January 2001

Assembly as a noncooperative game of its pieces: analysis of 1D sphere assemblies

H. Isil Bozma
Bogazii University

Daniel E. Koditschek
University of Pennsylvania, kod@seas.upenn.edu

Reprinted from *Robotica*, Volume 19, Issue 1, January 2001, pages 87-93.
http://journals.cambridge.org/action/displayJournal?jid=ROB

NOTE: At the time of publication, the author, Daniel Koditschek, was affiliated with the University of Michigan. Currently, he is a faculty member of the School of Engineering at the University of Pennsylvania.

Assembly as a noncooperative game of its pieces: analysis of 1D sphere assemblies

Abstract
We propose an event-driven algorithm for the control of simple robot assembly problems based on noncooperative game theory. We examine rigorously the simplest setting — three bodies with one degree of freedom and offer extensive simulations for the 2 DOF extension. The initial analysis and the accompanying simulations suggest that this approach may indeed, offer an attractive means of building robust event driven assembly systems.

Keywords
noncooperative game, robot assembly, event driven assembly

Comments
Reprinted from *Robotica*, Volume 19, Issue 1, January 2001, pages 87-93.
http://journals.cambridge.org/action/displayJournal?jid=ROB

NOTE: At the time of publication, the author, Daniel Koditschek, was affiliated with the University of Michigan. Currently, he is a faculty member of the School of Engineering at the University of Pennsylvania.

Assembly as a noncooperative game of its pieces: analysis of 1D sphere assemblies

H. Işil Bozma* and Daniel E. Koditschek†

(Received in Final Form: August 2, 1999)

SUMMARY

We propose an event-driven algorithm for the control of simple robot assembly problems based on noncooperative game theory. We examine rigorously the simplest setting – three bodies with one degree of freedom and offer extensive simulations for the 2 DOF extension. The initial analysis and the accompanying simulations suggest that this approach may indeed, offer an attractive means of building robust event driven assembly systems.

KEYWORDS: Noncooperative game; Robot assembly; Event-driven algorithm

1. INTRODUCTION

This paper, a sequel to that cited in reference [1], concerns the simple assembly problem depicted in Figure 1, where a set of objects lying on a table are managed by a robot manipulator. The parts are unactuated and cannot move unless gripped and dragged by the robot. We are interested in developing feedback based approaches to the automatic generation of actuator commands that cause the manipulator to move such a set of pieces from an arbitrary initial disassembled configuration to a specified final assembled configuration. Traditionally within the motion planning literature, assembly has been approached in an open-loop manner: an "off-line" geometric trajectory planning phase is followed by an "on-line" trajectory-tracking phase.[2,3] However, general problems of motion planning may alternatively be solved by an approach that employs feedback to achieve "planning" via event driven reactions. In this paradigm, the planning and action phases are consolidated: motion plans and control commands are generated simultaneously by a closed loop vector field – the result of applying the reaction rules at every state encountered along the way. In contrast to open loop plans, if the vector field is appropriately constructed and implemented, then the robustness against small disturbances as well as obstacle avoidance and convergence to the goal state may be guaranteed. Closed loop systems compensate as well for large unanticipated disturbances that are not too frequent and leave the state within the domain of attraction of the goal.

Recent work in extremely simplified problem settings suggests that such feedback techniques may be extended to the problems depicted in Figure 1 as well: the automatic generation of parts mating sequences along with the motion

* Intelligent Systems Laboratory, Department of Electrical and Electronic Engineering, Boğaziçi University, Bebek, Istanbul 80815 (Turkey). http://www.isl.ee.boun.edu.tr/~isil
† Artificial Intelligence Laboratory, EECS Department, College of Engineering, University of Michigan, Ann Arbor, Michigan 48109 (USA). http://ai.eecs.umich.edu/people/kod/kod.html

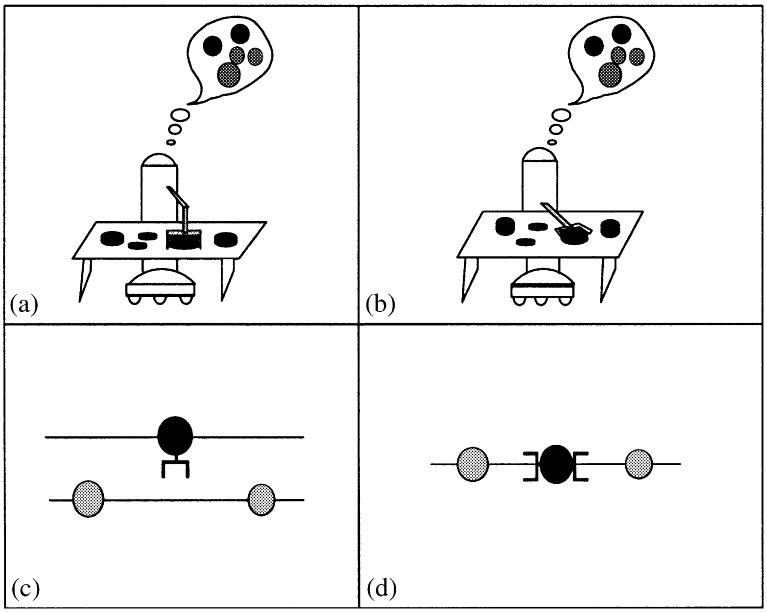

Fig. 1. Simplified 2 assembly scenarios: (a) 2 DOF Exogeneous assembly; (b) 2 DOF Endogeneous assembly; (c) 1 DOF Exogeneous assembly; (d) 1 DOF Endogeneous assembly.

control problems that arise at each step in the sequence of moves[1,4]. Although limited at present to such simplified versions of the problem wherein the parts have one or two degrees of freedom and have simple shapes, these techniques may well generalize to higher degrees of freedom and more complex shapes as does the original framework. Yet there is another complication arising in multiple parts assembly that has not yet been addressed in the closed loop motion planning literature: the situation wherein the robot inhabits the same configuration space as the parts being manipulated*.

Figure 1(a), depicts the previously investigated *exogenous* version of the problem. Since the pieces inhabit one copy of $\mathbb{R}^2$ and the robot is isolated in another, the free placements of the pieces in the associated configuration space are independent of the robot's location. A simulation study of a feedback based solution to this problem has been presented in reference [4]. In the one degree of freedom version of this exogenous problem, depicted in Figure 1(c), the pieces inhabit the line $\mathbb{R}$ and the robot is isolated in a parallel copy. A correctness proof for a feedback based solution to this one degree of freedom exogenous assembly problem is offered in reference [1].

However, in the most relevant settings of the assembly problem, the robot cannot be separated from the enviroment to be manipulated – a situation we will call the *endogenous assembly* problem. In this problem setting, the robot inhabits the same copy of $\mathbb{R}^2$ as the pieces, as depicted in Figure 1(b) for the two degree of freedom case. For the one degree of freedom case depicted in Figure 1(d) the robot lies on the same line as the pieces. Now, the free placements of the pieces and the robot become interdependant. For example, once the robot is included in the workspace, the topology of the free configuration space will potentially change depending on which part the robot mates.

1.1. Contribution of the paper

In this paper, we present a feedback based solution to the endogenous assembly problem, offer extensive simulation study of its generalization to the two degree of freedom case (Figure 1b), and prove its correctness for the 1 degree of freedom case (Figure 1d).

Specifically, we suggest in $\mathbb{R}^2$ and show in $\mathbb{R}^1$ that by sequentially switching among a family of feedback controllers, a plan can be generated in a completely reactive manner that is ensured of convergence – either to successful completion of the assembly or to termination in a spurious local minimum if and only if the task is infeasible. This is achived via the following steps:

Move part: We design a set of feedback controllers – one for moving each different part. Each of these controllers is defined by a navigation function[5] for the corresponding *part-mated-to-robot* pair that encodes the goal configuration for assembling that part along with the obstacles presented by all the other parts when doing so.

Mate part: The robot is sent to mate with one designated part at a time and if the mating succeeds, continues with the assembly of that part until it becomes blocked. The mating is achieved by a controller again arising from a navigation function that encodes the allowed mating configurations and presents all the parts as obstacles.

Next part: If a mating fails because the robot encounters a local minimum of the mating function prior to reaching the designated part, then next-part is chosen and mate-part is re-invoked. Similarly, when move-part terminates at a local minimum of move-part function, then a next-part is chosen and mate-part is re-invoked. Once blocked in this fashion, the robot switches to the assembly of the other part.

The assembly plan is *implicitly* defined by which and in what order the individual parts' controllers are selected during a given run. An assembly plan is correct if it implies the composition of controllers in a manner that ensures task achievement in case of feasible assembly and termination in case of infeasible assembly. Snapshots from a typical simulation run of our algorithm applied to the degree of freedom endogenous assembly problem depicted in Figure 1b, are presented in Figure 2.

1.2. Motivation

Why is this little problem worth studying? It is a matter of considerable interest to us that a "dumb" feedback policy is capable of making what appear to be "strategic" decisions. For example, the problem depicted in Figure 2 requires that a subset of initially correctly placed parts be moved out of the way in order to bring a blocked part into place and our feedback policy does indeed figure this out. We would like to understand how such capabilities might be predicted and generalized but the 1DOF case (Figure 1b), its utter simplicity notwithstanding, turns out to be the hardest endogenous problem for which we presently have a provably correct algorithm. Thus, it appears that novel techniques of analysis suited to this problem will be required in order to better understand what degree of "strategy" we may expect in general from such switching feedback controlled systems.

Of course, the problem is completely trivial when we remove the requirement that the task be based on feedback (reactive planning). But for the problem of interest, rather than developing a plan of assembly at the beginning of manipulation which is then executed, our plan must be generated as the assembly evolves. As will be shortly seen, this is not a standard problem in either control theory or optimization and the question now arises: how is the global convergence of such a hybrid system to be guaranteed? This paper develops a methodology for studying that question. Our present methods of proof rely on the notion of a noncooperative game. Convergence is established by showing that the equilibria of the resulting discretely iterated system have attracting properties – global asymptotic stability in case of feasible assembly and local asymptotic stability with no additional periodic limits in the infeasible assembly case.

We are of course interested ultimately in more realistic

* There does not seem to be too much attention paid even in the traditional motion planning literature to the distinction between exogenous and endogenous assembly situations. A notable exception is given in reference [6].

