nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-022-29640-7 OPEN

Check for updates

A hierarchy of biomolecular proportional-integral-derivative feedback controllers for robust perfect adaptation and dynamic performance

Maurice Filo[1], Sant Kumar[1] & Mustafa Khammash[1]✉

Proportional-Integral-Derivative (PID) feedback controllers are the most widely used controllers in industry. Recently, the design of molecular PID-controllers has been identified as an important goal for synthetic biology and the field of cybergenetics. In this paper, we consider the realization of PID-controllers via biomolecular reactions. We propose an array of topologies offering a compromise between simplicity and high performance. We first demonstrate that different biomolecular PI-controllers exhibit different performance-enhancing capabilities. Next, we introduce several derivative controllers based on incoherent feedforward loops acting in a feedback configuration. Alternatively, we show that differentiators can be realized by placing molecular integrators in a negative feedback loop, which can be augmented by PI-components to yield PID-controllers. We demonstrate that PID-controllers can enhance stability and dynamic performance, and can also reduce stochastic noise. Finally, we provide an experimental demonstration using a hybrid setup where in silico PID-controllers regulate a genetic circuit in single yeast cells.

[1] Department of Biosystems Science and Engineering, ETH Zürich, Mattenstrasse 26, 4058 Basel, Switzerland. ✉email: mustafa.khammash@bsse.ethz.ch

One of the most salient features of biological systems is their ability to adapt to their noisy environments. For example, cells often regulate gene expression to counteract intrinsic and extrinsic noise in order to maintain a desirable behavior in a precise and timely fashion. This resilience toward undesired disturbances is often achieved via feedback control that has proved to be ubiquitous in both natural (e.g.[1–3]) and engineered systems (e.g.[4,5]). In fact, synthetic engineering of biomolecular feedback controllers is gaining wide attention from biologists and engineers (e.g.[6–14]).

A standard general setup for feedback controllers is depicted as a block diagram in Fig. 1a. Refer to Supplementary Box 1 (A Primer on Block Diagrams). The “Plant" block represents the process to be controlled. It can be actuated through its input, denoted here by u, to dynamically manipulate its output of interest, denoted here by y. The objective of such control systems is to design a feedback controller that automatically actuates the plant in a smart autonomous fashion which guarantees that the output y meets certain performance goals despite the presence of disturbances in the plant. These performance goals, described in Fig. 1b, include (but are not limited to) robust steady-state tracking—also known as robust perfect adaptation (RPA) in biology, stability enhancement, desirable transient response and variance reduction. Control theory developed a wide set of tools to design feedback controllers that meet certain performance objectives. For instance, it is well known in control theory (Internal Model Principle[15]) that a controller should include integral (I) action to be able to achieve RPA. Furthermore, proportional-integral-derivative (PID) feedback controllers—first rigorously introduced by Nicolas Minorsky[16] around a hundred years ago—adds proportional (P) and derivative (D) action to the integrator (I) to be able to tune the transient dynamics and enhance stability while preserving RPA. Interestingly, after almost a century, PID controllers are still the most widely used controllers in industrial applications[17–19].

Originally, PID feedback controllers were designed to control mechanical (later, electrical and chemical) systems such as automatic ship steering[20]. Such control systems involve controlling quantities that can take both negative and positive values such as angles, velocities, electric currents, voltages, etc. Furthermore, traditional PID controllers possess linear dynamics since all three operations of a PID are linear. Two classes of linear PID controllers, adopted from Chapter 10 of ref. [21], are shown in Fig. 1c, d. In Fig. 1c, the error signal $e(t) := r - y(t)$ is fed into the three (P, I, and D) components. The outputs of the three components are summed up to yield the control action u that serves as the actuation input to the plant. However, in Fig. 1d, the controller has two degrees of freedom since both the error e and the output y are used separately and simultaneously. Particularly, the error is fed into the integrator, while the output is fed into the proportional and derivative components. Observe that both architectures require that the integrator operates on the error (and not the output). This is necessary to achieve RPA and can be easily seen using a very simple argument explained next. Let $u_I(t)$ denote the output of the integrator, that is,

$$u_I(t) := K_I \int_0^t e(\tau)d\tau \quad \Rightarrow \quad \dot{u}_I(t) = K_I e(t). \qquad (1)$$

Assuming that the dynamics are stable, then at steady state we have $\lim_{t\to\infty} \dot{u}_I(t) = 0$. This implies that, at steady state, the error $e := r - y$ has to be zero, and thus $\lim_{t\to\infty} y(t) = r$, hence achieving the steady-state tracking property. Observe that although this argument requires closed-loop stability, it does not depend on the particular structure and/or parameters of the plant, hence achieving the robustness property.

For mechanical and electrical systems, the linearity of the PID controllers is convenient because of the availability of basic physical parts (e.g., dampers, springs, RLC circuits, op-amps, etc.) that are capable of realizing these linear dynamics. However, this realization quickly becomes challenging when designing biomolecular controllers. This difficulty arises because (a) biomolecular controllers have to respect the structure of BioChemical Reaction Networks (BCRN) that are inherently nonlinear, and (b) the quantities to be controlled (protein copy numbers or concentrations) cannot be negative (see[22] for positive integral control). To achieve RPA, BCRN realizations of standalone integral (I) controllers received considerable attention[23–28]. In previous work[25], the antithetic integral (*a*I) feedback controller was introduced to realize integral action that ensures RPA. More recently, it was shown in[9] that the antithetic motif is necessary to achieve RPA in arbitrary intracellular networks with noisy dynamics. A detailed mathematical analysis of the performance tradeoffs that may arise in the *a*I controller is presented in[29,30], and optimal tuning is treated in[31]. Furthermore, practical design aspects, particularly the dilution effect of controller species, are addressed in[9,27]. Biological implementations of various biomolecular integral and quasi-integral controllers appeared in bacteria in vivo[6,8,9] and in vitro[13], and more recently in mammalian cells in vivo[14] and in yeast using the cyberloop in silico[32].

In the pursuit of designing high-performance controllers while maintaining the RPA property, BCRN realizations of PI and PID controllers are starting to receive more focused attention[33–38]. Particularly in[33], a proportional component is separately appended to the antithetic integral motif via a repressing Hill-type function to tune the transient dynamics and reduce the variance. The resulting PI controller follows the concept of Fig. 1d where error and output feedback are used to build separate (but nonlinear) P and I components. Several successful attempts were carried out to devise BCRN realizations that approximate derivatives[39–43]. A BCRN realization of a full PID controller was reported in[35], where the authors introduced additional controller species to obtain a derivative component. The resulting PID controller uses error feedback (similar to the concept of Fig. 1c) to build separate nonlinear P, I, and D components and successfully improves the dynamic performance in the deterministic setting. Using a different approach, [37] and [38] exploit the dual-rail representation from[23], where additional species are introduced to overcome the non-negativity challenge of the realized PID controller. The authors demonstrate the resulting improvement of the performance in the deterministic setting. On a different note, [36] analyzed the effects of separate proportional and derivative controllers on (bursty) gene expression models in the stochastic setting.

Interestingly, previous research in this direction shares two intimately related aspects. Firstly, the P, I, and D components are realized separately such that they enter the dynamics additively. This aspect is motivated by traditional PID controllers where the controller dynamics are constrained to be linear, and thus the three components have to be added up (rather than multiplied for example). However, since feedback mechanisms in BCRNs are inherently nonlinear, there is no reason to restrict the controller to have linear dynamics and/or additive components. Secondly, the proposed designs introduce additional species to mathematically realize the controller, and thus making the biological implementation more difficult. To this end, we consider in this paper (more general) nonlinear PID controllers that do not have to be explicitly separable into their three (P, I, and D) components. This allows controllers to involve P, I, and D architectures in one (inseparable) block as depicted in Fig. 1e where both, error and output, feedback are allowed. The nonlinearity and inseparability features of the proposed PI and PID controllers provide more flexibility in the BCRN design and allow simpler

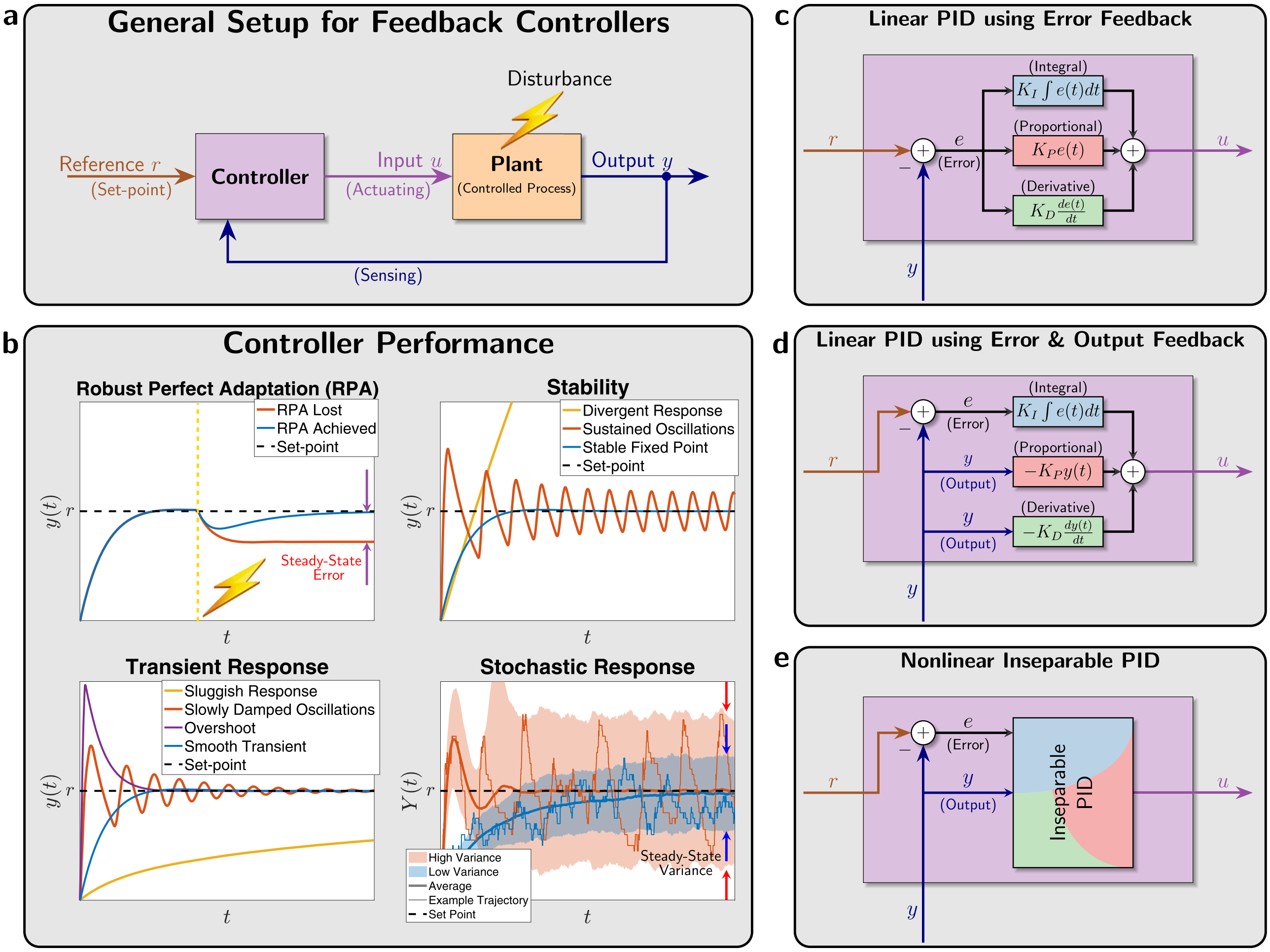


Fig. 1 Feedback controller design and performance. a The output to be controlled is fed back into the controller via a sensing mechanism. The controller exploits the setpoint, which is typically "dialed in" by the user and computes the suitable control action to be applied to the plant (or process) via an actuation mechanism. The goal of the control action is to steer the output to the desired setpoint despite external or even internal disturbances. **b** A demonstration of four performance goals that are typically targeted when designing the controller. Robust perfect adaptation (RPA): this is the biological analog of the notion of robust steady-state tracking (RSST) that is well known in control theory[61]. A controller achieves RPA if it drives the steady state of the plant output y to the setpoint (or reference, denoted by r) despite varying initial conditions, plant uncertainties and/or constant disturbances. Stability enhancement: a typical goal of a controller is to stabilize the dynamics. That is, it forces the output y to converge to a fixed steady-state value thus avoiding divergent responses and sustained oscillations. Desirable transient response: another typical control objective is to yield a smooth transient response that is fast enough but does not overshoot or oscillate too much. Variance reduction: for stochastic dynamics, it is common to study the time evolution of the output probability distribution and its moments such as the mean and variance. A natural performance objective is to design a controller that tightens the probability distribution around the mean, e.g., reduce the variance (cell-to-cell variability). Note that the shaded regions denote the mean ± three standard deviations. **c–e** Various PID control architectures. The classical designs in (**c**) and (**d**) involve separate linear P, I and D operations that are added together to yield the control action u. The difference between (**c**) and (**d**) is in the controller input: in (**c**) the error signal is the only input, while in (**d**) the error signal is fed into the I component whereas the output signal is fed into the P and D components. In this paper, we propose PID control architectures that fit in the more general class depicted in (**e**) where the PID components may be nonlinear and inseparable. This gives more design flexibility for biomolecular controllers.

architectures that do not require introducing additional species to the standalone integral controller. Next, we slightly increase the complexity (or order) of the controller designs by introducing up to two additional controller species. This provides more degrees of freedom for the controller and, as a result, offers a higher achievable performance. Furthermore, the higher the order of the controller, the more separable it is which facilitates the tuning of the PID gains by the biomolecular parameters. This hierarchical approach offers the designer a natural compromise between simplicity and performance enhancement. A rich library of biomolecular PID controllers of variable complexity is presented in this paper to offer the biologists a flexible and wide range of designs that can be selected depending on the desired performance and application at hand.

Results

General framework for biomolecular feedback controllers. The framework for feedback control systems is traditionally described through block diagrams (e.g., Fig. 1a). In this section, we lay down a general framework for feedback control systems where both the plant and the controller are represented by BCRNs. With this framework, the controller can either represent an actual biomolecular circuit or it can be implemented as a mathematical algorithm in silico[44–46] to regulate a biological circuit (through light for example[32,47]).

