Supplementary information for

Self–tuning PID controller based on analog–digital hybrid computing with double–gate SnS_2 memtransistor

Shania Rehman[1†], Muhammad Farooq Khan[2†], Hee-Dong Kim[1], and Sungho Kim[1]*

[1]Department of Semiconductor Systems Engineering and Convergence Engineering for Intelligent Drone, Sejong University, Seoul, Korea 05006

[2]Department of Electrical Engineering and Convergence Engineering for Intelligent Drone, Sejong University, Seoul, Korea 05006

[*]Corresponding author: *Sungho Kim*, email: sungho85kim@sejong.ac.kr

[†]These authors contributed equally to this work.

Supplementary Note 1. Preliminaries for the classical PID control algorithm

1.1 Characteristics of each gain

If the PID gains are changed, the resultant responses are summarized as follows.

1) Proportional gain (K_p) : From Eq. (1), increasing K_p has the effect of proportionally increasing the control signal $u(t)$ for the same level of error. As a result, increasing K_p will increase the speed of the control system response. However, if K_p is too large, the overshoot of the response is more, and the response will begin to oscillate.

2) Integral gain (K_i) : The addition of an integral term to the controller tends to help reduce steady-state error. The integral component sums the error over time. The result is that even a small error will cause the integral component to increase slowly. The integral response will continually increase over time unless the error is zero, so the effect is to drive the steady-state error to be zero. A drawback of the integral term, however, is that it can make the system more sluggish (and oscillatory) since when the error changes sign, it may take a while for the integrator to unwind.

3) Derivative gain (K_d) : The addition of a derivative term to the controller provides the ability of the controller to anticipate the error. With derivative term, the control signal can become large if the error begins sloping upward, even while the magnitude of the error is still relatively small. This anticipation tends to add damping to the system, thereby decreasing overshoot. The addition of a derivative term, however, has no effect on the steady-state error. Most practical control systems use very small K_d, because the drivative rsponse is highly sensitive to noise. If the sensor feedback signal is noisy, the derivative term can make the control system unstable.

gain	overshoot	settling time	steady-state error
K_p ↑	increase	small change	decrease
K_i ↑	increase	increase	decrease
K_d ↑	decrease	decrease	no change

Table S1. The summarized responses according to the change of each PID gain.

1.2 Ziegler-Nichols tuning method

The Ziegler-Nichols method is popular method for tuning the PID gains. It is very similar to the trial and error method. First, K_i and K_d are set to zero and K_p is increased until the response starts to oscillate. Once oscillation starts, the critical gain $K_p = K_{crit}$ and the period of oscillations T_{crit} are noted. The K_p, K_i and K_d are then selected as below.

gains	K_p	K_i	K_d
value	$0.6K_{crit}$	$1.2K_{crit}/T_{crit}$	$0.075K_{crit}\cdot T_{crit}$

Table S2. Typical tuning of PID gains through Ziegler-Nichols method.

Supplementary Note 2. The sensor fusion procedure to estimate current drone's attitude

The sensor fusion procedure for estimating the drone's attitude consists of two step; 1) measurement and 2) algorithm calculation.

<1st step : measurement>

The gyroscope measures the angular velocities (g_x, g_y, and g_z). In our experimental setup, because the drone setup is fixed to the ground, the body frame of the drone is equivalent to the inertial frame. Therefore, measured angular velocities directly represents the time derivatives of Euler angles (i.e., $\dot{\phi}_g$, $\dot{\theta}_g$, and $\dot{\psi}_g$) without any conversion.

$$\begin{pmatrix} \dot{\phi}_g \\ \dot{\theta}_g \\ \dot{\psi}_g \end{pmatrix} = \begin{pmatrix} g_x \\ g_y \\ g_z \end{pmatrix} \tag{S1}$$

By integrating of these time derivatives, Euler angles can be estimated.

Menwhile, the accelerometer measure the accelerations (a_x, a_y, and a_z), and the Euler angles can be estimated by

$$\begin{aligned} \phi_a &= \tan^{-1}\left(a_y / \sqrt{a_x^{\ 2} + a_z^{\ 2}}\right) \\ \theta_a &= \tan^{-1}\left(-a_x / \sqrt{a_y^{\ 2} + a_z^{\ 2}}\right) \end{aligned} \tag{S2}$$

The estimation of yaw angle from the accelerometer is not reliable; because the initial states of the drone is close to parallell with the axis of Earth's gravitational pull, any rotation around z-axis will have very little to no effect on the accelerometer output. Threrfore, a magnetometer is additionally required to accurately estimate yaw angle in general drones.

