What Happened at the DARPA Robotics Challenge, and Why?

C. G. Atkeson*, B. P. W. Babu†, N. Banerjee†, D. Berenson†, C. P. Bove†, X. Cui†, M. DeDonato†,
R. Du†, S. Feng*, P. Franklin†,M. Gennert†, J. P. Graff†, P. He†, A. Jaeger†, J. Kim, K. Knoedler,
L. Li†, C. Liu*, X. Long†, T. Padir†, F. Polido*†, G. G. Tighe†, X. Xinjilefu*

Abstract

This paper summarizes self-reports made by many of the DARPA Robotics Challenge teams on how much autonomy was used and why errors occurred. We also include observations about commonalities and differences in robot behavior across teams, and discuss lessons learned.

1 Introduction

This paper describes some of the lessons learned from the DARPA Robotics Challenge (DRC). We requested teams to voluntarily provide self-reports of what caused various events at the DRC, in an attempt to get a deeper understanding of what happened (DRC-Teams, 2015). We also provide our own observations based on reviewing the videos of the DRC.

There are several motivations for this paper. The first motivation is that we wanted data to test hypotheses about what led to success or problems in the DRC, including falls, failed attempts at tasks, long periods of robot inactivity, and operator errors.

The second motivation is that the dominant messages from the DRC were, unfortunately, superficial: *Robots can't do much, robots are slow, robots fall down.* It is ironic that the top failure compendiums were put out by DARPA (140,000 views in the first month) (DARPA, 2015), and IEEE Spectrum (1,400,000 views in first month) (IEEESpectrum, 2015). These videos represent a lost opportunity to mix more positive and substantive messages with entertaining blooper reels for a more balanced perception of what happened at the DRC. The team self-reports attempt to add more depth to the DRC legacy, by trying to explain why events occurred.

The third motivation is an attempt to resolve a dispute between the two CMU DRC teams, Tartan Rescue (CHIMP) and WPI-CMU (using a Boston Dynamics Atlas biped named WARNER). The robot that was supposed to be consistent and "not have to worry about falling down" (CHIMP) actually fell down and had two very different runs over the two days, while the biped did not fall down and performed essentially the same way on its runs on two different days. One explanation for these unexpected results was that an explicit goal of the CHIMP operators was to go as fast as possible to get a good time, while an explicit goal

**Robotics Institute, Carnegie Mellon University
††Robotics Engineering, Worcester Polytechnic Institute
[0]This paper is an extended version of a published conference paper (Atkeson et al., 2015). Atkeson is the corresponding author.

of the WPI-CMU team was not to rush and go only for successful execution (points) but not time. In fact, WPI-CMU's first run was only about 30 seconds under the one hour time limit. The claim was made that CHIMP would not have fallen and would have been more consistent if CHIMP had gone "as slow as the WPI-CMU team's robot." We attempted to explore this hypothesis, and discovered that operator errors (potentially due to time pressure) did explain a large fraction of the errors across the teams. However, we did not find any evidence that robots made errors or fell due to fast movement or dynamic effects during the DRC.

The fourth motivation is the contention in the Japanese press that a reason for Japan's worse than expected performance in the DRC was a lack of autonomy in the Japanese DRC robots (DRC-Teams, 2015). The desire to test this hypothesis led us to ask about how much autonomy was actually used during the DRC.

2 What Happened at the DRC?

The statements in this section are based on the submitted team self-reports and our observations of the DRC both at the event and from the resulting videos.

2.1 Human-Robot Interaction (HRI)

Did the time pressure or "speeding" lead to performance issues? We believe that no robot moved fast enough to have dynamic issues sufficient to cause it to lose control authority. However, it is clear from the team self-reports that trying to do the tasks quickly caused problems for the human operators. The top six teams all had major operator errors leading to a fall, a reset or large delay, or a task failure (DRC-Teams, 2015).

Operators want to control the robot at many levels. Across teams, robot operators wanted to be able to control the robot at many levels, and rapidly and conveniently switch between them: 1) command joint velocities or changes in positions, 2) command Cartesian velocities or changes in positions, 3) designate end effector targets such as desired grasps or footsteps, 4) provide task parameters such as speed, and 5) select tasks or task components to perform (DRC-Teams, 2015).

2.2 Behavior Design

Behaviors were fragile. KAIST reported that a slightly longer drill bit at the DRC compared to what it practiced with caused problems. CHIMP had issues with friction variations. WPI-CMU had several parameters that had been very stable and well tested in our DRC test setup replica, but that had to be changed slightly in the DRC (perhaps due to running on a battery for the first time). TRACLabs had problems with the BDI behaviors at the DRC. AIST-NEDO had a 4cm ground level perception error, and fell coming off the terrain. In the DARPA Learning Locomotion project, we and other teams had this problem frequently. Behaviors that worked perfectly and robustly in our labs did not work or were erratic when tested on an "identical" setup at a DARPA test site. To be useful, robots need to handle small variations of a task.

