

Stochastic Jumping Robots for Large-scale Environmental Sensing

Julian Hird^{1,2}, Andrew Conn^{1,2}, and Sabine Hauert^{1,2}

¹University of Bristol

²Bristol Robotics Laboratory

December 13, 2021

Abstract

Single-use jumping robots that are mass-producible and biodegradable could be quickly released for environmental sensing applications. Such robots would be pre-loaded to perform a set number of jumps, in random directions and with random distances, removing the need for onboard energy and computation. Stochastic jumpers build on embodied randomness and large-scale deployments to perform useful work. This paper introduces simulation results showing how to construct a large group of stochastic jumpers to perform environmental sensing, and the first demonstration of robot prototypes that can perform a set number of sequential jumps, have full-body sensing, and are well suited to be made biodegradable.

Corresponding author(s) Email: a.conn@bristol.ac.uk sabine.hauert@bristol.ac.uk

ToC Figure

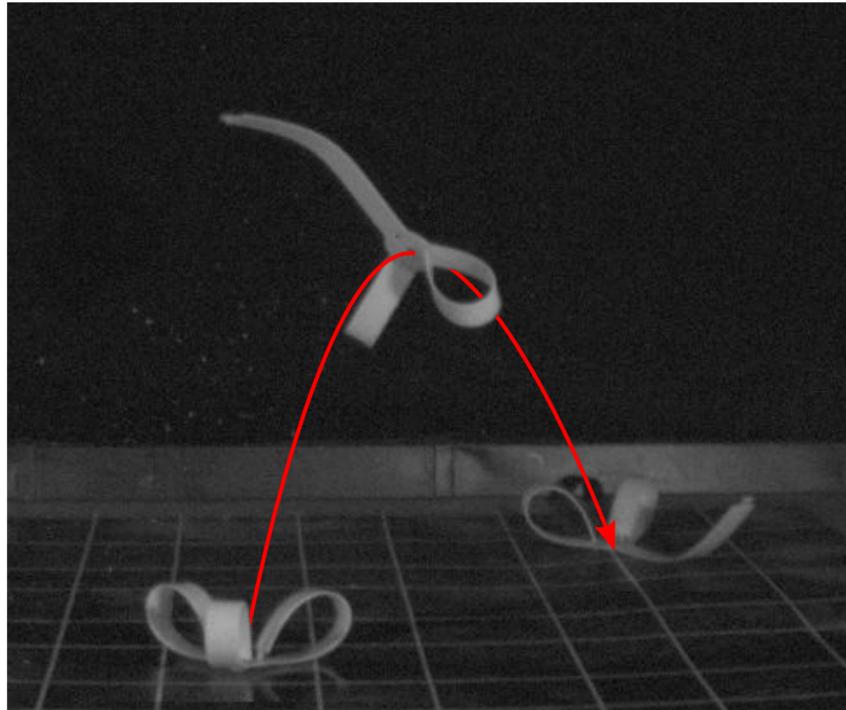


Figure 1: Stochastically jumping, mass producible robots could be used to perform large scale in situ sensing of outdoor environments. Herein, we report results from simulations of the system for an area coverage scenario, and the first robot prototypes. The physical robot developed can be rapidly assembled, and then is able to perform a series of jumps using environmentally triggered latches.

Introduction

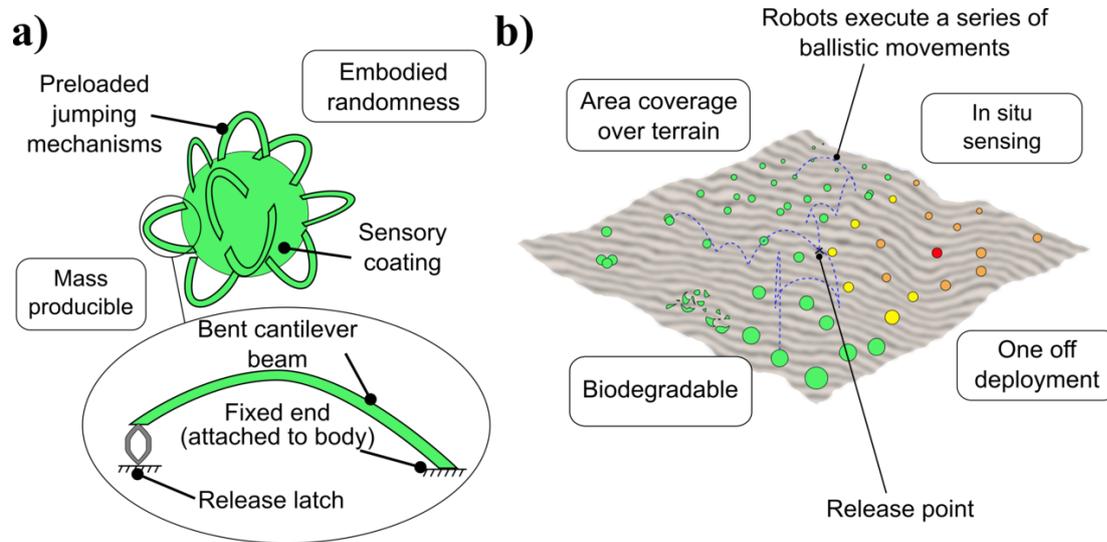


Figure 2: a) A Stochastic jumping robot covered in cantilever beam jumping mechanisms. b) The system spreading over an area and changing color in the presence of environmental stimuli. The proposed system releases robots in the center of the area of interest, from which they perform a series of jumps to spread themselves out.

Robots operating in large numbers have been proposed as a solution to perform large scale environmental monitoring (Schranz et al., 2020). Such robots can be scaled in number to fit area coverage needs, and redundancy in these systems favors robust deployment (Şahin, 2005). However, very few large multi-robot systems have been used in reality outside laboratories due to the challenges navigating uneven terrain, or producing sufficient robot numbers for meaningful area coverage in a cost-effective way. We propose to overcome both barriers by making single-use stochastic jumpers that are easy to mass produce, cheap, and effective. Our jumpers operate using embodied randomness, encoding stochastic jumping behavior in the design of the body of the jumper (see Figure 2). Jumps are initiated upon releasing pre-loaded elastic energy using mechanical components (latches) that are activated by an environmental stimulus. These latches control the sequence and timings of the jumps. Sensing capability is directly painted on the robot, and the lack of electronics makes it possible that in the future the robot can be made fully biodegradable. Large numbers of jumpers could provide in situ sensory information of an area for common tasks within agriculture or environmental remediation industries. As a first step towards real-world application, this work focuses on a deployment and sensing scenario over a target area.

Robots operating in large numbers are often individually simpler than those used in systems consisting of a few robots or a single robot. The simplicity of the robots in these systems is often compensated by their numbers and the design of strategies governing their deployment. Algorithms using artificial forces (Howard et al., 2002), minimal or noisy sensors (E. Ugur and Sahin, 2007; Ludwig and Gini, 2006; Ozdemir et al., 2019), and random walks (Beal, 2013; Siebold and Hereford, 2008; Dimidov et al., 2016) have all been proposed as methods of dispersing robots over an area. Large-scale indoor deployments up to 3000 ft² have been reported in (McLurkin and Smith, 2008) using the iSwarm system. Outdoor robot deployments have been demonstrated at large scales using drones (Hauert et al., 2014) or surface water vehicles (Zoss et al., 2018). However, large outdoor land based robotic deployments have not yet been realized.

Jumping robots have been explored in the past as a way to navigate challenging outdoor terrain, especially for small robots (Armour et al., 2007). Examples include miniature robots weighing under 10 g that exploit flea-inspired elastic release mechanisms driven by shape memory alloys (Noh et al., 2012) and DC motors (Kovač et al., 2008), although many of these platforms have not been designed for use in large numbers. This changes design priorities towards low individual robot cost, simplicity, and potential for mass fabrication. Previous jumping robots for environmental monitoring such as (Zhakypov et al., 2019; Dubowsky et al., 2008; Mintchev et al., 2018) all use electrical power and control components in their designs limiting their potential biodegradability (Rossiter et al., 2016; Hartmann et al., 2021). The dynamic simulations shown in (Dubowsky et al., 2008) demonstrated how jump-height and robot size had a strong influence on their robot's ability to traverse an obstructed tunnel without becoming entrapped. Meanwhile (Mintchev et al., 2018) performed physical trials of their robot, demonstrating the robot's ability to overcome obstacles 7 cm high and rapidly explore a flat 10 m by 10 m area by exploiting dynamic instabilities in its locomotion mechanism to perform a random walk. While (Mintchev et al., 2018) does explore the total area covered by the system based on their robot's trajectories, neither work examines the coverage capabilities of a large number of jumping robots operating simultaneously.

The principles of morphological computation represent an emerging view of intelligence in robotics, where mechanically pre-programmed control schemes and responses to environmental stimuli can be encoded in the robot's body (Laschi et al., 2016; Hauser et al., 2011). These principles are extended in this work to embody randomness within a robot's structure, so that the control of locomotion is encoded without pre-defined or deterministic path planning. Embodied intelligence can also include sensing modalities, such as using observations of body dynamics to sense environmental characteristics (Nakajima et al., 2015), and reactive pigmentation for thermally (Soter et al., 2018) or chemically (Ding et al., 2020) responsive robots.

Overall, the work presented here provides the first steps towards mass production and deployment of large numbers of stochastic robots, with embodied randomness, for outdoor applications. The potential for jumping robots to perform area coverage in simulation is demonstrated, and these simulations are used to inform the design of proof-of-concept single-use jumpers. These prototypes can be stored in a compact way, assembled quickly with minimal manipulation and are then capable of sensing their local environment via direct contact. The prototype designs also operate at a low price point (approximately US\$1.39 bulk cost of materials per robot). While the current design is not biodegradable, the limited number of materials used in its construction alongside the lack of toxic electronic elements make the design well-suited to be made fully biodegradable in the future.

Simulation-based Design

Simulations, programmed in Python, were carried out to evaluate the performance of the system in covering a 10 m by 10 m area of interest after being released at the center. In the future, we imagine a separate system might be able to produce and release the stochastic jumpers directly into the environment. Alternatively, the stochastic jumpers could be released at ground level or from the air by a human or robotic carrier.

Stochastic Robot

The ability of the system to cover the area of interest for environmental sensing is encoded in the design of the robot's body. Control of each robot is therefore determined not by a programmed microcontroller as would typically be the case, but by mechanically programming the robots to execute a specific number of jumps. By changing the body of the robot, these jumps could be triggered after a certain time has elapsed, or by environmental factors. These jumps have a noisy distribution of jump distances and directions due to the robot's interaction with the environment and open-loop operation.

In the simulation, the robots have n_j pre-loaded elastic jumping mechanisms which in total store a strain energy of E_{tot} joules. Each jumping mechanism is assumed identical as previous simulations showed no difference in the system's performance by having different energy release strategies when noise was present. Furthermore, having identical jumping mechanisms lends itself to mass production.

The robots are modeled as spheres ($r = 5$ cm) in continuous space. Their velocity is dictated by the finite state machine shown in Figure 3. Robots are introduced into the world after the previous robot has left the starting area, which is located at the origin. Robots start with a

random z rotation random z rotation (θ) and in the **Not Deployed** state. This can be imagined as a person or a robotic system manually placing the robots into the environment.

