Towards Designing Benchmarks for Humanoid Space Robots: Northeastern's NASA Valkyrie Dataset

---

Velin Dimitrov
Department of Electrical and Computer Engineering
Northeastern University, MA, USA
`v.dimitrov@northeastern.edu`

Murphy Wonsick
Department of Electrical and Computer Engineering
Northeastern University, MA, USA
`wonsick.m@husky.neu.edu`

Xianchao Long
Department of Electrical and Computer Engineering
Northeastern University, MA, USA
`long.x@husky.neu.edu`

Pauline Maurice
Department of Biology
Northeastern University, MA, USA
`pauline.maurice@polytechnique.org`

Dagmar Sternad
Departments of Biology and Electrical and Computer Engineering
Northeastern University, MA, USA
`d.sternad@northeastern.edu`

Taşkın Padır
Department of Electrical and Computer Engineering
Northeastern University, MA, USA
`t.padir@northeastern.edu`

Abstract

Designing benchmarks for algorithms developed to run on physical robot hardware remains to be a challenge. Towards achieving benchmarks for space humanoid robots, we present Northeastern's humanoid robot dataset containing physical sensor data from NASA's Valkyrie (R5), including robot pose estimate, joint angles and velocities, center of pressure, center of mass, ground reaction wrenches, and motion capture ground truth pose. The dataset includes various mobility and manipulation tasks as atomic robot behaviors including walking and reaching motions. Inspired by the NASA Space Robotics Challenge, the dataset is intended for use by the community that wishes to conduct humanoid robot research without direct access to a hardware platform. In addition, it will enable comparative studies in terms of hardware designs, as well as task and motion planning methods. The dataset will provide the humanoid robotics research community with a resource not only to bridge the gap between simulation-based and experimental algorithm validation but also to design task-level benchmarks for humanoid space robots. This paper describes the robot hardware, software, data collection process, post-processing steps, and structure of data for Northeastern's NASA Valkyrie dataset.

1 Introduction

Space robotics has been a vibrant research field ever since the beginning of the human spaceflight program. A variety of satellites, probes, landers, and rovers have explored various celestial bodies within our solar system (Pederson et al., 2015). More recently, NASA's announcement to put American astronauts on the Moons south pole by 2024 and establish sustainable missions by 2028, has generated significant enthusiasm and an urge to act among the government, industry and university research and technology development communities (Anderson and Warner, 2019). However, future manned and robotic space exploration missions pose grand challenges. As a result, significant advances in robot capabilities are required (Miller, 2015). For example, sustainable human exploration missions to Moon and then Mars will be achieved by deploying supplies and equipment in a number of consecutive launch opportunities in advance of the crew's arrival. These predeployed assets (habitats, supplies, ascend vehicles, etc.) will need to be maintained over the years to ensure their availability and functionality for the crew when they arrive to complete their scientific objectives (Drake, 2009; Craig et al., 2015; Bobskill et al., 2015; Drake et al., 2010). Because the predeployed assets are designed to be used by astronauts, a robot in humanoid form-factor is suitable to complete standard and emergency maintenance and support tasks that arise during the course of the mission.

In order to better understand the challenges that need to be addressed to enable the next generation of robotic space missions and to advance the capabilities of humanoid robots, NASA organized the Space Robotics Challenge (SRC) in 2017 (Porter, 2017). The SRC teams competed in a virtual environment to demonstrate humanoid robot competencies with mobility, manipulation, and perception tasks on a simulated Valkyrie robot. Example tasks that the SRC focuses on include: alignment of a communications dish, repair of a solar array, and finding/repairing an air leak in a habitat. These are representative tasks that demonstrate integrated mobility, manipulation and perception frameworks needed to effectively support habitat building, maintenance, and science support operations in unknown and unexplored environments. We provided the winners of the SRC with access to the physical hardware after the challenge. This simulation-to-real experience has motivated the design and creation of the dataset presented here. Many of the approaches and algorithms needed to be modified for testing and operation on the real robot. To the best of our knowledge, there is no other humanoid robot dataset containing proprioceptive sensor data and ground truth collected while robot is executing atomic tasks.

Why Humanoids? As the enabling technologies mature, humanoid robots will find numerous practical applications, personal assistants in the home, nursing assistants in hospitals, first-responders in disaster relief, and as forward deployed space exploration assets. Humans perform well in a diverse class of mobility tasks due to our form, shape and size: tasks such as climbing, crawling, and traversing different terrains including rubble and sand. Bipedal humanoid robots, such as Atlas by Boston Dynamics (Johnson et al., 2016), DRC-Hubo by Korea Advanced Institute of Science & Technology (Jung et al., 2018) and Valkyrie by NASA's Johnson Space Center (Radford et al., 2015), have great potential to perform a variety of complex human-like mobility and manipulation tasks, such as climbing, crawling, traversing rough terrain and reaching behind obstacles as compared to their wheeled or tracked counterparts. This makes them suitable for deployment in cluttered and dynamic environments that typically require making contacts with the surroundings, such as manipulation inside science glove boxes; in spaces specifically designed for humans, such as habitats to support future manned space missions; in operating human tools such as hand drills; and utilizing human interfaces such as switches, plugs, and door handles. The trade-off, however, is the complexity that comes with the high degrees of freedom. For example, the well-studied 2D navigation problem for a mobile robot becomes a multi-faceted research and development effort which incorporates path planning, step-planning, balancing and locomotion control. A bipedal humanoid robot has to maintain balance at all times.

Grand Challenges of Autonomous Humanoid Robots. Our team participated in the 2015 DARPA Robotics Challenge Finals (DeDonato et al., 2017). The DRC Finals was aimed at developing mobility, manipulation, perception and operator interface capabilities for human-robot teams in a simulated disaster response scenario. In a simulated environment, robots performed a variety of manipulation and mobility tasks under human supervision with a 1-hour mission completion time. For greater realism, the communications between the operator(s) and robot was degraded during parts of the mission. Figure 1 presents snapshots from the DRC field and the operator room. The Finals demonstrated that state-of-the-art humanoid robots are slower than humans by an order of magnitude in performing tasks such as turning valves, using hand drills and flipping electric switches. Furthermore, the DRC robots relied heavily on pre-scripted motions and hence lacked autonomy, a critical