Consider a general plant, depicted in Fig. 2, comprised of L species $\mathbf{X} := \{\mathbf{X}_1, \cdots, \mathbf{X}_L\}$ that react with each other through K reaction channels labeled as $\mathcal{R} := \{\mathcal{R}^1, \mathcal{R}^2, \cdots, \mathcal{R}^K\}$. Each reaction $\mathcal{R}^k$ has a stoichiometry vector denoted by $\zeta_k \in \mathbb{Z}^L$ and a propensity function

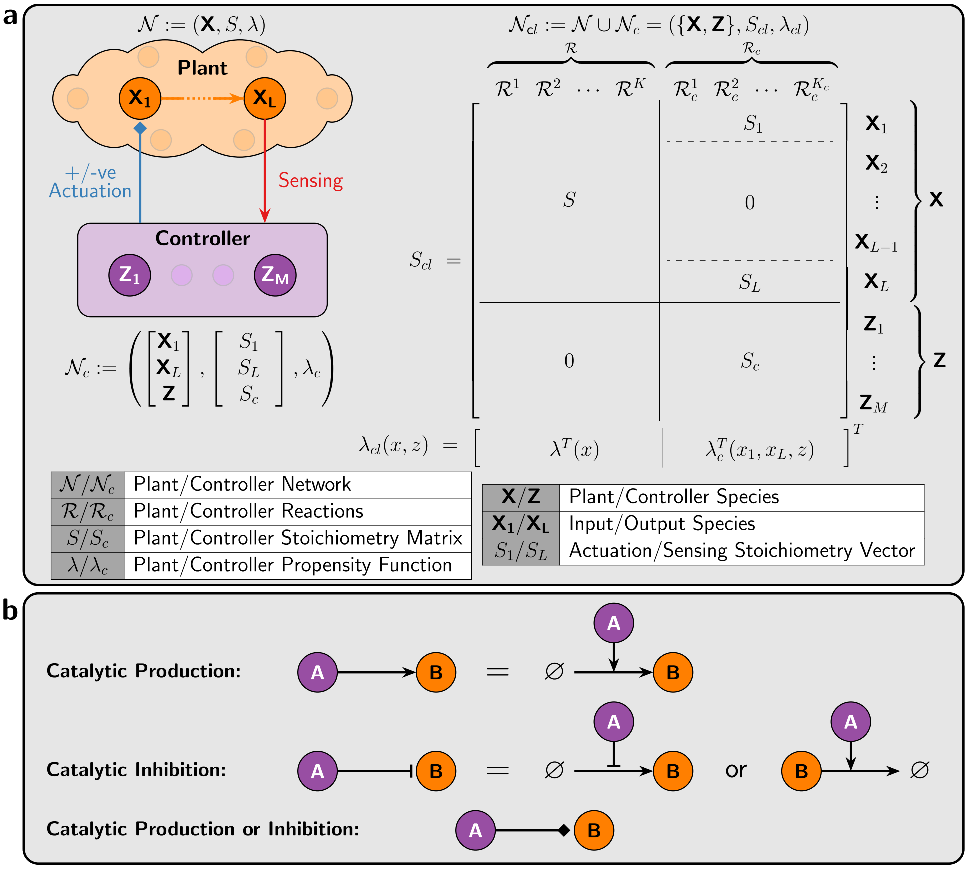

a
$\mathcal{N} := (\mathbf{X}, S, \lambda)$
Plant
X_1
X_L
+/-ve Actuation
Sensing
Controller
Z_1
Z_M
$\mathcal{N}_c := \left(\begin{bmatrix} X_1 \\ X_L \\ Z \end{bmatrix}, \begin{bmatrix} S_1 \\ S_L \\ S_c \end{bmatrix}, \lambda_c \right)$
$\mathcal{N}_{cl} := \mathcal{N} \cup \mathcal{N}_c = (\{\mathbf{X}, \mathbf{Z}\}, S_{cl}, \lambda_{cl})$
$\mathcal{R}$ $\mathcal{R}_c$
$\mathcal{R}^1$ $\mathcal{R}^2$ $\cdots$ $\mathcal{R}^K$ $\mathcal{R}_c^1$ $\mathcal{R}_c^2$ $\cdots$ $\mathcal{R}_c^{K_c}$
$S_{cl} =$ S S_1 0 S_L 0 S_c
X_1 X_2 $\vdots$ X_{L-1} X_L Z_1 $\vdots$ Z_M
X
Z
$\lambda_{cl}(x, z) = [\lambda^T(x) \quad \lambda_c^T(x_1, x_L, z)]^T$
$\mathcal{N}/\mathcal{N}_c$ Plant/Controller Network
$\mathcal{R}/\mathcal{R}_c$ Plant/Controller Reactions
S/S_c Plant/Controller Stoichiometry Matrix
λ/λ_c Plant/Controller Propensity Function
X/Z Plant/Controller Species
X_1/X_L Input/Output Species
S_1/S_L Actuation/Sensing Stoichiometry Vector
b
Catalytic Production:
A B = ∅ A B
Catalytic Inhibition:
A B = ∅ A B or B A ∅
Catalytic Production or Inhibition:
A B

Fig. 2 A framework for feedback control of chemical reaction networks. a An arbitrary plant is comprised of L species $\{\mathbf{X_1}, \cdots, \mathbf{X_L}\}$ reacting with each other. Species $\mathbf{X_L}$, by definition, is the output of interest to be controlled, while $\mathbf{X_1}$ is assumed to be the only accessible input species that can be "actuated" (positively and/or negatively) by the controller network which is comprised of M species $\{\mathbf{Z_1}, \cdots, \mathbf{Z_M}\}$. The closed-loop system, with stoichiometry matrix S_{cl} and propensity function λ_{cl}, denotes the overall feedback interconnection between the plant and controller networks. The partitioning of S_{cl} and λ_{cl} describes the various components of the closed-loop network. **b** A description of the compact graphical notation that is adopted throughout the paper. Arrows directed toward species indicate catalytic productions, whereas T-shaped lines indicate catalytic inhibitions that encompass either repressive production or degradation. Note that the propensities of degradation reactions are considered to be either $kAB/(B+\kappa)$ if two parameters (k, κ) are indicated on the arrow, or ηAB if only one parameter η is indicated on the arrow. Finally, diamonds indicate either production or inhibition.

$\lambda_k : \mathbb{R}_+^L \to \mathbb{R}_+$. Let $S := [\zeta_1 \quad \zeta_2 \quad \cdots \quad \zeta_K] \in \mathbb{Z}^{L\times K}$ denote the stoichiometry matrix and let $\lambda := [\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_K]^T$ denote the (vector-valued) propensity function. Then, the plant constitutes a BCRN that is fully characterized by the triplet $\mathcal{N} := (\mathrm{X}, S, \lambda)$, which we shall call the "open-loop" system.

The goal of this work is to design a controller network, denoted by $\mathcal{N}_c$, that is connected in feedback with the plant network $\mathcal{N}$, as illustrated in Fig. 2a, to meet certain performance objectives such as those mentioned in Fig. 1b. We assume that all the plant species are inaccessible by the controller except for species $\mathbf{X_1}$ and $\mathbf{X_L}$. Particularly, the controller "senses" the plant output species $\mathbf{X_L}$, then "processes" the sensed signal via the controller species $\mathbf{Z} := \{\mathbf{Z_1}, \cdots, \mathbf{Z_M}\}$, and "actuates" the plant input species $\mathbf{X_1}$. The controller species are allowed to react with each other and with the plant input/output species through K_c reaction channels labeled as $\mathcal{R}_c := \{\mathcal{R}_c^1, \mathcal{R}_c^2, \cdots, \mathcal{R}_c^{K_c}\}$. Let $\bar{S}_c \in \mathbb{Z}^{(M+2)\times K_c}$ and $\lambda_c : \mathbb{R}_+^{M+2} \to \mathbb{R}_+^{K_c}$ denote the stoichiometry matrix and propensity function of the controller, respectively. Since the controller reactions $\mathcal{R}_c$ involve the controller species $\mathbf{Z}$ and the plant input/output species $\mathbf{X_1}/\mathbf{X_L}$, the stoichiometry matrix $\bar{S}_c$ can be partitioned as

$$\bar{S}_c := [S_1 S_L S_c],$$

where S_1 and $S_L \in \mathbb{Z}^{1\times K_c}$ encrypt the stoichiometry coefficients of the plant input and output species $\mathbf{X_1}$ and $\mathbf{X_L}$, respectively, among the controller reaction channels $\mathcal{R}_c$. Furthermore, $S_c \in \mathbb{Z}^{M\times K_c}$ encrypts the stoichiometry coefficients of the controller species $\mathbf{Z_1}, \cdots, \mathbf{Z_M}$. Hence, the controller design problem boils down to designing S_1, S_L, S_c and λ_c. Note that, for simplicity, we consider plants with single-input-single-output species. However, this can be straightforwardly generalized to multiple-input-multiple-output species by adding more rows to S_1 and S_L. Finally, the closed-loop system constitutes the open-loop network augmented with the controller network so that it includes all the plant and controller species $\mathbf{X_{cl}} := \mathbf{X} \cup \mathbf{Z}$ and reactions $\mathcal{R}_{cl} := \mathcal{R} \cup \mathcal{R}_c$. Thus, the closed-loop network, $\mathcal{N}_{cl} := \mathcal{N} \cup \mathcal{N}_c$, can be fully represented by the closed-loop stoichiometry matrix S_{cl} and propensity function λ_{cl} described in Fig. 2a. We close this section, by noting that our proposed controllers range from simple designs involving $M = 2$ controller species, up to more complex designs involving $M = 4$ controller species.

Antithetic proportional-integral (*a*PI) feedback controllers. Equipped with the BCRN framework for feedback control systems, we are now ready to propose several PI feedback controllers that are capable of achieving various performance objectives. All of the proposed controllers involve the antithetic integral motif introduced in[25] to ensure RPA. However, other additional motifs are appended to mathematically realize a proportional (P) control action.

Consider the closed-loop network, depicted in Fig. 3, where an arbitrary plant is connected in feedback with a class of controllers that we shall call *a*PI controllers. Observe that there are three different inhibition actions that are color coded. Each inhibition action gives rise to a single class of the proposed *a*PI controllers. Particularly, when no inhibition is present, we obtain the standalone antithetic integral (*a*I) controller of[25] whose reactions are summarized in the left table of Fig. 3, whereas *a*PI of Class 1 (resp. Class 2) involves the inhibition of the input $\mathbf{X_1}$ by the output $\mathbf{X_L}$ (resp. controller species $\mathbf{Z_2}$), and *a*PI of Class 3 involves the inhibition of the controller species $\mathbf{Z_1}$ by the output $\mathbf{X_L}$. Furthermore, each *a*PI class encompasses various types of controllers depending on the inhibition mechanisms that enter the controller network as actuation reactions. We consider three types of biologically relevant inhibition mechanisms detailed in Fig. 3: additive, multiplicative (competitive) and degradation. Considering all three *a*PI classes with the various inhibition mechanisms, Fig. 3 proposes eight different *a*PI control architectures. Note that, it can be shown that a degradation inhibition in the case of *a*PI Class 3 would destroy the RPA property and is thus omitted. All of these controllers are compactly represented by a single general closed-loop stoichiometry matrix S_{cl} and propensity function λ_{cl} depicted in Fig. 3. The various architectures can be easily obtained by suitably selecting the functions $h := h^+ - h^-$ and g from the tables of Fig. 3. A theoretical linear perturbation analysis is carried out in Supplementary Information Section 1.1 to verify the proportional-integral control structure of the proposed controllers. In fact, the analysis applies to any smooth function h which is monotonically increasing (resp. decreasing) in z_1 (resp. z_2, x_1 and x_L), and any smooth function g that is monotonically increasing (resp. decreasing) in μ (resp. x_L). For example, linear terms in h such as kz_1 can be replaced with increasing Hill-type functions to model saturation whenever needed. The genetic implementation of the various control mechanisms presented here (and throughout the paper) can be carried out using basic activators, repressors and proteases which do not require any complicated biophysical mechanisms to implement the control designs. For instance, in the case of Class 1 *a*PI with degradation (see Fig. 3), the output of interest $\mathbf{X_L}$ can be fused to a protease capable of degrading the input $\mathbf{X_1}$ (see below for more details on the genetic implementations).

Deterministic steady-state analysis: robust perfect adaptation (RPA) of *a*PI controllers. The deterministic dynamics of the closed-loop systems, for all the *a*PI controllers given in Fig. 3 can be compactly written as a set of ordinary differential equations (ODEs) given by

$$\begin{cases} \dot{x} = S\lambda(x) + h(z_1, z_2, x_1, x_L)e_1 \\ \dot{z}_1 = g(\mu, x_L) - \eta z_1 z_2 \\ \dot{z}_2 = \theta x_L - \eta z_1 z_2, \end{cases} \tag{2}$$

where $e_1 := [1 \quad 0 \quad \cdots \quad 0]^T \in \mathbb{Z}^L$. Note that the total actuation and reference propensities h and g take different forms for different *a*PI control architectures as depicted in Fig. 3. The fixed point of the closed-loop dynamics cannot be calculated explicitly for a general plant; however, the output component (x_L) of the fixed point solves the following algebraic equation

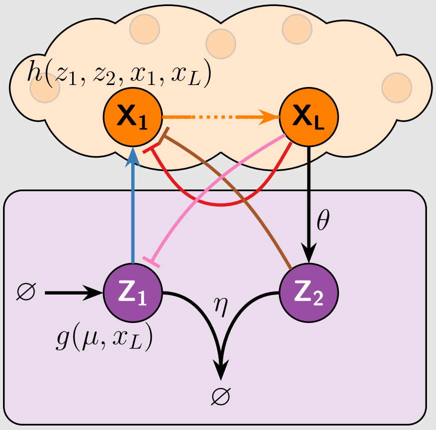
$$g(\mu, \bar{x}_L) = \theta \bar{x}_L, \tag{3}$$

where over-bars denote steady-state values (if they exist), that is $\bar{x}_L := \lim_{t\to\infty} x_L(t)$. Two observations can be made based on (3). The first observation is that (3) has a unique nonnegative solution since g is a monotonically decreasing function in x_L. The second observation is that (3) does not depend on the plant. As a result, if the closed-loop system is stable (i.e., the dynamics converge to a fixed point), then the output concentration converges to a unique setpoint that is independent of the plant. This property is valid for any initial condition, and is referred to as RPA. Particularly, for the *a*I and *a*PI controllers of Class 1 and 2, the reference propensity is $g(\mu, x_L) = \mu$, and thus $\bar{x}_L = \frac{\mu}{\theta}$. Furthermore, for the *a*PI of Class 3, $\bar{x}_L$ solves a polynomial equation of degree $n+1$, where n denotes the Hill coefficient depicted in Fig. 3 (see Supplementary Information Section 3). In conclusion, all the proposed *a*PI controllers maintain the RPA property that is obtained by the antithetic integral motif, while introducing additional control knobs as extra degrees of freedom to enhance other performance objectives.

Deterministic stability analysis and performance assessment of *a*PI controllers. Next, we show that Class 1 *a*PI controllers with degradation and multiplicative inhibitions yield superior stability and performance properties. To compare the stability properties of the various proposed *a*PI controllers, we consider a particular plant, depicted in Fig. 4a, that is comprised of two species $\mathbf{X_1}$ and $\mathbf{X_2}$ ($L = 2$). This plant may represent a gene expression network where $\mathbf{X_1}$ is the mRNA that is translated to a protein $\mathbf{X_2}$ at a rate k_1. The degradation rates of $\mathbf{X_1}$ and $\mathbf{X_2}$ are denoted by γ_1 and γ_2, respectively. The closed-loop stoichiometry matrix and propensity function are also shown in Fig. 4a. Using the Routh-Hurwitz stability criterion[48], one can establish the exact conditions of local stability of the fixed point (Supplementary Equation (17)) for the various proposed *a*PI controllers. These conditions, once satisfied, guarantee that the dynamics locally converge to the fixed point.

For the remainder of this section, we consider fast sequestration reactions, that is, η is large. Under this assumption, one can obtain simpler stability conditions that are calculated in Supplementary Information Section 3, and tabulated in Fig. 4b. The stability conditions are given as inequalities that have to be satisfied by the various parameters of the closed-loop systems. A particularly significant lumped parameter group is $\rho := \frac{kk_1\theta}{\gamma_1\gamma_2(\gamma_1+\gamma_2)}$ that depends only on the plant and standalone *a*I controller parameters. To study the stabilizing effect of the appended proportional (P) component, we fix all the parameters related to the plant and standalone *a*I controller (hence ρ is fixed), and investigate the effect of the other controller parameters related to the appended proportional component. By examining the table in Fig. 4b, one can see that, compared to the standalone *a*I, the *a*PI controller of Class 1 with multiplicative (resp. degradation) inhibition enhances stability regardless of the exact values of κ (resp. δ) and n. This gives rise to a structural stability property: adding these types of proportional components guarantees better stability without having to fine-tune parameters.