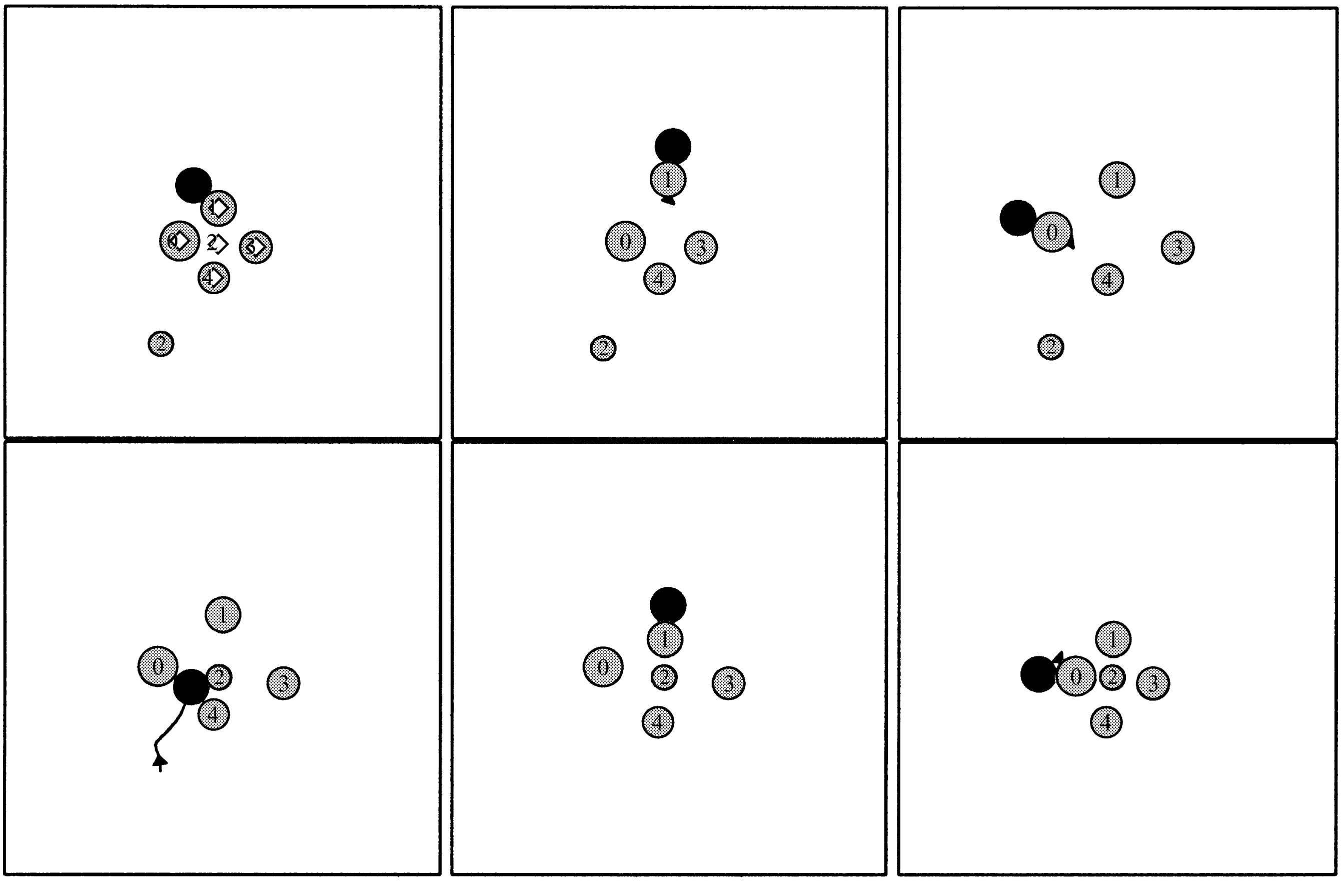

Fig. 2. Sequence of moves in a 5 part assembly. Diamonds represent goal positions of the shaded, numbered disks. The robot is depicted by the dark disk.

tasks such as 2DOF (and ultimately 3DOF) endogenous assemblies. Indeed, the 1DOF algorithm presented here appears to generalize in a straightforward manner to such settings as we try to suggest in the simulation study of Section 2.5. However, the analysis developed here has not yet found generalization to these settings and we offer the present work in the hope that others may be motivated to work on the problem as well.

1.2.1. Feedback. The tradeoffs between feedforward (predictive planning) and feedback (reactive planning) have been by now exhaustively debated both in the robotics literature and beyond[7,8] to the point that there seems little worth in holding forth for one or the other in abstraction*. Crudely speaking, feedforward achieves performance and feedback achieves safety: clearly, both are needed and may be applied at the various levels in the robot command hierarchy. Our view is that performance may always be added after a system is working safely but that the converse may not always be true.

The notion of safety in question here relates to the predictability of the inevitably encountered error detection and recovery cycle. Our experience suggests that failures in machine reliability frequently occur because of events which are not intrinsically unrecoverable but which violate dramatically our models and cannot be anticipated. Wire-wrapped boards occasionally send spurious signals, balls fly off paddles in completely "wrong" directions, defective parts slide off the gripping tool in a novel fashion; all manner of temporary setbacks occur which "might" have been made right with a little more a thought". But there can never be sufficient thought. While control and recovery policies founded on human anticipation are clever, they intrinsically take an "optimistic" view – that any possible environmental state transitions have been included in the exception handler. In contrast, feedback policies take the most "pessimistic" view in providing a response to *any possible state* the environment could be in at any moment.

To be a little more concrete, let the state of the environment be represented by some set of elements $b \in B$ (positions of each unactuated degree of freedom) and u be the means by which a robot can change the state of the environment according to the rule:

$$b = f(b, r) \tag{1}$$

In the specific problem posed in this paper, f represents the manner in which the object's position is affected by that of the robot r when it is being moved by the robot. We seek a means of assigning to the robot a next part to assemble as a function of its previous state, a function,

$$r = \Phi(b) \tag{2}$$

that induces a closed loop system governed by the iterates of the map:

$$T(b) = f(b, \Phi(b))$$

* A related discussion can be foun in reference [9].

in such a fashion that a large set of initial conditions are eventually drawn into the desired goal G after a number of moves. More preferably, we desire that *almost* all initial conditions can be guaranteed to eventually arrive at the goal.

1.2.2. Contrast with planning. In contrast, much work in robotics is concerned with developing plans,

$$u = \Pi(t; b_0) \tag{3}$$

to bring b from a specified initial condition b_0 to a desired final condition. In artificial intelligence, the tradition has been to write down Π in the form of "if-then-else" statements. In control theory, the tradition has been to write down Π in a form that effectively inverts the plant around a reference path from start to finish. Because they are written by humans, plans having the form of (3) can result in impressive behavior when all is as modelled. But Π is often very sensitive to b_0 (an open-loop move-box-to-pallet will fail badly if the box is not initially as assumed) and relies very strongly upon the predictive model as represented by Eq. (1). Of course most implemented robot systems surround (3) with periodic sensor derived "verification" checks and include "exception handling". But no human programmer can anticipate all the varied ways in which the real world will depart from the response model (1). And assuming, as is typical, that anticipated errors are recovered by invoking a variant of (3) with the new view of the present environment b_k there is established an effective closed loop,

$$b_{k+1} = f(b_k, \Pi(k, b_k))$$

a form of (2) whose steady state properties are almost never worked out and, moreover, rarely easy to ponder. Since we hope to study the reactions the world will have to our choice of actions, we prefer to start with (2).

1.3. Background literature

1.3.1. Robotic assembly and factory automation. Our focus on correctness proofs for geometrically simplified assembly feedback laws is motivated in part by the hope of helping to integrate geometrically detailed approaches to robotic assembly within factory automation settings. On the face of it, the coarse view of part shape taken here seems to limit the application of these ideas to relatively unstructured problems with simple components wherein unexpected and potentially persistent disturbances necessitate the reactive emphasis. One imagines tasks such as changing batteries, packing groceries, arranging furniture, and so on. In contrast, the last decade's advances in automated assembly[3,10,11] address the geometric and operational details of mating, seemingly to the exclusion of error detection and recovery procedures. For example, the Archimedes system[12] already functioning in an important practical application setting, incorporates fast collision detection applicable to very complex part geometry, sophisticated mating functions, and detailed provisions for respecting various user specified insertion constraints, but no checking for the success of the operations, nor any provision for handling failure. Historically, the experience reported in the robotics literature[2,13] suggests that both geometric detail and online error detection and recovery will be important even in structured factory assembly applications. Our attention is focused exactly on this problem – on the global convergence of the assembly operation from as large a set of initial conditions as possible – and we have intentionally "postponed" a careful treatment of the geometry in the interests of beginning to get this aspect of the assembly problem right.

Hybrid control schemes for factory automation, dating back to Lyons' pioneering work[14] represent an increasingly popular area of contemporary research[15,16]. The central difficulty in applying such discrete control methods to practical problems lies in choosing the "coarsening" – in effect, designing a partition of the underlying configuration space such that transitions between its cells can be exactly modeled at the higher level. The convergence of our discrete time game and its formal correspondence to continuous time motions suggests an alternative approach to the problem of hybrid control in assembly.

1.3.2. Game theory. Our analysis of the 1DOF discrete system is guided in part by Başar's study of noncooperative games[17]. Specifically, we have found their work on the existence, stability and iterative computation of non-cooperative equilibria[18] in nonquadratic convex Nash games particularly relevent to our studies. Motivated by their results, previous work has reported a noncooperative game formulation of robotic tasks in general[19,20] and a cooperative game – theoretic interpretation of exogenous assembly[1].

1.3.3. Nonholonomy. Assembly problems present more environmental (unactuated) degrees of freedom to be manipulated than there are robotic (actuated) degrees of freedom with which to manipulate. In consequence, as it is intuitively clear, contact with the environment must be repeatedly made and broken, and as seems less obvious but can be formally demonstrated, event driven robot strategies must have a hierarchical nature. This may be seen by noting that the formal correspondence to nonholonomical constrained dynamics[21]. In our view, the key observation in this context has been made by Bloch *et al.*[22] who have shown that all mechanical problems featuring nonholonomic kinematic constraint in mechanical systems fall into the class of control systems identified by Brockett[23] who showed that even when these systems are completely controllable, they fail to be continuously stabilizable. Our interpretation of this formal result animates much of our work in this area and indeed motivates the premises of this paper: since no single smooth feedback law can avail, we are led to introduce multiple families of feedback laws and then tune and switch between them.

2. PROBLEM SETUP

Given N unactuated disks in $\mathbb{R}^n$ (the "parts"), denote the location of the center of the ith by b_i, and its radius by ρ_i. The total configuration of the parts is denoted $b = [b_1, \ldots, b_N] \in \mathbb{R}^{Nn}$. Given an actuated disk in $\mathbb{R}^n$ (the "robot"), denote the position of its midpoint by r, and its radius by ρ_r. We will consider the simplest quasi-static (purely kinematic) version of the problem* adopting the simple first order generalized damper model for robot motion,

$$\dot{r} = \tau \tag{4}$$

where τ denotes an applied force. All the results of this paper can be generalized to the Newtonian model of motion at the cost of greater notational effort yet without changing the essential features of the problem or its solutions[1].

We will posit a robot "gripper" capable of engaging and releasing the parts as desired. As an extension of the generalized damper model of motion, the parts are assumed to move with the robot when engaged and are motionless when released. Reflecting these assumptions we write

$$b_i = c_i(b_i, r)\dot{r} \qquad \mathrm{i} = 1, \ldots, \mathrm{N}$$

where, c_i is the coupling function between the robot and the unactuated part that vanishes when the two bodies are not touching, $|b_i - r| \leqslant \rho_i + \rho_r$. In this paper, it is convenient to assume that $c_i(b, r) = 1$ when $\|b_i - r\| \leqslant \rho_i + \rho_r$ (and that it vanishes otherwise as stated). More realistic coupling rules that vary smoothly with the relative distance have been presented in reference [1] and do not change the essential features of this problem. In contrast, introducing a more realistic version of c_i that makes the mating sensitive to the relative orientation of the two disks adds another dimension to the robot's configuration space, raising attendant technical questions that we have not yet considered.

Assume, finally that perfect sensing information is available: the robot always knows exactly where it and all the parts are located. The robot's task is to move the pieces to their "assembled" positions while avoiding collisions.

2.1. Feedback-based solutions

A feedback based solution takes the form of a robot force law, $\tau = g(b)$, along with a gripper schedule that results in the robot visiting and re-visiting (if necessary) each body until the desired assembly is achieved and never permitting two bodies to collide. Thus, we require a solution that brings all initial configurations to the destination from within the connected component of the configuration space and that stops the robot after some time if the destination is infeasible.