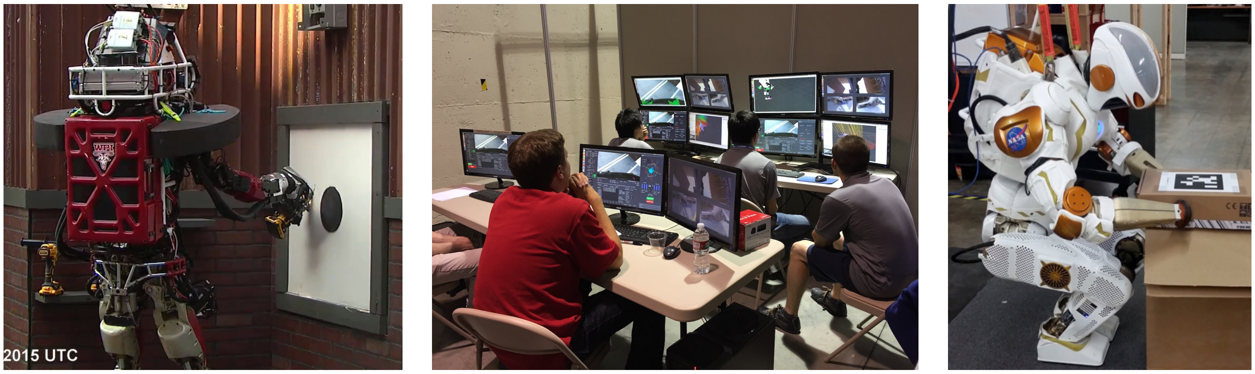

Figure 1: (Left) The Boston Dynamics Atlas robot performing the wall task during the official DRC run. (Center) A view from the operator control room of Team WPI-CMU)(DeDonato et al., 2017) while Atlas is performing the terrain task. (Right) NASA's Valkyrie is performing a simple pick task in a controlled experiment in our labs (Long et al., 2017a).

capability especially in performing tasks relevant to many application domains. We also learned that reliability in completing tasks is prohibitively too low to make humanoid robots practical even for a simplified set of tasks. It is also noted that there are many sources of errors that could lead to catastrophic robot failures. These error sources include sensor errors, controller errors, operator errors, state estimation drift, actuator limits, hardware failures, software bugs, self-collisions, and collisions with the environment. One can argue that many of these factors can be eliminated or predicted and corrected but uncertainties in system dynamics and sensor data could still pose risk of failure during robot operations. As a result, there is a tremendous need to develop new paradigms of autonomy for humanoid robots with a tightly coupled perception and action loops for reliable, and adaptive robot behaviors to perform tasks in dangerous, distant and daring environments.

This work is aimed at bridging research gaps in humanoid space robotics from two aspects: (i) there is a need to design benchmarks and performance evaluation techniques for our inventory of humanoid robots to collaboratively advance the field, (ii) accessibility to a full-size bipedal humanoid robot, due to size, complexity and costs associated with acquisition, infrastructure, and maintenance, remains to be the limiting factor for the research community to systematically validate new algorithms. This paper describes our methodology in creating a dataset for foundational tasks using NASA's humanoid robot Valkyrie and aimed at addressing the benchmarks and accessibility issues by releasing the dataset for the broader research community.

Utilizing NASA's humanoid robot Valkyrie, we have generated a dataset while the robot is performing foundational locomotion (such as stepping) and manipulation (such as reaching) tasks to enable research and development on humanoid robots to stay grounded with respect to the realistic capabilities of the robotic platforms without access to a full-size humanoid robot. The primary goal of the effort is taking a step towards developing methods for designing benchmarks for robotics. This paper describes the hardware and software architecture of the robot, the process for data collection, and the structure of the dataset itself. The intent of providing this information is to help research groups, citizen scientists, and other potential entities interested in working with humanoid robots better understand the limitations and considerations needed to successfully transition from simulation to real hardware. Furthermore, the dataset will provide the community with a resource to compare and contrast different humanoid robots performing simple yet foundational tasks. As a result, it is a step towards designing task-driven benchmarks.

The paper is organized as follows. Section 2 describes the design requirements, robot tasks and motions comprising the dataset, the sensor streams recorded, the hardware and software architecture of the robot, and motion capture system and post-processing details. Section 3 outlines the structure of the dataset, shows some sample data, and discusses potential uses of the dataset. Finally, Section 5 concludes with closing remarks and future directions.

2 Methodology

2.1 Dataset Design Requirements

Northeastern's NASA Valkyrie Dataset has been motivated by the SRC tasks. More specifically, Qualification Task 2 (Harbaugh, 2017), described below has been used to develop a set of requirements.

> **Qualification Task 2:** The second task requires teams to press a button that opens a door, and then walk through the doorway. The button will be located on a wall and will be brightly colored and textured. R5 will start in front of the wall. The doorway will be located next to the button, and will open when the button is pushed.
>
> Successful completion of the Qualification Task 2 will entail: (1)) Pushing the button. (2) Walking through the doorway, where R5 must walk one (1) meter beyond the door without falling.
>
> **Task 2 Qualification Scoring: Qualification score for Task 2 will be determined by the time it takes to complete Task 2. In Task 2, teams completing the task using less time will receive a higher score than teams using more time.**

It should be noted that SRC Qualification Task 1 requiring teams *to find a series of lights on a panel* was a perception only task while Valkyrie standing in front of the panel. Hence, it was deemed to be out of scope for the purposes of our dataset creation methodology.

Using the SRC Qualification Task 2 as a starting point, we enriched the use case by considering the placement of the door and its button in various positions and orientations within the robot's workspace including the case when the door is behind the robot. As a result of this process, the following set of requirements were used to drive the design of experiments and data collection processes. Northeastern's NASA Valkyrie Dataset must include motion and ground truth data while performing following atomic tasks:

1. walking forward for at least 2 meters with a standard step size,
2. walking backwards for at least 2 meters with a standard step size,
3. turning left for 180° with a standard step size,
4. turning right for 180° with a standard step size,
5. reaching to target points while standing using both arms with a specified grid pattern within the robot's workspace.

It is envisioned that the existing of this set of task-based requirements can be used to generate similar datasets for other full body humanoid robots for benchmarking of results. Moreover, new methods validated in simulation by composing these atomic tasks can be tied to data from the physical robot without having a direct access to the platform.

2.2 Data Contained in the Dataset

In view of the requirements presented in Section 2.1 Northeastern's NASA Valkyrie dataset includes data from locomotion and manipulation tasks in the following categories focused on isolating the different functionalities and modalities available on the robot.

- *Lower Body*:
  - Walk forward, backwards, to the left, right, and rotate
- *Upper Body*:
  - Move torso forward, backwards, to the left, right and rotate
  - Reach to target points (grid)
  - Draw a figure 8
- *Full Body*:
  - Pick and place box with both arms

The set of motions is selected to be representative of the capabilities already developed on the robot. Table 1 presents the parameters used for each motion. Parameters were experimentally derived to ensure that experiments could be run in both real-world and simulation to allow for direct comparison.

