

Manipulating Ergodic Bodies Through Gentle Guidance

Leonardo Bobadilla

Katrina Gossman

Steven M. LaValle

bobadill1@uiuc.edu kgossma2@uiuc.edu lavalle@uiuc.edu

Department of Computer Science

University of Illinois

Urbana, IL 61801 USA

Abstract—This paper proposes methods for achieving basic tasks such as navigation, patrolling, herding, and coverage by exploiting the wild motions of very simple bodies in the environment. Bodies move within regions that are connected by gates that enforce specific rules of passage. Common issues such as dynamical system modeling, precise state estimation, and state feedback are avoided. The method is demonstrated in a series of experiments that manipulate the flow of weasel balls (without the weasels) and Hexbug Nano vibrating bugs.

I. INTRODUCTION

In everyday life we see many examples of independently moving bodies that are gracefully corralled into behaving in a prescribed way. For example, when the free breakfast area closes in a hotel, the patron usually locks the door from the outside so that no one else can enter, but people eating are able to finish their meals and leave. This has the effect of clearing everyone from the room without people feeling that they have been tightly controlled or coerced. People install a “doggie door” on their house door to enable pets to move in either one direction or both. In a subway system, turnstiles cause people to flow in prescribed directions to ensure that proper fares are paid. The popular Hexbug Nano toy provides a habitat for simple vibrating bugs that can be channeled through rooms and corridors by reconfiguring static gates.

All of these scenarios, and many others, involve numerous bodies moving together in one environment with two important principles:

- 1) Each body moves independently, possibly with a “mind of its own”, in a way that seems to exhaustively explore its environment.
- 2) The bodies are effectively controlled without precisely measuring their state and without directly actuating them.

We propose a paradigm for developing robotic systems that is inspired by these examples and two common principles. Each body may be a robot, human, animal, or even some simple automaton in the mechanical sense. We want its behavior, however, to seem sufficiently wild, erratic, random, or systematic so that it appears to try

every kind of motion. Ideally, we would like the body to move on a trajectory that is *dense* in the boundary of any bounded region in which it is trapped. This means that it will strike every part of the boundary until eventually, there does not exist any open interval that has not contacted. One way to achieve this is through *ergodic* motion [24].¹ The sufficiently wild trajectory of each body is then exploited to yield effective control strategies that gently guide bodies into desired states, rather than trying to tightly manipulate them. This gives the feeling of herding, corraling, or even encouraging them to behave.

We want to solve tasks such as navigation, patrolling, coverage, herding, and separating into groups. Our approach is to divide the workspace of the bodies into a finite set of *regions*. Bodies are constrained to move within a region, but are able to transition to another region by the use of a *gate*. Sections II, III, and IV introduce, respectively, systems with increasingly sophisticated gates:

- **Static gates:** The gates are fixed in advance and allow one-way motions from region to region.
- **Pliant gates:** The gates have internal modes that affect how bodies are permitted to transition between regions and the modes may passively change via contact with bodies.
- **Controllable gates:** Based on sensor feedback, the gate modes are externally controlled during execution.

The ideas relate closely to many existing approaches to manipulation. Our work generally falls under the category of *nonprehensile manipulation* [12], [21], which includes tilting a tray to align parts [13], squeezing with a parallel-jaw gripper [16], batting at objects [1], pushing boxes [20], manipulation using passive fences [4], [26], self assembly [17], [22], [25], and inducing a knot in a thread by forcing it through a carefully designed block [3]. Field-based approaches to manipulation also closely fit, including MEMS [6] and vibrating plates [5], [23]. In addition,

¹Our notion of ergodicity is derived from dynamical systems analysis, having more to do with dense and uniform coverage of trajectories, rather than its more popular use of probabilistic analysis of Markov chains.

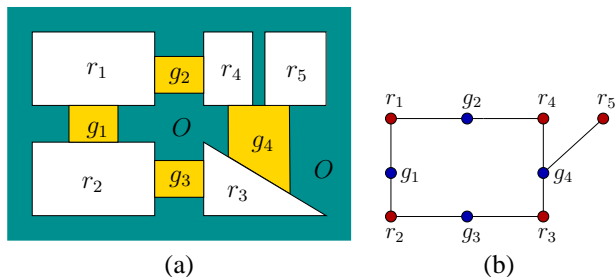


Fig. 1. a) A simple arrangement of five regions and four gates; b) the corresponding bipartite graph.

randomization was shown to help for manipulation in [11], which is similar to the wild motions exploited in our approach.

In addition to manipulation research, our approach has some similarity to *behavior-based robotics* in which teams of agents can achieve complex tasks by composing several simple behaviors [2]; however, in our case we exploit one constant “behavior”: The wild, systematic motions of a body. Even more closely related are designing *virtual fences* to control herds of cows [8] and designing fire evacuation strategies to safely “herd” humans out of a burning building [9].

Since our gates have discrete modes and the bodies transition between continuous regions, there are natural connections to hybrid systems and their use in the design and control of multi-robot systems (e.g., [10], [14], [15]).

