Developing a Robust Disaster Response Robot: CHIMP and the Robotics Challenge

G. Clark Haynes, David Stager, Anthony Stentz, J. Michael Vande Weghe, Brian Zajac, Herman Herman, Alonzo Kelly, Eric Meyhofer, Dean Anderson, Dane Bennington, Jordan Brindza, David Butterworth, Chris Dellin, Michael George, Jose Gonzalez-Mora, Morgan Jones, Prathamesh Kini, Michel Laverne, Nick Letwin, Eric Perko, Chris Pinkston, David Rice, Justin Scheifflee, Kyle Strabala, Mark Waldbaum, and Randy Warner
National Robotics Engineering Center, Carnegie Mellon University, 10 40th Street, Pittsburgh, Pennsylvania 15201
e-mail: gch@cs.cmu.edu, stager.robotics@gmail.com, tony@cmu.edu, vandeweg@cmu.edu, brian.zajac@gmail.com, herman@rec.ri.cmu.edu, alonzo@cmu.edu, emeyhofer@gmail.com

Received 8 October 2015; accepted 23 August 2016

CHIMP, the CMU Highly Intelligent Mobile Platform, is a humanoid robot capable of executing complex tasks in dangerous, degraded, human-engineered environments, such as those found in disaster response scenarios. CHIMP is uniquely designed for mobile manipulation in challenging environments, as the robot performs manipulation tasks using an upright posture, yet it uses more stable prostrate postures for mobility through difficult terrain. In this paper, we report on the improvements made to CHIMP—both in its mechanical design and its software systems—in preparation for the DARPA Robotics Challenge Finals in June 2015. These include details on CHIMP's novel mechanical design, actuation systems, robust construction, all-terrain mobility, supervised autonomy approach, and unique user interfaces utilized for the challenge. Additionally, we provide an overview of CHIMP's performance, and we detail the various lessons learned over the course of the challenge. CHIMP was one of the winners of the DARPA Robotics Challenge, completing all tasks and finishing in 3rd place out of 23 teams. Notably, CHIMP was the only robot to stand back up after accidentally falling over, a testament to the robustness engineered into the robot and a remote operator's ability to execute complex tasks using a highly capable robot. We present CHIMP as a concrete engineering example of a successful disaster response robot.

1. INTRODUCTION

The Defense Advanced Research Projects Agency (DARPA) has a long history of sponsoring competitions to advance the state of the art in robotics. The DARPA Robotics Challenge is one such competition, designed to accelerate the development of robots capable of responding to natural and manmade disasters, motivated by the 2011 Fukushima Daiichi nuclear disaster in Japan. This paper provides an overview of the performance of the CHIMP robot, developed by the Tartan Rescue team from Carnegie Mellon's National Robotics Engineering Center (NREC), at the culmination of the DARPA Robotics Challenge, the DRC Finals in June of 2016.

After placing 3rd in the DARPA Robotics Challenge (DRC) Trials in December 2013, the Tartan Rescue team set about to close the capability gap for what would be needed for the DRC Finals, a year and a half later. Due to the compressed Phase I schedule (14 months to design and build a robot), the CHIMP robot did not have a full complement of software to attempt all of the tasks at the Trials, and the robot itself was ill-prepared to withstand a fall that could result in an early exit from the competition.

After the DRC Trials, the team focused first on adding the missing capabilities, primarily the mobility tasks such as moving over uneven terrain, climbing stairs, and driving a vehicle. At the same time, the team upgraded the robot itself, inserting a battery for tetherless operation, strengthening joints, and adding fall protection.

Once CHIMP was able to perform all of the tasks, the team turned its attention to increasing the robot's reliability and reducing task execution time. The team realized these improvements through a combination of robot autonomy and remote teleoperation. Computer vision was employed to recognize objects, such as the drill, valve, and door handle, thereby increasing the reliability of a grasp and reducing the time of the operation. Scripted motions were used for well-understood tasks, such as climbing the stairs and vehicle egress, with guarded steps to ensure the operation

Direct all correspondence to: G. Clark Haynes, e-mail: gch@cs.cmu.edu

View this article online at wileyonlinelibrary.com • DOI: 10.1002/rob.21696

was proceeding as planned, thereby eliminating the need for the operators to control individual joints. More traditional manipulation planners were used to move CHIMP's arms through free space and grasp objects. Finally, the user interface was improved by adding various overlays to the 3D immersive display to give the remote operators better situational awareness.

Along the way, the team fine-tuned its approach to address the specifics of the competition as they became better known. For example, the severe restrictions on communications bandwidth for the indoor tasks mandated an approach whereby the environment model was transmitted only occasionally via a bandwidth burst while robot pose was transmitted continuously in between.

The team started practicing the tasks months in advance of the competition to identify areas for improvement and to train the operators. On the first day of the competition, CHIMP accomplished all eight tasks in under an hour despite several setbacks, including the robot falling as it entered the doorway. The run demonstrated the skill of the operators, the validity of the tools, and the ruggedness of the robot itself, even in the face of adversity. On the second day, CHIMP was plagued by problems in its communications software, resulting in dropped commands, and it ran out of time before it could complete the full slate of tasks. On the strength of the first run, Tartan Rescue captured 3rd place overall.

This paper tells the story of the CHIMP robot, going into detail on the improvements made in preparation for the DRC Finals and reviewing CHIMP's performance at the challenge. Building upon details provided in Stentz et al. (2015), we first overview the CHIMP robot, highlighting key features and specifications. In describing the software developed for the DRC Finals, we detail the approach taken for each DRC task while noting the overall software improvements performed. We also provide details on the hardware improvements made to the CHIMP robot, enabling tether-free operation via battery power and wireless communications while also strengthening and hardening the robot to survive potential falls. Lastly, the paper goes into significant detail on CHIMP's performance at the DRC Finals, providing a play-by-play account of the challenge and also focusing on lessons learned over the course of the project.

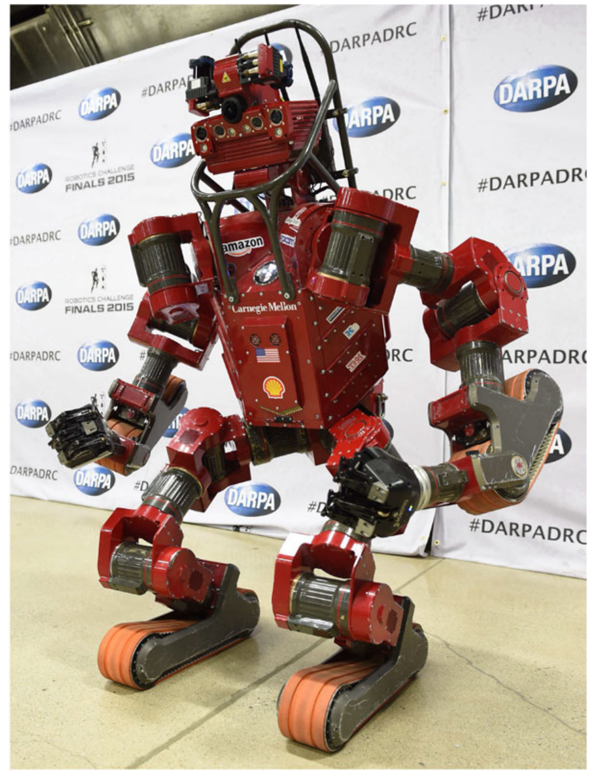

Figure 1. CHIMP, the CMU Highly Intelligent Mobile Platform, designed and built at Carnegie Mellon's National Robotics Engineering Center.

2. CHIMP OVERVIEW

Roughly the same size and form as a human, CHIMP was one of the more unique robot designs to emerge during the DARPA Robotics Challenge. CHIMP can operate in human environments, but it provides very unique locomotion capabilities. Although the robot is roughly anthropomorphic (Figure 1), CHIMP uses motorized track drives, embedded on the robot's limbs, to locomote. When traveling over uneven terrain and up stairs, CHIMP uses four track drive units—one on each limb—to maintain stability at all times. To manipulate objects, CHIMP stands upright on the two leg tracks, skid steering to move throughout an environment while grasping objects using grippers found on the robot's arms.

CHIMP was different from various DRC competitors in many ways. While most DRC teams pursued actively balanced humanoid robots, including some highly successful teams (Fallon et al., 2015; Johnson et al., 2015), CHIMP's design is one of static stability, thus reducing the overall need for balance control. Some teams pursued statically stable designs using four or more legs (Hebert et al., 2015), however CHIMP's design was a truly transformative one, allowing static stability in a variety of postures. After the DRC Trials, at least one team even added additional mechanisms to add static stability to an existing humanoid platform (Lim et al., 2015), thus, like CHIMP, transforming between various postures. Table I provides some of CHIMP's overall specifications.

CHIMP contains a total of 39 degrees-of-freedom (DOFs). Seven DOFs are dedicated to each of CHIMP's arms, in a traditional 3-1-3 kinematic arrangement where the shoulder and wrist DOFs create spherical joints. Six

Table I. CHIMP specifications.

Mass	201 kg
Standing Height	150 cm
Crawling Height	90 cm
Shoulder/Hip Width	74 cm
Degrees of Freedom	39
Actuator Type	Permanent Magnet Synchronous Motor
Drivetrain	Harmonic Drive
Brake	Parking Brake on all Drive Joints
Compliance	In-line torsional spring
Structure	Aluminum 7050-T7451
Components	800+ unique, 10,000+ total
Bus Voltage	60 VDC
Batteries	BB-2590 Li-Ion
Battery Energy	2.4 kWh
Computing	3x Intel Core i7 3820QM
Data	Gig-Ethernet, CAN bus
Positioning	Honeywell navigation grade IMU

DOFs are used on each leg of the robot. Four track drive motors help propel the robot using the rotating track belts. Manipulation is performed using two grippers, each of which contains Four DOFs. One last DOF is used to control the spinning of the LIDAR units atop CHIMP's head.

While many DRC teams utilized a hydraulic humanoid robot (Nelson et al., 2012), CHIMP uses a fully electric drive system. As humanoid robots place demands on both torque and power density, many unique motor designs have emerged during the DRC. New designs for liquid-cooled Urata, Nakanishi, Okada, & Inaba (2010) and air-cooled (Lim et al. 2015) electric motors are found on several competitors, thus enabling high-performance. CHIMP, in comparison, utilizes designs for extremely high-performance motors, but without the need for active cooling. Each of CHIMP's drive joints contains a custom frameless motor design with harmonic drive gearboxes and continuous output rotation (when kinematically feasible). Motor and joint output encoders provide accurate and absolute joint positioning, while a magnetically actuated parking brake allows each joint to hold position when powered off. A mechanical slip clutch, installed at the output flange of each motor, provides torque limiting, allowing a joint to slip rather than damage internal components. A torque tube that forms the output shaft provides additional protection against large impact loads via compliance. A temperature sensor on the motor windings allows the motor to be driven well above its continuous torque rating while ensuring that it does not overheat. These drive joints were designed in four different sizes and used throughout CHIMP's limb designs. Table II provides specifications for the drive joints.

Table II. CHIMP drive joint specifications.

	NGT-20	NGT-50	NGT-100	NGT-200
Continuous Motor Torque (Nm)	19	90	252	432
Peak Torque (Nm)	50	175	360	660
Continuous RPM	34.4	30.6	16.2	10.5
Mass (kg)	1.0	2.2	3.0	5.2
Length (mm)	90.5	113.5	130.5	135.0
Diameter (mm)	77.0	94.5	111.5	140.0

CHIMP includes several different sensing modalities to support navigation, situational awareness, and manipulation. Two LIDAR scanners capture 360 degrees of geometric data surrounding the robot, while panomorphic fisheye lenses add video texture for the geometric data. Multiple pairs of stereo cameras provide additional depth sensing and position estimation, while an internal inertial measurement unit (IMU) produces highly accurate inertial data. Six-DOF force/torque sensors on each wrist provide feedback during manipulation tasks.

In preparation for the DRC Trials, subsystem testing was performed to ensure individual component reliability prior to assembling the full robot. With well-understood performance and robustness by each subsystem, the overall uptime of the robot was extremely high. Over the course of a year and a half following the DRC Trials, the robot was unavailable (due to hardware maintenance) only a handful of times, thus allowing software development and testing to proceed at almost all times. Overall, the CHIMP hardware has been extremely rugged and reliable. Stentz et al. (2015) provides additional details on CHIMP's design.

3. SOFTWARE PREPARATION FOR THE DRC FINALS

As of the DRC Trials in December 2013, the CHIMP robot was capable of completing only a portion of the challenge tasks. Incomplete tasks largely included those focused on *mobility*—driving and egress from the vehicle, traveling across rough terrain, climbing the ladder/stairs—mostly due to our team's early focus on manipulation. Furthermore, the robot was assembled only six weeks prior to shipping to the DRC Trials, leaving the team with very limited time to develop the full body motions that were required for these mobility tasks. (In comparison, manipulation tasks had been tested for many months prior to the DRC Trials via the use of surrogate hardware).

