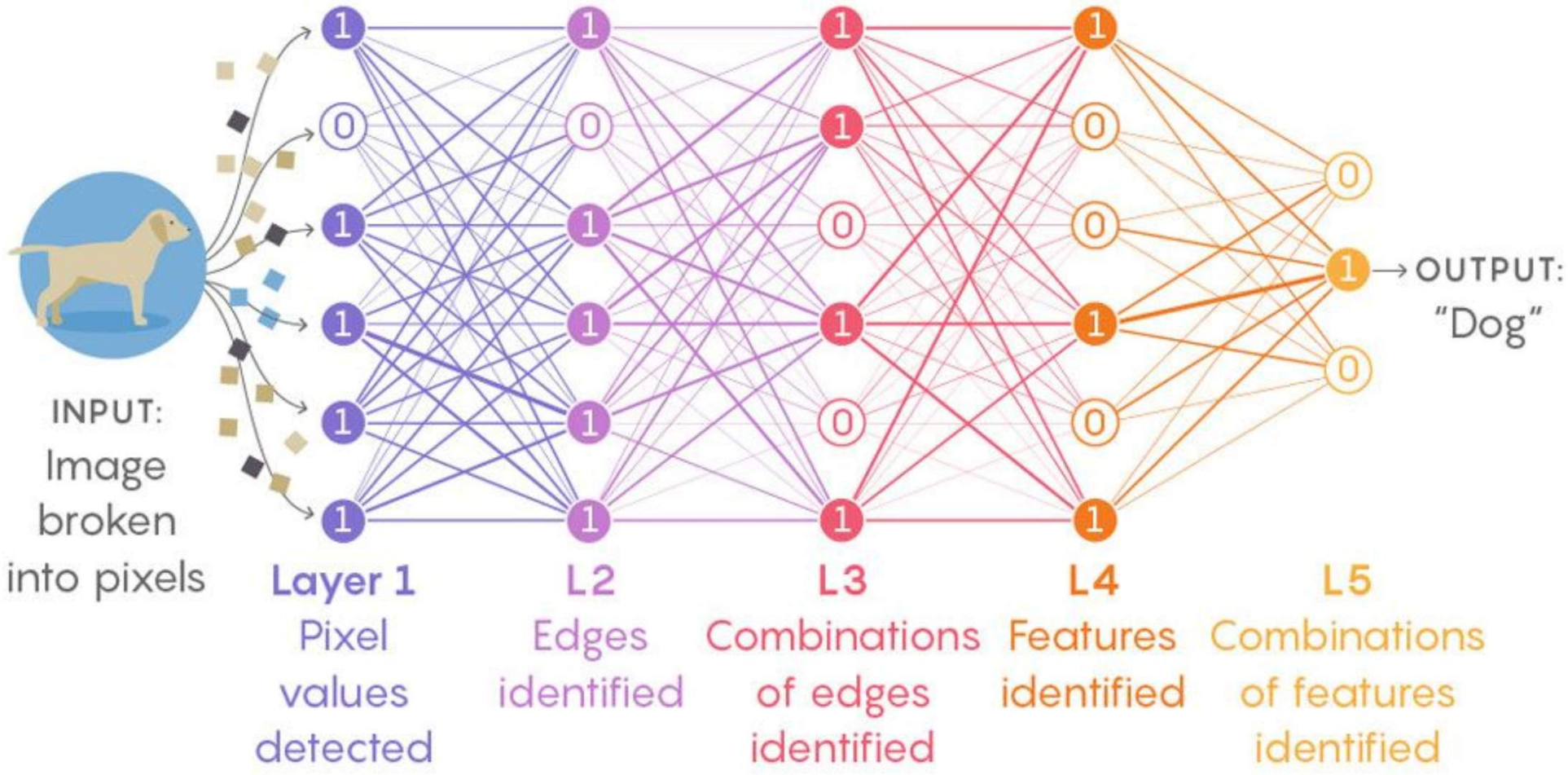
What Happens When Robots Leave Their Cages?

