Activation Steering with a Feedback Controller

Anonymous authors
Paper under double-blind review

Abstract

Controlling the behaviors of large language models (LLM) is fundamental to their safety alignment and reliable deployment. However, existing steering methods are primarily driven by empirical insights and lack theoretical performance guarantees. In this work, we develop a control-theoretic foundation for activation steering by showing that popular steering methods correspond to the proportional (P) controllers, with the steering vector serving as the feedback signal. Building on this finding, we propose Proportional-Integral-Derivative (PID) Steering, a principled framework that leverages the full PID controller for activation steering in LLMs. The proportional (P) term aligns activations with target semantic directions, the integral (I) term accumulates errors to enforce persistent corrections across layers, and the derivative (D) term mitigates overshoot by counteracting rapid activation changes. This closed-loop design yields interpretable error dynamics and connects activation steering to classical stability guarantees in control theory. Moreover, PID Steering is lightweight, modular, and readily integrates with state-of-the-art steering methods. Extensive experiments across multiple LLM families and benchmarks demonstrate that PID Steering consistently outperforms existing approaches, achieving more robust and reliable behavioral control.

1 Introduction

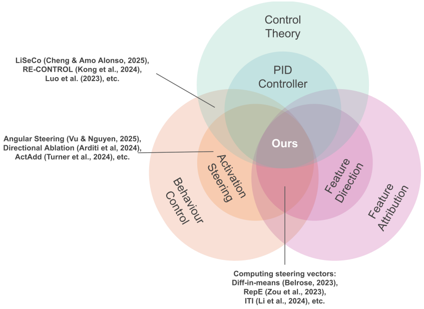

Control Theory
PID Controller
LiSeCo (Cheng & Amo Alonso, 2025), RE-CONTROL (Kong et al., 2024), Luo et al. (2023), etc.
Angular Steering (Vu & Nguyen, 2025), Directional Ablation (Arditi et al, 2024), ActAdd (Turner et al., 2024), etc.
Ours
Activation Steering
Behaviour Control
Feature Direction
Feature Attribution
Computing steering vectors: Diff-in-means (Belrose, 2023), RepE (Zou et al., 2023), ITI (Li et al., 2024), etc.

Figure 1: Our paper connects LLM Behavior Control, Feature Attribution for LLM and Control Theory. Specifically, we apply a PID-Controller to compute the steering vector for activation steering.

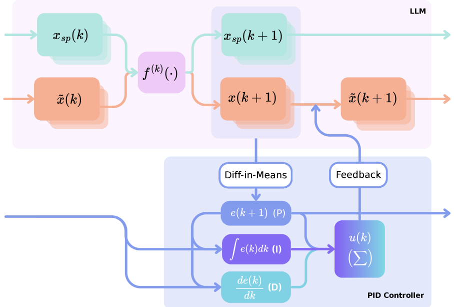


Figure 2: PID Steering: To compute the steering vector $u(k)$: a PID controller is applied at every layer $f^{(k)}(\cdot)$, using the diff-in-means between 2 contrastive data $x_{sp}(k)$ and $x(k)$ as the error signal $e(k)$.

Large language models (LLMs) have demonstrated remarkable capabilities across diverse domains, yet ensuring that their outputs align with desired behaviors remains a central challenge (Dang et al., 2025; Sclar et al., 2023; Kotha et al., 2023; Luo et al., 2025; Houlsby et al., 2019). Common post-training approaches (Wei et al., 2021; Ouyang et al., 2022) have proven effective for improving alignment. However, these techniques demand substantial computational resources (Houlsby et al., 2019) and require weight updates with new training data, which can unintentionally degrade fluency or performance on unrelated tasks (Templeton et al., 2024; Kotha et al., 2023; Luo et al., 2025).

An increasingly popular alternative is *activation steering*, which modifies a model's internal activations directly at inference time, avoiding costly retraining (Vu & Nguyen, 2025; Li et al., 2024; Turner et al., 2023; 2024; Lee et al., 2024; Rimsky et al., 2024; Rodriguez et al., 2025). This approach has been employed both to probe internal representations (Geiger et al., 2024; von Rütte et al., 2024; Vu & Nguyen, 2025) and to enable fine-grained behavioral control (Vu & Nguyen, 2025; Rodriguez et al., 2025; Turner et al., 2024; Zou et al., 2023a; Rimsky et al., 2024; Li et al., 2024). Recent work demonstrates that steering along carefully chosen low-dimensional directions can effectively alter model behavior (Turner et al., 2024; Rimsky et al., 2024; Arditi et al., 2024; Zou et al., 2023a; Vu & Nguyen, 2025), highlighting its potential as a lightweight yet powerful alignment strategy.

Steering through the Lens of Dynamical Systems. Recent methods leverage the geometric structure of the activation space (Marks & Tegmark, 2024; Park et al., 2024) using linear algebraic techniques (Turner et al., 2024; Zou et al., 2023a; Rimsky et al., 2024; Arditi et al., 2024; Vu & Nguyen, 2025) to compute the steering vectors. While effective, these works oversimplify the complex, dynamic behavior arising from the auto-regressive nature of LLMs. When viewed through this dynamical lens, activation steering can be interpreted as guiding the model's trajectory through activation space, from a region encoding one concept to another, analogous to steering a dynamical system from one state to a desired target state.

Contribution. Building on the aforementioned dynamical system insight, our work departs from the prevailing algebraic framing and instead adopts a control-theoretic perspective on activation steering. Although recent studies (Soatto et al., 2023; Kong et al., 2024; Luo et al., 2023) have begun exploring this direction, their focus has primarily remained at the level of the token-level generation proceses, treating high-level behaviors as control signals. In contrast, we take into account the internal mechanisms of LLMs by modeling the layer-wise construction of feature directions (Bricken et al., 2023; Park et al., 2024) as a dynamical system. These feature directions are then used as steering vectors (Turner et al., 2024; Zou et al., 2023a; Rimsky et al., 2024; Arditi et al., 2024; Vu & Nguyen, 2025). Specifically, we show that existing steering methods relying on difference-of-means feature directions (Rimsky et al., 2024), including Activation Addition (ActAdd) (Turner et al., 2024), Directional Ablation (Arditi et al., 2024), and Mean Activation Transport (Mean-AcT) (Rodriguez et al., 2025), can be interpreted as instances of a *proportional (P) controller*, thus suffering from the steady-state error due to the disturbance to the state of the system (Åström & Hägglund, 1995b). This new perspective enables the application of principled control-theoretic strategies for extracting effective feature directions and computing steering vectors, thereby offering stronger robustness and performance guarantees for activation steering methods. An overview of our approach is shown in Fig. 1 and 2. In this paper, we use the terms feature direction and steering vector interchangeably, noting that steering vectors represent a practical application of feature directions in activation steering. Our contribution is three-fold:

1. **Control-Theoretic Formulation for Feature Direction:** We develop a new control-theoretic framework for constructing feature directions/steering vectors along the layers of an LLM.
2. **PID-Based Steering:** We propose the novel *Proportional-Integral-Derivative (PID) Steering*, a control-theoretic framework for computing feature directions using a PID controller to reduce the steady-state error inherent in existing activation steering methods (see Fig. 2).
3. **Unified Theoretical Framework:** We demonstrate that common activation steering methods correspond to proportional (P) controllers. This connection enables a theoretical analysis that highlights PID Steering's advantages in reducing steady-state error and oscillations

We comprehensively validate our PID Steering across diverse *modalities* (text and image), *downstream applications* (toxicity mitigation, jailbreaking attack, and image style control), *steering paradigms* (ActAdd, Mean-AcT, and Angular Steering (Vu & Nguyen, 2025)), *model families* (Qwen2.5 (Yang et al., 2024), Gemma2 (Gemma Team et al., 2024), Llama3 (Llama Team, 2024), SDXL-Lightning (Lin et al., 2024), and Flux (Labs, 2024)), and *model scales* (3B-14B for language models and 3.5B-12B for diffusion models).

Organization. We organize the paper as follows: Section 2 reviews background; Section 3 links activation steering to P control and introduces PID Steering; Section 4 presents its theoretical analysis; Section 5 provides empirical validation; Appendix A discusses related work; and Section 6 concludes. Proofs, derivations, and additional experiments are in the Appendix.

2 Background

2.1 Transformers

Decoder-only transformers take an input token sequence $\boldsymbol{q} = [q_1,\dots,q_n]$ and map it to initial embeddings $\boldsymbol{x}(1) = [\boldsymbol{x}_1(1),\dots,\boldsymbol{x}_n(1)]^\top = \text{Embed}(\boldsymbol{q})$. The embeddings are then propagated through K layers. At each layer k, the residual activation $\boldsymbol{x}_i(k)$ for token p_i is updated by self-attention and an MLP block, with normalization applied before (and sometimes after) these modules:

$$\boldsymbol{x}_{i,\text{post-attn}}(k) = \boldsymbol{x}_i(k) + \text{SelfAttn}^{(k)}(\text{Norm}(\boldsymbol{x}_i(k)))$$
$$\boldsymbol{x}_i(k+1) = \boldsymbol{x}_{i,\text{post-attn}}(k) + \text{MLP}^{(k)}(\text{Norm}(\boldsymbol{x}_{i,\text{post-attn}}(k))).$$

In this paper, for notational brevity, we summarize the layered processing above as $\boldsymbol{x}_i(k+1) = f_i^{(k)}(\boldsymbol{x}(k)), i = 1,...,n$, where $f_i^{(k)}$ encapsulates both the Self-Attention mechanism and Multi-Layer Perceptron at layer k. Finally, the output activations from the last layer, $\boldsymbol{x}_i(L+1)$, are decoded over the model's vocabulary to get the next token $y_i = \text{Decode}(\boldsymbol{x}_i(L+1))$ for subsequent generation.

2.2 Activation Steering

Features such as behaviors or concepts are hypothesized to align with (approximately) orthogonal directions in activation space (Park et al., 2024; Bereska & Gavves, 2024; Elhage et al., 2022). Activation steering leverages this by modifying hidden states at inference to amplify or suppress specific features (Bayat et al., 2025; Konen et al., 2024; Li et al., 2024; Marks et al., 2025; Templeton et al., 2024). Recent approaches operationalize this idea by constructing feature directions, which act as *steering vectors* $\boldsymbol{r}$ for adjusting hidden states. These steering vectors are computed as layerwise differences in mean activations between datasets with contrasting concepts (e.g., harmful vs. harmless), a *difference-in-means* approach (Rimsky et al., 2024), shown to effectively isolate salient feature directions (Turner et al., 2023; 2024; Arditi et al., 2024).

2.2.1 Applying the Steering Vectors

Two popular activation steering approaches that use steering vectors are: *Activation Addition* (Turner et al., 2024), and *Directional Ablation* (Arditi et al., 2024). Both methods modify the token activation $\boldsymbol{x}(k)$ using the steering vector $\boldsymbol{r}(k)$ at layer k such that the activation expresses the target concept or behavior. By setting $\boldsymbol{x}(1,\boldsymbol{q}) = \text{Embed}(\boldsymbol{q})$ and $\boldsymbol{r}(1) = 0$, these methods apply the steering vectors $\boldsymbol{r}(k)$ to the activation $\boldsymbol{x}(k)$, $k = [K]$, at each layer via a steering function ρ_{steer} as follows:

$$\boldsymbol{x}(k-1,\boldsymbol{q}) = \rho_{\text{steer}}(\boldsymbol{x}(k-1,\boldsymbol{q}),\boldsymbol{r}(k-1)), \text{for } \boldsymbol{q} \in \mathcal{D}_{\text{source}} \tag{1}$$
$$\boldsymbol{x}(k,\boldsymbol{q}) = f^{(k)}(\boldsymbol{x}(k-1,\boldsymbol{q})), \text{for } \boldsymbol{q} \in \mathcal{D}_{\text{source}} \cup \mathcal{D}_{\text{target}}. \tag{2}$$

We discuss here the details on how to design the steering function ρ_{steer} for each method.

Activation Addition (ActAdd). ActAdd and sets $\rho_{\text{steer}}(\boldsymbol{x}(k), \boldsymbol{r}(k)) = \boldsymbol{x}(k) + \alpha \boldsymbol{r}(k)$, where the coefficient α controls the strength of the effect.

Directional Ablation (DirAblate). DirAblate removes the feature by projecting the token activation onto the orthogonal complement, $\rho_{\text{steer}}(\boldsymbol{x}(k),\boldsymbol{r}(k)) = \boldsymbol{x}(k) - \boldsymbol{r}(k)\,\boldsymbol{r}(k)^\top\,\boldsymbol{x}(k)$.

2.2.2 Computing the Steering Vectors

Non-sequential Mapping. Let us use the jailbreaking task as an example. In this task, we apply activation steering to force the LLM to respond to harmful prompts (Vu & Nguyen, 2025; Arditi et al., 2024). In order to compute the steering vectors, i.e., refusal direction, for each layer $k \in [K]$ and post-instruction token position $i \in I$, we calculate the mean activation $\boldsymbol{\mu}_{i,\text{target}}(k)$ for harmless prompts from $\mathcal{D}^{(\text{train})}_{\text{target}}$ and $\boldsymbol{\mu}_{i,\text{source}}(k)$ for harmful prompts from $\mathcal{D}^{(\text{train})}_{\text{source}}$:

$$\boldsymbol{\mu}_{i,\text{target}}(k) = \frac{1}{|\mathcal{D}^{(\text{train})}_{\text{target}}|} \sum_{\boldsymbol{q} \in \mathcal{D}^{(\text{train})}_{\text{target}}} \boldsymbol{x}_i(k,\boldsymbol{q}), \qquad \boldsymbol{\mu}_{i,\text{source}}(k) = \frac{1}{|\mathcal{D}^{(\text{train})}_{\text{source}}|} \sum_{\boldsymbol{q} \in \mathcal{D}^{(\text{train})}_{\text{source}}} \boldsymbol{x}_i(k,\boldsymbol{q}). \tag{3}$$

We then compute the difference-in-means vectors, $\boldsymbol{r}_i(k) = \boldsymbol{\mu}_{i,\text{target}}(k) - \boldsymbol{\mu}_{i,\text{source}}(k)$, and use them as steering vectors. Optionally, among the difference-in-means vector $\boldsymbol{r}_i(k)$ for each post-instruction token position $i \in I$ at layer k, we can select the single most effective vector $\boldsymbol{r}(k) = \text{Select}(\{\boldsymbol{r}_i(k)\}_{i \in I})$ from this set by evaluating each candidate vector over validation sets $\mathcal{D}^{(\text{val})}_{\text{source}}$ and $\mathcal{D}^{(\text{val})}_{\text{target}}$.

Sequential Mapping. A non-sequential mapping neglects the causal dependency across activations, where outputs from one layer are passed to the next, i.e., $\boldsymbol{x}_i(k+1) = f_i^{(k)}(\boldsymbol{x}(k))$. Consequently, any intervention applied at one layer must be accounted for before introducing an intervention at the subsequent layer. To capture this causal structure, *Mean Activation Transport (Mean-AcT)* in (Rodriguez et al., 2025) estimates the steering vectors incrementally at each layer as follows:

$$\boldsymbol{x}_i(k-1,\boldsymbol{q}) = \rho_{\text{steer}}(\boldsymbol{x}_i(k-1,\boldsymbol{q}),\boldsymbol{r}(k-1)),\text{for } \boldsymbol{q}\in\mathcal{D}_{\text{source}} \tag{4}$$

$$\boldsymbol{x}_i(k,\boldsymbol{q}) = f_i^{(k)}(\boldsymbol{x}(k-1,\boldsymbol{q})),\text{for } \boldsymbol{q}\in\mathcal{D}_{\text{source}}\cup\mathcal{D}_{\text{target}} \tag{5}$$

$$\boldsymbol{\mu}_{\text{target}}(k) = \frac{1}{|\mathcal{D}_{\text{target}}^{(\text{train})}|}\sum_{i\in I,\boldsymbol{q}\in\mathcal{D}_{\text{target}}^{(\text{train})}}\boldsymbol{x}_i(k,\boldsymbol{q}), \qquad \boldsymbol{\mu}_{\text{source}}(k) = \frac{1}{|\mathcal{D}_{\text{source}}^{(\text{train})}|}\sum_{i\in I,\boldsymbol{q}\in\mathcal{D}_{\text{source}}^{(\text{train})}}\boldsymbol{x}_i(k,\boldsymbol{q})$$

$$\boldsymbol{r}(k) = \boldsymbol{\mu}_{\text{target}}(k) - \boldsymbol{\mu}_{\text{source}}(k). \tag{6}$$

Like ActAdd, Mean-AcT sets $\rho_{\text{steer}}(\boldsymbol{x}(k),\boldsymbol{r}(k)) = \boldsymbol{x}(k)+\alpha\boldsymbol{r}(k)$.

2.3 Proportional–Integral–Derivative Controller

Proportional-Integral-Derivative (PID) control is a feedback mechanism extensively used in control systems (Minorsky, 1922). It is valued for its simplicity, robustness, and effectiveness in a broad range of applications, from industrial automation to robotics and aerospace systems (Visioli, 2006; Borase et al., 2021). The core idea behind PID control is to compute a control signal based on the error between a target reference signal and the actual output of a system. Specifically, consider a continuous-time dynamical system governed by a state space model

$$\dot{\boldsymbol{x}}(t) = g(\boldsymbol{x}(t),\boldsymbol{u}(t),t), \boldsymbol{y}(t) = h(\boldsymbol{x}(t),\boldsymbol{u}(t),t), \tag{7}$$

where $\boldsymbol{x}(t)\in\mathbb{R}^d$ denotes the state variable, $\boldsymbol{u}(t)\in\mathbb{R}^m$ is the control variable, and $\boldsymbol{y}(t)\in\mathbb{R}^{d'}$ represents the measured output signal. Here, $g:\mathbb{R}^d\times\mathbb{R}^m\to\mathbb{R}^d$ specifies the system dynamics, and $h:\mathbb{R}^d\times\mathbb{R}^m\to\mathbb{R}^{d'}$ is an output mapping. A PID controller applies the control variable $\boldsymbol{u}(t)$ to minimize the discrepancy between a target reference, or also known as the setpoint in the literature of PID control, $\boldsymbol{y}_{sp}(t)$ and the actual output $\boldsymbol{y}(t)$. This discrepancy, called control error, is defined as

$$\boldsymbol{e}(t) = \boldsymbol{y}_{sp}(t) - \boldsymbol{y}(t). \tag{8}$$

In a PID controller, the control variable $\boldsymbol{u}(t)$ is composed of the proportional (P), integral (I), and derivative (D) terms and given by:

$$\boldsymbol{u}(t) = K_p\boldsymbol{e}(t) + K_i\int_0^t \boldsymbol{e}(\tau)d\tau + K_d\frac{d\boldsymbol{e}(t)}{dt}, \tag{9}$$

where $K_p,K_i,K_d\geq 0$ are the proportional, integral, and derivation gains, respectively. In PID control design, the P, I, and D play different roles: *Proportional term (P)* outputs a correction proportional to the current error $\boldsymbol{e}_t$, but alone leaves a steady-state offset; *Integral term (I)* accumulates past errors to remove residual bias, ensuring offsets are corrected even as proportional effects fade; and *Derivative term (D)* responds to the error's rate of change, damping rapid growth to improve stability and reduce overshoot.

State-Feedback PID Controller. A special case of the PID controller is obtained by choosing the measured output $\boldsymbol{y}(t)$ to be the state variable $\boldsymbol{x}(t)$ in Eqn. 7, yielding the following state-space model

$$\dot{\boldsymbol{x}}(t) = g(\boldsymbol{x}(t),\boldsymbol{u}(t),t), \boldsymbol{y}(t) = \boldsymbol{x}(t). \tag{10}$$

The control error then becomes the state tracking error, $\boldsymbol{e}(t) = \boldsymbol{x}_{sp}(t) - \boldsymbol{x}(t)$, and the system is controlled through feedback of the state (Åström & Murray, 2021).

3 Steering with a Feedback Controller

In this section, we will formulate popular activation steering methods, such as ActAdd, DirAblate, and Mean-AcT, as a state-feedback P controller. Based on this new interpretation, we propose PID Steering, a novel steering method that uses a PID controller.

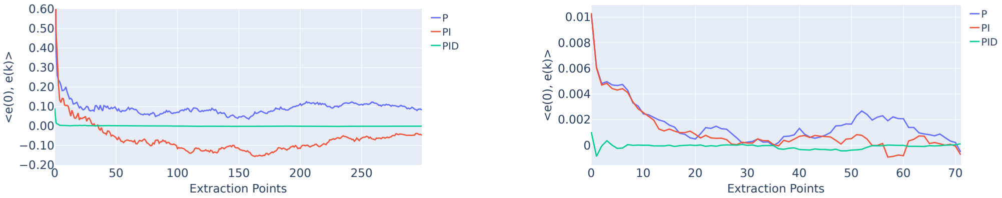
3.1 Activation Steering as a P Controller

We consider the state-feedback PID controller given in Eqn. 10 and the continuous steering vector $\boldsymbol{r}(t)$ in which we replace the layer index k by the time index t. Substituting the state tracking error $\boldsymbol{e}(t)$ by the difference-in-means vector $\boldsymbol{r}(t)$ and using the P controller whose system dynamics is governed by $g(\boldsymbol{x}(t),\boldsymbol{u}(t),t) = f(\rho_{\text{steer}}(\boldsymbol{x}(t),\boldsymbol{u}(t)),t) - \boldsymbol{x}(t)$, we obtain

$$\dot{\boldsymbol{x}}(t) = f(\rho_{\text{steer}}(\boldsymbol{x}(t),K_p\boldsymbol{r}(t)),t) - \boldsymbol{x}(t). \tag{11}$$

We discretize Eqn. 11 using Euler method (Euler, 1768; Hairer et al., 1993) to obtain

$$\boldsymbol{x}(k) - \boldsymbol{x}(k-1) = f^{(k)}(\rho_{\text{steer}}(\boldsymbol{x}(k-1),K_p\boldsymbol{r}(k-1))) - \boldsymbol{x}(k-1),$$



(a) Randomly Initialized LLama3

(b) Pretrained Qwen2.5-3B-Instruct

Figure 3: Scalar errors across time step of randomly initialized model after applying P, PI, and PID controller.

or equivalently,

$$\boldsymbol{x}(k)=f^{(k)}(\rho_{\text{steer}}(\boldsymbol{x}(k-1),K_p\boldsymbol{r}(k-1))), \tag{12}$$

where $f^{(k)}(\cdot)=f(\cdot,k)$, a function depending on index k.

