Turning-rate Selective Control : A New Method for Independent Control of Stress-engineered MEMS Microrobots

Igor Paprotny*†**††, Christopher G. Levey‡, Paul K. Wright†§ and Bruce R. Donald¶‖**
*Berkeley Sensor & Actuator Center (BSAC), University of California, Berkeley, CA, USA.
†Center for Information Technology Research in the Interest of Society (CITRIS), University of California, Berkeley, CA, USA.
‡Thayer School of Engineering, Dartmouth College, Hanover, NH, USA.
§Department of Mechanical Engineering, University of California, Berkeley, CA, USA.
¶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.
††Email: igorpapa@eecs.berkeley.edu

***Abstract*—We present a novel method for independently controlling multiple stress-engineered MEMS microrobots (MicroStressBots) through a single, global, control signal. Called Turning-rate Selective Control (TSC), this new technique employs designed variations in turning rates between individual microrobots to differentiate their motion. Despite all robots moving simultaneously and being identical except for exhibiting different turning rates, TSC can individually and independently position the robots' centers of rotation within a planar configuration space. This allows the individual robots to be independently maneuverable to within a distance equal to the turning radius (approximately half of a microrobot width) away from an arbitrary location (configuration excluding rotation) in $\mathbb{R}^2$. We introduce the theory behind TSC and, by using fabricated microrobots, show experimental results that confirm the feasibility of TSC for controlling multiple MicroStressBots through a single, global, control signal. We conclude by discussing how TSC can extend the maximum number of independently controllable MicroStressBots beyond previously published approaches.**

I. Introduction

The last decade has seen the development of several novel microscale mobile robotic systems, such as electrostatically-driven stress-engineered MEMS microrobots (MicroStressBots) [5], resonating stepping robots [9], stick-slip magnetic walkers [8], and microscrew-based swimmers [10]. Virtually all envisioned microrobotic applications (for example neurological; [1]) rely on the combined actions of many microrobots. However, control of many microrobots through a single global control signal has been demonstrated by only a few groups [6, 4]. Simultaneous control of several microrobots is significantly more challenging than control of single microrobots because one must overcome high level of underactuation present in such systems. The high level of underactuation is a result of the limited ability of the microrobots to decode a global control signal.

Donald et al. [7] analyzed the control voltage complexity of stress-engineered MEMS microrobots, defined as the number of distinct voltage levels of the control waveform required to independently maneuver n robots. It was shown that MicroStressBots can exhibit sub-linear control voltage complexity ($O(\sqrt{n})$) if their steering arms are designed to have *Symmetric Electromechanically Saturated* (SESat) hysteresis gaps. In this work, we report on an new method to independently maneuver (independently control) MicroStressBots on a planar substrate. Called *Turning-rate Selective Control* (TSC), this method allows for independent control of microrobots that are only differentiated by their turning rates. Because MicroStressBots can be *designed* to have different turning rates, TSC fits within the paradigm of Global Control Selective Response (GCSR), where independent control of multiple microrobots is enabled by engineering the robots to exhibit different behavior during portions of the global control signal. Our TSC idea is very simple, yet highly effective, as our experimental results show (Sec. IV). The mechanism of TSC is both different and independent of the mechanism of SESat-based control, allowing TSC to be used to complement SESat-based control and increase the maximum number of independently controllable MicroStressBots.

We present the concept behind TSC, and show experimental results confirming the feasibility of TSC for enabling independent control of stress-engineered MEMS microrobots. There are no simulations in this paper; as described below, the presented control method was validated (up to proof-of-concept) using physically fabricated MicroStressBots.

The paper is structured as follows: in Sec. II, we introduce the design, the kinematics, and notation for MicroStressBots. Sec. III describes the theory behind TSC, while experimental results confirming its feasibility are presented in Sec. IV. A concluding discussion summarizing the benefits and limitations of TSC is included in Sec. V.

II. STRESS-ENGINEERED MEMS MICROROBOT (MICROSTRESSBOT)

Scanning-electron micrograph (SEM) of a MicroStressBot is shown in Fig. 1. The robot is approximately 260 μm $\times$ 60 μm $\times$ 10 μm in size, and has two actuated internal degrees of freedom; an untethered scratch-drive actuator (USDA) that provides forward motion, and a steering-arm actuator that determines whether the robot moves in a straight-line or turns. The steering arm actuator consists of a cantilever beam (80-150 μm long) with a circular pad at its end (20-50 μm in diameter). The robot chassis, including the USDA and the steering arm actuator, is fabricated from polycrystaline silicon using a MEMS foundry process [3]. Post-processing is used to add a stress-layer to curve the initially planar steering arm out-of-plane. The stressor layer (consisting of evaporated chromium) is lithographically patterned to produce the exact amount of steering-arm deflection.

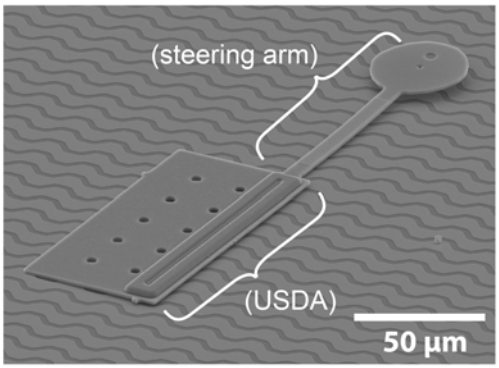


Fig. 1. SEM micrograph of a MicroStressBot. The robot consists of an untethered scratchdrive actuator (USDA) that provides forward motion (propulsion), and a steering-arm actuator that controls whether the robot moves in straight-line or turns.