Sensor data including proprioceptive data are logged internally on Valkyrie during every run. After processing these logs, we provide the following time-synchronized data streams in Matlab structure format:

- Robot pose estimate from the state estimator
- Joint displacements, velocities, torques

Table 1: Parameters used for experiments.

Motion	**Specs**
Walk Forward/Backwards	Number of Steps: 8 Step Size: 0.25 m Transfer Time: 1.5 sec Swing Time: 1.5 sec Feet Separation: 0.21 m
Side Steps Right/Left	Number of Steps: 10 Step Size: 0.15 m Transfer Time: 1.5 sec Swing Time: 1.5 sec Feet Separation: 0.1 m
180 Turn Right/Left	Number of Steps: 12 Step Size: 0.15 m Transfer Time: 1.5 sec Swing Time: 1.5 sec Feet Separation: 0.277 m
Pelvis Movement	Roll Range: -0.2 to 0.2 rad Pitch Range: 0 to 0.5 rad Yaw Range: 0.5 to -0.5 rad
Grid Right/Left	Size: 0.6 m x 0.3 m Points: 13
Figure 8 Right/Left	Size: 0.4 m x 0.4 m

- Estimated center of pressure (CoP), center of mass (CoM), ground reaction wrenches (force/torque)
- Pose ground truth for arms, feet, torso, and pelvis

Position and velocity ground truth for both lower and upper body tasks is provided through an external motion tracking system recording reflective markers placed on known landmarks. As the robot completes the tasks, the robot configuration is calculated and logged. Torque ground truth is not directly available, but before each experiment the robot is placed in a known configuration and gravity-compensation mode. This configuration allows any biases present in the series-elastic actuators to be nullified.

2.3 Valkyrie Hardware and Software

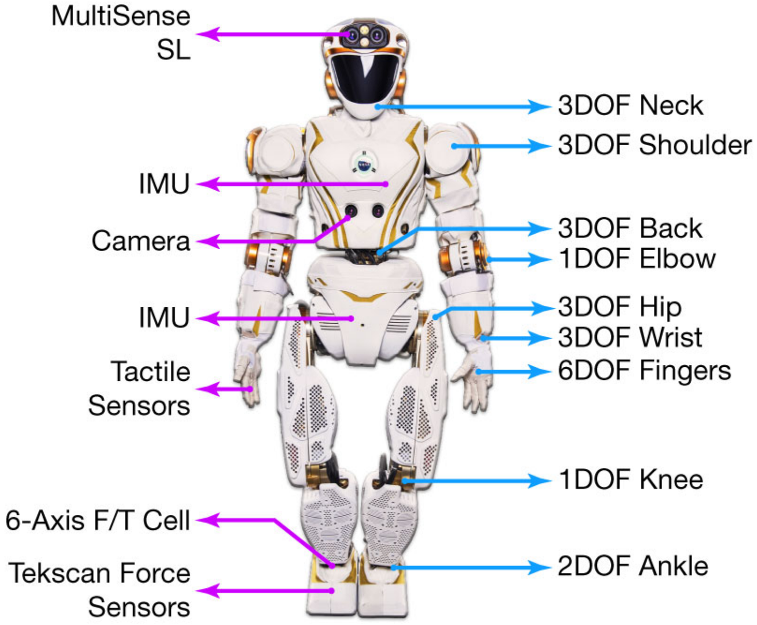

Figure 2: A hardware diagram showing the location of all the sensors and joint actuators on the Valkyrie robot. Valkyrie has a total of 44 degrees of freedom (DOF) with 6 DOF legs, 7 DOF arms, 3 DOF back, and 3 DOF neck. An additional 6 DOF are available on the fingers in each hand.

Valkyrie, shown in Figure 2, is a 44 degree of freedom (DOF) humanoid robot originally designed to compete in the DRC Trials in December 2013 (Radford et al., 2015). The robot consists of 5 major mechanical sub-assemblies: two arms, two legs, and a torso. Valkyrie's arms and leg flanges are connected to flanges on the torso through a Marman band clamp. The clamps have a pair of flats cut into them ensuring repeatable alignment for accurate forward kinematics after calibration. The arms consist of a total of 7 DOF arranged in a 3 DOF shoulder, 1 DOF elbow, and 3 DOF wrist. The first 5 DOF of each arm are implemented through series-elastic actuators (SEA) (Paine et al., 2015) enabling the implementation of torque control by measuring spring deflection with pre- and post-spring encoders. The 2 wrist DOF are implemented with a lead-screw on a linear guide assembly.

Each leg consists of a total of 6 DOF arranged as follows: 3 DOF hip, 1 DOF knee, and 2 DOF ankle. The first 4 DOF of each leg are also implemented with SEAs, while the last 2 DOF are implemented with lead-screw actuators similar to the wrist. Torque control of the ankle roll and pitch assembly is implemented with two load cells on the mechanical linkages of the last two joints. In addition to the joint-level torque sensing, ATI Mini58 6-axis force/torque (F/T) sensors mounted in the ankles provide data to calculate wrenches representing ground reaction forces and center of pressure (CoP). Finally, embedded in each foot is an array of Tekscan force sensors that after a firmware update can be used in future work for more accurate ground sensing, especially over uneven terrain.

There are 6 DOF in the torso, split between a 3 DOF back that enable the torso to rotate and lean, and a 3 DOF neck allowing the robot to turn and tip the head. Two IMUs are located in the torso, one near the pelvis and other near the left shoulder. Both IMUs can be used independently or jointly to estimate the pose of the torso. Finally, two cameras in a stereo pair arrangement are mounted in the stomach which can be used as hazard avoidance cameras or to provide a low-angle perspective for manipulation tasks.

Valkyrie's head consists of a Carnegie Robotics Multisense SL sensor unit combining a rotating Hokuyo LIDAR and a stereo camera pair. An integrated FGPA in the Multisense SL processes the stereo camera data and provides registered point clouds from the camera data and LIDAR data. Integrated fully-dimmable LED lights can be used to illuminate the scene. In addition, the LIDAR rotation speed can be varied, trading off denser point clouds for longer scan times.

Valkyrie has two on-board computers: Link and Zelda. They are single board computers (SBC) with Intel i7-3615QE at 2.3 Ghz paired with 16GB of DDR3 1600 and a 240GB SSD. All of that is on a Congatec BS77 Type2 COM Express module, and the EFK XV1 carrier board provides peripherals. Both computers run Ubuntu 14.04 Server edition, and Link, which runs the onboard low-level controllers, additionally has real-time kernel patches to ensure controller performance. In addition, it interfaces with the Synapse drivers implementing the custom LVDS Robonet protocol to communicate with the motor drivers and sensor boards throughout the robot. In general, Link only runs the low-level controllers, and Zelda runs any ROS nodes that are needed on the robot, including handling the Multisense SL sensor data.