Comparing Eqn. 12 with Eqn. 1 and 2 shows that applying the steering vectors as in Section 2.2.1 is equivalent to implementing the P controller, where $f^{(k)}$ is the k-th layer in an LLM, $\boldsymbol{u}(t)=K_p\boldsymbol{r}(t)$ is the new steering vector. Thus, activation steering computes the expected state tracking error.

$$\boldsymbol{r}(t)=\bar{\boldsymbol{e}}(t)=\mathbb{E}_{\boldsymbol{q}_{sp}\in\mathcal{D}^{(\text{train})}_{\text{target}}}[\boldsymbol{x}_{sp}(t,\boldsymbol{q}_{sp})]-\mathbb{E}_{\boldsymbol{q}\in\mathcal{D}^{(\text{train})}_{\text{source}}}[\boldsymbol{x}(t,\boldsymbol{q})]. \tag{13}$$

This expected state tracking error, i.e., the difference-in-means vector $\boldsymbol{r}(t)$, can be computed non-sequentially or sequentially, as explained in Section 2.2.2. When $\boldsymbol{r}(t)$ is computed non-sequentially and $\rho_{\text{steer}}(\boldsymbol{x}(k),\boldsymbol{u}(k))=\boldsymbol{x}(k)+\alpha\boldsymbol{u}(k)$ or $\boldsymbol{x}(k)-\boldsymbol{u}(k)\,\boldsymbol{u}(k)^\top\boldsymbol{x}(k)$, we obtain ActAdd or DirAblate, respectively. When $\boldsymbol{r}(t)$ is computed sequentially and $\rho_{\text{steer}}(\boldsymbol{x}(k),\boldsymbol{u}(k))=\boldsymbol{x}(k)+\alpha\boldsymbol{u}(k)$, we attain Mean-AcT.

Limitations of P Controller. There is always a steady state error in P control. The error decreases with increasing gain, but the tendency towards oscillation also increases. Since activation steering methods, i.e., ActAdd, DirAblate, and Mean-Act, are P controllers, they share the same limitations. We informally state our theoretical guarantees that P-control activation steering methods cannot alleviate the steady state error in Proposition 1 below and provide detailed proofs in Appendix B.4.

Proposition 1 (Steady-state error of P-control activation steering) *P-control activation steering ensures input-to-state stability (ISS) for an appropriate range of K_p. However, there still exists a steady-state error due to the disturbance $\boldsymbol{w}(k)$ to the state of the system. In the best case, when $\boldsymbol{w}(k)$ converges to $\boldsymbol{w}$, under a mild condition, the expected error, i.e., the difference-in-means, $\boldsymbol{r}(k)=\bar{\boldsymbol{e}}(k)$ eventually converges to a steady state $\bar{\boldsymbol{e}}_{ss}\propto\boldsymbol{w}$. Therefore, $\bar{\boldsymbol{e}}_{ss}\neq 0$ if $\boldsymbol{w}\neq 0$.*

We further provide empirical evidence to validate Proposition 1 and to illustrate the effect of adding integral and derivative terms (discussed in more detail in Section 4.2) in Fig. 3. To achieve this, we apply Sequential P-control activation steering (P Steering) on a randomly initialized model with 150 layers deep, and pretrained Qwen2.5-3B-Instruct. We plot the scalar signal $\langle\bar{e}(0),\bar{e}(k)\rangle$ which measures how much of the mean error at layer k remains aligned with the initial mean error (see Appendix B.6 for further explanation). A nonzero plateau of this quantity indicates a persistent component of the initial error (i.e., a steady-state error), whereas convergence to values near zero means that this component has been eliminated.

In this light, the behaviour in Figure 3 matches the theoretical discussion: the P curve (blue) decays but clearly settles on a nonzero plateau, confirming that P-control is ISS but admits a steady-state error. The PI curve (red) crosses zero and, after the trasient, remains close to it, meaning that the integral action eliminates the steady-state offsets in $<e(0),e(k)>$; the large negative dip before convergence is the classical overshoot of PI control, which we discuss in the “Limitations of PI control” in Section 4.2.1 of our main text and analysis in Appendix B.6. The PID curve (green) also settles near zero but with substantially reduced overshoot, showing that derivative action damps the PI transient while preserving its steady-state advantage. We analyze the overshoot reduction of PID control in Theorem 2 in Section 4.2.2 of our main text and in Appendix B.7.2.

3.2 Proportional–Integral–Derivative (PID) Steering

3.2.1 Overview

To overcome the steady-state error inherent in P-control activation steering, we extend the method by adding integral (I) and derivative (D) terms to the steering vectors. PID Steering thus (i) reacts immediately to errors via the P term for greater responsiveness, (ii) removes steady-state offsets with the I term, ensuring convergence to the desired set point, and (iii) anticipates error trends through the D term, improving stability and reducing overshoot. Together, these properties yield the following advantages:

- **Generalization.** PID Steering extends P-control methods like ActAdd, DirAblate, and Mean-AcT by adding integral and derivative components.
- **Methodological Agnosticism.** Our PID framework can be applied across different activation steering techniques, including ActAdd, DirAblate, and Mean-AcT.
- **Stability.** We theoretical prove and empirical demonstrate that PID Steering reduces steady-state error and overshoot in P-controllers, improving existing steering methods.
- **Interpretability.** Derived from classical feedback control (Minorsky, 1922), the framework inherits the simplicity and interpretability that underpin the wide use of PID controllers.

3.2.2 Computing the Steering Direction Using a PID Feedback Controller

Following Section 3.1, we consider the state-feedback PID controller in Eqn. 10, replacing the state tracking error $\boldsymbol{e}(t)$ with the difference-in-means vector $\boldsymbol{r}(t)$. With the PID controller governed by the system dynamics $g(\boldsymbol{x}(t),\boldsymbol{u}(t),t) = f(\rho_{\text{steer}}(\boldsymbol{x}(t),\boldsymbol{u}(t)),t) - \boldsymbol{x}(t)$, we obtain

$$\dot{\boldsymbol{x}}(t) = f(\rho_{\text{steer}}(\boldsymbol{x}(t), K_p\boldsymbol{r}(t) + K_i\int_0^t \boldsymbol{r}(\tau)d\tau + K_d\frac{d\boldsymbol{r}(t)}{dt}),t) - \boldsymbol{x}(t). \tag{14}$$

Eqn. 14 defines the continuous-time model of PID Steering, whose steering vector is given by:

$$\boldsymbol{u}(t) = K_p\boldsymbol{r}(t) + K_i\int_0^t \boldsymbol{r}(\tau)d\tau + K_d\frac{d\boldsymbol{r}(t)}{dt}. \tag{15}$$

In order to obtain the discrete-time formulation of PID Steering, we first discretize the sytem dynamics $\dot{\boldsymbol{x}}(t) = f(\rho_{\text{steer}}(\boldsymbol{x}(t),\boldsymbol{u}(t)),t) - \boldsymbol{x}(t)$ using Euler method (Euler, 1768; Hairer et al., 1993), same as in Section 3.1, and attain

$$\boldsymbol{x}(k) = f^{(k)}(\rho_{\text{steer}}(\boldsymbol{x}(k-1),\boldsymbol{u}(k-1))), \tag{16}$$

Next, we discretize $\boldsymbol{u}(t)$ given in Eqn. 15 to obtain $\boldsymbol{u}(k)$ using Lemma 1 below.

Lemma 1 (Discretizing PID steering vector) *Consider the continuous PID steering vector defined in Eqn. 15. The discrete-time PID steering vector is given by:*

$$\boldsymbol{u}(k) = K_p\boldsymbol{r}(k) + K_i\sum_{j=0}^{k-1}\boldsymbol{r}(j) + K_d(\boldsymbol{r}(k) - \boldsymbol{r}(k-1)). \tag{17}$$

Proof of Lemma 1 is in Appendix B.1. With Eqn. 16 and Lemma 1, we now define PID Steering.

Definition 1 (PID Steering) *Given a large language model whose layers are $\left\{f^{(k)}\right\}_{k=1}^{K}$ and a steering function ρ_{steer}, PID Steering constructs the steering vectors as follows:*

$$\boldsymbol{u}(k) = K_p\boldsymbol{r}(k) + K_i\sum_{j=0}^{k-1}\boldsymbol{r}(j) + K_d(\boldsymbol{r}(k) - \boldsymbol{r}(k-1)), \tag{18}$$

where for non-sequential mapping,

$$\boldsymbol{r}(k) = \mathbb{E}_{\boldsymbol{q}_{sp}\in\mathcal{D}^{\textit{(train)}}_{\textit{target}}}[\boldsymbol{x}_{sp}(k,\boldsymbol{q}_{sp})] - \mathbb{E}_{\boldsymbol{q}\in\mathcal{D}^{\textit{(train)}}_{\textit{source}}}[\boldsymbol{x}(k,\boldsymbol{q})],$$

and for sequential mapping,

$$\tilde{\boldsymbol{x}}(k) = f^{(k)}\left(\rho_{steer}\left(\boldsymbol{x}(k-1),\boldsymbol{u}(k-1)\right)\right), \quad \boldsymbol{r}(k) = \mathbb{E}_{\boldsymbol{q}_{sp}\in\mathcal{D}^{\textit{(train)}}_{\textit{target}}}[\boldsymbol{x}_{sp}(k,\boldsymbol{q}_{sp})] - \mathbb{E}_{\boldsymbol{q}\in\mathcal{D}^{\textit{(train)}}_{\textit{source}}}[\tilde{\boldsymbol{x}}(k,\boldsymbol{q})].$$

4 Theoretical Analysis of PID Steering

This section provides theoretical evidence for our claims: (i) adding integral action (PI) reduces steady-state error that remains under pure P-control (Proposition 3); and (ii) adding a derivative term (PID) preserves bias removal while mitigating oscillations/overshoot (Theorem 1 and 2). We denote $\boldsymbol{K}_p := K_p\boldsymbol{I}$, $\boldsymbol{K}_i := K_i\boldsymbol{I}$, and $\boldsymbol{K}_d := K_d\boldsymbol{I}$. Detailed proofs are provided in Appendix B.

4.1 Dynamics of the Average Error Across Layers

To formalize the problem, we consider N pairs of prompts/input tokens from two contrastive datasets, e.g. harmful and harmless, $\{(\boldsymbol{q}_i^+, \boldsymbol{q}_i^-)\}_{i=1}^N$ with corresponding activations $\boldsymbol{x}_i^\pm(k) \in \mathbb{R}^d$ at layer k. A steering input $\boldsymbol{u}(k)$ perturbs the undesired branch:

$$\boldsymbol{x}_i^-(k+1) = f_i^{(k)}\big(\boldsymbol{x}_i^-(k) + \boldsymbol{u}(k)\big). \tag{19}$$

Let $\bar{\boldsymbol{e}}(k) := \frac{1}{N}\sum_{i=1}^N (\boldsymbol{x}_i^+(k) - \boldsymbol{x}_i^-(k))$, the error dynamics of activation steering is then given by Proposition 2 below.

Proposition 2 (Error dynamics of activation steering) *The error dynamics $\bar{\boldsymbol{e}}(k)$ in activation steering is of the form:*

$$\bar{\boldsymbol{e}}(k+1) = \bar{\boldsymbol{A}}(k)\bar{\boldsymbol{e}}(k) - \bar{\boldsymbol{A}}(k)\boldsymbol{u}(k) + \boldsymbol{w}(k), \tag{20}$$

where $\bar{\boldsymbol{A}}(k)$ is the mean local Jacobian of $f_i^{(k)}$ at $\boldsymbol{x}_i^+(k)$ and the disturbance term $\boldsymbol{w}(k)$ collects heterogeneity. See Appendix B.3 for detailed proof and explanations of the terms.

Our control objective is to drive $\bar{\boldsymbol{e}}(k)$ to zero with input-to-state stability (ISS) for disturbed discrete system Eqn. 20 (Jiang et al., 1999).

4.2 Stability of the Error Dynamics: Roles and Caveats of PI and PID Control

In the following stability analysis, we consider the orthogonal decomposition for the disturbance $\boldsymbol{w}(k) = \boldsymbol{w}^\parallel(k) + \boldsymbol{w}^\perp(k)$, where $\boldsymbol{w}^\parallel(k) \in \mathrm{Im}\bar{\boldsymbol{A}}(k)$ and $\boldsymbol{w}^\perp(k) \in (\mathrm{Im}\bar{\boldsymbol{A}}(k))^\perp$.

4.2.1 PI Control

The following proposition provides a theoretical guarantee of PI Steering's steady-state error reduction.

Proposition 3 (Stabilizing the PI loop reduces steady-state error) *Let $\boldsymbol{M}_p(k) = \bar{\boldsymbol{A}}(k)(\boldsymbol{I} - \boldsymbol{K}_p)$, and denote $\|\boldsymbol{K}_i\| =: h$. Assume $\sup_k \|\bar{\boldsymbol{A}}(k)\| \le M < \infty$ and $\sup_k \|\boldsymbol{M}_p(k)\| \le q < 1$. If $q + Mh < 1$, then the PI closed-loop control is ISS. Furthermore, the integral part exactly cancels the matched disturbance component $\boldsymbol{w}^\parallel$. The remaining error is due only to the unmatched component $\boldsymbol{w}^\perp$, which cannot be compensated. Full proof and term explanations provided in Appendix B.5*

Limitations of PI control. Overshoot is common under PI: the closed loop oscillates about the setpoint before settling (Åström & Hägglund, 1995a, Ch. 3, §3.3, pp. 68-69), and large overshoot can arise with a high integral gain $\boldsymbol{K}_i$. In our steering setting, we explain this by scalarizing the dynamics along a reference direction . The scalarized integral state accumulates past error, pushing the trajectory beyond the setpoint; when the scalarized error changes sign, the integral discharges and the error subsequently approaches zero. See Fig. 7 for an illustration and Appendix B.6 for the formal derivation.

4.2.2 PID Control

The derivative action counteracts PI-induced oscillations near the setpoint by responding to decreases in the scalarized error, while preserving the integral term's bias-removal role, as shown in Theorems 1 and 2. For detailed proofs and explanations, see Appendix B.7.

Theorem 1 (Stabilizing the PID loop preserves bias removal) *Let $\boldsymbol{M}_p(k) = \bar{\boldsymbol{A}}(k)(\boldsymbol{I} - \boldsymbol{K}_p)$, and denote $\|\boldsymbol{K}_i\| =: h$, $\|\boldsymbol{K}_d\| =: \ell$. Assume $\sup_k \|\bar{\boldsymbol{A}}(k)\| \le M < \infty$ and $\sup_k \|\boldsymbol{M}_p(k)\| \le q < 1$. If $q + Mh < 1$ (stable PI loop), then there exists $\ell > 0$ such that the PID closed-loop control is ISS. Therefore, the integral part in PID design still cancels the matched disturbance component $\boldsymbol{w}^\parallel$.*

Theorem 2 (PID reduces the first-overshoot amplitude) *Let the first overshoot occur at index k_0 with amplitude A_0 (definition in Eqn. 54). Then, the first-overshoot amplitude under PID Steering, A_0^{PID}, satisfies $A_0^{\mathrm{PID}} \le A_0^{\mathrm{PI}}$, where A_0^{PI} denotes the corresponding amplitude under PI Steering.*

To support the theory, we present empirical evidence in Fig. 3. PI and PID controllers clearly improve over P-only control: PI removes steady-state error but causes large overshoot, while adding the derivative term mitigates overshoot and enables faster, cleaner convergence to zero.

5 Controlling the Steering Effect

In this section, we demonstrate the applicability and effectiveness of PID-Steering by using it as a drop-in replacement for the steering vector computation step across multiple steering frameworks.

Table 1: Toxicity mitigation results for Gemma-2B and Llama-8B, averaged over 10 runs. Lower is better for toxicity and perplexity; higher is better for MMLU. Bold = best, underline = second-best within each model.[1]

		Seq.	CLS Tox. (%) ↓	0-shot Tox. (%) ↓	QVQ (%) ↓	PPL Wikipedia ↓	PPL Mistral-7B ↓	MMLU ↑
Gemma2-2B	Original	–	4.17$_{\pm 0.32}$	13.42$_{\pm 1.08}$	14.17$_{\pm 0.08}$	13.98	6.68	53.1
	ActADD		3.96$_{\pm 0.24}$	13.43$_{\pm 1.42}$	14.17$_{\pm 0.08}$	14.69$_{\pm 0.22}$	**6.67**$_{\pm 0.15}$	**53.00**$_{\pm 0.51}$
	CAA		1.20$_{\pm 0.25}$	5.35$_{\pm 0.50}$	5.88$_{\pm 0.36}$	14.60$_{\pm 0.20}$	6.85$_{\pm 0.22}$	51.70$_{\pm 0.48}$
	AURA		2.12$_{\pm 0.27}$	9.04$_{\pm 0.66}$	9.72$_{\pm 0.27}$	**14.18**$_{\pm 0.14}$	7.04$_{\pm 0.34}$	**53.00**$_{\pm 0.30}$
	ITI-C		0.74$_{\pm 0.18}$	5.36$_{\pm 0.91}$	6.10$_{\pm 0.13}$	14.90$_{\pm 0.29}$	7.44$_{\pm 0.19}$	52.6$_{\pm 0.55}$
	Mean-AcT		1.12$_{\pm 0.23}$	5.20$_{\pm 0.42}$	5.80$_{\pm 0.15}$	14.53$_{\pm 0.21}$	6.81$_{\pm 0.19}$	51.74$_{\pm 0.55}$
	Linear-AcT		0.95$_{\pm 0.36}$	5.37$_{\pm 0.80}$	5.92$_{\pm 0.11}$	14.75$_{\pm 0.22}$	7.24$_{\pm 0.24}$	51.63$_{\pm 0.50}$
	Mean-AcT	✓	0.68$_{\pm 0.21}$	3.23$_{\pm 0.44}$	3.70$_{\pm 0.14}$	14.92$_{\pm 0.25}$	6.97$_{\pm 0.74}$	51.80$_{\pm 0.55}$
	Linear-AcT	✓	1.00$_{\pm 0.27}$	4.13$_{\pm 0.89}$	4.64$_{\pm 0.04}$	14.98$_{\pm 0.22}$	7.13$_{\pm 0.70}$	51.47$_{\pm 0.50}$
	PID-AcT (Ours)	✓	**0.51**$_{\pm 0.21}$	**2.90**$_{\pm 0.55}$	**3.40**$_{\pm 0.04}$	15.22$_{\pm 0.24}$	7.02$_{\pm 0.65}$	51.30$_{\pm 0.52}$
Llama3-8B	Original	–	5.80	15.00	15.81$_{\pm 0.09}$	9.06	5.68	65.30
	ActADD		5.57$_{\pm 0.45}$	15.73$_{\pm 0.21}$	16.48$_{\pm 0.19}$	9.71$_{\pm 0.46}$	5.85$_{\pm 0.26}$	**65.50**$_{\pm 0.34}$
	CAA		1.82$_{\pm 0.36}$	6.70$_{\pm 0.58}$	7.40$_{\pm 0.06}$	9.40$_{\pm 0.25}$	5.50$_{\pm 0.30}$	64.30$_{\pm 0.37}$
	AURA		1.90$_{\pm 0.61}$	8.12$_{\pm 0.85}$	8.80$_{\pm 0.17}$	9.52$_{\pm 0.32}$	6.05$_{\pm 0.30}$	**65.50**$_{\pm 0.33}$
	ITI-C		1.60$_{\pm 0.22}$	6.53$_{\pm 0.66}$	7.19$_{\pm 0.06}$	9.48$_{\pm 0.24}$	6.17$_{\pm 0.14}$	64.70$_{\pm 0.44}$
	Mean-AcT		1.78$_{\pm 0.33}$	6.56$_{\pm 0.54}$	7.30$_{\pm 0.25}$	9.36$_{\pm 0.28}$	**5.45**$_{\pm 0.34}$	64.35$_{\pm 0.39}$
	Linear-AcT		1.87$_{\pm 0.39}$	6.55$_{\pm 0.21}$	7.30$_{\pm 0.15}$	**9.35**$_{\pm 0.17}$	5.56$_{\pm 0.33}$	64.55$_{\pm 0.33}$
	Mean-AcT	✓	1.21$_{\pm 0.41}$	5.09$_{\pm 0.64}$	5.73$_{\pm 0.05}$	9.83$_{\pm 0.21}$	5.71$_{\pm 0.33}$	64.22$_{\pm 0.40}$
	Linear-AcT	✓	1.68$_{\pm 0.48}$	6.47$_{\pm 0.38}$	7.12$_{\pm 0.26}$	9.48$_{\pm 0.19}$	5.46$_{\pm 0.44}$	64.49$_{\pm 0.38}$
	PID-AcT (Ours)	✓	**0.72**$_{\pm 0.49}$	**4.36**$_{\pm 0.81}$	**4.90**$_{\pm 0.14}$	9.56$_{\pm 0.20}$	6.08$_{\pm 0.37}$	64.50$_{\pm 0.36}$

5.1 Toxicity Mitigation

We evaluate the effectiveness of PID Steering for toxic language mitigation in comparison to sequential steering methods, specifically Linear-AcT and Mean-AcT (Rodriguez et al., 2025), by closely following their experimental setup. We apply PID-Steering into Mean-AcT and call it PID-AcT. Results from other baselines, namely ActADD (Turner et al., 2023), CAA (Rimsky et al., 2024), AURA (Suau et al., 2024), ITI-C (Li et al., 2024), are also reported.