Full body behavior: Why didn't any robot use railings, walls, door frames, or obstacles for support and stabilization? It is startling to realize that all teams failed to use the stair railings, put a hand on the wall to help cross the rough terrain, or grab the door frame to more safely get through the door in the DRC Finals. Egress was the only example of the use of "contact-rich" behaviors (KAIST, CHIMP, RoboSimian, WPI-CMU). WPI-CMU also used a contact-rich strategy in ladder climbing in the DRC Trials (DeDonato et al., 2015). Maximizing contact led to using the car to stabilize the robot as much as possible. It is interesting that other Atlas teams tried to minimize contact with the car, spending a lot of time carefully balancing on one foot with no other supporting or stabilizing contacts. MIT released a video of their robot standing on one foot while the vehicle is being shaken (MIT, 2015b). These are impressive but

perhaps unnecessary demonstrations of high quality robot control. Both IHMC and MIT added a handle to the car within easy reach of the robot to hold while driving, but their robots did not use it to get out of the car. IHMC chose not to use the handrail on the stairs because during development the Atlas robots had very poor arm force control due to faulty wrist force/torque sensors, and the Atlas arms have a lot of static friction making implementing arm compliance difficult (IHMC, 2015). We had similar problems with our Atlas robot, and also discuss later the algorithmic challenges of using contacts. There is a risk in using contact-rich strategies. CHIMP did get caught on the car while trying to egress (DRC-Teams, 2015).

Behaviors need to be robust to robot hardware component failure. Hardware reliability is a limiting factor for mobile manipulators. Mobile manipulators including humanoids consist of many components, any of which can fail. As a result, for example, our ATLAS robot had a mean time between failures of hours or, at most, days. Since we could not repair the robot ourselves, we often ran the robot with various components not working. We had to develop behaviors that tolerated or worked around hardware failure. For example, it was rare that both electrical forearms in the final Atlas configuration were both working at the same time. Two handed drill task strategies turned out to be a big mistake, even though they were more robust than one handed strategies *when both arms were working.* We should have done a better job developing behaviors that worked when unreliable components actually turned out to be not working, rather than engage in wishful thinking that the problems would get fixed or go away before the DRC. The Atlas robots also needed to be able to do any one-handed task with either the left or the right hand.

Gentle touchdowns make walking easier or harder? Our footsteps seemed much softer with less of a shock wave travelling up the robot body than other Atlas teams. Does a firm footstep (a stomp) make sensing and control of locomotion easier or harder?

Is compliance useful? If so, how much, when, and where? We note that when Atlas robots fell, they often seemed very stiff with legs sticking up in the air and joints not moving during the fall (all of the videos of the other Atlas teams seem to have this behavior, similar to videos of the Honda Asimo falling). Perhaps very little compliance was actually being used in these so-called force-controlled robots.

2.3 Robot design

Leg/wheel hybrids were very successful when flat floors and sparse light debris are present. CHIMP, KAIST, UNLV, RoboSimian, and NimbRo are examples of this. It is an open question what combinations of wheels, tracks, and limbs will be useful in the spectrum of rubble typically found in a collapsed building or an explosion or earthquake site.

Some leg/wheel hybrids and quadrupeds did not do any rough terrain. NimbRo, Aero, and RoboSimian did not do the rough terrain task or the stairs task at the DRC. It was possible to do the debris task instead of the terrain task and robots with wheels chose to do the debris task (including the Hubo teams). CHIMP also plowed through the debris, and did not do (or need to do) the rough terrain task. CHIMP did do the stairs. This pattern of not doing rough terrain in the form of tilted cinder blocks or stairs is surprising, in that tracked and quadruped designs were originally advocated as better ways to deal with rough terrain than bipedal designs. All of the biped robots that got that far chose the rough terrain task instead of the debris task. We found that bipedal shuffling through the debris was unreliable due to jamming, and picking up the debris was too slow.

Wheeled vehicles need to balance and get un-stuck. To our surprise, many wheeled and tracked vehicles fell (CHIMP, Aero) or almost fell (KAIST, NimbRo). It is clear from the DRC that many of these robots were operating in the dynamic regime, and all needed to take dynamics into account to avoid roll-over, avoid getting stuck, or recover from being stuck.

Small mechanical details of the robot made a big difference in the DRC tasks. This was especially clear in the DRC Trials, where robots that could walk with knees pointing backwards did very well on the

ladder task (Schaft, Hubo), while robots that could not had a lot of trouble (Atlas). The Atlas robots hit ankle joint limits stepping down on the cinder block (rough) terrain, which caused problems. In addition to wheels or tracks, mechanical details such as the ratio of the size of the feet to the center of mass height could make a difference in how hard the DRC was. A practical limit to the foot size was the size of a cinder block. No robot went that far in taking advantage of the details of the DRC course. We believe the Atlas robots had small feet relative to other bipeds, but we do not have foot measurements and center of mass heights for the other robots.

