Supplementary Materials to

Memristor circuits for colloidal robotics: Temporal access to memory, sensing, and actuation

Jing Fan Yang[1], Albert Tianxiang Liu[1,2,3], Thomas A. Berrueta[4], Ge Zhang[1], Allan M. Brooks[1], Volodymyr B. Koman[1], Sungyun Yang[1], Xun Gong[1], Todd D. Murphey[4], and Michael S. Strano[1*]

[1]Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[2]Department of Molecular and Cellular Physiology, School of Medicine, Stanford University, Stanford, CA 94305, USA
[3]Department of Chemical Engineering, University of Michigan, Ann Arbor, MI 48109, USA
[4]Department of Mechanical Engineering, Northwestern University, Evanston, IL 60208, USA
[*]Corresponding author. Email: strano@mit.edu

Contents

1. Methods

The simulations in this work, including both the circuit simulations and pharmacokinetic modeling, are performed in MATLAB R2020a (The MathWorks, Inc. of Natick, MA).

1.1 Simulation of a Discrete 1D Memristor Array

In a discrete 1D memristor array, such as the one simulated in Figure 1G of the Main Text, the N cross-plane memristors are individually defined and bridged in between by resistors. The simulation of this scenario is simple. We define the instantaneous memristance of the n^{th} memristor $M_n(t) \in [R_{\text{ON}}, R_{\text{OFF}}]$ and the resistance of the n^{th} fixed resistor $R_{\text{R}n}$. The lumped resistance of the array to the right of the n^{th} memristor (itself included), $\text{R}_{\text{L}n}(t)$, can be expressed recursively as:

$$R_{\text{L}n}(t) = \left(M_n^{-1} + R_{\text{L}n+1}^{-1} \right)^{-1} + R_{\text{R}n} \tag{S1}$$

At the terminus with the n^{th} memristor, $R_{\text{L}N} = M_N + R_N$. With a supply voltage V_0, the combined current passing through the power source $I_0(t)$ is thus:

$$I_0(t) = V_0 / R_{\text{L}1} \tag{S2}$$

By Kirchhoff's first law, the current passing through the n^{th} memristor, $I_n(t)$, is:

$$I_n(t) = \left(I_0 - \sum_{j=1}^{n-1} I_j \right) \cdot \frac{R_{\text{L}n+1}}{M_n + R_{\text{L}n+1}} \tag{S3}$$

Returning to the dynamics of a single memristor outlined in Equations 1-3 of the Main Text, we arrive at the memristor array dynamics here:

$$\frac{d}{dt} w_n(t) = \frac{\mu R_{\text{ON}}}{L^2} I_n(t) f_{\text{W}}(w_n) \tag{S4}$$

μ, R_{ON}, and L are respectively the dopant mobility, the ON-state memristance, and the effective device length L, consistent with the Main Text. The additional term, f_{W}, is a window function dependent on the state variable which enforces numerical stability and adds nonlinearity near the device bounds. A number of window functions have been proposed in the literature, such as $f_{\text{W}}(w) = w - w^2$, $f_{\text{W}}(w, I) = 1 - \left[w - (I < 0) \right]^{2p}$, $f_{\text{W}}(w) = 1 - \left[(w - 0.5)^2 + 0.75 \right]^p$, and $f_{\text{W}}(w) = 1 - (2w - 1)^{2p}$ [1–3]. The last window function is used throughout this work, where $p = 10$. The instantaneous memristance is defined by the state variable by:

$$M_n(t) = w_n(t) \cdot R_{\text{ON}} + \left[1 - w_n(t) \right] \cdot R_{\text{OFF}} \tag{S5}$$

Setting the initial state variables of all N memristors to be 0.999, we can then trace the sequential switching behavior of the memristor array by solving the ordinary differential equations in S4.

Like a continuous memristor array, the sequential switching slows down as time elapses if the resistors bridging the memristors are identical. This phenomenon, described in the Discussions section of the Main

Text and §2.1 below, can be resolved in a discrete array via resistive tuning. That is, as the memristors and resistors are spatially defined and separate, the individual resistances may be tuned by, for instance, surface functionalization [4]. Figure 1G of the Main Text shows the switching dynamics of a tuned 7-memristor array where the intervals are uniform between each memristor's switching and the subsequent one. For the specific simulation, the tuned resistances of the 7 resistors are respectively 50.0, 7.5, 4.0, 3.2, 3.2, 3.6, and 6.0kΩ.

1.2 Simulation of a Continuous 1D Memristor Array

As Figure 1H in the Main Text shows, an 1D memristor array can be made from a continuous phase of a memristive material. In this case, the memristance M_n of a standalone memristor is replaced by the local memristance measured at a location x, or $M(x, t)$. Similarly, the local voltage V and longitudinal current along x I_L are now continuous functions of x and t as well. As shown in Figure S1, we can treat each memristor node as an infinitesimal unit within an otherwise homogenous layer of memristive material. In this limit, the spatial gradient of the local electrical potential down the longitudinal direction is:

$$\frac{\partial}{\partial x}V(x,t) = -R_R(t)\cdot I_L(x,t) \tag{S6}$$

in accordance with the classical cable theory originally used to calculate the current and voltage along neurons. Incidentally, the Nobel-prize-winning Hodgkin-Huxley circuit used to model a neuron's axon has a mathematically equivalent form consisting of memristors, bearing striking resemblance to the memristor array reported in our work [5,6]. Note that since $R_R(t)$ in the continuous model measures the instantaneous longitudinal resistance per unit length, it assumes the unit of Ω/μm. Anticipating experimental difficulty in doping the conductive surfaces with high spatial resolution, we assume R_R is not made a function of x. It may, however, be a function of time should chemiresistors be used such as in Figure 2 of the Main Text, as the chemiresistance is modulated by the time-dependent chemical analyte profile.

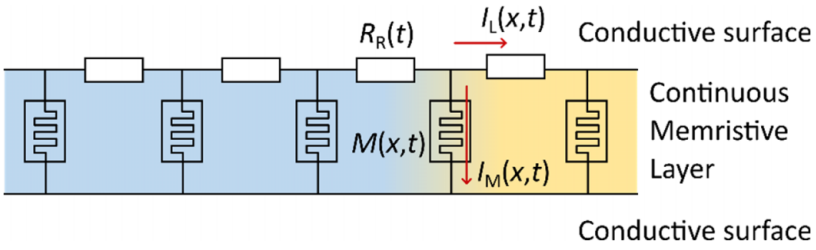

Figure S1. Schematic of the simulation of a continuous 1D memristor array fabricated by laying a continuous phase of memristive ink between two conductive surfaces.

Accounting for the transverse current $I_M(x, t)$ per length that goes through the memristive layer, we establish the following relation:

$$\frac{\partial}{\partial x}I_L(x,t) = -I_M(x,t) = -\frac{V(x,t)}{M(x,t)} = -\frac{V(x,t)}{w(x,t)\cdot R_{ON} + [1-w(x,t)]\cdot R_{OFF}} \tag{S7}$$

where $w(x, t)$ is naturally the instantaneous state variable for the particular locality on the memristive layer. Its dynamics is again described by Equation 1 of the Main Text:

$$\frac{\partial}{\partial t}w(x,t) = \frac{\mu R_{ON}}{L^2}I_M(x,t)f_W \tag{S8}$$

Note that the units for $I_M(x, t)$ and $M(x, t)$ are respectively A/μm and Ω·μm. Reorganizing Equations S6-8 yields the following system of highly coupled, non-linear partial differential equations (PDEs):

$$\begin{cases} \dfrac{\partial^2}{\partial x^2} V(x,t) = \dfrac{R_R(t)}{R_{\mathrm{ON}}} \cdot \dfrac{V(x,t)}{w(x,t) + \left[1 - w(x,t)\right] \cdot \alpha} \\ \dfrac{\partial}{\partial t} w(x,t) = \dfrac{\mu}{L^2} \cdot \dfrac{V(x,t)}{w(x,t) + \left[1 - w(x,t)\right] \cdot \alpha} f_{\mathrm{W}} \end{cases} \tag{S9}$$

where $\alpha = R_{\mathrm{OFF}}/R_{\mathrm{ON}}$ is the switching range of a memristor. We used the method of lines (MOL) to simulate the PDE system by discretizing the spatial domain with a stencil of accuracy 2. We can also nondimensionalize the PDE system to better understand its mathematical characteristics, which is documented in detail in §2.1.

1.3 2D Memristor Array Simulation

2D memristor arrays presented in Figure 3 of the Main Text feature 1D branch arrays along x grafted on a single 1D trunk array along y. The branch and trunk memristors, in addition to the circuit breaker memristors which switch off a branch's voltage access after a short window of time, are treated as distinct circuit elements in this simulation for generality. They may, however, be identical in practice if desired. For clarity, we define in Table S1 all quantities involved when simulating a 2D array.

Table S1. Physical quantities involved in simulating a 2D memristor array.

