

# Sensor-Based Legged Robot Homing Using Range-Only Target Localization

Vasileios Vasilopoulos<sup>†</sup>, Omur Arslan<sup>\*</sup>, Avik De<sup>\*</sup>, and Daniel E. Koditschek<sup>\*</sup>

**Abstract**—This paper demonstrates a fully sensor-based reactive homing behavior on a physical quadrupedal robot, using onboard sensors, in simple (convex obstacle-cluttered) unknown, GPS-denied environments. Its implementation is enabled by our empirical success in controlling the legged machine to approximate the (abstract) unicycle mechanics assumed by the navigation algorithm, and our proposed method of range-only target localization using particle filters.

## I. INTRODUCTION

Sensor-based homing is a generically useful mobile robot capability for which formal performance guarantees relative to prior knowledge about the environment are essential for reliable autonomous docking (e.g. to sleep and recharge) and rendezvous (e.g. to transfer data and accept mission updates) behavior. Although this problem has been studied in some detail in the past in the context of underwater vehicles [1]–[3], related literature has been far more sparse in the domain of legged robotics where both complicated body dynamics and the presence of unanticipated obstacles becomes a compelling concern. To the best of our knowledge, this work represents the first time a purely sensor-based homing algorithm (using online target range data) that is provably correct relative to a (sufficiently simple) unknown, obstructed, GPS-denied environment (sensed by online LIDAR) has been implemented on a legged robot.

### A. Motivation, Background and Prior Literature

The hallmark of feedback-based motion planners [4]–[7] is that robot trajectories are generated in real time in reaction to instantaneous state measurements by a vector field typically constructed offline, algorithmically, from prior information about the loci of obstacles and goal. Their dynamical nature makes them particularly applicable to robots governed by second order [8], [9] or nonholonomic [10]–[12] mechanics. With the further aim of reducing the need for prior information, we have introduced the term “doubly reactive” [13] to refer to schemes, e.g. [4]–[6], whose algorithmically constructed vector fields are themselves generated in real time from locally perceived structure. In addition to their virtue in mitigating perturbations arising from complex mechanics of the mobile platform, such sensor-driven reactive motion planners invite application to unknown, noisy, dynamic environments as well.

<sup>†</sup>Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA 19104, vvasilo@seas.upenn.edu.

<sup>\*</sup>Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, {omur, avik, kod}@seas.upenn.edu.

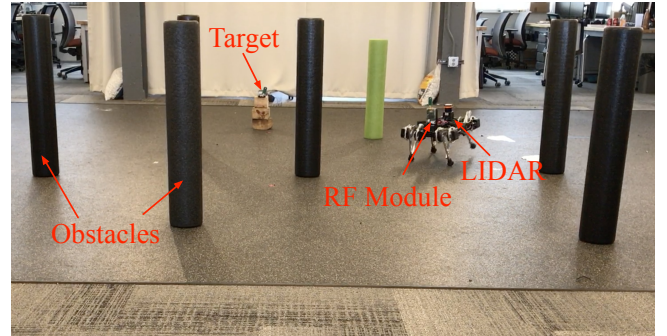


Fig. 1. Minitaur navigating through an artificial forest towards a target.

While the problem of exact navigation for nonholonomically constrained plants, such as the unicycle model, has been addressed in the past literature [11]–[14], the few applications to legged machines have been almost entirely restricted to quasi-static [15], [16] or small-scale legged platforms [17] whose dynamics is negligible. With the exception of very recent work focused on the problems of legged navigation for rugged, forested, hilly terrain [18]<sup>1</sup>, application of vector-field motion planners to highly dynamic legged platforms has heretofore been restricted to exteroceptive servoing applications in contractible (no obstacles puncturing the interior of the) workspaces [10], [20], [21].

Most prior legged locomotion literature has focused on the template-anchor paradigm [22]–[24] and related ideas [25]–[27] as a means of abstracting away the details of gait and other high DoF regulation problems so as to simplify the control of complex legged machines in the sagittal plane. Our goal in this paper is to further advance the abstraction away from the complicated kinematics and dynamics associated with sagittal plane legged locomotion to expose only the effective horizontal plane mechanics required for mobility of the mass center. We seek to establish the unicycle model as a *navigation template* [28] for the application to legged platforms of high-level planners originally designed for non-holonomically constrained wheeled or tracked vehicles.

### B. Contributions and Organization of the Paper

The contributions of this paper are largely empirical. Our main advance is to use the emerging understanding of

<sup>1</sup>To distinguish the present contribution relative to [18] where a body frame IMU reported online terrain gradients to a RHex machine [19] (guaranteeing only convergence to a local ridge), here, a designated target’s online beacon signals are used to guarantee homing using range-only information in the robot’s body frame.

Minitaur’s bounding [29] and walking trot gaits to improve its horizontal plane behavior to the point of exhibiting the dynamics of a horizontal plane unicycle, which we can then adopt as the *navigation template* assumed by the reactive navigation algorithms [13], [14] it must execute. A second contribution is to realize this algorithm in a GPS-denied environment by recourse to a body-frame, range-only target localization scheme. More specifically, the robot is assumed to possess only an RF sensor providing range measurements from the desired goal. In addition to the LIDAR signals used to avoid the unknown obstacles, our algorithm uses only this one-dimensional information to extract the (two-dimensional) position of the goal, and the reformulation of the navigation algorithm in [13] in the robot’s body frame allows for successful homing while guaranteeing obstacle avoidance along the way.