Heat dissipation, and planning for thermal management are important. One failure mode was that Atlas forearm motors overheated and were automatically shut down. We now believe we could have avoided these failures by paying attention to and planning about thermal behavior in addition to mechanical behavior. NimbRo also had motor overheating issues in their leg/wheel hybrid robot. The winning DRC Finals (KAIST) and Trials (Schaft) teams put a lot of emphasis on heat dissipation and thermal management in their robot and control system designs.

2.4 Software engineering

Have we figured out how to eliminate programming errors? No. IHMC, MIT, and WPI-CMU all had bugs that caused falls or performance issues that were not detected in extensive testing in simulation and on the actual robots doing the DRC tasks in replica DRC test facilities (DRC-Teams, 2015). IHMC in particular tried to have excellent software engineering practices, and still had an undetected bug that helped cause a fall on the stairs on day 1 of the DRC (IHMC, 2015).

2.5 Control

Switching behaviors: Changing dimensionality is difficult for current control approaches. Our quadratic programming-based full-body control typically introduces large command discontinuities when the dimensionality of the problem changes due to contact changes, stick/slip change on any contact, singularities, and hitting or moving off of joint stops or limits. A typical step involves heel strike, foot flat contact, and toe push off. Walking on stairs may involve walking on the toes as well as on the sole of the foot. A large step down may involve jumping (no foot contacts), toe strike and foot flat, and the knee may go from straight to quite bent. Grasping a railing, touching or bracing against a wall all involve structural change in the robot kinematics and dynamics. Prioritized or constraint hierarchy approaches have similar problems with structural change of kinematics or dynamics.

Blend, don't switch. We developed an approach to structural change smoothing in our quadratic programming. In a previous implementation, the inverse dynamics QP changed dimension based on the number of contacts. Discrete changes can also happen due to constraint manifold changes such as during toe-off, when the foot frame CoP is constrained at the toe. Such changes will cause sudden jumps in outputs that can induce undesired oscillations on the physical robot. These jumps are caused by structural changes in the problem specification, and cannot be handled properly by just adding cost terms that penalize changes. Our solution is to maintain the same QP dimension and gradually blend the constraints over a short period of time. We always assume the largest number of contacts, but heavily regularize the magnitude of the contact force and relax the acceleration constraints for the inactive contacts. Inequality constraint surfaces are changed smoothly when a change is required.

How to use foot force/torque sensors? Of the top three Atlas teams, our understanding is that IHMC and MIT both used the foot force/torque information just as a binary contact sensor (IHMC, 2015; MIT, 2015a). We used foot force/torque sensing in our state estimator and falling prediction, and controlled the robot using full state feedback and joint level force control, which includes joint (including ankle) torque feedback, but does not include direct contact force/torque feedback. Some ZMP-based robots use output feedback in the form of contact force/torque feedback to do foot force control. We need to introduce output

Figure 1: Successful sidestepping through the door (left, middle) and the failure in the DRC rehearsal (right) in which the protective cage (black padding) for the head is against the white door frame.

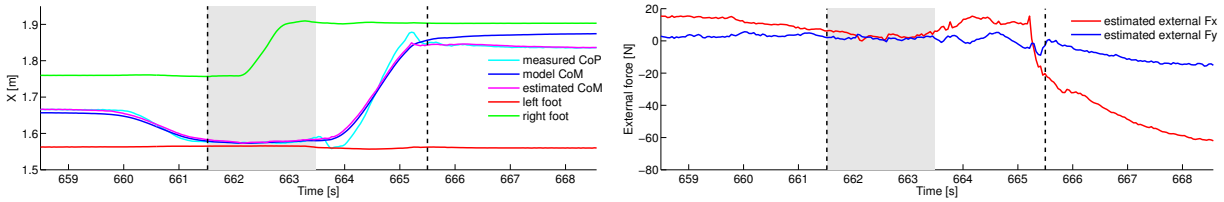

Figure 2: Atlas was caught on the door frame when sidestepping through it during the DRC rehearsal. The walking controller delayed liftoff and remained in double support when the external force estimator detected a large change in the estimated external force in the robot's sideways direction (F_x, through the door). The single support phase is shown by the shaded area, and the black dashed lines indicates the planned liftoff time. The estimated CoM is the sum of the model CoM and the estimated CoM offset.

feedback and contact force/torque control to our QP and state-based full body control and perhaps to higher level control based on simple models.

