# *Sensor-Based Reactive Execution of Symbolic Rearrangement Plans by a Legged Mobile Manipulator*

# Sensor-Based Reactive Execution of Symbolic Rearrangement Plans by a Legged Mobile Manipulator

Vasileios Vasilopoulos, T. Turner Topping, William Vega-Brown, Nicholas Roy, Daniel E. Koditschek

***Abstract*— We demonstrate the physical rearrangement of wheeled stools in a moderately cluttered indoor environment by a quadrupedal robot that autonomously achieves a user's desired configuration. The robot's behaviors are planned and executed by a three layer hierarchical architecture consisting of: an offline symbolic task and motion planner; a reactive layer that tracks the reference output of the deliberative layer and avoids unanticipated obstacles sensed online; and a gait layer that realizes the abstract unicycle commands from the reactive module through appropriately coordinated joint level torque feedback loops. This work also extends prior formal results about the reactive layer to a broad class of nonconvex obstacles. Our design is verified both by formal proofs as well as empirical demonstration of various assembly tasks.**

## I. INTRODUCTION

### A. Motivation

In this paper, we apply and extend prior formal results [1] to solve and empirically demonstrate a geometrically modest but mechanically challenging instance of the Warehouseman's problem [2]. We recruit the Minitaur quadruped [3] (see Fig. 1) as a legged "mobipulator" [4] — a mobile robot that uses only its native, general purpose mechanical appendages to effect work on itself and the surrounding environment — in order to re-arrange according to a user's command the location of wheeled stools in a known environment that is sparsely obstructed by unanticipated, immovable obstacles of unknown general placement and shape. The integration of deliberative and reactive layers introduced in [1] guarantees that a unicycle capable of pushing or releasing such wheeled stools at will must always accomplish its task so long as the unanticipated objects are all convex and sufficiently sparsely placed relative to the known floor plan.

Here, we seek to bring a greater degree of realism to that framework by two different but allied extensions. We relax the geometric restriction to convex obstacles and prove that the idealized unicycle will still succeed even when confronted with nonconvex unanticipated objects at run-time, so long as they are "moderately curved" and "sufficiently sparse". We relax the mechanical assumption of an idealized gripper by adding an entirely new "gait layer" that translates the erstwhile unicycle's velocity and gripper commands into a Minitaur joint-level architecture taking the form that we conjecture meets the requirements of a simple hybrid dynamical manipulation and self-manipulation system [5]. Although this would insure at least that the hybrid system is guaranteed to be live and non-blocking [5], we are only in the early stages of working out the formal relationships of the legged dynamics to the abstracted unicycle reference.

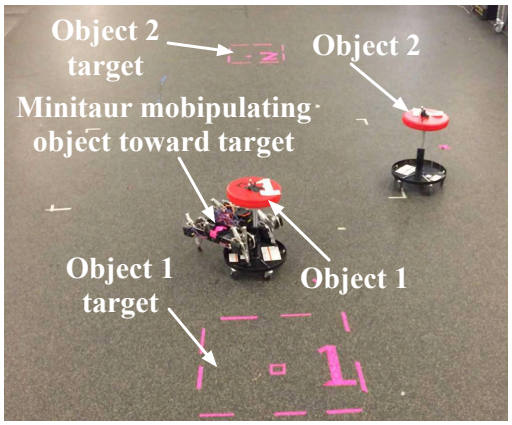


Fig. 1. LIDAR-equipped Minitaur [3] mobipulating [4] two stools using gaits [6] called out by a deliberative/reactive motion planner [1].

### B. Prior Work

Substantial computational benefits [7] can reward planners that commit to decisions at a high level of abstraction [8], learning STRIPS-like symbols [9] or performing an efficient satisficing search [10]. In contrast, aiming for highly dynamic implementation in uncertain, unstructured settings, in [1] we introduced a provably correct architecture for completing assembly plans in 2D environments based on the integration of a vector-field reactive controller with a deliberative planner [11] that uses the angelic semantics [12] to guarantee hierarchical optimality. While high level deliberation has previously been coordinated with reactive planners on hardware-specialized physical systems [13], [14], harnessing the inherent relationship between mobility and manipulation [4], as well as the potential for dynamics to ameliorate kinematic task mismatches [15], can preserve platform generality and, thereby, its fitness for a greater diversity of tasks.

### C. Contributions and Organization of the Paper

In this paper, we suggest by empirical demonstration the effectiveness of a hierarchical control structure (depicted in

Vasileios Vasilopoulos is with the Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA 19104, vvasilo@seas.upenn.edu

T. Turner Topping and Daniel E. Koditschek are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, {ttopping,kod}@seas.upenn.edu

William Vega-Brown and Nicholas Roy are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, {wrvb,nickroy}@csail.mit.edu

Fig. 2) for a highly dynamic physical system (illustrated in Fig. 3). Specifically, we believe this is the first provably correct deliberative/reactive planner to engage an unmodified general purpose mobile manipulator in physical rearrangement of its environment. At the same time, targeting more geometrically realistic environments as well, our new results offer the first formal extension of the recent online, doubly-reactive controller family (originated in [16] and extended to our line tracking application in [17]) to nonconvex obstacles. We believe that our "length scale" interpretation of prox-regularity [18] used to distinguish suitably "moderate" nonconvexities will also have independent broad future applicability in these settings.

Section II describes the problem and summarizes our approach. Section III describes each component of the hierarchical control structure (deliberative, reactive and gait controller) separately and Section IV presents our formal results on reactive wall following for nonconvex obstacles. Section V begins with the description of our hardware infrastructure based on ROS and continues with the presentation of our empirical results for different classes of experiments. Finally, Section VI concludes with some comments and ideas for future work.

## II. PROBLEM FORMULATION

In this work, Minitaur is assumed to operate in a closed and compact workspace $\mathcal{W} \subset \mathbb{R}^2$ whose boundary $\partial\mathcal{W}$ is assumed to be known, and is tasked to move each of $n \in \mathbb{N}$ movable disk-shaped objects, centered at $\mathbf{p} := (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n) \in \mathcal{W}^n$ with a vector of radii $(r_1, r_2, \ldots, r_n) \in (\mathbb{R}_{>0})^n$, from their initial configuration to a user-specified goal configuration $\mathbf{p}^* := (\mathbf{p}_1^*, \mathbf{p}_2^*, \ldots, \mathbf{p}_n^*) \in \mathcal{W}^n$. For our hardware implementation, the movable objects are stools with five caster wheels. We assume that both the initial configuration and the target configuration of the objects are known. In addition to the known boundary of the workspace $\partial\mathcal{W}$, the workspace is cluttered by an unknown number of fixed, disjoint, potentially nonconvex obstacles of unknown position and size, denoted by $\mathcal{O} := (O_1, O_2, \ldots)$. To simplify the notation, we also define $\mathcal{O}_w := \mathcal{O} \cup \partial\mathcal{W}$.