In contrast, although the *a*PI controller of Class 1 with additive inhibition may enhance stability, special care has to be taken when tuning α. In fact, if α is tuned to be larger than a threshold given by $\alpha_{TH} := \frac{\gamma_1\gamma_2}{k_1} r\left[1 + (r/\kappa)^n\right]$, then stability is lost. Figure 4c elaborates more on this type of *a*PI controller. Three cases arise here. Firstly, if $\rho < 1$, that is, the standalone *a*I already stabilizes the closed-loop dynamics, then the (α, κ)- parameter space is split into a stable and unstable region. In the latter ($\alpha > \alpha_{TH}$), z_2 grows to infinity, and the output x_2 never reaches the desired setpoint $r = \mu/\theta$. Secondly, if $1 < \rho < 2$, that is the standalone *a*I is unstable, then the (α, κ)-parameter space is split into three regions: (1) a stable region, (2) an unstable region with a divergent response



$$S_{cl} = \left[\begin{array}{c|ccccc}
\mathcal{R} & \multicolumn{5}{c}{\mathcal{R}_c} \\
\mathcal{R}^1\ \mathcal{R}^2\ \cdots\ \mathcal{R}^K & \mathcal{R}_s & \mathcal{R}_q & \mathcal{R}_r & \mathcal{R}_a^+ & \mathcal{R}_a^- \\
 & 0 & 0 & 0 & 1 & -1 \\
 & & & & & \\
S & & & 0 & & \\
 & & & & & \\
 & 0 & 0 & 0 & 0 & 0 \\ \hline
0 & 0 & -1 & 1 & 0 & 0 \\
 & 1 & -1 & 0 & 0 & 0
\end{array}\right]
\begin{array}{ll}
 & \\
 & \\
\mathbf{X}_1 & \\
\mathbf{X}_2 & \\
\vdots & \mathbf{X} \\
\mathbf{X}_{L-1} & \\
\mathbf{X}_L & \\
\mathbf{Z}_1 & \mathbf{Z} \\
\mathbf{Z}_2 &
\end{array}$$

$$\lambda_{cl}(x, z) = \left[\begin{array}{c|ccccc}
\mathcal{R} & \multicolumn{5}{c}{\mathcal{R}_c} \\
\mathcal{R}^1\ \mathcal{R}^2\ \cdots\ \mathcal{R}^K & \mathcal{R}_s & \mathcal{R}_q & \mathcal{R}_r & \mathcal{R}_a^+ & \mathcal{R}_a^- \\
\lambda^T(x) & \theta x_L & \eta z_1 z_2 & g(\mu, x_L) & h^+(z_1, z_2, x_L) & h^-(z_2, x_1, x_L)
\end{array}\right]^T$$

Total Actuation Propensity: $h(z_1, z_2, x_1, x_L) := h^+(z_1, z_2, x_L) - h^-(z_2, x_1, x_L)$

aI	Reference Rxn: $\mathcal{R}_r$
	$\varnothing \xrightarrow{\mu} \mathbf{Z_1}$
	Actuation Rxn: $\mathcal{R}_a$
	$\mathbf{Z_1} \xrightarrow{k} \mathbf{Z_1} + \mathbf{X_1}$
	Sensing Rxn: $\mathcal{R}_s$
	$\mathbf{X_L} \xrightarrow{\theta} \mathbf{X_L} + \mathbf{Z_2}$
	Sequestration: $\mathcal{R}_q$
	$\mathbf{Z_1} + \mathbf{Z_2} \xrightarrow{\eta} \varnothing$
	Total Actuation Propensity $h(z_1, z_2, x_1, x_L)$
	kz_1
	Reference Propensity $g(\mu, x_L)$
	μ

	Inhibition Type	Actuation Rxns	Total Actuation Propensity h
aPI Class 1	Additive	$\mathcal{R}_a^+ : \quad \varnothing \longrightarrow \mathbf{X_1}$	$kz_1 + \frac{\alpha}{1+(x_L/\kappa)^n}$
	Multiplicative	$\mathcal{R}_a^+ : \quad \varnothing \longrightarrow \mathbf{X_1}$	$\frac{kz_1}{1+(x_L/\kappa)^n}$
	Degradation	$\mathcal{R}_a^+ : \quad \mathbf{Z_1} \longrightarrow \mathbf{Z_1} + \mathbf{X_1}$ $\mathcal{R}_a^- : \quad \mathbf{X_1} + \mathbf{X_L} \longrightarrow \mathbf{X_L}$	$kz_1 - \delta x_L \frac{x_1}{x_1+\kappa_1}$

	Inhibition Type	Actuation Rxns	Total Actuation Propensity h
aPI Class 2	Additive	$\mathcal{R}_a^+ : \quad \varnothing \longrightarrow \mathbf{X_1}$	$kz_1 + \frac{\alpha}{1+(z_2/\kappa)^n}$
	Multiplicative	$\mathcal{R}_a^+ : \quad \varnothing \longrightarrow \mathbf{X_1}$	$\frac{kz_1}{1+(z_2/\kappa)^n}$
	Degradation	$\mathcal{R}_a^+ : \quad \mathbf{Z_1} \longrightarrow \mathbf{Z_1} + \mathbf{X_1}$ $\mathcal{R}_a^- : \quad \mathbf{X_1} + \mathbf{Z_2} \longrightarrow \mathbf{Z_2}$	$kz_1 - \delta z_2 \frac{x_1}{x_1+\kappa_1}$

	Inhibition Type	Reference Rxn	Reference Propensity g
aPI Class 3	Additive	$\mathcal{R}_r : \quad \varnothing \xrightarrow{g(\mu, x_L)} \mathbf{Z_1}$	$\mu + \frac{\alpha}{1+(x_L/\kappa)^n}$
	Multiplicative	$\mathcal{R}_r : \quad \varnothing \xrightarrow{g(\mu, x_L)} \mathbf{Z_1}$	$\frac{\mu}{1+(x_L/\kappa)^n}$

similar to the previous scenario where $\rho < 1$, and (3) another unstable region where sustained oscillations emerge as depicted in the bottom plot of Fig. 4c. Note that the closer ρ is to 2, the narrower the stable region is. Thirdly, for $\rho > 2$, the stable region disappears and thus this *a*PI controller has no hope of stabilizing the dynamics without re-tuning the parameters related to the standalone *a*I controller (e.g., k and/or θ). Clearly, multiplicative and degradation inhibitions outperform additive inhibition if stability is a critical objective. To this end, Fig. 4d shows how the settling time and overshoot can be tuned by the controller parameters α, κ, and δ for additive, multiplicative, and degradation inhibitions, respectively. It is shown that with multiplicative and degradation inhibitions, one can simultaneously suppress oscillations (settling time) and remove

Fig. 3 Antithetic proportional-integral (*a*PI) feedback controllers. Three different classes of *a*PI controllers are designed by appending the standalone *a*I controller with three inhibitions. Three biologically relevant inhibition mechanisms are considered. Additive Inhibition: The inhibitor species produces the inhibited species separately at a decreasing rate. For instance, in the case of *a*PI Class 1 with additive inhibition, both controller species $\mathbf{Z_1}$ and output $\mathbf{X_L}$ produce the input $\mathbf{X_1}$ separately, but $\mathbf{Z_1}$ acts as an activator while $\mathbf{X_L}$ acts as a repressor. This separate inhibition can be modeled as the production of the input $\mathbf{X_1}$ as a positive actuation reaction $\mathcal{R}_a^+$ with an additive Hill-type propensity given by $h^+(z_1, x_L) = kz_1 + \frac{\alpha}{1+(x_L/\kappa)^n}$, where n, α and κ denote the Hill coefficient, maximal production rate and repression coefficient, respectively. This *a*PI is the closest control architecture to[33] and[35], since the P and I components are additive and separable (see Fig. 1c, d). Multiplicative inhibition: the inhibitor competes with an activator over a production reaction. In the case of *a*PI Class 1 with multiplicative inhibition, the output $\mathbf{X_L}$ inhibits the production of the input $\mathbf{X_1}$ by the controller species $\mathbf{Z_1}$. This can be modeled as the production of $\mathbf{X_1}$ with a multiplicative Hill-type propensity given by $h^+(z_1, x_L) = kz_1 \times \frac{1}{1+(x_L/\kappa)^n}$. Observe that in this scenario, the proportional (P) and integral (I) control actions are inseparable, and the actuation reaction $\mathcal{R}_a^+$ encodes both PI actions simultaneously. Degradation inhibition: the inhibitor invokes a negative actuation reaction that degrades the inhibited species. For instance, in the case of *a*PI Class 1 with degradation inhibition, the controller species $\mathbf{Z_1}$ produces the input $\mathbf{X_1}$ (positive actuation reaction $\mathcal{R}_a^+$), while the output $\mathbf{X_L}$ degrades it (negative actuation reaction $\mathcal{R}_a^-$). The dynamics can be captured by using a positive actuation with propensity $h^+(z_1) = kz_1$ and a negative actuation with propensity $h^-(x_1, x_L) = \delta x_L \frac{x_1}{x_1+\kappa_1}$. The total actuation propensity is defined as $h(z_1, x_1, x_L) := h^+(z_1) - h^-(x_1, x_L)$. The three classes with different inhibition mechanisms give rise to eight controllers that are compactly represented by the closed-loop stoichiometry matrix S_{cl} and propensity function λ_{cl} by choosing the suitable h and g functions from the tables.

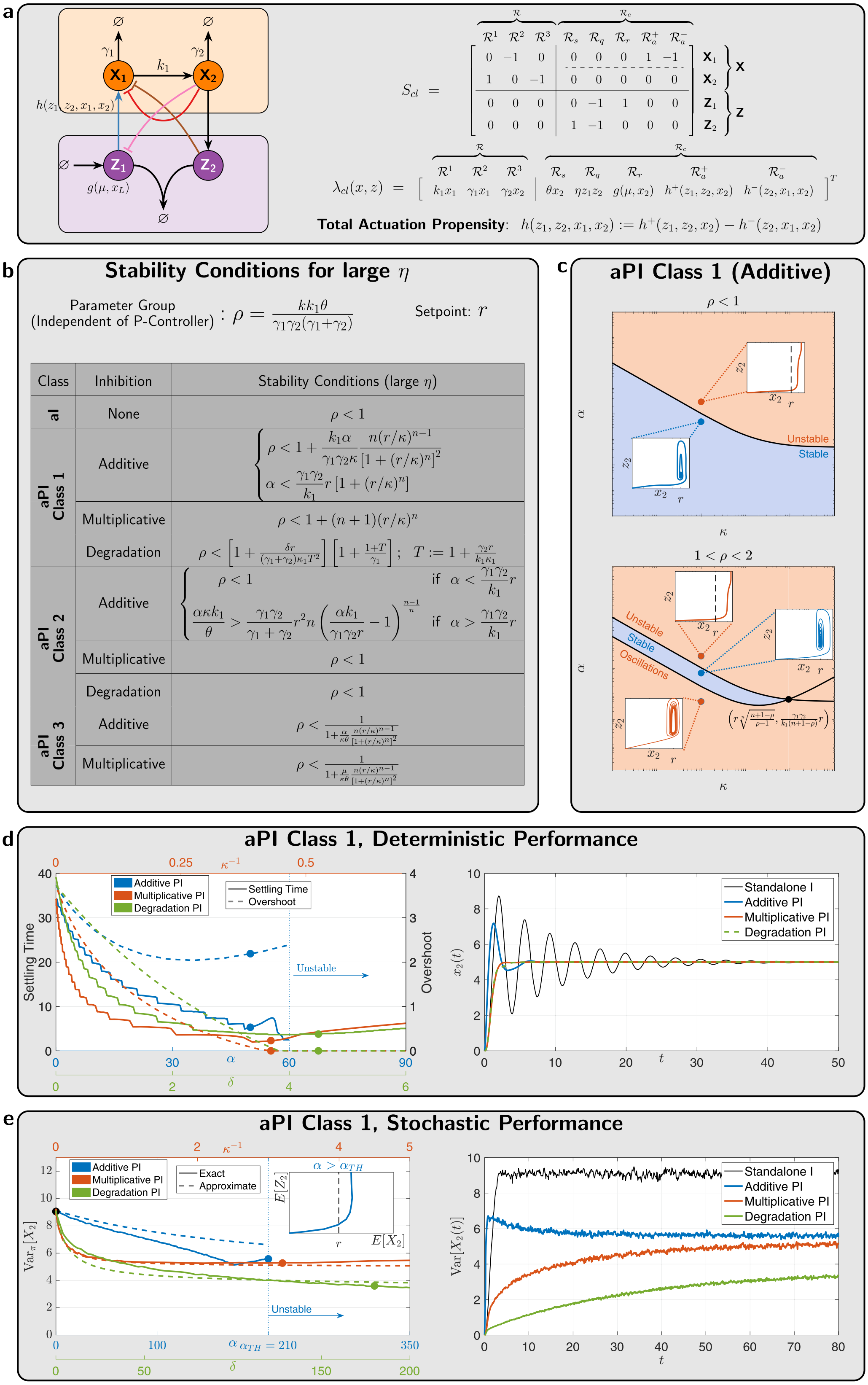
overshoots. In contrast, a proportional component with additive inhibition can suppress oscillations but is not capable of removing overshoots as illustrated in the simulations of Fig. 4d right panel. Furthermore, one can lose stability if α is increased above a threshold, as mentioned earlier. Nevertheless, for multiplicative and degradation inhibitions, increasing the controller parameters (κ^{-1}, δ) too much can make the response slower but can never destroy stability.

It can be shown that the other two classes (2 and 3) are undesirable in enhancing stability. For instance, observe that for Class 2, the stability conditions are the same as the standalone *a*I controller (in the limit as $\eta \longrightarrow \infty$) with the exception of the case of additive inhibition when $\alpha > \frac{\gamma_1\gamma_2}{k_1} r$. In this case, the inequality is structurally very different from all other stability conditions. In fact, the actuation via $\mathbf{Z_2}$ dominates $\mathbf{Z_1}$, and hence $\mathbf{Z_2}$ becomes responsible for the integral (I) action instead of $\mathbf{Z_1}$. The detailed analysis of this network is not within the scope of this paper, and is left for future work. Finally, *a*PI controllers of Class 3 deteriorate the stability margin, since the right-hand side of the inequalities is strictly less than one. However, this class of controllers can be useful for slow plants if the objective is to speed up the dynamics.

Stochastic analysis of the *a*PI controllers: RPA and stationary variance. We now investigate the effect of the *a*PI controllers on the stationary (steady-state) behavior of the output species $\mathbf{X_L}$ in the stochastic setting. First, we examine the stationary expectation $\mathbb{E}_\pi[X_L]$. It is shown in Supplementary Information Section 8.1 that for RPA to be achieved in the stochastic setting, the function g has to be affine in x_L. Hence only *a*PI controllers of Class 1 and 2 achieve RPA in the stochastic setting with $\mathbb{E}_\pi[X_L] = \mu/\theta =: r$. Next, we examine the stationary variance $\mathrm{Var}_\pi[X_L]$ to demonstrate that *a*PI controllers with degradation inhibition excel in reducing cell-to-cell variability. In Supplementary Information Section 8.2, we develop a tailored moment-closure technique based on[33] to derive an analytic closed formula for the stationary variance $\mathrm{Var}_\pi[X_L]$. Unfortunately, a general analysis for an arbitrary plant cannot be done. As a case study, we consider again the particular plant given in Fig. 4a in feedback with the *a*PI controller of Class 1 with the various inhibition mechanisms. Note, however, that the analysis can be generalized to any (affine-linear) plant with mono-molecular reactions (see Supplementary Information Section 8.2). The resulting formula is given in Supplementary Table 2 that shows that the proportional controller decreases $\mathrm{Var}_\pi[X_2]$. Figure 4e demonstrates this stationary variance reduction via simulations and the approximate formula. Unlike additive inhibition, multiplicative and degradation inhibitions provide a structural property of decreasing the $\mathrm{Var}_\pi[X_2]$ without risking the loss of ergodicity (similar to the deterministic setting).

Antithetic proportional-integral-derivative feedback (*a*PID) control topologies. In this section, we append a derivative (D) control action to the *a*PI (Class 1) controller of Fig. 3 to obtain an array of *a*PID controllers depicted in Fig. 5. The proposed *a*PID controllers range from simple second order (involving only two controller species $\mathbf{Z_1}$ and $\mathbf{Z_2}$) up to fourth order (involving four controller species $\mathbf{Z_1}$ to $\mathbf{Z_4}$). Furthermore, the various controllers are categorized as two types: N-type and P-type. N-type (negative feedback) controllers are usually suitable for plants with positive gain (increasing the input yields an increase in the output), while P-type (positive feedback) controllers are usually suitable for plants with negative gains. This ensures that the overall control loops realize negative feedback. Note that one can easily construct hybrid PN-type controllers, where the individual P, I, and D components have different P/N-types. This hybrid design is shown to be very useful for certain plants (see Fig. 7d for example).

