8-2021

Motivation dynamics for autonomous composition of navigation tasks

Paul B. Reverdy

Vasileios Vasilopoulos
University of Pennsylvania

Daniel E. Koditschek
University of Pennsylvania

Recommended Citation

Paul B. Reverdy, Vasileios Vasilopoulos, and Daniel E. Koditschek, "Motivation dynamics for autonomous composition of navigation tasks", *IEEE Transactions on Robotics (T-RO)* 37(4). August 2021.

Motivation dynamics for autonomous composition of navigation tasks

Abstract

We physically demonstrate a reactive sensorimotor architecture for mobile robots whose behaviors are generated by motivation dynamics. Motivation dynamics uses a continuous dynamical system to reactively compose low-level control vector fields using valuation functions which capture the potentially competing influences of external stimuli relative to the system's own internal state. We show that motivation dynamics 1) naturally accommodates external stimuli through standard signal processing tools, and 2) can effectively encode a repetitive higher-level task by composing several low-level controllers to achieve a limit cycle in which the robot repeatedly navigates towards two alternatively valuable goal locations in a commensurately alternating order. We show that these behaviors are robust to perturbations including imperfect models of robot kinematics, sensor noise, and disturbances resulting from the need to traverse difficult terrain. We argue that motivation dynamics can provide a useful alternative to controllers based on hybrid automata in situations where the control operates at a low level close to the physical hardware.

Keywords

Reactive and Sensor-Based Planning; Motion and Path Planning; Planning, Scheduling and Coordination; Hybrid Logical/Dynamical Planning and Verification

Disciplines

Electrical and Computer Engineering | Engineering | Systems Engineering

Motivation dynamics for autonomous composition of navigation tasks

Paul B. Reverdy, Vasileios Vasilopoulos, and Daniel E. Koditschek

***Abstract*—We physically demonstrate a reactive sensorimotor architecture for mobile robots whose behaviors are generated by motivation dynamics. Motivation dynamics uses a continuous dynamical system to reactively compose low-level control vector fields using valuation functions which capture the potentially competing influences of external stimuli relative to the system's own internal state. We show that motivation dynamics 1) naturally accommodates external stimuli through standard signal processing tools, and 2) can effectively encode a repetitive higher-level task by composing several low-level controllers to achieve a limit cycle in which the robot repeatedly navigates towards two alternatively valuable goal locations in a commensurately alternating order. We show that these behaviors are robust to perturbations including imperfect models of robot kinematics, sensor noise, and disturbances resulting from the need to traverse difficult terrain. We argue that motivation dynamics can provide a useful alternative to controllers based on hybrid automata in situations where the control operates at a low level close to the physical hardware.**

I. Introduction

This paper considers the problem of managing the execution of competing tasks in the setting of robot navigation. Specifically, we demonstrate the applicability and robustness of a dynamical systems approach to task management. An established approach to this problem uses a finite automaton to switch among various control vector fields in a discrete manner, yielding a hybrid control system. Constructing such an automaton requires designing a representation of the robot and its environment in terms of discrete symbols and mapping these symbols to low-level control actions. The discretization-based approach can fail for at least three reasons: first, the discrete symbols may be difficult to define (e.g., it may not be clear where to set thresholds that define symbol boundaries); second, the discretization is inherently rigid and dependent on a model of the robot and its environment, which will inevitably be incomplete; and third, it may be difficult to determine the discrete state of the system, particularly in situations where sensor data is corrupted by significant noise.

In contrast, our approach maintains a continuous internal representation of the control vector field selection process. The continuous internal representation avoids the need to explicitly define a discrete symbolic state and results in a system that can be considered a continuous relaxation of the hybrid system, which can encode a much richer set of control actions. The continuous nature of our approach naturally accommodates situations where the state of an equivalent discrete representation would be difficult to determine.

Our approach is inspired by biological decision-making mechanisms and admits formal analysis and performance guarantees. We aim to encode and control internal state variables that drive the robot to execute one or another of its tasks in a manner sensitive to both their internally-perceived value and externally-stimulated urgency. In other words, we seek to develop a reactive method that is capable of aspects of planning and of responding to external stimuli.

The focus of this paper is thus two-fold. First, we present our recently-developed dynamical systems framework, which we term *motivation dynamics* [1], in the context of concrete problems in mobile robotics. We argue that this framework can be a useful alternative to hybrid control systems in situations where the control operates at a low level close to the physical hardware. Second, we show two examples of how a motivation dynamics controller can be instantiated to produce desired behavior: one a stimulus response where the robot chooses among navigation behaviors in response to a noisy environmental stimulus, and another a stable recurrent behavior corresponding to persistent patrol of two locations.

A. Motivation

In this work, we demonstrate the effectiveness of our recently-developed motivation dynamics framework [1] in achieving reactive and recurrent patrol behaviors. The motivation dynamics framework is inspired by work in the neural and ecological sciences that aims to understand how organisms make decisions. A key feature of the decisions such organisms have evolved to make is that they are *embodied*, i.e., they take place in a physical context and decisions are made in order to take physical actions. The dynamics associated with taking physical actions then place important constraints on the types of actions that can be considered.

There is a large and growing body of evidence that for many types of embodied decisions, humans and other living organisms perform decision making by considering many possible actions in parallel [2]; furthermore there is evidence [3] that in certain contexts the decision to select one of the possible actions is made in a continuous fashion, where the human subject's physical dynamics smoothly transition from performing one action to another. A common hypothesis in the literature suggests that such a parallel operation mechanism improves performance by allowing agents to anticipate required actions and change their action choice if new sensory data arrives [4], [5], [6].

Inspired by these findings, we take a reactive control approach and encode individual robot actions in control vector fields $F_i, i \in \{1, \dots, N\}$. A standard method to compose

P. Reverdy is with the Department of Aerospace and Mechanical Engineering, University of Arizona, Tucson, AZ 85721 USA, and with Amazon, Seattle, WA 98109 USA. This work was performed before P. Reverdy joined Amazon. V. Vasilopoulos is with the Department of Mechanical Engineering and Applied Mechanics, and D. E. Koditschek is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA. Email: `preverdy@arizona.edu`.

these actions would be to select one action at a time using a discrete automaton, thereby producing a hybrid control system. In contrast, our method composes actions by taking convex combinations of the corresponding vector fields and varying the combination coefficients as needed to achieve high-level behaviors. Concretely, we control the robot state x with a composite vector field

$$F(x,t) = \sum_{i=1}^{N} m_i(t) F_i(x),$$

where each $m_i \in [0,1]$ and $m \in \Delta^N$, where Δ^N is the N-simplex defined by

$$\Delta^N = \left\{ x \in \mathbb{R}^{N+1} : x_i \geq 0, \sum_{i=1}^{N+1} x_i = 1 \right\}.$$

We call m the *motivation state* of the agent, and the component $m_i(t), i \in \{1,\ldots,N\}$ represents the fraction of attention which the agent places on task i at time t; component m_{N+1} represents the motivation not to perform any task. Encoding the action selection in this way can be thought of as a convex relaxation of a discrete selection process; in the discrete process the agent's discrete actions would correspond to the vertices of the simplex. Using a continuous motivation state m allows a system to encode a much richer set of behaviors than one could using a discrete switching variable. As an example, consider a scenario where the robot knows it must perform one of two tasks, but is currently unsure which one is correct. The robot can anticipate the need to perform both tasks by setting both motivations to nonzero values and then smoothly shifting weight to the appropriate motivation as the identity of the correct task becomes apparent.

In cases where the system being controlled is fundamentally discrete, e.g., where the control actions correspond to selecting one of a finite number of gear ratios, motivation dynamics would not be appropriate, because a convex combination of these control actions is not meaningful. However, motivation dynamics is applicable to the broad range of systems where hybrid compositional control approaches have been used, because the underlying continuous controllers are expressed in terms of vector fields which can be meaningfully combined using convex combinations.

In the context of a recurrent patrol mission, we wish the robot to exhibit a default behavior consisting of cyclic performance of a sequence of equally-valued tasks, i.e., visits to a sequence of waypoints. When external stimuli, e.g., sensor data indicating the presence of an anomaly, appear, we wish the robot to interrupt its default behavior and engage in some other task whose value it now perceives to be higher than that of the default behavior. We show how this can be achieved by extending the framework from [1] to incorporate an interface from sensor data to the system's value state.

The control system that results from this extension of [1] is capable of smoothly shifting its motivation among its default tasks in nominal conditions and to other tasks as external sensor data dictate. We implement and execute these robust, reactive motivation dynamics on physical machines and we report on indoor and outdoor experiments with both wheeled and legged robots.

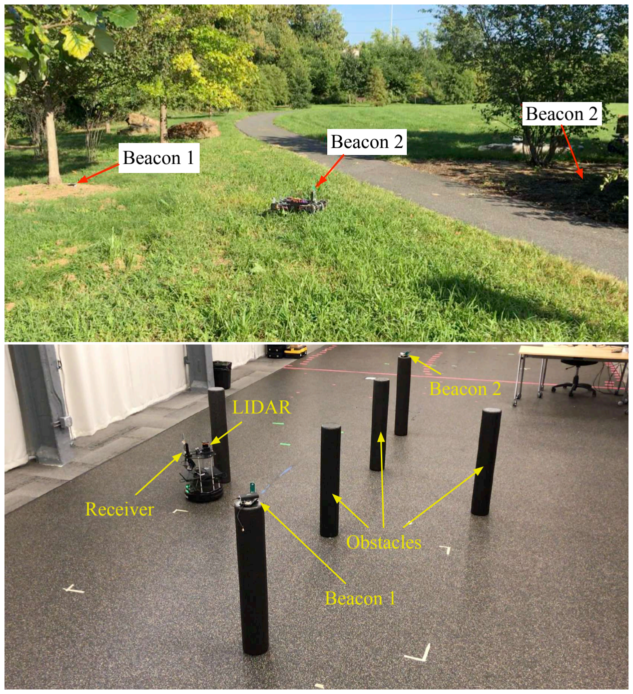


Fig. 1: Physical setups with Minitaur (top) in an outdoor experiment and Turtlebot (bottom) in an indoor experiment, equipped with a LIDAR for obstacle avoidance.