For (reactive) planning purposes, Minitaur is modeled as a first-order, nonholonomically-constrained, disk-shaped robot, centered at $\mathbf{x} \in \mathbb{R}^2$ with radius $r \in \mathbb{R}_{>0}$ and orientation $\psi \in S^1$, following our previous work [1]. The model dynamics are described by

$$(\dot{\mathbf{x}}, \dot{\psi}) = \mathbf{B}(\psi)\mathbf{u}_{ku} \tag{1}$$

with $\mathbf{B}(\psi) := \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}^T$ the differential constraint matrix and $\mathbf{u}_{ku} := (v, \omega)$ the input vector[1] consisting of a linear and an angular command. Similarly to [1], we adopt the following assumptions to facilitate the proofs of our formal results, which are not necessary for the existence of some solution to the problem.

[1]Throughout this paper, we will use the ordered set notation $(*, *, \ldots)$ and the matrix notation $\begin{bmatrix} * & * & \ldots \end{bmatrix}^T$ for vectors interchangeably.

**Assumption 1** (Obstacle separation) *The (potentially nonconvex) obstacles $\mathcal{O}$ in the workspace are separated from each other by clearance[2] of at least $d(O_i, O_j) > 2(r + \max_k r_k), i \neq j$, with $k$ an index spanning the set of movable objects. They are also separated from the boundary of the (potentially nonconvex) workspace $\mathcal{W}$ by at least $d(O_i, \partial\mathcal{W}) > 2(r + \max_k r_k)$ for all $i$.*

**Assumption 2** (Admissible object goals) *For any object $i \in \{1, \ldots, n\}$, $d(\mathbf{p}_i^*, \mathcal{O}_w) > r_i + 2r$.*

The robot is assumed to have access to its state[3] $(\mathbf{x}, \psi)$ and to possess a LIDAR for local obstacle avoidance, positioned at $\mathbf{x}$, with a $360^\circ$ angular scanning range and a fixed sensing range $R \in \mathbb{R}_{>0}$. It is also assumed to use a gripper for moving objects, which can be engaged or disengaged; we will write $g = 1$ when the gripper is engaged and $g = 0$ when it is disengaged. Of course, Minitaur is only an imperfect unicycle [20] and does not actually possess a gripper; it has to successfully coordinate its limbs and walk while following a path, avoid an obstacle or lock an object in place and move it to a desired location. Hence, the reactive planner's commands $(\mathbf{u}_{ku}, g)$ must in turn be translated to appropriate low-level commands on the robot's joint level.

The aforementioned description imposes a hierarchical structure, as shown in Fig. 2. The deliberative planner is endowed with a symbolic command set comprised of three actions: MOVETOOBJECT$(i, \mathcal{P})$, POSITIONOBJECT$(i, \mathcal{P})$ and MOVE$(\mathcal{P})$. Here $i$ is the desired object and $\mathcal{P}$ is a piecewise continuously differentiable path $\mathcal{P} : [0, 1] \to \mathcal{W}$ connecting an initial and a final position, which can be seen as a "geometric suggester" in the sense of [8] (see [1] for more details). This command set suggests the following problem decomposition into the complementary sub-problems:

1) In the *deliberative layer*, find a *symbolic plan*, i.e., a sequence of symbolic actions whose successful implementation is guaranteed to complete the task.
2) In the *reactive layer*, implement each of the symbolic actions by finding appropriate commands $(\mathbf{u}_{ku}, g)$ according to the robot's equations of motion shown in (1), while avoiding the perceived obstacles (unanticipated by the deliberative planner) encountered along the way.
3) In the *gait layer*, use a hybrid dynamical systems framework with simple guard conditions to choose between constituent gaits, providing a unicycle interface to the reactive layer, controllable by $(\mathbf{u}_{ku}, g)$, regardless of the state of the agent and objects.

## III. SYSTEM ARCHITECTURE

In this section, we describe the three-layer architecture used to accomplish the task at hand, shown in Fig. 2. After a description of the offline deliberative planner, we proceed with the features of the online reactive module and the new,

[2]Here the clearance between two sets $A$ and $B$ is defined as $d(A, B) := \inf\{\|\mathbf{a} - \mathbf{b}\| \,|\, \mathbf{a} \in A, \mathbf{b} \in B\}$

[3]Since legged state estimation falls beyond the scope of this work, localization is performed using a Vicon motion capture system [19].

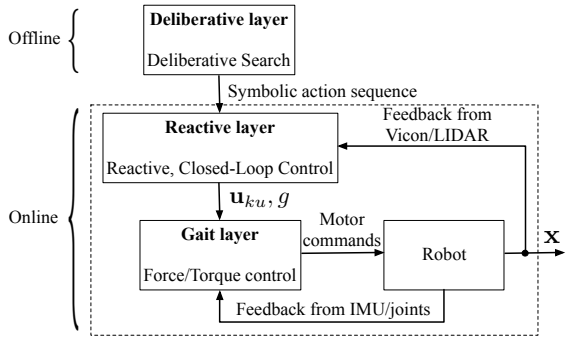


Fig. 2. A coarse block diagram of the planning and control architecture. In the deliberative layer, a high-level planner [11] outputs a sequence of symbolic actions that are realized and executed sequentially using a reactive controller that issues unicycle velocity ($\mathbf{u}_{ku}$) [1] and abstract gripper ($g$) commands (see Section III-B). The low-level gait layer uses the commands instructed by the reactive planner to call out appropriately parametrized joint-level feedback controllers (see [6] and Section III-C) for Minitaur.

low-level layer of control (the "gait" layer), used to achieve on Minitaur the commands instructed by the reactive layer.

### A. Deliberative Layer

The deliberative layer finds a feasible path through the joint configuration space of the robot and anticipated environment. It takes as input a metric description of the world, including object and obstacle geometry, and proceeds in two stages. First, it discretizes the environment by constructing a factored random geometric graph [21]. The factored graph is the product of $n+1$ probabilistic roadmaps, one for each object and one for the robot. Each edge in the factored graph represents a feasible motion; these motions are either paths of the robot while the other objects do not move, or paths of the robot carrying a single object. Paths through this graph then represent continuous paths through configuration space.

This graph construction is asymptotically optimal; as the number of vertices in each factor increases, the cost of the best path through the graph approaches the cost of the optimal path. In addition, the factored representation allows us to quickly construct graphs with an exponential number of vertices. However, the number of graph vertices is exponential in $n$. We can search for a near-optimal path through the graph in a reasonable amount of time using the angelic hierarchical A* algorithm [11], [12]. This algorithm interleaves the search over high-level decision, like which objects to grasp and in which order, and over lower-level details, like where objects should be placed, by using a hierarchy of abstract operators, which are implicitly-defined sets of plans that achieve a specified effect. For example, the operator MOVETOOBJECT$(i,\cdot)$ represents any plan that eventually reaches object $i$.

