



Published in final edited form as:

Int J Rob Res. 2013 February 1; 32(2): 218–246. doi:10.1177/0278364912467486.

Planning and Control for Microassembly of Structures Composed of Stress-Engineered MEMS Microrobots

Bruce R. Donald^{‡,§,¶,*}, Christopher G. Levey^{||}, Igor Paprotny^{¶,**}, and Daniela Rus^{††}

[‡]Department of Computer Science, Duke University, Durham, NC, USA

[§]Department of Biochemistry, School of Medicine, Duke University Medical Center, Durham, NC, USA

[¶]Duke Institute for Brain Sciences, Duke University Medical Center, Durham, NC, USA

^{||}Thayer School of Engineering, Dartmouth College, Hanover, NH, USA

^{**}Berkeley Sensor and Actuator Center, University of California, Berkeley, CA, USA

^{††}Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, USA

Abstract

We present control strategies that implement planar microassembly using groups of stress-engineered MEMS microrobots (MicroStressBots) controlled through a single global control signal. The global control signal couples the motion of the devices, causing the system to be highly underactuated. In order for the robots to assemble into arbitrary planar shapes despite the high degree of underactuation, it is desirable that each robot be independently maneuverable (independently controllable). To achieve independent control, we fabricated robots that behave (move) differently from one another in response to the same global control signal. We harnessed this differentiation to develop assembly control strategies, where the assembly goal is a desired geometric shape that can be obtained by connecting the chassis of individual robots. We derived and experimentally tested assembly plans that command some of the robots to make progress toward the goal, while other robots are constrained to remain in small circular trajectories (closed-loop orbits) until it is their turn to move into the goal shape.

Our control strategies were tested on systems of fabricated MicroStressBots. The robots are 240–280 $\mu\text{m} \times 60 \mu\text{m} \times 7\text{--}20 \mu\text{m}$ in size and move simultaneously within a single operating environment. We demonstrated the feasibility of our control scheme by accurately assembling five different types of planar microstructures.

1 Introduction

Microrobotics systems have the potential to enable applications in many different areas including medical science (Dario et al., 1992), surveillance (Kahn et al., 1999), and assembly (Popa & Stephanou, 2004). Our goal is to develop a microrobotic system capable of coordinated self-assembly. In this paper we present a multi-microrobotic system composed of several stress-engineered MEMS microrobots (MicroStressBots), together with planning and control strategies for directing these robots to assemble into planar shapes using a single, global, control signal. In (Donald et al., 2006), we described a globally-controllable 240 $\mu\text{m} \times 60 \mu\text{m} \times 10 \mu\text{m}$ mobile stress-engineered MEMS microrobot (MicroStressBot). This robot, although

*Corresponding author: P.O. Box 90129, Department of Computer Science, Duke University, Durham, NC 27708, USA. brd +ijrr12@cs.duke.edu.

complex to fabricate, contains only simple electromechanical 2-bit memory and logic. Such simplicity is common in other microrobotic systems as well (e.g., (Yesin et al., 2006; Frutiger et al., 2008).) Control of several such simple devices is desirable, but presents a significant challenge when only a single, global control signal can be used to control all the devices: the resulting system is highly underactuated. In this paper we present algorithms enabling independent control of several MicroStressBots, and directing the robots (also referred to throughout this paper as *devices*) as components that follow complex paths to assemble into larger, planar structures.

Our control scheme fits within the paradigm of *global control, selective response* (GCSR), where the individual devices are controlled by exploiting *fabricated* differences in their response to a broadcast-type control signal (Donald, 2007; Donald et al., 2008b). The GCSR paradigm enables the control of small and simple devices with their individual size approaching the scale-limitations of the fabrication processes. Until microfabrication technology advances to the point where one can embed on-board power and complex control on micro- or nanorobots, GCSR allows for *scalable* implementation of multi micro- and nanorobotic systems.

In this paper, we show that designing microrobots that move along different trajectories in response to voltage levels in the broadcasted control signal allows us to independently maneuver multiple microrobots to assemble into desired planar structures. Although the robots move in parallel, our algorithms decouple their motion by reducing the parallel motion of n microrobots to parallel motion of two robots, followed by sequential motion of single devices. Each robot either orbits, or pursues a trajectory towards the goal. Changes in the global control signal cause the robots to selectively switch (based on their differentiated physics) between orbiting trajectories and goal trajectories.

We implemented microassembly using GCSR by fabricating groups of MicroStressBots (240–280 $\mu\text{m} \times 60 \mu\text{m} \times 7\text{--}20 \mu\text{m}$ in size) with different steering-arm designs. A scanning-electron micrograph of one such microrobot is shown in Figure 1.a, while four microrobots in the process of assembling a shape are shown in Figure 1.b. Each robot consists of an untethered scratch-drive actuator (USDA) (Donald et al., 2003) that provides forward motion, and a steering-arm actuator (i) that determines whether the robot moves in a straight line or turns. Each device receives both power and a control signal remotely without the use of restrictive wires or tethers, using a capacitive coupling with an underlying engineered substrate. The substrate comprises the robot's operating environment. The robot is non-holonomic, and although it is not small-time locally controllable, in (Donald et al., 2006) we proved that interleaving of straight-line and turning motions is sufficient to ensure global controllability.

The steering-arm can be either raised to cause the robot to move in a straight line, or lowered to cause the robot to turn. We call the position of the steering-arm actuator the *hysteresis state* of the microrobot (arm raised, hysteresis state = 0; arm lowered, hysteresis state = 1). A system of n MicroStressBots contains 2^n possible hysteresis states (all 2^n combinations of n steering-arms being raised or lowered). However, in general, not all of the 2^n hysteresis states are electromechanically accessible, depending on the physical design of the steering arm actuators. For example, the steering arms may be designed such that we can raise the steering arms for some devices *only* while simultaneously raising the steering arms of certain other robots. Such couplings can constrain the number of reachable hysteresis states to be significantly less than 2^n . In this paper we show that during microassembly, it is sufficient to use only $n + 1$ hysteresis states to control n robots. While the control algorithm that uses $n + 1$ hysteresis states has poor time complexity, it requires the smallest number of independent voltage levels in the control signal. This is significant, since the number of

independent voltage levels in the control signal, called the *control bandwidth*, is the scarcest resource in our microrobotic system.

Details of fabrication, designs, and testing were reported in (Donald et al., 2008b). In this work, we present the control strategies and algorithms that allow us to implement simultaneous control of multiple microrobots to perform assembly of planar structures. These algorithms are crucial to the system, and are not covered in (Donald et al., 2008b). Enabling these control algorithms are new theorems, proven below, that minimize the required control voltage bandwidth; these were essential for controlling multiple untethered microrobots to move and assemble independently.

The major contributions of this paper can be summarized as follows:

- **Control Signal Engineering for Sub-linear Control Voltage Bandwidth:** Design of the *control primitives* i.e. waveforms encoding the hysteresis states of the MicroStressBots to minimize the number of independent control voltage levels. A key contribution of this paper is novel engineering of control primitives that requires only $2\sqrt{n}$ voltage levels to control n microrobots. This is a dramatic improvement over our previously proposed requirements of $2n$ voltage levels.
- **Heuristic Control Strategies:** A comprehensive description of the two-stage heuristic control strategy based on motion planning under uncertainty. The strategy maneuvers the microrobots towards the goal while reducing the accumulating control error. The control heuristic includes compliance and collision avoidance, and, as we show in the presented experimental results, allows us to generate virtually defect-free assemblies.

The remainder of this paper is organized as follows. Sec. 2 discusses related work. In Sec. 3 we describe the stress-engineered MEMS microrobots (MicroStressBots) used in our work. In Sec. 4 we show how a control signal can be engineered to maneuver our microrobots independently despite the coupling of their motion. Sec. 5 describes a heuristic to construct the assembly plan, which allows us to determine the target configurations of the individual microrobots within the assembling shape. In Sec. 6 we describe how we used the control matrix engineered in Sec. 4, to implement a two-stage Error-Detection and Recovery (EDR) control strategy, allowing us to control the robots from their initial configuration to their final configuration within the assembling shape specified by the assembly plan generated in Sec. 5. In Sec. 7 we show the experimental results with fabricated devices to demonstrate the control strategies, and present the results of the assembly of several types of planar structures. Finally, in Sec. 8 we conclude by discussing the potential for application of our control algorithms to a broader class of underactuated microrobotic systems. A preliminary and abbreviated report of the ideas in this paper appeared in the workshop WAFR (Donald et al., 2008e).

2 Related Work

2.1 Microrobotics

Small-scaled robotic devices are envisioned by previous authors to have numerous applications within biomedicine (Dario et al., 1992), surveillance (Kahn et al., 1999), or microassembly (as we show in (Donald et al., 2008b) and this paper). Small, often completely autonomous mobile robotic devices, with the size on the order of centimeters (Byrne et al., 2002; Stefanini et al., 2006; Jianghao et al., 2007; Churaman et al., 2011) are called *miniature robots*. Miniature robots containing microfabricated components (Kladitis & Bright, 2000; Hollar et al., 2003; Linderman & Bright, 2000; Mohebbi et al., 2001) are called MEMS robots. The components of MEMS robots are often manufactured separately and then assembled to form complete systems. The size of the MEMS robots ranges from few micrometers to several centimeters.

In (Donald et al., 2006, 2008b) and this paper, we use the term *microrobot* to denote mobile untethered MEMS robots with their dimensions strictly confined within a 1 mm³ cube. Several magnetically-actuated microrobots have been presented in (Yesin et al., 2006; Floyd et al., 2008; Frutiger et al., 2008; Pawashe et al., 2008; Frutiger et al., 2010).

2.2 Multirobotic Systems and Control

There are many examples of *macroscopic* multi-robotic systems that aggregate form or functionality through cooperation or mutual interaction. Self-reconfiguring robots are examples of physically-connected robotic systems that can reconfigure their shape and functionality (Kotay & Rus, 1999; Rus, 1998) to match the task at hand. The self-reconfiguration process has been controlled using distributed algorithms and message-passing (Butler et al., 2002). Multi-robotic cooperation can be also achieved in absence of explicit communication (Donald et al., 1997; Rus et al., 1995; Stilwell & Bay, 1993; Pagello et al., 1999). Specifically, (Donald et al., 1997; Rus et al., 1995) show how implicit communication can be used to coordinate several robots to manipulate larger objects. Coordination of multiple microrobots can be also achieved through the use of local rules (Jadbabaie et al., 2003; Martinoli & Agassounon, 2004; Pallottino et al., 2007).

Most previous work on decentralized multi-robotic control assumes the robots have sufficient onboard hardware resources to receive and process sensory inputs and/or communicate with other devices. However, our microrobots have a much simpler structure, and can only partially demultiplex individual commands transmitted through the global control signal. Bretl (Bretl, 2007, 2012) presented a related theoretical motion planning approach for systems of robots with limited controllability, showing that even simple devices controlled through a global signal can perform useful tasks. Our MicroStressBots are also simple, however, unlike the robots envisioned in (Bretl, 2007, 2012) they are fabricated to *behave* differently during portions of the control waveform. Control strategies presented later in this paper (Sec. 6) exploit this differentiation to maneuver the robots independently toward the goal. We call this control concept *Global Control, Selective Response (GCSR)* (Donald, 2007; Donald et al., 2008b). The problem of controlling the evolution of a dynamical system through a single control signal was proposed by Li and Khaneja (Li & Khaneja, 2005; Li, 2006), originally for controlling spin ensembles in NMR spectroscopy and magnetic resonance imaging (MRI). GCSR is related to the control of inhomogeneous ensembles coined by Li (Li & Khaneja, 2006), with the added difference that the devices that are being controlled in our work are specifically designed and fabricated to have sufficiently differentiated behavior to ensure controllability (as opposed to dynamic systems with *intrinsic* differences in behavior.) Previous work does not show how to *design* device physics for ensemble control, since in NMR the physics (e.g., nuclear spin) is determined by Nature. We use nanofabrication to explicitly design for different device physics to ensure differentiated responses that enable GCSR. We present scalable ways of designing this differentiation in MicroStressBots in Sec. 4. Recently, our group achieved independent control of multiple microrobots using differences in turning-rates, as opposed to transition voltages of the steering arms, to independently maneuver multiple robots on a planar substrate (Paprotny et al., 2012).

A similar class of control problem emerges in sensorless manipulation using vibrational surfaces (Bohringer et al., 2000; Branicky et al., 1999) or distributed manipulation (Böhringer et al., 1994; Bohringer et al., 1999; Suh et al., 1999), where *global* force-fields can be designed to geometrically filter parts differentiated by their geometry (similar to (Berretty et al., 1998)), implementing either GCSR or Ensemble Control. It can be argued however whether a global force-field applied through hundreds or even thousands (Böhringer et al., 1994) of actuators can be considered a single control signal.

2.3 Microassembly

Microassembly has been introduced in the literature as an application for cooperating microrobotic systems (Popa & Stephanou, 2004). Microassembly can be performed using pick-and-place macroscale robotic micromanipulators (Dechev et al., 2004; Skidmore et al., 2004), distributed manipulation (Böhringer et al., 1994; Bohringer et al., 1999; Suh et al., 1999), or via parallel but less controllable self-assembly (Whitesides & Grzybowski, 2002; Rothmund, 2000; Liang et al., 2004)

Pick-and place assembly using macroscale robotic micromanipulators has been shown to successfully assemble several microstructures (Skidmore et al., 2004). However, the actuator mass is several orders of magnitude larger than its payload. This restricts the speed of this assembly method compared with microactuator-driven microassembly described below. (For example, we have demonstrated virtually instantaneous acceleration and de-acceleration of our USDAs to speeds up to 1.8 mm/sec, due to their very low mass (Donald et al., 2003).) Magnetically-actuated microrobots have recently been shown to be able to manipulate microscale objects (Frutiger et al., 2010; Pawashe et al., 2011).

In distributed manipulation (Böhringer et al., 1994; Bohringer et al., 1999; Suh et al., 1999), arrays of micro-manipulators are embedded into a surface (so called active surface) and can be used independently to translate, orient and assembled parts. Virtual force fields, created by synchronous motion of actuators in an active surface, have been shown to orient and filter parts through sensorless manipulation. Optical manipulation using a pixelized light source (Ohta et al., 2007) is another form of distributed manipulation.

Self-assembly (SA) is an increasingly viable autonomous method for assembling micro- and nanoscale components (Gracias et al., 2000). In some instances, the complexity of the generated shapes is quite astounding (see (Rothmund, 2006)), however, unless one uses DNA with base-pair sequence matching (Seeman, 1998), the shape complexity is often inversely proportional to the yield due to difficulties in designing unique local energy minima for the various docking configurations. An interesting direction in SA is Directed Self-Assembly (DSA), which implies some control over the assembly process. This is done either through the use of optimization and distribution of docking sites (for example (Saeedi et al., 2006)) (also called templated self-assembly), or by actively changing the docking site affinity (Onoe et al., 2004). The latter approach has great potential for generation of complex shapes (Klavins, 2004) with high yield.

In this work, we implement microassembly by maneuvering multiple microrobots to dock together and form larger structures. This type of microassembly was first presented by our group in (Donald et al., 2008b), however related multi-microrobotic assembly using magnetically-actuated robots was recently presented in (Diller et al., 2011). The magnetic assembly (and disassembly) described in (Diller et al., 2011) relies on selective electrostatic clamping using the substrate as a temporary anchor to achieve independent control. A related multi-robotic control mechanism that does not rely on a specialized substrate was recently published in (Floyd et al., 2011). Similar to the mechanism presented in (Donald et al., 2008b) and this work, (Floyd et al., 2011) relies on engineered differences in the response of the microrobots to the same global control signal. The robots in (Floyd et al., 2011) are differentiated by how their entire chassis interact with the global magnetic field. However, large design differences were necessary to ensure significantly different motion to enable independent control. The need for such large design differences limited the number of robots that could be controlled simultaneously. In contrast, the robots presented in (Donald et al., 2008b) and this work *decode* a sequence of control pulses using the one-bit onboard memory stemming from the steering-arm hysteresis gap. As we show below, this allows us to greatly increase the number of simultaneously-controllable MicroStressBots. Furthermore, the mobility of the microcomponents in our assembly scheme is not restricted by discretization of

the substrate, as in the case of (Diller et al., 2011) or distributed manipulation. In contrast with SA, our microassembly scheme relies on *intersecting trajectories*, rather than component affinity and energy minimization to promote structure aggregation. This permits the control of defect formation through collision avoidance and non-intersecting trajectories. However, while our robots move in parallel, our assembly algorithms use a largely sequential order, and lack the vast parallelism present in biological SA systems.

3 Stress-engineered MEMS Microrobot (MicroStressBot)

All the control and assembly algorithms presented in this paper are implemented using groups of parallel-actuated stress-engineered MEMS microrobots (MicroStressBots) (Donald et al., 2006). A MicroStressBot has two actuated internal degrees of freedom (DOF); an untethered scratch-drive actuator (USDA) (Donald et al., 2003) that provides forward locomotion, and a steering-arm actuator that determines whether the robot moves in a straight-line or turns. The steering-arm consists of a cantilever beam with a circular pad and a $.75\text{--}1.2\ \mu\text{m}$ deep dimple. The cantilever beam is curved out-of-plane using a stress-engineering process (Donald et al., 2006), which determines the deflection of the steering arm. The microrobot operates on fields of zirconia-insulated interdigitated electrodes. When a voltage is applied across these electrodes, the electrodes and the conductive microrobot chassis form a capacitive circuit inducing an electric potential on the microrobot body. This voltage (waveform) is varied over time to provide power to the untethered scratch-drive actuator and to control the state of the steering-arm. This waveform is called the *control waveform*. Figure 2 illustrates one cycle of the control waveform. The waveform is divided into a *control cycle*, containing $j - 1$ *control pulses* ($V_{a,j}$), that sets the state of the steering-arm actuator, and a *power-delivery cycle* that provides power to the scratch-drive actuator. USDA. The power delivery cycle consisting of 250 stepping pulses, alternating between a maximum (V_s) and a minimum (V_b) (the subscripts a , s , and b , are abbreviations of arm, step, and bias). In order for the USDA to actuate reliably, $V_s > V_{flx}$ and $V_b < V_{rel}$, where V_{flx} is the minimum voltage at which the backplate of the USDA obtains enough curvature to produce a forward step, while V_{rel} is the maximum voltage at which that curvature is sufficiently relieved to generate forward motion. The relationship between V_{flx} and V_{rel} is described in more detail in (Akiyama & Shono, 1993; Linderman & Bright, 2000; Donald et al., 2006).

Similar to an electrostatic cantilever-beam (Nathanson et al., 1967), the steering-arm of the MicroStressBot has two distinct voltage-levels at which it abruptly changes state. These voltages are called the *transition voltages*. While the state of the microrobot includes the state of the steering arm and the state of the scratch-drive actuator, for the purpose of this section it suffices to consider only the states of the steering-arm actuators, which we call the robots' *hysteresis states*. Consequently, a single actuated MicroStressBot can be in one of only two hysteresis states; the steering-arm can be either raised (0) or lowered (1). When the voltage supplied to the robot reaches the steering-arm's *snap-down* transition voltage (V_d), the arm is pulled into contact with the substrate. When the voltage is reduced past the *release* transition voltage (V_u), the arm is released from the substrate. V_u is less than V_d because the electrostatic attraction is a strongly non-linear function of the gap between the steering arm and the substrate. The transition voltages are a function of the design of the individual steering-arm actuators: for example, a smaller air gap or larger steering pads primarily reduce V_d and V_u , respectively. Microrobots with identical steering-arms are classified as belonging to the same *microrobot species*. The difference between the snap-down and release voltage of a steering-arm is called the *hysteresis gap*.

The microrobot moves forward during the power-delivery cycle of the control waveform (Akiyama & Shono, 1993) with an average step size of 10 – 20 nm, such that many stepping pulses are required to produce micron-scale displacement. We have successfully operated our

microrobots using stepping frequencies as high as 20 kHz, achieving speeds of up to 200 $\mu\text{m/s}$, while the USDAs in isolation have been shown to travel at stepping frequencies of up to 100 kHz with speeds up to 1.8 mm/s. Applying the power-delivery cycle while the arm is lowered causes the microrobot to turn. As the robot turns, a portion of the arm remains in flat contact with the substrate, providing a resistive force. This force generates a moment on the scratch-drive actuator, causing it to turn. The robot turns around a fixed radius of curvature, which is defined by the design of the steering-arm actuator and the voltage V_b of the stepping cycle. If the power-delivery cycles are applied while the arm is raised, the robot moves in straight line. Both turning and straight line trajectories are shown in Fig. 3.

For a single robot, a control waveform defined by a specific voltage triple (V_a, V_b, V_s) (see Fig. 2) is called a *control primitive*. A control primitive with a $V_a > V_d$, and $V_b > V_u$ keeps the steering-arm lowered through the stepping cycle, causing the robot to turn. A control primitive with $V_a < V_d$ or $V_b < V_u$ causes the robot to move in straight-line motion. For simultaneous control of n microrobots, the control primitive may contain up to $2n$ control pulses, and has the form $(V_{a,1}, \dots, V_{a,k}, V_b, V_{step})$, where $k \in \{1, \dots, 2n\}$. In both cases, the control primitive defines the control and power-delivery waveform that is supplied to the robots through the operating environment.