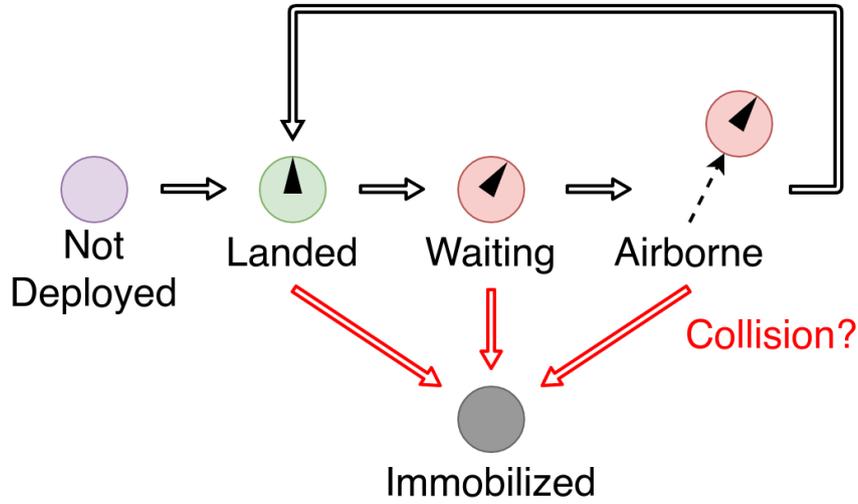


Figure 3: Finite state machine used to model each robot’s jumping behaviour in simulation.

Simulation Environment

The robots wait one second in the **Waiting** state, to model the delay caused by the release latch reacting to the environment. After this, robots jump and transition into the **Airborne** state. When the robots jump, ballistic physics with air resistance neglected is used to determine their jumping velocity (Figure 4).

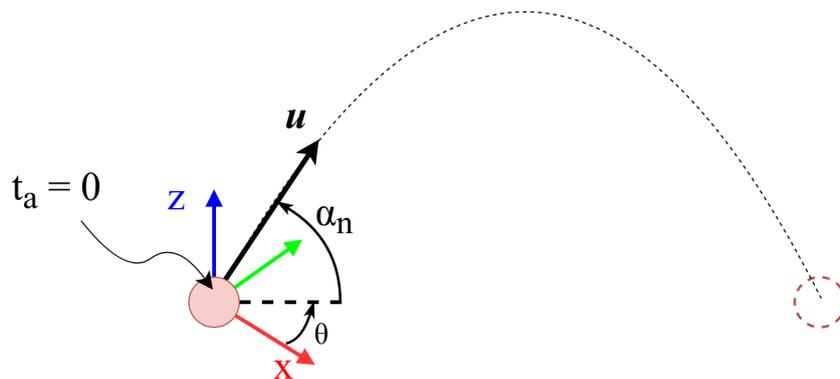


Figure 4: A robot jumping from the origin. Jumping is modeled using projectile motion. The direction of the jumping velocity \mathbf{u} is determined by a noisy take-off angle α_n and the robot’s z rotation θ , which can be considered to be the orientation of the next jumping mechanism. The z component of \mathbf{u} is a function of jumping time t_a which is zero at take-off.

This velocity is calculated from jumping energy and take-off angle after noise has been applied.

As each mechanism is identical, the ideal jumping energy used in a single jump E is an even fraction of E_{tot} .

$$E = \frac{E_{tot}}{n_j} \quad (1)$$

Noisy jumping energy E_n is calculated by multiplying E by a number sampled from the Gaussian distribution $\mathcal{N}(1.0, 1/9)$. Meanwhile, the noisy take-off angle α_n is obtained by sampling the distribution $\mathcal{N}(\pi/4, \pi^2/144)$. The robot's jumping velocity \mathbf{u} during the airborne state can then be calculated using the following.

$$\mathbf{u} = \begin{bmatrix} \cos(\theta)\cos(\alpha_n)\sqrt{2E_n/m} \\ \sin(\theta)\cos(\alpha_n)\sqrt{2E_n/m} \\ \sin(\alpha_n)\sqrt{2E_n/m - gt_a} \end{bmatrix} \quad (2)$$

Where t_a is the time the robot has been in the air. In these simulations $m = 50$ g and $g = 9.81$ m/s². When the robot lands, its velocity is set to zero and it enters the **Landed** state. When landing the robot's orientation is unpredictable so in this work the robot's θ after landing is chosen at random between $-\pi$ and π . If the number of jumps the robot has done does is less than n_j then the robot will return to the **Waiting** state.

During these movements, robots can collide if the distance between them is less than the sum of their radii and they are either both in the air or both on the ground. If collisions are being considered in the particular simulation then both robots involved in the collision are moved into the **Immobilized** state. In this state the robot will not move any further, but is considered to be laying on the ground, where it can still perform sensing. This can be considered the worst result of a collision. In reality it is likely that one or both of the robots involved in the collision would continue moving, if they had jumps left to perform.

Performance Evaluation

To calculate the area covered by the robots, their positions are recorded over time during the simulation. A 10 by 10 grid of squares is then used to divide the area of interest into coverable sections. If any of the robots' centers lie within a grid square then it is classified as being covered for sensing purposes, otherwise the square is classified as uncovered. Coverage is then given as the percentage of all grid squares that are covered (see Figure 5).

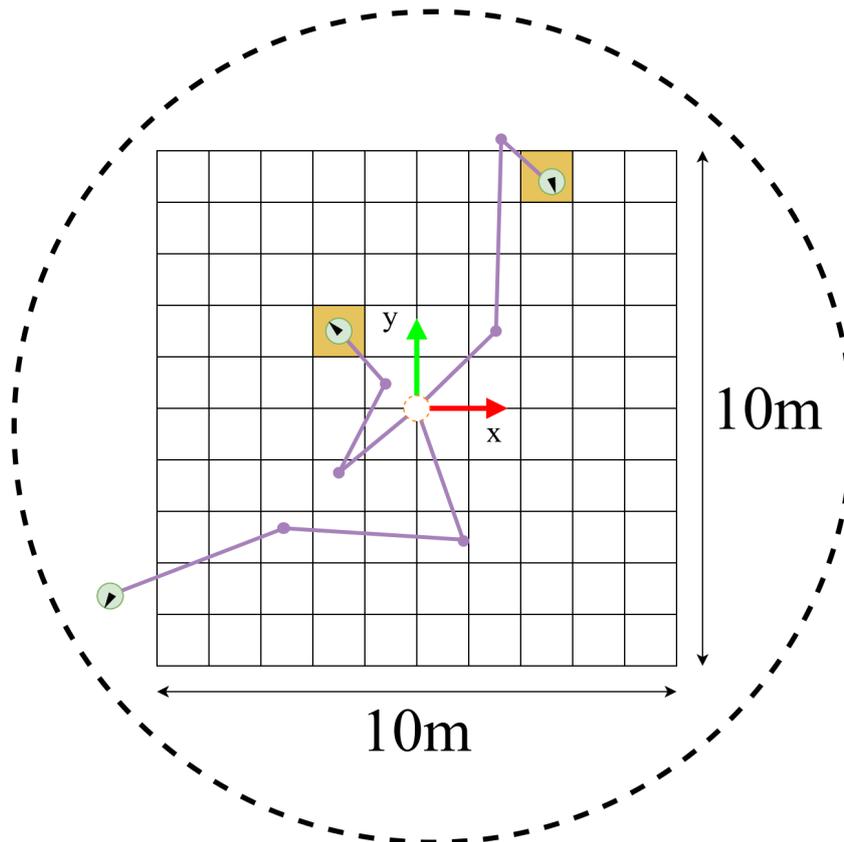


Figure 5: The deployment area modeled as an infinite plane with no obstacles (apart from other robots). A grid with a 1 m resolution is used to measure the coverage of the robots over a 10 m by 10 m area of interest. Here 2 out of 100 squares are covered (shown in yellow). The purple lines show the robots' trajectories through space from the deployment point at the origin. Along these lines the circular markers indicate where the robot has landed. The starting area at the origin is shown as the orange dotted circle. This circle has a radius of 5.25 cm.

Results

Here we present the insight from simulations in the design of area coverage strategies using stochastic jumpers, and the subsequent design of the first prototype robots. These prototypes fulfil the design requirements of area coverage, mass production, sensing, and potential for biodegradability.

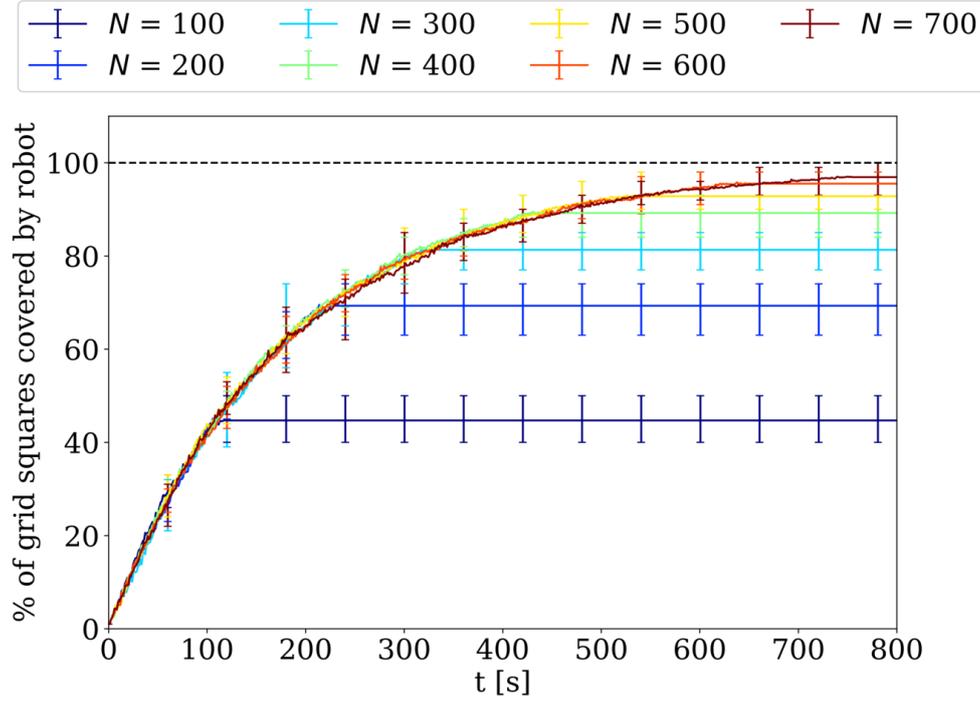


Figure 6: Coverage against time for different numbers of robots for $E_{tot} = 4.0$ J and $n_j = 7$. The solid lines show the mean coverage value over 10 trials, while the whiskers show minimum/maximum values over all trials.

Figure 6 shows how releasing up to N robots with $E_{tot} = 4.0$ J and $n_j = 7$ covers the 10 m by 10 m area of interest. As N is increased the system's total coverage also increases in a nonlinear fashion.