- We are entering an era in which robots are not, or will not in the near future, be behind safety cages
  - ‘Cobots’ are appearing alongside workers in manufacturing, logistics, picking, etc.
  - Self-driving cars have been tested over millions of kilometres on public roads (less in Canada…)
  - Aerial drones are now in regular use in many industries (film/TV, surveying) and might soon deliver packages to your door
- *Robots are (incrementally) becoming parts of our daily lives, and so are interacting with people much more regularly*

Challenges | Software Complexity

- The software that runs modern robotic systems can be incredibly complex, and huge
  - Consider the F-35 codebase: **8 million lines**
  - Statistics on the size of the codebase for, e.g., a Waymo vehicle are hard to come by…
- Estimates of the number of bugs per line of code (or defects per KLOC) vary…
  - *Code Complete* estimated 15—50 / 1000 lines
  - Microsoft claims 0.5 / 1000 lines in production
  - NASA achieved zero for the space shuttle, but at a cost of $1,000's of dollars **per line**

Challenges | Machine Learning and Interpretable AI



Challenges | Machine Learning and Interpretable AI

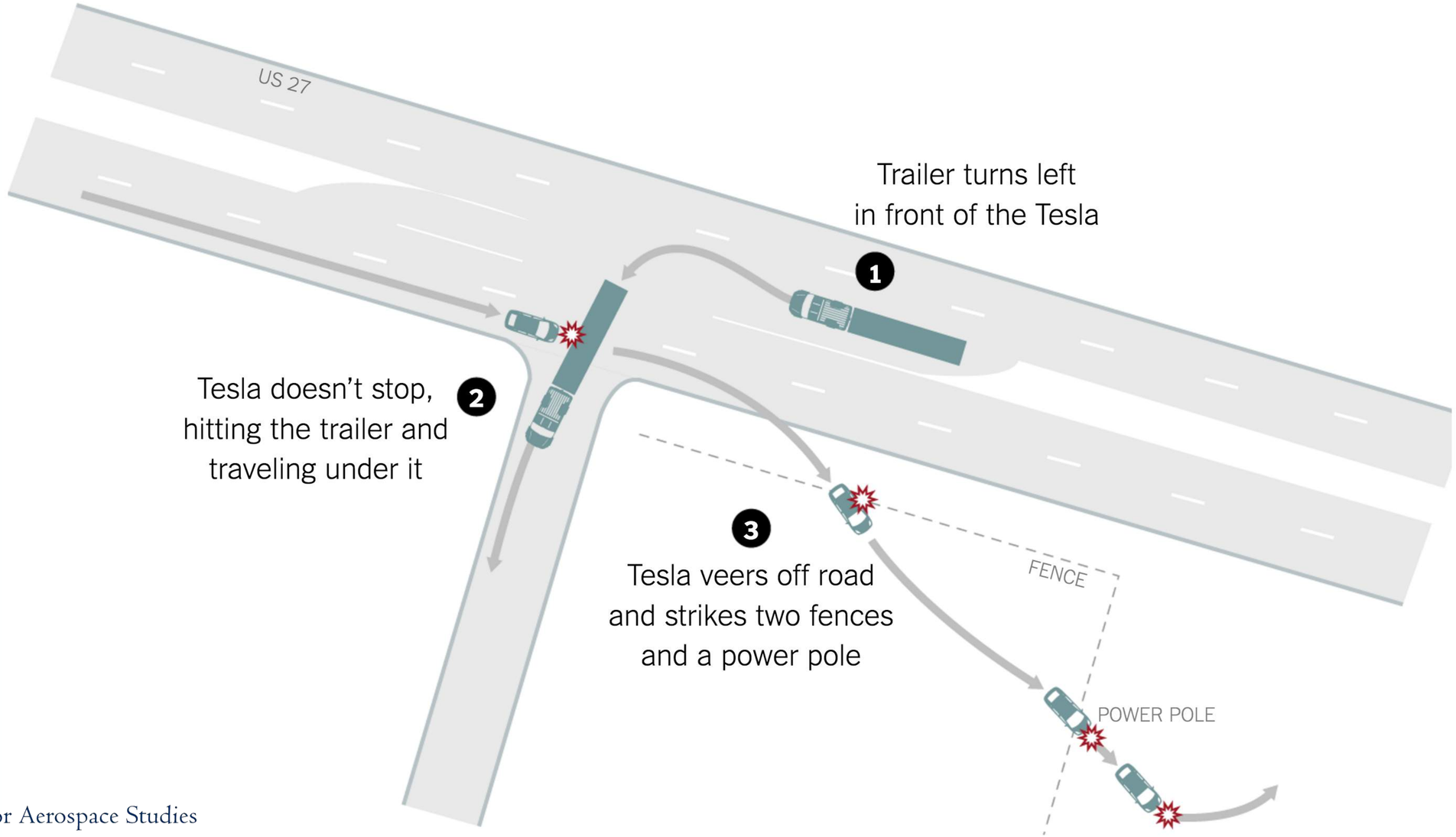
Challenges | The Human Element

- *Most* of the time, human beings do things that are *mostly* predictable (following traffic signals, pedestrian indicators, instructions, etc.)
- *Sometimes*, human beings do irrational and largely *unpredictable* things
  - Driving on the wrong side…
  - Swerving into traffic on a bicycle…
  - Jumping over a safety barrier…
- These events are very difficult for robots to handle! E.g., there may be little or no training data (for learning)

Failures in Edge Cases | Sensor Fusion Gone Wrong

- **Example:** Tesla Model S owner (and fan) Joshua Brown killed May 7, 2016 in Williston, Florida, with Tesla Autopilot system engaged and driving
  - Large 18-wheeler turned left in front of Tesla on divided highway
  - Neither Autopilot nor Brown applied brakes or steering, car drove full speed under trailer “with the bottom of the trailer impacting the windshield of the Model S”
- Tesla’s explanation: “Against bright spring sky background, vision system failed to distinguish white trailer cross-section, while radar mistook trailer for overhead road signage…”
  - Note that Elon Musk has said “LIDAR is a fool’s errand”

