Stochastic Modeling, Control, and Verification of Wild Bodies

Daniel Erik Gierl, Leonardo Bobadilla, Oscar Sanchez, and Steven M. LaValle*

***Abstract*— This paper presents strategies for controlling the distribution of large numbers of minimalist robots (ones containing no sensors or computers). The strategies are implemented by varying area, speed, gate length, or gate configuration in environments composed of regions connected by gates. We demonstrate the effectiveness and practical feasibility of our approach through physical experiments and simulation. We use Continuous Stochastic Logic to verify high level properties of our system and to evaluate the accuracy of our model. Also, we prove that our model is accurate and that our algorithms are efficient with respect to the number of regions and number of bodies.**

I. Introduction

Applications such as agriculture, environmental monitoring, surveillance, and search and rescue would benefit from the deployment of large numbers of robots that solve tasks such as navigation, patrolling, and coverage [9], [25]. However, there are several fundamental challenges that are present and need to be addressed before the full potential of such applications can be realized. These include modeling issues, such as the curse of dimensionality, where the state space and associated costs grow unacceptably as a function of the number of bodies. Other issues involve costs, energy consumption, ease of deployment, and robustness. Scaling issues are especially problematic in the area of micro- and nano-robotics, where very large numbers of bodies with little sensing or actuation capabilities are involved [27].

Most approaches for solving robotics tasks have followed a trend towards becoming more complex. They require precise sensors, robust and reliable actuators, complicated world models, high-bandwidth communication, and powerful computers and algorithms. This complication is unsurprising given the nature of technological progress and the difficulty of the tasks related to multiple robot deployments. On the other hand, no proof exists to support the necessity of this complexity. Although these information-rich approaches have achieved remarkable success, their resource intensive nature may make them hard to scale.

Our motivation is to tackle multiple robot deployment problems through a minimalist approach, where instead of asking ourselves: *what is the most we can do?*, we ask ourselves: *what is the least we need to do it?* From a theoretical standpoint, this becomes a very interesting question that has been explored by several authors [1], [8], [10], [12], [28].

* Daniel Erik Gierl, Oscar Sanchez, and Steven M. LaValle are at the Motion Strategy Lab (MSL), in the Department of Computer Science at the University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA {gierl1,sanche14,lavalle}@uiuc.edu. Leonardo Bobadilla is at the School of Computing and Information Sciences at Florida International University, Miami, FL 33199, USA {bobadilla}@cs.fiu.edu

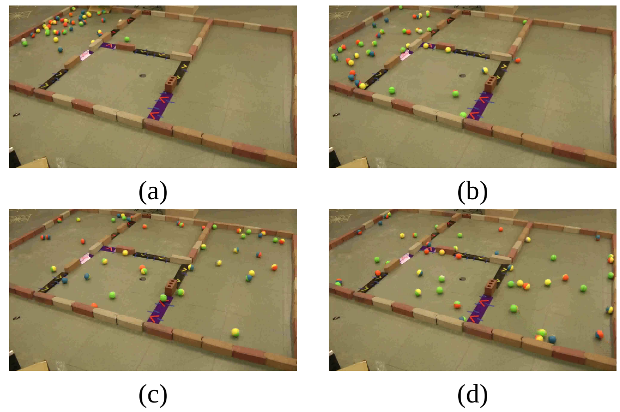


Fig. 1. A sequence of images illustrating a control policy using static gates in a large (6m × 4m) environment composed of regions separated by low brick walls, and connected by small static ramps. Forty-five Weasel balls were used in this experiment (see Fig. 3 for details). Here the mixing process is demonstrated as all of the bodies start in one region and are brought under control to the distribution $(2/9, 1/9, 4/9, 2/9)$ (or $(10, 5, 20, 10)$ when normalized to the number of bodies), clockwise from top-left. In (a) all the bodies save one are still in region 0. In (b) they have begun to disperse through the environment. In (c) they have reached the distribution $(12, 4, 21, 8)$. And in (d) the distribution is $(12, 5, 19, 9)$. Experiments starting in different regions yield similar results.

One approach explored is to build and use stochastic models, due to the non-deterministic aspects underlying many mobile robotics problems [16], [30]. Another approach is to take advantage of randomized motion in bodies to solve tasks [11]. In these, control is exercised on the behavioural configuration space of the robot [29]. Our methods build on previous work done on manipulating robots by means of passive control [2]–[4]; we expand and formalize such control using stochastic models. We propose and demonstrate four different means of control for minimalist bodies using Continuous-Time Markov chains (CTMCs) [7].

II. Preliminaries

A. Regions and Gates

Let $G = (V, E)$ be a connected directed graph representing the environment where V, the *regions*, is a set of disjoint closed subsets with finite boundaries in $\mathbb{R}^2$, and E, the *gates*, is a set of edges connecting V between their boundaries (see Fig. 1). Each $e = (i, j) \in E$ represents the collection of gates going from region i to region j.

B. Wild Bodies

Within this environment move *wild* bodies [4] having no sensors, computers, or communication abilities, each represented by its location in some $v \in V$, its direction of motion, and its bounded speed[1]. The body is actuated sufficiently to induce *wild* behavior in v. We use the definition for wild set forth in [3], in which a wild body is one that, when placed into a bounded region $r \subset \mathbb{R}^2$, it moves along a trajectory that strikes every open interval along the boundary of r infinitely often. This property is related to the notion of *topological transitivity* in dynamic billiards [14], [15], [17], [33].