Experimental Setup. Our evaluation is conducted on Gemma2-2B (Gemma Team et al., 2024) and Llama3-8B (Llama Team, 2024), using 1,000 randomly sampled prompts from the RealToxicityPrompts dataset (Gehman et al., 2020). Toxicity is quantified with a ROBERTA-based classifier (Logacheva et al., 2022), following the methodology of Suau et al. (2024). We also assess toxicity in a zero-shot setting by employing Llama3-8B-Instruct as an LLM-as-a-judge (Zheng et al., 2023). Additionaly, in Tab. 1, we report QVQ (%) metric measured by QVQ-72B-Preview (Qwen Team, 2024) as LLM-as-a-judge's.

To measure general utility of the intervened models, we report: (i) perplexity (PPL) on a fixed set of 20k Wikipedia sentences, (ii) PPL of model-generated outputs evaluated with Mistral-7B (Jiang et al., 2023), and (iii) 5-shot MMLU (Hendrycks et al., 2021) accuracy.

Results. PID-AcT achieves the strongest toxicity reduction with minimal utility loss across both models. As shown in Tab. 1, it lowers toxicity by up to **8.2**× on Gemma2-2B and **8.1**× on Llama3-8B under both classifier and LLM-judge evaluations. PID-AcT outperforms Mean-AcT and Linear-AcT within the sequential family (Seq., ✓) and surpasses strong activation-editing baselines (ActADD, AURA, ITI-C, CAA). Utility remains stable: perplexity is comparable to other baselines; MMLU accuracy aligns with AcT-based methods and slightly lower than non-AcT approaches, likely reflecting properties of the AcT framework rather than our method. Overall, AcT-style methods yield stronger toxicity mitigation, with PID-AcT consistently ranking highest.

5.2 Jailbreaking Large Language Models

We evaluate our method on ActAdd within the Angular Steering framework (Vu & Nguyen, 2025) on the jailbreaking task, which seeks to override a model's refusal behavior and elicit harmful outputs.

Experimental Setup. Following (Vu & Nguyen, 2025), we replace DIM with our method and baselines RePE (Zou et al., 2023a) and ITI (Li et al., 2024). Refusal directions are built from 80% of AdvBench (Zou et al., 2023b) and 512 harmless Alpaca (Taori et al., 2023) samples, with the remaining 20% for evaluation. General LM ability is tested on TinyBenchmarks (Maia Polo et al., 2024). We evaluate across Gemma2, LLaMA3, and Qwen2.5 models (3B–14B).

Table 2: Comparison of Original, DIM, ITI, RePE, and PID across models on ASR and general benchmarks. Bold = best, underline = second-best within each model (ASR column). Refer to Tab. 3 for results on all tested models.

	Method	ASR↑	tinyArc↑	tinyGSM8k strict↑	tinyMMLU↑	tinyTruthQA↑	tinyHellaSwag↑	tinyWinoGrande↑
Qwen2.5-3B Instruct	*Original*	–	62.29	17.64	68.03	56.43	73.18	70.65
	DIM	<u>74.03</u>	61.95	14.80	66.11	54.95	72.40	69.85
	ITI	70.19	61.28	15.57	66.62	54.75	72.71	70.12
	RePE	68.44	61.05	14.60	65.70	54.30	72.03	69.40
	PID (ours)	**76.07**	61.20	16.01	67.29	54.10	72.59	69.72
Qwen2.5-14B Instruct	*Original*	–	73.96	90.12	74.60	64.50	82.70	73.77
	DIM	<u>90.38</u>	72.74	87.01	74.30	63.01	81.94	72.93
	ITI	33.65	73.15	89.27	74.55	64.03	82.24	73.31
	RePE	25.42	72.40	86.20	73.90	63.20	81.52	72.60
	PID (ours)	**92.65**	72.13	88.96	74.52	63.60	82.60	73.04
Llama3.1-8B Instruct	*Original*	–	65.33	63.21	62.02	54.39	82.51	65.56
	DIM	<u>93.26</u>	62.01	60.57	60.96	54.17	81.73	64.81
	ITI	79.80	64.26	61.85	61.37	54.33	82.01	65.21
	RePE	70.42	61.40	60.00	60.20	53.70	81.35	64.45
	PID (ours)	**94.85**	62.30	61.99	61.54	54.24	81.87	64.93
Gemma2-9B Instruct	*Original*	–	69.31	83.19	76.60	55.07	82.31	72.34
	DIM	<u>77.88</u>	68.21	80.14	72.29	51.86	81.45	71.51
	ITI	35.57	68.32	81.47	75.33	53.13	81.70	71.82
	RePE	28.64	67.50	79.20	71.10	51.10	81.20	71.15
	PID (ours)	**79.50**	67.91	79.24	74.89	52.49	81.59	71.42
Gemma2-27B Instruct	*Original*	–	73.45	86.91	76.11	61.36	83.24	75.47
	DIM	<u>74.03</u>	72.13	84.70	74.59	59.49	81.79	74.21
	ITI	37.36	72.84	86.38	75.51	60.85	82.83	75.11
	RePE	24.19	71.03	84.00	73.90	58.79	80.82	73.44
	PID	**79.80**	72.93	86.60	75.67	60.74	82.95	75.06

(a) Cyberpunk concept.

(b) Steampunk concept.

Figure 4: Qualitative results of activation steering in FLUX-Schnell across two style concepts with the prompt *"Lady bent over with red polka dot umbrella inside a brick building."*

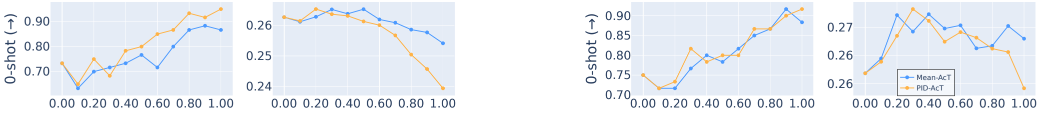
Results. PID Steering consistently outperforms DIM and scales robustly across models and metrics (see Tab. 2). On Qwen2.5-14B and LLaMA3.1-8B, PID achieves the largest ASR reductions of **92.7%** and **94.9%**, exceeding DIM by 1.5-2 points, while maintaining almost the same performance, with marginal cost, on TinyBenchmarks. Smaller models also see consistent gains: +2.0 ASR on Qwen2.5-3B and +1.3 on LLaMA3.2-3B. In contrast, ITI and RePE fail to scale, collapsing on larger models with ASR values of 33.7 and 25.4, respectively, on Qwen2.5-14B. A full version of Tab. 2 which also studies Qwen2.5-7B and Llama3.2-3B is provided in Appendix C.2.

5.3 Image Generation Styles Control

We study activation steering in diffusion models using FLUX.1.Schnell's denoising transformer (Labs, 2024), built on T5-XXL encoders (Raffel et al., 2020) and requiring just 4 diffusion steps.

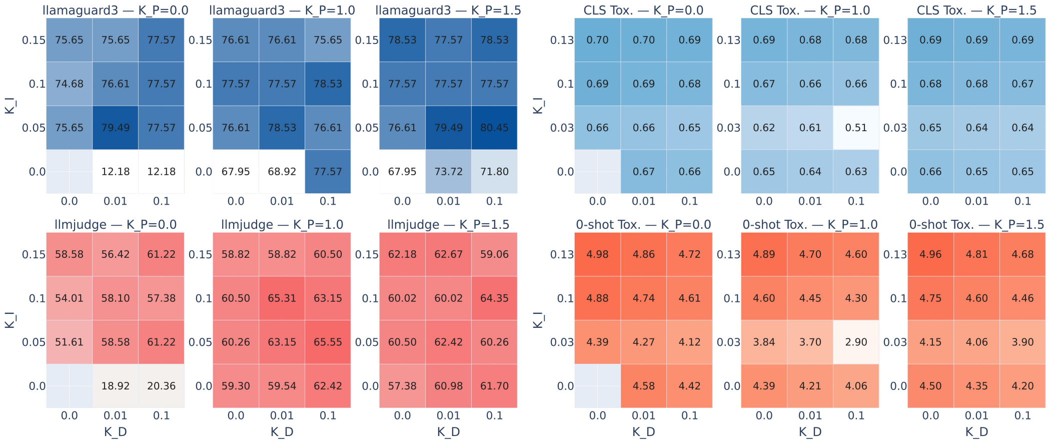
Experimental Setup. Following (Rodriguez et al., 2025), we intervene on all normalization layers after most residual blocks in FLUX. Style/concept expression is measured by a CLIP zero-shot classifier with two labels (`A picture of a {style/concept}''` vs. `A picture of something''`), and content preservation by CLIPScore (Hessel et al., 2021). Training uses 2,048 COCO Captions (Chen et al., 2015) prompts augmented with *cyberpunk/steampunk* modifiers from LLaMA-8B-Instruct (source = unmodified p, target = modified q). Evaluation samples 512 validation prompts to generate images across intervention strengths.

Results. In Fig. 4, increasing the intervention strength from 0 to 1 produces a smooth and coherent shift in style—e.g., neon tones for cyberpunk, metallic textures for steampunk—while preserving core content. Moderate strengths yield strong yet faithful stylization, and even at high strengths, semantic alignment remains largely intact. Quantitatively (Figs. 5), style intensity rises monotonically,



(a) Cyberpunk concept (b) Steampunk concept

Figure 5: 0-shot and CLIPScore results for ‘cyberpunk‘ and ‘steampunk‘ concept.



(a) Gemma-2-9b-it on Jailbreaking task. (b) Gemma-2-2B on Toxicity mitigation task.

Figure 6: Ablation Study on (K_p, K_i, K_d) parameters. We sweep through different values of K_p, K_i, K_d and report (a) Llamaguard3 (ASR in Tab. 2) and Llmjudge (evaluated using QVQ-72B-Preview) metrics for Gemma-2-9b-it, and (b) CLS Tox. and 0-shot Tox. metrics for Gemma2-2B.

with PID-AcT outperforming Mean-AcT at mid strengths (0.4–0.8). CLIPScore shows the expected trade-off: both methods decline steadily, with PID-AcT only slightly lower and by a small margin.

5.4 Ablation Study: Controller Generalization Across Architectures

Figs. 6a and 6b show that the detailed response surfaces of PID gains exhibit highly consistent structure across two very different LLMs and tasks. In the Gemma-2-9B-IT jailbreak setting (Fig. 6a), increasing K_i sharply reduces harmfulness under both LLaMAguard3 and LLM-Judge, with the strongest gains concentrated in the band $K_i \in [0.05, 0.10]$. However, large K_i introduces instability when K_p is small, visible as the pockets of degraded scores at high integral gain; adding a small derivative term $K_d \in [0.01, 0.05]$ reliably suppresses these oscillatory regions and produces smooth, monotone improvements across the grid. A nearly identical pattern appears in the Gemma-2-2B toxicity setting (Fig. 6b), despite the different objective and different model scale. The classifier-based toxicity drops most steeply as K_i increases from 0 to 0.05, after which the gains saturate; derivative action again suppresses the large overshoot visible at high K_i when $K_d = 0$, especially for $K_p = 0$. The zero-shot toxicity metric exhibits the same geometry: K_i consistently drives improvement, while K_d stabilizes the contours and prevents degradation at high integral strength. The overall layout, monotonic gains along K_i, stabilization along K_d, and mild scaling with K_p, remains aligned with Fig. 6a. Overall, these two figures demonstrate that the stabilization property of PID repeats across architectures, model sizes, and tasks: K_i improves bias removal, K_d damps instability, and K_p adjusts amplitude without altering the underlying structure. This consistency supports our claim that PID behavior arises from architecture-agnostic error dynamics and therefore generalizes without retraining.

6 Concluding Remarks

We introduced PID Steering, a control-theoretic approach to activation steering that models layer-wise representations as a dynamical system. This framework unifies prior methods, offers robustness guarantees, and leverages PID dynamics for computing steering vectors. Across language and diffusion models, PID Steering achieves stronger and more stable performance than existing approaches in toxicity mitigation, jailbreak prevention, and style control, while preserving model utility. Our results highlight control theory as a principled foundation for developing reliable and generalizable steering methods. A limitation of our work is the use of “stability-first, one-gain-at-a-time” analytical strategy to find controller gains: it clarifies the role of each component but may miss optimal choices and can overlook broader feasible regions. To address this, numerical methods, for example, LMI-based computations, can be employed. We leave these for future work.

Ethics Statement. Given the nature of the work, we do not foresee any negative societal and ethical impacts of our work. However, risks remain: by tightening activation-level control, PID Steering may inadvertently ease the generation of nuanced harmful content (e.g., persuasive misinformation or biased narratives). Although it does not expand the baseline risk profile of LLMs, robust safeguards, transparency, accountability, and ongoing ethical review are required for responsible use.

Reproducibility Statement. Source codes for our experiments are provided in the supplementary materials of the paper. The details of our experimental settings and computational infrastructure are given in Section 5 and the Appendix. All datasets that we used in the paper are published, and they are easy to access in the Internet.

LLM Usage Declaration. We use large language models (LLMs) for grammar checking and correction.

References

Andy Arditi, Oscar Balcells Obeso, Aaquib Syed, Daniel Paleka, Nina Rimsky, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=pH3XAQME6c.

Karl Johan Åström and Tore Hägglund. *PID Controllers: Theory, Design, and Tuning*. ISA - The Instrumentation, Systems and Automation Society, 1995a. ISBN 1-55617-516-7.

Karl Johan Åström and Tore Hägglund. *PID Controllers: Theory, Design, and Tuning*. ISA - The Instrumentation, Systems and Automation Society, 1995b. ISBN 1-55617-516-7.

Karl Johan Åström and Richard Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2021.

Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. Steering Large Language Model Activations in Sparse Spaces, February 2025.

Leonard Bereska and Efstratios Gavves. Mechanistic Interpretability for AI Safety – A Review, April 2024.

Rakesh P Borase, DK Maghade, SY Sondkar, and SN Pawar. A review of pid control, tuning methods and applications. *International Journal of Dynamics and Control*, 9(2):818–827, 2021.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server, 2015. URL https://arxiv.org/abs/1504.00325.

Emily Cheng and Carmen Amo Alonso. Linearly controlled language generation with performative guarantees. *arXiv preprint arXiv:2405.15454*, 2024. URL https://arxiv.org/abs/2405.15454.

Huu-Tien Dang, Tin Pham, Hoang Thanh-Tung, and Naoya Inoue. On effects of steering latent representation for large language model unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 23733–23742, 2025.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL ”https://transformer-circuits.pub/2022/toy_model/index.html”.

L. Euler. *Institutionum calculi integralis*. Number v. 1 in Institutionum calculi integralis. imp. Acad. imp. Saènt., 1768. URL https://books.google.com.sg/books?id=Vg8OAAAAQAAJ.

Zoran Gajic and Muhammad Tahir Javed Qureshi. *Lyapunov matrix equation in system stability and control*. Courier Corporation, 2008.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3356–3369, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.301. URL https://aclanthology.org/2020.findings-emnlp.301/.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning*, pp. 160–187. PMLR, 2024.

Google Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Ernst Hairer, Gerhard Wanner, and Syvert P Nørsett. *Solving ordinary differential equations I: Nonstiff problems*. Springer, 1993.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.595. URL https://aclanthology.org/2021.emnlp-main.595/.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Zhong-Ping Jiang, Eduardo Sontag, and Yuan Wang. Input-to-state stability for discrete-time nonlinear systems. *IFAC Proceedings Volumes*, 32(2):2403–2408, 1999. ISSN 1474-6670. doi: https://doi.org/10.1016/S1474-6670(17)56408-3. URL https://www.sciencedirect.com/science/article/pii/S1474667017564083. 14th IFAC World Congress 1999, Beijing, Chia, 5-9 July.

Kai Konen, Sophie Jentzsch, Diaoulé Diallo, Peer Schütt, Oliver Bensch, Roxanne El Baff, Dominik Opitz, and Tobias Hecking. Style Vectors for Steering Generative Large Language Models. In Yvette Graham and Matthew Purver (eds.), *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 782–802, St. Julian's, Malta, March 2024. Association for Computational Linguistics.

Lingkai Kong, Haorui Wang, Wenhao Mu, Yuanqi Du, Yuchen Zhuang, Yifei Zhou, Yue Song, Rongzhi Zhang, Kai Wang, and Chao Zhang. Aligning large language models with representation editing: A control perspective. *arXiv preprint arXiv:2406.05954*, 2024. URL https://arxiv.org/abs/2406.05954.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. Understanding catastrophic forgetting in language models via implicit inference. *arXiv preprint arXiv:2309.10105*, 2023.

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

Bruce W Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. *arXiv preprint arXiv:2409.05907*, 2024.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model, June 2024.

Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation, 2024. URL https://arxiv.org/abs/2402.13929.

AI @ Meta Llama Team. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. ParaDetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6804–6818, Dublin, Ireland, May 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.acl-long.469.

Yifan Luo, Yiming Tang, Chengfeng Shen, Zhennan Zhou, and Bin Dong. Prompt engineering through the lens of optimal control. *arXiv preprint arXiv:2310.14201*, 2023. URL https://arxiv.org/abs/2310.14201.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinybenchmarks: evaluating llms with fewer examples. *arXiv preprint arXiv:2402.14992*, 2024.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aajyHYjjsk.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse Feature Circuits: Discovering and Editing Interpretable Causal Graphs in Language Models, March 2025.

Nicolas Minorsky. Directional stability of automatically steered bodies. *Journal of the American Society for Naval Engineers*, 34(2):280–309, 1922.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Kiho Park, Yo Joong Choe, and Victor Veitch. The Linear Representation Hypothesis and the Geometry of Large Language Models, July 2024.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=6Mxhg9PtDE.

Alibaba Qwen Team. Qvq: To see the world with wisdom, December 2024. URL https://qwenlm.github.io/blog/qvq-72b-preview/.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), January 2020. ISSN 1532-4435.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, marco cuturi, and Xavier Suau. Controlling language and diffusion models by transporting activations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=l2zFn6TIQi.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*, 2023.

Stefano Soatto, Paulo Tabuada, Pratik Chaudhari, and Tian Yu Liu. Taming ai bots: Controllability of neural states in large language models. *arXiv preprint arXiv:2305.18449*, 2023. URL https://arxiv.org/abs/2305.18449.

Xavier Suau, Pieter Delobelle, Katherine Metcalf, Armand Joulin, Nicholas Apostoloff, Luca Zappella, and Pau Rodríguez. Whispering experts: neural interventions for toxicity mitigation in language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Alexander Turner, Sam Ringer, Rohin Shah, Andrew Critch, Victoria Krakovna, and Evan Hubinger. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*, 2023. URL https://arxiv.org/abs/2308.10248.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering Language Models With Activation Engineering, October 2024.

Antonio Visioli. *Practical PID control*. Springer, 2006.

Dimitri von Rütte, Sotiris Anagnostidis, Gregor Bachmann, and Thomas Hofmann. A Language Model's Guide Through Latent Space, February 2024.

Hieu M. Vu and Tan Minh Nguyen. Angular steering: Behavior control via rotation in activation space. *Advances in Neural Information Processing Systems*, 2025.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation Engineering: A Top-Down Approach to AI Transparency, October 2023a.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023b.

Supplement to “Activation Steering with a Feedback Controller”

Table of Contents

A Related Works

Recent works increasingly frame large language models (LLMs) as *dynamical systems*, where generation is a trajectory in latent space. This view shifts activation steering from heuristic nudging to principled *control*: rather than biasing outputs without guarantees, controllers enforce constraints on trajectories with formal assurances (Cheng & Amo Alonso, 2024). In our PID-steering framework, this distinction is key: we treat the model as a plant with hidden states evolving under controlled interventions.

Controllability of LLMs. Soatto et al. (2023) model decoder-only LLMs as discrete-time stochastic systems and prove that, under idealized assumptions (Euclidean embeddings, well-trained semantics), they are controllable in the space of meanings. This establishes that any semantic state is theoretically reachable, but probabilities may be negligible in practice, and emergent behaviors like chain-of-thought are not captured. Their results highlight both opportunities and risks: controllability opens adversarial attack surfaces but also suggests defensive controllers.