A navigation function, $\varphi : \mathbb{B} \to \mathbb{R}$ is a non-degenerate potential with one minimum in the interior of and taking its maximum uniformly on the boundary of its domain, $\mathbb{B}$, some smooth manifold with boundary. The associated gradient dynamical system

$$\dot{b} = -[D\varphi]^{\mathrm{T}}(b) \tag{5}$$

will asymptotically approach this minimum, without contacting the boundary (i.e., avoiding collisions), from all initial conditions outside a set of measure zero. It is guaranteed that such functions exist for any configuration space of interest including the present one [24]. Such a function relevant to the present case, $\tilde{\varphi}$, is constructed in reference [1], inspired by the general design introduced in reference [24] and may be written in the form:

$$\tilde{\varphi} = \frac{(\sum_{i=1}^{N} \gamma_i)^k}{\beta} \tag{6}$$

where the term γ_i encodes the body's distance from the desired position in the completed assembly, $d_i \in \mathbb{R}^n$, as $\gamma_i = \|b_i - d_i\|^2$, and the mutual intersections of parts comprising the configuration space "obstacle" to be avoided is encoded as

$$\beta(b, \rho) = \prod_{\substack{i=1 \\ j=i+1}}^{\substack{i=n \\ j=n \\ i\neq j}} (\|b_i - b_j\|^2 - (\rho_{ij})^2), \tag{7}$$

where $\rho_{ij} = \rho_i + \rho_j$.

Now consider the application of this function to the case of independently actuated bodies. The actuation vector (torques applied to the ensemble of bodies) is generated according to the gradient field (5). The curve $b(t)$ *simultaneously* specifies the *time varying* position of the ensemble of bodies in the configuration space. They appear in the workplace to find their way cooperatively in the specified "assembled" configuration if this is possible. This presumes a robot that can independently and simultaneously manipulate all the bodies at once – a most impractical assumption. Rather, we will assume the robot can move one body at a time and our original feedback controller must be adapted so that the robot attends to one body at a time.

2.2 The exogenous setting: A cooperative game

To do so, remove the bodies' independent actuators, and place an actuated robot in a space "parallel" to their workplace (refer to Figure 1(a) and (c)), leading to the exogenous assembly case. It can be shown that there exists no single continuous feedback control τ for (4) capable of forcing the robot to visit and re-visit each body until they are all brought into the desired goal locations[1]. Instead, the procedure in Reference [1] is to introduce a family of continuous feedback laws based upon the navigation function $\tilde{\varphi}$* and then design an effective rule to switch between them. We desire that the bodies' motions tend to decrease the "cost", $\tilde{\varphi}$. However, only the robot is actuated, thus only one body may move at a time. A high-level controller operates in principle by selecting a body i and applying: the low-level control law $b_i = -Db_i\tilde{\varphi}$ – the navigation function gradient evaluated while all the other bodies remain fixed in their positions. The body is halted at a relative minimum and the next body is chosen based on having the largest magnitude of the navigation function gradient at this minimum point. By interpreting $D_{b_i}\tilde{\varphi}$ as the

* This structure characterizes most of the classical nonholonomically mechanical systems and as shown in reference [21] also describes the essential features of assembly problems.

* Our function $\tilde{\varphi}$ takes exactly the same extrema as those presented in reference 1, but differs in that we do not bother to "squash" the unbounded derivatives at the boundary in order to facilitate the presentation. Adding such "squashing" terms is straightforward and does not change any of our results.

derivative of the projection of $\tilde{\varphi}$ to the $N-1$ dimensional subspace where the other bodies are at $(b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_N)$, the high-level controller may be seen as refereering an N-player game where each body's payoff is simply the projection of $\tilde{\varphi}$ onto its configuration space. Since each payoff is a coordinate slice of the same global function, this is a cooperative (identical payoff) game. The correctness of this two-level controller for the case $n=1$ is demonstrated by establishing the global convergence of this identical payoff game[1].

2.3 The endogenous setting: A noncooperative game

We use the term *game* to describe a discrete dynamical system on a state space of *players*, $\{b_i\}_{i=1,N}$* whose evolution is governed by the limiting properties of a set of coupled gradient vector fields in a manner that is now described.

We presume a set of "payoff" functions $\tilde{\varphi}(b_1, \ldots, b_N)_{i=1,N}$ – a collection of smooth scalar valued maps on the state space. Denote by $\bar{b}_i \triangleq (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_N)$ a vector in the subspace $\mathbb{R}^{(N-1)n}$, corresponding to the removal of the i^{th} component of $b \in \mathbb{R}^{Nn}$. Define the vector field f_i to be the negative gradient of the map $\tilde{\varphi}_i$ with respect to the vector b_i

$$f_i(b_i; \bar{b}_i) \triangleq -[D_{b_i}\tilde{\varphi}_i((b_1, \ldots, b_N))]^T. \tag{8}$$

Here, the semicolon notation is intended to call attention to the parametric role that the other players $\bar{b}_i = (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_N)$ will play in the motion of player b_i. When $\tilde{\varphi}_1 = \tilde{\varphi}_2 = \ldots = \tilde{\varphi}_N$, we have the situation of Section 2.2. Otherwise, we have a general noncooperative game. Motion on this subspace of the state space will be governed by the limit properties of the gradient dynamical system

$$b_i = f_i(b_i; \bar{b}_i)$$

whose integral curve through the initial condition b_i^0 will be denoted by $f_i^t(b_i^0; \bar{b}_i)$. When $f_i(b_i^*; \bar{b}_i)=0$ implies that $D_{b_i}f_i(b_i^*, \bar{b}_i)$ has full rank, it can be guaranteed that the limit set $f_i^\infty(b_i; \bar{b}_i) \triangleq \lim_{t\to\infty} f_i^t(b_i; \bar{b}_i)$ of every trajectory through any possible initial condition is some isolated singularity $f_i^\infty(b_i; \bar{b}_i) = \{b_i^*\}$. This rank condition holds generically over $\bar{b}_i$ but it will not be true for all $\bar{b}_i$ – that is, the vector field f_i passes through bifurcation points as the parameters $(b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_N)$ vary over the state space. In order to proceed, the same limiting properties must persist even at bifurcation.

With these assumptions and notation in force, each function $\tilde{\varphi}_i$ gives rise to a (generally discontinuous) map $f_i^\infty(b_i; \bar{b}_i)$ of the entire space into the ith projection. Letting $\iota: \mathbb{R}^{Nn} \to \{1, \ldots, N\}$ denote some indexing scheme, we refer to the iterates of the discrete map, $T(b) = (T_1(b), \ldots, T_N(b))$, with components

$$T_i(b) = \begin{cases} f_i^\infty(b) & \text{if } i = \iota(b),\ i = 1, \ldots, N \\ b_i & \text{otherwise} \end{cases} \tag{9}$$

as determining a game of the players $\{b_i\}_{i=1,N}$. The fixed points of the discrete system are the solutions of the game.

* In this case, the bodies to be assembled are the players.

Note that for a simple 2-player game, which we will shortly introduce, the indexing can be as simple as:

$$\iota(b_1, b_2) = \begin{cases} 1 & \text{if } b \in f_2^{-1}(0) \\ 2 & \text{if } b \in f_1^{-1}(0) \end{cases}$$

2.4 Endogenous assembly setting leads to a noncooperative game

When the robot is mated in the same space with different objects, the free configuration spaces for each mated robot-body pair changes depending on which body the robot is mated with.

For ease of exposition, we will specialize the discussion to the one degree of freedom case, $n=1$. The simulation study $n=2$ will make clear the appropriate generalization, and this more focused discussion will facilitate the formal presentation of Section 3. Consider 2 bodies*. The position of each body, $i=1, 2$ is denoted $b_i \in B_i$, $B_i \subseteq R$, its desired position $d_i \in B_i$, and its radius ρ_i. Let $b \in B_1 \times B_2$ denote the vector of all positions and $d \in R^2$ the vector of all the desired positions. The robot's position is denoted by $r \in \mathbb{R}$ and its radius is ρ_r.

The bodies must never be allowed to touch each other as they are dragged along the way to their respective goal positions. The physical constraint that the bodies cannot overlap results in a free configuration space consisting of 2 disjoint regions, only 1 of those being physically meaningful. For example, in the 2-body assembly case, the legal body configurations are in $B = B_u \cup B_l$ with

$$B_u \triangleq \{(b_1, b_2) : b_2 - b_1 > \rho_{12}\}; \qquad B_l \triangleq \{(b_1, b_2) : b_1 - b_2 > \rho_{12}\}$$

as depicted in Figure 3. A feasible task is one for which the desired destination is in the same connected component as the initial configuration.

When mated, the position of the robot and that of the body are coupled, describing a new 1-disk (an interval) centered at $b_i + o_i\rho_r$ with the radius $\rho_r + \rho_i$ where $o_i = sgn(r^0 - b_i^0)$. Note that o_1 and o_2 will be of the same magnitudes, but different signs where the sign of each will

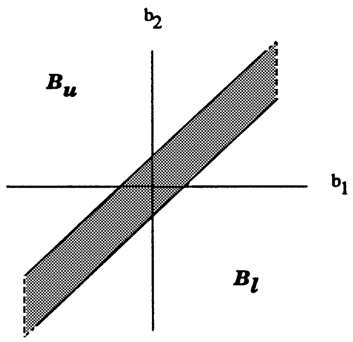

Fig. 3. Disconnected components of the assembly space.

* Notice that barring a push communicated through intermediate bodies, a robot on the same line as the spheres can manipulate only its two nearest neighbors. We will require that the bodies never be allowed to touch each other. In this setting, assemblies with a greater number of bodies are not feasible in one degree of freedom.

depend the relative position of the associated body with respect to the robot. If the robot is mated to body 1, the legal body configurations are $\tilde{B}=\tilde{B}_u \cup \tilde{B}_l$:

$$\tilde{B}_u \triangleq \{(b_1, b_2): b_2-(b_1+o_1\rho_r)>\rho_{12}+\rho_r\};$$

$$\tilde{B}_l \triangleq \{(b_1, b_2): b_1+o_1\rho_r-b_2>\rho_{12}+\rho_r\}$$

If the robot is mated to body 2, the legal body configurations are $\tilde{B}=\tilde{B}_u \cup \tilde{B}_l$:

$$\tilde{B}_u \triangleq \{(b_1, b_2): b_1-(b_2+o_2\rho_r)>\rho_{12}+\rho_r\};$$

$$\tilde{B}_l \triangleq \{(b_1, b_2): b_2+o_2\rho_r-b_1>\rho_{12}+\rho_r\}$$

Without loss of generality, assume $b_2<r<b_1$ so that $o_1=-1$ and $o_2=1$.

Now let us endow the controller with the objective functions defined previously,

$$\tilde{\varphi}_i(b_1, \ldots, b_N)=\frac{(b_i-d_i)^k}{\tilde{\beta}}$$

where $\tilde{\beta}_i$ encodes the $N-1$ remaining obstacles when the robot is mated with body i. The obstacle function can then be expressed as:

$$\tilde{\beta}(b)=(b_1-\rho_r-b_2)^2-(\rho_{12}+\rho_r)^2$$

Since only one body may move at a time, a two-level controller is once again required and operates as already explained. It chooses from among the low level controllers $f_i(b_i, \bar{b}_i)=-D_{b_i}\tilde{\varphi}_i$, applies it until the robot becomes blocked, navigates towards the next mating body selected based on the indexing scheme and then proceeds similarly. We can write the high level controller as the discrete dynamical system

$$b(k+1)=T(b(k)) \qquad (10)$$

where the $T:\mathbb{R}^{Nn}\to\mathbb{R}^{Nn}$ is the transition map from one "blocked" robot configuration to the next. The solutions of the game determine whether the assembly is to be succesfully completed or terminated. The analysis of these solutions in Section 3 forms the central contribution of the paper.

2.5. A simulation study: 2 DOF endogenous assembly

In conjunction with the 1DOF analysis, to be presented below, we have pursued an extensive simulation study of the 2DOF version of our feedback solution to the endogenous assembly problem*. This solution takes the form of the hybrid controller depicted in Figure 4. The next-part decision is made by an index function, ι (9), that chooses the part whose gradient field (8) has the greatest magnitude.