As a result of this wild motion, the bodies move randomly within the interiors of the regions of the environment, making it difficult to model the state of the body inside of a region. Instead, we model the state space as a n-dimensional vector μ, the distribution of bodies across the regions of the environment. Our notation is $\mu = (\mu_0, \mu_1, ..., \mu_n)$ for $n = |V|$, where μ_i is the proportion of bodies in region i; thus $\sum_i \mu_i = 1$.

C. Continuous Time Markov Chains

Transitions in this state space occur when a body moves across an $e \in E$. Such a transition along $e = (v_i, v_j)$ occurs at a rate R_{ij}, corresponding to the parameter of an exponential distribution from which the time spent in the region before such a transition is made is drawn. Given these transition rates, we model the system using a Continuous-Time Markov chain (CTMC) $C = (V, Q)$, where Q:

$$Q_{ij} = \begin{cases} R_{ij} & (i,j) \in E \\ 0 & o/w \end{cases}$$

is the $n \times n$ instantaneous transition matrix. This can be augmented to Q' by setting the diagonals to be minus the sum of rates for their row, $Q'_{ii} = -\sum_{j \neq i} Q_{ij}$. From this augmented form, the probability that a body is in state j after time t after having started in state i is $P_{ij}(t)$ where $P(t) = e^{tQ'}$. The limiting distribution is μ^* satisfying $\mu Q' = 0$, where $\sum_i \mu_i = 1$.

Our results can be extended from one body to multiple bodies in a straight-forward way by assuming that the bodies cover a small proportion of any region's area and that collisions between them do not disrupt significantly their rate w.r.t. the other bodies.

D. Problem Formulation

Now that we have formulated our model, we can briefly state the problem: *given a desired limiting distribution μ of bodies in an environment G, construct a control policy γ expected to achieve μ using as few additional resources as possible*.

The rest of the paper is organized as follows. In Section III, we show how Q can be found for an environment for any goal distribution μ. In Section IV, we show how a Q and some additional information about the environment can be used to construct a control policy γ to achieve the limiting distribution of Q. Lastly, in Section V we use Continuous Stochastic Logic (CSL) and a model checker to construct probabilistic and temporal bounds for our strategies based on Q.

[1]For our work here, each body is represented by a point particle, though in practice it need only be small with respect to the area of the regions and the widths of the gates.

III. Finding a Transition Matrix from an Environment's Connectivity and a Goal Distribution

Algorithm 1 takes a connected, undirected, anti-reflexive graph G and a positive goal distribution μ, and constructs an instantaneous transition matrix Q for a CTMC whose limiting distribution is μ. The speed of any particular chain's convergence to the limiting distribution is given by its transition matrix's Second Largest Eigenvalue Modulus (SLEM), in which the lower the second largest eigenvalue is, the faster it converges [5].

Algorithm 1 FindTransitionMatrix(G, μ)

Input: $G = (V, E)$ {Graph of the environment}
Input: μ {Desired limiting distribution}
Output: Q {An instantaneous transition matrix}
1: $Q_{ij} \leftarrow \begin{cases} \mu_j & (i,j) \in E \\ 0 & o/w \end{cases}$

Alternative algorithms for finding such Q exist, for example using semi-definite programming [5], [6]. These methods obtain the optimal solution, the Fastest Mixing Markov Chain (FMMC), but we have chosen Algorithm 1 for simplicity.

Proposition 1: **All Q such that $\frac{Q_{ij}}{Q_{ji}} = \frac{\mu_j}{\mu_i}, i \neq j$ have limiting distribution μ.**

Proof:

First, we augment Q to Q' by adding negative diagonals (see Sec. II). The limiting distribution μ^* is defined as $\mu^* Q' = 0$, $\sum_i \mu_i^* = 1$. By definition of the algorithm above we have

$$\begin{aligned} \forall i \forall j, \frac{Q'_{ij}}{Q'_{ji}} &= \frac{\mu_j}{\mu_i} \\ \forall j \forall i, \mu_i Q'_{ij} &= \mu_j Q'_{ji} \\ \forall j, \sum_{i \neq j} \mu_i Q'_{ij} &= \sum_{k \neq j} \mu_j Q'_{jk}, \end{aligned}$$

by breaking up the definition of the diagonal we get

$$\forall j, \sum_{i \neq j} \mu_i Q'_{ij} = -\mu_j Q'_{jj}.$$

By moving over the negative diagonal that was added during augmentation we get

$$\begin{aligned} \forall j, \sum_i \mu_i Q'_{ij} &= 0 \\ \mu Q' &= 0, \end{aligned}$$

therefore, $\mu = \mu^*$.

■

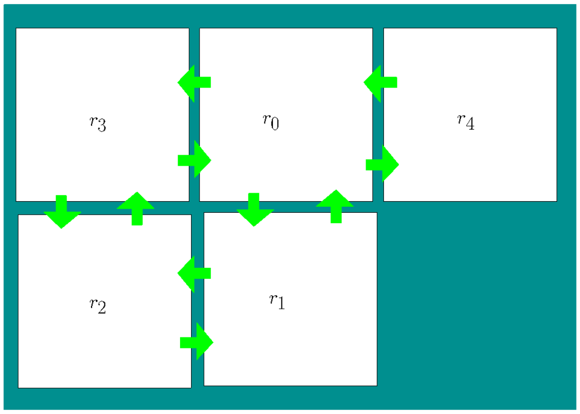


Fig. 2. An example labelled environment with five regions and two gates between every connected pair of regions.