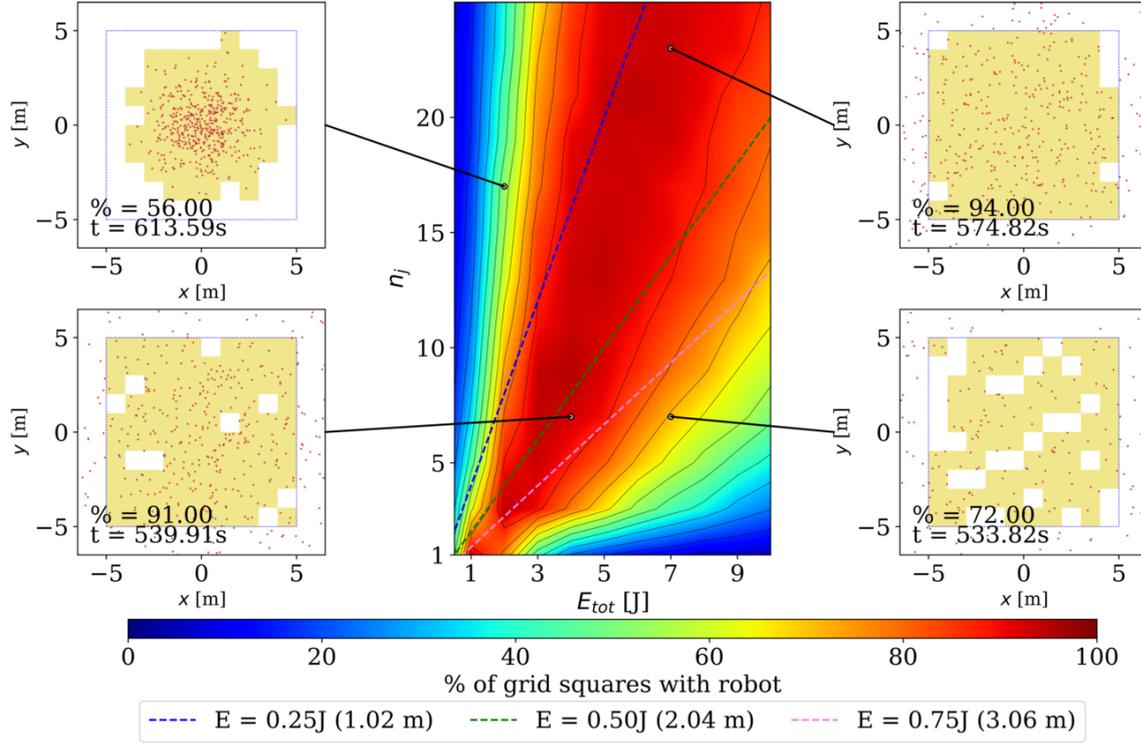


Figure 7: Heatmap of the final grid based area coverage for different deployments of robots that have different combinations of E_{tot} and n_j . The surrounding images show the final robot distribution for particular values of E_{tot} and n_j . In these images the area of interest is outlined in blue and the yellow squares show the covered sections of the grid where at least one robot lies. In these simulations collisions between robots have no effect.

To design the correct robot for the area of interest, Figure 7 shows how varying the values of n_j and E_{tot} affects the system’s final coverage performance for $N = 500$ robots. From Equation (1) it can be seen that straight lines that pass through the origin represent different ideal jump lengths. The large red area demonstrates the large design space that exists in choosing n_j and E_{tot} to achieve a high performance in both total coverage and coverage time. For example, a deployment of 500 robots which jump seven times ($n_j = 7$) and have an ideal jump length of 2.33 m ($E_{tot} = 4$ J) covers 91% the area of interest in 540 seconds (shown bottom left in Figure 7). Meanwhile a deployment of robots which jump 23 times ($n_j = 23$) but with a smaller ideal jump length of 1.24 m ($E_{tot} = 7$ J) is also able to cover over 90% of the area in a similar time of 574 seconds (shown top right in Figure 7). Across all of these simulations the average deployment time was 556 seconds. The size of the robot will scale with the number of jumping mechanisms n_j and the size of these mechanisms. Since larger mechanisms are able store more strain energy, the size of the robot is also proportional to $\frac{E_{tot}}{n_j}$. It is therefore

noteworthy that low values ($n_j < 5$ and $E_{tot} < 4.0$ J) are able to cover the area well, leading to the possibility of using very small robots for the area coverage task.

Introducing collisions between robots, as shown in Figure 8, demonstrates that inter-robot interference, while damaging at high values of E_{tot} and n_j , still leaves a large design space (shown in red) where the robot is able to cover more than 75% of the area. The main cause of this deterioration is due to robots landing on top of each other, causing clusters of immobilized robots to form. If these clusters also occur in close proximity to the deployment zone robots are prevented from reaching the outer regions of the area, lowering total coverage. For short jump lengths, these clusters are more likely to form as robots are less able to jump over one another. Robots with lower values of n_j perform better when considering collisions. We hypothesize this is due to the lower number of jumps leading to less situations (mainly landings) where the robot can enter into a collision. A mild improvement to the system’s coverage performance was found by waiting 60 seconds between each robot being deployed. However, this led to a longer average deployment time of 8.49 hours.

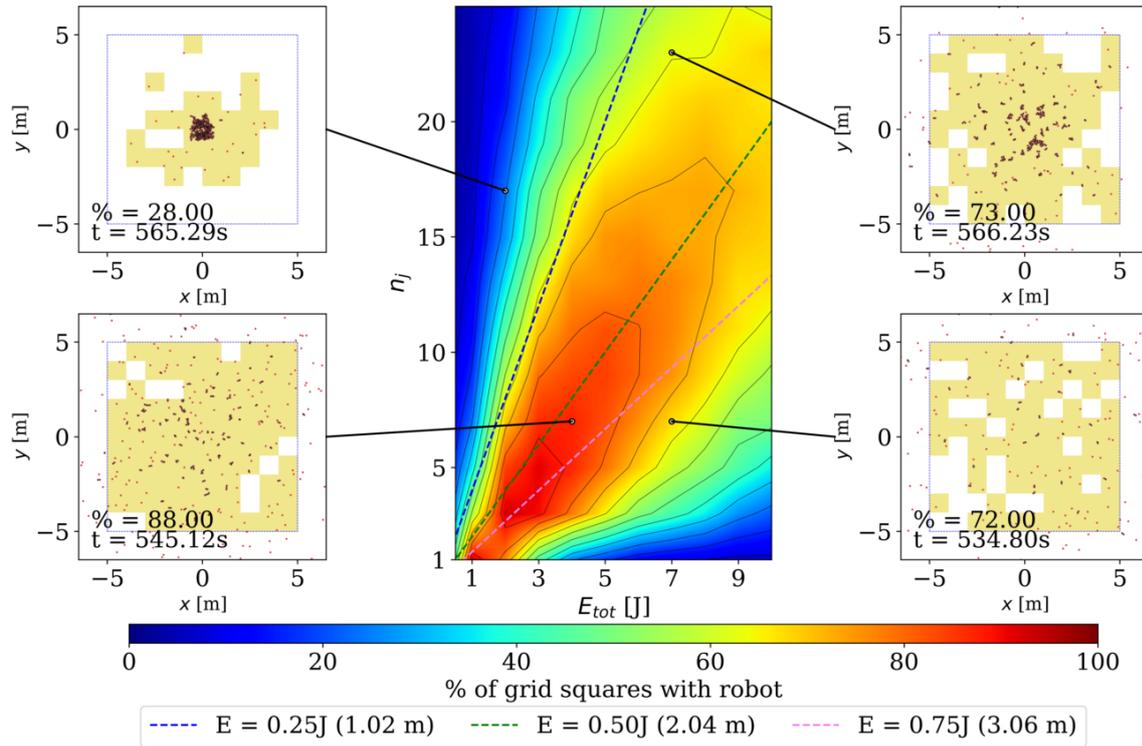


Figure 8: Heatmap of the final grid based area coverage for different deployments of robots that have different E_{tot} and n_j . In these simulations collisions between robots render both robots immobilized.

Insight from these simulations shows that there exists a large number of combinations of jump

numbers and jump lengths that allow 500 robots to cover more than 80% of the 10 m by 10 m area, with a resolution of 1 m. The amount of time taken varies between 556 seconds and 8.49 hours depending on the time between robot releases. This time would also depend on the period between robot jumps which was fixed at one second in these simulations.

Stochastic Robot Design

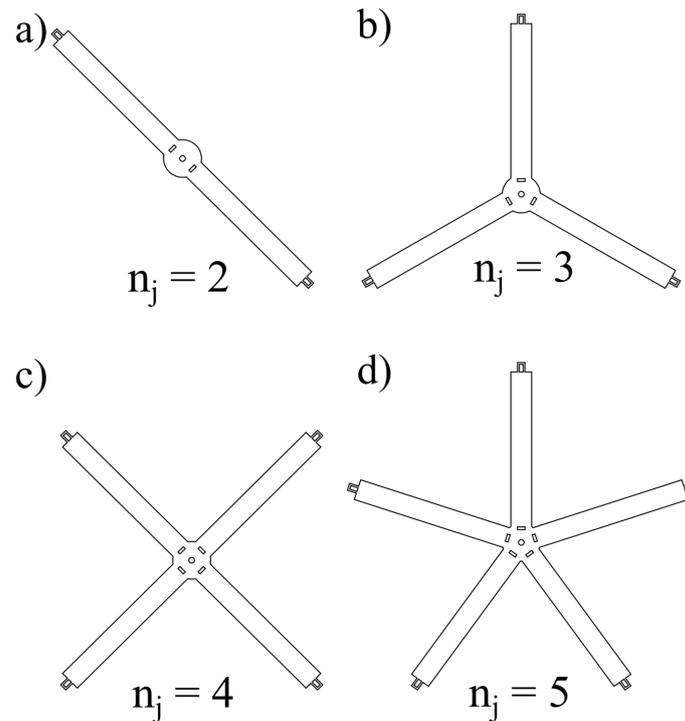


Figure 9: Top down view of the different types of prototype produced. Each prototype (a-d) is capable of a different number of jumps (n_j) depending on how many cantilever beams are featured in the design. Each prototype was laser cut from a sheet of Acetal Co-polymer.

As a first step towards making large deployments of stochastic jumpers a reality, we present a series of prototypes capable of between 2 and 5 jumps. Each prototype design consists of cantilever beams arranged around a central circular area (Figure 9). The number of cantilever beams determines the number of jumps the robot will perform. When these beams are bent, they are capable of storing the required energy for a jump. This removes the need to use separate spring components, simplifying robot assembly (Figure 10). Before being placed in the environment the robots are pre-loaded with strain energy. This is achieved by inserting the tips of the beams into slots inside the central area (see Figure 10 and Figure ??), which are then secured in place using 3D printed water soluble latches (PolyVinyl Alcohol, PVA). The water soluble latches facilitate sequential jumps to be triggered by moisture in the environment (e.g.

rain). The simplicity of the design opens up the possibility that it could be rapidly assembled by a robotic or human production line. Manual assembly of the current design for example takes less than a minute (as shown in Figure ??). The use of laser-cut scaffolds for the robot makes it easy to store the material, allowing for the production of large numbers of robots in a compact form. Currently the beams are constructed from Acetal Co-polymer which is not a biodegradable plastic. However, this sheet material could be replaced with a different compostable polymer (Naser et al., 2021) and enable the robot to be fully degradable. CAD designs can be found at (Bitbucket, 2021).

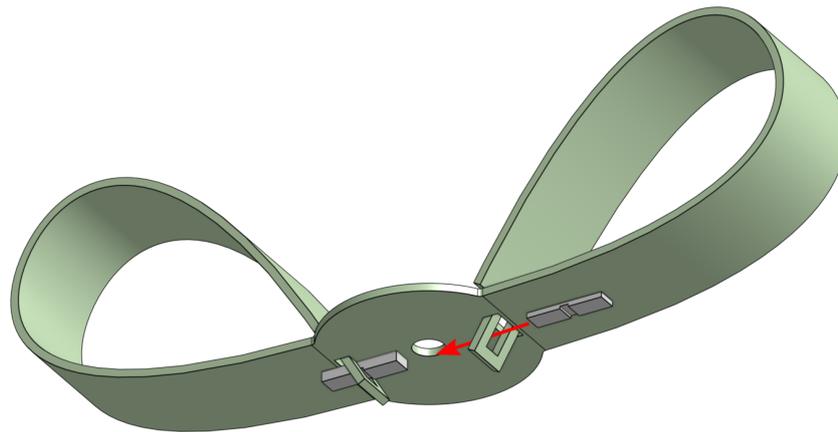


Figure 10: The robot is primed for jumping by bending the beams and securing them with PVA latches.