We can derive bounds on the cost of any primitive plan contained in an abstract operator. For example, the cost of any plan in MOVETOOBJECT$(i,\cdot)$ starting from a position $\mathbf{x}$ is greater than the Euclidean distance from $\mathbf{x}$ to object $i$. If we find some path from $\mathbf{x}$ to object $i$, its cost is an *upper* bound on the cost of the best plan from $\mathbf{x}$ to object $i$. Using these bounds, we can estimate the cost of plans composed of sequences of abstract operators, allowing us to prune bad plans early and refine promising plans first. More importantly, these bounds allow us to prove that a symbolic plan is feasible before providing it to the reactive layer.

### B. Reactive Layer

As shown in Fig. 2, the reactive layer is responsible for executing the symbolic action sequence, i.e. the output of the deliberative planner. The reader is referred to Algorithms 3, 4 and 5 of [22] for the reactive implementation of the MOVETOOBJECT, POSITIONOBJECT and MOVE symbolic actions. Here, similarly to [1], [22], we decompose the reactive behavior into two separate modes determined by the absence or presence of unanticipated obstacles:

*1) Anticipated Environment:* In the absence of unanticipated obstacles, the robot is in *path following mode*. Based on the results of [16], [17], this mode is responsible for steering the robot along a reference path $\mathcal{P}$ given by the deliberative planner. This is achieved by following the *projected-path goal* $\mathcal{P}(\alpha^*)$ with $\alpha^*$ determined as[4]

$$\alpha^* := \max\{\alpha \in [0,1] \,|\, \mathcal{P}(\alpha) \in B\left(\mathbf{x}, d(\mathbf{x}, \partial\mathcal{F})\right)\} \quad (2)$$

constantly updated as the agent moves along the path. Here $d(\mathbf{x}, \partial\mathcal{F})$ denotes the distance of the agent from the boundary of the free space $\mathcal{F}$, determined as

$$d(\mathbf{x}, \partial\mathcal{F}) = \min_{\theta} \rho_{\mathbf{x}}(\theta) - r \quad (3)$$

with $\rho_{\mathbf{x}}(\theta)$ denoting the polar curve[5] describing the LIDAR measurements [16].

*2) Unanticipated Obstacles:* In the presence of unanticipated obstacles, i.e., when $d(\mathbf{x}, \partial\mathcal{F}) < \epsilon$ with $\epsilon$ a desired tolerance, the robot switches to *wall following mode*. In this mode, the robot follows the *wall-following goal* $\mathbf{x}_p(\mathbf{x})$ defined as in [1]

$$\mathbf{x}_p(\mathbf{x}) := \mathbf{x}_{\text{offset}}(\mathbf{x}) + \frac{\epsilon}{2}\mathbf{n}_w(\mathbf{x}) + a\frac{\epsilon\sqrt{3}}{2}\mathbf{t}_w(\mathbf{x}) \quad (4)$$

with $\mathbf{x}_{\text{offset}}(\mathbf{x}) := \mathbf{x} - (\rho_{\mathbf{x}}(\theta_m) - r)\,\mathbf{n}_w(\mathbf{x})$ an offset point from the obstacle boundary, $\theta_m := \arg\min_{\theta} \rho_{\mathbf{x}}(\theta)$ the LIDAR angle corresponding to the minimum distance from the obstacle, $\mathbf{n}_w(\mathbf{x}) := -(\cos\theta_m, \sin\theta_m)$ the normal vector to the boundary of the obstacle at the point of minimum distance and $\mathbf{t}_w(\mathbf{x}) := (\sin\theta_m, -\cos\theta_m)$ the corresponding tangent vector. Finally, $a \in \{-1, 1\}$ denotes the wall following direction (1 for CCW motion and -1 for CW motion). The robot exits the wall following mode and returns to the path following mode once it encounters the path again, i.e. when $\alpha^* = \max\{\alpha \in [0,1] \,|\, \mathcal{P}(\alpha) \in B\left(\mathbf{x}, d(\mathbf{x}, \partial\mathcal{F})\right)\} > \alpha_s^*$, with $\alpha_s^*$ the saved path index at the beginning of the wall following mode. Based on the above definitions, we showed in Theorem 1 of [22] that the *wall following law*

$$\mathbf{u}(\mathbf{x}) = -k(\mathbf{x} - \mathbf{x}_p) \quad (5)$$

[4]Here $B(\mathbf{q}, t) := \{\mathbf{p} \in \mathcal{W} \,|\, ||\mathbf{p} - \mathbf{q}|| \leq t\}$, i.e. the ball of radius $t$ centered at $\mathbf{q}$.

[5]We use an idealized LIDAR model to derive the polar curve $\rho_{\mathbf{x}}(\theta)$, but in (3) we use the minimum observed LIDAR range.

provides an easy formula for wall following within specified bounds, even in the absence of obstacle curvature information. This allows for fast computation, which is critical in our legged robot setting. The reader is referred to [1, Equation (9)] for the choice of wall following direction and to [1, Theorem 3] for an extension to differential drive robots.

### C. Gait Layer

*1) Hybrid Dynamical System Structure:* The gait layer's primary function is to interpret simple unicycle commands $\mathbf{u}_{ku} = (v, \omega)$, as well as simple gripper commands by mapping them into physical joint level robot behaviors and transitions between them that realize the reactive layer's abstracted gripping/releasing unicycle model in the physical world. This structure naturally lends itself to the hybrid dynamical systems framework and we conjecture (but defer the proof to a future paper in progress) that the following architecture meets the requirements of a formal simple hybrid dynamical manipulation and self-manipulation system [5].

