IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

ALPHRED: A Multi-Modal Operations Quadruped Robot for Package Delivery Applications

Joshua Hooks, Min Sung Ahn, Jeffrey Yu, Xiaoguang Zhang, Taoyuanmin Zhu, Hosik Chae, and Dennis Hong

***Abstract*— Modern quadruped robots are more capable than ever before at performing robust, dynamic locomotion over a variety of terrains, but are still mostly used as mobile inspection platforms. This paper presents ALPHRED version 2, a multi-modal operations quadruped robot designed for both locomotion and manipulation. ALPHRED is equipped with high force bandwidth proprioceptive actuators and simple one degree of freedom end-effectors. Additionally, ALPHRED has a unique radially symmetric kinematic design that provides a superior end-effector workspace and allows the robot to reconfigure itself into different modes to accomplish different tasks. For locomotion tasks, ALPHRED is capable of fast dynamic trotting, continuous hopping and jumping, efficient rolling on passive caster wheels, and even has the potential for bipedal walking. For manipulation tasks, ALPHRED has a tripod mode that provides single arm manipulation capabilities that is strong enough to punch through a wooden board. Additionally, ALPHRED can go into a bipedal mode to allow for dual arm manipulation capable of picking up a box off a one meter tall table and placing it on the ground.**

I. INTRODUCTION

In recent years, the legged robotics community has seen a large number of successful platforms capable of impressive dynamic locomotion. In particular, there have been a large number of quadrupedal platforms due to their favorable stability properties: ANYmal C from ANYbotics [1] employs IP67 rated hardware and custom SEA's in order to provide robust inspection and mobility in even the most extreme environments. Cheetah 3 from MIT [2] demonstrates an impressive full 3D gallop at 3 m/s, popularizing the use of proprioceptive actuators [3] [4] for legged robotics. Lastly, Spot from Boston Dynamics [5] is capable of extremely robust walking in various terrains both indoors and out, and also can equip a single robotic arm to open doors and pick up small objects. With their ability to have robust mobility across different terrains, quadrupeds like these are starting to be deployed in surveillance and inspection applications that require such performance.

However, for applications that involve tasks such as picking up a package or using tools, a legged robot needs to have versatile limbs it can use for manipulation. Humanoid platforms like Boston Dynamic's Atlas [6] or Agility Robotics' Digit [7] have been shown performing package manipulation and delivery tasks thanks to the humanoid form factor's versatility. In addition, there were a number of humanoid-like robots showcased at the DARPA Robotics Challenge (DRC) that demonstrated tasks such as driving a car, opening a door, using power tools, and going over extreme terrain. However, bipeds (e.g. humanoids) are not statically stable during locomotion and require active balancing which makes their control much more difficult. For this reason, some of the most successful robots at the DRC utilized a multi-modal approach that allowed them to use wheels ([8]) or treads ([9]) on flat ground and legs for uneven terrain and stairs. Compared to bipeds, quadrupeds and hexapods are typically more stable and robust mobility platforms, but require additional features in order to be effective at manipulation.

Fig. 1: ALPHRED crossing a cross walk with a package.

Prior work on hexapod robots like [10], [11], and [12] utilize two out of the six limbs for both locomotion and manipulation, allowing the system to always have a stable quadruped base at the minimum. The main drawback of these systems is that they are more complex to control and costly to develop due to the increased number of actuated limbs. Some quadruped robots like [13], [14], and [15] have also been developed for manipulation, and are able to 'sit' on various parts of their body in order to make two limbs available for manipulation. Still, these platforms are typically restricted to interacting with objects at the ground level, due to the seated position typically requiring some part of the body or knees to be in contact with the ground.

Taking inspiration from the reliability of quadrupeds, the versatility of humanoids, and the success of the DRC multi-modal platforms, we developed the Autonomous Legged Personal Helper Robot with Enhanced Dynamics (ALPHRED) Version 2 [16] [17]. ALPHRED is a quadrupedal robotic platform that features proprioceptive actuation with all of the actuators collocated at the hip. However, ALPHRED utilizes a non-traditional kinematic leg configuration by swapping

All authors are with the Robotics and Mechanisms Laboratory (RoMeLa) at the University of California, Los Angeles (UCLA), Los Angeles, CA 90095, USA. {hooksj, aminsung, jeffcyu, hawkblizzard, tymzhu, hosikchae, dennishong}@ucla.edu

Copyright ©2020 IEEE

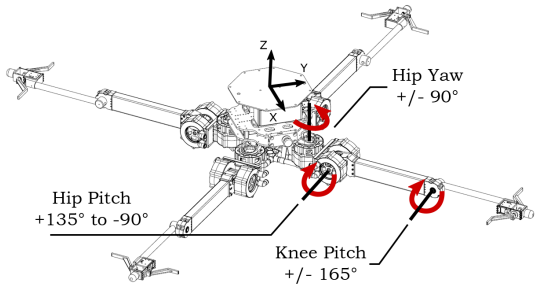


Fig. 2: ALPHRED's unique kinematic configuration providing a large continous workspace.

lateral ab/adduction (typically achieved with a hip-roll DoF) with medial rotation (with a hip-yaw DoF) (Fig. 2). This modification sacrifices some of the dynamical benefits of the traditional configuration in order to provide a much greater range of motion. Combining the robot's large workspace with simple end-effectors allows the robot to have different operating modes to handle a multitude of tasks, providing ALPHRED with capabilities beyond those of current quadrupeds.