As such, we focused our post-Trials efforts on completing all of the challenge tasks while also making certain CHIMP's entire skillset was both more robust and faster. In this section, we describe these improvements made to CHIMP's software. Stentz et al. (2015) provides an overview of the system architecture and individual

Figure 2. CHIMP in the Polaris vehicle at the DRC Finals, controlling the steering wheel with the track belt on the robot's left arm.

components that comprise CHIMP's software, and this section focuses largely on technical details of the improvements made to that system in the intervening year and a half between the Trials and the Finals.

3.1. Task Approaches

3.1.1. Vehicle Driving

For the vehicle driving task, we needed to situate a large, heavy robot inside the vehicle, actuate the various vehicle controls, and navigate through a moderately complex driving course. Since our team had skipped this task for the DRC Trials, we accelerated development for both vehicle driving and egress for the DRC Finals.

Given the extreme importance of vehicle egress, as described in the next section, determining the exact manner in which CHIMP sat in the vehicle was a critical first step. The team felt strongly that the robot should drive the vehicle without any additions or modifications to the vehicle itself. Early analysis showed it was possible for CHIMP to sit straight in the driver's seat, similar to a human driver, however we later found a "side saddle" posture that greatly reduced the complexity of egress. In this posture, CHIMP sat rotated approximately 45 degrees to the left, bracing its left leg track against the outside of the vehicle cab and grasping the roll-cage of the vehicle with its right arm. This posture allowed the use of the left arm and right leg for actuating the steering wheel and throttle. Figure 2 shows CHIMP operating the Polaris utility vehicle.

While we initially pursued grasping and turning the steering wheel with the gripper, we ultimately used CHIMP's track drive mechanism to turn the wheel, applying a small amount of pressure between the limb and wheel while turning the track motor. This approach allowed for fast turning of the steering wheel throughout its entire range. A self-retracting, ejectable paddle—attached to CHIMP's right heel—allowed the robot to push the throttle pedal from the side saddle posture. The self-retracting feature ensured that even if the software or safety systems disabled the hardware, the throttle paddle would back away from the throttle, allowing the vehicle to come to a halt without having to call for a manual emergency stop of the vehicle itself.

Given the larger distances covered by a robot in a vehicle, our pose system placed extra importance on visual odometry (compared to LIDAR odometry) when driving the vehicle. The presentation of the sensor data was also customized for vehicle driving, using full range LIDAR data to display voxelized representations of the robot's surroundings. The voxel updates together with a live video stream from the forward-facing cameras provided the operators with sufficient real-time situational awareness.

With real-time sensor data and telemetry from the robot, we decided it was feasible to teleoperate the vehicle rather than pursue vehicle autonomy. The operator used a joystick to command both steering and throttle, while onboard software translated the commands into motor controls. Safeguards from both software and hardware systems ensured the vehicle would come to a halt in case of any unforeseen issues (overheating motors, low battery, communications dropout, excessive CANbus errors, etc.).

3.1.2. Vehicle Egress

Exiting from the vehicle was the second task requiring new development. Given the complex motions necessary for CHIMP to safely exit the utility vehicle, we decided a simulation would be extremely useful for developing our egress behavior. We incorporated a full dynamics simulation of the CHIMP robot, allowing us to test tasks such as vehicle egress without requiring use of the physical robot. Our simulation was heavily based upon the Gazebo[1] simulator and the ODE[2] physics engine, and it was closely integrated into our control systems and software architecture. Tuning of the simulation focused on achieving acceptable simulation accuracy while maintaining near real-time performance. To reduce simulation complexity, we simplified our robot model to eliminate any joints that were not necessary for the egress maneuver. Simulation parameters were chosen by a brute force approach of automatically running a few test simulation scenarios hundreds of times while

[1]http://www.gazebosim.org/
[2]http://www.ode.org

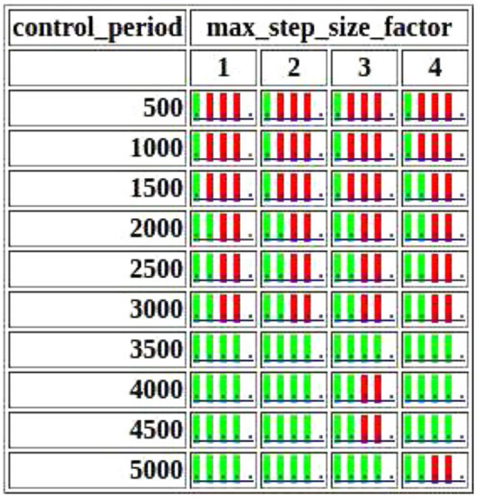

Figure 3. Simple simulation tests were repeated hundreds of times with varying parameters to select the best trade-off between speed and accuracy. This output shows the pass/fail result for two such parameters.

Table III. Key Gazebo parameters used for simulating CHIMP for egress, stair climbing, and fall recovery.

Gazebo parameter	CHIMP value
physics/type	ode
physics/ode/solver/type	quick
physics/ode/solver/iters	50
physics/ode/solver/sor	1.4
physics/ode/constraints/cfm	0.00001
physics/ode/constraints/erp	0.2
physics/ode/constraints/ contact_max_correcting_vel	100
physics/ode/constraints/ contact_surface_layer	0.003
physics/real_time_update_rate	1500
physics/max_step_size	0.00066667
robot/joint/dynamics/damping	3.2 – 23*
robot/joint/dynamics/friction	8 – 40*

Notes *Higher values for larger joints.

slightly varying simulation parameters, and scoring the results based on model stability and matching of real-world data (Figure 3). We present the resulting CHIMP-specific simulation values in Table III. We used these values to run Gazebo in lockstep with our software-based controller, and we achieved a real-time factor of between 1.2 and 1.5, depending on the complexity of the environment.

Egress was initially developed entirely in simulation, then tested and improved on the robot. Execution consisted of open-loop maneuvers that used preconditions to ensure each stage was completed successfully. CHIMP first released its right-arm grasp of the roll-cage, straightened out its left leg, pointing the tip of its leg track towards the ground, and braced its left arm against the vehicle exterior. It then actuated its right leg track against the vehicle floor to pivot the body out until the tip of the left track dropped down to contact the ground. Finally, it straightened its right leg out into the same extended position as the left leg and used its arm tracks to gracefully slide down the outside of the vehicle, returning the leg tracks to be flat on the floor. This strategy was primarily designed in simulation and tweaked afterwards using the robot and the Polaris utility vehicle. To ensure robustness, the approach was tested and tweaked while we varied starting conditions such as the robot seated position, vehicle height, pitch, roll, and the friction of the ground.

A common flaw with open-loop trajectories is their inability to recover when unexpected events occur. As such, we added a form of guarded autonomy by allowing the robot to monitor its own torso rotation at each segment of the egress trajectory, using accurate body orientation from CHIMP's pose system. If any segment did not result in the motion that was expected to occur, the robot halted and requested assistance from the human operator. The operator corrected the position as necessary and resumed the egress sequence.

3.1.3. Mobility

The DRC mobility task consisted of semi-random terrains constructed out of standard-sized cinder blocks, often sloped to provide unique shapes. One of the team's ultimate design choices was to develop a humanoid robot that could operate over these challenging terrains more like a vehicle, transforming into a posture on all four limbs and using the track drives to move the robot forward. While we explored four-limb mobility briefly prior to the DRC Trials (enough to secure a single point on the task in December 2013), for the DRC Finals we fully developed an "adaptive suspension" that allowed CHIMP to negotiate rough terrain.

CHIMP's adaptive suspension controlled each track's position relative to the torso to comply with the ground it traveled over. As each limb contains five degrees-of-freedom between the body and the track, there are a total of 20 DOFs available to program an adaptive suspension. Kinematic analysis showed that each track's position and orientation could be fully controllable relative to the other tracks, and another two degrees-of-freedom were available to shift the robot's body position. An early version of the adaptive suspension utilized these kinematic freedoms to adjust almost everything about CHIMP's posture on rough terrain. This included roll, pitch, and yaw angles of each track independently, relative x/y/z offsets between tracks, and shifting the body from side to side to balance its center of gravity. Multiple controllers ran in parallel, calculating corrective joint velocities that were summed together across

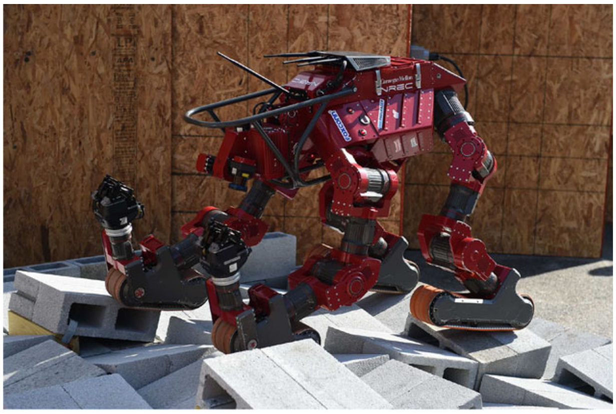

Figure 4. CHIMP crossing the mobility course at the DRC test bed.

all joints. While this software provided great versatility, we also found that it broke down easily whenever the robot encountered kinematic singularities or joint motion limits.

Back to the drawing board, we developed a second version of the adaptive suspension that greatly simplified use of the robot's kinematics. Each track's five DOFs were reduced to two controllable freedoms: the pitch angle of the track relative to the body, and the "throw" distance of the track (a mostly vertical translation similar to the travel of a physical suspension). This reduced the complex kinematics to only a handful of closed-form solutions that allowed each limb to independently conform to the environment while staying within a predetermined operating envelope.

With a goal of balancing the weight of the robot across multiple limbs, we developed additional software to estimate ground reaction forces on each track. This software used torque measurements from each joint (motor torques as well as the deflections of the torque tubes) to estimate ground reaction forces. A Similar difficulty arose when near kinematic singularities, however, our team sought a simpler solution. Through experimentation, we found that the ground reaction force estimation system could be reliably reduced to a binary ground contact sensor. To maintain balance, we developed a system that leveled the body, side to side, by adjusting the throw distance of each track. Augmented with the software-based ground contact sensing, this system aimed to keep all tracks in contact with the ground while maintaining minimal body roll. While this approach did not take advantage of the full force control nature of an adaptive suspension, we found it to be sufficient for CHIMP negotiating rough terrain. The technique was not foolproof, however, and it required a skilled operator using a lot of concentration in order to keep CHIMP out of unrecoverable situations. Figure 4 shows CHIMP on the mobility course at the DRC Testbed event in March 2015.

Ultimately, at the DRC Finals, teams were given the choice to operate on the mobility task *or* the debris task, and our team opted for debris. Thus, this behavior was not exercised during the DRC Finals.

3.1.4. Ascending Stairs

The software developed for mobility described above was adapted for climbing stairs. For both, CHIMP operated using four tracks to drive across uneven environments. Whereas our rough terrain behaviors required a control system to allow CHIMP to balance and comply with the terrain, we utilized largely open-loop sequences of actions to climb stairs.

Much of the stair-climbing development was done in the same simulation environment used for egress. Since the rise/run measurements of the stairs were not known until a few months before the finals, simulation allowed us to test our software on many different stair inclines.

To climb stairs, CHIMP ratcheted its way along. The front limbs slid forward until they rested upon the next step in a set of stairs. These limbs locked their tracks in place and used the friction between the tracks and the step to hold the robot as the rear limbs slid one step forward. The process was repeated as the robot climbed stairs. Because the robot's motions were parameterized based upon different stair dimensions, we augmented the stair climbing software with a custom perception system for estimating the rise and run of the stairs. While observing 3D sensor data, the user identified a set of stairs by clicking on the center of two different steps. The vector between the selected points defined a vertical plane along the stairs. We then robustly estimated the stairs' parameters by studying the LIDAR points in a region of interest defined by this plane. For a collection of points along a set of stairs, the Z coordinate was expected to cluster on the horizontal surface of each visible step. The constant spacing between peaks in a histogram along this dimension thus defined the rise parameter of the stairs. Similarly, LIDAR points along the horizontal axis of the stairs should cluster at each step's facing edge, thus defining the run of the stairs. Figure 5 visually describes this process. As the stair geometry was fully specified prior to the DRC Finals, this software was ultimately unnecessary.

Sequences of motions were developed initially to approach and mount the stairs from the four-limb mobility position, as well as to dismount the stairs onto a level platform atop. Later, in order to perform this task faster, we developed a method that allowed CHIMP to directly mount the stairs by leaning forward from the two-limb upright posture that was used at the Finals.

