Supplementary Information for

Continuous Learning of Emergent Behavior in Robotic Matter

Giorgio Oliveri, Lucas C. van Laake, Cesare Carissimo, Clara Miette, Johannes T.B. Overvelde

Corresponding Author Johannes T.B. Overvelde.
E-mail: overvelde@amolf.nl

This PDF file includes:

Supplementary text
Figs. S1 to S14
Legends for Movies S1 to S5
SI References

Other supplementary materials for this manuscript include the following:

Movies S1 to S5

Supporting Information Text

Fabrication of units

Each unit consists of a micro-controller (ESP32), a motion sensor (ADNS-9800) and a pressure sensor (Honeywell MPRLS0015 PG0000SA), mounted on a custom made PCB embedded in a 3D printed frame with a total dimension of $68 \times 44 \times 42$ mm. The frame is designed to support a micro-pump (Huizhou Yingyi Motor Co. YYP032), a 3.7V battery (Carson 500608131), and has two mechanical snap connectors to attach the soft actuators. The soft actuators consist of three bellows of 25 mm outer diameter, a length of 35 mm and a total inner volume of ≈ 2 ml, which were moulded using a soft elastomer (Smooth-ON Dragon Skin 30). The mould consists of two external components printed on a polyjet printer (Stratasys Eden260VS), and a soluble PVA core printed on a FDM printer (Ultimaker 3). After casting, the soft actuators were connected to a water pump (Eheim 1250-790) to flush out the soluble core over a period of approximately two days. The actuators were connected on both sides to luer lock couplings (Nordson Medical FTLL035-1 and MLRL035-1) with heat-shrink tubes (RS-458-068). One side of the actuator is then connected to a venting needle (Metcal 9922050-TE gauge 22, inner diameter of 0.412 mm and length of 11 mm), via a male-female luer elbow coupling (Nordson Medical LE87-1), and two luer lock caps (Nordson Medical FSLLR-3). The other side of the actuator is connected to the pump and the pressure sensor via a silicone tube (outer and inner diameter 3 mm and 1.5 mm, respectively), fitted with a barb to luer lock connector (Nordson Medical MLRL007-1). The mass m of each unit is equal to 63.7 g, including 10.2 g for the complete actuator and 10.8 g for the battery. After connecting two units using a soft actuator, the distance at rest between the two adjacent units equals 62 mm. Importantly, the assembled units touch the surface using four 5.5×8 mm screws. To maximize the accuracy of the motion sensor, the distance of the PCB from the surface was set to be ≈ 5 mm, achieved with a screw head of 3 mm thickness, and four nylon rings of 0.54 mm thickness per screw. The width of the units at the place they touch the ground (i.e., width between the screws), which fits into the track, has dimension of 55.4 mm. An overview of the components and assembly is given in Fig. S10.

Experimental setup

The learning experiments are launched by initiating each unit at a random phase ϕ_i. The initialization is done via a WiFi connection that is made with each unit sequentially. While the WiFi connection is not needed to perform the experiments, the connection is used to gather sensor data during the experiments at a 10 Hz rate. After initialization, the units try out new phases after every $n_{\text{act}} = 2$ cycles of $t_{\text{cycle}} = 2$ s duration. During these two cycles, the phase ϕ_i is kept constant. In the first cycle, the change to the phase is made by scaling the total duration of the cycle according to $1 - (\phi'_i - \phi_i)/t_{\text{cycle}}$. Note that for both cycles, the pump is only turned on in the first $\alpha = 0.4$ portion of each cycle. After performing two cycles, each unit determines its average velocity U' that is used in the learning algorithm. To determine the velocity, note that within each actuation cycle the optical motion sensors take 10 displacement measurements per second as shown in Fig. S11, and the unit displacement for the learning step is determined by comparing the absolute displacement at the beginning and end of the second actuation cycle (the measurements that are located at the dashed lines). The velocity can then be determined by dividing the displacement by the actuation cycle duration of $t_{\text{cycle}} = 2$ s. Note that the implementation of the phase change, which results in an elongating or shortening of the duration of the first cycle, introduces some differences in measured velocity between units of the same robots at the same learning step, because the evaluation periods start and finish at different times. By comparing the measured velocity U' with the velocity U stored in memory, each unit determines the acceptance probability of the current phase ϕ' according to $p(\Delta U) = e^{(U'-U)/T}$, in which $T = 0.1$. Using the phase that is stored in memory ϕ, each unit then perturbs its phase according to $\phi' = \phi + \epsilon \Delta s$ to obtain a new candidate phase, in which ϵ is a random number drawn from a uniform distribution on the interval$[-1, 1]$, and $\Delta s = 0.1$ is the used stepsize. Note that while the units are initialized at the same time, the internal clocks of the units are not further synchronized and will go out of sync over time. While this is not corrected during learning (which should also not be necessary as the units should be able to adapt to this), for plotting purposed we take the learning step as reference. While the units can operate on battery, to allow for continuous testing most of the experiments were run by powering the units with a common power supply, attached to one of the active units. Power is then delivered to the other units via a unit-to-unit connection placed below the soft actuator. Moreover, the experiments were performed on a lasercut circular POM track screwed on a MDF surface. The diameter of the track is $D \approx 0.854$ m, and the track width is $w = 56.4 \pm 0.4$ mm.

The learning algorithms

Each unit runs an identical algorithm. The pseudecode for the two algorithms (Thermal and Flaky) tested in this work are given in Fig. S1.

Numerical model

We model the robot's behavior with a mass-spring system that exhibits friction with its environment (Fig. S12a). Each i-th unit consists of a mass m_i, and an actuator with stiffness k_i and initial length L_i. Furthermore, we assume that each unit experiences a nonlinear friction $F(\dot{x}_i)$ with the surface that depends on its velocity. We can then write the system's equation of motion as follows,

$$\mathbf{M}\ddot{\mathbf{x}} = -\mathbf{K}\mathbf{x} - \mathbf{F}(\dot{\mathbf{x}}) + \mathbf{A}(t), \qquad [1]$$

where $\mathbf{M}$ is a mass matrix, $\mathbf{K}$ is a stiffness matrix, $\mathbf{F}$ is a friction force vector and $\mathbf{A}$ a vector containing the actuation forces exerted by the actuators on the connected masses. Note that the forces that are applied by each actuator are modelled by changing the rest length L_i of each spring by a deformation $\Delta L_i(t)$, such that the actual rest length of the actuator at any moment in time equals $l_i(t) = L_i + \Delta L_i(t)$. Specifically, the matrices for the assembled system containing $n-1$ active units (i.e., n masses) are equal to

$$\mathbf{M} = \begin{bmatrix} m_1 & & & \\ & m_2 & & \\ & & \ddots & \\ & & & m_n \end{bmatrix},$$

$$\mathbf{K} = \begin{bmatrix} k_1 & -k_1 & & & \\ -k_1 & k_1+k_2 & -k_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -k_{n-2} & k_{n-2}+k_{n-1} & -k_{n-1} \\ & & & -k_{n-1} & k_{n-1} \end{bmatrix},$$

$$\mathbf{A}(t) = \begin{bmatrix} k_1 l_1(t) \\ k_2 l_2(t) - k_1 l_1(t) \\ \vdots \\ k_{n-2} l_{n-2}(t) - k_{n-1} l_{n-1}(t) \\ k_{n-1} l_{n-1}(t) \end{bmatrix}.$$

As an example, if we assume that all units are identical (i.e., $m_i = m$ and $k_i = k$), for two active units we have

$$m \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \begin{bmatrix} \ddot{x_1} \\ \ddot{x_2} \\ \ddot{x_3} \end{bmatrix} = -\,k \begin{bmatrix} 1 & -1 & \\ -1 & 2 & -1 \\ & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} F(\dot{x_1}) \\ F(\dot{x_2}) \\ F(\dot{x_3}) \end{bmatrix} + k \begin{bmatrix} l_1(t) \\ l_2(t) - l_1(t) \\ l_2(t) \end{bmatrix}. \quad [2]$$