As outlined in the introduction, *assemble-part* is composed of two classes of controllers: *mate-part* and *move-part*. The *mate-part* controller is defined by the gradient vector field generated by a navigation function whose goal encodes the designated next mate and whose obstacles include all the other parts. In the case that this mating is impossible (i.e. the robot and the piece to be mated are not presently in the same connected component of the configuration space), the switching automaton goes back to the *next-part* state and chooses the part whose gradient magnitude is next greatest and the process repeats. If the designated contact is achieved, then the *move-part* algorithm defined by the gradient field (8) brings the robot-part pair towards that part's goal set until its motion becomes blocked, that is, the vector field (8) goes to zero. The switching automaton once more goes back to *find-next* state, and the process repeats. When all the parts' gradient fields are sufficiently small, the automaton declares the assembly task complete and the robot remains in its state. Thus, the whole assembly can be viewed as the robot refereeing a noncooperative game being played between subassemblies.

The nature of the present simulation study and the form of the presentation are directly inspired by the work reported in reference [4]. A typical anecdotal run illustrating the rudimentary "strategy" displayed by this scheme has been discussed in the introduction (Figure 2). However strategic, the robot's decisions will typically not yield optimal performance, and, depending upon the particular initial conditions and the difficulty of the final assembly, some runs may result in unnecesarily numerous switches between parts or arc length traveled. As an example, consider the situation depicted in Figure 5. Observe that in this particular case, no part except part 1 is near its goal configuration.

The sample run for this case is shown in Figure 6 – where the frames show sequentially (but not uniformly in time)

* Portions of this section are taken from reference [25].

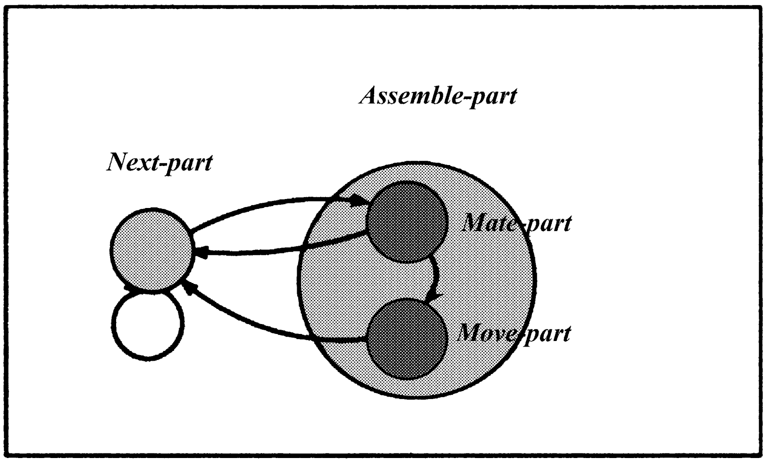

Fig. 4. The composition of behaviors.

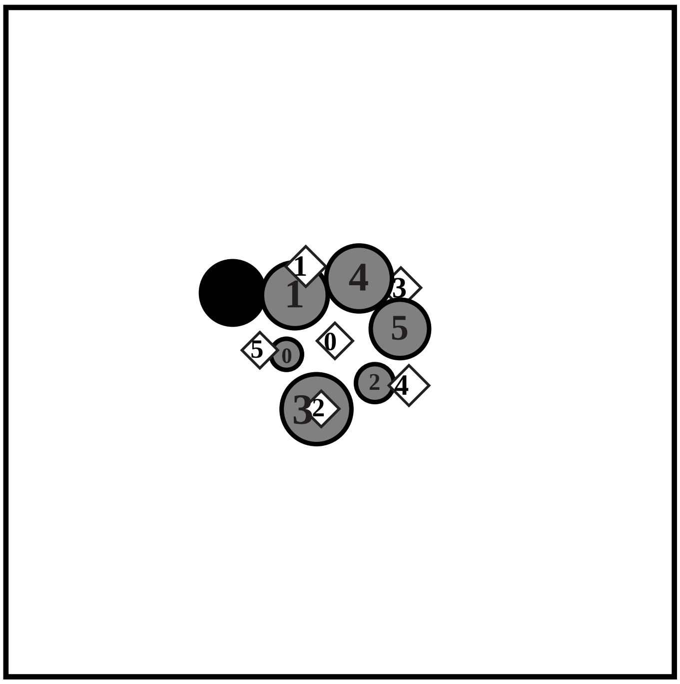

Fig. 5. A 6 sphere assembly sequence with destination $\beta = 8.8 \times 10^{48}$.

sampled moves of the robot-starting with the initial configuration. In the top center frame, the robot moves part 1 away from its goal position and then it moves part 4 closer to its goal position. It then moves part 5 closer to its goal position. In the next frame, we observe part 2 being moved to a closer neighborhood of its assembled position. Similarly, part 3 is moved to a closer neighborhood of its assembled position in frame 6. Part 0 is then moved to its goal position in the next frame. The robot then improves the positional accuracy of the parts. Thus, it is clear that this sequence of moves is not "intelligent" since the maneuver might have been made in a smaller number of moves. It is, however, fully automatic and an appropriate sequence of moves will be made from any initial condition.

2.5.1. Statistics. To test the performance we have given up in the interests of autonomous feedback plan generation, we have conducted an extensive simulation study of the problem domain depicted in Figure 7. Our assemblies contain six disk-like objects of varying radii. We consider six different randomly chosen final assembly configurations of increasing difficulty as measured by log $\beta(d)$ – the log of the destination's β value, corresponding to the "tightness" of final fit as shown in Figure 7.

In all the simulation runs reported, the initial position of the robot is the left upper corner of the workspace. In the graphs, each data point represents the mean and standard deviation of 25 runs with random initial configurations. In this study, we use four measures of performance:

1. Normalized assembly path length, $\mathrm{npl} = \frac{\int_{t_i}^{t_f} \|\dot{b}\|\, dt}{\|b(0) - d\|}$, as reported in Figure 8;
2. Normalized robot path length, $\mathrm{rpl} = \frac{\int_{t_i}^{t_f} \|\dot{r}\|\, dt}{\|b(0) - d\|}$ as reported in Figure 9;
3. The number of times the robot switched between the parts as reported in Figure 10;
4. Positioning inaccuracy $\mathrm{pi} = \|b(t_f) - d\|$ as reported in Figure 11;

where t_i and t_f denote, respectively, the starting and finishing times of an assembly.

Note that the assembly path length measures the distance traveled in $\mathbb{R}^{2N}$ by the disk-like parts from an initial configuration to a final "assembled" configuration. In order to account for the variations in the initial conditions, it is normalized by the Euclidean distance from the initial configuration to the goal configuration. Notice that this "straight line" from initial condition to goal in the collected configuration space is generally infeasible – it runs through obstacles wherein the bodies must touch or overlap – so the ratio must be greater than unity. How much greater than unity seems like a reasonable measure of the "awkwardness" of the plan realized in the particular run.

In contrast, the robot path length measures the distance traveled by the robot in its two dimensional configuration space as it shuttles to and fro between the parts, both mating to and then moving each one it visits. We now discuss the graph summaries of this simulation study.

a. Normalized Path Length vs. Assembly Difficulty. Figure 8 shows that normalized path length varies in manner that matches our intuitive expectation – the closer the parts need to be packed together, a greater distance they need to be moved. In other words, the path-length performance correlates inversely with the assembly difficulty: destinations with very small β values corresponding to tightly packed goals such as in Figure 7(f) are more difficult to assemble than loosely packed goals with higher β values such as in Figure 7(a). Note that path length is on average about five times longer than the Euclidean distance between the initial and final configurations. Two factors account for this: First, the parameter k_3 of the moving function ψ_m is chosen such that the obstacle avoiding term dominates unless the part is close to its destination which means that in general parts move away from their assembled positions before moving towards them. Secondly, in many of the randomly generated initial assembly configurations, some parts, although at their assembled positions, must be moved away before other parts can be assembled. This is another illustration of how the Euclidean straight line configuration space from start to goal is an overly optimistic normalization measure – it runs through infeasible points.

b. Robot Path Length vs. Assembly Difficulty. The normalized path traveled by the robot also matches our intuitive notions of assembly difficulty. Again, tightly packed assemblies such as in Figure 7(f) cause the robot to travel a longer path length than that of a more loosely packed assembly.

The path traveled by the robot is of magnitude about 30 times that of the Euclidean distance between the initial and final assembly configurations. Three factors contribute to this: First, as explained earlier on, the robot is initially located on the upper left corner of the workspace – far from the parts to be assembled and this fact is not accounted for in our normalization. Secondly, the k_2 parameter of the mating function φ_m is chosen such that the obstacle avoidance terms dominate which means that the robot travels in a path distant from all the parts. Finally, in some of the randomly generated initial configurations where some of the parts are located close to their assembled positions, the robot may move some parts away from their locations before moving them back.

c. Switches vs. Assembly Difficulty. Figure 10 shows the mean standard deviations for the number of switches. Here we observe that the number of switches required to complete an assembly rises as a function of the assembly difficulty. The easy assemblies require on average each part to be switched three times while the more difficult assemblies have both a greater mean of the number of switches as well as higher variance.

d. Positional Inaccuracy vs. Assembly Difficulty. One expects that the positional inaccuracy of the assembled parts should similarly increase with the difficulty of the assembly. The more closely the parts need to be assembled together, the more crucial it would seem that the robot place a part precisely at its first attempt since the chance of that part being blocked by other assembled parts increases once the parts are assembled. Accordingly, as the assembly task becomes more difficult (i.e. the destination lies close to the configuration space obstacle so that the final destination

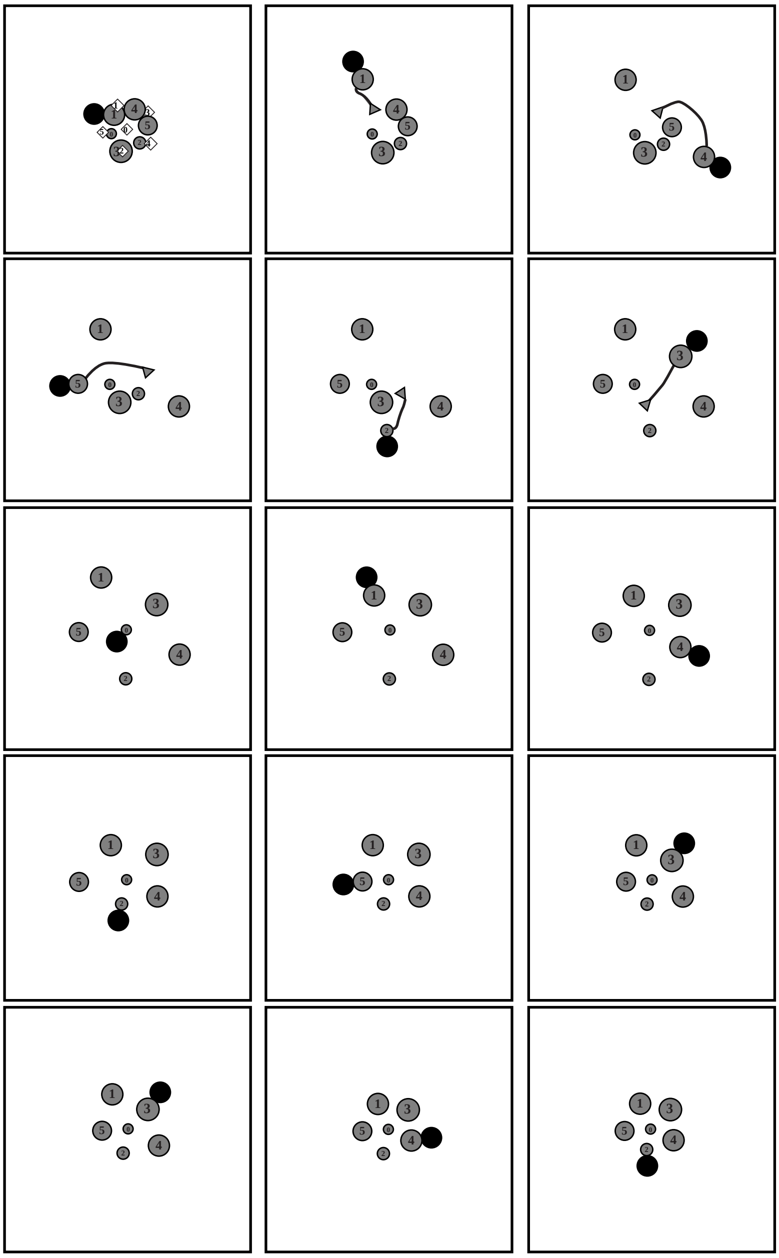

Fig. 6. Sampled assembly sequence.