<2nd step : sensor fusion alogithrm calculation>

It has been known that the gyroscope and the accelerometer are prone to errors due to drift or noise. In detail, the gyroscope exhibits a steadily growing error over time because noise accumulates during the integration process. Meanwhile, the accelerometer is reliable only for steady-state acceleration, and high-frequency noise is inevitably included. Therefore, neither the gyroscope nor the accelerometer can be used alone to accurately estimate Euler angles; thus, a sensor fusion technique is widely exploited to compensate for the limitations of individual sensors by combining data from two sensors.

Among several algorithms for sensor fusion, the Kalman filter (KF) and complementary filter (CF) are the most popular. In our experiment, the CF is implemented because it can be executed with fewer computing resources, and it does not require a mathematical model of the system.

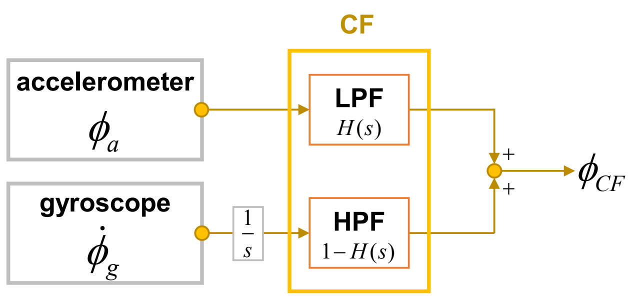

Figure S1. Basic structure of linear complementary filter

The basic structure of CF shown in Fig. S1 consists of a low-pass and a high-pass filter. The output of the CF, ϕ_{CF}, is given as

$$\phi_{CF} = [1 - H(s)]\left(\frac{\dot{\phi}_g}{s}\right) + H(s)\phi_a\,, \tag{S3}$$

where $\dot{\phi}_g$ and ϕ_a are obtained by the gyroscope and the accelerometer respsectively. $H(s)$ represents the transfer function for the low-pass filter (LPF), whereas $1-H(s)$ is the transfer function of the high-pass filter (HPF). This transfer function can be easily converted to the form of discrete-time by Tustin's method,

$$\phi_{CF} = \alpha \times \left(\phi_{prev} + \dot{\phi}_g \cdot dt \right) + \left(1 - \alpha \right) \phi_a \quad \text{(S4)}$$

where α is the smoothing factor which is within the range $0 \le \alpha \le 1$. In general, α is roughly determined as $\alpha = \tau / (\tau + \Delta t)$; Δt is the loop time of the CF alogirthm, and τ should be longer than the time constant of the accelerometer's noise. In our experiment, $\alpha = 0.996$ is used.

Supplementary Note 3. Double-gate SnS_2 memtransistor

Parameters	Single-gate memtransistor	Double-gate memtransistor
I_{on} (at V_G = +5V, V_D = 0.5V)	1.62 µA	6.04 µA
I_{off} (at V_G = −5V, V_D = 0.5V)	67 pA	11.9 pA
V_T (at $I_D = 10^{-7}$ A)	2.1 V	1.25 V
Subthreshold slope	0.8 V/dec	0.55 V/dec

Table S3. The comparison of performances between single-gate and double-gate memtransistors.

Step-1	**1st spin-coating layer of EL-9** • Spin-coating speed (1000 RPM: 10 sec/5000 RPM:30 sec) • Baking on hot plate: 10 min.
Step-2	**2nd spin-coating layer of PMMA** • Spin-coating speed (1000 RPM: 10 sec/5000 RPM: 30 sec) • Baking on hot plate: 10 min.
Step-3	**EBL process and parameters** • Beam spot size: 15 nm • Beam current: 30 pA
Step-4	**Developing** • 1st develop in MIBK for 5 sec. • 2nd develop in Xylene for 10 sec.

Table S4. The detail experimental process for e-beam lithography.