For an example of the algorithm in practice, consider the environment in Fig. 2. If we want the distribution $\mu = (0.1, 0.2, 0.3, 0.3, 0.1)$, we create the following Q:

$$Q = \begin{bmatrix} 0 & 0.2 & 0.3 & 0.3 & 0.1 \\ 0.1 & 0 & 0.3 & 0.3 & 0.1 \\ 0.1 & 0.2 & 0 & 0.3 & 0.1 \\ 0.1 & 0.2 & 0.3 & 0 & 0.1 \\ 0.1 & 0.2 & 0.3 & 0.3 & 0 \end{bmatrix}$$

This follows the intuition that the rates of transition between two regions should be inversely proportional to their desired distributions.[2]

IV. Developing a Control Policy from a Transition Matrix and Other Environment Information

A. Modeling

To construct a control policy γ to gently influence the motion of a body, first we must construct a model for how the body b moves through an environment G. This motion is captured by the rate of transition R_{ij} of b between connected regions i and j. Due to the body's *wild* motion, at a point in time t, the body's location in region i is uniformly distributed as is its direction.[3] Let $Q^i_j = \frac{Q_{ij}}{Q_{ji}}$, $m^i_j = \frac{Pr[m(e_{ij})=open]}{Pr[m(e_{ji})=open]}$, $w^i_j = \frac{w(i)}{w(j)}$, $a^i_j = \frac{a(i)}{a(j)}$, and $l^i_j = \frac{l(e_{ij})}{l(e_{ji})}$, in which $l : E \to \mathbb{R}^+$ is the length of a gate along a region boundary, $a : V \to \mathbb{R}^+$ is the area of a region, $w : V \to \mathbb{R}^+$ is the speed of a body in a region, and $m : E \to \{open, closed\}$ is the configuration of the gate $e = (i, j)$, in which $open$ indicates that the gate allows bodies from region i to j and $closed$ not.

[2]Note that because the ratios of values are important and not the values themselves, transpose-pairs of Q can be scaled by arbitrary factors. The algorithm presented in Section IV only uses the relative ratios, so such scaling has no effect on the end control strategy.

[3]Because the body's motion inside a region is deterministic over small intervals of time (up to bouncing off region boundaries and other bodies), for t' near t, this is not the case. As time progresses, the body's location and direction undergo topological mixing and thus if they are sampled at sufficiently large intervals they appear to be drawn from uniform random variables.

Proposition 2: **The ratio of the rates is**

$$\frac{R_{ij}}{R_{ji}} = m^i_j w^i_j a^j_i l^i_j$$

Proof:
We start with the definition of the rate, and observe that all transitions must occur through gates, when they are open.

$$\begin{aligned} R_{ij} &= Rate[b \text{ transitions from } i \text{ to } j] \\ &= Rate[b \text{ transitions through } e_{ij}] \\ &= Pr[m(e) = open] Rate[b \text{ hits } e_{ij}] \end{aligned}$$

where $Rate[]$ is the rate of an event per unit time, as the unit time approaches zero. Since gates occupy portions of region boundaries, we get

$$Rate[b \text{ hits } e_{ij}] = Rate[b \text{ hits boundary}] \tfrac{l(e_{ij})}{bnd(i)}.$$

The rate at which bodies hit the boundary is proportional to their speed and the boundary length, and inversely proportional to the region size, thus yielding

$$Rate[b \text{ hits boundary}] = C \tfrac{w(i) bnd(i)}{a(i)},$$

where $bnd : V \to \mathbb{R}^+$ is the length of a region's boundary, and $C \in \mathbb{R}$ is a constant factor. Thus we get

$$R_{ij} = C Pr[m(e_{ij}) = open] \frac{w(i) l(e_{ij})}{a(i)}$$

and finally as a ratio

$$\begin{aligned} \frac{R_{ij}}{R_{ji}} &= \frac{Pr[m(e_{ij}) = open]}{Pr[m(e_{ji}) = open]} \frac{w(i)}{w(j)} \frac{a(j)}{a(i)} \frac{l(e_{ij})}{l(e_{ji})} \\ \frac{R_{ij}}{R_{ji}} &= m^i_j w^i_j a^j_i l^i_j \end{aligned}$$

■

Given that the limiting distribution will be the goal distribution so long as the ratio $\frac{R_{ij}}{R_{ji}} = \frac{Q_{ij}}{Q_{ji}}$ is preserved by the implementation of γ onto G (see Proof III), the control strategy can be implemented by adjusting $\frac{Pr(m(e_{ij})=open)}{Pr(m(e_{ji})=open)}$, $\frac{w(i)}{w(j)}$, $\frac{a(j)}{a(i)}$, or $\frac{l(e_{ij})}{l(e_{ji})}$. Combinations of these factors can also be used (see Fig. 7), but for simplicity we do not consider them here.

B. Control Policy

Construction of a control policy assumes a Q, and knowledge of three of the following: area (a), gate configuration switching (m), speed (w), and gate length (l). For area and speed control to be effective alone, detailed balance must hold, for example, as when all gates are symmetric with respect to their lengths and configurations.

1) Area: **Area:** The first way to control the distribution is by modifying each region's area, such that for region i, the area is $\gamma_{area}(i) := \gamma_{area}(j) Q_j^i l_j^i w_j^i m_j^i$, for some adjacent region j, the first such region being given any constant value. The order of this control policy evaluation is any brepth-first search of the environment. If all gates are identical, this simplifies to $\gamma_{area}(i) := \mu_i w(i)$.

2) **Length:** The next means of control is by manipulating the lengths of gates across the region boundaries, in which the length of the sum of gates $e = (i, j)$ is $\gamma_{length}(i, j) := Q_j^i m_i^j a_i^j w_i^j$.

