Supporting Information for

Programmable Lipid Bilayer Tension–Control Apparatus for Quantitative Mechanobiology

Yuka Matsuki[1,3], Masayuki Iwamoto[2,3], Takahisa Maki[2,3], Masako Takashima[2], Toshiyuki Yoshida[4], Shigetoshi Oiki[5]*

1 Department of Anesthesiology and Reanimatology, Faculty of Medical Sciences., University of Fukui, Yoshida-gun, Fukui, 910-1193, Japan
2 Department of Molecular Neuroscience, Faculty of Medical Sciences., University of Fukui, Yoshida-gun, Fukui, 910-1193, Japan
3 Life Science Innovation Center, University of Fukui, Fukui-shi, Fukui, 910-8507, Japan
4 Division of Engineering Information Science, Faculty of Engineering, University of Fukui, Fukui-shi, Fukui, 910-8507, Japan
5 Biomedical Imaging Research Center, University of Fukui, Yoshida-gun, Fukui, 910-1193, Japan

*Correspondence
oiki@u-fukui.ac.jp
Biomedical Imaging Research Center, University of Fukui, Fukui 910-1193, Japan

The methodological procedures and mathematical handling of the feedback control are described in detail in the Supporting Information.

Bilayer tension-clamp system

Bubble pressure control

Figure S1 shows two pneumatic manipulators for manual operation (Narishige), on which a stepping motor (PKP523N12B, ORIENTAL MOTOR Co., Ltd, Japan) was mounted. A software switch is set to choose either manual or automatic control, which allows immediate switching-on manipulation.

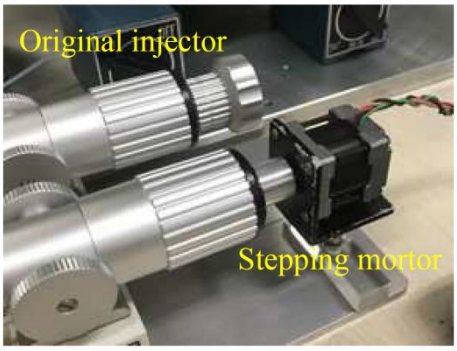

Figure S1. Computer-controllable pneumatic injector. The stepping motor (PKP523N12B manufactured by ORIENTAL MORTOR Co., Ltd, Japan) was attached to the rotation axis of the original pneumatic microinjector (IM-11-2 manufactured by NARISHIGE Co., Ltd, Japan). The power lines of the motor were carefully shielded to avoid electromagnetic interference for a channel current measurement.

Image analysis

A bubble image provides the geometry metrics prerequisite for tension evaluation involving bubble radii and contact angles. For the captured bubble images, we first detected the bubble boundaries. To avoid disturbances from the blurred pipette contour, two vertical lines were manually located at the bubble-pipette borderline (Fig. S2) on a graphical user interface (GUI). The image is then trimmed into a region limited by two vertical lines. The upper and lower boundaries of the bubbles were detected in the region using a suitable edge-detection operator[1]. They were further separated into left and right boundaries at the corner points (Fig. S2A) based on corner detection operations in fundamental image processing techniques[1].

The shape of the bubbles was well modeled with a slanted ellipse, and their geometric parameters were obtained by fitting each of the boundaries with the parametric equation of an ellipse:

$$F(x,y) = a\,x^2 + b\,xy + c\,y^2 + d\,x + e\,y = 1 \qquad \text{Eq. S1}$$

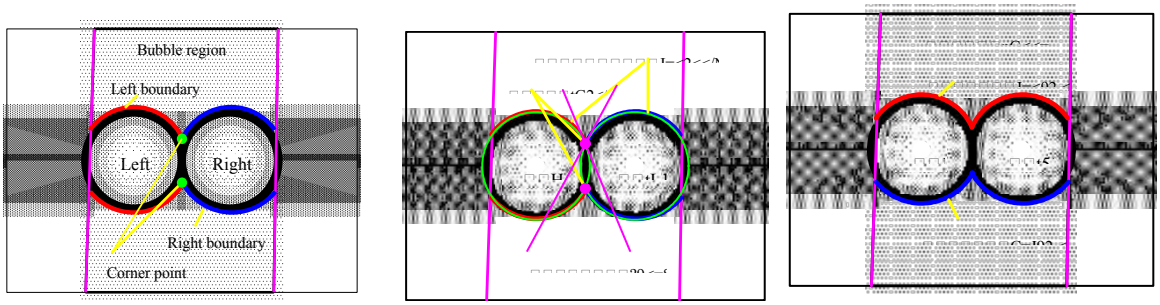

Figure S2. Extraction of the upper and lower boundaries of the bubbles. **A.** The vertical lines were placed to separate the bubbles from the blurred pipettes. The upper and lower boundaries were readily detected using an edge-detection operator. **B.** Separation of the upper and the lower boundaries into the left and right at the corner points. **C.** Approximation of the left and right boundaries by the ellipse $F_z(x,y) = 1$, from which the geometric parameters can be extracted.

The coefficients *a* - *e* were obtained by the least square technique that minimizes

$$E = \sum_n^{N-1} \left(a\, x_n^2 + b\, x_n\, y_n + c\, y_n^2 + d\, x_n + e\, y_n - 1 \right)^2 \qquad \text{Eq. S2}$$

with respect to parameters *a* - *e*, where (x_n, y_n) (n = 0, 1, …, N) represents the coordinates of each pixel on the bubble boundary. The geometric parameters of the bubbles were readily calculated from the coefficients *a* - *e*, yielding the center coordinate (c_x, c_y), the principal radii R_l and R_r, and the tilt angle ϕ, where ϕ is the angle between a principal radius and the horizontal line.

The contact angles θ were necessary to calculate γ_{bil} using Eqs. (1) – (5), which can be calculated by first solving $F_L(x,y) = 1$ and $F_R(x,y) = 1$ simultaneously to find the upper and lower contact points followed by calculating the tangential lines as shown in Fig. S2. The Newton-Raphson technique was employed to determine the coordinates of the contact points.

Software control unit

A block diagram of the control system is shown in Fig. 5, including three building blocks and internal signals:

r(*t*): the reference (clamping target) value
y(*t*): the measured tension
e(*t*): the difference tension
u(*t*): the control block output
ω(*t*): the rotation angle of the injector's stepping motor
p(*t*): the bubble pressure
r(*t*): the bubble radius
ϑ(*t*): the contact angle

A simple proportional-integral-differential (PID) control technique was employed in the control block [38]. Control block outputs, *u*(t),

$$u(t) = k_p e(t) + k_d \frac{de(t)}{dt} \qquad \text{Eq. S3}$$

for the difference $e(t)$, where k_p and k_d are proportionality constants. The feedback signal into the pressure operating unit is the pulse count for the rotating axis $u(t)$. Thereafter, the motor drive yields the rotation angle, $\omega(t)$, for the stepping motor. As the motor maintains its angle after rotation, the motor block acts as an integrator.

$$\omega(t) \propto \int_0^t u(t)dt \qquad \text{Eq. S4}$$

The fine pressure gauge continuously monitored changes in the bubble pressure p(t). Simultaneously, the image-analysis unit retrieved the bubble radii ($R_{avg}(t)$) that were fed into the size-clamp system.

Taking the Laplace transformation of each internal value, the transfer function $H(s)$ from the input reference $r(t)$ to the output $y(t)$ is given by:

$$H(s) = \frac{Y(s)}{R(s)} = \frac{\frac{U(s)}{E(s)} \cdot \frac{\Omega(s)}{U(s)} \cdot \frac{Y(s)}{\Omega(s)}}{1+\frac{U(s)}{E(s)} \cdot \frac{\Omega(s)}{U(s)} \cdot \frac{Y(s)}{\Omega(s)}} \qquad \text{Eq. S5}$$

where $Y(s)/\Omega(s)$ denotes the transfer function of the bubble block: the ratio of the response $y(t)$ (= $R_{avg}(t)$) against a variation of the injector pressure governed by $\omega(t)$ in the transformed domain. We have empirically confirmed that the transfer function $Y(s)/\Omega(s)$ can be well modeled as a first-order lag system and is represented as

$$\frac{Y(s)}{\Omega(s)} \propto \frac{1}{s\tau+1} \qquad \text{Eq. S6}$$

where τ is a time constant of the response determined by the bubble system's chemical and physical features.