The paper is organized as follows. Section II gives a description of the control strategy that empirically anchors a kinematic unicycle on Minitaur while it is executing a bounding or a walking trot gait. Section III summarizes the ideas behind both the locally sensed and the sensor-based motion planning strategy. Section IV describes the proposed body-frame, range-only target localization algorithm that allows for successful homing. Section V continues with a description of our experimental setup. Section VI begins by demonstrating the effectiveness and robustness of the doubly reactive motion planning scheme in a small-sized and densely cluttered developed experimental environment (Section VI-A) using the bounding gait and only offboard sensing, and continues with more experiments using the full sensor-based version of the algorithm and the target localization scheme (Section VI-B). Finally, in Section VII, we conclude with a summary of the results reported here.

## II. EMPIRICAL UNICYCLE ANCHORING

This section describes our experimental platform, the Minitaur quadruped [30], focusing on the empirical anchoring [28] of a first order unicycle in its bounding and walking gait.

### A. Minitaur Hardware

Minitaur (Fig. 1 [30], [31]) is a 6kg direct drive quadruped that has already demonstrated a variety of interesting behaviors, including a 48 cm vertical leap [30], bounding at a continuum of speeds up to 2 m/s, pronking, trotting, etc. [29]. In this paper, we only make use of the “bounding” and “walking trot” (for moving with a desired fore-aft and angular velocity), and “standing” (employed before the beginning and after the end of any motion for safely starting and terminating experiments) behaviors.

The bounding gait can achieve higher speeds, but it induces a strong body pitching motion which makes the application of onboard, sensor-based navigation techniques quite hard. For this reason, we use the bounding gait in the context of a navigation algorithm using only local but “bird’s eye” information about the surrounding obstacles from the motion capture arena, and the walking trot gait for

fully sensor-based navigation with a LIDAR (for obstacle avoidance) and a range RF sensor (for target localization in the body frame), as described in Section III. We use a Ghost Minitaur [31] for the experiments in this paper.

### B. Bounding Gait as a Kinematic Unicycle

Bounding is a virtual bipedal gait, wherein the front pair and rear pair of legs are phase-locked to each other, and the steady state stepping pattern is an alternation of front and rear stance periods, typically with substantial aerial phases in between. Minitaur’s bounding is implemented using compositional principles [24] yielding a controller which requires few parameters, and exerts no feedback phase coordination between the front and rear hips [29].

The bounding controller exposes two commands: horizontal plane translational speed  $v_c$  and yaw rate  $\omega_c$ . Heretofore, these parameters have been set by a human operator, but in this paper, for the first time, we supply these parameters from a higher-level controller. It is worth noting here that BigDog was able to generate such control commands autonomously for following a leader [32], but here we focus on the autonomous navigation problem. Though we don’t make any formal claims of anchoring [28] in this paper, we present an empirical characterization of bounding Minitaur as a kinematic unicycle as assumed in Section III and use this working model as a trial *navigation template* for our legged platform. Our ultimate goal is to abstract away the complicated bounding dynamics of Minitaur and allow the robot to be controlled by a high-level motion planner as a differential drive robot.

However, bounding Minitaur is very much a dynamic system, and requires a non-trivial amount of time to accelerate between different speeds and yaw rates. In fact, the stride rate (3Hz) limits the control authority available, since the body cannot be actuated in flight. We hypothesize that a dynamic unicycle model [33], [34] with limits on acceleration [18] would be the most appropriate horizontal template for Minitaur, but here, we instead smooth the inputs with an auto-regressive filter to reduce the magnitude of the acceleration.

Given as inputs a desired speed  $v_d \in \mathbb{R}$ , and yaw rate  $\omega_d \in \mathbb{R}$ , let  $v_c$  and  $\omega_c$  be the commands sent to Minitaur. Then, we set

$$\dot{v}_c = -\sigma_v(v_c - v_d), \quad \dot{\omega}_c = -\sigma_\omega(\omega_c - \omega_d), \quad (1)$$

for some  $\sigma_v, \sigma_\omega \in \mathbb{R}_+$ . Note that smaller  $\sigma_*$  results in a smoother output, and vice versa.

For the empirical characterization of our strategy, we send Minitaur time-varying signals, and plot its response. Fig. 2 shows the time trajectories of the observed speed and yaw (measured by the motion capture system described in Section V) for a commanded sinusoidal signal of a fixed frequency. The 3Hz cutoff filter removes periodicities caused by Minitaur’s 3Hz stride rate.

Minitaur’s response to smooth commands is very accurate in yaw, and more lagged in speed. We believe that this is due to the very small  $\sigma_v$  that had to be used in (1) in order

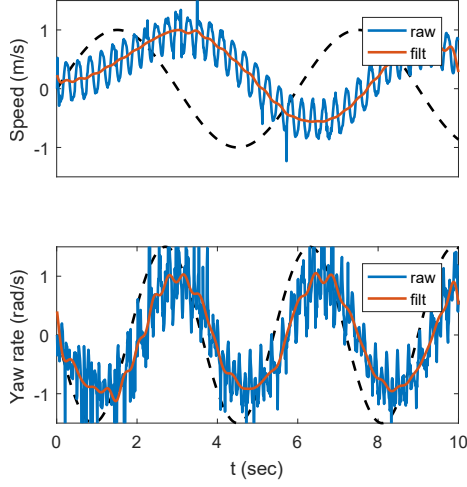


Fig. 2. Frequency domain characterization of Minitaur's bounding response to smooth input signals  $v_d, \omega_d$  (1): raw speed  $v$  and yaw response  $\omega$  (blue), with a 3Hz cutoff filter (red), and the reference signals  $v_d, \omega_d$  (black dashed).

to limit acceleration, since lower speeds are necessitated in the case of vector fields with high curvature. The robot's response to speed and yaw commands in typical experiment runs is shown at the end of the results section in Fig. 9.

