Verification of Autonomous Systems Workshop ICRA 2018 - Session 1 Notes

Dejanira - industrial - A*STAR development scientist - challenges in industries of ARTC (aerospace & food) - remanufacturing (components come in, check for defects/problems, manufacturing processes to fix them), manufacturing, industry 4.0. In Singapore - aerospace - highly customized, manufacturing - plane components - inspection and repair - lot of part-to-part variation - dismantle, inspect, check - don't what you'll find - compared to traditional manufacturing - extremely variable - many processes - designs are human-centric - can't be reached by robots - tough to automate - industry wants to (safer, business case) - well-defined standards - processes very very hard to change (2 years to change type of bolts to something automatable). Food industry - many processes still manual - demands of store/chain of stores - lines are flexible and change a lot - conveyer belts - goods have lots of different forms - materials, shapes, forms - try to use same robots, but jobs very diverse in lots of different places - don't know what robot will be doing tomorrow. Also highly customized.

Challenges - human safety - traditional industriai robots - use generic robot arm rather than customize industrial robot - human needed to reprogram, some processes can be done collaboratively - human handles perception and sensing, robot performs actions - close collaboration process - cobots - not massive industrial caged robots, have slower robot arms working with humans. - lighter payload, but sensors in the joints so they stop automatically to avoid collisions along length of arm. Compliance wrt safety standards - hits then stops, but can hurt human partners. Trend/research/effort to add more sensors and intelligence (prevent injury, speed interfaces, simplified reprogramming). Before, had fixed cell - robot does X, human motions are also routine and predictable, calculate max speed that will ensure robot only bruises human, not break. Smarter programming, actions, more flexibility in the lines - open challenge - taking things that are very constrained and rigid and becoming more open and flexible. Learning/changing - how to prove operator's safety = sensors & systems? Is there a methodology - if I follow, can I be certified as safe? When reprogrammed, make sure doesn't delete or overwrite safety protocols - Not introduce harm to environment (e.g. pick up something sharp in an awkward way.

Mobile manipulators - robot arm on a moving platform - no safety standards for these kinds of things - how to V&V a manipulate a manipulator/AGV hybrid system? Usual problems plus reliability of wifi - lots of metal and concrete

Industry 4.0 - data, automated control of everything, digital twins(?) machine controls everything remotely How to protect data? Do we really need all of it? Ensure control working correctly? Prove simulated arm is correct representation of real? Prevent unpredicted actions. Truthful model? Security against attacks and intrusions? Can people hack in and change the robot's actions? What about safety when you mix all the pieces?

Intellectual property - robots, conveyers, automation equipment, software from different vendors, s/w supposed to connect it all together - even if can see and control software, everything starts out as a black box - how to verify interactions between them? Had projects where we programmed kuka robots, file somewhere gets overwritten, program lost - no way to know how it's happening. Ensure it doesn't have gridlocks at higher level, but no visibility and assurances at lower level. Really hard to buy things, connect them, model and verify them.

Once solution proposed: ROS-i - need industry group to open more. Really hard! Concurrency and serialization, protect IP and be open at the same time, ensure open source libraries protect the integrity of the equipment? Many libraries out there, but no culture of testing and offering assurances for what you develop - people (esp in academia) rarely test thoroughly! industry - not going to use that!

Craig - NIST - bridge gap between research and industry needs - what we've heard from industry - different perspective - manufacturing - discrete industrial processes. work comes to the robot - trying to get robots more sensing and knowledge-enabled, minimal up-front programming, intelligent navigation around factory, keep humans safe - need level of confidence that system will perform as expected. Add adaptation, agility, much harder to verify. Why we're here! What does industry want? Company created big things - factories set aside to create individual things. High demand for part B and low demand for part A - use part of factory A to create part B - tried to figure out if they could - couldn't - couldn't switch quickly and verify - gave up. Kind of challenges trying to solve for industry. What we do at NIST - lots of videos. Test methods to verify that systems perform the way they're supposed to - agility and maneuverability and manipulability - see grip strength (top right video), different robots work together even thought fundamentally different. Put together test methods, characterize how well they perform - but anecdotal. Challenge of AI. At NIST, AI is a big deal, lot of work going into it. Learning -> almost impossible to verify. AI is the future, going to move forward - need to be able to verify - if we can figure it out, in good shape. Reason over ontologies - easier to verify if you have a reasoning chain - understand why it did the thing it did - starting to be used in banking industry. Introspection important, but how to get there?