Symbol	Definition	Unit	Value
$V_x(x, t)$	Local voltage on the trunk at location x and time t	[V]	-
$V(x, y, t)$	Local voltage on the branches at location (x, y) and time t	[V]	-
$I_x(x, t)$	Local longitudinal current on the trunk at location x and time t	[A]	-
R_x	Resistance of fixed resistors on the trunk	[Ω / μm]	3×10^3
$R_y(t)$	Instantaneous resistance of chemiresistors on the branches at time t	[Ω]	-
$R_{\mathrm{ON},x}$	ON state resistance of memristors on the trunk	[Ω · μm]	1×10^2
$R_{\mathrm{ON},y}$	ON state resistance of memristors on the branches	[Ω · μm^2]	1×10^2
$R_{\mathrm{ON},s}$	ON state resistance of the circuit breaker memristors	[Ω · μm]	1×10^2
$w_x(x, t)$	State variable of the differential memristor on the trunk at location x and time t	[-]	[0, 1]
$w_y(x, y, t)$	State variable of the differential memristor on a branch at location (x, y) and time t	[-]	[0, 1]
$w_s(x, t)$	State variable of the circuit breaker memristor for the branch at location x and time t	[-]	[0, 1]
α_x	Switching range of memristors on the trunk, i.e., the ON/OFF ratio	[-]	2×10^2
α_y	Switching range of memristors on the branches	[-]	2×10^2
α_s	Switching range of the circuit breaker memristors	[-]	5×10^5
μ_x	Mobility of memristors on the trunk normalized by L^2	[min^{-1}V^{-1}]	4
μ_y	Mobility of memristors on the branches normalized L^2	[min^{-1}V^{-1}]	3.5×10^3
μ_s	Mobility of the circuit breaker memristors normalized L^2	[min^{-1}V^{-1}]	1×10^3

We again start by applying the classical cable theory to the trunk:

$$\frac{\partial V_x(x,t)}{\partial x} = -R_x I_x(x,t) \tag{S10}$$

The reduction of the longitudinal current along the *x*-axis is due to two factors: current flowing through trunk memristors and that entering the local branch array:

$$\frac{\partial I_x(x,t)}{\partial x} = \underbrace{-\frac{1}{R_{\mathrm{ON},x}} \frac{V_x(x,t)}{w_x(x,t) + \left[1 - w_x(x,t)\right]\alpha_x}}_{\text{Current via trunk memristors}} \underbrace{+\left(\frac{1}{R_y(t)} \frac{\partial}{\partial y} V(x,y,t)\right)_{s+}}_{\text{Current via branch array}} \tag{S11}$$

where *s*+ denotes the location right after the circuit breaker memristor along a branch array. Combining the two equations above yields:

$$\frac{\partial^2 V_x(x,t)}{\partial x^2} = R_x \left\{ \frac{1}{R_{\mathrm{ON},x}} \frac{V_x(x,t)}{w_x(x,t) + \left[1 - w_x(x,t)\right]\alpha_x} - \left(\frac{1}{R_y(t)} \frac{\partial}{\partial y} V(x,y,t)\right)_{s+} \right\} \tag{S12}$$

Along the *y*-axis, on the other hand, Equation S9 still applies for each branch array:

$$\frac{\partial^2}{\partial y^2} V(x,y,t) = \frac{R_y(t)}{R_{ON,y}} \frac{V(x,y,t)}{w_y(x,y,t) + \left[1 - w_y(x,y,t)\right]\alpha_y} \tag{S13}$$

As with before, the state variables w_x, w_y, and w_s are regulated by the currents going through the respective memristors:

$$\begin{cases} \dfrac{\partial}{\partial t} w_y(x,y,t) = -\mu_y \dfrac{V(x,y,t)}{w_y(x,y,t) + \left[1 - w_y(x,y,t)\right]\alpha_y} f_{\mathrm{W}}(w_y) \\ \dfrac{\partial}{\partial t} w_x(x,t) = -\mu_x \dfrac{V_x(x,t)}{w_x(x,t) + \left[1 - w_x(x,t)\right]\alpha_x} f_{\mathrm{W}}(w_x) \\ \dfrac{\partial}{\partial t} w_s(x,t) = -\mu_s \dfrac{V_x(x,t) - V(x,y_{s+},t)}{w_s(x,t) + \left[1 - w_s(x,t)\right]\alpha_s} f_{\mathrm{W}}(w_s) = \mu_s f_{\mathrm{W}}(w_s) R_{\mathrm{ON},s} \left(\dfrac{1}{R_y(t)} \dfrac{\partial}{\partial y} V(x,y,t)\right)_{s+} \end{cases} \tag{S14}$$

The PDE system consisting of Equations S12-14 is again solved using the method of lines. We discretize the *x* and *y* dimension respectively into N_i and N_j points with step sizes Δx and Δy. As a result, $V_x(x, t)$, $V(x, y, t)$, $w_x(x, t)$, $w_y(x, y, t)$, and $w_s(x, t)$ become $V_i^x(t)$, $V_{i,j}(t)$, $w_i^x(t)$, $w_{i,j}(t)$, and $w_i^s(t)$.

The current entering a branch can be discretized with a finite difference stencil of accuracy 2:

$$\left(\frac{1}{R_y(t)} \frac{\partial}{\partial y} V(x,y,t)\right)_{s+} = \frac{1}{2\Delta y R_y(t)} \left(-3V_{i,1}(t) + 4V_{i,2}(t) - V_{i,3}(t)\right) \tag{S15}$$

The PDE system is then transformed into the following form:

$$\left\{\begin{array}{l}\left.\begin{array}{l}\dfrac{dw_{i,j}(t)}{dt}=-\mu_y\gamma_{i,j}(t)R_{\mathrm{ON},y}V_{i,j}(t)f_{\mathrm{W}}(w_{i,j})\\ \dfrac{dw_i^x(t)}{dt}=-\mu_x\gamma_i^x(t)R_{\mathrm{ON},x}V_i^x(t)f_{\mathrm{W}}(w_i^x)\\ \dfrac{dw_i^s(t)}{dt}=\dfrac{\mu_s R_{\mathrm{ON},s}}{2\Delta y R_y(t)}\left[-3V_{i,1}(t)+4V_{i,2}(t)-V_{i,3}(t)\right]f_{\mathrm{W}}(w_i^s)\end{array}\right\}\text{ODEs}\\ \left.\begin{array}{l}\gamma_i^s(t)\left[V_i^x(t)-V_{i,1}(t)\right]+\dfrac{1}{2\Delta y R_y(t)}\left[-3V_{i,1}(t)+4V_{i,2}(t)-V_{i,3}(t)\right]=0\\ V_{i-1}^x(t)-\left\{2+R_x\left(\Delta x\right)^2\left[\gamma_i^x(t)+\gamma_i^s(t)\right]\right\}V_i^x(t)+V_{i+1}^x(t)+R_x\left(\Delta x\right)^2\gamma_i^s(t)V_{i,1}(t)=0\\ V_{i,j-1}(t)-\left\{2+R_y(t)\left(\Delta y\right)^2\gamma_{i,j}(t)\right\}V_{i,j}(t)+V_{i,j+1}(t)=0\end{array}\right\}\text{Linear Equations}\end{array}\right. \tag{S16}$$

where $\gamma_{i,j}=\left\{R_{ON,y}\left[w_{i,j}+\left(1-w_{i,j}\right)\alpha_y\right]\right\}^{-1}$, $\gamma_i^s=\left\{R_{\mathrm{ON},s}\left[w_i^s+\left(1-w_i^s\right)\alpha_s\right]\right\}^{-1}$, $\gamma_i^x=\left\{R_{\mathrm{ON},x}\left[w_i^x+\left(1-w_i^x\right)\alpha_x\right]\right\}^{-1}$ are all functions of time. At each time point, therefore, we update voltages at all locations over the 2D space by solving the linear equations. We then use these updated parameters, as well as the chemiresistance R_{y} (t) determined by the instantaneous analyte concentration, to progress the ODE solution. To make use of MATLAB's extremely efficient backslash operator, it is best to put the linear equations in their matrix form:

$$\underbrace{\left(\begin{array}{ccccccccccccccc}1&&&&&&&&&&&&&&\\ 1&-2-R_x\left(\gamma_2^x+\gamma_2^s\right)\left(\Delta x\right)^2&1&&&&R_x\gamma_2^s\left(\Delta x\right)^2&&&&&&&&\\ &\ddots&\ddots&\ddots&&&&\ddots&&&&&&&\\ &&1&-2-R_x\left(\gamma_{N_i-1}^x+\gamma_{N_i-1}^s\right)\left(\Delta x\right)^2&1&&&&R_x\gamma_{N_i-1}^s\left(\Delta x\right)^2&&&&&&\\ &&½&-2&³⁄₂&&&&&&&&&&\\ \hline \gamma_1^s&&&&&-\dfrac{3}{2\Delta yR_y}-\gamma_1^s&&&\dfrac{2}{\Delta yR_y}&&&-\dfrac{1}{2\Delta yR_y}&&&\\ &\ddots&&&&&\ddots&&&\ddots&&&\ddots&&\\ &&\gamma_{N_i}^s&&&&&-\dfrac{3}{2\Delta yR_y}-\gamma_{N_i}^s&&&\dfrac{2}{\Delta yR_y}&&&-\dfrac{1}{2\Delta yR_y}&\\ \hline &&&&&1&&&-2-R_y\gamma_{1,2}\left(\Delta y\right)^2&&&1&&&\\ &&&&&&\ddots&&&\ddots&&&\ddots&&\\ &&&&&&&1&&&-2-R_y\gamma_{i,j}\left(\Delta y\right)^2&&&1&\\ &&&&&&&&\ddots&&&\ddots&&&\ddots\\ &&&&&&&&&1&&&-2-R_y\gamma_{N_i,N_j-1}\left(\Delta y\right)^2&&1\\ \hline &&&&&&&½&&&-2&&&³⁄₂&\\ &&&&&&&&\ddots&&&\ddots&&&\ddots\\ &&&&&&&&&½&&&-2&&³⁄₂\end{array}\right)}_{\mathbf{A}}\underbrace{\left(\begin{array}{c}V_1^x\\ \vdots\\ V_{N_j}^x\\ \hline V_{1,1}\\ \vdots\\ V_{N_i,1}\\ \hline V_{1,2}\\ \vdots\\ V_{i,j}\\ \vdots\\ V_{N_i,N_j-1}\\ \hline V_{1,N_j}\\ \vdots\\ V_{N_i,N_j}\end{array}\right)}_{\mathbf{V}}=\underbrace{\left(\begin{array}{c}V_0\\ \vdots\\ 0\\ \hline 0\\ \vdots\\ 0\\ \vdots\\ 0\\ \hline 0\\ \vdots\\ 0\end{array}\right)}_{\mathbf{B}} \tag{S17}$$

where **A** is a N_i (N_j+1)-by-N_i (N_j+1) square matrix and **V** is a N_i (N_j+1)-by-1 vector. At each time point, the linear equations are then solved simply by computing **V** = **A\B**. For the simulations shown in Figure 3 of the Main Text, all w, w^x, and w^s are set to 0.999 initially.