Prompting as open-loop control. Luo et al. (2023) recast multi-round prompt engineering as an optimal control problem, where each prompt is a control input maximizing task reward. This provides a unifying formalism for prompt strategies, akin to open-loop control. Yet, the framework remains conceptual: metrics in discrete language space are poorly defined, and no guarantees of stability or convergence are provided.

Closed-loop activation control. Cheng & Amo Alonso (2024) propose *Linear Semantic Control* (LiSeCo), which projects activations into safe subspaces at each decoding step via a closed-form controller. This yields lightweight, guaranteed control of simple attributes (e.g., toxicity, sentiment).

However, the linearity assumption only approximates LLM embeddings, guarantees are local rather than global, and long-horizon stability remains unaddressed.

Dynamic representation editing. Kong et al. (2024) introduce *RE-CONTROL*, which learns a value function on hidden states and applies gradient-based interventions at test time. This dynamic approach generalizes steering into a Bellman-optimal control problem, balancing alignment with fluency. Still, accuracy of the learned value function is critical, test-time optimization adds overhead, and local interventions may not guarantee global alignment.

Together, these works move activation steering from heuristics to control theory. Soatto et al. (2023) prove fundamental controllability (but under strong assumptions), Luo et al. unify prompt strategies as open-loop control (without guarantees), Cheng & Amo Alonso (2024) derive closed-form activation control (limited to linear approximations), and Kong et al. (2024) extend to dynamic optimal control (with overhead and approximation risks).

B Theoretical Proofs

B.1 Discretized PID controller

Implementing a continuous-time controller on digital hardware, such as PID, requires discretizing its derivative and integral terms (Åström & Hägglund, 1995a, p.95)

Lemma 1 (Discretizing PID steering vector) *Consider the continuous PID steering vector defined in Eqn. 15. The discrete-time PID steering vector is given by:*

$$\boldsymbol{u}(k)=K_p\boldsymbol{r}(k)+K_i\sum_{j=0}^{k-1}\boldsymbol{r}(j)+K_d(\boldsymbol{r}(k)-\boldsymbol{r}(k-1)). \tag{17}$$

Proof. We follow the discretization procedure for PID controllers in (Åström & Hägglund, 1995a, Sec. 3.6, Ch. 3). For simplicity, the sampling period is normalized to $h=1$.

Proportional term in Eqn. 15.

$$P(t)=K_p\boldsymbol{r}(t).$$

The discrete-time form is obtained by substituting sampled variables for their continuous counterparts:

$$P(k)=K_p\boldsymbol{r}(k). \tag{21}$$

Integral term in Eqn. 15.

$$I(t)=K_i\int_0^t\boldsymbol{r}(\tau)d\tau \qquad \Rightarrow \qquad \frac{dI}{dt}=K_i\boldsymbol{r}(t).$$

Using forward Euler with $h=1$,

$$I(k+1)-I(k)=K_i\boldsymbol{r}(k).$$

Hence

$$I(k+1)=I(k)+K_i\boldsymbol{r}(k),$$

which is equivalent to

$$I(k)=I(0)+K_i\sum_{j=0}^{k-1}\boldsymbol{r}(j)=K_i\sum_{j=0}^{k-1}\boldsymbol{r}(j), \tag{22}$$

since $I(0)=0$.

Derivative term in Eqn. 15.

$$D(t)=K_d\frac{d\boldsymbol{r}(t)}{dt}.$$

Approximating the derivative by the backward Euler difference with $h=1$ gives

$$D(k)=K_d\big(\boldsymbol{r}(k)-\boldsymbol{r}(k-1)\big). \tag{23}$$

Combining equation 21, equation 22, and equation 23 yields

$$\boldsymbol{u}(k)=K_p\boldsymbol{r}(k)+K_i\sum_{j=0}^{k-1}\boldsymbol{r}(j)+K_d\big(\boldsymbol{r}(k)-\boldsymbol{r}(k-1)\big).$$

□

B.2 Background on Input-to-state Stability & Notations

Background on Input-to-state Stability (ISS) In our proofs, the input-to-state stability (ISS) of a system can be established either through the definition of an ISS system in (Jiang et al., 1999, Def. 2.1) or via the use of an ISS-Lyapunov function as in (Jiang et al., 1999, Def. 2.2, Prop. 2.3). We also rely on the definition of a Lyapunov function and the difference Lyapunov equation for linear discrete-time homogeneous dynamical systems in (Gajic & Qureshi, 2008, Ch. 1, p. 8). The existence of a solution to the Lyapunov equation, together with its bound, is stated in (Gajic & Qureshi, 2008, Ch. 4, p. 110).

For reference, we briefly note that input-to-state stability (ISS) extends the classical notion of Lyapunov by explicitly accounting for external inputs: the state remains bounded and eventually converges whenever the input is bounded. A Lyapunov function provides an energy-like certificate for stability, while the associated Lyapunov equation offers a constructive method for obtaining such functions in linear settings. These notions are central for analyzing stability and will be used throughout our proofs.

Conventions and assumptions (used throughout). Let $\|\cdot\|$ denote the Euclidean norm on $\mathbb{R}^d$; for a matrix $M \in \mathbb{R}^{d\times d}$ we also write $\|M\|$ for the operator norm induced by the Euclidean norm, i.e. $\|M\| := \sup_{\|x\|=1}\|Mx\|$ (the spectral norm) (Horn & Johnson, 2012, pp. 343–346). We assume (i) $\sup_k\|\bar{\boldsymbol{A}}(k)\| < \infty$; (ii) $\boldsymbol{w}(k)$ is bounded (for a signal $\boldsymbol{w}$ we set $\|\boldsymbol{w}\|_\infty := \sup_k\|\boldsymbol{w}(k)\|$) ; (iii) the controller gains are static and time-invariant scalar multiples of the identity, i.e., $\boldsymbol{K}_p := K_p\boldsymbol{I}$, $\boldsymbol{K}_i := K_i\boldsymbol{I}$, and $\boldsymbol{K}_d := K_d\boldsymbol{I}$. We use the standard meaning of the classes $\mathcal{K}$ and $\mathcal{KL}$ as in (Jiang et al., 1999).

B.3 Dynamics of the Average Error Across Layers

To formalize the problem setup, we consider N pairs of contrastive prompt/input tokens $\{(\boldsymbol{q}_i^+, \boldsymbol{q}_i^-)\}_{i=1}^N$, where $\boldsymbol{q}_i^+$ carries the desired property and $\boldsymbol{q}_i^-$ represents the opposite. For discrete time (layer) k, let $\boldsymbol{x}_i^\pm(k) \in \mathbb{R}^d$ denote the corresponding activation vectors. The layer-to-layer evolution is

$$\boldsymbol{x}_i(k+1) = f_i^{(k)}\big(\boldsymbol{x}(k)\big), i=1,\dots,N, \tag{24}$$

with $f_i^{(k)} : \mathbb{R}^d \to \mathbb{R}^d$ differentiable on the operating region. A steering input $\boldsymbol{u}(k)$ is applied on the undesired branch:

$$\boldsymbol{x}_i^-(k+1) = f_i^{(k)}\big(\boldsymbol{x}_i^-(k)+\boldsymbol{u}(k)\big). \tag{25}$$

Defining $\bar{\boldsymbol{x}}^\pm(k) := \frac{1}{N}\sum_{i=1}^N \boldsymbol{x}_i^\pm(k)$, we track the per-pair and average errors as

$$\boldsymbol{e}_i(k) := \boldsymbol{x}_i^+(k) - \boldsymbol{x}_i^-(k), \qquad \bar{\boldsymbol{e}}(k) := \bar{\boldsymbol{x}}^+(k) - \bar{\boldsymbol{x}}^-(k), \qquad \tilde{\boldsymbol{e}}_i(k) = \boldsymbol{e}_i(k) - \bar{\boldsymbol{e}}(k). \tag{26}$$

Furthermore, we define $\boldsymbol{A}_i(k)$ as the Jacobian of $f_i^{(k)}$ at $\boldsymbol{x}_i^+(k)$:

$$\boldsymbol{A}_i(k) := J_{f_i^{(k)}}\big(\boldsymbol{x}_i^+(k)\big), \qquad \bar{\boldsymbol{A}}(t) := \frac{1}{N}\sum_{i=1}^N \boldsymbol{A}_i(k), \qquad \tilde{\boldsymbol{A}}_i(k) = \boldsymbol{A}_i(k) - \bar{\boldsymbol{A}}(k). \tag{27}$$

The dynamic of the average error $\bar{\boldsymbol{e}}(k)$ is then given by Proposition 2.

Proposition 2 (Error dynamics of activation steering) *The error dynamics $\bar{\boldsymbol{e}}(k)$ in activation steering is of the form:*

$$\bar{\boldsymbol{e}}(k+1) = \bar{\boldsymbol{A}}(k)\bar{\boldsymbol{e}}(k) - \bar{\boldsymbol{A}}(k)\boldsymbol{u}(k) + \boldsymbol{w}(k), \tag{20}$$

where $\bar{\boldsymbol{A}}(k)$ is the mean local Jacobian of $f_i^{(k)}$ at $\boldsymbol{x}_i^+(k)$ and the disturbance term $\boldsymbol{w}(k)$ collects heterogeneity. See Appendix B.3 for detailed proof and explanations of the terms.

Proof. The evolution of the average error $\bar{\boldsymbol{e}}(k)$ through layers can be described as follows:

$$\bar{\boldsymbol{e}}(k+1) = \bar{\boldsymbol{x}}^+(k+1) - \bar{\boldsymbol{x}}^-(k+1) = \frac{1}{N}\sum_{i=1}^N \Big[f_i^{(k)}\big(\boldsymbol{x}_i^+(k)\big) - f_i^{(k)}\big(\boldsymbol{x}_i^-(k)+\boldsymbol{u}(k)\big)\Big]. \tag{28}$$

Linearizing $f_i^{(k)}$ around $\boldsymbol{x}_i^+(k)$, we obtain

$$f_i^{(k)}\big(\boldsymbol{x}_i^+(k)+\delta_i(k)\big) \approx f_i^{(k)}\big(\boldsymbol{x}_i^+(k)\big) + J_{f_i^{(k)}}\big(\boldsymbol{x}_i^+(k)\big)\cdot\delta_i(k), \tag{29}$$

where $J_{f_i^{(k)}}$ denotes the Jacobian of $f_i^{(k)}$.

Setting $\delta_i(k) = -\boldsymbol{e}_i(k) + \boldsymbol{u}(k)$ yields

$$f_i^{(k)}\big(\boldsymbol{x}_i^+(k)+\delta_i(k)\big) \approx f_i^{(k)}\big(\boldsymbol{x}_i^+(k)\big) + \boldsymbol{A}_i(k)\big(-\boldsymbol{e}_i(k)+\boldsymbol{u}(k)\big),$$

Insert this into Eqn.28 we obtain

$$\bar{\boldsymbol{e}}(k+1) = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{A}_i(k)\boldsymbol{e}_i(k) - \bar{\boldsymbol{A}}(k)\boldsymbol{u}(k). \tag{30}$$

Recall that

$$\boldsymbol{e}_i(k) = \bar{\boldsymbol{e}}(k) + \tilde{\boldsymbol{e}}_{(i)}(k), \text{ then } \frac{1}{N}\sum_{i=1}^{N}\tilde{\boldsymbol{e}}_{(i)}(k) = 0,$$

and

$$\boldsymbol{A}_{(i)}(t) = \bar{\boldsymbol{A}}(k) + \tilde{\boldsymbol{A}}_{(i)}(k), \text{ then } \frac{1}{N}\sum_{i=1}^{N}\tilde{\boldsymbol{A}}_{(i)}(k) = 0.$$

Therefore,

$$\begin{aligned}
\bar{\boldsymbol{e}}(k+1) &= \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{A}_i(k)\boldsymbol{e}_i(k) - \bar{\boldsymbol{A}}(k)\boldsymbol{u}(k) \\
&= \frac{1}{N}\sum_{i=1}^{N}\bar{\boldsymbol{A}}(k)\bar{\boldsymbol{e}}_i(k) - \bar{\boldsymbol{A}}(k)\boldsymbol{u}(k) + \frac{1}{N}\sum_{i=1}^{N}\tilde{\boldsymbol{e}}_{(i)}(k)\tilde{\boldsymbol{A}}_{(i)}(k) \\
&\quad \underbrace{+\bar{\boldsymbol{A}}(k)\frac{1}{N}\sum_{i=1}^{N}\tilde{\boldsymbol{e}}_{(i)}(k) + \bar{\boldsymbol{e}}(k)\frac{1}{N}\sum_{i=1}^{N}\tilde{\boldsymbol{A}}_{(i)}(k)}_{=0}
\end{aligned} \tag{31}$$

We then obtain the final state-space model for the dynamics of $\bar{e}(t)$ as

$$\bar{\boldsymbol{e}}(k+1) = \bar{\boldsymbol{A}}(k)\bar{\boldsymbol{e}}(k) - \bar{\boldsymbol{A}}(k)\boldsymbol{u}(k) + \boldsymbol{w}(k), \tag{32}$$

where

$$\boldsymbol{w}(k) = \frac{1}{N}\sum_{i=1}^{N}\tilde{\boldsymbol{A}}_i(k)\tilde{\boldsymbol{e}}_i(k),$$

which acts as a time-dependent exogenous disturbance to the model □

B.4 Proportional (P) Control

Consider proportional control with

$$\boldsymbol{u}(k) = \boldsymbol{K}_p\bar{\boldsymbol{e}}(k), \quad (\boldsymbol{K}_i = \boldsymbol{K}_d = 0). \tag{33}$$

The dynamics Eqn. 20 then become

$$\bar{\boldsymbol{e}}(k) = \boldsymbol{M}_p(k)\bar{\boldsymbol{e}}(k) + \boldsymbol{w}(k), \tag{34}$$

where $\boldsymbol{M}_p(k) = \bar{\boldsymbol{A}}(k)(I - \boldsymbol{K}_p)$.

With a suitable choice of $\boldsymbol{K}_p$, the system can be made input-to-state stable (ISS); that is, there exist a $\mathcal{KL}$-function β and a $\mathcal{K}$-function γ such that, for all disturbance $\boldsymbol{w}$ with bounded sup norm and all initial states $\bar{\boldsymbol{e}}(0)$,

$$\|\bar{\boldsymbol{e}}(k)\| \le \beta(\|\bar{\boldsymbol{e}}(0)\|, k) + \gamma(\|\boldsymbol{w}\|_\infty), \quad k \in \mathbb{Z}_{\geq 0}, \tag{35}$$

see (Jiang et al., 1999, Def. 2.1).

In particular, the error decays from the initial condition and remains bounded under bounded disturbances.

Proposition 1 (Steady-state error of P-control activation steering) *P-control activation steering ensures input-to-state stability (ISS) for an appropriate range of K_p. However, there still exists a steady-state error due to the disturbance $\mathbf{w}(k)$ to the state of the system. In the best case, when $\mathbf{w}(k)$ converges to $\mathbf{w}$, under a mild condition, the expected error, i.e., the difference-in-means, $\boldsymbol{r}(k) = \bar{\mathbf{e}}(k)$ eventually converges to a steady state $\bar{\mathbf{e}}_{ss} \propto \mathbf{w}$. Therefore, $\bar{\mathbf{e}}_{ss} \neq 0$ if $\mathbf{w} \neq 0$.*

Proof. Assume $\sup_k \|\bar{\boldsymbol{A}}(k)\| \le M < \infty$, $\boldsymbol{K}_p = K_p I = pI$ with $p > 0$. Since $\boldsymbol{M}_p(k) = \bar{\boldsymbol{A}}(k)(I - \boldsymbol{K}_p) = \bar{\boldsymbol{A}}(k)(1-p)I$, by sub-multiplicative property of matrix norm we have

$$\|\boldsymbol{M}_p(t)\| \le \|\bar{A}(t)\| \|(1-p)I\| \le M|1-p| =: q. \tag{36}$$

For $p \in \left(1 - \frac{1}{M}, 1 + \frac{1}{M}\right)$, we have $q < 1$.

Expanding recursively,

$$\bar{\boldsymbol{e}}(k) = \boldsymbol{M}_p(k-1) \cdots \boldsymbol{M}_p(0)\bar{\boldsymbol{e}}(0) + \sum_{j=0}^{k-1} \boldsymbol{M}_p(k-1) \cdots \boldsymbol{M}_p(j+1) w(j).$$

Hence,

$$\|\bar{\boldsymbol{e}}(k)\| \le q^k \|\bar{\boldsymbol{e}}(0)\| + \sum_{j=0}^{k-1} q^{k-1-j} \|w(j)\| \le q^k \|\bar{e}(0)\| + \frac{1-q^k}{1-q} \|w\|_\infty \le q^k \|\bar{e}(0)\| + \frac{1}{1-q} \|w\|_\infty. \tag{37}$$

Since $q < 1$, we can set $\beta(s,k) = q^k s$, which is a $\mathcal{KL}$-function (decaying to zero as $k \to \infty$), and $\gamma(s) = \frac{1}{1-q} s$, which is a $\mathcal{K}$-function, satisfying Eqn. 35. Therefore, the system is ISS.

However, there exists a steady-state error due to the disturbance $\boldsymbol{w}(k)$. In the best case, when $\bar{\boldsymbol{A}}(k)$ converges to $\bar{\boldsymbol{A}}$ and $\boldsymbol{w}(k)$ converges to $\boldsymbol{w}$, the error $\bar{\boldsymbol{e}}(k)$ eventually converges to a steady state given by

$$\bar{\boldsymbol{e}}_{ss} = (I - \bar{\boldsymbol{A}}(1-pI))^{-1} \boldsymbol{w}.$$

Therefore, $\bar{\boldsymbol{e}}_{ss} \neq 0$ if $\boldsymbol{w} \neq 0$. □

Remark 1 (Convergence rate versus $\boldsymbol{K}_p$.) *From Ineq. 37, smaller q yields faster convergence. Because*

$$q(p) = M|1-p| = \begin{cases} M(1-p), & p \in \left(1 - \frac{1}{M}, 1\right), \\ M(p-1), & p \in \left[1, 1 + \frac{1}{M}\right), \end{cases}$$

we have $\frac{d}{dp} q(p) = -M < 0$ for $p < 1$ and $\frac{d}{dp} q(p) = M > 0$ for $p > 1$. Therefore the contraction factor $q(p)$ is minimized at

$$p^\star = 1 \quad \Longrightarrow \quad q^\star = 0,$$

and increases as p moves away from 1 within the admissible interval.

B.5 Proportional-Integral (PI) Control

To reduce the steady-state error, the proportional controller is extended with an integral action, resulting in a proportional-integral (PI) control law:

$$\boldsymbol{u}(k) = \boldsymbol{K}_p \bar{\boldsymbol{e}}(k) + \boldsymbol{K}_i \boldsymbol{s}(k), \qquad \boldsymbol{s}(k+1) = \boldsymbol{s}(k) + \bar{\boldsymbol{e}}(k), \qquad (\boldsymbol{K}_d = 0). \tag{38}$$

The dynamics Eqn. 20 then become

$$\bar{\boldsymbol{e}}(k+1) = \bar{\boldsymbol{A}}(k)(I - \boldsymbol{K}_p)\bar{\boldsymbol{e}}(k) - \bar{\boldsymbol{A}}(k)\boldsymbol{K}_i \boldsymbol{s}(k) + \boldsymbol{w}(k). \tag{39}$$

We use the following orthogonal decomposition for $\boldsymbol{w}(k)$:

$$\boldsymbol{w}(k) = \boldsymbol{w}^{\parallel}(k) + \boldsymbol{w}^{\perp}(k),$$

where $\boldsymbol{w}^{\parallel}(k) \in \operatorname{Im} \bar{\boldsymbol{A}}(k)$ and $\boldsymbol{w}^{\perp}(k) \in (\operatorname{Im} \bar{\boldsymbol{A}}(k))^{\perp}$.

The impact of $\boldsymbol{w}^{\parallel}(k)$ on the error can be eliminated by PI control, as discussed below. On the other hand, P-only control is not able to do so, because keeping $\bar{\boldsymbol{e}}(k) = 0$ requires $\boldsymbol{u}(k) = 0$, leaving no component in $\boldsymbol{u}(k)$ that can compensate for $\boldsymbol{w}^{\parallel}(k)$.