Rich media available at <https://www.youtube.com/watch?v=2RLQsvjq33M>

Jumping Mechanism

The cantilever beams allow the robot to jump by releasing their stored strain energy and colliding with the ground. This converts some of the stored energy into kinetic energy of the robot body (see Figure 11).

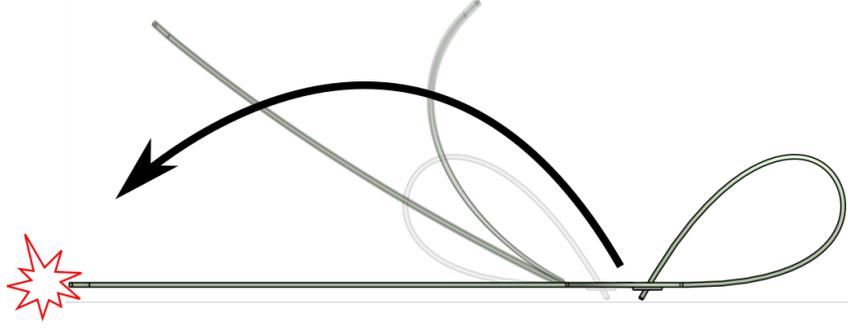


Figure 11: The jumping mechanism propels the robot into the air through the latch dissolving when in contact with water, releasing the compressed beam. As the beam unfurls it collides with the ground, propelling the robot into the air.

As shown by the earlier simulations, designing the robot with a certain ideal jump length is important to ensure good system performance. The jumping energy in the robot prototypes is controlled through the dimensions of the beam. The strain energy (W) in an axially loaded beam can be approximated using Equation (3), derived from work in (Scirè Mammano et al., 2017).

$$W = -\frac{E_{ac} b t^3 \pi^2}{3l} - \frac{4 E_{ac} b t^3 \pi^2}{3d - 12l} \quad (3)$$

Where E_{ac} is the Young's modulus of the beam material, d is the tip displacement, l is the beam length, b is the beam width and t is the beam thickness. The beam material was chosen to be Acetal Co-polymer ($E_{ac} = 2800$ MPa) due its low density and high yield strength. In the prototype the distance between the beam ends when they are primed for jumping is essentially zero, making the displacement in the direction of loading equal to the beam length ($d = l$) leading to Equation (3) becoming:

$$W = \frac{E_{ac} b t^3 \pi^2}{9l} \quad (4)$$

The thickness t and length l of the beam were chosen based on the available material sizes, the dimensions of the laser cutting bed, and to minimize the stress in the material to avoid plastic deformation. The final values used were $t = 1.5$ mm and $l = 165$ mm. This leaves the beam width b as a free parameter which determines the energy stored in the beam. This was chosen to be $b = 22$ mm, resulting in an energy per jump of 1.38 J according to Equation (4). This single mechanism (shown in Figure 12) is repeated around a central circular area to give the desired total number of jumps. Figure 9 shows the resulting designs for total jump numbers of

2 to 5 jumps.

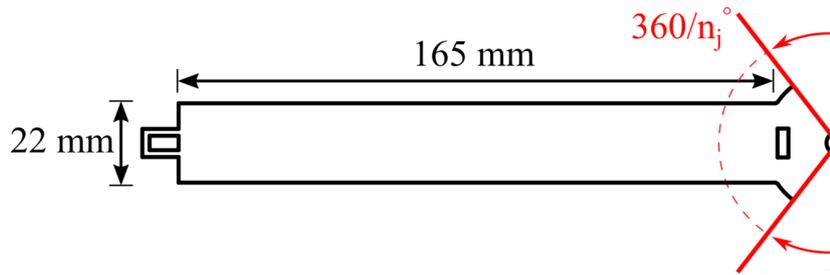


Figure 12: Detailed top view of the cantilever beam that forms part of the jumping mechanism used in the prototypes. The solid red lines indicate where the design would be repeated, with the angle depending on the number of beams in the prototype.

Environmentally Triggered Latches

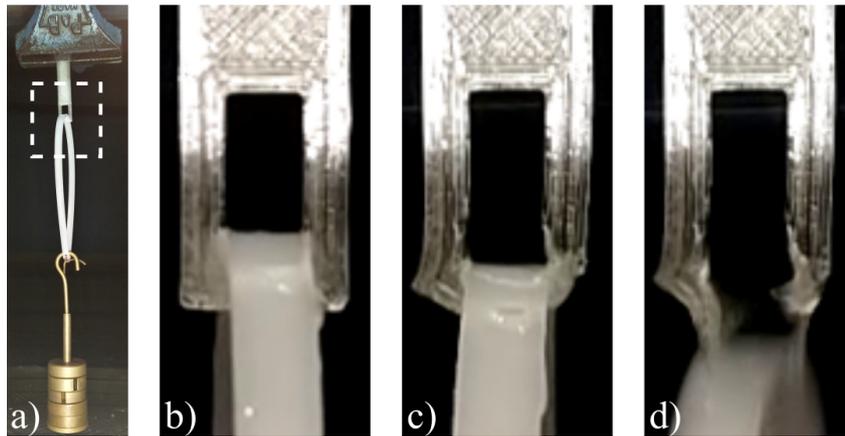


Figure 13: The working principle behind the PVA latches a) A 500 g load is placed on a 1 mm thick PVA part. The applied load represents the applied force from the bent beam. In b-d) 50 ml of water was applied to the part every 20 minutes leading to the PVA dissolving and the part failing after 85 minutes.

When the robot comes into contact with water (e.g. from rain) the latches dissolve (see Figure 13), eventually releasing the loaded beam and triggering the jump. The different thicknesses of latches cause the beams to release sequentially, allowing for consecutive jumps. The latches were 3D printed using a WANHAO i3 Mini and PVA filament.

To characterize the time it would take for each latch to yield under load a series of latch specimens of varying thickness were put in an experimental rig as shown in Figure 14. The rig mimicked the loading conditions on the latch when loaded by the bent beam by using a replica

of the beam end and slot on the robot's body. The applied load (6.38 N) was chosen based on measurements made using a digital force meter (Fk-50 Sauter) and a beam from one of the robot prototypes. The latches were then submerged underwater and the time taken until the latch failed was measured.

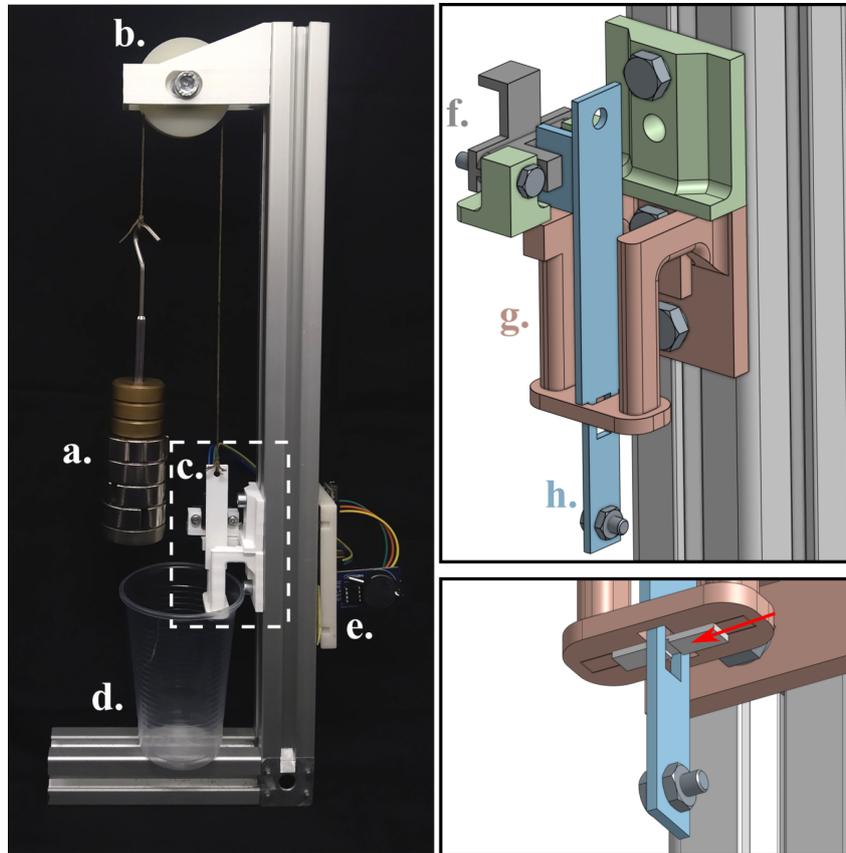


Figure 14: The experimental rig used to characterize the yielding times of the latches (left) with a close up view of the latch holding area (top right). Slotted masses (a) are used to load latches via a pulley system (b). Latches are placed in the latch holding area (c) and submerged in a water container (d). The time till yield is measured using timing circuitry (e) and a light gate (f). A dummy robot body (g) and beam end (h) replicate the loading conditions the latch experiences when used in the prototype. Latches are loaded into the rig in a similar manner to how they are loaded on the robot (bottom right).

The results of these experiments (shown in Figure 15) demonstrate that varying the thickness of the latches can be used to precisely control their yield time, hence ensuring jumps are released sequentially. In reality, the latches experience additional loading forces beyond just those from the bent beam including forces involved in robot assembly and during landing. Hence, to ensure sequential release, the thicknesses of the latches in the prototypes were increased in 1 mm increments with the thinnest being 1 mm thick.

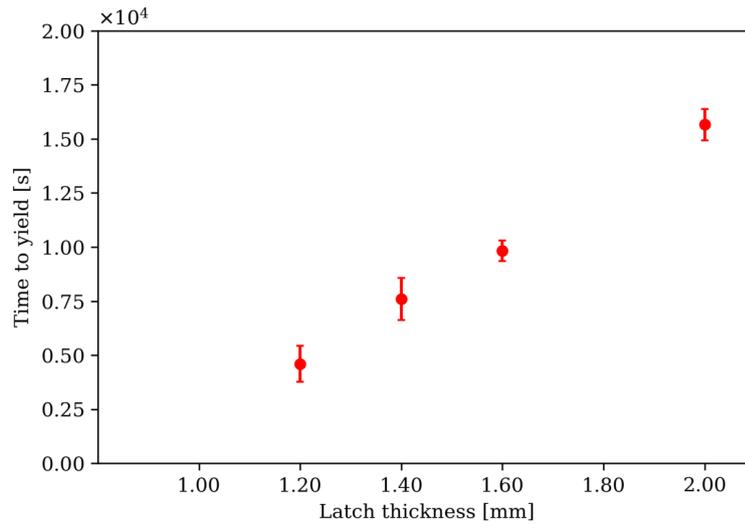


Figure 15: Time it took for different latch specimens to fail when placed in the latch testing rig. The experiment was repeated 6 times for each latch and the mean time to yield with one standard deviation is shown.

Sensory Coating

The sensory coating of the robot allows it to communicate the presence of stimuli in the environment through the use of color change. To demonstrate this concept the prototype robot was coated in thermochromic paint. Figure 16 demonstrates how the robot changes color in the presence of heat; this color change approximately happens at 31°C.