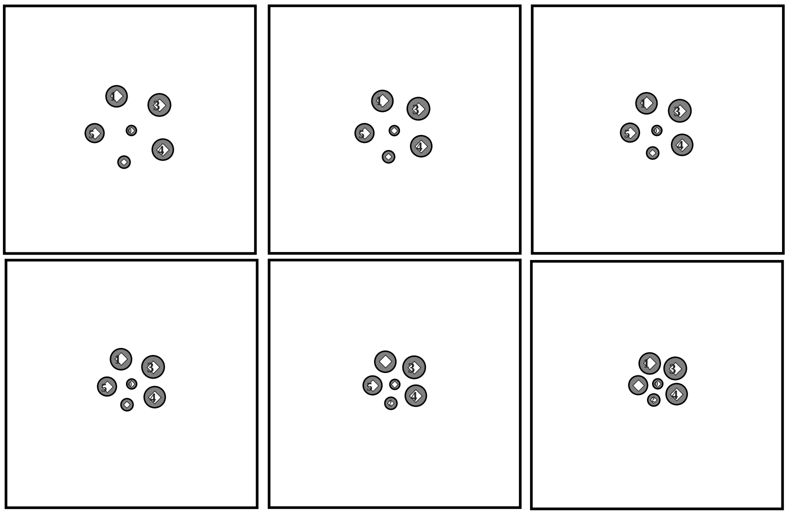

Fig. 7. Assemblies of increasing difficulty.

entails a densely packed arrangement), we have seen above that the robot spends more time in transit between the part transportation episodes. In contrast, the data show that once the parts' destinations start almost touching each other, positioning accuracy starts increasing. Our observation is that the robot's placement of the middle part in Figure 7 becomes increasingly "sloppy" (i.e. after placement, the part's center is not exactly, but rather almost at its destination) as the difficulty of final packing increases. More densely packed destinations may incur a steeper cost

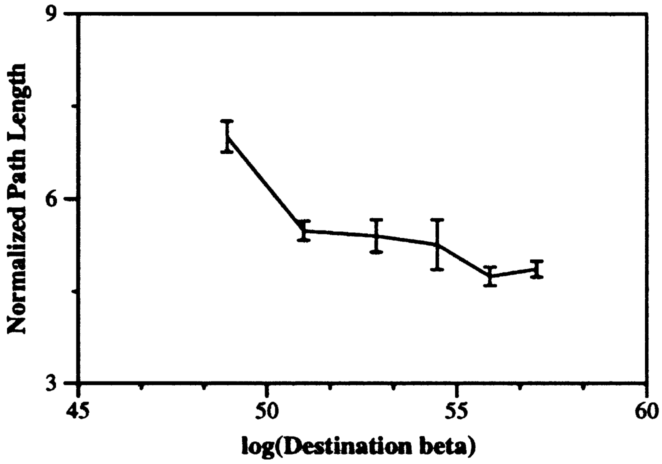

Fig. 8. Normalized path length statistics.

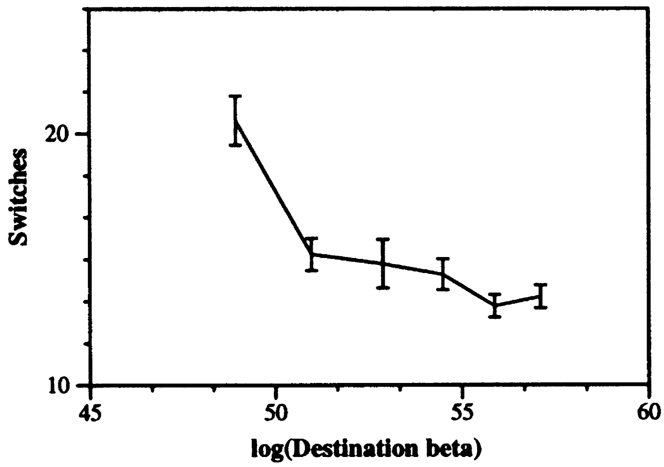

Fig. 10. Switching statistics.

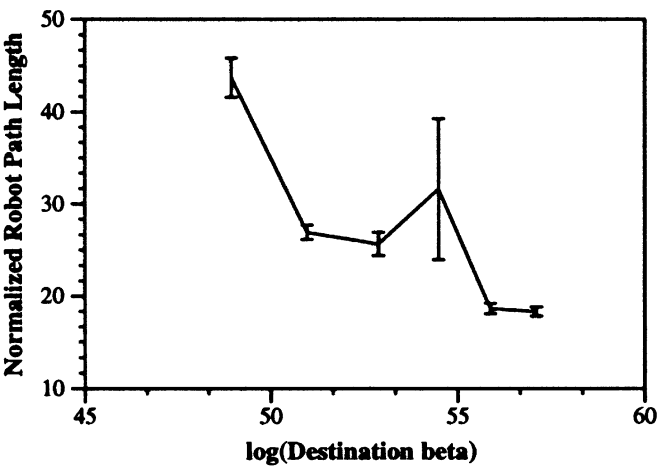

Fig. 9. Normalized robot path length statistics.

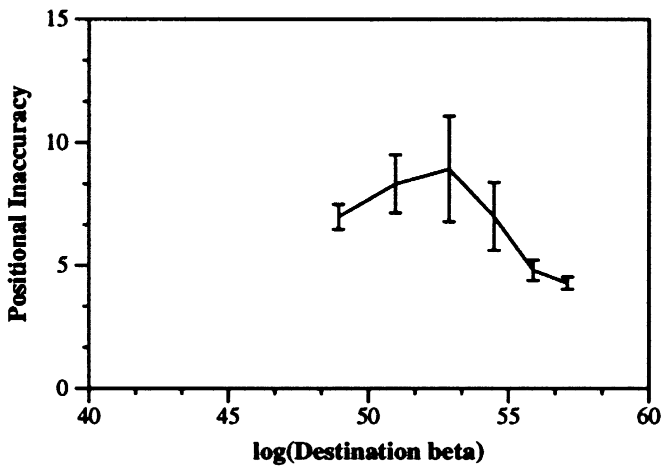

Fig. 11. Positional inaccuracy statistics.

function, so that the small gradients cannot occur until the center part has been placed almost exactly at its designated destination.

3. 1 DOF 2-BODY ENDOGENOUS ASSEMBLY

In this section, we will limit our attention to the particular case of two parts, $N=2$, on the line, $n=1$. We will first construct the game transformation function, $T: \mathbb{R}_2 \rightarrow \mathbb{R}_2$, and then simplify the associated dynamical system defined by its iterates through a diagonalization argument. That is, we will note that there exist two scalar valued functions, $r_i: \mathbb{R} \rightarrow \mathbb{R}$, $i=1, 2$ such that

$$\begin{aligned}(T \circ T)(b) &= ((T \circ T)_1(b_1, b_2), (T \circ T)_2(b_1, b_2)) \\ &= (r_1 \circ r_2(b_1), r_2 \circ r_1(b_2)).\end{aligned}$$

These diagonalizing functions are called ‘reaction functions’ in the game theoretic literature. Their existence is guaranteed when $\tilde{\varphi}_1$, $\tilde{\varphi}_2$ are convex – a nearly universal assumption within that literature. However, for the present application, such an assumption would make no sense: the space in question is not even convex, so there is no possibility of defining convex functions upon it! Nevertheless, one fact is key: all the bodies must remain in the connected component of the feasible assembly space they start in. This constraint eliminates all but one branch of our reaction sets which can then be represented as the graph of “reaction-like” functions (that we denote $r_{li}(\bar{b}_i)$, $r_{ui}(\bar{b}_i)$, respectively, on each disjoint component of the feasible assembly space). These functions are piecewise algebraic and can be solved in closed form, as shown graphically in Figure 12.

In turn, the availability of simple closed form expressions for $T \circ T$ enables us to exhaustively analyze the steady state properties of this game.

One final note in passing concerns the complexity of this analysis relative to the extreme simplicity of the problem setting. Indeed, in most numerical examples, the conclusion depicted in Figure 13 emerges from straightforward graphical analysis. Unfortunately, graphs do not constitute proofs, and of course, we are concerned with developing analytical tools that may achieve insight in higher dimensional settings such as that simulated in Section 2.5.

3.1. Summary of analysis

Recall, that the only component, $T_i(b)$, of $T(b)$ in Eq. 9 that moves at all must move the component of its argument, b_i, to the limit set of the *ith* gradient system. This limit set consists of those n-vectors, b_i, that make the gradient vanish