B. Prior work

In a recent theoretical work [1], we have developed a dynamical system that uses a motivation state $m(t)$ to encode its decision of which navigation task to perform at time t. Our dynamical system associates a positive value v_i to each task i and endows m with dynamics (3) studied in [7]. The state $v_i(t)$ encodes the degree to which the agent values task i at time t, and the dynamics of m are such that the system will tend to place the majority of its attention on the highest-value task. It can be shown formally [8], [9] that the dynamics (3) embed an unfolded pitchfork bifurcation with the task values v_i serving as unfolding parameters. The value state also provides a natural interface for injecting external stimuli using standard signal processing tools such as likelihood functions. The components of $m(t)$ are then used as weights for the individual task vector fields F_i and the system navigates using the weighted sum of the tasks.

As reported in [1], the motivation-based control system exhibits a limit cycle under appropriate conditions. The more relevant result for applications is given by Theorem 1 below, which states that there is a single control parameter to tune, and that as long as the value of this parameter is sufficiently high, the dynamics exhibit a stable limit cycle in which the robot visits its goal states in a cyclically-repeating order.

This work is part of a project that aims to develop a continuous dynamical systems approach for constructing switching protocols for robotic mission planning. It should be viewed as bridging the literatures on hybrid switching [10], [11], [12], [13], [14] and vector field or "blending"-based control [15], [16], [17] and is philosophically in the tradition of vector field planning, e.g., [18]. We argue that such an approach could facilitate the development of robustly autonomous robots capable of flexibly prioritizing their various low-level tasks in reaction to the environment, in the same way as vector field planning has proven effective in low-level motion control [19].

A key design problem for implementing a controller based on either the standard hybrid or our motivation dynamics framework is synthesizing a switching protocol that guarantees the controlled system will achieve the desired behavior. For the standard hybrid control approach, there has been significant work developing synthesis techniques that take behavioral specifications expressed in terms of linear-temporal logic and automatically construct hybrid controllers that are guaranteed to achieve behavior that satisfies the specification [20], [21]. Developing analogous general synthesis tools for motivation dynamics is an open problem. The previous theoretical result [1] is a first step towards solving this problem, as it provides a guarantee of correct behavior for motivation dynamics in the context of a two-point persistent patrol.

C. Contributions

The main contribution of this paper is the first physical implementation of the motivation dynamics system developed and first studied in [1]. First, we show how to integrate external stimuli by properly defining the value state of the motivation system, develop Theorem 2 to bound the performance of the resulting system, and demonstrate that the resulting robot control system is able to accurately and quickly respond to noisy stimuli. Second, we consider the limit cycle guaranteed by Theorem 1 and demonstrate its existence in several different robotic systems, both indoors and outdoors, and with and without obstacles. As mentioned above, this behavior would be referred to as a recurrent patrol or coverage mission in the logic-based synthesis literature.

The empirical results go beyond the previously-reported theoretical results from [1] in several ways. First, the integration of external stimuli, mentioned only in passing in [1], is now instantiated in a working physical implementation. Second, the formal results from [1], including Theorem 1, only address the case where the robot's workspace is obstacle free. Here, we show empirically that the same controller can produce a limit cycle even in the presence of obstacles.

The remainder of the paper is structured as follows. Section II presents the motivation dynamics system. Section III summarizes the formal results from [1] that characterize the limit cycle behavior. Section IV shows how the control strategy can be modified to integrate external stimuli. Section V describes the physical implementation of the controller on mobile robots, Section VI presents our empirical results, and Section VII concludes with our main observations.

II. Motivation dynamics

In this section, we specify the parts of our motivation dynamics control system. A block diagram of the system is shown as Figure 2. In Section III we show that particular choices of $f_{\mathbf{v}}, f_{\mathbf{m}}$, and F result in a controller that produces a recurrent patrol behavior. We address the integration of external stimuli (i.e., the input $\mathbf{u}$) in Section IV. For the remainder of the paper we assume that we have $N = 2$ tasks. The following sections follow the blocks of Figure 2 from bottom to top.

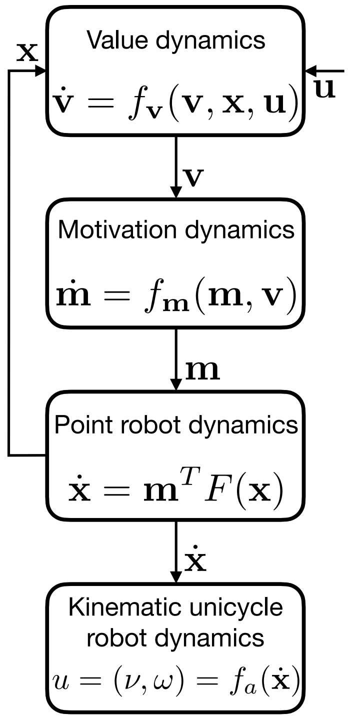

Value dynamics
$\dot{\mathbf{v}} = f_{\mathbf{v}}(\mathbf{v}, \mathbf{x}, \mathbf{u})$
$\mathbf{x}$
$\mathbf{u}$
$\mathbf{v}$
Motivation dynamics
$\dot{\mathbf{m}} = f_{\mathbf{m}}(\mathbf{m}, \mathbf{v})$
$\mathbf{m}$
Point robot dynamics
$\dot{\mathbf{x}} = \mathbf{m}^T F(\mathbf{x})$
$\dot{\mathbf{x}}$
Kinematic unicycle robot dynamics
$u = (\nu, \omega) = f_a(\dot{\mathbf{x}})$

Fig. 2: Block diagram of the motivation dynamics system. The unicycle inputs u are computed from the desired point robot dynamics $\dot{\mathbf{x}}$ via an anchoring relationship f_a; $\mathbf{u}$ represents external sensor input. The functions defining $\dot{\mathbf{m}}, \dot{\mathbf{v}}$, and $\dot{\mathbf{x}}$ are given by Equations (3), (4), and (2), respectively.

A. Kinematic unicycle model

We consider first a vehicle with fully-actuated first-order dynamics: $\dot{x} = u$, where $x \in \mathcal{D} \subseteq \mathbb{R}^2, u \in \mathbb{R}^2$. We develop our vector field planner for the fully-actuated case and then, for implementation purposes, map the planner into a version that works with a vehicle modeled as a kinematic unicycle, i.e., with dynamics

$$\begin{aligned} \dot{x}_1 &= \nu \cos\theta \\ \dot{x}_2 &= \nu \sin\theta \\ \dot{\theta} &= \omega, \end{aligned} \tag{1}$$

where $u = (\nu, \omega) \in \mathbb{R}^2$ are the control inputs given in terms of fore-aft speed and turning rate, respectively. This component is depicted in the bottom panel of Figure 2.

B. Point robot dynamics

The point robot dynamics consist of a set of behavior vector fields composed into a single navigation vector field by convex combination with weights defined by the motivation state m.

1) Behavior vector fields: We consider navigation as a prototypical task for our robots. In the spirit of the reactive control formalism, we encode individual navigation tasks in vector fields $F_i : \mathcal{D} \to T\mathcal{D}, i \in \{1,2\}$ with the property that

$$\dot{x} = F_i(x) \implies \lim_{t\to+\infty} x(t) = x_i^*.$$

In other words, each goal state x_i^* is an attractor of the associated vector field F_i. Furthermore we assume the existence of functions $\varphi_i : \mathcal{D} \to \mathbb{R}_+, i \in \{1,2\}$ that obey $\varphi_i(x_i^*) = 0$ and increase with distance from the goal state. The functions φ_i measure the degree to which the i^{th} task has been completed, and can be constructed, e.g., by normalizing Lyapunov functions for the vector fields F_i. In the navigation function approach [18], φ_i is the navigation function for the i^{th} task and F_i is its negative gradient.

Consider as a concrete example the case of navigation in the absence of obstacles. In this case, we define the fields by $F_i(x) = -(x - x_i^*)$, representing proportional control, and the functions φ_i by $\varphi_i(x) = \|x - x_i^*\|_2$, i.e., the Euclidean distance to the goal.

2) Navigation dynamics: We then define the navigation dynamics of the point robot as the sum of the task vector fields F_i weighted by the motivation state m:

$$\dot{x} = f_x(x, m) := m_1 F_1(x) + m_2 F_2(x). \tag{2}$$

Recall that $m \in \Delta^2 = \left\{x \in \mathbb{R}^3 : x_i \geq 0, \sum_{i=1}^3 x_i = 1\right\}$, so m takes the form $m = (m_1, m_2, 1-m_1-m_2)$ where $m_1, m_2 \in [0,1]$ and $m_1 + m_2 \leq 1$. These dynamics can be interpreted as a convex relaxation of a hybrid system, where only one of the two task vector fields can be activated at any given time. The corresponding hybrid system would allow m to take values in $\{(1,0,0), (0,1,0), (0,0,1)\}$, i.e., the vertices of the simplex.

