Team VALOR's ESCHER: A Novel Electromechanical Biped for the DARPA Robotics Challenge*

Coleman Knabe, Robert Griffin, James Burton, and Graham Cantor-Cooke
Virginia Tech, Blacksburg, Virginia 24061
e-mail: knabe@vt.edu, robert.griffin@vt.edu, james.burton@vt.edu, grahamcc@vt.edu
Lakshitha Dantanarayana
University of Technology, Sydney (UTS), Ultimo NSW 2007, Australia
e-mail: lakshitha.dantanarayana@uts.edu.au
Graham Day, Oliver Ebeling-Koning, Eric Hahn, Michael Hopkins, Jordan Neal, Jackson Newton, Chris Nogales, Viktor Orekhov, John Peterson, Michael Rouleau, John Seminatore, Yoonchang Sung, Jacob Webb, Nikolaus Wittenstein, Jason Ziglar, Alexander Leonessa, Brian Lattimer, and Tomonari Furukawa
Virginia Tech, Blacksburg, Virginia 24061
e-mail: gad266@vt.edu, oliverek12@gmail.com, erichahn@vt.edu, hopkns@vt.edu, jdneal@vt.edu, newtonj@vt.edu, nogalesc@vt.edu, vlorekhov22@gmail.com, jrpeter@vt.edu, mrouleau@vt.edu, johnsemi@vt.edu, yooncs8@vt.edu, jdwebb@vt.edu, nikolaus@vt.edu, jpz@vt.edu, leonessa@vt.edu, blattimer@jensenhughes.com, tomonari@vt.edu

Received 8 October 2015; accepted 19 August 2016

The Electric Series Compliant Humanoid for Emergency Response (ESCHER) platform represents the culmination of four years of development at Virginia Tech to produce a full-sized force-controlled humanoid robot capable of operating in unstructured environments. ESCHER's locomotion capability was demonstrated at the DARPA Robotics Challenge (DRC) Finals when it successfully navigated the 61 m loose dirt course. Team VALOR, a Track A team, developed ESCHER leveraging and improving upon bipedal humanoid technologies implemented in previous research efforts, specifically for traversing uneven terrain and sustained untethered operation. This paper presents the hardware platform, software, and control systems developed to field ESCHER at the DRC Finals. ESCHER's unique features include custom linear series elastic actuators in both single and dual actuator configurations and a whole-body control framework supporting compliant locomotion across variable and shifting terrain. A high-level software system designed using the robot operating system integrated various open-source packages and interfaced with the existing whole-body motion controller. The paper discusses a detailed analysis of challenges encountered during the competition, along with lessons learned that are critical for transitioning research contributions to a fielded robot. Empirical data collected before, during, and after the DRC Finals validate ESCHER's performance in fielded environments.

1. INTRODUCTION

Robots that are capable of serving as first responders will allow for faster, safer, and more capable service in the aftermath of disasters. The DARPA Robotics Challenge (DRC) pushed robots to leave the security of the lab, taking on new roles in challenging, unpredictable disaster response environments. Team VALOR (Virginia Tech Advanced Legged Operations Robots) was selected as a Track A competitor for the DRC, requiring the design and fabrication of a hardware platform, and the development of software and control systems for the DRC. As many disaster scenarios take place in manmade environments, bipedal humanoid robots provide a means of navigating such environments while using tools originally designed for human responders (Kajita, Hirukawa, Harada, & Yokoi, 2014). Although traditional wheeled platforms exhibit advantages over bipeds, such as inherent stability, high payload capacity, and ease of control and state estimation, bipeds offer a unique potential for mobility and flexibility in a variety of environments. The DRC effort focused on developing the key enabling technologies to field a robot capable of robust bipedal locomotion and manipulation in unstructured environments, while supporting varying levels of autonomous operation.

*This manuscript was originally submitted for the special issue on the DARPA Robotics Challenge Finals.
This work was performed while with the Terrestrial Robotics Engineering, and Controls (TREC) Lab at Virginia Tech
This material is based upon work supported by (while serving at) the National Science Foundation.
Currently at Jensen Hughes, Blacksburg, VA.
Direct correspondence to Tomonari Furukawa: tomonari@vt.edu

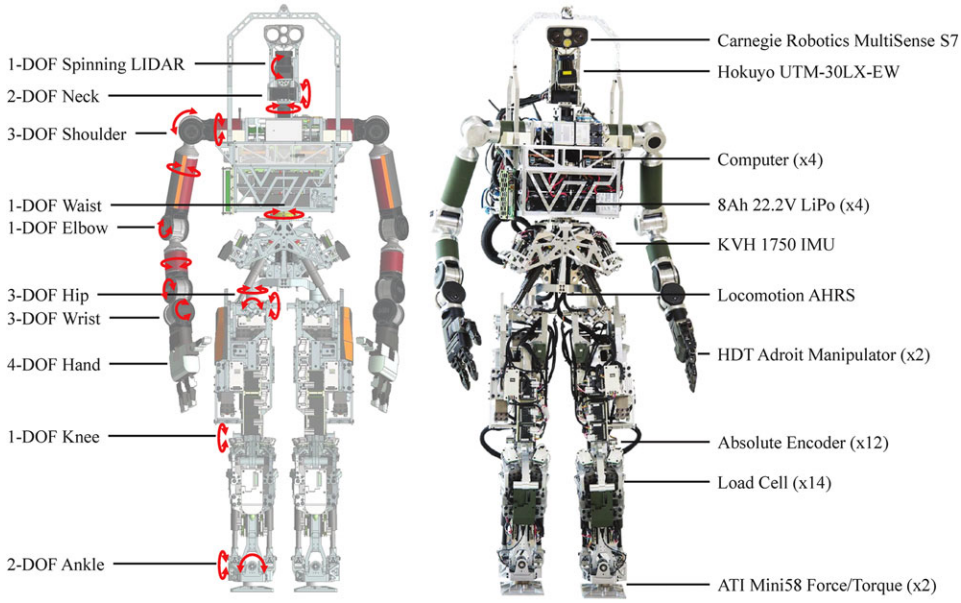


Figure 1. Left: overview of ESCHER's degrees of freedom. Right: hardware components of ESCHER.

Researchers on the team had been involved in developing full-sized humanoid robots to assist in fighting fires aboard ships as part of the Office of Naval Research (ONR) Shipboard Autonomous Fire Fighting Robot (SAFFiR) project. This project spurred the development the Tactical Hazardous Operations Robot (THOR) (Lee, 2014). Although this robot fulfilled the initial requirements of the SAFFiR project, it did not meet locomotion, onboard computing, and battery life requirements of the DRC, prompting the development of the Electric Series Compliant Humanoid for Emergency Response (ESCHER) platform.

Humanoid research has traditionally focused on designing robots, such as ASIMO (Hirai, Hirose, Haikawa, & Takenaka, 1998) and HRP-2 (Kaneko et al., 2002), with rigid joints for accurate position control. However, as researchers try to mimic the adaptability and fluidity of natural motions, research on compliant robots is becoming increasingly common (Englsberger et al., 2014b; Lahr, Orekhov, Lee, & Hong, 2013; Pratt & Krupp, 2008; Tsagarakis, Morfey, Cerda, Zhibin, & Caldwell, 2013). Utilizing low-impedance actuators improves the ability to adapt quickly to external uncertainties and disturbances over high mechanical impedance position controlled actuators (Stephens & Atkeson, 2010; Tsagarakis et al., 2013) while also decreasing risk to the environment (Pratt & Williamson, 1995). Series elastic actuators (SEAs) represent an effective means of realizing compliant, force controllable actuation by introducing an elastic element inline with the transmission (Pratt & Williamson, 1995). While hydraulic actuation offers high bandwidth and power density, it is typically much heavier than its electric counterpart (Pratt & Krupp, 2004) and it has high output impedance. To achieve low-impedance force controllable actuation, the custom electric linear SEAs presented in Knabe, Lee, Orekhov, & Hong (2014b) were developed and implemented on the THOR for the SAFFiR program. To enable the multicontact behaviors required for compliant locomotion and manipulation on THOR, the novel whole-body control framework presented in Hopkins, Hong, & Leonessa (2015b) was developed, inspired by de Lasa & Hertzmann (2009), Feng et al. (2015), Herzog et al. (2014), Kuindersma et al. (2014), and Saab et al. (2013). This quadratic program (QP) -based optimizer was designed to track the unstable divergent component of motion (DCM) of the center of mass (CoM) (Englsberger, Ott, & Albu-Schaffer, 2013; Hopkins, Hong, & Leonessa, 2014) for balance by computing joint torques that minimize tracking errors for multiple objectives, such as pelvis angular acceleration and swing foot acceleration, simultaneously. A platform capable of compliant locomotion was realized through the design of a compliant bipedal humanoid utilizing SEAs and the development of the controls framework outlined in Hopkins et al. (2015b).

In addition to improved locomotion, ESCHER, displayed in Figure 1, incorporates a new software system. Previous challenges in developing and integrating higher-level software on THOR led to using the robot operating system (ROS) to leverage open-source software and integrate with the existing whole-body control framework. This allowed Team VALOR's software team to collaborate with Team ViGIR for perception, human-robot interaction, and path and manipulation planning. This use of open-source packages enabled Team VALOR to deploy unique research

contributions while rapidly building up key subsystems. When incorporated onto ESCHER, a robot capable of both compliant locomotion and semiautonomous behaviors was realized, resulting in a platform capable of walking and manipulation robust enough for the DRC Finals. ESCHER is one of only four custom bipedal humanoids designed for the DRC Finals.

This paper describes the approach taken throughout the development of ESCHER for the DRC Finals by overviewing the design of the mechanical, electrical, software, and control systems. Insights gained from the design and fielding of a compliant bipedal humanoid are provided, including methods for addressing challenges encountered when implementing model-based whole-body control on hardware. ESCHER's capabilities as a fieldable robot are demonstrated through testing and practice results leading up to and following the DRC Finals. Empirical data on the platform's performance at the DRC Finals are provided along with lessons learned through the design, testing, and fielding of the robot.

2. ESCHER PLATFORM ARCHITECTURE

ESCHER is a fully electric, torque-controlled humanoid standing 1.78 m tall and weighing 77.5 kg with 38 degrees of freedom (DOF) as illustrated in Figure 1 (left). The body consists of a lightweight aluminum alloy frame with locomotive power provided by custom linear SEAs. A whole-body motion framework enables walking across uneven and shifting terrain by resolving multiple motion tasks using the optimization-based formulation presented in Section 4. Integrating HDT's ruggedized Adroit manipulators improved payload capacity, added force-sensing capabilities, and improved reliability. The perceptive and proprioceptive sensors, computers, and batteries depicted in Figure 1 (right) allow semiautonomous operation with over 2 h of runtime. This section describes ESCHER's hardware design, beginning with the series elastic actuators in the lower body, the design of the lower and upper bodies, the selected sensors and computers, and finally concluding with the power and safety systems.

2.1. Series Elastic Actuator

Each 6-DOF leg on ESCHER is driven by seven custom electric linear SEAs, depicted in Figure 2 (Knabe et al., 2014b). Three variations of SEAs are used on ESCHER, each composed of modular subassemblies to reduce the burden of redesign, maintenance, or replacement in the event of damage. The transmission consists of a low-friction ball screw driven by a brushless direct current (BLDC) motor to satisfy efficiency, power density, and load capacity requirements, similar to many other linear SEAs (Lee, Knabe, Orekhov, & Hong, 2014; Paine, Oh, & Sentis, 2014; Paluska & Herr, 2006; Pratt & Krupp, 2004; Robinson et al., 1999). A tension/compression load cell mounted inline with each actuator directly measures actuator force to a peak load of 2,225 N. The actuator lacks a linear guide to reduce weight and friction; instead, universal joints at each end of the actuator constrain it as a two-force member and provide the necessary degrees of freedom to allow parallel actuation of robotic joints.

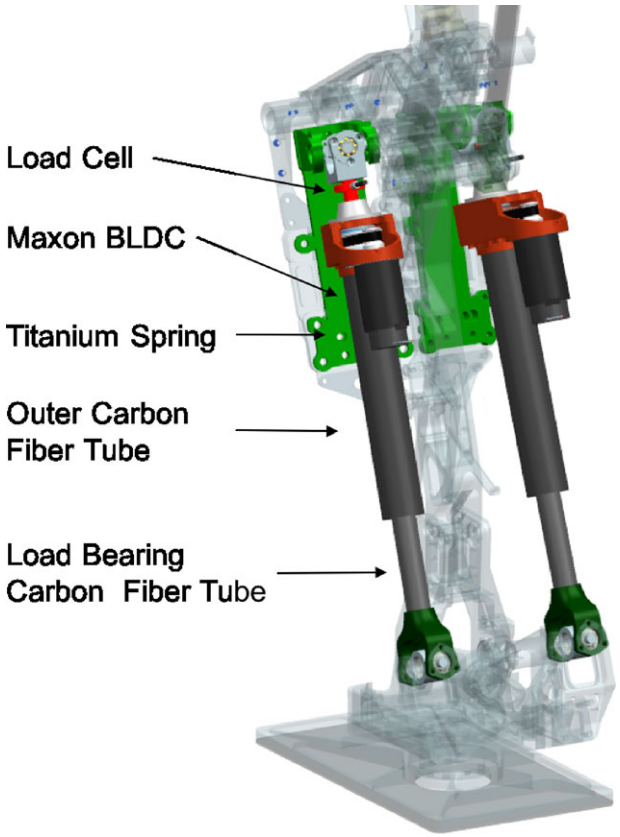


Figure 2. Labeled schematic of SEA used in the right ankle pitch/roll joint.

Unlike many linear SEAs that use one or more compression die springs with relatively linear spring rates (Edsinger-Gonzales & Weber, 2004; Paine, Oh, & Sentis, 2014; Pratt, Krupp, & Morse, 2002; Sensinger et al., 2006), the elastic element in this design is a cantilevered titanium leaf spring mounted parallel to the actuator. This positions the region required for spring deflection outside the travel axis of the actuator, resulting in a smaller effective packaging volume. A lever arm connects the actuator to load the beam in moment, allowing larger actuator forces prior to yield than a similarly sized beam loaded in bending. A removable pivot allows for configurable compliance, similar to that described in Orekhov, Lahr, Lee, & Hong (2013), with two selectable spring rates: 372 or 655 kN/m. However, every actuator on ESCHER utilized the stiffer (655 kN/m) of

the two compliance settings, since the higher spring rate increases force bandwidth.

A custom dual-axis motor controller handles low-level joint level impedance control of the linear SEAs, including parallel actuated joints on ESCHERs legs (Hopkins, Ressler, Lahr, Hong, & Leonessa, 2015c; Ressler, 2014). These motor controllers allow for custom algorithms to be quickly implemented, and they are more compact than commercially available options. They communicate at 1 Mbps using the CANopen protocol, a well-known industrial automation and control standard. Configuration values and joint-space set points are transmitted at a rate of 500 Hz, while joint-space estimates are transmitted back using a synchronous read-write scheme.

In parallel to SEA development, a new model of ball screw driven SEAs was derived that decouples the translational, or sprung, mass from the rotary inertia of the motor and drivetrain, described in an intuitive rack and pinion representation (Orekhov, Knabe, Hopkins, & Hong, 2015). This model more accurately describes the actuator dynamics when driving a moving output, and it has shown that robust force control can be achieved regardless of the location of the elastic element. This enabled more flexibility when designing ESCHER by allowing positioning of the elastic element between the motor and chassis ground, rather than between the motor and load.

2.2. Lower Body Design

The lower body of ESCHER is a substantial design improvement upon the THOR leg architecture (Lee, 2014), with the most significant change occurring in the thigh. The THOR leg design focused on achieving a humanlike range of motion utilizing inverted Hoeken's straight line linkages to convert linear motion from the SEAs to rotary motion at the hip and knee pitch joints (Knabe, Lee, & Hong, 2014a). This configuration delivered a peak torque of 115 N m, i.e., there was ample overhead for locomotion on relatively flat terrain, but it was insufficient for the stair and rubble tasks at the DRC Finals.