3) **Speed:** The third means of control is over the speed of the bodies in each region, in which for region i, the speed is $\gamma_{speed}(i) := \gamma_{speed}(j) Q_i^j l_j^i w_j^i m_j^i$, for some adjacent region j, the first such region being given any constant value. The order of this control policy evaluation is any brepth-first search of the environment. If all gates are identical, this simplifies to $\gamma_{speed}(i) := w(i)/\mu_i$.

Speed control can be implemented in mobile robots by varying the motor power. In micro- or nano-robots that move across voltage differentials, speed control can be implemented by varying voltage.

4) **Configuration Switching:** The last of the means of control is over the probability that a gate from region i to region j is configured to be open, $\gamma_{time}(i, j) := \frac{\gamma'(i,j)}{\gamma'(i,j)+\gamma'(j,i)}$ proportion of the time, in which $\gamma'(i, j) := Q_j^i a_i^j w_i^j l_i^j$

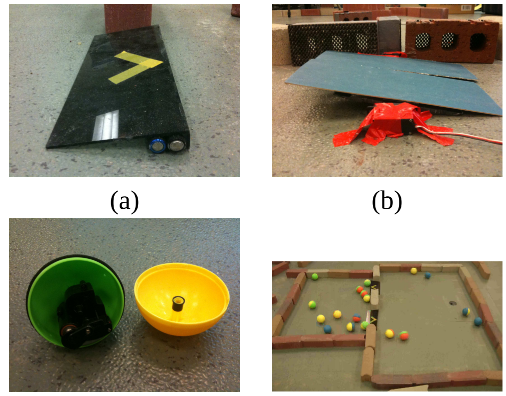
Gate configuration control can be implemented by opening and closing gates (e.g. a door).

These strategies can all be scaled by a constant and remain correct, although the constant will effect the speed at which the distribution of bodies approaches the goal. Calculating time and length control are $O(n^2)$ in time, in which n is the number of regions, and calculating area and speed control can be done in $O(n)$ time.

C. Experimental Testing

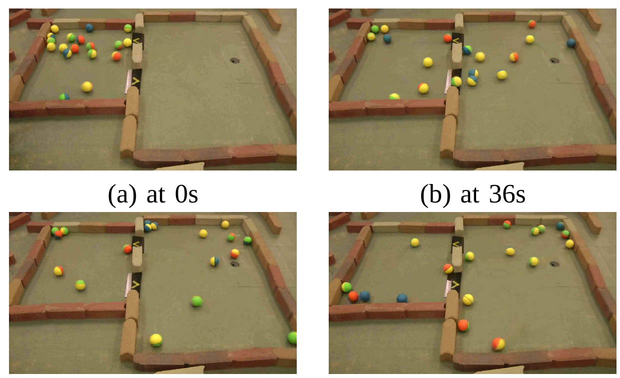
Experiments are used to demonstrate the strategy's effectiveness, using both a physical implementation as well as in a simulator. The physical implementation consists of Weasel Balls, a cheap ($4) toy for children and pets, inside an environment whose regions are constructed out of low brick walls and whose gates are either single-configuration static gates whose configuration is *open*, or controllable two-way gates whose configuration is either *open* or *closed*. See Fig. 3. These mechanisms are similar to the ones presented in [2]. The simulation consisted of polygonal regions in $\mathbb{R}^2$ and bodies that implement a simple approximation of Weasel Balls' motion; they travel along straight trajectories and bounce at random angles off of region boundaries [13]. See Fig. 7.

In Figs. 4, 5, and 6, we demonstrate achieving a distribution of $\mu = (1/3, 2/3)$ across two regions by means of



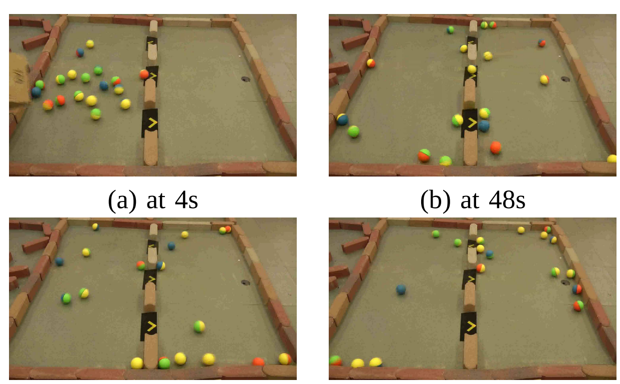
(a) (b) (c) (d)

Fig. 3. (a) A static, one-way gate; (b) A dynamic, two-way gate; (c) The inside of a Weasel Ball, which is 10cm in diameter; (d) A physical environment with bodies. The walls of the environment are composed of $9 \times 9 \times 28$cm bricks.



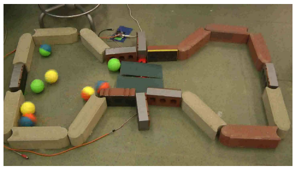
(a) at 0s (b) at 36s (c) at 102s (d) at 204s

Fig. 4. Control by means of area in a physical implementation. One region is twice as large as the other, and are connected by two equally large gates. Initially in (a) all bodies begin in the smaller region. By (b) they are distributed half-and-half, by (c) they have reached the goal distribution and it is maintained in (d).



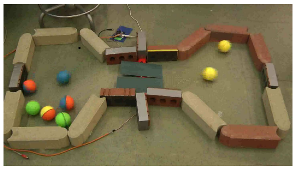
(a) at 4s (b) at 48s (c) at 224s (d) at 404s

Fig. 5. Control by means of gate length in a physical implementation. Two gates of the same length are pointing in one direction, and a single gate of the same size is pointing in the other direction. Initially in (a) all bodies begin in the left region. Again, by (b) they are distributed half-and-half, by (c) it has reached the goal distribution and it is maintained in (d).