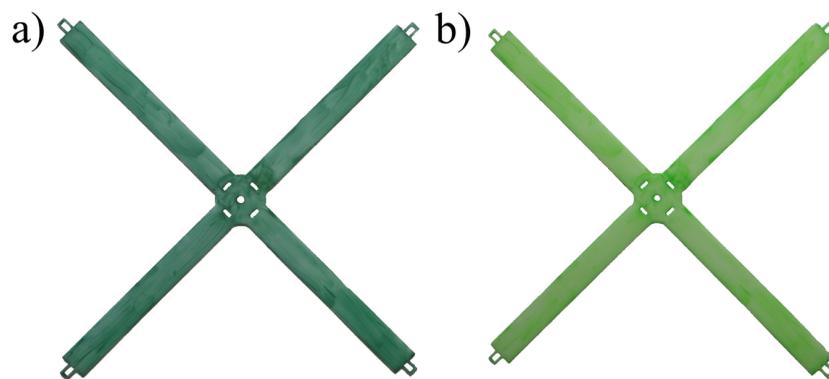


Figure 16: The coating of the robot prototype reacts to the temperature of its surroundings. The robot at room temperature (a) and after heating (b).

These readings could then be recorded with an aerial photograph. Figure 17 demonstrates how an overhead image of deployed jumpers over an area can be used to locate a heat source by

observing the robots' colors. In the future, larger areas could be imaged by combining many photos together that have been captured using a drone (Remondino et al., 2011). The sensory coating could also offer sensory information to other agents on the ground. Additionally, various stimuli could be detected using colorimetric (Burgess et al., 2013) or paper based sensors (Nery and Kubota, 2013). These could be laminated on top of the sheet material used to construct the robot.

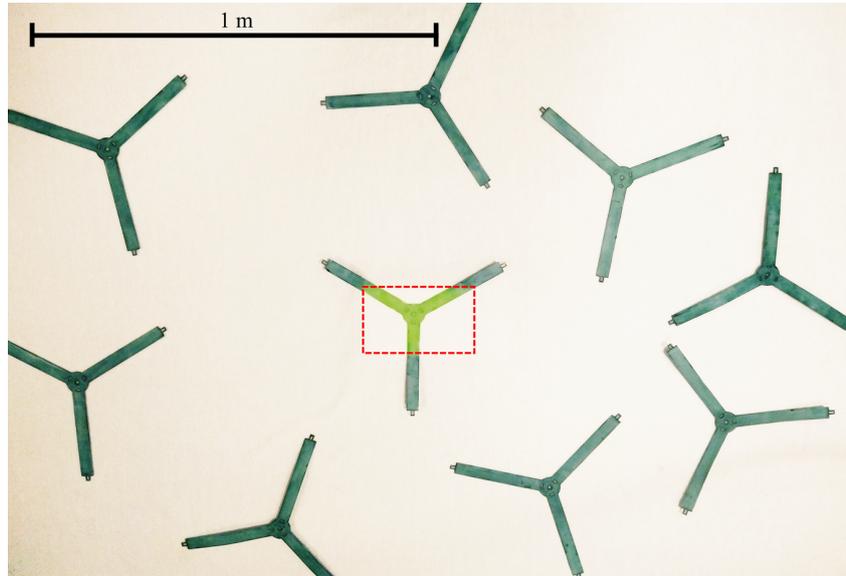


Figure 17: A collection of robots manually distributed over an indoor area containing a heated metal plate (outlined in red) which was placed under the white backdrop. The heat emitted by the plate changes the color of the nearby robot from dark green to light green.

Jumping Performance

The jumping performance of four different prototypes, each capable of a different number of jumps, was evaluated by carrying out a series of jumping trials within a flat experimental arena. Two cameras were used to track the robot's movement and also measure jumping characteristics such as jump height and distance (see Figure 18). The side camera (FLIR Blackfly S BFS-U3-16S2M) had a framerate of 200 frames per second to capture the robot's motion during a jump. Meanwhile a separate top down camera (Mermaid MM-USB8MP02G-MFV) was used to accurately measure the robots position before and after jumps. Image capture and processing was done using Python with the Spinview SDK (FLIR) and OpenCV library (ope).

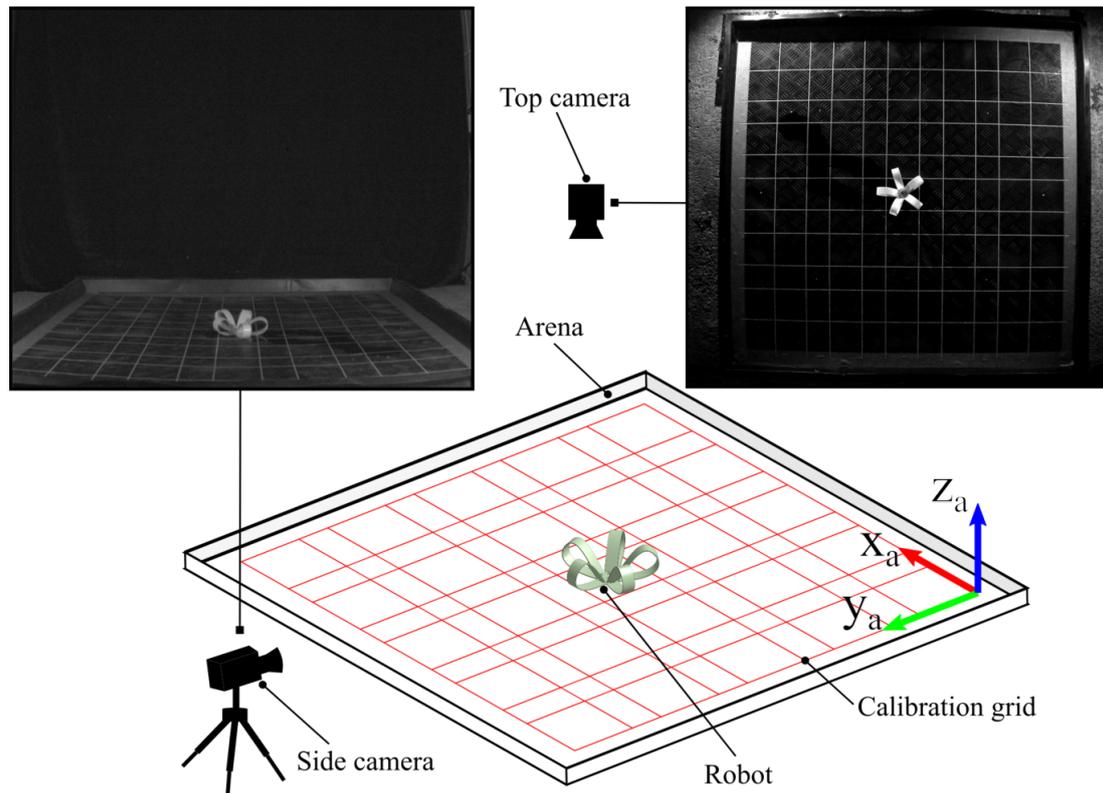


Figure 18: The experimental setup used to measure the prototype’s performance inside an arena. The setup consists of two cameras that are used to track the robots movement and a computer used to store the captured images. A grid taped to the arena floor is used to calibrate the system so measurements can be made by converting between pixels and millimeters. The co-ordinate system (x_a, y_a, z_a) used for measurements is also shown.

At the start of each experiment the robot prototype under test was assembled and placed in the center of the arena. To avoid the effect of any material fatigue freshly manufactured robots were used during each trial and three trials were performed for each of the designs shown in Figure 9. Once the robot was in place, the recording software was activated and 50 ml of water was then added to the robot. Water was added to the robot periodically throughout the experiment to mimic how water would reach the robot outdoors (eg. rain). During the experiment, the latches within the robot would yield once exposed to the water causing the robot to perform a sequence of jumps. Top down images were captured every minute to track the robot’s movement. Meanwhile, side images were continually captured into a circular buffer that had a capacity to store 2.5 seconds of footage. When motion was detected in the side view image an additional 2 seconds of images were captured and then the entire buffer would be written to disk. This method was used due to the long timescale of the experiments and the limited speeds at which images could be written to a hard drive. Typical footage of the robot jumping is shown in Figure ??.

Rich media available at <https://www.youtube.com/watch?v=FTCM2WkV7x4>

Experiment Calibration and Measurement

Once the robot had performed all its jumps the images from both cameras were processed and labeled to obtain the measurements of interest. Processing images from both cameras consisted of discarding irrelevant images that did not show robot motion and then removing distortion from the remaining images using each camera's distortion coefficients. These were established by capturing a series of images of a chessboard before the experiment. The top-down images underwent an additional processing step where they were re-projected onto the co-ordinate space shown in Figure 19 in order to align the grid axes and image axes.

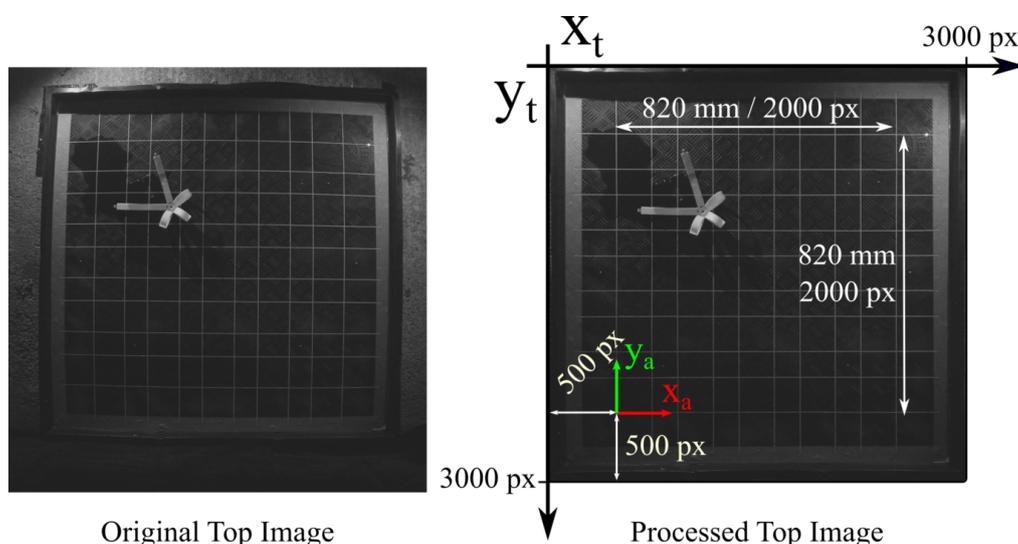


Figure 19: The images from the top camera had distortions removed and then were re-projected onto a square image of 3000 x 3000 px to take measurements of the robots position within the arena. Pixel positions are then converted into the arena co-ordinate system.

Pixel positions in the resulting images (x_t, y_t) were then converted to real world positions (x_a, y_a) using the following equations, which are derived from the known size of the processed image and the grid on the arena floor.

$$x_a = \frac{x_t - 500 \text{ px}}{2000 \text{ px}} 820 \text{ mm} \quad (5)$$

$$y_a = \left(1 - \frac{y_t - 500 \text{ px}}{2000 \text{ px}}\right) 820 \text{ mm} \quad (6)$$

The labeling process involved selecting the pixel position of the center of the robot within

top-down or side images. These positions were then used to calculate jump height and distance measurements.