Since $\boldsymbol{w}^{\parallel}(k) \in \operatorname{Im} \bar{\boldsymbol{A}}(k)$, it can be expressed as

$$\boldsymbol{w}^{\parallel}(k) = \bar{\boldsymbol{A}}(k) \boldsymbol{K}_i s^*(k). \quad \Longleftrightarrow \quad \boldsymbol{s}^*(k) = \boldsymbol{K}_i^{-1} \bar{\boldsymbol{A}}(k)^{\dagger} \boldsymbol{w}^{\parallel}(k)$$

Let $\tilde{\boldsymbol{s}}(k) = \boldsymbol{s}(k) - \boldsymbol{s}^*(k)$ and $d(k) = s^*(k+1) - s^*(k)$. Therefore,

$$\tilde{\boldsymbol{s}}(k+1) = \tilde{\boldsymbol{s}}(k) + \bar{\boldsymbol{e}}(k) - \boldsymbol{d}(k) \tag{40}$$

Insert $\boldsymbol{s}(k) = \boldsymbol{s}^*(k) + \tilde{\boldsymbol{s}}(k)$ and $\boldsymbol{w}(k) = \boldsymbol{w}^{\parallel}(k) + \boldsymbol{w}^{\perp}(k) = \bar{\boldsymbol{A}}(k)\boldsymbol{K}_i s^*(k) + \boldsymbol{w}^{\perp}(k)$ into Eqn. 39,

$$\begin{aligned}\bar{\boldsymbol{e}}(k+1) &= \bar{\boldsymbol{A}}(k)(I - \boldsymbol{K}_p)\bar{\boldsymbol{e}}(k) - \bar{\boldsymbol{A}}(k)\boldsymbol{K}_i\boldsymbol{s}^*(k) - \bar{\boldsymbol{A}}(k)\boldsymbol{K}_i\tilde{\boldsymbol{s}}(k) + \bar{\boldsymbol{A}}(k)\boldsymbol{K}_i s^*(k) + \boldsymbol{w}^{\perp}(k) \\ &= \bar{\boldsymbol{A}}(k)(I - \boldsymbol{K}_p)\bar{\boldsymbol{e}}(k) - \bar{\boldsymbol{A}}(k)\boldsymbol{K}_i\tilde{\boldsymbol{s}}(k) + \boldsymbol{w}^{\perp}(k)\end{aligned} \tag{41}$$

We introduce the lifted state $\tilde{\zeta}_{\mathrm{PI}}(k) = \begin{bmatrix}\bar{\boldsymbol{e}}(k) \\ \tilde{\boldsymbol{s}}(k)\end{bmatrix}$ with its dynamic derived from Eqn. 40-41 as follow

$$\tilde{\zeta}_{\mathrm{PI}}(k+1) = \boldsymbol{M}_i(t)\tilde{\zeta}_{\mathrm{PI}}(k) + \tilde{\boldsymbol{w}}_{\mathrm{PI}}(k), \tag{42}$$

where

$$\boldsymbol{M}_i(k) = \begin{bmatrix}\boldsymbol{M}_p(k) & -\boldsymbol{G}(k) \\ I & I\end{bmatrix},$$

with $\boldsymbol{M}_p(k) = \bar{\boldsymbol{A}}(p)(I - \boldsymbol{K}_p)$, $\boldsymbol{G}(k) = \bar{\boldsymbol{A}}(k)\boldsymbol{K}_i$ and

$$\tilde{\boldsymbol{w}}_{\mathrm{PI}}(k) = \begin{bmatrix}\boldsymbol{w}^{\perp}(k) \\ -\boldsymbol{d}(k)\end{bmatrix},$$

Proposition 3 (Stabilizing the PI loop reduces steady-state error) *Let $\boldsymbol{M}_p(k) = \bar{\boldsymbol{A}}(k)(\boldsymbol{I} - \boldsymbol{K}_p)$, and denote $\|\boldsymbol{K}_i\| =: h$. Assume $\sup_k \|\bar{\boldsymbol{A}}(k)\| \le M < \infty$ and $\sup_k \|\boldsymbol{M}_p(k)\| \le q < 1$. If $q + Mh < 1$, then the PI closed-loop control is ISS. Furthermore, the integral part exactly cancels the matched disturbance component $\boldsymbol{w}^{\parallel}$. The remaining error is due only to the unmatched component $\boldsymbol{w}^{\perp}$, which cannot be compensated. Full proof and term explanations provided in Appendix B.5*

Proof. Using the sub-multiplicativity of the induced matrix norm and the triangle inequality, and noting that $\|\boldsymbol{M}_p(k)\| \le q$, $\|\boldsymbol{G}(k)\| = \|\bar{\boldsymbol{A}}(k)\boldsymbol{K}_i\| \le \|\bar{\boldsymbol{A}}(k)\|\|\boldsymbol{K}_i\| \le Mh$, we obtain

$$\begin{aligned}\|\bar{\boldsymbol{e}}(k+1)\| &\le \|\boldsymbol{M}_p(k)\|\|\bar{\boldsymbol{e}}(k)\| + \|\boldsymbol{G}(k)\|\|\tilde{\boldsymbol{s}}(k)\| + \|\boldsymbol{w}^{\perp}(k)\| \\ &\le q\|\bar{\boldsymbol{e}}(k)\| + Mh\|\tilde{\boldsymbol{s}}(k)\| + \|\boldsymbol{w}^{\perp}(k)\|,\end{aligned} \tag{43}$$

$$\|\tilde{\boldsymbol{s}}(k+1)\| \le \|\bar{\boldsymbol{e}}(k)\| + \|\tilde{\boldsymbol{s}}(k)\| + \|d(k)\|. \tag{44}$$

Introduce

$$z(k) := \begin{bmatrix}\|\bar{\boldsymbol{e}}(k)\| \\ \|\tilde{\boldsymbol{s}}(k)\|\end{bmatrix}, \qquad H := \begin{bmatrix}q & Mh \\ 1 & 1\end{bmatrix}, \qquad v(k) := \begin{bmatrix}\|\boldsymbol{w}^{\perp}(k)\| \\ \|\boldsymbol{d}(k)\|\end{bmatrix}.$$

Then Eqn. 43- 44 give the comparison system

$$z(k+1) \le Hz(k) + v(k). \tag{45}$$

Expanding Eqn. 45 recursively yields

$$z(k) \le H^k z(0) + \sum_{i=0}^{k-1} H^{k-1-i} v(i). \tag{46}$$

Consider the characteristic equation of H:

$$(\lambda - q)(\lambda - 1) - Mh = 0 \iff \lambda^2 - (q+1)\lambda + (q + Mh) = 0.$$

Since $q + Mh < 1$, the maximal root $\lambda^\star$ satisfies $\lambda^\star < 1$, hence the spectral radius $\rho(H) < 1$.

Let $r := \rho(H) < 1$ be the spectral radius of H. By the Gelfand formula for induced (operator) norms,

$$\lim_{k\to\infty} \|H^k\|^{1/k} = r \qquad \text{(Horn \& Johnson, 2012, p. 349).}$$

Fix any $\rho \in (r, 1)$. Then, by the definition of the limit, there exists $N \in \mathbb{N}$ such that

$$\|H^k\|^{1/k} \le \rho \quad \text{for all } k \ge N \implies \|H^k\| \le \rho^k \quad \forall k \ge N.$$

Define the constant

$$C := \max\Big\{1, \max_{0 \le k \le N} \|H^k\|\rho^{-k}\Big\}.$$

Then:

- If $k \geq N$, we have $\|H^k\|\rho^{-k} \leq 1 \leq C$, hence $\|H^k\| \leq C\rho^k$.
- If $0 \leq k \leq N$, we have $\|H^k\|\rho^{-k} \leq \max_{0\leq k\leq N}\|H^k\|\rho^{-k} \leq C$, hence $\|H^k\| \leq C\rho^k$.

Therefore,

$$\|H^k\| \leq C\rho^k \qquad \text{for all } k \geq 0. \tag{47}$$

Applying Eqn. 47 to Eqn. 46 gives

$$\begin{aligned}\|z(k)\| &\leq \|H^k\|\|z(0)\| + \sum_{i=0}^{k-1}\|H^{k-1-i}\|\|v(i)\| \\ &\leq C\rho^k\|z(0)\| + C\sum_{i=0}^{k-1}\rho^{k-1-i}\|v(i)\| \\ &\leq C\rho^k\|z(0)\| + \frac{C}{1-\rho}\|v\|_\infty,\end{aligned} \tag{48}$$

where $\|v\|_\infty := \sup_{i\geq 0}\|v(i)\|$.

By construction,

$$\|\tilde{\zeta}_{PI}(k)\| = \left\|\begin{bmatrix}\bar{\boldsymbol{e}}(k)\\ \tilde{\boldsymbol{s}}(k)\end{bmatrix}\right\| = \left(\|\bar{\boldsymbol{e}}(k)\|^2 + \|\tilde{\boldsymbol{s}}(k)\|^2\right)^{1/2} = \|z(k)\|.$$

Combining this identity with Eqn. 48, the ISS estimate follows with

$$\beta(s,k) := C\rho^k s \in \mathcal{KL}, \qquad \gamma(s) := \frac{C}{1-\rho}s \in \mathcal{K},$$

which proves that the PI closed loop Eqn. 42 is ISS. □

The integral part exactly cancels the matched disturbance component $\boldsymbol{w}^{\|}$. The remaining error is due only to the unmatched component $\boldsymbol{w}^{\perp}$, which cannot be compensated, and to the variation rate $\boldsymbol{d}(k)$ when $\bar{\boldsymbol{A}}(k)$ and $\boldsymbol{w}(k)$ change over time. In the best scenario, if a steady state exists, i.e., $\bar{\boldsymbol{A}}(k) \to \bar{\boldsymbol{A}}$ and $\boldsymbol{w}(k) \to \boldsymbol{w}$ with $\boldsymbol{w} \in \mathrm{Im}\bar{\boldsymbol{A}}$, then $\boldsymbol{w}^{\perp} \equiv 0$, $d \equiv 0$, and thus $\bar{\boldsymbol{e}}(k) \to 0$.

Remark 2 (Convergence rate versus $\boldsymbol{K}_i$) *From proposition 3, the convergence rate of $\tilde{\zeta}_{PI}(t)$ depends on ρ: the smaller ρ, the faster the convergence. We also adopt the convention (as in the proof) that $\rho \in (r,1)$. Equivalently, we examined $r(h) = \rho(H)$ and proved that with $h = \frac{(1-q)^2}{4M}$ this quantity is minimized.*

Proof Consider the characteristic polynomial of H:

$$\lambda^2 - (q+1)\lambda + (q+Mh) = 0.$$

Its discriminant is

$$\Delta(h) = (q+1)^2 - 4(q+Mh) = (q-1)^2 - 4Mh.$$

If $\Delta(h) \geq 0$ (i.e., $0 \leq h \leq \frac{(1-q)^2}{4M}$), then

$$r(h) = \frac{q+1+\sqrt{\Delta(h)}}{2},$$

and $r(h)$ decreases as h increases.

If $\Delta(h) < 0$ (i.e., $\frac{(1-q)^2}{4M} < h < \frac{1-q}{M}$), then

$$r(h) = \sqrt{q+Mh},$$

and $r(h)$ decreases as h decreases.

Hence $r(h)$ can achieve its best (smallest) value at

$$h = \frac{(1-q)^2}{4M}, \tag{49}$$

for which the error converges to zero the fastest. □

Nevertheless, as we discuss in the next section, in some situations such a large value of h may become a practical obstacle for PI control.

B.6 Oveshoot Mechanism

Phenomenon. A common issue in standard PI settings is *overshooting*: the closed loop oscillates around the setpoint before settling (see Åström & Hägglund (1995a, Ch. 3, §3.3)). In our terms, the integral part accumulates past error and can push the output beyond the setpoint; subsequent sign changes of the error gradually "discharge" the integral, producing a decaying oscillation. The big overshoot is undesirable when we prefer a more stable response near zero. Below we analyze the same mechanism for our PI steering setting.

Citing an observation from Vu & Nguyen (2025): in the *absence* of steering, the cosine similarity between error vectors at different layers is consistently *positive*, i.e.,

$$\cos\angle\big(\bar{\boldsymbol{e}}(i),\bar{\boldsymbol{e}}(j)\big) > 0 \quad \text{for all layers } i,j,$$

so the layerwise errors share (approximately) the same direction. Consequently, with $\{\bar{\boldsymbol{x}}^+(k)\}$ serving as the trajectory setpoints and $\{\bar{\boldsymbol{x}}^-(k)\}$ the system output, an *overshoot event* occurs when the instantaneous error reverses its initial orientation, namely when

$$\langle \bar{\boldsymbol{e}}(k),\bar{\boldsymbol{e}}(0)\rangle < 0.$$

We now introduce the definitions used below.

Scalarization along a direction. Let

$$v := \frac{\bar{\boldsymbol{e}}(0)}{\|\bar{\boldsymbol{e}}(0)\|} \qquad \text{and project onto } v: \qquad \boldsymbol{e}_v(k) := v^\top \bar{\boldsymbol{e}}(k), \quad \boldsymbol{s}_v(k) := v^\top \tilde{\boldsymbol{s}}(k).$$

From the PI loop dynamics Eqn. 42 we obtain the scalar PI pair

$$\boldsymbol{e}_v(k+1) = a(k)\boldsymbol{e}_v(k) - b(k)\boldsymbol{s}_v(k) + \boldsymbol{w}_v^\perp(k), \tag{50}$$
$$\boldsymbol{s}_v(k+1) = \boldsymbol{s}_v(k) + \boldsymbol{e}_v(k) - \boldsymbol{d}_v(k), \tag{51}$$

with $a(k) = v^\top \bar{\boldsymbol{A}}(k)(I - \boldsymbol{K}_p)v = v^\top \boldsymbol{M}_p(k)v$, $b(k) = v^\top \bar{\boldsymbol{A}}(k)\boldsymbol{K}_i v = v^\top \boldsymbol{G}(k)v$, and projected disturbances $\boldsymbol{w}_v^\perp(k) := v^\top \boldsymbol{w}^\perp(k)$, $\boldsymbol{d}_v(k) := v^\top \boldsymbol{d}(k)$. Empirically (and consistently with the angular-steering observation in our setup), we have $v^\top \bar{\boldsymbol{A}}(k)v \geq 0$ for all k; together with the gain $\boldsymbol{K}_i = hI$ with $h \geq 0$, this implies $b(k) \geq 0$.

Also, the assumption $q := \sup_k \|\boldsymbol{M}_p(k)\|$ and $M := \sup_k \|\bar{\boldsymbol{A}}(k)\|$ yeilds

$$a(k) \leq q < 1, \qquad 0 \leq b(k) \leq Mh, \tag{52}$$

Since the system Eqn. 42 is ISS, so is the system Eqn. 50-Eqn. 51. In other words, both $\boldsymbol{e}_v(k)$ and $\boldsymbol{s}_v(k)$ decay. Recall that

$$\boldsymbol{s}_v(k) = \sum_{i=0}^{k-1} \boldsymbol{e}_v(i).$$

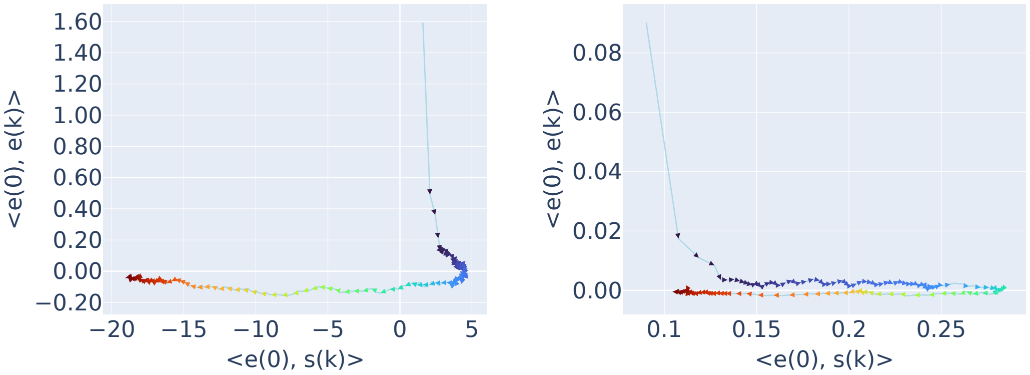
Hence, $\boldsymbol{s}_v(k)$ can only decrease when $\boldsymbol{e}_v(k) < 0$, which is precisely the moment when overshoot occurs. These overshooting and decaying phenomena are observed in empirical simulation, see Fig. 7. Below, we define the overshoot in our setting.

Definition 2 (Overshoot and its amplitude) *We say an* overshoot *occurs from time k_a to k_a+m if $\boldsymbol{e}_v(k) < 0\, \forall k = k_a, k_a+1, \ldots, k_a+m-1$ and $\boldsymbol{e}_v(k) \geq 0$ for $k = k_a-1, k_a+m$. Its amplitude is defined as*

$$A_a := \max_{kt_a \leq i \leq k_a+m-1} |\boldsymbol{e}_v(i)|\,. \tag{53}$$

In standard PID settings illustrated in Åström & Hägglund (1995a, Ch. 3, §3.3)), it is observed that the overshoot amplitude decays over time. This decay is also consistent with the ISS property of the closed loop: as both $\boldsymbol{e}_v(t)$ and $\boldsymbol{s}_v(t)$ are driven down, subsequent oscillations tend to diminish in magnitude. In our simulation (see Fig. 7), the first overshoot appear to be representative. Hence, while we are not yet able to provide a formal proof, the empirical evidence and ISS intuition justify the first overshoot which is typically the dominant one and serves as a representative indicator of oscillatory behavior.

This assumption is for Proposition 4. Suppose that $a(t) \geq 0$; equivalently, $p \in (1-\frac{1}{M}, 1]$, which is the result of proposition 1. This assumption is expected to entail no loss of generality relative to the $|a(t)| \leq q < 1$ assumption.



(a) PI

(b) PID

Figure 7: Scalar errors across time step of randomly initialized model after applying PI and PID controller. Colors from blue to red denote the time (layer) dimension.

Proposition 4 (Agressive PI gain leads to a large first overshoot) *Let k_0 be the first sign-change time, i.e.,*

$$\boldsymbol{e_v}(j) \geq 0 \;\; \forall j = 0,1,\ldots,k_0-1, \qquad \boldsymbol{e_v}(k_0) < 0,$$

and let k_1 be the first time the trajectory returns to the nonnegative side,

$$\boldsymbol{e_v}(k_1) \geq 0, \qquad \boldsymbol{e_v}(i) < 0 \;\; \forall i = k_0, k_0+1, \ldots, k_1-1.$$

As in Eqn.53, the first overshoot amplitude is

$$A_0 \;=\; \max_{k_0 \leq i \leq k_1-1} |\boldsymbol{e_v}(i)| \;=\; |\boldsymbol{e_v}(i_{max})|. \tag{54}$$

Assume $\sup_k \|\bar{\boldsymbol{A}}(k)\| \leq M < \infty$ and $\sup_k \|\boldsymbol{M}_p(k)\| \leq q < 1$. Denote $\|\boldsymbol{K}_i\| =: h$. and given $q + Mh < 1$. Therefore,

$$A_0 \;\leq\; Mh\Big(\frac{1}{1-q} + \frac{1}{(1-q)^2}\Big)\boldsymbol{e_v}(0) \;+\; \Big(\frac{Mh}{1-q}(k_0-1) + \frac{Mh}{1-q}\Big)\|\boldsymbol{d}\|_\infty + \frac{Mh(k_0-1)+1}{1-q}\|\boldsymbol{w}\|_\infty; \tag{55}$$

Proof. Before the first crossing ($j \leq k_0 - 1$) we have $\boldsymbol{s}_v(j) \geq 0$, hence from Eqn. 50

$$\boldsymbol{e}_v(j+1) = a(j)\boldsymbol{e}_v(j) - b(j)\boldsymbol{s}_v(j) + \boldsymbol{w}_v^\perp(j) \;\leq\; a(j)\boldsymbol{e}_v(j) + \boldsymbol{w}_v^\perp(j) \;\leq\; q\boldsymbol{e}_v(j) + \|\boldsymbol{w}\|_\infty,$$

so by induction $\boldsymbol{e}_v(j) \leq q^j \boldsymbol{e}_v(0) + \frac{1}{1-q}\|\boldsymbol{w}\|_\infty$. Summing Eqn. 51,

$$\boldsymbol{s}_v(k_0-1) \;=\; \sum_{i=0}^{k_0-2} \boldsymbol{e}_v(i) \;-\; \sum_{i=0}^{k_0-2} \boldsymbol{d}_v(i) \;\leq\; \frac{\boldsymbol{e}_v(0)}{1-q} \;+\; (k_0-1)\|\boldsymbol{d}\|_\infty + \frac{k_0-1}{1-q}\|\boldsymbol{w}\|_\infty. \tag{56}$$

Since $a(k) \geq 0$ and $\boldsymbol{e}_v(k_0-1) \geq 0$, at the crossing step,

$$\begin{aligned} |\boldsymbol{e}_v(k_0)| \;=\; -\boldsymbol{e}_v(k_0) \;&\leq\; b(k_0-1)\boldsymbol{s}_v(k_0-1) \;+\; \|\boldsymbol{w}\|_\infty \\ &\leq Mh\left(\frac{\boldsymbol{e}_v(0)}{1-q} + (k_0-1)\|\boldsymbol{d}\|_\infty + \frac{k_0-1}{1-q}W_\infty\right) + \|\boldsymbol{w}\|_\infty. \\ &= \frac{Mh}{1-q}\boldsymbol{e}_v(0) + Mh(k_0-1)\|\boldsymbol{d}\|_\infty + (Mh\frac{k_0-1}{1-q}+1)\|\boldsymbol{w}\|_\infty \end{aligned} \tag{57}$$

Assume that $\boldsymbol{d}_v(k)$ is small enough s.t during the overshoot time, $\boldsymbol{s}_v$ is nonincreasing (since $\boldsymbol{e}_v < 0$ a.e. on $[k_0, k_1-1]$), so $\boldsymbol{s}_v(i) \leq \boldsymbol{s}_v(k_0-1)$ for $i \in [k_0, k_1-1]$. Using Eqn. 50 again and unrolling m steps from k_0,

$$\begin{aligned} |\boldsymbol{e}_v(k_0+m)| \;&\leq\; q^m |\boldsymbol{e}_v(k_0)| \;+\; \sum_{k=0}^{m-1} q^k \big(Mh\boldsymbol{s}_v(k_0-1) + \|\boldsymbol{w}\|_\infty\big) \\ &\leq\; |\boldsymbol{e}_v(k_0)| \;+\; \frac{Mh\boldsymbol{s}_v(k_0-1) + \|\boldsymbol{w}\|_\infty}{1-q}. \end{aligned} \tag{58}$$

Taking the maximum over $m\in\{0,1,...,k_1-k_0\}$ and substituting Ineq. 56 and Ineq. 57 into Ineq. 58 yields

$$A_0 \le Mh\Big(\frac{1}{1-q}+\frac{1}{(1-q)^2}\Big)\boldsymbol{e}_v(0) + \Big(\frac{Mh}{1-q}(k_0-1)+\frac{Mh}{1-q}\Big)\|\boldsymbol{d}\|_\infty + \frac{Mh(k_0-1)+1}{1-q}\|\boldsymbol{w}\|_\infty \tag{59}$$

□

Consequently, the right-hand side of Ineq.59 is *monotone increasing in* h (via the factor Mh) and increases as q decreases (through the factors $\frac{1}{1-q}$). In particular, more aggressive PI leads to a larger first-overshoot amplitude.