In experiments, the pump is cyclically turned on and off for t_{on} and t_{off} seconds, respectively. When the pump is turned on, the flow into the soft actuator is approximately constant. Simultaneously, the actuator vents air through a needle. To model the inflation and deflation cycle of the actuator characterized by the actuation function $l_i(t)$ in Eq. (1), we turn to an electronic analogy (1). If we assume the pump acts like a current source, the actuator as a capacitor, and the needle as a resistor (with parameters I, C and R, respectively), the voltage V in the capacitor during one full cycle (representing the pressure in the actuator) is given by

$$V = \begin{cases} V_{\mathrm{off}} + (IR - V_{\mathrm{off}})(1 - e^{-t/(RC)}) & \text{for } 0 \leq t \leq t_{on} \\ V_{\mathrm{on}} e^{-t/(RC)} & \text{for } t_{\mathrm{on}} \leq t \leq t_{\mathrm{off}} + t_{\mathrm{on}}, \end{cases} \quad [3]$$

where V_{off} is the voltage in the capacitor at the end of the previous deflation step, and V_{on} is the voltage in the capacitor at the end of the inflation step. We next transform this voltage to extension of an actuator by assuming $\Delta L = \beta V$, such that

$$l_i(t) = L_i + \beta V_i, \quad [4]$$

where we assumed a linear relation between the pressure inside the actuator and the extension of the actuator.

We model the friction force $F(\dot{x})$ for each mass as a combination between a Coulomb $F_{\mathrm{C}}(\dot{x})$ and a viscous $F_{\mathrm{v}}(\dot{x})$ friction (2). The Coulomb friction is given by

$$F_{\mathrm{C}}(\dot{x}) = (\mu_{\mathrm{Brk}} mg) \tanh\left(\frac{\dot{x}}{v_{\mathrm{C}}}\right), \quad [5]$$

in which μ_{Brk} is the breakaway friction coefficient between each mass and the surface, g is the gravitational constant, and v_{C} is the Coulomb velocity threshold which scales the speeds at which the friction barrier is broken. Moreover, the viscous friction equals

$$F_{\mathrm{v}}(\dot{x}) = c\dot{x}, \quad [6]$$

where c is the viscous damping coefficient. The total friction force that acts on each mass equals

$$F(\dot{x}) = F_{\mathrm{C}}(\dot{x}) + F_{\mathrm{v}}(\dot{x}). \quad [7]$$

Note that the μ_{Brk} parameter for the Coulomb friction can be observed experimentally. In contrast, from an experimental perspective v_{C} is ideally equal to zero. However, in simulations we used $v_{\mathrm{C}} = 0.001$ to smooth the transition and prevent numerical problems. Moreover, we included the viscous term F_{v} to numerically dampen the system and avoid possible numerical instabilities or high-frequency oscillations. We noticed that the choice of parameter c only effects the quantitative agreement between simulations and experiments shown in Fig. 1g, and we found a value of $c = 5$ to be appropriate.

Simulation setup

To determine the motion of the assembled system of equations, we numerically solve Eq. (1) using the implicit Radau method implemented in a custom script in Python with the SciPy integration library (version 1.4.1). To perform a learning simulation, in the setup phase a worm object is instantiated and all parameters are set to their initial values. This includes the resting length of the springs which is always set to a constant value of $L = 60$ mm. A random phase ϕ_i is assigned to each unit, along with a fixed number of learning steps and a fixed number of actuation cycles $n_{\text{act}} = 10$ per learning step, during which the phases ϕ_i are kept constant. After the last learning step, units calculate their score U' as the average distance travelled relative to the number of actuation cycles. Units use this score to update their phases as done in experiments. Similarly, the phases are also perturbed in the same way after each learning step, and new candidate phases ϕ_i' are set for the next learning step. Note that at the beginning of every learning step, we reset the displacements of the units to their rest length, such that the system undergoes transient behavior during the first actuation cycle, as can be seen in the first seconds in Fig. S12d. This is the result of the difference between the natural rest length of the spring 60 mm, and the new rest length as determined by the new actuation phase ϕ_i'. This does not significantly affect the U', as we use a relatively large number of actuation steps.

Model parameter measurements

To characterize the behavior of the soft actuator (i.e., stiffness k and the extension ΔL), we measured the force-displacement response of the actuator using a mechanical testing machine (Instron 5965L9510). To determine the stiffness, we start with an actuator that is open to the surroundings and perform a pulling experiment four times, in which we deform the actuator at a constant deformation rate of 120 mm/min to an extension of 20 mm, return to the initial position, and compress 3 mm beyond the initial position (Fig. S13a). Based on these results, we extract the slope of the curve to determine the stiffness k of the actuator, as shown in Fig. S13b.

To determine if the stiffness of the actuator depends on the internal pressure, we next performed a second experiment where we start with an unpressurized, but closed actuator, and performed the same extension/compression testing routine. We then add 1 ml of air to the actuator using a syringe pump, leading to an increase in the internal pressure. We then closed a valve in order to disconnect the syringe during the pulling experiment and repeated this procedure for a total of four measurement cycles. In Fig. S13c-d we show the measured force-deformation curves and the corresponding stiffness k for the closed cases, respectively. We find that the stiffness is not significantly affected by the internal pressure when the actuator is closed to the surroundings. It is important to note that during compressive loading at small or negative extensions the actuator undergoes a global buckling instability that suddenly changes the stiffness (Fig. S13c-d). However, such an instability has never been observed in the actuators during normal operation of the units. Moreover, we also find that the closing of the actuator to the surroundings increases the stiffness. This is likely the result of the enclosed air that acts as an additional spring. Since the experimental situation is between open and closed, and preliminary results showed that the response is qualitatively identical independently of the exact numerical value of stiffness, we chose $k = 100$ N/m for the numerical simulations.

To determine the parameters used to specify the actuation cycle, we measure the maximum extension $\Delta L_{\max}$ and corresponding pressure $P_{\max}$ of the actuator during cyclic operation. We find that $\Delta L_{\max} = 12.5$ mm and $P_{\max} = 6$ kPa. Given that the maximum pressure in the actuator equals the maximum voltage $V_{\text{on}} = 6$ in our fluidic-electric model (Eq. (4)), we find that $\beta \approx 2.08$ mm/kPa. We next tune I, R and C such that $V_{\text{on}} \approx 6$, and $V_{\text{off}} \approx 0.01 V_{\text{on}}$ (such that the actuator is empty at the end of the deflation cycle). We find that values of $I = 0.35$, $R = 18$ and $C = 0.0145$ give a realistic inflation and deflation behavior according to Eq. (3), as shown in Fig. S12b.

To characterize the friction $F(\dot{x})$ of the unit moving on the experimental table, we attached one unit to a horizontal material testing machine (Instron 5900 equipped with 100 N load cell) as shown in Fig. S14a), and measured the reaction force when prescribing a displacement (Fig. S14b) to the unit that is moving on a similar surface as the one used in the learning experiments. Note that we used the bottom part of one of the units, and placed additional weight on it to a total of $m_{\text{Test}} = 0.811$ kg, in order to increase the resolution in our measurements. To understand if this coefficient is influenced by the relative speed of the units, we performed five tests in which a maximum displacement of 126 mm is reached after ramping up and ramping down the displacement rate as shown in Fig. S14b. Assuming that the friction force is equal to $\mu_{\text{Brk}}\, m_{\text{Test}}\, g$, the resulting friction coefficient μ_{Brk} is shown in Fig. S14c. Apart from the initial increase in friction coefficient when the unit is set into motion, we find an approximate constant friction force as a function of the unit's relative velocity. Based on this result, we chose $\mu_{Brk} = 0.24$ as the breakaway friction coefficient in our numerical model (Eq. (5)). The final friction behavior is shown in Fig. S12c.