Failures in Edge Cases | Sensor Fusion Gone Wrong

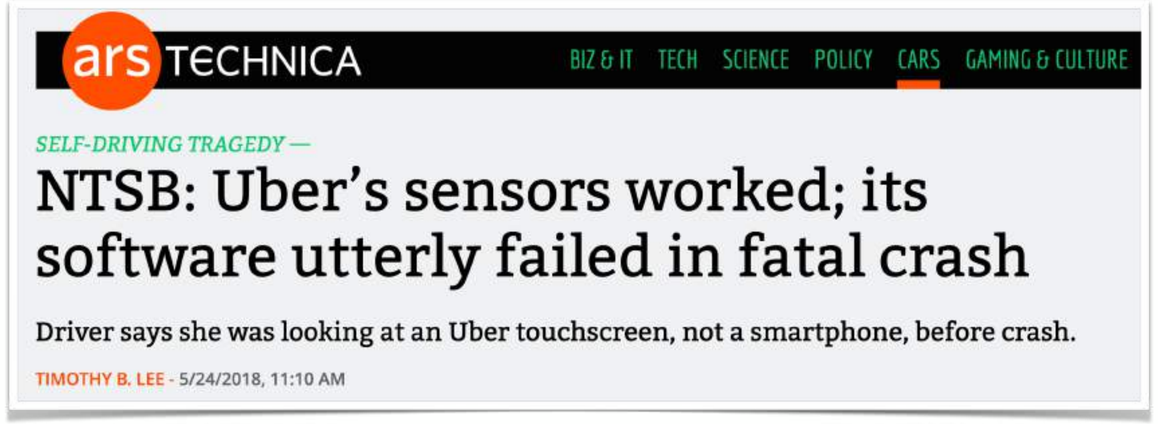


Institute for Aerospace Studies
UNIVERSITY OF TORONTO


And in March 2019, it happened again…

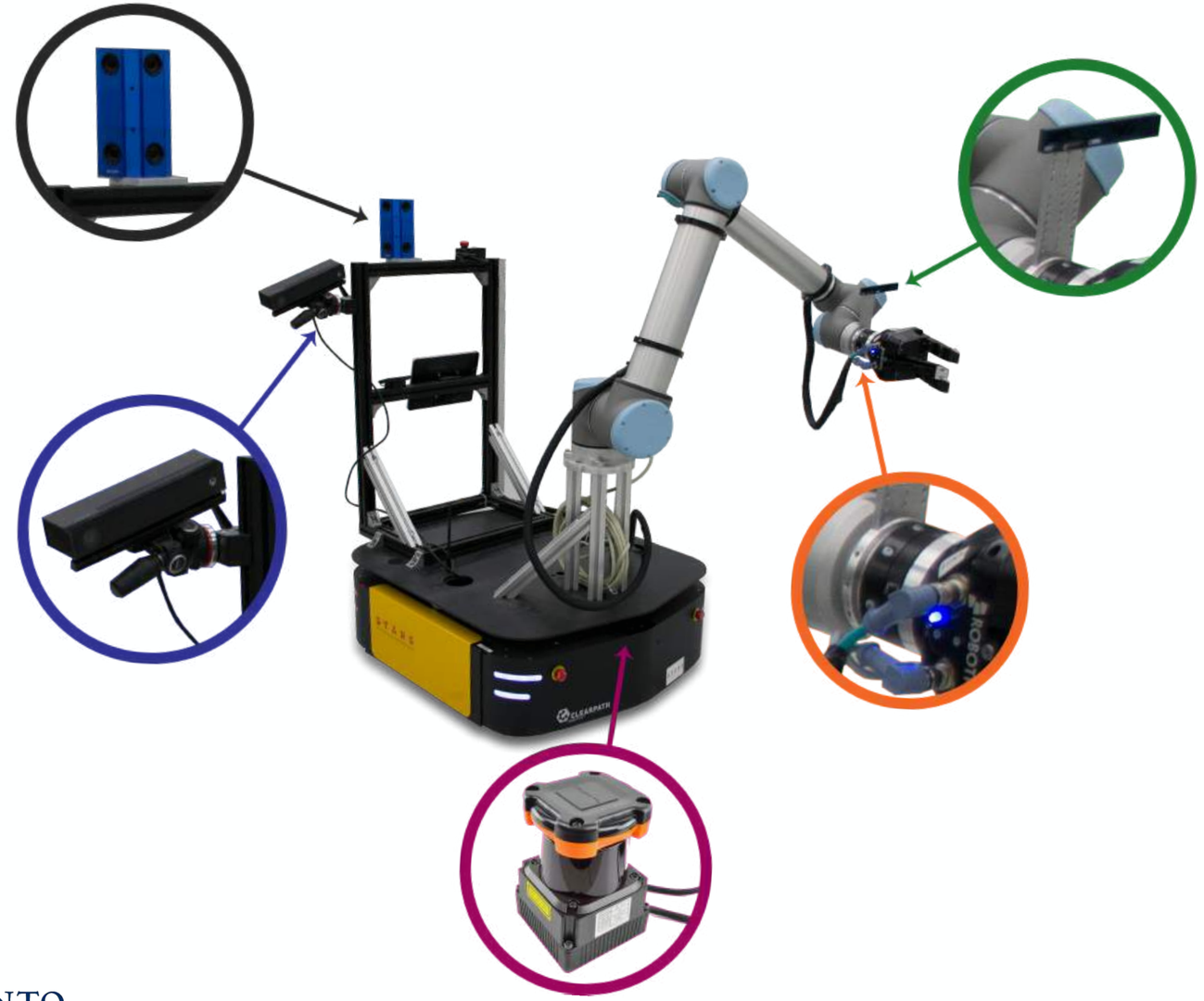
Failures in Edge Cases | Interactions Gone Wrong

- **Example:** Fatal crash involving an Uber self-driving SUV in Tempe, Arizona, March 2018, while in autonomous mode, with safety driver on board…
  - Elain Herzberg, 49, was killed while walking across a highway at night with her bicycle by her side
- Sensors on the car (LIDAR and radar, but not the cameras) did detect the woman, but did not correctly identify her (*unknown object* → *car* → *bike*)
- At 1.3 seconds before impact, software determined an emergency maneuver was needed—*disabled* on Uber cars
  - And the safety driver was *not* alerted; driver was watching her phone…




NTSB

Towards Robustness, Reliability, and Safety | Rich Sensory Systems



Institute for Aerospace Studies
UNIVERSITY OF TORONTO

Towards Robustness, Reliability, and Safety | Formal Verification

- Formal verification involves proving the correctness of a piece of code
  - Answers the question: “Have we made what we were trying to make?”
  - Tractable for some algorithms, but remains very difficult for robotics software due to: *concurrency*, *distributed processing*, *embedded nature of code*, *real time requirements*, and *heavy data processing needs*
- This is an active area of robotics research, and new methods are certainly needed
  - Learning systems, in particular, are very difficult to verify (as discussed previously)
  - Predicted to be a hot topic going forward