### C. Walking Trot Gait as a Kinematic Unicycle

The walking trot gait we use in this work is also a virtual bipedal gait, wherein the diagonal pairs of legs are phase-locked to each other and the steady state stepping pattern is an alternation of diagonal stance periods with rapid flight phases in between. Although the formal analysis of this gait is still work in progress, we attempt an empirical characterization of the walking Minitaur as a kinematic unicycle, in an effort to use this model as the *navigation template* for sensor-based navigation and range-only target localization, since its negligible pitching motion allows for the straightforward use of a LIDAR for collision avoidance.

Similarly to the bounding gait, the walking trot controller exposes two commands: horizontal plane translational speed  $v_c$  and yaw rate  $\omega_c$ , set by a higher-level controller. For the generation of smooth commands  $v_c$  and  $\omega_c$  from the desired inputs  $v_d$  and  $\omega_d$  we employ a first-order filter similar to (1) with lower gains  $\sigma_v, \sigma_\omega$ , since we noticed that rapid changes in the inputs  $v_d, \omega_d$  resulted in easier loss of traction and more falls compared to the bounding gait. We suspect that this occurs due to the more complicated stance kinematics of walking that make turning harder, but further investigation is currently underway.

As in the bounding gait trials, for the empirical characterization of our strategy, we send Minitaur time-varying signals and plot its response. Fig 3 shows the time trajectories of the observed speed and yaw (measured by a Vicon motion capture system [35]) for a commanded sinusoidal signal of a fixed frequency. We use a 3Hz cutoff filter to remove periodicities and numerical noise from the differentiation of the position signals.

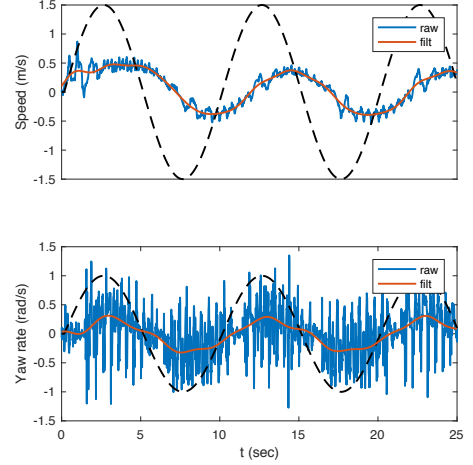


Fig. 3. Frequency domain characterization of Minitaur's walking trot response to smooth input signals  $v_d, \omega_d$  (1): time-domain plots of raw speed  $v$  and yaw response  $\omega$  (blue), with a 3Hz cutoff filter (red), and the reference signals  $v_d, \omega_d$  (black dashed).

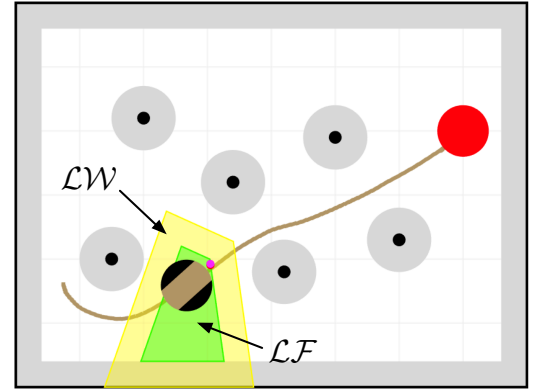


Fig. 4. Reactive navigation with local but "bird's eye" information: A depiction of the "local workspace"  $\mathcal{LW}$  (yellow polygon) and "local free space"  $\mathcal{LF}$  (green polygon) concepts that illustrates the local nature of the control strategy [13]. The goal position is shown as a solid red disk, and the local goal as a dot on one edge of the local free space. The dark disks correspond to the physical obstacles, while the grey regions delimits the free space (for the robot's centroid) boundary. The trajectory corresponds to an experimental trial also shown in Fig. 8.

Similarly to the bounding gait trials, we observe lagged response in speed and better frequency tracking in yaw. Small magnitudes in both speed and yaw can be attributed to the low gains  $\sigma_v, \sigma_\omega$  we used, as well as physical limitations of the gait, which was developed for easier navigation over rough, uncluttered terrain rather than high-speed, energetically efficient motion.

## III. REACTIVE NAVIGATION ALGORITHMS

In this section, we give an overview of the reactive navigation schemes that guarantee almost global navigation in convex workspaces using only local knowledge of the environment. We find it important to distinguish between the reactive navigation algorithm using local but "bird's eye" information, implemented on top of the bounding gait, and the fully sensor-based reactive navigation algorithm,

implemented on top of the walking trot gait along with a LIDAR and a RF sensor, as described below.

In every case, it is assumed that the robot's motion is described by unicycle kinematics

$$\dot{x} = v \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}, \quad (2)$$

$$\dot{\theta} = \omega, \quad (3)$$

#### A. Reactive Navigation Using Local but “Bird’s Eye” Information

We use the algorithm in [13] as an example of a high-level strategy, capable of solving the navigation problem for a differential drive robot, in order to test the limits of the kinematic unicycle navigation template for bounding legged robots (as empirically validated in Section II-B) in a real-world setting.

In brief, its construction utilizes power diagrams—generalized Voronoi diagrams with additive weights [36]—to identify a local workspace  $\mathcal{LW}$  and a collision-free local free space  $\mathcal{LF} \subset \mathcal{LW}$  of a disk-shaped robot in a sphere world, and continuous motion towards the closest point in the robot's local safe neighborhood to a designated goal location is proven to asymptotically drive almost all robot configurations to the destination location with no collisions along the way, as in the example shown in Fig. 4 with Minitaur bounding towards a goal location. It is also shown in [13] that this construction can be further adapted to a nonholonomically constrained “unicycle” robot model while maintaining the stability and collision avoidance properties.