2.6 Error detection: Early and all the time

Early detection of robot or operator errors can prevent falls. Often falling is caused by errors in robot locomotion or manipulation, where the robot pushes or pulls itself over, rather than outside perturbations such as wind, support movement, or external agents pushing the robot. Our field has over-emphasized research on responses to outside perturbations such as "push recovery" over handling robot-generated errors. We developed an automatic balance error detector based on estimating forces between the robot and the world represented as a "corrected" capture point, which saved us from several falls both in practices and in the DRC (Figure 1, 2, and 3). We found that simply stopping what the robot was doing gave the remote operator enough time to successfully intervene (Atkeson et al., 2015). Other teams provided displays or other types of feedback (such as audio) of various danger indictors (such as IHMC (DRC-Teams, 2015)).

2.7 Planning

Inverse kinematics is still a difficult problem. Tuning optimization-based inverse kinematics algorithms to avoid wild motions of the arm and still achieve task goals proved difficult, and constraint-based inverse kinematics algorithms eliminated too much of the workspace. We need more work in this area to achieve human levels of performance in trading off desired end effector motion in task space and undesired motion of the rest of the arm.

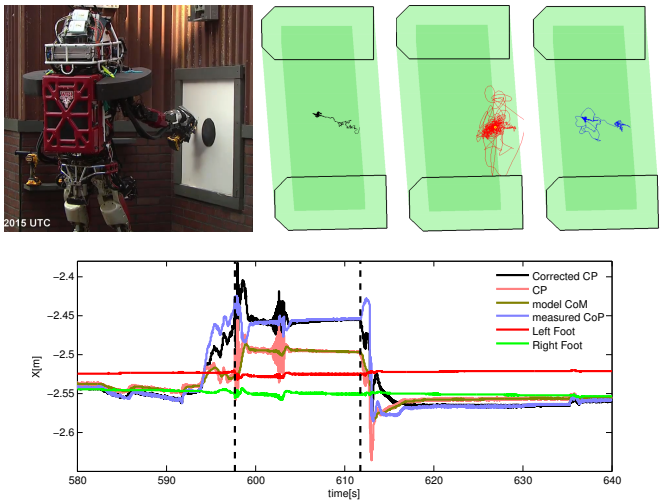

Figure 3: Our robot nearly pushing itself over during the drill task. Top Left: Robot posture after error detection. Top Right: Black trace: The corrected capture point (CCP) up to the error detection, Red trace: The CCP during the "frozen" period, and Blue trace: The CCP moves back to the center of the polygon of support during manual recovery. Bottom: Plots of candidate fall predictors in the fore/aft direction during this event. The black vertical dashed lines mark the freeze time and the start of manual recovery. We can eliminate some candidate fall predictors easily. The center of mass (CoM) (and a "corrected" CoM (not shown)) usually provide a fall warning too late, because the CoM velocity is not included. The capture point (CP) does not include information about external forces. The center of pressure (CoP) is noisy and gives too many false alarms. It can warn of foot tipping, but it is less reliable about warning about robot falling, which is not the same thing as foot tipping in a force controlled robot or if there are non-foot contacts and external forces. In this plot, we see that the CoP moves away from the safe region during recovery, predicting that the robot is falling again, while the corrected capture point (CCP) moves towards the interior of the safe region.

More degrees of freedom makes motion planning much easier ($7 \gg 6$). The original six degree of freedom arms on Atlas led to many problems finding reasonable inverse kinematic solutions. Adding an additional arm degree of freedom and including the whole body in manipulation planning greatly reduced the challenges of inverse kinematics. Similarly, a neck that allowed head turning (rotation about a vertical axis) would have been vary useful. The Atlas robot only allowed head nodding (rotation around a horizontal axis in the frontal plane (a pitch axis)).

Approximate self-collision detection was adequate. We used crude capsule-based approximations of the robot link shapes to avoid self-collisions.

3 Discussion

Humans rescuing robots is not acceptable. We believe that robots are useless for disaster recovery if they require humans to rescue the robots, rather than the other way around. We were surprised at how many robots required physical human intervention during the DRC. Many team's robots fell or got stuck due to operator errors, robot errors, and hardware failures. If this had been an actual disaster in which physical human intervention was risky, costly, inefficient, or not possible (such as recovering from a nuclear or toxic chemical disaster, a "dirty" bomb attack, or a dangerous epidemic such as the recent Ebola epidemic), of the teams that attempted all tasks the only teams that would have proven useful or perhaps even survived the two missions over the two days of the DRC would be the CMU CHIMP team and WPI-CMU. However, none of the robots could currently deal with the complexity and difficulty of an actual disaster site such as

the Fukushima Daiichi disaster site.