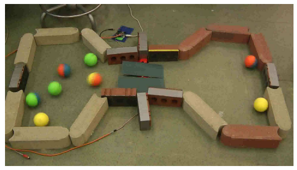


(a) at 0s

(b) at 18s

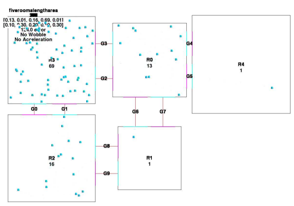


(c) at 30s

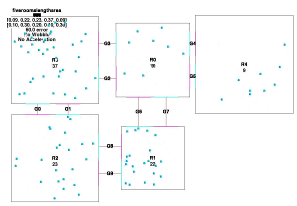


(d) at 133s

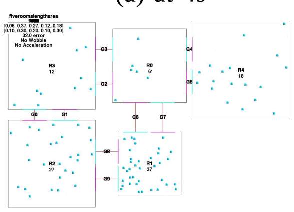
Fig. 6. Control by means of the probability that a gate is in an accepting configuration through time, in a physical implementation. The gate alternates direction randomly such that the ratio of probabilities is the desired ratio.



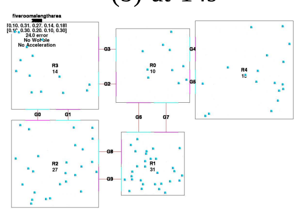
(a) at 4s



(b) at 14s



(c) at 33s



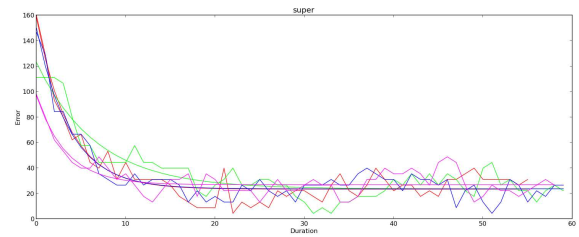
(d) at 49s

Fig. 7. Control by means of the speed of 100 bodies, in simulation. The control strategy here has to control for varying region sizes in addition to the goal distribution. Across (a), (b), (c), and (d) there is a decreasing error (L-2 distance between current distribution and goal distribution) of 1.24, 0.60, 0.32, and 0.24 respectively.

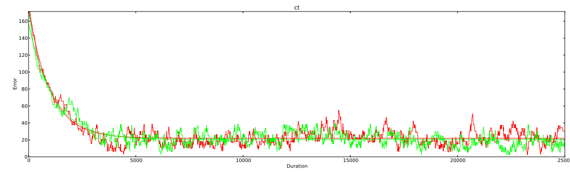
controlling region area, gate length, and the probability that a gate is accepting, respectively. In Fig. 7, we control the distribution across five regions using a combination of area and speed control. Four physical experiments were also run in a larger environment with 45 bodies; one of which is shown in Fig. 1. Additional experiments were performed on a wide variety of environments, starting distributions, and goal distributions (see Fig. 8). Videos of these experiments and more can be found at: *http://users.cis.fiu.edu/∼jabobadi/sc/*.

D. Transition Sequence for Goal Distributions with Zeros

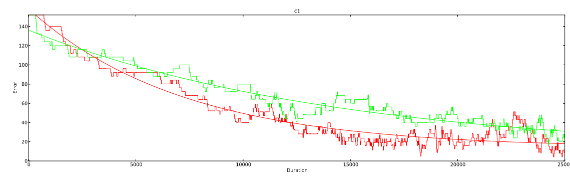
In Section III, we specified that μ must be positive. Consider the case of the goal distribution $\mu' = (0, .3, .2, .1, .4)$ for the environment in Fig. 2. Our control policy construction would divide by 0, and is thus undefined. In some cases (e.g. for the distribution $\mu = (0.25, 0.25, 0.25, 0.25, 0)$) we can get around that problem by approximating $\forall c \in \mathbb{R}^+, c/0 := \infty$,



(a)



(b)



(c)

Fig. 8. Graphs of the L-2 distance between the actual distribution and the goal distribution as a function of time. Different colors represent different experiments. (a) Convergence starting from each region (one of which is in Fig. 1) of a physical experiment; (b) Convergence in simulation to a distribution proportional to their area; (c) Convergence in simulation to a distribution inversely proportional to their area

which would correspond to a scenario in which bodies only move out of regions with limiting distribution 0 and not ever into them. However this does not work in the case of our original μ', since it would result in G's transition matrix ceasing to be an ergodic chain. The limiting distribution of a non-ergodic Markov Chain is a function of the initial distribution. Consider how an initial distribution $\mu_0 = (0, 0, 0, 0, 1)$ would result in the limiting distribution $\mu^* = \mu_0$, whereas any initial distribution $\mu_0 = (0, a_1, a_2, a_3, .4), \sum_i a_i = .6$ would result in the limiting distribution $\mu^* = \mu'$, our goal.

Though Q cannot be constructed for impossible goal distributions μ', Qs can be constructed for an infinite sequence of possible, non-negative goal distributions $\mu_0, \mu_1, \mu_2, ..., \lim_{i \to \infty} \mu_i = \mu'$. This subsequent sequence of transition matrices $Q_0, Q_1, Q_2, ...$ converges to Q and can be used to implement the otherwise unreachable control policy.