ARIAC competition - teams to apply AI to robotic challenges appropriate for industry - teamed with OSRF, develop control systems, try to solve challenges - what happens when robot drops part, when need to reprogram quickly, when change to order comes in. Next week, $10k winner will be announced. Underlying goal - test our testbeds. www.nist.gov/ariac

NISTpost-doc program - us citizen - professor or pdh student, want to work on this, cycle closes in august, flyers outside, check out the website

Pulkit - MathWorks - model-based design - managing complexity, model-based design, verify what the system will not do, especially for adaptive systems. autonomous systems with matlab - navigation and path planning, learning, manipulators, mapping - all can be safety critical - making important decisions - safety - critical functionality, real-time operation, predictable behavior, robust - don't want to shut down or reboot during critical operation - no emergent behavior, be able to trace cause of failure. design -

most of the errors here, only found out later in testing and integration phase. Results from DARPA grand challenge and robotics challenge. Learning systems making bad decisions (sometimes spectacularly bad). Use model-based design - model algorithms, environments, explore design alternatives, run simulations on wide range of options. Want lots of models, doing lot of simulation - want to be able to generate code, test and do static analysis on it. Certifiable model-based design workflow Documented in MathWorks IEC Certification Kit - component or system testing (brown arrows) - starts from requirements and goes to object code. Advantages - automate manual tasks, deliver high quality s/w, utilize workflows approved by safety and testing authorities - produce artifacts and metrics to provide to certification for assurance case.

Recommendations - must ehibit certain functional and safety properties to ensure acceptance - must be part of system requirements - system level validation needs to be done, include test and code analysis. Verify BOTH what the system will do and what it will not do. Fans of formal methods - should be used, learned state too broad to provide adequate assurance - express mathematical notation helpful. Used in a number of cases - aerospace -> oil and gas, automatice. Example 33DoF robot - hand coding - a mess when trying to develop overall control loop - model actuators and auto-generating code - maintain and verify - deployed.

Hadrien - IIS - AUV in collaboration with DST - domain - ISR - support defense and law enforcement agencies - autonomously survey large area - monitoring, search and rescue. Industrial scale AUVs available - Hugin, Bluefin, Kongsberg gliders, waveRider (?) - consequences of error unacceptably severe - trust is a major challenge! V&V essential to progress of AUV trustworthiness and acceptance. Anticipation of regulatory legal framework. Traditional - composition of task-specific components, but complexity is an issue - requires highly skilled and expensive workforce - need automated extraction methods - want to evaluate performances in realistic uncertain/unstructured environments. sTatistical techniques - lack of methods and tools to V&V them. V&V of time-varying systems, traceability, quantifying resistance adversarial examples - manipulate environment to intentionally provoke mission/AUV failure - need to deal with components from both trusted and untrusted components. Build trusted system from untrustworthy components? Monitor and manage untrusted components? Working on it! Griffith University.

Panel Discussion:
Raja: Black boxes - what knowledge you have about the black boxes - don't know contents, but do you have a description of constraints or ... - how did you get the boxes? When you program the robots - scripts in python - don't want to install anything on computer on robot - scripting language - get pose, status - get a lot of data on current state of robot, but no visibility on code inside robot - hard to model. Latency can cause problems. Hardware simpler. All devices are black boxes, but have specs. Don't tell you how, just what. What's different? Don't get a specification for behaviors. Signe - specification - Pulkit - known unknowns are good - unknown unknowns are bad - black box okay if known unknowns - black box - no spec, unpredictable, can measure some things, can have some guarantees, but might randomly fail. Specifications - should be

quantifiable and measurable - did I meet this, what is the threshold or range within which acceptable - should be testable - how are you going to testable, include test protocol in spec - other things - Dejanira - agree - test chips - model them in s/w before printed - execute s/w to check chip function - can't check function if don't know what it's _supposed_ to do.

Use of verification in traditional computer systems - autonomous systems - different mindset - not explainability - do we move from V&V to qualification? how we've qualified/tested, when we're going to do it again?Opinion? Treat them as s/w and not more like humans is going after the wrong problem. Thoughts? Craig - very scary - system that you trust to fly you somewhere - it's qualified, ran it through tests, handled everything we though of. I agree - don't know what other options we have. Good - may be as good as we can get. Q: Already have that - driving - safety system is lines of paint on the road. Pulkit - interesting idea - doesn't have to be either/or - should be both. Licensing vs. certification. Need to at least explore as a community. What are the things we don't want it to do. Craig - bad things can happen to humans if they violate norms. Hadrien - pilot - also have co-pilot checking pilot's work and determinisitc controls preventing some problems.