3.1.5. Opening Doors

Significant autonomy was added to the opening of doors. During the DRC Trials, CHIMP was able to open doors,

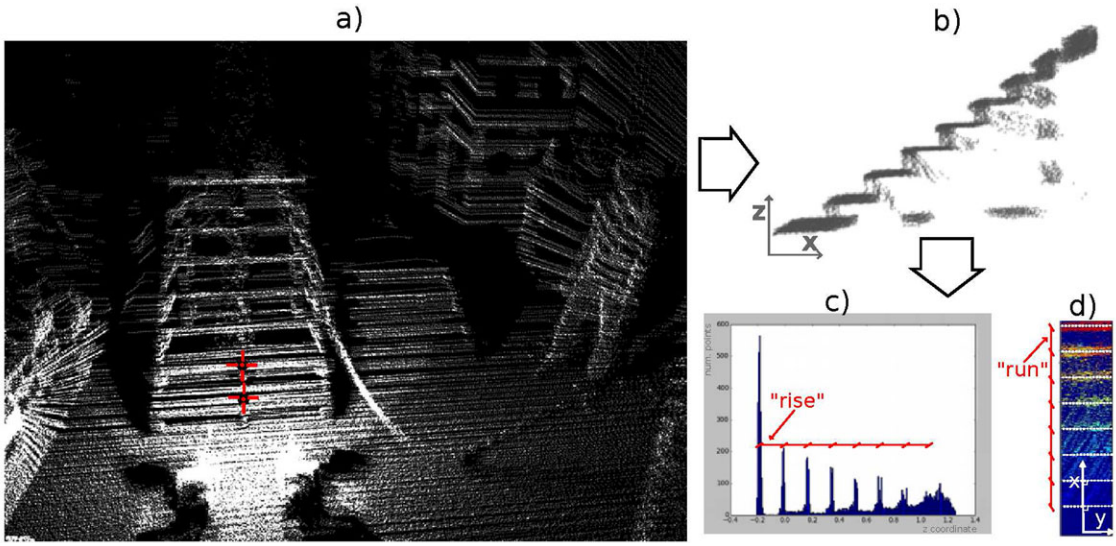

Figure 5. Perception routine for "rise" and "run" estimation. (a) Operator inputs: clicks on two different steps. (b) LIDAR points in the region of interest around the stairs (lateral view). (c) Height histogram for points in the considered region of interest. (d) Top projection of the stairs points colored by step.

largely through human operators manually guiding the robot into place and executing planned and teleoperated motions on a door handle. For the DRC Finals, we developed parametrized trajectories augmented with perception algorithms that could open both push doors and pull doors in a variety of configurations.

The first step was localizing the door. Perception systems were developed to simplify the process of estimating the position and orientation of both the door and its handle (see Figure 6). The operator triggered the process by clicking on the 3D interface twice: first click on the hinge of the door and second at the door's handle. A plane-fitting algorithm estimated the position and orientation of the door. To prevent bias in the estimated plane, the algorithm ignored points close the door's handle as well as points near the ground surface. Once the door was located, the handle's position was estimated using the height distribution of points in front of the estimated door plane.

Given the small size of a typical door handle, a major limiting factor was acquiring enough LIDAR points to accurately determine the shape and location of the handle. As such, this algorithm was only utilized once the robot was within 1.5 meters of the door.

With perception thus defining the geometry of the door, the operator manually positioned the robot's base at an appropriate location relative to the door, defined a priori through experimentation. The robot utilized the fist of the gripper to unlatch the door handle and give the door a small push, a change in approach since the DRC Trials (in which we used individual fingers), as fingers were likely to catch on the door handle with our previous approach.

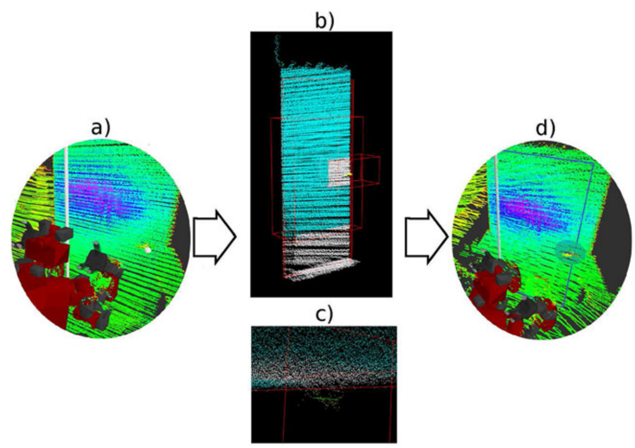

Figure 6. Perception routine for door detection. (a) Operator inputs: clicks on door hinge and handle. (b) Estimation of door leaf position and orientation using plane fitting. (c) Estimation of lever handle position and extent. (d) Algorithm outputs: door plane and handle "fixtures".

A fully autonomous system handled the actual opening of the door. Similar to our egress system, guarded autonomy allowed the robot to proceed with opening the door, halting execution and informing the operator only if an unexpected event occurred. The system, a state machine based upon Boost's Statechart,[3] first moved the robot's second arm up and out of the way so that it could clear the doorway.

[3]http://www.boost.org/

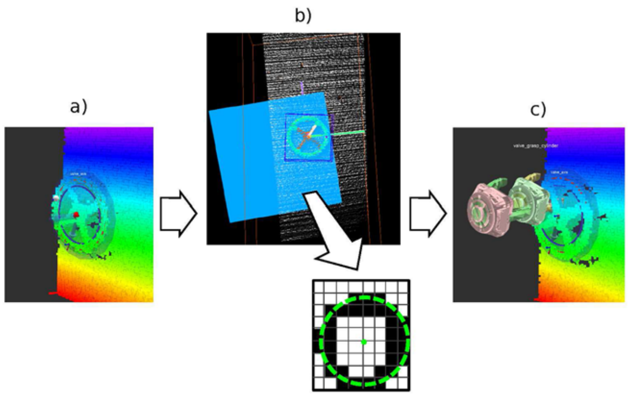

Figure 7. Perception routine for valve localization. (a) Operator inputs: clicks on valve center and rim. (b) Point cloud segmentation and circumference fitting. (c) Algorithm outputs: valve "fixture" and candidate grasp locations. The image shows a reachable "grasp" goal (green phantom gripper) but an unreachable "pregrasp" goal (in light red).

The unlatching procedure was then executed, after which the operator manually verified that the door had opened through visual inspection of imagery and LIDAR data. If the door failed to unlatch, the operator was able to restart the procedure to reattempt.

With the door unlatched and slightly open, the robot's arm was extended forward and it held the door open while the robot passed through. This additional step of holding the door open was incorporated after the DRC Trials, during which our team lost a considerable amount of time after a gust of wind closed a previously-open door. After the robot proceeded forward and through the door, the robot's arms were brought back to the nominal upright posture. At this time, the autonomy software was complete and the human operator controlled the robot once more.

3.1.6. Turning the Valve

We greatly simplified operation of the valve-turning task from the DRC Trials, largely due to perception algorithms that accurately determined the size, shape, and position of a valve.

Similar to the door task, the operator first teleoperated CHIMP to a favorable position near the valve, as determined by *a priori* guides visualized on the operator interface. These markers showed regions of maximum reachability, computed using a discrete cost-map of kinematic reachability, similar to Ruehl, Hermann, Xue, Kerscher, & Dillmann (2011). Navigating to favorable base locations such as this greatly reduced the time required to complete the task, as kinematic reachability greatly influenced the robot's ability to fully turn a valve.

Perception algorithms were similarly developed to speed up valve turning, as shown in Figure 7. The operator clicked on the center and the rim of the valve, with LIDAR data near to the valve used for a parametric circumference fitting algorithm. Our approach used a Random Sample Consensus (RANSAC) technique Fischler & Bolles (1981) to locate LIDAR points that matched the shape of a circular valve with the center and radius defined by the clicked points. This algorithm fully localized the valve by fitting a plane to the front-facing points on the valve, and estimating the radius of the valve from the outer points from the approximate center.

CHIMP was capable of grasping a valve either on the rim, thus using its arm kinematics to turn the valve, or by placing the gripper at the center of the valve, threading its fingers between the spokes, and spinning the wrist joint to turn the valve.

The grasp goal position was the final position to be attained by the gripper before closing its fingers on the valve while the pregrasp goal position was at a short standoff distance away from the grasp position along the direction of the valve axis. The idea was that the inverse kinematics engine first solved for the pregrasp location, which was in relatively voxel-free space, and then planned a simple straight line move from the pregrasp to grasp location. These goal positions were displayed on the operator interface using models of CHIMP's grippers to give the operator better situational awareness.

Moreover, our inverse kinematics system ran in the background, checking for solutions as soon as these positions were set. It then color-coded the aforementioned gripper models either green or red depending on whether it could find a solution or not. If either of the models was colored red, the operator knew right away that modifications had to be made to either to CHIMP's base position or the goal positions.

Once the positions were finalized, the operator then started the motion-planning system, which returned a set of trajectories for the robot to follow. The trajectory to get to the "pregrasp" location was a joint space plan using OMPL's implementation of the RRT-Connect algorithm,[4] while the simple move to the "grasp" location was done by incorporating a straight line constraint into the same algorithm. The trajectory to turn the valve at its rim was generated using the CBiRRT algorithm with a circular constraint defined using Task Space Regions Berenson, Srinivasa, Ferguson, and Kuffner (2009), while that to turn the valve at its axis was generated by a continuous rotation of the wrist joint angle. The user could then preview these trajectories if required before finally executing the motion.

3.1.7. Cutting the Wall

Given the extreme overall complexity of the wall-cutting task, we separated this task into multiple sub-tasks:

[4]http://ompl.kavrakilab.org/

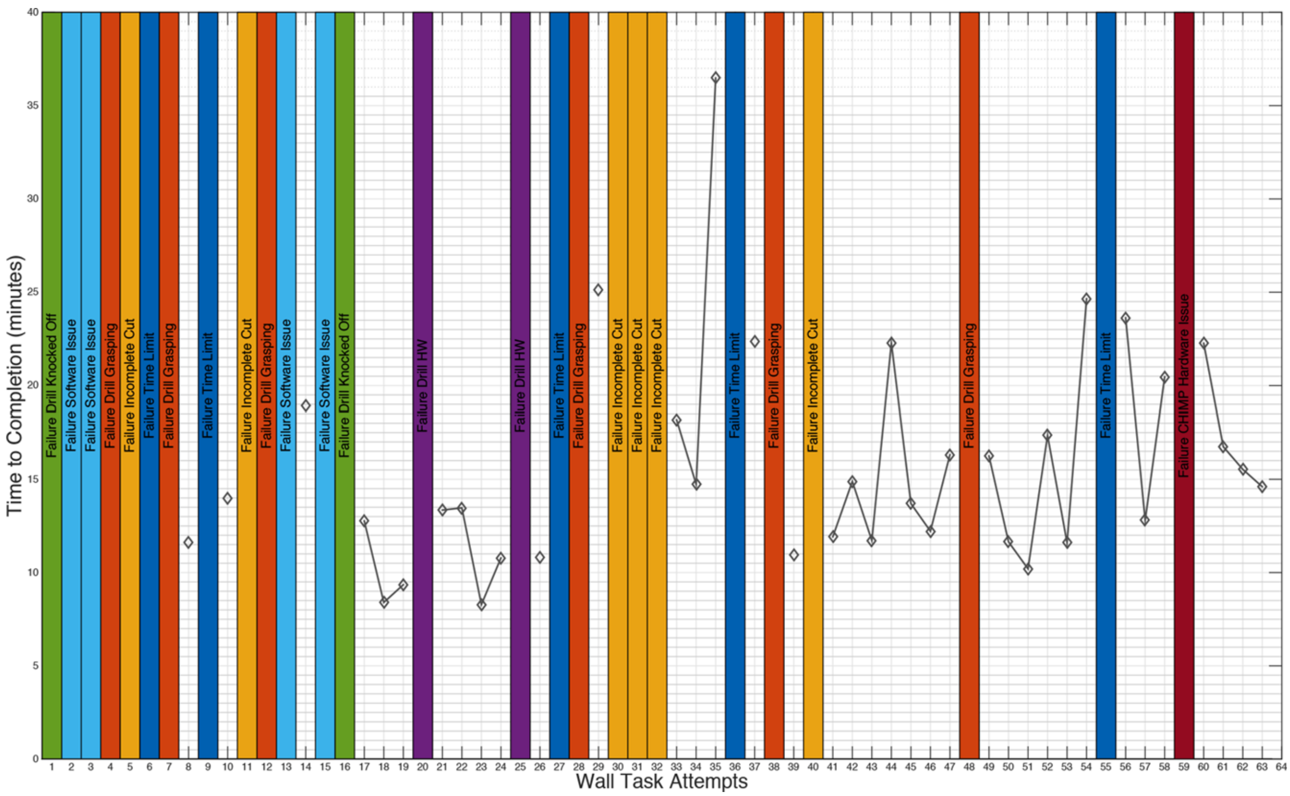

Figure 8. Evolution of task time and success rate over time for the wall task. In early attempts, both software and process needed improvement to fix reliability due to overly aggressive drill grasping and cutting strategies. As development continued, the success rate improved and execution times stabilized.

localizing and picking up the power tool, ensuring the robot could actuate the tool's trigger, and finally using the tool to cut the wall. In the DRC Trials, our team completed this task almost entirely under manual control by the operator. For the Finals, however, we implemented shared autonomy using a hierarchical state machine architecture called the "grasping agent" similar to Pitzer, Styer, Bersch, DuHadway, & Becker, (2011). This system, also using the Boost Statechart library, defined a state machine that could be utilized to automate large portions of the wall-cutting task. CHIMP was capable of grasping the drill autonomously, however if a problem occurred, the robot halted itself and alerted the operator. These functions helped the team recover from possible failure scenarios during each sub-task that would have affected our ability to complete the overall task (Figure 8).