MicroStressBots are actuated on a field of zirconia-insulated interdigitated electrodes. When a voltage waveform is applied between the opposing pairs of these electrodes, alternating electric potential is capacitively coupled to the microrobot chassis supplying both power for the actuation of the USDA (propulsion) and a control signal for the actuation of the steering arm. An alternating voltage waveform containing both a steering arm control signal and an USDA power-delivery waveform is called a *control primitive* [6]. When the steering arm is elevated and the USDA receives power, the robot moves along a straight-line trajectory. When the steering arm is lowered as the USDA is powered, the robot turns with a turning rate ω around its center of rotation, c_r.

The kinematics of the MicroStressBots are summarized in Fig. 2. Although a single robot is globally controllable within its $\mathbb{R}^2 \times \mathbb{S}^1$ configuration space (C-space) [11], the robot is not small-time locally controllable, since it can only turn to one side and cannot back up. MicroStressBots can be either right- or left-handed, depending on whether the steering-arm is attached to the right of left side of the USDA, respectively.

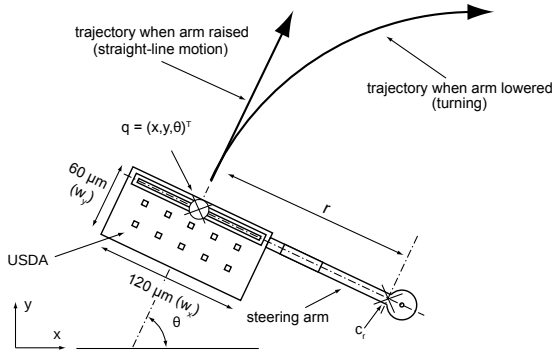


Fig. 2. Kinematics of the stress-engineered MEMS microrobot. When the steering arm is lowered, the robot turns around a center of rotation c_r.

A. Turning Mechanism

The interaction between the steering arm and the underlying substrate as the USDA is powered causes the MicroStressBots to follow a curved trajectory, i.e. to turn. The turning mechanism is illustrated in Fig. 3. During pull-down, a portion s of the steering arm comes into flat contact with the substrate (Fig. 3.a). When the USDA is subsequently actuated, s acts as a temporary anchor, restricting the motion of the tip of the steering arm. The robot follows a curved trajectory, flexing the steering arm until the restoring force of the arm equals the force applied by the USDA (Fig. 3.b). When the arm is released during periodic polarity reversal of the AC waveform, the flexure in the arm is relieved, resulting in a net change in the heading θ of the microrobot (Fig. 3.b). The amount of the steering arm flexure is highly dependent on the geometry of the steering arm actuator, making the corresponding turning rate design-specific.

B. Notation for MicroStressBot Control

Consider a system of n MicroStressBots R_i, $i \in \{1, \ldots, n\}$. Let P_j be a control primitive, drawn from an *alphabet* Σ of m available control primitives; $\Sigma = \{P_1, \ldots, P_m\}$. Application of each control primitive P_j to the power delivery substrate will cause all the MicroStressBots on that substrate to move simultaneously. However, the differences in the designs of their steering-arm actuators may cause individual robots to move along different trajectories (exhibit different behavior).

Because each robot R_i can exhibit different behavior during each control primitive P_j, we use a *control matrix* to define the relationship between the applied control primitives and the resulting microrobot behavior. An $m \times n$ control matrix $M = (a_{ji})$, where a_{ji} is a pair containing the velocity v_{ji} and the turning radius r_{ji} of robot R_i during the application of control primitive P_j, $a_{ji} = (v_{ji}, r_{ji})$. If robot R_i moves forward in straight line during the application of P_j, r_{ji} approaches infinity. While turning, the velocity v_{ji} and turning radius r_{ji} does not vary much between the control primitives

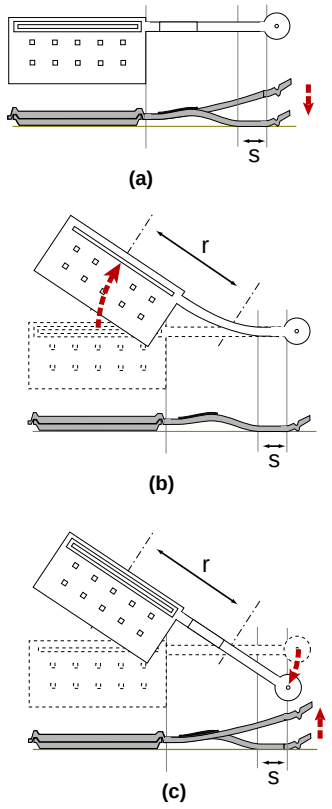


Fig. 3. Turning mechanism of the MicroStressBot (top and cross-section views). **(a)** The steering arm is pulled in contact with the substrate, and a flat region s is temporarily anchored in place by electrostatic forces. **(b)** The USDA is actuated, causing the steering arm to flex while the robot follows a trajectory with the radius of curvature r. **(c)** As the arm is released during a polarity reversal of the drive voltage waveform, the flexure of the steering arm is released. This cycle is continuously repeated, causing the robot to turn.

that cause turning, and for the remainder of this paper we shall assume that $v_i = v_{ji}$ and $r_i = r_{ji}$ for all j that correspond to turning control primitives. The turning rate of the robots is consequently $\omega_i = \frac{v_i}{r_i}$. We define the center of rotation c_r of robot R_i as $c_r(R_i)$.