We expected bipedal robots to have trouble in the DRC. We used an Atlas humanoid robot built by Boston Dynamics. Compared to wheeled and quadruped robots, bipedal humanoids usually have higher centers of mass and a smaller support region, which results in a smaller stability margin and a higher risk of hardware damage when falling. In a real-world scenario where humanoid robots have to interact with the environment without a safety system such as a belay, fall prevention and recovery behaviors are necessary. Avoiding getting stuck and getting un-stuck are also important. The DRC attempted to mimic a real-world scenario, in which robots had to perform disaster response related tasks involving physical contact with the environment. What we observed during the contest was that most humanoids fell down and no biped that fell got back up. We were also surprised that wheeled robots had so much trouble with balance, falling, and getting stuck in the DRC.

The most cost effective research area to improve robot performance is Human-Robot Interaction (HRI). After entering the contest believing that robots were the issue, we now believe it is the human operators that are the real issue. They are the source of the good performance of the DRC robots, but they are also the source of many of the failures as well. The biggest enemy of robot stability and performance in the DRC was operator errors. Developing ways to avoid and survive operator errors is crucial for real-world robotics. Human operators make mistakes under pressure, especially without extensive training and practice in realistic conditions. Interfaces need to help eliminate errors, reduce the effect of errors, and speed the recovery or implement "undo" operations when errors happen. Interfaces need to be idiot proof, require no typing, have no check boxes, and minimize the information displayed and the options the operator has to decide between. The best interface is perhaps a large green **go** button and a much larger red **STOP** button.

What are we going to do about software bugs? It is frustrating that several teams did not detect more of their software bugs after attempting to use good software engineering practices and extensive testing in replicas of the DRC environments. We do believe more time would have helped Team WPI-CMU do a better job testing on our DRC replica setup and the actual robot, and this is probably true for other teams as well. We could have done better in terms of planning and software process. Another big factor in our software development was the late change to electric forearms for our Atlas robot.

However, we do not believe that more tests in simulation or regression testing would have made a substantial difference. We do not believe that formal methods and verification techniques would have helped much either, as much of robotics involves the analog and messy real world, and the assumptions and limitations necessary to prove something meaningful about a piece of software typically do not fully address what can go wrong in the real world. Here is a list of some of the modeling errors that have happened for us for rigid body kinematic and dynamic models of simple robots with pin joints and rigid links: wrong kinematic and inertial parameters, cogging and other magnetic effects in electric motors, actuator dynamics and hysteresis, variable viscous or nonlinear dynamic friction depending on the state of hydraulic oil leakage, dust, and dirt, thermal (both external weather and actuator heating) effects, humidity effects, constant delay, variable delay due to computer communications. joint/actuator/transmission stiction and other static friction effects, foot slip-stick on touchdown and during stance, six dimensional bearing play, structural link deformation, and material aging. Such models can be difficult to improve if velocity, acceleration, and joint force and torque measurements are not available, inaccurate, or noisy.

Is the route to reliable robotics orders of magnitude more testing in diverse environments in the real world? This is how we verify automobiles, and they still have accidents and recalls. This process is, of course, hugely expensive.

How much autonomy was used in the DARPA Robotics Challenge? In our view there was a surprising amount of commonality of approach and use of autonomy across teams, particularly the top Atlas teams. Operators designated targets by indicating regions of interest in images and point clouds (in various ways). Locomotion was planned and controlled autonomously, with the opportunity for human operators to correct errors. Robots with legs typically provided ways for the operator to adjust footsteps. Balance was

handled autonomously in a similar way across the Atlas robots. No robot was either fully teleoperated or fully autonomous (even at the task level).

It is clear that some commentators had expectations of greater autonomy (Sofge, 2015; Dzieza, 2015; **?**; Gaudin, 2015). The highest level of autonomy that could be expected was performing tasks based on high level verbal descriptions one might give to a person: "Turn valve 204 counterclockwise 360^{deg}". The robot might have a map indicating where valve 204 was in the building, a plan library that indicated how to turn many types of valves, and a method to find and adapt the most relevant plan in the library. No robot operated at this level in the DRC.

Instead, plans and planners were specialized for the particular tasks. The location and sizes of objects such as valves were parameterized. During a run, most teams used remote human operators to locate objects. Many teams then refined the location and identified the object size autonomously from vision and laser scan data. This was typically the only use of the human operator if everything went according to plan. Because much was known about the tasks (door handle characteristics, what type of valve, the vehicle model, and the nature of the surprise tasks), most teams did not program approaches to handle wide variations. The plan libraries were typically trivial, having only one plan for the current task (CHIMP had much richer plan libraries).