The following practical heuristic is provided for the initial element of the sequence μ_0: the best μ_0 is the initial distribution of the bodies if known, since convergence to μ_0 would then be instantaneous, or the distribution corresponding to a random walk on G, because it has a low $SLEM$ and is easy to implement.

V. Finding Probabilistic Bounds on the Control Strategy using Model Checking

In this section, we use a temporal logic and a model checker to present techniques for verifying the limiting distribution, calculating the average expected error, and comparing the mixing rates of systems under the control of the policy we presented. Additionally, we present an analytical technique for quantifying the difference between a body under wild motion and an ergodic body. We are motivated by ongoing work uses temporal logic in high-level control for motion planning problems [18]–[21], [23], [24], [31]. In our work, we found *continuous stochastic logic* (CSL) to be suitable for our needs.

The syntax for CSL is

$$\Phi = \textbf{true} \mid a \mid \neg\Phi \mid \Phi \wedge \Phi \mid \mathbf{P}_{\sim p}[\phi] \mid \mathbf{S}_{\sim p}[\Phi]$$
$$\phi = \bigcirc\Phi \mid \Phi\mathcal{U}^I\Phi$$

in which a is a proposition, $\sim \in \{<, \leq, >, \geq\}$, $p \in [0,1]$ is a probability, and I is an interval of time on $[0,\infty)$. In this syntax, Φ are state formulas, which are assertions about the model state, and ϕ are path formulas, which are temporal assertions about the sequence of states that the model may take. The operator $\bigcirc$ is *next*, and is satisfied if in the next state the formula will be true, and the operator $\mathcal{U}$ is satisfied if the first state formula will be true *until* the second one is true. From these all of the common operators and propositions, such as $\vee$ and **false**, can be derived, and additionally temporal operators such as *eventually*: $\Diamond\Phi = \textbf{true}\ \mathcal{U}^{[0,\infty)}\Phi$ can be defined. The syntax $\mathbf{P}_{\sim p}[\phi]$ asserts that the probability of the path satisfying ϕ is $\sim p$. The syntax $\mathbf{S}_{\sim p}[\Phi]$ asserts that the limiting probability of Φ satisfies $\sim p$. A complete and detailed definition, syntax, semantics, and examples for CSL can be found in [26]. The reader is encouraged to consult [26] for more information.

We used the probabilistic temporal model checker PRISM, described at *http://www.prismmodelchecker.org/* [22]. This program takes a problem formulation and a list of propositions to be checked, all formulated in an application specific grammar, and evaluates the propositions efficiently. Our model's state is the space of all distributions; it consists of a variable for each region whose value indicates the proportion of bodies in that region. Our model's transition scheme consists of a rate between two states i and j proportional to R_{ij} and the number of bodies in i.

A. Steady State Verification

We can use PRISM to verify properties about our system, such as the limiting distribution. The limiting distribution for region i in a single body model can be found by evaluating the formula $\mathbf{S}[\mu_i = 1]$.[4] This is the proposition that, in the steady state, the body can be found in that region. The results from PRISM corroborate the proof provided in section IV. These calculations are fast too, verifying for the environment in Fig. 1 in 0.005s on an Ubuntu 12.04 LTS install on a Intel Core 2 Duo CPU with 4GB memory.

[4]There are not probabilistic bounds on this or other formulas here because $\mathbf{S}[\mu_i = 1] = \mu_i^*$ s.t. $\mathbf{S}_{\leq \mu_i^*}[\mu_i = 1]$ and $\mathbf{S}_{\geq \mu_i^*}[\mu_i = 1]$. Using PRISM, this can be calculated without a search.

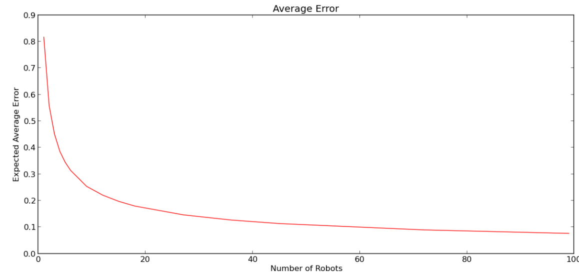


Fig. 9. The L-2 distance between the expected limiting distribution and the goal distribution as a function of the number of bodies in the environment, for the environment in Fig. 1. Derived using the CSL model checker PRISM.

B. Average Error

The proofs and verification provided thus far show that the expected distribution under this control strategy is the goal distribution. At any given time, the actual distribution is probably not the goal one (and for some goals, cannot be). We call this difference the *error*, and is measured as the L-2 norm distance between μ and μ^*: $\langle \mu, \mu^* \rangle_2$. The average error can be calculated as

$$\sum_{\mu} Pr(\mu)\langle \mu, \mu^* \rangle_2$$

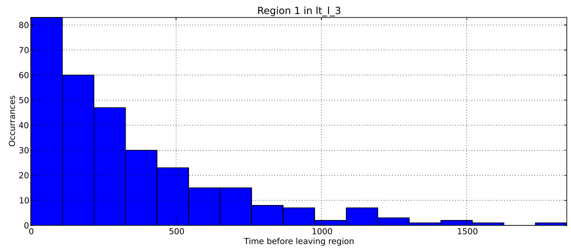
This computation can be expensive and complicated, especially the enumeration of all possible states and their probabilities. Fortunately PRISM can again be used to simplify this process. PRISM allows costs to be assigned to states that meet any proposition, and the cost is accumulated over the time the system spends in the state. Assigning a cost of the L-2 norm distance allows the expected distance to be measured. Experimental analysis shows that the error tends to zero as the number of bodies tends towards infinity.[5]

Consider the environment shown in Fig. 1, whose empirically observed error appears in Fig. 8 (a), and whose expected error in the limiting distribution is shown in Fig. 9. The model checker shows that for that environment and the number of bodies used (45), the average error of the limiting distribution is 11.35%. The observed average error is 25%; thus we can conclude that the discrepancies between our model and the physical system accounts for 56% of the observed error. This technique is generalized and can be applied to analyze how closely any body's motion approximates ergodicity.