#### B. Sensor-Based Reactive Navigation

The algorithm in [13] was extended in [14], by replacing the Voronoi power diagrams with separating hyperplanes to account for a broader than spheres class of convex bodies and to accommodate a realistic 2D LIDAR sensor model for obstacle detection. As shown in Fig. 5, the algorithm relies again on the construction of a local workspace  $\mathcal{LW}$  and a collision-free local free space  $\mathcal{LF} \subset \mathcal{LW}$  and continuous motion towards the closest point in the local free space brings the robot to a designated goal location. However, as shown in [14] and in Fig. 5, the construction of these cells is now based on the intersection of the (local) LIDAR footprint with appropriately defined hyperplanes, one for each local minimum observed within this footprint.

### IV. BODY FRAME TARGET LOCALIZATION

In the sensor-based framework of Section III-B, the problem of homing on a beacon using range-only measurements can become quite challenging (see e.g [3]). In the absence of global information, both the target localization and the navigation control strategy must be appropriately modified for the robot's body frame [20], [37]. Thankfully, as we will see next, the algorithm in Section III-B can be reformulated in the robot's body frame, provided successful target localization. To facilitate our analysis, we refer to Fig. 6.

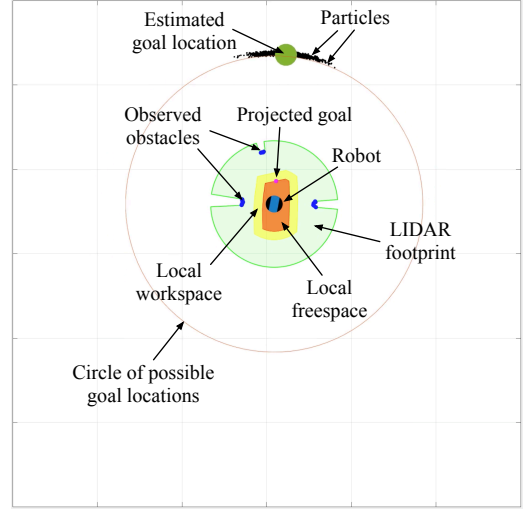


Fig. 5. Sensor-Based Reactive Navigation: A depiction of the “local workspace”  $\mathcal{LW}$  (yellow polygon) and “local free space”  $\mathcal{LF}$  (orange polygon) constructed from a LIDAR footprint (green) [14]. The estimated goal position (dark green dot) is calculated using range-only information and a particle filter. Notice how the particles spread on the circle with radius equal to the projection of the current range measurement. The local goal is computed from the projection of the estimated goal position onto  $\mathcal{LF}$ .

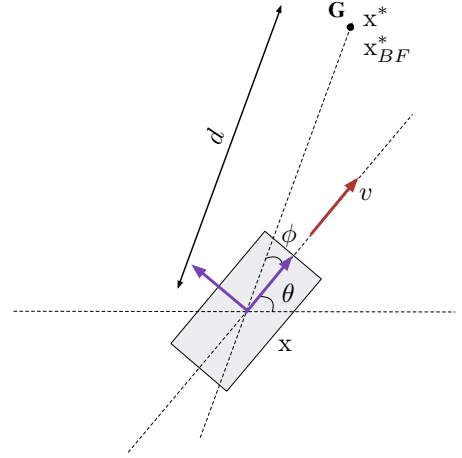


Fig. 6. Range-only target localization in the robot's body frame (purple).

#### A. Localization Model

We assume that the robot is located at  $x$  with an orientation  $\theta$ , which are both unknown. The goal is to navigate to point  $G$ , whose position in the global frame  $x^* := (x^*, y^*) \in \mathbb{R}^2$  is also unknown to the robot. The robot can only measure (with some accuracy) its distance  $d := \|x^* - x\|$  from  $G$ . Let  $x_{BF}^* := (x_{BF}^*, y_{BF}^*) \in \mathbb{R}^2$  denote the target position in the robot's body frame.

**Lemma 1** For the unicycle dynamics described in (2)-(3), if  $x^* = \text{const.}$ , then

$$\dot{x}_{BF}^* = -v + \omega y_{BF}^* \quad (4)$$

$$\dot{y}_{BF}^* = -\omega x_{BF}^* \quad (5)$$

*Proof.* In the global frame, define  $r_x := x^* - x = d \cos(\phi +$



$\theta$ ),  $r_y := y^* - y = d \sin(\phi + \theta)$ , with  $\phi = \arctan2(x_{BF}^*, y_{BF}^*)$ . Since the goal does not move, we see that  $\dot{r}_x = -\dot{x} = -v \cos \theta$  and  $\dot{r}_y = -\dot{y} = -v \sin \theta$ . On the other hand, we see from Fig. 6 that

$$\begin{aligned}\phi + \theta &= \arctan\left(\frac{y_{BF}^* - y}{x_{BF}^* - x}\right) = \tan^{-1}\left(\frac{r_y}{r_x}\right) \\ \Rightarrow \dot{\phi} + \omega &= \frac{1}{d^2}(\dot{r}_y r_x - \dot{r}_x r_y) = \frac{v}{d} \sin \phi \\ \Rightarrow \dot{\phi} &= \frac{v}{d} \sin \phi - \omega\end{aligned}$$

from the definitions above. Also, since  $d = \sqrt{r_x^2 + r_y^2}$ , we can easily derive  $\dot{d} = \frac{\dot{r}_x r_x + \dot{r}_y r_y}{d} = -v \cos \phi$ . Focusing now on the robot's body frame, we can see that  $x_{BF}^* = d \cos \phi$  and  $y_{BF}^* = d \sin \phi$ , so that by differentiation  $\dot{x}_{BF}^* = \dot{d} \cos \phi - d \dot{\phi} \sin \phi$  and  $\dot{y}_{BF}^* = \dot{d} \sin \phi + d \dot{\phi} \cos \phi$ . Simple substitution of  $\dot{d}$  and  $\dot{\phi}$  from above yields  $\dot{x}_{BF}^* = -v + \omega y_{BF}^*$  and  $\dot{y}_{BF}^* = -\omega x_{BF}^*$  and this concludes the proof. ■