In math and physics, there is extensive literature on ergodic systems. One of the most relevant branches here is *dynamical billiards* [24], in which conditions are determined under which a ball that “rolls forever” will fill the entire space and achieve other uniformity properties (a famous example is the Bunimovich stadium, which looks like a hockey rink in which the puck ricochets and travels forever).

II. STATIC GATES

Consider a planar workspace \mathbb{R}^2 that is partitioned into an obstacle set O and a finite set of bounded cells with connected open interior, each of which is either a *region* or a *gate*; Figure 1 shows a simple example. The following conditions are imposed: 1) No region shares a boundary with any other region, 2) No gate shares a boundary with any other gate; 3) Every region shares a boundary with at least one gate; 4) If a gate and a region share a boundary, then the boundary is a connected interval (rather than being a point or being disconnected). Let R denote the set of all regions and G denote the set of all gates. The union of all $r \in R$, all $g \in G$, and O yields \mathbb{R}^2 .

Now place a *body* b into the workspace. The body is assumed to be “small” with respect to the sizes of

regions, gates, and their shared boundaries. It is therefore modeled geometrically as a point even though it may have complicated kinematics and dynamics. For any region $r \in R$, it is assumed that b moves on a trajectory that is dense on the boundary of r . One sufficient condition to achieve this is *ergodicity* (see the appendix for a formal definition and example), which implies that from any initial state, no open subset of its state space, confined to the region r , will be unvisited. We will exploit this property to ensure that b repeatedly strikes every open interval in ∂r (the boundary of r), from various directions.

The precise locations or configurations of b do not need to be measured in our work. The only information that we will use for design and analysis is which region or gate contains it. Therefore, we can immediately obtain a combinatorial description of the states and state transitions.

We define a bipartite graph \mathcal{G} with set of vertices $V = R \cup G$. The edge set E corresponds to every region-gate pair that shares a boundary. Due to the constraints on regions and gates, note that \mathcal{G} is bipartite: There are no edges between regions and none between gates.

Each gate has the ability to manipulate the body b , sending it from one of the gate’s adjacent regions to another. The simplest gate behavior is assumed here; more sophisticated gates are introduced in later sections. Let each pair (r, g) of adjacent regions and gates be called an *interface*. For each interface, a *direction* is associated: 1) *incoming*, which means that a body approaching the common boundary from r is sent into g ; however, it cannot enter r from g . 2) *outgoing*, which means that a body approaching the common boundary from g is sent into r . The gates are considered *static*, meaning that once their directions are set they cannot be changed during execution.

Now we want to use the setup to solve a task. Suppose that an environment is given with regions, gates, and a body. We have the ability to set the direction of every interface to force a desirable behavior. What can be accomplished? In this section, we demonstrate two tasks: 1) Navigation to a specified region and 2) traveling along a cyclic patrolling route without termination.

To develop an implementation, it would be ideal to design bodies that have ergodic behavior. This is an interesting direction for future research; here, we instead borrowed some existing low-cost mechanical systems that have very similar properties. For most experiments, we used a *weasel ball*, pictured in Figure 2. This inexpensive (around \$5 US) toy consists of a plastic ball of radius 8.5cm that has only a single offset motor inside that oscillates at about 2Hz. There is no other circuitry: Only a battery and motor. It was designed to roll wildly while appearing to chase a small stuffed weasel toy. After

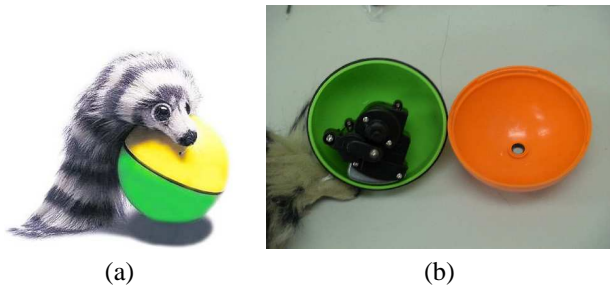


Fig. 2. a) A weasel ball serves as a wildly behaving body; b) it consists entirely of a battery and slowly oscillating motor mounted to a plastic shell.

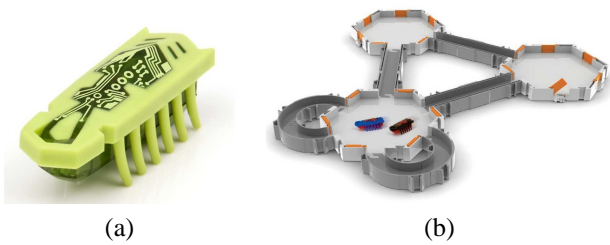


Fig. 3. a) The vibrating Hexbug Nano toy also has wild motion properties and is useful for our experiments; b) it in fact comes equipped with a “habitat” that it nicely explores.

removing the weasel, we discovered in experiments that its exploration properties are remarkably systematic. For other experiments, we used the *Hexbug Nano* (Figure 3), which is a cheap (around \$10 US) vibrating toy that looks like the end of a toothbrush with rubberized bristles with a vibrating motor mounted on top (an earlier YouTube video demonstrated precisely this). This highly popular toy has been demonstrated to explore complex habitats with regions and gates, which can be purchased. In this case, the gates are static and can be made either *closed* or *bidirectional*.