The remainder of this paper outlines the ALPHRED platform starting with Section II, detailing the mechanical design of the robot as well as the software architecture. The controllers used for legged and wheeled locomotion are described in Sections III and IV. Both single and dual manipulation are described in Section V. Finally, the results from both locomotion and manipulation experiments are presented in Section VI.

II. System Overview

A. Concept of Operation

ALPHRED was designed to test the feasibility of having legged robots perform package delivery tasks in urban settings, such as picking up packages from the back of a vehicle and placing objects on elevated surfaces such as tables or desks. ALPHRED's unique kinematics allow it to reconfigure into several different operational *modes*. The robot's nominal configuration is a wide sprawl symmetric stance that allows the robot to easily and stably move in any direction (Fig. 3a). This mode is used to transition between modes and to accurately position the robot for manipulation tasks. The robot can switch into a more traditional quadrupedal mode, increasing the front-to-back and decreasing the left-to-right distance of the legs (Fig. 3c). In this mode the platform is capable of a stable 1.5 m/s trot over real world terrain. For more stability, the robot can go into a broader stance and use jumping/pronking for locomotion (Fig. 3e). For improved efficiency on flat terrain, the robot can enter a wheeled mode by lowering itself onto passive caster wheels (Fig. 3d). Once the robot is resting on the four wheels the robot uses its limbs to push itself, providing an efficient mode of transportation on flat smooth terrain.

The four previously mentioned modes of transportation provide ALPHRED with the ability to safely and efficiently

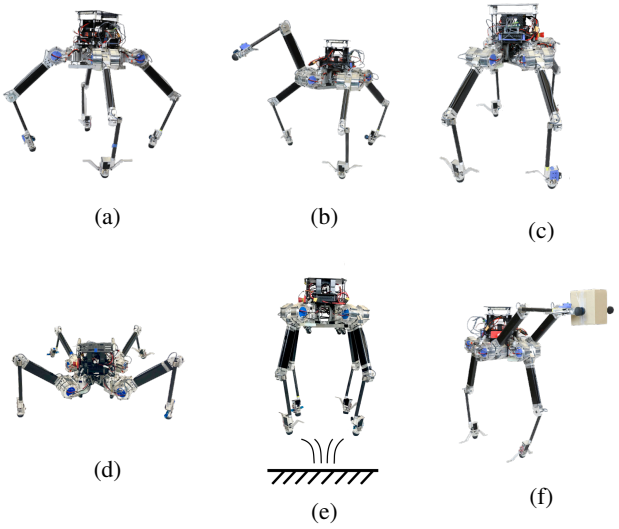


Fig. 3: a) Symmetric stable mode, b) Tripod single manipulation mode, c) Dynamic mode, d) Caster/wheeled mode, e) Jumping/pronking mode, f) Biped dual manipulation mode.

traverse many different types of terrain, but it is ALPHRED's ability to perform simple manipulation tasks that really differentiates it from other quadrupedal robotic platforms. By lifting one limb off of the ground, ALPHRED can enter tripod mode, giving the robot a stable base for single manipulator tasks such as pushing a button (Fig. 3b). This mode offers enough support and strength for the manipulating limb to punch through a wooden board. Finally, the robot is capable of going into a bipedal mode by using its custom end-effectors to provide a large enough support polygon for the robot to balance on two limbs, freeing up the other two for dual limb manipulation (Fig. 3f). ALPHRED's dual arm manipulation is capable of picking up and transporting boxes of various size and weight. This mode also has the potential to be used for bipedal walking as shown on the NABi-2 platform in [18] [19].

B. Design

ALPHRED is composed of four identical limbs that are attached to a central body in a radially symmetric distribution with a total mass of 17.9 kg. The body of ALPHRED (Fig. 4) is comprised of an upper body which houses the electronics, and lower body frame which acts as the chassis. All parts on the ALPHRED body were designed to be lightweight while also ensuring that the structure does not interfere with the limbs' large kinematic workspace. The upper body structure without sensors and electronics weighs 0.63 kg and the lower body frame weighs 0.62 kg, resulting in a total weight of 1.25 kg, which is less than 7% of the total weight of ALPHRED.

ALPHRED's limbs have 3 DoF: hip yaw, hip pitch, and knee pitch (Fig. 2). These joints are actuated by proprioceptive BEAR modules [20] mounted at the hip to reduce limb inertia and mass (Table I provides a brief summary of the BEAR specifications). The knee and hip actuators use a

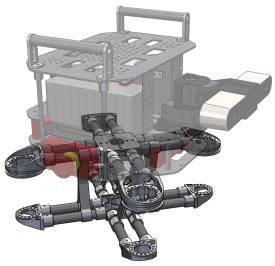


Fig. 4: The transparent part is the upper body structure that houses the electronics and the solid part is the robot chassis.

parallel configuration, with the knee joint being driven by a 1:1 belt transmission which runs through the carbon fiber leg structure and provides the joint with a range of motion that is nearly 360°, as shown in Fig. 5. The parallel configuration of the hip and knee actuator resulted in an antagonistic coupling [21] of the two actuators. Equations (1) and (2) describe how the torques are coupled

$$\tau_h = -(\tau_h^d - \tau_k^d) \tag{1}$$

$$\tau_k = \tau_k^d \tag{2}$$

where, τ_h and τ_k are the commanded torques for the hip and knee actuators respectively, and τ_h^d and τ_k^d are the desired torques for a serially coupled design. The parallel configuration was chosen over a serial configuration to simplify the design, decrease the mass of the limb, and decrease the inertia of the limb. The limb lengths and range of motion were the result of a kinematic study that was driven by the requirement that ALPHRED is able to reach into a vehicle and pick up and drop off a package. The resultant kinematic design was then used in a preliminary locomotion simulation to ensure that the BEAR actuators were capable of the required torque. Table II provides a summary of all design parameters and Fig. 2 shows the full range of motion of each joint.