Let $\mathbf{x}_M \in \mathcal{X}_M$ be the robot pose and joint state and let $\mathbf{g} := (g_s, g_v, g_a) \in \{0,1\}^3$ be a vector representing gripper state, where $g_s \in \{0,1\}$ representing "open" and "closed" respectively, $g_v \in \{0,1\}$, representing zero and non-zero gripper transition velocity, and $g_a \in \{0,1\}$ representing zero and non-zero gripper command from the reactive layer, with $\mathbf{g}$ arranged as yet another component of the gait layer's state. Thus, taking $\mathbf{x}_{M^+} = (\mathbf{x}_M, \mathbf{g}) \in \mathcal{X}_M \times \{0,1\}^3$ as the hybrid state, our hybrid system modes arise from the disjoint union of the geometric placement indexed by the 4 mutually exclusive gripper conditions as follows:

$$\mathbf{x}_{M^+} \in \begin{cases} \mathcal{M}_W = \{\mathbf{x}_{M^+} \,|\, g_s = 0, g_v = 0\} & \textit{"Walk"} \\ \mathcal{M}_M = \{\mathbf{x}_{M^+} \,|\, g_s = 0, g_v = 1\} & \textit{"Mount"} \\ \mathcal{M}_P = \{\mathbf{x}_{M^+} \,|\, g_s = 1, g_v = 0\} & \textit{"Push Walk"} \\ \mathcal{M}_D = \{\mathbf{x}_{M^+} \,|\, g_s = 1, g_v = 1\} & \textit{"Dismount"} \end{cases}$$

where the guard condition, $g_a = 1$, triggers appropriate resets so that the hybrid mode system changes in the recurring sequence: $\mathcal{M}_W \to \mathcal{M}_M \to \mathcal{M}_P \to \mathcal{M}_D \to \mathcal{M}_W \ldots$

*2) Mode Dynamics:* A formal representation of the legged controllers and resulting closed loop dynamics used to realize the abstracted unicycle grip/release behaviors lies beyond the scope (and page constraints) of this paper. Instead, we now provide a brief, informal account of each mode as follows.

*a) Walk:* Incorporated here as reported in [20], this behavior is adapted from the still developing insights of [6]. While the kinematic model of the Minitaur platform prevents it from literal unicycle behavior in quasi-static operation, the underlying family of controllers overcomes this deficiency by dynamically exploiting higher-order effects, such as bending of the limbs and frame, as well as toe-slipping.

*b) Mount:* The mounting behavior, a physical realization of the abstract $(g_s, g_v) = (0,1)$ state, comprises a sequential composition that we conjecture can be placed within the formal framework of [23]. Informally, the behavior begins by leaping with the front legs, while maintaining ground contact with the rear, as shown in Fig. 3-2. During

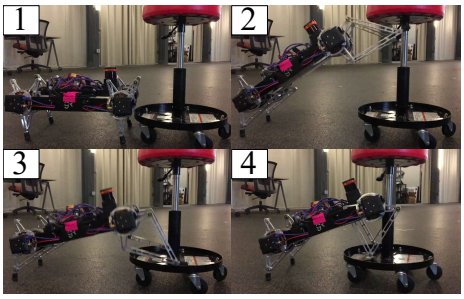


Fig. 3. Consecutive snapshots of a successful "Mount" onto an object.

this "flight" phase, an attempt to servo to the desired yaw is made by generating a difference in ground reaction forces in the stance legs according to the control law:

$$F_{LH} = -\frac{1}{2}(k_p(\psi_{des} - \psi) - k_d\dot{\psi})$$
$$F_{RH} = \frac{1}{2}k_p(\psi_{des} - \psi) - k_d\dot{\psi}$$

where $F_{LH}$ is the ground reaction force on the body generated by the left hip, and $F_{RH}$ is the analogous force generated by the right hip. Note that this method does not use Minitaur's kinematic configuration as a means of measuring $\psi$, and as such is able to continue to servo to the desired heading even in the presence of toe slipping or bending in the body. However, as contact modes are not assured and Vicon data is not available to the gait layer, the measurement of $\psi$ is obtained by integrating gyroscope data, which for this short behavior (less than a second) is reasonably accurate. In this work, we implicitly assume that the mounting behavior is always successful. Since failures might occur, we intend to relax this assumption in the future by introducing feedback in the hybrid mode system presented above.

*c) Push-Walk:* This behavior attempts to mask the underlying dynamics of the system consisting of the Minitaur platform with the front two limbs in various contact modes with a holonomic (albeit not friction-less) stool, and the rear two in varying contact modes with the ground. In [1], we introduced a method for generating virtual commands for different points of interest in the holonomic robot-object pair when a gripper is utilized, and translating them to actual commands for the differential drive robot using simple kinematic maps. The goal of this behavior is to exploit this result and use Minitaur's front legs as a virtual gripper.

The behavior is divided into two components; the fore-aft push-walk, and the yaw push-walk. The fore-aft push-walk is simply the previously described walking gait [6], modified such that the front limbs cannot retract to break contact with the stool. The yaw push-walk is a bit more dynamic, as the empirical application of the fore-aft walk in turning situations proved to have prohibitively small radius of curvature. To improve upon this, the front legs are allowed to retract as they would during walking, breaking and re-establishing contact with the stool on each step. The result is that the Minitaur is "freed" from the kinematic constraint of being unable to turn sharply enough in a manner described intuitively in Fig 4,

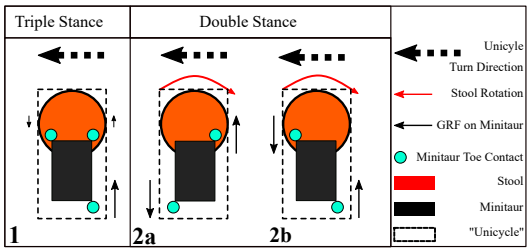


Fig. 4. Intuition underlying how intermittent contact (yaw push-walk) provides larger moments on the system than the moments produced in a triple stance (fore-aft push-walk). In (1), the presence of both toes on the stool kinematically constrains it so that any reaction forces generated by those toes are internal forces of the Minitaur-Stool system, where as in (2a) and (2b), the stool is free to rotate, allowing the single front toe to generate a moment on the Minitaur body.

avoiding triple stance (Fig. 4.1) in favor of the more strongly yawing torques arising from double stance (Fig. 4.2a,b).

*d) Dismount:* Finally, the $(g_s, g_v) = (1, 1)$ state is encoded by employing the walking behavior with controller parameters set as - 1) the height of the walk, or the nominal length of a stance leg is made nearly maximum, and 2) a simple open-loop fore-aft trajectory is programmed to linearly ramp up the speed to a pre-determined backward rate and then back down to zero.

## IV. EXTENSION OF REACTIVE LAYER TO NONCONVEX OBSTACLES

In this Section, we extend the result of Theorem 1 of [1] regarding the wall following law (5) to a class of nonconvex obstacles satisfying specific criteria. We begin with some notation and basic definitions for nonconvex obstacles.

**Definition 1** ([24]) Let $\mathcal{X}$ be a Hilbert space and $\mathcal{S}$ a closed set of $\mathcal{X}$. For $\mathbf{x} \in \mathcal{X}$ we denote by $\text{Proj}_{\mathcal{S}}(\mathbf{x})$ the (possibly empty) set of nearest points of $\mathbf{x}$ in $\mathcal{S}$. When $\text{Proj}_{\mathcal{S}}(\mathbf{x})$ is a singleton, its single point is called the *metric projection* and denoted by $\Pi_{\mathcal{S}}(\mathbf{x})$, i.e., $\text{Proj}_{\mathcal{S}}(\mathbf{x}) = \{\Pi_{\mathcal{S}}(\mathbf{x})\}$.