Drawing upon our experience at the DRC Trials, we chose to use the drill tool (Dewalt DCD995) to cut the wall, as it was possible to pick up and actuate this tool with a single gripper. A major design choice was the manner of grasping and actuating the drill's trigger, shown in Figure 9, in which CHIMP's spread fingers encompassed the handle and trigger of the drill, while a third finger wrapped around the upper body of the drill.

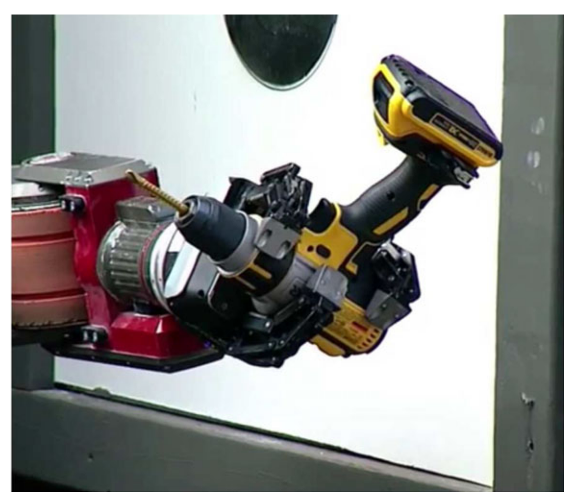

Figure 9. CHIMP grasping the drill with two fingers around the drill trigger and handle and a third finger holding the drill body firmly in place.

CHIMP was first required to pick up the drill from the shelf. To do so, the robot operator drove CHIMP near the

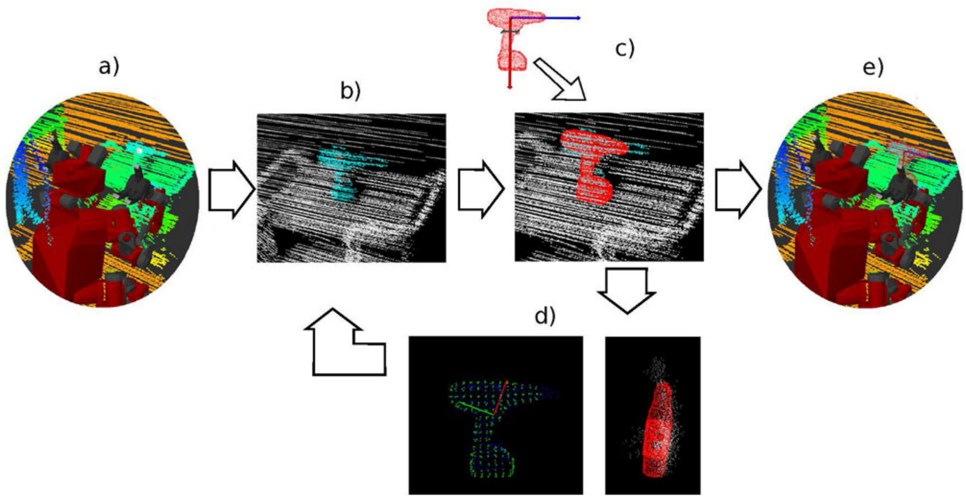

Figure 10. Perception routine for drill localization. (a) Operator input: click in the proximity of the tool. (b) Segmenting the point cloud in the volume of interest. (c) Model registration. (d) Evaluation of the model registration accuracy. (e) Algorithm output: position and orientation of the drill.

tool shelf with the help of visual guides in the operator interface. Base placement was crucial as the operator had to ensure that the robot arm could reach the drill and that the sensor head had a clear line of sight to it. Clear visibility was required to obtain accurate perception results and to allow manual verification of the gripper finger positions prior to grasping the drill.

We developed a perception system for detecting the drill's pose that the operator initialized by clicking on the 3D LIDAR points of the drill (see Figure 10). The first step of the algorithm segmented the points corresponding to the tool surface from other background elements (such as the horizontal support surface). This segmentation was based on certain assumptions identified in all DRC scenarios (e.g. the tool was in the vertical position with enough clearance space from background walls and other elements). As part of the preprocessing step, the system downsampled and smoothed the selected region of the point cloud to ensure uniform point density and to mitigate the effect of noisy LIDAR measurements. The next step was registering pre-existing models of the tool with the point cloud to accurately estimate its position in the scene. The algorithm computed an initial rough estimation of the pose using principal component analysis on the 3D points to find out the drill's dominant axes along the handle and chuck of the drill. The pose was then refined by aligning the reference model with the point cloud.

We experimented with different alignment algorithms, using both dense point-to-point distance metrics on the original point cloud and discrete salient features detected on the tool surface. For the sake of simplicity and computational efficiency, the final implementation was based on an iterative closest point (ICP) Besl & McKay (1992) algorithm. One of the limitations of this technique is the sensitivity to local minima when iteratively optimizing a pose using noisy point-to-point correspondences. To mitigate this problem, we added heuristics to assess the quality of the solution computed by the ICP subroutine. This process exploited the fact that large portions of the tool's body are convex volumes, thereby penalizing solutions that left unmatched points from the original point cloud inside the registered drill model. The algorithm explored the search space by launching multiple instances of ICPs seeded with slightly different versions of the original pose, ranking the solutions using the above-mentioned heuristics, and returning the best candidate pose of the drill.

Using a predefined grasp for picking up the drill, the robot's target gripper location was immediately visualized to the human operator, after which the operator could make small tweaks if necessary. The robot then autonomously planned a joint space trajectory using the RRT-Connect algorithm (Kuffner & LaValle 2000) to place the gripper around the drill. Next, the robot executed a sequence of small motions that lightly touched the drill from three different directions in an attempt to guide and center the gripper around the drill handle accurately. The force-torque sensor in CHIMP's wrist provided the required force feedback to confirm that the fingers had touched the drill each time. Finally, the gripper moved forward until the force-torque sensor indicated the palm had touched the drill's body, after which the operator closed the gripper's fingers around the drill.

Once the tool was grasped, the robot lifted the tool and the operator moved the robot away from the shelf. The software system then added a "collision object" approximately the size of the drill to CHIMP's robot model, which made the motion planners aware of the drill in hand. CHIMP then moved its arm to point the drill towards the cameras on its head, and it activated the trigger by squeezing its fingers. Simultaneously, the state machine obtained imagery of the drill before and after squeezing the trigger, sending highly compressed images to the operator over the low bandwidth link. The operator could then confirm that the drill was on by looking for a light on the drill that indicated that triggering was successful. If the drill was not triggered (light was off), the operator could attempt various methods to shift the drill in the palm of the hand before any reattempts.

Next, with the drill correctly held in the robot's gripper, the operator guided the robot into place to cut the wall, again with the help of guides on the operator interface. The operators on this task carried out a number of test runs to decide favorable base locations for performing the cut. Since the drill had to be maintained perpendicular to the wall during the cut, we had to ensure that the arm was kinematically capable of making the required movements from the base location. Next, similar to the detection of the drill pose, the operator clicked on the 3D LIDAR points of the wall to specify the cutting path. This, in turn, triggered the perception system that detected the wall plane and added a wall marker containing a cutting path to the operator interface.

The trajectory to cut the wall was computed by our meta-planning framework that concatenated together trajectories created by multiple kinds of planners. An unconstrained joint space trajectory was planned using RRT-Connect to bring the drill near the wall, and the same planner was used to make a constrained straight-line trajectory to push the drill bit into the wall. The task-space planner CBiRRT was then used to plan the square cuts, which were defined using four Task Space Regions (Berenson et al., 2009). The planning times for CBiRRT vary greatly, as does the path length and velocity. For this reason, we parallelized the planning system across all operator stations. As a result, we had multiple computers racing to produce the highest quality plan. The operator finally previewed these motions, and the robot autonomously executed the cut. Cartesian speed limits were imposed on the path by scaling the joint velocity limits for each individual trajectory section so that the max speed at the end-effector was at or below the Cartesian limit.

The operators monitored image and joint torque data during the cut for signs of potential faults. If the images showed the autonomous cut moving off the desired path, or if the motion faulted midway, the operator could continue the cut by teleoperating the robot arm relative to the wall surface. If joint torques were approaching their safety limit, the operator could reduce load on the motors by reducing the travel speed. As a final measure, the team developed functionality to slightly bump and push the drill body into the wall in case the cut section had not completely fallen

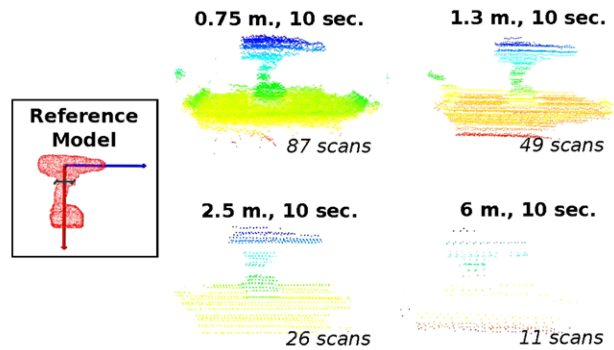

Figure 11. Density of the point clouds captured for the drill at different distances (using a 10 second accumulation buffer for the LIDAR scans).

Table IV. Number of successful model alignments for different drill orientations. A total of 145 experiments were performed for each orientation using different click (seed) locations.

orientation (deg.)	0	45	90	135	180	225	270	315
num. successful alignments	139	132	81	108	101	138	31	139

out. After the cut was completed, the robot backed away from the wall and placed the drill on the ground.

One factor that had a major impact on the performance of the drill detection (and all the perception algorithms in general) was the length of the accumulation buffer for LIDAR scans. The optimal choice for this parameter was a trade off between the latency in the perception result and the density achieved in the processed point clouds. Similarly, the distance to the target had a major impact on the number of registered LIDAR returns on the object (as illustrated in Figure 11). Based on experimental tests, we opted for an accumulation window of 10 seconds and operational distances up to 2 meters.

To evaluate the performance of the model registration algorithm, we captured point clouds of the drill at different orientations and quantified the error in reported position and orientation. Table IV illustrates the results with a distance from the drill of 60 cm and a maximum error acceptance criteria of 5cm position and 10 degrees orientation. Figure 12 displays the test orientations, and it shows that there is a correlation between orientations in which it is difficult to disambiguate the drill orientation due to the point of view, and to a higher number of failed alignments.

3.1.8. Debris

For the debris task, we dramatically changed the approach taken in the DRC Trials. Whereas at the Trials robots

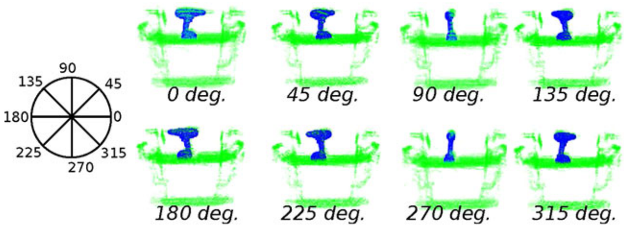

Figure 12. Test drill orientations and examples of successful model registrations.

typically performed precision pick-up maneuvers to move objects out of the way, for the DRC Finals robots simply had to negotiate a pile of loose debris. To do so, we tested many new postures that allowed CHIMP to push through the debris pile while maintaining stability and avoiding damage.

The result was two "bulldozing" modes. The first had the robot on four limbs extending its closed grippers straight out at ground level in order to push and sweep away loose debris. The grippers prevented debris from getting stuck in the tracks, and both sets of tracks could be slightly inclined to facilitate driving over uneven terrain or smaller debris. This posture has a shorter wheelbase than our typical four-limb mobility position, facilitating more responsive turning when skid-steering the tracks. The sensor head was also positioned higher off the ground, giving the operator a better view of the debris and reducing the risk of damage. The second mode had the robot on two limbs with its arms extended downwards such that the closed grippers were just in front of the leg tracks. This mode with its shorter base footprint was primarily intended for clearing debris in narrow passages and for pushing its way out of tighter debris-filled environments. This posture with arms extended downwards also ensured that any debris on top of CHIMP's arms in four-limb postures fell off while it stood up to the two-limb posture. In addition, we developed predefined "sweep" trajectories for both bulldozing modes that moved one of the front limbs off the ground and swept it to the side in order to clear lightweight debris from the path. Figure 13 shows what the operators could visualize from the control stations and the various options available to them. These trajectories were created in simulation and then tested on the robot against a variety of debris configurations.