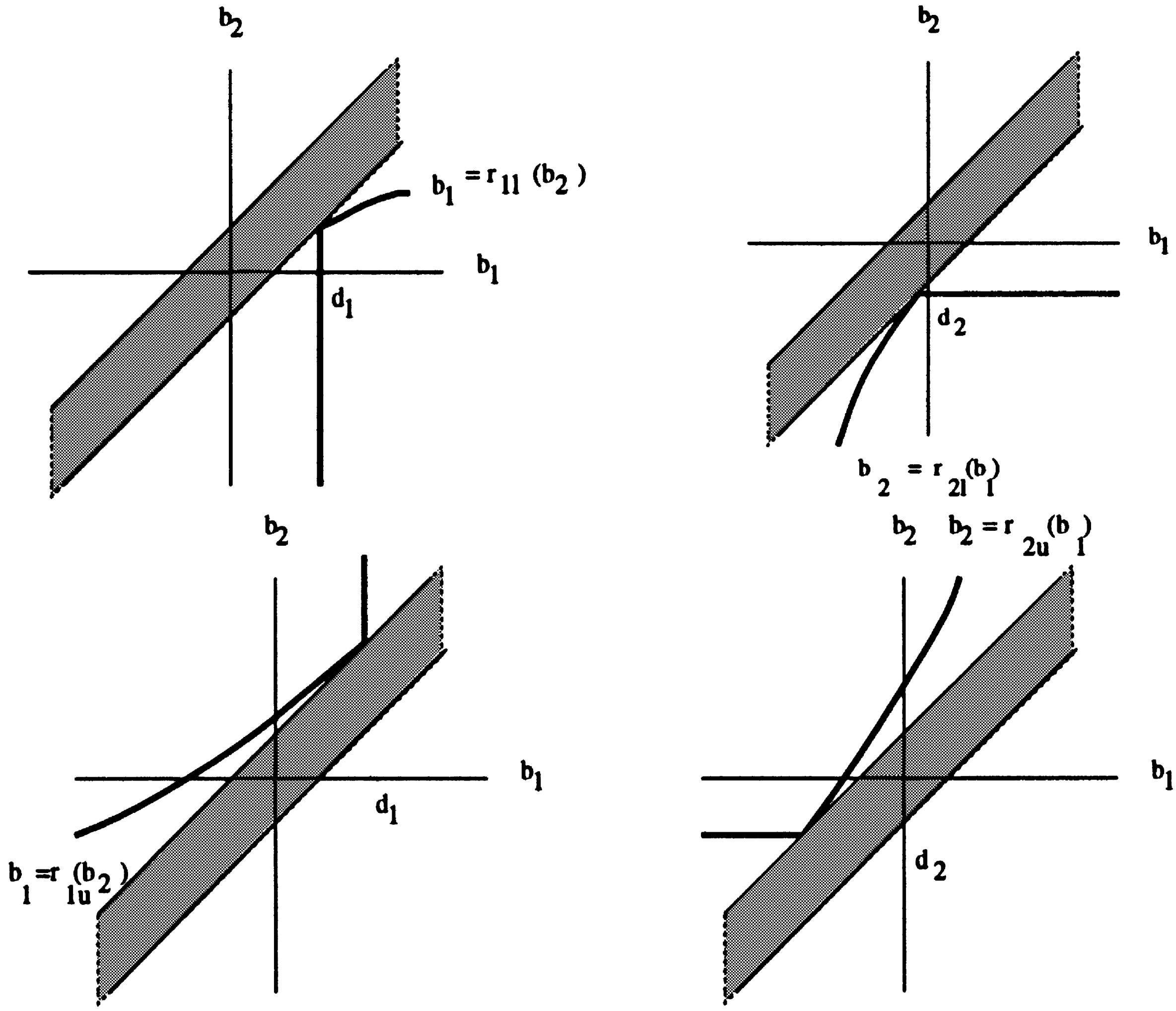

Fig. 12. Top: Reaction function for feasible assembly space for part 1 (left) and part 2 (right); Bottom: Reaction function for infeasible assembly space for part 1 (left) and for part 2 (right).

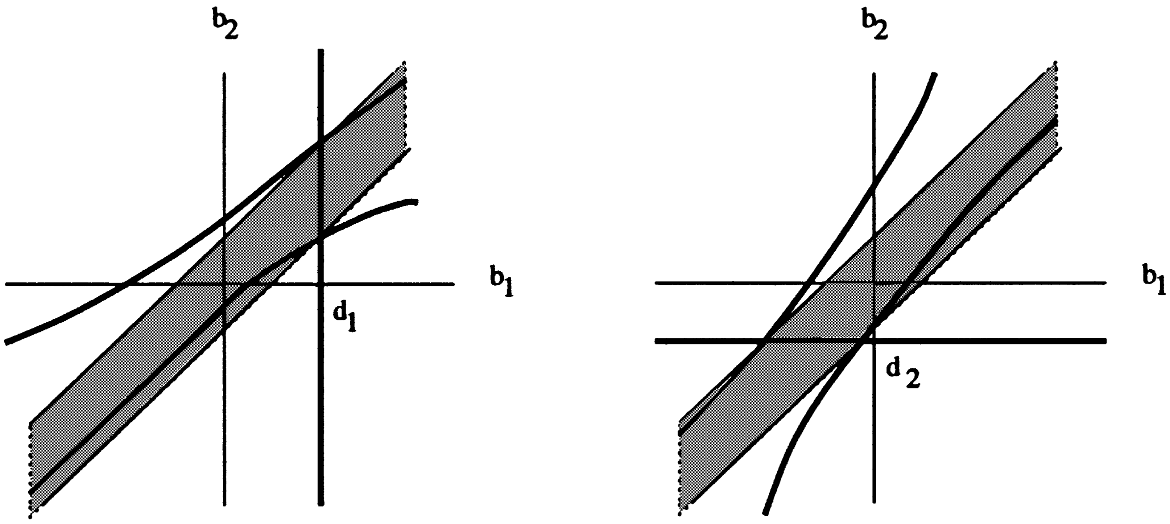

Fig. 13. Graphical depiction of reaction sets. Left: Part 1; Right: Part 2.

as parametrized by the possible values of $\bar{b}_i$,

$$R_i(\bar{b}_i) \stackrel{\triangle}{=} \{b_i \in \mathbb{R}^n : f_i(b_i, \bar{b}_i) = 0\}.$$

In the game-theoretic setting, the set $R_i = \{b_i : b_i = r_i(\bar{b}_i)\}$ is known as the reaction set of part i – since it represents the set of optimal moves for each body i given those $\bar{b}_i$ of other bodies[17]. In our case, they have the form as shown in Figure 14. Suppose for some $\bar{b}_i \in \mathbb{R}^{N(n-1)}$ it is the case that the Hessian, $F_i(b_i^R, \bar{b}_i) \stackrel{\triangle}{=} D_{b_i} f_i(b_i, \bar{b}_i)$, has full rank at each critical point, $b_i^R \in R_i(\bar{b}_i)$. Standard arguments from dynamical systems theory now imply (e.g., consult [26]) that T_i will map all but a set of measure 0 of $b_i \in \mathbb{R}^n$ to the local relative minima of $\tilde{\varphi}$ – i.e., those $b_i^{R+} \in R_i(\bar{b}_i)$ at which $F_i(b_i^{R+}, \bar{b}_i)$ is positive definite. Suppose, further, that there is one and only one such minimum, $b_i^{R+} \in R_i(\bar{b}_i)$, for each parameter value, $\bar{b}_i$. According to the implicit function theorem, we may now express the surface of minima

$$R_i^+(\bar{b}_i) \stackrel{\triangle}{=} \{b_i \in \mathbb{R}^n : f_i(b_i, \bar{b}_i) = 0 \text{ and } F_i(b_i, \bar{b}_i) > 0\}$$

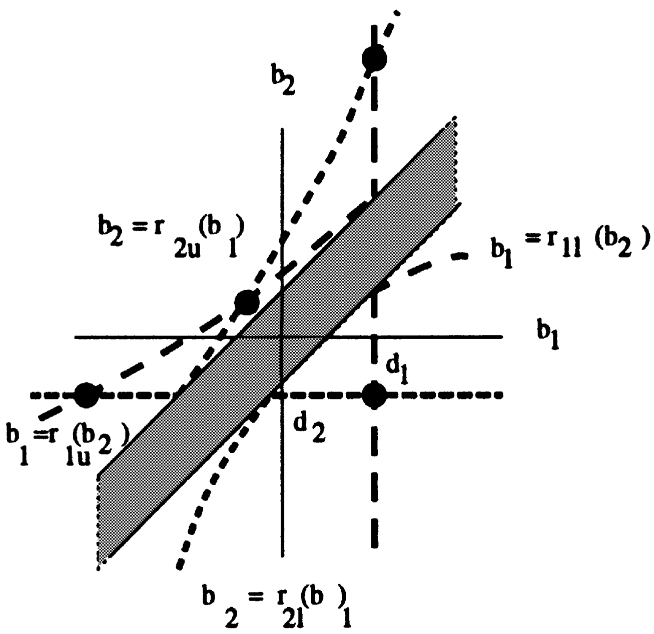

Fig. 14. Fixed points of discrete maps (shown with black dots).

as the graph of a function $r_i : \mathbb{R}^{N(n+1)} \longrightarrow \mathbb{R}^n$

$$R_i^+(\bar{b}_i) \stackrel{\triangle}{=} \{b_i \in \mathbb{R}^n : b_i = r_i(\bar{b}_i)\},$$

that solves for the root of $f_i(b_i, \bar{b}_i) = 0$. Under these circumstances, we might very simply parametrize the component of T_i as

$$T_i(b_i, \bar{b}_i) = r_i(\bar{b}_i).$$

where $r_i \in C[R^{(N-1)n}, R]$ is referred to as the reaction function[17]. In the present setting, this would correspond to the situation that one part's intermediate destination when mated to the robot is determined completely by the other part's location, independent of its own initial placement. The singleton property does not hold for the reaction sets of our game, however their restriction to each disconnected component of the free configuration space – $R_i^+(\bar{b}_i) \cap \tilde{B}_u$ and $R_i^+(\bar{b}_i) \cap \tilde{B}_l$ respectively – does turn out to have only one branch. These we will indeed parametrize as the graph of the "reaction functions," whose appropriate compositions, $r_{im} \circ r_{jm}(b_i)$, govern the motion of each mated part on each disjoint component of the feasible assembly space.* The points at which the reaction sets (functions) intersect – shown in Figure 13 – constitute the fixed points of the discrete map, $T \circ T$, which in turn, determine the properties of the solution to the game and, hence, whether the assembly is successfully completed (feasible assembly) or not (infeasible assembly).

3.2. Notation and preliminaries

Define $b \in R^2$ as $b \stackrel{\triangle}{=} [b_1\, b_2]^T$ and $d \in R^2$ as $d \stackrel{\triangle}{=} [d_1\, d_2]^T$. In the rest of the sequel, we assume w.l.o.g that $(d_1, d_2) \in \tilde{B}_l$. Denote the canonical unit vectors as $e_1 = [10]^T$, $e_2 = [01]^T$ and a rotated basis as $\upsilon_1 \stackrel{\triangle}{=} e_1 - e_2$ and $\upsilon_2 \stackrel{\triangle}{=} e_1 + e_2$. Define $\tilde{\varphi}_i = \gamma_i^k / \beta$ where

$$\gamma_i = e_i^T (b - d)$$

and

* Here and throughout the sequel, the index j is taken to be the opposite of i and the index m is either l or u respectively.

$\tilde{\beta} = -\tilde{\beta}_u \tilde{\beta}_l$; $\quad \tilde{\beta}_u(b) = -\upsilon_1^T b - o_1\rho_r - (\rho_{12}+\rho_r)$;

$$\tilde{\beta}_l(b) = \upsilon_1^T b + o_1 \rho r - (\rho_{12}+\rho_r)$$

Note that $\tilde{\beta}_l^{-1}(0)$ and $\tilde{\beta}_u^{-1}(0)$ corresponds to the two boundaries of the obstacle space respectively. It is clear that γ_i vanishes on the boundary at two distinct points l_i and u_i

$$l_i = d_i\upsilon_2 + \sigma_j(\rho_{12}+\rho_r - o_1\rho_r)\, e_j;\ u_i = d_i\upsilon_2 + \sigma_i(\rho_{12}+\rho_r + o_1\rho_r)e_j;$$
$$\sigma_1 = 1;\ \sigma_2 = -1$$

lying on the lower $\tilde{\beta}_l^{-1}(0)$ and the upper $\tilde{\beta}_u^{-1}(0)$ respectively. Observe that if we define

$$\beta = -\beta_u\beta_l \qquad \beta_u(b) = -\upsilon_1^T b - \rho_{12} \qquad \beta_l(b) = \upsilon_1^T b - \rho_{12}$$

then

$$\tilde{\beta} = \beta(b-\Delta) - \tau_r; \qquad \Delta = \rho_r\begin{bmatrix} -o_1 \\ 0 \end{bmatrix}; \qquad \tau_r = \rho_r(2\rho_{12}+\rho_r).$$

It will prove convenient in the sequel to distinguish regions of freespace – “feasible orthants” – defined by half planes through these points both parallel and orthogonal to the goal lines $\gamma_{\mathrm{i}} = 0$. Accordingly, define

$$\nu_{u_i}(b) \stackrel{\triangle}{=} e_j^T(-b+u_i) \qquad \nu_{li}(b) \stackrel{\triangle}{=} e_j^T(b-l_i) \tag{11}$$

Now take

$$H_{u_i}^+ = \nu_{u_i}^{-1}(0,\infty] \cap \tilde{B}_u; \qquad H_{u_i}^0 = \nu_{u_i}^{-1}[0] \cap \tilde{B}_u;$$
$$H_{u_i}^- = \nu_{u_i}^{-1}[-\infty, 0) \cap \tilde{B}_u; \tag{12}$$

and similarly for $\tilde{B}_l$.