TABLE I: BEAR Specifications.

Parameter	Value	Unit
Reduction Ratio	10	
Torque Constant	0.88	Nm/A
Peak Torque	32	Nm
Max Velocity	27	Rad/s
Voltage	30	V

The ability to manipulate items often requires 6 DoF if not more; however because we reduced the scope of items to only boxes we were able to add only one additional degree of freedom and still achieve the desired manipulation capabilities. The one additional DoF was added at the foot of each limb and is actuated by a Dynamixel XM430 actuator. This DoF can be used to enhance ALPHRED's capabilities by attaching various accessory components. In the case shown in Fig. 5, a foot extension bar is attached to provide stability when in biped mode, whereas in Fig. 7a a passive joint is added to interface with boxes.

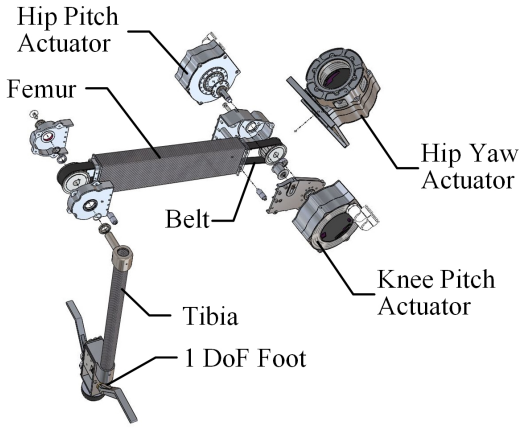


Fig. 5: ALPHRED leg exploded view.

TABLE II: ALPHRED Design Parameters.

Parameter	Value	Unit
Total body mass	17.55	kg
Body[1]mass	13.58	kg
Body ${I_{xx}}^2$	0.288	kgm^2
Body ${I_{yy}}^2$	0.474	kgm^2
Body ${I_{zz}}^2$	0.287	kgm^2
Femur length	0.350	m
Femur mass	0.463	kg
Tibia length	0.350	m
Tibia mass	0.495	kg

[1] Body includes the body structure, electronics, and all actuators.
[2] All inertia values were calculated from a model and use the coordinate system shown in Fig. 2.

C. System Architecture

The main computer is an off-the-shelf Intel NUC equipped with an Intel Core i7-6670HQ @ 2.60 GHz with 32 GB of RAM. ALPHRED's codebase is mostly written in Python, but utilizes external C/C++ libraries such as Eigen and Boost when applicable. ALPHRED runs an architecture tailored for simplicity and modularity to invite multiple developers to easily contribute to modular components that can be easily assembled, similar to the setup developed in [18].

ALPHRED has many concurrently running processes which include a motor communication thread, user input thread, finite state machine (FSM) thread, and a state estimation thread. All communication between threads is done using a custom shared memory library for the sharing of structured data. The motor communication process operates at 1000 Hz and is used as the interface between the controllers and actuators. The FSM process runs all of the high-level planners and controllers making it easy to integrate a new controller via a new state in the state machine. The user input process allows for tele-operation of the robot through a GPD (small handheld computer). Lastly, the state estimation process communicates with the VectorNav 200 IMU and uses the IMU-encoder sensor fusion approach described in [22] to provide full state estimation of the robot. This type of approach provides great flexibility in

further developing ALPHRED as initially demonstrated in the past with incorporating vision and other sensor data into the architecture while minimally modifying the currently existing code [23] [24].

The entire system is powered by two 14.8V 2250 mAh LiPo MaxAmps batteries that are connected in series to a custom power board. The power board provides a 30V line for the BEAR actuators, a 19V line for the Intel NUC, and a 12V line for the Dynamixel XM430 at the feet. The current batteries allow for the robot to continuously trot for approximately 15 minutes.

III. LOCOMOTION CONTROL

One of the main forms of locomotion for ALPHRED is a tele-operated trot gait with equal length stance and swing phases where the operator controls the robot by commanding desired linear velocities v_d and yaw rate $\dot{\psi}_d$. These references are tracked by using Raibert [25] and Capture Point [26] heuristics to plan footsteps and a simple PD controller that adjusts limb lengths to correct for orientation errors. An event-based state machine is used to determine what state the robot is in and which control to take. All calculations are done in the body frame shown in Fig. 2.

A. Limb State Machine

Locomotion controllers for legged robots need to manage both the continuous dynamics of the system as well as the discrete phase transitions determined by changing contact conditions. For this reason, we have developed a Finite State Machine to organize the control and detection of the changing states of each limb as shown in Fig. 6. The events that trigger transitions between states are determined by the actual and desired state of contact of the foot. Contacts are detected via contact switches built into the tip of each end-effector, the design of these sensors is out of the scope of this work. All events are shown in Table III and a short description of each state is provided below.