ESCHER's thigh design utilizes higher power versions of the SEAs used on THOR, addressing the need for higher peak torques while reducing development costs. Torque requirements were generated using the Gazebo physics simulator (Koenig & Howard, 2004) by traversing a 0.23 m block with an 80 kg mass-augmented model of THOR, the design set point for ESCHER. Figure 4 contains plots of the torque requirements overlaid with peak joint torques available on THOR and ESCHER for the required joint angles to complete the motion. Experimental validation of the redesign demonstrates that the hip and knee pitch torque requirements enable the locomotive capabilities required for the DRC (Knabe et al., 2015).

ESCHER features SEAs driving the leg joints arranged in two configurations: parallel actuation, in which a pair of SEAs collaboratively drive two orthogonal DOFs, and serial actuation, where one or more SEAs drive a single rotary joint through a crank arm. These configurations are shown in Figure 3. The new hip and knee pitch joints are serially actuated through 0.075 m crank arms configured such that the peak mechanical advantage occurs at joint angles corresponding to the peak demanded torques found in Figure 4. The knee joint utilizes two identical linear SEAs in a parallel configuration to power the single DOF, doubling available joint torque without requiring modification of the fundamental actuator design. The 2-DOF hip and ankle joints rely on a parallel actuation arrangement to decrease limb inertias and increase maximum joint torques for individual joints, thereby reducing ESCHER's peak power requirements. These actuation schemes, coupled with an intelligent structural design, give ESCHER an impressive range of motion (ROM) in the lower body, as shown in

Figure 3. Rear view of ESCHER's right leg showing the placement of the linear SEAs.

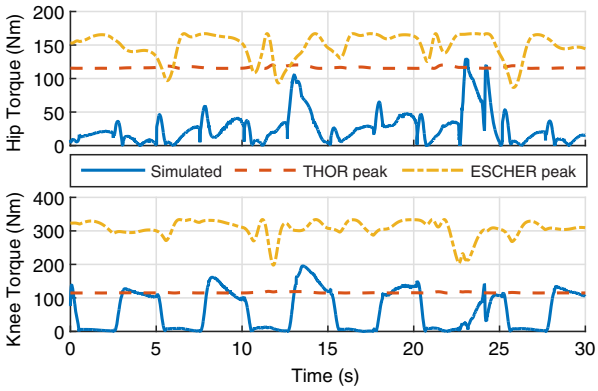


Figure 4. Comparison of THOR and ESCHER's peak torques to requirements from the 80 kg simulation model.

Table I. Range of motion limits in ESCHER's left leg.

Joint axis	Min. angle (deg)	Max. angle (deg)
Hip Yaw	−20	45
Hip Roll	−30	45
Hip Pitch	−115	15
Knee Pitch	0	130
Ankle Pitch	−55	35
Ankle Roll	−30	30

Table I. Closed-cell PVC foam board and molded polystyrene composite covers protect critical sensors and likely points of impact, providing a lightweight means of reducing shock transmission to the robot frame in the event of a fall. Lee and Knabe et al. detail the design, analysis, and torque profiles of THOR and ESCHER in Lee (2014) and Knabe et al. (2015), respectively.

Accurate measurement of the robot's state requires properly biased joint encoders. At the expense of additional power, communications, and design complexity, platforms incorporating limit switches can automate this biasing procedure. However, experience with THOR and previous legged robots indicate this is typically an intensive manual process requiring accurate positioning of joints to a set zero pose. To improve this process, the ESCHER leg frame contains mounting locations for biasing jigs which, when installed, properly align and distance the thighs, shins, and feet. This constrains the yaw and roll DOFs to their respective zero pose, thereby reducing the 12-DOF legs to a 3-DOF system in which only the collective hip, knee, and ankle pitch joints require manual positioning. The use of biasing jigs ensures that the legs are symmetrically biased, and reduces the average biasing time from 1.5 h to under 45 min.

2.3. Upper Body Design

The THOR arms consisted of commercial off-the-shelf ROBOTIS Dynamixel Pro motors arranged in a 7-DOF configuration and outfitted with custom 2-DOF underactuated grippers (Rouleau & Hong, 2014), resulting in a 2 m arm span. Each 6.6 kg arm possessed a limited payload capacity of approximately 3 kg (Robotnik, 2015), lacked direct joint torque sensing, and exhibited reliability issues. Furthermore, a redundant yaw-roll-yaw wrist configuration increased the possibility of gimbal lock, introducing challenges to manipulation planning. Historical testing informed the decision to equip ESCHER with adroit manipulator arms made by HDT Global with a 2.4 m arm span, more kinematically advantageous yaw-pitch-roll wrist, support for joint impedance control, reliable through-actuator wiring offering continuous rotation, and an improved payload capacity of 13.6 kg (HDT-Global, 2015). Each 8.3 kg, 7-DOF arm connects to a 4-DOF manipulator featuring an opposable thumb and force sensing to determine grip. Figure 5 shows the results of a reachability study conducted using OpenRAVE (Diankov & Kuffner, 2008) to compare the Dynamixel and HDT arms, highlighting the advantages of the longer arm links and improved wrist configuration. This improved workspace enables more flexibility in the positioning of the robot and enables larger motions to be executed for manipulation tasks.

The chest houses the computation and power suites required for untethered operation, as discussed in Sections 2.5 and 2.6, respectively, including sufficient overhead to expand capabilities for future research. Removable panels on the front and back allow access for rapid battery changes

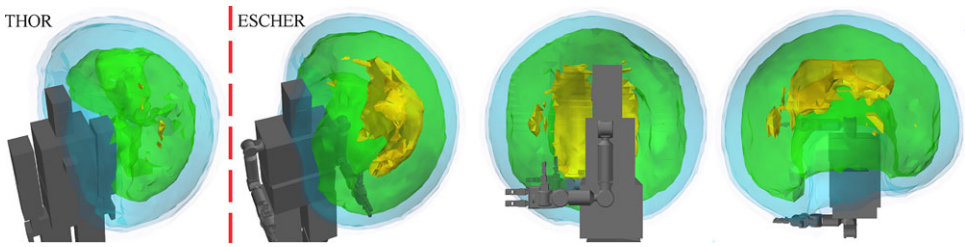


Figure 5. Reachability regions of Dynamixel (far left) and HDT (left center, right center, and far right) arms. Warmer colors represent a higher number of kinematic solutions to each desired target pose (Wittenstein, 2015).

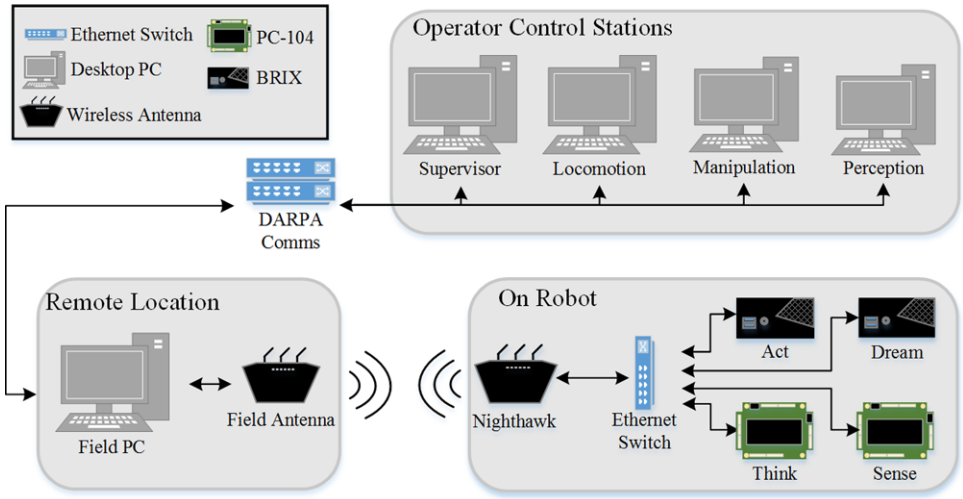


Figure 6. Team VALOR's computer architecture at the DRC Finals. Four interconnected computers on the robot are connected by a wireless link to a field computer that interfaces with the OCSs through DRAPA's degraded communications network.

and decreased maintenance time. A roll cage frame arches over the head to protect the perception sensors in the event of a fall, and it also serves as the primary means of attachment to an overhead gantry system through a quick release pin joint.

2.4. Sensors

In environments with restricted mobility, it is often necessary to articulate sensor packages to enable observation with adjustment of a robot's orientation. In addition, humanoid robots need the versatility to interact with objects and the environment that may not be within the field of view of sensors fixed to the robot. Realizing the need for an articulated sensor package, ESCHER's perception sensors are packaged in a 2-DOF head. A Carnegie Robotics MultiSense S7 stereo camera provides high-resolution wide-field-of-view color imagery and stereoscopic depth images to operators. A rolling Hokuyo UTM-30LX-EW LIDAR generates accurate three-dimensional (3D) point clouds of the environment, providing higher sampling density close to the axis of rotation and a larger field of view than the stereo camera. The rolling LIDAR located on the head enables the robot to *see* over its shoulder, improving the robot's situational awareness.

The lower body features a variety of proprioceptive sensors, providing the required state feedback needed for compliant whole-body locomotion. Gurley A19 absolute encoders at each degree of freedom measure joint position. Futek LCM-200 tension/compression load cells directly measure forces within each SEA. Incremental actuator encoders on each Maxon BLDC measure motor angle prior to the large gear reduction of the ball screw transmission, providing a high-resolution estimate of actuator and joint velocities. An ATI Mini-58 six-axis force/torque transducer in each foot measures ground contact and reaction forces. A MicroStrain 3DM-GX3-25 attitude and heading reference system (AHRS) in the pelvis provides dead reckoning pose estimation for the locomotion framework. However, it introduced too much noise to integrate exteroceptive data during motion, so a KVH 1750 Fiber Optic Gyro Inertial Measurement Unit (IMU) was incorporated into the pelvis.

2.5. Onboard Computation and Network Architecture

ESCHER's chest houses two Gigabyte Brix computers and two ADLQM87PCs, referred to as *Brix* and *PC-104*, respectively. Each Brix contains a quad-core i7 processor nominally operating at 3.2 GHz for high single threaded performance, while each ADLQM87PC contains a quad-core i7 processor operating nominally at 2.4 GHz for improved power efficiency, and an additional Ethernet adapter to connect directly to the Multisense S7. These computers strike a balance between performance, power consumption, and I/O peripheral availability, providing the required hardware interfaces for communicating with onboard sensors and actuators while also providing enough processing power to minimize any requirements to heavily optimize software during initial development. The onboard computers communicate through a Gigabit Ethernet switch, as shown in Figure 6.

The competition allowed teams to deploy a field computer anticipating advances in processing power. The onboard computers communicate with the remote field computer over a wireless bridge using a DARPA-furnished router. The field computer then communicates with the operator control station (OCS) computers over DARPA's

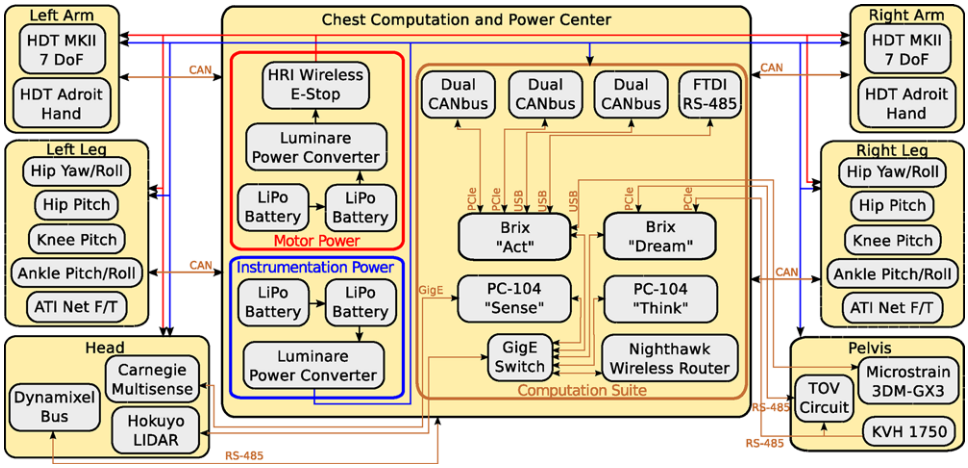


Figure 7. ESCHER power and communication bus structure. ESCHER's actuators and proprioceptive sensors communicate over CAN and RS-485, while its exteroceptive sensors communicate over Ethernet.

degraded communications network. In ideal conditions, all computers in the system are connected via a single, unified network; however, under the degraded communications present at the Finals, the computers were split into two distinct networks separated by the DARPA communications link described in Section 3.8.

2.6. Power, Communication, and Safety Systems

Figure 7 summarizes the power and communication protocols for ESCHER's sensors, actuators, and computers. Sensors in the head and pelvis communicate over the Ethernet with the onboard computers. Each limb of the robot communicates with the onboard computers over separate CAN channels. The power system is split into two separate power buses, one for instrumentation and the other for the actuators.

Commercial-grade lithium polymer (LiPo) batteries provide power to the all-electric ESCHER platform, offering high-energy density at a relatively low cost. Batteries were chosen to meet worst-case estimates based on peak sensor, computer, and motor power draw, while satisfying a minimum factor of safety of 2. The resulting system was to operate with four 25.9 V MaxAmps battery packs, providing a total nominal energy capacity of 2,279 Wh. This large capacity enables extended operation and allows for future expansion through sensor additions or computer upgrades. For testing purposes, the team utilized a set of four smaller 22.2 MaxAmp batteries that provided a nominal energy capacity of 710 Wh. The large and small battery sets weighed 11.2 and 4.7 kg, respectively. To simplify testing logistics, ESCHER can also run off external power supplies.

Two pairs of batteries wired in series provide 48 V power buses to motors and instrumentation in order to distribute the wide variety of supply voltages for onboard electrical components. This split-series architecture, shown in Figure 7, reduces noise coupling between motors and more sensitive electronics, as well as reducing resistive losses in high-power components. Lower body motors run directly off a 48 V bus, while the arms receive power from a 24 V step downconversion from the same bus. A stack of high-efficiency, low-noise, commercial-grade buck converters converts the second 48 V bus to 15, 12, and 5 V buses to power onboard sensors and computers.

A Humanistic Robotics control unit integrated in the chest receives telemetry from a wireless e-stop. The e-stop controller drives doubly redundant relays in series with the 48 and 24 V motor buses, ensuring rapid deenergizing of all onboard actuators. An LED strip attached to the outer rim of the head roll cage frame visually conveys the current state of the motor power system to indicate when the robot is safe to approach. In addition to the indicator strip, each motor controller displays an internal state through an LED bank, reporting idle, active, or fault conditions at a glance.