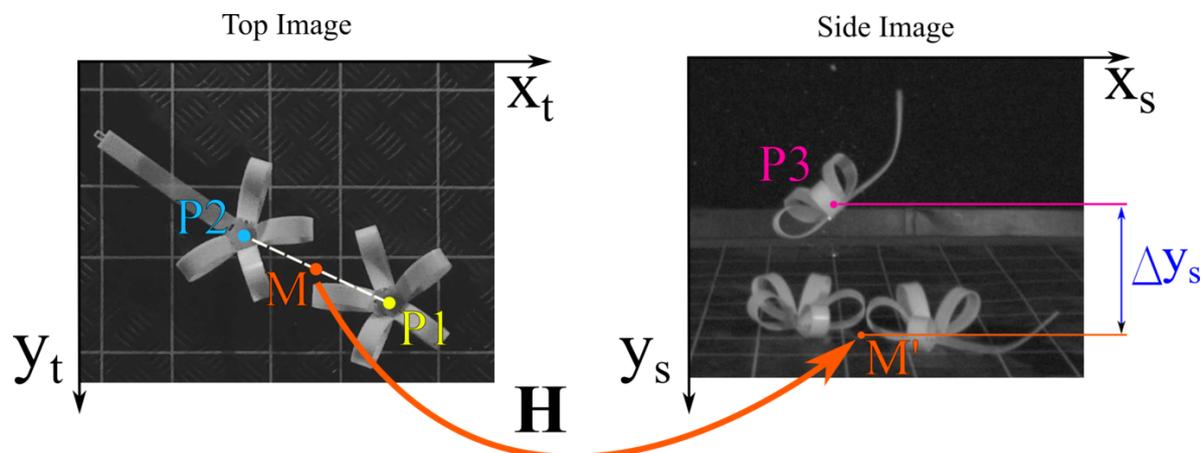


Figure 20: Jump height was measured by establishing the midpoint (M) between the robot's start position ($P1$) and end position ($P2$) from the top down images. This midpoint was then re-projected into the side image (M') using the homography H . The robot's peak position in the side image ($P3$) was then manually labeled and the difference in y pixels between the two points was converted to mm. M' is not directly below $P3$ in the side image as the robot bounced upon landing before coming to rest at $P2$.

For each jump, the robot's central position in the top down image was manually labeled both before and after the jump ($P1$ and $P2$ in Figure 20). These pixel positions were then converted into the arena co-ordinate system to find the robot's planar trajectory and jump length.

Measuring jump height required finding the distance between the position of the robot at the peak of its jump and the ground underneath the robot at this time. These positions were obtained using both the top down and side view images, as shown in Figure 20. The peak position of the robot was found in the side image and labeled manually in the frame where the robot was at the peak of its jump. The ground position in the side view was difficult to determine accurately by eye. Hence the ground position was calculated by assuming the robot followed a ballistic trajectory and would be at its peak height when it had travelled halfway from its starting position to its ending position, which had previously been labeled in the top down images. In reality, the robot often bounced a small distance away from its initial landing spot. However, this distance was found to be negligible compared the distance of the jump. To convert the mid-point position in the top-down image into a ground position in the side image a homography between the two images was used. It was assumed the arena grid was planar. Hence the homography H between the top down and side images could be found by selecting corresponding points in the arena grid within both images.

Once the ground and peak positions had been found the pixel distance in the y axis (Δy_s) needed to then converted to height (z_a) using a conversion factor (m).

$$z_a = m\Delta y_s$$

Since the robot changes its distance from the camera during the experiment the conversion factor m depends on the position of the robot in the arena. The side camera was carefully aligned using a spirit level so that the image plane was parallel to the $x_a z_a$ plane. This allowed the conversion from pixels to millimeters to be represented as a linear function of the robot's y position in the top image (y_t).

$$m = c_1 y_t + c_2$$

To find the two calibration constants c_1 and c_2 , a vertical jig marked with two targets was moved around the grid while images from both cameras were captured. The targets on the jig were manually labeled and the physical distance between them was known. This gave m for various positions in the grid. Then c_1 and c_2 could be found by fitting a linear regression model to the data with a high accuracy ($R^2 = 0.99$), as shown in Figure 21.

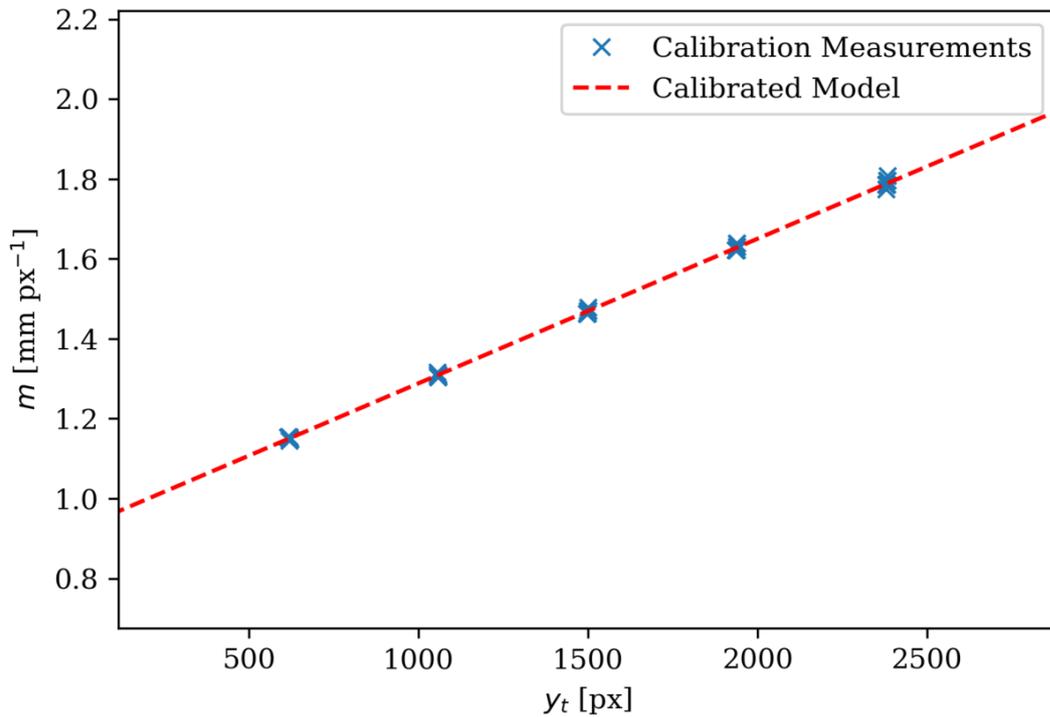


Figure 21: The relationship between the number of y pixels in the side image per mm (m) and y pixel position of the object in the top image (y_t).

The error that this calibration produces against the known height of the top target (416.5 mm) and bottom target (116.5 mm) as the tool was moved around the grid is shown in Figure 22. The mean of the error for both targets is close to zero ($\mu = 0.610$ mm). Meanwhile, the measurements made by the system can be said to be within ± 4 mm based on three standard deviations ($3 \times 1.3 \approx 4$ mm).

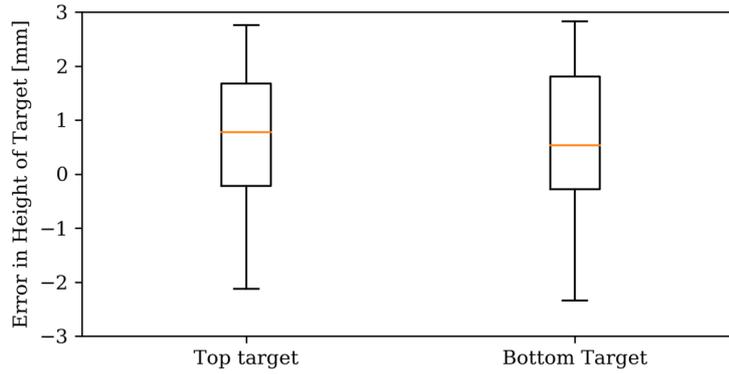


Figure 22: Error in height measurements of the targets on the calibration tool once the system had been calibrated. Whiskers indicate the upper and lower quartiles. The mean of the error for both targets was 0.61 mm while the standard deviation was 1.31 mm for the top target and 1.32 mm for the bottom target.

Experiment Results

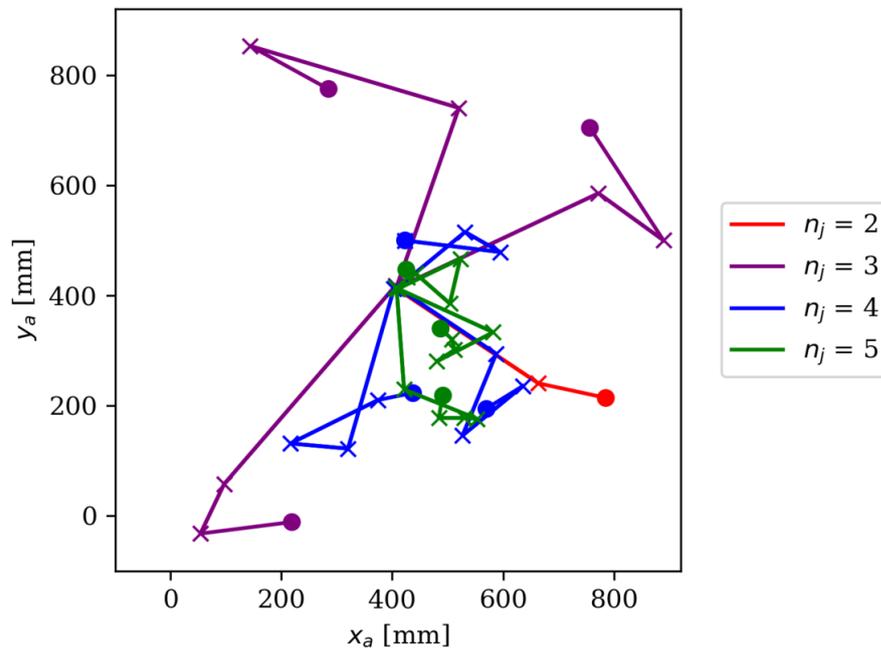


Figure 23: The planar trajectories of the robot prototypes. Crosses indicate where the robot landed after completing jumps. Filled circles indicate the final positions of the robots. Two of the $n_j = 2$ robots jumped out of the arena on their first jump and so are not shown.

The results from the jumping trials are detailed below. Firstly, Figure 23 shows the trajectories of the robots throughout the arena across all trials. Each robot executed its jumping sequence successfully and was able to move away from the starting area. All the robots were able to jump regardless of the orientation they landed in. This included the $n_j = 2$ design which was prone to falling on its side. In one particular trial this design landed on its side after its first jump and then was able to move a further 122 mm during its second jump. This design also left the arena in two trials and had to be placed back into the center of the arena so its second jump could be measured. The first jump of these trials could not be measured accurately, however they do show that this design is capable of jumping distances larger than half the size of the arena (> 560 mm).