### B. Measurement and Estimation

We use the localization model laid out in (4)-(5) to perform state estimation for  $x_{BF}^*$  using a particle filter [38] implemented in the ParticleFilter class of the MATLAB Robotics toolbox [39]. We assume that the only measurement provided for the propagation of the particle filter is the distance of the robot to the target  $d = \sqrt{(x_{BF}^*)^2 + (y_{BF}^*)^2}$  and use a measurement model of the form

$$y(t) = d(t) + \epsilon(t) \quad (6)$$

with  $\epsilon(t)$  representing the measurement noise. We note here that various statistical distributions have been considered for  $\epsilon(t)$  in the RF literature, but, consistent with other work [40], a Gaussian distribution with mean zero and a specified standard deviation according to the range sensor's characteristics was determined to be sufficient for our purposes.

By supplying an initial estimate for  $x_{BF}^*$ , an initial estimate covariance  $\Sigma_0$ , suitable process noise estimates for the proprioceptive linear speed and yaw rate provided by the robot, a suitable measurement noise standard deviation and a proper number of particles (please refer to Section V-B for more details), the particle filter provides an estimate of the goal location  $x_{BF}^*$ , which is constantly updated and gets better as the robot moves. Some tuning on the number of particles is required to balance between the needs for fast filter updates and the achievement of good convergence properties. It must be noted that the problem of beacon homing using RF sensors is worthy of independent study due to issues related to multipath interference etc., which go beyond the scope of this work.

### C. Body Frame Navigation Algorithm

With the localization algorithm supplying an estimate of  $x_{BF}^* = (x_{BF}^*, y_{BF}^*)$  at every control iteration already in place, we construct the homing behavior by writing the control law in [13, Equation 33] in the body frame and setting

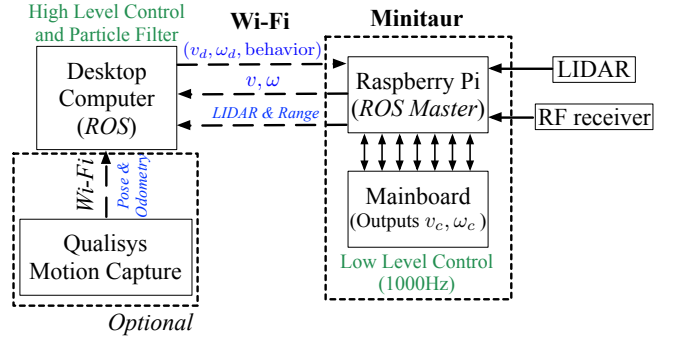


Fig. 7. A schematic demonstrating the system structure of the experimental setup. Minitaur's Raspberry Pi, the central element of this configuration, acts as the ROS Master and forwards any LIDAR and range readings. The external computer runs the high level controller which gives the desired linear and angular velocities  $v_d, \omega_d$ , while Minitaur's mainboard runs the low level controller by calculating the actual commands  $v_c, \omega_c$  using (1), and provides proprioceptive speed and yaw rate feedback  $v, \omega$ , forwarded to the desktop computer by the Raspberry Pi.

the inputs

$$v_d = k \bar{x}_{v,BF}^* \quad (7)$$

$$\omega_d = k \arctan(\bar{y}_{\omega,BF}^* / \bar{x}_{\omega,BF}^*) \quad (8)$$

with  $\bar{x}_{v,BF}^* = (\bar{x}_{v,BF}^*, \bar{y}_{v,BF}^*)$  and  $\bar{x}_{\omega,BF}^* = (\bar{x}_{\omega,BF}^*, \bar{y}_{\omega,BF}^*)$  the linear and angular projected goals respectively, calculated by projecting  $x_{BF}^*$  on the local free space  $\mathcal{LF}$  as described in [13] and demonstrated in Fig. 5. In this way, we have constructed a minimalistic sensory-driven approach to the homing problem, that uses a LIDAR for obstacle avoidance and an RF sensor, providing only one-dimensional information (range), for the target location.

## V. EXPERIMENTAL SETUP

Here, we detail the ROS networked environment, in which Minitaur operates, that generates its high level ("unicycle-like") control inputs by implementing the reactive navigation algorithm summarized in either Section III-A (bounding) or Section III-B (walking). As shown in Fig. 7, this environment consists of a computer implementing the high-level controller and of Minitaur's ROS infrastructure, exchanging messages over a Wi-Fi network.

In order to provide a hardware abstraction commensurate with the behavioral abstraction of Sec. II-B, Minitaur's computational subsystem is enhanced with a Raspberry Pi Model 3, which is able to both run ROS and connect to a Wi-Fi access point. A custom ROS node on the Raspberry Pi receives  $(v_d, \omega_d)$  and the desired mode of operation (bounding, walking, standing) as ROS messages (from the desktop computer) and forwards them to the Minitaur mainboard (microcontroller implementing the functionalities shown in (1) to produce the actual commands  $v_c, \omega_c$ ) at 100Hz over a 115.2 Kbps USART connection. The Raspberry Pi acts as the ROS Master that resolves networking for the rest of the ROS nodes: a dedicated ROS node is activated as soon as the system boots and automatically subscribes to the  $(v_d, \omega_d)$

ROS topics, as well as an additional one capable of defining the desired behavior.

#### A. Bounding-Specific Infrastructure Components

In the case of bounding and in the absence of any onboard sensor, the odometry information consisting of the linear speed  $v$  and the yaw rate  $\omega$  is extracted from a Qualisys Motion Capture System [41] (QMCS) at 100 Hz, using a set of motion capture cameras positioned around a  $20\text{m} \times 6\text{m}$  arena. The desktop computer receives the online data from QMCS using the ROS package `mocap_qualisys` [42] and outputs specific desired linear and angular velocity values  $(v_d, \omega_d)$  for Minitaur as described in Section III-A. The high level control loop runs at approximately 100Hz, which is more than enough for the robot to recover if any obstacle is detected, and the low-level (bounding or walking trot) controller runs at 1KHz.