TABLE III: Table of events, 1 indicates contact and 0 indicates no contact.

	Desired	**Actual**
s_1	1	1
s_2	0	1
s_3	1	0
s_4	0	0

Stance (ST): In stance the limb is securely on the ground allowing for ground reaction forces to control the center of mass dynamics of the robot. Section III-C describes the controller used in this phase.

Swing (SW): The swing state is used to transition the foot's position from one location to the next. Cycloidal functions are used to create the trajectory of the swing legs [16]. If using torque control, the following impedance controller was used:

$$\tau = \mathbf{H}\mathbf{J}^{-1}[\mathbf{u} - \dot{\mathbf{J}}\dot{\mathbf{q}}] + \mathbf{C} + \mathbf{G} - \mathbf{J}^{\mathbf{T}}\lambda \tag{3}$$

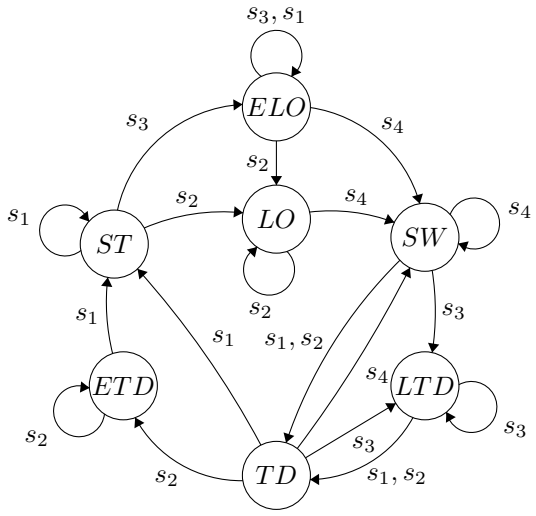


Fig. 6: Visual representation of the limb FSM.

$$\mathbf{u} = \ddot{\mathbf{x}}_\mathbf{d} + \mathbf{M}_\mathbf{d}^{-1}[\mathbf{K}_\mathbf{d}(\mathbf{x}_\mathbf{d} - \mathbf{x}) + \mathbf{B}_\mathbf{d}(\dot{\mathbf{x}}_\mathbf{d} - \dot{\mathbf{x}}) - \lambda] \tag{4}$$

where $\tau \in \mathbb{R}^3$ is the desired torque vector, $\dot{\mathbf{q}} \in \mathbb{R}^3$ is the current joint velocities, $\mathbf{x} \in \mathbb{R}^3$ is the end effector position, $\dot{\mathbf{x}} \in \mathbb{R}^3$ is the end effector velocity, $\mathbf{J} \in \mathbb{R}^{3\times3}$ is the Jacobian, $\mathbf{H} \in \mathbb{R}^{3\times3}$ is the mass matrix, $\mathbf{C} \in \mathbb{R}^{3\times1}$ is the Coriolis term, $\mathbf{G} \in \mathbb{R}^{3\times1}$ is the gravitational term, and $\lambda \in \mathbb{R}^{3\times1}$ is the ground reaction forces. In (4) $\mathbf{M_d}, \mathbf{B_d}$, and $\mathbf{K_d} \in \mathbb{R}^{3\times3}$ represent the mass, damping and stiffness of the desired impedance as described in [27] while $\mathbf{x_d}, \dot{\mathbf{x}}_\mathbf{d}$, and $\ddot{\mathbf{x}}_\mathbf{d} \in \mathbb{R}^{3\times1}$ describes the desired position, velocity, and acceleration of the foot. Note that during the swing phase λ is set to zero as there is no ground reaction forces.

Touchdown (TD): The touchdown state is used to determine if the foot is securely on the ground. Once a rising edge is detected from the contact sensor the limb will immediately transition into the touchdown state where the impedance controller (3) is used to prevent the foot from bouncing upon impact. In this phase the B_d matrix is increased to provide more damping and λ is set to 25% of the weight of the robot, both of these changes were determined through experimentation and were found to help mitigate foot bouncing. The state machine will stay in this state until the foot contact sensor reads consistently for 7ms.

Early Touchdown (ETD): This state performs the same function as the stance state but additionally ensures that the limb does not transition to the early lift-off state. This state allows for an extended stance period.

Late Touchdown (LTO): When this state occurs the stance controller is lacking control authority over at least one contact point, which can have a severe impact on performance. To mitigate this problem, the limb is commanded to hold it's X and Y position and move downwards in the Z direction, thus trying to shorten the time that the robot is in this state, similar methods to this were used in [28].

Liftoff (LO): This state performs the same function as the swing state but additionally prevents the transition into the early touchdown until 60% of the swing time has passed.

At 60% the foot is now on its downward trajectory and is capable of establishing a stable foothold, whereas this is not the case for the upward trajectory.

Early Liftoff (ELO): Early liftoff occurs when a foot unexpectedly loses contact with the ground. In this state the foot is commanded to continue to use the controller used in the stance phase in hopes that the foot will again make contact shortly and will be able to contribute to the control of the robotic system.