The Institute for Human Motion and Cognition (IHMC) has integrated Valkyrie's model into the Simulation Construction Set (SCS), the simulator built specifically for testing humanoid walking performance and used for the DRC by Team IHMC. In addition, IHMC integrated the walking controller used on the Atlas robot during the DRC on the Valkyrie robot (Johnson et al., 2015) (Koolen et al., 2016). The SCS simulator, along with the Valkyrie models, is open source. Development on Gazebo support is a work in progress, lagging behind the performance of SCS, but allows for manipulation-based research to be conducted in simulation. SCS is not structured to support simulation of dexterous manipulators, and therefore is limited to mobility simulation. In addition, Gazebo allows the leveraging of previously available code and approaches from the DRC. For those reasons, the SRC and our simulated dataset uses the Gazebo simulator.

Valkyrie has a ROS master running on the Zelda computer which allows for simplified development on the robot utilizing the standard ROS tools, messages, and services. Controllers, such as the custom forearm controller, use the ROS control framework and can be loaded like any other controller in ROS. Coupled with the IHMC walking controller, users can queue footstep messages in ROS to make the robot walk, arm pose messages to control the arms either in joint space or Cartesian space, and full body messages for control of all the joints simultaneously. The IHMC walking controller provides relevant information for balancing such as the ground contact forces as wrenches in the foot frame.

Sensor data are also provided through the standard ROS messages. In addition, effort and position information is provided for each SEA joint in order to exert precise forces for prehensile and non-prehensile manipulation. The Multisense SL data are passed through without modification from the embedded computers and is provided in appropriate ROS topics by the standard Multisense driver.

2.4 Ground Truth Measurement System

To obtain a ground truth measurement for the pose of the robot pelvis, feet, and arms, we used an optical motion capture system from Qualisys (Göteborg, Sweden). The system consisted of 12 Oqus type cameras distributed all around the robot workspace, at a height between 2.6 and 3.5 m. One camera was occluded by the robot's safety gantry, meant to catch the robot in case of falls. Therefore, only 11 cameras were utilized for the dataset. This setup was nevertheless highly redundant since each marker only requires to fall within view of 2 cameras to be recorded. At the beginning of each session of data collection, the cameras were calibrated according to the Qualisys standard calibration procedure.

The motion capture data were recorded at 100 Hz. In order to synchronize the motion capture data with the data from the robot sensors, the motion capture acquisition was triggered by an external analog signal (5 V TTL rising edge). The signal was triggered by the operator computer which recorded the Unix time at the start of the trial. The IHMC logger, which logs all the internal robot state information, ran on a dedicated network attached storage (NAS). Both the operator computer and NAS were synchronized using network time protocol (NTP), and the recorded start time was used to calculate the time offset between the ground truth from the motion capture system and the logged robot data in post-processing.

2.5 Motion Planning and Control

The GitHub repository that contains the dataset, described in detail later, also contains the scripts used for the mobility motions. The scripts send footstep messages to the IHMC controller to move and place the robot feet in each task. The specific details of the walking parameters are contained in the corresponding script. For example in the forward and backward walking motions, we utilize transfer times of 1.5 seconds, swing times of 1.5 seconds, step sizes of 0.25 meters, and the distance between the feet is 0.21 meters. These values are experimentally derived to work well in both simulation and on the real robot so that direct comparisons are possible.

The manipulation motions are generated utilizing our framework which combined whole body motion planning and control, first utilized on the Atlas robot in the DRC and now modified onto Valkyrie. We utilize this constrained motion planning framework to generate the trajectory for the figure 8, grid, and box pick and place manipulation motions. The details of the approach from (Long et al., 2016; Long et al., 2017b) are summarized next for the sake of completeness.

The framework can calculate a sequence of collision-free robot configurations efficiently by solving a trajectory optimization problem which is constrained by the kinematics requirements of the task and the collision avoidance conditions. The details of the problem formulation of the IK engine and the constraint setups can be found in (Long et al., 2016).

The objective of the motion planing problem, which contains a sequence of T joint configurations representing motion trajectory of a K DoF humanoid robot system as decision variables $\mathbf{q}_{1:T}$, where $\mathbf{q}_t \in \mathbb{R}^K$ describes the robot joint configuration at the t-th time step, has the following form:

$$\begin{aligned} f(\mathbf{q}_{1:T}) = \sum_{t=1}^{T} ((\mathbf{q}_{t+1} - \mathbf{q}_t)^\top Q_1 (\mathbf{q}_{t+1} - \mathbf{q}_t) + (\mathbf{q}_t - \\ \mathbf{q}_{nom})^\top Q_2 (\mathbf{q}_t - \mathbf{q}_{nom}) + \Delta\mathbf{d}(\mathbf{q}_t)^\top Q_3 \Delta\mathbf{d}(\mathbf{q}_t)) \end{aligned} \tag{1}$$

where $Q_1, Q_2, Q_3 \geq 0$ are weight matrices, $\mathbf{q}_{nom}$ represents a nominal posture, and $\Delta\mathbf{d}(\mathbf{q}_t)$, ordered as $[\Delta x(\mathbf{q}_t), \Delta y(\mathbf{q}_t), \Delta z(\mathbf{q}_t), \Delta roll(\mathbf{q}_t), \Delta pitch(\mathbf{q}_t), \Delta yaw(\mathbf{q}_t)]$, is the Cartesian deviation between a link's pose at the robot state $\mathbf{q}_t$ and its desired posture. These quadratic cost terms represent penalizations of the weighted sum on the joint displacements between the waypoints, joint configuration deviation from a nominal posture and Cartesian displacement from a link frame to a desired target frame. The first term can limit the movement of the robot and smooth the trajectory. The second term is used to push the joints to the nominal configuration when all the constraints have been met. Similarly, the third term is used to push links to specific positions and orientations.

Many constraints can be specified on the robot's motion, which range from simple joint limits, to position/orientation constraints of the robot's links, to collision avoidance constraints, to constraints on keeping the horizontal projection of CoM on the support polygon. The Cartesian posture constraint plays an important role for generating a motion solution.

A Cartesian posture constraint can be established through the following two formulations:

- Type 1 is an equality constraint which can fix the link at a specific pose:

  $$diag(c_1, c_2, ..., c_6)\Delta\mathbf{d}(\mathbf{q}_t) = \mathbf{0} \tag{2}$$

  where $diag(c_1, c_2, ..., c_6)$ is a 6x6 diagonal matrix with scaling factors from c_1 to c_6. Eqn. 2 can eliminate the posture displacement. Multiplying $\Delta\mathbf{d}(\mathbf{q})$ with a diagonal matrix allows us to relax some position and orientation constraints by setting the corresponding entry to 0.