3. SOFTWARE ARCHITECTURE

Addressing the entire span of capabilities required for the DRC Finals in a single, integrated system involved tackling a wide array of technical topics. When looking to the challenges involved in the software system, the initial approach for software development leading up to the DRC Trials could be described by several overarching concepts: a *bottom-up* development approach, focusing on *evolving capability* with *proprietary* software using a *decentralized* team structure. This approach began by decomposing proposed capabilities into six key areas: motion, perception, path planning, behaviors, communication, and human-robot interaction. Subteams assigned to these areas developed in parallel, rapidly iterating on subsystems to build up

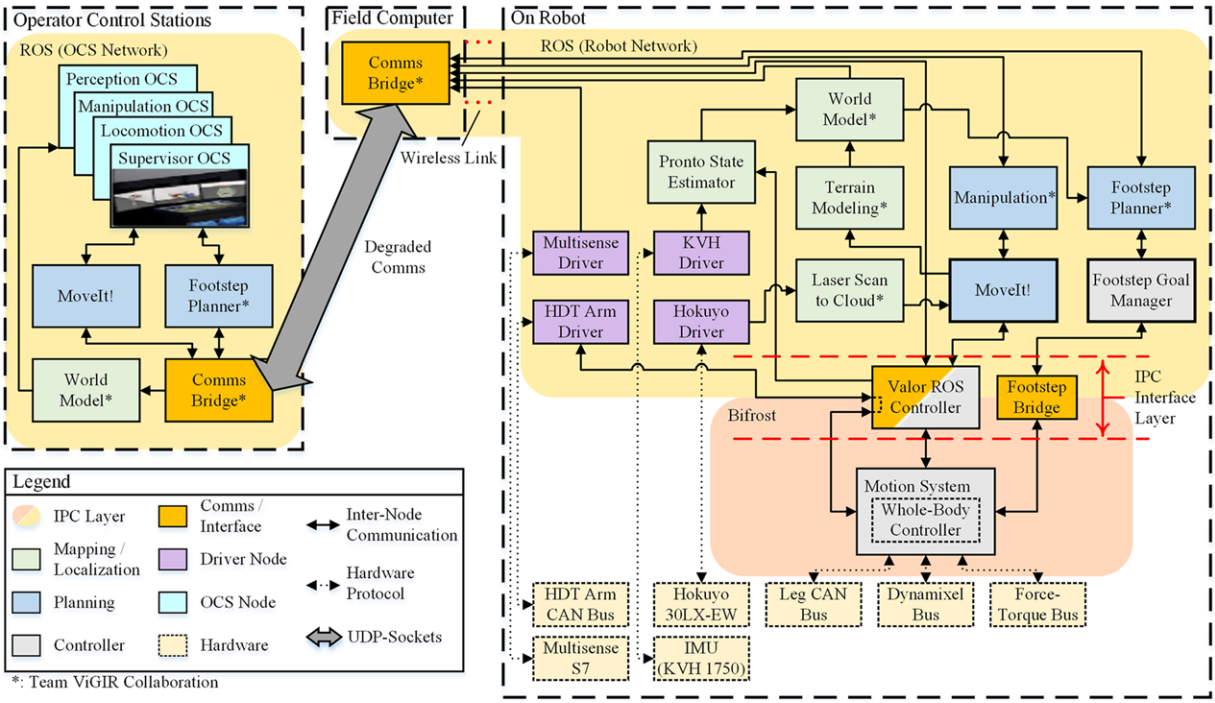


Figure 8. Overall software design for Team VALOR. The VALOR ROS Controller acts as a bridge between the motion system, which runs the whole body controller and actuator interfaces with higher-level onboard components of the system including the Footstep Planner and MoveIt!. The onboard planning components are duplicated on the OCS side of the Comms bridge, enabling plans to be previewed there by operators and sent to the robot.

capability. Developing software in a custom framework enabled fine-grained control of how to address challenges, and built up in-team expertise in key areas. While this approach used to develop software for THOR resulted in a mature, well-tested motion subsystem demonstrating high-fidelity force control (Hopkins et al., 2015c) and robust walking and balancing (Hopkins et al., 2015b), other components struggled to integrate and test in a timely fashion. Delaying integration of initial path planning and perception systems in order to extend decentralized development time compounded the effort required to eventually integrate. Revisiting the software approach post-Trials served to accelerate development and ensure systemic capability.

For the DRC Finals, the software system targeted different concepts for development: a *hybrid* development approach, using open-source software to *implement required functionality*, allowing researchers to identify and focus on *key enabling technologies*. Software development used a bottom-up approach for developing novel subsystems, while top-down design provided a system-wide view to ensure a cohesive method for addressing system requirements (e.g., ensuring safe state transitions during operation and providing localization sufficient for world modeling). To handle the aggressive timeline of developing and integrating the necessary functionality, efforts shifted to provide base-level capability through integration of open-source software. This not only provided a path for rapidly introducing functionality, but it also provided a baseline to compare novel research and identify areas on which to focus future efforts. This approach relied on collaboration with Team ViGIR, a Track B entry in the DRC headquartered local to Team VALOR, who developed software with the explicit goal of open-sourcing the results (Kohlbrecher et al., 2015). Software packages shared between the teams benefited from additional testing with differing hardware configurations, with more manpower available for improving reliability and developing novel algorithms.

Starting from scratch with a rudimentary software infrastructure, and through integration of open-source software, Team VALOR built a full system capable of performing DRC tasks. The overall software architecture implemented for the Finals is shown in Figure 8. The majority of software running on ESCHER uses ROS Indigo, a popular open-source middleware system for robotics (Quigley et al., 2009). Previously developed motion and controls software remained in Bifrost, a custom interprocess communication (IPC) framework developed for the Trials, and a bridge created to provide an interface to the higher-level system.

3.1. Motion System

As software developed for the SAFFiR program evolved to meet the requirements of the DRC Trials, the framework shifted from a core shared-memory approach described in McGill, Brindza, Yi, & Lee (2010) to a message passing IPC known as Bifrost. Bifrost's design offers integrated bandwidth management, support for unreliable network links, and handles scaling system complexity. However, it requires custom implementations to utilize the Bifrost framework, which limits the ability to integrate with already developed systems.

The Motion System is a set of programs implemented in Bifrost that acts as ESCHER's hardware interface and controls balancing and locomotion. The Motion System implements a whole-body controller that is managed by a finite-state machine that assigns task-level weights to the controller and makes available task-specific interfaces to higher-level systems. The whole-body controller optimizes actuator set points based on proprioceptive data and dynamic models of the robot to maintain balance while standing or executing motions. The theoretical basis of this whole-body controller will be discussed in Section 4. In addition to controlling actuator set points, the Motion System also plans leg trajectories given a sequence of footholds. The Motion System on ESCHER implements the critical balancing and bipedal locomotion capabilities necessary to operate the robot.

3.2. Motion Interface

Leveraging open-source software to achieve full systemic capability introduced an additional layer of complexity when interfacing with the Motion System. The ROS offers a large body of software applicable to the DRC, but it could not be directly integrated into the Bifrost framework. By this point, system testing validated walking performance, and porting the system to a new language and framework incurred major technical risks. To minimize risk and meet performance requirements, it was decided that implementing a controller in the ROS framework to serve as a bridge to Bifrost offered the most direct path for integrating the Motion System. This controller, known as the VALOR ROS Controller (Figure 8), implements the ROS interface expected by higher-level planning, perception, and user interface software by publishing appropriate messages in Bifrost, while simultaneously converting feedback from motion into standard ROS messages (Burton, 2016). ROS Controllers within the VALOR ROS Controller interpolate commanded trajectories to spool out a smooth series of set points for the Motion System. The controller also handles the mapping between high-level operational behaviors (e.g., manipulation while balancing) and low-level Motion System states. In addition to the VALOR ROS Controller, the Footstep Bridge, also shown in Figure 8, provides state management and translation of footstep messages. While the VALOR ROS Controller introduced an additional layer of indirection and control parameters for tuning, it successfully avoided larger efforts in porting well-tested motion software to a new framework and language.

3.3. State Estimation

Reliable and accurate state estimation is at the foundation of ESCHER's operation at every level of the system. Without it, ESCHER cannot maintain balance, walk to a desired location, integrate sensor data into a cohesive model, or plan actions to interact with the environment. Given imperfect state estimation, different subsystems place conflicting requirements on how state estimation should behave; for instance, low-level controls favor smooth low-latency estimates, while accumulating perception data into a single model requires accurate low-drift estimates.

The Motion System applies Kalman filters to joint level absolute and incremental encoders to estimate joint positions and velocities. The pose of the floating base frame is estimated through forward kinematics using the estimated joint positions and the Microstrain AHRS. Updating the joint estimates in lockstep with the whole-body controller simplifies reasoning about relative timing between state estimates and controller updates, further improving efficiency and operation. This state estimator provides low-latency state estimates required for balancing and tracking task-space objectives.

Empirical testing indicated that the state estimate produced by the Motion System included too much drift for sufficiently detailed mapping of the environment. Several approaches for supplying localization for higher-level systems were investigated, ranging from developing a visual simultaneous localization and mapping (SLAM) system based on Dellaert (2012), to using open-source SLAM systems (Grisetti, Stachniss, & Burgard, 2005; Zhang & Singh, 2014), settling on Pronto (Bry, Bachrach, & Roy, 2012; Fallon, Antone, Roy, & Teller, 2014) based on empirical performance with the rest of the system. Pronto provides explicit modeling of a legged odometry model, integration of a high-performance IMU, and it supports both visual odometry and LIDAR-based localization. The state estimate provided by Pronto supports perception, planning, and visualization for the OCS.

3.4. Perception

ESCHER's primary perception task focused on ensuring safe execution of walking and manipulation tasks in a remote environment. Toward that end, perception efforts concentrated on three main objectives: obstacle detection, terrain modeling, and supplying relevant data to human operators. Obstacle detection started with the Octomap (Hornung, Wurm, Bennewitz, Stachniss, & Burgard, 2013) integrated into the MoveIt! manipulation planning

package (Sucan & Chitta, 2012), which uses the head LIDAR to generate a 3D occupancy voxel space indicating binary obstacles. The voxel space is sampled in two height ranges to generate 2D binary obstacle maps, with detected obstacles in the ankle to knee height recorded in the lower map, and binary obstacles in the knee to shoulder height in the upper map. These maps represent obstacles that can be stepped over and navigated around during footstep planning, respectively. Terrain modeling is performed using the approach described in Stumpf, Kohlbrecher, Conner, & von Stryk (2014), which estimates terrain as a height map with normal estimates for the support surfaces in the environment to be used in path planning. Remote operators viewed 2D obstacle maps, camera images from the Multisense, and point clouds to understand the environment surrounding the robot.

3.5. Locomotion Planning

In open environments with level terrain and few obstacles, the primary walking challenge comes from generating footstep plans to travel in a desired direction while staying within the kinematic constraints of the robot. By deferring obstacle avoidance to the operator and assuming a locally level ground plane, the footstep planning problem reduces to determining footholds (i.e., x-y position and orientation) in a 2D plane, which head toward the desired goal. A planner known as the footstep pattern generator provides parametrized walking patterns representing a set of walking primitives (e.g., walk forward, sidestep left, turn left) that expand into a sequence of footsteps. The parameters provide some level of control over the primitive expansion, such as the number of steps, distance traveled per step, and speed of each step.

Once the assumptions of a simple world model are violated, safe operation of ESCHER requires more rigorous footstep planning. For activities such as traversing rough terrain, stair climbing, or navigating in cluttered environments, an extended version of the Anytime Repairing A* (ARA*) algorithm generates either 2D or 3D footstep plans (Stumpf et al., 2014). This planner conforms footsteps to rough terrain, avoids obstacles, and estimates risk for individual footsteps, which allows more autonomous motion planning.

3.6. Manipulation Planning

To manage midlevel manipulation planning, the system integrates MoveIt!, a ROS based open-source manipulation planning stack. MoveIt! handles planning obstacle-free trajectories in static environments through the integration of inverse kinematic solvers, 3D perception data, and high-level controllers. In conjunction with its collision avoidance determination, MoveIt! also acts as the self-filter for ESCHER, filtering out the robot from the LIDAR point clouds.

While MoveIt! works well for placing end-effectors at arbitrary goal poses in static environments, it does not handle more complex actions that require sequences of plans. Additionally, MoveIt! has no knowledge of adjustments made by the Motion System to maintain balance, which thus violates the underlying assumption of a static base throughout execution. To address these challenges, a template-based approach provides an interface for specifying sequences of actions to perform for completion of complex manipulation tasks (Romay, Kohlbrecher, Conner, Stumpf, & von Stryk, 2014). This system includes an infrastructure for defining a database of object templates containing a set of parametrized actions pertaining to each object, such as moving the end effector through affordance trajectories (e.g., twist, rotate, pull) or to an adjacent pose relative to the object location. The combination of fine-tuned end-effector approach and grasp poses, along with object-specific affordance trajectories, contains the necessary information to compute end-effector goals and planning constraints to perform safe and reliable manipulation planning.

3.7. Human-robot Interaction

Human operators utilize the OCS, shown in Figure 9, to visualize relevant perception, planning, and state data transmitted through the degraded communication link in three distinct views. This tool provides a single configurable interface for representing all pertinent data in a unified framework, as well as providing command and control interfaces for ESCHER. A camera view, shown on the left window of Figure 9, provides the operator with the most recent camera image received from the robot overlaid with virtual templates and footstep plans, and includes controls for the neck joints. An overhead view, shown on the right window of Figure 9, displays obstacle maps, footstep plans, and the robot's bounding box for navigation. The main view, shown on the center window of Figure 9, provides a 3D model of the robot with registered assembled laser scans, virtual fixtures, and footsteps. Surrounding this view are the manipulation interface, including menus to select virtual fixtures, grasps, and stances, and the locomotion interface, including the footstep generator, footstep planning goal selection, and planning parameter set selection. A small status window provides feedback from the planning and execution of the footstep and manipulation subsystems, both onboard and offboard the robot.

To reduce operator training time, fully exploit expert guidance, and minimize individual cognitive load throughout operation at the DRC Finals, a multioperator configuration covered four key roles identified through early system testing: locomotion, manipulation, perception, and supervisor. Locomotion focused on adjusting footstep planner parameters to improve walking speed while walking the driving course bypass, and to perform rough positioning for manipulation tasks. Manipulation operators were

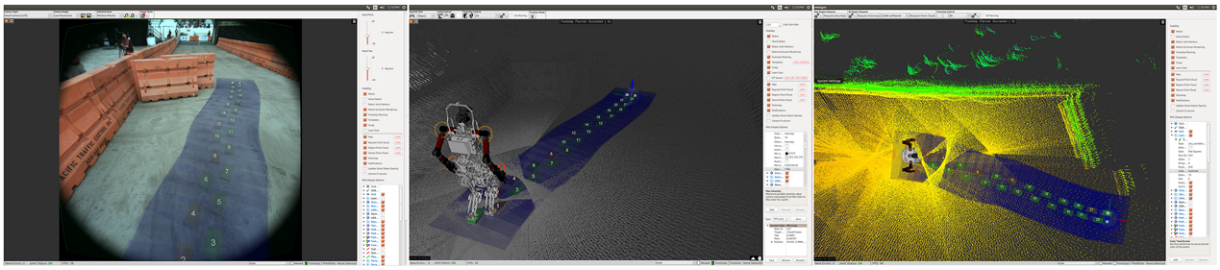


Figure 9. OCS from the Finals driving course, showing the camera (left), main (center), and overhead (right) views, each overlaid with the current footstep plan.

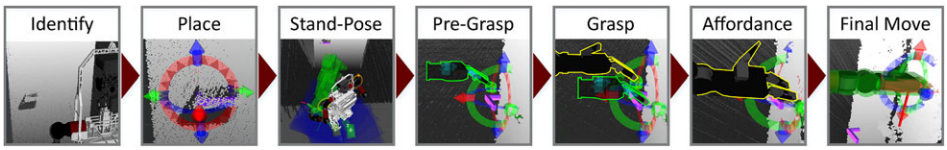


Figure 10. Operator manipulation process illustrating the steps of grasping a door handle. An operator identifies an object, places a template, moves the robot to a stand-pose, commands the end-effector to pregrasp and grasp poses, and executes an affordance. The selected pregrasp/grasp and actual robot end-effector are outlined in green and yellow, respectively, to improve visibility.

responsible for moving virtual fixtures within the OCS and commanding affordances to complete the manipulation tasks of the competition. Perception monitored the world modeling and localization systems to ensure the data visualized by locomotion and manipulation operators provided a consistent view of the environment. The supervisor handled safely launching the robot, emergency debugging, transitioning robot control between operators, and acting as the official point of contact for DARPA during runs. The team benefited from this distributed approach by allowing operators to test individual strategies for particular tasks and to switch operators based on individual performance over the course of testing and development.