B. Footstep Planner

The cadence of the robot's gait is controlled via timers that are monitored by the limb state machines. Once the timer goes below zero the desired state switches from stance phase to swing phase or vice versa for that limb, and the timer is reset to either the swing time or the stance time depending on which state it was in previously. This formulation makes it simple to construct different gaits by separately adjusting the swing and stance times of each of the limb state machines. During the swing and liftoff phase the desired foot position is planned using Raibert and Capture Point heuristics (5), similar to the technique used in [2]. All calculations for foot placement are done in the body frame.

$$\mathbf{p} = \underbrace{\mathbf{R}(\dot{\psi}_d \frac{1}{2} T_s)\mathbf{p_0} + \frac{1}{2} T_s \upsilon_d}_{\text{Raibert}} + \underbrace{\frac{1}{2}\sqrt{\frac{p_z}{g}}(\upsilon - \upsilon_d)}_{\text{Capture Point}} \quad (5)$$

where $\mathbf{p} \in \mathbb{R}^2$ is the desired X and Y position for the next footstep location, p_z is the nominal height of the robot, g is the scalar value for gravity, $\upsilon \in \mathbb{R}^2$ is the current velocity of the body, $\upsilon_\mathbf{d} \in \mathbb{R}^2$ is the desired velocity in the body frame given by the operator, $\dot{\psi}_d$ is the desired yaw rate given by the operator, T_s is the total time that the limb will be in stance on the next step, $\mathbf{p_0} \in \mathbb{R}^2$ is nominal position of the limb, and $\mathbf{R} \in \mathbb{R}^{2\times 2}$ is a rotation matrix. In practice, the Capture Point contribution is only used when $|\upsilon - \upsilon_d| > 0.15$ which helps correct major disturbances but does not come into play when operating normally. The clipping of the Capture Point term was found to help prevent oscillatory behavior at small velocities but still mitigate big disturbances, this parameter was chosen through experimentation.

C. Stance Phase Controller

During the stance and early touchdown states the X and Y Cartesian positions of the limb are commanded to match the inverse of the desired velocity of the robot. The PD controller, described by (6), changes the length of the limb to correct height and rotation errors of the robot, enabling the robot to traverse uneven terrain and mitigate disturbances. In position mode the output of the PD controller is a Z velocity and in torque mode the output is a Z force; both result in changing the length of the given limb. If in torque mode, the torques are calculated using the mapping from the Jacobian.

$$z_{out} = k_{p,h}(p_{0,z} - p_z) + k_{d,h}\dot{p}_z + s(\mathbf{K_{p,r}} \log(\mathbf{R}_d \mathbf{R}^T) - \mathbf{K_{d,r}}\omega) \quad (6)$$

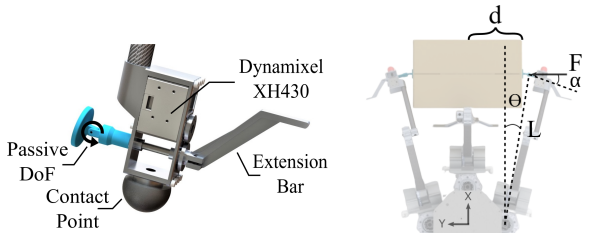
where $p_{0,z}$ is the nominal Z position of the limb, p_z is the current Z position of the limb, $\mathbf{R}_d, \mathbf{R} \in \mathbb{R}^{3\times 3}$ are the desired rotation matrix and current rotation matrix representing the transform from the body frame to the world frame (desired pitch and roll angles are always zero), $\omega \in \mathbb{R}^3$ is the angular rates of the robot in the body frame, and $s = [1, 1, 0]$ is a selection matrix that selects the roll and pitch errors (this controller does not adjust for yaw errors). The orientation errors are derived using the matrix log mapping described in [29] [30]. $k_{p,h}$ and $k_{d,h}$ are the scalar proportional and derivative gains for the height component. $\mathbf{K_{p,r}}$ and $\mathbf{K_{d,r}}$ are the diagonal $\mathbb{R}^{3\times 3}$ proportional and derivative gain matrices for the orientation error component. For all four limbs the magnitude of the gains are the same but the orientation gain matrices will differ in signs between elements in order for the limbs to appropriately respond to errors.

The combination of the footstep planner and the stance phase controller is essentially a Raibert controller as described in [25]. During the swing phase the footstep location is based off of the desired velocity of the robot while during the stance phase simple PD controllers are used to correct for height and orientation errors.

IV. ADDITIONAL LOCOMOTION MODES

ALPHRED can currently perform two additional forms of locomotion: *pronking*, which is jumping on four legs, and *skating*, which is rolling on passive wheels while being propelled by the legs. Pronking can be utilized as an alternative mode of transport, or to overcome obstacles. Skating can be used to move very quickly and efficiently over flat terrain, since the robot no longer needs to continuously support its body weight. These approaches to locomotion are available to ALPHRED thanks to the proprioceptive actuators and unique limb kinematics which can handle the large impacts from pronking and the sprawled leg configuration of skating, respectively. These modes of locomotion are implemented using the same framework described in Section III, with only minor modifications.

For pronking, the early/late liftoff/touchdown states are removed, leaving just the stance, liftoff, flight, and touchdown phases. Reference trajectories are still tracked with the PD controller, and the footsteps are planned with the Raibert and capture point planner. However, the foot schedule no longer alternates between pairs of limbs and instead schedules all four legs with the same sequence. The skating controller is very similar to that used for pronking, except that it is executed with the passive caster wheels in contact with the ground. The main difference between all three locomotion implementations (trot, pronk, skate) is the stance phase locomotion controller. The pronk controller no longer attempts to track the commanded input during the stance phase and simply embeds spring-mass dynamics into the legs, allowing the Raibert and capture point footstep planner to regulate velocity. The skating controller executes the same inverse velocity matching done for trotting, but with a much longer swing time phase.