- Type 2 is an inequality constraint which limits a link's movement range:

  $$\mathbf{A}\mathbf{d}(\mathbf{q}_t) - \mathbf{b} \leq \mathbf{0} \tag{3}$$

  where $\mathbf{d}(\mathbf{q})$ is the pose of the target link. $\mathbf{A}$ and $\mathbf{b}$ are decided by the link's movement range parameters. To allow the robot to grab a box with both arms while solving motion planning problem, a kinematic reachability region for the arm placement needs to be defined based on the dimension of the box.

  $$a_i x(\mathbf{q}_t) + b_i y(\mathbf{q}_t) + c_i z(\mathbf{q}_t) + d_i \leq 0, i = 0, 1, 2, 3 \tag{4}$$

  where a_i, b_i, and c_i are scalars which can be computed according to the vertex parameters of the reachability region. The yaw angle of the arm can range from θ^- to θ^+. This limit can be formed as:

  $$yaw(\mathbf{q}_t) - \theta^+ \leq 0, and \ \ \theta^- - yaw(\mathbf{q}_t) \leq 0 \tag{5}$$

  All the inequalities in (4) and (5) can be composed into the left arm inequality constraint with the form (3).

Therefore, for implementation on the Valkyrie robot, a variety of costs and constraints need to be set to generate a feasible motion solution. There are a number of general costs and constraints, such as joint displacement costs using a normal standing pose as the nominal pose, pelvis height and orientation costs, torso orientation costs, joint limit constraints, end-effector target constraints, collision avoidance constraints, and CoM constraints.

The figure 8 and grid reaching tasks can be accomplished with Type 1 constraints, since they require just one arm's motion. The box pick and place requires an additional Type 2 constraint on the other arm so the arm can rotate with respect to the box while also maintaining a clamping force in order to lift the box. The selection of the specific costs for each task consists of starting with a known set of nominal costs, and then experimental tuning based on multiple runs of each task.

2.6 Data Collection and Processing

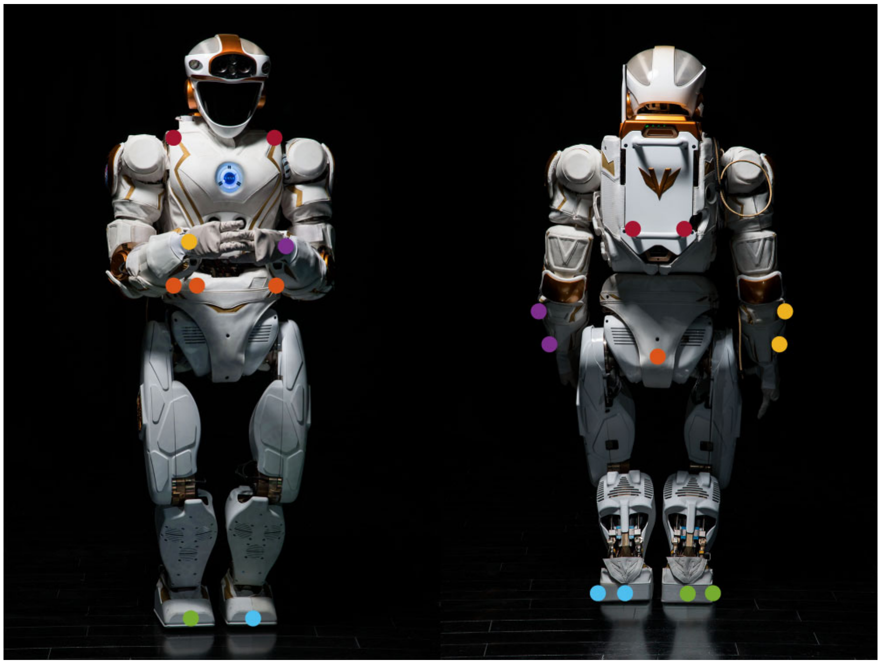

Figure 3: Positions of the reflective markers on the robot pelvis(orange), forearms(yellow/purple), feet(green/cyan), and back(red). The colors correspond to the same ones in Figure 4 which show the data from the forward walking trial.

Spherical reflective markers (12.7 mm diameter) were attached to Valkyrie's torso, feet and arms. In all movements, 4 markers were placed on Valkyrie's pelvis, 3 markers on each foot, and 3 markers on each forearm. In the box pick-up movement, four additional markers were placed on Valkyrie's torso. Fig. 3 depicts the marker positions on the robot. All markers were placed on unique landmarks of Valkyrie's body so that their position in the robot frame were accurately determined from the robot CAD model.

All markers were tracked 100 % of the time, except in the box pick-up movement where one of the pelvis marker was partly lost. However, given that four markers were placed on the pelvis, the pelvis pose could still be determined without ambiguity.

The same procedures for data collection and processing were followed for each individual manipulation and mobility trial. Valkyrie was suspended from a gantry placed inside the workspace of the motion capture system. One full week was scheduled in order to conduct the tests to allow ample time for system setup and data collection. The dataset itself was recorded over the span of two days after all procedures were thoroughly tested and practiced.

Each motion started with the robot initialization where the robot computers are booted and the IMUs are powered and perform their built-in bias compensation. The actuators were then enabled, and the robot was placed in a bias compensation mode. In this mode, the robot hangs suspended from the safety harness in a known configuration with the feet off the ground. The measured torques on each joint are compensated given the expected torque due to gravity. This procedure removes any torque biases that have accumulated in the actuators.

After the gravity compensation was performed, the robot feet were placed on the ground, and the gains on the IHMC controller were slowly raised bringing the robot to a standing state. At that point, the recording was ready to begin. The operator started the script that controls the robot throughout the experiment. This script also triggered the time synchronization system described above. While the robot performed the movement tasks, the motion capture system logged ground truth data, and the NAS logged all robot state information. All the scripts used to generate the movements are provided in a GitHub repository, described in detail later.

After the experiment concluded, both the motion capture and robot logger were stopped, and the raw data were saved for later post-processing and checked for corruption. Additionally, the motion capture data were checked for marker visibility. If any marker disappeared for a significant period of time, the experiment was rerun. Markers disappear for a variety of reasons out of our control such as reflections and occlusions, but for this dataset, almost every marker was completely visible for the entirety of the experiment. Good marker positions that lead to good visibility throughout the experiments were determined through testing, shown in Figure 3, and thus only a few experiments had to be rerun.

The post-processing of the data consists of five distinct steps: (i) housekeeping to get all the different data streams together into structs, (ii) calculating the appendages world frame positions based on the marker positions, (iii) time synchronizing the motion capture streams and internal robot logging, (iv) finding the rotation between the motion capture world frame and the robot's internal world frame, and finally (v) rotating all the robot data so the ground truth and robot data can be directly compared.