The manipulation process utilized by Team VALOR relies on human operators to identify manipulation targets and to select an appropriate approach to interact with them. Team VALOR utilized a template-based approach based on the work done by Romay et al. in order to create an interface for specifying sequences of actions to perform (Romay et al., 2014). This system defined a database of object templates that associates sets of parametrized grasps, affordance actions (e.g., twist, rotate, pull), and stand poses unique to each object. Operators follow the manipulation process shown in Figure 10, utilizing the OCS to identify objects, place virtual fixtures, and execute motions. After identifying a manipulation target from perception data presented in the OCS, an operator places the appropriate object template by aligning the the template with the LIDAR point cloud and visual data. Once aligned, the operator selects a robot stand pose suited to the environmental constraints; the robot then generates a footstep plan for the operator to review and approve. Once at the stand pose, the operator selects a grasp and commands the selected manipulator to go to a pregrasp position associated with the selected grasp. This pregrasp ensures that the end effector approaches the object from the correct direction and helps to produce an initial move trajectory that is free of collisions. After the end effector reaches the final grasp pose, the operator may chose to execute a predefined affordance on the object or to move either the object template or end-effector template to a new goal pose.

3.8. Communication Management

The Finals included a degraded network link between the fielded robot and human operators, simulating the realistic conditions common in disaster zones. The communication network consisted of three links: Link 1, a 300 Mbit/s bidirectional wireless link between the robot and the field computer; Link 2, a 300 Mbit/s unidirectional link from the field computer to the human operators; and Link 3, a 0.0096 Mbit/s bidirectional link between the field computer and the operators. For the "indoor" tasks, Link 2 suffered from periodic blackouts ranging between 1 and 30 s in duration. The network connection dropped any traffic exceeding these limits, requiring bandwidth management to ensure a consistent and usable control interface.

Managing bandwidth between operators and the field computer focused largely around compressing data and throttling message topics. Both compression and throttling were performed by the Comms Bridge, one of the ROS packages shared by Team ViGIR. The Comms Bridge sends low-level motion, path planning, and state change commands to the robot while streaming LIDAR, video, pose, and state data back to the operators. It does this by interfacing with specified topics in ROS, handling

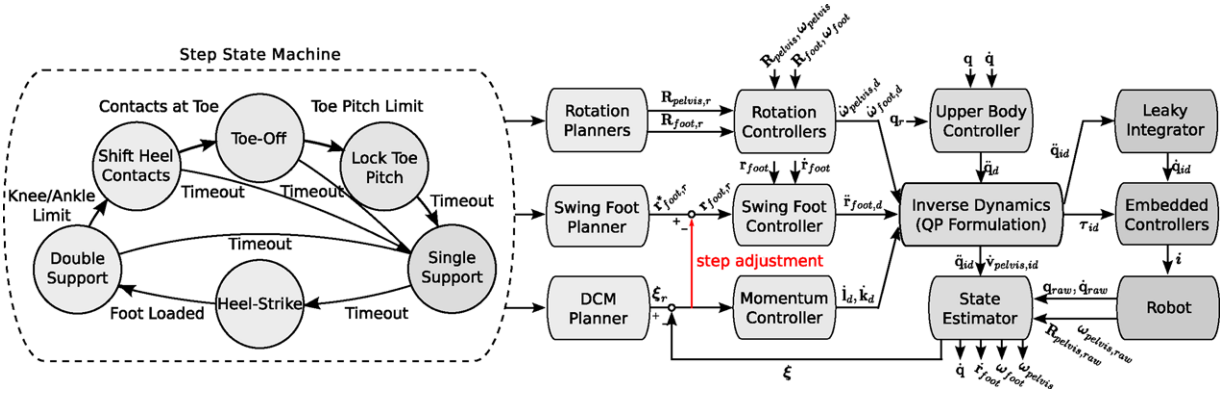


Figure 11. High-level block diagram of ESCHER's stepping controller. Note that some control paths have been omitted to improve readability.

compression/decompression, message throttling, and serialization, with data being sent over the links via direct TCP/UDP connections. This semitransparent connection between the onboard and operator ROS networks firewalled the challenge-specific network limitations from the rest of the system, and provided a smooth path from early, single ROS network testing through both nondegraded and degraded managed communications.

4. THEORETICAL PRINCIPLES OF THE MOTION SYSTEM

The Motion System developed by Team VALOR implements a compliant whole-body control strategy that relies on the time-varying DCM to regulate momentum during locomotion. While the Motion System involves numerous subsystems, its core functionality rests on a few key control principles and algorithms. This section presents the controls approach used to enable robust whole-body control and locomotion on ESCHER. This is an extension of the compliant locomotion framework described in Hopkins, Griffin, Leonessa, Lattimer, & Furukawa (2015a) and Hopkins et al. (2014), for which a high-level block diagram is included in Figure 11. While much of the theoretical background is published in previous works (Hopkins et al., 2014, 2015a,b,c), this section covers the work already done and introduces new features developed for the DRC Finals.

4.1. Dynamic Models

As in Feng et al. (2015), Kuindersma et al. (2014), and Koolen et al. (2013), the controls approach employed by Team VALOR for the DRC approximates the whole-body dynamics of the robot using a rigid body model. Since it is possible to treat SEAs as pure torque sources, the joint torques, τ, are a linear function of the contact forces, $\mathbf{f}_c$, and joint accelerations given estimated joint positions and velocities.

The controls methodology presented here makes extensive use of the DCM to stabilize the centroidal dynamics of the rigid body system while walking. The DCM represents a linear transformation of the CoM state that separates the second-order linear inverted pendulum dynamics into coupled, first-order unstable and stable systems (Englsberger et al., 2013), with the time-varying formulation presented in Hopkins et al. (2014) is defined as

$$\xi = \mathbf{x} + \frac{1}{\omega}\dot{\mathbf{x}}, \tag{1}$$

where $\mathbf{x}$ is the CoM position and $\omega(t) > 0$ is the time-varying natural frequency of the CoM dynamics. This unstable dynamic representation defines the 3D point at which the CoM converges, a similar concept to the instantaneous capture point introduced in Pratt, Carff, Drakunov, & Goswami (2006). Through appropriate planning of the DCM, the CoM can be stabilized for one or more steps. The DCM can be controlled with the virtual repellent point (VRP), which lies above the enhanced centroidal moment pivot (eCMP) point. The VRP repels the DCM at a rate proportional to its distance, simultaneously repelling the CoM. By definition, the VRP maps the position of the CoM to the total desired linear momentum rate of change acting on the system. The eCMP encodes the contact forces by mapping the CoM position to the net contact force (Englsberger et al., 2013; Hopkins et al., 2014). By using the time-varying formulation, better control of the CoM height is possible when traversing uneven terrain by relaxing the assumption of a constant pendulum length (Hopkins et al., 2014).

4.2. State Machine

A finite-state machine transitioning between various contact phases, depicted on the far left of Figure 11, is used to enable single and multistep behaviors. In double support, both feet are assumed to be in contact with the ground and have contact points enabled through which the robot

exerts contact forces. During single support, the swing foot travels to a new foothold position. After a specified duration, the state machine transitions to heel-strike, where the swing foot is lowered at a constant velocity until it achieves a desired contact force. To guarantee adequate contact between the swing foot and the ground at heel-strike, decoupling of the swing foot timing from the CoM motion proved to be critical. Transitioning to double support is delayed until the foot is sufficiently loaded for a defined duration, and additional time can be added to the single support phase to prevent preemptive loading of the swing foot. When switching between double and single support, the enabling/disabling of contact points can result in discontinuities in the desired contact forces if not handled appropriately. The abrupt change in contact force can result in jerky motions, reducing stability. Applying a linear ramp to the maximum contact forces when approaching heel-strike and single support transitions ensures these transitions are piecewise linear continuous.

To compensate for leg ROM limits if a predefined ankle or knee pitch limit is encountered by the swing leg prior to single support, the robot transitions to a reactive toe-off state. The contact points on the heel are shifted toward the toe at a specifiable rate to avoid instantaneous restriction of the support polygon, which can result in the center of pressure (CoP) lying outside the base of support. Once the heel is unloaded, the swing foot rotates about the toe contact points at a constant velocity, shifting the ankle pitch away from the soft joint limit.

4.3. Task-space Planning

To determine whole-body motions, a series of reference trajectories are generated. A high-level footstep planner populates a footstep queue with desired foothold poses and step durations, which are used to compute task-space reference trajectories for the swing foot, pelvis, and DCM at the beginning of each double support phase. The reference DCM trajectory is generated after defining CoP and CoM height trajectories using the real-time numeric planner described in Hopkins et al. (2014). Although this approach is more computationally intensive than the analytical DCM planners proposed by Englsberger et al. (Englsberger et al., 2013; Englsberger et al., 2014a), it permits the usage of generic CoP and vertical CoM trajectories.

The pelvis rotation and swing foot pose are generated using piecewise minimum jerk trajectories that interpolate between intermediate waypoints calculated from the initial and final foothold poses to ensure smooth motions. Based on height change in the desired foothold, velocities at the waypoints are found using a sigmoid function that blends the interpolated and average waypoint velocities. Intermediate waypoints can also be defined by a higher-level planner; however, planning of swing-foot trajectories at the motion system level removes the burden from the high-level planner and eases tuning by using predefined motion strategies.

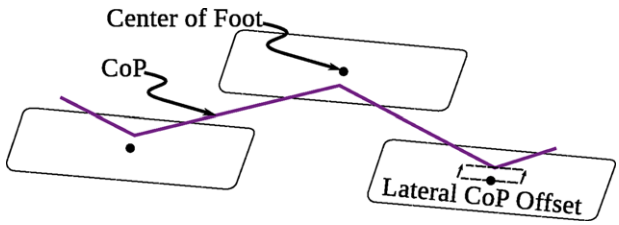


Figure 12. The nominal CoP trajectory is shifted toward the inside of the support foot to improve stability on soft terrain and reduce lateral motion of the CoM during walking.

The upper body joint trajectories, $\mathbf{q}_r(t)$, are defined by an internal planner that manages arm swinging during stepping using simple offsets on the shoulder pitch derived from the current hip pitch. Additionally, shoulder roll limits are adjusted based on the current hip roll and yaw to avoid collisions with the lower body. During manipulation, these trajectories are determined by an external planner.

4.3.1. CoP Trajectory Planning for Compliant Terrain

As the controller assumes a rigid contact model, soft terrain such as grass and dirt introduces significant unmodeled dynamics during walking. Although low-impedance control of the lower body provides some robustness to surface compliance, poor DCM tracking can lead to tipping when the CoP deviates to the edge of the support polygon. During the course of development, the authors found that introducing a lateral offset to shift the CoP reference trajectory toward the inside of the support foot, shown in Figure 12, significantly improves performance on soft and uncertain terrain, as presented and demonstrated in Hopkins et al. (2015a). This decreases the lateral motion of the CoM similar to narrowing the step width without the additional risk of self-collision. By increasing the nominal distance of the CoP to the outer edge of the foot, additional horizontal torque is available to correct for DCM overshoot, decreasing the risk of outward tipping. In the event of inward tipping, the controller can still regain stability through appropriate step adjustment.

4.3.2. CoM Height Trajectory Planning

To enable safe navigation of terrain with significant height changes, careful design of the nominal CoM height trajectory was required to cope with limited ankle and knee ROMs. Figure 13 includes the vertical CoM, DCM, and VRP reference trajectories corresponding to a 20 cm step up and down. Note that the CoM begins to accelerate at the beginning of the double support phase when ascending, and then at the end of single support when descending. Raising the CoM height early in the step cycle tends to rotate the

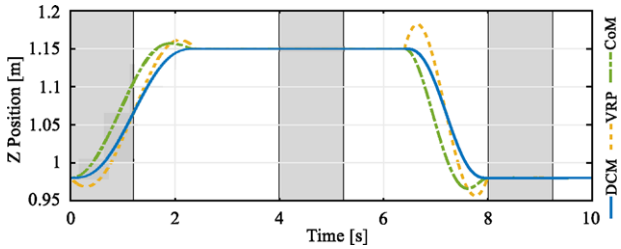


Figure 13. During ascension, vertical CoM acceleration initiates at the beginning of the double support phase (shaded in gray), reducing the required knee torque. During descension, CoM acceleration occurs at the end of single support.

Table II. Whole-body controller weights and gains[1].

Motion task	Units	Weight	P-Gain	D-Gain
$\dot{\mathbf{l}}_d$	N	5, 5, 1[2]		
$\dot{\mathbf{k}}_d$	Nm	5, 5, 0		
$\dot{\omega}_{\text{pelvis},d}$	rad/s^2	100, 100, 100	70, 70, 30	30, 30, 15
$\dot{\omega}_{\text{foot},d}$	rad/s^2	500, 500, 500	100, 100, 75	5, 5, 5
$\ddot{\mathbf{r}}_{\text{foot},d}$	m/s^2	1e3, 1e3, 1e3	50, 50, 100	35, 35, 35
$\ddot{\mathbf{r}}_{\text{contact},d}$	m/s^2	1e5, 1e5, 1e5	0, 0, 0	0, 0, 0
$\ddot{\mathbf{q}}_{\text{arm},d}$	rad/s^2	15	45	10
$\ddot{\mathbf{q}}_{\text{waist},d}$	rad/s^2	100	40	20

[1]Cartesian weights and gains are specified for the x, y, and z axes.
[2]These weights are decreased to (2.5, 5, 1) during the single support phase to reduce oscillations in the sagittal plane.

support ankle away from the soft position limit, with the added benefit of straightening the support knee for lower knee torques. Significant toe-off is also required to allow the foot to remain in contact with the ground prior to lift-off. Lower knee torques result by moving the height change to the end of single support when stepping down, as the knee is significantly less bent for the majority of single support.

4.4. Task-space Control

To realize the planned task-space motions, task-space position and velocity errors are regulated using a set of feedback controllers. The upper body joint space trajectories and 6-DOF Cartesian trajectories are tracked using individual PID controllers presented in Hopkins et al. (2015b). Table II lists the proportional and derivative gains used to compute desired linear and angular momentum rates of change, pelvis and swing foot accelerations, and upper body joint accelerations used in the Finals. To maintain rotation invariance, x- and y-axis gains are represented in pelvis yaw coordinates.

The desired linear momentum rate of change is calculated using a DCM tracking controller defined in Hopkins et al. (2014), designed to stabilize the centroidal dynamics that includes both proportional and integral actions. Proportional and integral gains of 3 and 1 m/m were selected for the Finals, noting that the integral action is disabled during single support and heel-strike to prevent windup (Hopkins et al., 2015b).

The desired angular momentum rate of change is defined as $\dot{\mathbf{k}}_d = \mathbf{0}$. This reflects the assumption that the eCMP and CoP are collocated during walking, in which case the horizontal moment about the CoM is equal to zero. Note that the robot's angular momentum is not directly regulated using feedback control. The approach used relies on Cartesian and joint-space PID controllers to track the pelvis, swing foot, and upper body trajectories and prevent excessive angular momentum during walking.

4.5. Whole-body Control and Inverse Dynamics (QP Formulation)

Given desired task-space forces and accelerations, an inverse dynamics solver presented in Hopkins et al. (2015b) is used to compute optimal joint accelerations, $\ddot{\mathbf{q}}$, and generalized contact forces, $\rho = [\rho_1^T, \ldots, \rho_N^T]^T$, by minimizing a quadratic cost function in the form

$$\min_{\ddot{\mathbf{q}},\rho} \left\| \mathbf{C}_b \left(\mathbf{b} - \dot{\mathbf{J}}\dot{\mathbf{q}} - \mathbf{J}\ddot{\mathbf{q}} \right) \right\|^2 + \lambda_{\ddot{\mathbf{q}}} \|\ddot{\mathbf{q}}\|^2 + \lambda_\rho \|\rho\|^2, \qquad (2)$$

where $\mathbf{b}$ represents the vector of desired motion tasks, $\mathbf{J}$ represents the corresponding matrix of task-space Jacobians, $\mathbf{Q}_\mathbf{b} = \mathbf{C}_\mathbf{b}^T \mathbf{C}_\mathbf{b}$ represents the task weighting matrix, and $\lambda_{\ddot{\mathbf{q}}}$ and λ_ρ define the regularization parameters. The experimentally selected weights for each motion task used in the Finals are listed in Table II. The QP also includes Newton-Euler constraints for the rigid body dynamics and Coulomb friction constraints for each contact point, as well as constraints on joint position and torque limits, as presented in Hopkins et al. (2015b).