Finally, define Π_i to be the set projection, $\Pi_i(B) = \{b_i : b \in B\}$.

3.3 Analysis

We show in this section that the reaction set – the zero set of the “self”-gradient which takes the form

$$D_i\tilde{\varphi}_i = \frac{\gamma_i^{k-1}}{\tilde{\beta}^2}\,\tilde{\xi}_i \text{ where } \tilde{\xi}_i(b) = k\tilde{\beta} - \gamma_i D_i\,\tilde{\beta} \tag{13}$$

is the graph of a function – when restricted to each $\tilde{B}_u$ and $\tilde{B}_l$ respectively. The proofs of all but the most central of these results are presented in [27].

Lemma 1: *The zero set* $\tilde{\Xi}_i = \tilde{\xi}_i^{-1}[0]$ *is an hyperbola both of whose distinct branches intersect transversally the free-space boundary at* u_i *and* l_i *respectively.*

Lemma 2: *The branches of* $\tilde{\Xi}_i$ *both admit parametrizations* g_{iu} *and* g_{il} *by* b_j.

Taken together, these observations lead to the following summary.

Proposition 1: *The reaction set for part i consists of a single connected curve in each component* $\tilde{B}_u$ *and* $\tilde{B}_l$ *that intersects the boundary at exactly points* u_i *and* l_i*, respectively.*

Lemma 3: *The reaction set for part i is, when restricted to the closed freespace* $\tilde{B}_l$ *or* $\tilde{B}_u$*, parametrized by a piecewise smooth implicit function – that is there exists a piecewise smooth and continuous scalar valued map* r_i *such that*

$$\{b \in \tilde{B}_l : D_i\tilde{\varphi} = 0\} \equiv \{b \in \tilde{B}_l : b_i = r_i(\bar{b}_i)\}$$

and similarly for $\tilde{B}_u$.

Proof: Let us first consider the case $\tilde{B}_l$. First we show that the reaction set in $\tilde{B}_l$ is a graph of some function and next we will exhibit the function explicity. Based on lemma 6 presented in reference [27], the following holds:

(i) $H_{l_1}^- \cap \Xi_1 = \emptyset$; $H_{l_2}^+ \cap \Xi_2 = \emptyset$;
(*ii*) $H_{l_i}^0 \cap \gamma_i^{-1}[0] \cap \tilde{\Xi}_i = \{l_i\}$
(*iii*) $H_{l_1}^+ \cap \gamma_1^{-1}[0] = \emptyset$; $H_{l_2}^+ \cap \gamma_2^{-1}[0] = \emptyset$

Thus, each constituent open half space $H_i^{\mp}$ of $\tilde{\beta}_l$ includes either $\gamma_i^{-1}[0]$ or $\tilde{\Xi}_i$ but not both. Since lemma 1 shows that $\tilde{\Xi}_i$ has only one branch in $\tilde{B}_l$ $\tilde{B}_u$, this demonstrates that the reaction set itself has one branch. Note that the branches of $\tilde{\Xi}_i$ and $\gamma_i^{-1}[0]$ join at l_i or (u_i) as shown in lemma 5 presented in reference [27] and $\Pi_j(\gamma_i^{-1}[0] \cap \tilde{\Xi}_i) = \mathrm{B}_j$. Thus, the reaction set is the graph of some continuous function r_{i_m} defined on B_j – which can be constructed as follows:

$$r_{1_m}(b_2) = \begin{cases} d_1 & \text{if } b_2 \in \Pi_2(H_{m_1}^-) \\ g_{1m}(b_2) & \text{otherwise} \end{cases}$$

$$r_{2_m}(b_1) = \begin{cases} d_2 & \text{if } b_1 \in \Pi_1(H_{m_2}^-) \\ g_{2m}(b_1) & \text{otherwise} \end{cases}$$

To see that r_{i_m} is piecewise smooth, observe that each branch is differentiable. To see that r_{i_m} is continuous, first consider part 1. Take $b_2 \in B_2$. For $b_2 \in \Pi_2(H_{m_1}^- \cap H_{m_1}^+)$, the result follows from the fact that it is differentiable at b_2. If $b_2 \in \Pi_2(H_{m1}^0)$, then $b_2 = e_2^T l_1$ by definition. Then $\lim_{b_2 \to (e_2^T l_1)^+} r_{1_m}(b_2) = d_1$ and $\lim_{b_2 \to (e_2^T l_1)^-} r_{1_m}(b_2) = g_{1m}(e_2^T l_1)$. Using proposition 1, we know $g_{1_m}(e_2^T l_1) = d_1$. Hence the result. □

We can now define the discrete map governing the motion of each player i as $r_{i_l} \circ r_{j_l}$.

Proposition 2: *When restricted to* $\tilde{B}_l$, $r_{il} \circ r_{j_l}(b_i) = d_i$.

Proof: First consider part 1 and write r_{j_l} explicitly in $r_{1_l} \circ r_{2_l}$:

$$r_{1_l} \circ r_{2_l}(b_1) = \begin{cases} r_{1_l}(d_2) & \text{if } b_1 \in \Pi_1(H_{l_2}^+) \\ r_{1_l}(g_{2_l}(b_1)) & \text{otherwise} \end{cases} \tag{14}$$

Noting that $d_2 \in \Pi_2(H_{l_1}^-)$, write r_1, explicitly in eq. 14,

$$r_{1_l} \circ r_{2_l}(b_1) = \begin{cases} d_1 & \text{if } b_1 \in \Pi_1(H_{l_2}^+) \\ d_1 & \text{if } g_{2_l}(b_1) \in \Pi_2(H_{l_1}^-) \\ g_{1_l}(g_{2_l}(b_1) & \text{if } g_{2_l}(b_1) \in \Pi_2(H_{l_1}^0 \cap H_{l_1}^+) \end{cases} \tag{15}$$

Now $\Xi_2 \cap H_{l_2}^+ = \emptyset$ and $H_{l_1}^+ \subset H_{l_2}^+$ implies that $\Xi_2 \cap H_{l_1}^+ = \emptyset$, which implies that $g_{2_l}(b_1) \in \Pi_2(H_{l_1}^-)$ which then implies $g_{1_l}(g_{2_l}(b_1)) = d_1$. Thus, $r_{1_l} \circ r_{2_l}(b_1) = d_1$. Similar reasoning can be used to show $r_{2_l} \circ r_{1_l}(b_2) = d_2$. □

It can be seen that each player reaches its destination in at most two moves – depending on their relative configurations and the first player.

Let us now consider the case for B_u. It will prove convenient to establish the g_{i_u} are increasing functions.

Lemma 4: $\|g'_{i_u}\| > 1 + \dfrac{1}{k-1}$.

Let us also define intervals of B_i – seperated by points $e_j^T u_j$ and $h_i = g_{j_u}^{-1}(e_i^T u_i)$ as:

$$G_i^+ = \{b_i : \sigma_i b_i \geqslant \sigma_i h_i\}; \qquad G_i^0 = \{b_i : \sigma_i e_j^T u_j < \sigma_i b_i < \sigma_i h_i\};$$

$$G_i^- = \{b_i : \sigma_i b_i \leqslant \sigma_i e_j^T u_j\}; \quad (16)$$

and note that they yield a partition of $\tilde{B}_i$ by showing that $\sigma_i e_j^T u_j < \sigma_1 h_i$. First, let it be remarked that since g_{i_u} is an increasing function and $\sigma_i e_j^T u_j < \sigma_i e_i^T u_i$, $g_{j_u}^{-1}(\sigma_i e_j^T u_j) < g_{j_u}^{-1}(\sigma_i e_i^T u_i)$. As $g_{j_u}^{-1}(\sigma_i e_j^T u_j) = d_j$, it then follows that $\sigma_i e_j^T u_{j\text{-}} < \sigma_i h_i$. Finally, it is easy to show that $G_i^- = \Pi_i(H_{uj}^x)$ where

$$x = \begin{cases} + & \text{if } i = 1 \\ - & \text{if } i = 2 \end{cases} \quad (17)$$

The following proposition shows that the reaction set is the graph of a piecewise smooth and continuous function in $\tilde{B}_u$.

Proposition 3: *When restricted to* $\tilde{B}_u$,

$$r_{i_u} \circ r_{j_u}(b_i) = \begin{cases} d_i & \text{if } b_i \in G_i^+ \\ g_{iu} \circ g_{j_u}(b_i) & \text{if } b_i \in G_i^0 \\ g_{iu}(d_j) & \text{if } b_i \in G_i^- \end{cases}$$

Proof: Write r_{j_u} explicitly in $r_{i_u} \circ r_{j_u}$ explicitly as:

$$r_{i_u} \circ\ r_{j_u}(b_i) = \begin{cases} r_{i_u}(d_j) & \text{if } \sigma_i b_i \in \Pi_i(H_{m_j}^x) \\ r_{i_u}(g_{j_u}(b)) & \text{otherwise} \end{cases} \quad (18)$$

Now write r_{i_u} explicitly in eq. 18

$$r_{i_u} \circ\ r_{j_u}(b_i) = \begin{cases} g_{i_u}(d_j) & \text{if } b_i \in \Pi_i(H_{u_j}^x) \\ d_i & \text{if } g_{j_u}(b_i) \in \Pi_j(H_{u_i}^y) \\ g_{i_u} \circ g_{j_u}(b_i) & \text{otherwise} \end{cases} \quad (19)$$

where y is understood to be opposite of x. First note that $\Pi_i(H_{u_j}^x) = G_i^-$. Secondly note $g_{j_u}(b_i) \in \Pi_j(H_{u_i}^y)$ implies that $\sigma_i g_{j_u}(b_i) > \sigma_i e_i^T u_i$ and since g_{j_u} is an increasing function, $\sigma_i b_i > \sigma_i g_{j_u}^{-1}(e_i^T u_i)$ which then implies that $b_i \in G_i^+$. Hence, the result. Finally, since $r_{i_u} \circ r_{j_u}$ is the composition of two piecewise smooth and continuous functions, it itself is also so. □

Proposition 4 in reference [27] shows that a discrete map having the form of $r_{i_u} \circ r_{j_u}$ has no limit cycles.

Corollary 1: $r_{i_u} \circ r_{j_u}$ *has no periodic orbits other than fixed points.*

Proof: Since $r_{i_u} \circ r_{j_u}$ is of the same form as that given in proposition 4, it has no periodic orbits other than its fixed points.

Remark: Our computations show that we have case (i) of the proposition.

3.4. Example case

In this example, the desired destination is arbitrarily set to $d = (5, -1)$. Figure 15 shows the configurations spaces for the edogenous case. The reaction sets are as shown in Figure 16. The discrete maps governing the motion of each player are then as shown in Figure 17.

4. CONCLUSION

We have argued that endogenous assembly – assembly of parts into a goal configuration by a robot that inhabits the same workspace – is a generalization of the exogenous assembly problem explored by the second author in a previous paper [1]. This paper proposes a noncooperative game-theoretic formulation for endogenous assembly

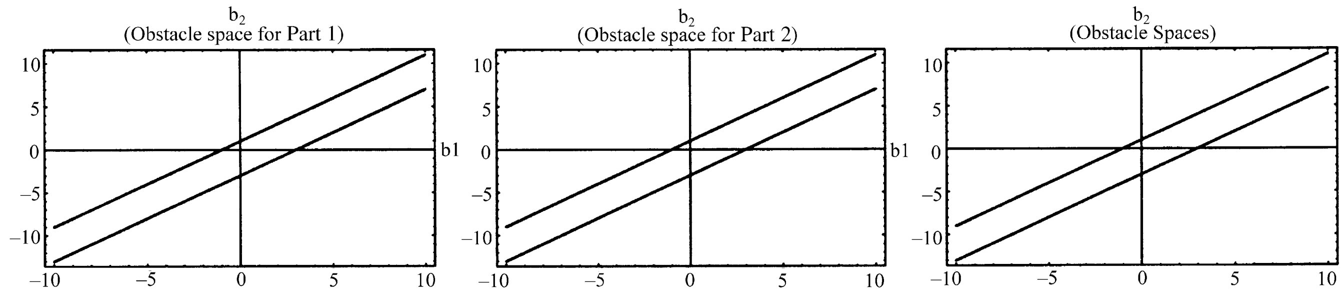

Fig. 15. Configuration spaces: The case when $O_1 = -pr$ and $O_2 = pr$.