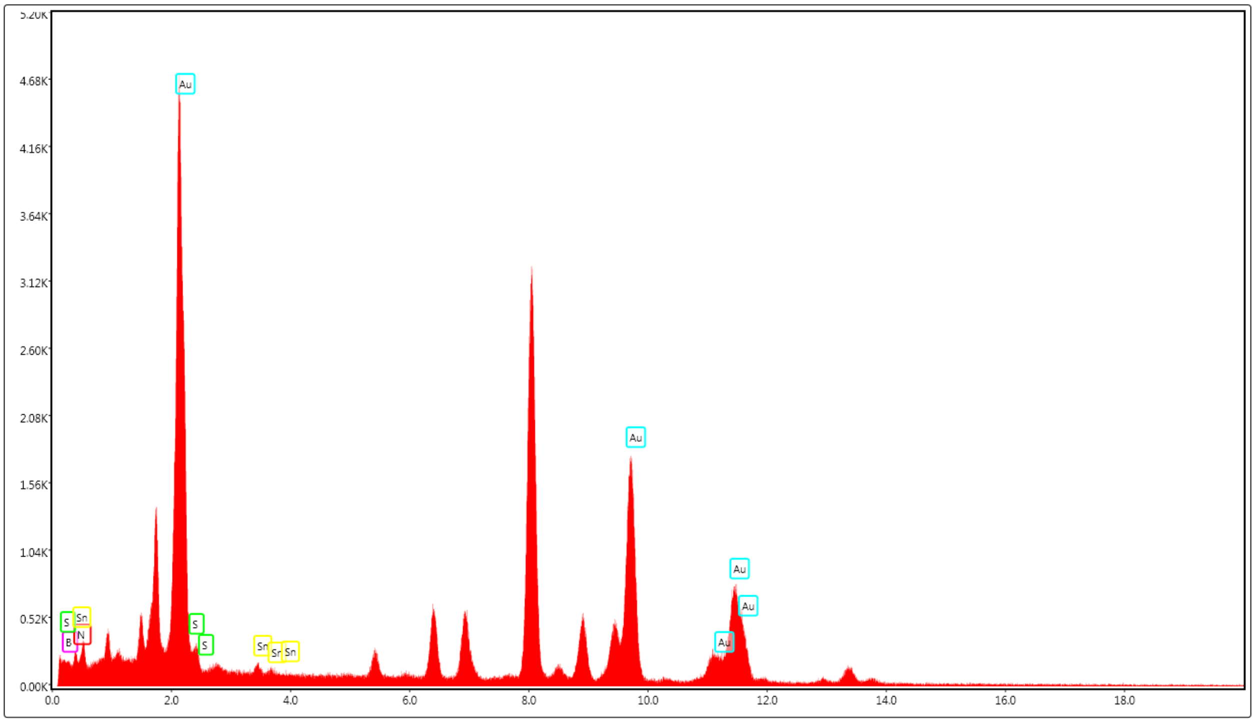

Figure S2. EDS spectrum of SnS_2 and h-BN nanosheets.

Fig. S3 depicts the differences between the transfer characteristics obtained from the pulsed I-V method and the conventional voltage sweep method. In a conventional voltage sweep method, a read voltage is applied for a specific duration (t_{mea}) during each voltage step. If t_{mea} is excessively prolonged, the measurement process itself modulates the channel conductance. As depicted in Fig. S3b, when t_{mea} is set to 100 ms, hysteresis occurs due to the continuous application of voltage for a relatively longer t_{mea}, resulting in the trapping/detrapping of electrons. In contrast, the pulsed I-V method (Fig. S3a) utilizes short duration voltage pulses (t_{mea} = 1 ms) to the gate and drain, while the drain current is measured only during the pulse application. The reduced t_{mea} prevents the occurrence of hysteresis in the transfer characteristic by ensuring the channel conductance remains stable within a sufficiently short t_{mea}. Moreover, the long interval time (100 ms) between pulses prevents any accumulation effect.