We place these bodies into planar environments for which obstacles are formed using bricks and cinder



Fig. 4. a) A static, directional gate can be implemented making a flexible “door” from a stack of paper; in this case, the body can transition only from the bottom region to the top; b) this works much like a “doggie door”.

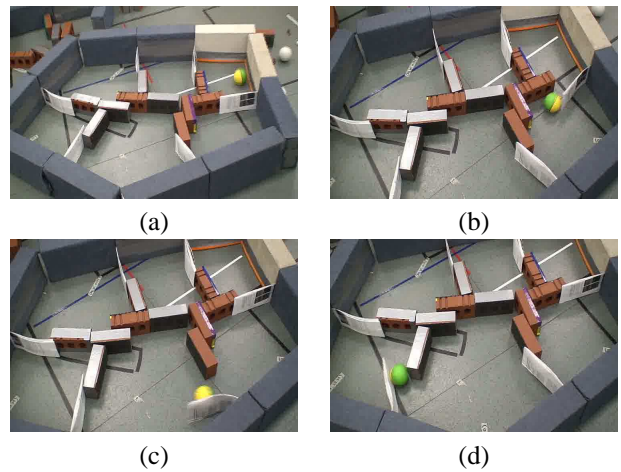


Fig. 5. A basic navigation experiment: a) The weasel ball is placed initially in the upper right corner; b) after 25 seconds it changes regions; c) after 32 it changes again; d) after 45 seconds it reaches its destination.

blocks. The only part remaining is to design directional gates. A simple way to achieve this is illustrated in Figure 4(a). A body moving from the bottom region to the top region can pass through the right side by bending the paper; a body in the other direction is blocked. This simple setup proved to reliably implement the directional gate in numerous experiments. Using these experimental components, we solve the tasks below.

TASK 1: Navigation to a specified region. The task here is to force the body to a particular region, r_{goal} , regardless of where it starts. This can be accomplished by following the classical “funneling” approach whereby in each vertex (region or gate) the body is forced into another vertex that brings it closer to the goal [7], [19], [21]. We simply set all of the interfaces so that all discrete flows in \mathcal{G} lead into r_{goal} . This can be computed in time linear in the size of \mathcal{G} by a simple wavefront propagation algorithm (see Chapter 8 of [18]). Alternatively, Dijkstra’s algorithm can be used. These result in a discrete navigation function in which the cost-to-go decreases from vertex to vertex until the goal is reached. The interface edge directions are then set to “point” from the higher-cost vertex to the lower-cost vertex.

We implemented the navigation approach for a weasel ball in an environment of approximately 2 by 3 meters and six gates; see Figure 5. The gate directions were set so that the body is led from the region in the upper right to the lower left. It was allowed to achieve this by traveling along the top chain of regions or along the bottom; in this particular run it chose the bottom. The videos for this execution and all others in this paper can be found at:

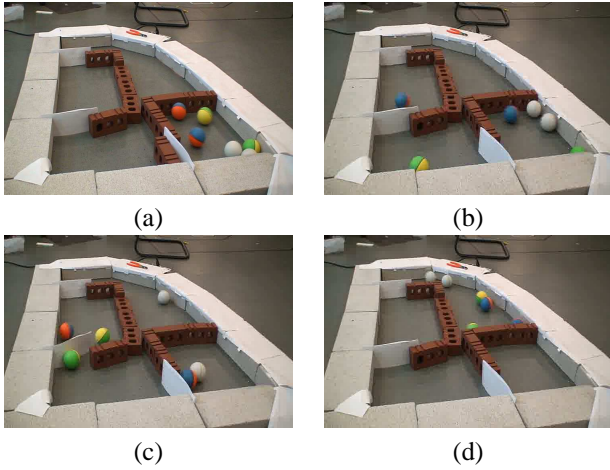


Fig. 6. Navigation with multiple balls: a) Six weasel balls are started in the lower right; b) after 20 seconds some progress is made; c) after 45 seconds one ball has arrived in the destination region in the upper right; after 120 seconds, all six balls have arrived.