We finally address the calibration of 2D memristor arrays illustrated in Figure S2. As discussed in the Main Text and mathematically analyzed in §2.1, the ON/OFF boundary propagates slower with time. In the context of a 2D memristor array, the voltage is initially passed down the trunk (x-axis) swiftly but slows down gradually: Figure S2A shows that although a 2D memristor array can log a temporal sinusoidal profile, the corresponding map of raw memristor states is skewed, with the original periodic oscillation compressed at later times (i.e., larger x). We consequently calibrate the x-axis so that the memristor states faithfully represent the form of the sensed chemical profile. The calibration relation between elapsed time and coordinates along x, such as the one shown in Figure S2B, is established with a quick 1D simulation of just the standalone trunk array. Specifically, we track the frontier location of the ON/OFF boundary, defined as the x coordinate where w = 0.99, as a function of time. Rescaling the x coordinates of the raw data with the

calibration relation yields the periodic readout shown in Figure S2C, where the peaks and valleys map very well onto those of the sinusoidal profile sensed by the array, despite the deteriorating resolution at later times. The result validates using a 1D memristor array as a proxy for the trunk, suggesting that the branches impact only to a minimal degree the sequential switching along *x*. We note that the calibration relation is an intrinsic property of the circuit and independent of the analyte profile: the same calibration relation is applied to Figure S8Figure S11.

Somewhat surprisingly, the analyte-responsive sequential switching along the branches (*y*) is unaffected by the distortion along *x*. For example, while the four peaks in Figure S2A are not evenly spaced along the *x* dimension, they correspond to the same *y* location on the branches. A more telling example is given in Figure S10A, where the constant profile is not distorted along *y* and the final ON/OFF boundary locations on all branches are identical. This observation is caused by the deceleration along *x* and that along *y* cancelling each other. As §2.1 shows mathematically, the slowing switching further down the trunk is a direct result of the lower voltage across the memristor. Incidentally, this is the same voltage sustaining the switching of the corresponding branch. The sluggish switching along *y*, caused by the lower input voltage at the trunk, is therefore sustained for a proportionately longer time window as the propagation along *x* slows down as well. The two factors thus cancel each other and ensure the same analyte concentration always corresponds to the same set of memristor states on all branches.

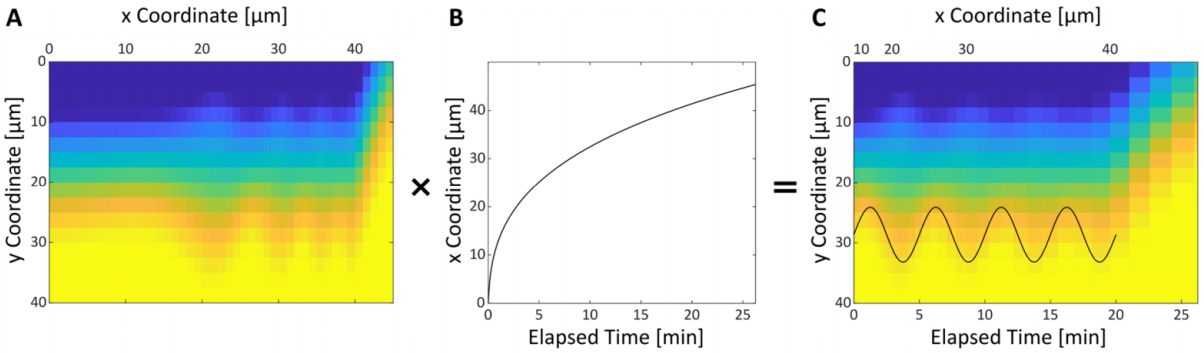

Figure S2. Calibration of a 2D memristor array. (A) Memristor states as measured show that the ON/OFF front resembles the sinusoidal analyte profile but is distorted. (B) A calibration relation between location of the activated branch on *x* and time can be established by simulating the trunk as a 1D memristor array. (C) Scaling the x-axis with the calibration location yields a memristor state map that agrees well with the sensed analyte profile.

1.4 Pharmacokinetic and Circuit Simulations of Glucose-Responsive Insulins

In the Main Text, we optimized the design of a microrobotic glucose-responsive insulin (GRI) through circuit and pharmacokinetic simulations. While the latter determined constraints on the intermediate GRI parameters such as the threshold glucose level $[G]_{th}$, they are translated by circuit simulation into optimal ranges of memristor array parameters such as V_0, R_{ON}, and α (Figure S3).

We used a pharmacokinetic model named PAMERAH that has been published and validated in the literature [7]. The model is composed of two parts: a physiological model of the full-body glucoregulatory system and a user-defined GRI kinetics. As seen in Figure S4, the physiological model approximates the organs participating in glucose metabolism as an interconnected network of well-mixed compartments, in which glucose, glucagon, native insulin, and GRI are circulated, produced, consumed, or cleared. For example, the concentration of active insulin [I] in the adipose compartments is governed by:

$$\frac{d[\mathrm{I}]_\mathrm{v}}{dt} = \underbrace{\frac{Q}{V_\mathrm{v}}([\mathrm{I}]_\mathrm{heart}\text{-}[\mathrm{I}]_\mathrm{v})}_{\text{Inflow and outflow}} - \underbrace{\frac{V_\mathrm{i}}{V_\mathrm{v}\tau_\mathrm{T}}([\mathrm{I}]_\mathrm{v} - [\mathrm{I}]_\mathrm{i})}_{\text{Transcapillary diffusion}} + \underbrace{r_{\mathrm{GRI,v}}}_{\text{Changes due to GRI action}} \tag{S18}$$

The subscripts v and i denote vascular and interstitial volumes respectively. τ_T is the transcapillary diffusion time constant, V the adipose compartmental volumes, and Q the local blood flow rate. In PAMERAH, a GRI affects glucoregulation via the final r_GRI term, which depends on the mode of action of the particular GRI. Readers are referred to the original publication for the detailed mathematical formulations. In the current work, we did not modify the physiological model and only specified the glucose-responsive release kinetics of our microrobotic GRI. If at a certain time the insulin reservoir contains N_I units of insulin to be released and the local glucose concentration sensed is [G], the instantaneous release rate is:

$$r_\mathrm{GRI} = -\frac{dN_\mathrm{I}}{Vdt} = \underbrace{\frac{\ln 2 N_\mathrm{I}}{t_{1/2}V}}_{\text{Nonlinear depletion}} \cdot \underbrace{\left\{1+\exp\left[-0.25\left([\mathrm{G}]-[\mathrm{G}]_\mathrm{th}\right)\right]\right\}^{-1}}_{\text{Glucose responsiveness}} \tag{S19}$$

where V is the compartmental volume. The first term on the right-hand side signifies that the release becomes slower as the reservoir depletes, characterized by the half-life $t_{1/2}$. The second term takes care of the glucose responsiveness: minimal insulin release for [G] < $[\mathrm{G}]_\mathrm{th}$ and vice versa. Note that in a microrobotic GRI, this comparator is realized with the voltage passed down to the branch terminus. If the terminus voltage exceeds a threshold voltage V_th, the reservoir is activated. As a result, $[\mathrm{G}]_\mathrm{th}$ is an intermediate parameter dependent upon circuit parameters like V_0, R_ON, and α. For the simulation in Figure 4C of the Main Text, we started with 100 robots each with an initial reservoir load ($N_{\mathrm{I}0}$) of 0.007 to 0.018 IU. Assuming a 2500 μm^3 reservoir filled with 20%wt of insulin [8], each robot has an insulin capacity of 14.4 IU, some three orders-of-magnitude more than the conservative range we chose. The ranges we explored for $t_{1/2}$ and $[\mathrm{G}]_\mathrm{th}$ are respectively 0.5 to 18 hours and 100 to 300 mg/dL. We simulated the GRI's action in a model parameterized previously with data from male Lewis rats [7]. By definition, a satisfactory GRI should:

(i) maintain the blood glucose concentration with in the normoglycemic range – 75 to 240 mg/dL for rats [9] – for 24 hours unless the GRI is just administered or a meal is consumed;
(ii) bring the initial hyperglycemic glucose level down to the normal range within two hours following the GRI's administration;
(iii) curb the postprandial glucose hikes by returning the glucose level to the normal range within two hours following 1 g/kg oral glucose meals.

Mathematically, any deviation from the said criteria is counted towards either a hyper- or hypoglycemia risk score. Only GRIs free from both risks are considered adequate, which are represented by the shaded region in Figure 4C.

The subsequent circuit simulation is straightforward and not so different from the earlier sections. Considering the computational efficiency, we approximated the dynamics of each memristor branch as a standalone 1D array described in §1.2. The approximation is valid as the distance travelled by a branch's ON/OFF boundary along y is independent of the branch location x on the trunk, evidenced by the final memristor states in a sinusoidal profile (Figure 3D of the Main Text) as well as Figure S10A. In a 2D memristor array, each branch is supplied power for only a short window of time. This temporary voltage input to the branch, $V_0(t)$, is approximated as a Gaussian pulse with a peak voltage ranging from 0.5 to 5V and a standard deviation of 20min. The peak location is inconsequential so long as the entire pulse is

supplied. Note that the Gaussian approximation is exact if the transmission line model is adopted (§2.2). The variable resistance of the branch chemiresistors R_R is ultimately a function of the analyte concentration, in the GRI case [G]:

$$R_R([G]) = \left[1+100\tanh\left(\left[G\right]-150/200\right)\right]^{-1} \text{ MΩ/μm} \tag{S20}$$

which follows the theoretical fit used in Koman *et al.* [10]. This chemiresistor offers a 20-fold reduction in its resistance as the analyte concentration increases from 150 to 250mg/dL, which is within the typical range of chemiresistor responsiveness. Furthermore, the sensor sensitivity is reduced towards larger [G], typical of chemiresistors as well.

The voltage passed down the 1D memristor array as a function of time is tracked by an ODE solver. For a specific combination of the peak voltage, R_{ON}, and α, we can find a corresponding threshold glucose concentration $[G]_{th}$ (V_0, R_{ON}, α) that causes the maximal voltage at the branch terminus to reach V_{th} = 0.15V. The insulin reservoir is set to be 75μm away from V_0. We may then compare the corresponding $[G]_{th}$ with its desired range from the pharmacokinetic model to determine if the V_0-R_{ON}-α combination is acceptable (Figure S3B).