The trajectory of a microrobot is a concatenation of straight-line motion and turning, and is generated through the *execution* of a *control sequence*. A control sequence is a sequence of several control primitives, where each primitive is applied for a specific length of time. For example, a control sequence S consisting of two control primitives, P_1 and P_2 , applied for 10 and 20 seconds respectively, is written as $S = (P_1(10), P_2(20))$. We refer to a *nominal* (error-free) microrobot trajectory as the ideal trajectory of a control sequence executed in the absence of the control error. We summarize the technical terms introduced in this section, and Section 4, in Table 1. The design types (NHG, STRING, ESat, and SESat) may appear unfounded. However, we will show that they have provable properties that are crucial for multirobot assembly and are also possible to implement in hardware within the constraints of the microfabrication process. Hence, these design classes are a melding of theoretical and physical constraints.

The kinematics of our robot is illustrated on Fig. 3. The configuration of the robot is given by the vector $q = (x, y, \theta)^T$ in configuration space (C-space (LaValle, 2006)). The configuration of the robot is measured at the point Z_o in the middle of its bushing. The velocity of the robot is $\dot{q} = v(\sin\theta, \cos\theta, \frac{ah}{r})^T$, where $h \in \{-1, 1\}$ and denotes whether the steering arm is on the right or the left side, v is the velocity of the scratch-drive actuator, r is the turning radius and $a \in \{0, 1\}$ is the state of the steering arm (0 = up, 1 = down). The velocity v can be varied by changing the frequency of the stepping pulses, however for the remainder of this paper we will consider v to be a positive constant (positive because the robot can not back up). We define a distance r^* as the turning radius of the most extreme point on the microrobot chassis, i.e., the point farthest away from the center of curvature, Z_c , for the robots turning motion. The radius r^* allows us to define the necessary separation between the orbiting microrobots (see Appendix Sec. C.4). Our robot is not small-time locally controllable: it can only turn one way ($\dot{\theta} = \frac{ah}{r}$) and cannot back up ($v > 0$).

4 STRING Theory: Control Signal Engineering

The control strategies presented in Sec. 6 depend on a set of control primitives that couple the motion of the microrobots in a specific way (later called *STRING control primitives*). In this section (Sec. 4), we show how to design such set of control primitives given the snap-down and release voltages of a group of MicroStressBots. First, we provide formulas for generating

STRING control primitives. Next, we prove two theoretical results; (1) we show that STRING control primitives can be created for any non-degenerate (distinct snap-down and release voltages) system of microrobots, and (2) we show that in certain cases the control signal can be engineered to achieve sub-linear ($O(\sqrt{n})$) complexity of independent voltage levels required to control n microrobots. For example, this last result permits the control of 100 microrobots with only 20 control voltage levels, compared to 200 voltage levels from our previously proposed algorithm ((Donald et al., 2006)) that used Nested Hysteresis-gaps (NHG).

The snap-down and release voltages are a function of the robots design parameters, such as width of the steering arms, diameter of the steering pads, or the height of the steering arm dimple. In Secs. 4.1 and 4.2 we assume that these parameters are set, and the corresponding snap-down and release voltages are given. We refer the interested reader to Appendix F for a list of design parameters used for the systems of microrobots presented in this paper. In Sec. 4.3 we show how to maximize the number of independently controllable MicroStressBots with our control strategies by designing the robots snap-down and release voltages in a specific way.

4.1 The Control Voltage Bandwidth

We now define the concepts of the control voltage bandwidth and the control voltage bandwidth requirement, which will help us to discuss the scalability of our microassembly scheme. For a given system of n microrobots, let $V_{d,i}$ and $V_{u,i}$ denote the snap-down and release voltages of microrobot i . Let V_{Ω} be the breakdown voltage of the operating environment. For the robots to function properly, their snap-down and release voltages must conform to the following constraints:

1. $V_{d,i} < V_{\Omega}$: snap-down voltage cannot exceed the break-down voltage of the operating environment.
2. $V_{d,i} > V_{u,i}$: dictated by the electromechanics of cantilever beams.
3. $V_{rel} > V_{u,i}$ for all i : ensures the microrobot can receive power during all the hysteresis states. Recall that V_{rel} is the voltage at which the backplate of the USDA actuator relaxes, allowing the USDA to take a step forward during the power delivery cycle. If $V_{rel} \leq V_{u,i}$, the USDA will not be able to actuate the microrobot during turning.
4. $V_{flx} \leq \min_i V_{d,i}$: Ensures that the USDA flexes sufficiently to produce forward motion during the power i delivery cycle. Recall that V_{flx} is the minimum voltage at which the backplate of the USDA generates enough flexure to produce a forward step.

We define the *control voltage bandwidth* ξ_i of a MicroStressBot system as the number of independent electromechanically-addressable transition voltage levels of the control signal. ξ_i depends on four parameters: 1) the break-down voltage of the operating environment, V_{Ω} , 2) the inherent variability of the power coupling between the robot and the underlying substrate, 3) the precision of the fabrication process, and 4) the minimum range of voltages required to reliably power the USDA, V_{SDA} ($V_{SDA} = V_{flx} - V_{rel}$).

The variability in the power coupling causes deviations in the potential induced between the steering arm and the substrate, while inaccuracies in the fabrication process cause deviations in ms. Let δ_v be the maximum deviation of the transition voltage manifested during the microrobot operation, determined by these two parameters. We define two transition voltages to be *significantly independent* if they are separated by at least $2\delta_v$. Note that, although in general, $V_{SDA} = V_{flx} - V_{rel}$, it is possible for the stepping pulse to overlap with the lowest snap-down voltage, $\min_{i \in Z_n} V_{d,i}$ where $Z_n = \{1, \dots, n\}$. Consequently, we define V'_{SDA} as the *additional* control voltage gap required by the power delivery cycle to ensure reliable actuation of the USDA, as follows:

$$V'_{SDA} = \begin{cases} 0, & \text{if } V_{flx} - V_{rel} \leq 2\delta_v, \\ V_{flx} - V_{rel} - 2\delta_v, & \text{otherwise.} \end{cases} \quad (1)$$

The control voltage bandwidth of a microrobot system is then $\xi = \left\lfloor \frac{V_{\Omega} - V'_{SDA}}{2\delta_v} \right\rfloor$, assuming V_{flx} and V_{rel} can vary by at least $2\delta_v$ (otherwise $\xi = \left\lfloor \frac{V_{rel}}{2\delta_v} \right\rfloor + \left\lfloor \frac{V_{\Omega} - V_{flx}}{2\delta_v} \right\rfloor$). It tells us how many independent voltage levels are available for control. How much of ξ ; need actually be used for controlling our microrobots is related to the number of accessible hysteresis states. We define the *control voltage bandwidth requirement*, ξ_n , of a n -microrobot system as the number of significantly-independent transition voltage levels necessary to implement microassembly under the control strategy presented in Sec. 6. Clearly, in order to be able to achieve microassembly, it must hold that $\xi_n \geq \xi$. In general, a microrobotic system with fewer accessible hysteresis states has a lower control voltage bandwidth requirement. More specifically, the accessibility of the hysteresis states depends on the relation between the hysteresis gaps of the individual robots.

For example, consider a system of two microrobots, D_1 and D_2 , with steering arms that have *Nested Hysteresis Gaps* (NHG) ((Donald et al., 2006)). Fig. 4(a) shows the relation between the transition voltages for such system. The snap-down and release voltages are shown as circles and rectangles, respectively. Each transition voltage corresponds to an independent voltage level of the control signal (labeled V_a , V_b , V_c and V_d). Fig. 4(b) shows the programming cycles for the four control primitives that access the four hysteresis states (11), (10), (01) and (00) (we assume $V_b = V_{rel}$). More generally, we classify the system of n steering arms, sorted according to ascending $V_{d,i}$, as having NHG when $(V_{d,i} + 2\delta_v < V_{d,j})$ and $(V_{u,i} - 2\delta_v > V_{u,j})$, for all $i < j$. NHG systems can access all 2^n hysteresis states. However each device requires two unique control voltage levels, and so the control voltage bandwidth requirement of this system is $\xi_n = 2n$.

4.2 The STRING Control Matrix

We now describe how to generate control primitives to be used by the control strategies presented in Sec. 6 to implement microassembly using *fewer* control voltage levels than in the case of NHG. The recursive formulas (Eq. (2), p. 11, and Eq. (4), p. 12) generate a set of such control primitives given a set of snap-down and release voltages.

Assume a system of m microrobots. The application of a control primitive will cause either straight-line or turning motion in each of the m robots. A mapping between the control primitives and the motion of the individual microrobots is expressed through a *control matrix* M of size $n \times m$, where each entry $M_{i,j}$ contains the hysteresis state of microrobot j during the application of the control primitive i . The control matrix corresponds to the coupling of the microrobot motion as a function of the control signal, providing a layer of abstraction between the electromechanical functionality of the steering-arm with the motion of the individual devices. An example of the control matrix is shown in Eq. (3) on p. 12. The control strategies presented in Sec. 6 require the control matrix to be structured such that the robots progressively start turning as the control primitives with higher index i are applied.

NHG is sufficient but not necessary, to control multiple devices during assembly. Consider a two-robot system where the hysteresis gaps of the robots are not nested, as shown in Fig. 5(a). In this particular system, only three hysteresis states are electromechanically accessible. The

programming cycles that access all three hysteresis states ((00), (10) and (11)) are shown in Fig. 5(b). Control state (01) can not be accessed, because pulling down the steering-arm of D_2 also pulls down the steering-arm of D_1 , and the steering-arm of D_1 can not be released without also releasing the arm of D_2 .

It is convenient for us to define a lexicographic sorting of the robots, using two keys. In general, an n -microrobotic system, primarily sorted according to ascending values of V_d , and secondarily sorted according to ascending values of V_u , has non-nested hysteresis gaps if $(V_{d,i} > V_{d,j})$ and $(V_{u,i} > V_{u,j})$, for all $i < j$. However, in the case when $V_{d,j} - V_{d,i} < 2\delta_v$ and $V_{u,j} - V_{u,i} < 2\delta_v$, the behavior of robots i and j is indistinguishable, and such two devices cannot be controlled independently. We call such two robots a *degenerate pair*. Let a *STRICtly Non-nested hysteresis Gaps* (STRING) system be a non-nested hysteresis gap system with no degenerate pairs of devices.

Lemma 4.1—An n -robot STRING system has exactly $n + 1$ accessible hysteresis states.

The proof for Lemma 4.1 is provided in Appendix A. We now construct the control primitives and corresponding control matrix that can access the $n + 1$ hysteresis states of a n -robot STRING system. The ordering of the robots is determined by the transition voltages of the steering arms, i.e., the robots must be primarily sorted according to increasing order of $V_{u,i}$ and secondarily sorted according to increasing order of $V_{d,i}$. We construct the control primitive P_j such that it snaps down the arms of devices D_i for $i \leq j$, and releases the arms of devices D_i for all $i > j$. P_j is defined by a control cycle containing two control pulses, $P_j = (V_{a,1}, V_{a,2})$, assume $V_b = V_{rel}$ and $V_s = V_{flx} - V_{d,1}$. Consider the STRING system shown on Fig. 6, where V_a, \dots, V_e represent significantly independent control voltage levels. We define P_j as:

$$P_j = \begin{cases} (V_0, V_0), & \text{if } j=0; \\ (V_{d,j}, V_{u,j+}), & \text{if } j \in Z_{n-1}; \\ (V_{d,n}, V_{d,n}), & \text{if } j=n, \end{cases} \quad (2)$$

where $V_{u,j+} = V_{u,j} + 2\delta_v$. In practice, $V_{u,j+}$ is the next significantly independent release voltage above $V_{u,j}$. Also, note that in order for $V_{d,j}$ to cause *reliable* snap down, it must be δ_v *above* the designed (nominal) $V_{d,j}$ level. Correspondingly, $V_{u,j}$ must be δ_v *below* the designed (nominal) $V_{u,j}$ level to ensure reliable steering arm release.

The first control pulse ($V_{a,1}$) snaps down the steering arms of all the devices D_i , $i \in Z_j$, as well as any devices D_k , $k > j$ with $V_{d,k} = V_{d,j}$. The second control pulse ($V_{a,2}$) releases all the devices D_k , $k > j$ that were snapped down by the first control pulse, because in the case when $V_{d,k} = V_{d,j}$, it must hold that $V_{u,j} < V_{u,j+} - V_{u,k}$. An example control cycle is also shown in Fig. 6.

The $n + 1$ control primitives generated by P_j form a $(n + 1) \times n$ control matrix M . An example of such control matrix for four devices is:

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}. \quad (3)$$

We refer to M as the *STRING control matrix*, the $n + 1$ control primitives contained in M as the *STRING control primitives*, and the $n + 1$ hysteresis states accessible using these control primitives as the *STRING hysteresis states*. Note that because adding a new control state to a

STRING system requires the addition of another independent voltage level (per Lemma 4.1), the control bandwidth requirement for a STRING system is $\xi_n = n + 1$.

4.3 Beyond STRING: Sublinear Reduction of ξ_n through SESat

The reduction of the control bandwidth requirements from $\xi_n = 2n$ (NHG) to $\xi_n = n + 1$ (STRING) enabled our implementation of microassembly using a group of four microrobots. We can however further reduce ξ_n , assuming we can freely design the snap-down and release voltages around the constraints presented in Sec. 4.1.

Lemma 4.2—Any stress-engineered n -microrobotic system with no degenerate pairs of robots can be sorted such that all $n + 1$ STRING hysteresis states are accessible.

For any n microrobots sorted primarily according to increasing release voltage $V_{u,i}$ and secondarily sorted according to increasing snap-down voltage $V_{d,i}$, the $n + 1$ STRING control primitives can be generated using the following recursive formula:

$$P_j = \begin{cases} (V_0, V_0), & \text{if } j=0; \\ (P_{j-1}, V_{d,j}, V_{u,j+}), & \text{if } j \in Z_n. \end{cases} \quad (4)$$

The complete proof of Lemma 4.2 is provided in Appendix B.

Theorem 4.3—A system of n STRING microrobots contains the minimum number ($n + 1$) of electromechanically accessible hysteresis states of any stress-engineered microrobot system without degeneracy.

Proof: Per Lemma 4.1; an n -microrobot STRING system has exactly $n + 1$ accessible hysteresis states, and by Lemma 4.2, any n stress-engineered microrobotic system without degenerate pairs of robots contains at least $n + 1$ hysteresis states.

Theorem 4.4—An algorithm that can plan the motion (i.e. finds the control sequence S) for a STRING system can be applied to plan the motion for any non-degenerate system of stress-engineered microrobots.

Proof: A consequence of Lemma 4.2; a STRING control matrix can be constructed for any n -robot stress-engineered microrobotic system.

Theorem 4.4 allows us to further *reduce* the control bandwidth requirements (ξ_n) for a microrobotic implementing microassembly using the control strategies described in Sec. 6. The control voltage bandwidth requirement for a microrobot system with k independent snap-down voltage levels and l independent release voltage levels is $\xi_n = k + l$. In an *electromechanically saturated* (ESat) system (ESat contains all permutations of $V_{d,j}$ and $V_{u,j}$, see Appendix Sec. B for a complete definition), the number of non-degenerate microrobots, is $n = kl$. It follows that n is maximized when $l = k = \xi_n/2$, and $n = \xi_n^2/4$. We call such system *symmetric electromechanically saturated*, or *SESat*. As a consequence, the control bandwidth requirement for an ESat system under our control strategy is $\xi_n \geq 2\lceil \sqrt{n} \rceil$, but it is merely $\xi_n = 2\lceil \sqrt{n} \rceil$ for an SESat system. As a consequence of Theorem 4.4 we can control SESat and ESat systems using a STRING control matrix and control algorithms that we will discuss in detail in Sec. 6, greatly reducing the control voltage bandwidth requirement for microassembly (under control strategy from Sec. 6). It follows that an SESat system maximizes the number of simultaneously controllable microrobots.

Table 2 compares the control voltage bandwidth requirements (under the control strategy from Sec. 6) and the number of control pulses in the control cycle of the STRING control primitives for the three classes of micro-robotic systems: a) NHG, b) STRING, c) SESat. Note that although the NHG system can access all the possible hysteresis states using $O(2n)$ control pulses, only a single control pulses per control primitive is required to access all the STRING hysteresis states. Although the SESat system has the smallest control voltage bandwidth requirements under our control strategy, the control cycle of each of its control primitives requires $O(2n)$ control pulses. In contrast, each control cycle in every control primitive for a STRING system requires only two control pulses. Clearly, which type of system is preferable is dictated by the number of robots, n , and the available control voltage bandwidth. Table 2 also shows the number of robots that can be independently controlled using NHG, STRING, and SESat, respectively, under the assumption that the transition voltages require a minimum separation $2\delta_v = 10$ V, and assuming $V_Q - V_{SDA} = 200$ V, which is consistent with our experimental setup. Note that under these conditions SESat in principle enables the control of 100 robots, 10 times more microrobots than can be simultaneously controlled using NHG.

Under the control strategies presented in Section 6, the time and physical space (area of assembly) complexity of both NHG, STRING, and SESat are similar, because they are all based on sequential ordering (two robots in parallel, followed by one robot at a time) and consequently sequential assembly of a goal shape \mathcal{G}_i . This sequential motion is promoted by the structure of the STRING control matrix, but is also important to enable collision avoidance and error correction during during the assembly process. However, the ability of an NHG system to access all of its 2^n hysteresis states provides a potential advantage compared to both STRING and SESat. Recall that for both STRING and SESat, the index i of the microrobot D_i , and correspondingly the order in which the robots are maneuvered towards the goal shape, is unique and predetermined by the structure of the STRING control matrix. Because an NHG system can access *all* of its hysteresis states, one can generate a STRING control matrix with arbitrary bijection to the set of all n microrobots. This allows the robots to assemble the target shape in arbitrary order, enabling any robot to occupy any configuration within the goal-shape \mathcal{G}_i . This may be of great advantage if the robots have the same shape (chassis) but are non-homogenous, such as carrying different payloads.

Furthermore, the merely sub-linear control voltage bandwidth requirement of a SESat system requires that the microrobot USDAs can be fabricated such that $V_{SDA}/2 + V_{rel} = V_Q/2$, or that the steering-arm actuators can be fabricated such that the number of significantly independent snap-down and release voltages is equal.

5 The Assembly Plan

An assembly plan (de Mello & Sanderson, 1990) contains an ordering of the assembly process, specifying precedence constraints between the assembling parts (certain parts must be assembled before others). In this section, we describe a simple method to generate an assembly plan for our microrobots given a desired target shape by enumerating the accessible (i.e., stable) shapes. Note, that the order of the assembly is set by the STRING control matrix. Consequently, the target configuration for the robots will contain enough information to represent the assembly plan.

For simplicity of exposition, we assume that all the robots are identical, and do not consider collisions between the out-of-plane curved steering arms. The input to the algorithm is the target shape, \mathcal{G}_i , represented as a list of robot configurations, $\mathcal{G}_i = (\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_p)$, where g_i represents the configuration of i^{th} robot within the overall goal-shape \mathcal{G}_i . The output of the algorithm is the goal configuration $\mathbf{g} = (\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n)$, which contains the individual goal

configurations of the n robots composing \mathcal{G}_k , sorted according to the order of motion imposed by the structure of the STRING control matrix.

The goal-shape is progressively assembled using *compliant-stable structures*. A compliant-stable structure is a structure where the forces and moments generated by all the microrobots and friction sum to zero. Consequently, since MEMS systems such as ours are quasi-static, compliant-stable structures do not change their configuration while USDAs are powered, allowing them to be used as building-blocks to form larger assemblies.

We calculate the assembly sequence for a target shape \mathcal{G}_k by enumerating all the intermediate stable structures leading up to \mathcal{G}_k . Our stability model dictates there exists only one feasible structure for $n = 2$, \mathcal{G}_1 , called the initial seed shape. Consequently, stable structures composed of three robots ($n = 3$) are generated by adding a single microrobot to each of the restricted docking locations (Fig. 7(b) and (c)) of \mathcal{G}_1 , testing for stability, and discarding the unstable structures. This procedure is repeated generating stable structures containing progressively more robots, until \mathcal{G}_k is generated. The resulting tree-structure, as shown on Fig. 7.c for $n = 4$, has \mathcal{G}_1 at the root, while each branch represents the successive addition of a single robot at a permitted docking location. \mathcal{G}_k is located in one of the leaves of the generated tree-structure. A path from \mathcal{G}_1 to \mathcal{G}_k (the root of the tree) generates an assembly plan for \mathcal{G}_k (For example see \mathcal{G}_8 in Fig. 7.c).