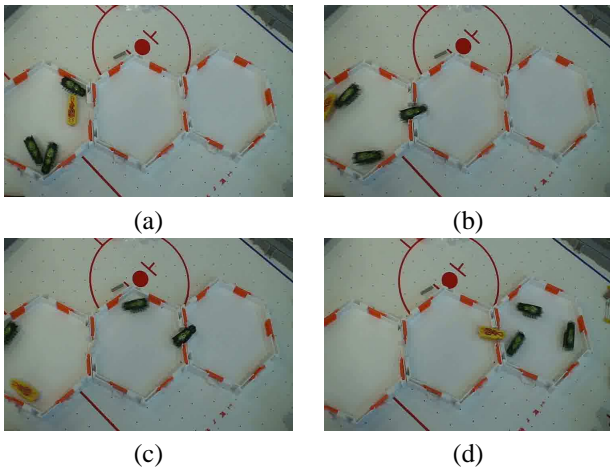


Fig. 7. Navigation of Nanos: a) Initially, all the four Nanos are together in the left region; b) after 10 seconds one Nano changes regions; c) after 17 seconds one Nano crosses from the second region to the third; d) after 70 seconds all Nanos are in the third region.

<http://msl.cs.uiuc.edu/ergodic/2010/>

We then tried the method for multiple bodies moving together. This should work provided that the bodies do not interfere with each others' systematic motions and they do not interfere with gate operation. We designed an environment with three gates and six weasel balls (see Figure 6) and set up the gate configuration to lead the bodies from the bottom right to the upper right.

To illustrate the navigation task on another platform, four Hexbug Nanos were placed in an environment with three regions; see Figure 7.

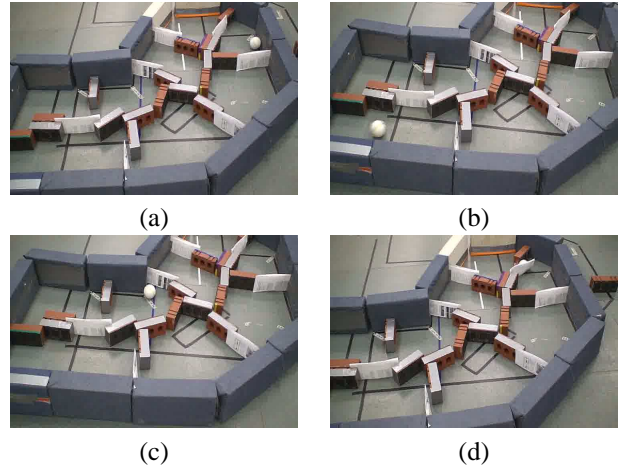


Fig. 8. A patrolling example: a) The ball starts its route; b) after 40 seconds it has entered three new regions; c) after 50 seconds it has visited most regions; d) after 120 seconds, it completes a tour of all regions, and continues.

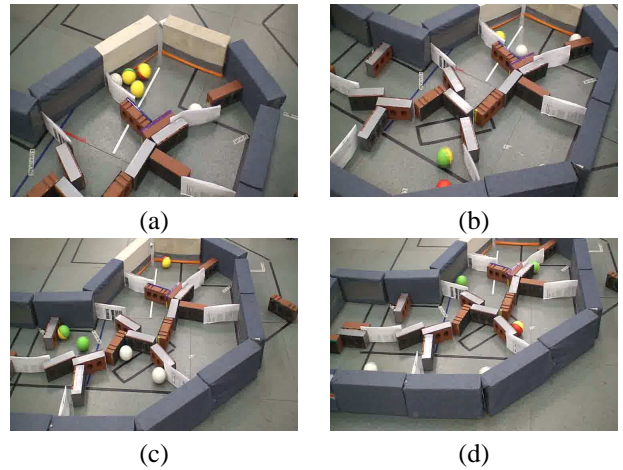


Fig. 9. Patrolling with multiple bodies: a) Five balls are initially in one region; b) 20 seconds later, two balls have advanced two regions; c) after 105 seconds one ball has already returned to the starting region; d) after 165 seconds several have returned, and the patrolling continues.

TASK 2: A cyclic patrolling route: As opposed to reaching a particular goal, various forms of patrolling may be performed. In this case, the directions are set in \mathcal{G} to make directed cycles that the bodies traverse without termination.

We implemented patrolling by creating an environment with seven regions, seven gates, and one weasel ball; see Figure 8. We also implemented patrolling for five weasel balls in the same environment; see Figure 9.

III. PLIANT GATES

In Section II, every gate was configured at the outset and remained fixed during execution. In this section, we



Fig. 10. A gate with two modes: a) a ball can pass from left to right, but its blocked the other way; b) a ball can only pass from right to left.

allow *pliant* gates, which change their behavior while interacting with bodies, thereby having their own internal state. These changes are caused entirely by the motions of bodies, rather than being induced from some external forces. The gates passively configure themselves based on interactions with bodies.

A wide variety of mechanisms can achieve this, leaving many interesting, open design issues. Each gate g has an associated discrete mode space $M(g)$. Based on its mode $m \in M(g)$, transitions between its neighboring regions are allowed or prohibited. In terms of \mathcal{G} from Section II, imagine that the edge directions for all neighboring edges of g are determined as a function of m . Furthermore, a *mode transition equation* is made for each gate: $m' = f(m, r)$, which indicates that the gate enters mode m' , when it starts in mode m and a body enters from region r .