Difference between experiments and simulations

We believe that the noise observed in experiments is mainly caused by three effects, related to the sensing hardware, decentralization, and environment.

First, while the position sensor is relatively precise (steps of 0.016 mm) and can handle much higher speeds (up to 150 mm/s, we do observe some rotation when the units are moving along the track. Such rotation is not captured by the sensors, and could therefore cause an integration error. This causes a relatively high noise for phases that result in approximately no movement.

Second, since measurements of the position are done separately by each unit, the moment at which the position is measured by each unit varies in time (e.g., see Fig. S11). Therefore, when one unit is measuring its displacement, another unit might still be moving. This effect is furthermore strengthened by the fact that units' internal clocks go out of sync as the learning progresses. This noise therefore depends on the state of the system, and the time during experiments.

Third, experimental noise is also caused by variations in the track (the track has been the same for all experiments). As mentioned in the section on experimental setup, some regions of the tracks are narrower than others, where we observe a standard deviation of ~ 0.4 mm between the 20 measurements at different track locations. This causes the units' feet to experience more play in some regions and less in others, leading to small rotations besides the tangential translation when pushed by the actuators. This is also demonstrated by the results in Fig. 2d, where we show the average measured velocities across the system for 20 measurements starting from different initial positions. In Fig. 2d we show the standard deviation between these measurements highlighting the large variation in performance for fixed phase combinations and different positions. Some of the individual experimental scan can be seen in Fig. S4, where the landscape is explored by fixing $\phi_1 = 0$ and ϕ_2, and varying ϕ_3. There is a significant dependency of the potential behavior the units can exhibit and where they are on the track, and so this effect is rather a dynamic condition, rather than noise.

To determine if the velocity measurement taken by the local sensors is accurate, we also compare the units' displacement measurements with an external camera. Considering the experiment with fixed phases for three active units in Movie 2, we see that it takes approximately 1182 seconds to complete a full circle of length 2.68 meters. Comparing this to the distance measured with the sensors, we find the three motion sensors underestimate the traveled distance with relative errors of $1\%, 9.7\%$ and 10.7% respectively.

Importantly, we did not try to further reduce the above effects, but rather embrace them, as one of our goals is to design a robust learning algorithm that is capable of dealing with dynamic (and noisy) situations. Note that all three cases have not been included in most simulations, except for a noise term similar to sensor noise that we have studied in Fig. S5, and changing friction and damage shown in Fig. S6 and Fig. S7, respectively.

Algorithm 1 Thermal algorithm in pseudocode

Initialize random phase ϕ
Evaluate velocity U
for Step in n_{learn} **do**
    Perturb phase stored in memory $\phi' = \phi + \epsilon\Delta s$
    Evaluate new velocity U'
    Determine acceptance probability $p(\Delta U) = e^{(U'-U)/T}$
    if $p(\Delta U) \geq rand(0,1)$ **then**
        Store new phase in memory $\phi = \phi'$
        Store new velocity in memory $U = U'$

Algorithm 2 Flaky algorithm in pseudocode

Initialize random phase ϕ
Evaluate velocity U
for Step in n_{learn} **do**
    Perturb phase stored in memory $\phi' = \phi + \epsilon\Delta s$
    Evaluate new velocity U'
    Determine acceptance probability $p(\Delta U) = e^{(U'-U)/T}$
    if $p(\Delta U) \geq rand(0,1)$ **then**
        Store new phase in memory $\phi = \phi'$
    Store new velocity in memory $U = U'$

Fig. S1. Pseudocode of the Thermal and Flaky algorithms.

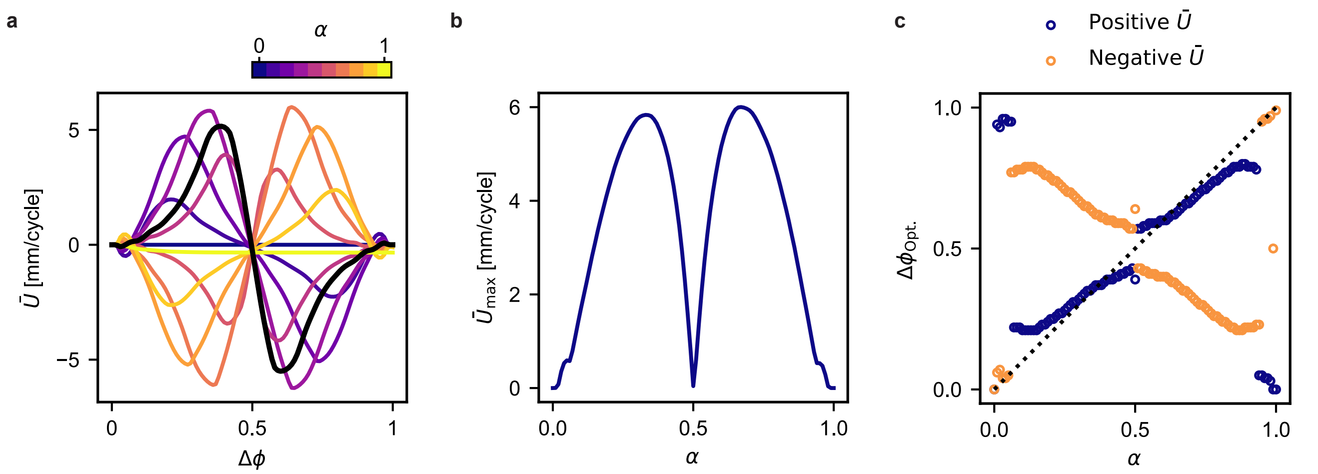

Fig. S2. Simulations to determine the influence of the actuation duration α on the displacement of a two active unit robot. a, Average velocity of the assembled robot as a function of the phase difference $\delta\phi = \phi_2 - \phi_1$, for different values of α. The black line shows the response for $\alpha = 0.4$, that has been used throughout our studies. **b**, Maximum system velocity $\bar{U}_{\max}$ as a function of α. **c**, Optimal phase difference $\Delta\phi_{\mathrm{Opt.}}$ (for both positive and negative $\bar{U}$) as a function of α.