Compared with the complexity required to pick up and move each object, having the robot push through was a much faster, simpler, and more pragmatic solution. The operators of the debris task spent a significant amount of time practicing with large debris piles in order to become comfortable with all of the tools available to them.

3.2. Additional Software Development

3.2.1. Positioning System

CHIMP used an advanced positioning system that fused many different data sources using a Kalman filter on an on-board embedded system (Stentz et al., 2015; George, Tardif, & Kelly, 2015). In addition to the data sources used previously—a navigation grade IMU, visual odometry, and limb odometry (measuring joint angles and track positions)—we incorporated LIDAR odometry measurements for the DRC Finals.

Whereas most of our data sources provide estimates of position change since the last measurement, for example visual odometry estimating speed and direction from optical flow, our LIDAR odometry provided an absolute measurement of position, similar to a GPS. The LIDAR odometry system did this by capturing a keyframe (a collection of LIDAR observations made over 10–20 seconds) and registering new data against this keyframe. In indoor environments with walls and surfaces to register, keyframes could last tens of minutes while CHIMP performed tasks, providing a drift-free pose estimate using the keyframe.

Highly accurate orientation direct from CHIMP's navigation-grade IMU greatly simplified our LIDAR odometry implementation. Rather than performing a full

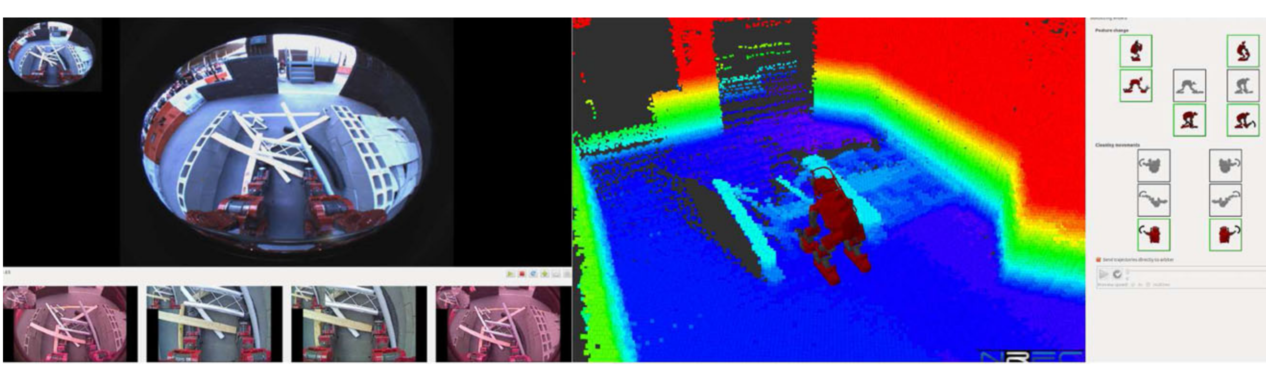

Figure 13. The operators' view at the interface during the debris task, containing images from the robot's cameras (left), a 3D model of the robot in its environment using voxel points (center), and a control panel for the operator to switch between different postures and/or activate debris clearing motions (right).

Figure 14. Colorized LIDAR point cloud resulting from CHIMP's drift-free pose estimates, captured over the course of an hour navigating the DRC Finals course.

six-dimensional ICP between incoming LIDAR data and keyframes, our algorithm only estimated the translational differences between the data. For a small translational offset from the keyframe, this reduced the ICP algorithm to a single iteration through the data, and our LIDAR odometry system ran at full sensor rate (40 Hz) with overall latencies less than 100 milliseconds. The approach was successfully used at the DRC Finals to largely cancel out position drift over the course of a run. Figure 14 shows an example of the overall point cloud produced during CHIMP's test run on the DRC Finals course.

Our positioning system had a few different modes it could be run under, namely, vehicle mode, mobility mode, and manipulation mode, each with its specific applications. Each mode activated a different set of aiding inputs for use in the Kalman filter to produce the pose result. The vehicle mode, used when CHIMP drove the vehicle, aided the INS with visual odometry, which worked better than LIDAR odometry in the outdoor environment due to its faster update rate and ability to detect features beyond the range of LIDAR. The mobility mode was used in tasks such as egress, debris, and stairs, which had the robot moving around a lot, often on four limbs. This mode used LIDAR odometry and IMU-based zero motion detection to calculate the pose result. Visual odometry was not used in this mode due to the often-occluded field of view of the cameras when driving in four-limb mode, and the additional slip that occurred when driving in four-limb mode caused the limb odometry to overestimate motion. The manipulation mode was used for the rest of the tasks, when CHIMP was upright and manipulating objects. This included limb odometry, taking advantage of the fact that CHIMP's legs and tracks were mostly stationary during manipulation, utilizing limb kinematics to estimate any motion at the IMU in the center of CHIMP's body.

In the absence of the LIDAR odometry input, the system demonstrated a maximum translational error of approximately 0.6% of the distance traveled, typical results for an INS aided with visual odometry, for instance. The addition of LIDAR odometry, however, greatly improved the accuracy of the pose system and limited pose drift. During the DRC Finals test course run, we estimated total pose drift to be only 6 cm over the course of 42 m of total motion on the "inside" portion of the course, equivalent to a drift of only 0.14% of total distance traveled.

3.2.2. Fall Recovery

An additional improvement made for the DRC Finals was our fall recovery system. Given early descriptions that robots would be required to get up after falls, we developed simulated fall recovery motions during the year leading up to the DRC Finals. The fall recovery system utilized the same Gazebo simulation used for our egress development, and it allowed rapid discovery of various strategies to recover from falls.

We identified some of the more likely configurations the robot would end up in after a fall, considering in particular tasks on uneven terrain such as mobility. These were (a) the robot falling on its side from a four-limb stance, (b) the robot rolling over onto its back from a four-limb stance, and c) the robot falling backwards from a two-limb stance, shown in Figure 15. We designed a chain of open-loop trajectories, starting from these positions, going through stable intermediate points and ending in the nominal four-limb position.

With CHIMP on its back, the first goal was to roll the robot onto its side. By shifting all four limbs in the direction of the roll, the robot's center of mass was offset enough to cause the entire robot to roll. Once on its side, CHIMP's limbs prevented the torso from rolling further. By outstretching the limbs lying on the ground along the axis of the torso, the limbs no longer prevented rotation and the robot pivoted over this axis. The other two limbs were used once again to shift the center of mass and cause CHIMP to roll onto its front side. Finally, the robot used all four limbs to lift its torso to its nominal four-limb position. Significant testing was performed to ensure that the power and torque values calculated in the Gazebo simulation matched the actual capabilities of the robot.

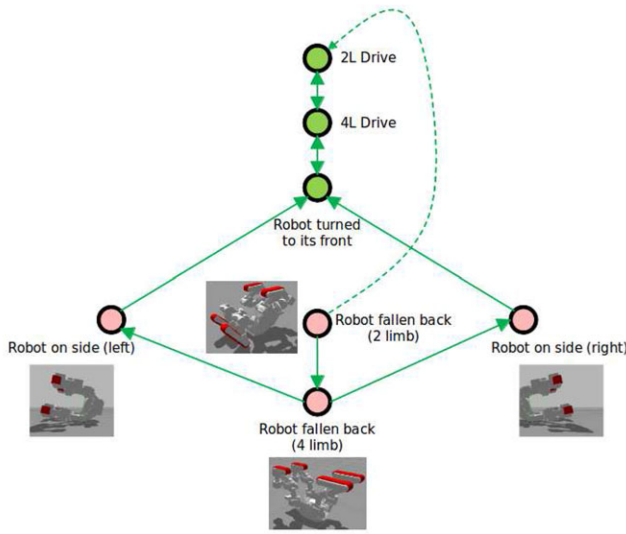

Figure 15. Open loop trajectories for CHIMP to recover from a fall.

While most of CHIMP's exterior is ruggedized to withstand a fall, some components such as the sensor head and communication setup are more sensitive than others, and the trajectories were developed keeping this in mind. The drive tracks were run in a "zero-torque mode" while executing fall recovery to prevent them from resisting the limb motions. In addition to these trajectories, the operator had joint teleoperation modes and the adaptive suspension system at his disposal to maneuver out of unforeseen positions.

In the final months leading up to the DRC Finals, we only tested fall recovery twice on the physical robot. Both sets of tests were performed on padded ground mats (to protect the robot from unintentional injury), but we proved that fall recovery was possible with the robot. We did not expect to ever utilize this software during competition.

3.2.3. Safety Systems

Additional improvements were made to how CHIMP responded when unexpected events occurred. As described in Stentz et al. (2015), CHIMP already halted motion in the case of unexpected power issues (such as a sudden drop or increase in voltage) and when the control system produced unexpectedly high torques. To this system we added several more conditions in which the robot halted operation and waited for user input before proceeding.

Although CHIMP is designed to be statically stable, it has a rather high center of gravity when standing on the two lower limbs, and thus it can maintain stability only on relatively level terrain. During testing, we occasionally encountered situations in which the remote operator unintentionally commanded CHIMP to drive over inclines or pieces of debris that were large enough to upset the stability and tip the robot over. In response, we developed a component to monitor the roll and pitch of the leg tracks when standing upright, and to automatically deactivate robot motion if the values were greater than a small threshold. The monitor could be disabled by the operator for situations in which they wanted to proceed regardless, but doing so caused a warning to be displayed on the screens of all operators until the monitor was re-enabled.

Another problem that the operators encountered was overloading particular joints while traversing the mobility course in the four-limb posture. The most common situation occurred when there were large lateral forces on one of the limbs due to either the lateral slope of the course or to a pair of limbs being held apart by a concrete block in between. The operator was typically unaware that the joints were overloaded, and as a result would make the situation worse until the joint started to backdrive and the robot posture collapsed. These failures were unrecoverable and required human assistance to extract the robot.

CHIMP drive units feature output clutches that slip beyond a particular torque threshold. We added software to detect clutch slipping by looking for excessive motion of the motor encoder relative to the output shaft encoder. The onset of clutch slipping triggered an onscreen warning; if it persisted for more than two seconds the joints were deactivated, which automatically engaged parking brakes to hold the position. Deactivating the robot prevented the situation from worsening, gave the operator a chance to assess the situation and decide on a new strategy, and forced the controller back to a reasonable target position in place of the previously unreachable one.

In all of these scenarios, CHIMP's ability to maintain static postures with parking brakes applied was a critical feature. Unlike humanoids that must balance even when not performing tasks, CHIMP was able to shut off all motor control and wait for human operator commands.

3.2.4. Additional Improvements

The DRC challenged teams in terms of the data transmitted between operators and robot. We continued with the same communications prioritization system we utilized for the DRC Trials (Stentz et al., 2015), but we made many improvements. Foremost, DARPA changed the network topology for the DRC Finals. Rather than having a single transmission link with variable bandwidth, the DRC Finals provided a constantly on low-bandwidth bidirectional link to the robot (limited to 9.6 Kbps of UDP/TCP traffic and 4.8 Kbps of ICMP traffic) augmented with a 300 Mbps unidirectional UDP link from the robot to the operators that occasionally turned on for only one second. We demarcated and prioritized data for each link, sending robot telemetry and critical information via the low-bandwidth connection, with sensor data transmitted over the high-bandwidth link. Each link had its own communications bridge to prioritize and transmit these data.

In addition, we made major improvements in terms of the bandwidth required to transmit individual messages to and from the robot. Rather than transmitting messages using their default, naive serialization methods, we developed custom encodings on a per message type basis to minimize bandwidth while ensuring data resolution was sufficient. On the low-bandwidth links, we added intelligent UDP packet grouping to reduce the amount of packet overhead required. We also developed methods to very efficiently transmit the robot state. Our protocol transmitted a full robot state once every 10 s, and it only transmitted changes to the robot state in between, utilizing a variable resolution data packing strategy to accommodate both large and small changes to joint angles. The robot state compression (30:1 ratio) coupled with reducing the update frequency (25:1 ratio) allowed the operators to receive telemetry at 4 Hz while using only 1.44 Kbps. The balance of the 9.6 Kbps downstream bandwidth was used for sending network confirmations and diagnostic messages as necessary. A protocol similar to the robot state compression was used for sending planned motion trajectories back to the robot, whereby the first point was sent in full detail but subsequent points were sent as differential values for only the joints that changed.

We also made use of the available ICMP bandwidth in order to keep the system clocks on the operator computers synchronized with the clocks on the robot side. ICMP protocol includes a time-stamp option that was within the allowable use and was perfectly suited for clock synchronization. With system clocks synchronized via an independent pathway, we were able to detect and measure unusual latencies on the 9.6 Kbps UDP/TCP link, which we indicated on-screen to the operators so that they would expect extra feedback delay when teleoperating the robot.