In contrast to the hybrid system, the motivation dynamics system can take any convex combination of the task vector fields and is thus able to encode a richer set of behaviors than would be possible using a hybrid system. For example, consider a scenario where the robot is unsure of which task to perform. The hybrid system would have to choose one of the tasks or to perform neither. In contrast, the motivation dynamics system has the flexibility to put nonzero weight on both tasks, thereby hedging its bets until further information arrives. This greater flexibility of behavior is one of the virtues of the motivation dynamics approach.

C. Motivation dynamics

The dynamics of m are given by

$$\dot{m} = f_m(m, v; \sigma) \tag{3}$$

where the components of f_m are given by

$$\begin{aligned}\dot{m}_1 &= v_1 m_U - m_1(1/v_1 - v_1 m_U + \sigma m_2)\\ \dot{m}_2 &= v_2 m_U - m_2(1/v_2 - v_2 m_U + \sigma m_1),\end{aligned}$$

$\sigma > 0$ is a fixed parameter, and $m_U = 1 - m_1 - m_2$ completes the simplex Δ^2. The dynamics of m_U are inherited from this relation, which implies that $\dot{m}_U = -(\dot{m}_1 + \dot{m}_2)$. These dynamics were previously studied in [7] as a model of decision making in biological systems.

The dynamics (3) can be thought of as a dynamical soft maximum operation which put maximum weight m_i on the task i with the highest corresponding value v_i. More formally, the dynamics (3) embeds an unfolded pitchfork bifurcation with the task values v_i serving as unfolding parameters [8], [9]. When both task values v_i are low, (3) has a single stable equilibrium corresponding to a deadlock phenomenon in which the system attaches little weight to either task. As the values v_i are increased, the dynamics (3) exhibit a bifurcation which destabilizes the deadlock equilibrium and produces equilibria that correspond to definite choices for one task or the other. Tools from bifurcation theory [22] can be used to analyze the details of the bifurcation, but such analysis is outside the scope of this paper.

The dynamics (3) apply to the case of $N = 2$ tasks. Extensions to the case $N \geq 2$ have been considered in recent and ongoing work, e.g., [23], [24]. In this paper, we focus on the case $N = 2$ for simplicity of exposition.

D. Value dynamics

The robot maintains a value state v that encodes the robot's perception of the importance of its various tasks. Maintaining an explicit representation of the task values allows task valuation and selection to be decoupled, and facilitates interfacing motivation dynamics with external information, such as the sensors considered in Section IV. The value state v has dynamics

$$\dot{v} = f_v(v, x) + f_s(t) \tag{4}$$

where f_v represents internal state behavior and f_s represents external stimuli. The dynamics are structured as the sum of two functions representing our desired distinction between internally-driven default behavior and externally-driven behavior performed in response to stimuli.

For the recurrent two-point patrol, we pick f_v such that the i^{th} value state has dynamics

$$\epsilon_\lambda \dot{v}_i = -(v^* \varphi_i(x) - v_i), i \in \{1,2\}, \tag{5}$$

where $v^* > 0$ is a positive gain. The parameter $\epsilon_\lambda \geq 0$ is the inverse of the proportional gain in the feedback loop coupling φ_i to v_i and can be interpreted as the time scale of the coupling: larger values of ϵ_λ correspond to v_i following φ_i more slowly. In the limit $\epsilon_\lambda \to 0$, the function values φ_i are coupled directly to v_i, i.e., $v_i = v^* \varphi_i(x)$.

In the following section, we review conditions under which these dynamics produce a cyclic behavior corresponding to recurrent patrol.

III. Limit cycles

For particular choices of the system functions φ_i, F_i, and f_v, there is a finite value V^* such that setting the parameter $v^* > V^*$ results in the dynamics shown in Figure 2 exhibiting a stable limit cycle. In this section we review a result which appears as Theorem 3 of [1] and establish the existence of the threshold V^* beyond which cyclic behavior is guaranteed.

A. Task specification

We focus on a specific beacon-homing task, where the task functions φ_i and F_i are defined as follows. We consider a convex, obstacle-free domain $\mathcal{D} \subseteq \mathbb{R}^2$ and define

$$\varphi_i(x) = \|x - x_i^*\|_2 \tag{6}$$

as the Euclidean distance to the i^{th} goal state and

$$F_i(x) = -(x - x_i^*) = -\nabla(\|x - x_i^*\|_2^2)/2 \tag{7}$$

to be the vector field that maps each point x to the negative of the spatial error $x - x_i^*$.

B. Value dynamics

We specify value dynamics (4) where $f_s = 0$ and the components of f_v are those introduced above as (5), namely

$$\epsilon_\lambda \dot{v}_i = -(v^* \varphi_i(x) - v_i), i \in \{1, 2\}. \tag{8}$$

The parameter $\epsilon_\lambda \geq 0$ is the inverse of the proportional gain in the feedback loop coupling φ_i to v_i. In the limit $\epsilon_\lambda \to 0$, the function values φ_i are coupled directly to v_i, i.e., $v_i = v^* \varphi_i(x)$.

The intuition behind this particular choice of dynamics is as follows. When the robot is far from its i^{th} goal, φ_i takes a positive value and v_i increases towards $v^* \varphi_i(x)$. This motivates the robot to perform task i, i.e., navigate towards the goal x_i^*. At the goal, φ_i takes value zero and thus v_i tends exponentially to zero, thus encouraging the robot to shift its motivation to other tasks.

C. Formal guarantee

Define the closed-loop system state vector $z = (x, m, v) \in \mathcal{D} \times \Delta^2 \times \mathbb{R}^2$. The dynamics of z are given by

$$\dot{z} = f_z(z) := (f_x(x, m), f_m(m, x), f_v(v, x)). \tag{9}$$

Now, we study the dynamics of (9) with φ_i, F_i, and f_v given by (6), (7), and (8), respectively. Theorem 3 of [1] shows that there exists an $\epsilon > 0$ such that for each $\epsilon_\lambda \in [0, \epsilon)$, there exists a finite $V_{\epsilon_\lambda}^*$ such that for $v > V_{\epsilon_\lambda}^*$, the system (9) exhibits a stable limit cycle. This is the result that we illustrate through a physical implementation in this paper. Precisely, we have

Theorem 1. *[1, Theorem 3] Accepting [1, Conjecture 20], for* $\sigma > 6$, *there exists a stable limit cycle of* (9) *for sufficiently small, but finite, values of* ϵ_λ *and* ϵ_v. *Equivalently, fixing* λ, *there exists a stable limit cycle of* (9) *for sufficiently large, but finite, values of* v^*.

Proof. See [1]. □

The practical implication of Theorem 1 is that by tuning the single parameter v^*, one can ensure that the motivation dynamics system (9) exhibits a stable limit cycle. This greatly simplifies the process of implementing the motivation dynamics. In the context of logic-based synthesis, the limit cycle would be referred to as a recurrent patrol mission.

IV. Integrating external stimuli

The motivation dynamics system presented in [1] and reviewed in Section III did not include any external stimuli. In this section, we show how to modify the previously-published motivation dynamics strategy to integrate external stimuli, thereby permitting the system to modify its actions in response to the outside world. The stimuli are represented in Figure 2 as the input $\mathbf{u}$ entering in the value dynamics block.

For the purposes of demonstration, we consider a scenario where the stimulus is binary and corresponds to one of the robot's two navigation tasks. Upon receiving a noisy stimulus, the robot must respond by navigating to the correct target. Methods developed for this scenario can readily be extended to develop more complex stimulus responses, much as binary representations of stimuli are used as the basis of reactive logic-based planners [20]. We present the essential components of our approach here in the main text and provide full details in the Appendix.

Our method for integrating external stimuli is inspired by the cognitive science literature on perceptual decision making. This is consistent with our choice of decision-making mechanism, as the dynamics (3) were originally developed to model perceptual decision making in biological swarms. Often, the stimulus comes as a signal to be integrated over time, and in our scenario the decision maker must decide between two competing classes to which the signal may belong. Wald [25] showed that the optimal strategy for this task is the sequential probability ratio test (SPRT). Such two-alternative choice tasks have been studied in human and animal subjects using visual and auditory stimuli, and models based on the SPRT have been shown to provide a unifying account for a wide variety of behavioral data [26].

Logic-based techniques for synthesizing hybrid controllers have also been extended to integrate external stimuli. In order to connect external stimuli with the symbolic internal representations used by hybrid controllers, logic-based techniques require defining a discrete set of possible external stimuli which will then be used in a synthesis protocol. For example, the authors of [20] assume that the raw sensor data can be processed into a set of discrete state variables corresponding to states of the external world. Such processing can be achieved by recourse to standard statistical signal processing tools, for example by building detectors (using, e.g., the SPRT).

In contrast, our technique maintains a continuous representation of the filtered stimuli. In applications where the stimulus signal-to-noise ratio is low, the time required for a stand-alone detector to integrate sufficient evidence to make a confident decision may be similar to the time available to execute the corresponding action. In such an application, retaining a less-filtered signal in the decision-making system could improve performance by using real-time information about the relative likelihood of the different classifications. These are the types of scenarios where we envision our motivation dynamics system carrying out low-level control to complement a higher-level logic-based controller. In parallel work [27], we have begun to develop performance metrics that quantitatively justify this intuition.