Note that the set of goal configurations (planar shapes) which can be reached (assembled) by our system is also restricted by the kinematics of our robots (the robots only move forward) and the reliance on force closure to maintain the position of the assembling shape. To relax both the force closure and unidirectional motion restrictions would require a redesign of our microrobots, an important direction for future work. (For instance, to relax the force closure requirement, our robots would require the ability to stop without pushing against other objects.)

6 Control Strategies for Microassembly

In this section (Sec. 6) we describe the control strategies that allow us to implement microassembly using groups of MicroStressBots. In our implementation, a control strategy consists of a control algorithm used to maneuver the robot to its goal, and recognizable goal and failure regions of C-space (Donald, 1987). (Recognizable region are regions where the robots' entry can be reliably measured.) All our control strategies were implemented using an iterative replanning control algorithm, RePlan, described in Sec. 6.1.

Two important features of our control strategies enable the control of our microrobots to assemble into planar shapes. First, we use self-aligning compliance (a form of pairwise self-assembly) to facilitate the generation of the initial seed shape \mathcal{G}_1 , simplifying simultaneous control of two robots to assemble \mathcal{G}_1 . Second, we use the structure of the STRING control matrix M to reduce the parallel motion of n robots to parallel motion of two robots, followed by sequential motion for single robots (while awaiting their turn, the robots are confined to circular trajectories called *closed-loop orbits*). Sequential motion towards the goal permits the correction of control error in the trajectories of only one or two robots at a time, which simplifies our solution to the accumulating control error.

We use planning under uncertainty (Lozano-Perez et al. (1984)), Error Detection and Recovery (EDR) (Donald (1987)), and compliance to define intermediate goal regions A_i , extended final goal regions G_i^* , and to construct global and local control strategies (σ_G and σ_L) that guarantee our robots to enter A_i and G_i^* (A brief review of the EDR terminology is presented in Appendix D). The resulting two-step EDR control strategy, combined with compliance, allows our robots to achieve high docking accuracy, as demonstrated by the experiments described in Sec. 7.2.

In particular, Figs. 11, 14, and 16 highlight how the concepts presented in the sections below were used in our experiments. In these figures, the theoretical model is illustrated as an annotated line drawing on the left, while on the right we show optical micrographs (of actual microrobots) that were captured during the corresponding physical experiments performed to validate the theory.

To model the inherent uncertainty in the pose and control of the robots described using the EDR framework, we define R_i to be the starting region for robot i , typically a ball around the nominal initial configuration \mathbf{r}_i . Correspondingly, let G_i be the region of goal configurations for robot i , typically an open set around the nominal goal region \mathbf{g}_i . Our objective is to maneuver the robots from their start region $R = R_1 \times R_2 \times \dots \times R_n$, to their goal region $G = G_1 \times G_2 \times \dots \times G_n$.

The process of assembling the target goal shape \mathcal{G}_* described using the EDR framework is shown in Fig. 8. The n STRING microrobots are labeled D_i , where $i \in \{1, \dots, n\}$ according to the convention described in Sec. 4. Initially, two robots (D_n and D_{n-1}) are simultaneously maneuvered to form the initial seed shape \mathcal{G}_1 through self-aligning compliance. Following the assembly of \mathcal{G}_1 , single robots (D_i where $i \in \{n-2, \dots, 1\}$) are maneuvered sequentially, progressively assembling the target shape \mathcal{G}_* . A summary of the symbols used throughout this section is presented in Table 3.

6.1 Implementing Control Strategies: Iterative Re-planning Control Algorithm (*RePlan*)

All our control strategies were implemented using an iterative re-planning control algorithm (*RePlan*), which is illustrated by Algorithm 1. *RePlan* consists of planning and execution stages that are repeated iteratively until the robots reach a goal or a failure region. In the planning stage, *RePlan* constructs a nominal trajectory, described through control sequence S , from a current ($\mathbf{q} \in Q$) to a goal ($\mathbf{g} \in G$) configuration using some planning function $\mathcal{F}(\mathbf{q}, \mathbf{g})$, which depends on the control strategy that is being implemented. Following the planning stage, the sequence S is partially executed by actuating the robots for a duration of t_x seconds (in our experiments t_x varied between 1 and 20 seconds). After the execution, the new position of the robots (\mathbf{q}) is registered (using an optical microscope and a digital camera), a new sequence S is generated, and the cycle is repeated. This loop continues until the robots enter the goal region G or the failure region H , where the algorithm exits.

6.2 Generating the Initial Seed Shape \mathcal{G}_1

The initial seed shape \mathcal{G}_1 provides the initial structure to which the consecutive robots dock to progressively assemble the target shape \mathcal{G}_* , and provides a nucleus for further assembly. The seed shape \mathcal{G}_1 is generated using force closure, and by necessity requires the simultaneous control of two microrobots. Using the order imposed by the STRING control matrix, robots D_n and D_{n-1} generate \mathcal{G}_1 , while D_i , $i \in \{1, \dots, n-2\}$ are confined to following closed-loop orbits. We further restrict \mathcal{G}_1 to be assembled from robots with steering-arms on opposing sides, such that their steering arms remain confined to one side of the seed shape \mathcal{G}_1 , minimizing interference during subsequent assembly process.

The structure of the STRING control matrix dictates that the motion of D_n and D_{n-1} is coupled; D_n can only turn if D_{n-1} is also turning. Although it is possible to generate an EDR control strategy for \mathcal{G}_1 that takes into account this coupling, we leave this derivation for future work. Instead, we used a simplified control heuristic that directs D_n and D_{n-1} towards a common intersecting point of their trajectories, and rely on self-aligning compliance to align the robots to form \mathcal{G}_1 . This simplified control heuristic, which implements *RePlan* as a subroutine, is shown on the flow-chart shown on Fig. 9. It consists of three steps: In step one, the robots were oriented (Fig. 10.a), such that they can collide using a single Dubins trajectory in the the center

region of the field. If the robots were already aligned properly at the start of the assembly, this step was omitted. In step two, the robots were maneuvered along a modified Dubins trajectory (Dubins (1957)) to make initial contact (Fig. 10.b). The initial contact may include large misalignment of the two robots, consequently in step 3, self-aligning compliance (step 3) was used to align the robots D_n and D_{n-1} to generate an accurate seed shape \mathcal{G}_1 .

Self-aligning compliance is a form of pairwise self-assembly. In our experiments, self-aligning compliance was assumed successful if the maximum allowed lateral misalignment w_s was less than $90 \mu\text{m}$ ($2/3$ of the bushing width), while the relative rotational misalignment α was less than 60° (see Fig. 11.a). Fig. 11.b shows a series of optical micrographs taken from the initial contact (i), and up to completed alignment (iv) of the robots forming \mathcal{G}_1 . Self-aligning compliance requires that the robots D_n and D_{n-1} are actuated with their steering arms lifted (i.e. straight-line motion). The structure of the STRING matrix however ensures that the next robot (D_{n-2}) is controlled using primitives P_{n-2} and P_{n-3} , which both cause straight-line motion of D_n and D_{n-1} . Consequently, the robots D_n and D_{n-1} are self-aligned as consecutive devices are maneuvered to dock with the seed shape \mathcal{G}_1 .

6.3 Two-step EDR Control Strategy for Sequential Assembly of \mathcal{G}_k

We now construct an EDR control strategy to assemble the goal shape \mathcal{G}_k by sequentially adding single robots to the seed shape \mathcal{G}_1 . To simplify the construction of the control strategy, the robots will be maneuvered to goal G_i via an intermediate region A_i . A control strategy σ_G , called the *global control strategy* is used to maneuver the robots from R_i to A_i . The global control strategy lacks sufficient accuracy to complete the docking operation, but in the absence of obstacles or collisions, it can maneuver the robots anywhere within their operating environment. From region A_i , the robots are maneuvered by control strategy σ_L , also referred to as the *local control strategy*, towards the goal region G_i . This strategy permits further reduction of the control error, however unlike σ_G , σ_L requires the robots to start in a specific region (A_i) in order to guarantee that they reach G_i . Region A_i contains configurations that are *provably* reachable from R_i using σ_G , and from which the robots are *guaranteed* to enter the goal using σ_L . The complete σ_G - σ_L control strategy is called a *two-step EDR control*

strategy, and can be represented by the following reachability diagram: $R_i \xrightarrow{\sigma_G} A_i \xrightarrow{\sigma_L} G_i$. Fig. 12 shows the projection of the C-space onto the x - y plane with annotated regions R_i , A_i , G_i , the trajectories representing both the global and the local control strategies, the configuration of the microrobot D_i at the start of σ_G (a), at the end of σ_G and start of σ_L (b), and at the end of σ_G .

In EDR, the failure region H signals the failure of assembly. Because our control strategies were implemented using manual registration of microrobot configurations, entry to the H region was visually recognized when the robots were stuck in a configuration that was not one of the intermediate goal shapes. In all our experiments we observed entry to H only once, resulting from an intermediate goal shape destabilizing and not the accuracy of the control strategies.

The iterative implementation of the two-step EDR control strategy on robots D_i , $i \in \{n-2, \dots, 1\}$ for assembling the shapes \mathcal{G}_i , $i \in \{2, \dots, k\}$ is shown by the flow chart in Fig. 13. The individual sub-blocks of the flow-chart are described in Secs. 6.3.1 – 6.3.4 below. Because the pose of each intermediate assembly \mathcal{G}_i depends on the pose of \mathcal{G}_1 , the regions G_i , G_i^* , A_i , and A_i^* are derived at the start of the motion of the individual robots D_i , $i \in \{n-2, \dots, 1\}$ towards their goal.

6.3.1 Expanding the Goal-regions G_i to G_i^* through Compliance—The goal region G_i is derived geometrically based on the configuration of the robots D_n, \dots, D_{i+1} that make up the thus far assembled shape. We use compliance between the docking robots and the assembling shape to expand the goal region G_i to an expanded goal region G_i^* from which the robot D_i is guaranteed to enter G_i via compliant interaction with the assembling shape. The phenomenon of compliance is illustrated in Fig. 14. When a single robot docks with an assembling shape, the docking robot will successfully align the front of its scratch-drive actuator with the edge of the shape as long as its angle of approach $\theta \in [\theta_0 - \alpha, \theta_0 + \alpha]$, where θ_0 is the nominal docking angle, and α is the angle at which the corner will slip. In our experiments, α was found to be approximately 45° . The tolerance for angular misalignment is smaller than in the case of the self-aligning compliance (Section 6.2) because only the docking robot is free to rotate and align. This alignment enlarges the goal configuration G_i to an expanded goal configuration G_i^* that include all the valid ($\theta \in [\theta_0 - \alpha, \theta_0 + \alpha]$) approach angles. Because the center of rotation as the robot aligns passes through the robot's corner, the expanded goal configuration G_i^* is a torus in C-space.

6.3.2 Deriving the Intermediate Goal-regions A and A^* —Region A marks the change from the global control strategy σ_G to a local control strategy σ_L , and is chosen such that it lies outside the *proximity space* (C_B) of the assembling shape, which is the region of C-space where the robots may collide with the assembling shape B under control strategy σ_G . C_B can be defined geometrically by expanding the boundary of B in C-space by a distance $d = zr^*$ where $z \approx 1.5$ in our experiments. A robot inside C_B may not be able to avoid colliding with the assembling shape under σ_G , however a robot outside C_B will be always able to avoid colliding with B under σ_G . Let C_F be the region outside the proximity space of B , $C_B, C_F = C - C_B$. Simply put, a robot D_i may abort and re-attempt the docking while it is under σ_G and outside of C_B , however once robot D_i is inside the C_B and progressing to G_i^* under σ_L , D_i is generally committed to attempt to enter G_i or enter H (fail).

We define A_i to be the intersection of C_F , the strong preimage of G_i^* under σ_L , and the forward projections of R_i under σ_G (from A_i); formally $A_i = P_{A_i, \sigma_L}(G_i^*) \cap C_F \cap F_{\sigma_G}(R_i)$. In addition, for A_i to be guaranteed reachable from R_i , it must hold that R_i is the subset of the preimage of A_i under σ_G (from R_i); formally $R_i \subset P_{R_i, \sigma_G}(A_i)$. A_i must contain the forward projection of the global control strategy, σ_G , from R_i , $F_{\sigma_G}(R_i)$. Because σ_G is implemented using a re-planning control algorithm (RePlan), $F_{\sigma_G}(R_i)$ is a cylinder around a target configuration \mathbf{a}_i , $C_{a,i} = B_{r_{a,i}}(\mathbf{a}_i) \times [\theta_{a,i} - h_{a,i}, \theta_{a,i} + h_{a,i}] \subset \mathbb{R}^2 \times S^1$, where $B_{r_{a,i}}(\mathbf{a}_i)$ is a ball of radius $r_{a,i}$ around \mathbf{a}_i . The bounds $r_{a,i}$ and $h_{a,i}$ on $F_{\sigma_G}(R_i)$ are described in detail in Appendix E. In order to ensure that $F_{\sigma_G}(R_i)$ is completely contained by the preimage of G_i^* , $P_{A_i, \sigma_L}(G_i^*)$, while remaining outside C_B , we define the region A_i^* which contains all the configurations \mathbf{a}_i that are at least $r_{a,i}$ away from the boundaries of $P_{A_i, \sigma_L}(G_i^*)$ and C_B in \mathbb{R}^2 ; $h_{a,i}$ in S^1 . Note that if $A_i^* = \emptyset$ we can not guarantee that the robot will reach G_i^* . The regions A_i, A_i^*, C_F , and $F_{\sigma_G}(R_i)$ projected onto the x - y plane of the C-space are illustrated on Fig. 15.

During our experiment the robots were maneuvered towards a configuration $\mathbf{a}_i^* \in A_i^*$ manually selected to fit within $P_{A_i, \sigma_L}(G_i^*)$ while sufficiently far away from the assembling shape to allow the robots to complete a full turn without risking a collision with B, i.e. allow docking to be aborted if exurban amount of control error would cause our robot to risk entering H . However, the necessity to abort docking never occurred during in our experiments.

6.3.3 The Global Control Strategy σ_G —The global control strategy σ_G is implemented using RePlan algorithm towards $\mathbf{a}_i^* \in A_i^*$ until robot D_i entered A_i or H . In principle any *global* path-planning algorithms can be used as \mathcal{F} to generate control sequence S from Q_i to A_i^* ; we chose a Dubins trajectory (Dubins (1957)) that was manually adjusted to avoid collisions with other robots and the assembling shape. A more general trajectory planning scheme which includes a collision avoidance heuristic is presented in Appendix C.

6.3.4 The Local Control Strategy σ_L —The local control strategy σ_L allows for more precise control of the robot D_i towards the expanded goal configuration G_i^* , however it requires D_i to start in a specific region of C-space in the vicinity of the assembling shape (region A). Strategy σ_L uses iterative re-planning of an interpolated turning trajectory (Donald et al., 2006). Interpolated turning interleaves straight-line and curved trajectory segments to effectively vary the turning radius r'_i of the microrobot between ∞ (straight-line motion) and, r_i (the maximum turning radius of the microrobot). The trajectory of σ_L follows a single arch along r'_i that intersects the expanded goal region G_i^* . As the trajectory of the robot is perturbed by the control error during microrobot motion, r'_i is adjusted such that it again passes through G_i^* . This in turn induces a change in the docking approach angle θ , however this accumulating error is later removed through compliance.

Recall the current configuration of D_i as $\mathbf{q}_i = (x_{q,i}, y_{q,i}, \theta_{q,i})$ and a goal configuration $\mathbf{g}_i = (x_{g,i}, y_{g,i}, \theta_{g,i})$. The radius r'_i of a local trajectory that allows D_i to reach a goal *location* $\mathbf{p}_i = (x_{g,i}, y_{g,i})$, $p_i \in G_i^*$ from \mathbf{q}_i is

$$r'_i = \frac{\Delta x^2 + \Delta y^2}{2(\Delta x \cos \theta_{q,i} - \Delta y \sin \theta_{q,i})}, \quad (5)$$

where $\Delta x = x_{g,i} - x_{q,i}$ and $\Delta y = y_{g,i} - y_{q,i}$. Let the control sequence S_L represent an interpolated trajectory of D_i , as $S_L = (P_{i,a}, P_{i-1,b}, \dots, P_{i,a}, P_{i-1,b})$, where a and b denotes the length of time the control primitives P_i and P_{i-1} are applied, respectively. Let ρ_a and ρ_b be the fraction of the time primitives P_i and P_{i-1} are applied in the control sequence S_L ; $\rho_a = \frac{a}{a+b}$ and $\rho_b = \frac{b}{a+b}$. The radius of curvature r'_i for the trajectory defined by S_L can now be fully described by either ρ_a or ρ_b , since $\rho_a + \rho_b = 1$, defined as:

$$r'_i = r \left(1 + \frac{\rho_a}{\rho_b} \right). \quad (6)$$

Eqs. (5) and (6), implemented in the RePlan algorithm as \mathcal{F} , can now be used to construct a control sequence $S = S_L$ between $\mathbf{q}_i \in Q_i$ and $\mathbf{p}_i \in G_i^*$. We use the shorthand notation $S_L(\rho_a)$ to denote the control sequence S_L defined by ρ_a . $S_L(1)$ denotes a straight line trajectory while $S_L(0)$ denotes turning at $r' = r$.

In our experiments, the replanning interval t_x in RePlan was reduced to approximately 1 second as robot D_i was approaching the assembling shape. As mentioned in Sec. 6.3.1, it was simpler to track the configuration of the corner that will first make contact with the assembling shape, $\tilde{\mathbf{q}}_i$. As the robot approached the assembling shape during our experiments, $\tilde{\mathbf{q}}_i$ was tracked in lieu of \mathbf{q}_i , adjusting S_L to ensure that $\tilde{\mathbf{q}}_i$ entered \tilde{G}_i^* . Fig. 16.b shows optical micrographs overlaid with planned local trajectories during our experiments as a robot approaches the

assembling shape ((i) – (iii)). Although the control primitive pair P_i and P_{i-1} can be repeated many times in S_L , we discovered that repeating the pair three times yielded satisfactory results.

7 Experimental Results

The control strategies described in previous sections have been tested experimentally on groups of fabricated MicroStressBots. This section (Sec. 7) uses experimental data that has been previously reported in (Donald et al., 2008b), but describes how this data validates the algorithms above, and gives the explicit construction of the control matrices necessary to replicate the results. The control strategies in Secs. 4–6 were not described in (Donald et al., 2008b) and form the heart of this paper. However, this section (Sec. 7) is necessary to present experimental validation of the algorithms. All data and images were obtained using real robots; there are no simulation results in this paper.

7.1 STRING Microrobots

We fabricated 15 MicroStressBots classified into five microrobot *species*. The microrobot species are differentiated by the designs of their steering arm actuators. Two of the species (1a and 1b) form a degenerate pair, i.e. although their designs are different, the snap-down and release voltages of their steering arms can not be differentiated using the available control primitives. While this degeneracy was initially unintentional, it is interesting to observe that two robots with vastly different steering-arm designs can exhibit such a similar behavior. Because of this degeneracy, microrobot species 1a and 1b were never simultaneously used in any of our microassembly experiment. This limited the number of microrobots that could be independently controlled at any given time to four. Fig. 17 shows scanning-electron micrographs of all the five microrobot species, including the two degenerate species 1a and 1b.

The robots were fabricated such that the transition voltages of their steering arms, ($V_{d,i}$ and $V_{u,i}$), were reproducibly confined to the ranges shown in Fig. 18(a). Snap-down voltages ($V_{d,i}$) are marked by circles, while the rectangles denote the release voltages ($V_{u,i}$). The ranges marked with two vertical dots signifying that the respective bound is not fixed or measured. Groups of robots from species 1a or 1b,2,3,4 form a four-robot STRING system and are independently controllable. The exact parameters of the steering-arms defining all five species are described in (Donald et al., 2008b), for reference these parameters are (with permission) reproduced in the Appendix F. The waveforms (control pulse and three power delivery pulses only) for the five control primitives P_0, \dots, P_4 used to control the microrobots are shown in Fig. 18(b). Average V_a , V_b and V_s voltage levels across all species are shown. The actual voltage-levels used to control the individual groups of microrobots could vary by up to ± 10 V.

Matrix M_r (Eq. (7)) contains experimentally recorded average and standard deviation of the turning radii r (in μm), across three independent stress-engineering runs for all five species, and represent the experimentally-measured behavior of the individual microrobots during the application of the control primitives. Notice that by replacing the cells representing turning behavior ($r < 550 \mu\text{m}$) with 1, and the cells representing straight-line motion ($r \geq 550 \mu\text{m}$) with 0 (thresholding), we obtain the STRING control matrix (Eq. (3)). The only difference between M and M_r is that the columns 1 and 2 in the thresholded M_r are linearly dependent, corresponding to degeneracy between species 1a and 1b. Consequently, a STRING control matrix will contain columns 1,3,4,5 or 2,3,4,5 corresponding to the four independently-controllable microrobot species.

$$M_r = \begin{pmatrix} 1699 \pm 348 & >2000 & 1570 \pm 720 & 1457 \pm 742 & >2000 \\ 211 \pm 43 & 273 \pm 3 & 1650 \pm 606 & 1549 \pm 373 & 1720 \pm 484 \\ 157 \pm 1 & 227 \pm 8 & 172 \pm 7 & >2000 & 1406 \pm 595 \\ 160 \pm 4 & 195 \pm 42 & 195 \pm 23 & 322 \pm 112 & 1990 \pm 17 \\ 158 \pm 7 & 202 \pm 32 & 210 \pm 32 & 316 \pm 92 & 167 \pm 14 \end{pmatrix} \quad (7)$$

The reproducibility of motion during the application of the five control primitives (P_0, \dots, P_4) is shown in Fig. 19. Each track represents a single, independent experiment. In this figure, a total of 140 tracks are shown, with 28 tracks for each of the 5 control primitives. For each of the device species, two tracks are shown for each of the robots fabricated through independent stress-engineering runs. The species 1a, 1b and 4 are left-handed, hence they turn counter-clockwise. The species 2 and 3 are right-handed, and turn clockwise. Species 3 and 4 show a slight tendency to turn in the opposite direction of their steering-arms when the arms are elevated, however in all cases the radii of curvature are larger than $550 \mu\text{m}$.