To cope with the limited speed capabilities of the HDT arms, additional constraints inspired by the soft joint limit constraints in Saab et al. (2013) were introduced,

$$k_{\dot{q}}(\underline{\dot{\mathbf{q}}} - \dot{\mathbf{q}}) \le \ddot{\mathbf{q}} \le k_{\dot{q}}(\bar{\dot{\mathbf{q}}} - \dot{\mathbf{q}}), \qquad (3)$$

where $k_{\dot{q}}$ acts as the constraint stiffness, and $\underline{\dot{\mathbf{q}}}$ and $\bar{\dot{\mathbf{q}}}$ refer to the lower and upper speed limits, respectively. Limiting the arm joint velocities proved to be critical, as the motor speed and tracking ability of the arm's embedded controller were limited. The resulting lower velocity set points also yielded a decreased power draw from the arms.

To further alleviate the effects of rapidly changing contact forces when switching between single and double support, the joint torques were smoothed by limiting the joint torque rates of change through

$$\Delta T \underline{\dot{\tau}} + \tau_{k-1} \le \tau \le \Delta T \bar{\dot{\tau}} + \tau_{k-1}, \qquad (4)$$

where τ_{k-1} is the previous joint torque set point, ΔT is the optimization time step, and $\underline{\dot{\tau}}$ and $\bar{\dot{\tau}}$ are the maximum torque rates of change. This helped prevent rapid changes in torque set points, which can lead to shaky CoM motions.

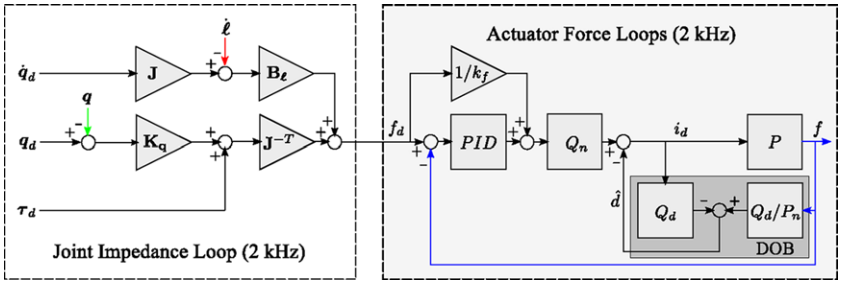


Figure 14. Diagram of cascaded low-level controller, containing an outer joint impedance feedback loop and inner actuator force feedback loop.

Joint velocity set points, $\dot{\mathbf{q}}_a^*$, are required for the lower body impedance and upper body velocity joint controllers. These set points must be computed from the optimized joint accelerations, $\ddot{\mathbf{q}}_a$, in a way that does not diverge from the estimated values for stability. To do this, a leaky integrator defined in Hopkins et al. (2015b) as

$$\ddot{\mathbf{q}}_i^* = \alpha\,(\dot{\mathbf{q}} - \dot{\mathbf{q}}_i^*) + \ddot{\mathbf{q}}_a \tag{5}$$

is used, where $\dot{\mathbf{q}}_i^* = \int \ddot{\mathbf{q}}_i^* dt$ represents the integrated joint velocity. The integral drift rate toward the estimated joint velocity, $\dot{\mathbf{q}}$, is set by the leak rate $\alpha \geq 0$. Increasing the leak rate was found to have dramatic effects on stability and joint tracking when implemented on hardware.

To reduce high-frequency oscillations in the lower body joints, viscous joint space damping was added by applying $\dot{\mathbf{q}}_a^* = \gamma \dot{\mathbf{q}}_a^*$. Here γ is a viscous damping term that damps the joint velocity values toward zero. For the high-gain velocity controlled upper body, no viscous damping was used, while the lower body used $\gamma = 0.6$. This combination of tuning the leak rate and viscous damping was found to be critical to maintain stability and ensure tracking of the desired joint velocities. By leaking toward the current joint velocity, the desired joint velocity does not diverge from the estimated value, ensuring the joint impedance commands do not expend control authority tracking diverging dynamics, while the viscous damping prevents actuator jitter.

4.6. Joint-level Control

The computed torque and velocity set points are relayed to the embedded motor controller outlined in Figure 14 at a rate of 150 Hz. Upper body trajectories are tracked using a high-gain velocity controller for each DOF, while lower body trajectories are tracked using the outer loop low-gain joint impedance controller presented in Hopkins et al. (2015c) using actuator velocity feedback. This approach significantly improves velocity tracking because of the collocation of the motor and incremental encoder, and due to velocity feedback being actuator-independent for parallely actuated joints. The force error is then regulated with inner loop PID feedback, which also converts force commands to the desired motor current. A disturbance observer (DOB) based on the plant dynamics was implemented on the current feedback loop to reduce errors resulting from factors such as stiction and dynamic coupling of the actuators. This results in excellent torque tracking, a critical component for successful stabilization of the CoM (Hopkins et al., 2015c).

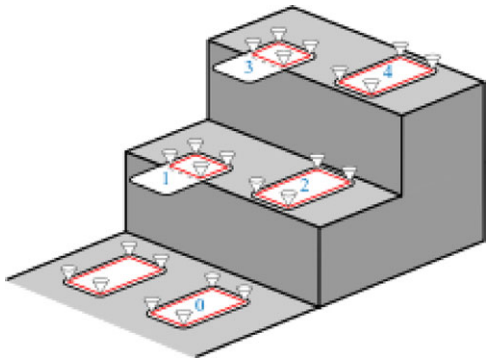


Figure 15. The half-step strategy implemented increases the available collision-free workspace of the support leg when ascending stairs.

4.7. Increased Step Workspace

The DRC course included an industrial stairway and a cinder block rubble course designed to test the robot's mobility over uneven terrain. A number of difficulties arose while testing due to the risk of collision between the support leg and the upper stair when stepping up. To increase the available workspace, a half-step strategy was implemented, allowing the robot to place the center of the support foot on the lip of the stair. As shown in Figure 15, the rear contact points were shifted to the middle of the foot to appropriately constrain the CoP trajectory. To prevent knee and shin collisions, a soft position limit was enacted on the ankle pitch joint corresponding to the predicted angle of collision. The whole-body controller was able to adjust the ankle pitch using this approach to avoid collision during stepping.

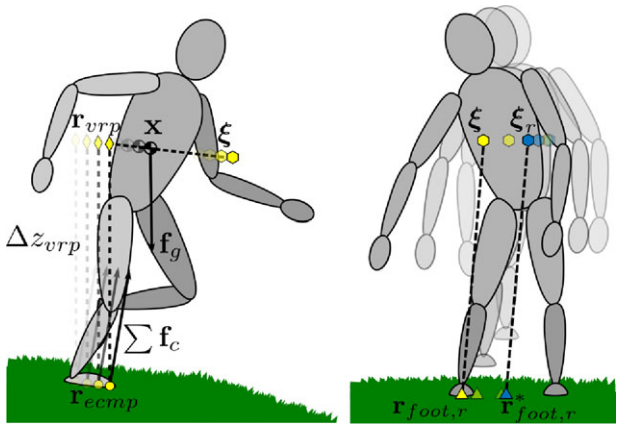


Figure 16. Left: DCM dynamics and external forces acting on an articulated humanoid. Right: lateral step adjustment based on the estimated DCM error. Here $\mathbf{r}^*_{\text{foot},r}$ and $\mathbf{r}_{\text{foot},r}$ are the nominal and adjusted swing foot positions.

4.8. Step Adjustment

To stabilize the DCM in the face of poor tracking or external disturbances, the momentum controller will shift the eCMP and CoP away from the nominal reference position. This can induce a significant moment about the CoM if the eCMP leaves the base of support. The corresponding angular momentum rates of change required to prevent a fall are not always possible to sustain due to the robot's limited range of motion. A fall can often be avoided in this situation by modifying the base of support through step adjustment. A simple heuristic presented and demonstrated in Hopkins et al. (2015a) was used to implement real-time step adjustment in response to significant disturbances based on modifying the swing foot position by the DCM tracking error. Figure 16 illustrates an ideal lateral step adjustment in response to a large DCM error, where the offset between the final DCM and the swing foot position is equivalent to the offset between the reference DCM and the unadjusted swing foot position. The future foothold positions are specified relative to the current support foot position, requiring the high-level footstep planner to correct for horizontal drift in foothold tracking.

5. TESTING AND DRC FINALS RESULTS

5.1. Pre-Finals Testing

The original rules set for the DRC Finals required vehicle egress following the driving task. Given the need to build a completely new robot, hardware testing needed to occur on an extremely shortened timeline. Due to these restrictions, the combined difficulty of driving and egressing from the vehicle, and the team's expertise in bipedal locomotion, the decision was made to bypass the driving task by walking. Though the rules were later changed to allow a reset to bypass egress, ESCHER was mechanically committed to walking the course and implementing the driving capability required resources that were unavailable.

Testing prior to the DRC Finals consisted of both subsystem and full system tests. Subsystem tests focused on debugging software and tuning parameters used for locomotion and manipulation. Full-system tests focused on mimicking the conditions expected at the DRC Finals as closely as possible. The robot was commanded through a 61 m walking course, a door task, and a valve task both indoors and outdoors by operators that were isolated from the robot. This testing demonstrated ESCHER's capability to walk on a variety of terrains, the platform's ample battery life, and it validated the strategy of bypassing the driving task, focusing on the door and valve tasks, and attempting the surprise task with the remaining time.

5.1.1. Knee Torque Testing

The dual-actuator knee design was validated by having ESCHER step onto a 0.23 m step in both simulation and hardware. Figure 17 shows a comparison between simulation and hardware test results for the left knee, the most heavily loaded joint in this experiment. This figure demonstrates that the simulation is a close enough representation of reality for design purposes, and that the dual-actuator knee design provides sufficient torque for completing DRC locomotion tasks. For additional testing and validation of ESCHER's design, see Knabe et al. (2015).

5.1.2. Walk Testing

As reliable walking is imperative for bipeds, locomotion testing formed a key part of the test regime leading up to the Finals. Despite ESCHER being a new platform, it bore enough mechatronic and inertial similarities to THOR that it required only two weeks to tune the motion system using an aggressive testing schedule. This rapid start-up time enabled full system testing closely resembling competition conditions to begin within one month of mechanical completion of the robot. Throughout locomotion endurance testing, ESCHER walked over 1,500 m on various types of terrain, as shown in Figure 18, including gravel, grass, brick, and concrete.

One of the primary locomotion tests consisted of walking a 61 m course with randomly placed large-scale obstacles to mimic bypassing the driving task. While ESCHER demonstrated the ability to travel 61 m in an average of 11 min, the introduction of obstacles into the environment required planning for safe navigation. Many factors impact walking velocity at the task level, such as planning time, validating safe execution, length of executed footstep sequences, and perception limitations due to limited line of sight. Testing revealed that increasing the locomotion duty cycle, defined as $\text{Duty Cycle} = \frac{\text{Walking Time}}{\text{Planning Time} + \text{Walking Time}}$,

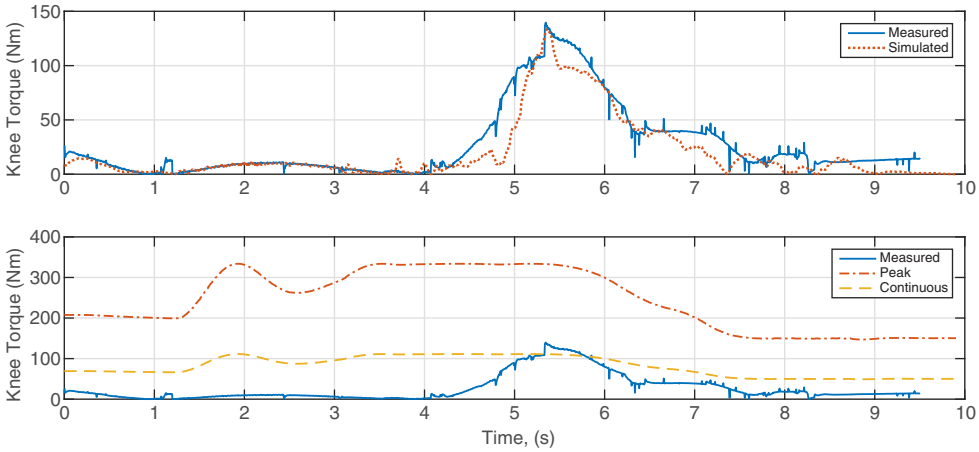


Figure 17. Validation of the dual-actuator knee. Top: a comparison of simulated torques vs measured torques of the left knee. Bottom: a comparison of measured torques of the left knee with the estimated continuous and peak limits (Knabe et al., 2015).

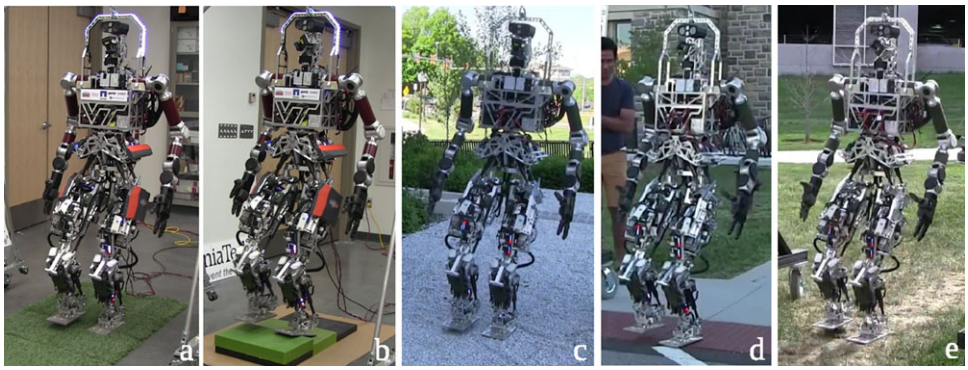


Figure 18. ESCHER walking on a variety terrain including grass, gravel, and 3.8 cm blocks. In each case, the controller assumes a rigid contact surface and has no knowledge of height variations or surface compliance.

would have the greatest impact on reducing overall locomotion times throughout the competition. As shown in Figure 19, planning time using the footstep planner described in Section 3.5 is non-negligible compared to the time spent walking, with an average time of 4.185 s per step in planning compared to the 2 s steps typical for ESCHER, resulting in a locomotion duty cycle of just 36% during these full system tests. Furthermore, the footstep planner tends to produce plans with an excessive number of footsteps at the end to achieve the precise footholds specified by the user, which is not useful when attempting to maximize speed in open areas. This resulted in ESCHER walking with an average moving velocity of 0.066 m/s during motion execution, 12% less than the maximum speed of 0.075 m/s governed by the footstep planner's maximum step length, but only an average velocity of just 0.024 m/s if planning time is also included. At this speed, the walking segment was expected to take almost 43 min, the vast majority of the allotted 60 min time.

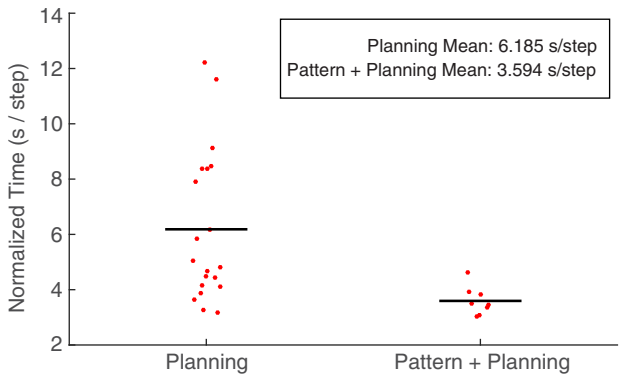


Figure 19. Average time to plan and execute a step in a sequence. Planning covers traversals with footstep sequences generated purely by the planner, while Pattern + Planning refers to traversals completed using a combination of pattern generation and footstep planning.