A. Stimulus model

For our experiments, we suppose that our robot is endowed with a sensor that can measure a noisy scalar signal from the environment which indicates which of its navigation tasks to perform. This models, e.g., a perception module that processes raw sensor data and outputs a signal that captures relevant features of the environment. Specifically, at time t the robot can measure an environmental signal

$$e(t) = \mu s(t) + \sigma_s z(t), \tag{10}$$

where $z(t)$ is Gaussian noise with mean zero, unit standard deviation, and which is uncorrelated over time,

$$s(t) = \begin{cases} 0, & \text{remain idle (task 0)} \\ +1, & \text{perform task 1} \\ -1, & \text{perform task 2,} \end{cases}$$

and $\mu, \sigma_s > 0$ are known parameters. The two parameters μ and σ_s can be interpreted as the signal and sensor noise strengths, respectively.

The model (10) is defined in continuous time. However, in implementations, the signals are measured in discrete time. For purposes of analysis, we consider a continuous-time limit of the discrete-time signals in the Appendix. We distinguish between continuous- and discrete-time signals by using different brackets for the time argument. Concretely, the signal e measured at continuous time t is denoted by $e(t)$ and at discrete time t by $e[t]$.

B. Probability ratio test

Given the stimulus model (10), deciding which action to perform at time t reduces to deciding whether the uncorrupted signal $s(t)$ is equal to $+1$ or -1. The SPRT provides an optimal statistical test for making this decision based on the log-likelihood ratio. Given a sequence of discrete-time measurements $\{e[\tau]\}_{\tau=1}^{t}$, the log-likelihood ratio of two hypotheses i, j is

$$\Lambda_{ij}[t] = \sum_{\tau=1}^{t} \log \frac{p_i(e[\tau])}{p_j(e[\tau])}, \tag{11}$$

where $p_i(e)$ is the probability density function of the signal e conditional on the true signal $s(t)$ being associated with task $i \in \{0, 1, 2\}$.

The SPRT then uses the log-likelihood ratio $\Lambda_{12}[t]$ to classify the signal by comparing it against thresholds $\log Z_2 < 0 < \log Z_1 < \infty$. If $\Lambda_{12}[t] > \log Z_1$, then the signal is classified as corresponding to task 1; if $\Lambda_{12}[t] < \log Z_2$, the signal is classified as corresponding to task 2; otherwise, the signal cannot be confidently classified as corresponding to either task and more information is required. The thresholds Z_1, Z_2 are tuned to obtain desired rates of Type I (false positive) and Type II (false negative) errors [25], [26].

C. Using log-probability ratios as values

We connect the environmental stimulus with the motivation dynamics system using the log-probability ratios as values. In particular, for each task i, we set the associated value $v_i(t)$ equal to

$$v_i(t) = v^* \max\{\Lambda_{i0}(t), \varepsilon_v\}, \tag{12}$$

where $\Lambda_{i0}(t)$ is the log-likelihood ratio which compares the probability that $s(t)$ corresponds to task i to the probability that $s(t)$ corresponds to performing neither task, $\varepsilon_v > 0$ is a constant which we choose to ensure that the value of v_i passed to the motivation dynamics (3) is positive, as required, and $v^* > 0$ is a gain. The intuition is as follows: the more likely that the signal corresponds to task i, the higher its associated value v_i and corresponding motivation value m_i will be. This results in the motivation dynamics system putting more weight on the task.

Note that we have used the continuous-time quantity $\Lambda_{i0}(t)$ rather than the discrete-time quantity $\Lambda_{i0}[t]$ defined in (11). In Appendix VII-A we show that $\Lambda_{i0}(t)$ is well defined and derive the equation that governs its time evolution. This equation allows us to derive performance guarantees such as Theorem 2 using existing techniques from the decision-making literature, as shown in Appendix VII-B.

D. Comparison to hybrid system

Here, we show a comparison between the motivation dynamics system with log-probability ratio values (12) and an analogous hybrid control system. The precise details of the implementation of (12) are provided in Appendix VII-A. We consider a one-off decision where the robot, initially located at x_0, is presented with a stimulus following the model (10) and the two tasks correspond to navigating to goals located at x_1^* and x_2^*, respectively. The robot is required to respond by navigating to the correct goal. This decision models, e.g., a situation where a robot is navigating through a partially-known environment and only one of two possible passages is free of obstacles. The robot has to use its sensor data to decide which passage is open and navigate to it.

A standard hybrid control strategy for this decision problem would be to construct a system whose discrete mode state a takes one of three values according to the value of the log-likelihood ratio Λ_{12} (11):

$$a(t) = \begin{cases} U, & \text{if } \Lambda_{12} \in [\log Z_2, \log Z_1] \\ 1 & \text{if } \Lambda_{12} \in (\log Z_1, +\infty) \\ 2 & \text{if } \Lambda_{12} \in (-\infty, \log Z_2) \end{cases}, \tag{13}$$

where $Z_2 < Z_1$ are chosen to obtain desired rates of classification errors. The discrete state implements the SPRT and therefore represents the fastest possible classification scheme. When $a(t) = U$, the continuous dynamics are the null dynamics $\dot{x} = 0$, i.e., do not move, while $a(t) = 1$ and $a(t) = 2$ correspond to dynamics $\dot{x} = -(x - x_1^*)$ and $\dot{x} = -(x - x_2^*)$, respectively, that navigate towards the goals. One could design other dynamics for the case $a(t) = U$, e.g., to drive the robot towards the midpoint between the two goals, but this will not necessarily generalize well either to a situation where the robot has to make a series of decisions or where there are more than two goals.

To illustrate the benefits of the motivation dynamics system, we choose stimulus parameters $\mu = 1$ and $\sigma_s = 100$ to represent a situation where the stimulus is very noisy. In such a noisy environment, a detector like the one in (13) would require a long time to classify the stimulus and make a decision. In the absence of a confident classification, the hybrid controller would not move. In contrast, the richer internal representation of the motivation dynamics system allows it to begin to move before it is fully confident about its decision. The parameters v^* and ε_v from (12) are set equal to 1 and 10^{-4}, respectively, and the parameter α from (18) was set equal to zero.

We simulated 1,000 instances of the one-off decision; ten representative trajectories of the motivation dynamics system are shown in Figure 3. The motivation dynamics system converged to the correct goal 59.7% of the time, with a mean response time of 165.6. For an equivalent hybrid system, setting the thresholds $-\log Z_2 = \log Z_1 = \log(0.6/(1-0.6)) \approx 0.405$, results in an accuracy rate of 60%, equivalent to the performance of the motivation dynamics system. In contrast, with these parameters the SPRT at the heart of the hybrid control system requires a mean time of 405.5 just to classify the signal, and a further time of 3.16 to navigate to the chosen goal. Motivation dynamics thus achieves the same probability of converging to the correct goal while taking 60% less time to decide and respond. This time savings are driven by the system's anticipatory behavior, as seen in Figure 3: the trajectories begin moving towards both goals and then smoothly commit to one goal or the other as the system accumulates information from the noisy sensor.

The physical experiments 1 and 2 presented in Section VI-A extend the one-off decision depicted in Figure 3 to the case where the true signal $s(t)$ switches at times that are unknown to the robot and the robot has to respond accordingly by making a series of decisions.

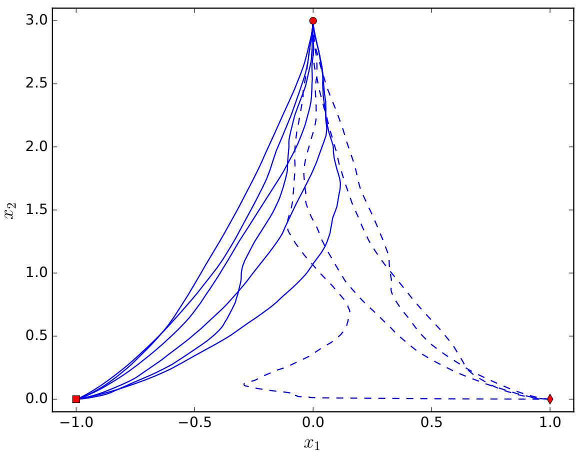


Fig. 3: Ten trajectories using motivation dynamics to perform a one-off decision. The robot starts at $x_0 = (0, 3)$ (marked by a circle) and is given a signal according to (10) instructing it navigate to the goal at $x_1^* = (-1, 0)$ (square). Solid lines show trajectories where the robot navigated to the correct goal; dashed lines the incorrect goal located at $x_2^* = (1, 0)$. Note how, in all cases, the trajectory anticipates the need to move toward both goals and smoothly commits to one goal or the other as time passes and the system accumulates information.

V. Experimental methods

For our empirical work, we use two robots: the Ghost Minitaur, a quadrupedal legged machine [28], and the Dabit Turtlebot 2e [29]. We chose Minitaur in order to show the limit cycle both indoors where we could capture ground truth data and outdoors in a less-structured environment, on a highly dynamic legged platform. We use the Turtlebot to develop the extensions including obstacle avoidance with a LIDAR indoors. We implement both the motivation dynamics scheme and the reactive controller on these robots using sensors that give measurements of the range to beacons. Using these sensors we implement beacon homing controllers in a manner similar to that documented in [30]. We provide details in this section.

A. Robot model

We model each robot as a unicycle, i.e., with dynamics (1)

$$\dot{x}_1 = \nu \cos\theta,\ \ \dot{x}_2 = \nu \sin\theta$$
$$\dot{\theta} = \omega,$$

where $u = (\nu, \omega) \in \mathbb{R}^2$ are the control inputs given in terms of fore-aft speed and turning rate, respectively. This is the standard model for the Turtlebot robot; an empirical characterization of this model for Minitaur is given in [30].