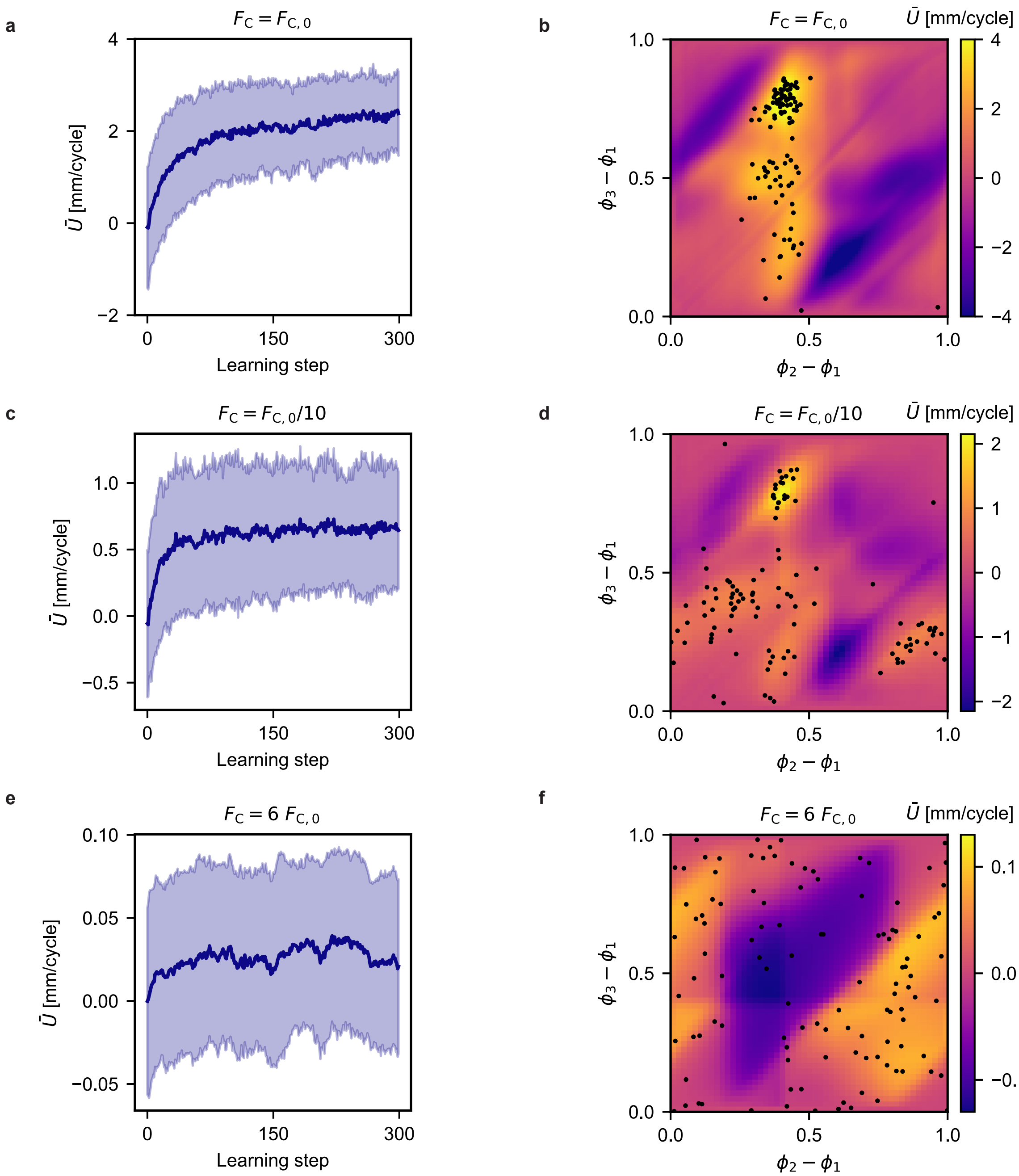

Fig. S3. Effect of friction on the potential velocity and ability to learn of a three active units robot running the Flaky algorithm. We performed 112 simulations for each of three different frictions, where we determine both the average velocity $\bar{U}$ as a function of the learning step (where the shaded area represents the standard deviation), and the optimal phase combinations reached at the end of the last learning step, as indicated by the dots in the contour plot. **a-b**, Results for the friction similar to the experiments ($F_{\mathrm{C}} = F_{\mathrm{C},0}$) as used throughout our studies. **c-d**, Results for reduced friction ($F_{\mathrm{C}} = F_{\mathrm{C},0}/10$), qualitatively similar to the robot moving on ice. **e-f**, Results for increased friction ($F_{\mathrm{C}} = 6\ F_{\mathrm{C},0}$).

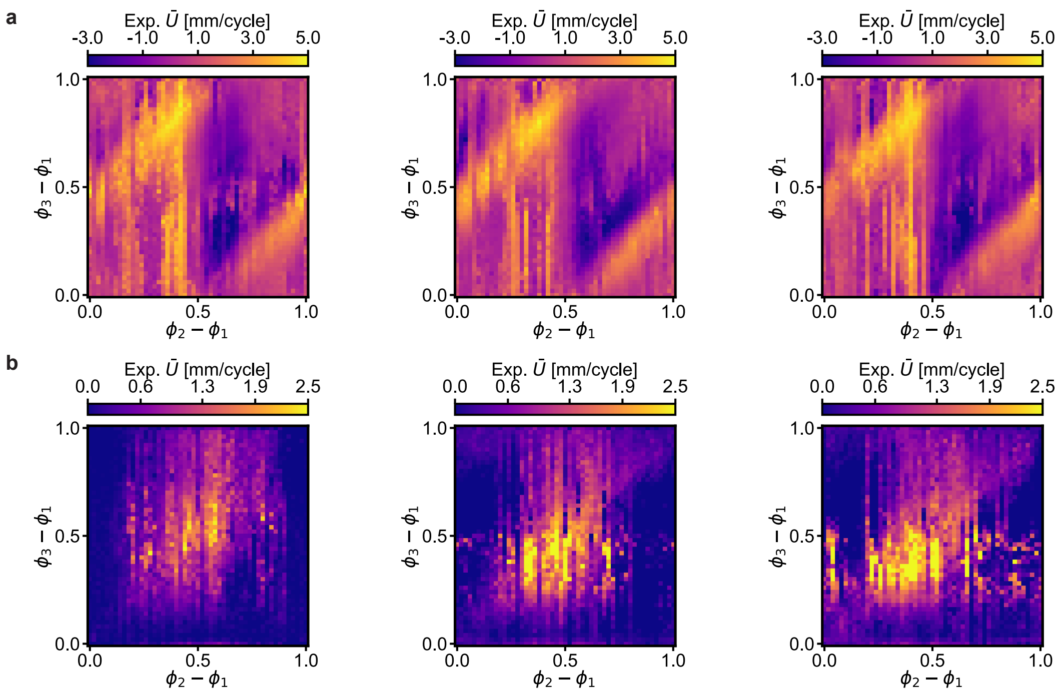

Fig. S4. Selection of experimental velocity scans for an assembled robot consisting of three active units and one dummy unit. The results show three out of 20 experimental runs that were used to determine the average speed $\bar{U}$ as a function of all the possible combinations of phases $\phi_3 - \phi_1$ and $\phi_2 - \phi_1$ (with $\phi_1 = 0$), for (**a**) an undamaged and (**b**) a damaged robot. Each scan is initialized by placing the robot at different starting positions.

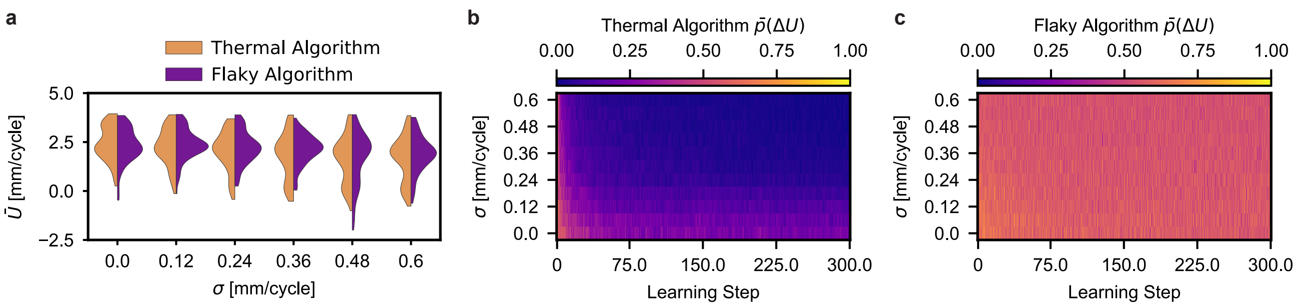

Fig. S5. Effect of noise on learning strategies. To understand how the performance of the Thermal and Flaky algorithms are effected by noise, we performed simulations on a assembled robot consisting of three active and one dummy unit, in which the measured speed U_i of each unit was altered by a random value u drawn from a Gaussian distribution with standard deviation ranging from $0 \leq \sigma \leq 0.6$. For each selected value of σ we performed 100 simulations. **a**, Distribution of measured velocities after 300 learning steps, as a function of the standard deviation σ of the noise applied. We notice that for both algorithms, even for larger values of the standard deviation, the system learns how to move (although less effectively). However, for the Thermal algorithms and for an applied noise with $\sigma > 0.24$ $[\mathrm{mm/cycle}]$ we see an increasing number of simulations with $\bar{U} \approx 0$. **b**, Important to note is that for the Thermal algorithm the noise level has a large effect on the acceptance probability $\bar{p}(\Delta U)$, which quickly tends to go to zero for increasing number of learning steps and for $\sigma > 0.024$ $[\mathrm{mm/cycle}]$. **c**, In contrast, noise does not seem to effect the acceptance probability $\bar{p}(\Delta U)$ when using the Flaky algorithm.