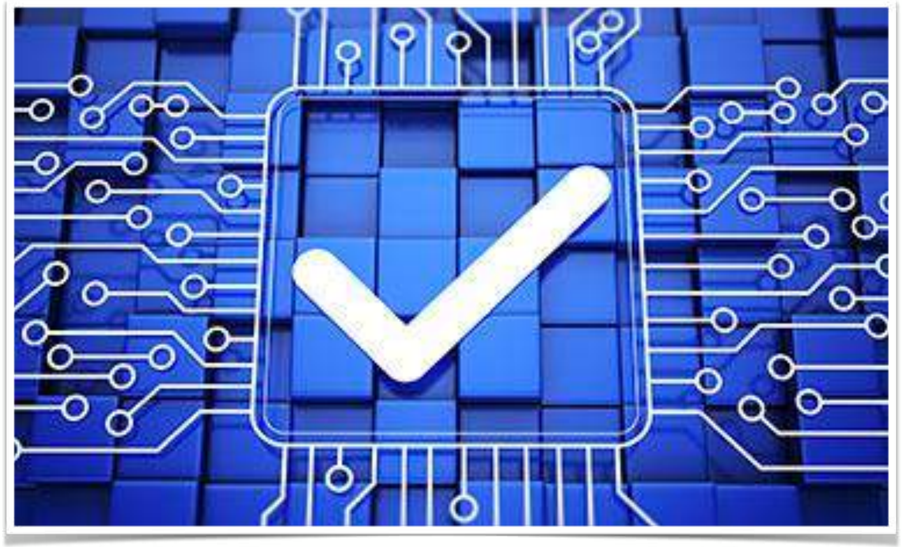
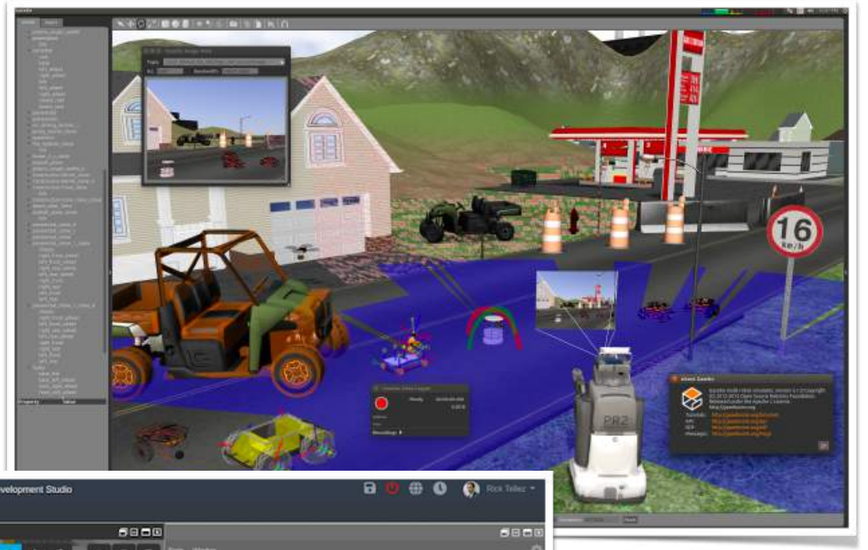
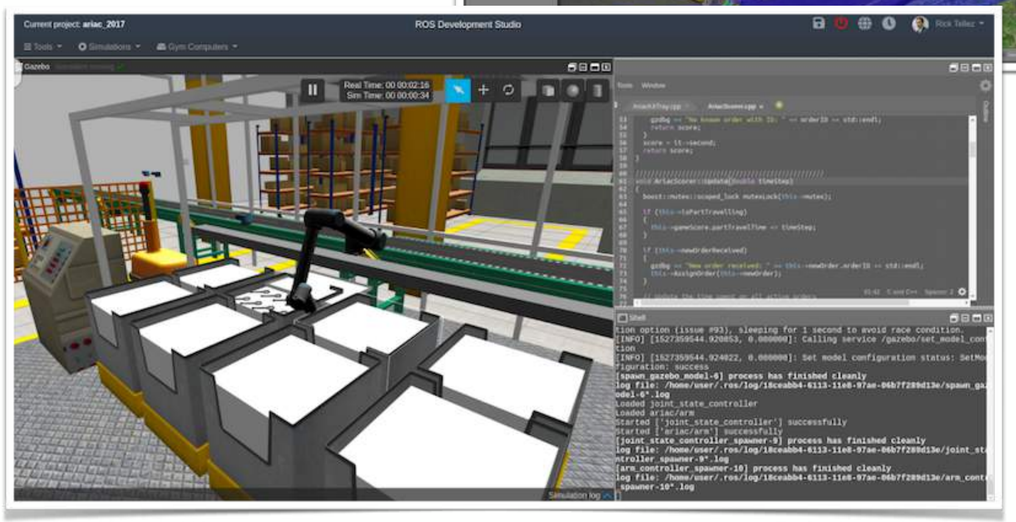