B. Localization model

The system kinematics are then given as follows. Let x_i^* be the i^{th} goal location. As is common in the sensor-based planning literature, we work in the body frame of the robot and denote the coordinates of x_i^* in the body frame by (x_{i1}, x_{i2}). Assuming that x_i^* is not moving in the inertial frame and that the robot kinematics are given by (1), the kinematics of (x_{i1}, x_{i2}) are given by

$$\dot{x}_{i1} = -\nu + \omega x_{i2},\ \ \dot{x}_{i2} = -\omega x_{i1}.$$

In these coordinates, the distance to x_i^* is given by

$$d_i = \sqrt{x_{i1}^2 + x_{i2}^2}. \tag{14}$$

C. Measurement and estimation

To close the estimation loop, we measure the distance d_i to each beacon using the Pulson P-440 RF sensors [31], which yield measurements at approximately 100 Hz. The i^{th} beacon reports a value of d_i corrupted by noise, which we assume is iid over measurements and independent of measurement location. Therefore, we use the measurement model

$$y_i(t) = d_i(t) + \varepsilon(t), \tag{15}$$

where $\varepsilon(t)$ represents the measurement noise. Consistent with other work in the robotics literature [32], we found that it was sufficient to model ε as a Gaussian random variable with mean zero and standard deviation of 10 cm.

We perform state estimation using the ParticleFilter class contained in the Matlab Robotics toolbox [33] to estimate the position of each beacon in the body frame and apply the control law. For simplicity of implementation, we maintain one particle filter for each beacon, and each filter is responsible for maintaining an estimate of the location of its beacon.

D. Control model

Using the estimation of x_i^* in the robot's body frame, we supply the system with the goal

$$x^* = m_1 x_1^* + m_2 x_2^* \tag{16}$$

and construct the homing behavior to x^* based on Equations (7) and (8) of [30], which are based on the control law given in [34, Equation (33)]. Concretely, in the absence of obstacles, we set the robot control inputs equal to

$$\nu = k(x^*)_1,\ \omega = k \arctan((x^*)_2/(x^*)_1), \tag{17}$$

where $k > 0$ is a constant; this mapping constitutes the anchoring relationship f_a shown in Figure 2. When LIDAR data is available, the control inputs are computed based on a projection of x^* into the robot's local free space as described in [30, Equations (7) and (8)]. For our experiments, $\varphi_i = d_i$ and the constants in dynamics (9) are set as $\sigma = 10, \epsilon_\lambda = 1$, and $v^* = 10$.

E. System integration

The system is integrated using the Matlab Robotics toolbox and ROS. We use the libraries [35], [36] to interface the RF sensors and the Hokuyo LIDAR with ROS respectively. Multipath interference is a significant concern when using these sensors, but we were able to minimize it by carefully choosing the experiment location within the indoor environment to avoid metallic objects. With Minitaur, 5000 particles for each beacon with an effective particle ratio of 0.8 provided good performance, while, with Turtlebot, we had to increase the number of particles to 8000 due to less accurate measurements from the onboard IMU.

Adding process noise in the measured linear and angular velocities ν, ω of the robot was important to achieving reasonable filter convergence. For reference, we use a process noise of 0.2 m/s and 0.2 rad/s for the linear and angular speeds, respectively. Additionally, the filter convergence is sensitive to the initial distribution of the particles; we observe faster filter convergence if we initialize the particles on the circle defined by the first range measurement.

The exchange between Matlab and the robot for the commanded and sensed ν, ω is achieved through WiFi, with control update rates of approximately 50 Hz, using the interface between Matlab and the ROS control architecture described in [30].

VI. Empirical results

In this Section we report the results of a series of experiments that implement the robot control law detailed in the previous section. We first demonstrate the effectiveness of motivation dynamics in responding to external stimuli using (12). Next we show that the control law (9) results in a limit cycle in several experiments, thereby verifying the relevance of the theoretical results reported in [1]. A comprehensive list of experiments is given in Table I. The first four experiments on which we report took place indoors at the University of Pennsylvania's PERCH facility, which is equipped with a Vicon [37] motion capture system that we use to record ground truth data. The fifth experiment took place outdoors.

Experiment	Description	Section
1	Stimulus response, no obstacles, Turtlebot	VI-A
2	Stimulus response, obstacles, Turtlebot	VI-A
3	Indoor limit cycle, no obstacles, Minitaur	VI-B1
4	Indoor limit cycle, obstacles, Turtlebot	VI-B2
5	Outdoor limit cycle, Minitaur	VI-C

TABLE I: List of experiments discussed in Section VI.

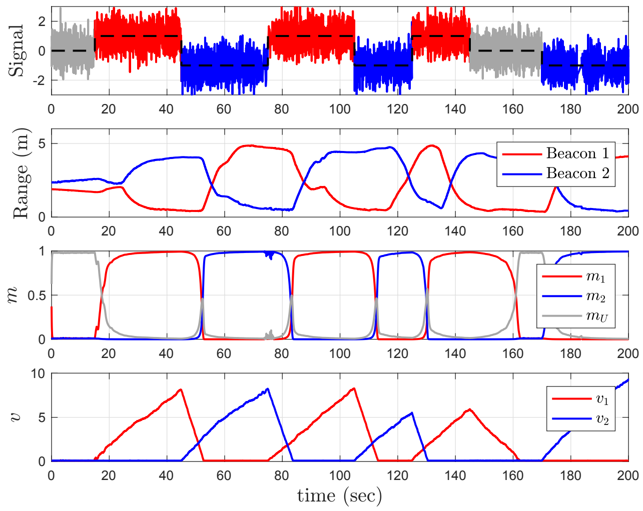


Fig. 4: Experiment 1: stimulus response in an obstacle-free environment with a Turtlebot. From top to bottom, the panels show the signal provided, range to the beacons, and the trajectory of the internal states m and v. The environmental signal in the top panel is modeled by (10) with the mean following the trajectory shown with a black dashed line. The value state v evolves according to (18); the value state v then drives the motivation state m through the dynamics (3).

A. Stimulus-response experiment

Experiments 1 and 2 focus on demonstrating the stimulus response control developed in Section IV. We used the Turtlebot robot to perform navigation tasks in an indoor environment in response to a fictitious stimulus modeled by (10). The stimulus parameters in (10) were set equal to $\mu = 1$ and $\sigma_s = 0.7$, respectively. The leak term α from (18) was set equal to 0.001. All other control parameters were set as reported in Section V.

Figure 4 shows the results of a typical run of Experiment 1, where the environment was free of obstacles. The panels show the stimulus signals $s(t), e(t)$, range to beacons $\varphi(t)$, and internal states $m(t), v(t)$, respectively. The value state v_i increases when the associated task i is indicated by the signal $s(t)$ and decreases otherwise due to the leak term α in (18). The motivation dynamics (3) then put weight on the appropriate task, which results in the robot carrying out the correct navigation behavior. Figure 5 shows the results

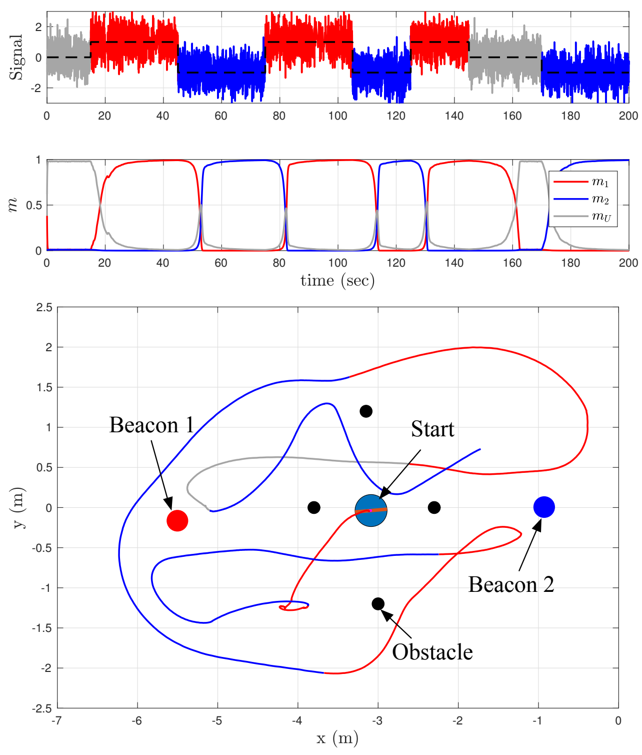


Fig. 5: Experiment 2: stimulus response in an obstacle-strewn environment with a Turtlebot. The top two panels show the signal provided and trajectory of the internal state m as in Figure 4. The bottom panel shows the ground truth robot trajectory and obstacle locations, expressed in the lab frame. Trajectory color represents the behavior commanded by the signal at the corresponding time. Obstacles shown in black.

of a typical run of Experiment 2, where the Turtlebot robot was equipped with a LIDAR sensor to avoid obstacles in the environment. The main panel shows the ground truth physical trajectory of the robot; the color of the trace represents the behavior commanded by the signal at the corresponding time. We see that the robot quickly picks the correct task and effectively navigates towards the corresponding goal while avoiding obstacles.