7.2 Microassembly

We applied the control algorithms described in Sec. 6 to groups of four STRING microbots (1a or 1b,2,3,4), thereby assembling a total of 14 planar structures. The assembled structures belong to five types of target shapes, labeled $\mathcal{G}_1 - \mathcal{G}_5$. Optical micrographs of microstructures for each type of target shape is shown on Fig. 20.

The robots were operated on a 2 mm^2 environment, and their position was recorded using a digital video-camera attached to an optical microscope ($6.7 \times$ objective lens). The position of the devices was measured with a precision of $\pm 2.1 \mu\text{m}$. The humidity was kept below 4% RH using a continuous stream of dry nitrogen. The waveforms for the control primitives were produced using an Agilent 33120A arbitrary waveform generator, and amplified with a Trek PZD700-1 high-voltage power amplifier with a gain of 200. The duration of the individual primitives was manually controlled during the execution of the control sequence S . We considered the assembly a success as long as the assembled shape is a rigid-body transformation of the target goal shape, i.e. the rotation and position of the assembled shape on the operating environment was not important.

Table 4 shows the average match (portion of the target structure covered by the assembled shape) for the five assembled shapes, $\mathcal{G}_1 - \mathcal{G}_5$. The assembly experiments were conducted starting from two different classes of initial configurations: \mathcal{R}_1 – robots are arranged along the corners of a rectangle with sides 1 by 0.9 mm, all devices oriented along the y -axis (see Fig. 21(a) for a representative example), and \mathcal{R}_2 – robots are arranged in a line with average separation of $360 \mu\text{m}$, and with variable orientation. The initial position of the microrobots was set using batch-transfer structures called *transfer-frames* (Donald et al., 2008b) and microprobes. We used common geometric shapes (a line and a rectangle) to demonstrate the ability to achieve successful assembly from arbitrary different initial configurations.

The results in Table 4 do not include completely failed assemblies. We recorded a 11% failure rate during the consecutive assembly of nine structures over the course of three assembly experiments. This reflects that the assembly of one of the nine structures failed due to the loss of stability of an intermediate structure, which we attribute to an initial unfortunate misalignment between the microrobots forming the intermediate assembly. Fig. 21 illustrates a representative assembly experiment. In this experiment, the target shape \mathcal{G}_5 is generated via the assembly of \mathcal{G}_1 and \mathcal{G}_4 . The experiment terminated when all four microrobots were successfully incorporated in the assembled structure. A movie of this assembly experiment is freely available online at (Donald et al., 2008c,d,a).

Table 5 shows the average misalignment after docking (position only), representing the remaining control error. For the assembly of \mathcal{G}_i , we report misalignment before and after self-aligning compliance, underscoring its contribution to the reduction of the control error. In order to further test the self-aligning compliance, large ($> 50 \mu\text{m}$) misalignment was purposefully introduced to robots D_n and D_{n-1} in two of the five assembly experiments. Because compliance during docking of single robots corrects only error in θ , and does not affect the position misalignment, only the misalignment after compliance was recorded during the assembly of structures \mathcal{G}_2 – \mathcal{G}_5 .

8 Conclusions

In this work, we addressed the planning and control challenges for achieving independent microrobot control of stress-engineered MEMS microrobots (MicroStressBots) assembling into planar structures. We presented control strategies that build on robotic planning under uncertainty, EDR, and compliance, allowing us to plan for and execute the microassembly of several types of planar shapes. The experimental data, reprinted from (Donald et al., 2008b), shows that our control scheme is feasible. To our knowledge, the microassemblies described in (Donald et al., 2008b) were the first implementation of a planar multi-microrobotic system capable of coordinated self-assembly using a single, global, control signal. The present paper describes the control strategies used for these microassemblies, which were not described in (Donald et al., 2008b).

We have shown that our control scheme minimizes the control voltage bandwidth requirements of an n -microrobotic system. The sub-linear ($\mathcal{O}(\sqrt{n})$) control voltage bandwidth requirement is a large improvement over $2n$ in our previously-proposed approach (Donald et al., 2006). Reducing the control voltage bandwidth requirements below the previous $2n$ bound was the enabling technology that allowed us to experimentally demonstrate simultaneous control of four devices.

We were able to efficiently implement error correction while controlling our robots towards the assembling shape using the STRING control matrix to reduce parallel motion of n robots to parallel motion of only two robots, followed by sequential motion of single devices. Furthermore, we used planning under uncertainty and EDR to construct control strategies to maneuver the robots to regions from which they could reliably enter the goal configuration, using compliance to correct for accumulating control error. These strategies have allowed us to demonstrate precise assembly of planar microstructures, achieving control accuracy on the order of the minimum feature size of the microfabrication process. The implementation and experiments we reported above use some heuristic steps for motion planning and EDR. However, rigorous and combinatorial versions of most of these algorithms have been reported in literature (e.g., (Donald, 1990b,a, 1993; Donald et al., 1993)). It would be interesting future work to replace the heuristic steps of our system with these provable algorithms to obtain an implementation that was end-to-end provable.

The assembly scheme presented in this work has two main limitations. First, the inability of our robots to move backwards or stop in place restricts the set of goal configurations (shapes) that can be reached (assembled) by our robots. This limitation is reflected in both our planning and control algorithms throughout this paper; our assembly scheme can only form shapes which are maintained through force closure and can be assembled by sequential addition of single microrobots to an initial seed shape formed by a pair of robots. It is easy to design a robot to stop on collision. It may be interesting future work to revisit this restriction and to investigate how our microrobots could be redesigned to stop in free space, for example a 2-armed robot could potentially stop in place by lowering both of its arms at the same time. Second, the lack of a control mechanism to correct the trajectories of orbiting robots necessitates sufficient

separation to account for the drift of their closed-loop orbits due to accumulating control error. This impacts scalability, since the orbiting robots will require large separation to avoid collisions as the number of robots increases (resulting in an increasing size of their orbits). However, our data suggest that the control error during turning is much smaller than the control error during straight-line motion, limiting their drift while following closed-loop orbits.

Although not completely general, the methodology presented in this paper tackles the control of stress-engineered multi-microrobotic systems by selectively addressing the behavior of individual devices through a global common control signal. We believe the concept of GCSR will be important for controlling future multi-microrobotic systems. The ability for a subset of the robots to follow a closed-loop orbit, i.e. to remain within a limited spatial region, while other devices progress toward the goal, is an example of how fabricated differences in device physics can be exploited to complete a cooperative task (such as microassembly) in a multi-microrobotic systems. GCSR allows for control of devices with limited computational resources, which will become important as we further reduce the size of the individual robots comprising multi-microrobotic systems. GCSR has interesting biological parallels, and may be the paradigm of choice for controlling groups of future micro- and nano-robots.

Acknowledgments

This work was supported by a grant to B.R.D. from the Duke Institute of Brain Sciences, grant numbers GM-65982, GM-78031, and NS-79929 to B.R.D. from NIH, and 2000-DT-CX-K001 to B.R.D., from the Office for Domestic Preparedness, Department of Homeland Security, USA. The electron micrographs were taken at the Dartmouth Ripple Electron Microscopy Laboratory, with the help of C. P. Daghljan. We further thank D. Balkcom, C. Bailey-Kellogg, A. Lebeck, J. Reif, and K. Böhringer for their advice and suggestions.

References

- Akiyama T, Shono K. Controlled stepwise motion in polysilicon microstructures. *Journal of Micro-electromechanical Systems*. 1993; 2(3):106–110.
- Berretty, R-P.; Goldberg, K.; Overmars, M.; vdStappen, F. Geometric algorithms for trap design. *The Proceedings of the 15th ACM Symposium on Computational Geometry*; 1998.
- Bohringer K-F, Bhatt V, Donald BR, Goldberg K. Algorithms for sensorless manipulation using a vibrating surface. *Algorithmica*. 2000; 26:389–429.
- Bohringer KF, Donald BR, MacDonald NC. Programmable force fields for distributed manipulation, with applications to MEMS actuator arrays and vibratory parts feeders. *International Journal of Robotics Research*. 1999; 18(2):168–200.
- Böhringer, K-F.; Donald, BR.; Mihailovich, RR.; MacDonald, NC. A theory of manipulation and control for microfabricated actuator arrays. *The Proceedings of the IEEE Workshop on Micro Electro Mechanical Systems MEMS*; 1994.
- Branicky, MS.; Causey, GC.; Quinn, RD. Towards a science of flexible feeding. *The Proceedings of the 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*; 1999. p. 380-385.
- Bretl, T. Control of many agents using few instructions. *the Proceedings of Robotics: Science and Systems 2007 conference*; Atlanta, GA. 2007.
- Bretl T. Minimum-time optimal control of many robots that move in the same direction at different speeds. *IEEE Transactions on Robotics*. 2012; 28(2):351–363.
- Butler Z, Fitch R, Rus D. Distributed control for unit-compressible robots: Goal-recognition, locomotion and splitting', *IEEE/ASME Transactions on Mechatronics*. 2002; 7(4):418–430.
- Byrne RH, Adkins DR, Eskridge SE, Harrington JJ, Heller EJ, Hurtado JE. Miniature mobile robots for plume tracking and source localization search. *Journal of Micromechatronics*. 2002; 1(3):253–261.
- Churaman, WA.; Gerratt, AP.; Bergbreiter, S. First leaps toward jumping microrobots. *Proceedings of 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; San Francisco, CA. 2011. p. 1680-1686.

- Dario P, Vallengi R, Carozza MC, Montesi MC, Cocco M. Microactuators for microrobots: a critical survey. *Journal of Micromechanics and microengineering*. 1992;141.
- de Mello LSH, Sanderson AC. And/or graph representation of assembly plans. *IEEE Transactions on Robotics and Automation*. 1990; 6(2):188–199.
- Dechev, N.; Cleghorn, WL.; Mills, JK. Tether and joint design for micro-components used in microassembly of 3D microstructures. the proceedings of SPIE Micromachining and Microfabrication; Photonics West. 2004; 2004.
- Diller E, Pawashe C, Floyd S, Sitti M. Assembly and disassembly of magnetic mobile micro-robots towards deterministic 2-D reconfigurable micro-systems. *The International Journal of Robotics Research*. 2011; 30(14):1667–1680.
- Donald, BR. Error Detection and Recovery in Robotics. Vol. 336. *Lecture Notes in Computer Science* Springer-Verlag; Berlin: 1987.
- Donald BR. The complexity of planar compliant motion planning with uncertainty. *Algorithmica*. 1990a; 5(3):353–382.
- Donald BR. Planning multi-step error detection and recovery strategies. *International Journal of Robotics Research*. 1990b; 9(1):3–60.
- Donald BR. Computational robotics: Manipulation, planning and control. *Algorithmica*. 1993; 10(2/3/4): 91–101.
- Donald, BR. Building very small mobile micro robots. Inaugural Lecture, Nanotechnology Public Lecture Series, MIT (Research Laboratory for Electronics, Electrical Engineering and Computer Science, and Microsystems Technology, Laboratories; Cambridge, MA: 2007. available online at: <http://mitworld.mit.edu/video/463/>
- Donald BR, Jennings J, Rus D. Information invariants for distributed manipulation. *International Journal of Robotics Research*. 1997; 16(10):673–702.
- Donald, BR.; Levey, CG.; Paprotny, I. Microrobots assembling a planar microstructure [Online]. 2008a. Available: <http://www.youtube.com/watch?v=pQuS5SkB8Kw>
- Donald BR, Levey CG, Paprotny I. Planar microassembly by parallel actuation of MEMS micro-robots. *Journal of Microelectromechanical Systems*. 2008b; 17(4):789–808.
- Donald, BR.; Levey, CG.; Paprotny, I. Planar microassembly by parallel actuation of MEMS microrobots – supplementary videos (2008) [Online]. 2008c. Available: www.cs.duke.edu/donaldlab/Supplementary/jmems08/
- Donald, BR.; Levey, CG.; Paprotny, I. Technical Report 553. The Department of Computer Science, Dartmouth College; 2008d. Planar microassembly by parallel actuation of MEMS micro-robots – supplementary videos (2008). [Online]. Available: www.cs.dartmouth.edu/reports/abstracts/TR2008-553/
- Donald BR, Levey CG, McGray C, Paprotny I, Rus D. An untethered, electrostatic, globally-controllable MEMS micro-robot. *Journal of Microelectromechanical Systems*. 2006; 15(1):1–15.
- Donald BR, Levey CG, McGray C, Rus D, Sinclair M. Power delivery and locomotion of untethered micro-actuators. *Journal of Microelectromechanical Systems*. 2003; 10(6):947–959.
- Donald, BR.; Levey, CG.; Paprotny, I.; Rus, D. Simultaneous control of multiple MEMS microrobots. Proceedings of the 9th International Workshop on Algorithmic Foundations of Robotics (WAFR); Guanajuato, Mexico. 2008e. p. 69-84. Chirikjian, GS.; Choset, H.; Morales, M.; Murphey, T., editors. Springer Tracts on Advanced Robotics. Vol. 57. Springer-Verlag; Berlin: 2010. p. 69-84.
- Donald BR, Xavier P, Canny J, Reif J. Kinodynamic motion planning. *Journal of the ACM*. 1993; 40(5): 1048–1066.
- Dubins LE. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*. 1957; 79(3):497–516.
- Floyd S, Diller E, Pawashe C, Sitti M. Control methodologies for a heterogeneous group of untethered magnetic micro-robots. *International Journal of Robotics Research*. 2011; 30(13):1553–1565.
- Floyd, S.; Pawashe, C.; Sitti, M. An untethered magnetically actuated micro-robot capable of motion on arbitrary surfaces. the Proceedings of IEEE International Conference on Robotics and Automation (ICRA); 2008.

- Frutiger, DR.; Kratochvil, BE.; Vollmers, K.; Nelson, BJ. Magnetics wireless resonant magnetic microrobots. the Proceedings of IEEE International Conference on Robotics and Automation (ICRA); 2008.
- Frutiger DR, Vollmers K, Kratochvil BE, Nelson BJ. Small, fast, and under control: Wireless resonant magnetic micro-agents. *International Journal of Robotics Research*. 2010; 29(5):613–636.
- Gracias DH, Tien J, Breen TL, Hsu C, Whitesides GM. Forming electrical networks in three dimensions by self-assembly. *Science*. 2000; 289:1170–1172. [PubMed: 10947979]
- Hollar, S.; Flynn, A.; Bellew, C.; Pister, KSJ. Solar powered 10 mg silicon robot. the proceedings of the The Sixteenth Annual International Conference on Micro Electro Mechanical Systems, 2003. MEMS-03 Kyoto; Kyoto, Japan. 2003. p. 706-711.
- Jadbabaie A, Lin J, Morse AS. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*. 2003; 48(6):988–1001.
- Jianghao L, Zhenbo L, Jiapin C. An omni-directional mobile millimeter-sized microrobot with 3-mm electromagnetic micromotors for a micro-factory. *Advanced Robotics*. 2007
- Kahn, JM.; Katz, RH.; Pister, KSJ. Mobile networking for smart dust. the proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 99); Seattle, WA. 1999. p. 17-19.
- Kladitis PE, Bright VM. Prototype microrobots for micro-positioning and micro-unmanned vehicles. *Sensors and Actuators A (Physical)*. 2000; 80(2):132–137.
- Klavins, E. Directed self-assembly using graph grammars. *Foundations of Nanoscience: Self Assembled Architectures and Devices*; 2004.
- Kotay K, Rus D. Locomotion versatility through self-reconfiguration. *Robotics and Autonomous Systems*. 1999; 26:217.
- LaValle, S. *Planning Algorithms*. Cambridge University Press; 2006.
- Li, J-S. PhD thesis. Harvard University; 2006. Control of inhomogenous ensembles.
- Li J-S, Khaneja N. Noncommuting vector fields, polynomial approximations and control of inhomogenous quantum ensembles. *Quantum Physics*. 2005:0510012.
- Li J-S, Khaneja N. Control of inhomogenous quantum ensembles. *Physics review A*. 2006
- Liang, SH.; Xiaoang, X.; Böhringer, K-F. Towards optimal design for self-alignment in surface tension driven micro-assembly. the proceedings of the IEEE Conference on Micro Electro Mechanical Systems (IEEE MEMS 2004); 2004. p. 9-12.
- Linderman, RJ.; Bright, VM. Technical Digest Solid-State Sensor and Actuator Workshop. 2000. Optimized scratch drive actuator for tethered nanometer positioning of chip-sized components; p. 214-217.
- Lozano-Perez T, Mason MT, Taylor RH. Automatic synthesis of fine-motion strategies for robots. *International Journal of Robotics Research*. 1984
- Martinoli A, Agassounon KEW. Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *The International Journal of Robotics Research*. 2004; 23(4–5):415–436.
- Mohebbi, MH.; Terry, ML.; Böhringer, KF.; Kovacs, GTA.; Suh, JW. Omnidirectional walking microrobot using MEMS thermal cilia arrays. the proceedings of ASME, International Mechanical Engineering Congress and Exposition; 2001. p. 1-7.
- Nathanson HC, Newell WE, Wickstrom RA, Davis JJR. The resonant gate transistor. *IEEE Transactions on Electron Devices*. 1967; ED-14(3):117–133.
- Ohta AT, Chiou P-Y, Han TH, Liao JC, Bhardwaj U, McCabe ERB, Yu F, Sun R, Wu MC. Dynamic cell and microparticle control via optoelectronic tweezers. *Journal of Microelectromechanical Systems*. 2007; 16(3):491–499.
- Onoe H, Matsumoto K, Shimoyama I. Three-dimensional micro-self-assembly using hydrophobic interaction controlled by self-assembled monolayers. *Journal of Microelectromechanical Systems*. 2004; 13(4):603–611.
- Pagello E, D'Angelo A, Montesello F, Garelli F, Ferrari C. Cooperative behaviors in multi-robot systems through implicit communication. *Robotics and Autonomous Systems*. 1999; 29(1):65–77.
- Pallottino L, Scordio VG, Frazzoli E, Bicchi A. Decentralized cooperative policy for conflict resolution in multi-vehicle systems. *IEEE Transactions on Robotics*. 2007; 23(6):1170–1183.

- Paprotny, I.; Levey, CG.; Wright, PK.; Donald, BR. Turning-rate selective control : A new method for independent control of stress-engineered mems microrobots. the Proceedings of 2012 Robotics: Science and Systems (RSS 2012); 2012.
- Pawashe C, Floyd S, Sitti M. Modeling and experimental characterization of an untethered magnetic micro-robot. *International Journal of Robotics Research*. 2008; 28(8):1077–1094.
- Pawashe C, Floyd S, Diller E, Sitti M. Two-dimensional autonomous microparticle manipulation strategies for magnetic microrobots in fluidic environments. *IEEE Transactions on Robotics*. 2011; (99):1–11.
- Popa D, Stephanou HE. Micro and meso scale robotic assembly. *SME Journal of Manufacturing Processes*. 2004; 6(1):52–71.
- Rothemund PWK. Using lateral capillary forces to compute by self-assembly. *Proceedings of the National Academy of Sciences of the United States of America*. 2000; 97(3):984–989. [PubMed: 10655471]
- Rothemund PWK. Folding dna to create nanoscale shapes and patterns. *Nature*. 2006; 446:297–302. [PubMed: 16541064]
- Rus D. Self-reconfiguring robots. *IEEE Intelligent Systems*. 1998; 13:2.
- Rus, D.; Donald, BR.; Jennings, J. Moving furniture with teams of autonomous mobile robots. the proceedings of the IEEE/Robotics Society of Japan International Workshop on Intelligent Robots and Systems, (IROS); Pittsburg, PA. 1995.
- Saeedi E, Abbasi S, Bohringer K-F, Parviz BA. Molten-alloy driven self-assembly for nano and micro scale system integration. *Fluid Dynamics and Materials Processing*. 2006; 2(4):221–245.
- Seeman NC. Nucleic acid nanostructures and topology. *Angewandte Chemie International Edition*. 1998; 37(23):3220–3238.
- Skidmore, G.; Ellis, M.; Geisberger, A.; Tsui, K.; Tuck, K.; Saini, R.; Udeshi, T.; Nolan, M.; Stallcup, R., 2nd; JVE. Assembly technology across multiple length scales from the micro-scale to the nano-scale. the proceedings of the 17th IEEE International Conference on MEMS; 2004. p. 588-592.
- Stefanini C, Menciassi A, Dario P. Modeling and experiments on a legged microrobot locomoting in a tubular, compliant and slippery environment. *International Journal of Robotics Research*. 2006; 25 (5–6):551–560.
- Stilwell, D.; Bay, J. Toward the development of a material transport system using swarms of ant-like robots. the proceedings of IEEE International Conference on Robotics and Automation; Atlanta, GA. 1993. p. 766-771.
- Suh JW, Daring RB, Bohringer KF, Donald BR, Baltes H, Kovacs GTA. CMOS integrated ciliary actuator array as a general-purpose micromanipulation tool for small objects. *Journal of Microelectromechanical Systems*. 1999; 8(4):483–496.
- Whitesides GM, Grzybowski B. Self-assembly at all scales. *Science*. 2002; 295:2418–2421. [PubMed: 11923529]
- Yesin KB, Vollmers K, Nelson BJ. Modeling and control of untethered boimicrorobots in fluid environment using electromagnetic fields. *The International Journal of Robotics Research*. 2006; 25 (5–6):527–536.