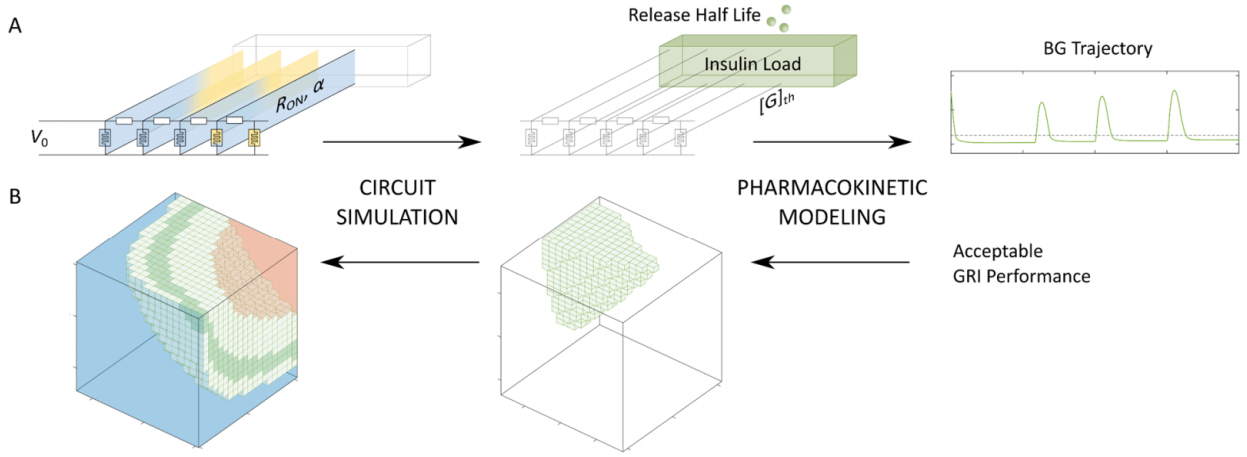

Figure S3. (A) Starting with a set of circuit parameters such as V_0, R_{ON}, and α, we run circuit simulations to obtain the corresponding threshold glucose level beyond which the insulin reservoir is activated by the terminal voltage. $[G]_{th}$ and the reservoir parameters are then used to predict the GRI's performance in the body of a diabetic subject. (B) The parameter optimization process goes the opposite way. Based on criteria for adequate GRIs, requirements on the GRI parameters can be established via pharmacokinetic modeling. With circuit simulations, optimal ranges of the circuit parameters are then found which would yield acceptable GRI parameters.

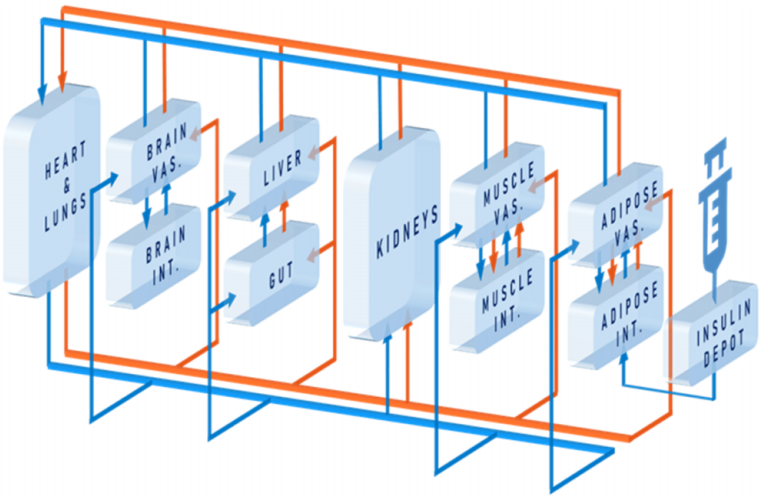

Figure S4. Schematic of PAMERAH's physiological model. The blue and red arrows represent respectively the circulation of glucose and insulin/GRI. Adapted from Ref. [7].

1.5 Spatiotemporal Simulations of a Microrobotic GRI Swarm

This simulation is designed to be illustrative of the use of microrobotic GRIs to regulate nontrivial spatial distributions of glucose over an environment (e.g., glucose agglomeration in specific tissues). We begin by framing the assumptions and design choices of our microrobotic GRI swarm simulation. First, we do not endow the glucose and insulin concentrations with diffusive dynamics. This choice is made so as to illustrate the role movement and spatial effects play on the performance of the microrobotic GRIs in an environment with persistent structural features. Otherwise, the initial glucose and insulin distributions diffuse evenly through the simulated volume, making the glucose regulation problem spatially homogenous and visually opaque. Here, by insisting on spatially heterogenous glucose concentrations, we highlight the dynamic nature of the closed-loop insulin release of each microrobotic agent. Second, we assume that due to well-mixed flow conditions the agent dynamics are captured by Langevin diffusion with drift (i.e., a spatially isotropic diffusion tensor with a linear drift term capturing a net flow). An important note is that in this class of diffusion processes (or flow conditions) the movement of particles through space is ergodic. That is, for any finite volume of space particles are guaranteed to visit every location in space in a finite amount of time (also referred to as achieving coverage over the space). Third, we make use of periodic boundary conditions to maintain a constant microrobot density without the need for deploying additional agents over time.

The simulation procedure itself requires tracking and evolving the state of three different objects over time: The environmental glucose concentration [G] (x, y, z, t), the environmental insulin concentration [I] (x, y, z, t), and the state of each agent. The state of glucose and insulin concentrations are given by the local levels of each chemical throughout the environment domain, and their joint evolution is described by a simplified set of coupled ordinary differential equations adapted from the PAMERAH model [7]:

$$\begin{cases} \underbrace{\frac{d[\mathrm{GRI}]}{dt}}_{\text{Depletion of reservoir}} = -r_{\mathrm{GRI}} = -\underbrace{\frac{\ln 2[\mathrm{GRI}]}{t_{1/2}}}_{\text{Nonlinear depletion}} \cdot \underbrace{\left\{1+\exp\left[-0.25\left([\mathrm{G}]-[\mathrm{G}]_{\mathrm{th}}\right)\right]\right\}^{-1}}_{\text{Glucose responsiveness}} \\ \underbrace{d[\mathrm{G}]/dt}_{\text{Glucose uptake rate}} = -r^{\mathrm{B}}_{\text{glucose uptake}} \underbrace{\left[6.28+6.52\tanh\left(0.78[\mathrm{I}]-1.98\right)\right]}_{\text{Impact of insulin on glucose uptake rate}} \\ d[\mathrm{I}]/dt = r_{\mathrm{GRI}} - \underbrace{[\mathrm{I}]/9.39}_{\text{Insulin clearance rate}} \end{cases} \tag{S21}$$

At each moment in time, the insulin released by each agent is added locally to the environmental insulin distribution, which then jointly evolves with the glucose distribution according to their coupled dynamics. The variables and their values are listed in Table S2. With only these basic interactions, agents are able to regulate glucose levels via closed-loop insulin release. To simulate the environmental glucose-insulin dynamics we make use of a fourth-order deterministic Runge-Kutta scheme. The state of each agent is given by their location in space, as well as the amount of insulin in their internal reservoir [GRI] (i, t). Since the insulin reservoir levels deplete according to a simple nonlinear function of the local glucose levels, we make use of a computationally efficient Euler scheme to describe the slow-moving insulin reservoir dynamics of each agent. To integrate the stochastic differential equations describing the diffusive trajectories of each microrobotic GRI over time we make use of a stochastic second-order Runge-Kutta scheme. At each time step, all agents move one step, measure their local glucose levels, release insulin to the environment in response, update their insulin reservoir levels, then the glucose and insulin distributions interact and evolve. This main simulation loop is repeated with a time resolution of 0.1s for 100 microrobotic GRIs for the duration of the simulations in the manuscript.

Table S2. Variables and their values used in the spatiotemporal simulations of a GRI swarm.

Symbol	Definition	Unit	Value
[GRI] (i, t)	Concentration of remaining insulin in the reservoir of the i^{th} agent	mU/L	$[\mathrm{GRI}](i,\ t=0)=7.12\times10^{3}$
[I] (x, y, z, t)	Local concentration of insulin at (x, y, z)	mU/L	$[\mathrm{I}](x,y,z,t=0)=0$
[G] (x, y, z, t)	Local concentration of glucose at (x, y, z)	mg/dL	$[\mathrm{G}](x,y,z,t=0)\in[230,425]$
r_{GRI}	Rate of insulin release	mU/L/min	-
$r^{B}_{\text{glucose uptake}}$	Base rate of glucose uptake	mU/L/min	1.67
$[\mathrm{G}]_{\mathrm{th}}$	Threshold glucose concentration beyond which insulin is released from the reservoir	mg/dL	225
$t_{1/2}$	Reservoir half life	min	360

2 Supplementary Simulations and Analyses

2.1 Nondimensionalized Model of the 1D Memristor Array

Equation S9 governing the 1D memristor array dynamics can be nondimensionalized as follows:

$$\begin{cases} \dfrac{\partial^2 U}{\partial X^2} = \dfrac{U}{w + [1-w]\cdot\alpha} \\ \dfrac{\partial w}{\partial \tau} = \dfrac{U}{w + [1-w]\cdot\alpha} f_{\mathrm{W}} \end{cases} \tag{S22}$$

where $U(X,\tau) = V(x,t)/V_0$, $X = x\sqrt{R_{\mathrm{R}}(t)/R_{\mathrm{ON}}}$, and $\tau = \left(\mu V_0 / L^2\right)t$ are the dimensionless voltage, location, and time, respectively. U and w are both functions of X and τ, and f_{W} denotes the window function of choice. While we used the dimensional simulations in the Main Text to put the actual electrical element parameters in quantitative perspective, the nondimensionalized model in Equation S22 is easier to analyze and particularly helpful in elucidating the deceleration of an array's ON/OFF boundary at a later time. This phenomenon is visible in Figure 1H of the Main Text and discussed in the Discussions section. If we adopt the quasi-steady state approximation and view w as just a function of X for a given time, Equation S22 becomes an ODE of U:

$$\frac{d^2 U}{dX^2} = \frac{1}{w + [1-w]\cdot\alpha} U \tag{S23}$$

Qualitatively, as the voltage polarity stays the same along the memristor array for a fixed source and $|U(X>0)| < |U(0)|$, magnitude of the voltage unavoidably diminishes further away. Qualitatively, if we make the easy but numerically inaccurate assumption that w is uniform, the voltage profile can be analytically solved and U exponentially decays along X. In reality, the memristors further away remain ON while the ones closer to the voltage source start to be switched OFF not too long after $t = 0$. In the region already affected by the voltage source, then, the distribution of w is approximately linear. Solution of this ODE suggests that the voltage distribution along the array takes the form of a modified Bessel function, which also decays with X. Substituting the decaying U into the second PDE in Equation S22 reveals how memristors further away switch slower. As described in the Discussions section of the Main Text, the decelerating sequential switching further away from the voltage source limits the usable length of a continuous memristor array: Beyond a certain x, the sluggish propagation of the ON/OFF boundary can no longer be detected due to the measuring probe's spatial resolution. The limitation is fortunately countered by the increasing availability of high-resolution characterization platforms like conductive atomic force microscopy (cAFM), which presents a thousand-fold enhancement compared to traditional micromanipulator probes. In a discrete memristor array, on the other hand, resistive tuning may be used (§1.1).