Our experience with general tools such as TrajOpt (Schulman et al., 2013) was not good, particularly for egress. Current tools did not do a good job resolving redundant inverse kinematics, choosing robust task strategies, or handling contact-rich strategies. We believe the over-emphasis on finding feasible paths based on geometric information only has limited the applicability of motion planning in the real world. We believe a new generation of motion planning tools are needed that generate robust contact-rich strategies using reasoning about both geometry **and physics.**

Learning to plan better is hard. To be computationally feasible for real-time control, a cascade of smaller optimizations is usually favored, which for us and many others eventually boiled down to reasoning about long term goals using a simplified model and a one-step constrained convex optimization that uses the full kinematics and dynamics. Although effective, this approach suffers from the common problem that plagues hierarchical systems: how to generate a different high-level plan when something goes wrong at a lower level, which is not even modeled or represented in the high-level planner? We think the ideal solution is to include as complete a model as possible in the high-level planner and replan as often as possible, but this is unfortunately currently computationally impractical. A second alternative is to give limited foresight to the low-level controller and guide it towards the original high-level goal in that limited horizon. We have experimented with providing the approximated local value function computed by DDP to the inverse dynamics controller, but we have yet to observe a significant increase in either performance or stability. We think this is due to not previewing far enough into the future. For walking, we had the most issues with kinematic related constraints, such as rear ankle joint limits when taking a step down and knee singularity when it goes straight. Just adding inequality constraints on accelerations was not sufficient in most cases. In the end, we worked around these issues with special purpose heuristics that operate in joint space and incur increasingly higher costs as the system approaches these constraints. Although crude, it is a way of injecting some notion of the future into greedy local optimization. Some high-level changes are also necessary, such as limiting step length and allowing toe-off in various situations. A third direction is to insert a simpler trajectory optimizer that has a shorter horizon, but replans much more rapidly. We are currently exploring this idea by adding a receding horizon control component at the 1kHz time scale rather than at the 1Hz time scale, as is currently done.

Sensing is cheap, so let's use as much as possible. We strongly believe that humanoids should be outfitted with as much sensing as possible. The absence of horizontal force and yaw torque sensing in the Atlas feet limited our ability to avoid foot slip, reduce the risk of falling, and optimize gait using learning. We commissioned Optoforce to build true six axis foot force/torque sensors for the Atlas feet, but the sensors were not ready in time for the DRC. We will explore how much they improve performance in future work. Redundant sensor systems make calibration and detection of hardware failure much easier.

We need full-body robot skin. An important research area to work on is full-body robot skin, both for the skin's mechanical properties, but most importantly for sensing. It is an interesting research question as to how to distinguish between robot errors and external disturbances, or combinations of the two, especially with no full body skin tactile sensing. The correct response to robot errors and external disturbances are often quite different and conflicting.

Super-human sensing is useful. Super-human sensing (whole body vision systems, for example) is a useful research area which could greatly improve humanoid robustness and performance. We used vision systems on the wrists to guide manipulation and on the knees to guide locomotion. Time prevented us from implementing planned vision systems located on the feet. We plan to build super-human feet with cameras viewing in all directions and IMUs (accelerometers and gyros). Our goal is to measure foot translational acceleration (accelerometers), linear and angular velocity (optical flow and gyros), and track foot translation, orientation, and relative positions (IMU orientation tracking and image matching). Longer term, we intend to build robust high resolution tactile sensing for the soles of the feet, as well as similar robust high resolution sensing skin. We intend to build many types of optical sensing into the robot's skin, to do obstacle and collision detection at a variety of distances and resolutions (Lumelsky, 2015).

Walk with your entire body. Humans use stair railings, and brace themselves by reaching out to a nearby wall when walking over difficult rough terrain. Except for egress, no robots in the DRC Finals used the stair railings or any form of bracing. Even drunk people are smart enough to use nearby supports. Full body locomotion (handholds, bracing, leaning against a wall or obstacles) should be easier than our current high performance minimum contact locomotion approaches. There is a reason that adults re-learning walking are told: "Walk with your entire body." Why didn't our robots do this? In programming robots we avoid contacts and the resultant structural changes in our models and in reality. More contacts make tasks mechanically easier, but algorithmically more complicated for planning, and the transitions are difficult to both plan and control. Human's "fall" or step down into new contact situations easily. We have seen very few robot planners that are capable of generating this behavior (Mordatch et al., 2012). In the DARPA Learning Locomotion program, it was widely recognized that the "belly-slide" was the best way for a dog-like robot to exit a difficult terrain. This behavior had to be programmed manually, and manually inserted into plans (this was true of several research groups).

Dynamic approaches are often easier than static/quasi-static approaches. Although dynamic behavior is often more impressive to view, dynamic approaches are often easier than static/quasi-static approaches for the robot to actually do. However, perception, planning, and control must keep up with the higher movement velocities. It is easier to ride a bicycle at a higher velocity (within reason) than at a very slow velocity. In situations where footholds are plentiful (such as flat floors) and obstacles are sparse, dynamic locomotion approaches often provide more control authority and are more robust to modeling errors, due to the higher rate of foot placements (cadence). Static and quasi-static locomotion is limited by the ability to control center of pressure location under the stance foot in single support. Faster gaits don't need to carefully control center of pressure location under a foot, but simply place the foot appropriately. An example of this is that point feet (such as stilts) can often be used for dynamic gaits, but not for biped static gaits. We had a lot of trouble stepping up and down. In this case dynamic approaches can use momentum, and rely on the rear leg less, escaping kinematic constraints on the rear leg. Stepping down would involve falling to the next foothold instead of having full control at all times. Jumping is an extreme case of giving up control in the flight phase.