#### B. Walking-Specific Infrastructure Components

For the walking trot experiments, the setup is enhanced with two Pulson P-440 RF modules [43] (one beacon for the goal and one receiver for the robot), along with a Hokuyo UTM-30LX LIDAR [44].

Since the fully sensor-based navigation approach described in Sections III-B and IV is used, a second ROS node reads the proprioceptive odometry feedback<sup>2</sup> from the mainboard and forwards it to the desktop computer for use in the particle filter propagation [38]. Also, a third node, adapted from the ROS library in [47], is responsible of sending the range measurements from the RF sensor to the desktop computer. A final ROS node, taken from [48], forwards the LIDAR measurements to the desktop computer.

The desktop computer is responsible for running the high-level control algorithm outlined in Section III-B, along with the particle filter propagation for target localization, as described in Section IV. For the particle filter, we use a process noise of 0.2m/s for the linear speed and 0.4rad/s for the angular speed. Also, a range measurement noise (standard deviation) of 10cm is used, consistent with the Pulson P-440 RF module datasheet. We use 2000 particles, systematic resampling and an effective particle ratio of 0.8.

As we show in Section VI and the accompanying video, this infrastructure works robustly and without any discernible network-induced latency. The high level control loop here is slower and runs at approximately 50Hz, since several sensor readings have to be sent and processed, but this frequency is still more than enough for the robot to recover if any obstacle is detected.

## VI. EXPERIMENTAL RESULTS

We illustrate the qualitative features and performance of the navigation algorithm by presenting empirical results for both the bounding and the walking trot gait. Section VI-A reports on experiments run using the bounding gait and local but “bird’s eye” information as described in Section III-A in a

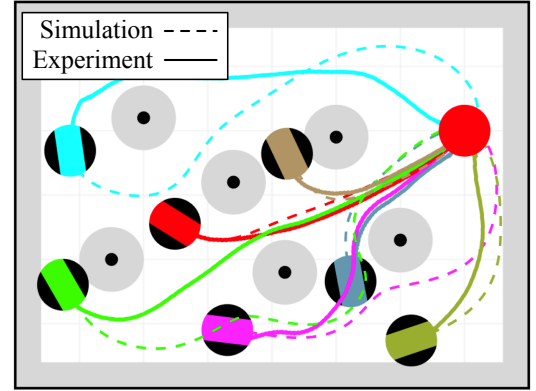


Fig. 8. Trajectories extracted from simulations and bounding experiments in a small and dense with obstacles environment. The goal position (shown as a solid red disk) is fixed but initial robot configurations vary.

small and cluttered rectangular environment with disk-shaped obstacles, and Section VI-B describes the results in a similar workspace with the walking gait, the use of the sensor-based navigation algorithm described in Section III-B and the range-only target localization scheme presented in Section IV. In all of our experiments, Minitaur is approximated as a disk-shaped robot with radius<sup>3</sup> 0.4 m, and a margin of 0.1 m is added to the robot’s radius for safety reasons.

#### A. Bounding Experiments

Fig. 8 depicts our results in the small and obstacle-dense environment. The workspace is rectangular with length 8 m and width 6 m; the obstacles have common radius  $\rho = 0.1$  m and are randomly placed throughout the environment (also see Fig. 1). The goal position is near the top right corner of the workspace behind several obstacles.

As it is evident from Fig. 8, Minitaur manages to converge to the desired location from a variety of initial configurations. In a total of over 50 trials, Minitaur reaches the goal and avoids all the obstacles each time. In Fig. 8, we also overlay trajectories from a MATLAB simulation of a differential-drive robot with the same initial conditions and similar control gains. The simulation and physical platform follow similar trajectories in 4 out of the 7 cases. Even when the trajectory is quite different, the robot always safely navigates to the goal location.

Finally, to illustrate Minitaur’s bounding performance as a kinematic unicycle (Sec. II-B), we plot in Fig. 9 Minitaur’s response to the commanded fore-aft and yaw speeds during an experimental trial. Similarly to Section II, we use a 3Hz cutoff filter to remove periodicities caused by Minitaur’s bounding.

#### B. Sensor-Based Walking Experiments

As mentioned in Section II, the change from bounding to walking allows for the use of the fully sensor-based algorithm described in Sections III-B and IV. Fig. 10 depicts our results

<sup>2</sup>Here the forward speed  $v$  is estimated with the use of leg kinematics as shown in [45] and  $\omega$  is provided by a VN-100 IMU [46].

<sup>3</sup>Minitaur’s length (hip-to-hip) is 0.4 m and an extra length of 0.4 m due to fore and hind leg extensions in the sagittal plane (typically about 0.2 m) has to be accounted for.

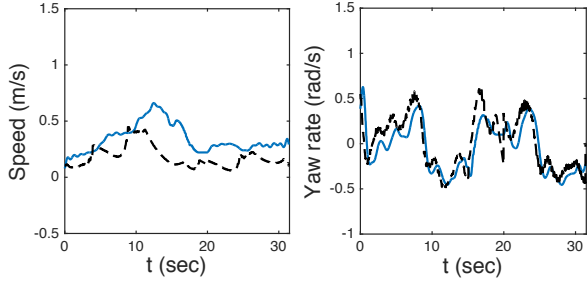


Fig. 9. Minitaur’s response (blue) to speed and yaw reference signals (black) during a bounding experimental trial.