2.2 2D Memristor Array with a Transmission Line

In addition to grafting memristor array branches onto a memristor array trunk, we studied the feasibility of voltage propagation by a transmission line due to its mathematical simplicity and uniform speed. The transmission line, or telegraph line, which supplies voltage to the branches in this design, is governed by the elegant telegrapher's equations by Heaviside:

$$\begin{cases} \dfrac{\partial V(x,t)}{\partial x} = -L_{\mathrm{T}} \dfrac{\partial I(x,t)}{\partial t} - R_{\mathrm{T}} I(x,t) \\ \dfrac{\partial I(x,t)}{\partial x} = -C_{\mathrm{T}} \dfrac{\partial V(x,t)}{\partial t} - G_{\mathrm{T}} V(x,t) \end{cases} \tag{S24}$$

Coupling Equation S24 with Equation S9 which describes the 1D memristor array dynamics, we arrive at the governing system of PDEs:

$$\begin{cases} \dfrac{\partial V(x,0,t)}{\partial x} = -L_{\mathrm{T}} \dfrac{\partial I(x,0,t)}{\partial t} - R_{\mathrm{T}} I(x,0,t) \\ \dfrac{\partial I(x,0,t)}{\partial x} = -C_{\mathrm{T}} \dfrac{\partial V(x,0,t)}{\partial t} - G_{\mathrm{T}} V(x,0,t) + \left(\dfrac{\partial V(x,y,t)}{R_y(t) \partial y} \right)_{y=0+} \\ \dfrac{\partial^2}{\partial y^2} V(x,y,t) = \dfrac{R_y(t)}{R_{\mathrm{ON}}} \dfrac{V(x,y,t)}{w(x,y,t) + \left[1 - w(x,y,t)\right]\alpha} \\ \dfrac{\partial}{\partial t} w(x,y,t) = \mu_y \dfrac{V(x,y,t)}{w(x,y,t) + \left[1 - w(x,y,t)\right]\alpha} f_{\mathrm{W}} \end{cases} \tag{S25}$$

V (x, 0, t) and I (x, 0, t) are respectively the voltage and current on the transmission line. L_{T}, C_{T}, G_{T}, and R_{T} are parameters describing the telegraph line characteristics, among which the latter two determine the energy loss. As with previous simulations, $w \in [0, 1]$ represents the instantaneous state variable that determines the local memristance at (x, y): $M(x,y,t) = \left\{ w(x,y,t) + \left[1 - w(x,y,t)\right]\alpha \right\} R_{\mathrm{ON}}$ where α is the switching range and R_{ON} is the ON-state memristance with the unit of $\Omega \cdot \mu\mathrm{m}^2$ (*cf.* $R_{\mathrm{ON},y}$ in Table S1).

Again, we simulated the set of PDEs in Equation S25 with the method of lines. We approximated the spatial derivatives via finite differencing of accuracy 2, transforming the PDEs into a system of ODEs:

$$\begin{cases} \left.\begin{aligned} \dfrac{\partial I_{i,1}(t)}{\partial t} &= -\dfrac{1}{2L_{\mathrm{T}}\Delta x}\left[V_{i+1,1}(t) - V_{i-1,1}(t)\right] - \dfrac{R_{\mathrm{T}}}{L_{\mathrm{T}}} I_{i,1}(t) \\ \dfrac{\partial V_{i,1}(t)}{\partial t} &= -\dfrac{1}{2C_{\mathrm{T}}\Delta x}\left[I_{i+1,1}(t) - I_{i-1,1}(t)\right] - \dfrac{G_{\mathrm{T}}}{C_{\mathrm{T}}} V_{i,1}(t) + \dfrac{-3V_{i,1}(t) + 4V_{i,2}(t) - V_{i,3}(t)}{2C_{\mathrm{T}} R_y(t) \Delta y} \\ \dfrac{\partial w_{i,j>1}(t)}{\partial t} &= \mu \gamma_{i,j} R_{\mathrm{ON}} V_{i,j}(t) f_{\mathrm{W}} \end{aligned} \right\} \text{ODEs} \\ V_{i,j-1}(t) - \left[2 + \gamma_{i,j} R_y(t) (\Delta y)^2\right] V_{i,j}(t) + V_{i,j+1}(t) = 0 \end{cases} \tag{S26}$$

where $\gamma_{i,j}(t) = \left\{ R_{\mathrm{ON}} \left[w_{i,j} + \left(1 - w_{i,j}\right)\alpha \right] \right\}^{-1}$. The (i, j) tuple refers to the location at the i^{th} node along the x axis and j^{th} node along y on the mesh with a resolution of Δx by Δy. The voltage pulse V_0 (t) is initiated at (1, 1). As with §1.3, the linear algebraic equation in Equation S26 can be solved efficiently with the backslash operator:

$$\mathbf{V} = \begin{pmatrix} V_{1,1} \\ \vdots \\ V_{N_i,1} \\ \hline V_{1,2} \\ \vdots \\ V_{i,j} \\ \vdots \\ V_{N_i,N_j-1} \\ \hline V_{1,N_j} \\ \vdots \\ V_{N_i,N_j} \end{pmatrix} = \begin{pmatrix} 1 & & & & & & & & & & & & \\ & \ddots & & & & & & & & & & & \\ & & 1 & & & & & & & & & & \\ \hline 1 & & \beta_{1,2} & & 1 & & & & & & & & \\ & \ddots & & \ddots & & & \ddots & & & & & & \\ & & 1 & & \beta_{i,j} & & & & 1 & & & & \\ & & & \ddots & & \ddots & & & & \ddots & & & \\ & & & & 1 & & \beta_{N_i,N_j-1} & & & & 1 & & \\ \hline & & & & & & -\tfrac{1}{2} & -2 & \tfrac{3}{2} & & & & \\ & & & & & & & \ddots & \ddots & \ddots & & & \\ & & & & & & & & -\tfrac{1}{2} & -2 & \tfrac{3}{2} & & \end{pmatrix}^{-1} \begin{pmatrix} V_{1,1} \\ \vdots \\ V_{N_i,1} \\ \hline 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \hline 0 \\ \vdots \\ 0 \end{pmatrix} \quad \text{(S27)}$$

where $\beta_{i,j}(t) = -2 - \gamma_{i,j} R_y(t)(\Delta y)^2$ and any $V_{i,1}(t)$ is updated from the differential equations. The ODEs are solved as a standard initial value problem with the initial condition of $w = 0.999$ at all locations. For each branch to receive only temporary power restricted to a short time window, a voltage pulse is applied at (0, 0) and passed down the trunk along x. For the simulations in Figure S5, we used no-flux boundary conditions at large x and y, as well as a transient voltage input at (0, 0) in the form of a Gaussian pulse with a peak voltage of 5V and a standard deviation of 10min. This 2D memristor array powered by a transmission line works well for both a sinusoidal and a realistic analyte profile, evident from the agreement between the profiles and the ON/OFF fronts. Furthermore, the memristor states are not distorted as in §1.3 thanks to the constant propagation speed along the transmission line, and hence no calibration is required. Nonetheless, we note that the transmission line 2D array is more of an analytical tool than an engineering design, considering the challenges involved in patterning micron-scale inductors as well as the large $L_T C_T$ product required to slow down the trunk transmission. In comparison, a trunk in the form of a memristor array – the design we eventually chose – can be designed with experimentally relevant parameters and fabricated by facile patterning of materials.

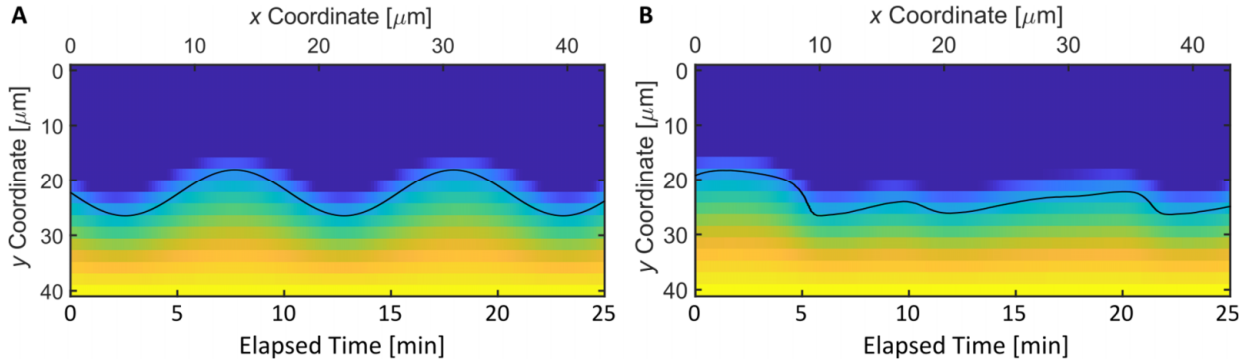

Figure S5. Simulated final memristor states of a 2D memristor array powered by a transmission line. The ON/OFF fronts conform well to both the (A) sinusoidal and (B) realistic temporal analyte profile shown as black curves. The transmission line (trunk) parameters used are: $C_T = 0.18$F/μm, $L_T = 0.6$MH/μm, $R_T = 0$,

and $G_T = 0$, i.e., a "lossless" line. The branch parameters are: $\mu_y = 0.05 s^{-1}V^{-1}$, $\alpha = 10$, and $R_{ON} = 50k\Omega$. The realistic profile is extracted from Ref. [11].