TASK 3: Maintaining Fixed Coverage: To illustrate this principle, we designed the gate shown in Figure 10. An “L” shaped door is attached to hinges on a vertical post. It can rotate 90 degrees back and forth, depending on forces from the body. The gate has two modes. In one mode, the body is allowed to pass from left to right, but is blocked in the other direction. In the other other mode, it instead permitted from right to left, but blocked from left to right. Furthermore, when a body passes through the gate in either direction, its mode is forced to change.

For a single body, this has the effect of allowing it to double back on its course. If there are multiple bodies, then the gate keeps the number of bodies per adjacent room roughly constant. Two bodies are not allowed to pass sequentially through the gate; the second one must wait for a body to pass in the other direction. Figure 11 shows an experiment that illustrates this for five weasel balls.

We also designed and implemented a four-way revolving door, which is a pliant gate that has four adjacent regions. It is only allowed to rotate up to 90 degrees and alternates between two modes: 1) Allowing a clockwise transfer and 2) allowing a counterclockwise transfer. An

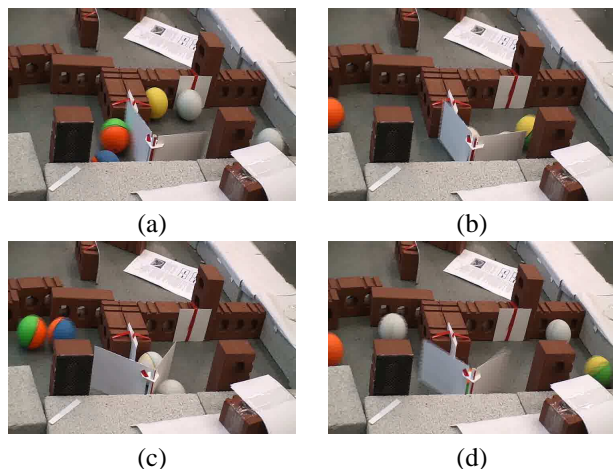


Fig. 11. a) Initially, two balls are in the right region and three are in the left; the gate mode allows a right to left transition; b) after 18 seconds a body crosses right to left, changing the gate mode; c) after 40 seconds a body moves left to right, changing the gate mode again; d) number of bodies in each region alternates between two and three for the rest of the experiment.

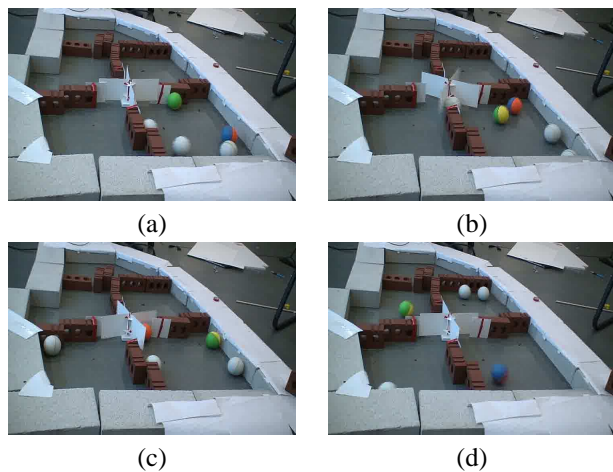


Fig. 12. a) Initially, the five balls are together in one region and only clockwise transfers are allowed; b) after 20 seconds a ball changes regions and counterclockwise transfers are allowed; c) 5 seconds later a ball changes regions; d) after 92 seconds, the balls occupy all four regions

experiment with five weasel balls is illustrated in Figure 12.

IV. CONTROLLABLE GATES

In this section, we propose actuating the gates as opposed to passively allowing gate modes to change as in Section III. We now have the ability to set the mode at will during execution; however, a crucial issue arises:

What information will be used during execution to change gate modes?

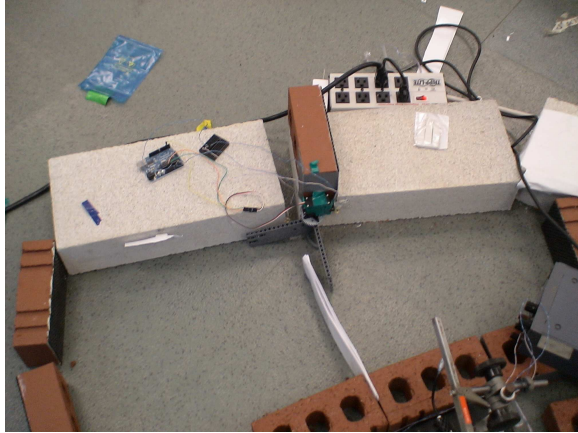


Fig. 13. The controllable door is designed with a L-shape connected to a servo motor that is commanded by a microcontroller. Two photo diodes are connected to each side of the gate to detect the direction of crossing

Let M be the composite mode space, obtained as the $|G|$ -fold Cartesian product of $M(g)$ for every $g \in G$. A plan or control law can generally be expressed as a mapping $\pi : \mathcal{I} \rightarrow M$, in which \mathcal{I} is an *information space* that takes into account actuation histories and sensor observation histories (see Chapter 11 of [18]).