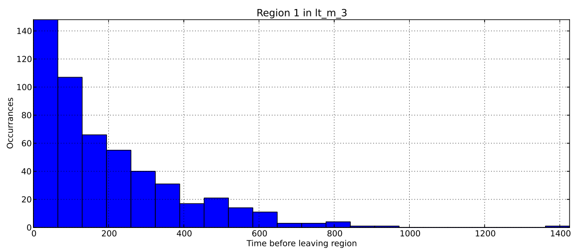
C. Mixing Rates

In Section III, the Second Largest Eigenvalue Modulus (SLEM) is introduced as an analytical method for comparing the mixing rate of two transition matrices. Using PRISM's rewards-based properties and the reward in Sec. V-B, the

[5]Expected average error is invariant to the connectivity of the environment, but does vary based on the goal distribution, the number of agents, and the number of regions.



(a)



(b)

Fig. 10. Histograms of the time a body spent in two regions in simulator. Region (a) is 50% larger than region (b). These clearly demonstrate that the time spent in a region can be approximately modeled as an exponential distribution whose parameter is proportional to the region's area. Such a distribution exists for each gate $(i, j) \in E$ for the time spent in i before transitioning to j. The parameter of this distribution is the rate R_{ij}.

same can be done with PRISM. The cumulative error of a system can be calculated, as can be the instantaneous error with PRISM. Since these rewards are monotone decreasing, we can compare the mixing rate of different graph structures (in which the goal distribution is kept constant), and the mixing rate of different starting distributions (in which the topologies are kept constant), by calculating either of these measures for a given constant time. This comparison can then be used to conduct a search of the region connectivity space or initial distribution space to find the fastest-mixing systems. This method is more powerful than using the SLEM because it is sensitive to initial distributions.

VI. Discussion and Conclusions

In this work we present a methodology to control a distribution of bodies by controlling the variables for the 1) area of the regions, 2) length of the gates between regions, 3) speed of the bodies in the regions, and 4) the probability that a gate between two regions is open.

The results from the physical and simulated experiments show that this control strategy is correct and efficient with respect to the number of regions and number bodies being subject to control. All of the factors subject to control: the area of regions, the speed of bodies within the regions, the sizes of the gates, and the likelihood that a body can transition through a gate, are shown to be relevant factors, and can be ignored only where they are constant across the environment. Other factors may exist that could influence the effect of the control strategy; however any such factors must have been unintentionally held constant across all experiments we conducted. Additionally, our core assumption that the motion of wild bodies in such an environment can be modeled by a CTMC is supported by experimental evidence (see Fig. 10). This strategy is general enough that it can be applied to any environment that meets the assumptions laid out in Sec. II, and any goal distribution. Furthermore, the model checking techniques in Sec. V can be used to analyze any strategy.

A. Significance

The significance of these results and this control strategy can be limited by the assumptions made. The assumption about the *wildness* of the robotic motion turns out to be rather robust; in Fig. 10 we see that our simulated bodies transition according to an exponential distribution. We conducted additional experiments, in which we manipulated the quality of the body's motion in simulation by adding varying degrees of systematic wobble, skewed direction of travel, and acceleration after inelastic collisions. In all cases the exponential distribution was preserved, albeit with a scaled parameter.

On the other hand, the *a priori* knowledge requirements about the environment, as well as the necessary control over the environment to implement some of the strategies, can limit certain applications. The power in our approach is that it can be applied to a wide variety of problems; the strategy is ignorant to the number of bodies being controlled, ignorant to their state (region, location, speed, etc), and works in any connected controllable environment.

In practice, indoor environments lend themselves to being easily broken up into regions and gates by rooms and doors respectively. Whereas establishing control by means of area or length modulation may be difficult or impossible in such environments, time control is achievable merely by opening and closing one-way gates, and speed control can be achieved by varying the speed of a robot's motors in each region [3]. For nanorobots that swim according by control of voltage differentials, speed can be adjusted with the voltage. Furthermore, time- and speed-based control strategies also lend themselves to being easily scaled down or up so that the fastest speed or shortest duration are manageable.

B. Future Work

Most foreseeable future work, in keeping with the goals of minimalism set out in Sec. I, would involve reducing the severity of the assumptions the strategy requires. For example, some of the information assumed by our strategy, such as the area of regions and the topology of the environment, could be approximated if gates were capable of communicating with each other as well as distinguishing between the bodies in the environment. The topology can be learned by observing which gates are reachable from each other (if a body visits two gates in sequence, they must share a region). The base rates between regions, taking into account their area, the speed within them, and the size of the gates connecting them, can be calculated by approximating how long an agent spends in a region i on average before transitioning through any given gate to region j. This value corresponds to $a(i)w(i)l(e_{ij})$ when $m_j^i = 1$.

There may be more accurate models at the expense of being more complicated [32]. However we believe that CTMCs capture the phenomenon to a large enough extent to make them useful models. Some future work may involve exploring other modeling options.

Another area of development for future work would be to design robots whose motion best fits the definition of *wild* put forth in Sec. II. This measurable (see Sec. V-B) development would improve mixing rates and decrease average error, and its relation to *ergodicity* would make it an interesting problem in its own right.