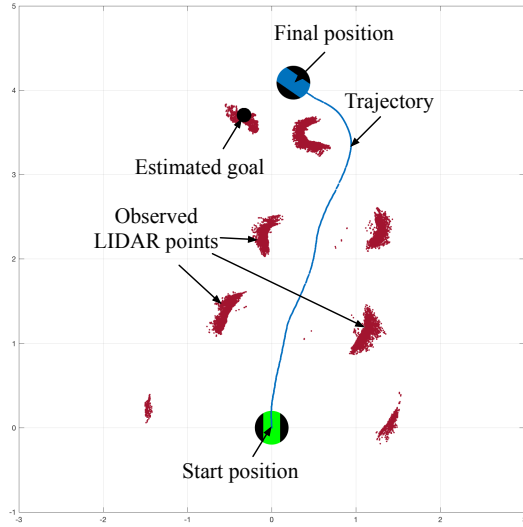


Fig. 10. A suggestive path reconstructed from Minitaur’s proprioceptive data in the environment shown in Fig. 1. The black dot corresponds to the (converged) estimated goal location at the end of the trial. The brown points consist the corresponding pointcloud of observed obstacle points; in the absence of ground-truth their exact location cannot be precisely determined.

in a workspace cluttered with obstacles of common radius  $\rho = 0.1$  m and randomly placed throughout the environment. Because of the lack of a portable ground-truth mechanism, the path shown in Fig. 10 was obtained by numerically integrating all the saved proprioceptive linear speed  $v$  and yaw rate  $\omega$  estimates, and is thus suggestive but not exact. This also explains the non-convex shape of the observed “obstacles” in the workspace, reconstructed from the union of all the LIDAR readings. From this figure, it is evident that the robot managed to successfully localize the target, navigate there and stop within a predefined distance from it. In the absence of ground-truth, we plot in Fig. 11 the range measurements obtained by the RF sensor for several trials, showing convergence to the target. Finally, to illustrate Minitaur’s walking performance as a kinematic unicycle (Sec. II-C), we plot in Fig. 12 its response to commanded fore-aft and yaw speeds during an experimental trial.

In the accompanying video, we demonstrate several of the numerous successful experimental trials for both bounding and walking, with the robot indefatigably seeking the goal.

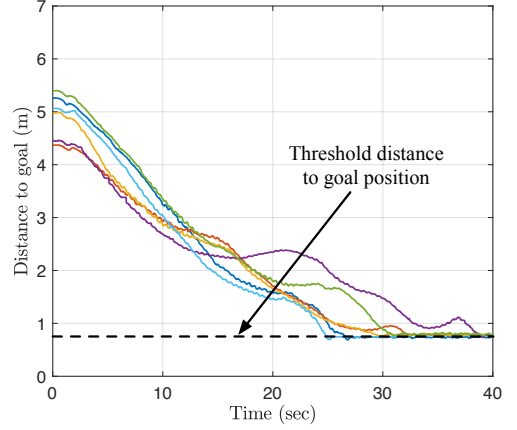


Fig. 11. The distance to the goal position as a function of time for several initial conditions with the walking trot gait. In every case, the robot was commanded to stop as soon as it got within a distance of 0.8m from the target position.

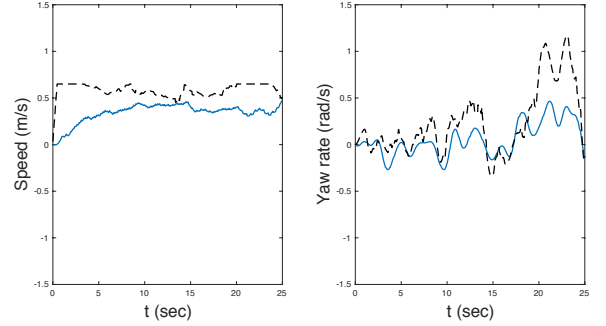


Fig. 12. Minitaur’s response (blue) to speed and yaw reference signals (black) during a walking trot experimental trial.

## VII. CONCLUSION

This work demonstrates the empirical anchoring of a kinematic unicycle model on the dynamically complicated bounding and walking trot gaits of a quadrupedal robot and the robustness and efficiency of a sensor-based doubly reactive homing scheme, as an example of a high level motion planning strategy for legged robots. The realization of this algorithm in a GPS-denied environment is largely enabled by a proposed body-frame, range-only target localization algorithm which uses one-dimensional, range information to estimate the goal position in the body frame. The empirical results to date are very promising: the robot is driven to the desired goal location from any initial position and configuration in the workspace, while avoiding obstacles.

Work currently in progress aims to make formal arguments about the anchoring of the kinematic unicycle model on different gaits and implement this navigation strategy with range-only target localization in an outdoor setting.

## VIII. ACKNOWLEDGEMENTS

This work was supported in part by AFRL grant FA865015D1845 (subcontract 669737-1) and in part by ONR grant #N00014-16-1-2817, a Vannevar Bush Fellowship held

by the last author, sponsored by the Basic Research Office of the Assistant Secretary of Defense for Research and Engineering.