Remark 3 (”Fast-PI” specialization.) *With the tuning used in our analysis in remark 2,* $h=\frac{(1-q)^2}{4M}$ *(the value that minimizes the comparison-system rate), so* $Mh=\frac{(1-q)^2}{4}$*. Plugging into equation 59 gives*

$$\boldsymbol{A}_0 \le \Big(\frac{1-q}{4}+\frac{1}{4}\Big)\boldsymbol{e}_v(0) + \Big(\frac{1-q}{4}(k_0-1)+\frac{1}{4}\Big)\|\boldsymbol{d}\|_\infty + \frac{k_0}{1-q}\|\boldsymbol{w}\|_\infty.$$

In particular, in the disturbance-free case ($\|\boldsymbol{w}\|_\infty=\|\boldsymbol{d}\|_\infty=0$*) we obtain*

$$\boldsymbol{A}_0 \le \Big(\frac{1-q}{4}+\frac{1}{4}\Big)\boldsymbol{e}_v(0),$$

so stronger proportional action (smaller q*) comes with a larger first-overshoot envelope, even though the closed-loop settles faster.*

B.7 PID Control

B.7.1 Stability of PID closed loop

We consider the PID update

$$\boldsymbol{u}(k) = \boldsymbol{K}_p\bar{\boldsymbol{e}}(k) + \boldsymbol{K}_i\boldsymbol{s}(k) + \boldsymbol{K}_d\big(\bar{\boldsymbol{e}}(k)-\bar{\boldsymbol{e}}(k-1)\big),$$

and define the auxiliary matrices

$$\boldsymbol{M}_p(k) := \bar{\boldsymbol{A}}(k)\big(I-\boldsymbol{K}_p\big), \qquad \boldsymbol{G}(k) := \bar{\boldsymbol{A}}(k)\boldsymbol{K}_i, \qquad \boldsymbol{H}(k) := \bar{\boldsymbol{A}}(k)\boldsymbol{K}_d,$$

together with the error increment

$$\Delta\bar{\boldsymbol{e}}(k) := \bar{\boldsymbol{e}}(k)-\bar{\boldsymbol{e}}(k-1), \qquad \Delta\bar{\boldsymbol{e}}(-1)=0$$

Using the plant relation, we obtain

$$\Delta\bar{\boldsymbol{e}}(k+1) = \big(\boldsymbol{M}_p(k)-I\big)\bar{\boldsymbol{e}}(k) - \boldsymbol{G}(k)\tilde{\boldsymbol{s}}(k) - \boldsymbol{H}(k)\Delta\bar{\boldsymbol{e}}(k) + \boldsymbol{w}^{\perp}(k), \tag{60}$$

Introduce the lifted state from the auxiliary PI state in Eqn. 42

$$\tilde{\zeta}_{\mathrm{PID}}(k) := \begin{bmatrix} \bar{\boldsymbol{e}}(k) \\ \tilde{\boldsymbol{s}}(k) \\ \Delta\bar{\boldsymbol{e}}(k) \end{bmatrix},$$

Then the closed-loop evolution reads

$$\tilde{\zeta}_{\mathrm{PID}}(k+1) = \boldsymbol{M}_d(k)\tilde{\zeta}_{\mathrm{PID}}(k) + \tilde{\boldsymbol{w}}_{\mathrm{PID}}(k), \tag{61}$$

where

$$\boldsymbol{M}_d(k) := \begin{bmatrix} \boldsymbol{M}_p(k) & -\boldsymbol{G}(k) & -\boldsymbol{H}(k) \\ I & I & 0 \\ \boldsymbol{M}_p(k)-I & -\boldsymbol{G}(k) & -\boldsymbol{H}(k) \end{bmatrix}, \qquad \tilde{\boldsymbol{w}}_{\mathrm{PID}}(k) := \begin{bmatrix} \boldsymbol{w}^{\perp}(k) \\ -\boldsymbol{d}(k) \\ \boldsymbol{w}^{\perp}(k) \end{bmatrix}.$$

Theorem 1 (Stabilizing the PID loop preserves bias removal) *Let* $\boldsymbol{M}_p(k) = \bar{\boldsymbol{A}}(k)(\boldsymbol{I}-\boldsymbol{K}_p)$*, and denote* $\|\boldsymbol{K}_i\| =: h$*,* $\|\boldsymbol{K}_d\| =: \ell$*. Assume* $\sup_k\|\bar{\boldsymbol{A}}(k)\| \le M < \infty$ *and* $\sup_k\|\boldsymbol{M}_p(k)\| \le q < 1$*. If* $q+Mh<1$ *(stable PI loop), then there exists* $\ell>0$ *such that the PID closed-loop control is ISS. Therefore, the integral part in PID design still cancels the matched disturbance component* $\boldsymbol{w}^{\parallel}$*.*

Proof. We establish the ISS for system 61 using the method of ISS-Lyapunov function, see (Jiang et al., 1999, Def. 2.2, Prop. 2.3). It then suffices to construct a candidate ISS-Lyapunov function $\boldsymbol{V}_{\mathrm{PID}}(k)$ satisfying that there exist class $\mathcal{K}_\infty$ functions α_1,α_2,α_3 and a class $\mathcal{K}$ function σ such that

$$\alpha_1(\|\tilde{\zeta}_{\mathrm{PID}}(k)\|) \le \boldsymbol{V}_{\mathrm{PID}}(\tilde{\zeta}_{\mathrm{PID}}(k)) \le \alpha_2(\|\tilde{\zeta}_{\mathrm{PID}}(k)\|), \tag{62}$$

and

$$\boldsymbol{V}(\tilde{\zeta}_{\mathrm{PID}}(k+1)) - \boldsymbol{V}(\tilde{\zeta}_{\mathrm{PID}}(k)) \le -\alpha_3(\|\tilde{\zeta}_{\mathrm{PID}}(k)\|) + \sigma(\|\boldsymbol{w}\|). \tag{63}$$

Step 1: Candidtate $\boldsymbol{V}_{PID}(k)$ and PI-closed loop baseline

Define

$$\boldsymbol{V}_{\mathrm{PID}}(k) := \boldsymbol{V}_{\mathrm{PI}}\big(\tilde{\zeta}_{\mathrm{PI}}(k), k\big) + r\|\Delta\bar{e}(k)\|^2, \qquad r > 0, \tag{64}$$

where $\boldsymbol{V}_{\mathrm{PI}}(\tilde{\zeta}_{\mathrm{PI}}, k) = \tilde{\zeta}_{\mathrm{PI}}^\top \boldsymbol{P}(k)\tilde{\zeta}_{\mathrm{PI}}$ with $\boldsymbol{P}(k) = \boldsymbol{P}(k)^\top \succ 0$ and there exists some $\mu_{\mathrm{PI}} > 0$ such that

$$\boldsymbol{M}_i(k)^\top \boldsymbol{P}(k) \boldsymbol{M}_i(k) - \boldsymbol{P}(k) \le -\mu_{\mathrm{PI}} I, \qquad \forall k, \tag{65}$$

Regarding the existence of such $\boldsymbol{V}_{\mathrm{PI}}$, recall the homogeneous PI-loop $\tilde{\zeta}_{\mathrm{PI}}(k+1) = \boldsymbol{M}_i(k)\tilde{\zeta}_{\mathrm{PI}}(k)$ with $\boldsymbol{M}_i(k) = \begin{bmatrix} \boldsymbol{M}_p(k) & -\boldsymbol{G}(k) \\ I & I \end{bmatrix}$, being asymptotically stable. Suppose there is $\boldsymbol{Q}(k) = \boldsymbol{Q}(k)^\top \succeq 0$ bounded so that the pair $(\boldsymbol{M}_i(k), \sqrt{\boldsymbol{Q}(k)})$ is observable for all k, hence the difference Lyapunov equation

$$\boldsymbol{M}_i^\top(k)\boldsymbol{P}(k+1)\boldsymbol{M}_i(k) - \boldsymbol{P}(k) = -\boldsymbol{Q}(k)$$

admits a unique positive definite solution $\boldsymbol{P}(k) = \boldsymbol{P}^\top(k) \succ 0$ for all k, and a uniform bound $\|\boldsymbol{P}\|_\infty := \sup_k \|\boldsymbol{P}(k)\| < \infty$ (see (Gajic & Qureshi, 2008, Ch. 4, p. 110))

Step 2: Condition (i) as in Eqn. 62

We write $\boldsymbol{V}_{\mathrm{PID}}$ as a quadratic form

$$\boldsymbol{V}_{\mathrm{PID}}(k) = \tilde{\zeta}_{\mathrm{PID}}(k)^\top \boldsymbol{P}_*(k)\tilde{\zeta}_{\mathrm{PID}}(k), \qquad \boldsymbol{P}_*(k) := \begin{bmatrix} \boldsymbol{P}(k) & 0 \\ 0 & rI \end{bmatrix}. \tag{66}$$

Clearly $\boldsymbol{P}_*(k) = \boldsymbol{P}_*(k)^\top$ and $\boldsymbol{P}_*(k) \succ 0$ because $\boldsymbol{P}(k) \succ 0$ and $rI \succ 0$; hence $\boldsymbol{P}_*(k)$ is symmetric positive definite for all k.

By the spectral theorem, there exists an orthogonal matrix $U_*(k)$ and a diagonal $\Lambda_*(k) = \mathrm{diag}(\lambda_1(k), \dots, \lambda_{n_*}(k))$ with positive entries such that $\boldsymbol{P}_*(k) = U_*(k)\Lambda_*(k)U_*(k)^\top$. Moreover, because $\boldsymbol{P}_*(k)$ is block diagonal, its eigenvalues are precisely the union of the eigenvalues of $\boldsymbol{P}(k)$ and the repeated eigenvalue r. Using the uniform bounds already established for $\boldsymbol{P}(k)$ (there exists $\underline{\lambda} > 0$ with $\lambda_{\min}(\boldsymbol{P}(k)) \ge \underline{\lambda}$ and $\lambda_{\max}(\boldsymbol{P}(k)) \le \|\boldsymbol{P}\|_\infty := \sup_k \|\boldsymbol{P}(k)\| < \infty$), we obtain the k-independent bounds

$$\lambda_{\min}\big(\boldsymbol{P}_*(k)\big) \ge \underline{\lambda}_* := \min\{\underline{\lambda}, r\} > 0, \qquad \lambda_{\max}\big(\boldsymbol{P}_*(k)\big) \le \overline{\lambda}_* := \max\{|\boldsymbol{P}\|_\infty, r\} < \infty.$$

For every vector z and every symmetric positive definite M, $\lambda_{\min}(M)\|z\|^2 \le z^\top M z \le \lambda_{\max}(M)\|z\|^2$. Applying this to $M = \boldsymbol{P}_*(k)$ and $z = \tilde{\zeta}_{\mathrm{PID}}(k)$ in Eqn. 66 yields

$$\underline{\lambda}_* \big\|\tilde{\zeta}_{\mathrm{PID}}(k)\big\|^2 \le \boldsymbol{V}_{\mathrm{PID}}(k) \le \overline{\lambda}_* \big\|\tilde{\zeta}_{\mathrm{PID}}(k)\big\|^2.$$

Therefore, choosing the class-$\mathcal{K}_\infty$ functions

$$\alpha_1(\boldsymbol{s}) := \underline{\lambda}_* \boldsymbol{s}^2, \qquad \alpha_2(\boldsymbol{s}) := \overline{\lambda}_* \boldsymbol{s}^2,$$

we obtain the desired bound

$$\alpha_1\Big(\big\|\tilde{\zeta}_{\mathrm{PID}}(k)\big\|\Big) \le \boldsymbol{V}_{\mathrm{PID}}(k) \le \alpha_2\Big(\big\|\tilde{\zeta}_{\mathrm{PID}}(k)\big\|\Big), \tag{67}$$

which establishes condition (i) in 62.

Step 3: Condition (ii) as in Eqn. 63.

From Eqn. 64,

$$\Delta\boldsymbol{V}_{\mathrm{PID}}(k) = \underbrace{\Delta\boldsymbol{V}_{\mathrm{PI}}\Big(\tilde{\zeta}_{\mathrm{PI}}(k)\Big)\Big|_{\mathrm{PID}}}_{\text{PI part under PID update}} + r\big(\|\Delta\bar{e}(k+1)\|^2 - \|\Delta\bar{e}(k)\|^2\big). \tag{68}$$

Bounding the PI part under the PID update.

Under PI rule ($\boldsymbol{K}_d = 0$),

$$\tilde{\zeta}_{\mathrm{PI}}(k+1)\big|_{\mathrm{PI}} = \boldsymbol{M}_i(k)\tilde{\zeta}_{\mathrm{PI}}(k) + \tilde{\boldsymbol{w}}_{\mathrm{PI}}(k), \qquad \boldsymbol{M}_i(k) := \begin{bmatrix} \boldsymbol{M}_p(k) & -\boldsymbol{G}(k) \\ I & I \end{bmatrix}$$

then

$$\begin{aligned} \Delta \boldsymbol{V}_{\mathrm{PI}}\Big(\tilde{\zeta}_{\mathrm{PI}}(k)\Big)\Big|_{\mathrm{PI}} &= \tilde{\zeta}_{\mathrm{PI}}(k)^\top \big(\boldsymbol{M}_i(k)^\top \boldsymbol{P}(k+1)\boldsymbol{M}_i(k) - \boldsymbol{P}(k)\big)\tilde{\zeta}_{\mathrm{PI}}(k) \\ &\quad + 2\tilde{\zeta}_{\mathrm{PI}}(k)^\top \boldsymbol{M}_i(k)^\top \boldsymbol{P}(k+1)\tilde{\boldsymbol{w}}_{\mathrm{PI}}(k) + \tilde{\boldsymbol{w}}_{\mathrm{PI}}(k)^\top \boldsymbol{P}(k+1)\tilde{\boldsymbol{w}}_{\mathrm{PI}}(k) \end{aligned}$$

By Eqn. 65, Cauchy-Schwarz inequality and Young's inequality,

$$\begin{aligned} \Delta \boldsymbol{V}_{\mathrm{PI}}\Big(\tilde{\zeta}_{\mathrm{PI}}(k)\Big)\Big|_{\mathrm{PI}} &\le -\mu_{\mathrm{PI}}\|\tilde{\zeta}_{\mathrm{PI}}(k)\|^2 + 2\|\boldsymbol{M}_i\|_\infty \|\boldsymbol{P}\|_\infty \|\tilde{\zeta}_{\mathrm{PI}}(k)\| \|\tilde{\boldsymbol{w}}_{\mathrm{PI}}(k)\| + \|\boldsymbol{P}\|_\infty \|\tilde{\boldsymbol{w}}_{\mathrm{PI}}(k)\|^2 \\ &\le -(\mu_{\mathrm{PI}} - \varepsilon_1)\|\tilde{\zeta}_{\mathrm{PI}}(k)\|^2 + \Big(\frac{\|\boldsymbol{M}_i\|_\infty^2 \|\boldsymbol{P}\|_\infty^2}{\varepsilon_1} + \|\boldsymbol{P}\|_\infty\Big)\|\tilde{\boldsymbol{w}}_{\mathrm{PI}}(k)\|^2 \\ &= -\mu^*_{\mathrm{PI}}\|\tilde{\zeta}_{\mathrm{PI}}(k)\|^2 + C_1\|\tilde{\boldsymbol{w}}_{\mathrm{PI}}(k)\|^2, \end{aligned}$$

for any $\varepsilon_1 > 0$.

Under PID rule ($\boldsymbol{K}_d \neq 0$),

$$\tilde{\zeta}_{\mathrm{PI}}(k+1)\big|_{\mathrm{PID}} = \boldsymbol{M}_i(k)\tilde{\zeta}_{\mathrm{PI}}(k) - \delta(k) + \boldsymbol{w}_{\mathrm{PI}}(k),$$

with the "perturbation"

$$\delta(k) := \begin{bmatrix} H(k)\Delta\bar{\boldsymbol{e}}(k) \\ 0 \end{bmatrix}$$

Hence,

$$\begin{aligned} \Delta \boldsymbol{V}_{\mathrm{PI}}\big(\tilde{\zeta}_{\mathrm{PI}}(k)\big)\Big|_{\mathrm{PID}} &= \boldsymbol{V}_{\mathrm{PI}}\big(\tilde{\zeta}_{\mathrm{PI}}(k+1)\big|_{\mathrm{PI}}\big) - \boldsymbol{V}_{\mathrm{PI}}\big(\tilde{\zeta}_{\mathrm{PI}}(k)\big) \\ &= \boldsymbol{V}_{\mathrm{PI}}\big(\tilde{\zeta}_{\mathrm{PI}}(k+1)\big|_{\mathrm{PI}}\big) - \boldsymbol{V}_{\mathrm{PI}}\big(\tilde{\zeta}_{\mathrm{PI}}(k)\big) + 2\big(\tilde{\zeta}_{\mathrm{PI}}(k+1)\big|_{\mathrm{PI}}\big)^\top \boldsymbol{P}\delta(k) + \delta(k)^\top \boldsymbol{P}\delta(k) \end{aligned} \tag{69}$$

Bounding each term in Eqn. 69

- $\boldsymbol{V}_{\mathrm{PI}}\big(\tilde{\zeta}_{\mathrm{PI}}(k+1)\big|_{\mathrm{PI}}\big) - \boldsymbol{V}_{\mathrm{PI}}\big(\tilde{\zeta}_{\mathrm{PI}}(k)\big) \le -\mu^*_{\mathrm{PI}}\|\tilde{\zeta}_{\mathrm{PI}}(k)\|^2 + C_1\|\tilde{\boldsymbol{w}}_{\mathrm{PI}}(k)\|$
- Applying Young's inequality for inner product, there exists $\varepsilon > 0$ s.t

$$2\big(\tilde{\zeta}_{\mathrm{PI}}(k+1)\big|_{\mathrm{PI}}\big)^\top \boldsymbol{P}\delta(k) \le \varepsilon \Big\|\tilde{\zeta}_{\mathrm{PI}}(k+1)\big|_{\mathrm{PI}}\Big\|^2 + \tfrac{1}{\varepsilon}\delta(k)^\top \boldsymbol{P}\delta(k) \tag{70}$$

$$\le \varepsilon \|\boldsymbol{P}\| \Big\|\tilde{\zeta}_{\mathrm{PI}}(k+1)\big|_{\mathrm{PI}}\Big\|^2 + \tfrac{\|\boldsymbol{P}\|}{\varepsilon} M^2 \ell^2 \|\Delta\bar{\boldsymbol{e}}(k)\|^2 \tag{71}$$

  Since $\Big\|\tilde{\zeta}_{\mathrm{PI}}(k+1)\big|_{\mathrm{PI}}\Big\|^2 \le 2\|\boldsymbol{M}_i\|_\infty^2 \|\tilde{\zeta}_{\mathrm{PI}}(k)\|^2 + 2\|\tilde{\boldsymbol{w}}_{\mathrm{PI}}(k)\|^2$

$$\begin{aligned} \Rightarrow 2\big(\tilde{\zeta}_{\mathrm{PI}}(k+1)\big|_{\mathrm{PI}}\big)^\top \boldsymbol{P}\delta(k) &\le 2\varepsilon\|\boldsymbol{P}\|\|\boldsymbol{M}_i\|_\infty^2 \|\tilde{\zeta}_{\mathrm{PI}}(k)\|^2 + 2\varepsilon\|\boldsymbol{P}\|\|\tilde{\boldsymbol{w}}_{\mathrm{PI}}(k)\|^2 \\ &\quad + \tfrac{\|\boldsymbol{P}\|}{\varepsilon} M^2\ell^2 \|\Delta\bar{\boldsymbol{e}}(k)\|^2 \\ &= 2\varepsilon\|\boldsymbol{P}\|\|\boldsymbol{M}_i\|_\infty^2 \|\tilde{\zeta}_{\mathrm{PI}}(k)\|^2 + \tfrac{\|\boldsymbol{P}\|}{\varepsilon} M^2\ell^2 \|\Delta\bar{\boldsymbol{e}}(k)\|^2 \\ &\quad + C_2\|\tilde{\boldsymbol{w}}_{\mathrm{PI}}(k)\|^2, \end{aligned}$$

  where $C_2 = 2\varepsilon\|\boldsymbol{P}\|$
- $\delta(k)^\top \boldsymbol{P}\delta(k) \le \|\boldsymbol{P}\|\|\delta(k)\|^2 \le \|\boldsymbol{P}\| M^2\ell^2 \|\Delta\bar{\boldsymbol{e}}(k)\|^2$

Therefore,

$$\begin{aligned} \Delta \boldsymbol{V}_{\mathrm{PI}}\big(\tilde{\zeta}_{\mathrm{PI}}(k)\big)\Big|_{\mathrm{PID}} &\le -\mu^*_{\mathrm{PI}}\|\tilde{\zeta}_{\mathrm{PI}}(k)\|^2 + C_1\|\tilde{\boldsymbol{w}}_{\mathrm{PI}}(k)\|^2 \\ &\quad + 2\varepsilon\|\boldsymbol{P}\|\|\boldsymbol{M}_i\|_\infty^2 \|\tilde{\zeta}_{\mathrm{PI}}(k)\|^2 + \tfrac{\|\boldsymbol{P}\|}{\varepsilon} M^2\ell^2 \|\Delta\bar{\boldsymbol{e}}(k)\|^2 + C_2\|\tilde{\boldsymbol{w}}_{\mathrm{PI}}(k)\|^2 \\ &\quad + \|\boldsymbol{P}\| M^2\ell^2 \|\Delta\bar{\boldsymbol{e}}(k)\|^2 \qquad (72) \\ &= -(\mu^*_{\mathrm{PI}} - 2\varepsilon\|\boldsymbol{P}\|\|\boldsymbol{M}_i\|_\infty^2)\|\tilde{\zeta}_{\mathrm{PI}}(k)\|^2 + \|\boldsymbol{P}\| M^2\ell^2 \Big(\tfrac{1}{\varepsilon} + 1\Big)\|\Delta\bar{\boldsymbol{e}}(k)\|^2 \\ &\quad + C_3\|\tilde{\boldsymbol{w}}_{\mathrm{PI}}(k)\|^2, \end{aligned}$$

where $C_3 = C_1 + C_2$.