We begin with the N-type second-order design (first row of Fig. 5) whose main advantage is its simplicity. Note that the rationale behind the various P-type designs is similar. Intuitively, the antithetic integral motif is cascaded with an incoherent feedforward loop (IFFL) to yield a PID architecture whose P, I and D components are inseparable as described in Fig. 1e. More precisely, the output $\mathbf{X_L}$ directly inhibits the input $\mathbf{X_1}$ and simultaneously produces it via the intermediate species $\mathbf{Z_1}$. As a result, $\mathbf{Z_1}$ simultaneously plays the role of both an intermediate species for the IFFL and the integral control action. It is shown in Supplementary Information Section 1.2.1 that this simple design embeds a low-pass-filtered PID controller. The N-type third-order design (second row of Fig. 5) involves one additional controller species $\mathbf{Z_3}$ to realize an IFFL that is disjoint from the antithetic motif. This yields an inseparable PD component appended to the separate I controller. It is shown in Supplementary Information Section 1.2.2 that this design embeds a low-pass-filtered PD + I controller when η is large enough. In contrast, the N-type fourth-order design (third row of Fig. 5) involves two additional controller species $\mathbf{Z_3}$ and $\mathbf{Z_4}$ to realize a completely separable PID control architecture. It is shown in Supplementary Information Sections 1.2.3 and 1.2.4 that this design embeds a PI + low-pass-filtered D controller when η and η_0 are large enough. The key idea behind mathematically realizing the derivative component here is fundamentally different from the previous two designs. This



$$S_{cl} = \begin{array}{c} \\ \\ \end{array}\overbrace{\begin{array}{ccc}\mathcal{R}^1 & \mathcal{R}^2 & \mathcal{R}^3\end{array}}^{\mathcal{R}}\ \overbrace{\begin{array}{ccccc}\mathcal{R}_s & \mathcal{R}_q & \mathcal{R}_r & \mathcal{R}_a^+ & \mathcal{R}_a^-\end{array}}^{\mathcal{R}_c}$$

$$S_{cl} = \left[\begin{array}{ccc|ccccc} 0 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \end{array}\right] \begin{array}{l} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{array} \begin{array}{l} \left.\begin{array}{l} \\ \\ \end{array}\right\} \mathbf{X} \\ \left.\begin{array}{l} \\ \\ \end{array}\right\} \mathbf{Z} \end{array}$$

$$\lambda_{cl}(x, z) = \Big[\ \overbrace{\begin{array}{ccc}\mathcal{R}^1 & \mathcal{R}^2 & \mathcal{R}^3 \\ k_1 x_1 & \gamma_1 x_1 & \gamma_2 x_2\end{array}}^{\mathcal{R}}\ \Big|\ \overbrace{\begin{array}{ccccc}\mathcal{R}_s & \mathcal{R}_q & \mathcal{R}_r & \mathcal{R}_a^+ & \mathcal{R}_a^- \\ \theta x_2 & \eta z_1 z_2 & g(\mu, x_2) & h^+(z_1, z_2, x_2) & h^-(z_2, x_1, x_2)\end{array}}^{\mathcal{R}_c}\ \Big]^T$$

Total Actuation Propensity: $h(z_1, z_2, x_1, x_2) := h^+(z_1, z_2, x_2) - h^-(z_2, x_1, x_2)$

Stability Conditions for large η

Parameter Group (Independent of P-Controller) : $\rho = \frac{kk_1\theta}{\gamma_1\gamma_2(\gamma_1+\gamma_2)}$ Setpoint: r

Class	Inhibition	Stability Conditions (large η)
aI	None	$\rho < 1$
aPI Class 1	Additive	$\begin{cases} \rho < 1 + \frac{k_1\alpha}{\gamma_1\gamma_2\kappa}\frac{n(r/\kappa)^{n-1}}{[1+(r/\kappa)^n]^2} \\ \alpha < \frac{\gamma_1\gamma_2}{k_1}r\left[1+(r/\kappa)^n\right] \end{cases}$
	Multiplicative	$\rho < 1 + (n+1)(r/\kappa)^n$
	Degradation	$\rho < \left[1 + \frac{\delta r}{(\gamma_1+\gamma_2)\kappa_1 T^2}\right]\left[1 + \frac{1+T}{\gamma_1}\right]; \quad T := 1 + \frac{\gamma_2 r}{k_1\kappa_1}$
aPI Class 2	Additive	$\begin{cases} \rho < 1 & \text{if} \quad \alpha < \frac{\gamma_1\gamma_2}{k_1}r \\ \frac{\alpha\kappa k_1}{\theta} > \frac{\gamma_1\gamma_2}{\gamma_1+\gamma_2}r^2 n\left(\frac{\alpha k_1}{\gamma_1\gamma_2 r} - 1\right)^{\frac{n-1}{n}} & \text{if} \quad \alpha > \frac{\gamma_1\gamma_2}{k_1}r \end{cases}$
	Multiplicative	$\rho < 1$
	Degradation	$\rho < 1$
aPI Class 3	Additive	$\rho < \frac{1}{1+\frac{\alpha}{\kappa\theta}\frac{n(r/\kappa)^{n-1}}{[1+(r/\kappa)^n]^2}}$
	Multiplicative	$\rho < \frac{1}{1+\frac{\mu}{\kappa\theta}\frac{n(r/\kappa)^{n-1}}{[1+(r/\kappa)^n]^2}}$

controller realizes an "antithetic differentiator", whereby the antithetic motif feeds back into itself: $\mathbf{Z}_3$ feeds back into $\mathbf{Z}_4$ via the rate function $g(z_3, x_L)$. In fact, this idea is inspired by a mathematical trick in control theory (see Supplementary Information Section 7) that basically exploits an integral controller, in feedback with itself to implement a low-pass-filtered derivative controller. For this fourth-order design, the derivative action can be achieved in two ways. One way is by mutually producing $\mathbf{Z}_4$ and $\mathbf{X}_1$ at a rate proportional to $g(z_3, x_L)$ such that g is monotonically increasing (resp. decreasing) in z_3 (resp. x_L). This implementation

Fig. 4 Performance of *a*PI feedback controllers. a Gene expression network controlled by *a*PI controllers. **b** Inequalities that need to be respected by the various controllers (with η being large enough) to guarantee closed-loop stability in the deterministic setting. Multiplicative and degradation inhibition mechanisms exhibit superior structural stability properties. **c** *a*PI controllers of Class 1 with an additive inhibition mechanism, exhibit different stability properties for different ranges of the parameter group ρ (that depends solely on the plant and the standalone *a*I controller). In particular, for $\rho < 2$, the additive proportional control action can stabilize the dynamics, while for $\rho > 2$, it cannot stabilize without re-tuning the integral component. **d** Settling time and overshoot of the output ($\mathbf{X_2}$) response as a function of controller parameters that are related to the appended proportional components. Multiplicative and degradation inhibition mechanisms are capable of ameliorating the performance without risking instability as opposed to the additive inhibition mechanism. **e** Reduction of the output stationary variance (cell-to-cell variability) with *a*PI controllers. The *a*PI controllers of Class 1 with all three inhibition mechanisms are capable of reducing the stationary variance of the output. This is demonstrated here via the simulations and the approximate formula shown in Supplementary Table 2 as well. For additive inhibition, α has a threshold value α_{TH} above which ergodicity is lost similar to the deterministic setting. Furthermore, observe that for values of α that are close to α_{TH}, the analytic approximation is less accurate. In contrast, the multiplicative and degradation mechanisms are capable of reducing the variance without the risk of losing ergodicity, and the analytic approximation remains accurate. The degradation inhibition, demonstrating superiority, is capable of reducing the stationary variance to levels even lower than the mean $\mathbb{E}_\pi[X_2] = 5$—a level that is not achievable by the other inhibition mechanisms. The numerical values of all the parameters can be found in Supplementary Information Section 11.

is treated separately in Supplementary Information Section 1.2.3. The other way is by producing $\mathbf{Z_4}$ while degrading $\mathbf{X_1}$ at a mutual rate of $g(z_3, x_L)$ such that g is monotonically increasing in both z_3 and x_L. This implementation is treated separately in Supplementary Information Section 1.2.4. Both designs have the same underlying PID control structure, but one might be easier to experimentally implement than the other.

Deterministic analysis and properties of the various *a*PID controllers. It is straightforward to show that the setpoint for the second-order design is given by $\bar{x}_L = \frac{\mu}{\theta-\beta}$ with the requirement that $\beta < \theta$; whereas the set-points for both higher-order designs are given by $\bar{x}_L = \frac{\mu}{\theta}$. Furthermore, the effective PID gains, denoted by (K_P, K_I, K_D), and cutoff frequency ω of the embedded low-pass filter for each of the proposed *a*PID controllers can be designed by tuning the various biomolecular parameters: $\beta, \eta, \eta_0, \gamma_0, \mu_0$ and the parameters of the propensity functions h and g. These functions depend on the specific implementation adopted. In particular, they can be picked in a similar fashion to the functions used to realize the three inhibition mechanisms (additive, multiplicative, or degradation) of the *a*PI controllers shown in Fig. 3. In the subsequent examples, we use degradation inhibitions, but the other mechanisms can also be used.

Next, we demonstrate various properties of the proposed controller designs in the deterministic setting and highlight the benefits of the added complexity. The mappings between the effective PID parameters (K_P, K_I, K_D, ω) and the biomolecular parameters $(\mu, \theta, \eta, \beta, \gamma_0, \eta_0, \ldots)$ are given in Supplementary Information Section 4 for each controller. It is fairly straightforward to go back and forth between the two parameter spaces. For control analysis, these mappings can compute the various PID parameters from the biomolecular parameters, whereas for control design, these mappings can compute the various biomolecular parameters that achieve some desired PID gains and cutoff frequency. As a result, one can use existing methods in the literature (e.g.[49]) to carry out the controller tuning in the PID parameter space, and then map them to the actual biomolecular parameter space. Nevertheless, it is of critical importance to note that different controllers yield different coverage over the PID parameter space. For instance, for the fourth-order design, there are enough biomolecular degrees of freedom to design any desired positive $(K_P, K_I, K_D, \omega) \in \mathbb{R}_+^4$. The lower the order of the controller, the fewer the biomolecular degrees of freedom, and hence the more constrained the coverage in the PID parameter space. For instance, for the third-order design, the achievable PID parameters are constrained to satisfy $K_P \le K_D \omega$. For the second-order design, the constraint becomes even stricter. The details are all rigorously reported in Supplementary Information Section 4.

Flexibility of *a*PID controllers. We first show the limitation of *a*PI controllers, and then demonstrate the flexibility that comes with an added derivative component. More precisely, we show that the *a*PI controller alone is incapable of speeding up the response beyond a certain threshold without incurring oscillations—a limitation that a full *a*PID controller overcomes. We also show that the higher-order *a*PID controllers exhibit more flexibility in shaping the transient response. Consider the controlled gene expression network depicted in Fig. 6a where the ODEs of the various controllers are shown to explicitly specify the adopted propensity functions h and g. In this example, we consider both the P and D components acting on the input species $\mathbf{X_1}$ as negative actuation via degradation reactions. In fact, inhibition with degradation is used whenever possible because it outperforms the other inhibition mechanisms by achieving better stability properties and dramatically reducing the stationary variance (see Fig. 4). We start by highlighting the fundamental limitation of *a*PI controllers alone (without a D) in Fig. 6b. Using simple root-locus arguments (see Supplementary Information Section 5.1), it is shown that as the proportional gain K_P is increased, two complex eigenvalues of the linearized dynamics around the fixed point approach a vertical asymptote intersecting the real axis at $-\frac{\gamma_1+\gamma_2}{2}$, while one real eigenvalue approaches the origin (due to integral control). This is numerically demonstrated in the two root-locus plots of Fig. 6b, where $K_P \approx \delta$ (for a sufficiently small κ_1). Clearly, the asymptotic limit is independent of all other parameters, including the integral gain K_I. This analysis highlights a fundamental limitation of the *a*PI controller, because no matter how we tune K_P and K_I, two of the eigenvalues are constrained to remain close to the imaginary axis when γ_1 and γ_2 are small. In the time domain, these constraints impose either a slowly rising response or a faster-rising response but with lightly damped oscillations, as illustrated in the simulation examples of Fig. 6b. These limitations can be mitigated by appending a derivative control action via the various *a*PID controllers. To demonstrate this, we consider a design problem where the end goal is to achieve a fast response without oscillations and with minimal overshoot. This can be achieved by placing the eigenvalues far to the left on the real axis. Hence the design problem can essentially be translated to the following objective: place the four most dominant eigenvalues (or poles) at $s = -a$ where $a > 0$ and make a sufficiently large. The design steps start by first (1) deciding where to place the poles $s = -a$ for some desired a, then (2) computing the PID parameters so as to place the poles as desired; this can be achieved using Supplementary Equation (46), and finally (3) mapping the PID parameters to the actual biomolecular parameters using the formulas in Supplementary Information Section 4. This is

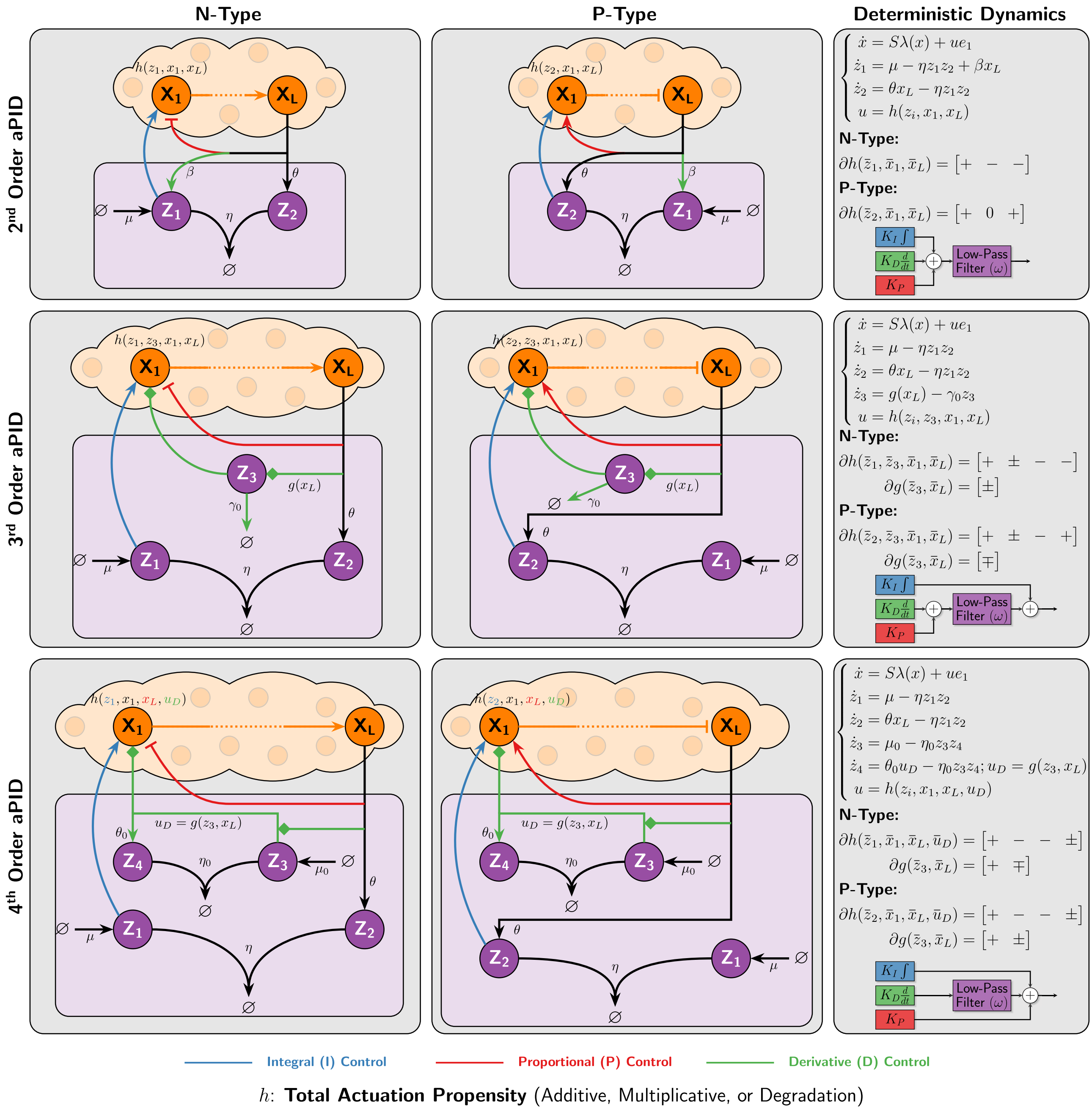
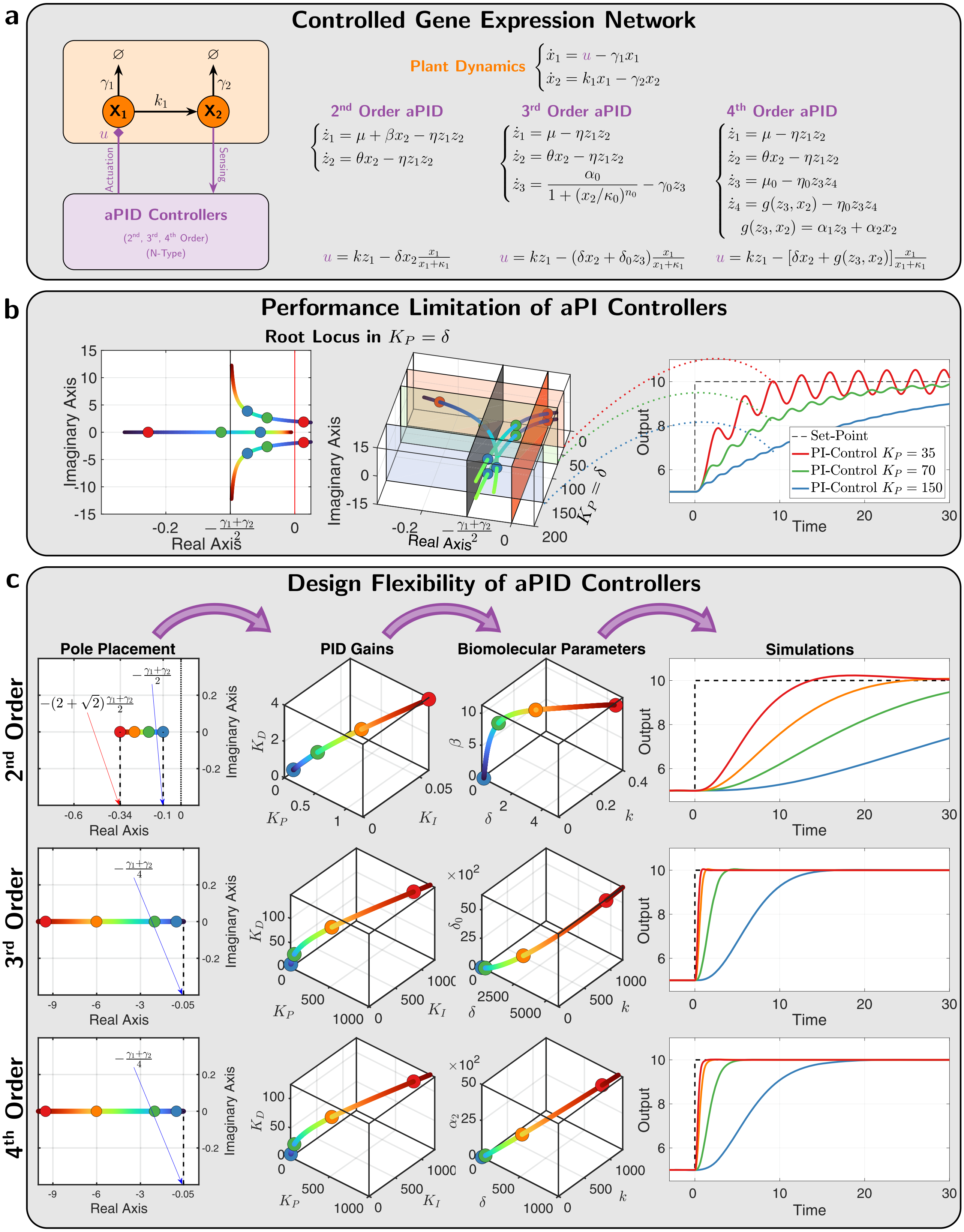