Design robots to survive failure and recover. Robustness to falls (avoiding damage) and fall recovery (getting back up) need to be designed in from the start, not retro-fitted to a completed robot design. The Atlas robot was too top heavy and its arms were too weak (similar to a Tyrannosaurus Rex) for it to reliably get up from a fall, especially in complex patterns of support and obstacles such as lying on the stairs, rough terrain, debris, or in the car or door frame. We believe lighter and structurally flexible robots with substantial soft tissue (similar to humans) are a better humanoid design. We have been exploring inflatable robot designs as one approach to fall-tolerant robots (Atkeson, 2015).

How do we get better perception and autonomy? A more subtle issue is that many DRC teams were dominated by people who work on robot hardware and control, and were weaker in perception, reasoning, and autonomy. Our impression is that many teams used standard libraries such as OpenCV, OpenSLAM. the Point Cloud Library, and other libraries, some of which are available through ROS. We also used LIBVISO2, a library for visual odometry. In some cases one can get excellent performance from general software libraries, but usually software libraries make it easy to get mediocre performance. In the DRC, this problem was solved by over-relying on the human operator and the always-on 9600 baud link. We need to figure out ways to get the perception and autonomy research community interested in helping us make robots that are more aware and autonomous, and going beyond standard libraries. It takes a real commitment to making perception and autonomy software work on a real robot, and there need to be rewards for doing so. In academia, this type of work, going the "last mile", is not respected or rewarded. We note that any student with perception and autonomy expertise is being snapped up by companies interested in automated driving, so it will be a while before the rest of robotics moves forward in this area.

Are we ready for the real world? After 3 DARPA Challenges involving self-driving cars, we are now seeing autonomous cars in the real world. After 1 DARPA Robotics Challenge, it is clear there is much work to be done before robots can usefully work in the real world or help in disasters. Although some have a pessimistic view, we believe that we would see dramatic improvements in robot performance if follow-on Robotics Challenges were held.

Some bipedal DRC teams took the position that the use of safety belays delayed or stunted robot development, and made operators too cautious, and therefore prohibited the use of safety belays early in development. It may be the case that we need to be more aggressive about testing robots in real world conditions.

Paradigm shift needed. We note that in the DRC rehearsal both IHMC and our team did not practice most tasks, since safety belays were not allowed. We did not have enough faith in our software or robot, and we believed a fall would be fatally damaging to our Atlas robots. This lack of faith in our own robot performance does not bode well for deployment of humanoid robots in the near future, unless there is a paradigm shift in both hardware and software design (Atkeson, 2015).

There is something wrong with robotics. We have collectively produced many videos showing convincing robot performance. However, in a situation where the researchers did not control the test, most robots, even older and well-tested designs, performed poorly and often fell. It is clear that our field needs to put much more emphasis on building reliable systems, relative to simulations and theoretical results. We need to understand why our current hardware and software approaches are so unreliable.

4 Improvements For Next Time

We can do better in terms of control. One of our goals is coming closer to human behavior in terms of speed, robustness, full body locomotion, and versatility. A very simple step recovery behavior based on the corrected capture point has been implemented. We think high cadence dynamic walking, heel strike, and toe push off are essential to fast robust walking. To achieve this, better system identification will further bridge the gap between simulation and hardware and improve the overall performance. Optimizing angular momentum in the high-level controller will also benefit fast walking and will increase the safety margin for balancing.

We can do much better in terms of human-robot interaction. Supervisory control of the DRC robots was only possible by operators who had extensively practiced for months, and even then errors were made. We couldn't touch our robot without two emergency stops and rubber gloves to prevent 480V electric shocks. We could safely poke our robot with a stick. Robots and humans aren't really working together until control of a robot can be learned in minutes and physical interaction isn't life threatening, let alone

easy and fun (Atkeson, 2015).

We should do more to provide early fall predictions. In manipulation, we tracked the corrected capture point. In walking, the corrected capture point moves around quite a bit. Its deviation from its expected motion can be used to predict falling. Vertical foot forces (F_z) too high or not high enough, and other foot forces and torques being large are warning signs of large external forces, Joint torque sensors can be used to attempt to locate a single external force to either a hand (manipulation), a foot (tripping), or another part of the body (collision). Robot skin-based sensing would have been and can be expected to be extremely useful as part of an early warning system. Horizontal foot forces (F_x, F_y), yaw torque (torque around a vertical axis: M_z) going to zero, foot motion due to deflection of the sole or small slips measured using optical sensors, and vibration measured in force/torque sensors or IMUs in the foot are early warning signs of possible slipping. The center of pressure going to a foot edge and foot tipping measured by a foot IMU are early warning signs of individual foot tipping and resultant slipping or possible collapse of support due to ankle torque saturation. Any part of the body such as the foot or hand having a large tracking error is a useful trigger for freezing the robot and operator intervention.