A common approach is to let \mathcal{I} represent the full state space and design powerful sensors and filters to estimate the state (all gate and body configurations) at all times. We instead take a minimalist approach and control the gates using as little sensing information as possible.

One approach is to simply time the switching of the gates. Let $T = [0, t]$ be the execution time interval. We let $\mathcal{I} = T$ and a plan is represented as $\pi : T \rightarrow M$. In this case, the expected behavior of the body or bodies needs to be carefully measured so that the gates are switched at the best times. Without no other feedback, however, this approach is limited.

The shortcomings of time feedback motivate the introduction of simple sensors that can detect whether one or more bodies has crossed part of a region or has traveled through a gate. This allows strategies based on *sensor feedback*. In this case, let Y denote the set of all sensor outputs. A plan is expressed as $\pi : Y \rightarrow M$. More complex plans are possible by reasoning about sensing and actuation *histories*; however, this is left for future work.

We now present some simple experiments with controllable gates; much more is left to do in this direction. Our controllable gate is made from a stack of paper, as the static gates in Section II; however, we additionally have a controllable, “L” shaped part. By rotating the part 90

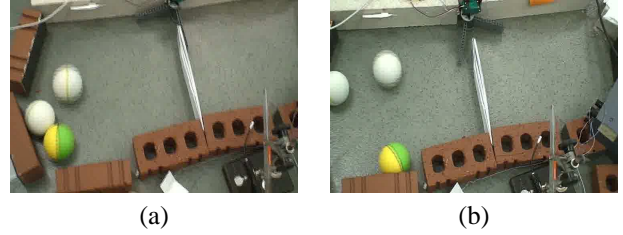


Fig. 14. a) Three balls are trapped in the left region because only right-to-left transfers are allowed. b) After one minute, the gate automatically changes mode and allows the balls to make the left-to-right transition.

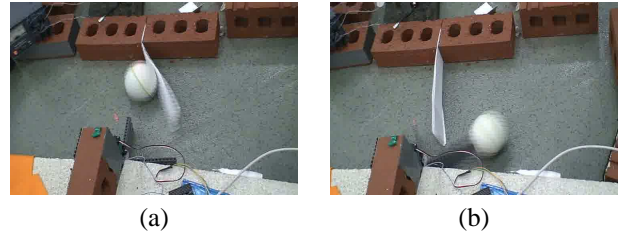


Fig. 15. Sensor beam feedback for one body: a) the weasel ball is placed initially in the left region and then that ball passes through the gate; b) the gate direction is then changed, allowed it to return.

degrees, the direction of the gate is altered. The L-shaped rotating element is made out of Lego pieces attached to a servo motor that is commanded by the an Arduino 8-bit Atmel microcontroller, costing about \$30 US; see Figure 13. The microcontroller has an in built clock that that provides time feedback.

We tested the time feedback gate in an environment of about 2 by 2 meters; see Figure 14. We started with three weasel balls in the left region and the initial gate mode allows movement only to the right region, thereby confining the balls to the left region. We programmed the gate to automatically change its mode every minute, causing the balls to oscillate back and forth.

We then developed some simple sensor beams to provide alternative feedback. An emitter-detector pair was placed on each side of the gate, indicating whether a body passes. The detector is a photodiode (about \$2 US) and the emitter is a cheap laser pointer (about \$3 US).

We show two experiments to illustrate sensor feedback with sensor beams. In Figure 15, one weasel ball is placed in the right region. The gate was programmed to change direction as soon as the pair of beams is crossed. This will keep the body alternating between two regions. In Figure 16, the system was programmed to react to the crossings of two balls: As soon as two balls have crossed, the gate responds by changing directions.

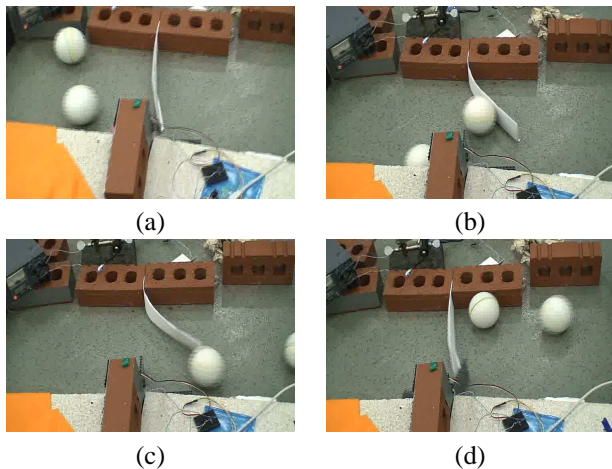


Fig. 16. Using sensor beam feedback for two bodies: a) Initially, two weasel balls are in left region; b) after 15 seconds one ball transitions; c) after 30 seconds the other body follows; d) the sensor detects both crossings and changes the gate mode.