## REFERENCES

- [1] P. Jantapremjit and P. A. Wilson, "Optimal control and guidance for homing and docking tasks using an autonomous underwater vehicle," in *International Conference on Mechatronics and Automation*, 2007, pp. 243–248.
- [2] P. Batista, C. Silvestre, and P. Oliveira, "A sensor-based controller for homing of underactuated AUVs," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 701–716, 2009.
- [3] G. Vallicrosa and P. Ridao, "Sum of gaussian single beacon range-only localization for AUV homing," *Annual Reviews in Control*, vol. 42, no. Supplement C, pp. 177 – 187, 2016.
- [4] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.
- [5] —, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, 1989.
- [6] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [7] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.
- [8] D. E. Koditschek, "The control of natural motion in mechanical systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 113, pp. 547–551, 1991.
- [9] A. Rizzi, "Hybrid control as a method for robot motion programming," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, 1998, pp. 832–837.
- [10] G. A. D. Lopes and D. E. Koditschek, "Navigation functions for dynamical, nonholonomically constrained mechanical systems," *Advances in Robot Control*, pp. 135–155, 2006.
- [11] G. Roussos, D. Dimarogonas, and K. Kyriakopoulos, "3d navigation and collision avoidance for a non-holonomic vehicle," in *American Control Conference*, 2008, pp. 3512–3517.
- [12] D. Conner, H. Choset, and A. Rizzi, "Integrating planning and control for single-bodied wheeled mobile robots," *Autonomous Robots*, vol. 30, no. 3, pp. 243–264, 2011.
- [13] O. Arslan and D. E. Koditschek, "Exact robot navigation using power diagrams," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1–8.
- [14] O. Arslan and D. E. Koditschek, "Sensor-based reactive navigation in unknown convex sphere worlds," in *Workshop on Algorithmic Foundations of Robotics*, Dec 2016, p. (in press).
- [15] J. Burdick and B. Goodwine, "Quasi-static legged locomotors as nonholonomic systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2000.
- [16] B. Goodwine and J. Burdick, "Gait controllability for legged robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1998.
- [17] D. Zarrouk, D. W. Haldane, and R. Fearing, "Dynamic legged locomotion for palm-size robots," in *SPIE DSS*, 2015.
- [18] B. D. Ilhan, A. Johnson, and D. E. Koditschek, "Autonomous legged hill ascent," *Journal of Field Robotics*, p. (under review), 2016.
- [19] U. Saranlı, M. Buehler, and D. E. Koditschek, "Rhex: A simple and highly mobile hexapod robot," *The International Journal of Robotics Research*, vol. 20, no. 7, p. 616, 2001.
- [20] A. De and D. E. Koditschek, "Toward dynamical sensor management for reactive wall-following," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 2400–2406.
- [21] A. De, K. Bayer, and D. Koditschek, "Active sensing for dynamic, non-holonomic, robust visual servoing," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 6192–6198.
- [22] U. Saranlı and D. E. Koditschek, "Template based control of hexapedal running," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2003, pp. 1374–1379.
- [23] R. Altendorfer, D. Koditschek, and P. Holmes, "Stability analysis of a clock-driven rigid-body SLIP model for RHex," *The International Journal of Robotics Research*, vol. 23, no. 1011, p. 10011012, 2004.
- [24] A. De and D. E. Koditschek, "Parallel composition of templates for tail-energized planar hopping," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4562–4569.
- [25] M. Raibert, *Legged Robots that Balance*. MIT Press, 1986.
- [26] H.-W. Park, S. Park, and S. Kim, "Variable-speed quadrupedal bounding using impulse-planning: Untethered high-speed 3d running of MIT cheetah 2," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5163–5170.
- [27] I. Poulakakis *et al.*, "Modeling and experiments of untethered quadrupedal running with a bounding gait: The Scout II robot," *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 239–256, 2005.
- [28] R. J. Full and D. E. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *Journal of Experimental Biology*, vol. 202, no. 23, pp. 3325–3332, Dec. 1999.
- [29] A. De and D. E. Koditschek, "Vertical hopper compositions for preflexive and feedback-stabilized quadrupedal bounding, pacing, prinking and trotting," *under review*, Dec. 2016.
- [30] G. Kenneally, A. De, and D. Koditschek, "Design principles for a family of direct-drive legged robots," *IEEE Robotics and Automation Letters*, 2016.
- [31] "Ghost robotics minitaur," 2016. [Online]. Available: <http://www.ghostrobotics.io/minitaur/>
- [32] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert, "Autonomous navigation for BigDog," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 4736–4741.
- [33] D. Panagou and H. Tanner, "Modeling of a hexapod robot; kinematic equivalence to a unicycle," *UDMETR-2009-0001*, 2009.
- [34] P. Ogren and N. E. Leonard, "Obstacle avoidance in formation," in *IEEE International Conference on Robotics and Automation*, 2003, p. 24922497.
- [35] Vicon. Vantage V16. [Online]. Available: <https://www.vicon.com/file/vicon/vantagebrochure-021216-web-94650.pdf>
- [36] F. Aurenhammer, "Power diagrams: Properties, algorithms and applications," *SIAM Journal on Computing*, vol. 16, no. 1, pp. 78–96, 1987.
- [37] E. W. Justh and P. S. Krishnaprasad, "Natural frames and interacting particles in three dimensions," in *IEEE Conference on Decision and Control*, 2005, pp. 2841–2846.
- [38] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- [39] P. Corke, *Robotics, Vision & Control*. Springer, 2017.
- [40] D. Kurth, G. Kantor, and S. Singh, "Experimental results in range-only localization with radio," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2003, pp. 974–979.
- [41] Qualisys. Oqus platform. [Online]. Available: <http://www.qualisys.com/cameras/oqus/>
- [42] KumarRobotics. Drivers for motion capture systems. [Online]. Available: [https://github.com/KumarRobotics/motion\\_capture\\_system](https://github.com/KumarRobotics/motion_capture_system)
- [43] Time Domain. P-440. [Online]. Available: <http://www.timedomain.com/products/pulson-440>
- [44] Hokuyo. UTM-30LX. [Online]. Available: <https://www.hokuyo-aut.jp/search/single.php?serial=169>
- [45] G. Kenneally and D. E. Koditschek, "Leg design for energy management in an electromechanical robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.
- [46] VectorNav Technologies. VN-100. [Online]. Available: <https://www.vectornav.com/products/vn-100>
- [47] [Online]. Available: [https://github.com/ashariati/pulson\\_ros](https://github.com/ashariati/pulson_ros)
- [48] [Online]. Available: [http://wiki.ros.org/urg\\_node](http://wiki.ros.org/urg_node)