Image: edX

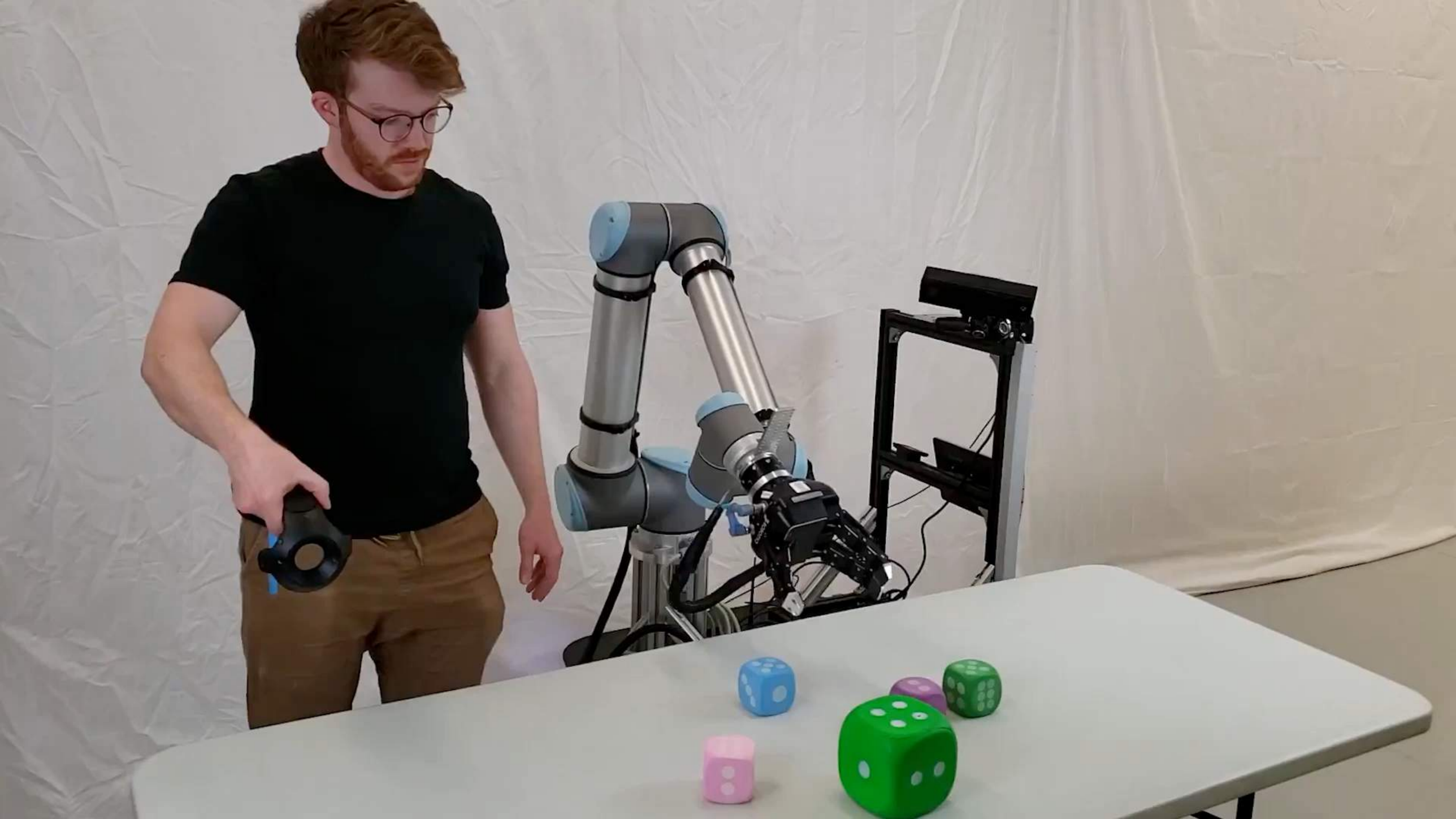
Towards Robustness, Reliability, and Safety | Simulation