(a) (b)

Fig. 7: a) End-effector with attachment for picking up boxes, b) Calculating the required clamping force to pick up the box.

V. MANIPULATION

ALPHRED's versatility is demonstrated by its ability to perform various simple manipulation tasks using one or two limbs. The more stable form of manipulation is *single arm manipulation* in which three support legs form a support polygon in the shape of an isosceles triangle, as seen in Fig. 3b. In this mode, ALPHRED is capable of single arm manipulation tasks by using a combination of its remaining single limb and body to reach and orient the end-effector to its desired pose. Because of ALPHRED's size and solid support stance, it can handle single hand tasks such as pressing an elevator button, pushing open a door, opening door levers, and inspecting tight angles with an end-effector mounted camera.

A. Dual Arm

One of the most unique features of ALPHRED is its ability to perform *dual arm manipulation* in order to pick up various size boxes. As mentioned previously, ALPHRED's yaw joints have a ± 90 degree range of motion. This kinematic range allows for two opposing limbs to become parallel, enabling them to work in tandem to perform simple manipulation tasks. The manipulation limbs are equipped with simple, passive end-effectors (Fig. 7a), while the limbs in support are equipped with structural foot extension bars that act as lateral stabilizers. The simple end-effectors have a passive joint perpendicular to the active joint that ensures that the gripping surface is always parallel to the surface of the box during manipulation tasks. However, it still allows for torque to be transferred so that the active joint can control the pitch of the box. If manipulation is performed in bipedal mode, the structural extension bars on the stance legs will be oriented down to buttress the support feet, increasing the area of the support polygon and making it easier to balance on two legs (Fig. 3f).

In order to pick up a box, ALPHRED will orient its limbs to be on either side of the box as shown in Fig. 7b. Then, the limbs performing the manipulation can be tele-operated to be approximately in-line with the box's center of mass. Once aligned, an auto-clamping routine that utilizes the back-drivability of the BEAR modules will gently clamp onto the box by limiting the maximum torque of the yaw actuators. Adequate clamping is detected when a significantly large position error at the end-effector is observed, after which the max torque of the yaw actuators is slowly increased until the position error starts to decrease (this means that the box is being deflected) or maximum torque is applied. After the auto-clamping routine is complete the clamping force is calculated from (7).

$$F_c = \frac{cos(\alpha)\tau}{L} \tag{7}$$

where F_c is the clamping force, τ is the yaw actuator torque, L is the horizontal distance from the yaw actuator to the manipulator, and α is the angle between the force created by the limb and the normal vector of the box surface.

Once ALPHRED has clamped a box, it can place it onto its body for transport using a hybrid position-force controller. The hip pitch and knee pitch actuators are commanded with position control to track a trajectory that will place the box onto the body. The yaw actuators are commanded (with position control) to penetrate the box by 1 cm, but their maximum torques are limited to track the desired clamping force calculated using (7) and the current configuration of the end-effectors. This approach ensures that the resulting clamping force will not crush or drop the box. This approach resulted in control of the end-effector's position in the X and Z plane and control of the force along the Y axis in the box frame shown in Fig. 7b. This is similar to the torque based hybrid position-force controller described in [31] where the problem is broken up into both position and force constraints that must be satisfied.

VI. RESULTS AND DISCUSSION

This section presents the results found from experiments ran on ALPHRED, all results are shown in the supplementary video.

A. Trotting

The primary gait used by ALPHRED is a trotting gait with swing and stance times ranging from 0.18-0.3 seconds. Using a 0.2 second swing and stance time ALPHRED can reliably walk at 1.0 m/s and achieve a turning rate of 0.4 rad/s (Fig. 8). From this experiment the cost of transport (CoT) was calculated to be 2.6, the CoT formula used is the same as the one presented in [32]. After lowering both the swing and stance time times to 0.18 seconds ALPHRED achieved a max velocity of 1.5 m/s. Using the same trot gait, ALPHRED was able to successfully walk over an artificial obstacle course comprised of scattered blocks as obstacles ranging in height from 2-4 cm. In addition, the robot has also successfully walked around in the real world walking on side walks, streets, and dirt paths which also included terrain with slopes $> 15°$.

Developing walking strategies for the ALPHRED platform revealed two limitations imposed by the unique mechanical design of the robot. To increase the velocity of a walking gait the robot can either decrease the gait cycle time or increase the stride length. A quadruped with a typical kinematic design has a nominal foot position directly under the hip. During locomotion, the leg will fully stretch out in front