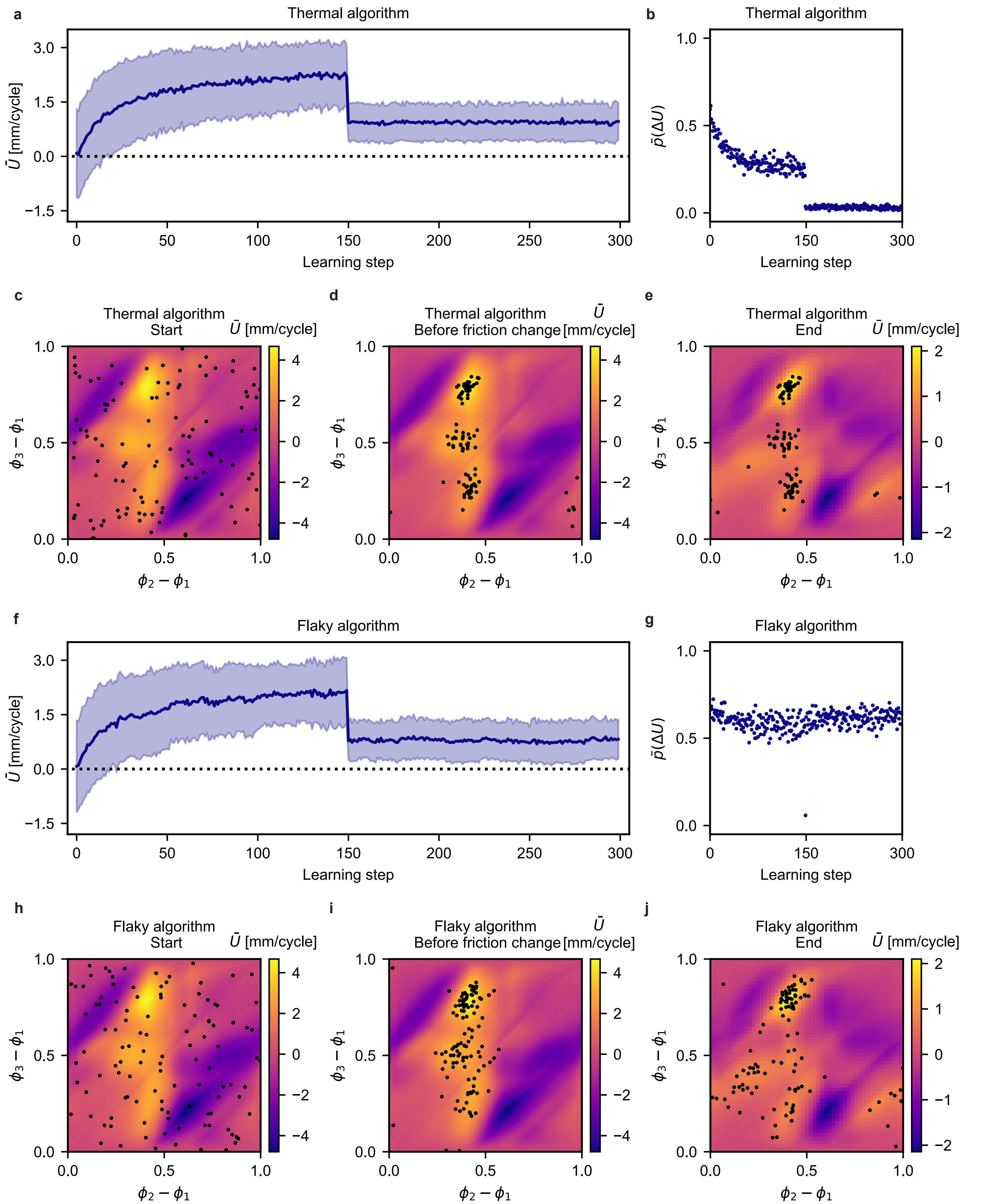

Fig. S6. Learning simulations of a three active units robot with sudden and sharp drop in friction at the 150^{th} learning step for both Thermal (a-e) and Flaky algorithms (f-j). The results of 112 simulations per algorithm type are shown in terms of average velocity $\bar{U}$ and average acceptance rate $\bar{p}$, as a function of the learning step (**a-b** and **f-g**). In **c-e** and **h-j** we show the optimal phases combinations at the beginning of the learning, just before the friction change, and at the end of the last learning step.