Bounding the increment term.

From Eqn. 60 and applying the inequality $(x+y+z)^2 \le 3(x^2+y^2+z^2)$,

$$\|\Delta\bar{\boldsymbol{e}}(k+1)\|^2 \le 3\Big(\|\boldsymbol{M}_p(k)-I\|_\infty^2 + Mh\Big)\|\tilde{\zeta}_{\mathrm{PI}}(k)\|^2 + 3M^2\ell^2\|\Delta\bar{\boldsymbol{e}}(k)\|^2 + 3\|\tilde{\boldsymbol{w}}_{\mathrm{PID}}(k)\|^2,$$

and so

$$\begin{aligned} r\big(\|\Delta\bar{\boldsymbol{e}}(k+1)\|^2 - \|\Delta\bar{\boldsymbol{e}}(k)\|^2\big) \le 3r\Big(\|\boldsymbol{M}_p(k)-I\|_\infty^2 + Mh\Big)\|\tilde{\zeta}_{\mathrm{PI}}(k)\|^2 \\ - r\big(1-3M^2\ell^2\big)\|\Delta\bar{\boldsymbol{e}}(k)\|^2 + 3r\|\tilde{\boldsymbol{w}}_{\mathrm{PID}}(k)\|^2. \end{aligned} \tag{73}$$

Combination

Combining Eqn. 68, Ineq. 72, and Ineq. 73,

$$\begin{aligned} \Delta\boldsymbol{V}_{\mathrm{PID}}(k) \le -\Big(\mu_{\mathrm{PI}}^* - 2\varepsilon\|\boldsymbol{P}\|_\infty\|\boldsymbol{M}_i\|_\infty^2 - 3r\big(\|\boldsymbol{M}_p(k)-I\|_\infty^2 + Mh\big)\Big)\|\tilde{\zeta}_{\mathrm{PI}}(k)\|^2 \\ - \Big(r(1-3M^2\ell^2) - \|\boldsymbol{P}\|_\infty M^2\ell^2\Big(\tfrac{1}{\varepsilon}+1\Big)\Big)\|\Delta\bar{\boldsymbol{e}}(k)\|^2 + C\|\tilde{\boldsymbol{w}}_{\mathrm{PI}}(k)\|^2, \end{aligned}$$

where $C = C_3 + r$. Define

$$S(r,\varepsilon) := \mu_{\mathrm{PI}}^* - 2\varepsilon\|\boldsymbol{P}\|_\infty\|\boldsymbol{M}_i\|_\infty^2 - 3r\big(\|\boldsymbol{M}_p(k)-I\|_\infty^2 + Mh\big), \tag{74}$$

$$T(r,\varepsilon,\ell) := r(1-3M^2\ell^2) - \|\boldsymbol{P}\|_\infty M^2\ell^2\Big(\tfrac{1}{\varepsilon}+1\Big). \tag{75}$$

ISS of the PID loop follows if $S(r,\varepsilon) > 0$ and $T(r,\varepsilon,\ell) > 0$.

Feasible choices.

We are free to choose any $\varepsilon > 0$ and $r > 0$ such that $S(r,\varepsilon) > 0$. One convenient selection is

$$\varepsilon = \frac{\mu_{\mathrm{PI}}^*}{8\|\boldsymbol{P}\|_\infty\|\boldsymbol{M}_i\|_\infty^2}, \qquad r = \frac{\mu_{\mathrm{PI}}^*}{8\big(\|\boldsymbol{M}_p(k)-I\|_\infty^2 + Mh\big)} \quad \Rightarrow \quad S(r,\varepsilon) = \tfrac{3}{8}\mu_{\mathrm{PI}}^* > 0.$$

With ε, r fixed as above, pick $\ell > 0$ small enough to satisfy $T(r,\varepsilon,\ell) > 0$, namely

$$\ell^2 < \frac{r}{\big(\|\boldsymbol{P}\|_\infty\big(\frac{1}{\varepsilon}+1\big) + 3r\big)M^2},$$

Under these choices, $\Delta\boldsymbol{V}_{\mathrm{PID}}(k) \le -\alpha_3\|\zeta_{\mathrm{PI}}(k)\|^2 - \alpha_4\|\Delta\bar{\boldsymbol{e}}(k)\|^2 + \beta\|\boldsymbol{w}_{\mathrm{PI}}(k)\|^2$ for some $\alpha_3, \alpha_4, \beta > 0$, which satisfies condition (ii) as in Eqn. 63 and proves ISS of the PID closed loop. □

B.7.2 Overshooting under PID law of control

Developing from Sec. B.6, we introduce scalar PID recursion along v:

$$\boldsymbol{e}_v(k+1) = a(k)\boldsymbol{e}_v(k) - b(k)\boldsymbol{s}_v(k) - c(k)\Delta\boldsymbol{e}_v(k) + \boldsymbol{w}_v^\perp(k), \qquad \boldsymbol{s}_v(k+1) = \boldsymbol{s}_v(k) + \boldsymbol{e}_v(k) - d_v(k), \tag{76}$$

where

$$a(k) := v^\top \boldsymbol{M}_p(k) v, \qquad b(k) := v^\top \boldsymbol{G}(k) v, \qquad c(k) := v^\top \boldsymbol{H}(k) v,$$
$$\Delta\boldsymbol{e}_v(k) = v^\top \Delta\bar{\boldsymbol{e}}(k), \quad \boldsymbol{w}_v^\perp(k) = v^\top \boldsymbol{w}^\perp(k).$$

By construction $a(k) \le q < 1$, $b(k) \le Mh$, $c(k) \le M\ell$ with $M := \sup_k \|\bar{\boldsymbol{A}}(k)\|$, $h := \|\boldsymbol{K}_i\|$ and $\ell := \|\boldsymbol{K}_d\|$.

We now impose an additional requirement on the derivative gain $\boldsymbol{K}_d$ so that, without the effect of noise, the PID update secures the monotonic decrease of $\boldsymbol{e}_v(k)$ before the first negative peak of scalar error $\boldsymbol{e}_v(k)$.

Remark 4 (Pre-overshoot monotonic decrease of scalar errors) *Assume the setting of Proposition 4 and further suppose the scalar error trajectory before the first largest overshoot under PID law is* smooth *in the sense that there exists $R \ge 1$ such that*

$$\frac{\boldsymbol{e}_v(k-1)}{\boldsymbol{e}_v(k)} \le R \qquad \textit{for all } k = 1, 2, \ldots, i_{max} - 1, \tag{77}$$

where $\boldsymbol{A}_0 := \max_{k_0 \le i \le k_1} |\boldsymbol{e}_v(i)| = |\boldsymbol{e}_v(i_{max})|$ *from Eqn.*54.

Assume that $\mathbf{w}_v^{\perp} \equiv 0$ *and* $\boldsymbol{d}_v \equiv 0$.

If, in addition, the derivative gain satisfies

$$l = \|\boldsymbol{K}_d\| \le \frac{1-q}{(R-1)M}, \tag{78}$$

then under PID law

$$\boldsymbol{e}_v(k+1) \le \boldsymbol{e}_v(k) \qquad \textit{for all } \ k=0,1,\ldots,i_{max}-1.$$

Proof. Before k_0, we have $\boldsymbol{e}_v(k) > 0$, so $\boldsymbol{s}_v(k) \ge 0$ (since $\boldsymbol{s}_v$ accumulates $\boldsymbol{e}_v$ and $\boldsymbol{s}_v(0) = 0$). For $k_0 \le k \le i_{max}-1$, $\boldsymbol{s}_v(k) \ge 0$ proved in remark 5

Hence

$$\begin{aligned}\boldsymbol{e}_v(k+1) &= a(k)\boldsymbol{e}_v(k) - b(k)\boldsymbol{s}_v(k) - c(k)\big(\boldsymbol{e}_v(k) - \boldsymbol{e}_v(k-1)\big)\\ &\le a(k)\boldsymbol{e}_v(k) + c(k)\big(\boldsymbol{e}_v(k-1) - \boldsymbol{e}_v(k)\big)\\ &\le \Big[a(k) + c(k)(R-1)\Big]\boldsymbol{e}_v(k) \le \Big[q + (R-1)M\ell\Big]\boldsymbol{e}_v(k) \le \boldsymbol{e}_v(k),\end{aligned}$$

where the last inequality is exactly Eqn. 78. □

Note for R*:* In practice, one may estimate a conservative R from PI-law traces and use a small safety factor

Adding Disturbance: With bounded disturbances, the scalar update reads

$$\boldsymbol{e}_v(k+1) \le \big[q + (R-1)c_{\max}\big]\boldsymbol{e}_v(k) + |\boldsymbol{w}_v^{\perp}(k)|,$$

so the same one-step monotonicity conclusion holds whenever

$$|\boldsymbol{w}_v^{\perp}(k)| \le \Big(1 - \big[q + (R-1)c_{\max}\big]\Big)\boldsymbol{e}_v(k) \quad \text{for all pre-first-largest-overshooting steps.}$$

If this smallness condition on disturbances fails at some step, one-step monotonicity may be lost, but the ISS bounds proved earlier still guarantee geometric decay up to a disturbance-dependent radius.

Remark 5 (before the first negative peak, the integral state is positive) *Assume the setting of Remark.* 4*. Hence,*

$$\boldsymbol{s}_v(k) > 0 \qquad \textit{for all } k = k_0,\ldots,i_{\max}-1.$$

Proof. We argue by contradiction. Suppose there exists the first $\tau \in [k_0, i_{\max}-1]$ such that $\boldsymbol{s}_v(\tau) \le 0$. Then $\boldsymbol{s}_v(\tau-1) > 0$, and since we are on the first negative lobe, $\boldsymbol{e}_v(\tau) < 0$. Compute the one-step change of $\boldsymbol{e}_v$:

$$\begin{aligned}\boldsymbol{e}_v(\tau+1) - \boldsymbol{e}_v(\tau) &= (a(\tau)-1)\boldsymbol{e}_v(\tau) - b(\tau)\boldsymbol{s}_v(\tau)\\ &= (1-a(\tau))|\boldsymbol{e}_v(\tau)| + b(\tau)(-\boldsymbol{s}_v(\tau)) > 0,\end{aligned}$$

because $a(\tau) \le q < 1$, $\boldsymbol{e}_v(\tau) < 0$ and $\boldsymbol{s}_v(\tau) \le 0$. Hence $\boldsymbol{e}_v(\tau+1) > \boldsymbol{e}_v(\tau)$. By the same reasoning, as long as both $\boldsymbol{e}_v(k) < 0$ and $\boldsymbol{s}_v(k) \le 0$ hold, we have

$$\boldsymbol{e}_v(k+1) - \boldsymbol{e}_v(k) \ge (1-q)|\boldsymbol{e}_v(k)| + b_{\min}(-\boldsymbol{s}_v(k)) > 0,$$

where $b_{\min} := \inf_k b(k) > 0$. Meanwhile $\boldsymbol{s}_v(k+1) = \boldsymbol{s}_v(k) + \boldsymbol{e}_v(k) \le \boldsymbol{s}_v(k)$ on that interval, so $\boldsymbol{s}_v(k)$ is non-increasing; equivalenty $-\boldsymbol{s}_v(k)$ is non-decreasing. If $\boldsymbol{e}_v$ stayed negative forever, then $\sum_{k=0}^{N} \boldsymbol{e}_v(\tau+k) \to -\infty$, so $-\boldsymbol{s}_v(k)$ would grow without bound and the increments $\boldsymbol{e}_v(k+1) - \boldsymbol{e}_v(k)$ would eventually be arbitrarily large, forcing $\boldsymbol{e}_v$ to cross 0 in finite time. This contradicts the choice of $i_{\max}$ as the first negative peak. Therefore such τ cannot exist and $\boldsymbol{s}_v(k) > 0$ for all $k = k_0, \ldots, i_{\max}-1$. □

Theorem 2 (PID reduces the first-overshoot amplitude) *Let the first overshoot occur at index* k_0 *with amplitude* A_0 *(definition in Eqn.* 54*). Then, the first-overshoot amplitude under PID Steering,* A_0^{PID}*, satisfies* $A_0^{\mathrm{PID}} \le A_0^{\mathrm{PI}}$*, where* A_0^{PI} *denotes the corresponding amplitude under PI Steering.*

Proof. Under the PI law we have

$$A_0^{\mathrm{PI}} = -a(i_{max}-1)\boldsymbol{e}_v(i_{max}-1) + b(i_{max}-1)\boldsymbol{s}_v(i_{max}-1) - \boldsymbol{w}_v^{\perp}(i_{max}-1),$$

while under the PID law

$$A_0^{\text{PID}} = -a(i_{max}-1)\boldsymbol{e}_v(i_{max}-1) + b(i_{max}-1)\boldsymbol{s}_v(i_{max}-1) + c(i_{max}-1)\Delta\boldsymbol{e}_v(i_{max}-1) \tag{79}$$

$$-\boldsymbol{w}_v^{\perp}(i_{max}-1), \tag{80}$$

Due to the monotone decrease before this first largest overshooting condition stated in the previous part and the fact that $c(k) > 0$, we have $c(k_0-1)\Delta\boldsymbol{e}_v(k_0-1) < 0$. Therefore,

$$A_0^{PID} \leq b(i_{max}-1)\boldsymbol{s}_v(i_{max}-1) - a(i_{max}-1)\boldsymbol{e}_v(i_{max}-1) - \boldsymbol{w}_v^{\perp}(i_{max}-1) = A_0^{PI}. \tag{81}$$

□

Remark 4 is neccessary because monotonic decrease of $e_v(t)$ before the first peak is both a key technical property for proving Theorem 2 and a desirable feature of PID control itself. Indeed, as noted by Åström & Hägglund (1995a, p.70), poorly tuned derivative gains may produce non-monotonicity, in which case reducing only the first overshoot does not translate into improved overall behavior.

B.8 Beyond Locally Linearized Activation Dynamics

In the main text, we analysed the averaged error dynamics under P/PI/PID steering by applying a first-order Taylor expansion of the activation map around the desired "plus" trajectory $x_i^+(k)$, leading to a linear time-varying (LTV) model with an additive disturbance term. In this appendix, we make explicit how the higher-order nonlinear terms enter the dynamics and discuss their impact on the closed-loop behaviour. We also justify why the perturbations

$$\delta_i(k) = -\boldsymbol{e}_i(k) + \boldsymbol{u}(k)$$

remain controlled in norm under the proposed controllers, even under strong steering.

B.8.1 Nonlinear remainder in the averaged error dynamics

Recall that the minus activations after steering are given by

$$x_i^-(k) + \boldsymbol{u}(k) = x_i^+(k) + \delta_i(k), \qquad \delta_i(k) = -\boldsymbol{e}_i(k) + \boldsymbol{u}(k),$$

where $\boldsymbol{e}_i(k) = x_i^+(k) - x_i^-(k)$ denotes the layer-wise error for pair i, and $\boldsymbol{u}(k)$ is the steering control shared across pairs. We consider the exact Taylor expansion of the activation map $f_i^{(k)}$ around the desired activation $x_i^+(k)$:

$$f_i^{(k)}\big(x_i^+(k) + \delta_i(k)\big) = f_i^{(k)}\big(x_i^+(k)\big) + J_{f_i^{(k)}}\big(x_i^+(k)\big)\delta_i(k) + O\big(\|\delta_i(k)\|^2\big), \tag{82}$$

where $J_{f_i^{(k)}}(x)$ denotes the Jacobian of $f_i^{(k)}$ at x and the $O(\cdot)$ term collects the higher-order Taylor remainder.

Averaging over i and expressing everything in terms of the error variables leads to the following averaged error dynamics:

$$\bar{\boldsymbol{e}}(k+1) = \bar{\boldsymbol{A}}(k)\bar{\boldsymbol{e}}(k) - \bar{\boldsymbol{A}}(k)\boldsymbol{u}(k) + \boldsymbol{w}(k), \tag{83}$$

where

$$\bar{\boldsymbol{e}}(k) = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{e}_i(k), \qquad \tilde{\boldsymbol{e}}_i(k) = \boldsymbol{e}_i(k) - \bar{\boldsymbol{e}}(k), \tag{84}$$

$$\bar{\boldsymbol{A}}(k) = \frac{1}{N}\sum_{i=1}^{N}A_i(k) = \frac{1}{N}\sum_{i=1}^{N}J_{f_i^{(k)}}\big(x_i^+(k)\big), \tag{85}$$

$$\tilde{A}_i(k) = A_i(k) - \bar{\boldsymbol{A}}(k). \tag{86}$$

The disturbance term $\boldsymbol{w}(k)$ now has two contributions:

$$\boldsymbol{w}(k) = \frac{1}{N}\sum_{i=1}^{N}\tilde{A}_i(k)\tilde{\boldsymbol{e}}_i(k) + \frac{1}{N}\sum_{i=1}^{N}O\big(\|-\boldsymbol{e}_i(k) + \boldsymbol{u}(k)\|^2\big). \tag{87}$$

The first part is the "heterogeneity" term that already appears in the linearized analysis (capturing pair-to-pair variation in Jacobians and errors), while the second part arises purely from the higher-order Taylor remainder in Eqn. 82.

In other words, when we do not truncate the Taylor series after the linear term, the closed-loop dynamics still take the form

$$\bar{\boldsymbol{e}}(k+1) = \bar{\boldsymbol{A}}(k)\bar{\boldsymbol{e}}(k) - \bar{\boldsymbol{A}}(k)\boldsymbol{u}(k) + \boldsymbol{w}(k),$$

but $\boldsymbol{w}(k)$ is augmented by the additional nonlinear contribution $\frac{1}{N}\sum_i O(\|-\boldsymbol{e}_i(k) + \boldsymbol{u}(k)\|^2)$.

From the perspective of input-to-state stability, this is a benign modification: all ISS results in the main text only require that the disturbance $\boldsymbol{w}(k)$ is uniformly bounded. The higher-order terms in Eqn. 87 simply enter $\boldsymbol{w}(k)$, and as long as $\|-\boldsymbol{e}_i(k) + \boldsymbol{u}(k)\|$ remains bounded and the activation maps $f_i^{(k)}$ have bounded higher derivatives along the trajectories of interest, these terms are also uniformly bounded across layers. In that case, all ISS statements and Lyapunov-based bounds derived in the main text continue to hold; the only effect is that the (conservative) upper bound on $\|\bar{\boldsymbol{e}}(k)\|$ scales with the enlarged disturbance norm $\|w\|_\infty$.

B.8.2 Magnitude of $\delta_i(k)$ under P/PI/PID steering

A natural concern is whether $\|\delta_i(k)\| = \|-\boldsymbol{e}_i(k) + \boldsymbol{u}(k)\|$ may become large under strong steering, potentially making the higher-order terms in Eqn. 87 significant. We show here that, under our gain conditions, $\delta_i(k)$ primarily measures a residual mismatch that remains controlled in norm.

By definition,

$$\delta_i(k) = -\boldsymbol{e}_i(k) + \boldsymbol{u}(k) = -\big(x_i^+(k) - x_i^-(k)\big) + \boldsymbol{u}(k), \tag{88}$$

so that

$$x_i^-(k) + \boldsymbol{u}(k) = x_i^+(k) + \delta_i(k). \tag{89}$$

Thus, $\delta_i(k)$ measures the residual mismatch between the desired activation $x_i^+(k)$ and the steered activation $x_i^-(k) + \boldsymbol{u}(k)$ at the same layer. When the controller functions as intended, the steered trajectory $x_i^-(k) + \boldsymbol{u}(k)$ stays close to $x_i^+(k)$, and hence $\|\delta_i(k)\|$ remains small in norm. We make this more precise by examining $\delta_i(k)$ for the three controller classes.