The first step is conducted by a Matlab script called *preprocess_data.m* included in the corresponding Github repository for the dataset. This script repacks all the robot logged data into a struct called *valkyrie* and all the marker positions into a struct called *mocap*. In addition, it pulls in the start time recorded by the time synchronization setup, and initializes the Q matrices used by the Kabsch algorithm in later steps. The script is commented with how these Q matrices change for a couple experiments as some markers on the back and pelvis were occluded by the robot motion itself, and thus discarded for that specific experiment.

The second script, *process_data.m*, implements the next four steps. The second step utilizes the Kabsch algorithm (Kabsch, 1976; Kabsch, 1978) (Kabsch.m[1]) to calculate a ground truth for the pelvis, right arm, left arm, right foot, and left foot robot frames. The algorithm calculates the optimal rotation matrix between paired sets of points. In this case, the Q matrices represent the distances between markers and the tracked robot frames. The Kabsch algorithm calculates where the frame is based on the two distances: one from the motion capture system, and one measured from the CAD model.

---

[1] `http://www.mathworks.com/matlabcentral/fileexchange/25746-kabsch-algorithm`

The third step in the post-processing consists of time synchronizing the internal robot data streams with the motion capture data. The IHMC logger logs the Unix time stamp as the experiment runs, and the start time recorded. The script finds the index corresponding to the starting time stamp in the robot logged data, and trims all data from before the experiment was triggered. It also trims the end of the internal robot-logged data to match the end of the motion capture data.

Once the data consist of the tracked frames and are time-synchronized, the rotation between the robot sense of the world frame and the motion capture world frame needs to be calculated. This allows for the evaluation of the internal robot state estimator. We utilize *absor.m*[2] which is an implementation of Horn's quaternion method (Horn, 1987). The algorithm finds the optimal rotation, translation, and scaling that aligns two separate but dependent data streams. In our case, we only utilize the optimal rotation to align the frames. The initial bias is removed between the beginning of both data streams manually. The optimal translation is not used, because the two data streams should start from a single point. The optimal translation would align the two streams in order to minimize the bias between the two streams which is not beneficial in this instance. The scaling is also discarded since it calculates the trivial solution, an identity scaling since both data streams use the same scale.

Finally, all the internal robot data, given in the robot's world frame, are rotated to match the ground truth world frame using the optimal rotation calculated in the previous step. There is a second struct that holds robot data called *valkyrie_dec*. The struct is that same as *valkyrie* with the exception that it was decimated to match the 100 Hz of the motion capture system. The *valkyrie* struct has the finer grained data recorded at the nominal 500 Hz by the robot logger. Lastly, the script plots the pelvis ground truth and the robot state estimator as a qualitative check to confirm all the rotations and fits were done well. The next section describes the structure of the datasets in greater detail.

3 Access to Dataset and Associated Tools

3.1 Live Repository

A GitHub repository holds all the scripts used for post-processing the data, sample scripts to visualize the dataset, and an issue tracker where questions and concerns about the dataset are answered (`https://github.com/RIVeR-Lab/neu_valkyrie_dataset/`). In addition, instructions for accessing the dataset are included. The datasets consist of Matlab data files named after the particular experiment run in that dataset. Once imported, each data file will expand into 6 structs and 1 parameter.

- *gt*: The ground truth from the motion capture system from Kabsch algorithm (*Kabsch.m*)
  - U: 3x3 orientation matrix of the frame

---

[2] `https://www.mathworks.com/matlabcentral/fileexchange/26186-absolute-orientation-horn-s-method`

  - *r*: 3x1 position vector of the frame
  - *lrms*: scalar least root mean square
- *mocap*: 3xn matrices holding 3D position values of n markers on each appendage
  - Marker positions in world frame
- *Q*: 3xn matrices holding 3D position values of n markers on each appendage
  - Marker position in appendage frame
- *regParams*: Output parameters from Horn's method (*absor.m*)
  - *R*: 3x3 rotation matrix
  - *t*: 3x1 translation vector (not used)
  - *s*: scalar scaling factor (not used)
  - *M*: 4x4 homogeneous transformation matrix (not used)
- *valkyrie*: Robot logged data
  - *timestamp*: Unix time
  - *robot_time*: Robot time
  - *cop*: 2x1 center of pressure position vector
  - *com*: 3x1 center of mass position vector
  - *state_estimator*: 3x1 robot state estimate of pelvis frame
  - *grf*: 3x1 ground reaction force and torque vectors
  - *tau*: scalar joint torque values
  - *q*: scalar joint positions
  - *qd*: scalar joint velocities
  - *pelvis_imu*: acceleromter and gyroscope data from pelvis IMU
- *valkyrie_dec*: Robot logged data decimated to 100Hz
  - Same as *valkyrie*

3.2 Performance Analysis for a Use Case

Motivated by the NASA Space Robotics Challenge tasks, Northeastern's NASA Valkyrie Dataset contains data from a rich set of atomic tasks performed by the robot. In order to illustrate its utility we present a use case on how to obtain performance metrics for Valkyrie from the dataset. Figure 4 depicts various plots obtained using the data from the walking forward task. The ground truth plots corresponding to robot's feet, arms, torso and pelvis are sketched using the motion capture data. In addition state estimator output for the pelvis position and orientation is included. A video corresponding to this plot is presented in the Github repository and it demonstrates the robot motions from a data-centered perspective. In order to evaluate the performance of the default state estimator, the state estimator error has been calculated at the pelvis for the duration of the task. Figure 5 shows the evolution of the Euclidean distance between the state estimator (*valkyrie_dec.state_estimator*) and the pelvis ground truth (*gt.r.pelvis*) changes as the experiment progresses. Quantitatively, it is observed that the total error reaches to ≈ 55 mm at the end of the task when the robot walked a distance of 2 m. Qualitatively it is observed that there are dynamics not well accounted