Appendices

A Proof of Lemma 4.1

Lemma 4.1

An n -robot STRING system has exactly $n + 1$ accessible hysteresis states

Proof

(By induction.)

Base case: a STRING system with $n = 1$, has two accessible hysteresis states, (0 - arm up) and (1 - arm down).

Inductive step: adding a single device, (changing the size of the system from n to $n+1$) extends the number of accessible control state by exactly one, provided that both the n and $n+1$ microrobotic systems remain STRING.

Let n microrobots, labeled D_1, \dots, D_n , be a STRING system sorted according to $V_{u,i}$ and $V_{d,i}$. Without loss of generality, $V_{d,n} < V_{d,n+1}$ and $V_{u,n} < V_{u,n+1}$ (If this is not the case, we can simply relabel the voltages and generate an equivalent system sorted as described above). Fig. A.1 shows the ranges for the transition voltages of D_{n+1} , such that the new, $n+1$ robotic system retains STRING. Let $V_\alpha, \dots, V_\delta$ be significantly independent transition voltage levels, ordered such that $V_\delta < V_\gamma < V_u < V_\beta < V_\alpha < V_\Omega$. Let $V_{d,n} = V_\alpha$ and $V_{u,n} = V_\gamma$. It follows that the snap-down voltage $V_{d,n+1}$ can have a value V_1 in the range $[V_\alpha, V_\Omega]$, or voltage $V_2 = V_\alpha$. Similarly, the release voltage, $V_{u,n+1}$, can have the value V_3 in the range $[V_{rel} - 2\delta_v, V_\gamma]$, or voltage $V_4 = V_\gamma$ ($V_{u,n+1}$ can not exceed $V_{rel} - 2\delta_v$ without risking that V_{rel} might release the steering arm during the power delivery cycle). Consequently, for the $(n+1)$ robot system to remain STRING, one of the following combinations of the snap-down and release voltages for D_{n+1} must hold: (V_1, V_3) , (V_1, V_4) and (V_2, V_3) . We examine each case separately:

(V_1, V_3) : Because the snap-down voltage of D_{n+1} is greater than the snap-down voltage of $D_1 \dots D_n$, $V_{d,n+1} > V_{d,i}$, $i \in Z_n$ where $Z_n = \{1, \dots, n\}$, we can only snap-down the arm of D_{n+1} after we snap-down the arms of all other devices. Since the release voltage of D_{n+1} is greater than the release voltage of D_1, \dots, D_n , $V_{u,n+1} > V_{u,i}$, $i \in Z_n$, we can only release the arm of D_1, \dots, D_n after we have released the arm of D_{n+1} . Consequently, we can only change the state of D_{n+1} when D_1, \dots, D_n are in state 1. During all other states of the system, the state of D_{n+1} must remain 0. Consequently, the number of accessible hysteresis states increases by exactly one.

(V_1, V_4) : This case is identical to (V_1, V_3) , except that the arm of D_n is released at the same time as the arm of D_{n+1} . As long as $V_{d,n+1} > V_{d,i}$, we can snap down the arm of D_{n+1} only after all other devices D_1, \dots, D_n are in state 1. As a consequence, the number of accessible hysteresis states increases by one.

(V_2, V_3) : The snap-down voltage of D_{n+1} is equal to the snap-down voltage of D_n , $V_{d,n+1} = V_{d,n}$. In this case, the arm of D_{n+1} is snapped down at the same time as the arm of D_n . Because the release voltage of D_{n+1} is greater than the release voltage of D_1, \dots, D_n , $V_{u,n+1} > V_{u,i}$, where $i \in Z_n$, we can only release the arm of D_n (or any other devices) after we release the arm of D_{n+1} . As in the (V_1, V_3) case, the state of D_{n+1} must be 0 except when D_1, \dots, D_n are all snapped down, then D_{n+1} can be in either 0 or 1 by varying the release voltage. Consequently, the number of accessible hysteresis states increases by one.

We have now shown that adding a device to STRING system, such that the resulting system remains an STRING system, increases the number of accessible hysteresis states by exactly one. Combined with the base case ($n = 1$, two hysteresis states), it follows by induction that every n -robot STRING system has exactly $n+1$ accessible hysteresis states.

B Proof of Lemma 4.2

Lemma 4.2

Any stress-engineered n -microrobotic system with no degenerate pairs of robots can be sorted such that all $n+1$ STRING hysteresis states are accessible.

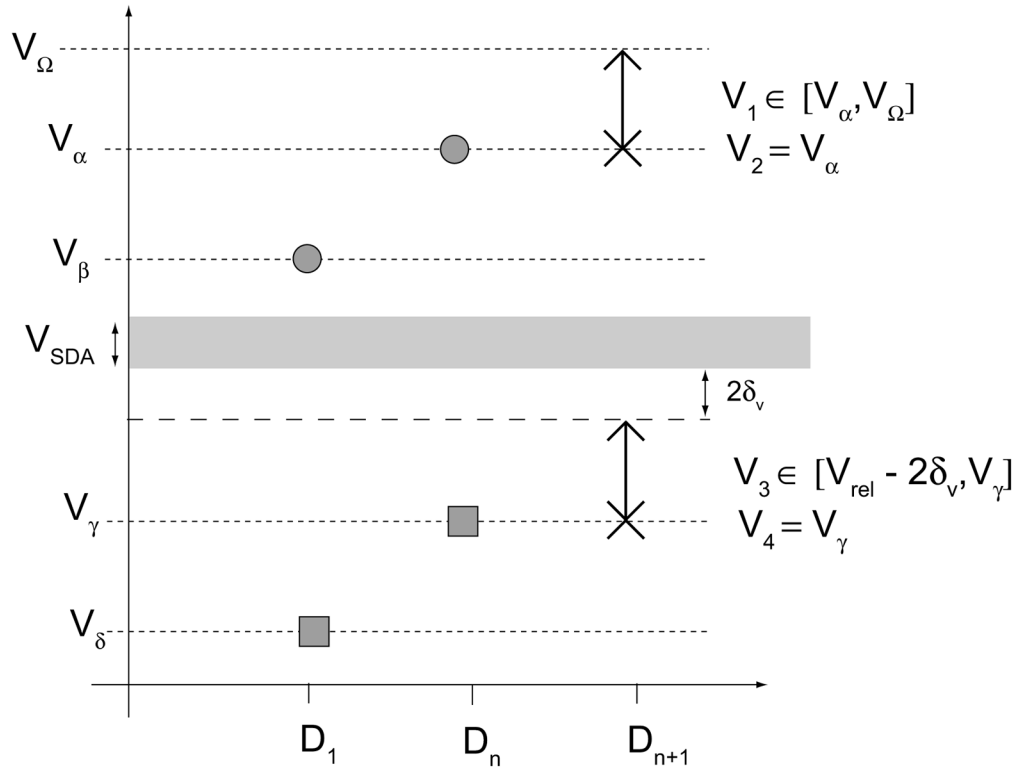


Figure A.1.
Illustration of the proof of Lemma 4.1.

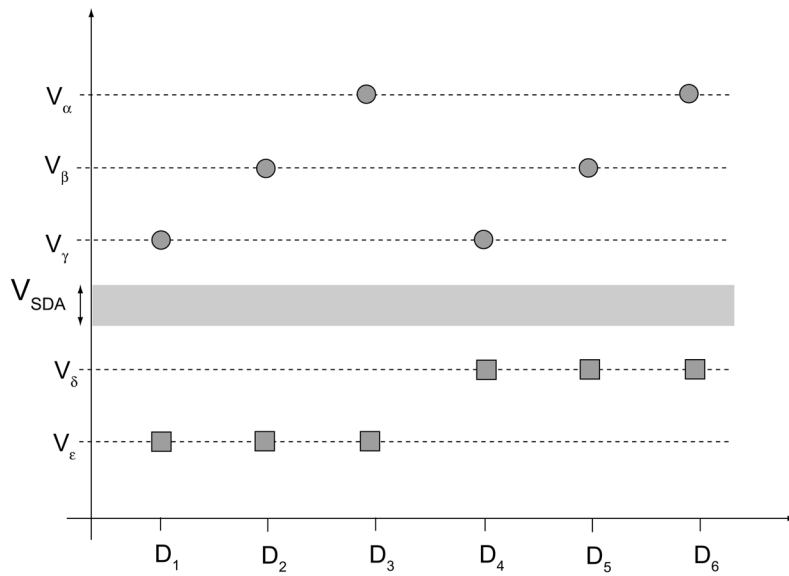


Figure B.1.
Example of an ESat system with $k = 3$ and $l = 2$.

Proof

By construction. Consider a microrobot system with k independent snap-down voltages, and l independent release voltages. Assuming no degenerate pairs of devices, it follows that n kl . In the case when $n = kl$ the n steering arms encode all the possible kl combinations of snap-down and release voltages. We call such system for *electromechanically saturated* (ESat). We can enumerate the hysteresis gaps for an ESat system given both k and l . Consider an ESat system, sorted primarily according to increasing release voltage $V_{u,i}$ and secondarily sorted according to increasing snap-down voltage $V_{d,i}$. Fig. B.1(b) shows such a system when $k = 3$ and $l = 2$. Note that sorting ensures a monotonic increase of $V_{u,i}$ with increasing index i . For such an order, there exists a recursive formula, shown in Eq. (B.1), that generates all $n + 1$ STRING control primitives. The control cycle for each control primitive defined by Eq. (B.1) contains a sequence of up to $2n$ control pulses (in contrast with 2 control pulses in Eq. (2)). Again, we assume $V_b = V_{rel}$ and $V_s = V_{flx} = V_{d,1}$. We construct the control primitive P_j ,

$$P_j = \begin{cases} (V_0, V_0) & \text{if } j=0; \\ (P_{j-1}, V_{d,j}, V_{u,j+}), & \text{if } j \in Z_n. \end{cases} \quad (\text{B.1})$$

P_j generates $n + 1$ control primitives that form a STRING control matrix, by causing devices D_i ($i = j$) to be in state 1, while robots D_i ($i > j$) are in state 0. Consider the base case P_0 , where all D_j , ($j \in Z_n$) is in state 0. We make the inductive argument that after application of the recursive part of P_j , P_{j-1} , all D_1, \dots, D_{j-1} robots are in control state 1. It is clear that $V_{d,j}, V_{u,j+}$, ($j \in Z_n$), will snap down D_j . The $V_{u,i} - V_{d,i}$ sorting implies that, for a device D_k , $k > j$, only two cases are possible with respect to its transition voltages: (a) $V_{d,j} < V_{d,k}$ (e.g., $j = 2$ and $k = 3$ in Fig. B.1), or (b) $V_{u,j} < V_{u,j+} < V_{u,k}$ (e.g., $j = 3$ and $k = 5$ in Fig. B.1). It is clear that in case (a), $V_{d,j}$ sets D_j to state 1, while D_k , $k > j$ is in state 0. The sorting ensures that any previously applied control primitive P_i , $i < j$ with $V_{a,1} < V_{d,k}$ (which also inadvertently snaps down the arm of D_k) must have been followed by a control pulse $V_{a,2} = V_{u,k} - 2\delta_v$ (which would release the arm of D_k). In case (b), $V_{u,j+}$ releases any devices D_k , $k > j$.

Note that because the devices are sorted according to $V_{u,i}$ and $V_{d,i}$, Eq. (B.1) also holds for any microrobotic system, even one that is not ESat.

C Motion Planning for Microassembly

This section is included for completeness, and describes how the structure of the STRING control matrix can be used to plan the motion of our robots (in the absence of the control error) from an initial configuration $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n)$, to a goal configuration $\mathbf{g} = (\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n)$. The output of the motion planning is a control sequence S which describes the trajectory of the robots from \mathbf{r} to \mathbf{g} using control primitives in M . The structure of M is used to reduce the parallel motion of n robots to parallel motion of two robots, followed by sequential motion for single devices. The robots that are not progressing toward the goal are confined to follow closed-loop orbits.

The motion planning presented in this section can plan the assembly of an arbitrary shape generated by assembly planning from Sec. 5 from a set of STRING robots with the center of curvature of their turning trajectories are separated by at least $3r^*$ (A minimum requirement to enable collision avoidance, see Sec. C.4.) A simplified version of this motion planning was used in experimental validation of the control strategies presented in Sec. 7. The complete motion planning scheme is presented below for completeness.

The assembly planning is very similar to the implementation of our control scheme presented in Sec. 6. The initial stable shape \mathcal{G}_1 is assembled using two robots, while the remaining $n - 2$ robots orbit without making progress towards the goal (Sec. C.1). Consecutive robots are then sequentially added to the assembling structure; the robots that have already docked with the assembling shape are kept in place by force closure, while the robots that are awaiting their turn to dock orbit without making progress towards the goal (Sec. C.2). The assembly of a structure composed of n robots takes place in $n - 1$ steps.

C.1 Step 1: Assembly of \mathcal{G}_1

\mathcal{G}_1 is always assembled through the simultaneous motion of the two robots with the highest indexes, i.e. D_n and D_{n-1} (the two robots that turn during the smallest number of control primitives), using primitives P_n, P_{n-1} and P_{n-2} only. This allows the remaining robots D_1, \dots, D_{n-2} to follow closed-loop orbits without making progress towards the goal. The assembly of \mathcal{G}_1 is divided into two stages, as shown on Figs. C.1(a) and C.1(b), respectively (for clarity, the orbits of robots $D_{n-2} \dots D_1$ are shown in Fig. C.1.c but are not depicted in Figs. C.1.a and C.1.b).

During stage 1 (Fig. C.1(a)) microrobot D_n is maneuvered from an initial configuration, \mathbf{r}_n , to an intermediate goal configuration \mathbf{a}_n , using only control primitives P_n and P_{n-1} (trajectory T_1). The reason for maneuvering robot D_n to \mathbf{a}_n rather than directly to its goal \mathbf{g}_n is that D_n will only be able to move in a straight line during stage 2. Thus during stage 1, D_n must be maneuvered to a configuration from which it can enter the goal, \mathbf{g}_n , using solely straight-line motion in stage 2 (trajectory T_2). As D_n is maneuvered to the intermediate configuration \mathbf{a}_n , robot D_{n-1} orbits without making any progress towards the goal \mathcal{G}_1 (because control primitives P_n and P_{n-1} invoke only turning motion in D_{n-1}). In order to calculate the length of the trajectory of robot D_{n-1} in stage 2 (trajectory T_3), and hence the intermediate configuration \mathbf{a}_n to which we must drive robot D_n , we must know where is D_{n-1} at the beginning of stage 2. To achieve this, we ensure that robot D_{n-1} always orbits back to initial configuration \mathbf{r}_{n-1} at the end of stage 1 by adjusting the length of T_1 and correspondingly the lengths of the orbit of D_{n-1} using the method described in Sec. C.3. This allows us to use the initial configuration \mathbf{r}_{n-1} as the starting configuration for planning the trajectory of robot D_{n-1} at the beginning of stage 2.

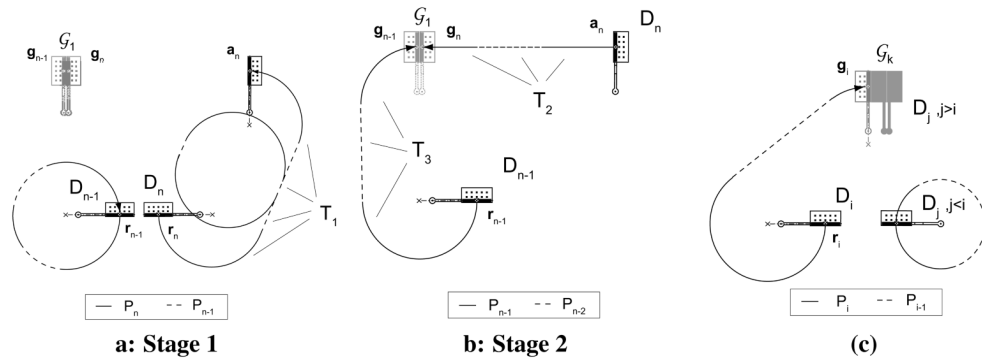


Figure C.1.

Step 1 of microassembly; the assembly of the initial stable shape, \mathcal{G}_1 using microrobots D_n and D_{n-1} . **a: Stage 1:** D_n is maneuvered to \mathbf{a}_n while D_{n-1} orbits. **b: Stage 2:** D_{n-1} is maneuvered to \mathbf{g}_{n-1} while D_n moves in straight line to \mathbf{g}_n . **c:** Progressive docking of single microrobots to the assembling stable shape. D_j orbits while D_i is maneuvered to dock with the stable shape.

In stage 2, microrobot D_{n-1} is maneuvered from \mathbf{r}_{n-1} to its target configuration, \mathbf{g}_{n-1} , using only primitives P_{n-1} and P_{n-2} (see Fig. C.1(b)). Both these primitives are sufficient to maneuver robot D_{n-1} to an arbitrary configuration, but cause only straight-line motion in D_n . However, as described above, we ensured that the intermediate configuration \mathbf{a}_n is chosen such that D_n moves into its target configuration \mathbf{g}_n during its straight-line motion along T_2 .

Whenever the configuration of the robot swept along the trajectories T_1 , T_2 , and T_3 would intersect the $2r^*$ region swept by the orbiting robots, a collision might occur. The collision heuristic described in Sec. C.4 must be used, to ensure that the robot stays outside the orbiting region. If the length of T_3 changed, T_1 must be adjusted again per. Sec. C.3.

C.2 Steps 2: ..., $n - 1$. Subsequent addition of single robots

After the assembly of \mathcal{G}_i , single robots are sequentially maneuvered towards their goal configurations while the remaining robots are either docked or orbiting along limit cycles (See Fig. C.1.c.). The structure of the control matrix M allows robot D_i to be maneuvered to its target configuration \mathbf{g}_i using control primitives P_i and P_{i-1} , while robots $D_j, j < i$, orbit in place. Control primitives P_i and P_{i-1} normally would also cause straight-line motion in robots $D_j, j > i$, but, since our robots are assembled in decreasing order of i , these robots are already docked and immobilized as part of an intermediate stable structure shape.

As in step 1, if the trajectory of robot D_i intersects the $2r^*$ region swept by the orbiting robots, the collision heuristic described in Sec. C.4 must be used to ensure that the robot D_i stays outside the orbiting regions.

C.3 Adjusting Trajectory T_1 such that Robot D_{n-1} returns to \mathbf{r}_{n-1}

Here we modify control sequence S_1 , corresponding to the trajectory T_1 (Fig. C.1.a), so that the corresponding trajectory could be easily extended without changing the destination of robot D_n , and then *adjusting* S_1 such that robot D_{n-1} orbits back to \mathbf{r}_{n-1} . This is done in three steps, shown in Fig. C.2 a, b and c.

In step 1 (Fig. C.2.a), we insert the control primitive $P_{n,t_2\pi}$ to the middle of a straight segment in S_1 , where $t_2\pi$ corresponds to the time it takes robot D_n (and D_{n-1}) to turn 2π , and $P_{n,t_2\pi}$ is control primitive P_n applied for $t_2\pi$ time. Naturally, the structure of the control matrix ensures that robot D_{n-1} follows a closed-loop orbit during the entire duration of S_1 .

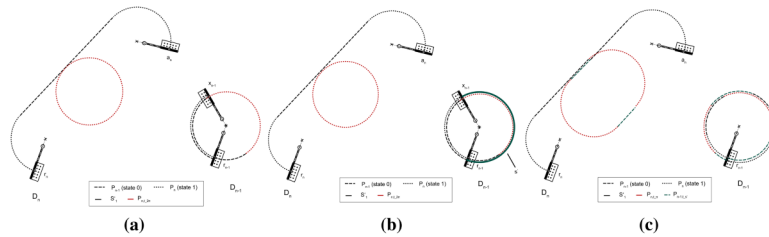


Figure C.2.