N-Type
P-Type
Deterministic Dynamics
2nd Order aPID
3rd Order aPID
4th Order aPID
$h(z_1, x_1, x_L)$
$h(z_2, x_1, x_L)$
X_1
X_L
Z_1
Z_2
Z_3
Z_4
β
θ
μ
η
$\varnothing$
$\dot{x} = S\lambda(x) + ue_1$
$\dot{z}_1 = \mu - \eta z_1 z_2 + \beta x_L$
$\dot{z}_2 = \theta x_L - \eta z_1 z_2$
$u = h(z_i, x_1, x_L)$
N-Type:
$\partial h(\bar{z}_1, \bar{x}_1, \bar{x}_L) = [+ \ - \ -]$
P-Type:
$\partial h(\bar{z}_2, \bar{x}_1, \bar{x}_L) = [+ \ 0 \ +]$
$K_I \int$
$K_D \frac{d}{dt}$
K_P
Low-Pass Filter (ω)
$h(z_1, z_3, x_1, x_L)$
$h(z_2, z_3, x_1, x_L)$
$g(x_L)$
γ_0
$\dot{x} = S\lambda(x) + ue_1$
$\dot{z}_1 = \mu - \eta z_1 z_2$
$\dot{z}_2 = \theta x_L - \eta z_1 z_2$
$\dot{z}_3 = g(x_L) - \gamma_0 z_3$
$u = h(z_i, z_3, x_1, x_L)$
N-Type:
$\partial h(\bar{z}_1, \bar{z}_3, \bar{x}_1, \bar{x}_L) = [+ \ \pm \ - \ -]$
$\partial g(\bar{z}_3, \bar{x}_L) = [\pm]$
P-Type:
$\partial h(\bar{z}_2, \bar{z}_3, \bar{x}_1, \bar{x}_L) = [+ \ \pm \ - \ +]$
$\partial g(\bar{z}_3, \bar{x}_L) = [\mp]$
$K_I \int$
$K_D \frac{d}{dt}$
K_P
Low-Pass Filter (ω)
$h(z_1, x_1, x_L, u_D)$
$h(z_2, x_1, x_L, u_D)$
$u_D = g(z_3, x_L)$
θ_0
η_0
μ_0
$\dot{x} = S\lambda(x) + ue_1$
$\dot{z}_1 = \mu - \eta z_1 z_2$
$\dot{z}_2 = \theta x_L - \eta z_1 z_2$
$\dot{z}_3 = \mu_0 - \eta_0 z_3 z_4$
$\dot{z}_4 = \theta_0 u_D - \eta_0 z_3 z_4; u_D = g(z_3, x_L)$
$u = h(z_i, x_1, x_L, u_D)$
N-Type:
$\partial h(\bar{z}_1, \bar{x}_1, \bar{x}_L, \bar{u}_D) = [+ \ - \ - \ \pm]$
$\partial g(\bar{z}_3, \bar{x}_L) = [+ \ \mp]$
P-Type:
$\partial h(\bar{z}_2, \bar{x}_1, \bar{x}_L, \bar{u}_D) = [+ \ - \ - \ \pm]$
$\partial g(\bar{z}_3, \bar{x}_L) = [+ \ \pm]$
$K_I \int$
$K_D \frac{d}{dt}$
K_P
Low-Pass Filter (ω)
Integral (I) Control
Proportional (P) Control
Derivative (D) Control
h: Total Actuation Propensity (Additive, Multiplicative, or Degradation)

Fig. 5 Antithetic proportional-integral-derivative (*a*PID) feedback controllers. N-type (negative feedback) controllers are usually suitable for plants with positive gain (increasing the input yields an increase in the output), while P-type (positive feedback) controllers are usually suitable for plants with negative gains. The order of the controllers indicates the number of controller species $\mathbf{Z_i}$. The second-order *a*PID controller has the simplest design where no additional species are added to the *a*PI design, and only one reaction is added to produce $\mathbf{Z_1}$ catalytically from the output $\mathbf{X_L}$ at a rate $\beta < \theta$. The third-order *a*PID controller adds a single species to the *a*PI design. This intermediate species $\mathbf{Z_3}$ is produced by the output $\mathbf{X_L}$ and actuates the input species $\mathbf{X_1}$. These actions (indicated by the diamonds) are allowed to be either activations or inhibitions. Finally, the fourth-order *a*PID controller adds two species to the *a*PI design. These two species form an "antithetic differentiator" where $\mathbf{Z_3}$ is constitutively produced at a rate μ_0 and participates with $\mathbf{Z_4}$ in a sequestration reaction with a rate η_0. For the N-type design, the derivative action enters the plant either by mutually producing $\mathbf{Z_4}$ and $\mathbf{X_1}$ at a rate proportional to $g(z_3, x_L)$ (see Supplementary Fig. 4) such that g is monotonically increasing (resp. decreasing) in z_3 (resp. x_L), or by producing $\mathbf{Z_4}$ while degrading $\mathbf{X_1}$ at a rate proportional to $g(z_3, x_L)$ (see Supplementary Fig. 5) such that g is monotonically increasing in both z_3 and x_L. Intuitively, the second and third-order *a*PID controllers mathematically realize a derivative action using an incoherent feedforward loop from the output $\mathbf{X_L}$ to the input $\mathbf{X_1}$ via $\mathbf{Z_1}$ and $\mathbf{Z_3}$, respectively, whereas the fourth-order *a*PID controller realizes a derivative action by placing an additional antithetic integral motif in feedback with the plant and itself ($\mathbf{Z_3}$ feeds back into $\mathbf{Z_4}$).

pictorially demonstrated in Fig. 6c for each *a*PID controller. However, it is shown in Supplementary Information Section 5.2 that the second-order *a*PID imposes a lower bound on the achievable poles given by $-(2+\sqrt{2})\frac{\gamma_1+\gamma_2}{2}$ as demonstrated in Fig. 6c. As a result, with a second-order *a*PID, the performance can be made better than the *a*PI controller; however, the performance is also limited and cannot be made faster than a threshold, dictated by γ_1 and γ_2, without causing overshoots and/or oscillations. In contrast, it is also shown in Supplementary Information Section 5.2 that the third- and fourth-order *a*PID controllers can make *a* as large as desired without



any theoretical upper bound. This means that the added complexity of the higher-order controllers is capable of shaping the response of the gene expression network freely and as fast as desired with no overshoots nor oscillations. This is also demonstrated in the simulations depicted in Fig. 6c.

Deterministic performance of *a*PID control of complex networks. We consider a more complex plant to be controlled. The plant, comprised of $L = 6$ species, is depicted in Fig. 7a where $\mathbf{X_i}$ degrades at a rate γ_i and catalytically produces $\mathbf{X_{i+1}}$ at a rate k_i. Furthermore, the output species $\mathbf{X_6}$ feeds back into $\mathbf{X_2}$ by

Fig. 6 *a*PID control of a gene expression network. a Closed-loop dynamics. A gene expression network is controlled by the various N-type *a*PID controllers of Fig. 5. The deterministic dynamics and the overall control action u are shown here for each controller to explicitly specify the adopted propensity functions h and g in this example. **b** Fundamental limitation of *a*PI controllers. Without the derivative component, the response cannot be sped up beyond a certain threshold without inflicting oscillations. The left and middle plots demonstrate the same root locus of the linearized dynamics as the proportional gain K_P is increased. The left plot depicts the complex plane, while the middle plot explicitly shows the complex plane together with the values of the proportional gain K_P which is shown to be approximately equal to δ. These plots verify that two eigenvalues are confined within a small region close to the imaginary axis when γ_1 and γ_2 are small, and thus imposing a limitation on the achievable performance as demonstrated in the simulations shown in the right plot. **c** Design flexibility offered by derivative control actions. Exploiting all the components of the full *a*PID controllers offers more flexibility in achieving superior performance compared to the *a*PI controllers. This panel shows the steps of a pole placement, control design problem where the four dominant poles are placed on the real axis of the left-half plane to ensure a stable and non-oscillating response with minimal overshoot. The second-order *a*PID exhibits a restriction on how far to the left the poles can be placed; whereas the higher-order controllers can place the poles arbitrarily as far to the left as desired and thus achieving a response that is as fast as desired without overshoots nor oscillations. The design problem starts by picking the poles, then computing the PID gains (shown here) and cutoff frequency (not shown here), and finally computing the actual biomolecular parameters that allow us to obtain the nonlinear simulations to the right.

catalytically degrading it at a rate γ_F. This plant is adopted from[35]; however, to challenge our controllers in a way that demonstrates their features, the feedback degradation rate γ_F is chosen to be larger than that reported in[35], which yields a plant that is unstable when operating in an open loop, as shown in Fig. 7a. In fact, the root locus in the integral gain $K_I \approx k$ (for large η) demonstrates that this plant cannot be stabilized with a standalone *a*I controller, that is no matter how we tune k, the response will remain unstable. It was shown in[35] that, for this plant, the P control is not useful. This is the case because the proportional gain K_P was restricted to have a positive value. One of the nice features of our proposed second- and third-order *a*PID controllers is their ability to achieve negative proportional gains K_P (see Supplementary Equations (27) and (32)) without having to rewire, that is without switching topologically from N-type to P-type. This is a consequence of the inseparability of the P component from other components (I and D for the second order, and D for the third order). In Fig. 7b, c we show that, for this plant, tuning the proportional gain K_P to be negative is critical to achieving high performance, whereby oscillations and overshoots are almost completely removed while maintaining a fast response. This is demonstrated using the intensity plots of a performance index that quantifies the overshoot, settling time, and rise time of the output response over a range of the relevant biomolecular controller parameters. With the completely separable fourth-order *a*PID, the gains cannot be tuned to be negative; however, one can always switch between N-type and P-type topologies or even resort to hybrid designs where different PID components are of different P/N-types. Indeed, Fig. 7d shows that by using a fourth-order hybrid *a*PID controller, high performance can be attained.

To demonstrate the effectiveness of *a*PID control of high dimensional plants, we carry out a simulation study of cholesterol control in the plasma using the second-order *a*PID controller. The mathematical whole-body model of cholesterol metabolism is adopted from[50] that involves 34 state variables (species). The simulation description and results can be found in Supplementary Information Section 6 that demonstrates that *a*PID controllers are also capable of achieving high performance for high dimensional systems.

Effect of derivative control on the stationary variance. In Supplementary Information Section 9, we examine the effect of the derivative component in the various *a*PID controllers on the cell-to-cell variability (e.g., stationary variance). We consider two plants: the gene expression network of Fig. 6a and the six-species network of Fig. 7a. The results in Supplementary Information Section 9 demonstrate that, for both plants, the third- and fourth-order *a*PID controllers are capable of reducing the stationary variance, whereas the second-order *a*PID increases it. Unfortunately, moment-closure techniques similar to the one used for the *a*PI controllers failed to approximate the stationary variance here. Hence, the conclusion here is based on simulations only, but seems to be consistent. Further stochastic analysis similar to[51] that exploits linear noise approximations is left for future work.

Genetic circuit designs. Here we propose and describe a particular genetic design in *Escherichia coli* that realizes the third-order *a*PID controller topology presented in Fig. 5 (See Supplementary Information Section 10 for another genetic design of the second-order *a*PID controller). We also perform numerical simulations using biologically realistic parameters to demonstrate the effectiveness of the controllers in ameliorating the dynamic performance. The genetic circuit is depicted in Fig. 8a where the controller circuit augments the I-control module (in blue), which is based on[9], with additional circuitry to implement additional P and D controls (in red and green). The only difference between our I-control module and that of[9] is the choice of the promoter P_{RM} driving the expression of the anti-σ factor (rsiW) that is activated by the transcription factor cI acting as a dual activator for P_{RM} and repressor for P_R (see[52,53]). The P and D control modules are implemented via the Mflon protease which is capable of degrading the input species $\mathbf{X_1}$. The additional disturbance circuit (in yellow) serves as a source of external perturbation to the closed-loop circuit by degrading the regulated output $\mathbf{X_2}$. The set of ODEs describing the deterministic dynamics is also shown Fig. 8a and the various parameters are chosen to reflect biologically realistic regimes and account for controller species dilution δ_c (see Supplementary Information Section 11). Figure 8b shows the simulation results for I, PI, and PID control. The responses are shown for a step change of setpoint μ/θ, which is tunable with HSL[9], at $t = 8$ h, and for a step change of disturbance Δ, which is tunable with aTc, at $t = 16$ h. The simulations demonstrate that the full PID controller is capable of dramatically enhancing the stability and performance by not only shaping the transient dynamics but also reducing the steady-state error that can be incurred by the dilution effect (see[9,25,27]).