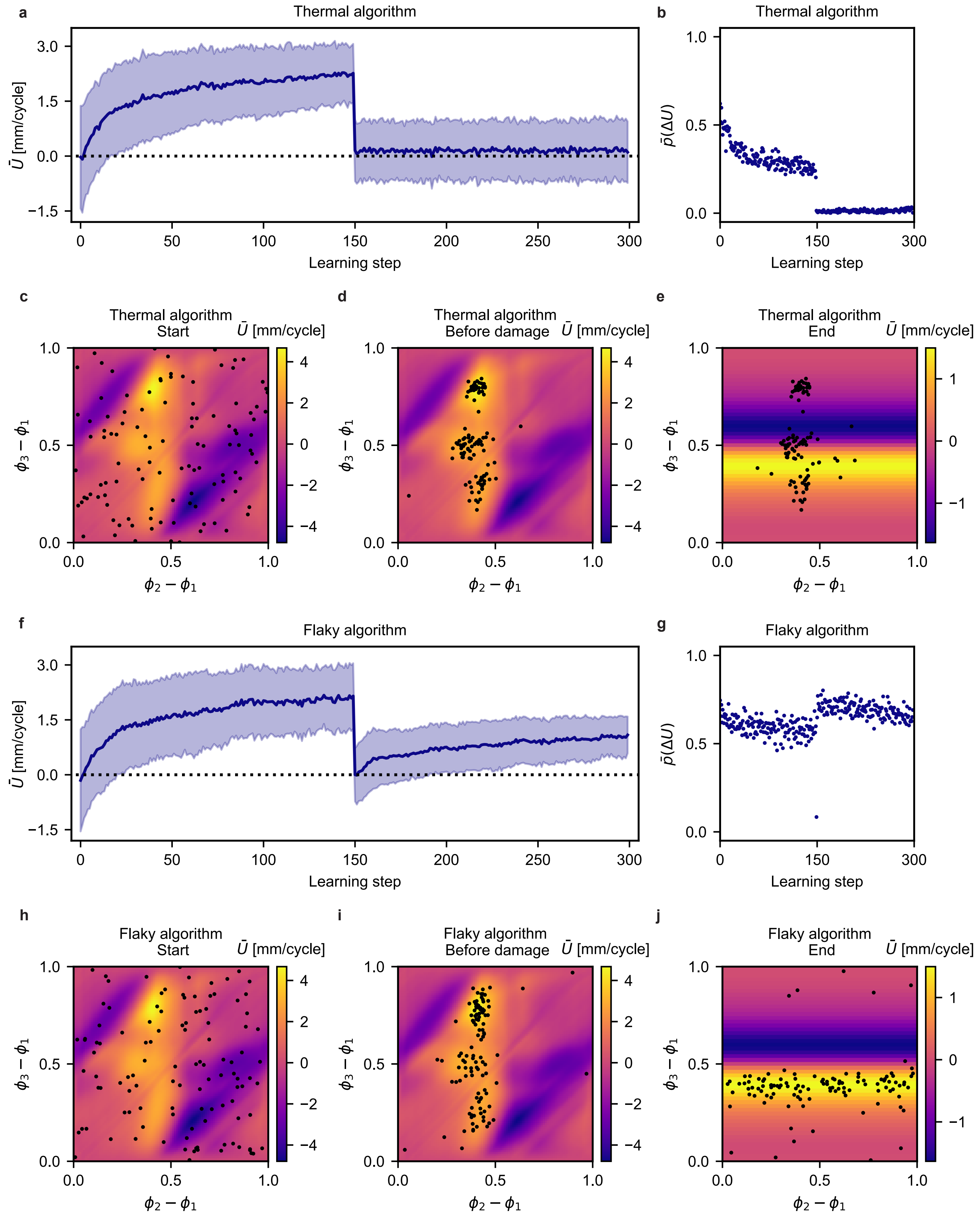

Fig. S7. Learning simulations of a three active units robot with sudden damage on the second unit at the 150^{th} learning step for both Thermal (a-e) and Flaky algorithms (f-j). The results of 112 simulations per algorithm type are shown in terms of average system velocity $\bar{U}$ and average acceptance rate $\bar{p}$, as a function of the learning step (**a-b** and **f-g**). In **c-e** and **h-j** we show the optimal phases combinations at the beginning of the learning, just before the damage is applied, and at the end of last learning step

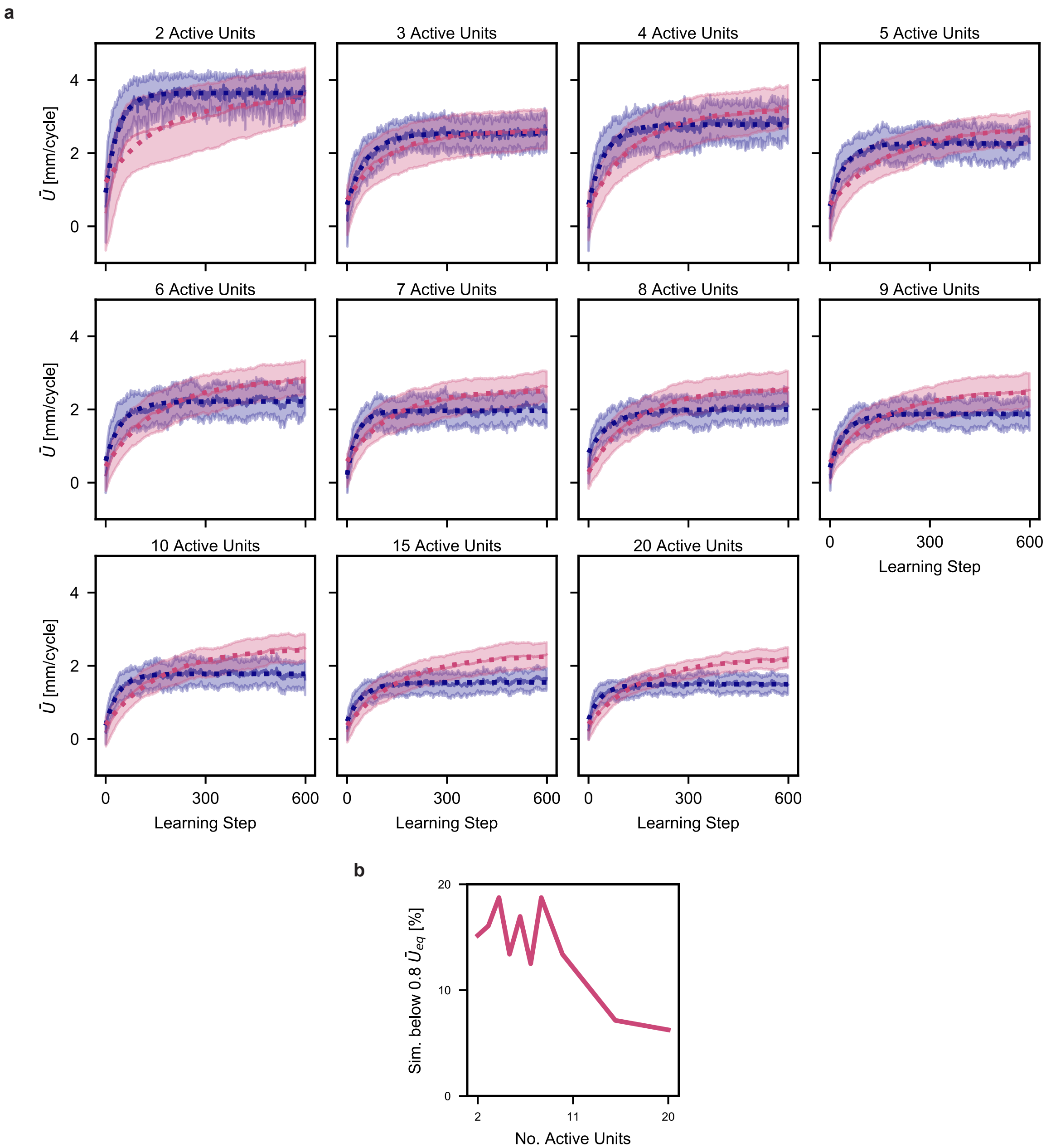

Fig. S8. Scalability of the decentralized learning approach. To study the effect of the number of units on the effectiveness of our updated learning algorithm, we performed numerical learning experiments on assembled robots consisting of two to twenty active units (and one dummy unit). We performed these simulations for two different stepsizes of $\Delta s = 0.1$ and $\Delta s = 0.025$, in blue and pink respectively. **a,** Solid lines represents the average over 112 simulations, while the shaded area represents the standard deviation. The average curves have been fitted by an exponential function $y = \bar{U}_{eq} - \left(\bar{U}_{eq} - \bar{U}_0\right) e^{-\gamma\, x}$, shown by the dashed line. **b,** Percentage of simulations for $\Delta s = 0.025$ not reaching the $0.8\,\bar{U}_{eq}$ threshold within 600 learning steps. Note that for $\Delta s = 0.1$ all simulations reach $0.8\,\bar{U}_{eq}$.

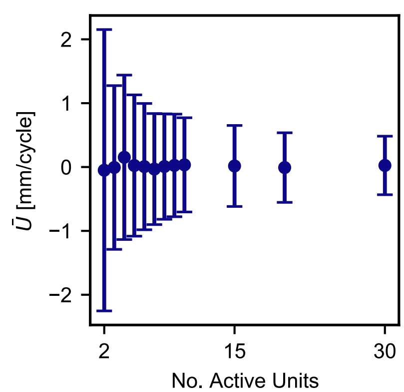

Fig. S9. Sampling the scalable behavior. Simulated system velocity as a function of the number of active units. The dots represents the average value, and the bars indicate standard deviation values over the 1000 random samples per system size.