Recall that we decompose the per-pair error into its mean and heterogeneous components,

$$\boldsymbol{e}_i(k) = \bar{\boldsymbol{e}}(k) + \tilde{\boldsymbol{e}}_i(k), \qquad \bar{\boldsymbol{e}}(k) = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{e}_i(k), \qquad \tilde{\boldsymbol{e}}_i(k) = \boldsymbol{e}_i(k) - \bar{\boldsymbol{e}}(k).$$

(i) P control. For P control we have

$$\boldsymbol{u}(k) = \boldsymbol{K}_p \bar{\boldsymbol{e}}(k),$$

and therefore

$$\begin{aligned} \delta_i(k) &= -\boldsymbol{e}_i(k) + \boldsymbol{u}(k) = -\bar{\boldsymbol{e}}(k) - \tilde{\boldsymbol{e}}_i(k) + \boldsymbol{K}_p \bar{\boldsymbol{e}}(k) && (90)\\ &= (\boldsymbol{K}_p - 1)\bar{\boldsymbol{e}}(k) - \tilde{\boldsymbol{e}}_i(k). && (91) \end{aligned}$$

Our P-gain stability condition requires

$$1 - \frac{1}{M} < \boldsymbol{K}_p < 1 + \frac{1}{M},$$

so $\boldsymbol{K}_p - 1$ is necessarily small in norm. This cancels most of the contribution of $\bar{\boldsymbol{e}}(k)$ in $\delta_i(k)$: in particular, the component of $\delta_i(k)$ along $\bar{\boldsymbol{e}}(k)$ is scaled by $(\boldsymbol{K}_p - 1)$, not by $\boldsymbol{K}_p$. As a result, $\|\delta_i(k)\|$ is dominated by the heterogeneity term $\|\tilde{\boldsymbol{e}}_i(k)\|$ rather than by the overall steering magnitude $\|\bar{\boldsymbol{e}}(k)\|$.

(ii) PI control. For PI control the steering law is

$$\boldsymbol{u}(k) = \boldsymbol{K}_p \bar{\boldsymbol{e}}(k) + \boldsymbol{K}_i \boldsymbol{s}(k), \qquad \boldsymbol{s}(k) = \sum_{j=0}^{k-1} \bar{\boldsymbol{e}}(j),$$

so

$$\begin{aligned} \delta_i(k) &= -\boldsymbol{e}_i(k) + \boldsymbol{u}(k) && (92)\\ &= -\bar{\boldsymbol{e}}(k) - \tilde{\boldsymbol{e}}_i(k) + \boldsymbol{K}_p \bar{\boldsymbol{e}}(k) + \boldsymbol{K}_i \boldsymbol{s}(k) && (93)\\ &= (\boldsymbol{K}_p - 1)\bar{\boldsymbol{e}}(k) + \boldsymbol{K}_i \boldsymbol{s}(k) - \tilde{\boldsymbol{e}}_i(k). && (94) \end{aligned}$$

All contributions to $\delta_i(k)$ remain linear in $\bar{\boldsymbol{e}}(\cdot)$ and its history through $\boldsymbol{s}(k)$. The gain conditions derived in the main text ensure closed-loop stability and imply that

$$\bar{\boldsymbol{e}}(k), \quad \boldsymbol{s}(k)$$

remain uniformly bounded. Consequently, $\|\boldsymbol{u}(k)\|$ remains in a comparable bounded range, and so does $\|\delta_i(k)\|$; the PI controller does not cause $\delta_i(k)$ to blow up, even under strong steering.

(iii) PID control. For PID control we add a derivative-like term,

$$\boldsymbol{u}(k)=\boldsymbol{K}_p\bar{\boldsymbol{e}}(k)+\boldsymbol{K}_i\boldsymbol{s}(k)+\boldsymbol{K}_d\Delta\bar{\boldsymbol{e}}(k), \qquad \Delta\bar{\boldsymbol{e}}(k)=\bar{\boldsymbol{e}}(k)-\bar{\boldsymbol{e}}(k-1),$$

which yields

$$\begin{aligned}\delta_i(k)&=-\boldsymbol{e}_i(k)+\boldsymbol{u}(k) &(95)\\ &=-\bar{\boldsymbol{e}}(k)-\tilde{\boldsymbol{e}}_i(k)+\boldsymbol{K}_p\bar{\boldsymbol{e}}(k)+\boldsymbol{K}_i\boldsymbol{s}(k)+\boldsymbol{K}_d\Delta\bar{\boldsymbol{e}}(k) &(96)\\ &=(\boldsymbol{K}_p-1)\bar{\boldsymbol{e}}(k)+\boldsymbol{K}_i\boldsymbol{s}(k)+\boldsymbol{K}_d\Delta\bar{\boldsymbol{e}}(k)-\tilde{\boldsymbol{e}}_i(k). &(97)\end{aligned}$$

Again, all terms are linear in $\bar{\boldsymbol{e}}(\cdot)$, $s(\cdot)$, and $\Delta\bar{\boldsymbol{e}}(\cdot)$. Under the PID gain conditions, the closed-loop system is stable and the signals

$$\bar{\boldsymbol{e}}(k), \quad \boldsymbol{s}(k), \quad \Delta\bar{\boldsymbol{e}}(k)$$

remain bounded in norm, implying bounded $\|\boldsymbol{u}(k)\|$ and consequently bounded $\|\delta_i(k)\|$.

In summary, even when the raw error $\|\boldsymbol{e}_i(k)\|$ may become large under strong steering, the controller is designed to cancel the dominant component of $\bar{\boldsymbol{e}}(k)$ in $\boldsymbol{u}(k)$ and to keep the integral and derivative terms bounded. The residual mismatch $\delta_i(k)$ is primarily governed by heterogeneity and bounded controller memory, rather than by the overall steering magnitude. As long as $\|\delta_i(k)\|$ stays bounded, the higher-order nonlinear contributions in Eqn. 87 remain bounded as well, so the ISS guarantees derived from the locally linearized model continue to apply, with the disturbance norm $\|\boldsymbol{w}\|_\infty$ capturing both heterogeneity and curvature effects.

C Additional Experimental Results

C.1 Qualitative Examples of Concept Steering

Fig. 8 and 9 show that varying the intervention strength $\alpha\in[0,1]$ produces a smooth and controllable progression of stylistic traits in the generated images. At low strengths ($\alpha\approx0.2$), subtle cues emerge, such as faint neon accents for the *cyberpunk* style or mild metallic shading for *steampunk*, while the overall image remains close to the original prompt. At moderate strengths ($\alpha\approx0.5$), stylistic features become more salient: cyberpunk generations exhibit vivid neon lighting and futuristic cityscapes, whereas steampunk outputs show prominent brass textures, gears, and industrial motifs. Importantly, in this regime, the central semantic content of the prompt (i.e., objects, entities, and spatial composition) is preserved with high fidelity. At high intervention strengths ($\alpha\geq0.8$), stylistic traits dominate the visual appearance, often saturating the scene with strong color palettes or dense textures, yet semantic alignment to the original prompt remains largely intact, indicating that the steering primarily affects style without eroding core content.

C.2 Jailbreaking Large Language Models

Tab. 3 reports a comprehensive comparison of attack success rate (ASR) and general benchmark performance across multiple instruction-tuned models under different defense methods. Overall, PID consistently achieves the highest ASR among defenses, while maintaining comparable performance on downstream benchmarks.

We further compare our PID steering with recent steering-vector generation methods on a better safety-aligned variant of Gemma-9B-IT, namely Gemma2-9B-Instruct-With-Deeper-Safety-Alignment (Qi et al., 2025). Deeper safety alignment here trains the model to sustain refusal behavior beyond the first few tokens: instead of only shaping the opener, it conditions on partially harmful or misleading prefixes and optimizes the model to ”recover” back to safe behavior later in the sequence. Practically, this extends safety pressure across positions so refusals remain stable under mild coercion, prefilling, or decoding variance, while seeking to preserve general-task utility. Under this strong safety-aligned regime, PID-based steering still achieves a non-trivial attack success rate and outperforms other state-of-the-art activation steering methods considered in our study.

C.3 Empirical evidence for the stability interval

Proposition 4 and Remark 3 (Appendix B.6) show that the PI gains that maximise the asymptotic convergence rate of the linearised error dynamics (the “fastest convergence” choice of K_p and K_i) also induce a large overshoot in the correlation trajectory $\langle\bar{e}(0),\bar{e}(k)\rangle$. Empirically, this manifests as poorer steering performance: aggressive integral action speeds up convergence but increases oscillation and overshoot, which is consistent with classical PI tuning principles (see, e.g., (Åström & Hägglund, 1995a)).

Theorem 2 further shows that the derivative term K_d provides damping: for fixed K_p and K_i, increasing K_d reduces the overshoot of the error trajectory. Although we do not establish a formal

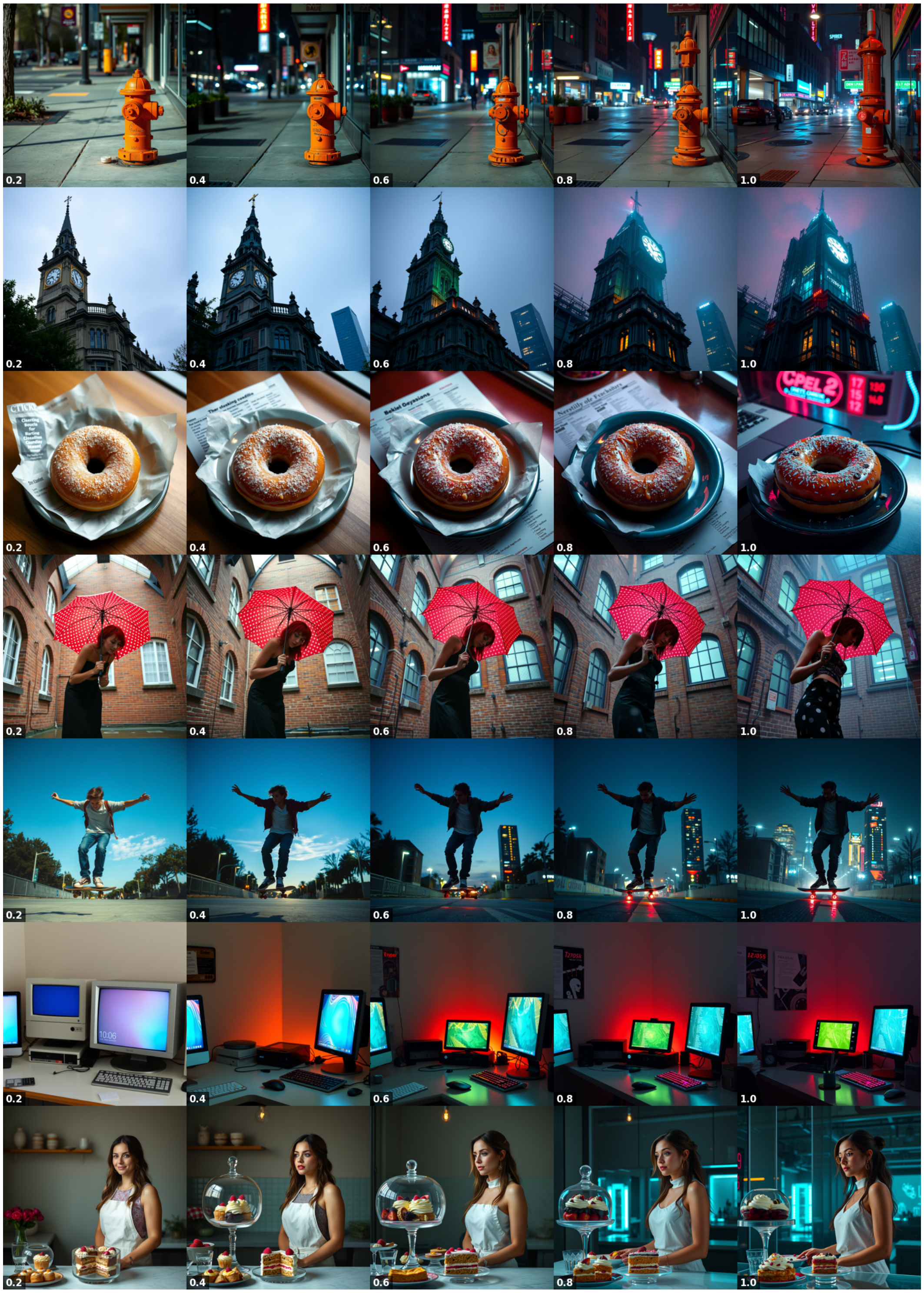


Figure 8: concept *cyberpunk*.

optimality theorem, it is natural to expect that the theoretically fastest PI gains, when combined with a suitably chosen derivative term, can outperform more conservative K_i values that lie strictly inside the stability interval (under the same $K_p = 1$). Intuitively, because transformer depth is finite and we do not know at which layer the error will effectively settle, it is desirable to drive the error down quickly while preventing excessive overshoot, with K_d acting as the compensating damping term.

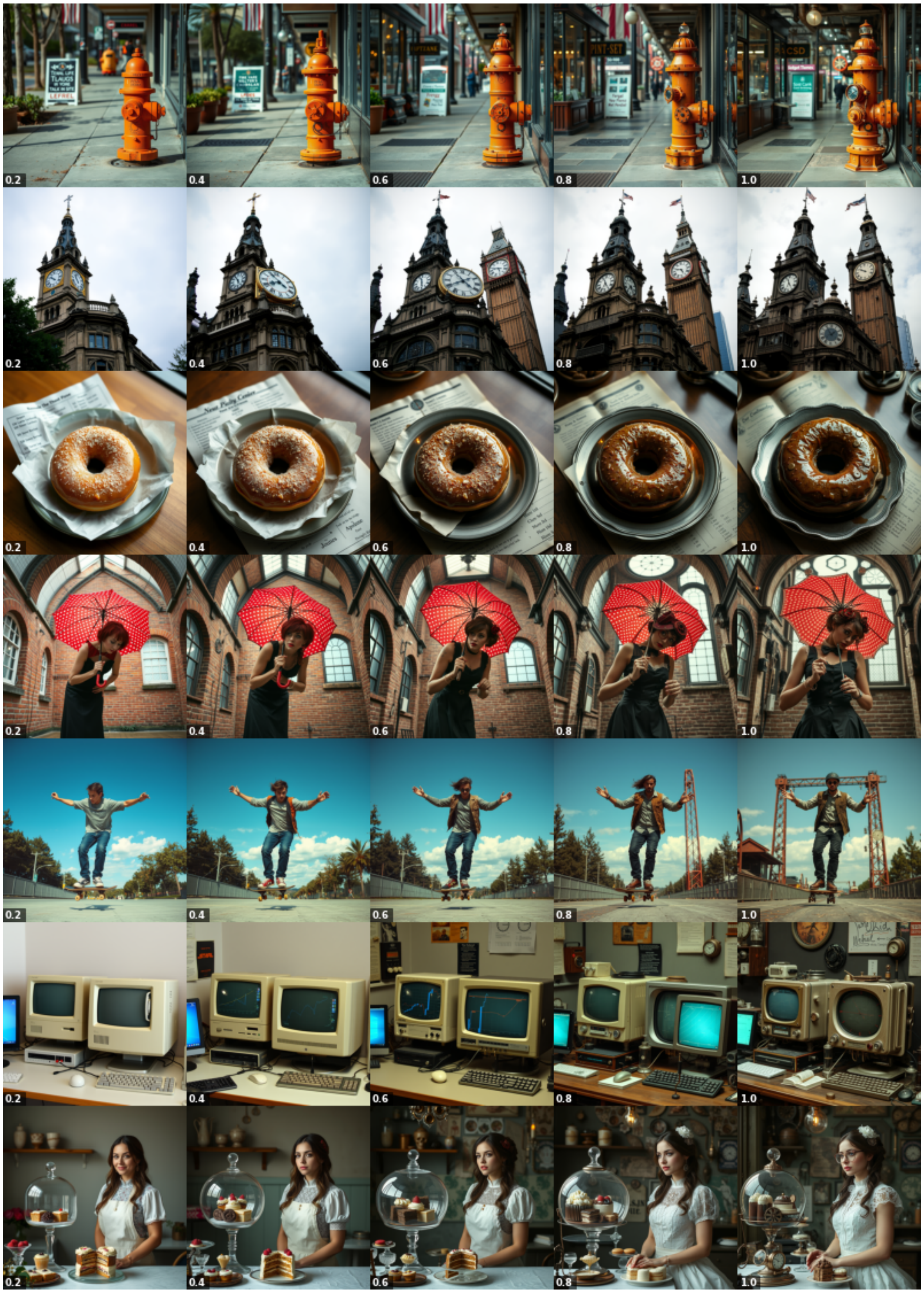


Figure 9: Concept *steampunk*

Our empirical results corroborate this interpretation. The stability intervals for K_i at $K_p = 1$ are $(-0.23, 0.23)$ for Gemma-2-9B-it and $(-0.1355, 0.1355)$ for Gemma-2-2B. In Figs. 6a and 6b, we sweep across multiple K_i values in these ranges. For Gemma-2-9B-it, the K_i that yields the fastest theoretical convergence rate is 0.056. When K_d increases from 0.0 to 0.01, the Llamaguard3 score rises from 76.61 to 78.53. This pattern suggests that with $K_d = 0$, the aggressive integral term produces noticeable overshoot that harms performance, whereas adding a small derivative term introduces sufficient damping to recover, and in some cases improve, steering performance.

Table 3: Full comparison of Original, DIM, ITI, RePE, and PID across models on ASR and general benchmarks on all tested models. Bold = best, underline = second-best within each model (ASR column).

	Method	ASR↑	tinyArc↑	tinyGSM8k strict↑	tinyMMLU↑	tinyTruthQA↑	tinyHellaSwag↑	tinyWinoGrande↑
Qwen2.5-3B Instruct	*Original*	–	62.29	17.64	68.03	56.43	73.18	70.65
	DIM	74.03	61.95	14.80	66.11	54.95	72.40	69.85
	ITI	70.19	61.28	15.57	66.62	54.75	72.71	70.12
	RePE	68.44	61.05	14.60	65.70	54.30	72.03	69.40
	PID	**76.07**	61.20	16.01	67.29	54.10	72.59	69.72
Qwen2.5-7B Instruct	*Original*	–	68.36	81.68	72.57	56.41	78.87	75.19
	DIM	96.15	65.15	80.81	71.19	55.22	78.14	74.42
	ITI	84.61	65.76	79.48	71.23	55.63	78.36	74.75
	RePE	80.32	65.00	78.90	70.60	55.00	77.73	74.15
	PID	**96.46**	66.61	80.78	71.22	55.52	78.28	74.58
Qwen2.5-14B Instruct	*Original*	–	73.96	90.12	74.60	64.50	82.70	73.77
	DIM	90.38	72.74	87.01	74.30	63.01	81.94	72.93
	ITI	33.65	73.15	89.27	74.55	64.03	82.240	73.31
	RePE	25.42	72.40	86.20	73.90	63.20	81.52	72.60
	PID	**92.65**	72.13	88.96	74.52	63.60	82.60	73.04
Llama3.2-3B Instruct	*Original*	–	55.86	59.40	63.48	50.19	75.91	58.63
	DIM	88.46	54.24	58.63	61.68	49.78	75.10	57.94
	ITI	76.92	53.67	57.77	61.85	49.95	75.22	58.16
	RePE	70.15	53.40	57.00	61.10	49.50	74.75	57.53
	PID	**89.76**	53.93	57.26	62.01	50.19	75.07	57.83
Llama3.1-8B Instruct	*Original*	–	65.33	63.21	62.02	54.39	82.51	65.56
	DIM	93.26	62.01	60.57	60.96	54.17	81.73	64.81
	ITI	79.80	64.26	61.85	61.37	54.33	82.01	65.21
	RePE	70.42	61.40	60.00	60.20	53.70	81.35	64.45
	PID	**94.85**	62.30	61.99	61.54	54.24	81.87	64.93
Gemma2-9B Instruct	*Original*	–	69.31	83.19	76.60	55.07	82.31	72.34
	DIM	77.88	68.21	80.14	72.29	51.86	81.45	71.51
	ITI	35.57	68.32	81.47	75.33	53.13	81.70	71.82
	RePE	28.64	67.50	79.20	71.10	51.10	81.20	71.15
	PID	**79.50**	67.91	79.24	74.89	52.49	81.59	71.42
Gemma2-27B Instruct	*Original*	–	73.45	86.91	76.11	61.36	83.24	75.47
	DIM	74.03	72.13	84.70	74.59	59.49	81.79	74.21
	ITI	37.36	72.84	86.38	75.51	60.85	82.83	75.11
	RePE	24.19	71.03	84.00	73.90	58.79	80.82	73.44
	PID	**79.80**	72.93	86.60	75.67	60.74	82.95	75.06
Gemma2-9B-IT-Deeper-Align	*Original*	–	69.07	82.93	76.18	54.82	82.27	72.11
	DIM	11.91	67.88	79.76	72.12	51.83	81.33	71.47
	ITI	23.12	68.19	81.41	75.22	53.05	81.58	71.79
	RePE	11.23	67.37	78.96	70.92	51.06	81.08	71.09
	PID	**34.75**	67.83	79.11	74.77	52.44	81.46	71.33