**Definition 2** ([24]) A vector $\mathbf{v} \in \mathcal{X}$ is said to be a *proximal normal vector* of $\mathcal{S}$ at $\mathbf{x} \in \mathcal{S}$ whenever there exists $t > 0$ such that $\mathbf{x} \in \text{Proj}_{\mathcal{S}}(\mathbf{x} + t\mathbf{v})$. The set of such vectors is the *proximal normal cone of* $\mathcal{S}$ *at* $\mathbf{x}$, denoted by $N^P(\mathcal{S}; \mathbf{x})$.

**Definition 3** ([24]) Given an extended real $r \in [0, +\infty]$ and a real $\alpha > 0$, we say that a closed set $\mathcal{S}$ of $\mathcal{X}$ is $(r, \alpha)$-*prox-regular at* $\mathbf{x}_0 \in \mathcal{S}$ if for every $\mathbf{x} \in \mathcal{S} \cap B(\mathbf{x}_0, \alpha)$ and every direction $\zeta \in N^P(\mathcal{S}; \mathbf{x}) \cap B(\mathbf{0}, 1)$, we have that $\mathbf{x} \in \text{Proj}_{\mathcal{S}}(\mathbf{x} + t\zeta)$ for every real $t \in [0, r]$.

We say that $\mathcal{S}$ is $r$-*prox-regular at* $\mathbf{x}_0 \in \mathcal{S}$ if it is $(r, \alpha)$-prox-regular at $\mathbf{x}_0$ for some $\alpha > 0$ and we simply say that $\mathcal{S}$ is *prox-regular at* $\mathbf{x}_0$ if there exists $r \in [0, +\infty]$ such that $\mathcal{S}$ is $r$-prox-regular at $\mathbf{x}_0$. Finally, $\mathcal{S}$ is *prox-regular* (resp. $r$-*prox-regular*) if it is prox-regular (resp. $r$-prox-regular) at every point $\mathbf{x} \in \mathcal{S}$. It is known [24] that $\mathcal{S}$ is prox-regular if and only if there exists a continuous function $\rho : \mathcal{S} \to [0, \infty]$, called the *prox-regularity function*, such that for every $\mathbf{x} \in \mathcal{S}$ and every $\zeta \in N^P(\mathcal{S}; \mathbf{x}) \cap B(\mathbf{0}, 1)$ one has $\mathbf{x} \in \text{Proj}_{\mathcal{S}}(\mathbf{x} + t\zeta)$ for every real $t \in [0, \rho(\mathbf{x})]$. The definition of prox-regularity itself is relatively abstract, but we attempt to ground it in the following paragraphs and Fig. 5.

It is also useful to include the definitions of the following enlargements of the set $\mathcal{S}$, according to [24]

$$\mathcal{R}_{\mathcal{S}}(\mathbf{x}_0, r, \alpha) := \{\mathbf{x} + t\mathbf{v} \,:\, \mathbf{x} \in \mathcal{S} \cap B(\mathbf{x}_0, \alpha), t \in [0, r], \mathbf{v} \in N^P(\mathcal{S}; \mathbf{x}) \cap B(\mathbf{0}, 1)\}$$

$$U_{\rho(\cdot)}(\mathcal{S}) := \{\mathbf{x} \in \mathcal{X} \,:\, \exists \mathbf{y} \in \text{Proj}_{\mathcal{S}}(\mathbf{x}) \text{ with } d_{\mathcal{S}}(\mathbf{x}) < \rho(\mathbf{y})\}$$

With these definitions, we are led to the following lemma.

**Lemma 1** *If $\mathcal{S}$ is $\rho(\cdot)$-prox-regular, then the collection of sets $\{\mathcal{R}_{\mathcal{S}}(\mathbf{x}, \rho(\mathbf{x}), \alpha) \,:\, \mathbf{x} \in \mathcal{S}\}$ with $\alpha > 0$ corresponding to the prox-regularity condition forms an open cover of $U_{\rho(\cdot)}(\mathcal{S})$.*

*Proof.* It is shown in [24] that the extended local sets $\mathcal{R}_{\mathcal{S}}(\mathbf{x}_0, r, \alpha)$ are open. Now consider a point $\mathbf{u} \in U_{\rho(\cdot)}(\mathcal{S})$. Then, by definition, there exists a $\mathbf{y} \in \mathcal{S}$, a direction $\zeta \in N^P(\mathcal{S}; \mathbf{y}) \cap B(\mathbf{0}, 1)$ and a real $t \in [0, \rho(\mathbf{y})]$ such that $\mathbf{u} = \mathbf{y} + t\zeta$. This shows that $\mathbf{u}$ also belongs in the set $\mathcal{R}_{\mathcal{S}}(\mathbf{y}, \rho(\mathbf{y}), \alpha)$ for some $\alpha > 0$, and concludes the proof. ■

In [24, Theorem 2.3], it is also shown that if $\mathcal{S}$ is $(r, \alpha)$-prox-regular at a point $\mathbf{x}_0$, then $\Pi_{\mathcal{S}}$ is well-defined and locally Lipschitz continuous on the set $\mathcal{R}_{\mathcal{S}}(\mathbf{x}_0, r, \alpha)$. Hence, using Lemma 1, we arrive at the following result.

**Lemma 2** *If $\mathcal{S}$ is $\rho(\cdot)$-prox-regular, then $\Pi_{\mathcal{S}}$ is well-defined and locally Lipschitz continuous on $U_{\rho(\cdot)}(\mathcal{S})$.*

In this way, we can formulate the following theorem, that extends the guarantees of our wall-following control law to $\rho(\cdot)$-prox-regular, nonconvex obstacles.

**Theorem 1** *In the presence of $\rho(\cdot)$-prox-regular, convex*[6] *or nonconvex isolated obstacles $\mathcal{O} := (O_1, O_2, \ldots)$ in the workspace satisfying Assumption 1 with $\min \rho_{O_i} > r + \epsilon$ for each $O_i$ and $\epsilon$ chosen as in Equation (11) of [22], the wall following law* (5) *has no stationary points, leaves the robot's free space $\mathcal{F}$ positively invariant under its unique continuously differentiable flow, and steers the robot along the boundary of a unique obstacle in $\mathcal{O}$ in a clockwise or counterclockwise fashion (according to the selection of a) with a nonzero rate of progress, while maintaining a distance of at most $(r + \epsilon)$ and no less than $\left(r + \frac{\epsilon}{2}\right)$ from it.*

*Proof.* The proof is almost identical to the proof of [22, Theorem 1]. The only difference here is that the wall following law is not piecewise continuously differentiable but just locally Lipschitz. This, however, does not change its basic properties. The key in the proof is the requirement that $\min \rho_{O_i} > r + \epsilon$ for each $O_i$. Since it is shown in [22, Proposition 1] that $\left\{\mathbf{p} \in \mathcal{W} \,\middle|\, \frac{\epsilon}{2} < d(\mathbf{p}, \partial\mathcal{F}) < \epsilon\right\}$ is positively invariant under the flow of the wall following law, we are guaranteed that the robot will never exit $U_{\rho(\cdot)}(\mathcal{S})$, which ensures local Lipschitz continuity of the vector field. ■

We then include the following definition to provide some intuition on the abstract definition of prox-regularity.