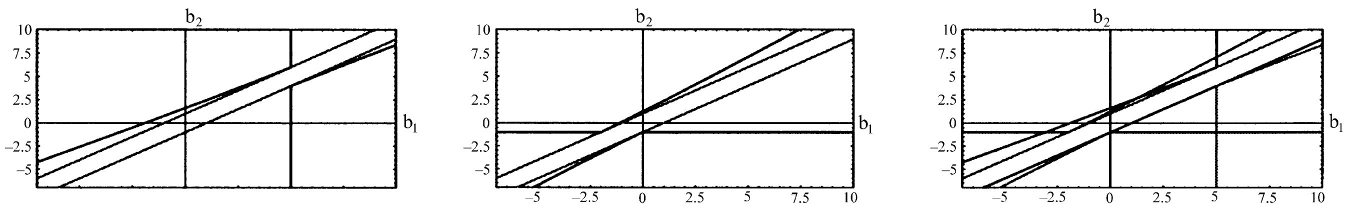

Fig. 16. Reaction sets.

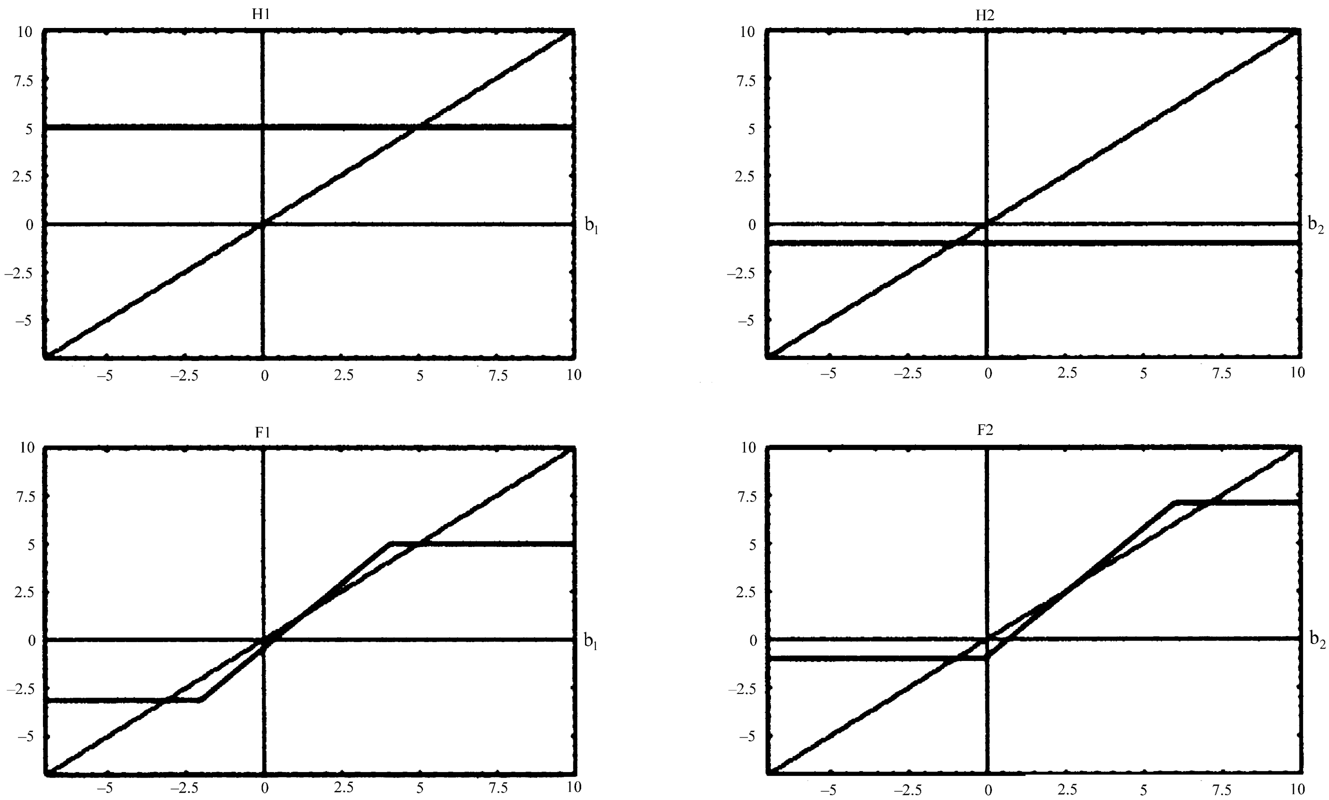

Fig. 17. Reaction functions.

problems that seems to also effectively generalize the perspective of a cooperative game introduced in that earlier work.

Including the robot as a body with physical extent within the workspace presents each robot mated part with a different free configuration space geometry (and, likely, topology) defined by the remaining ungrasped parts. We have constructed a set of distinct artificial potential functions φ_i, each encoding a navigation procedure for the robot mated part moving among these obstacles. We develop from these constructions an algorithm for choosing the next part for the robot to mate with and moving that mated pair against the backdrop of the stationary unmated remaining parts.

We present the analysis of convergence of this algorithm for the simplest instance of 1 DOF 2-body sphere assemblies within this framework along with an extensive simulation study of its implementation in a 2 DOF workspace. We have started to study the convergence properties of the noncooperative game interpretation of exogeneous assembly to 2 DOF N-body case. Our simulations indicate that in the case of feasible assembly, the assembly is always successfully completed where the number of switches made is dependent on the indexing scheme used. It remains to be proved analytically that the scheme ensures the completion of the assembly task for all arbitrary initial configurations in the case of feasible assembly or its termination otherwise.

ACKNOWLEDGEMENTS

The first author has been supported in part by the Turkish Scientific Research Agency TÜBİTAK under grants BAYG-1994 and MISAG 65-1995. The second author has been supported in part by the NSF under grant 9510673.

References

1. D.E. Koditschek "An approach to autonomous robot assembly" *Robotica* **12** part 2, 137–155 (1994).
2. T. Lozano-Peréz "Handey: A robot system that recognizes, plans and manipulates" *Proceedings of IEEE Int. Conference on Robotics and Aut.* (1987) pp. 843–849.
3. T. Lozano-Peréz and R.H. Wilson "Assembly sequencing for arbitrary motions" *Proceedings of IEEE Int. Conference on Robotics and Aut.* (1993) pp. 527–532.
4. L.L. Whitcomb, D.E. Koditschek and J.B.D. Cabrera "Toward the automatic control of robot assembly tasks via potential functions: The case of 2D sphere assemblies" *Proceedings of IEEE Int. Conference on Robotics and Aut.* (1992) pp. 2186–2191.
5. E. Rimon and D.E. Koditschek "Exact robot navigation using artificial potential fields" *IEEE Transactions on Robotics and Automation.* **8** (5), 501–518 (1992).
6. R. Alami, T. Simon and J. Laumond "A geometrical approach to planning manipulation tasks: The case of discrete placemements and grasps" *Preprint of 5th International Symposium of Robotics Research.* (1989) pp. 113–123.
7. T. Dean and M. Wellman *Planning and Control* Morgan-Kaufmann, USA (1991).
8. R. Brooks "New approaches to robotics" *Science* **254** 1227–1232 (1991).
9. R.R. Burridge, A. Rizzi and D.E. Koditschek "Sequential composition of dynamically dexterous robot behaviors" *Int. J. Robotics Research (In Press).*
10. L.S. Homem de Mello and A.C. Sanderson "A correct and complete algorithm for the generation of mechanical assembly sequences" *IEEE Transactions on Robotics and Automation.* **7** (2) 228–240 (1991).
11. L.S. Homem de Mello and A.C. Sanderson "Two criteria for the selection of assembly plans: Maximizing the flexibility of sequencing the assembly tasks and minimizing the assembly

time through parallel execution of assembly tasks” *IEEE Transactions on Robotics and Automation* **7** (5) 626–633 (1991).

12. S.G. Kaufman, R.H. Wilson, R.E. Jones, T.L. Calton and A.L. Ames “The archimedes 2 mechanical assembly planning system” *Proceedings of IEEE Workshop and Robots and Automation.* (1996) **Vol. 4** pp. 3361–3408.
13. C.P. Tung and A.C. Kak “Integrating sensing, task planning and execution” *Proceedings of IEEE Int. Conference on Robotics and Aut.* (1994) **Vol. 3** pp. 2030–2037.
14. D.M. Lyons “Representing and analyzing action plans as networks of concurrent processes” *IEEE Transactions on Robotics and Automation.* **9**(3) 241–256 (1993).
15. Yangmin Li “Hybrid control approach to the peg-in-hole problem” *IEEE Robotics and Automation Magazine.* **4**(2) 52–60 (1997).
16. Brenan J. McCarragher, Geir Hovland, Pavan Sikka, Peter Aigner and David Austin “Hybrid dynamic modeling and control of constrained manipulation systems” *IEEE Robotics and Automation Magazine.* **4**(2) 27–45 (1997).
17. T. Başar and G.J. Olsder *Dynamic Noncooperative Game Theory* Academic Press, New York, (1982).
18. S. Li and T. Başar “Distributed algorithms for the computation of noncooperative equilibria” *Automatica.* **23**(4) 523–533 (1987).
19. H.I. Bozma and J.S. Duncan “Noncooperative games for decentralized integration architectures in modular systems” *Proceedings of IEEE Workshop and Robots and Intelligent Systems* (1991) **Vo.l 3** pp. 1221–1228.
20. I. Bozma and J. Duncan “A game-theoretic approach to integration of modules” *IEEE Transactions on PAMI* **16**(11) 1074–1086 (1994).
21. D.E. Koditschek “Task encoding: Toward a scientific paradigm for robot planning and control” *Robotic and Autonomous Systems* **9**, 5–39 (1992).
22. A.M. Bloch, M. Reyhanoglu and N.H. McClamroch “Control and stabilization of nonholonomic dynamic systems” *IEEE Transactions on Automatic Control.***37**(11) 1746–1757 (1992)
23. R.W. Brockett “Asymptotic stability and feedback stabilization” *Differential Geometric Control Theory* (Birkhauser, Switzerland, 1983) pp. 181–191.
24. D.E. Koditschek and E. Rimon “Robot navigation functions on manifolds with boundary,” *Advances in Applied Mathematics* 412–442 (1990).
25. H.I. Bozma, C.S. Karagoz, and D.E. Koditschek “Assembly as a noncooperative game of its pieces: The case of endogenous disk assemblies” *Proceedings of IEEE International Symposium on Assembly and Task Planning* (1995) pp. 2–8.
26. M.W. Hirsch and S. Smale *Differential Equations Dynamical Systems and Linear Algebra* Academic Press, New York (1974).
27. H.I. Bozma, and D.E. Koditschek “Assembly as a noncooperative game of its pieces: Analysis of 1d sphere assemblies” *Technical Report CSE-TR-372-98,* (EECS, The University of Michigan, 1998) An electronic copy is obtainable from http://www.eecs.umich.edu/home/techreports/cse98 html.