In contrast, the combination of pattern generation for gross forwards movements and planning for course correction results in an average planning time of 1.566 s per step, resulting in a 65% walking duty cycle, an average walking velocity of 0.074 m/s, and an overall average velocity of 0.048 m/s. With double the average velocity of a purely planned walking approach, the expected time for traversing the Finals Driving course was halved to 21 min.

5.1.3. Battery Life Testing

Battery testing of ESCHER focused on validating that the robot would meet the required runtime of the DRC Finals. Before and after battery measurements, along with average current drawn during tethered testing, indicated that the robot consumes 271 Wh per hour under nominal conditions. All testing was conducted with the smaller 710 Wh set of batteries since they would provide ample run time, running the batteries from 100% charge to 20% charge to reduce wear. Three full system tests, with a peak duration of 58:44 min, before the competition confirmed that battery life was sufficient. In these tests, onboard computers ran all software required for the competition while the robot performed competition tasks. Across 5 days of trials with a minimum of 4 h of testing, the batteries showed an average life of 195 min decreasing to 168 min as the robot was more active during testing. Finally, for Day 2 of the DRC Finals, the robot was powered up and standing for a full 14 min before the beginning of the 60 min competition run and was shut down at the conclusion of the run with charge remaining, confirming that the designed power system was sufficient to power the robot for the duration of the competition.

5.1.4. Manipulation Testing

Manipulation testing served as an opportunity for operators to familiarize themselves with Team ViGIR's manipulation system, verify grasps and stances developed in simulation for each task, and evaluate task completion times. The results presented in this section represent ESCHER's manipulation capabilities using Team VALOR's DRC Finals OCS configuration. The slow pace of the robot revealed by locomotion testing led Team VALOR to focus on the door and valve tasks, since completion of the door task was necessary to reach the indoor tasks, and the valve task was the next closest task to the door.

The reachability analysis shown in Figure 5 led to adjustments in stand-poses facing the robot up to 90 degrees away from the object in order to place its grasps in regions with more kinematic solutions. For example, posing the robot 75 degrees relative to the valve task generally provided the best combination of increased reachability while maintaining the operator's line of sight to the target. The door task was executed oriented at 90 degrees to eliminate the need to rotate the robot prior to side-stepping through the door.

Manipulation testing focused on improving both the speed and safety of the manipulation tasks. During practice, operators focused on accomplishing the task consistently without making repeated, time-wasting attempts. Operators also focused on executing tasks safely to avoid making a mistake that might end the run prematurely, such as knocking the robot over, or requiring a reset, which would also waste valuable time.

Manipulation testing focused on the door and valve tasks with additional preparation for a few potential surprise manipulation tasks. Table III shows the average times of each phase of the door and valve tasks; each task took about 10 min to complete. For timing purposes, the door task was broken into three phases. The Approach phase encompasses the time required to assume a position needed to reach the handle from approximately 2 m away. The Manipulation phase is the time required to turn the door handle and push it open. Finally, the Egress phase is the time taken to walk through the doorway. For the door task, the robot spent the majority of its time side-stepping through the door to avoid collisions with the door frame resulting from the sway of the CoM while walking.

The valve task was divided into two phases: Approach and Manipulation. The Approach phase covers the time required to reach the specified position relative to the valve. The Manipulation phase covers the time required to interact with the valve to turn it one full revolution. For the valve task, the manipulation phase took the most time because the configuration of the HDT manipulators required repeated grasping, turning, and releasing actions turning the valve on average 88 degrees and 98 s per cycle averaged over 20 successful interactions with the valve.

5.2. DRC Finals Results

System testing leading to the Finals provided empirical data to refine strategies for completing the given tasks. Locomotion testing indicated that walking the driving course would require between 20 and 40 min of the 1 h available for the competition. Maintaining a high walking duty cycle minimizes traversal time, which could be done by mostly utilizing the pattern generator and only resorting to full footstep planning only when the local environment demanded careful navigation. Operators demonstrated ESCHER's potential to score points on the door and valve tasks, each requiring 10 min to complete. The upper bound estimate of these tasks required the full hour allotted for the challenge, resulting in a conservative strategy that focused on walking the driving course, and completing the door and valve tasks. Any additional time would go toward completing the surprise task, which operators prepared for by introducing additional virtual fixtures and practicing in simulation.

Table III. Average practice times for the valve and door tasks. The asterisk refers to individual grasp, turn, and release cycles.

Tasks		Approach	Manipulation	Egress	Total time
Door	Average Time (min)	1:40	3:12	5:03	9:55
	Standard Deviation (min)	0:38	2:37	3:44	4:36
	Trials	7	9	3	
Valve	Average Time (min)	2:21	7:44		10:05
	Standard Deviation (min)	0:34	3:08		3:11
	Trials	3	20*		

5.2.1. Run 1

Figure 20 shows a timeline of the events and footstep plans during the first run at the DRC Finals. The start of the first run was delayed due to a change in how the degraded communication link operated during the final event. In testing, enabling degraded communication involved physically altering the link between the field computer and degraded link. For the Finals, the link was not physically altered, but instead routing tables inside the DARPA configuration were changed with the link remaining active. The computers were left in the default network configuration, in which recently used routing table entries are cached for performance reasons and cleared when network links are disconnected. Changing the routing tables without disconnecting the link disrupted network flow until the problem was identified; manually clearing the caches resolved the issue and started the run.

A software configuration error was discovered that corrupted raw footstep messages transmitted from the OCS, requiring the operator to utilize the footstep planner onboard ESCHER to generate all walking commands. Since the original plan to minimize traversal time relied on the footstep pattern generator, which was rendered inoperable by the above issue, the operator requested longer plans than had previously been tested. To further improve speed, footstep planning requests were started during execution of the previous plan in order to exploit the plan repair capabilities of the footstep planner once ESCHER stopped walking. This approach was only possible when strict tracking of the plan was not critical, such as during the straight, unobstructed sections of the track. ESCHER walked at approximately 0.07 m/s with a walking duty cycle of 65%, following footstep plans with an average length of 25 steps. After walking a third of the course, the onboard footstep planner attempted to transmit too large a footstep plan consisting of 36 steps back to the OCS. This overflowed the corresponding compression algorithm in the Comms Bridge, triggering a bug that disabled the ability to execute additional footstep plans. After unsuccessful attempts to work around the problem, the team called a second reset at 33 min to restart the Comms Bridge. The robot fell during the third footstep plan following the second reset due to a loose electrical connection to the left hip pitch motor controller. This caused the motor controller to enter a fault state, resulting in a constant velocity set point leading to loss of balance.

Strategically placed polystyrene foam covers on the thighs and pelvis mitigated damage from the fall. During the fall, the left arm impacted first, followed closely by the left thigh, pelvis, and KVH IMU housing on the rear of the pelvis, which were protected by the covers. In attempting to maintain balance during motor controller failure, the whole-body controller commanded the left knee joint to its

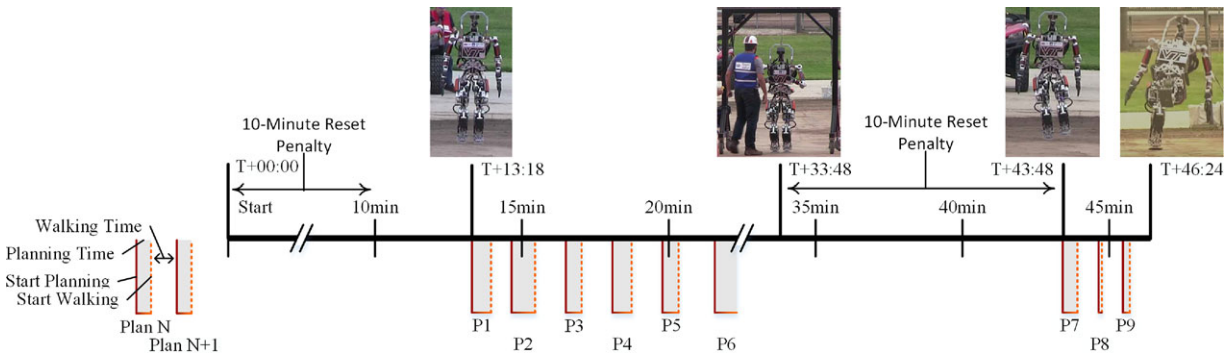


Figure 20. Day one timeline of the Finals illustrating key events, and footstep planning and execution times (photos courtesy of DARPA and Virginia Tech / Logan Wallace).

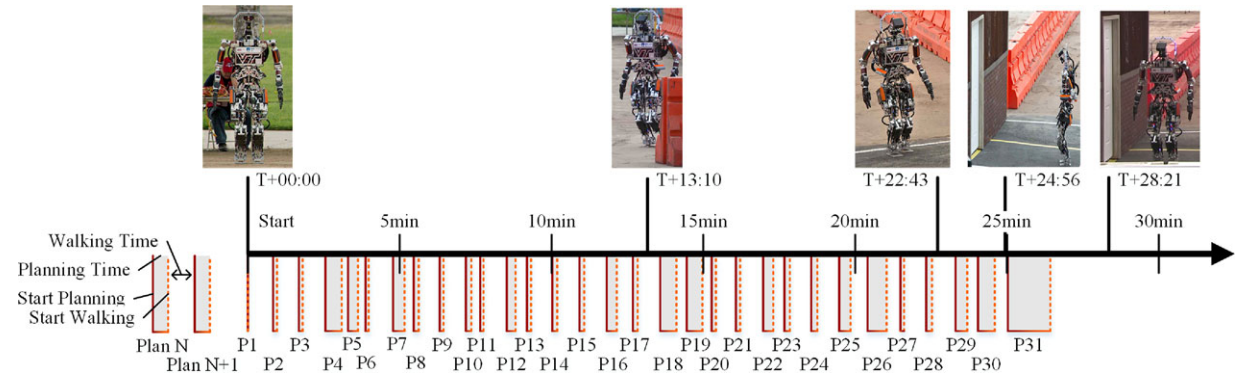


Figure 21. Day two timeline of the Finals illustrating key events, and footstep planning and execution times (photos courtesy of DARPA and Virginia Tech / Logan Wallace).

soft limit. During impact, the joint extended to its mechanical hard stop and induced full spring deflection overtraveling both ball nuts of the knee actuators and destroying the antitipping bushings at the end of both ball screws. The left leg then collapsed onto the left hand and impacted one of the two knee actuators, cracking the load bearing carbon fiber tube. The modular design of SEA subassemblies, repair knowledge amassed from designing and building the robot in-house, and foresight to bring a repair kit complete with spare components allowed rapid repair of the two damaged SEAs overnight. Following actuator repair work, all load cells and joint encoders in the lower body were rebiased. With limited time remaining prior to the second-day run, however, only basic locomotion and manipulation pose testing was performed to verify the functionality of the system.

In addition to hardware repairs, several bugs in the shared degraded communications bridge were fixed and tested. Testing and validating these fixes were distributed among affected teams, ensuring that operations for the second day would proceed according to the original strategy.

5.2.2. Run 2

Figure 21 shows a timeline from the start of the run to arrival at the door for the second attempt at the Finals course. The robot traversed the 61 m dirt course in 23 min using 28 total footstep plans, successfully walking the bypass of the driving course, as shown in Figure 22. This run shows a significant change from day 1 as a result of using the footstep generator. While the global mean duty cycle remained relatively unchanged at 67%, day 1 did not include the more complicated plans required for obstacle avoidance. A breakdown of the run resulted in a duty cycle of 81% for unobstructed walking, while the duty cycle during full footstep planning used for course correction around obstacles was a much lower 47%. The final footstep plan initiated at the

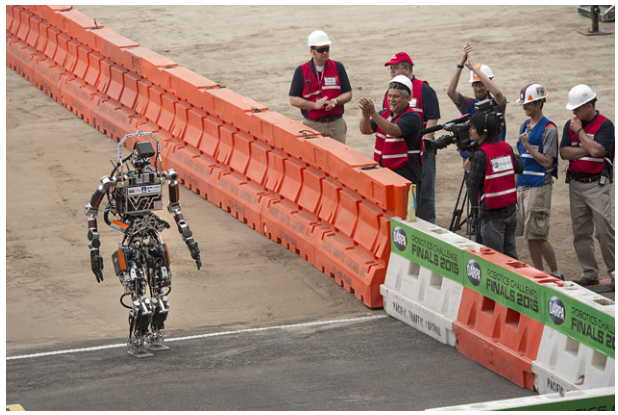


Figure 22. ESCHER finishing the dirt track at the Finals (photo courtesy of Virginia Tech / Logan Wallace).

25 min mark positioned ESCHER to attempt to open the door.

The manipulation operator attempted the door task as planned, however errant sensor readings in the wrist, discovered during execution, prevented successful completion. Figure 23 displays time-synchronized views of the head camera perspective, the 3D model, and the competition video. These images, the first two of which were reconstructed from data logs, expose a clear difference between the joint angle reported by the wrist abduction encoder and the actual position of that joint. The inconsistent scene in the OCS produced by this difference rendered the virtual fixture and affordance approach ineffective. A significant amount of time elapsed prior to the manipulation operator discovering this issue. The remaining time was spent unsuccessfully attempting several workarounds, such as manually adjusting the handle virtual fixture location, the manipulator goal pose, the robot's center of mass, and the robot's stance. These attempts were hindered by the end

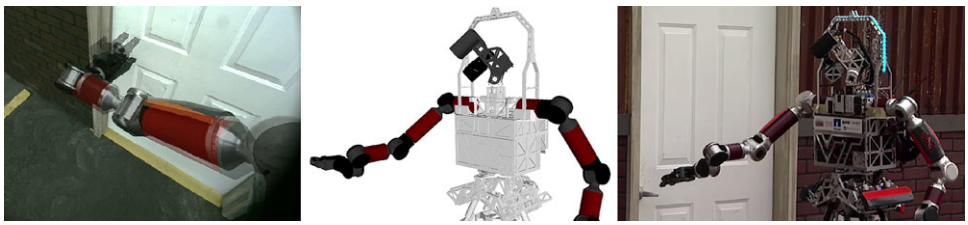


Figure 23. Operator view of door task through the MultiSense overlaid with ghost robot model (left), robot measured pose from proprioceptive sensors visualized in RViz (center), and photograph of ESCHER at door task (right) displaying discrepancies in measured and actual end-effector position.

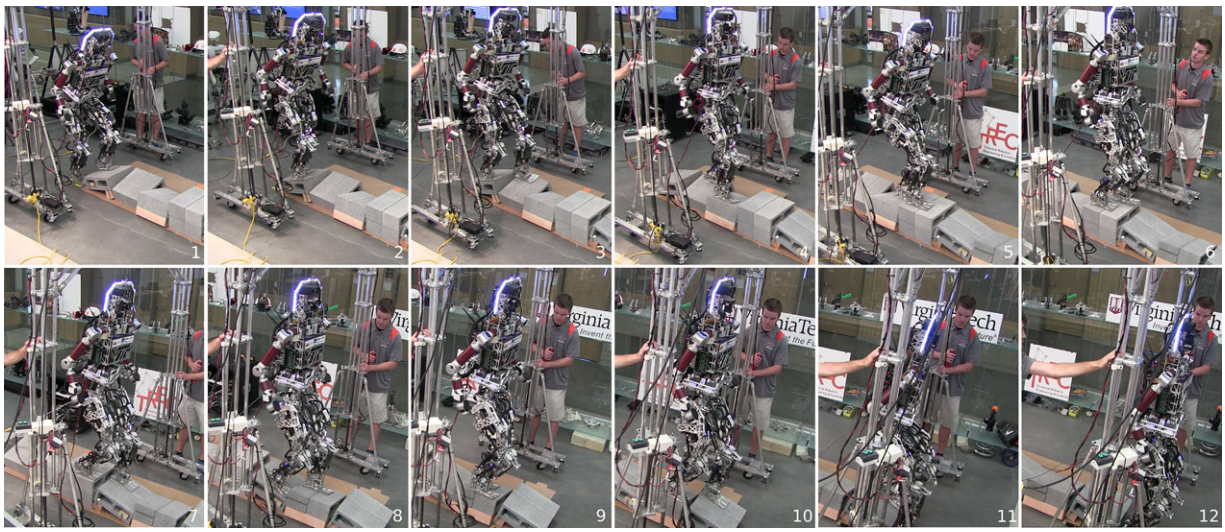


Figure 24. Time-lapse of ESCHER traversing a cinder block course similar to the Finals rubble task.

effector occluding the door handle, which limited the potential feedback from perception. In the final minutes of the run, the door handle was partially turned, but the robot was nearing an unsafe pose.