2.3 Further Analyses of Application-Specific Temporal Resolution Constraints

We described in the *Discussion* section of the Main Text how the sequential switching in a memristor array slows down over time. This feature leads to a high temporal resolution attainable earlier on, which gradually worsens as time elapses. For example, Figure 1H of the Main Text shows that the same spacing among the ON/OFF boundaries corresponds to a larger time window at a larger t. In this section, we further analyze how memristor arrays can be designed and measured such that even the worst temporal resolution towards the end of the robot's journey can suit the intended application. Note that the minimum resolution an application requires typically scales with the residence time; A task over the course of several hours typically do not require each second to be accounted for.

The 1st row of Table S3 shows that the memristor array studied in Figure 1H offers sufficient temporal resolution to pinpoint, within a 6-stage microfluidic chip reactor, which stage corresponds to a recorded event. However, the same memristor array fails to provide enough temporal resolution in an industrial flow reactor with a tenfold residence time (2nd row) if it is still read with a micrometer-resolution probe station.

We present two solutions to enhance the resolution. First, as thoroughly discussed in *Discussion*, an array's temporal resolution at each timepoint is ultimately limited by the spatial measurement resolution of the readout platform. As a result, enhancing the spatial accuracy of circuit readout effectively helps tell apart ON/OFF boundaries corresponding to different times, even when t is large (3rd vs. 2nd row, also Figure S6). This can be achieved by, for instance, replacing the probe station with cAFM, which is an alternative tool for memristor characterization [12,13]. Alternatively, the memristor array parameters may be designed differently to suit the temporal resolution this specific application targets (4th vs. 2nd row). Similarly, we designed a memristor array to satisfy the requirements of a canine GI tract sensor as well (5th row). While the final (worst-case) temporal resolution is ±28.5min at the end of 8 hours, the resolution is at least ±7.2min within the first 2 hours and ±3.6min within the first hour, as expected.

Table S3. Analyses of memristor arrays designed to satisfy the temporal resolution requirements of several specific application scenarios.

Example Application	Ref.	Typical Residence Time	Memristor Array Parameters			Spatial Measurement Resolution	Worst-Case Temporal Resolution[a]	
			μ/L^2 [$s^{-1}V^{-1}$]	R_R [kΩ/μm]	α		Target	Simulated
6-stage microfluidic synthesis	[14]	7min	1[b]	1.75[bc]	100[b]	±1μm (probe station)	±35s[d]	±27.6s
Industrial flow reactor	[15]	70min	1	1.75	100	±1μm (probe station)	±210s	±251.5s
Industrial flow reactor	[15]	70min	1	1.75	100	±1nm (cAFM)	±210s	±0.25s
Industrial flow reactor	[15]	70min	1	1[c]	100	±1μm (probe station)	±210s	±190.1s
Canine gastric emptying	[16]	8hr	1	1.75	100	±1μm (probe station)	±30min	±28.5min

[a] Because the temporal resolution is varying over time, we consider here the minimum (worst-case) temporal resolution towards the end of the residence time and compare it to the minimum requirement (the target resolution) specific to each intended application. The temporal resolution is much higher earlier on in all cases.

[b] This set of memristor array parameters is identical to that simulated in Figure 1H of the Main Text as well as Figure S6 below.

[c] Both simulated parameter values of R_R = 1kΩ/μm and 1.75kΩ/μm are consistent with material properties reported in the literature. For example, the sheet resistance of a graphene film ranges from 2×10^2 to 6×10^5Ω/□ depending on the method of

synthesis and chemical modification [17,18]. This corresponds to an R_R between 0.4 and 1200kΩ/μm assuming a 2μm-wide 1D memristor array.

[d] The target temporal resolution of ±35s is chosen such that even in the worst-case scenario, one stage out of the 6-stage microfluidic platform can be identified which corresponds to the sensing event picked up by the memristor array.

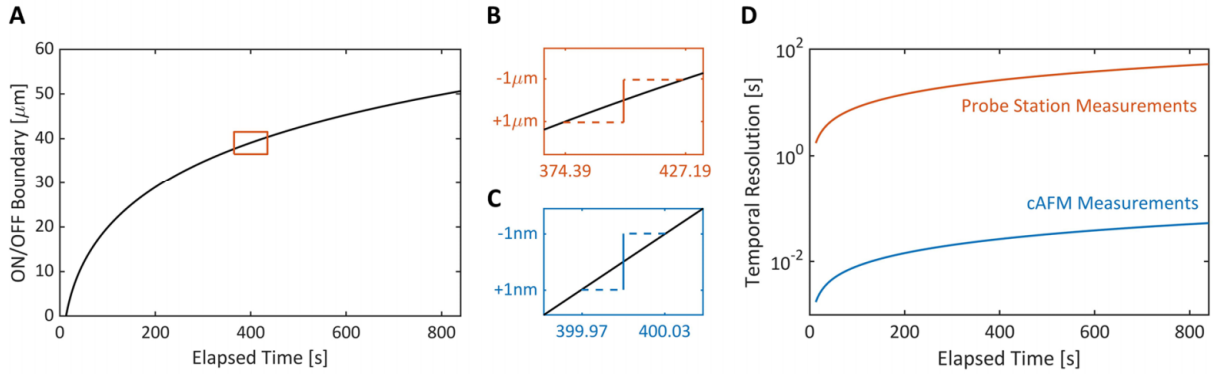

Figure S6. (A) The simulated relation between the ON/OFF boundary location along a continuous 1D memristor array and time elapsed shows that the sequential switching slows down gradually. This relation is calculated the same way as the calibration curve used for correcting a 2D memristor array as discussed in §1.3. The red box corresponds to the magnified region shown in (B). (B) The spatial resolution of a traditional probe station measurement platform, which is approximately 1μm as marked by the vertical interval, translates to a temporal resolution of ±26.3s at t = 400s shown as the horizontal dashed lines. (C) In comparison, the nanometer-range spatial resolution of a cAFM drastically refines the temporal resolution to ±0.03s. (D) The temporal resolution as a function of time and the spatial resolution of the measurement platform.

2.4 Overview of Additive Fabrication of Memristor Arrays

The design of an array of memristors connected in parallel is inspired by previous findings that matrices of memristors can be facilely stamped out *en masse* from materials additively stacked together, a technology termed *autoperforation* [4]. Specifically, as seen in Figure S7A, user-defined inks are printed onto a 2D sheet of graphene, MoS_2, or hexagonal boron nitride in the form of a dot array. The ink can be made memristive by, for instance, compositing polystyrene beads with black phosphorus or MoS_2 nanoflakes. Subsequently, a second 2D sheet is transferred on top, thereby forming a sandwich stack. Finally, the said stack is lifted off in solution, during which individual microparticles are "stamped out" and collected, the size of each dictated by the size of the printed ink disc (20 to 1000μm in diameter). The height of each microparticle is typically less than a micrometer, thus making the particles effectively planar.

Mechanistically, the particles are cut out from a continuous materials stack by strain-guided fracture. Seen in Figure S7B, the strain field imposed by the printed ink disc ensures any crack in the materials propagate along the disc's periphery, acting like a pair of invisible microscopic scissors. The autoperforation technique does not require the ink spots to be printed. Alternative micrometer-scale additive platforms like stamping, coating, and colloidal self-assembly are applicable as well.

Figure S7C presents the full set of data from which Figure 1F in the Main Text is adapted. Parts of the 3-by-5 grid on a single memristive microparticle, defined by the measurement probe, can be selectively switched ON by applying a writing voltage to each spot. The letters "M", "I", and "T" spelt out on the same autoperforated particle demonstrate that each particle is equivalent to an array of cross-plane memristors wired in parallel. Of note, the technology has been shown to be highly customizable, as the microparticle's

size, geometry, and ink compositions are all tunable. More importantly, the fabrication process allows for facile surface chemical modifications and patterning as well [19]. For the current work, this enables spatial tuning of surface resistivity (R_R) as well as definition of memristor array branches.

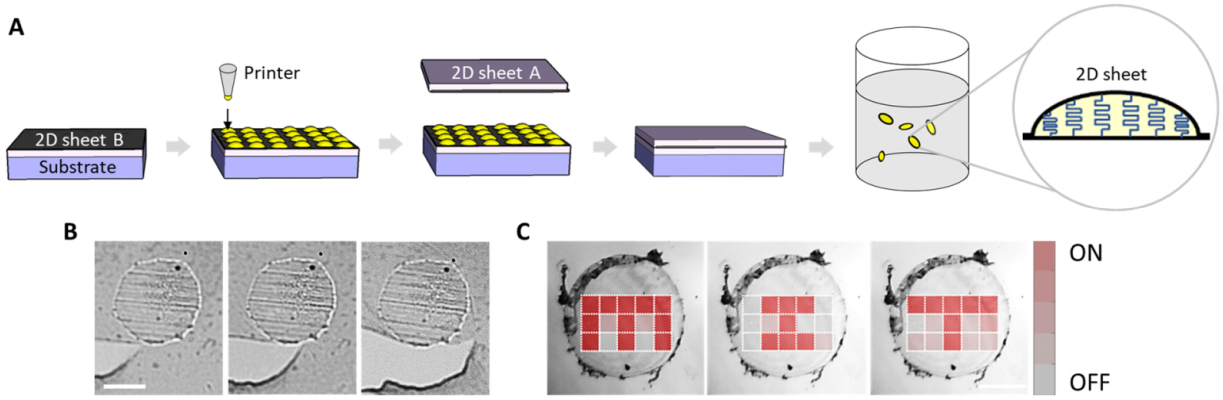

Figure S7. (A) Overview of the autoperforation process for fabricating micrometer-scale ink particles encapsulated between 2D sheets. (B) Microscopic process of the fracture-assisted lift-off step during which microparticles are stamped out of the continuous stack of materials. Scale bar, 100µm. (C) Selective switching experiments demonstrated that each memristive microparticle is effectively an 2D array of memristors connected in parallel. Scale bar, 100µm. All panels are adapted from Ref. [4].