Q: System will not do: autonomous car - "will not run people over" -> will stop in certain situation. deterministic bounds like braking distance. constrain the problem from both ends. Behavior one is more open-ended - need both - how do you encode behavioral into more formal approach. Does the right thing, never does the wrong thing - formal requirements language not expressive enough - do this here, but not always.

Q: accept these things. Also accept ramifications - medical malpractice, X deaths /year that we accept. Holding autonomous systems to a higher standard. What's the notion of acceptable failure? If cars didn't exist, someone proposed it and said will kill X people per year - would we accept it? Going to shut down program if autonomous cars kill anyone - not perfect, but dramatically better - not implies that killing X differential is okay. Won't get to perfect systems, need to transition earlier. Very difficult - concrete example - example - car in israel ran a red light, no one got hurt. Uber car has a fatality - nighttime, snow - higher level of safety lower level of risk vs. reasonable - must manage different things - safety system becomes more complex. Lower standard might not be acceptable. Craig - process of getting used to any new technology. Probably wouldn't have cars - but get accustomed. We're not going to set the standards, general public, see more good than bad - tolerance will go up. Dejanira - tesla - shiny surface causes problems with vision. Train for driving - don't train for just in case tree falls on car - as good as human, but very hard - HRI - distrust because of high expectation, other people don't. Understand more about trust.

Don - troubled by apparent double standard - acceptable level of risk/safety/loss for human/non-autonomous systems - what determines what is acceptable for a human-operated system? At what point do you say "unsafe" or take it off the market? US: consumer products safety commission does this, but don't know what the metrics are. Eventually comparable ones for autonomous systems.

Raja - aviation systems - airplanes safest way to travel because industry went to the trouble of quantifying rate of catastrophic accidents (10^-8?) - chance of failure very low. Don't understand why not have similar processes for autonomous systems. Same constraints. Need to find rules to qualify failure modes of those systems. Techniques exist for some - need to qualify safety bounds - acceptable operation. 10^-8 - design failure, not historical data. Aviation - didn't solve the problem - airplanes are largely autonomous, but they enforce obligation for human-on-the-loop - still have trust issues with autonomy - rely on trust given to human. Technical specification - plane to be trustworthy - manufacturer must provide data that they comply with standards and certification process - make you feel comfortable. Trustworthiness of the programs and hardware components. Dependency and resilience - autonomy adds complexity - automated pilot is qualified, processes for s/w and h/w so it can fly and is trustworthy. techniques should be applied to autonomous systems. not easy - environmental aspects. Obligation to provide safety cage. Hadrien - certification - human still relied on to handle uncertainty and crises. Raja - point is that problem is low probability. Hadrien - human is port of last recourse if there's a problem. Craig - airlines have backups on backups - maybe that's the answer. Deterministic bounds - what if one fails? Chaining?

CASE - autonomous system crash - what failed? autonomy, sensor? False positive ignored by systems.

Investment - airlines - lots of resources AND regulartory agency - what would help as much as V&V, if what we need for autonomous systems - similar federal regulator for each nation for autonomous systems - same investigative powers like NTSB - system wasn't hooked up properly - evaluate problem - everyone learns from each individual problem. Need to look at regulatory esbalishment - help get to low probability failure. AU - can report drone failures to regulatory - additional work, is it worth reporting - only applies to drones. Underground robots - experience failures not their fault - investigators said nobody involved, mandates are about human safety - no human involved, they just leave.

Explanation and visibility - very important -

Industry - not only typical robotics business has less resources - users allowed to add application code - no longer amenable to manufacturer verification. Pulkit - push back - resources an issue - apples to oranges/bigger oranges - onus on industry to do more test cases, open source software, ROS is not certifiable right now - no test cases - closed source ROS and never release it back to the community - tools available to support effort, but also up to industry to invest in those areas - industrial automation, robotics - certain industries with more industries will hopefully show us the way. Q: not all code needs to be certified. Lot of misconceptions - depends on applications. Medical.

Closing statements: Craig - qualification approach way to go, but interesting and no answers. Pulkit: good discussion. Need to evolve - more deterministic bounds, share knowledge and experience. Good place to start. Dejanira - explainability,visibility, transparency. Hadrien - trustworthiness. International collaboration - don't have to wait for the cool stuff.