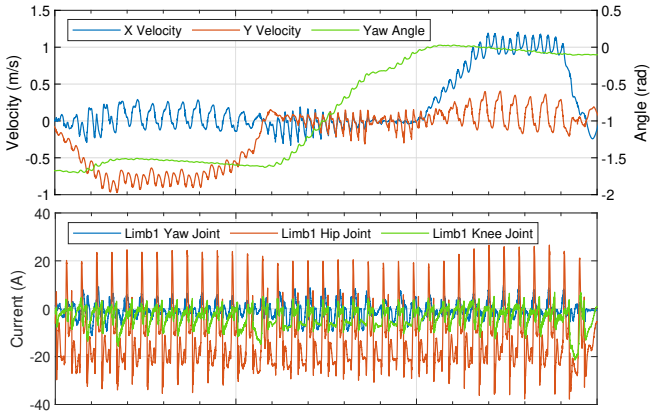


Fig. 8: The data comes from a test in which ALPHRED was commanded to move forward at 1.0 m/s, make a 90 degree turn, and then move forward at 1.0 m/s. **top)** Linear velocities and yaw recorded from the state estimator. **bottom)** Joint currents for a single limb.

of this nominal position and during the stance phase it will pass underneath the hip until once again the limb is fully extended, at which point the swing phase will start. With ALPHRED's kinematic design this full stride is not possible, because as the foot approaches the hip, the limb reaches a singularity that causes extremely large hip yaw movement. For this reason, ALPHRED's nominal foot position was chosen to be mid-way between the fully extended foot position and the projected hip position on the ground. The swing phase would start just in front of the projected hip position on the ground rather than passing under the hip yaw joint. This strategy resulted in a $\sim 50\%$ reduction in stride length compared to that of a traditional quadruped.

The second limitation is a result of the antagonistic relationship of the hip pitch actuator and the knee pitch actuator. As shown in (1), the knee actuator will work against the hip actuator when the desired torques are in opposite directions. Unfortunately, this is the case for the stances used for the resultant trotting experiments, meaning that the hip actuator was doing significantly more work than the knee actuator. This limitation dictated the nominal foot placements and the body height of the robot to ensure that the hip torque actuator would not over heat nor saturate when trotting.

B. Jumping/Pronking

To demonstrate the speed and strength of the BEAR modules a single jump was performed by commanding all four limbs to go from the crouched to standing position in 0.05 seconds. The test resulted in a vertical jump of 0.7 m as measured from the top of the robot at its apex and the top of the robot in it's final standing pose. For controlled jumping, the pronking gait described in IV enabled the robot to move in the X and Y directions at 0.3 m/s (Fig. 9).

C. Caster Mode

The skating locomotion enabled ALPHRED to reach velocities of up to 2.1 m/s on smooth, flat ground using tele-operation. Interestingly, the limiting factor for skating seemed to be the low quality of the caster wheels, as the robot would experience significant deceleration during the swing/flight phase of the gait. However, even with the inefficiencies of the casters, a CoT of 0.55 (40% of power was used by the computer) was recorded, which is not surprisingly much more efficient than the trotting gait.

D. Manipulation

To test the robot's two limb manipulation we had the robot pick up and drop off packages that we continuously increased in size until failure. This resulted in the smallest package being $16 \times 13.2 \times 11$ cm and the largest being $48 \times 34.5 \times 17.5$ cm. A similar test was repeated with increasing weight, using a nominal package of size $31 \times 21.5 \times 14$ cm. The weight test resulted in ALPHRED being able to pick up 3 Kg. Finally we tested what was the tallest height at which the robot could grab the nominal sized package from which resulted in a height of 1.02 m.

VII. CONCLUSION

This paper presented the second version of the ALPHRED platform. This platform utilizes a unique kinematic design giving it the ability to operate in different modes for both transportation and manipulation. We believe that we have successfully shown that this configuration change coupled with high torque proprioceptive actuators preserves the benefits of quadrupedal robots, stability and reliability, while also providing capabilities that go beyond that of traditional quadrupeds. While we are excited by these results we believe that this is only the first step in moving towards a world with fully automated robotic package delivery. To move closer to our ultimate goal we believe the next steps include improving the maximum run time of the robot, improving the autonomous capabilities of the robot, and adding a package storage system.

To increase the robot's operating time a redesign of the limbs is being considered. Currently, the hip pitch actuator is undersized due to the parallel coupling of the hip and knee pitch joints. A simple change being considered is doubling the max torque of the hip pitch actuator. This would allow for larger batteries to be used as well as prevent the hip pitch actuator from overheating, while still preserving the robots large range of motion.

Moving towards a fully autonomous robotic platform, we are working on integrating a vision system on to the platform to incorporate high level path planning as well as fully automating the package pick up and drop off. Additionally, a package carrier is being designed to sit atop the body in order to secure the package during transportation. Finally, we hope to develop a bipedal walking gait to further increase the different ways in which ALPHRED is equipped to handle a task.

ACKNOWLEDGMENT

This work was jointly supported by a grant (code R2016001) from Gyeonggi Technology Development Pro-

Fig. 9: A time series depicting one full pronking gait cycle. The robot is moving at 0.3 m/s to the left.

gram Funded by Gyeonggi Province and by a grant (N00014-15-1-2064) from the ONR. Thank you Donghun Noh for helping with the logistics of this project.

REFERENCES

[1] "ANYmal C," http://www.anybotics.com/anymal-legged-robot/, accessed: 2019-08-20.

[2] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 2245–2252.

[3] S. Seok, A. Wang, D. Otten, and S. Kim, "Actuator design for high force proprioceptive control in fast legged locomotion," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1970–1975.

[4] G. Kenneally, A. De, and D. E. Koditschek, "Design principles for a family of direct-drive legged robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 900–907, 2016.