V. CONCLUSION

We developed a manipulation paradigm based on placing wildly moving bodies into a complicated environment and then gently guiding them through gates that can be reconfigured. Several encouraging experiments were shown for weasel balls and Hexbug Nanos performing tasks such as navigation, patrolling, and coverage. Various types of gates were designed, including static, pliant, and controllable with sensor feedback. With the successes so far, it seems we have barely scratched the surface on the set of possible systems that can be developed in this way to solve interesting tasks.

As one example that extends beyond the models of this paper, Figure 17 shows an experiment in which there are two *types* of bodies. There are two regions joined by a special gate that allows one body type to flow in one direction, and the other type to flow in the other direction. All other situations are blocked. When a mixed collection of bodies are placed in both regions, the effect of the gate is to separate the bodies into their respective types over time. In this case, weasel balls end up in one region and Hexbug Nanos end up in the other, regardless of the starting condition.

Numerous other questions and issues remain for future research. What types of manipulations are generally possible? What can be accomplished if gates have the capacity to store balls as part of their mode (a kind of capacitance)? Can simple mechanisms be designed that permit very complex patterns of flow? For example, a gate might allow 3 bodies from left to right, then 1 from right to left, then 2 from left to right, then 4 from right to left, and then repeat. There is a natural connection to formal

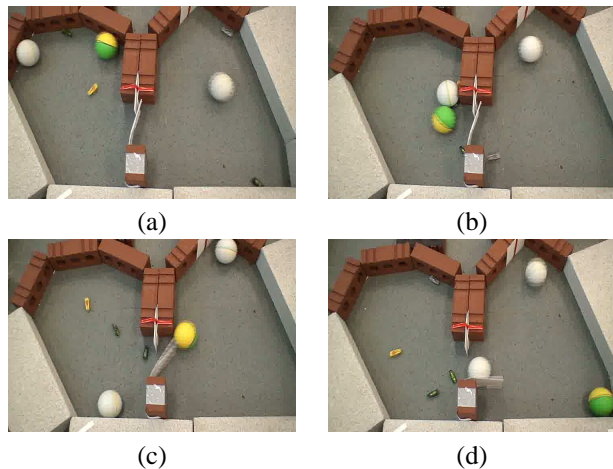


Fig. 17. Separating bodies: a) Two weasel balls and two Nanos are in the left region and one weasel ball and one Nano is in the right one; b) after 12 seconds one Nano changes regions; c) after 15 seconds one weasel ball changes regions; d) after 40 seconds the bodies have been separated.

languages and automata theory, as strings “accepted” by gates are considered. We have also just begun to explore the possibilities for sensor feedback and plans that chose actions based on information states. It will be interesting to see what tasks can be solved through combinations of ergodic bodies, cheap sensors, and simple actuators. As the settings become more complicated, interesting algorithmic questions will also arise. Can effective plans be computed automatically from the task and environment descriptions?

Other possible experimental platforms come to mind. What other simple mechanisms can be designed to yield effective behavior? Which designs lead to more effective overall systems with regions and gates? Can “virtual” gates be designed on simple robots using sensor feedback, rather than relying on bouncing from walls. What other media are possible? Instead of a planar surface, we can imagine manipulating ergodic boats, underwater vessels, and helicopters. Perhaps even insects can be effectively manipulated.

It is also interesting to study asymptotic properties of these systems in various settings to understand the limiting distribution of bodies across the rooms. What limiting distributions can be achieved by simple gate designs? Can the expected running time be calculated for certain combinations of rooms, gates, and body models?

Finally, the experiments appear like a macro-scale version of Maxwell’s demon, which was a thought experiment that violates the Second Law of Thermodynamics. Our approach uses controlled gates to interfere with a natural equilibrium that should otherwise result from

ergodic particles.

Acknowledgments: This work was supported in part by NSF grants 0904501 (IIS Robotics) and 1035345 (Cyberphysical Systems), DARPA SToMP grant HR0011-05-1-0008, and MURI/ONR grant N00014-09-1-1052. The authors thank Noah Cowan for helpful comments.