Three steps of the geometric planning of the control sequence S_1 such that robot D_n reaches the intermediate configuration \mathbf{a}_n at the same time as the robot D_{n-1} returns to its initial configuration \mathbf{r}_{n-1} .

In step two (Fig. C.2.b) we calculate the remaining distance, s' , along the circular closed-loop orbit of D_{n-1} between \mathbf{q}_{n-1} and its initial configuration \mathbf{r}_{n-1} . For now we assume that the turning rates of robots D_n and D_{n-1} are identical. It follows that the length of the trajectory for D_n must also be increased by s' in order for D_{n-1} to reach \mathbf{r}_{n-1} during execution of S_1 .

Thus, in step 3 (Fig. C.2.c) we use s' to adjust S_1 such that robot D_{n-1} returns to \mathbf{r}_{n-1} as robot D_n reaches a_n . This is simply done by replacing $P_{n,t_2\pi}$ with the sequence $\{P_{n,t\pi}, P_{n-1,t_{s'}}, P_{n,t\pi}\}$, where $t\pi$ is the time it takes robot D_n to turn π , and $t_{s'}$ is the time it takes D_n to move distance s' . S_1 can follow the torus around the orbits or assembling shapes required by the collision avoidance heuristic (see Sec. C.4).

C.4 Collision Avoidance

Our collision avoidance heuristic is based on the conservative space requirement; the orbiting devices need $\pi(r^*)^2$ space to orbit, surrounded by a free-space annulus of $2r^*$ thickness (See Fig. C.3). Given an initial trajectory S' , we test for collisions between the space covered by the orbiting robot (i) and the sweep of the area of the robot during its desired trajectory (ii). If these areas intersect, we conservatively declare a collision.

Generating the adjusted trajectory is done as follows; we first convert the initial microrobot trajectory to the trajectory of its center of rotation (iii). The trajectory to maneuver the robot around the space swept by the orbiting device (i) is calculated in the following manner: We plan the motion of the robot along its old trajectory until the robot's center of rotation is $2r^*$ away from the center of rotation of the orbiting robot (configuration 1). We then plan turning motion until the steering arm of the robot is pointing towards the center of rotation for the orbiting robot (configuration 2). We then plan the motion of the robot to configuration 3, where the center of rotation for the robot again intersects the initially generated trajectory of its center of rotation (iii). To reach 3 while remaining within the free-space torus, we plan a two-primitive local trajectory (See Sec. 6.3.4) with radius $r' = 2r^* + r$. Finally, we plan a turning trajectory until the robot faces in its original orientation (configuration 4). The robot can now continue along its initially planned trajectory. A similar approach can be used to avoid the assembling shape.

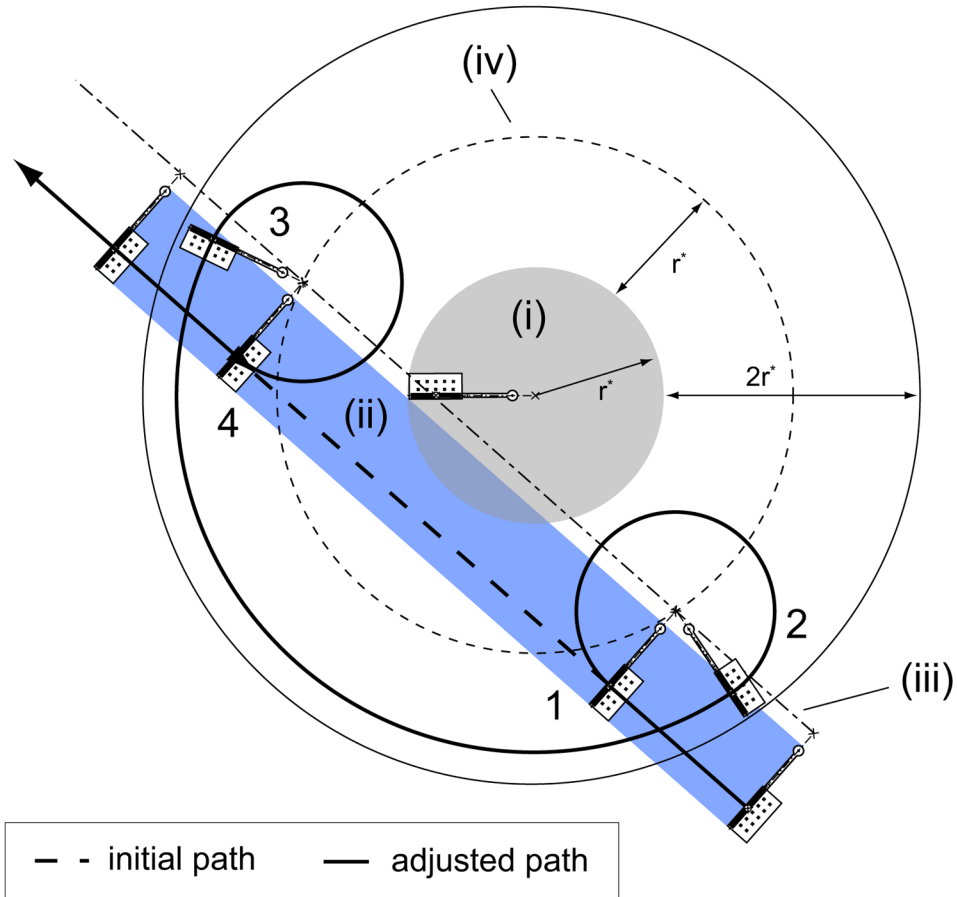


Figure C.3.
Geometric collision avoidance heuristic.

D Brief Review of EDR Terminology

We use preimages and forward projection (Lozano-Perez et al. (1984)) to define regions which the robots are guaranteed to enter during the implementation of a control strategy. This terminology is illustrated using the peg-in-a-hole example reproduced in Fig. D.1. Given starting region X and control strategy θ , the (strong) preimage of region Y , $P_{X,\theta}(Y)$, is the region of configuration space (C-space) from which the robot is guaranteed To Whom It May Concern: recognizably enter the region Y . A weak preimage of Y , $\hat{P}_{X,\theta}(Y)$, is the region of C-space from which the robot might recognizably enter Y , given fortuitous sensing and control events, when starting in X and applying control strategy θ . The forward projection, $F_{\theta}(X)$ of X under θ is the region of C-space which the robot might reach after the execution of the control strategy θ when starting in region X .

E Error Bounds

Error bounds are primarily used to bound the size of forward-projection regions for the global control strategies, $F_{\sigma_G}(R_i)$, defining the regions $C_{a,i}$ and $\hat{C}_{a,i}$. We derive simple error bounds from the kinematic model shown in Sec. 3 by substituting $\omega = \frac{v\theta}{r}$ and adding error components v_e and ω_e to the microrobot turning rate (ω) and linear velocity (v):

$$\dot{q} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \sin\theta \\ \cos\theta \\ 0 \end{pmatrix} (v+v_e) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} (\omega+\omega_e). \quad (E.1)$$

Let $\hat{v} = v + v_e$, and $\hat{\omega} = \omega + \omega_e$. The displacement with error, $\Delta q_e(t)$, can be written as:

$$\Delta q_e(t) = \begin{pmatrix} \int_0^t \hat{v} \cos(\theta - t\hat{\omega} + \frac{\pi}{2}) \\ \int_0^t \hat{v} \sin(\theta - t\hat{\omega} + \frac{\pi}{2}) \\ t\hat{\omega} \end{pmatrix}. \quad (E.2)$$

Eq. E.2 represents the error bound for a single control primitive. Let $\delta(t) = \int_S \Delta q_e(t) - \Delta q(t) dt$ be the error integrated over the control sequence S , where $\delta(t) = (\delta_x(t), \delta_y(t), \delta_\theta(t))^T$, and

$$\delta_{xy}(t) = \sqrt{\delta_x^2(t) + \delta_y^2(t)} \text{ (error in } \mathbb{R}^2 \text{)}.$$

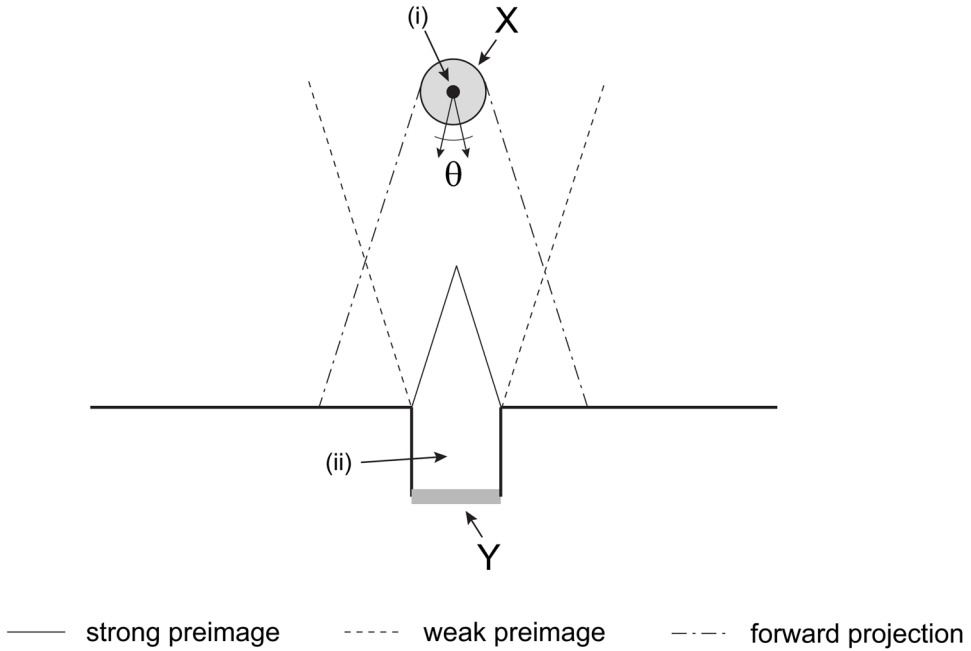


Figure D.1. Reproduced from Lozano-Perez et al. (1984), the peg-in-a-hole example illustrates the terminology of preimages and forward projections within a $\mathbb{R}^2 \times \mathbb{S}^1$ C-space. Regions X and Y represent the start and goal regions for the task of inserting a peg (i) into a hole (ii). The angle bracket represents the uncertainty in the commanded velocity (control strategy) θ . Boundaries of the strong and weak preimages, as well as forward projection of X are illustrated.

E.1 Bounding $F_{\sigma_G}(R_i)$

The size of the forward-projection of R_i with the global control strategies σ_G , $F_{\sigma_G}(R_i)$, is bounded by the maximum error $\delta(t)$ that can accumulate during the execution of the global control strategy. We start by deriving $C_{a,i}$ as the bound for $F_{\sigma_G}(R_i)$ for robot D_i , where $i \in Z_{n-2}$, using the global control strategy σ_G .

Let $t_\theta = \theta / \hat{\omega}$ be the time it takes the robot to rotate by angle θ while in control state 1. The forward-projection of R_i using σ_G for the control of a single robots from R_i to A_i , $F_{\sigma_G}(R_i)$ for $i \in Z_{n-1}$, is equal to $\delta(t_{2\pi})$. The reason for this is that our microrobots can only turn one way, and correcting 3 error may require a trajectory up to $2\pi r$ in length (the robot may have to complete a full circle). Consequently, global control strategies σ_G will not be able to reduce the control error to below $\delta(t_{2\pi})$, thus for robots D_i with $i \in Z_{n-1}$, $F_{\sigma_G}(R_i)$ is bounded by cylinder $C_{a,i}$ with $r_{a,i} = \delta_{xy}(t_{2\pi})$ and $h_{a,i} = \delta_\theta(t_{2\pi})$. $C_{a,i}$ is centered around the target configuration \mathbf{a}_i , as mentioned in Sec. 6.3.3.

F Design Specifications for Five STRING Microrobots

This section, provided for reference, is reproduced with permission from (Donald et al., 2008b). The steering-arms in all the five species are fabricated out of $1.5 \mu\text{m}$ thick polysilicon layer. Table F.1 summarizes the design parameters for the steering-arms defining each of the species. An annotated design of the steering-arm is shown on Fig. F.1 to provide a reference for the parameters in Table F.1. The layer of evaporated chromium is 76 nm thick, except for species 4, where the nominal thickness of chromium ranges from 76 to 92 nm to compensate for design-specific local effects of galvanic attack. The steering-arm designs were determined based on closed-form equations (Donald et al., 2006), finite-element models, and empirical data, such that their transition voltages are *reproducibly* confined to the voltage ranges shown in Fig. 18(b).

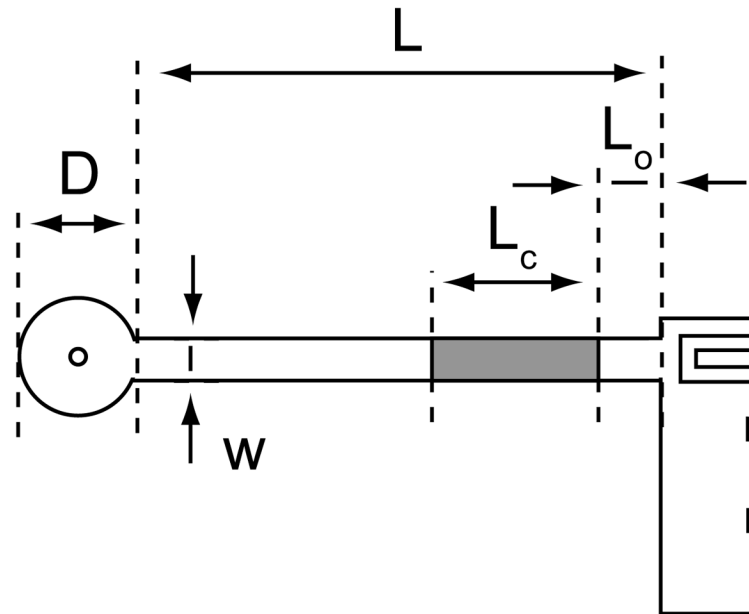


Figure F.1.

Reprinted with permission from (Donald et al., 2008b). Parameters of the steering-arms specified in Table F.1 classifying the five microrobot species.

Table F.1

Reprinted with permission from (Donald et al., 2008b). Design parameters for the steering-arms for the five microrobot species.

Species	Arm-orientation	D [μm]	w [μm]	L [μm]	L _c [μm]	L _o [μm]	Dimple height [μm]
1a	left side	20	10	120	33.5	9.5	0.75
1b	left side	40	8	80	59.5	9.5	1.2
2	right side	20	10	120	33.5	1.5	0.75
3	right side	36	8	107	94.5	1.5	0.75
4	left side	30	8	130	91.5	1.5	0.75

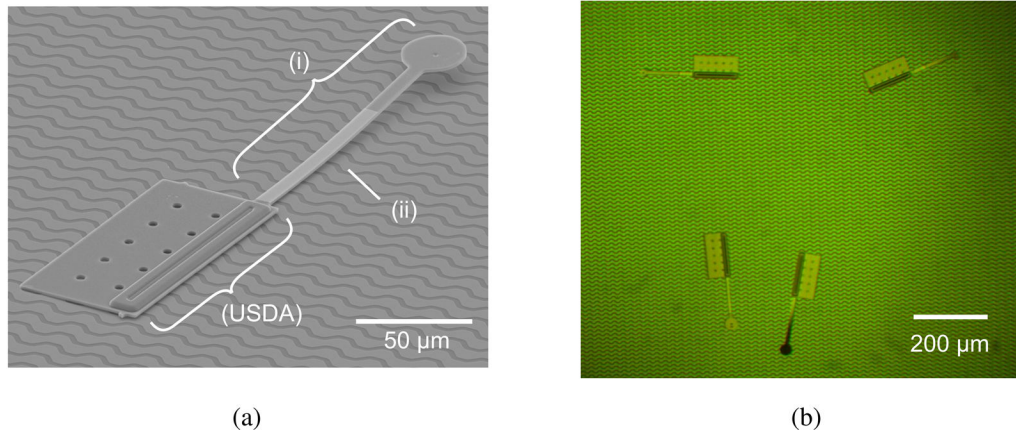


Figure 1. Scanning-electron micrograph of a stress-engineered MEMS microrobot (MicroStressBot) (a), and optical micrograph of four microrobots (b). **a:** The microrobot consists of an untethered scratch-drive actuator (USDA) that provides forward motion, and a curved steering-arm actuator (i) that determines whether the robot moves in straight-line motion or turns. A lithographically-patterned layer of chromium defines the curvature of the steering arm (ii). **b:** Four different MicroStressBots on their operating environment. The robots are differentiated by the design of their steering-arm actuators.

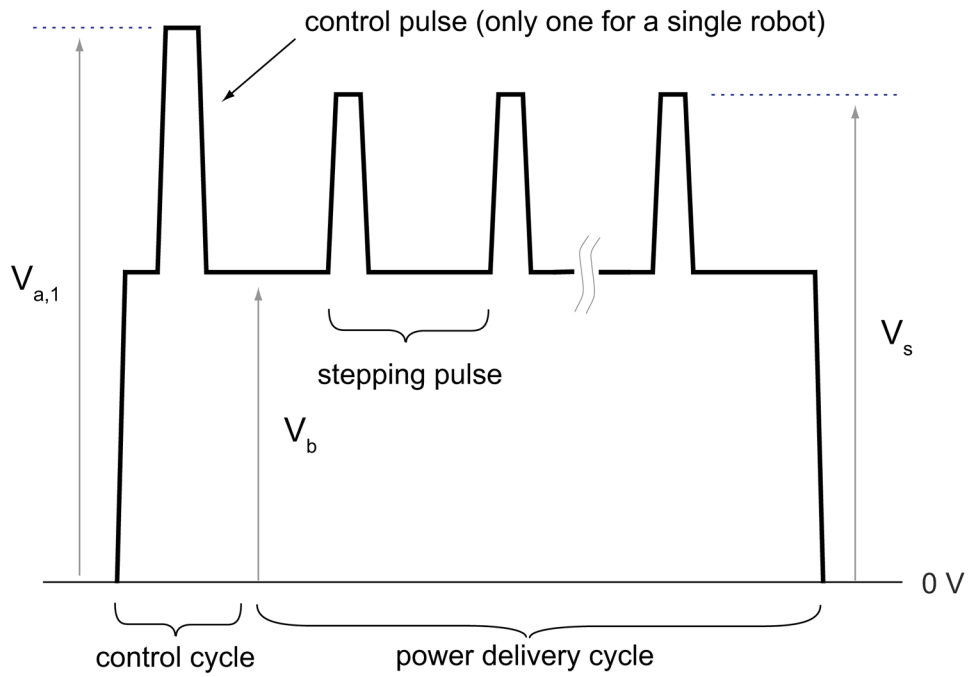


Figure 2.
The control and power delivery waveform for a single MicroStressBot.

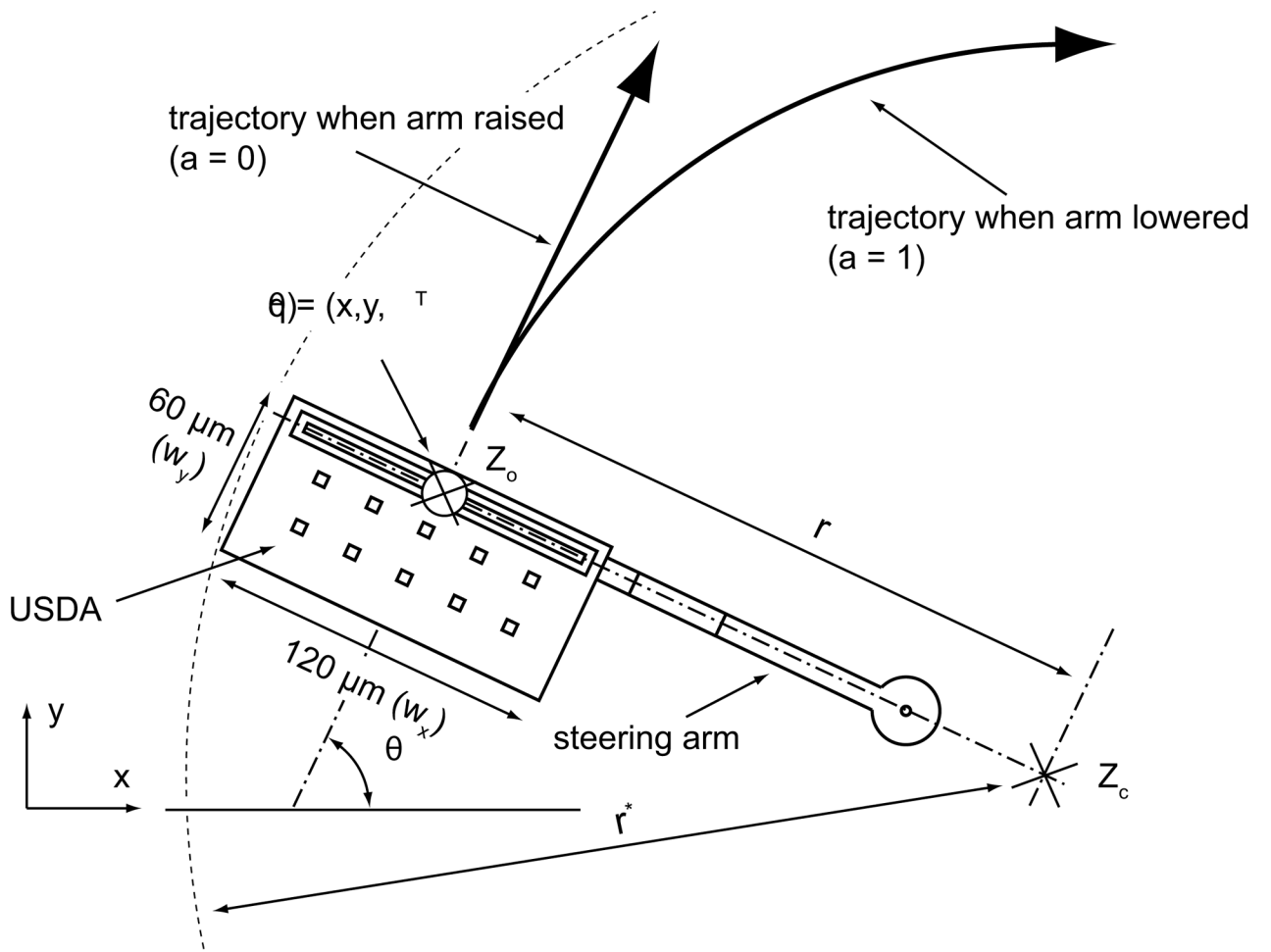


Figure 3.
Kinematics of a MicroStressBot.