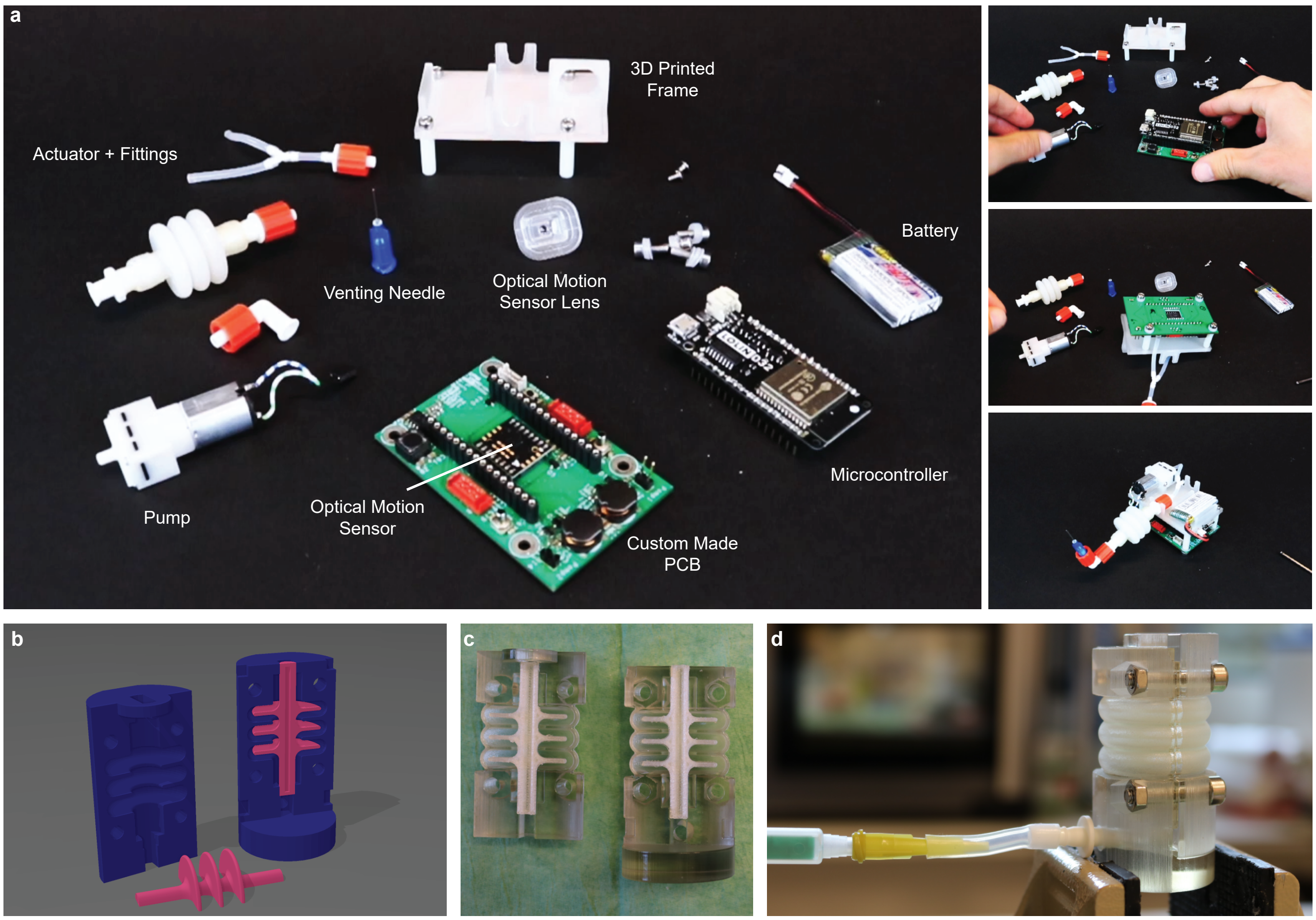

Fig. S10. Unit assembly and bellow actuator fabrication. **a,** Overview of unit components and assembly process. **b**, CAD models of bellow actuator moulds. The outer mould (in blue) was printed in VeroClear (Stratasys Eden260VS), and the inner core (in pink) was printed in soluble PVA (Ultimaker 3). **c**, 3D printed parts before casting. **d**, Injection casting of bellow actuator with Dragon Skin 30 (Smooth-ON).

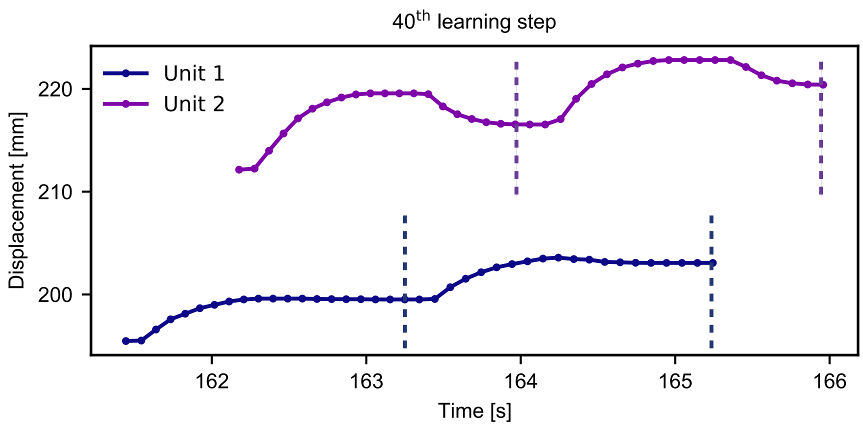

Fig. S11. Measured displacements at the 40^{th} learning step by the two active units in the learning experiment shown in Fig. 1d. The vertical dashed lines mark the beginning and end of the second cycle, that is used to calculate the units' average velocities U_1 and U_2.

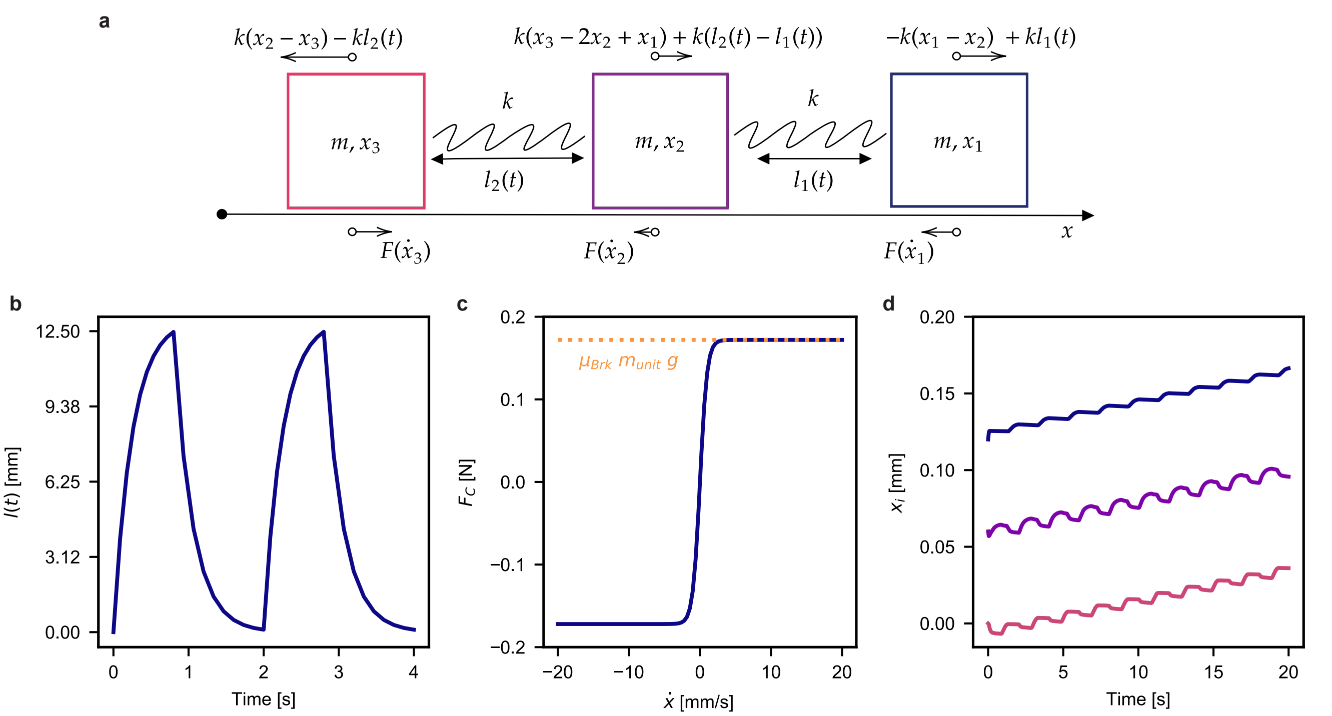

Fig. S12. Numerical model schematic and implemenation. **a**, Mass-spring model schematic for an assembled robot consisting of two active units (and one dummy unit). **b,** Actuator's preferred spring length $l(t)$ as a function of time for the specific case of $\phi = 0$. **c**, Modelled Coulomb friction F_C as a function of the relative velocity $\dot{x}$ to the ground. **d**, Example of displacements x_i for the three masses in a learning step of $n_{\rm act} = 10$ actuation cycles, with $\phi_1 = 0$ and $\phi_2 = 0.4$.