[6]Convex bodies are $\rho(\cdot)$-prox-regular by construction [24].

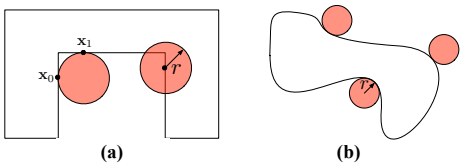


Fig. 5. (a) An example of a nonconvex body that fails to be $r$-prox-regular; since $0 < ||\mathbf{x}_1 - \mathbf{x}_0|| < 2r$, the existence of a tangent closed ball of radius $r$ to both $\mathbf{x}_0$ and $\mathbf{x}_1$ violates the $r$-oval-segment criterion, (b) An example of an $r$-prox-regular nonconvex body in $\mathbb{R}^2$, satisfying Proposition 1.

**Definition 4** ([18]) For any $r > 0$ and $\mathbf{x}_0, \mathbf{x}_1 \in \mathcal{X}$ with $||\mathbf{x}_1 - \mathbf{x}_0|| < 2r$, the *$r$-oval segment* $\Delta_r(\mathbf{x}_0, \mathbf{x}_1)$ in $\mathcal{X}$ with endpoints $\mathbf{x}_0, \mathbf{x}_1$ is defined as the intersection of all closed balls with radius $r$ containing $\mathbf{x}_0, \mathbf{x}_1$.

Then it can be shown that a closed set $\mathcal{S}$ of $\mathcal{X}$ is $r$-prox regular if and only if for any pair of points $\mathbf{x}_0, \mathbf{x}_1 \in \mathcal{S}$ with $0 < ||\mathbf{x}_1 - \mathbf{x}_0|| < 2r$, the $r$-oval segment $\Delta_r(\mathbf{x}_0, \mathbf{x}_1)$ contains a point of $\mathcal{S}$ different from $\mathbf{x}_0, \mathbf{x}_1$, or equivalently $\mathcal{S} \cap \Delta_r(\mathbf{x}_0, \mathbf{x}_1) \neq \{\mathbf{x}_0, \mathbf{x}_1\}$. Prox-regularity can, therefore, be seen as a means of defining an appropriate "length-scale" for the "nonconvexities" (e.g. valleys or traps) of the obstacle that do not result in a controller failure. Fig. 5 provides one example of nonconvex body for which this prox-regularity criterion fails and one example for which it succeeds. As a guide, we provide the following sufficient condition, based on curvature, for prox-regularity in $\mathbb{R}^2$ without proof.

**Proposition 1** *A closed, compact, simply-connected body $\mathcal{S} \subset \mathbb{R}^2$ is $r$-prox-regular if any tangent closed ball of radius $r$ at its boundary $\partial\mathcal{S}$ has only one common point with $\mathcal{S}$.*

In the future, we would like to use the formal guarantees of Theorem 1, whose assumptions are mere sufficient conditions, to extend the application of doubly-reactive planners [16] to nonconvex obstacles in Hilbert spaces.

## V. EXPERIMENTAL RESULTS

In this Section, we begin with a brief description of the hardware and software setup and continue with a description of the experiments run and our empirical results.

### A. Setup

*1) ROS Infrastructure:* For the hardware and software experimental setup, we use a system structure similar to that presented in [20]. A custom ROS node on the Raspberry Pi receives $\mathbf{u}_{ku}$ and the desired mode of operation ("Walk", "Mount", "Push-Walk", "Dismount") as ROS messages from the desktop computer and forwards them to the Minitaur mainboard (microcontroller implementing the gait layer functionalities) 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 subscribes to the $\mathbf{u}_{ku}$ ROS topic (using the `Twist` message type), as well as an additional one capable of defining the desired behavior. A final ROS node running on the Raspberry Pi, taken from [25], forwards LIDAR measurements (using the `LaserScan` message type) to the desktop computer.

The pose information, consisting of the horizontal plane coordinates of the robot and all the objects and the orientation of the robot, is extracted from a Vicon Motion Capture System [19] at 100 Hz, using a set of motion capture cameras positioned around a 20m $\times$ 6m arena. The desktop computer receives the online data from Vicon using the ROS package `mocap_vicon` [26] and forwards it to the desktop computer running ROS. The reactive layer runs at approximately 30Hz, which is more than enough for the robot to recover if any obstacle is detected, and the gait controller runs at 1KHz.

*2) LIDAR Measurement Handling:* The LIDAR measurements are pre-processed by the desktop computer before being used by the reactive planner. First of all, following the requirements of [16], range measurements greater than the limit $R$ are set to $R$. All measurements are projected on the horizontal plane using the robot pitch angle measurement provided by the motion capture system. Finally, when the reactive layer executes a symbolic action MOVETOOBJECT$(i, \mathcal{P})$ or POSITIONOBJECT$(i, \mathcal{P})$, it is critical to recognize the points of the LIDAR pointcloud associated with the object $i$ and not use them for the calculation of the local freespace [1], since $i$ should not be an obstacle. Hence, we look for points of the LIDAR pointcloud that are "close-enough" (within a $\delta_{object}$ tolerance) of the object $i$ position and set the associated ranges to infinity. Unfortunately, this results in the object blocking the robot's line of sight during POSITIONOBJECT$(i, \mathcal{P})$, meaning part of the workspace (that may or may not contain an obstacle) is completely invisible to the robot. However, as shown in the following experimental datasets and in the video, this was not an important issue that prohibited experimental success.

*3) Experimental Parameters:* For the experiments reported in this paper, we use a wall following offset $\epsilon = 65$cm, an object detection threshold $\delta_{object} = 60$cm, an angular precision of $12^\circ$ for successful alignment with each of the objects, a linear gain $k_l = 0.8$, an angular gain $k_a = 0.01$ and a maximum allowable LIDAR range of $R = 3$m. The stool-objects and the robot are treated as disks of radius $r = r_i = 0.2$m and we discretize the paths provided by the deliberative layer with a resolution of 1cm. Finally, the $\delta$ values (precision tolerances for landing zones [22]) used for the MOVETOOBJECT, POSITIONOBJECT and MOVE symbolic actions are 20cm, 40cm and 45cm respectively.