B. *Indoor limit cycles with ground truth*

We now present the results of two experiments that implement the limit cycle controller described in Section III. Both experiments were performed indoors in the PERCH facility, using the control implementation reported in Section V. As mentioned in Section V, we faced some issues due to multipath interference from the building structure, but were able to mitigate these issues by careful choice of the experimental domain.

1) Minitaur without obstacles: In Experiment 3 we implement the limit cycle controller on a Minitaur robot in a domain without obstacles. Figure 6 shows the ground truth robot trajectory from a typical run of Experiment 3 as measured in the lab frame. The two circles show the two beacon locations, and the circle with line shows the initial pose of the robot. Several features are immediately visible upon inspecting the trajectory. First, the robot has an initial transient and then settles into a cyclic behavior in which it approaches one beacon, then the other. This is the physical manifestation of the limit cycle whose existence is proven in Theorem 1. Second, the cyclic behavior is not symmetric: the points of closest approach to Beacon 2 are much more closely grouped than the points of closest approach to Beacon 1. This is likely due to asymmetries in the robot, which make it prefer walking forwards (towards Beacon 2 in this experiment).

Figure 7 shows the measured range to the beacons, i.e., $\varphi_i(x(t))$, and the internal state consisting of motivation $m(t)$ and value $v(t)$ from a typical run of Experiment 3 with Minitaur. Note that the limit cycle behavior is much more clearly manifest in the internal state trajectory than in the physical trajectory shown in Figure 6, likely because the dynamics of the internal states m and v are numerically integrated in the controller and therefore follow the assumed dynamics more closely than the physical dynamics of x. The asymmetries in the limit cycle are due to the same factors as discussed for Figure 6, as well as an asymmetry in the RF sensor antenna placement due to physical constraints.

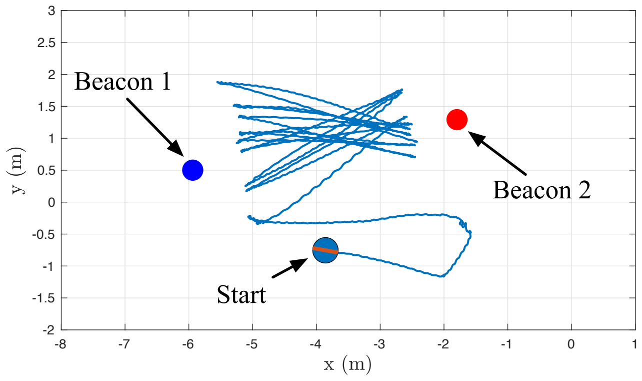


Fig. 6: Experiment 3: indoor limit cycle with Minitaur. Ground truth robot trajectory and beacon locations expressed in the lab frame. The robot persistently patrols back and forth between the beacons.

2) Turtlebot with obstacles: In Experiment 4, we implemented the same limit cycle controller as in Experiment 3 on a Turtlebot 2e robot using an obstacle avoidance behavior as in [30]. The behavior is reactive and uses sensor data to compute the robot's local free space, projects the goal location onto this free space, and drives the robot towards this projected goal. Figures 8 and 9 are analogous to Figures 6 and 7, respectively, for this new experiment. The behavior is qualitatively similar to the obstacle-free results with Minitaur, suggesting that the theoretical guarantee from Theorem 1 can be extended to this new, obstacle-strewn case.

As in the Minitaur experiments shown in Figures 6 and 7, the robot does not come arbitrarily close to the beacon but rather maintains some minimum distance, in this case approximately 2 m. Experimentation suggests that this minimum distance can be controlled by adjusting the parameter σ in the motivation dynamics (3). Extending Theorem 1 to the obstacle-strewn case and analyzing the relationship between σ and the minimum distance to the goal can both be understood

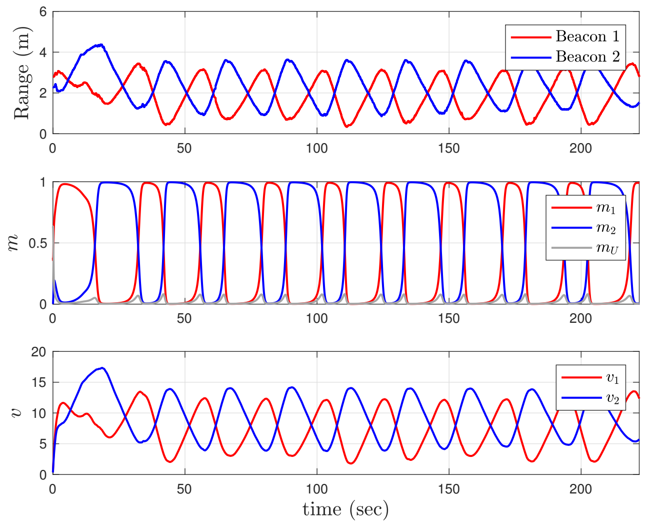


Fig. 7: Experiment 3: indoor limit cycle with Minitaur. From top to bottom, the panels show the range to beacons, i.e., $\varphi_i(x(t))$, and the trajectory of the internal states m and v. The states m and v follow the dynamics (3) and (8), respectively.

using formal dynamical systems tools. This work is currently underway.

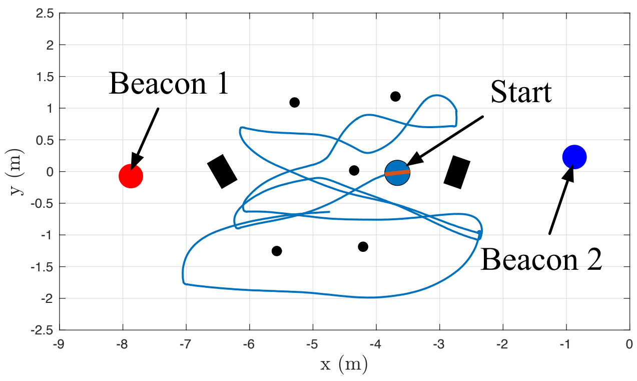


Fig. 8: Experiment 4: indoor limit cycle in an obstacle-strewn domain with Turtlebot. Ground truth robot trajectory and obstacle locations, expressed in the lab frame. Obstacles are shown in black. The robot persistently patrols back and forth between the beacons while reactively avoiding the obstacles.

C. Outdoor experiment

Experiment 5, whose setup is shown in Figure 1, was carried out in a public park in Philadelphia, PA. Outside, multipath interference was insignificant. As in Experiment 3, we implemented the limit cycle controller on a Minitaur robot.

Figure 10 shows the measured range to the beacons, i.e., $\varphi_i(x(t))$, and the internal states $m(t)$ and $v(t)$. As in the indoor experiment, the robot exhibits a cyclic behavior of repeatedly coming closer to one beacon and then the other. Also as in the indoor experiment, the cyclic behavior appears more clearly in the internal state than the range measurements. The irregular appearance of the cyclic behavior in the outdoor experiment is due to the greater difficulty that the robot had

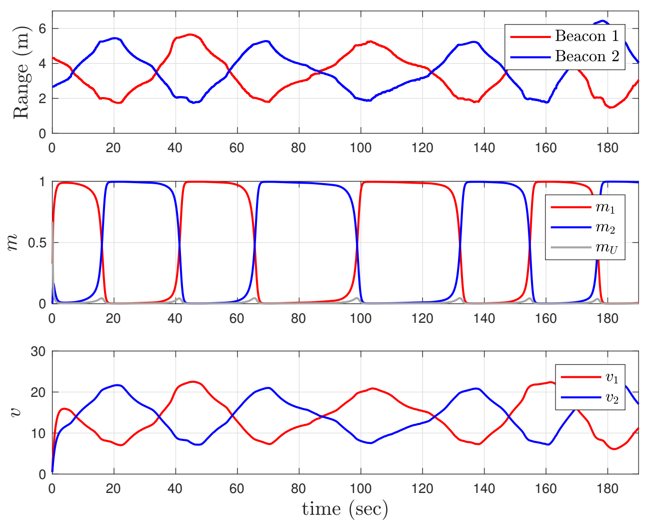


Fig. 9: Experiment 4: indoor limit cycle in an obstacle-strewn domain with Turtlebot. From top to bottom, the panels show the range to beacons, i.e., $\varphi_i(x(t))$, and the trajectory of the internal states m and v. The states m and v follow the dynamics (3) and (8), respectively.

in negotiating the domain, which was a moderately-well-kept grass field with an asphalt path in the middle. Locomotion on this terrain is less predictable than locomotion indoors on hard tiles, which accounts for the irregular cyclic behavior in this experiment.

Despite the difficulties posed by this more challenging environment, the cyclic behavior of the robot persists over several switches of target, showing the attracting nature of the limit cycle. In this environment, the unicycle model of the robot is even more poorly respected, which results in the control law being less effective. It should further be noted that these results were intended to show the robustness of the basic control strategy, and so were obtained without any special efforts being made to tune the control or estimation system parameters to account for the features of the environment. Moderate parameter tuning would likely yield more regular oscillatory behavior if such behavior were desired.

VII. Conclusion

In this paper we present the first physical implementation of the reactive vector field planner (9) introduced and studied in [1]. We show that the planner can naturally integrate external stimuli and respond by producing appropriate behaviors. In the absence of external stimuli, for a sufficiently-high value of a gain parameter, this planner exhibits stable limit cycles where the robot repeatedly navigates towards two target locations. We show that this same behavior is achieved in the physical implementation, thereby showing that our planner encodes a recurrent patrol mission covering the two targets. Furthermore, the planner is robust, as the identical control strategy exhibited the limit cycle behavior both indoors and outdoors without additional tuning.