Our motion-planning system incorporated various changes that made it faster to use. Other than predefined maneuvers, all motion planning at the DRC Trials utilized the CBiRRT algorithm (Berenson et al., 2009), an algorithm tailored for workspace constraints, such as those encountered when turning valves, moving a drill bit along a desired path, etc. For simplicity, this algorithm was utilized even when moving the limbs in unconstrained ways, however it often took several seconds to obtain motion plans even in simple surroundings. For the DRC Finals, we chose to instead utilize the OMPL package Şucan, Moll, & Kavraki (2012) whenever planning unconstrained limb motions. With a collision model comprised of only spheres for fast self-collision checking (Figure 16), these planning algorithms dramatically sped up motion plans.

When operating CHIMP, motions plans were typically generated by the operator and then transmitted to the robot for execution. Given the dramatically reduced bandwidth from operator to robot, we began to utilize motion planning on the robot whenever possible. This was utilized by onboard autonomy such that trajectories were not sent across the communications link for confirmation by the operators.

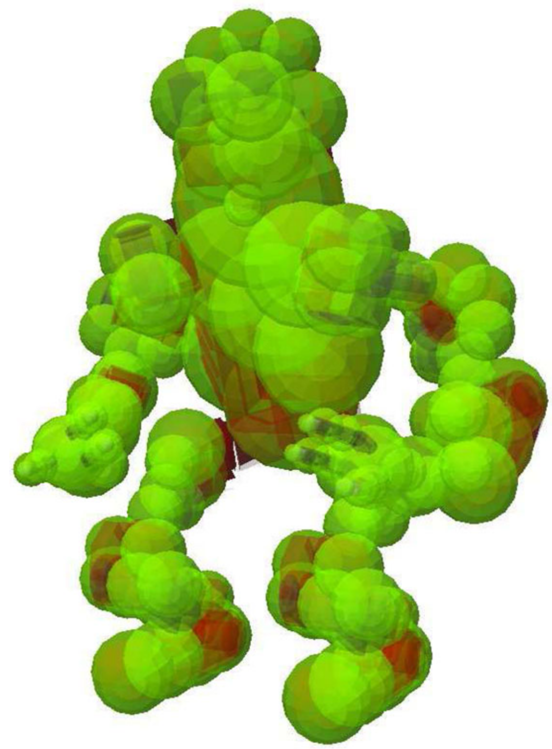

Figure 16. A collision model of CHIMP composed entirely of spheres in order to minimize 3D collision-checking time.

Switching CHIMP to battery power brought with it a need for power management. While using tether-supplied power for the trials phase, we only concerned ourselves with keeping each motor within its safe operating range in terms of both peak current and winding temperature. If we commanded multiple motors to draw high levels of power simultaneously, our Sorenson supply dropped the voltage as necessary to keep the total current below a preset limit. The battery modules, however, had specific current limits and would cut power entirely if the limits were exceeded, triggering a reboot of the entire robot. Managing the total power became the responsibility of our control software. To do so, we required an accurate estimate of total power and a strategy for keeping it within limits. The power estimate was calculated by taking the commanded torque for each motor and multiplying it by the measured velocity and an efficiency factor (mechanical power), adding in the resistive loss in the windings (electrical power), and adding a constant “hotel load” to account for the relatively fixed draw of the on-board computers, sensors, and communications gear. The management strategy consisted of scaling back the torques of all motors by a factor that

Figure 17. CHIMP's battery pack and Li-ion battery modules.

kept the total power to within a specified percentage of the maximum that the batteries could supply. This approach allowed the operators to push the robot as hard as they liked without triggering a battery blackout.

4. HARDWARE IMPROVEMENTS TO THE CHIMP ROBOT

In addition to the wide variety of software developments, we made multiple improvements to CHIMP's hardware in order to support the additional rigors of the DRC Finals.

4.1. Enabling Tether-free Operation

The first critical step in preparing CHIMP for the DRC Finals was to ensure that robot could be completely functional while untethered, requiring removal of all off-board power, wired communications, and belay support. This necessitated the switchover to battery power, wireless communications, and a stable machine built to avoid falls.

At the DRC Trials, CHIMP was operated exclusively via a power tether, even though the robot was originally designed for battery use. CHIMP's battery (Figure 17) was finished in early 2014, a large pack that loads into slots in CHIMP's back. The pack consists of eight high-capacity BB-2590 battery modules, a common battery design often found in military electronics and portable robots. The modules were selected for their battery chemistry (lithium ion), high-energy density compared to alternatives, rugged design for hostile environments, and multiple layers of built-in protection inside each battery module. Each module has a rating of 10 Ah at a nominal voltage of roughly 30 V. An entire battery pack consists of four parallel stacks of two modules, for a total energy of 2,400 Wh, enough for CHIMP to continuously perform tasks for approximately 2 hours. The pack provides details on each module's voltage and current via a CANbus interface, allowing the software to protect the modules from excessive discharge and keep the operators apprised of the remaining battery energy.

With added battery mass—a total of 16 kg, bringing CHIMP's total mass to 201 kg—additional upgrades were made to CHIMP's lower limbs to compensate. CHIMP's knees, joints that typically carry a large portion of CHIMP's mass when in upright postures, were capable of carrying the extra load, however the parking brakes on the joints were slightly undersized, meaning that CHIMP could fall backwards when parked in statically stable postures. These brakes were strengthened and CHIMP's typical postures were adjusted slightly forward. Additionally, CHIMP's thigh joints were not strong enough to hold the legs steady when skid steering on high friction surfaces (thus causing clutch slips). These joints were upgraded to provide approximately 2.8 times the torque they previously had. This upgrade required minimal effort and downtime due to our use of common joint sizes and a modular limb structure.

In addition to incorporating the battery and upgrading lower limb joints, we additionally made wireless communications possible. A Magnatek wireless MSTOP (Motion Stop) solution and a Ubiquiti wireless radio were integrated into CHIMP's torso, with antennas mounted on the robot's shoulders, to provide full wireless control of the robot. Before the DRC Finals, these were replaced with DARPA's requested hardware, an HRI wireless MSTOP integrated into the torso (requiring minor modifications to CHIMP's safety subsystem) and the Netgear R7000 wireless radio. The wireless radio was unfortunately too large to integrate within CHIMP's torso, so a custom enclosure was designed to house and slightly ruggedize the COTS component.

4.2. Surviving Falls and Unexpected Failures

The last major step in removing all of the wires, cables, and supports from CHIMP was ensuring the robot could survive a fall. While CHIMP was designed to minimize the risk of falling over, we designed the robot anticipating the worst, a decision that paid off during the competition. For the most part, CHIMP was already an extremely rugged machine, however we found several opportunities to improve the robot, making certain all components were rugged enough to continue functioning even after sudden impact loads.

Many components on CHIMP's body were stiffened, for instance side panels that cover electronics and computing were reinforced so they would not deform under high impact loads. CHIMP's joints utilize integrated slip clutches to give way when high impact torques occur, thus the robot mostly crumples when it falls and hits a hard object. The biggest danger, however, was that the robot could collapse and hit its sensorhead, an extremely critical component housing relatively fragile sensors. A roll cage was

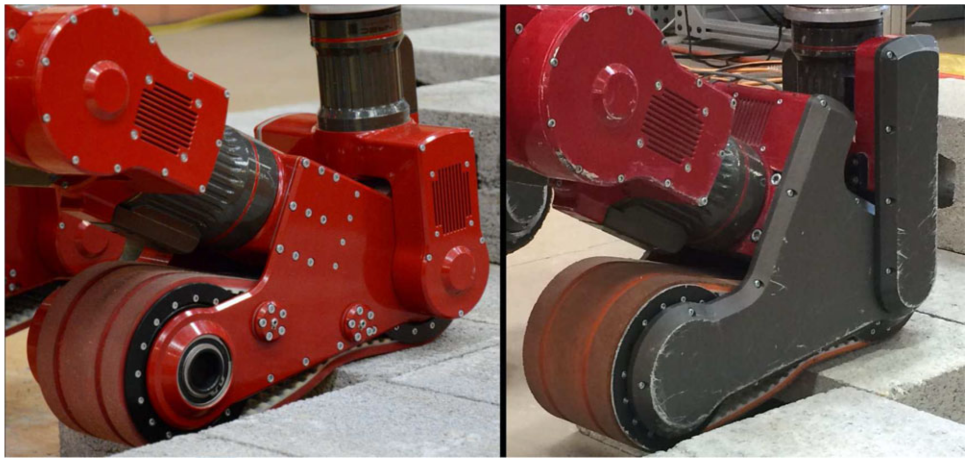

Figure 18. CHIMP's original track (left) and the redesigned version (right), with fewer protrusions to catch on ground obstacles.

added to surround CHIMP's head, in order to protect it from forward and backward falls. The roll cage also provided a space to mount the DARPA wireless radio unit, behind CHIMP's sensorhead.

The connectors on CHIMP's Robotiq grippers were another component susceptible to impact failure. The stock grippers have external connectors near the wrist, so our team worked with Robotiq to utilize internal cable routing, dramatically reducing the risk of difficult-to-repair damage to the grippers.

4.3. Additional Improvements

Two more design improvements were made to allow CHIMP to operate on the DRC Finals course. The rough terrain mobility task consisted of CHIMP using all four limbs to drive across piles of cinder blocks. While CHIMP has four tracks to provide rough terrain mobility, the structure on the tracks did not have sufficient ground clearance and would often get caught on rough cinder block edges. The entire track structure was redesigned and components resituated into different locations, thus reducing the amount of track structure that was exposed and improving CHIMP's overall mobility (Figure 18). This modification was made without affecting CHIMP's kinematic range of motion. Similarly, we modified the gear ratio on the track motors to increase overall track torque by approximately 43%, providing extra ability to negotiate steep slopes and climb stairs.

Second, an approach was developed to press the gas pedal of the Polaris vehicle using a normally unused joint near the distal end of CHIMP's lower limbs. For the DRC Trials, fold-out feet were installed on these actuators and were used for ladder climbing, but they had remained unused since that time. A custom paddle (Figure 19) was designed that was installed onto the output of the actuator. This paddle provided additional reach to press the throttle of the

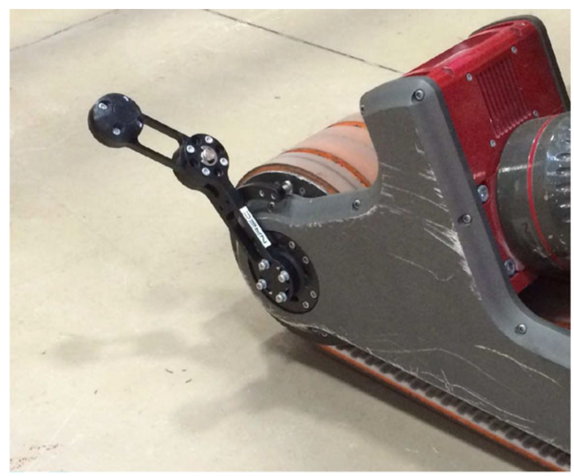

Figure 19. An extension allowed a motor in CHIMP's heel to depress the throttle pedal during vehicle driving.

utility vehicle, it had a spring return mechanism to auto-retract if software control was turned off, and it could be completely ejected by pushing it against a hard stop at the extent of its range of motion. This design allowed CHIMP to discard the paddle after driving the utility vehicle.

5. DARPA ROBOTICS CHALLENGE FINALS

Competitors at the DARPA Robotics Challenge were first scored on the number of tasks completed in under an hour. Only if a tie occurred did total time factor into a team's ranking. As such, after the DRC Trials, we focused our attention on reaching feature completeness with the CHIMP robot, ensuring it could complete all challenge tasks. After reaching this point months prior to the challenge, we focused our efforts on practicing as much as possible in the time remaining up until the challenge. This strategy proved critical to the team, as the lessons learned from many hours of practice helped our operators deal with unforeseen issues that arose during our challenge runs. CHIMP, working in tandem with the robot's human operators, completed all eight tasks in the DRC Finals.

5.1. Focused Robot Testing

Our team reached feature completeness with the CHIMP robot in early 2015, with the robot capable of completing all DRC tasks. The amount of time required, however, was far greater than the 1 h limit, thus our remaining development focused on speeding up operations, primarily through focused robot testing and increasing robustness.

Full system testing was critical in preparing CHIMP for the DRC Finals. After the DARPA testbed event in early March (where teams were given a preview of what to expect at the DRC Finals), our team was capable of completing only approximately two tasks (of eight) in one continuous hour of testing. While software issues hindered performance, the role of the operators and operator training were critical to speeding up CHIMP operations. Afterwards, our team began conducting full end-to-end tests on a weekly basis, providing an opportunity to incorporate new software updates while continuously analyzing our performance to focus our efforts. As the challenge neared, these tests increased in frequency to multiple times per week, then ultimately every day in the last few weeks.