Experimental demonstration—Cyberloop implementation. To validate the performance benefits of the proposed *a*PID controllers, we implemented and tested our fourth-order PID controller (presented in Fig. 5) in a hybrid in vivo–in silico optogenetic platform[54] using the rapid prototyping "Cyberloop" framework developed in[32]. This platform provides an interface (at single-cell resolution) between real biological circuits (in vivo) in cells placed under the microscope and stochastic computer simulation (in silico) of controllers via light stimulation and

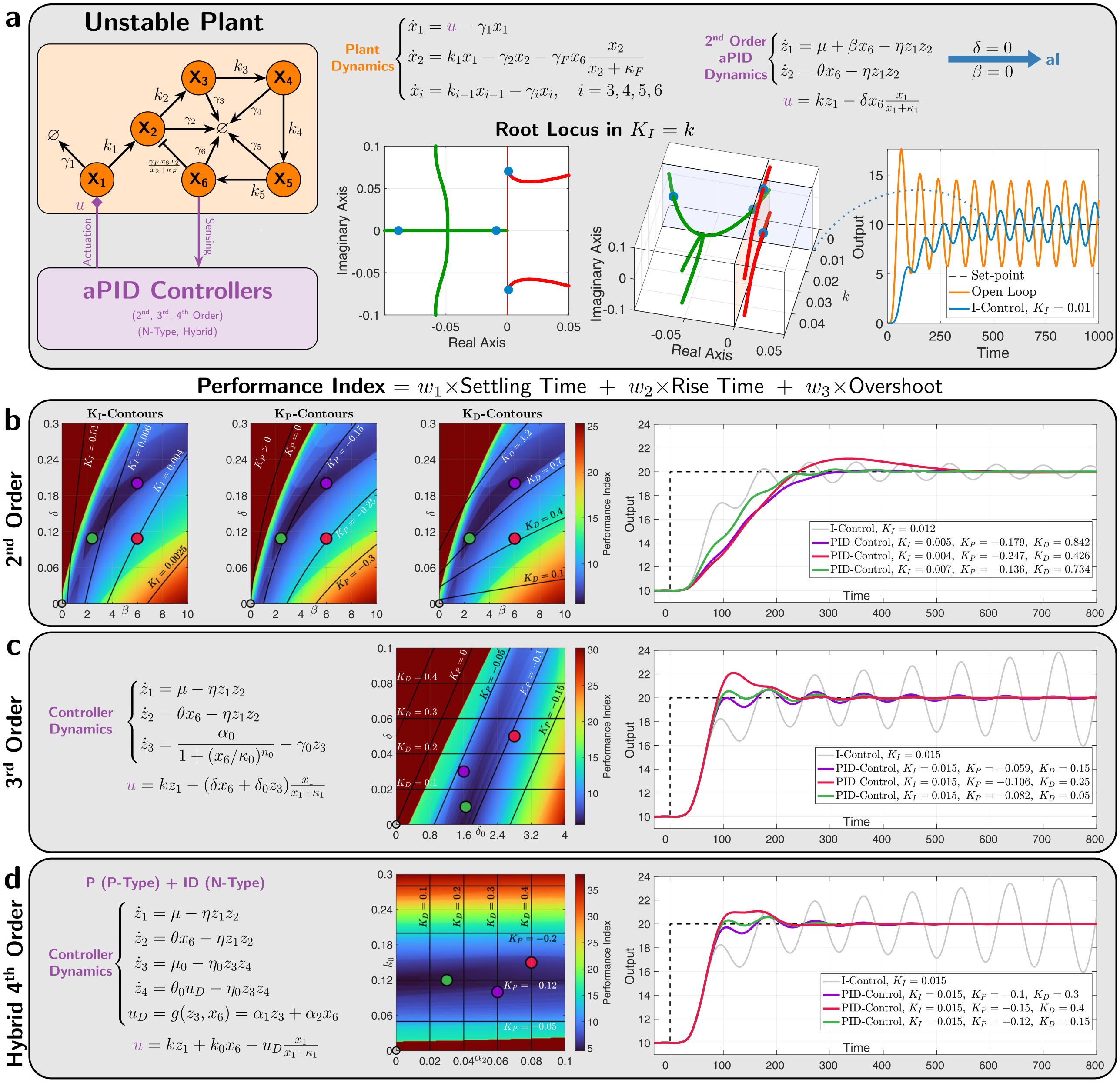


Fig. 7 *a*PID control of an unstable and more complex plant. a Plant description. The plant considered here involves $L = 6$ species and embeds a negative feedback from the output $\mathbf{Z_6}$ to $\mathbf{X_2}$ via an active degradation reaction. The underlying deterministic dynamics of the plant and the second-order *a*PID controller are shown in this panel. It is demonstrated that the open loop is unstable (orange response), and integral control alone cannot stabilize the dynamics since two eigenvalues carry on a positive real part for any $k \geq 0$. **b–d** Performance of the various *a*PID controllers. The intensity plots show the performance index over a range of biomolecular parameter values. These plots are overlaid with contours where the PID gains K_P, K_I or K_D are constant. For the third and fourth-order *a*PID in (**c**) and (**d**), $K_I \approx k$ and thus can be tuned separately with k that is held constant throughout this figure. For the fourth-order *a*PID in (**d**), the K_P- and K_D-contours are orthogonal to the k_0- and α_2-axes, respectively, and hence can also be tuned separately. For the third-order *a*PID in (**c**), the K_D-contours are orthogonal to the δ-axis and hence K_P can be tuned separately with δ_0, whereas the inseparability of the PD components forces the K_P-contours to be oblique, and thus δ tunes both K_P and K_D simultaneously. Finally, for the second-order *a*PID in (**b**), all three contours are not orthogonal to the axes and, as a result, all three PID gains have to be mutually tuned by the biomolecular parameters. This is due to the inseparability of all PID components. Note that each set of contours is displayed on a separate intensity plot here for clarity. Observe that the optimal performance for each controller is located in the dark blue regions where the proportional gains K_P are negative. Three different examples, red, green, and purple (along with the unstable standalone *a*I control in gray), are picked to demonstrate the achievable high performances depicted in the response plots to the right. For the second and third-order *a*PID, negative K_P can be achieved by properly tuning the biomolecular parameters without having to switch the topology from N-type to P-type. However, for the (separable) fourth-order *a*PID controller, a hybrid design with N-type ID and P-type P can also achieve a negative K_P that is critical for controlling this plant.

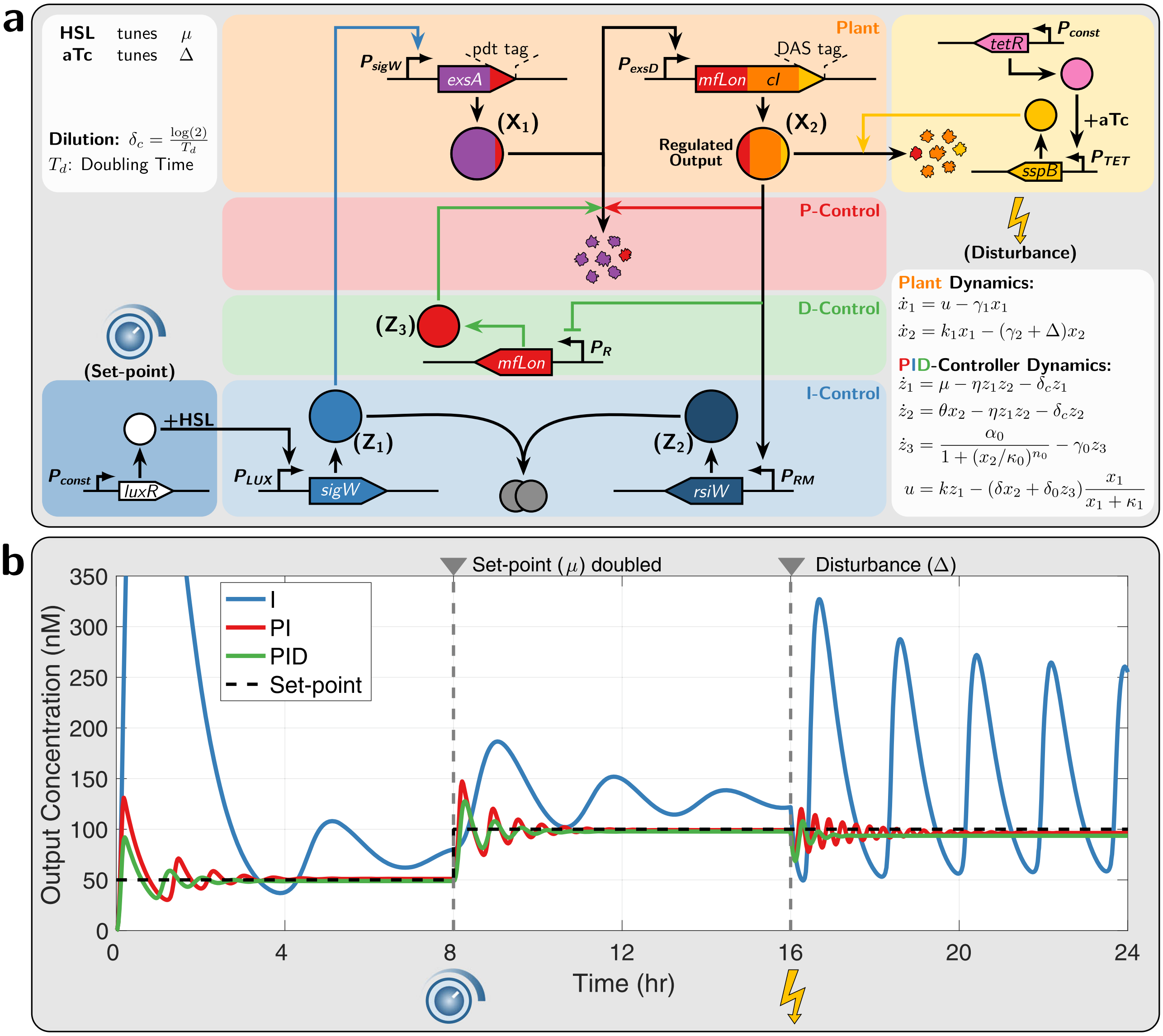


Fig. 8 A genetic implementation of the third-order *a*PID controller. a Circuit design. The genetic closed-loop circuit is comprised of the plant (in orange) involving two species $\mathbf{X_1}$ and $\mathbf{X_2}$, the *a*PID controller involving $\mathbf{Z_1}$, $\mathbf{Z_2}$ and $\mathbf{Z_3}$ and the disturbance network (in yellow). The objective is to force the regulated output $\mathbf{X_2}$ to track a tunable setpoint, despite the injected disturbance, and enhance the transient dynamic response. The antithetic integral control is implemented via the sequestration between the σ factor (SigW denoted by $\mathbf{Z_1}$) and anti-σ factor (RsiW denoted by $\mathbf{Z_2}$)[9] that are driven by the promoters P_{LUX} and P_{RM}[52], respectively. The setpoint is encrypted in the expression rate μ of $\mathbf{Z_1}$ that is tunable with homoserine lactone (HSL). The plant is comprised of two genes. The first gene encodes *exsA*[62] fused to the degradation pdt tag (recognized by Mflon), and is driven by a SigW-responsive promoter P_{sigW}. The second gene is driven by the ExsA-responsive promoter P_{exsD}[62] and encodes the protease MfLon and the transcription factor *cI* fused to the ssrA(DAS) degradation tag. The disturbance circuit (in yellow) increases the degradation rate of $\mathbf{X_2}$ by expressing *sspB* at a tunable rate, with anhydrotetracycline (aTc), which in turn recognizes the DAS tag in $\mathbf{X_2}$ and sends it to the endogenous degradation machinery[63]. The MfLon in the output $\mathbf{X_2}$ is capable of degrading the input $\mathbf{X_1}$, while the dual activator/repressor cI is capable of activating P_{RM} and repressing P_R[52]. The promoter P_{RM} drives the integral control module, while P_R drives another gene that expresses MfLon denoted by the control species $\mathbf{Z_3}$ that is also capable of degrading the input $\mathbf{X_1}$. Note that the protease MfLon encoded in $\mathbf{X_2}$ and $\mathbf{Z_3}$ implements an incoherent feedforward loop connected in feedback with the plant and thus realizing a PD-controller. **b** Deterministic simulations. Simulation of the closed-loop dynamics with I, PI, and PID control. The plot shows the dynamic response of the regulated output $\mathbf{X_2}$ to a step change in the setpoint at $t = 8$ h and to a disturbance injection at $t = 16$ h. The simulations are carried out using biologically realistic numerical values for the various parameters (see Supplementary Information Section 11).

fluorescence measurement. Multiple cells can be targeted and observed individually in parallel on this platform. Under the cyberloop framework, at first, individual cellular outputs are observed and quantified via fluorescence imaging and subsequent image processing. The quantified value for each cell is then fed to a stochastic simulation[55] of controller reactions (one controller simulation per cell) that computes the light intensity (based on the controller species abundance) which the corresponding cell should be stimulated with. The light intensity data, once computed for every target cell in the microscope field of view, are sent to a specialized custom-built projection hardware[54] that then stimulates target cells with their corresponding light intensities in a parallel fashion, thus closing the control loop. This fluorescence measurement and subsequent light stimulation steps are repeated every fixed interval providing us with single-cell time-course data for the controller performance and output behavior.

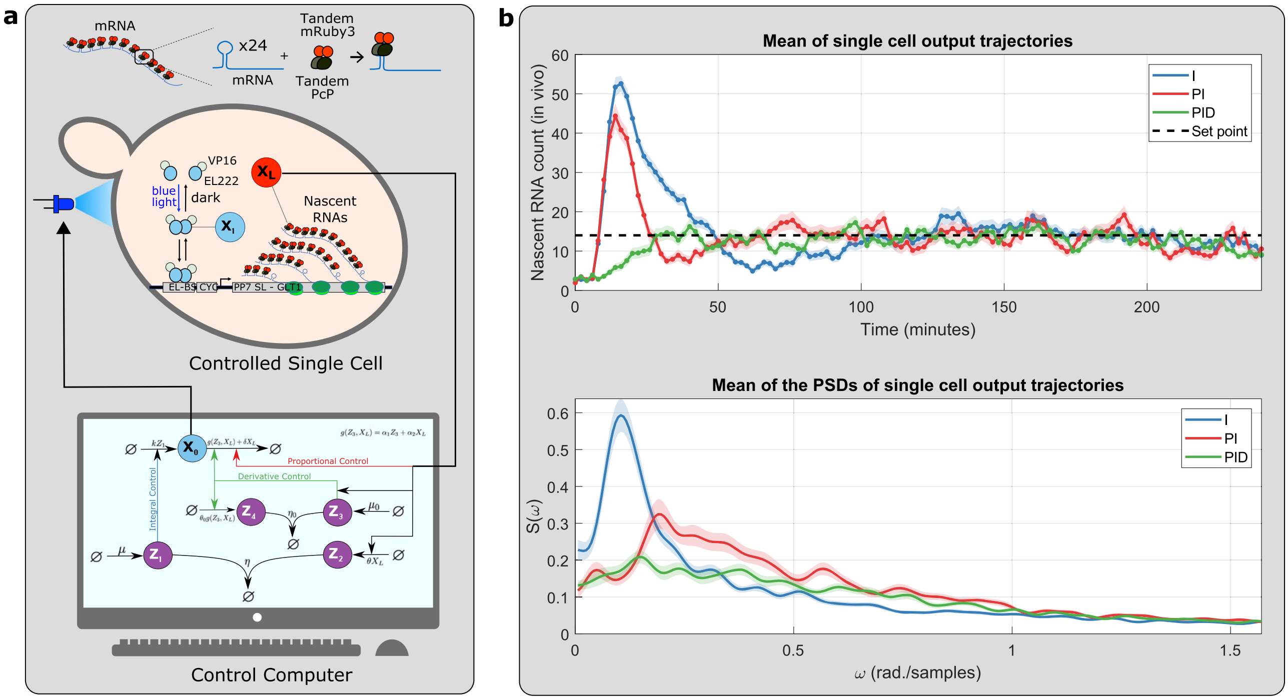


Fig. 9 Experimental demonstration of the performance of *a*PID controllers in the cyberloop platform[32] with a transcription circuit in *Saccharomyces cerevisiae*. a A Cyberloop implementation of the fourth-order *a*PID controller. Using an optogenetic framework proposed in[32, 54], an in silico stochastic simulation of the proposed fourth-order *a*PID controller (one controller per cell) is interfaced with a real biological circuit/plant (in vivo) genetically engineered in *Saccharomyces cerevisiae* cells placed under a microscope. The circuit includes a blue light optogenetic tool allowing light-stimulated regulation of transcription, making nascent RNAs ($\mathbf{X_L}$) as the output of interest to be controlled. These nascent RNAs (fused with fluorescent proteins) can be quantified via fluorescence imaging under the microscope and subsequent image processing in the control computer. The single-cell nascent RNA measurements are used to simulate stochastic dynamics of the controller network for each cell. Besides controller species ($\mathbf{Z_1}$, $\mathbf{Z_2}$, $\mathbf{Z_3}$ and $\mathbf{Z_4}$), an additional in silico plant species $\mathbf{X_0}$ was added to the network to facilitate implementation of proportional and derivative control reactions. $\mathbf{X_0}$ acts as an actuation species whose abundance defines the blue light intensity that the corresponding cell is then stimulated with using a custom-built projector setup[54] attached to the microscope. This setup allows one to stimulate and observe multiple cells in parallel. The single-cell measurements, controller simulations, and blue light intensity updates are done every 2-min interval. **b** The top plot shows the mean temporal response with the I controller, the PI controller and the fourth-order PID controller. The shaded region represents mean ± standard error. This plot demonstrates the effectiveness of the PI controller in reducing the oscillations of the mean response across the cells. It also demonstrates the added benefit of the PID controller in reducing the overshoot as well. The bottom plot shows the mean power spectral density (PSD) of various responses. The shaded region represents mean ± standard error. The PSD is useful in uncovering the stochastic oscillations on the single-cell level: a sharp peak in the PSD reveals the persistence of stochastic single-cell oscillations. The plot demonstrates the effectiveness of the PID controller in smoothing out the peak and thus considerably reducing the single-cell oscillations. Two separate experiments were performed per controller to track in total 168 cells for the I controller, 128 cells for the PI controller and 178 cells for the PID controller. The experimental parameters are provided in Supplementary Table 3. Source data are available in the Source Data file.