To maneuver the microrobots on the power delivery substrate, control primitives will be applied in some sequence to actuate the robots. All the MicroStressBots on the substrate receive the same single global power-delivery and control signal, and all the robots will move during the application of the control primitives. We define a *control sequence* S to be a sequence of control primitives

$$S = \{\mathcal{P}_1^{t_1}, \mathcal{P}_2^{t_2}, \ldots, \mathcal{P}_k^{t_k}\}, \tag{1}$$

where $\mathcal{P}_l^{t_l}$, $l \in \{1, \ldots, k\}$, is a control primitive drawn from the alphabet of available control primitives Σ, and t_l denotes the duration the voltage waveform described by the control primitive is applied to the substrate. Consequently, S defines nominal (i.e. error free) trajectories T_i, $i \in \{1, \ldots, n\}$ for all n microrobots on the substrate. The nominal trajectories described by the control sequence S are especially useful for planning the motion of microrobots. The relationship between the control sequence and the trajectories of a system of three MicroStressBots subject to a hypothetical control sequence S and a control matrix M is illustrated on Fig. 4.

An *orbit* is a trajectory that returns the microrobot to its initial configuration. An example of a circular orbit is illustrated on Fig. 4. A c_r*-orbit* is a trajectory that returns the *center of rotation* of a microrobot to its initial location. All orbits are c_r-orbits, but the inverse is not necessary the case.

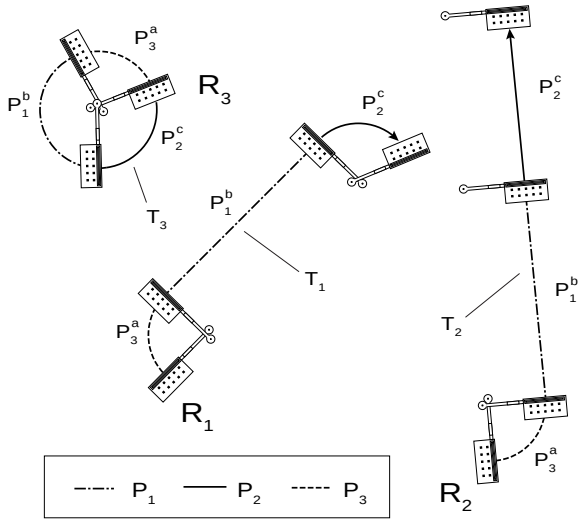


Fig. 4. Relationship between the control sequence S and trajectories for a group of three microrobots, R_1, R_2, and R_3. Nominal trajectories T_1, T_2, and T_3 for a control sequence $S = \{P_3^a, P_1^b, P_2^c\}$ are shown. Note that the behavior of R_1 and R_2 only differ during the application of control primitive P_2, while robot R_3 always turns, completing an orbit.

C. Interpolated Turning

Interpolated turning is a technique that allows a MicroStressBot to follow a curved trajectory with an effective radius of curvature (r') that can be set to any value between the turning radius (r) and infinity by interleaving straight-line and turning trajectory segments. The concept of interpolated turning is illustrated in Fig. 5. Let an interpolated turning trajectory T_I be defined by the control sequence $S_I = \{P_1^a, P_2^b, \cdots, P_1^a, P_2^b\}$, where P_1 and P_2 are straight-motion and turning control primitives, respectively. Let ρ_a and ρ_b be the fraction of the time primitives P_1 and P_2 are applied in the control sequence S_I; $\rho_a = \frac{a}{a+b}$ and $\rho_b = \frac{b}{a+b}$.

The the trajectory T_I will now approximate a turning trajectory with a radius of curvature r', $r \leq r' < \infty$, according to

$$r' = r\left(1 + \frac{\rho_a}{\rho_b}\right). \tag{2}$$

The approximation improves as the number of primitive pairs P_1-P_2 repeated within S_I increases. In our experiments,

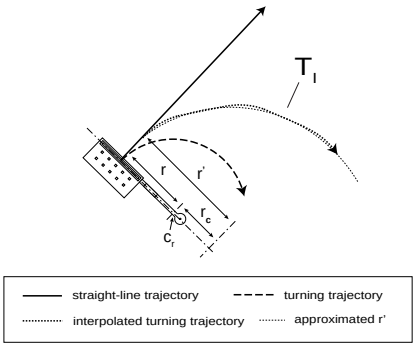


Fig. 5. Interpolated turning. The robot follows an interpolated trajectory T_I defined by the control sequence $S_I = \{P_1^a, P_2^b, \cdots, P_1^a, P_2^b\}$, where P_1 and P_2 are straight-motion trajectory and turning control primitives, respectively. $\rho_a = \frac{a}{a+b}$ and $\rho_b = \frac{b}{a+b}$, and the trajectory of S_I approximates a turning trajectory with turning radius r' according to Eq. (2).

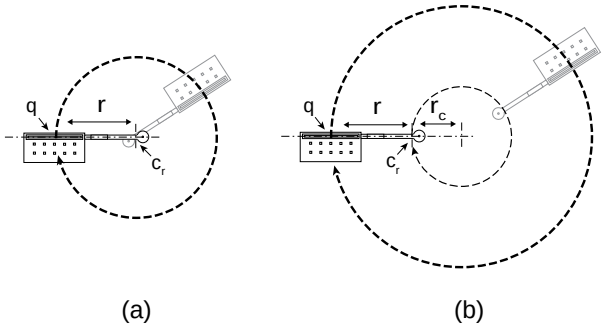


Fig. 6. Orbit trajectories. **(a)** During an orbit following a turning trajectory, the microrobot rotates with radius r around its center of rotation c_r. The location of c_r does not change (the orientation θ does change; c_r is turning in place). **(b)** During an orbit based on an interpolated turning trajectory, the microrobot follows a trajectory with a radius of curvature $r' = r + r_c$. The center of turning c_r follows a circular trajectory (its own orbit) with radius r_c, and returns to its starting location.

the P_1-P_2 pair was merged into a new control primitive where the pair was repeated more than 20 times producing a good approximation to r'.