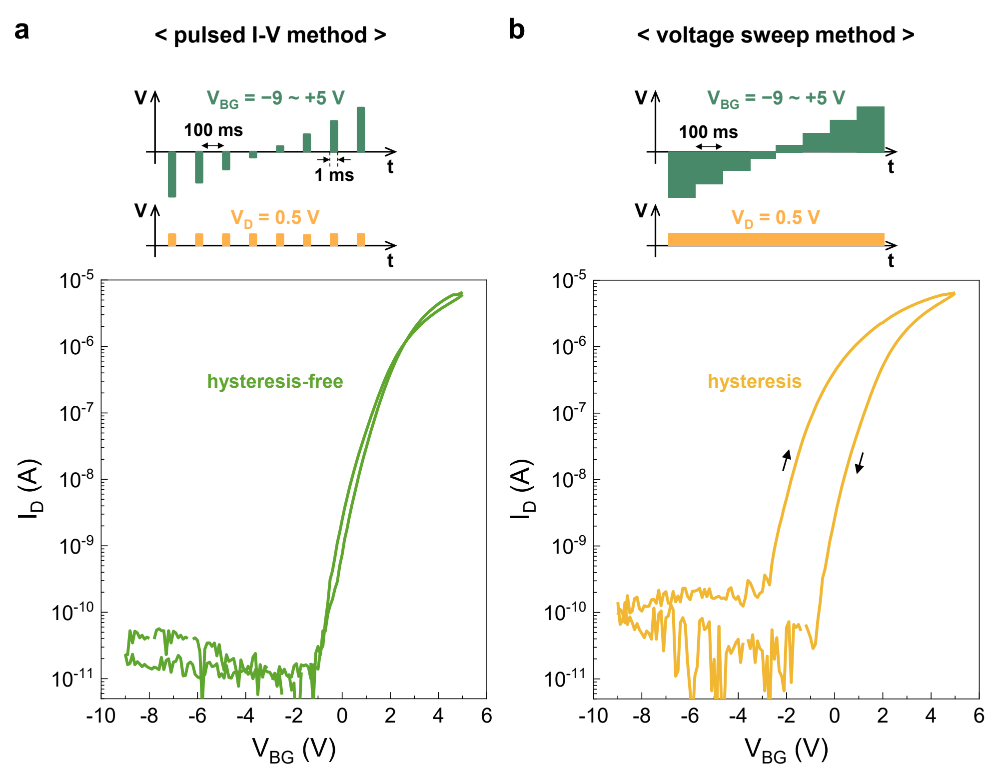

Figure S3. Measured transfer characteristics using (a) pulsed I-V method, and (b) general voltage sweep method, where $V_{TG} = V_{BG}$.

Fig. S4 shows a flow chart for the update-verify feedback process to precise control of channel conductance (G), which is executed in the microcontroller. By virtue of the iterative feedback process, G can be precisely adjusted to the desired value within a predetermined error range.

Each feedback cycle is based on a sequence of update & verify process, with each pulse pair consisting of an updating (potentiation or depression) pulse and a subsequent read pulse (V_G = 2 V, V_D = 0.5 V, 50 ms). During each cycle, the relative error (G_{var}) is calculated. If this relative error is within a predefined range (*e.g.*, ±5 %), the feedback process is finished. Otherwise, action is taken based on the sign of the relative error. For negative relative error, a potentiation pulse (V_{BG} = V_{TG} = −5 V, V_S = V_D = 0 V, 50 ms) is applied to increase G. Meanwhile, for positive relative error, a depression pulse (V_{BG} = V_{TG} = +5 V, V_S = V_D = 0 V, 50 ms) is applied to decrease G. In order to adjust desired G value (Fig. 2e), about 20 update-verify cycles are usually required.

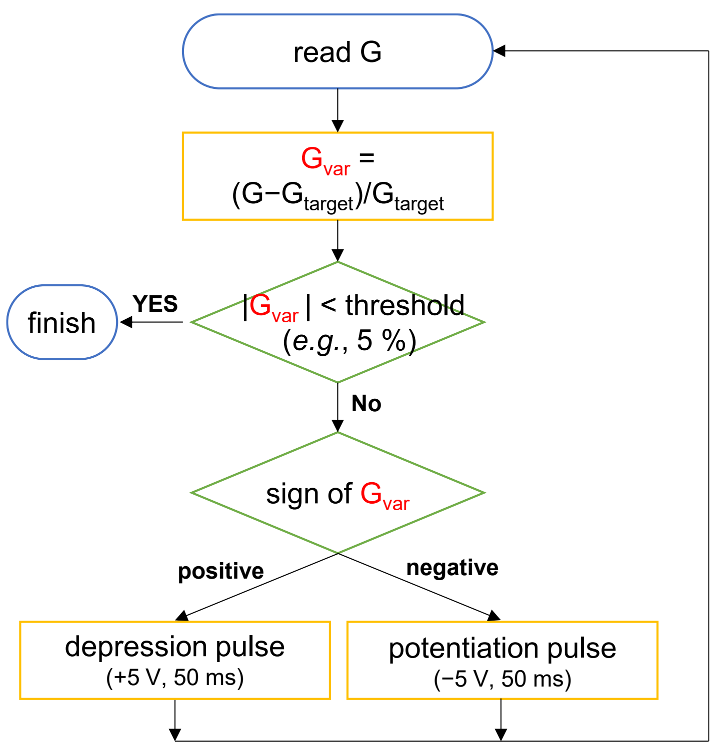

Figure S4. The flow chart for the update-verify feedback method.