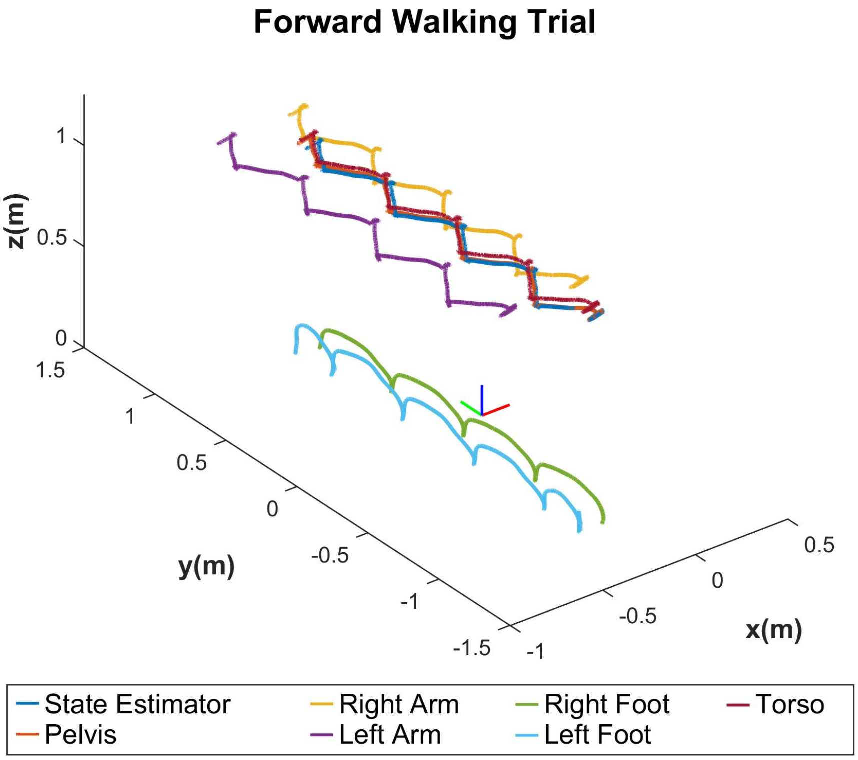

Figure 4: A visualization of the ground truth from the forward walking trial. The ground truth data for the pelvis (orange), forearms (yellow/purple), feet (green/cyan), and torso (red) is shown, along with the robot state estimate (blue). The markers on the back were only used in the box pick-up movement.

for by the state estimator during the large spikes where the most error accumulates. Future improvements of the state estimators can be tested on the dataset, and the improvement can be quantitatively compared to the current state of the art.

4 Lessons Learned

This data design and collection effort aimed at developing task-based benchmarks for full-size humanoid robots provided the research team with invaluable insights. For the sake of completeness, we present lessons learned from this effort.

General guidelines for humanoid designs can be developed through collaborative benchmarking. Generation of datasets with the goal of achieving research benchmarks provide significant insights into designs of robot hardware, software, and interfaces. Fostering a research community centered around benchmarking with standard robot platforms will potentially improve competencies of robot hardware through iterative design processes.

Sensor placement and selection is a critical design consideration. Through

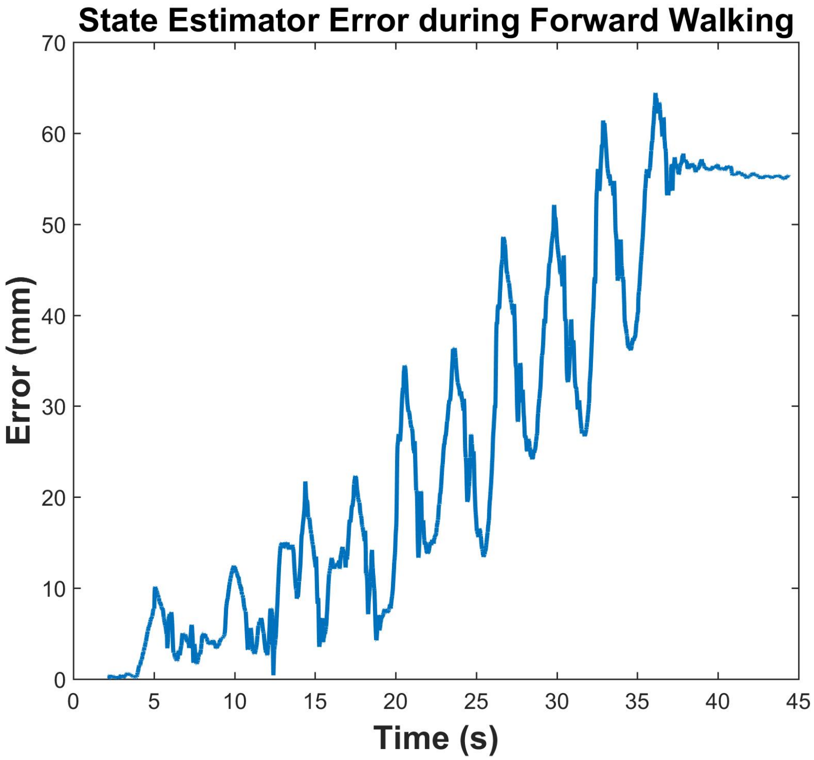

Figure 5: A graph showing the accumulated position error in the robot state estimator in the forward walking trial. The Euclidean distance between the state estimator and pelvis ground truth is calculated and plotted as a function of time.

the Valkyrie (R5) design iterations, sensors have been both upgraded and removed to meet the need of the software and to reduce complexity of the robot for redundant sensors. For humanoid robots, addition of cameras at the knees, wrists and torso, as well as inertial sensors at each limb will enable newer capabilities for benchmarking but this comes at the cost of increased complexity. Minimalistic yet redundant sensor suites will facilitate benchmarking without relying on an infrastructure for ground truth data collection.

Design of experiments for complex robot platforms can reveal design iterations to hardware. Overall, Valkyrie is a capable platform for realizing generalized movements, however, there are trade-offs that need careful attention. For example, iterations of Valkyrie designs had different knee specs, one was faster with lower torque allowing the robot to move and react quickly, but unable to step-up on a standard stair, while another had increased torque with slower speeds to allow for step-up behavior, but decreased the walking speed of the robot. These trade-off are usually discovered experimentally as the robot is programmed to perform progressively complex tasks.

Super-human capabilities can potentially generate humanoid robot perfor-

mance that can match human performance. Valkyrie is designed to be a generalized humanoid platform. However, there are potentially additional design elements that could be added to improve the platform. For example, the current iteration of Valkyrie has two sets of "forearms" that can be manually changed. One forearm is the actual forearm with a 3-DOF wrist and hand (with 3 fingers and a thumb) while the other is a mass simulator that mimics the size and weight of the actual forearms, but is simply a weight with a rubber end tip. The idea behind the mass simulators is to allow for walking testing to be completed without the need to worry about damaging the actual forearms, but due to the simplicity of the design and robustness of the mass simulators they can be used for other simple operations as well that the hands are not designed for (i.e. punching a wall). Therefore, one potential capability to add to Valkyrie is to design a way to allow the robot to switch between the tools without the needs for physical human intervention, as well as, identify and design more useful tools. Industrial robots benefited from this feature greatly.

There is much more to do in terms of achieving research benchmarking. Humanoid robots provide a rich set of interactions with their environments. Humanoid robot datasets need to be designed and generated for walking over rough or deformable terrain, tasks involving multi-contacts such as climbing stairs, and tasks that require dexterous manipulation such as repairing an instrument.