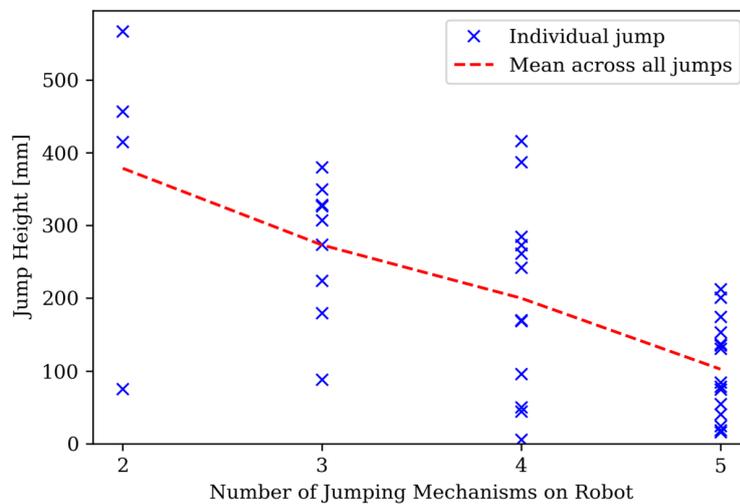


Figure 24: Jump height of the robots against the number of jumps it performed. The two jumps for the $n_j = 2$ design have been omitted as the robot landed outside the arena so the jump height could not be measured.

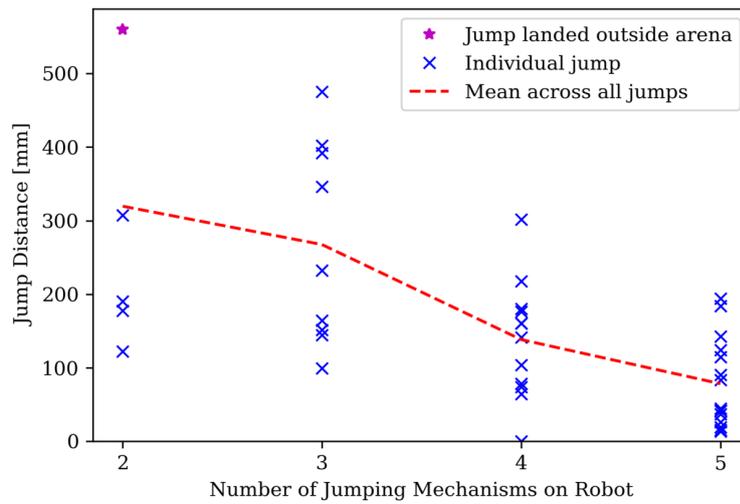


Figure 25: Jump distance of the robot against the number of jumps it performed. Two jumps for the $n_j = 2$ design left the arena could not be measured accurately. These jump distances have been estimated as the minimum distance they must have travelled (560 mm).

The jump distances and jump heights achieved by the prototypes are shown in Figure 24 and Figure 25 respectively, with the greatest height (567 mm) and distance (> 560 mm) achieved by the $n_j = 2$ design. The largest measured distance (475 mm) was achieved by the $n_j = 3$ design. Both jumping distance and height decreased as the number of jumps the design could perform increased. This can be explained by the fact that the energy per jumping mechanism is constant. However, the mass of the robot increases by around 8.5 g with each jumping mechanism added.

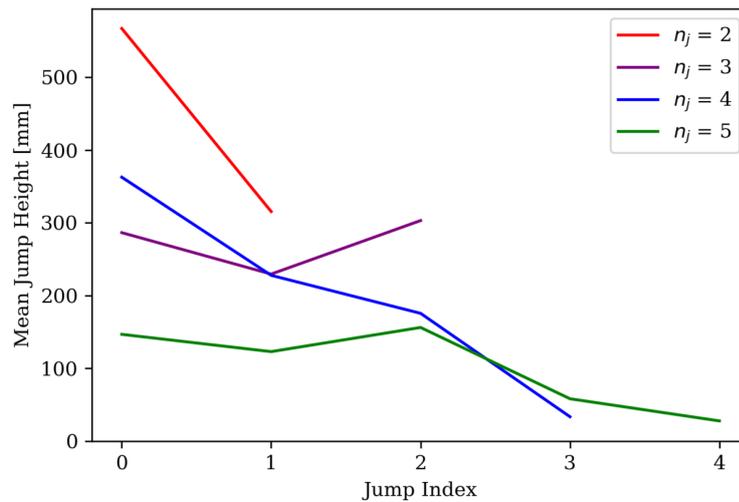


Figure 26: Mean jump height across all trials against the index of the jump in the jumping sequence (with zero being the first jump).

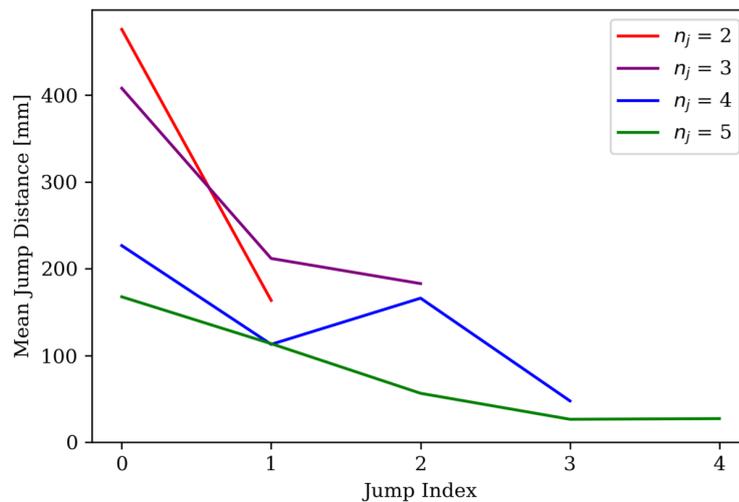


Figure 27: Mean jump distance across all trials against the index of the jump in the jumping sequence (with zero being the first jump).

It also appears that when comparing earlier jumps to later jumps, the earlier jumps achieve greater heights (see Figure 26) and travelled further on average (see Figure 27). This could be due to a number of factors. Firstly, the shape of the robot's body changes as beams are unfurled, altering its mass distribution. Additionally, when observing the footage of the later jumps there is a noticeable increase in oscillations in the robot body. This could be due to the fact that the stiffness of the robot structure is lower in these later jumps and so energy from the

jump goes into deforming the robot body and not into the jumping motion. Furthermore, during later jumps the robot has a larger surface area in contact with the ground leading to increased adhesive forces between the robot body and water on the arena floor.

The time taken for the each latch to release the jump is shown in Figure 28. The first latch (with a thickness of 1 mm) released the beam at around 0.938 hours on average, while the thickest latch (5 mm) had an average release time of 5.87 hours.

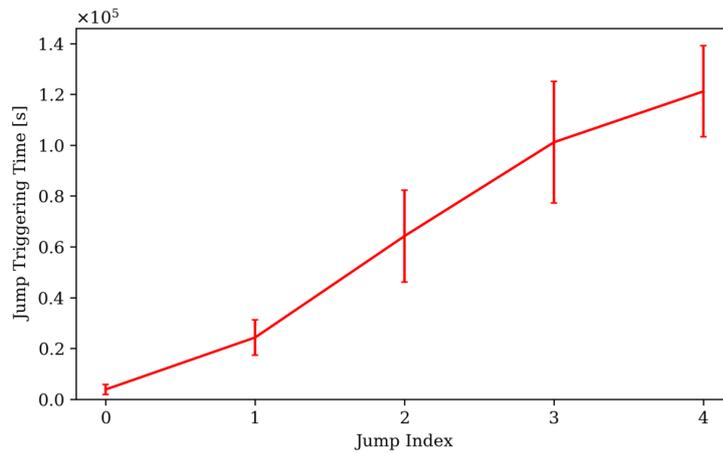


Figure 28: The mean time it took for each jump to trigger across all designs and trials. Jump index refers to the index of the jump in the jumping sequence, with zero being the first jump of each robot. Error bars show one standard deviation.

Outdoor Demonstration

To demonstrate the potential of the system to operate outdoors, we performed a jumping trial of one of the prototypes in rugged terrain as shown in Figure ???. The prototype is able to execute its jump sequence successfully and traverse various obstacles. This demonstration is a first step towards deploying stochastic jumpers outdoors.

Rich media available at <https://youtu.be/VZJJFzGvZFK>

Discussion

We have taken the concept of stochastic jumpers from simulation through to a first prototype design. The prototype achieves many features of the simulated jumpers, including jumping motion, the sequential jump release of a finite number of jumps and random re-orientation. However, they do differ from the simulated robots in a number of ways. Firstly, the simulated

robot’s jumping distance was independent of the number of jumps and the jump index. However, in the prototype designs this is not the case. Furthermore, the jumping distances achieved by the prototypes were smaller than the simulated robots that were able to cover a 10 m by 10 m area. The length of time it takes the robot prototype to complete each jump is also longer than in the simulation. However, this could be acceptable in scenarios where the speed of the deployment is not important. Deployment times could also be reduced by releasing the robots in parallel. Releasing robots from multiple points in parallel would also allow the system to cover larger areas as shown in Figure 29. This preliminary work shows how over 80% of a 100 m by 100 m area can be covered using as few as 25 deployment points. Future work will examine how best to choose these deployment points.

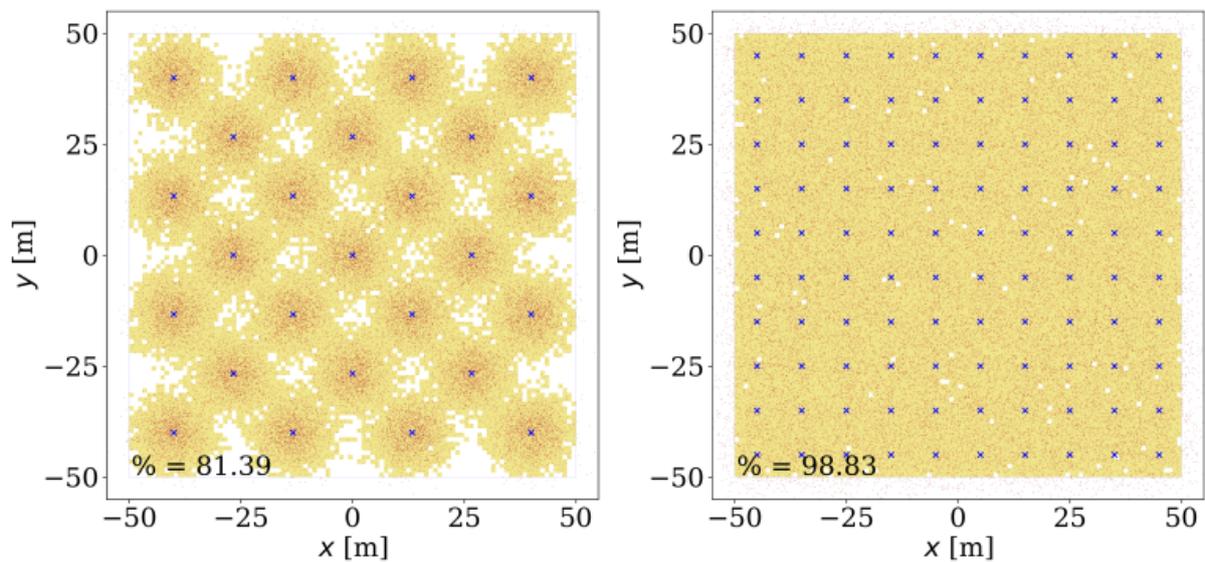


Figure 29: Snapshots of preliminary simulations that use multiple release points to increase the area covered by the system. Jumpers that are able to cover the 10 m by 10 m area ($n_j = 7$ and $E_{tot} = 4$) are introduced in equal numbers from multiple deployment points (shown by blue crosses). The 100 m by 100 m area is then covered by 50,000 robots using 25 deployment points (left) and 100 deployment points (right). As with previous simulations, the area is divided up into 1 m by 1 m squares, with covered squares shown in yellow alongside the percentage of total squares covered. These simulations do not consider collisions.