Supplementary Note 4. The analog PID controller circuit

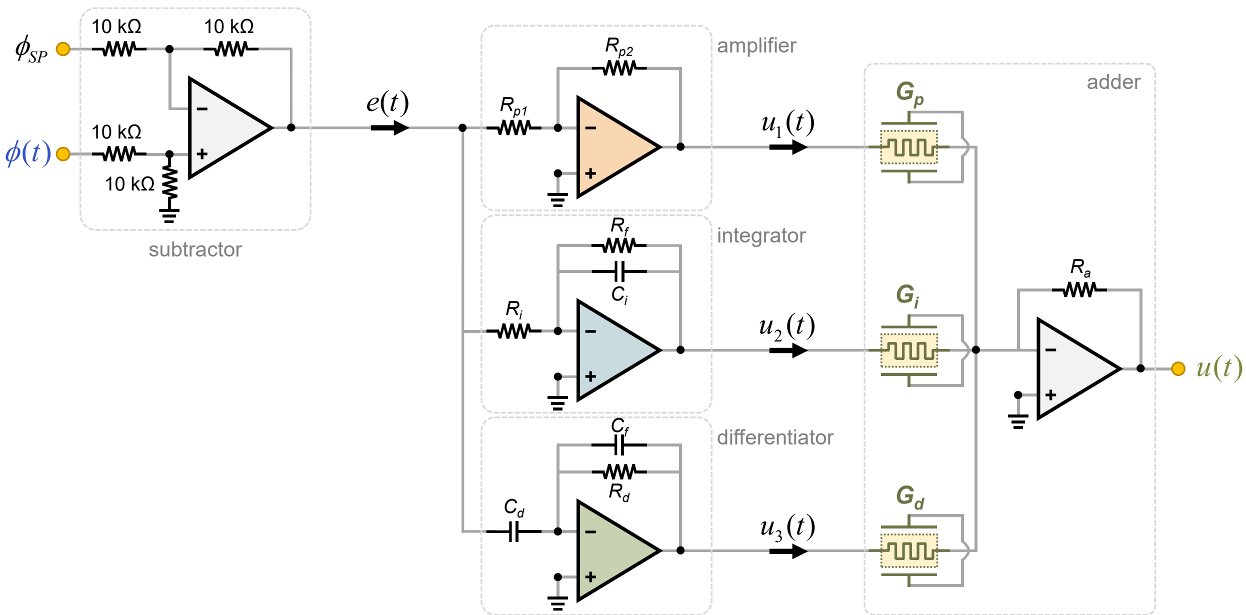

Figure S5. Circuit diagram of the analog PID controller circuit

Fig. S5 shows the designed analog circuit for executing the PID control algorithm. This analog PID controller circuit produces an output signal $u(t)$ from two input signals ($\phi(t)$ and ϕ_{SP}), through the following process.

1) The subtractor calculates the tracking error, i.e., $e(t) = \phi(t) - \phi_{SP}$.

2) $e(t)$ is input to an amplifier, integrator, and differentiator circuits respectively. The outputs of the amplifier, integrator, and differentiator circuits are

$$u_1(t) = -\left(\frac{R_{p2}}{R_{p1}}\right) e(t)$$
$$u_2(t) = -\left(\frac{1}{R_i C_i}\right) \int e(t)dt \qquad \text{(S4)}$$
$$u_3(t) = -\left(R_d C_d\right) \frac{de(t)}{dt}$$

Note that we add R_f and C_f additionally to the integrator and the differentiator circuits respectively. R_f avoids the saturation of $u_2(t)$ at low input frequency. C_f stabilizes the circuit at high input frequency, and also reduces the effect of noise on the circuit.