- Real-world robot testing is difficult, time consuming, and expensive…
- A promising avenue is to develop better and more realistic simulation software (for learning, etc.)
  - Many robot simulators and physics engines already exist, but there is still a need to better emulate, e.g., grasping and contact
- In addition to various open source efforts, numerous companies have become involved (e.g., NVIDIA) for domains such as autonomous driving

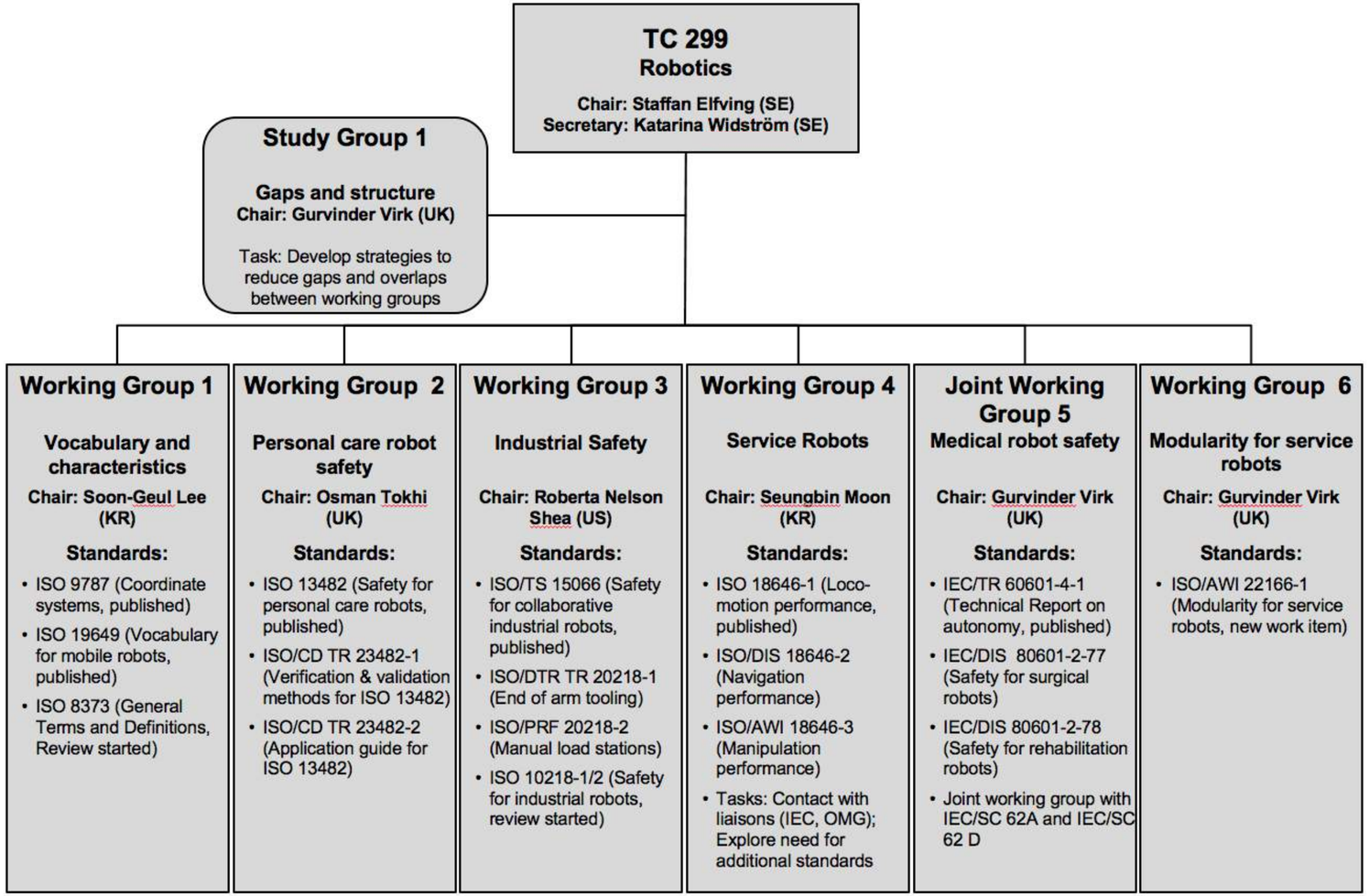




Institute for Aerospace Studies
UNIVERSITY OF TORONTO



Towards Robustness, Reliability, and Safety | Standardization

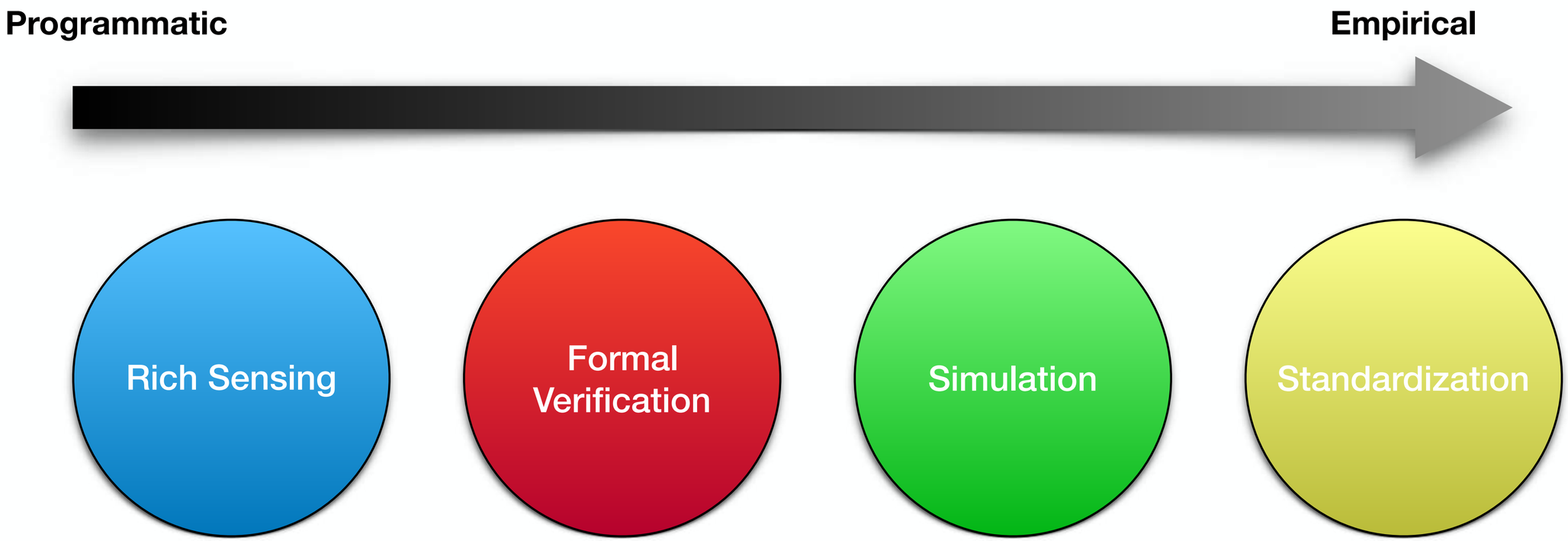


Towards Robustness, Reliability, and Safety | Human Understanding

- To some extent ‘peaceful coexistence’ depends on our own understanding of the capabilities and limitations of robotic systems
- There is a need to educate the general public about robotics
  - Machines will become both smarter and safer over time, but that doesn’t mean we should ignore training
- The field of Human-Robot Interaction has explored, and continues to actively explore, this domain

Image: Hacker Noon

Summary | A Path to Robustness, Reliability, and Safety



Thank You | Questions?

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

http://starslab.ca/

STARS
Space & Terrestrial Autonomous Robotic Systems
L A B O R A T O R Y