5.3. Post-Finals Validation

Due to Team VALOR's pre-Finals focus on walking and completion of the door and valve tasks, little time was allotted for testing ESCHER's capability to perform the remaining DRC tasks. As validation of the hardware platform, preliminary tests demonstrated that the robot could step up 0.23 m for the stairs task and on slanted cinder blocks for the rubble task, but full task demonstrations were not completed until after the DRC. Likewise, manipulation testing conducted prior to the Finals only represented some of the functionality of ESCHER's hardware, software, and controls systems. In an effort to demonstrate the platform's potential for emergency response, the team ran ESCHER through each remaining task from the competition.

5.3.1. Rubble and Stairs Tasks

Figure 24 contains images of ESCHER traversing a rough terrain course resembling the Finals rubble task. Due to the precise footstep placement required to properly navigate the various cinder block orientations, operator-specified footsteps were chosen over those generated with the footstep planner. To decrease planning time and ensure foot placements that were dynamically realizable, footstep patterns were empirically determined for each possible cinder block orientation using a step duration of 4 s. While descending from the final cinder block poses a risk of reaching ROM limits on the ankle pitch, using the CoM height plan proposed in Section 4.3.2 addressed this issue and rendered adaptation of the half-step strategy described in Section 4.7 unnecessary. Using this approach resulted in ESCHER completing the rubble task in 5 min and 53 s.

Figure 25 includes images of the robot stepping up industrial stairs manufactured to specifications of those in the Finals (23 cm × 28 cm). Sequence 3 was captured during toe-off, as the robot pitched the right ankle to extend the effective length of the swing leg. Footsteps were generated

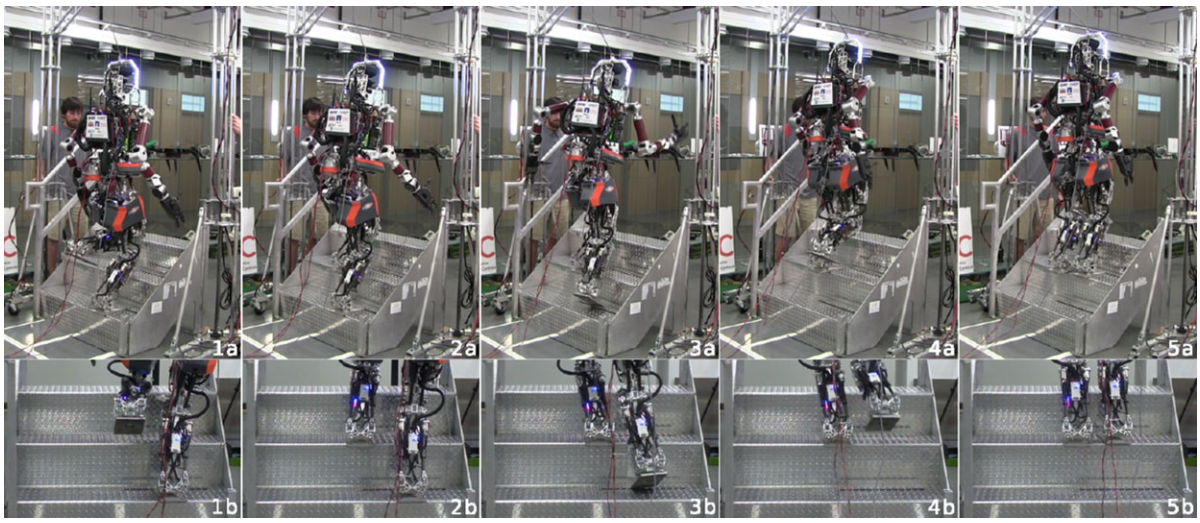
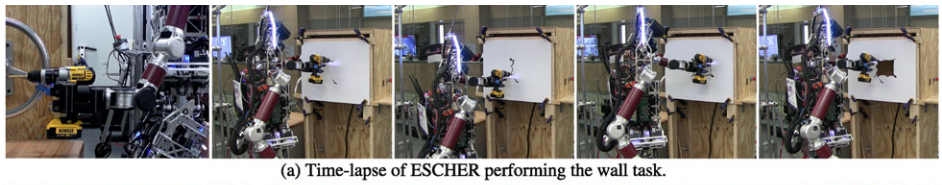
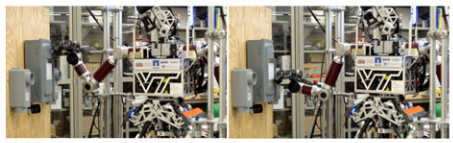


Figure 25. Time-lapse of ESCHER stepping up 23 cm industrial stairs equivalent to the Finals stair task.

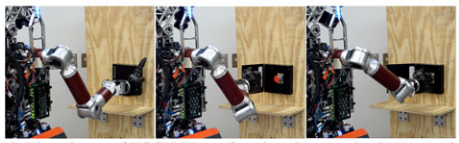


(a) Time-lapse of ESCHER performing the wall task.

(b) Time-lapse of ESCHER performing the surprise plug task.



(c) Time-lapse of ESCHER performing the surprise lever task.



(d) Time-lapse of ESCHER performing the surprise button task.

Figure 26. Time-lapse of ESCHER performing remaining Finals manipulation tasks.

by the user instead of the footstep planner to guarantee precision placement as in the rubble task. It was found that positioning the heel of the left foot approximately 10 cm from the lip of the stair provided a sufficient safety margin on the base of support to prevent the foot from tipping. Table IV lists the average completion times for both rubble and stairs traversal times.

Table IV. Completion times for remaining Finals locomotion tasks.

Task	Time (min)
Stairs	0:41
Rubble	5:53

5.3.2. Manipulation Task Demonstrations

ESCHER was mechanically capable of, and had the software foundation to complete, every manipulation task at the DRC Finals. Following the finals, Team VALOR finished defining grasps and stances for each task and validated ESCHER's manipulation capabilities through completion of the wall cutting and surprise tasks shown in Figure 26.

Although the manipulation operators did not rehearse these tasks for rapid completion, they still achieved reasonable times, as shown in Table V. In each of these tasks,

Table V. Completion times for remaining Finals manipulation tasks.

Task	Time (min)
Wall Cutting	10:54*
Move Plug	5:28
Flip Lever	4:45*
Press Button	4:29

*Time includes approaching the task area.

placing virtual fixtures and verifying grasp quality through camera feedback took the majority of the time. Of particular interest was the wall cutting task; interactions between the wall and whole-body control framework made it difficult to accurately execute arm trajectories. Since the controller's weighting scheme prioritized maintaining stability over tracking end-effector poses, interaction forces between the wall and drill tended to cause the end effector to deviate from the commanded cutting trajectory. Commanding ESCHER's CoM position exploited this prioritization to better achieve straight cuts. The surprise tasks were completed using the virtual fixtures, stances, and grasps developed and tested in simulation before the Finals. These tests validated estimates of capability empirically generated in the team's approach to the Finals.

6. DISCUSSION AND LESSONS LEARNED

Though constrained by a compressed timeline, Team VALOR was able to develop, manufacture, and field a DRC capable humanoid in 12 months through proper reuse of existing technologies combined with simulation and modeling to predict system performance. In house expertise in manufacturing allowed for rapid corrections to design flaws as they became apparent, while software development was accelerated through the use of THOR and simulation environments before the final hardware was complete. Collaborative tools and agile techniques were vital to assigning limited resources to critical areas and mitigating risk. With the short testing time of 43 days, beginning with the completion of the hardware on April 15, software issues were still being discovered at the DRC. The design still performed admirably, walking the 61 m course and having demonstrated the capability to perform all of the DRC tasks.

6.1. Discussion of Results

With the limited testing time, the team set the goal of completing at least one of the tasks during the competition and demonstrating the capability to complete the remaining tasks. Though walking the 61 m course did not score points, the team felt that as one of only two teams to complete this task, it demonstrated the potential of ESCHER and Team VALOR's whole body controller. In testing before the competition, ESCHER successfully climbed a 0.23 m step, opened doors, and turned valves. Given a longer testing period, the team is confident that it would have been capable of reliably scoring several points during the DRC. This section discusses hardware improvements between THOR and ESCHER, locomotion, including the whole body controller and midlevel footstep planning, and thoughts on the competition requirements.

6.1.1. Hardware Improvements

ESCHER exceeds the design requirements of the DRC in computation, battery capacity, and payload, however significant redesign of components originally developed for THOR was required to achieve these results. The ability of the custom linear SEAs to be packaged into a dual-actuator configuration driven by the preexisting motor controllers, along with the experience gained from developing THOR, proved key to producing ESCHER on an accelerated time line. Simulation was used to accurately predict performance, greatly increasing the robustness of the design by allowing confidence in the safety margins designers placed on the system. The DRC Finals and lab testing demonstrated ESCHER's physical ability to complete a wide variety of tasks while operating for extended periods of time.

Initial battery requirements were developed based on a worst-case analysis of simultaneous full power consumption of all electronics onboard the robot. Including a safety factor of 2 led to the selection of a set of large batteries with over 2,200 Wh of energy capacity. However, based on the average consumption results, the team chose to run ESCHER with the smaller 710 Wh test batteries in testing and at the DRC Finals. This reduced the overall mass of the robot by 6.5 kg while maintaining a long run time of 168 min. In addition to the increase in payload capacity, using the smaller batteries reduced the charge time between runs, thus simplifying testing and competition logistics. The ability to use the smaller batteries is attributable to the conservative worst-case analysis during the design phase as well as the efficiency of the robot. The DRC Finals demonstrated ESCHER's ability to operate for over an hour on a single charge, with additional space for larger batteries, with almost three times the energy capacity, offering an extended runtime estimated at 460 min.

6.1.2. Walking Speed

While the current limitation on robot walking speed is due to actuator speed limitations, additional speed can be achieved using more efficient locomotion strategies. Robot walking gaits typically involve highly bent knees, partly to avoid joint singularities, and partly to maintain a constant CoM height throughout the gait. By improving toe-off and developing better plans capable of more natural, oscillatory CoM height trajectories, more straight leg walking will be possible. Additionally, creating more dynamic plans that achieve closer to constant CoM velocity

in the forward direction will require lower joint speeds, as there is less of a "stop and start" motion. When combined with lower torques from straight leg walking, robots will be able to walk more dynamically and quickly using the same hardware than previously possible.

6.1.3. Walking Robustness

Considerable bipedal locomotion research effort has focused on achieving precise, accurate motions to realize stable plans. Humans are capable of fluidly transitioning between robust, imprecise gaits and precision step placement strategies. DRC manipulation tasks require precise alignment of the robot to position the objects within a reachable region. Conversely, traversing the driving course could benefit from relaxed footstep placement approaches to enable stable, high-speed walking. Thus, the ability to actively relax the desired step precision in favor of rapid adaptation could lead to more robust strategies while maintaining the ability to execute precise step plans when needed.

6.1.4. Whole-body Control

A surprising result of manipulation testing on ESCHER was the observation that the presence of low-impedance actuation somewhere in a closed loop between contact points effectively lends some degree of compliance to the entire loop. At the Finals, the HDT arms were operated in velocity control mode due to its better trajectory tracking than impedance mode. Despite the loss of local compliance, the low-impedance lower body provided some degree of global compliance, which helped the platform maintain stability during multipoint contact states.

Whole-body control was shown to be a powerful method for achieving compliant, humanlike motions. The current implementation relies heavily on the inverse dynamics formulation, requiring accurate knowledge of the robot's inertial model. Manipulating objects or carrying payloads can introduce significant inaccuracies to the inertial model, requiring methods to estimate the physical properties of the unmodeled object. Alternatively, development of techniques to compensate for inertial uncertainties can improve the robustness of whole-body control when interacting with poorly characterized objects and environments.

Another limitation of current whole-body control techniques is their inability to consider a future time window of the robot dynamics. The existing implementation minimizes only the instantaneous joint torques and contact forces, which may result in much higher required forces as the motion progresses. While consideration of future forces will make the dynamic performance much better, it comes at the cost of using the full rigid-body equations of motion, rather than the instantaneous linearization possible when only considering the current time step. Current nonlinear optimization techniques for high degree of freedom systems are not currently fast enough for online use. Toward that end, Kuindersma et al. (2016) implemented an LQR cost into their whole-body controller's quadratic program (QP) to consider the future ZMP dynamics, while maintaining fast solve times. Incorporating additional linear and quadratic costs in the QP provides a promising avenue for online whole-body control capable of highly dynamic motions.

6.1.5. Planning versus Pattern Generation

The walk testing results in Section 5.1.2 show a distinct difference in average walking speed based on whether the footstep planner or the footstep pattern generator produced the footstep plan. Distant goal locations often require multiple footstep plans to account for planner limitations and drift while walking. Running the footstep planner multiple times adds extra footsteps to reach precise footholds specified by the user at each intermediate goal. However, while precision at the final goal is desirable, intermediate goals may be less strict in position and orientation. Since long distances in open areas are covered in a piecewise manner, utilizing the footstep pattern generator increases average speed by eliminating the computational time associated with the footstep planner and avoids wasting extra steps. This suggests that instead of planning each individual foothold, a long-distance footstep planner may be more efficient if it plans over a series of footstep patterns. Additionally, as noted in Griffin & Leonessa (2016), planners that incorporate the robot's centroidal dynamics could improve locomotion performance by ensuring that the generated footstep plan is not simply heuristically optimal, but dynamically optimal as well.

6.1.6. Midlevel Autonomy is Sufficient for DRC

The DRC Finals sought to encourage the deployment of autonomous systems by imposing a degraded communications scenario to hinder teleoperation. However, with a sufficiently robust communications system, high-level autonomy was not required. The Comms Bridge developed by Team ViGIR exploited the available network links, discussed in Section 3.8, to enable continuous low-level state information feedback and operator goal transmission. Midlevel planners executing operator assigned goals for locomotion and manipulation tasks proved capable of completing every DRC Finals task and could be executed onboard with no additional intervention from the operator.

6.2. Lessons from the DRC

The members of Team VALOR learned many lessons from preparing and participating in the DRC Finals. A few are presented here to benefit others looking to construct and work with similar systems.

A fundamental realization in developing the software for ESCHER is the importance of common frameworks and existing infrastructures. Rapid development cycles that introduce key enabling technologies are realizable by leveraging previous work. However, this is only effective if there is a common software framework. Bridging between the motion system in Bifrost and the higher-level systems in ROS required significant implementation work. The team utilized a significant number of third-party ROS packages in a relatively short period of time that would have been impossible to implement from scratch given Team VALOR's limited resources. Likewise, having infrastructures in place for knowledge transfer, data storage, and code maintainability increased the team's overall effectiveness. Prior to using a centralized data server, Team VALOR stored test data in an *ad hoc* manner that made processing and review difficult, and increased the likelihood of data loss. Fieldable robots often require immense software systems that integrate numerous subcomponents; however, common frameworks increase code reusability and speed up development work.

With complex systems, it is important that each subsystem responds appropriately and safely to failures of other subsystems. Due to the speed of development, this safety and robustness was not present throughout the entire software system, leaving certain components vulnerable to failure. This had resulted in intermittent issues during testing, but none as severe as the communication failure on the first day of the Finals. While unreliability is undesirable in any system, it can be tolerated in research-grade hardware and software. However, for production of safety-critical systems, it is clear that there is a need to explicitly address the reliability of software and hardware through extensive testing and validation.

With all robots, calibration of sensors, biasing of joint angles, and accurate time stamping of sensor readings are all crucial for perception. For high-DOF robots, any inaccuracies in these details become much more apparent, making operations such as self-filtering and registration of locations between sensor frames less accurate. Even if the robot may be adequately biased in some offline procedure, it is always possible for some physical mishap to occur that may damage joints or bend hardware. Therefore, it seems prudent to have online validation and rebiasing procedures that may be used to identify whether there is an issue, and if possible work to address it.