5 Conclusion

We envision that the dataset will be used by researchers who do not have access to a full-size humanoid robot. They can better understand the realistic dynamic response of the system performing various tasks, which is difficult to accurately simulate. In addition, the dataset will provide a basis for comparing the Valkyrie robot and other humanoid robots. The set of experiments are described in detail to reproduce them in future to enable direct comparison of the performance. Finally, we envision the dataset will allow researchers to test new state estimation and humanoid localization without physical access to a robot.

Acknowledgments

This material is based upon work supported by the National Aeronautics and Space Administration under Grant No. NNX16AC48A issued through the Science and Technology Mission Directorate, by the National Science Foundation under Award No. 1451427, 1637854, 1701023, and by the Department of Energy under Award Number DE-EM0004482.

References

Anderson, G. and Warner, C. (2019). NASA taps 11 American companies to advance human lunar landers. `https://www.nasa.gov/press-release/nasa-taps-11-`

`american-companies-to-advance-human-lunar-landers/`. Accessed: 2019-05-26.

Bobskill, M. R., Lupisella, M. L., Mueller, R. P., Sibille, L., Vangen, S., and Williams-Byrd, J. (2015). Preparing for Mars: Evolvable Mars Campaign "Proving Ground" Approach. In *2015 IEEE Aerospace Conference*, pages 1–19.

Craig, D. A., Herrmann, N. B., and Troutman, P. A. (2015). The evolvable mars campaign - study status. In *2015 IEEE Aerospace Conference*, pages 1–14.

DeDonato, M., Polido, F., Knoedler, K., Babu, B. P. W., Banerjee, N., Bove, C. P., Cui, X., Du, R., Franklin, P., Graff, J. P., He, P., Jaeger, A., Li, L., Berenson, D., Gennert, M. A., Feng, S., Liu, C., Xinjilefu, X., Kim, J., Atkeson, C. G., Long, X., and Padir, T. (2017). Team wpi-cmu: Achieving reliable humanoid behavior in the darpa robotics challenge. *Journal of Field Robotics*, 34(2):381–399.

Drake, B. G., editor (2009). *Human Exploration of Mars Design Reference Architecture*. NASA.

Drake, B. G., Hoffman, S. J., and Beaty, D. W. (2010). Human exploration of mars, design reference architecture 5.0. In *Aerospace Conference, 2010 IEEE*, pages 1–24.

Harbaugh, J. (2017). NASA STMD Space Robotics Challenge. `https://www.nasa.gov/directorates/spacetech/centennial_challenges/space_robotics/index.html`. Accessed: 2019-05-26.

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642.

Johnson, M., Shrewsbury, B., Bertrand, S., Calvert, D., Wu, T., Duran, D., Stephen, D., Mertins, N., Carff, J., Rifenburgh, W., Smith, J., Schmidt-Wetekam, C., Faconti, D., Graber-Tilton, A., Eyssette, N., Meier, T., Kalkov, I., Craig, T., Payton, N., McCrory, S., Wiedebach, G., Layton, B., Neuhaus, P., and Pratt, J. (2016). Team ihmc's lessons learned from the darpa robotics challenge: Finding data in the rubble. *Journal of Field Robotics*, pages n/a–n/a.

Johnson, M., Shrewsbury, B., Bertrand, S., Wu, T., Duran, D., Floyd, M., Abeles, P., Stephen, D., Mertins, N., Lesman, A., Carff, J., Rifenburgh, W., Kaveti, P., Straatman, W., Smith, J., Griffioen, M., Layton, B., de Boer, T., Koolen, T., Neuhaus, P. D., and Pratt, J. E. (2015). Team ihmc's lessons learned from the DARPA robotics challenge trials. *J. Field Robotics*, 32(2):192–208.

Jung, T., Lim, J., Bae, H., Lee, K. K., Joe, H.-M., and Oh, J.-H. (2018). Development of the humanoid disaster response platform drc-hubo+. *IEEE Transactions on Robotics*, 34(1):1–17.

Kabsch, W. (1976). A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32:922–923.

Kabsch, W. (1978). A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 34:827–828.

Koolen, T., Bertrand, S., Thomas, G., de Boer, T., Wu, T., Smith, J., Englsberger, J., and Pratt, J. E. (2016). Design of a momentum-based control framework and application to the humanoid robot atlas. *I. J. Humanoid Robotics*, 13(1).

Long, X., Wonsick, M., Dimitrov, V., and Padır, T. (2017a). Anytime multi-task motion planning for humanoid robots. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 4452–4459. IEEE.

Long, X., Wonsick, M., Dimitrov, V., and Padir, T. (2017b). Anytime multi-task motion planning for humanoid robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Long, X., Wonsick, M., Dimitrov, V., and Padr, T. (2016). Task-oriented planning algorithm for humanoid robots based on a foot repositionable inverse kinematics engine. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 1114–1120.

Miller, D. (2015). *2015 NASA Technology Roadmaps, TA4: Robotics and Autonomous Systems*. National Aeronautics and Space Administration Office of the Chief Technologist.

Paine, N., Mehling, J. S., Holley, J., Radford, N. A., Johnson, G., Fok, C.-L., and Sentis, L. (2015). Actuator control for the nasa-jsc valkyrie humanoid robot: A decoupled dynamics approach for torque control of series elastic robots. *J. Field Robot.*, 32(3):378–396.

Pederson, L., Kortenkamp, D., Wettergreen, D., and Nourbakhsh, I. (2015). A survey of space robotics. *ISAIRAS*.

Porter, M. (2017). NASA awards citizen inventors top prizes in Space Robotics Challenge. `https://www.nasa.gov/directorates/spacetech/centennial_challenges/space_robotics/nasa-awards-citizen-inventors-top-prizes-in-space-robotics-challenge/`. Accessed: 2019-05-26.

Radford, N. A., Strawser, P., Hambuchen, K., Mehling, J. S., Verdeyen, W. K., Donnan, A. S., Holley, J., Sanchez, J., Nguyen, V., Bridgwater, L., Berka, R., Ambrose, R., Myles Markee, M., Fraser-Chanpong, N. J., McQuin, C., Yamokoski, J. D., Hart, S., Guo, R., Parsons, A., Wightman, B., Dinh, P., Ames, B., Blakely, C., Edmondson, C., Sommers, B., Rea, R., Tobler, C., Bibby, H., Howard, B., Niu, L., Lee, A., Conover, M., Truong, L., Reed, R., Chesney, D., Platt, R., Johnson, G., Fok, C.-L., Paine, N., Sentis, L., Cousineau, E., Sinnet, R., Lack, J., Powell, M., Morris, B., Ames, A., and Akinyode, J. (2015). Valkyrie: Nasa's first bipedal humanoid robot. *Journal of Field Robotics*, 32(3):397–419.