III. Turning-rate Selective Control (TSC)

The ability to vary the turning rates by varying the steering arm designs, as well as to adjust the effective turning radius r' during interpolated turning allows us to propose a new method for independent control of MicroStressBots. Called *Turning-rate Selective Control* (TSC), this method uses a novel mechanism to achieve independent control of MicroStressBots different from our previous presented approaches. In TSC, the microrobots are differentiated *only* through design-induced variation of their turning rates and not through selective pull-down and release of their steering-arm actuators [6]. This implies that that during TSC, all the robots are either turning or moving straight at any given time.

TSC is based on engineering orbit trajectories that maneuver some robots towards goal, while other robots are returned to their starting configurations. By combining several orbits, TSC can independently translate the centers of rotation c_r of the individual robots in $\mathbb{R}^2$.

Consider two orbits that follow a turning trajectory and an interpolated turning trajectory, shown in Figs. 6.a and 6.b, respectively. When the robot is turning, the center of rotation c_r remains in the same location in $\mathbb{R}^2$, as illustrated in Fig. 6.a. However, when the robot is following an interpolated turning trajectory, the center of rotation c_r is translated along a curved path with radius of curvature $r_c = r' - r$. When the robot completes this orbit, c_r returns to its starting location (Fig. 6.b).

Because each MicroStressBot makes a full rotation when it completes an orbit, MicroStressBots with different turning rates will complete their orbits at different times. Robots that are maneuvered along interpolated turning trajectories and do not complete an orbit are *translated* in C-space. Fig. 7 illustrates the concept of TSC. Consider robots R_1 and R_2 designed such that $\omega_2 = \frac{1}{3}\omega_1$. An orbit for R_1 based on an interpolated turning trajectory translates the center of rotation $(c_r(R_2))$ for robot R_2 by a vector γ. The vector γ can be expressed as

$$\gamma = r_c \begin{pmatrix} \sin\theta_2 - \sin\left(\theta_2 + 2\pi h \frac{\omega_2}{\omega_1}\right) \\ \cos\left(\theta_2 + 2\pi h \frac{\omega_2}{\omega_1}\right) - \cos\theta_2 \end{pmatrix}, \qquad (3)$$

where θ_2 is the orientation of robot R_2, h denotes the side the steering-arm is attached to ($h = 1$ when the robot is lefthanded, $h = -1$ when robot is righthanded), $r_c = r_2' - r_2$, where r_2 and r_2' are the turning radius and the interpolated turning radius (from Eq. 2), respectively, of robot R_2, and ω_1 and ω_2 are the turning rates of robots R_1 and R_2, respectively. Note that in Fig. 7 as well as the other figures in this section, the length of the trajectories is not to scale. The starting and final locations of the centers of rotation c_r for the robots are marked by green and yellow dots, respectively. Overlapping dots indicate no change in location. In addition, where appropriate, grey color denotes past trajectories and intermediate configurations.

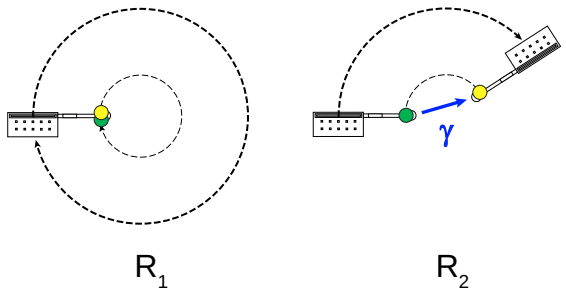


Fig. 7. The concept of Turning-rate Selective Control (TSC). Trajectories of two robots R_1 and R_2 with different turning rates: $\omega_2 = \frac{1}{3}\omega_1$. As R_1 follows a circular orbit trajectory based on interpolated turning, the center of rotation $(c_r(R_2))$ for robot R_2 is displaced by vector γ.

After translating the center of rotation $c_r(R_2)$ by displacement vector γ, $c_r(R_2)$ can be translated by a second linearly independent displacement vector γ_2 (see Fig. 8). The vectors γ and γ_2 effectively allow $c_r(R_2)$ to be translated anywhere within $\mathbb{R}^2$ without moving the center of rotation $c_r(R_1)$.

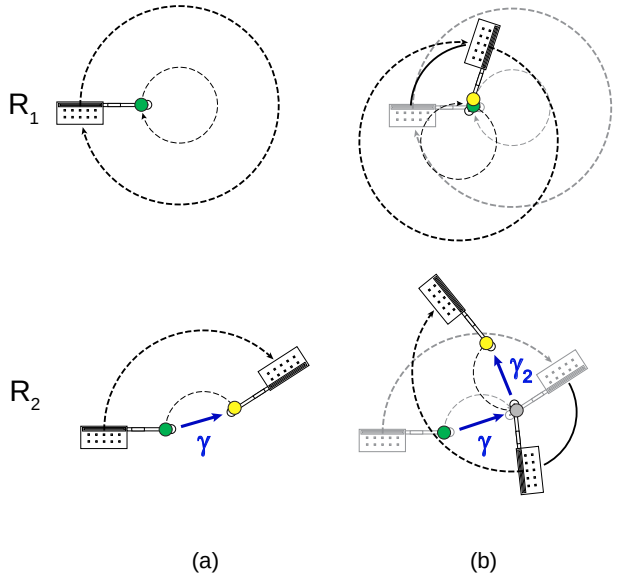


Fig. 8. Translating the center of rotation $c_r(R_2)$ by two displacement vectors that span $\mathbb{R}^2$. **(a)** Translating $c_r(R_2)$ by vector γ, while robot R_1 completes an orbit. **(b)** Both robots turn (non-interpolated) to an arbitrary angle, and $c_r(R_2)$ is translated by another vector γ_2, while robot R_1 completes another orbit. The angle between the vectors, as well as their length, can be varied arbitrary, allowing $c_r(R_2)$ to be translated in $\mathbb{R}^2$.