### B. Task #1 - Single Object Positioning

In Fig. 6, we document the ability of the reactive layer's abstract unicycle control outputs (Section III-B) to drive the gait layer's hybrid self-manipulation dynamics (Section III-C) to follow the paths and manipulation directives given by the deliberative layer (Section III-A). Starting from an initial position, the robot has to move to an object, mount it, push it to a desired location, dismount from it and then move to a predefined location. In order to validate the performance of the wall following law, presented in Section IV, a similar experiment is repeated, with the robot having to

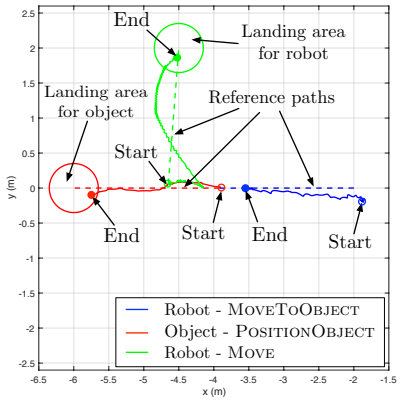


Fig. 6. Task #1 - No Obstacles (Section V-B): Vicon data showing the robot successfully following paths provided by the deliberative layer (dotted line segments): the robot has to approach (and then mount) the object (action MoveToObject), push the object inside a desired landing area (action PositionObject) and (first dismount) then retire to move to a predefined position (action Move), while following the reference paths (dotted lines).

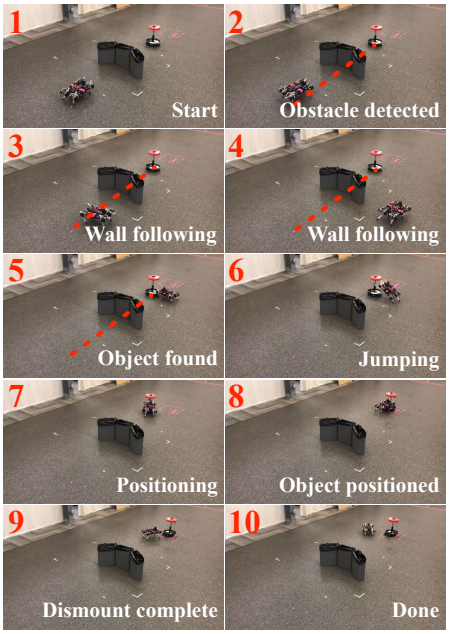


Fig. 7. Task #1 - Unanticipated Obstacle (Section V-B): The reactive layer allows for successful task completions even in the presence of nonconvex obstacles, that have not been accounted for by the deliberative layer. The red dashed line represents the original (blocked by the obstacle) path given by the deliberative planner, associated with the action MoveToObject.

avoid a nonconvex obstacle blocking its path to the object. As shown in Fig. 7 and in the accompanying video, the task is successfully completed, using the wall following algorithm.

### C. Task #2 - Swapping Object Positions

The second task is more demanding for the deliberative planner, since the robot has to successfully swap the positions of two objects and then move to a "nest" location. As expected, the deliberative planner outputs a plan which includes an intermediate position for one of the objects. Using the reactive layer, the robot completes this task, as shown in Fig. 8 and in the video. Notice how the robot

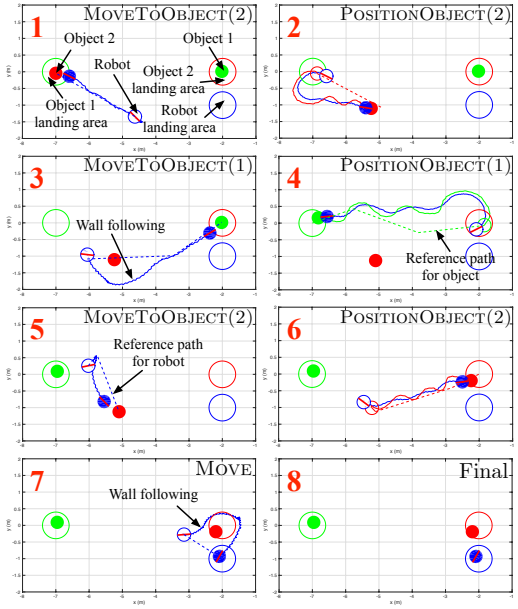


Fig. 8. Task #2 (Section V-C): Vicon data showing Minitaur swapping the positions of two objects. The dashed lines represent the reference paths for the robot or for the objects, provided by the deliberative layer. Non-filled and filled circles depict the start and end positions for each action execution. Any discrepancies of the final trajectories with the reference paths are caused by the controller's reactive nature and do not affect task completion.

switches to wall following when necessary and avoids any obstacles that block its path. The gait layer successfully executes the commands provided by the reactive layer.

### D. Task #3 - Object Blocking the Position of Another Object

Finally, in the third set of experiments, we explore a similar task where the robot has to move an object in a location occupied by another object. We demonstrate several successful trials in the accompanying video, but here we focus on a special case where the online execution is incommoded by the presence of an obstacle and terrain irregularities, shown in Fig. 9. The robot also has to face other unfortunate events, such as network delays and getting the wheels of the stool stuck in the platform's step, but eventually completes the task. This illustrates the role of the reactive layer whose "persistence" can handle changes in the environment not predicted beforehand. It also highlights the value of legged over wheeled locomotion when "mobipulation" in unstructured environments with rough terrain is needed. We hope to report more on that in the future.

## VI. CONCLUSION AND FUTURE WORK

The proxy scenarios of rearrangement tasks with stools documented in this paper verify that our three-layer control architecture is capable of completing challenging tasks for a dynamic platform in an effective way. Future work will address the problem of closely integrating the reactive and deliberative planners, while maintaining provable properties. Also, we hope to analyze the dynamic behaviors employed here in earnest, make them more robust, and develop new behaviors to improve the effectiveness of the gait layer.