3) The final output $u(t)$ is a weighted sum of $u_1(t)$, $u_2(t)$ and $u_3(t)$.

$$u(t) = \left(\frac{R_{p2}}{R_{p1}} \cdot G_p R_a \right) e(t) + \left(\frac{1}{R_i C_i} \cdot G_i R_a \right) \int e(t)dt + \left(R_d C_d \cdot G_d R_a \right) \frac{de(t)}{dt} \qquad \text{(S5)}$$

Here, G_p, G_i, and G_d are the conductance value of each memtransistor. As a result, PID gains are determined as

$$\begin{aligned} K_p &= \left(R_{p2} / R_{p1} \right) \cdot \left(G_p R_a \right) \\ K_i &= \left(G_i R_a \right) / \left(R_i C_i \right) \\ K_d &= \left(R_d C_d \right) \cdot \left(G_d R_a \right) \end{aligned} \qquad \text{(S6)}$$

In our experiment, R_{p1} = 100 kΩ, R_{p2} = 50 kΩ, R_i = 100 kΩ, R_f = 1 kΩ, C_i = 200 µF, R_i = 100 kΩ, R_d = 10 MΩ, C_d = 5 nF, C_f = 0.1 nF, and R_a = 1 MΩ. When G_p = G_i = 1 µS and G_d = 0.5 µS, K_p, K_i, and K_d become 0.5, 0.05, 0.025 respectively.

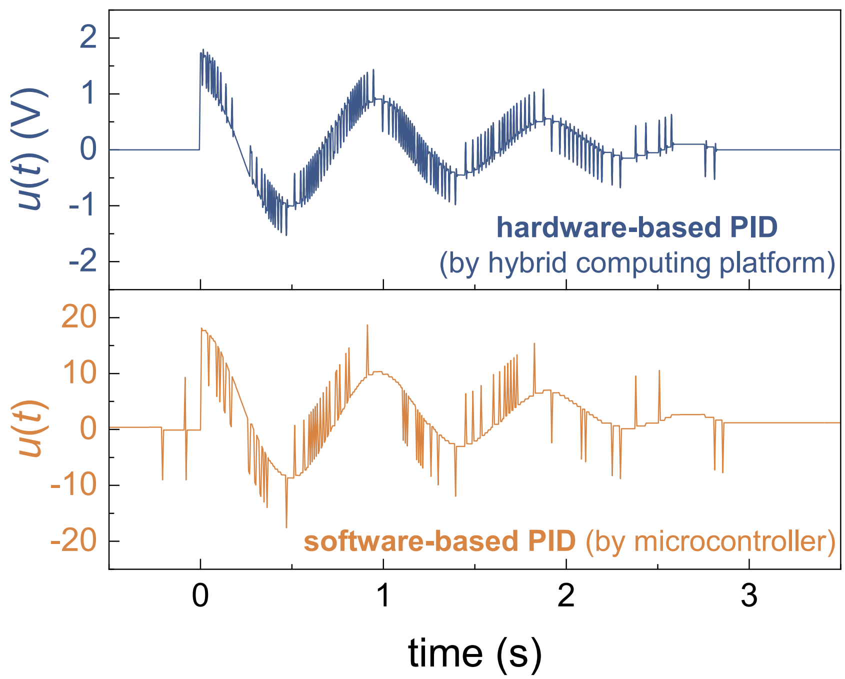

Figure S6. The comparison between $u(t)$ obtained by our analog PID controller circuit and by the calculation in the microcontroller.

	Variables	Values
Software-based PID (using only microcontroller)	V_{supply}	+5 V
	$I_{digital}$	40.5 – 43 mA (Fig. 4b)
	E_{PID}	$E_{PID} = \int \left[V_{supply} \times I_{digital}(t) \right] dt = 0.523\ J$
Hardware-based PID (using our hybrid computing platform)	V_{supply}	+5 V
	$I_{digital}$	~26.5 mA (Fig. 4c)
	I_{analog}	0.01 – 0.5 mA (Fig. 4c)
	E_{PID}	$E_{PID} = \int \left[V_{supply} \times \left(I_{digital}(t) + I_{analog}(t) \right) \right] dt = 0.332\ J$

Table S5. The comparison of energy consumption between software-based and hardware-based PID contollers.