In our cyberloop experiments, we used a target biological circuit genetically engineered in *Saccharomyces cerevisiae* (previously presented and used in[32,54]). As shown in Fig. 9a, this circuit includes an optogenetic tool for gene expression regulation designed in such a way that the transcription rate in a target cell can be changed by varying blue light intensity which the cell is stimulated with. This provides a light control over nascent RNA abundance in the cell. The nascent RNAs are engineered with multiple stem loops that can bind with available fluorescent proteins in the cell and hence, they can be observed and quantified via fluorescence imaging under the microscope. The reader is referred to[54] and[32] for further technical details about this target circuit and the cyberloop framework, respectively.

Following the approach in[32], we implemented our fourth-order PID controller network. As shown in Fig. 9b, the PID controller was capable of reducing the oscillations on both the population and single-cell levels. This is demonstrated by plotting the time response of the population average, and the power spectral density (PSD) where sharp peaks indicate single-cell oscillations[56]. Particularly, the added derivative control action was capable of considerably enhancing the response by getting rid of the overshoot of the mean response across the cells, while simultaneously smoothing out the peak of the PSD and as a result suppressing the stochastic single-single oscillations.

Alternative differentiators. In Fig. 5, the antithetic integral motif is exploited to yield an antithetic differentiator; however, other integral motifs such as zero-order[57,58] and auto-catalytic[24] integrators can also be similarly exploited as depicted in Supplementary Fig. 9. These differentiators can be carefully appended to the *a*PI controllers of Class 1 (see Fig. 3) to obtain an alternative set of *a*PID controllers depicted in Fig. 10. These differentiators act on the concentration x_L of the output species to approximate its derivative as a rate $u_D := g(z_3, x_L)$. This is one of the differences between our differentiators and those proposed in[43] where the computed derivative is encoded as a concentration of another species. Having the computed derivative encoded directly as a rate rather than a concentration is particularly convenient for controllers with a fewer number of species. Another technical difference is that our differentiators realize a derivative with a first-order low-pass filter, whereas the differentiators in[43] realize derivatives with a second-order low-pass filter due to the additional species introduced. We close this section by noting that it is also possible to replace the

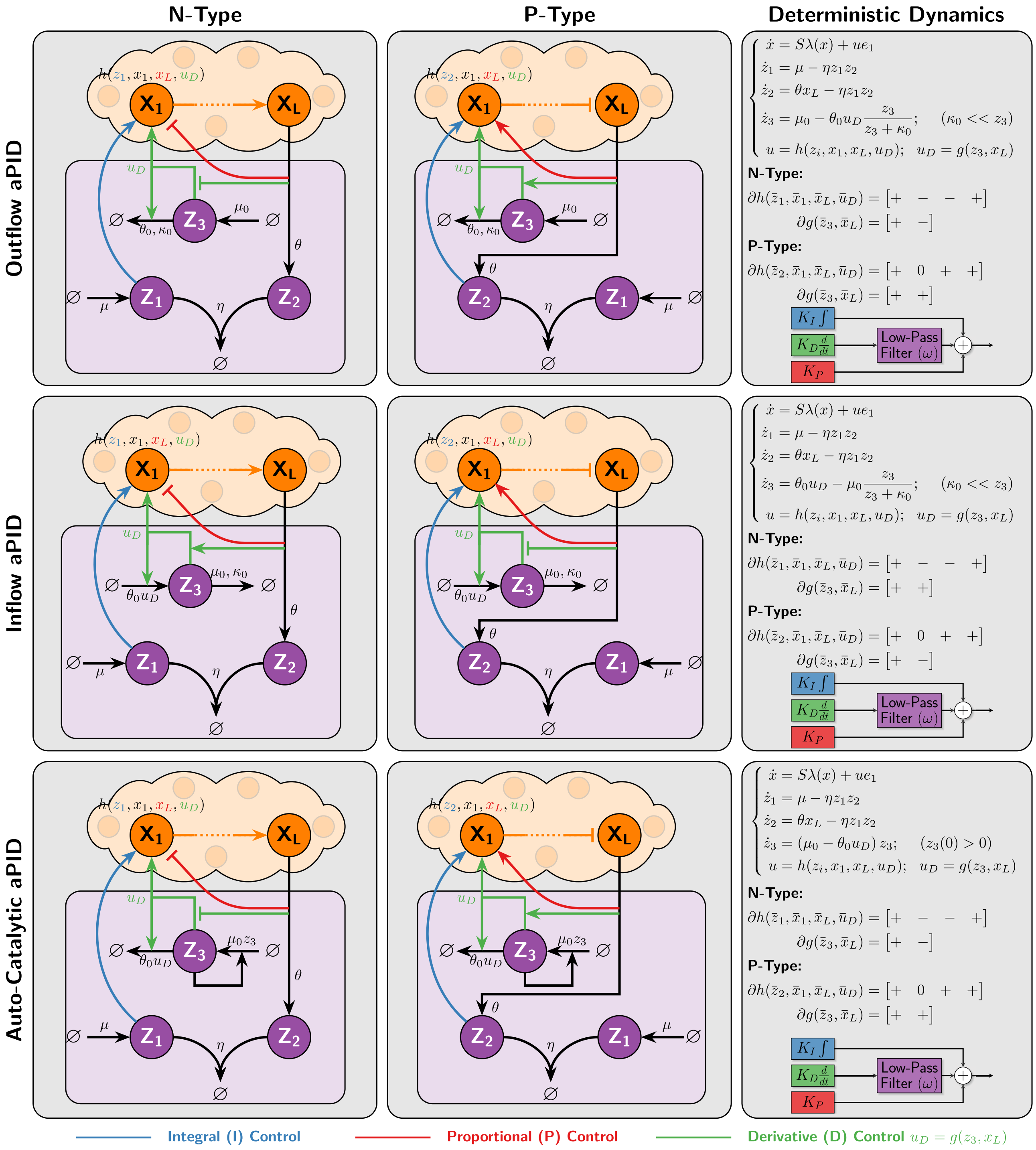

N-Type
P-Type
Deterministic Dynamics
Outflow aPID
Inflow aPID
Auto-Catalytic aPID
$h(z_1, x_1, x_L, u_D)$
$h(z_2, x_1, x_L, u_D)$
X_1
X_L
Z_1
Z_2
Z_3
u_D
μ
η
θ
μ_0
θ_0, κ_0
μ_0, κ_0
$\theta_0 u_D$
$\mu_0 z_3$
$\varnothing$
$\dot{x} = S\lambda(x) + ue_1$
$\dot{z}_1 = \mu - \eta z_1 z_2$
$\dot{z}_2 = \theta x_L - \eta z_1 z_2$
$\dot{z}_3 = \mu_0 - \theta_0 u_D \frac{z_3}{z_3 + \kappa_0}; \quad (\kappa_0 << z_3)$
$u = h(z_i, x_1, x_L, u_D); \quad u_D = g(z_3, x_L)$
N-Type:
$\partial h(\bar{z}_1, \bar{x}_1, \bar{x}_L, \bar{u}_D) = [+ \ - \ - \ +]$
$\partial g(\bar{z}_3, \bar{x}_L) = [+ \ -]$
P-Type:
$\partial h(\bar{z}_2, \bar{x}_1, \bar{x}_L, \bar{u}_D) = [+ \ 0 \ + \ +]$
$\partial g(\bar{z}_3, \bar{x}_L) = [+ \ +]$
$K_I \int$
$K_D \frac{d}{dt}$
K_P
Low-Pass Filter (ω)
$\dot{x} = S\lambda(x) + ue_1$
$\dot{z}_1 = \mu - \eta z_1 z_2$
$\dot{z}_2 = \theta x_L - \eta z_1 z_2$
$\dot{z}_3 = \theta_0 u_D - \mu_0 \frac{z_3}{z_3 + \kappa_0}; \quad (\kappa_0 << z_3)$
$u = h(z_i, x_1, x_L, u_D); \quad u_D = g(z_3, x_L)$
N-Type:
$\partial h(\bar{z}_1, \bar{x}_1, \bar{x}_L, \bar{u}_D) = [+ \ - \ - \ +]$
$\partial g(\bar{z}_3, \bar{x}_L) = [+ \ +]$
P-Type:
$\partial h(\bar{z}_2, \bar{x}_1, \bar{x}_L, \bar{u}_D) = [+ \ 0 \ + \ +]$
$\partial g(\bar{z}_3, \bar{x}_L) = [+ \ -]$
$K_I \int$
$K_D \frac{d}{dt}$
K_P
Low-Pass Filter (ω)
$\dot{x} = S\lambda(x) + ue_1$
$\dot{z}_1 = \mu - \eta z_1 z_2$
$\dot{z}_2 = \theta x_L - \eta z_1 z_2$
$\dot{z}_3 = (\mu_0 - \theta_0 u_D)\, z_3; \quad (z_3(0) > 0)$
$u = h(z_i, x_1, x_L, u_D); \quad u_D = g(z_3, x_L)$
N-Type:
$\partial h(\bar{z}_1, \bar{x}_1, \bar{x}_L, \bar{u}_D) = [+ \ - \ - \ +]$
$\partial g(\bar{z}_3, \bar{x}_L) = [+ \ -]$
P-Type:
$\partial h(\bar{z}_2, \bar{x}_1, \bar{x}_L, \bar{u}_D) = [+ \ 0 \ + \ +]$
$\partial g(\bar{z}_3, \bar{x}_L) = [+ \ +]$
$K_I \int$
$K_D \frac{d}{dt}$
K_P
Low-Pass Filter (ω)
Integral (I) Control
Proportional (P) Control
Derivative (D) Control $u_D = g(z_3, x_L)$

Fig. 10 PID controllers using integral-based differentiators. Three differentiators are constructed based on three different integrators. The differentiators appended to the *a*PI controllers of Class 1 (see Fig. 3) give rise to another collection of *a*PID controllers of both N- and P-types. The inflow and outflow *a*PID controllers are based on integrators realized via zeroth-order degradation reactions[58,57]. It is shown in Supplementary Information Section 7 that if these degradation reactions are tuned to operate in a saturating regime ($\kappa_0 << z_3$), then a low-pass filtered derivative action is mathematically realized. The difference between the outflow and inflow *a*PID controllers is that the feedback action $u_D := g(z_3, x_L)$ which approximates the derivative of x_L enters through a degradation and production reaction of the additional controller species $\mathbf{Z_3}$, respectively. In contrast, the auto-catalytic *a*PID controller is based on an auto-catalytic integrator[24] where the additional control species $\mathbf{Z_3}$ produces itself. It is shown that for this component to properly function as a differentiator, the initial concentration of $\mathbf{Z_3}$ has to be non-zero and g has to be designed such that $g(0, x_L) = 0$ (see Supplementary Information Section 7).

antithetic integral motif with other integrators to design yet another collection of PID controllers (see Supplementary Fig. 10).

Discussion

This paper proposes a library of PID controllers that can be realized via BCRNs. The proposed PID designs are introduced as a hierarchy of controllers ranging from simple to more complex designs. This hierarchical approach that we adopt offers the designer a rich library of controllers that gives rise to a natural compromise between simplicity and achievable performance. At the lower end of the hierarchy, we introduce simple PID controllers that are mathematically realized with a small number of biomolecular species and reactions making them easier to implement biologically. As we move up in the hierarchy, more biomolecular species and/or reactions are introduced to push the limit on the achievable performance. More precisely, higher-order PID controllers cover a wider range of PID gains that can be tuned to further enhance performance. Of course, this comes at the price of more complex designs making the controllers more difficult to implement biologically.

In this work, we start by introducing a library of PI controllers based on the antithetic integral motif[25] and an appended feedback control action where the input species is directly actuated by the output species. This is similar in spirit to previous works in[33] and[35] where the proportional control action enters the dynamics additively via a separate repressive production reaction. While this mechanism succeeds in enhancing the overall performance, we introduce other biologically relevant mechanisms, for the P component, that are capable of achieving even higher performance without risking instability and further reducing the stationary variance (see Fig. 4). However, it is shown rigorously and through simulations (see Fig. 6) that a PI controller alone is limited, while adding a D component adds more flexibility. Interestingly, it is shown that the performance of a gene expression network can be arbitrarily enhanced with full PID controllers: the PID can be tuned to achieve an arbitrarily fast response without triggering any oscillations or overshoots. This example highlights the power of full PID control. Another nice feature of PID control is the availability of various systematic tuning methods in the literature (see[49] for example). Well-known design tools in control theory (such as the pole placement performed in Fig. 6) can be exploited to perform the tuning in the PID parameter space instead of the biomolecular parameter space. Then the obtained PID parameters (PID gains and cutoff frequency) can be mapped by the formulas we derived (see Supplementary Information Section 4) to the actual biomolecular parameters. This approach considerably facilitates the biomolecular tuning process. It is worth mentioning that the biomolecular tuning is the easiest for the fourth-order *a*PID due to the separability of its components which allows tuning each PID gain separately with a different biomolecular parameter. In contrast, the lower order *a*PID controllers mix the various P, I, and D components and render them inseparable (see Fig. 1e) that results in each biomolecular parameter tuning multiple gains simultaneously. This is the price one has to pay for obtaining simpler designs. However, this can also be leveraged in some cases. For example, a single biomolecular parameter can tune both the integral and proportional gains simultaneously to enhance the dynamics and variance without risking instability (see the multiplicative *a*PI in Fig. 4). This inseparability also offers a nice advantage where the proportional gains can be tuned to be negative without having to switch topologies from N-type to P-type. For certain plants, achieving negative gains is critical to achieve a high performance (see Fig. 7).

We would like to point out that the proposed control structures are all designed based on linear perturbation analysis (see Supplementary Information Section 1). This is motivated by the rich set of existing tools to design and analyze linear control systems; whereas nonlinear control design and analysis is challenging and is often treated on a case-by-case basis. In the linearization, the PID structures are verified and hence the dynamics behave exactly like what is expected from classical PID control. However, full nonlinear simulations are always carried out to back up the theoretical analyses and implications. Of course, the dynamical behavior of the nonlinear PID controllers may deviate from their linear counterparts when the dynamics are (initially) far from the fixed point. This is a limitation that we believe can serve as a good future research direction where small signal analysis should be extended to large signal analysis as well. Another possible future direction is to analyze the effects of dilution on the full *a*PID controllers in a similar fashion to the analysis carried out for I-controllers only in[9] and[27]. In fact, the simulations in Fig. 8b show promising results on the roles of P and D controls in reducing the steady-state error incurred by dilution. Furthermore, in our work, we lay down a general mathematical framework for biomolecular feedback control systems that can be used to pave the way for other possible controllers in the future. We believe that research along these directions helps building high-performance controllers that are capable of reliably manipulating genetic circuits for various applications in synthetic biology and bio-medicine in the same way that PID controllers revolutionized other engineering disciplines such as navigation, telephony, aerospace, etc.

Methods

Yeast strain. No new strains were engineered in this study. Strain DBY96 from[54] was used for the cyberloop experiments. All plasmids, strains, and related details are summarized in the Key Resources Table in[54].

Culture media and initialization. Yeast cell cultures were started from a −80 °C glycerol stock at least 24 h prior to the experiment, and were grown in an incubator (Innova 42R, New Brunswick) at 30 °C in SD dropout medium (2% glucose, low fluorescence yeast nitrogen base (ForMedium), 5 g/L ammonium sulfate, 8 mg/L methionine, pH 5.8). The cell density was maintained at $OD_{600} < 0.2$ in the incubator (30 °C) for the last 12 h leading to the experiment. Approximately 400 μL of cell culture was centrifuged at 3000 RCF for 6 min, and then sufficient volume of supernatant was removed to get a concentrated culture with $OD_{600} \sim 4$.

Microfluidic chip loading protocol. The microfluidic chip proposed in[54] was used in the cyberloop experiments in this study. As mentioned in[54], this chip is a single layer poly(dimethylsiloxane) (PDMS, Sylgard 184, Dow Corning, USA) device, attached to a cover glass (thickness: 150 mm, size: 24 mm × 60 mm, Menzel-Glaser, Germany). Before loading, the PDMS device and cover glass were rinsed with acetone, isopropanol, deionized water and dried using an air gun. The chip loading protocol in[54] was followed: using a conventional pipette, 0.4 μL of the concentrated cell solution (as described before) was loaded into each chamber of the clean and dried microfluidic chip. The cover glass was placed on top of the PDMS device and pressed down very gently, creating an electrostatic bond between the glass and the PDMS. The loaded microfluidic chip was placed onto a custom-built microscope holder. A syringe pump (Model no. 300, New Era Pump Systems, Inc.) was used to maintain 30 μL/min of media flow through the loaded microfluidic chip. Cells were allowed to settle in the new conditions for 2 h prior to the start of any experiment.