Correspondingly, the center of rotation $c_r(R_1)$ of robot R_1 can be translated while robot R_2 completes an orbit, and $c_r(R_2)$ returns to its starting location. One approach is to initially move robot R_2 using interpolated turning, and then construct a non-orbit trajectory that maneuvers $c_r(R_2)$ back to its original location, while displacing robot R_1 by vector η. This approach is illustrated in Fig. 9.

A. Extension to n Microrobots

TSC can be extended to independently control n MicroStressBots by individually translating a single robot while the remaining $n-1$ robots complete their orbits. Consider adding a third microrobot R_3 with turning rate $\omega_3 = \frac{2}{3}\omega_1$ to the system shown on Fig. 7. The center of rotation $c_r(R_3)$ and not $c_r(R_2)$ is translated by engineering the displacement vectors γ and $\boldsymbol{\gamma_2}$ such that they cancel each other, resulting in a non-circular c_r-orbit returning the center of rotation $c_r(R_2)$ to its starting location. Because all the trajectories complete orbits for robot R_1, its center of rotation, $c_r(R_1)$, also remains in its starting location. The center of rotation $c_r(R_3)$ is translated by a vector $\boldsymbol{\zeta_3}$. The trajectories for this implementation of TSC on three robots are shown on Fig. 10. By induction, two displacement vectors $\boldsymbol{\zeta_3}$ and $\boldsymbol{\zeta_3'}$ can be combined to form a c_r-orbit for robot R_3, allowing the translation of the center of rotation of a fourth robot R_4, and so on. We will use the shorthand TSC(n) to denote TSC of $n > 2$ microrobots.

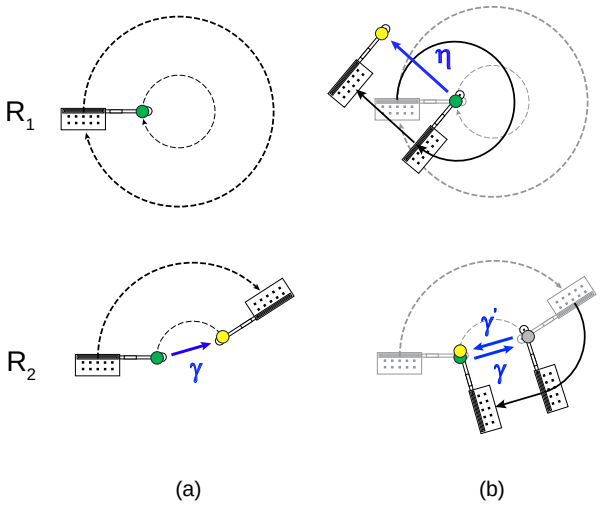


Fig. 9. Translating robot R_1 and not robot R_2. **(a)** Initially, the center of rotation $c_r(R_2)$ is translated by vector γ using an interpolated turning trajectory while robot R_1 completes an orbit. **(b)** Both robots are then rotated such that center of rotation $c_r(R_2)$ can return to its starting location using straight-line motion along vector γ' (bottom), while robot R_1 and center of rotation $c_r(R_1)$ are both translated by vector η (top).

TABLE I
EXPERIMENTALLY MEASURED TURNING RATES ω AND TURNING RADII r FOR FABRICATED MICROROBOTS R_1 AND R_2

	ω [rad/s]	r [μm]
$\mathbf{R_1}$	0.46 (0.05)	136.66 (7.6)
$\mathbf{R_2}$	0.20 (0.01)	177.50 (5.0)

IV. EXPERIMENTAL RESULTS

We fabricated two MicroStressBots designed to exhibit different turning rates. The designs and optical micrographs of the two robots are shown in Figs. 11 and 12 respectively. L and w correspond to the length and width of the steering arm, respectively. d is the diameter of the steering-arm pad and c is the length of the stressor layer applied to the steering arm.

The large difference in the turning rate of the robots was experimentally verified. Measured average turning rates ω and turning radii r for both robots are shown in Tab. I. Values in parentheses show the standard deviations. Differences of their turning rates are also clearly visible in the optical micrographs in Fig. 13. The four micrographs were taken at times $t = 2, 6, 8, 12$ seconds from the start of the experiment and include traces of the completed trajectories (black) for the two robots turning.

Fig. 14 shows the basic implementation of TSC on MicroStressBots R_1 and R_2, where one of the robots is displaced while the second robot orbits back to its initial configuration, analogous to the concept illustrated in Fig. 7. The center of rotation $c_r(R_2)$ is translated by vector γ using a interpolated