The experimental results presented here extend the formal results from [1] in two ways. First, we show that the same

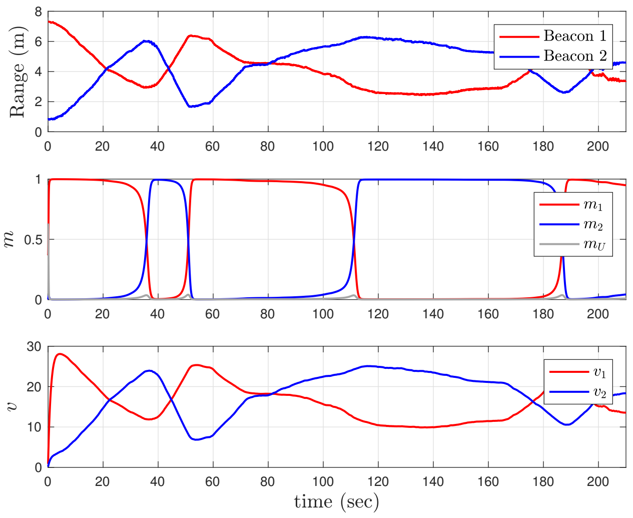


Fig. 10: Experiment 5: outdoor limit cycle with Minitaur. From top to bottom, the panels show the range to beacons, i.e., $\varphi_i(x(t))$, and the trajectory of the internal states m and v. The states m and v follow dynamics (3) and (18), respectively.

motivation dynamics planner can be readily extended to the case where the robot domain is strewn with obstacles. These experimental results suggest extending the formal results to include the case where the vector fields F_i have generic Lyapunov functions. This work is underway. Second, we show how to integrate external stimuli into the decision-making mechanism. For a situation where a stimulus requires performing an associated action, we show that using log-likelihood ratios as the action values results in a system that can quickly and accurately respond to noisy stimuli, and we provide an associated performance guarantee.

The formal results demonstrated in this paper are restricted to the case of two low-level tasks. However, work is in progress that extends both the formal and the experimental results presented here to an arbitrary number of low-level tasks. This work will be the subject of a separate report.

Acknowledgement

This work was funded in part by AFRL grant FA8650-15-D-1845 (subcontracts 669737-1 and 669737-6). We wish to acknowledge the support of Ghost Robotics in this work.

Appendix

A. *Likelihood ratio and value dynamics*

In this appendix, we show that the continuous-time likelihood ratio $\Lambda_{i0}(t)$ that arises from the discrete time ratio (11) is well defined and derive the equation that governs its time evolution.

We implement the value model (12) in discrete time by setting

$$\Lambda_{i0}[t] = (1-\alpha)\Lambda_{i0}[t-1] + \log(p_i(e[t])) - \log(p_0(e[t])), \quad (18)$$

where $[t]$ denotes the discrete time step t associated with a given sensor reading and $\alpha > 0$ is a constant. The parameter α encourages the system to discount old stimuli, thereby allowing it to adjust to a changing environment. Such a structure is commonly used in the cognitive science literature, where the α term is referred to as a "leak" or a "forgetting factor" [26]. In the absence of external stimuli, these dynamics will push v_i towards the value of zero, which corresponds to hypotheses i and 0 being equally probable.

For purposes of interpretation and analysis, it is useful to relate the discrete-time dynamics (18) to the general continuous-time value dynamics (4). This can be done by taking the continuous-time limit of (18) and interpreting the resulting dynamics as setting the i^{th} component of the stimulus function f_s equal to

$$f_{s,i}(t) = \left(-\alpha v_i(t) + v^* \log(p_i(e(t))) - v^* \log(p_0(e(t)))\right),$$

where $e(t)$ is the sensor reading at a continuous time t. The limiting continuous-time dynamics must be treated with care for two reasons. First, the stochastic nature of the stimulus $e(t)$ means that the function f_s is a stochastic function of time. Thus, the dynamics $\dot{v}_i = f_{s,i}(t)$ are more properly modeled as a stochastic differential equation (SDE), for which [38] is a standard reference. Second, the discrete-time equation (18) implicitly assumes a sensor sampling rate, i.e., the time Δt between samples. Taking the appropriate continuous-time limit of (18) thus implicitly takes the sensor sampling frequency to zero at the same rate as Δt. Fortunately for our purposes, the limiting process has been well studied in the literature, e.g., [26, Appendix A, pp. 744–746], [38].

The most straightforward way to understand the limiting process is to think in terms of numerical integration. Continuous-time ordinary differential equations of the form $\dot{x} = f(x)$ can be numerically integrated using discrete-time algorithms, such as the Euler method. Given an initial condition $x(0) = 0$, the Euler method constructs a discrete-time approximation $x[n+1] = x[n] + \Delta t f(x[n])$, where $x[n]$ approximates $x(n\Delta t)$. SDEs admit analogous numerical integration schemes [38]; the SDE analog of the Euler method is known as Euler-Maruyama.

Using the model (10), the discrete-time likelihood ratio evolution (18) becomes

$$\begin{aligned}\Lambda_{i0}[t] &= \frac{\mu}{2\sigma_s^2}(\pm 2e[t] - \mu) + (1-\alpha)\Lambda_{i0}[t-1] \quad (19)\\ &= \frac{\mu^2}{2\sigma_s^2}(\pm 2s[t] - 1) + (1-\alpha)\Lambda_{i0}[t-1] \pm \frac{\mu}{\sigma_s} z[t-1],\end{aligned}$$

where the positive sign corresponds to the case $i = 1$, the negative sign to the case $i = 2$, and $z[t] \sim \mathcal{N}(0,1)$ is again a Gaussian random variable. Comparing terms shows that the discrete-time process (19) is precisely the Euler-Maruyama approximation [38, Section 10.2] for the continuous-time SDE

$$\mathrm{d}\Lambda_{i0}(t) = a_i \mathrm{d}t + b_i \mathrm{d}W(t), \quad (20)$$

where the constants a_i and b_i are defined by

$$\begin{aligned}a_i &= -\frac{\alpha}{\Delta t}\Lambda_{i0}(t) + \frac{\mu^2}{2\sigma_s^2 \Delta t}(\pm 2s(t) - 1)\\ b_i &= \pm \frac{\mu}{\sigma_s \sqrt{\Delta t}},\end{aligned}$$

and the positive and negative signs again correspond to $i = 1$ and $i = 2$, respectively.

The equation (20) is the continuous-time representation of $f_{s,i}$ in (4). The terms in (20) can be interpreted as follows. The constant a_i scales the deterministic evolution due to the mean stimulus and the forgetting factor and b_i scales the stochastic Weiner increment $\mathrm{d}W(t)$ which is the formal mathematical representation of the stimulus noise in the continuous-time limit [39], [26]. The SDE (20) is sometimes referred to as a drift-diffusion equation, with a_i being a drift and b_i a diffusion term. Note that μ/σ_s can be interpreted as the signal-to-noise ratio of individual sensor readings, and $\mu/(\sigma_s\sqrt{\Delta t})$ can be interpreted as the signal-to-noise ratio scaled to account for the sampling time Δt.

The difference equation (18) and the SDE (20) are equivalent in the sense that solutions $\Lambda_{i0}[t]$ of (18) are a so-called strongly-convergent approximation of solutions of (20) [38]. This equivalence allows us to derive performance guarantees such as the following Theorem 2 using existing techniques from the decision-making literature.

B. Performance guarantee

Performance guarantees for the motivation dynamics stimulus response behavior can be readily obtained using techniques from the decision-making literature. For example, the following theorem holds.

Theorem 2. *Consider the motivation dynamics* (3) *with values* v_i *following* (18) *under the stimulus model* (10). *Let* $\bar{\Lambda} = (\Lambda_{10} + \Lambda_{20})/2$ *and* $\Delta\Lambda = \Lambda_{10} - \Lambda_{20}$ *and let* $\beta = \mu/\sigma_s$ *denote the signal-to-noise ratio of the stimulus. Then,*

1) $\bar{\Lambda}$ *is stably attracted to the value* $-\beta^2/2\alpha$*, and*
2) *For small* α*, the expected time to decision DT and error rate ER, i.e., probability of selecting the wrong goal, are approximately given by*

$$\mathit{DT} = \frac{K(\sigma)\Delta t}{2\alpha}\tanh(K(\sigma)\beta^2/8\alpha),$$

$$\mathit{ER} = \frac{1}{1+\exp(K(\sigma)\beta^2/4\alpha)},$$

where $K(\sigma)$ *is a constant given by [7, Equation 5].*

Proof. The dynamics of $\bar{\Lambda}$ and $\Delta\Lambda$ follow from (20). The stochastic terms in $\dot{\bar{\Lambda}}$ cancel, yielding

$$\dot{\bar{\Lambda}} = -\frac{\alpha}{\Delta t}\bar{\Lambda} - \frac{\beta^2}{2\Delta t}.$$

Similarly, the dynamics of $\Delta\Lambda$ are

$$\mathrm{d}\Delta\Lambda = \left(-\frac{\alpha}{\Delta t}\Delta\Lambda + \frac{\beta^2}{\Delta t}\right)\mathrm{d}t + 2\frac{\beta}{\sqrt{\Delta t}}\mathrm{d}W(t),$$

where $\mathrm{d}W(t)$ is a Wiener increment as in (20). The first statement follows directly from the dynamics of $\bar{\Lambda}$.