REFERENCES

- [1] S. Akella, W. H. Huang, K. M. Lynch, and M. T. Mason. Sensorless parts feeding with a one joint robot. In J.-P. Laumond and M. Overmars, editors, *Algorithms for Robotic Motion and Manipulation*, pages 229–237. A.K. Peters, Wellesley, MA, 1997.
- [2] T. Balch and R. C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.
- [3] M. Bell and D. Balkcom. Knot tying with single piece fixtures. In *IEEE International Conference on Robotics and Automation*, page 44294436, 2008.
- [4] R.-P. Berretty, K. Goldberg, M. H. Overmars, and A. F. van der Stappen. On fence design and the complexity of push plans for orienting parts. In *ACM Symposium on Computational Geometry*, pages 21–29, 1997.
- [5] K.-F. Böhringer, V. Bhatt, B. R. Donald, and K. Goldberg. Algorithms for sensorless manipulation using a vibrating surface. *Algorithmica*, 26:389–429, 2000.
- [6] K. F. Bohringer, B. R. Donald, R. Mihailovich, and N. C. MacDonald. Sensorless manipulation using massively parallel microfabricated actuator arrays. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 826–833, 1994.
- [7] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *International Journal of Robotics Research*, 18(6):534–555, 1999.
- [8] Z. Butler, P. Corke, R. Peterson, and D. Rus. Virtual fences for controlling cows. In *IEEE International Conference on Robotics and Automation*, page 44294436, 2004.
- [9] L. G. Chalmet, R. L. Francis, and P. B. Saunders. Network models for building evacuation. *Fire Technology*, 18(1):90113, 1982.
- [10] M. Egerstedt and X. Hu. A hybrid control approach to action coordination for mobile robots. *Automatica*, 38(1):125–130, January 2001.
- [11] M. A. Erdmann. Randomization in robot tasks. *International Journal of Robotics Research*, 11(5):399–436, October 1992.
- [12] M. A. Erdmann. An exploration of nonprehensile two-palm manipulation using two zebra robots. In J.-P. Laumond and M. Overmars, editors, *Algorithms for Robotic Motion and Manipulation*, pages 239–254. A.K. Peters, Wellesley, MA, 1997.
- [13] M. A. Erdmann and M. T. Mason. An exploration of sensorless manipulation. *IEEE Transactions on Robotics & Automation*, 4(4):369–379, August 1988.
- [14] R. Fierro, A. Das, V. Kumar, and J. P. Ostrowski. Hybrid control of formations of robots. In *Proceedings IEEE International Conference on Robotics & Automation*, pages 157–162, 2001.
- [15] E. Frazzoli, M. A. Dahleh, and E. Feron. Robust hybrid control for autonomous vehicles motion planning. Technical Report LIDS-P-2468, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, 1999.
- [16] K. Y. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10:201–225, 1993.
- [17] E. Klavins. Toward the control of self-assembling systems. In A. Bicchi, H. I. Christensen, and D. Prattichizzo, editors, *Control Problems in Robotics*, pages 153–168. Springer-Verlag, Berlin, 2002.
- [18] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Also available at <http://planning.cs.uiuc.edu/>.
- [19] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor. Automatic synthesis of fine-motion strategies for robots. *International Journal of Robotics Research*, 3(1):3–24, 1984.
- [20] K. M. Lynch and M. T. Mason. Stable pushing: Mechanics, controllability, and planning. *International Journal of Robotics Research*, 15(6):533–556, 1996.
- [21] M. T. Mason. *Mechanics of Robotic Manipulation*. MIT Press, Cambridge, MA, 2001.
- [22] N. Napp, S. Burden, and E. Klavins. The statistical dynamics of programmed self-assembly. In *IEEE International Conference on Robotics and Automation*, pages 1469–1476, 2006.
- [23] D. Reznik, E. Moshkoich, and J. Canny. Building a universal planar manipulator. In K. F. Bohringer and H. Choset, editors, *Distributed Manipulation*, page 147171. Kluwer, Norwell, MA, 2000.
- [24] S. Tabachnikov. *Geometry and Billiards*. American Mathematical Society, Providence, Rhode Island, 2005.
- [25] G. M Whitesides and B. Grzybowski. Self-assembly at all scales. *Science*, 295:2418–2421, 2002.
- [26] J. Wiegley, K. Goldberg, M. Peshkin, and M. Brokowski. A complete algorithm for designing passive fences to orient parts. In *Proceedings IEEE International Conference on Robotics & Automation*, pages 1133–1139, 1996.

APPENDIX

This section briefly provides a formal definition and an example of an ergodic system. First, a *measure-preserving dynamical system* is a four-tuple (X, \mathcal{B}, μ, T) for which: 1) X is a set, 2) \mathcal{B} is a σ -algebra over X , 3) $\mu : \mathcal{B} \rightarrow [0, 1]$ is a measure, and 4) $T : X \rightarrow X$ is a measurable transformation that preserves measure (each $A \in \mathcal{B}$ satisfies $\mu(T^{-1}A) = \mu(A)$).

A measurable set $A \in \mathcal{B}$ is called T -invariant mod 0 if $\mu(T^{-1}(A) \triangle A) = 0$, in which \triangle denotes the symmetric difference. Note that if this is true then A is T^n -invariant mod 0 for all n .

A measure-preserving transformation T as above is *ergodic* if for every T -invariant element mod 0 measurable set A , either A or its complement $X \setminus A$ has measure zero.

Here is a simple example. Let $X = [0, 1]$ with 0 and 1 identified to form a circle. Let the transformation be $T : x \mapsto x + \theta$, in which θ is irrational. In this case, T is ergodic. However, if $\theta = p/q$ is rational (in lowest terms) then T is periodic, with period q . This implies that T cannot be ergodic. For any interval of length a , with $0 < a < 1/q$, its orbit under T is a T -invariant mod 0 set that is a union of q intervals of length a ; therefore, it has measure $qa \in (0, 1)$.