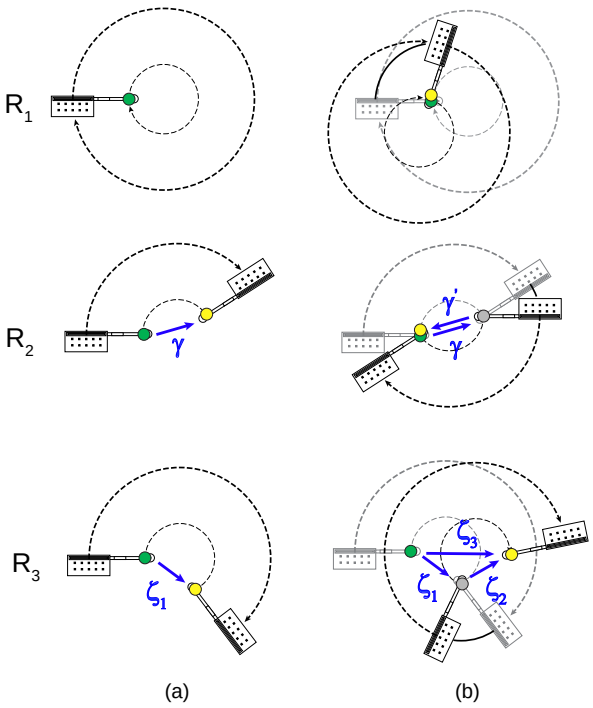


Fig. 10. Extending TSC to three MicroStressBots. A third robot R_3 ($\omega_3 = \frac{2}{3}\omega_1$) is added to the system from Fig. 7. **(a)** The robots are actuated along interleaved trajectories that results in an orbit of R_1 (top), while the centers of rotation $c_r(R_2)$ and $c_r(R_3)$ are translated by vectors γ (middle) and ζ_1 (bottom), respectively. **(b)** The robots rotate (solid line trajectory) such that an interleaved turning trajectory translates the center of rotation $c_r(R_2)$ back to starting location (middle). The center of rotation $c_r(R_3)$ is translated by a vector ζ_2 (bottom), while robot R_1 completes another orbit. The overall displacement of $c_r(R_3)$ is $\zeta_3 = \zeta_1 + \zeta_2$, while the location of $c_r(R_1)$ and $c_r(R_2)$ remain unchanged.

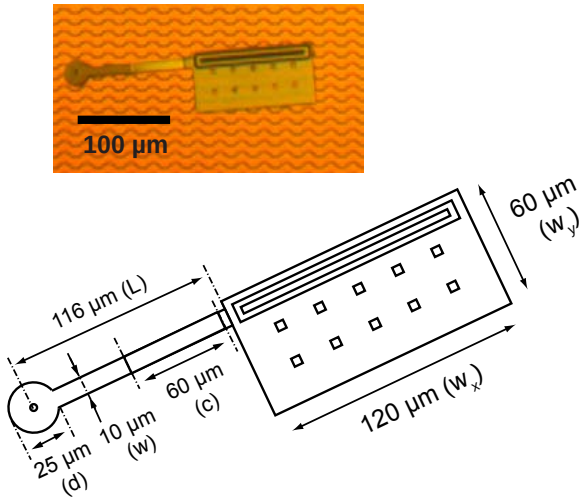


Fig. 11. Design and optical micrograph (inset) of robot R_1.

turning trajectory while robot R_1 completes an orbit. Control error attributed to a slight displacement of the radius of rotation $c_r(R_1)$.

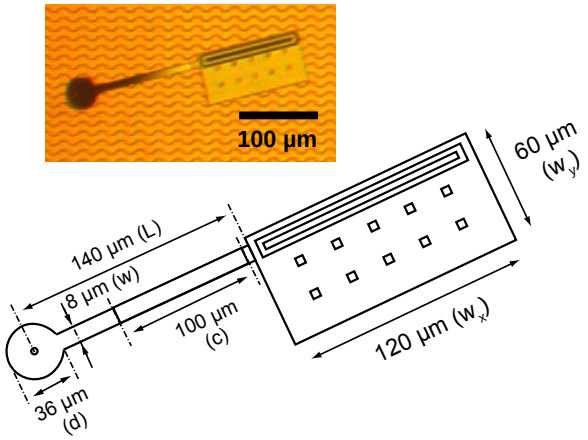


Fig. 12. Design and optical micrograph (inset) of robot R_2.

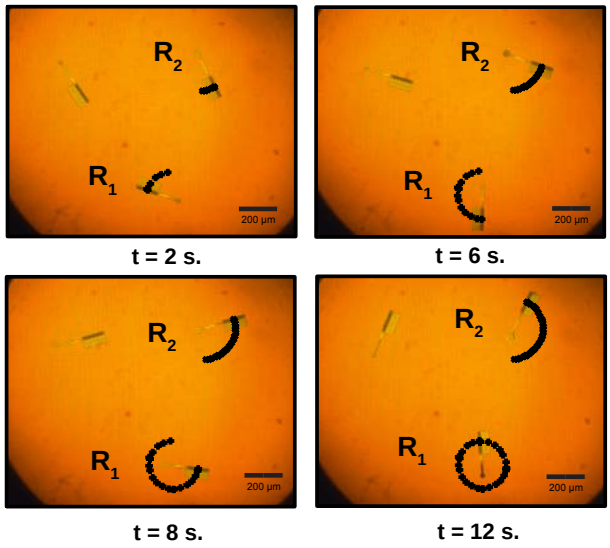


Fig. 13. Optical micrographs of robots R_1 and R_2 at times $t = 2, 6, 8, 12$ seconds while following turning trajectories. Traces of the completed trajectories for both robots are shown in black.

Fig. 15 shows the implementation of TSC where the center of rotation of one robot is translated in $\mathbb{R}^2$ by two independent vectors, analogous to the concept illustrated in Fig. 8. The center of rotation for robot R_2, $c_r(R_2)$, is translated by two vectors γ and γ_2. The robot R_1 completes two closed-loop orbits during this experiment. Again, a slight drift in the center of rotation $c_r(R_1)$ is apparent due to control error. During the experiment, the robot R_2 moves partially outside the view of the optical microscope.