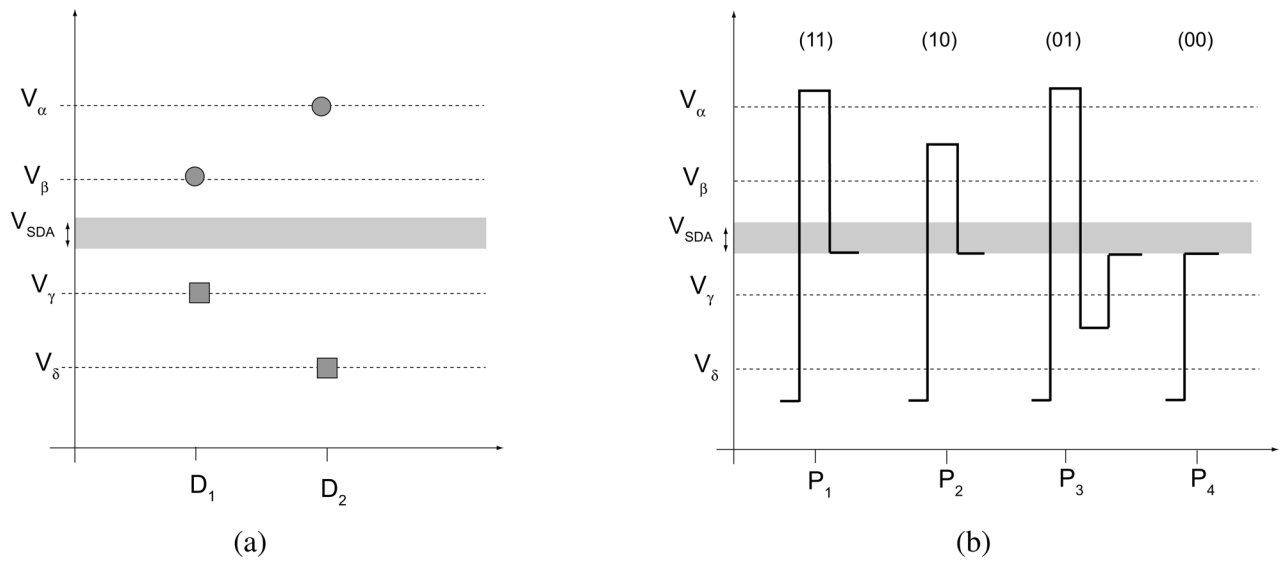


Figure 4. (a) Transition voltages for a system of two microrobots with nested hysteresis gaps (NHG), and (b) programming cycles for the four control primitives that access all system hysteresis states.

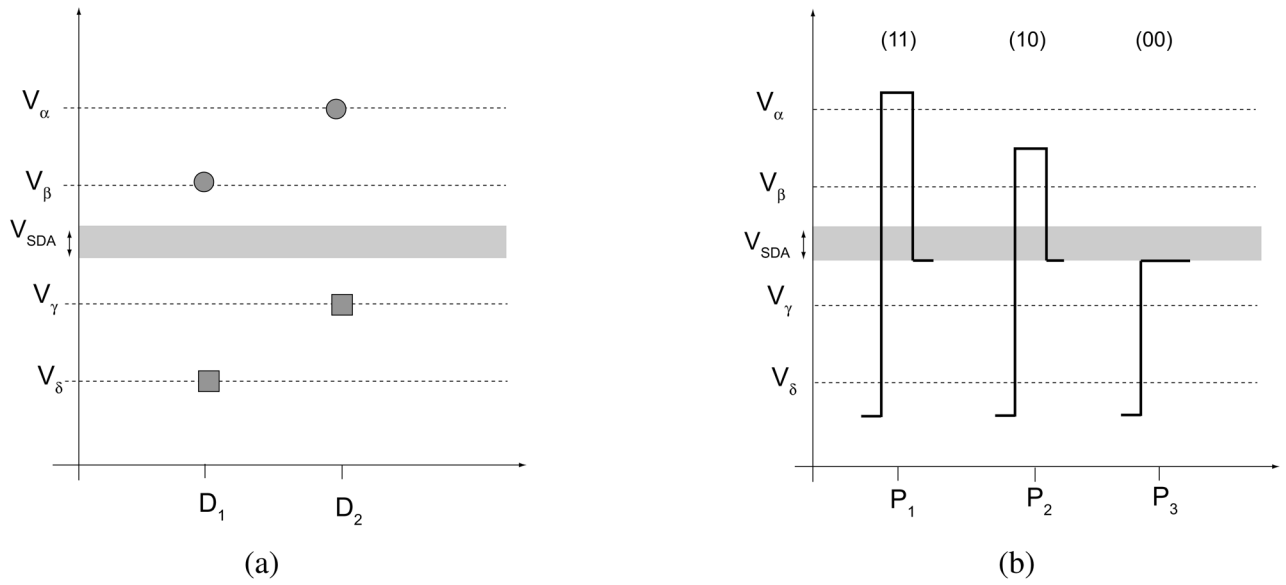


Figure 5. (a) Transition voltages for a system of two microrobots with non-nesting hysteresis gaps, and (b) programming cycles for the control primitives that access three of the hysteresis states for the system.

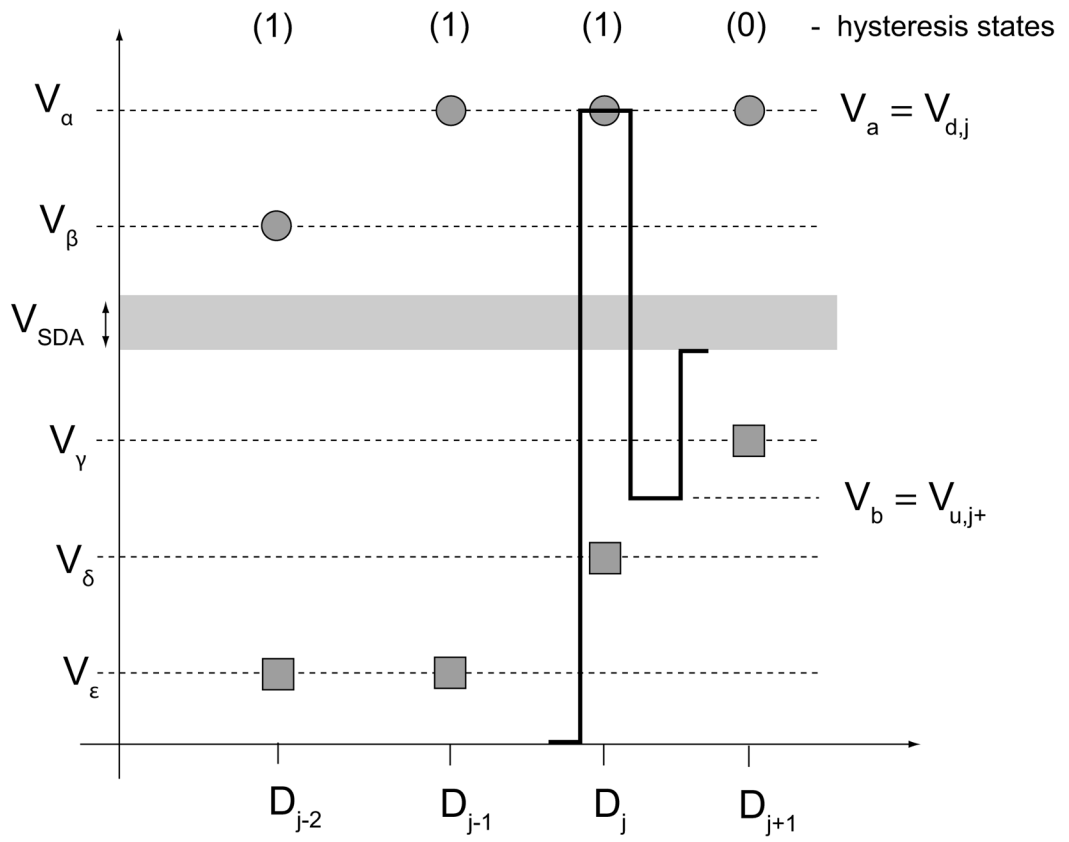


Figure 6.
Construction of a STRING control matrix.

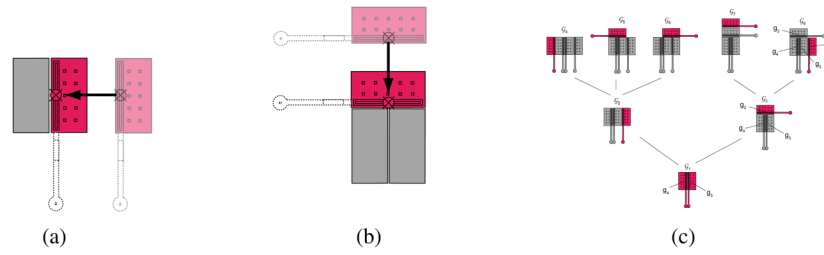


Figure 7.

(a,b) docking locations for the MicroStressBot. Restricted docking locations simplify the calculation of the stable shapes by ensuring that all torques sum to zero. Robots can dock either aligned along the long edge of a single device (a), or aligned with the short edges of two devices (b). Both in (a) and (b) the robots in the stable structure can face in either direction. **(a)** A tree-structure enumerating valid assembly sequences for a system of four microrobots. Note that not all feasible assemblies are shown. The assembly plan for G_8 is shown.

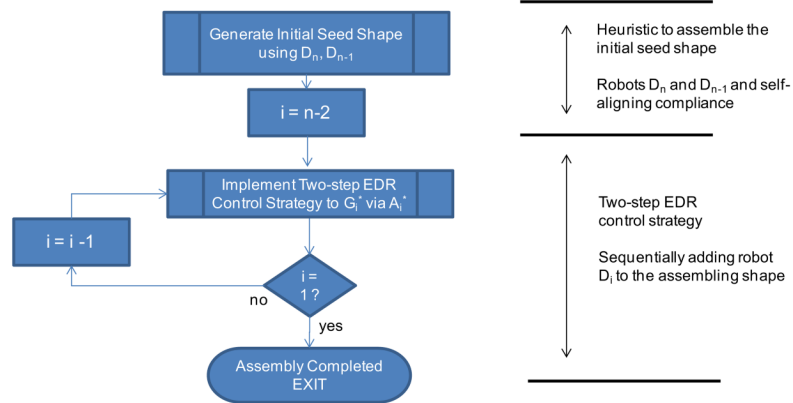


Figure 8.

Flow chart illustrating the process of assembling the target goal shape G_A using EDR framework. The shape is assembled by first generating the initial seed shape, and then by iteratively implementing a two-step EDR control strategy to sequentially maneuver single robots to dock with the seed shape, progressively assembling the target goal shape.

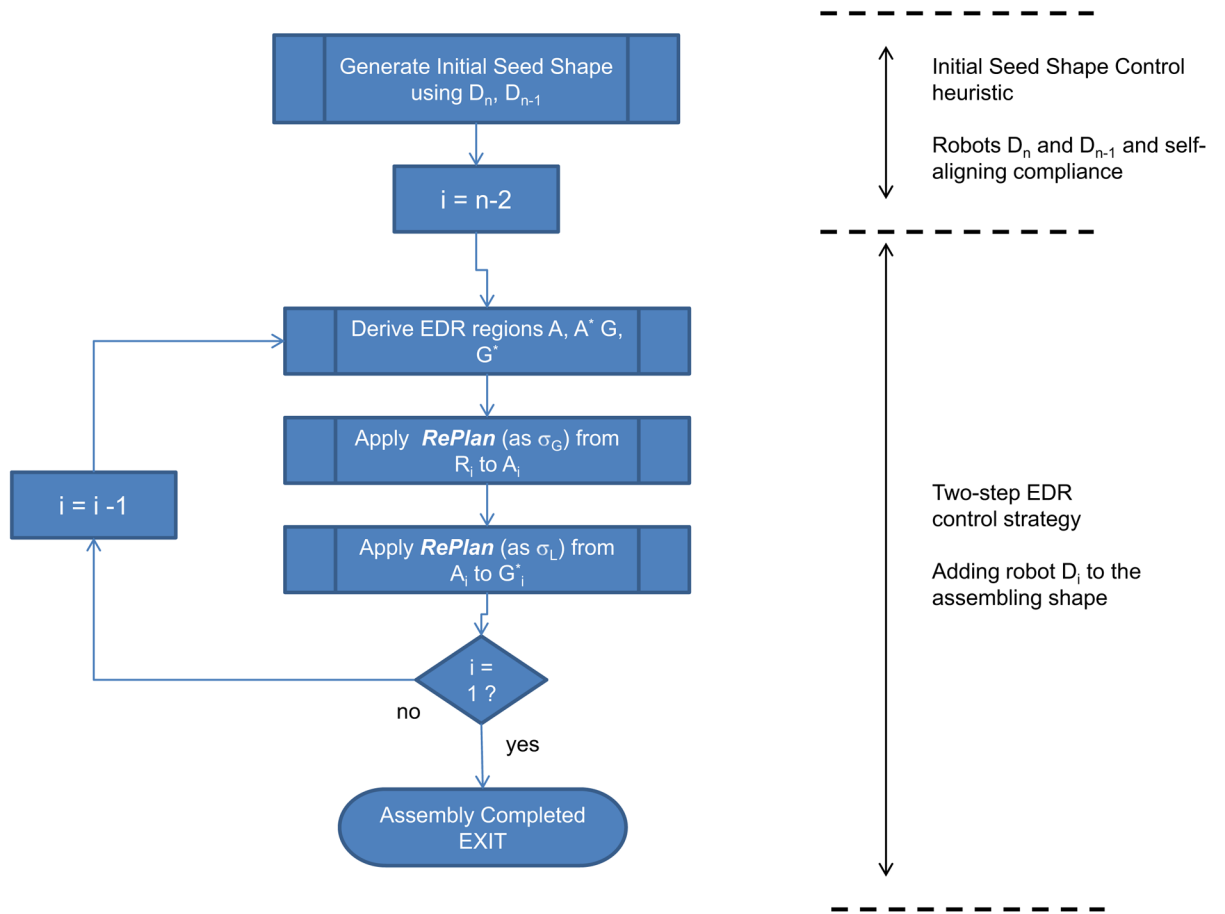


Figure 9. A flow chart outlining the control heuristic that was used to assemble the initial seed shape \mathcal{G}_i using robots D_n and D_{n-1} .

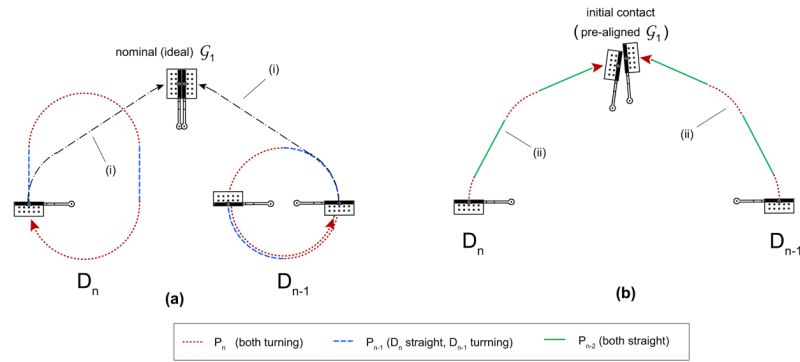


Figure 10.

Assembly of the seed shape \mathcal{G}_1 : **(a)** Orienting robots D_n and D_{n-1} to allow them to collide by following a single Dubins trajectory (i). **(b)** Maneuvering of D_n and D_{n-1} towards a common point using a modified Dubins trajectory ((ii), extra turn segments added to permit the trajectories to be adjusted if both robots turn less than expected due to control error). In the event of increased accumulation of the control error, the docking could be aborted and the heading of D_n and D_{n-1} re-adjusted, however due to the ability of self-aligning compliance to generate \mathcal{G}_1 with relatively large initial misalignment, aborting docking was never required during our experiments.

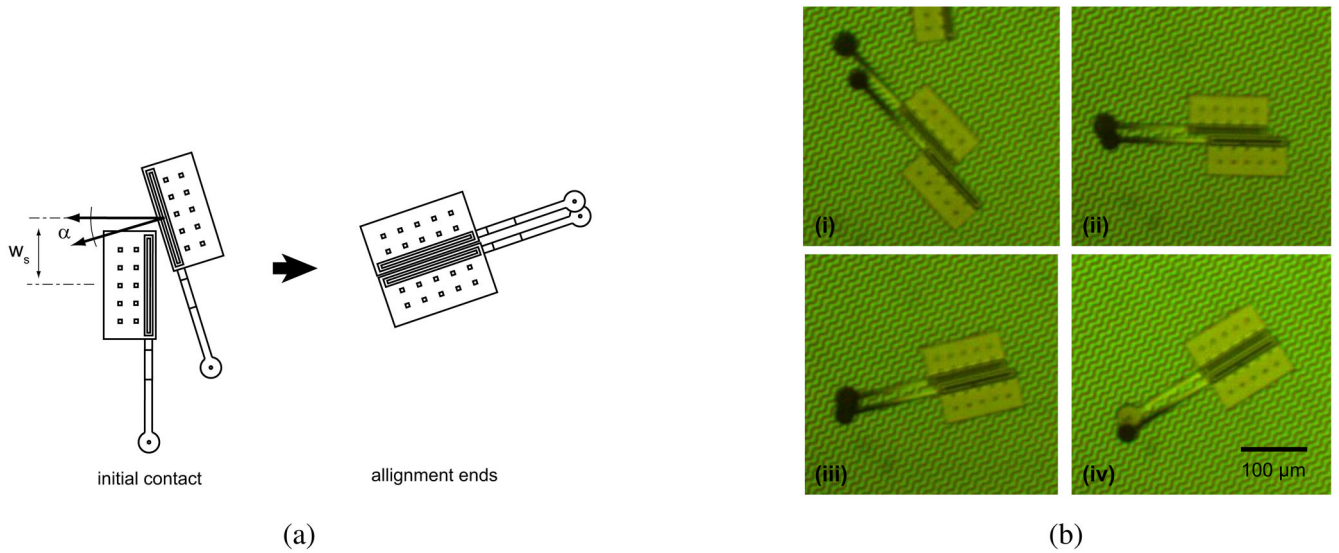


Figure 11.

Self-aligning compliance of \mathcal{G}_1 : **(a)** At initial contact, the robots are misaligned by an offset w_s and a relative rotation α . Under subsequent application of straight-line motion control primitives (this occurs automatically when actuating robot $D_{n-2} \dots D_1$ towards the goal due to the structure of the STRING control matrix), the robots rotate around slightly offset axis until their chassis are aligned. **(b)** Optical micrographs taken at the time of initial contact (i) and successively through the alignment process (ii)–(iv) showing the self-alignment of two robots. The initial seed shape \mathcal{G}_1 rotated by 79° during this self-alignment.