VII. Acknowledgements

This work was supported in part by NSF grant 0904501 (IIS Robotics), NSF grant 1035345 (CNS Cyberphysical Systems), DARPA SToMP grant HR0011-05-1-0008, and MURI/ONR grant N00014-09-1-1052.

References

[1] M. Blum and D. Kozen. On the power of the compass (or, why mazes are easier to search than graphs). In *Proceedings Annual Symposium on Foundations of Computer Science*, pages 132–142, 1978.

[2] L. Bobadilla, K. Gossman, and S. M. LaValle. Manipulating ergodic bodies through gentle guidance. In *Proceedings IEEE Conference on Robot Motion and Control*, 2011.

[3] L. Bobadilla, F. Martinez, E. Gobst, K. Gossman, and S. M. LaValle. Controlling wild mobile robots using virtual gates and discrete transitions. In *American Control Conference*, 2012.

[4] L. Bobadilla, O. Sanchez, J. Czarnowski, K. Gossman, and S. M. LaValle. Controlling wild bodies using linear temporal logic. In *Proceedings Robotics: Science and Systems*, 2011.

[5] S. Boyd, P. Diaconis, and L. Xiao. Fastest mixing markov chain on a graph. *SIAM Review*, 46(4):667–689, 2004.

[6] Stephen Boyd. Convex optimization of graph laplacian eigenvalues. In *Proceedings oh the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, pages 1311–1320, 2006.

[7] Pierre Bremaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulations, and Queues*. Springer, 1991.

[8] J. F Canny and K. Y Goldberg. A RISC approach to sensing and manipulation. *Journal of Robotic Systems*, 12(6):351–364, 1995.

[9] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Cambridge, MA, 2005.

[10] B. Donald, J. Jennings, and D. Rus. Minimalism + distribution = supermodularity. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(2-3):293–321, 1997.

[11] M. A. Erdmann. Randomization in robot tasks. *International Journal of Robotics Research*, 11(5):399–436, October 1992.

[12] M. A. Erdmann and M. T. Mason. An exploration of sensorless manipulation. *IEEE Transactions on Robotics & Automation*, 4(4):369–379, August 1988.

[13] K. Gossman. Design of environments to control robotic behavior, 2012.

[14] E. Gutkin. Billiards in polygons. *Physica D*, 19:311–333, 1986.

[15] E. Gutkin. Billiards in polygons: A survey of recent results. *Journal of Statistical Physics*, 83(1/2):675–737, 1996.

[16] Maryam Kamgarpour, Sean Summers, and John Lygeros. Control design for specifications on stochastic hybrid systems. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, HSCC '13, pages 303–312, New York, NY, USA, 2013. ACM.

[17] S. Kerckhoff, H. Masur, and J. Smillie. Ergodicity of billiard flows and quadratic differentials. *Annals of Mathematics*, 124(2):293–311, 1986.

[18] M. Kloetzer and C. Belta. Automatic deployment of distributed teams of robots from temporal logic motion specifications. *IEEE Transactions on Robotics and Automation*, 26(1):48–61, 2010.

[19] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas. Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics and Automation*, 25(6):1370–1381, December 2009.

[20] H. Kress-Gazit, G.E. Fainekos, and G.J. Pappas. Temporal logic motion planning for mobile robots. In *Proceedings IEEE International Conference on Robotics and Automation*, 2005.

[21] H. Kress-Gazit, G.E. Fainekos, and G.J. Pappas. Where's Waldo? sensor-based temporal logic motion planning. In *Proceedings IEEE International Conference on Robotics and Automation*, 2007.

[22] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In G. Gopalakrishnan and S. Qadeer, editors, *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.

[23] M. Lahijanian, J. Wasniewski, S. B. Andersson, and C. Belta. Motion planning and control from temporal logic specifications with probabilistic satisfaction guarantees. In *Proceedings IEEE International Conference on Robotics & Automation*, pages 3227–3232, 2010.

[24] Morteza Lahijanian, Sean B Andersson, and Calin Belta. Temporal logic motion planning and control with probabilistic satisfaction guarantees. *Robotics, IEEE Transactions on*, 28(2):396–409, 2012.

[25] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Also available at http://planning.cs.uiuc.edu/.

[26] G. Norman M. Kwiatkowska and D. Parker. Stochastic model checking. *Lecture Notes in Computer Science (Tutorial Volume)*, 4486:220–270, 2007.

[27] B.J. Nelson, L. Dong, and F. Arai. Micro/nanorobots. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*. Springer-Verlag, 2008.

[28] J. M. O'Kane and S. M. LaValle. On comparing the power of robots. *International Journal of Robotics Research*, 27(1):5–23, 2008.

[29] D. A. Shell, C. V. Jones, and M. J. Mataric. Ergodic dynamics by design: A route to predictable multirobot systems. In *Multi-Robot Systems: From Swarms to Intelligent Automata*, pages 291–297, 2005.

[30] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.

[31] Eric M Wolff, Ufuk Topcu, and Richard M Murray. Robust control of uncertain markov decision processes with temporal logic specifications. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 3372–3379. IEEE, 2012.

[32] A. I. Zeifman. Quasi-ergodicity for non-homogeneous continuous-time markov chains. *Journal of Applied Probability*, 26(3):pp. 643–648, 1989.

[33] A.N. Zemlyakov and A.B. Katok. Topological transitivity of billiards in polygons. *Mathematical notes of the Academy of Sciences of the USSR*, 18(2):760–764, 1975.