We collected performance data from all tests, both quantitative (time spent for each task compared to target completion times) and qualitative (whether the robot completed the task and what issues arose during testing). Analyzing these data each week allowed the team to focus efforts on tasks that lagged behind others—sometimes spending multiple days of effort on a single task as necessary—while also providing an estimate as to how the robot would perform at the DRC Finals. This testing additionally allowed further shakeout of the robot's hardware, giving a sense of the frequency and type of hardware failures, along with expected times to repair.

Table V provides additional detail on the types of failures that occurred during these full tests on a practice DRC course built at NREC. By and large, operator errors were the dominant mode of failure, with the number one cause being operator performing an action that resulted in the robot falling or becoming stuck. Software issues were the second cause of failure during our full test runs. Surprisingly, hardware errors only very rarely caused failure. This result is a testament both to CHIMP's overall hardware reliability as well as our team's diligence at keeping the robot in shape for our test runs.

CHIMP and its operators completed a total of 38 test runs in the months before the DRC Finals. Figure 20 shows both the tally of tasks completed in each run as well as the overall time of completion. Both improved dramatically after only a few tests, presumably due to operators becoming more experienced through practice. Additional practice over the course of dozens of tests further improved our team's performance.

Table V. Tabulation of all success and failure types during task testing leading up to the DRC Finals.

Test Result	Subtotal	Total
Success		**261**
Operator Error		**22**
Operator caused robot to fall or become stuck	10	
Operator aborted grasping maneuver	7	
Operator failed to complete task	5	
Software Failure		**10**
General software issue on robot	5	
Perception system failure	3	
Software configuration error	1	
Operator control software issue	1	
Hardware Failure		**4**
Track drive system failure	3	
Robot drive joint failure	1	
Other		**1**
Task not configured correctly	1	

Going into the DRC Finals, CHIMP was able to complete all tasks in under an hour, and our team was doing so with near 100% reliability. We had very high confidence that CHIMP would perform well at the Finals.

5.2. Performance at the Robotics Challenge Finals

CHIMP was one of the top performers at the DRC Finals, completing all eight tasks in under an hour. More impressive to us, however, was that our team was successful despite a variety of unexpected events, system failures, and operator mistakes that transpired during the run. The greater story told here is how our team reacted to and recovered from all of these failures, demonstrating the robustness of our robot and the problem solving capabilities of the team of human operators.

5.2.1. Day 1 Performance

After a successful practice run at the DRC Finals (CHIMP completed all eight tasks in under an hour), our team was one of the last to perform on Day 1 of the competition. We summarize here the events that transpired as CHIMP completed the challenge.

- *5:00 PM*: CHIMP began its run, driving the Polaris utility vehicle from the start line. All times noted below are given in minutes and seconds after 5:00 p.m.

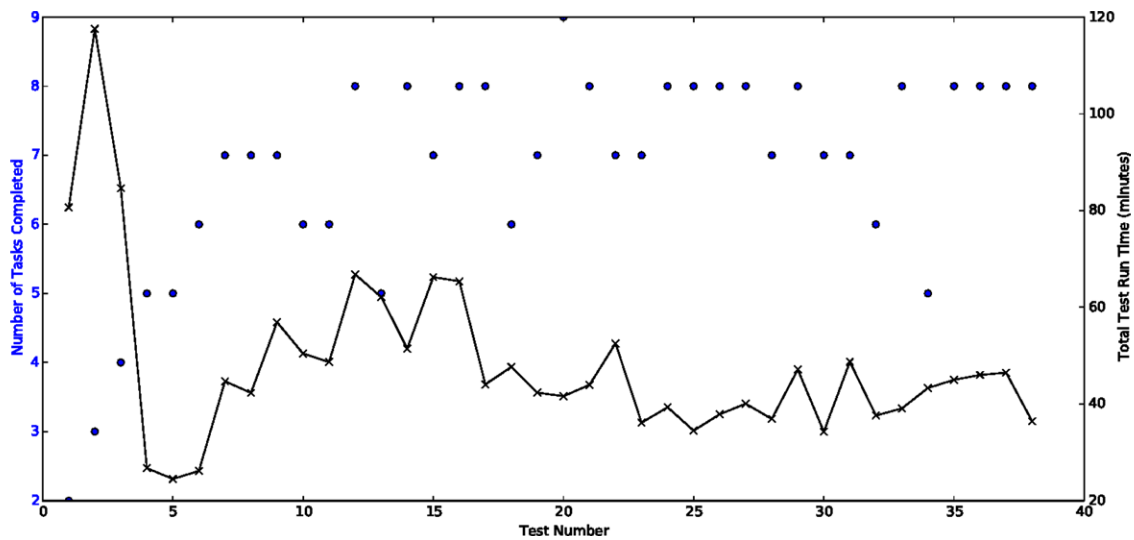

Figure 20. Overall task completion and total time during practice DRC course runs, conducted during March, April, and May, 2015.

- 1:41: CHIMP successfully drove the Polaris vehicle through the vehicle course, coming to a stop near the doorway.
- 2:04: CHIMP began the egress maneuver.
- 3:43: Partway through the egress maneuver, CHIMP paused, alerting the operator that it had not fully completed one of the motions necessary for egress. After studying the robot posture, the operators resumed egress, noting that the robot was only marginally off from desired posture. Post run analysis indicated that the robot failed to fully turn due to excessive dust on the floor of the vehicle cab, thus affecting friction with the track belts on CHIMP's legs.
- 4:39: After CHIMP had exited the vehicle and was down with tracks on the ground, one of the arm tracks became stuck on the vehicle cab. With no autonomous software to support automatically fixing this issue, the operators halted egress and began to attempt various motions to free the stuck arm.
- 13:09: After multiple attempts (including attempting to move away from the vehicle while still attached, almost resulting in a fall), the robot became unhooked from the vehicle and was able to continue operation.
- 15:51: Guided by its human operators, CHIMP adjusted its posture and navigated around the vehicle to approach the door.
- 17:30: At the door, CHIMP struggled a bit as it applied a lot of pressure while opening the door, causing the robot's knees to sag downward. The door swung open and the team decided to manually teleoperate the robot through the door.
- 17:47: With CHIMP's arms outstretched and knees sagging, the team applied a small position correction to the knee joints. Unfortunately, with the center of mass forward due to the outstretched arms, the position correction pushed CHIMP past the limit of static stability.
- 17:52: CHIMP fell forward and landed largely on its front, halfway through the door, with its right arm pinned under the body.
- 18:16: As the field team realized what occurred and began to prepare recovery tools for a robot intervention (and a 10 minute penalty), the remote operators decided to proceed with semi-autonomous fall recovery maneuvers.
- 22:32: After several minutes attempting various motions, the operators were successful in getting CHIMP to roll over onto its side (Figure 21), thus allowing the robot to free the arm pinned under the body.

Figure 21. CHIMP, having fallen through the door opening, during its successful fall recovery maneuver.

- 24:14: CHIMP continued to execute fall recovery motions, stretching all limbs fore-aft, rolling back onto its front, then rolling back into the prostrate four-limb posture.
- 24:55: CHIMP finished recovering from the fall, leaving all four tracks on the ground but still straddling the doorway.
- 25:54: The team initiated a 60 second recalibration of the pose system, during which CHIMP aligned its IMU by sensing the rotational motion of the Earth.
- 26:22: Operators determined that the fall disrupted communication to the embedded system that interfaces with the LIDAR sensors on CHIMP's head (an extremely rare event that was previously encountered only during high-temperature testing in a thermal chamber), and they executed a remote command to power cycle the embedded system.
- 27:03: CHIMP moved forward and cleared the doorway. Disrupted communications took effect, per DARPA rules.
- 27:22: CHIMP transitioned back to an upright two-limb posture. After a total of 9 min and 30 s, CHIMP was finally ready to continue the challenge tasks. While the time required to complete the fall recovery and system checkout was close to the time penalty that would have been incurred had the field team intervened, the robot was able to demonstrate extreme robustness and recovery from failure.
- 28:43: CHIMP navigated to the valve task and turned the valve with its right arm. The operator display indicated a near perfect valve turn, however sporadic camera imagery and joint feedback suggested that the arm was not quite perfectly aligned with the valve while turning it.
- 31:09: CHIMP navigated to and completed the surprise task, using the robot's left arm to push down a large knife switch.
- 34:07: CHIMP moved to the wall cutting task and prepared to pick up the drill with the right arm. As CHIMP moved its gripper toward the drill, the operators noticed a large amount of kinematic error between the visualized arm position and reality (similar to what was observed when turning the valve). The team decided the arm's calibration was no longer valid (due to the fall onto the arm minutes prior), and switched strategy to use the left arm instead.
- 37:39: CHIMP maneuvered to turn its body and pick up the second drill using the its left arm.
- 39:45: After making certain it could properly pull the drill trigger, the robot moved to the wall to begin the cut.
- 42:12: CHIMP began cutting out a shape from the wall using the handheld drill.
- 44:06: Halfway through the cut, the drill embedded too deep into the wall, and started to pull CHIMP's arm away from the nominal cutting path, thus risking task failure. The remote operators took over control and manually teleoperated the arm to trace out a shape, instead of the normal autonomous operation.
- 44:06: While the operators were manually finishing the cut maneuver, the robot's normally reliable pose system crashed. The pose system is a critical piece that provides both robot positioning as well as time synchronization to all of CHIMP's computers and embedded systems. With only two tasks remaining, the team decided to forgo precise 3D data and instead rely upon live video feeds from the robot. All system clocks on the robot were free running for the remainder of the hour.
- 46:48: The robot finished the wall cutting maneuver and gently placed the drill on the ground.
- 49:53: The robot transformed into "bulldozing" posture and pushed its way through the pile of debris.
- 50:10: After completing the debris task, the operators commanded a transition from "bulldozing" posture back to nominal upright two-limb posture, but shortly into the motion the robot encountered a control system fault, aborting the trajectory.
- 51:12: The operators attempted to move directly to two-limb posture, but the robot became unstable and tipped forward onto its elbows. The team quickly recovered and executed a proper transition back to the two-limb posture.
- 51:50: CHIMP began climbing up the stairs.
- 52:39: CHIMP repeatedly slipped off the top step while attempting to drive and climb up the stairs using its track drives. This failure was later determined to also be caused by excessive dust on the surface of the metal stairs.
- 54:08: The remote operators realized that friction was the problem and performed a third attempt on the stairs, this time slowing down CHIMP's motions by 50%. CHIMP's front tracks reach the top step and the robot continued to climb.
- 55:15: The robot, with all four tracks fully atop the stairs, finished the DARPA Robotics Challenge course, completing all eight tasks.

In the end, CHIMP consumed unnecessary time due to forced and unforced errors. While not the most facile completion of all eight DRC tasks, CHIMP's run on Day 1 demonstrated extreme robustness to failure, due to a heavily ruggedized machine that could survive and recover from a fall, as well as a well-practiced team of operators able to react to unexpected events.

5.2.2. Day 2

The second day of the competition was less successful for the team. Two major issues prevented us from answering to the 1st and 2nd place runs that HUBO (KAIST) Lim et al. (2015) and Running Man (IHMC) Johnson et al. (2015) had completed minutes before CHIMP began its Day 2 run.

First was a network error that seemed to prevent operator commands from being received by the robot. Analysis afterwards suggested this was a preventable configuration issue that had not arisen during previous testing. This network issue resulted in problems throughout the run, causing, for instance, CHIMP to attempt to open a door handle at the wrong height multiple times, and the robot to drop the first drill tool before even cutting the wall.

Second, during the plug task, in which the robot must move a magnetic plug from one socket to another, we had an extreme amount of difficulty due to the plug partially falling out of the robot's gripper during grasping. With the plug lying askew, our teleoperation tools were not suited to the awkward angle relative to the gripper, making a straight insertion of the plug into the hole rather difficult. Three different remote operators were able to alternate before ultimately succeeding at the task, but taking a rather long time to do so. CHIMP completed most of the tasks but was unable to finish the course in the time allowed.

5.3. Lessons Learned

There were a variety of lessons our team learned while preparing the robot for competition at the DRC Finals. We note in this section important decisions crucial to our success, efforts that were ultimately not utilized during the DRC Finals, and areas for future design improvements.

5.3.1. Important Decisions Validated

Many decisions our team made were based upon past experience and intuition. We highlight a handful of these decisions and how we felt they were critical to our success as a team.

First and foremost was the overall simple design of CHIMP. We chose a statically stable humanoid platform for pragmatic reasons, as we felt that a balancing humanoid was only one way to build a disaster-response robot. CHIMP's statically stable design with built-in parking brakes was extremely useful during the challenge, and we felt this decision was validated by other teams who also chose statically stable designs, including teams that incorporated such designs after the DRC Trials.

We opted to build only a single robot. While multiple robots would have been useful at times, it would have created additional hardware to support and keep functional. With a single robot, hardware failures had to be fixed as soon as possible, and we focused our efforts on making the single robot as robust as possible with minimal downtime.