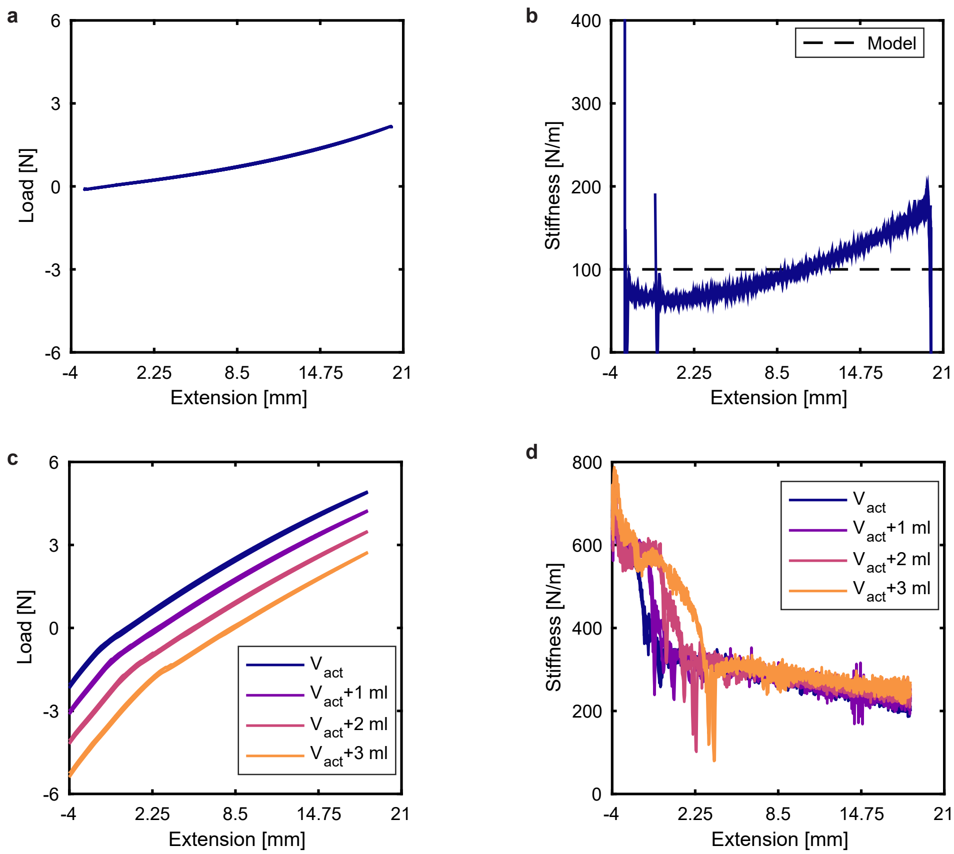

Fig. S13. Characterization of a soft actuator. **a**, **b**, Force-extension and stiffness-extension response of an actuator open to the environment. The value $k = 100\ N/m$ was chosen as actuator stiffness in the numerical model. **c**, **d**, Force-extension and stiffness-extension response for an actuator closed to the environment (i.e., contains a fixed amount of air), for increasing enclosed air volumes from $V_{tot} = V_{act}$ to $V_{tot} = V_{act} + 3\ ml$, in steps of $1\ ml$.

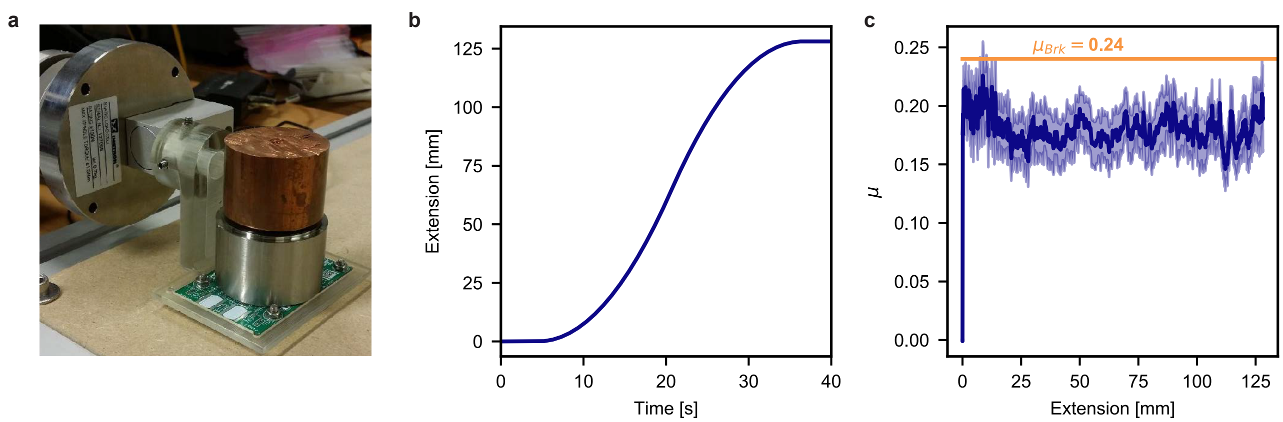

Fig. S14. Friction characterization between the unit and the table. **a**, overview of the horizontal friction characterization experiment. Additional weight has been added to the unit to increase the magnitude of the force and reduce relative noise. **b**, Applied displacement (extension) profile as a function of time. **c**, Measured friction coefficient, and standard deviation, as function of the applied extension. The value $\mu_{Brk} = 0.24$ was chosen as breakaway friction coefficient to use in the numerical model.

Movie S1. Learning experiment for an assembled robot. In this video we show learning experiments for two and three active unit (and one dummy unit) with Thermal algorithm, and a learning experiment with three active unit (and one dummy unit) with Flaky algorithm.

Movie S2. Influence of track on the robot motion. In this video we show the variability of the track by performing two experiments with fixed actuation phases on a robot assembled from three active units (and one dummy unit).

Movie S3. Adaptability to damage. In this video we show how the assembled robot adapts to damage, by performing two experiments on a robot consisting of three active units (and one dummy unit), using the Thermal and Flaky algorithm.

Movie S4. Scalability of the algorithm. In this video we demonstrate that the Flaky algorithm can also be applied to larger systems, and perform an experiment on an assembled robot consisting of seven active units (and one dummy unit).

Movie S5. Autonomous modular robot that learns how to move. In this video we demonstrate that the proposed learning strategy does not need any digital connection between units, and that the robot can operate fully untethered outside lab settings. We show an experiment in which we assembled three active units (and a dummy unit), after which we let the assembled robot learn how to move. We subsequently shuffle the unit position such that a different locomotion pattern needs to be learnt.

References

1. KW Oh, K Lee, B Ahn, EP Furlani, Design of pressure-driven microfluidic networks using electric circuit analogy. *Lab on a Chip* **12**, 515–545 (2012).
2. B Armstrong, C de Wit, *Friction Modeling and Compensation, The Control Handbook.* (CRC Press), (1995).