Imaging and light delivery system. All image acquisitions were performed as described in[54]. Briefly, images were taken under an automated Nikon Ti-Eclipse inverted microscope (Nikon Instruments), equipped with a 40× oil-immersion objective (MRH01401, Nikon AG, Egg, Switzerland) and CMOS camera ORCA-Flash4.0 (Hamamatsu Photonic, Solothurn, Switzerland). Brightfield imaging was done using LED 100 (Märzhäuser Wetzlar GmbH & Co. KG) with diffuser and green interference filter placed in the light path. Fluorescence (mRuby3) imaging was done using Spectra X Light Engine fluorescence excitation light source (Lumencor, Beaverton, USA) with 550/15 nm LED line from the light source, 561/4 nm excitation filter, HC-BS573 beam splitter, 605/40 nm emission filter (filters and beam splitter acquired from AHF Analysetechnik AG, Tubingen, Germany). The microscope sample temperature was maintained at 30 °C by enclosing the microscope with an opaque environment box setup (Life Imaging Systems, Switzerland), which also shielded the cell sample from external light.

To achieve optogenetic stimulation with single-cell resolution under the microscope, a Digital Micromirror Device (DMD)-based projection hardware developed in[54] was used. An additional neutral density filter (ND 1.3, 25 mm absorptive filter from Thorlabs) was placed in the light stimulation pathway to reduce blue light intensity reaching the cells. The microscope and DMD projector was operated using an open source microscope control software YouScope[59].

Image analysis. In this study, each of the cyberloop experiments was run for 4 h duration with imaging/sampling done every 2 min. At every imaging step, two brightfield images above and below the focal plane (±5 a.u. Nikon Perfect Focus System) were acquired, with an exposure of 100 ms each. These images were used for cell segmentation and tracking over the course of the experiment. For nascent RNA count quantification, five fluorescence images (Z stacks with step size ~0.5 µm) were also captured, with an exposure of 300 ms each. The software tools developed in[54] and[32] were employed for cell segmentation, tracking and (nascent RNA) quantification. These image analysis software routines were run in MATLAB (MathWorks) environment.

Stochastic simulation of proposed controllers. The in silico simulations of the proposed biomolecular controllers were run in MATLAB (MathWorks) environment. Routines developed in[32] were used in the cyberloop experiments. Briefly, at every sampling time, the following steps were performed:

1. The quantified cellular readout (nascent RNA count) was used to compute and update controller reaction network propensities for every tracked cell.
2. Gillespie's Stochastic Simulation Algorithm[55] was then employed to obtain the controller species abundance.
3. These abundance values for individually tracked cells were used to compute blue light intensities (proportional to the $\mathbf{X_0}$ abundance) which the corresponding cells were stimulated with.

Stimulation of individually tracked cells was done via a light delivery system mentioned previously.

Data analysis and formatting. All data obtained from Cyberloop experiments (Fig. 9b) were analyzed and plotted using MATLAB R2018a (academic use) platform. For these experiments, data from non-responding cells were manually removed from the analysis and further consideration. These outliers (non-responding cells) constituted around 5–10% of total cells tracked throughout the experiment. To identify and remove these cells from our data, we first observed the temporal profile of the actuation species $\mathbf{X_0}$ abundance, which determines the blue light intensity a cell receives, for each tracked cell. Cells that were receiving constantly increasing blue light intensity and showed no appreciable increase in the output response were then removed from our final analysis. All data analyses and simulations in this work were performed on MATLAB R2018a and R2021a (academic use) platforms. Different plots and figures were structured and formatted using Inkscape (v0.92, open source), MATLAB and TexStudio (v3.1.1, open source) software.

Reporting summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

The raw data (MATLAB .mat files) for the results of Fig. 9b are available in the Source Data file. They are also available at https://doi.org/10.5281/zenodo.6373177[60]. Source Data are provided with this paper.

Code availability

The MATLAB code for generating all the figures is available at a dedicated GitHub repository: https://github.com/Maurice-Filo/Biomolecular-PID-Control[60]. In the same repository, a MATLAB application is also available and is capable of simulating various biomolecular controllers using a graphical user interface. The custom code for the cyberloop experiments presented in Fig. 9b is run on an integrated experimental setup and hardware[54], and cannot be executed without the full associated hardware–software suite. Codes for experiments and hardware configuration files are available upon request.

Received: 28 August 2021; Accepted: 25 March 2022;
Published online: 19 April 2022

References

1. Yi, T.-M., Huang, Y., Simon, M. I. & Doyle, J. Robust perfect adaptation in bacterial chemotaxis through integral feedback control. *Proc. Natl Acad. Sci. USA.* **97**, 4649–4653 (2000).
2. Muzzey, D., Gómez-Uribe, C. A., Mettetal, J. T. & van Oudenaarden, A. A systems-level analysis of perfect adaptation in yeast osmoregulation. *Cell* **138**, 160–171 (2009).
3. El-Samad, H., Goff, J. P. & Khammash, M. Calcium homeostasis and parturient hypocalcemia: an integral feedback perspective. *J. Theor. Biol.* **214**, 17–29 (2002).
4. Dunlop, M. J., Keasling, J. D. & Mukhopadhyay, A. A model for improving microbial biofuel production using a synthetic feedback loop. *Syst. Synth. Biol.* **4**, 95–104 (2010).
5. Stapleton, J. A. et al. Feedback control of protein expression in mammalian cells by tunable synthetic translational inhibition. *ACS Synth. Biol.* **1**, 83–88 (2012).
6. Lillacci, G., Aoki, S., Schweingruber, D. & Khammash, M. A synthetic integral feedback controller for robust tunable regulation in bacteria. Preprint at *bioRxiv*, p. 170951 (2017).
7. Ng, A. H. et al. Modular and tunable biological feedback control using a de novo protein switch. *Nature* **572**, 265–269 (2019).
8. Huang, H.-H., Qian, Y. & Del Vecchio, D. A quasi-integral controller for adaptation of genetic modules to variable ribosome demand. *Nat. Commun.* **9**, 1–12 (2018).
9. Aoki, S. K. et al. A universal biomolecular integral feedback controller for robust perfect adaptation. *Nature* **570**, 533–537 (2019).
10. Lillacci, G., Benenson, Y. & Khammash, M. Synthetic control systems for high performance gene expression in mammalian cells. *Nucleic Acids Res.* **46**, 9855–9863 (2018).
11. Hsiao, V., De Los Santos, E. L. C., Whitaker, W. R., Dueber, J. E. & Murray, R. M. Design and implementation of a biomolecular concentration tracker. *ACS Synth. Biol.* **4**, 150–161 (2015).
12. Kelly, C. L. et al. Synthetic negative feedback circuits using engineered small rnas. *Nucleic Acids Res.* **46**, 9875–9889 (2018).
13. Agrawal, D. K., Marshall, R., Noireaux, V. & Sontag, E. D. In vitro implementation of robust gene regulation in a synthetic biomolecular integral controller. *Nat. Commun.* **10**, 1–12 (2019).
14. Frei, T., Chang, C.-H., Filo, M., Arampatzis, A. & Khammash, M. Genetically engineered proportional-integral feedback controllers for robust perfect adaptation in mammalian cells. Preprint at *bioRxiv* (2021).
15. Francis, B. A. & Wonham, W. M. The internal model principle of control theory. *Automatica* **12**, 457–465 (1976).
16. Minorsky, N. Directional stability of automatically steered bodies. *J. Am. Soc. Nav. Eng.* **34**, 280–309 (1922).
17. Vilanova, R. & Visioli, A. *PID control in the Third Millennium* (Springer, 2012).
18. Blevins, T. L. Pid advances in industrial control. *IFAC Proc. Vol.* **45**, 23–28 (2012).
19. Li, J. & Li, Y. Dynamic analysis and pid control for a quadrotor. In *2011 IEEE International Conference on Mechatronics and Automation*. 573–578 (IEEE, 2011).
20. Bennett, S. A brief history of automatic control. *IEEE Control Syst. Mag.* **16**, 17–25 (1996).
21. Åström, K. J. & Murray, R. M. *Feedback Systems: An Introduction for Scientists and Engineers* (Princeton University Press, 2010).
22. Briat, C. A biology-inspired approach to the positive integral control of positive systems: the antithetic, exponential, and logistic integral controllers. *SIAM J. Appl. Dyn. Syst.* **19**, 619–664 (2020).
23. Oishi, K. & Klavins, E. Biomolecular implementation of linear i/o systems. *IET Syst. Biol.* **5**, 252–260 (2011).
24. Briat, C., Zechner, C. & Khammash, M. Design of a synthetic integral feedback circuit: dynamic analysis and dna implementation. *ACS Synth. Biol.* **5**, 1108–1116 (2016).
25. Briat, C., Gupta, A. & Khammash, M. Antithetic integral feedback ensures robust perfect adaptation in noisy biomolecular networks. *Cell Syst.* **2**, 15–26 (2016).
26. Xiao, F. & Doyle, J. C. Robust perfect adaptation in biomolecular reaction networks. In *2018 IEEE Conference on Decision and Control (CDC)*, 4345–4352 (IEEE, 2018).
27. Qian, Y. & Del Vecchio, D. Realizing 'integral control'in living cells: how to overcome leaky integration due to dilution? *J. R. Soc. Interface* **15**, 20170902 (2018).
28. Samaniego, C. C. & Franco, E. Ultrasensitive molecular controllers for quasi-integral feedback. *Cell Syst.* **12**, 272–288.e3 (2021).
29. Olsman, N., Xiao, F. & Doyle, J. C. Architectural principles for characterizing the performance of antithetic integral feedback networks. *Iscience* **14**, 277–291 (2019).
30. Olsman, N. et al. Hard limits and performance tradeoffs in a class of antithetic integral feedback networks. *Cell Syst.* **9**, 49–63 (2019).
31. Filo, M. & Khammash, M. Optimal parameter tuning of feedback controllers with application to biomolecular antithetic integral control. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 951–957 (IEEE, 2019).

32. Kumar, S. Rullan, M. & Khammash, M. Rapid prototyping and design of cybergenetic single-cell controllers. *Nat. Commun.* **12**, 5651 (2021).
33. Briat, C., Gupta, A. & Khammash, M. Antithetic proportional-integral feedback for reduced variance and improved control performance of stochastic reaction networks. *J. R. Soc. Interface* **15**, 20180079 (2018).
34. Gupta, A. & Khammash, M. An antithetic integral rein controller for bio-molecular networks. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2808–2813 (IEEE, 2019).
35. Chevalier, M., Gómez-Schiavon, M., Ng, A. H. & El-Samad, H. Design and analysis of a proportional-integral-derivative controller with biological molecules. *Cell Syst.* **9**, 338–353 (2019).
36. Modi, S., Dey, S. & Singh, A. Proportional and derivative controllers for buffering noisy gene expression. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2832–2837 (IEEE, 2019).
37. Paulino, N. M. G., Foo, M., Kim, J. & Bates, D. G. Pid and state feedback controllers using dna strand displacement reactions. *IEEE Control Syst. Lett.* **3**, 805–810 (2019).
38. Whitby, M. et al. PID control of BioChemical Reaction Networks. *IEEE Trans. Automat. Contr.* **67**, 1023–1030 (2021).
39. Halter, W., Tuza, Z. A. & Allgöwer, F. Signal differentiation with genetic networks. *IFAC-PapersOnLine* **50**, 10938–10943 (2017).
40. Halter, W., Murray, R. M. & Allgöwer, F. Analysis of primitive genetic interactions for the design of a genetic signal differentiator. *Synth. Biol.* **4**, ysz015 (2019).
41. Samaniego, C. C., Giordano, G. & Franco, E. Practical differentiation using ultrasensitive molecular circuits. In *2019 18th European Control Conference (ECC)*, 692–697 (IEEE, 2019).
42. Samaniego, C. C., Kim, J. & Franco, E. Sequestration and delays enable the synthesis of a molecular derivative operator. In *2020 59th IEEE Conference on Decision and Control (CDC)*, 5106–5112 (IEEE, 2020).
43. Alexis, E., Schulte, C. C. M., Cardelli, L. & Papachristodoulou, A. Biomolecular mechanisms for signal differentiation. *Iscience* **24**, 103462 (2021).
44. Guiver, C. et al. Integral control for population management. *J. Math. Biol.* **70**, 1015–1063 (2015).
45. Briat, C. & Khammash, M. Computer control of gene expression: robust setpoint tracking of protein mean and variance using integral feedback. In 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), 3582–3588 (IEEE, 2012).
46. Briat, C. & Khammash, M. Integral population control of a quadratic dimerization process. In *52nd IEEE Conference on Decision and Control*, 3367–3372 (IEEE, 2013).
47. Milias-Argeitis, A., Rullan, M., Aoki, S. K., Buchmann, P. & Khammash, M. Automated optogenetic feedback control for precise and robust regulation of gene expression and cell growth. *Nat. Commun.* **7**, 1–11 (2016).
48. Anagnost, J. J. & Desoer, C. A. An elementary proof of the routh-hurwitz stability criterion. *Circuits Syst. Signal Process.* **10**, 101–114 (1991).
49. Åström, K. J. & Hägglund, T. *PID Controllers: Theory, Design, and Tuning*, Vol. 2 (Instrument Society of America, Research Triangle Park, NC, 1995).
50. Mc Auley, M. T., Wilkinson, D. J., Jones, J. J. L. & Kirkwood, T. B. L. A whole-body mathematical model of cholesterol metabolism and its age-associated dysregulation. *BMC Syst. Biol.* **6**, 1–21 (2012).
51. Modi, S., Dey, S. & Singh, A. Noise suppression in stochastic genetic circuits using pid controllers. *PLoS Comput. Biol.* **17**, e1009249 (2021).
52. Brödel, A. K., Jaramillo, A. & Isalan, M. Engineering orthogonal dual transcription factors for multi-input synthetic promoters. *Nat. Commun.* **7**, 1–9 (2016).
53. Hochschild, A. & Lewis, M. The bacteriophage λ ci protein finds an asymmetric solution. *Curr. Opin. Struct. Biol.* **19**, 79–86 (2009).
54. Rullan, M., Benzinger, D., Schmidt, G. W., Milias-Argeitis, A. & Khammash, M. An optogenetic platform for real-time, single-cell interrogation of stochastic transcriptional regulation. *Mol. Cell* **70**, 745–756 (2018).
55. Gillespie, D. T. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.* **81**, 2340–2361 (1977).
56. Gupta, A. & Khammash, M. Frequency spectra and the color of cellular noise. Preprint at *bioRxiv* (2020).
57. Yu Ni, X., Drengstig, T. & Ruoff, P. The control of the controller: molecular mechanisms for robust perfect adaptation and temperature compensation. *Biophys. J.* **97**, 1244–1253 (2009).
58. Khammash, M. H. Perfect adaptation in biology. *Cell Syst.* **12**, 509–521 (2021).
59. Lang, M., Rudolf, F. & Stelling, J. Use of youscope to implement systematic microscopy protocols. *Curr. Protoc. Mol. Biol.* **98**, 14–21 (2012).
60. Filo, M. A hierarchy of biomolecular proportional-integral-derivative feedback controllers for robust perfect adaptation and dynamic performance. GitHub Repository: Biomolecular-PID-Control. https://doi.org/10.5281/zenodo.6373177 (2022)
61. Khammash, M. H. Robust steady-state tracking. *IEEE Tran. Automat. Contr.* **40**, 1872–1880 (1995).
62. Shopera, T. et al. Robust, tunable genetic memory from protein sequestration combined with positive feedback. *Nucleic Acids Res.* **43**, 9086–9094 (2015).
63. McGinness, K. E., Baker, T. A. & Sauer, R. T. Engineering controllable protein degradation. *Mol. Cell* **22**, 701–707 (2006).

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (CyberGenetics; grant agreement 743269), and from the European Union's Horizon 2020 research and innovation programme (COSY-BIO; grant agreement 766840).

Author contributions

M.K. conceived the study; M.F. and M.K. designed the controller topologies; M.F. performed the control analysis and simulations, developed the code, and wrote the initial version of the paper; S.K. performed the cyberloop experiments with input from M.F.; M.K. supervised the project and secured funding; all authors contributed to the final version of the paper.

Competing interests

ETH Zürich has filed a patent application on behalf of the inventors T.F., C.H.C., M.F. and M.K. that includes the designs described (application no. EP21187316.1). The remaining authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-022-29640-7.

Correspondence and requests for materials should be addressed to Mustafa Khammash.

Peer review information *Nature Communications* thanks Noah Olsman and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.