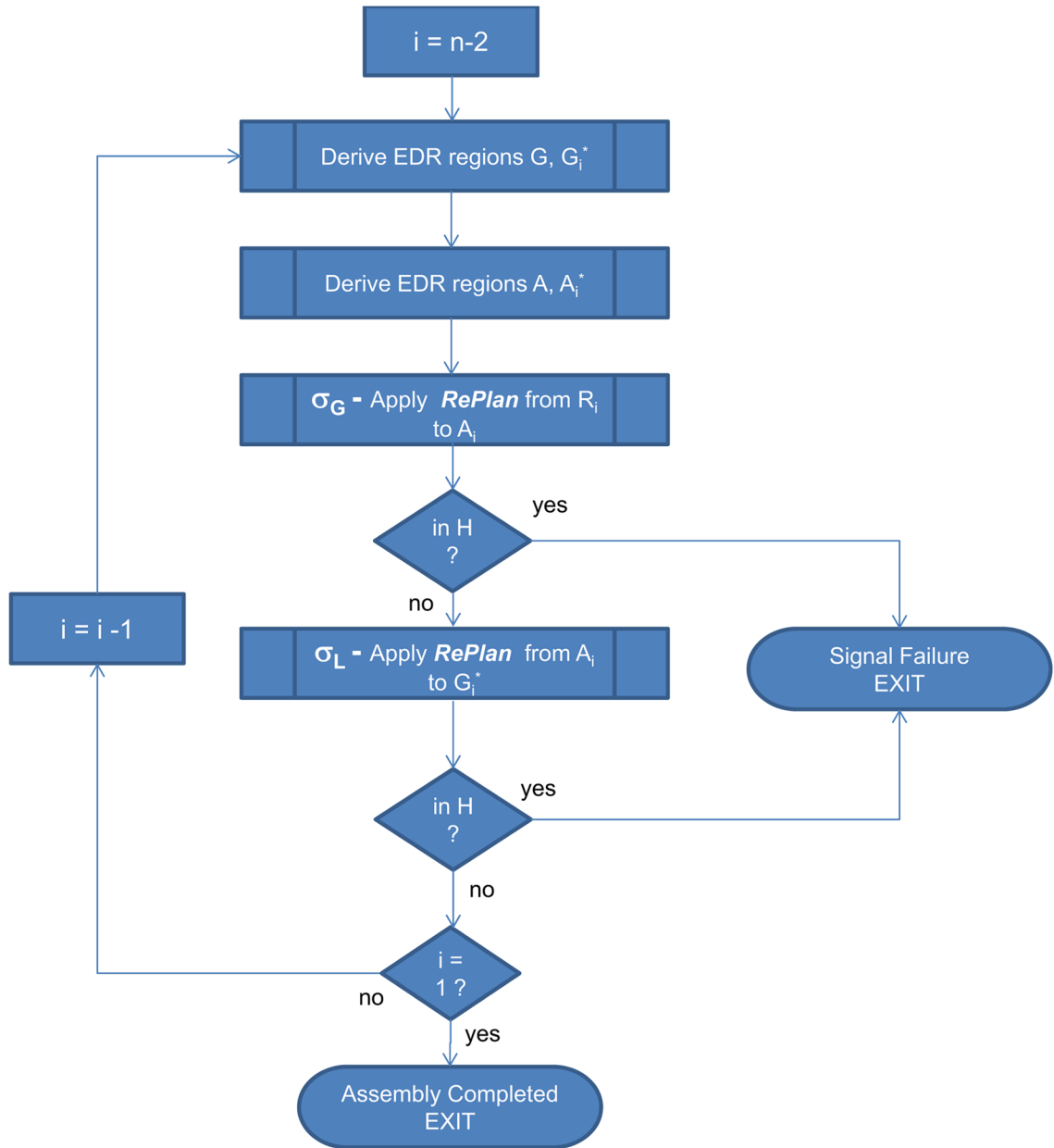


Figure 13. Flow chart illustrating the implementation of the two-step EDR control strategy for progressive assembly of the goal shape \mathcal{G}_n using robots $D_i, i \in \{n - 2, \dots, 1\}$.

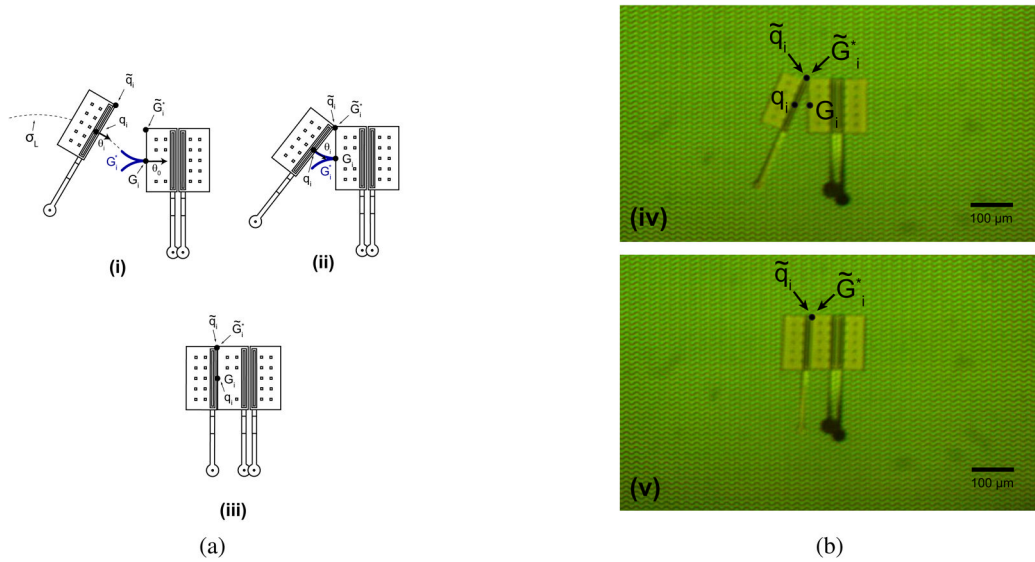


Figure 14.

Compliant interaction of a single robot docking with the assembling shape: **(a)** The mechanism of compliance between a docking robot and the assembling shape. The robot D_i approaches the assembling shape along a local trajectory defined by σ_L (i). G_i is the goal configuration of D_i , but because D_i will pivot around the corner that first makes the contact (q_i), the expanded goal configuration G_i^* (blue) forms a torus in C-space. It is thus easier to track the configuration of the corner, q_i , with respect to the expanded corner goal configuration \tilde{G}_i^* , which simply forms a cylinder in C-space. The robot enters G_i^* as q_i enters \tilde{G}_i^* (ii), and upon the application of a straight-line control primitive, D_i aligns with the assembling shape. **(b)** Optical micrographs showing a docking robot at the time of the initial contact with the assembling shape (iv) and after successful alignment (v). In our experiments the alignment was successful as long as θ_i had a deviation less than approximately 45° from that of the nominal goal configuration $g_i \in G_i$. The tolerance for angular misalignment is smaller than in the case of the self-aligning compliance (Section 6.2) because only the docking robot is free to rotate and align.

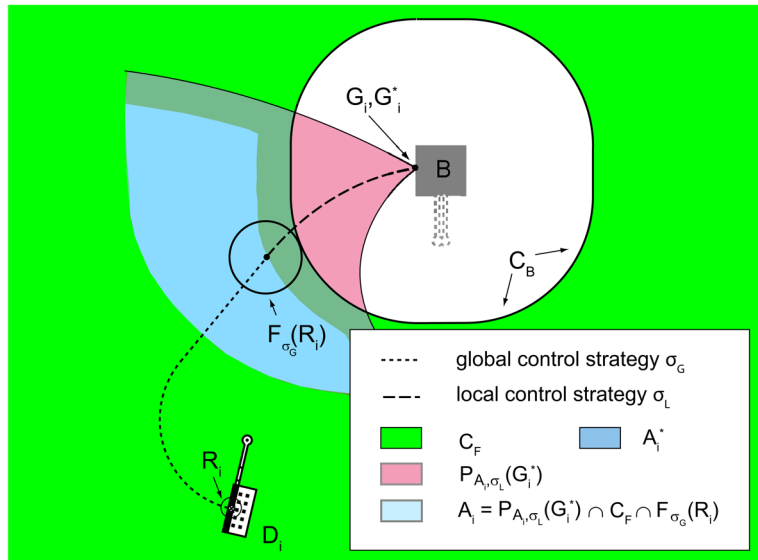


Figure 15. Regions A_i , A_i^* , C_F , and $F_{\sigma_G}(R_i)$ defining the two-stage EDR control strategy to maneuver the individual robots to their expanded goal regions G_i^* .

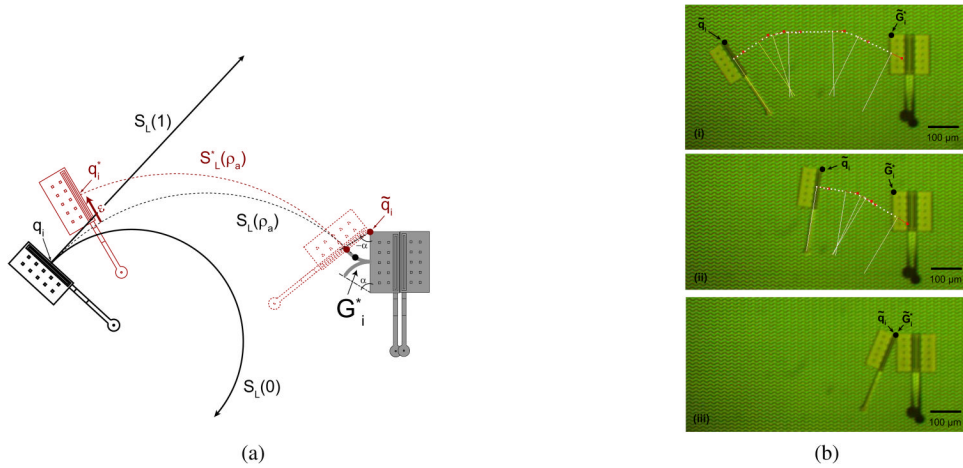


Figure 16.

Local control strategy σ_L : **(a)** The trajectory of the local control strategy projected onto the x - y plane of the C -space. A nominal (error-free) interpolated turning trajectory $S_L(\rho_a)$ is generated by interleaving straight line motion and turning control primitives. The curvature of S_L can vary between straight line motion, $S_L(1)$, and turning, $S_L(0)$. $S_L(\rho_a)$ is constructed such that it intersects the expanded goal region G_i^* . As the motion of the robot is perturbed by control error (ϵ), the interpolated turning trajectory is adjusted ($S'_L(\rho_a)$) such that it again intersects G_i^* . In practice we were tracking the configuration of the corner \tilde{q}_i as it was approaching the assembling shape. **(b)** Optical micrographs of a microrobot approaching an assembling shape. Overlay of planned nominal interpolated turning trajectories are shown in white. We implemented the local control strategies by tracking the corner of the robot \tilde{q}_i as it was approach the expanded goal configuration of that corner, \tilde{G}_i^* .

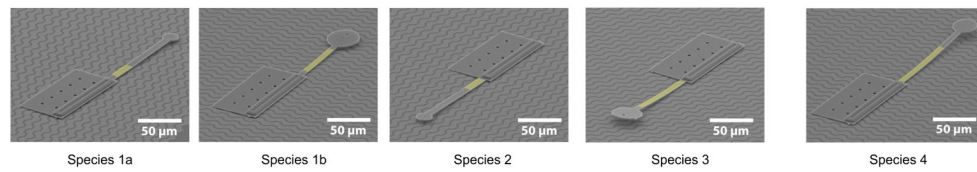


Figure 17. Scanning-electron micrographs of the five microrobot species used to implement microassembly. Yellow color is used to highlight the areas of the steering-arms covered by the layer of chromium. *Reprinted with permission from (Donald et al., 2008b).*

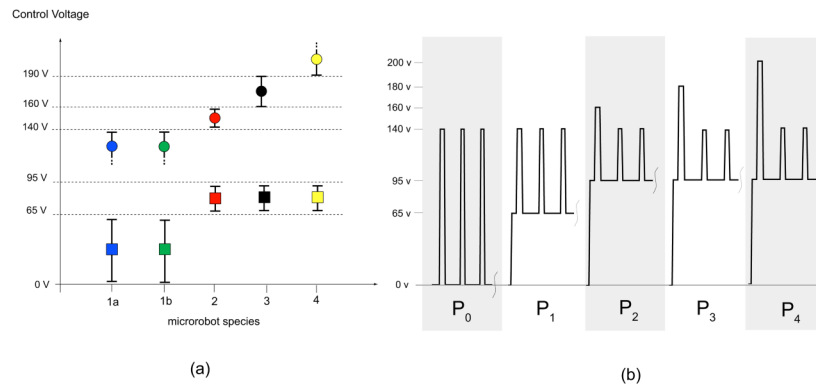


Figure 18. (a) Transition voltage ranges and (b) corresponding control primitives used to control the five microrobot species. *Based on data from (Donald et al., 2008b).*

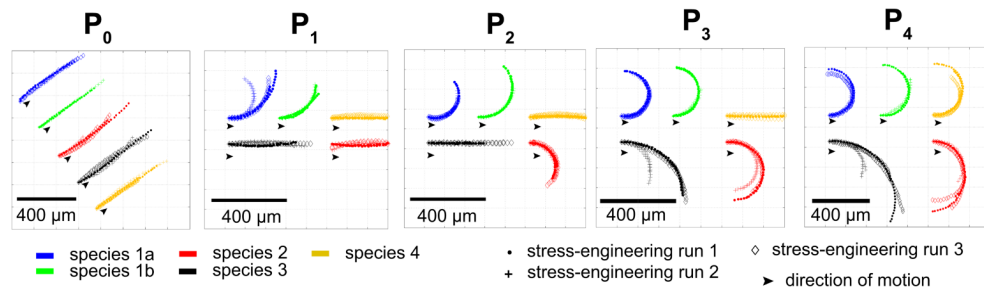


Figure 19. Reproducibility of motion during the application of the five control primitives P_0 , P_1 , P_2 , P_3 , P_4 . Reprinted with permission from (Donald et al., 2008b).

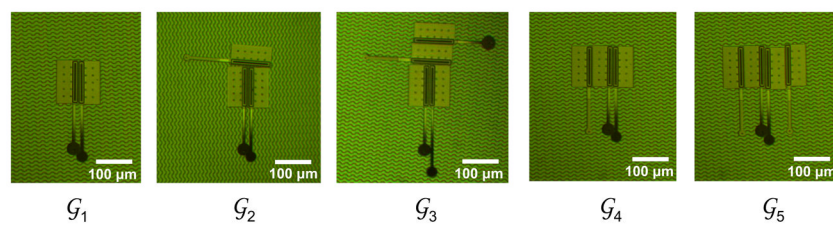


Figure 20. Optical micrographs of five types of target shapes assembled using our microrobots. *Reprinted with permission from (Donald et al., 2008b).*

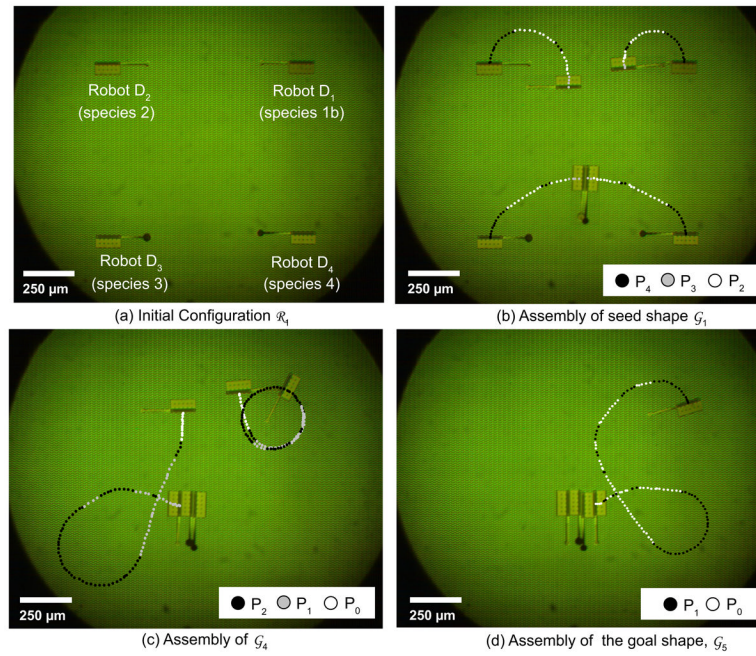


Figure 21.

Composite optical micrograph of experimental assembly-data using four robot D_i , $i \in \{1, \dots, 4\}$ from species 1b, 2, 3 and 4, respectively. **a:** Initial configuration (\mathcal{R}_i) of the microrobots. **b:** Docking of robot D_3 and D_4 to form \mathcal{G}_1 , while robot D_1 and D_2 follows a closed-loop orbit. **c:** Docking of robot D_2 to \mathcal{G}_1 , forming \mathcal{G}_4 , while robot D_1 follows a closed-loop orbit. **d:** Docking of robot D_1 , forming the goal shape \mathcal{G}_5 .

Table 1

Summary of technical terms.

Term	Definition
control waveform	The voltage waveform used to control the state of the steering-arm actuator and to power the untethered scratch-drive actuator.
transition voltage	The voltage at which the steering arm changes its state.
snap-down voltage (V_d)	The control waveform raised above this voltage causes the steering-arm to snap down.
release voltage (V_u)	The control waveform lowered below this voltage causes the steering arm to be released from the substrate.
hysteresis gap	The difference between the snap-down and release voltage.
microrobot species	A set of identical microrobots. Different species are differentiated by the designs of their steering arm actuators.
control primitive	A specific control waveform that actuates the robots along straight-line or curved trajectories.
control sequence	A sequence of control primitives that directs the motion of our microrobots
nominal microrobot trajectory	A microrobot trajectory in absence of control error.
NHG	(Sec. 4) Nested hysteresis gaps. Denotes a system of nested pairs of transition voltages.
STRING	(Sec. 4) Strictly non-nested hysteresis gaps. Denotes a system of transition voltages that is not nested.
ESat	(Sec. 4) Electromechanically saturated system. A system of steering arms that contains all combinations of independent snap-down and release voltages.
SESat	(Sec. 4) Symmetric ESat system. An ESat system containing an equal number of snap-down and release voltages.
degenerate pair	(Sec. 4) Two microrobots with steering-arms having identical snap-down and release voltages.

Table 2

Comparison of the control voltage bandwidth requirements, ξ_n , and the number of control pulses in the STRING control primitives of n -robot NHG, STRING and SESat systems.

	NHG	STRING	SESat
ξ_n	$2n$	$n+1$	$2\lceil\sqrt{n}\rceil$
Number of control pulses	1	2	$O(2n)$
Number of robots at $2\delta_v = 10$ V	10	20	100

Table 3

Summary of symbols used in this section.

Symbol	Definition
$D_i, i \in \{1, \dots, n\}$	Robot i
R_i and \mathbf{r}_i	The starting region and configuration of robot i , respectively
G_i and \mathbf{g}_i	The goal region and configuration of robot i , respectively
A_i and \mathbf{a}_i	The intermediate goal region and configuration of robot i , respectively
Q_i and \mathbf{q}_i	The current region and configuration of robot i , respectively
H	The failure region (signaling failed assembly)
σ_G	The global control strategy
σ_L	The local control strategy
C	The configuration space (C-space)
B	The region of the C-space occupied by the assembling shape
C_B	The proximity space of B (where robots <i>might</i> collide with B)
\mathcal{G}_1	The initial seed shape
$\mathcal{G}_l, l \in \{2, \dots, k-1\}$	An intermediate assembling shape
\mathcal{G}_k	The target goal shape

Table 4

Precision of Microassembly.

		Goal Configurations					
Initial Configurations	\mathcal{G}_1	\mathcal{G}_2	\mathcal{G}_3	\mathcal{G}_4	\mathcal{G}_5	Average	
\mathcal{R}_1	96 ± 4% (3 runs)	98 ± 3% (2 runs)	96 ± 2% (2 runs)	96% (1 run)	93% (1 run)	96 ± 3% (3 runs)	
\mathcal{R}_2	99 ± 2% (2 runs)	98% (1 run)	93% (1 run)	89% (1 run)	na	95 ± 4% (2 runs)	
Average	97 ± 3% (5 runs)	98 ± 2% (3 runs)	95 ± 2% (3 runs)	93 ± 5% (2 runs)	93% (1 run)	96 ± 3% (5 runs)	

Table 5

Docking Accuracy.

Goal Shape	Before Compliance	After Compliance
\mathcal{G}_1	$6 \mu\text{m} \pm 7 \mu\text{m}$	$2 \mu\text{m} \pm 3 \mu\text{m}$
	$> 50 \mu\text{m}$ (purposefully misaligned)	$9 \mu\text{m} \pm 8 \mu\text{m}$
$\mathcal{G}_2 - \mathcal{G}_5$	–	$3 \mu\text{m} \pm 3 \mu\text{m}$

Algorithm 1***RePlan***: Iterative Re-planning Control Algorithm

Input: M STRING control matrix
 G the goal region (strategy specific)
 F the planner function (strategy specific)
Output: $\mathbf{q} \in \{G, H\}$ (Success or Failure)
REGISTER $\mathbf{q} \in Q$
repeat
 PLAN $S \leftarrow \mathcal{F}(\mathbf{q}, \mathbf{g} \in G)$
 EXECUTE S for t_x seconds
 REGISTER $\mathbf{q} \in Q$
until $\mathbf{q} \in G$ or $\mathbf{q} \in H$
RETURN \mathbf{q}