7. FUTURE WORK

The DRC demonstrated that the software system deployed by Team VALOR needs further improvements in order to take better advantage of the whole body controller's capabilities. While the controller is capable of keeping the robot balanced and enables robust walking, it only reasons over contact points between the robot's feet and the environment. During manipulation tasks, control errors can occur from unmodeled contacts between the hands and the environment, which impose unexpected constraints on the robot's motion. Furthermore, the capability to detect decreasing stability margins while performing actions could be added to the motion system. This capability could be utilized to interrupt higher-level actions and activate recovery actions to keep the robot safe.

The whole body controller constantly adjusts the pose of the robot to maintain balance, even while performing manipulation tasks. The midlevel manipulation planner, which computes arm trajectories to achieve desired end-effector poses and motions, should be modified to take the motion of this floating base into account. Additional work for more reliable manipulation was done in parallel to DRC efforts by Wittenstein (2015), however there was not enough time to integrate it with the system. Wittenstein's work focused on improving door opening reliability by specifying a policy of motions dictated by force feedback interactions to successfully complete the task. By using tactile feedback, his system was more robust to errors in the estimated door location than the affordances employed at the competition.

The team expended a significant amount of software engineering effort simply to ensure that the correct software was launched on the appropriate computers both on and off the robot. A method to automate the allocation of hardware resources to run required software nodes would improve the flexibility and ease of use of the system.

8. CONCLUSION

The DRC was a significant driving force behind the development of robot hardware, software, and sensing technology in the past three years. This work has presented the hardware and software designs of ESCHER, a novel bipedal humanoid platform developed by Team VALOR, a Track A competitor in the DRC Finals. ESCHER is a state-of-the-art, compliant, electrically driven robot designed with sufficient computational power and spare payload to support future research while maintaining an extended two and a half hour run time and a low total weight of 77.5 kg. ESCHER features a custom motion system using a whole body controller, which, together with the compliant lower body, were demonstrated to be capable of the DRC locomotion tasks through simulation and hardware tests. The platform integrated in-house and open-sourced software packages facilitating software collaboration with Team VIGIR, improving algorithms utilized by both teams and enabling semiautonomous operation of ESCHER at the DRC Finals in a short amount of time. Subsystem tests before and after the DRC Finals demonstrated that the system is capable of the DRC tasks, and through several full-system tests meant to mimic the conditions of the DRC finals, Team Valor developed a strategy to employ at the competition.

This work also presents Team VALOR's account of the DRC Finals competition. On the first day, issues related to the degraded communications link between the operators and robot led to a delayed start and a request for a reset half-way through the run. A fault in the onboard communications between a low-level motor controller and the computer hosting the motion system resulted in an incorrect torque at the left hip knocking the robot over and ending the trial. Repairs to the robot made overnight allowed ESCHER to be fielded on day 2 of the competition, where it completed the walking bypass of the driving course in the expected amount of time. ESCHER failed to complete the door task, in part due to the amount of time it took the operators to realize that the wrist encoder was not reporting the correct value. Despite setbacks faced during the Finals, ESCHER is a field-tested platform capable of advanced locomotion and manipulation tasks, including those encountered at the Finals. The development of ESCHER's unique mechanical design and novel controls system has successfully pushed the boundary of humanoid research, particularly the field of bipedal locomotion. Through these developments, robots are on the cusp of capable performance as first responders, enabling fast and safe operation in the life-threatening environments that result from natural and manmade disasters.

ACKNOWLEDGMENTS

This material is supported by the Office of Naval Research (http://dx.doi.org/10.13039/100000006) through Grant No. GS100000006N00014-11-1-0074 and by the Defense Advanced Research Projects Agency (http://dx.doi.org/10.13039/100000185) through Grant No. GS100000185N65236-12-1-1002. We would like to thank Virginia Tech's Department of Engineering for providing support through student funding and travel compensation. We would also like to thank the following people: the members of Team ViGIR for their collaboration, especially David Conner, Stefan Kohlbrecher, and Ben Waxler for taking time to help with documenting and debugging; Derek Lahr, Bryce Lee, Steve Ressler, Joe Holler, and all of the TREC undergraduate volunteers, without whose help the design, assembly, and testing would not have been possible; Ruihao Wang and Sam Blanchard for the collaborative design of ESCHER's covers; and Christian Rippe for providing FEA throughout the chest design process. Lastly, thank you to the additional team sponsors, including the General Motors Foundation, HDT Global, Maxon Precision Motors, NetApp, Rapid Manufacturing, and THK.

REFERENCES

Bry, A., Bachrach, A., & Roy, N. (2012). State estimation for aggressive flight in GPS-denied environments using onboard sensing, in IEEE International Conference on Robotics and Automation (ICRA) (pp. 1–8).

Burton, J. D. (2016). Development and characterization of an interprocess communications interface and controller for bipedal robots.

de Lasa, M., & Hertzmann, A. (2009). Prioritized optimization for task-space control, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 5755–5762).

Dellaert, F. (2012). Factor graphs and GTSAM: A hands-on introduction, Technical Report GT-RIM-CP&R-2012-002, GT RIM.

Diankov, R., & Kuffner, J. (2008). OpenRAVE: A planning architecture for autonomous robotics, Technical Report CMU-RI-TR-08-34, Robotics Institute, Pittsburgh, PA.

Edsinger-Gonzales, A., & Weber, J. (2004). Domo: A force sensing humanoid robot for manipulation research, in IEEE/RAS International Conference on Humanoid Robots (Humanoids) (vol. 1, pp. 273–291).

Englsberger, J., Koolen, T., Bertrand, S., Pratt, J., Ott, C., & Albu-Schaffer, A. (2014a). Trajectory generation for continuous leg forces during double support and heel-to-toe shift based on divergent component of motion, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 4022–4029).

Englsberger, J., Ott, C., & Albu-Schaffer, A. (2013). Three-dimensional bipedal walking control using divergent component of motion, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 2600–2607).

Englsberger, J., Werner, A., Ott, C., Henze, B., Roa, M., Garofalo, G., Burger, R., Beyer, A., Eiberger, O., Schmid, K., & Albu-Schaffer, A. (2014b). Overview of the torque-controlled humanoid robot TORO, in IEEE/RAS International Conference on Humanoid Robots (Humanoids) (pp. 916–923).

Fallon, M., Antone, M., Roy, N., & Teller, S. (2014). Drift-free humanoid state estimation fusing kinematic, inertial and LIDAR sensing, in IEEE/RAS International Conference on Humanoid Robots (Humanoids) (pp. 112–119).

Feng, S., Whitman, E., Xinjilefu, X., & Atkeson, C. (2015). Optimization-based full body control for the DARPA Robotics Challenge, Journal of Field Robotics, 32(2), 293–312.

Griffin, R. J., & Leonessa, A. (2016). Model predictive control for dynamic footstep adjustment using the divergent component of motion, in IEEE International Conference on Robotics and Automation (ICRA).

Grisetti, G., Stachniss, C., & Burgard, W. (2005). Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling, in IEEE International Conference on Robotics and Automation (ICRA) (pp. 2432–2437).

HDT-Global (2015). Adroit mk2 systems robotic arm and manipulator.

Herzog, A., Righetti, L., Grimminger, F., Pastor, P., & Schaal, S. (2014). Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 981–988).

Hirai, K., Hirose, M., Haikawa, Y., & Takenaka, T. (1998). The development of Honda humanoid robot, in IEEE International Conference on Robotics and Automation (ICRA) (vol. 2, pp. 1321–1326).

Hopkins, M., Griffin, R., Leonessa, A., Lattimer, B., & Furukawa, T. (2015a). Design of a compliant bipedal walking controller for the DARPA Robotics Challenge, in IEEE/RAS International Conference on Humanoid Robots (Humanoids).

Hopkins, M., Hong, D., & Leonessa, A. (2014). Humanoid locomotion on uneven terrain using the time-varying divergent component of motion, in IEEE/RAS International Conference on Humanoid Robots (Humanoids) (pp. 266–272).

Hopkins, M., Hong, D., & Leonessa, A. (2015b). Compliant locomotion using whole-body control and divergent component of motion tracking, in IEEE International Conference on Robotics and Automation (ICRA).

Hopkins, M., Ressler, S., Lahr, D., Hong, D., & Leonessa, A. (2015c). Embedded joint-space control of a series elastic humanoid, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany.

Hornung, A., Wurm, K., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). Octomap: An efficient probabilistic 3d mapping framework based on octrees, Autonomous Robots, 34(3), 189–206.

Kajita, S., Hirukawa, H., Harada, K., & Yokoi, K. (2014). Springer tracts in advanced robotics (vol. 101). Berlin-Heidelberg: Springer.

Kaneko, K., Kanehiro, F., Kajita, S., Yokoyama, K., Akachi, K., Kawasaki, T., Ota, S., & Isozumi, T. (2002). Design of prototype humanoid robotics platform for HRP, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 2431–2436).

Knabe, C., Lee, B., & Hong, D. (2014a). An inverted straight line mechanism for augmenting joint range of motion in a humanoid robot, in ASME International Design Engineering Technical Conference (IDETC).

Knabe, C., Lee, B., Orekhov, V., & Hong, D. (2014b). Design of a compact, lightweight, electromechanical linear series elastic actuator, in ASME International Design Engineering Technical Conference (IDETC).

Knabe, C., Seminatore, J., Webb, J., Hopkins, M., Furukawa, T., Leonessa, A., & Lattimer, B. (2015). Design of a series elastic humanoid for the DARPA Robotics Challenge, in IEEE/RAS International Conference on Humanoid Robots (Humanoids).

Koenig, N., & Howard, A. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 2149–2154).

Kohlbrecher, S., Romay, A., Stumpf, A., Gupta, A., von Stryk, O., Bacim, F., Bowman, D., Goins, A., Balasubramanian, R., & Conner, D. (2015). Human-robot teaming for rescue missions: Team ViGIR's approach to the 2013 DARPA Robotics Challenge Trials. Journal of Field Robotics, 32(3), 352–377.

Koolen, T., Smith, J., Thomas, G., Bertrand, S., Carff, J., Mertins, N., Stephen, D., Abeles, P., Englsberger, J., McCrory, S., van Egmond, J., Griffioen, M., Floyd, M., Kobus, S., Manor, N., Alsheikh, S., Duran, D., Bunch, L., Morphis, E., Colasanto, L., Hoang, K. L., Layton, B., Neuhaus, P., Johnson, M., & Pratt, J. (2013). Summary of team IHMC's Virtual Robotics Challenge entry, in IEEE/RAS International Conference on Humanoid Robots (Humanoids) (pp. 307–314).

Kuindersma, S., Deits, R., Fallon, M., Valenzuela, A., Dai, H., Permenter, F., Koolen, T., Marion, P., & Tedrake, R. (2016). Optimization-based locomotion planning, estimation, and control design for the Atlas humanoid robot, Autonomous Robots, 40(3), 429–455.

Kuindersma, S., Permenter, F., & Tedrake, R. (2014). An efficiently solvable quadratic program for stabilizing dynamic locomotion, in IEEE International Conference on Robotics and Automation (ICRA) (pp. 2589–2594).

Lahr, D., Orekhov, V., Lee, B., & Hong, D. (2013). Development of a parallelly actuated humanoid, SAFFiR, in Proceedings of the ASME International Design Engineering Technical Conference (IDETC).

Lee, B. (2014). Design of a humanoid robot for disaster response, Master's thesis, Virginia Polytechnic Institute and State University.

Lee, B., Knabe, C., Orekhov, V., & Hong, D. (2014). Design of a human-like range of motion hip joint for humanoid robots, in ASME International Design Engineering Technical Conference (IDETC).

McGill, S., Brindza, J., Yi, S. J., & Lee, D. (2010). Unified humanoid robotics software platform, in The 5th Workshop on Humanoid Soccer Robots.

McNaughton, M., Baker, C., Galatali, T., Salesky, B., Urmson, C., & Ziglar, J. (2008). Software infrastructure for an autonomous ground vehicle, Journal of Aerospace Computing, Information, and Communication, 5(12), 491–505.

Orekhov, V., Knabe, C., Hopkins, M., & Hong, D. (2015). An unlumped model for linear series elastic actuators with ball screw drives, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Orekhov, V., Lahr, D., Lee, B., & Hong, D. (2013). Configurable compliance for series elastic actuators, in ASME International Design Engineering Technical Conferences (IDETC).

Paine, N., Oh, S., & Sentis, L. (2014). Design and control considerations for high-performance series elastic actuators, IEEE/ASME Transactions on Mechatronics, 19(3), 1080–1091.

Paluska, D., & Herr, H. (2006). The effect of series elasticity on actuator power and work output: Implications for robotic and prosthetic joint design, Robotics and Autonomous Systems, 54(8), 667–673.

Pratt, G., & Williamson, M. (1995). Series elastic actuators, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 399–406).

Pratt, J., Carff, J., Drakunov, S., & Goswami, A. (2006). Capture point: A step toward humanoid push recovery, in IEEE/RAS International Conference on Humanoid Robots (Humanoids) (pp. 200–207).

Pratt, J., & Krupp, B. (2004). Series elastic actuators for legged robots, in Defense and Security (pp. 135–144).

Pratt, J., & Krupp, B. (2008). Design of a bipedal walking robot, in SPIE Defense and Security Symposium (pp. 69621F1–69621F13).

Pratt, J., Krupp, B., & Morse, C. (2002). Series elastic actuators for high fidelity force control, Industrial Robot: An International Journal, 29(3), 234–241.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. (2009). ROS: An open-source Robot Operating System, In ICRA Workshop on Open Source Software (vol. 3, p. 5).

Ressler, S. A. (2014). Design and implementation of a dual axis motor controller for parallel and serial series elastic actuators, Master's thesis, Virginia Polytechnic Institute and State University.

Robinson, D., Pratt, J., Paluska, D., Pratt, G., et al. (1999). Series elastic actuator development for a biomimetic walking robot, in IEEE/ASME International Conference on Advanced Intelligent Mechatronics (pp. 561–568).

Robotnik, R. (2015). Robotis manipulator.

Romay, A., Kohlbrecher, S., Conner, D., Stumpf, A., & von Stryk, O. (2014). Template-based manipulation in unstructured environments for supervised semi-autonomous humanoid robots, in IEEE/RAS International Conference on Humanoid Robots (Humanoids) (pp. 979–986).

Rouleau, M., & Hong, D. (2014). Design of an underactuated robotic end-effector with a focus on power tool manipulation, in ASME International Design Engineering Technical Conference (IDETC).

Saab, L., Ramos, O., Keith, F., Mansard, N., Soueres, P., & Fourquet, J. (2013). Dynamic whole-body motion generation under rigid contacts and other unilateral constraints, IEEE Transactions on Robotics, 29(2), 346–362.

Sensinger, J. et al. (2006). Improvements to series elastic actuators, in IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (pp. 1–7).

Stephens, B., & Atkeson, C. (2010). Dynamic balance force control for compliant humanoid robots, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1248–1255).

Stumpf, A., Kohlbrecher, S., Conner, D., & von Stryk, O. (2014). Supervised footstep planning for humanoid robots in rough terrain tasks using a black box walking controller, in IEEE/RAS International Conference on Humanoid Robots (Humanoids) (pp. 287–294).

Sucan, I., & Chitta, S. (2012). MoveIt!

Tsagarakis, N., Morfey, S., Cerda, G., Zhibin, L., & Caldwell, D. (2013). COMpliant huMANoid COMAN: Optimal joint stiffness tuning for modal frequency control, in IEEE International Conference on Robotics and Automation (ICRA) (pp. 673–678).

Wittenstein, N. (2015). Force feedback for reliable robotic door opening, Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA.

Zhang, J., & Singh, S. (2014). LOAM: Lidar odometry and mapping in real-time, in Robotics: Science and Systems Conference (RSS).