[5] "Spot: Good Things Come in Small Packages," https://www.bostondynamics.com/spot, accessed: 2019-08-20.

[6] "Atlas: The Worlds Most Dynamic Humanoid," https://www.bostondynamics.com/atlas, accessed: 2019-08-20.

[7] K. Washington. (2019, May) Meet digit: A smart little robot that could change the way self-driving cars make deliveries. [Online]. Available: https://media.ford.com/content/fordmedia/fna/us/en/news/2019/05/22/meet-digit-robot-self-driving-delivery.html

[8] J. Lim, I. Lee, I. Shim, H. Jung, H. M. Joe, H. Bae, O. Sim, J. Oh, T. Jung, S. Shin *et al.*, "Robot system of drc-hubo+ and control strategy of team kaist in darpa robotics challenge finals," *Journal of Field Robotics*, vol. 34, no. 4, pp. 802–829, 2017.

[9] A. Stentz, H. Herman, A. Kelly, E. Meyhofer, G. C. Haynes, D. Stager, B. Zajac, J. A. Bagnell, J. Brindza, C. Dellin *et al.*, "Chimp, the cmu highly intelligent mobile platform," *Journal of Field Robotics*, vol. 32, no. 2, pp. 209–228, 2015.

[10] N. Koyachi, H. Adachi, M. Izumi, and T. Hirose, "Control of walk and manipulation by a hexapod with integrated limb mechanism: Melmantis-1," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 4. IEEE, 2002, pp. 3553–3558.

[11] T. Takubo, T. Arai, K. Inoue, H. Ochi, T. Konishi, T. Tsurutani, Y. Hayashibara, and E. Koyanagi, "Integrated limb mechanism robot asterisk," *Journal of Robotics and Mechatronics*, vol. 18, no. 2, pp. 203–214, 2006.

[12] K. Inoue, K. Ooe, and S. Lee, "Pushing methods for working six-legged robots capable of locomotion and manipulation in three modes," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 4742–4748.

[13] T. Omata, K. Tsukagoshi, and O. Mori, "Whole quadruped manipulation," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 2028–2033.

[14] S. Hirose, S. Yokota, A. Torii, M. Ogata, S. Suganuma, K. Takita, and K. Kato, "Quadruped walking robot centered demining system-development of titan-ix and its operation," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 1284–1290.

[15] W. Wolfslag, C. McGreavy, G. Xin, C. Tiseo, S. Vijayakumar, and Z. Li, "Optimisation of body-ground contact for augmenting whole-body loco-manipulation of quadruped robots," *arXiv preprint arXiv:2002.10552*, 2020.

[16] J. Hooks and D. Hong, "Implementation of a versatile 3d zmp trajectory optimization algorithm on a multi-modal legged robotic platform," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3777–3782.

[17] J. R. Hooks, "Real-time optimization for control of a multi-modal legged robotic system," Ph.D. dissertation, UCLA, 2019.

[18] J. Yu, J. Hooks, X. Zhang, M. S. Ahn, and D. Hong, "A proprioceptive, force-controlled, non-anthropomorphic biped for dynamic locomotion," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–9.

[19] J. Yu, J. Hooks, S. Ghassemi, A. Pogue, and D. Hong, "Investigation of a non-anthropomorphic bipedal robot with stability, agility, and simplicity," in *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. IEEE, 2016, pp. 11–15.

[20] T. Zhu, J. Hooks, and D. Hong, "Design, modeling, and analysis of a liquid cooled proprioceptive actuator for legged robots," in *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE/ASME, 2019, pp. 1–8.

[21] A. Abate, J. W. Hurst, and R. L. Hatton, "Mechanical antagonism in legged robots." in *Robotics: Science and Systems*, vol. 6. Ann Arbor, MI, 2016.

[22] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and imu," *Robotics*, vol. 17, pp. 17–24, 2013.

[23] M. S. Ahn, H. Chae, and D. W. Hong, "Stable, autonomous, unknown terrain locomotion for quadrupeds based on visual feedback and mixed-integer convex optimization," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3791–3798.

[24] J. Hooks, M. S. Ahn, and D. Hong, "Online trajectory optimization for legged robotics incorporating vision for dynamically efficient and safe footstep locations," in *ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2019.

[25] M. H. Raibert, *Legged robots that balance*. MIT press, 1986, pp. 44–47.

[26] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*, Dec 2006, pp. 200–207.

[27] J. H. Park, "Compliance/impedance control strategy for humanoids," in *Humanoid Robotics: A Reference*, A. Goswami and P. Vadakkepat, Eds. Dordrecht: Springer Netherlands, 2016, pp. 1–20.

[28] R. J. Griffin, G. Wiedebach, S. Bertrand, A. Leonessa, and J. Pratt, "Straight-leg walking through underconstrained whole-body control," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–5.

[29] F. Bullo and R. M. Murray, "Proportional derivative (pd) control on the euclidean group," in *European Control Conference*, vol. 2, 1995, pp. 1091–1097.

[30] R. M. Murray, *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[31] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *Journal of dynamic systems, measurement, and control*, vol. 103, no. 2, pp. 126–133, 1981.

[32] P. A. Bhounsule, J. Cortell, and A. Ruina, "Design and control of ranger: an energy-efficient, dynamic walking robot," in *Adaptive Mobile Robotics*. World Scientific, 2012, pp. 441–448.