The second statement follows by noting that the dynamics of $\Delta\Lambda$ constitute an Ornstein-Uhlenbeck process and that $v_i(t) = v^*\Lambda_{i0}(t)$ when $\Lambda_{i0} > 0$. As shown in [7, Equation 5], the motivation dynamics makes a decision when $K(\sigma) < |\Delta v|/\bar{v} \approx |\Delta\Lambda|/\bar{\Lambda}$, where $K(\sigma)$ is a constant. Then the expressions for DT and ER follow from Equations A65 and A64 of [26], respectively. □

The value of this result is to give an analytical measure of two performance criteria for the motivation dynamics technique. These analytical results help elucidate trends; for example, the expected decision time scales with the sampling interval Δt, and both performance metrics scale with the ratio $K(\sigma)\beta^2/\alpha$.

References

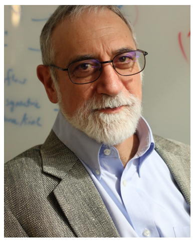
[1] P. B. Reverdy and D. E. Koditschek, “A dynamical system for prioritizing and coordinating motivations,” *SIAM J. Appl. Dynamical Syst.*, vol. 17, no. 2, pp. 1683–1715, 2018.
[2] P. Cisek and J. F. Kalaska, “Neural mechanisms for interacting with a world full of action choices,” *Annual rev. of neuroscience*, vol. 33, pp. 269–298, 2010.
[3] L. Barca and G. Pezzulo, “Tracking second thoughts: Continuous and discrete revision processes during visual lexical decision,” *PLoS One*, vol. 10, no. 2, p. e0116193, 2015.
[4] N. F. Lepora and G. Pezzulo, “Embodied choice: how action influences perceptual decision making,” *PLoS computational biology*, vol. 11, no. 4, p. e1004110, 2015.
[5] P. C. Trimmer, A. I. Houston, J. A. Marshall, R. Bogacz, E. S. Paul, M. T. Mendl, and J. M. McNamara, “Mammalian choices: combining fast-but-inaccurate and slow-but-accurate decision-making systems,” *Proc. Roy. Soc. London B*, vol. 275, no. 1649, pp. 2353–2361, 2008.
[6] J.-H. Song, “Abandoning and modifying one action plan for alternatives,” *Phil. Trans. R. Soc. B*, vol. 372, no. 1718, p. 20160195, 2017.
[7] D. Pais, P. M. Hogan, T. Schlegel, N. R. Franks, N. E. Leonard, and J. A. Marshall, “A mechanism for value-sensitive decision-making,” *PloS one*, vol. 8, no. 9, p. e73216, 2013.
[8] R. Gray, A. Franci, V. Srivastava, and N. E. Leonard, “Multi-agent decision-making dynamics inspired by honeybees,” *IEEE Transactions on Control of Network Systems*, vol. 5, no. 2, pp. 793–806, 2018.
[9] P. B. Reverdy, “Two paths to finding the pitchfork bifurcation in motivation dynamics,” in *Proc. IEEE Conf. Decision and Control*, 2019, pp. 8030–8035.
[10] D. Liberzon, *Switching in systems and control.* Springer Science & Business Media, 2003.
[11] 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. [Online]. Available: http://dx.doi.org/10.1177/02783649922066385
[12] R. Fierro, A. K. Das, V. Kumar, and J. P. Ostrowski, “Hybrid control of formations of robots,” in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 1, 2001, pp. 157–162 vol.1.
[13] C. Altafini, A. Speranzon, and K. H. Johansson, “Hybrid control of a truck and trailer vehicle,” in *Hybrid Systems: Computation and Control*, C. J. Tomlin and M. R. Greenstreet, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 21–34.
[14] R. G. Sanfelice and E. Frazzoli, “A hybrid control framework for robust maneuver-based motion planning,” in *2008 American Control Conference*, 2008, pp. 2254–2259.
[15] S. R. Lindemann and S. M. LaValle, “Smoothly blending vector fields for global robot navigation,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 3553–3559.
[16] D. Panagou, “Motion planning and collision avoidance using navigation vector fields,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2513–2518.
[17] D. Gopinath, S. Jain, and B. D. Argall, “Human-in-the-loop optimization of shared autonomy in assistive robotics,” *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 247–254, 2017.
[18] D. E. Koditschek and E. Rimon, “Robot navigation functions on manifolds with boundary,” *Advances in appl. mathematics*, vol. 11, no. 4, pp. 412–442, 1990.
[19] 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. [Online]. Available: https://doi.org/10.1177/027836498600500106
[20] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, “Temporal-logic-based reactive mission and motion planning,” *IEEE Trans. Robot.*, vol. 25, no. 6, pp. 1370–1381, 2009.

[21] J. Liu, N. Ozay, U. Topcu, and R. M. Murray, "Synthesis of reactive switching protocols from temporal logic specifications," *IEEE Trans. Autom. Control*, vol. 58, no. 7, pp. 1771–1785, 2013.
[22] M. Golubitsky and D. Schaeffer, *Singularities and Groups in Bifurcation Theory*, ser. Applied Mathematical Sciences. Springer, 1985, vol. 51.
[23] A. Reina, J. A. R. Marshall, V. Trianni, and T. Bose, "Model of the best-of-n nest-site selection process in honeybees," *Physical Review E*, vol. 95, no. 5, 2017.
[24] P. Reverdy, "Dynamical, value-based decision making among N options," 2020. [Online]. Available: https://arxiv.org/abs/2003.03874
[25] A. Wald, "Sequential tests of statistical hypotheses," *Ann. of Math. Stat.*, vol. 16, no. 2, pp. 117–186, 1945.
[26] R. Bogacz, E. Brown, J. Moehlis, P. Holmes, and J. D. Cohen, "The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks." *Psychological Rev.*, vol. 113, no. 4, p. 700, 2006.
[27] P. Reverdy, "Performance metrics for a physically-situated stimulus response task," in *The 4th Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM)*, 2019.
[28] Ghost Minitaur. [Online]. Available: https://www.ghostrobotics.io/minitaur
[29] D. Industries. Turtlebot 2e. [Online]. Available: https://dabit.industries/products/turtlebot-2e
[30] V. Vasilopoulos, O. Arslan, A. De, and D. E. Koditschek, "Sensor-Based Legged Robot Homing Using Range-Only Target Localization," in *IEEE Int. Conf. Robotics and Biomimetics*, 2017, pp. 2630–2637.
[31] P-440. [Online]. Available: http://www.timedomain.com/products/pulson-440/
[32] D. Kurth, G. Kantor, and S. Singh, "Experimental results in range-only localization with radio," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2003, pp. 974–979.
[33] Mathworks. Robotics system toolbox. [Online]. Available: https://www.mathworks.com/help/pdf_doc/robotics/robotics_ug.pdf
[34] 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.
[35] [Online]. Available: https://github.com/ashariati/pulson_ros
[36] [Online]. Available: https://wiki.ros.org/urg_node
[37] Vicon Vantage V16. [Online]. Available: https://www.vicon.com/file/vicon/vantagebrochure-021216-web-94650.pdf
[38] P. Kloden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, ser. Applications of Mathematics. Springer Berlin, Germany, 1992, vol. 23.
[39] C. Gardiner, *Stochastic Methods: A Handbook for the Natural and Social Sciences*, 4th ed., ser. Springer Series in Synergetics. Springer, 2009, vol. 13.

Paul B. Reverdy Paul Reverdy received the B.S. degree in Engineering Physics and the B.A. degree in Applied Mathematics from the University of California, Berkeley in 2007, and the M.A. and Ph.D. degrees in Mechanical and Aerospace Engineering from Princeton University in 2011 and 2014, respectively. He is an Assistant Professor in Aerospace and Mechanical Engineering at the University of Arizona. Since July 2020, he has been at Amazon, working on navigation for the Scout delivery robot.

From 2007 to 2009, he worked as a research assistant at the Federal Reserve Board of Governors, Washington, DC. From 2014 to 2017, he was a postdoctoral fellow in Electrical and Systems Engineering at the University of Pennsylvania, where he was affiliated with the GRASP laboratory. His research interests lie at the intersection of human and machine decision making and control, with applications in robotics, machine learning, and engineering design optimization. Dr. Reverdy's awards include a National Defense Science and Engineering Graduate (NDSEG) Fellowship for graduate study and the best student paper award from the 2014 European Control Conference.

Vasileios Vasilopoulos is a Ph.D. candidate in the Department of Mechanical Engineering and Applied Mechanics (MEAM) of the University of Pennsylvania, affiliated with the GRASP Lab. He received his Diploma from the National Technical University of Athens (NTUA) in 2014 and his M.S.E. from the University of Pennsylvania in 2018, both in Mechanical Engineering. His research interests are in the areas of task and motion planning, mobile manipulation and legged robotics. He is a Student Member of the IEEE and the ASME.



Daniel E. Koditschek (S80 M83 SM93 F04) received the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, USA, in 1983.

He is the Alfred Fitler Moore Professor of electrical and systems engineering at the University of Pennsylvania, Philadelphia, PA, USA, where he served as the Chair from 2005–2012. He holds secondary appointments in the Departments of Computer and Information Science and Mechanical Engineering and Applied Mechanics. Prior to joining Penn, he held faculty positions in the Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor and the Electrical Engineering Department, Yale University. His current research interests include robotics, the application of dynamical systems theory to intelligent machines, and nonlinear control. Dr. Koditschek is a member of the AMS, ACM, MAA, SIAM, SICB, and Sigma Xi. He is a fellow of the AAAS. He received the Presidential Young Investigators Award in 1986.