Walking on real rubble involves making footsteps, not just finding footsteps. It turns out you have to plan more in terms of *making footholds* on terrains made up of aggregates (loose soil, gravel, pebble beaches or piles of rounded stones, sand, snow, stream beds, and realistic rubble) rather than *finding footholds.* You use your foot to compact a foothold so it remains solid when you put weight on it. In this case footstep planning needs to choose good places to make footholds, rather than good existing footholds.

Be ready for the worst case. In our Day 2 run we lost the DARPA provided communications between the operators and the robot in a full communication zone (standing before the stairs) for at least six minutes, which counted against our time. The lesson here is that for real robustness, we should have planned for conditions worse than expected. We could have easily programmed behaviors to be initiated autonomously if communications unexpectedly failed. Our understanding is that CHIMP lost communications during several tasks on day 2, including the driving task, when full communications were scheduled. CHIMP crashed the car, when it could have easily autonomously stopped, or autonomously driven the entire driving course.

Provide balanced messages. Blooper reels are popular, but they are also an opportunity to educate. Mix meaningful messages and blooper reels. Explain why failures occurred. Show some success.

5 Conclusions

The DARPA Robotics Challenge was a useful reality test for robotics. It showcased areas where robotics is "solved", and exposed areas where there is a great deal of work to do. We feel it is important to provide more background explaining what happened and why, in an accessible way, to the media and to the public. We hope that by making this information available to the community, we can work together to get a balanced message out.

References

Atkeson, C. G. (2015). Big Hero 6: Let's Build Baymax. `build-baymax.org`.

Atkeson, C. G., Babu, B. P. W., Banerjee, N., Berenson, D., Bove, C. P., Cui, X., DeDonato, M., Du, R., Feng, S., Franklin, P., Gennert, M., Graff, J. P., He, P., Jaeger, A., Kim, J., Knoedler, K., Li, L., Liu, C., Long, X., Padir, T., Polido, F., Tighe, G. G., and Xinjilefu, X. (2015). No falls, no resets: Reliable humanoid behavior in the darpa robotics challenge. In *Humanoid Robots (Humanoids), 15th IEEE-RAS International Conference on.*

DARPA (2015). A celebration of risk (a.k.a., robots take a spill). `https://www.youtube.com/watch?v=7A\_QPGcjrh0`.

DeDonato, M., Dimitrov, V., Du, R., Giovacchini, R., Knoedler, K., Long, X., Polido, F., Gennert, M. A., Padir, T., Feng, S., Moriguchi, H., Whitman, E., Xinjilefu, X., and Atkeson, C. G. (2015). Human-in-the-loop control of a humanoid robot for disaster response: A report from the DARPA Robotics Challenge Trials. *Journal of Field Robotics*, 32(2):275–292.

DRC-Teams (2015). What happened at the DARPA Robotics Challenge? `www.cs.cmu.edu/~cga/drc/events`.

Dzieza, J. (2015). Behind the scenes at the final darpa robotics challenge: The robots at the final darpa challenge have come a long way, but they still need a lot of human help. `http://www.theverge.com/2015/6/12/8768871/darpa-robotics-challenge-2015-winners`.

Gaudin, S. (2015). Separating science fact from science fiction in robotics. `http://www.computerworld.com/article/2939331/robotics/separating-science-fact-from-science-fiction-in-robotics-with-video.html`.

IEEESpectrum (2015). A compilation of robots falling down at the DARPA Robotics Challenge. `https://www.youtube.com/watch?v=g0TaYhjpOfo`.

IHMC (2015). Personal communication.

Lumelsky, V. (2015). Home page. `http://directory.engr.wisc.edu/me/faculty/lumelsky\_vladimir`.

MIT (2015a). Personal communication.

MIT (2015b). Egress robustness tests. `https://www.youtube.com/watch?v=F5CBRmDQXTk`.

Mordatch, I., Todorov, E., and Popovic, Z. (2012). Discovery of complex behaviors through contact-invariant optimization. *ACM Trans. Graph.*, 31(4):43:1–43:8.

Schulman, J., Lee, A., Awwal, I., Bradlow, H., and Abbeel, P. (2013). Finding locally optimal, collision-free trajectories with sequential convex optimization. In *Robotics: Science and Systems (RSS)*, Berlin, Germany.

Sofge, E. (2015). The DARPA Robotics Challenge was a bust. `http://www.popsci.com/darpa-robotics-challenge-was-bust-why-darpa-needs-try-again`.