Fig 16 shows the implementation of TSC to independently translate the robot with a larger turning rate, analogous to the implementation illustrated in Fig. 9. The steps in this particular implementation were conducted in the reverse order than what was suggested in Sec. III. In this implementation, robot R_2 was first translated using a combination of turning and straight-line trajectories, displacing the center of rotation

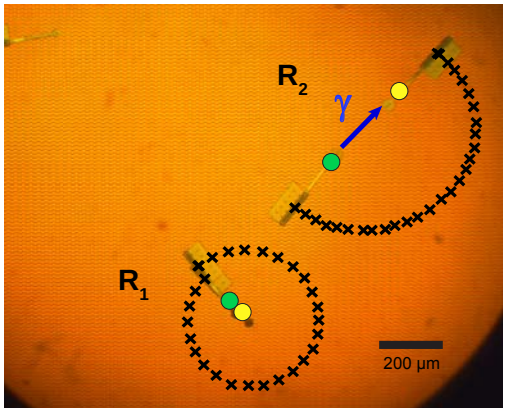


Fig. 14. Composite optical micrograph showing the basic implementation of TSC. Robots R_1 and R_2 follow interpolated turning trajectories. As robot R_1 completes an orbit and returns back to starting configuration, the center of rotation of robot R_2, $c_r(R_2)$, is translated by a vector γ.

$c_r(R_1)$ by vector η (Fig. 16.a). Consequently, the center of rotation $c_r(R_2)$ was translated back to its starting location, completing a c_r-orbit for R_2. Also in this experiment, a slight drift in the location of $c_r(R_2)$ can be observed. In practice, this control error will define the ultimate accuracy with which the TSC can be implemented, analogous to forward projections described in [7].

V. Conclusion

We presented a new method for independent control of stress-engineered MEMS microrobots (MicroStressBots). Called Turning-rate Selective Control (TSC), this method allows for the independent translation of centers of rotation c_r of MicroStressBots that are identical except for their turning rates. This allows the individual robots to be maneuvered independently to within the turning radius (r) away from arbitrary locations in $\mathbb{R}^2$. The concept of TSC was experimentally validated using physical fabricated MicroStressBots.

The mechanism of TSC is both different from, and independent of, SESat-based control [7]. This allows TSC to *augment* SESat and *increase* the number of microrobots that can be independently controlled using a single, global, control signal. When properly designed, n MicroStressBots can be controlled using $2\sqrt{n}$ independent control voltage levels [7]. Consider a hybrid SESat-TSC(u) system containing $n = u \times w$ robots, where u is the number of *groups* of w SESat MicroStressBots that are differentiated through their u different turning rates. The w robots within a group are controlled using SESat, while the robots between the u groups are controlled using TSC. Such SESat-TSC(u) system can independently control n robots using the same number of control voltage levels as a w-robot SESat system, and has control voltage complexity of $2\sqrt{\frac{n}{u}}$.

For example, suppose one wants to independently control 100 MicroStressBots through a single global control signal. According to SESat-based control, one would need 20 control voltage levels to achieve independent control. Suppose we divide the 100 microrobots into four groups, each containing

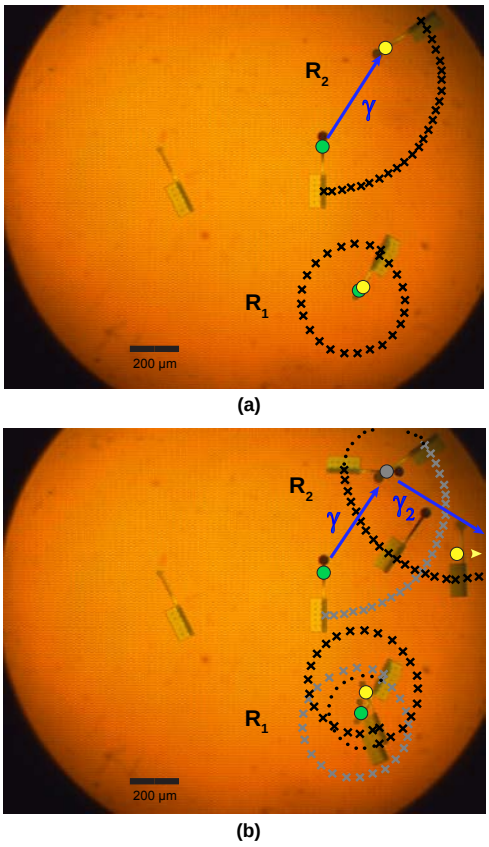
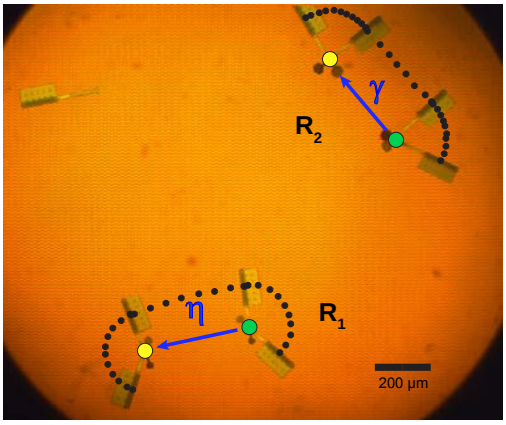


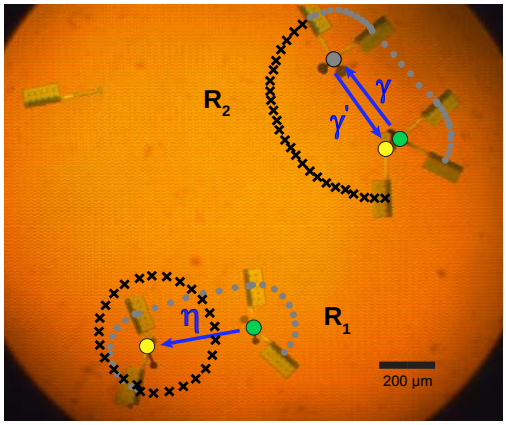
Fig. 15. Composite optical micrographs showing the translation of the center of rotation for robot R_2, $c_r(R_2)$, by two linearly independent vectors. **(a)** Both robots follow interpolated turning trajectories. The center of rotation $c_r(R_2)$ is displaced by vector γ while robot R_1 completes an orbit. **(b)** Both robots turn (turning trajectories are denoted by doted lines). During turning, the centers of rotation of both robots remain in place, allowing the angle between γ and γ_2 to be set to an arbitrary value. The center of rotation for robot R_2, $c_r(R_2)$, is then translated using interpolated turning by a second vector γ_2, as R_2 moves outside of the view of the microscope (intermediate configurations of robot R_2 are included for clarity). The yellow dot and arrow point to the approximate location of $c_r(R_2)$ after the completion of the experiment.