Supplementary Note 5. Reconfigurability of the PID controller

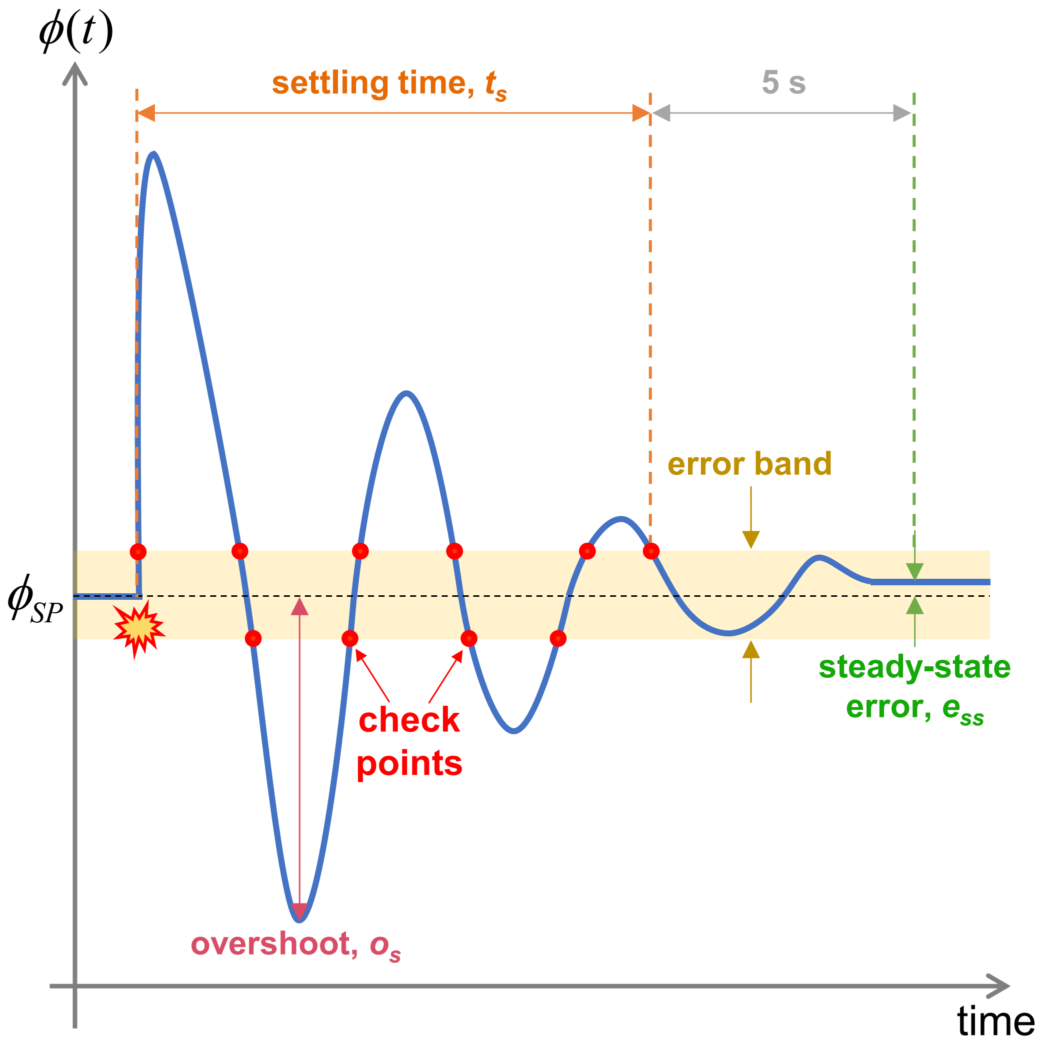

Figure S7. The schematic for the definition of parameters (t_s, o_s, and e_{ss}).