3 Supplementary Figures

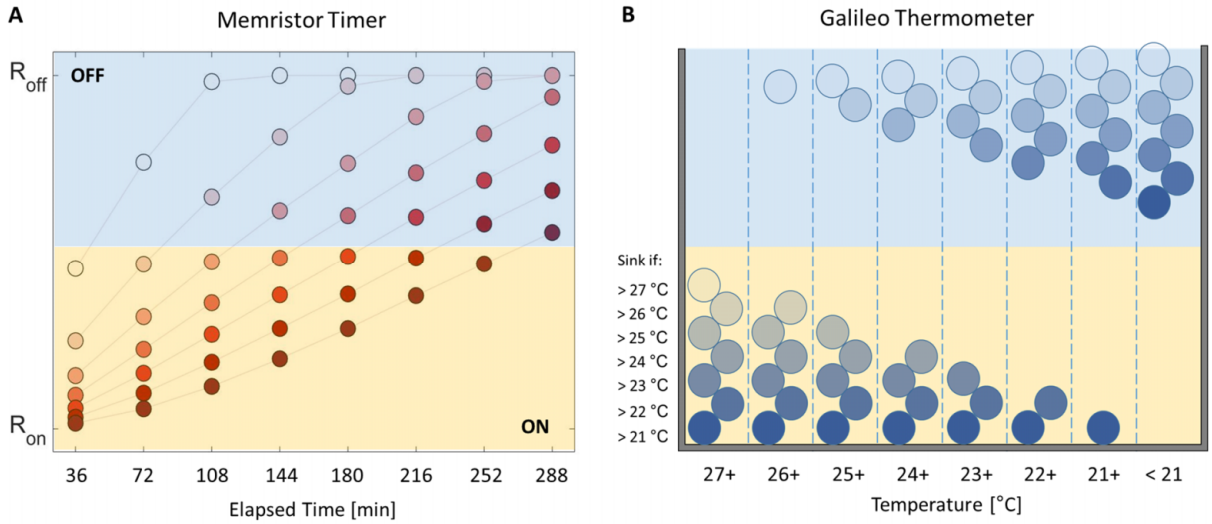

Figure S8. The 1D memristor array (A) records time exactly how a Galileo thermometer (B) records temperature. The simulated data in panel (A) are identical to those in Figure 1G of the Main Text but are visualized differently to show the parallelism.

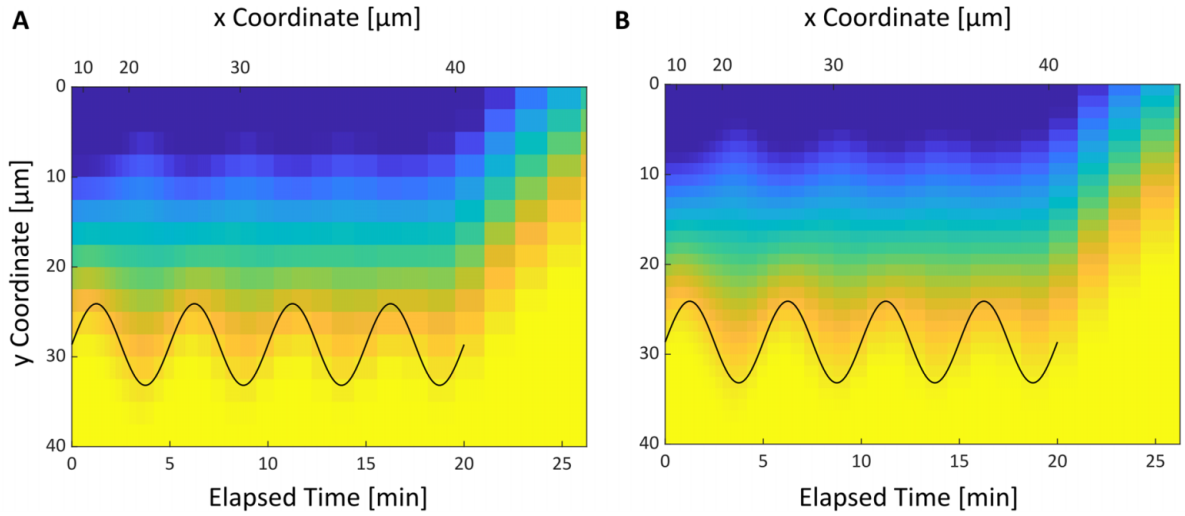

Figure S9. States of a 2D memristor array at t = 20min simulated with a (A) coarse and (B) fine degree of branch discretization with a resolution (Δy) of 0.25 and 0.125µm, respectively. The simulations are comparable.

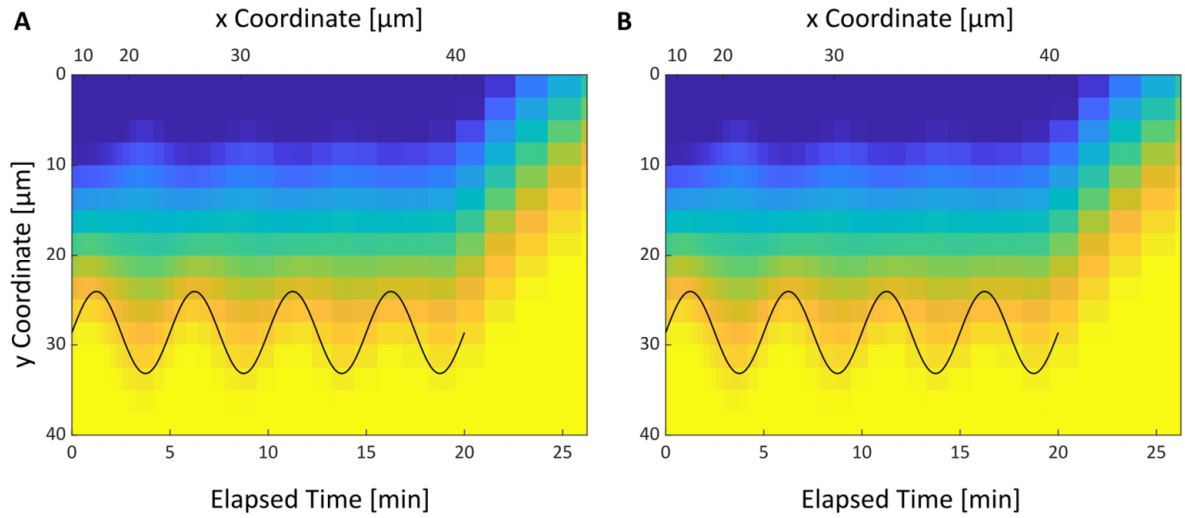

Figure S10. States of a 2D memristor array at t = 20min simulated with a simulation boundary at (A) x = 100µm and (B) x = 80µm. The comparable results suggest that a dimension of 80µm along x is sufficient for the time span.

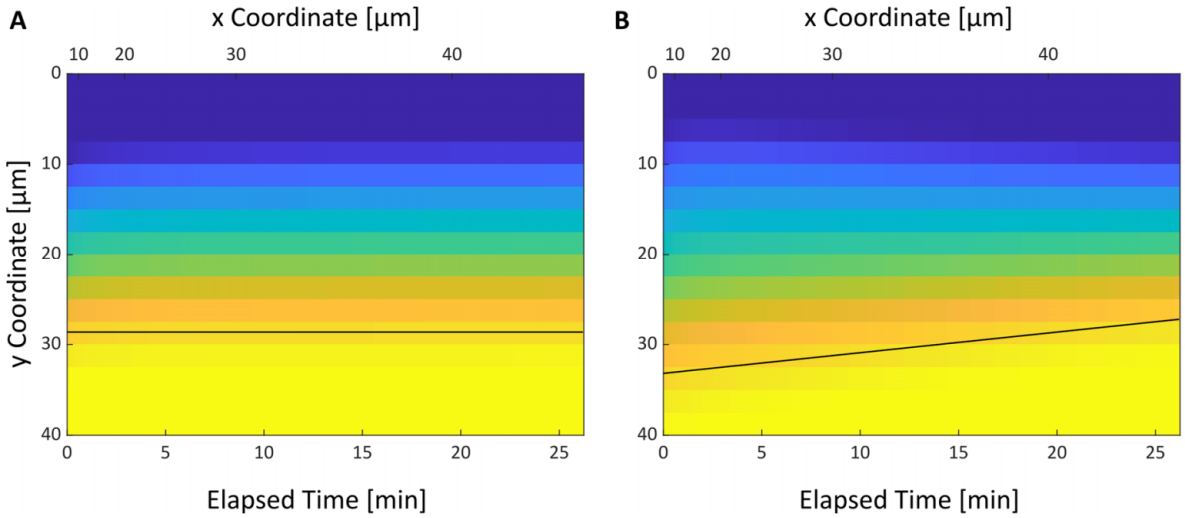

Figure S11. States of a 2D memristor array at t = 40min responding to (A) a constant and (B) a linearly ramping profile. Note the absence of distortion across all branches in (A). The agreement between the linear profile in (B) and the linear ON/OFF front demonstrates the use of the calibration process.