25 robots that are independently controlled using SESat, while robots between the groups are differentiated using TSC. The robots within this hybrid SESat-TSC(4) system can now be controlled using only 10 independent control voltage levels. Ofcourse, TSC can only maneuver the centers of rotation for the robots in $\mathbb{R}^2$. However, the SESat system can decouple the rotation of the individual robots. It would be interesting to analyze whether a SESat-TSC(n) system can be engineered to independently maneuver the robots in $\mathbb{R}^2 \times \mathbb{S}^1$

Because TSC cannot selectively snap-down or release the steering arms of the individual robots, all robots either all move in a straight line or all turn at the same time. Consequently, the orientations θ of the robots controlled by TSC remain coupled, i.e. cannot be independently controlled. Because $r > 0$, the MicroStressBots do not turn in place. To *approximate* desired configurations in $\mathbb{R}^2 \times \mathbb{S}^1$, the robots must orbit around their centers of rotation (c_r) until mutual



(a)



(b)

Fig. 16. Composite optical micrographs showing the implementation of TSC to independently translate the robot with a larger turning rate (R_1). **(a)** The center of rotation of robot R_2, $c_r(R_2)$, is translated using a combination of turning and straight-line trajectories (denoted by doted lines), and is translated by a vector γ. There resulting trajectory of robot R_1 does not form a an orbit, causing the center of rotation $c_r(R_1)$ to be translated by vector η. **(b)** The center of rotation of robot R_2, $c_r(R_2)$, is translated back to its starting location by vector γ' using interpolated turning, completing a c_r-orbit for R_2. Because the interpolated turning trajectory completes a c_r-orbit for robot R_1, the center of rotation $c_r(R_1)$ *maintains* its displacement by vector η from step **(a)**.

differences in their turning rates cause their orientations to all align close to the desired configurations. Hence, there is a trade-off between the lengths of the robot trajectories and the resulting proximity to the goal configurations, which is especially important when we consider perturbations due to the effects of control error. It would be interesting to fully investigate such trade-offs in future work.

Finally, it is also worth mentioning that interesting related theoretical ideas have been also presented in e.g. [2], and it would be interesting to investigate how these methods fit with the SESat-TSC framework.

Acknowledgments

This work was supported by a grant to B.R.D. from the Duke Institute of Brain Sciences, grant numbers GM-65982 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, and funded in part by the Center for Information Technology Research in the Interest of Society (CITRIS). The authors wish to thank D. Balkcom, C. Bailey-Kellogg, and Kris Pister for their advice and suggestions. The electron micrograph was taken at the Dartmouth Ripple Electron Microscopy Laboratory, with the help of C.P. Daghlian.

References

[1] I. V. Borzenets, I. Yoon, M. W. Prior, B. R. Donald, R. D. Mooney, and G. Finkelstein. Ultra-sharp metal and nanotube-based probes for applications in scanning microscopy and neural recording. *Journal of Applied Physics*, 111(7):074703, 2012.

[2] T. Bretl. Control of many agents using few instructions. In *the Proceedings of Robotics: Science and Systems 2007 conference*, Atlanta, GA, 2007.

[3] A. Cowen, B. Hardy, R. Mahadevan, and S. Wilcenski. *PolyMUMPs Design Handbook*. MEMSCAP, 2011.

[4] E. Diller, C. Pawashe, S. Floyd, and M. Sitti. Assembly and disassembly of magnetic mobile micro-robots towards deterministic 2-D reconfigurable micro-systems. *The International Journal of Robotics Research*, 30(14): 1667–1680, December 2011.

[5] B. R. Donald, C. G. Levey, C. McGray, I. Paprotny, and D. Rus. An untethered, electrostatic, globally-controllable MEMS micro-robot. *Journal of Microelectromechanical Systems*, 15(1):1–15, January 2006.

[6] B. R. Donald, C. G. Levey, and I. Paprotny. Planar microassembly by parallel actuation of MEMS microrobots. *Journal of Microelectromechanical Systems*, 17(4):789–808, August 2008.

[7] B. R. Donald, C. G. Levey, I. Paprotny, and D. Rus. Simultaneous control of multiple mems microrobots. In *the proceedings of the 9th International Workshop on Algorithmic Foundations of Robotics (WAFR)*, December 2008, and also in *Springer Tracts on Advanced Robotics*, **57**, G. S. Chirikjian, H. Choset, M. Morales, and T. Murphey (eds.), Springer-Verlag (Berlin), (2010).

[8] S. Floyd, C. Pawashe, and M. Sitti. An untethered magnetically actuated micro-robot capable of motion on arbitrary surfaces. In *the Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2008.

[9] D. R. Frutiger, B. E. Kratochvil, K. Vollmers, and B. J. Nelson. Magmites wireless resonant magnetic microrobots. In *the Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, May 2008.

[10] A. Ghosh and P. Fischer. Controlled propulsion of artificial magnetic nanostructured propellers. *Nano letters*, 9 (6):2243–2245, 2009.

[11] S. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.