To improve on the limitations of the current prototypes there are a number of avenues for future work. The jumping performances could be improved by investigating various beam cross-sections to improve the efficiency of strain energy stored per gram. A hammer like element attached to the end of the beam may also help in energy transfer by ensuring beam contact with the ground. The latch structure and material could also be investigated further. Including key-

stone elements into the latch structure could decrease the period between jumps dramatically. These latches would remain strong while the key-stone element is in place, but their strength would rapidly decrease once the keystone element dissolved. Alternatively, the material of the latches could be triggered by the presence or absence of a compound of interest in the environment. Responsive hydrogels (Liu et al., 2021; Koetting et al., 2015) could be used to make latches respond to various stimuli such as pH and temperature. This could allow the robots to sense their environment, and physically move based on whether a triggering material is present. Additionally, latches are not limited to controlling jumping, but could also release a payload when triggered. Combining these two behaviors could create a system of robots which would accumulate in certain areas and then selectively release fertilizer or a remedial agent into its vicinity over a long time scale. Robots could also release a payload that interacts with other robots for the purposes of communication. This could lead to swarm-like behaviors and improve the system's performance. For example, a payload released by one robot could cause nearby robots to jump. This could move robots away from each other, distributing robots more evenly over the area. Another possible application of stochastic robots could involve the system spreading out over an area to act as localization beacons for more sophisticated robots in noisy hazardous environments.

Conclusion

This work presents the first steps towards using large numbers of randomly jumping robots to cover an area of interest. Simulations demonstrate the flexible design space which would allow many robot configurations, with different numbers of jumps and total stored elastic energy, to achieve good coverage. For example, 500 robots can achieve over 90% area coverage with the robot design biased towards either reduced number of jumps (7 total jumps) or reduced jump length (1.24 m). The demonstrated robot prototypes contains all the core functionalities of the simulated system, including pre-loaded sequential jumps, environmental triggering and sensing, ease of production, low cost, and potential for biodegradability. Future work will focus on scaling up the system towards outdoor demonstrations.

Acknowledgements

Julian Hird was supported by the EPSRC Center for Doctoral Training in Future Autonomous and Robotic Systems (FARSCOPE) at the Bristol Robotics Laboratory. All authors contributed equally to the work.

Conflict of Interest

The authors declare no conflict of interest.

References

- opencv. URL <https://github.com/opencv/opencv>. Accessed on Fri, October 01, 2021.
- Rhodri Armour, Keith Paskins, Adrian Bowyer, Julian Vincent, and William Megill. Jumping robots: A biomimetic solution to locomotion across rough terrain. In *Bioinspiration and Biomimetics*, volume 2, sep 2007. doi: 10.1088/1748-3182/2/3/S01.
- Jacob Beal. Superdiffusive dispersion and mixing of swarms with reactive levy walks. *International Conference on Self-Adaptive and Self-Organizing Systems, SASO*, pages 141–148, 2013. ISSN 19493673. doi: 10.1109/SASO.2013.9.
- Bitbucket. Prototype CAD Files. 2021. URL https://bitbucket.org/jhird11/stochastic_swarm_cad/src/master/.
- Ian B Burgess, Marko Lončar, and Joanna Aizenberg. Structural colour in colourimetric sensors and indicators. *Journal of Materials Chemistry C*, 1(38):6075–6086, 2013.
- Cristina Dimidov, Giuseppe Oriolo, and Vito Trianni. Random walks in swarm robotics: an experiment with kilobots. In *International Conference on Swarm Intelligence*, pages 185–196. Springer, 2016.
- M. Ding, L. Jing, H. Yang, C.E. Machnicki, X. Fu, K. Li, I.Y. Wong, and P.-Y. Chen. Multifunctional soft machines based on stimuli-responsive hydrogels: from freestanding hydrogels to smart integrated systems. *Materials Today Advances*, 8:100088, 2020. ISSN 2590-0498. doi: <https://doi.org/10.1016/j.mtadv.2020.100088>. URL <https://www.sciencedirect.com/science/article/pii/S2590049820300357>.
- Steven Dubowsky, Samuel Kesner, Jean Sébastien Plante, and Penelope Boston. Hopping mobility concept for search and rescue robots. *Industrial Robot*, 35(3):238–245, 2008. ISSN 0143991X. doi: 10.1108/01439910810868561.
- A. E. Turgut E. Ugur and E. Sahin. Dispersion of a swarm of robots based on realistic wireless intensity signals. In *2007 22nd international symposium on computer and information sciences*, pages 1–6, 2007.

Teledyne FLIR. SDK. URL <https://www.flir.co.uk/products/spinnaker-sdk/>.

Florian Hartmann, Melanie Baumgartner, and Martin Kaltenbrunner. Becoming Sustainable, The New Frontier in Soft Robotics. *Advanced Materials*, 33(19):2004413, 2021. doi: <https://doi.org/10.1002/adma.202004413>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.202004413>.

Sabine Hauert, Sara Mitri, Laurent Keller, and Dario Floreano. Evolving cooperation: From biology to engineering. Technical report, MIT press Cambridge, MA, 2014.

Helmut Hauser, Auke J Ijspeert, Rudolf M Fuchsli, Rolf Pfeifer, and Wolfgang Maass. Towards a theoretical foundation for morphological computation with compliant bodies. *Biological cybernetics*, 105(5-6):355–370, 2011.

Andrew Howard, Maja J. Matarić, and Gaurav S. Sukhatme. Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem. *Distributed Autonomous Robotic Systems 5*, pages 299–308, 2002. doi: 10.1007/978-4-431-65941-9_30.

Michael C Koetting, Jonathan T Peters, Stephanie D Steichen, and Nicholas A Peppas. Stimulus-responsive hydrogels: Theory, modern advances, and applications. *Materials Science and Engineering: R: Reports*, 93:1–49, 2015.

Mirko Kovač, Martin Fuchs, André Guignard, Jean Christophe Zufferey, and Dario Floreano. A miniature 7g jumping robot. *Proceedings - IEEE International Conference on Robotics and Automation*, (c):373–378, 2008. ISSN 10504729. doi: 10.1109/ROBOT.2008.4543236.

Cecilia Laschi, Barbara Mazzolai, and Matteo Cianchetti. Soft robotics: Technologies and systems pushing the boundaries of robot abilities. *Science Robotics*, 1(1), 2016. doi: 10.1126/scirobotics.aah3690. URL <https://robotics.sciencemag.org/content/1/1/eaah3690>.

Jing Liu, Zhen Chen, Yujie Chen, Hafeez Ur Rehman, Yutong Guo, Hua Li, and Hezhou Liu. Ionic Conductive Organohydrogels with Dynamic Pattern Behavior and Multi-Environmental Stability. *Advanced Functional Materials*, 31(24):2101464, 2021. doi: <https://doi.org/10.1002/adfm.202101464>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/adfm.202101464>.

Luke Ludwig and Maria Gini. Robotic swarm dispersion using wireless intensity signals. *Distributed Autonomous Robotic Systems 7*, pages 135–144, 2006. doi: 10.1007/4-431-35881-1-14.

- James McLurkin and Jennifer Smith. Distributed Algorithms for Dispersion in Indoor Environments Using a Swarm of Autonomous Mobile Robots. *Distributed Autonomous Robotic Systems 6*, (June):399–408, 2008. doi: 10.1007/978-4-431-35873-2_39.
- S. Mintchev, D. Zappetti, J. Willemin, and D. Floreano. A Soft Robot for Random Exploration of Terrestrial Environments. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 7492–7497, 2018. ISSN 10504729. doi: 10.1109/ICRA.2018.8460667.
- Kohei Nakajima, Helmut Hauser, Tao Li, and Rolf Pfeifer. Information processing via physical soft body. *Scientific Reports*, 5, 2015. doi: 10.1038/srep10487.
- Ahmed Z Naser, I Deiab, and Basil M Darras. Poly (lactic acid)(PLA) and polyhydroxyalkanoates (PHAs), green alternatives to petroleum-based plastics: a review. *RSC Advances*, 11 (28):17151–17196, 2021.
- Emilia W Nery and Lauro T Kubota. Sensing approaches on paper-based devices: a review. *Analytical and bioanalytical chemistry*, 405(24):7573–7595, 2013.
- Minkyun Noh, Seung Won Kim, Sungmin An, Je Sung Koh, and Kyu Jin Cho. Flea-inspired catapult mechanism for miniature jumping robots. *IEEE Transactions on Robotics*, 28(5): 1007–1018, 2012. ISSN 15523098. doi: 10.1109/TRO.2012.2198510.
- Anil Ozdemir, Melvin Gauci, Andreas Kolling, Matthew D. Hall, and Roderich Gros. Spatial coverage without computation. *Proceedings - IEEE International Conference on Robotics and Automation*, 2019-May:9674–9680, 2019. ISSN 10504729. doi: 10.1109/ICRA.2019.8793731.
- F. Remondino, L. Barazzetti, F. Nex, M. Scaioni, and D. Sarazzi. UAV PHOTOGRAMMETRY FOR MAPPING AND 3D MODELING – CURRENT STATUS AND FUTURE PERSPECTIVES. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII-1/C22:25–31, 2011. doi: 10.5194/isprsarchives-XXXVIII-1-C22-25-2011. URL <https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXVIII-1-C22/25/2011/>.
- Jonathan Rossiter, Jonathan Winfield, and Ioannis Ieropoulos. Here today, gone tomorrow: biodegradable soft robots. *Electroactive Polymer Actuators and Devices (EAPAD) 2016*, 9798:97981S, 2016. ISSN 1996756X. doi: 10.1117/12.2220611.
- M Schranz, M Umlauf, M Sende, and W Elmenreich. Swarm Robotic Behaviors and Current Applications. *Front Robot AI*, 7:36, 2020.
- Giovanni Scirè Mammano, Eugenio Dragoni, and L. F.M. Da Silva. Mechanical design of buckled beams for low-stiffness elastic suspensions: Theory and application. *Proceedings of*

the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications, 231(1-2):140–150, 2017. ISSN 20413076. doi: 10.1177/1464420716670930.

Michael Siebold and James Hereford. Easily scalable algorithms for dispersing autonomous robots. *Conference Proceedings - IEEE SOUTHEASTCON*, pages 545–550, 2008. ISSN 07347502. doi: 10.1109/SECON.2008.4494354.

Gabor Soter, Andrew Conn, Helmut Hauser, Nathan Lepora, and Jonathan Rossiter. Multi-Tip: A multimodal mechano-thermal soft fingertip. *IEEE International Conference on Soft Robotics (RoboSoft)*, pages 239–244, 2018. doi: 10.1109/ROBOSOFT.2018.8404926.

Zhenishbek Zhakypov, Kazuaki Mori, Koh Hosoda, and Jamie Paik. Designing minimal and scalable insect-inspired multi-locomotion millirobots. *Nature*, 28, 2019. ISSN 14764687. doi: 10.1038/s41586-019-1388-8. URL <http://dx.doi.org/10.1038/s41586-019-1388-8>.

Brandon M. Zoss, David Mateo, Yoke Kong Kuan, Grgur Tokić, Mohammadreza Chamanbaz, Louis Goh, Francesco Vallegra, Roland Bouffanais, and Dick K.P. Yue. Distributed system of autonomous buoys for scalable deployment and monitoring of large waterbodies. *Autonomous Robots*, 42(8):1669–1689, dec 2018. ISSN 15737527. doi: 10.1007/s10514-018-9702-0.

Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. *Lecture Notes in Computer Science*, 3342:10–20, 2005. ISSN 03029743. doi: 10.1007/978-3-540-30552-1_2.