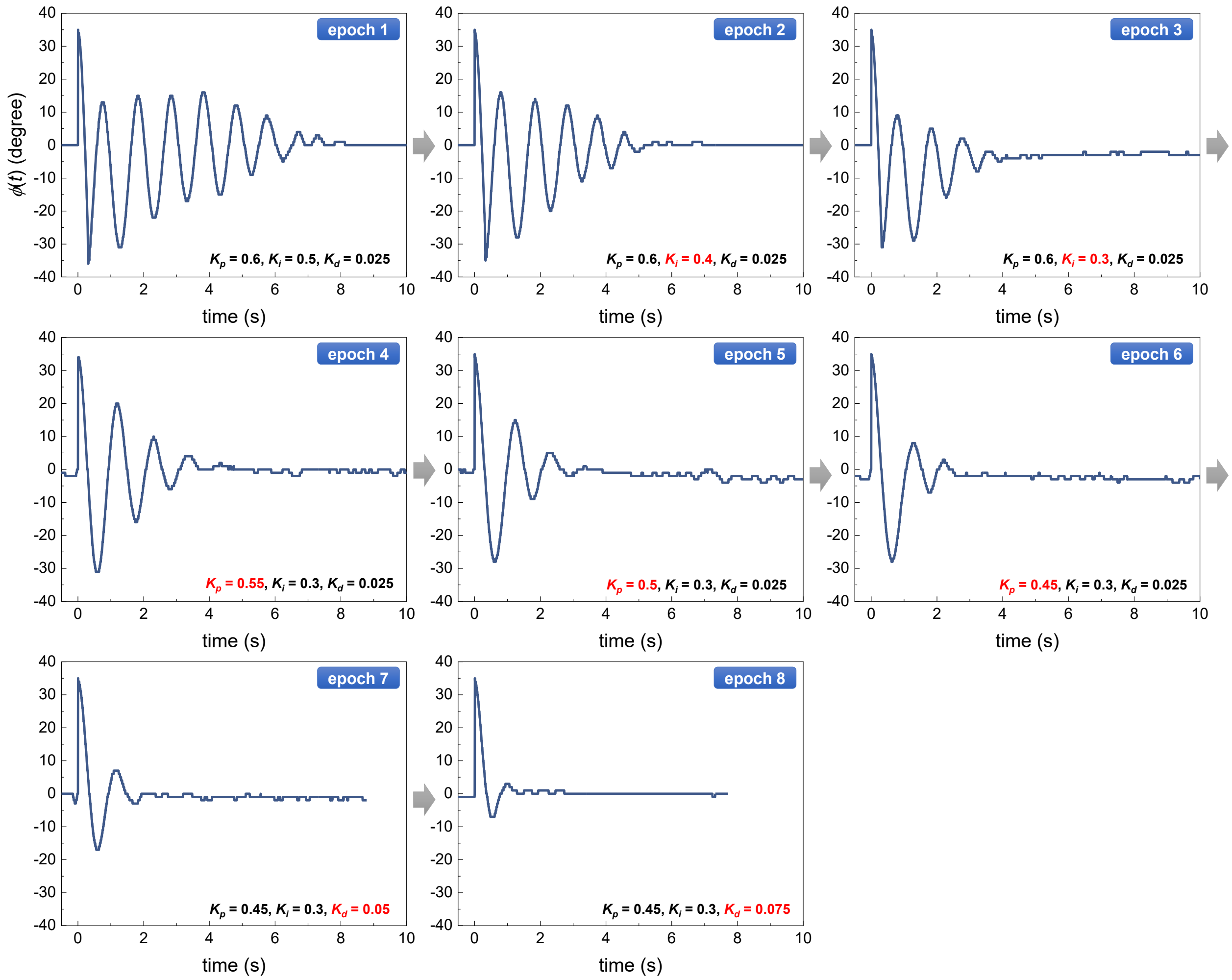

Figure S8. The evolution of PID control performance through the self-tuning algorithm.

Supplementary Note 6. The suumary of all abbreviations used in the text

category	abbreviation	full name or definition
Generally used terms	PID	Proportional-Integral-Derivative
	UAV	Unmanned Aerial Vehicle
	memtransistor	Memristor with a three-terminal transistor structure
	IMU	Inertial measurement unit
	DAC	Digital-To-Analog converter
	ADC	Analog-To-Digital converter
	ESC	Electronic Speed Controller
	TEM	Transmission electron microscopy
	I-V	Current-Voltage
	SnS_2	Tin disulfide
	h-BN	Hexagonal boron nitride
The attitude of drones	ϕ, θ, ψ	roll, pitch, and yaw angles
	$\hat{\phi}$	Actual roll angle of the drone
	$\phi(t)$	The estimated roll angle of the drone from IMU
	ϕ_{SP}	The target (set point) roll angle of the drone
Sensing data from IMU	a_x, a_y, a_z	Measured linear accelerations from the accelerometer
	g_x, g_y, g_z	Measured angular velocities from the gyroscope
PID control	K_p, K_i, K_d	Proportional, integral, and derivative gain in the PID controller
	o_s	Overshoot during the PID control
	t_s	Settling time during the PID control
	e_{ss}	Steady-state error during the PID control
Generated control signals for PID control	$u(t)$	Output signal of the PID controller circuit
	$y(t)$	Digitized $u(t)$ signal using ADC in the microcontroller
memtransistor	G	Channel conductance value of the memtransistor
	G_p, G_i, and G_d	G of three memtransistors, which determines the amount of K_p, K_i, K_d respectively
	V_{TG}, V_{BG}	Top and bottom gate voltages
	V_{write}	The gate voltage pulse to modulate G
Energy consumption	V_{supply}, I_{supply}	The amount of voltage and current provided from the power supply equipment
	I_{analog}, $I_{digital}$	I_{supply} for the analog and digital components
	E_{PID}	Total energy consumption for the PID control

Table S6. The summarized all abbreviations and their full name or definition.