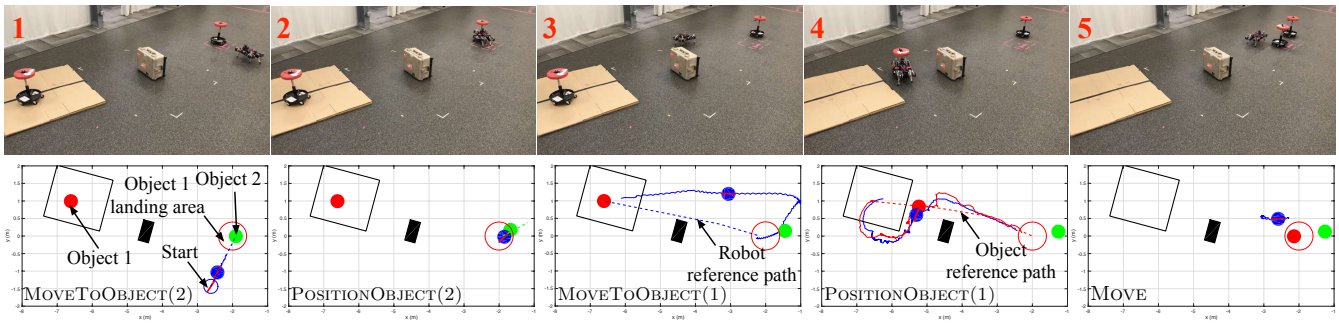


Fig. 9. Task #3 (Section V-D): Consecutive snapshots from a successful completion of a task where the robot must move an object that blocks the desired location of another object, highlighting the robustness of the approach. Apart from the presence of a convex obstacle (depicted in black) and terrain irregularities in the form of a 4cm-tall platform (depicted by a solid black line), the robot loses track of its pose estimation due to unfortunate network delays while executing MoveToObject(1). However, with the successful coordination of the reactive and the gait layer, it manages to find the reference path again once it reconnects. Also, as shown in the accompanying video (and discernible from the relatively large oscillations of the robot's path in frame 4), although the wheels of the stool get caught by the platform during PositionObject(1), the persistence of the reactive layer allows for successful task completion while avoiding unexpected obstacles.

## VII. 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] V. Vasilopoulos, W. Vega-Brown, O. Arslan, N. Roy, and D. E. Koditschek, "Sensor-Based Reactive Symbolic Planning in Partially Known Environments," in *IEEE International Conference on Robotics and Automation*, May 2018, pp. 5683–5690.

[2] J. E. Hopcroft, J. T. Schwartz, and M. Sharir, "On the Complexity of Motion Planning for Multiple Independent Objects; PSPACE-Hardness of the "Warehouseman's Problem"," *The International Journal of Robotics Research*, vol. 3, no. 4, p. 7688, 1984.

[3] "Ghost Robotics Minitaur," 2016. [Online]. Available: http://www.ghostrobotics.io/minitaur/

[4] M. T. Mason, D. Pai, D. Rus, L. R. Taylor, and M. Erdmann, "A Mobile Manipulator," in *IEEE International Conference on Robotics and Automation*, vol. 3, May 1999, pp. 2322 – 2327.

[5] A. M. Johnson, S. A. Burden, and D. E. Koditschek, "A hybrid systems model for simple manipulation and self-manipulation systems," *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1354–1392, 2016.

[6] A. De and D. E. Koditschek, "Event-driven Reactive Dynamical Quadrupedal Walking," *In Prep*.

[7] J. Wolfe, B. Marthi, and S. Russell, "Combined Task and Motion Planning for Mobile Manipulation," in *Proceedings of the International Conference on Automated Planning and Scheduling*, 2010.

[8] L. P. Kaelbling and T. Lozano-Perez, "Hierarchical task and motion planning in the now," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 1470–1477.

[9] G. Konidaris, L. P. Kaebling, and T. Lozano-Perez, "From Skills to Symbols: Learning Symbolic Representations for Abstract High-Level Planning," *Journal of Artificial Intelligence Research*, vol. 61, pp. 215–289, 2018.

[10] C. R. Garrett, T. Lozano-Perez, and L. P. Kaebling, "FFRob: Leveraging symbolic planning for efficient task and motion planning," *The International Journal of Robotics Research*, vol. 37, no. 1, pp. 104–136, 2018.

[11] W. Vega-Brown and N. Roy, "Admissible Abstractions for Near-optimal Task and Motion Planning," in *27th International Joint Conference on Artificial Intelligence*, July 2018.

[12] B. Marthi, S. Russell, and J. Wolfe, "Angelic hierarchical planning: Optimal and online algorithms," in *International Conference on Automated Planning and Scheduling*, 2008.

[13] R. A. Knepper, S. Srinivasa, and M. T. Mason, "Hierarchical Planning Architectures for Mobile Manipulation Tasks in Indoor Environments," in *IEEE International Conference on Robotics and Automation*, May 2010, pp. 1985–1990.

[14] J. Scholz, N. Jindal, M. Levihn, C. L. Isbell, and H. I. Christensen, "Navigation Among Movable Obstacles with learned dynamic constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2016, pp. 3706–3713.

[15] T. T. Topping, G. Kenneally, and D. E. Koditschek, "Quasi-static and dynamic mismatch for door opening and stair climbing with a legged robot," in *IEEE International Conference on Robotics and Automation*, May 2017, pp. 1080–1087.

[16] O. Arslan and D. E. Koditschek, "Sensor-Based Reactive Navigation in Unknown Convex Sphere Worlds," in *The 12th International Workshop on the Algorithmic Foundations of Robotics*, 2016.

[17] ——, "Smooth Extensions of Feedback Motion Planners via Reference Governors," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 4414–4421.

[18] R. A. Poliquin, R. T. Rockafellar, and L. Thibault, "Local Differentiability of Distance Functions," *Transactions of the American Mathematical Society*, vol. 352, no. 11, pp. 5231–5249, 2000.

[19] Vicon. Vantage V16. [Online]. Available: https://www.vicon.com/file/vicon/vantagebrochure-021216-web-94650.pdf

[20] V. Vasilopoulos, O. Arslan, A. De, and D. E. Koditschek, "Sensor-Based Legged Robot Homing Using Range-Only Target Localization," in *IEEE International Conference on Robotics and Biomimetics*, 2017, pp. 2630–2637.

[21] W. Vega-Brown and N. Roy, "Asymptotically optimal planning under piecewise-analytic constraints," in *The 12th International Workshop on the Algorithmic Foundations of Robotics*, 2016.

[22] V. Vasilopoulos, W. Vega-Brown, O. Arslan, N. Roy, and D. E. Koditschek, "Technical Report: Sensor-Based Reactive Symbolic Planning in Partially Known Environments," *Technical Report*, September 2017. [Online]. Available: https://arxiv.org/abs/1709.05474

[23] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential Composition of Dynamically Dexterous Robot Behaviors," *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 534–555, Jun 1999.

[24] R. Correa, D. Salas, and L. Thibault, "Smoothness of the metric projection onto nonconvex bodies in Hilbert spaces," *Journal of Mathematical Analysis and Applications*, vol. 457, no. 2, pp. 1307–1332, 2018.

[25] [Online]. Available: http://wiki.ros.org/urg_node

[26] KumarRobotics. Drivers for motion capture systems. [Online]. Available: https://github.com/KumarRobotics/motion_capture_system