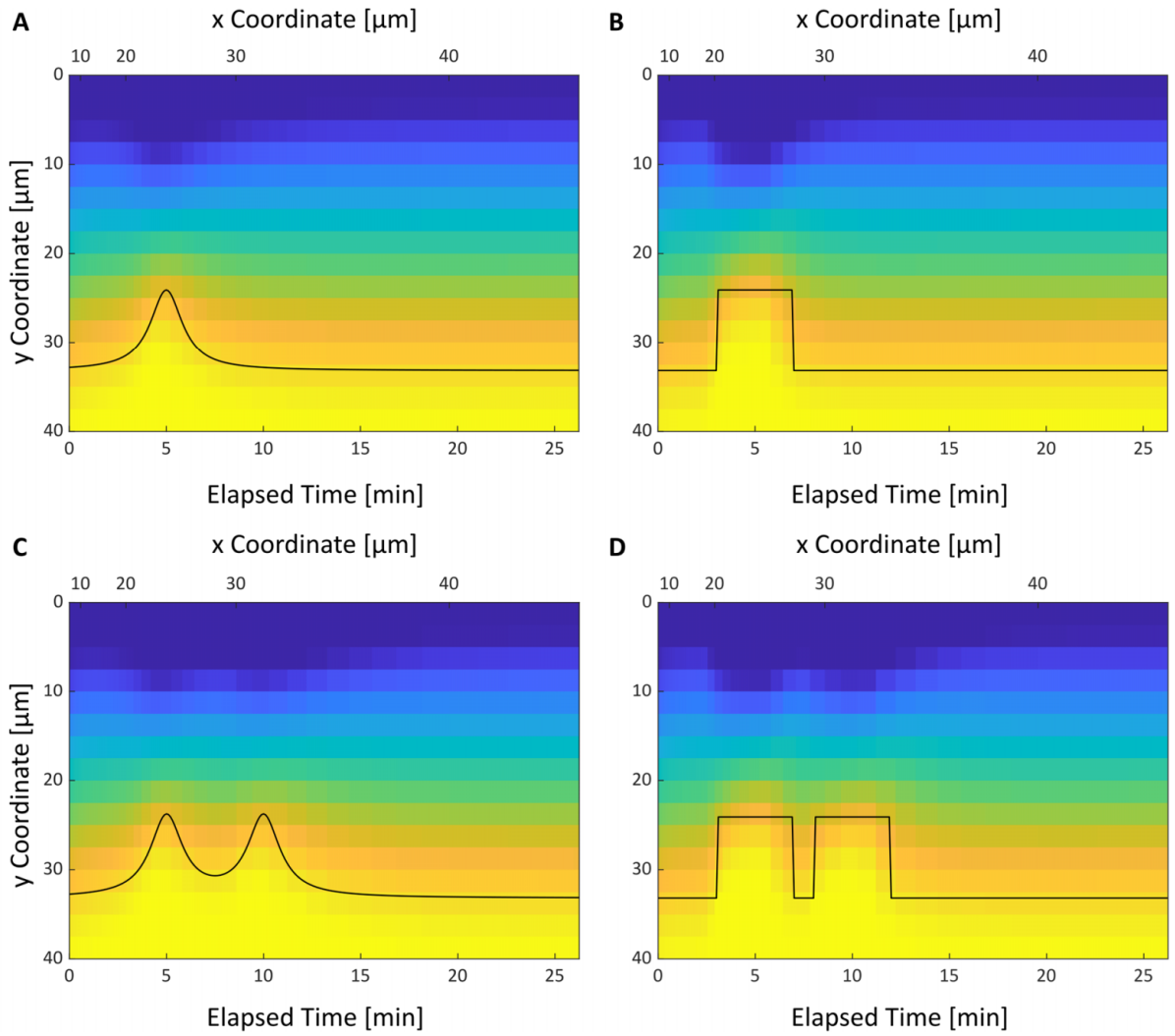

Figure S12. States of a 2D memristor array at t = 40min responding to analyte concentration profiles with (A) a smooth pulse, (B) an abrupt pulse, (C) two closely spaced smooth pulses, and (D) two closely spaced abrupt pulses. The smooth profiles (A, C) are modeled following Ref. [20] as $R_y(t) = 5 + 10 / [1 + (t - 5)^2]$ and $R_y(t) = 5 + 10 / [1 + (t - 5)^2] + 10 / [1 + (t - 10)^2]$, respectively. The square profiles (B, D) respectively follow $R_y(t) = \{15, t \in [3, 7]; 5, \text{otherwise}\}$ and $R_y(t) = \{15, t \in [3, 7] \cup [8, 12]; 5, \text{otherwise}\}$.

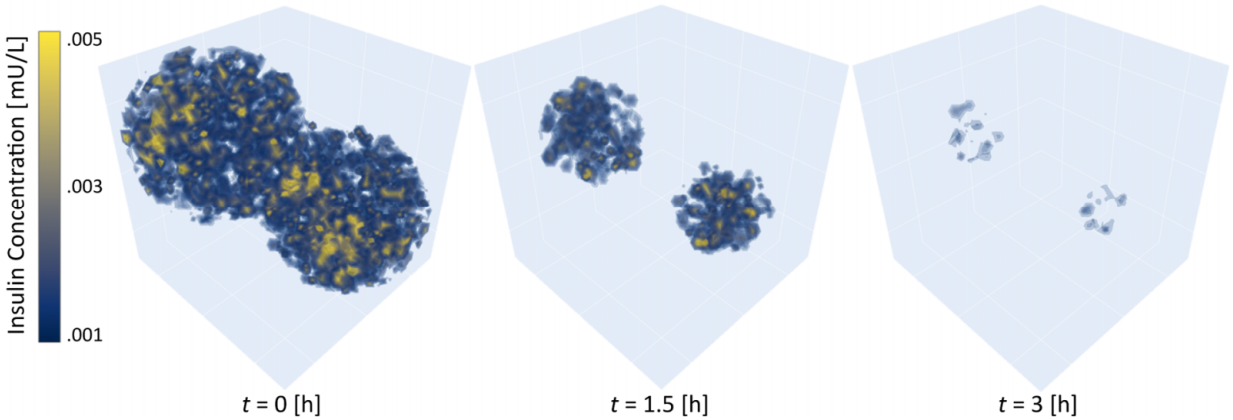

Figure S13. Progression of the spatial distribution of insulin released over time by a swarm of 100 microrobotic GRIs with a $[G]_{th}$ of 225 mg/dL in response to the glucose distribution shown in Fig. 4E of the Main Text. When the microrobotic agents sense a local glucose level above the threshold they release insulin in accordance with the measured magnitude. This closed-loop insulin release in concert with their diffusive motion allows the robots to regulate glucose levels over the spatial domain. As the agents successfully reduce glucose levels, they release less insulin over time.

4 References

[1] D. B. Strukov, G. S. Snider, D. R. Stewart, R. S. Williams, *Nature* **2008**, *453*, 80.

[2] Z. Biolek, D. Biolek, V. Biolková, *Radioengineering* **2009**, *18*, 210.

[3] T. Prodromakis, B. P. Peh, C. Papavassiliou, C. Toumazou, *IEEE Trans. Electron Devices* **2011**, *58*, 3099.

[4] P. Liu, A. T. Liu, D. Kozawa, J. Dong, J. F. Yang, V. B. Koman, M. Saccone, S. Wang, Y. Son, M. H. Wong, M. S. Strano, *Nat. Mater.* **2018**, *17*, 1005.

[5] L. Chua, V. Sbitnev, H. Kim, *Int. J. Bifurc. Chaos* **2012**, *22*, 1230011.

[6] L. Chua, *Nanotechnology* **2013**, *24*, DOI 10.1088/0957-4484/24/38/383001.

[7] J. F. Yang, X. Gong, N. A. Bakh, K. Carr, N. F. B. Phillips, F. Ismail-Beigi, M. A. Weiss, M. S. Strano, *Diabetes* **2020**, *69*, 1815.

[8] J. Yu, J. Wang, Y. Zhang, G. Chen, W. Mao, Y. Ye, A. R. Kahkoska, J. B. Buse, R. Langer, Z. Gu, *Nat. Biomed. Eng.* **2020**, 1.

[9] M. Maheandiran, S. Mylvaganam, C. Wu, Y. El-Hayek, S. Sugumar, L. Hazrati, M. del Campo, A. Giacca, L. Zhang, P. L. Carlen, *PLoS One* **2013**, *8*, DOI 10.1371/journal.pone.0083168.

[10] V. B. Koman, P. Liu, D. Kozawa, A. T. Liu, A. L. Cottrill, Y. Son, J. A. Lebron, M. S. Strano, *Nat. Nanotechnol.* **2018**, *13*, 819.

[11] The MathWorks Inc., "Non-adiabatic continuous stirred tank reactor: MATLAB file modeling with simulations in Simulink®," can be found under https://www.mathworks.com/help/ident/ug/non-adiabatic-continuous-stirred-tank-reactor-matlab-file-modeling-with-simulations-in-simulink.html#d122e43522, **n.d.**

[12] F. Fan, B. Zhang, Y. Cao, X. Yang, J. Gu, Y. Chen, *Nanoscale* **2017**, *9*, 10610.

[13] S. Brivio, G. Tallarida, D. Perego, S. Franz, D. Deleruyelle, C. Muller, S. Spiga, *Appl. Phys. Lett.* **2012**, *101*, 223510.

[14] J. Baek, Y. Shen, I. Lignos, M. G. Bawendi, K. F. Jensen, *Angew. Chemie Int. Ed.* **2018**, *57*, 10915.

[15] K. P. Cole, J. M. Groh, M. D. Johnson, C. L. Burcham, B. M. Campbell, W. D. Diseroad, M. R. Heller, J. R. Howell, N. J. Kallman, T. M. Koenig, S. A. May, R. D. Miller, D. Mitchell, D. P. Myers, S. S. Myers, J. L. Phillips, C. S. Polster, T. D. White, J. Cashman, D. Hurley, R. Moylan, P. Sheehan, R. D. Spencer, K. Desmond, P. Desmond, O. Gowran, *Science (80-.).* **2017**, *356*, 1144.

[16] C. S. Boillat, F. P. Gaschen, G. L. Hosgood, *Am. J. Vet. Res.* **2010**, *71*, 898.

[17] G. Jo, M. Choe, S. Lee, W. Park, Y. H. Kahng, T. Lee, *Nanotechnology* **2012**, *23*, DOI 10.1088/0957-4484/23/11/112001.

[18] P. Liu, A. L. Cottrill, D. Kozawa, V. B. Koman, D. Parviz, A. T. Liu, J. F. Yang, T. Q. Tran, M. H. Wong, S. Wang, M. S. Strano, *Nano Today* **2018**, *21*, 18.

[19] A. T. Liu, J. F. Yang, L. N. LeMar, G. Zhang, A. Pervan, T. D. Murphey, M. S. Strano, *Faraday Discuss.* **2021**, *227*, 213.

[20] N. Ghods, M. Krstic, *J. Dyn. Syst. Meas. Control* **2011**, *133*, DOI 10.1115/1.4003639.