With our sensor selection, we made heavy use of the LIDAR data, and relatively little use of stereo disparity calculations. Part of this was due to CHIMP's rigid neck, thus restricting the usable stereo field of view, but it was also due to the less reliable and environmentally dependent performance of the stereo measurements. We did make heavy use, however, of CHIMP's visual odometry measurements (using a wide baseline stereo system).

Our software approach utilized human operators and the robot working in concert with one another. Humans made high-level decisions regarding scene understanding and overall task strategies, while the robot took precision measurements using its sensors and executed fine-grained motion control. We did not place heavy emphasis on robot autonomy, instead opting to execute robot motions with guarded autonomy, alerting the user when problems arose. The user performed all high-level decision-making before proceeding.

We placed emphasis on having a highly accurate pose system that could closely track the robot's motion through the environment. This decision was critical, as it allowed robot operators to remotely navigate the robot through cluttered scenes even with extremely low bandwidth, relying upon live but low-bandwidth telemetry while using a previously transmitted map of the environment. Without this map and accurate pose, such operation would be nearly impossible.

Throughout the project, we placed extreme importance on testing. This required operators to be extremely comfortable working with the robot, while also requiring that the robot be robust, with all potential hardware issues understood and dealt with rapidly. We had high confidence that a robot that could withstand days' long continuous testing could survive the DRC Finals.

Lastly, our team management strategy was critical to our success. Our hardware and software engineers worked full-time, side-by-side, under one roof, to get the robot working. Bug tracking was used for all software and many hardware issues, project deadlines were set to ensure systems came together, and code freezes and code reviews ensured that new bugs did not creep into already functional software. Furthermore, we attempted to distribute overall responsibilities amongst team members. For instance, multiple operators were trained for each challenge task, thus ensuring that the unavailability of a single person would not prevent us from completing tasks.

5.3.2. Underutilized Efforts

Due to the immense scope but limited budget, it was extremely difficult to prioritize development for the DARPA Robotics Challenge. At many junctures throughout the project, we reprioritized efforts to better match what we felt would be necessary at the DRC Finals. In this section, we describe multiple efforts on which we halted development and the reasons why.

With CHIMP driving a utility vehicle during the challenge, we often considered applying our significant past experience with vehicle autonomy to the DRC. This was one of the earliest efforts cut from our development timeline, once we understood that a human operator could more easily teleoperate the vehicle, using the always on, low-bandwidth link, rather than integrating vehicle autonomy.

In a similar vein, we invested significant effort for CHIMP to autonomously navigate itself through indoor environments. This interface allowed an operator to direct CHIMP to a goal location, after which the robot handled all path planning, navigation, and tracking to reach the desired location (with LIDAR data used to generate obstacle-free paths). We developed and tested this system and ensured it could operate autonomously despite communication blackouts. After it became clear the DARPA communications system would allow an always-on, low-bandwidth link, we ultimately decided that a human operator, working with accurate telemetry and an occupancy map of the environment, would be able to perform this task sufficiently and more reliably.

We additionally developed software to compute inverse reachability maps of CHIMP performing various tasks. Given a typical manipulation task, this software calculated the best location for CHIMP in order to provide maximum kinematic flexibility. While we developed this software, we ultimately determined that a simpler approach was to provide the human operator with manually placed task guides at body-relative locations. These virtual guides, for instance a valve floating in space but visually attached to the robot, were used by the operator when lining the robot up to perform a task. These guides were heuristically determined on a per task basis and incorporated directly into the user interface.

On the perception side, we developed an easy-to-use generic object grasping framework that determined how best to grasp arbitrary objects. This system was integrated with our planning system and allowed the robot to grasp and manipulate objects with very little operator interaction. In the DRC Finals, however, the debris task placed more emphasis on mobility, and the manipulation tasks dealt with predetermined or structured objects (such as the tools the robots used or the typical valves to be turned). As such, we did not utilize this generic grasping framework at the DRC Finals.

With a statically stable humanoid robot, we placed little emphasis on active balance or fall prevention. Early on, deciding between pursuing fall recovery and fall prevention, our team chose to focus on the former, given that the ability to get back up after a fall could be potentially more critical than a system that prevented falls. This decision was made before DARPA amended the rules allowing teams to conduct interventions to assist fallen robots. Active balance algorithms would have been very helpful to CHIMP, and might have prevented the fall that the robot encountered. Given the short timeline of the project, our team simply did not have enough development cycles to pursue this effort. Our team did have the ability to actively monitor an estimated center of gravity, however we did not place this at high enough importance during the challenge.

In each of these scenarios, we balanced the individual pros and cons of continuing development versus focusing our efforts on other necessary work to be completed. The work our team did complete was sufficient to complete the DARPA Robotics Challenge, however there are ways we could have performed even better.

5.3.3. Areas of Future Improvement

There are several ways in which we could still improve the CHIMP robot. First and foremost, despite having an extremely robust robot with high uptime, hardware failures still occurred on occasion. One area that had repeated failures was the optical motor encoders on CHIMP's joints. These encoders failed from time to time, typically due to particulate debris inside the motor. We feel a small redesign could dramatically improve their reliability.

Similarly, when clutches slipped on individual joints, it was likely that the robot would lose its stored calibration of these joints. We surmise that the high impact that typically causes a clutch to slip also causes link components to shift very slightly. This could likely be fixed with a hardware revision, and further software improvements could make the calibration procedure faster. This failure is what caused our right arm to become less accurate during our Day 1 run.

CHIMP's unique track drives suffered from slipping on dusty surfaces during both vehicle egress and stair climbing. Continued operation in dusty environments would likely require a different belt material. Additionally, the track drive mechanisms rely on mechanical guides to keep the rubber belts on the tracks, but the belts could slip off when in an upright two-limb posture and on high friction environments, such as rough asphalt and concrete. We determined experimentally that this was more likely to occur when executing skid steer turns while moving forward. As such, during the DRC Finals, CHIMP executed turns only while moving backwards in order to keep the lower portion of the belt under higher tension. With track mechanism improvements, the belt could be better guided in place, thus providing better overall robot mobility.

As described earlier, we developed a system that could crudely estimate ground reaction forces using only the joint torques of individual motors along with the kinematic arrangement of CHIMP's limbs. This system was unreliable, however, and would be improved by incorporating strain gauges directly onto the series elastic elements in each of CHIMP's drive joints. Additionally, force-torque sensors connected to the track link itself would provide the best possible measurements of ground reaction forces.

As mentioned in the previous section, we did not develop a balancing controller or a fall prevention system. Despite the static stability design of the CHIMP robot, such a system would allow even greater robustness and reliability, and we will consider its incorporation in the future.

With network failures on Day 2 of the competition, we could have placed greater emphasis on overall network testing while operating CHIMP. Post-run analysis showed that our communication software was not able to scale from the six operator stations we used during testing to the eight stations we used at the finals. This conclusion served as a reminder to refrain from making last-minute changes to the configuration of any tested system, no matter how innocuous they may seem.

Lastly, we feel the DRC provides an excellent blueprint for adding further autonomy to robots such as CHIMP. When operating CHIMP, the human operator still provides critical, often real-time instructions to the robot. By adding more and more low-level autonomy to the robot, such as in our curtailed developments including autonomous navigation, the human will be freed from many of the low-level operation tasks, thus decreasing the overall interaction between CHIMP and its human operator.

6. CONCLUSION

At the DARPA Robotics Challenge, CHIMP demonstrated that a true disaster robot is within reach. By blending autonomy with teleoperation, CHIMP was able to deal with real impediments, such as doors, stairs, debris, and uneven surfaces, while performing real work, such as turning valves, throwing a switch, and operating a drill. The robot incorporated task-based behaviors as well as configuration and joint level control to handle a range of situations from those fully understood to the truly unexpected. The team developed a great set of tools for configuring, controlling, and monitoring a robot from a remote location, even in the face of severely restricted communications.

Throughout this paper, we have highlighted precisely how CHIMP was able to complete the DARPA challenge, in the form of unique approaches to solve each DRC task, overall software improvements made to the system, and the strengthening and hardening of the robot's mechanical design. With a robot capable of completing all of the challenge tasks and, ultimately, able to stand back up after accidentally falling, the CHIMP robot was one of the winners of the DARPA Robotics Challenge, dealing with significant adversity as it went on to complete the challenge.

CHIMP was shown to be general purpose, flexible, and rugged. That said, the robot will require another round of engineering to handle a true disaster environment, such as hardening to radiation, thermal extremes, and water immersion; however, that type of engineering is well understood.

The DARPA Challenge itself has been a great way to galvanize and focus a research community, give it stretch goals that push the state of the art but are still attainable, rally other partners and resources to help with the agenda, and inject the excitement of a competition into the research process.

ACKNOWLEDGMENTS

Development of the CHIMP robot has been supported by DARPA/SPAWAR under Contract No. N65236-12-C-3886. This work would not have been possible without the dedication of the entire Team Tartan Rescue and the National Robotics Engineering Center at Carnegie Mellon University. Additional team sponsors have provided generous support, notably from Foxconn, Amazon, and Carnegie Robotics, with additional support by Accurate Gear and Machine, Brentronics, Eclipse Metal Fabrication, Elmo Motion Control, Faulhaber, Glenair, Google, Harmonic Drive, Honeywell, Kollmorgen, Micromo, Oshkosh/JLG, Pratt & Miller, Robotiq, Sepac, Shell, and THK.

REFERENCES

Berenson, D., Srinivasa, S., Ferguson, D., & Kuffner, J. (2009). Manipulation planning on constraint manifolds. In IEEE International Conference on Robotics and Automation (ICRA '09).

Besl, P. J., & McKay, H. D. (1992). A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2), 239–256.

Fallon, M., Kuindersma, S., Karumanchi, S., Antone, M., Schneider, T., Dai, H., D'Arpino, C. P., Deits, R., DiCicco, M., Fourie, D., et al. (2015). An architecture for online affordance-based perception and whole-body planning. Journal of Field Robotics, 32(2), 229–254.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381–395.

George, M., Tardif, J.-P., & Kelly, A. (2015). Visual and inertial odometry for a disaster recovery humanoid. In Field and Service Robotics, pages 501–514. Springer.

Hebert, P., Bajracharya, M., Ma, J., Hudson, N., Aydemir, A., Reid, J., Bergh, C., Borders, J., Frost, M., Hagman, M., et al. (2015). Mobile manipulation and mobility as manipulation design and algorithms of robosimian. Journal of Field Robotics, 32(2), 255–274.

Johnson, M., Shrewsbury, B., Bertrand, S., Wu, T., Duran, D., Floyd, M., Abeles, P., Stephen, D., Mertins, N., Lesman, A., et al. (2015). Team ihmc's lessons learned from the darpa robotics challenge trials. Journal of Field Robotics, 32(2), 192–208.

Kuffner, J. J., & LaValle, S. M. (2000). Rrt-connect: An efficient approach to single-query path planning. In Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on, volume 2, pages 995–1001 vol.2.

Lim, J., Shim, I., Sim, O., Joe, H., Kim, I., Lee, J., & Oh, J.-H. (2015). Robotic software system for the disaster circumstances: System of team kaist in the darpa robotics challenge finals. In Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on, pages 1161–1166. IEEE.

Nelson, G., Saunders, A., Neville, N., Swilling, B., Bondaryk, J., Billings, D., Lee, C., Playter, R., & Raibert, M. (2012). Petman: A humanoid robot for testing chemical protective clothing. Journal of the Robotics Society of Japan, 30(4), 372–377.

Pitzer, B., Styer, M., Bersch, C., DuHadway, C., & Becker, J. (2011). Towards perceptual shared autonomy for robotic mobile manipulation. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 6245–6251.

Ruehl, S. W., Hermann, A., Xue, Z., Kerscher, T., & Dillmann, R. (2011). Graspability: A description of work surfaces for planning of robot manipulation sequences. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 496–502.

Stentz, A. T., Herman, H., Kelly, A., Meyhofer, E., Haynes, G. C., Stager, D., Zajac, B., Bagnell, J. A. D., Brindza, J., Dellin, C., George, M., Gonzalez-Mora, J., Hyde, S., Jones, M., Laverne, M., Likhachev, M., Lister, L., Powers, M. D., Ramos, O., Ray, J., Rice, D. P., Scheifflee, J., Sidki, R., Srinivasa, S., Strabala, K., Tardif, J. P., Valois, J.-S., Vandeweghe, J. M., Wagner, M. D., & Wellington, C. (2015). Chimp, the cmu highly intelligent mobile platform. Journal of Field Robotics (JFR), Special Issue: Special issue on DARPA Robotics Challenge (DRC), 32(2), 209–228.

Şucan, I. A., Moll, M., & Kavraki, L. E. (2012). The Open Motion Planning Library. IEEE Robotics & Automation Magazine, 19(4), 72–82. http://ompl.kavrakilab.org.

Urata, J., Nakanishi, Y., Okada, K., & Inaba, M. (2010). Design of high torque and high speed leg module for high power humanoid. In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pages 4497–4502. IEEE.