By substituting Eq. S6, and the Laplace transforms of Eqs. S3 and S4 into Eq. S5, the transfer function of the control system is given by:

$$H(s) = \frac{Y(s)}{R(s)} = \frac{k_d\, s + k_p}{\tau\, s^2 + (k_d+1)\, s + k_p} \qquad \text{Eq. S7}$$

which is a second-order transfer function. Application of the final value theorem,

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} s\, H(s)\, R(s) \qquad \text{Eq. S8}$$

where $R(s) = 1/s$ (unit-step function) in Eq. S8 yields $y(\infty) = 1$. This indicates that the control system produces no steady-state errors and justifies the exclusion of the integration term in the control block, as in Eq. S3.

In the PID control theory, the parameters k_p and k_d are determined such that the step response $y(t)$ approaches critical dumping.[2] However, the time-constant τ in Eq. S6 is highly dependent on the chemical and physical features of the system and is thus difficult to estimate. Accordingly, these values were empirically determined and set in our system using a GUI.

Thus far, the control system has been treated in the continuous-time domain for simplicity. The system was implemented as a discrete-time system on a control PC, which was temporally discretized to work in synchronization with the strobe signal of the video camera (currently f_{video} = 10 or 20 [Hz]). More specifically, it is a discretized version of the control system.

1. captures a frame and reads out the injector pressure from the A/D converter
2. extracts the geometric parameters of the bubble and calculates the target output in focus
3. calculates the next pulse count by which the motor is rotated
4. issues a rotation instruction for the motor
5. returns to step 1 and repeats steps 1–5

Let r_k, y_k, e_k, and u_k be discretized versions of their corresponding values $r(t)$, $y(t)$, $e(t)$, and $u(t)$, respectively, at t =

$T_s k$, where T_s is the time interval ($T_s = 1/f_{video} = 0.05$ [s] and k is an integer temporal index. In this case, the time constants τ of the bubble in Eq. S6 are significantly larger than T_s, and the differentiation $de(t)/dt$ in Eqs. S3 can be replaced by the finite difference $e_k - e_{k-1}$. Based on this replacement, the control block in Figure 5 calculates the pulse count, u_k by

$$u_k = k_p(r_k - y_k) + k_d((r_k - y_k) - (r_{k-1} - y_{k-1})) \qquad \text{Eq. S9}$$

in step 3.

Chart program: Graphical interface for tension-control experiment

We developed a program for the experimental protocol by setting the time course of tension changes (chart program) using a graphical interface (Fig. S3). The time course of the bilayer tension with the start and end times for a certain tension (tension protocol) was manipulated on the display. The software was organized into an image panel.

Figure S3. Graphical interface of the Chart program for programmed tension control experiments. A screenshot of the program is shown. Moving a cursor for an arbitrary time course can set a tension schedule. Here, the left and right leaflet tension is jumped from 2 mN/m to 4 mN/m simultaneously, leading the bilayer tension changes from 4 mN/m to 8 mN/m. Upon a step, the rate of tension changes is set arbitrarily at the left middle section, where the rate is set at 1.0 mN/m/sec. Moderate tension changes are recommended for stable operation in a long-run experiment covering a broad range of tension changes.

Tension step-changes

Upon step changes in the tension with the simultaneous operation of both bubbles, the time course of the changes in the bubble shape is shown as a video during 2 sec before and after the jump (Video 1; see also Fig. 4E). Upon simultaneous step changes in the both bubble pressure, the bubbles moved with a push-and-pull fashion due to bubble pressure unbalance. The contact angle changed immediately and slightly, reaching a new force balance between the

leaflets and monolayers (Eq. 6 and 7).

References

1. Bovik, A. Handbook of Image and Video Processing; Bovik, A., Ed.; *Communications, Networking, and Multimedia;* Academic Press: San Diego, 2000; pp 891.
2. Frank, S. A. Control Theory Tutorial: Basic Concepts Illustrated by Software Examples. *SpringerBriefs in Applied Sciences and Technology;* Springer Open, **2018**; pp 111.