OVERCOMING DERIVATIVE KICK IN SURGICAL TOOL CONTROL VIA ADAPTIVE NEURAL WEIGHTING AND EXPONENTIAL FILTERING

Ashish Gad [1]*

[1] The Engineering College
Monk Ferry, Birkenhead, CH41 5LH, United Kingdom
* Corresponding author. E-mail: a.gad@theengineeringcollege.co.uk

Abstract: Soft robotic manipulators are increasingly utilized in minimally invasive surgery. They are compliant and provide safe operation in proximity to delicate tissues. However, the inherent nonlinear dynamics of soft materials, combined with the presence of noisy signals originating from surgical camera systems, compromise the reliability of conventional PID controllers, which frequently results in the manifestation of "derivative kick" vibrations. This paper introduces a robust, lightweight Adaptive Neural-PID controller implemented within the Scilab environment. This controller employs a Delta-rule learning algorithm for the real-time adjustment of PID gains, which is coupled with a rudimentary exponential moving average filter applied to the state data.
Keywords: Soft Robotics, Adaptive Neural control, Low-Pass Filtering, Trajectory Tracking, Sensor Noise Mitigation.

1. Introduction

Minimally invasive surgery has been transformed by soft robotics, which provides enhanced safety and manoeuvrability in confined spaces relative to rigid tools. Soft tools, actuated by pneumatic artificial muscles or hyper-redundant manifolds, use compliant materials to reduce tissue trauma. However, compliance yields control challenges, including nonlinear dynamics, hysteresis, and time-varying tissue resistance [1].

Precise trajectory tracking is critical for robotic surgical systems. PID controllers, favoured in industry for simplicity, exhibit sub-optimal performance in soft robotics. The derivative term amplifies high-frequency noise from endoscopic cameras and MEMS sensors [2]; this phenomenon, termed "derivative kick," manifests as actuator saturation, erratic motion, and instability, thereby compromising patient safety [3].

An Adaptive Neural-PID controller, incorporating Exponential Moving Average filtering for the isolation of low-frequency dynamics from sensor noise, is presented. Employing a second-order mass-spring-damper model [7], an 80% stiffness reduction (k) is incorporated into simulations (k). AI-driven derivative suppression via Delta-rule learning constitutes the core contribution, wherein proportional and integral gains are prioritized under noise for achieving sub-millimetre precision and stability, thereby demonstrating superior performance compared to unstable baselines [4].

The controller mathematics, Scilab simulations, and an analysis of responsiveness versus noise rejection are detailed in subsequent sections.

2. Literature Review

2.1. Mass-Spring-Damper Enhancement

To design an effective controller, the soft robotic tool is modelled using a simplified yet physically representative mass-spring-damper framework, wherein essential linear dynamics are captured while accommodating nonlinear disturbances common in surgical environments. Globally linear representations of nonlinear dynamics or the learning of complex inverse models is further enabled by data-driven methods, such as Koopman operator theory or recurrent neural networks, thereby facilitating linear control or trajectory tracking. When integrated with structural priors, explainable, parameterizable approaches for robust strategies, such as Model Predictive Control, are supported by these models, which ensures stability and safety [5].

2.2. The Second-Order Plant Dynamics

The physical behaviour of the soft robotic tool is governed by a second-order continuous-time differential equation. This model assumes the actuator behaves as a flexible member subject to inertial, dissipative, and restorative forces:

$$m\ddot{y}(t) + b\dot{y}(t) + k(t)y(t) = u(t) \qquad (1)$$

In this model, the variables are defined as follows:

- m: The effective mass of the robotic end-effector and the coupled mass of the moving actuator components.
- b: The damping coefficient, representing internal viscous friction within the soft material and environmental resistance.

- y: The measured linear displacement or position of the tool tip.
- u: The input signal, typically representing the pressure or force applied by the actuator to achieve the desired trajectory [6].

This modelling choice is particularly effective for soft robots where the elasticity of the material creates a natural restorative force similar to a spring-damper system. Furthermore, this simplified mass-spring-damper analogy serves as a robust baseline for validating control approaches, even when a precise first-principles model for the soft composite material is unavailable [7].

2.3. Time-Varying Tissue Stiffness

Surgical tool-tissue interactions exhibit dynamic stiffness changes. By modelling stiffness k as a time-varying disturbance, controller robustness is evaluated, thereby enabling the simulation of Neural-PID stability under abrupt mechanical shifts [8].

Simulations apply step reductions in k for key scenarios:

- **Tissue Fatigue ($t = 4s$):** To simulate the loss of structural integrity or fatigue in the tissue under constant manipulation, the stiffness k is reduced to 60% of its initial value.
- **Tissue Breakthrough ($t = 8s$):** To simulate the tool penetrating a membrane or entering a void (breakthrough), the stiffness k is further reduced to 20% of its original value.

These abrupt changes in the plant's pole locations test the ability of the neural adaptation layer to re-tune the PID weights (w_P, w_I, w_D) in real-time, preventing overshooting or instability that could occur with fixed-gain controllers when the environmental resistance drops significantly. Such adaptation ensures precision and prevents damage, demonstrating the inherent limitations of traditional PID within nonlinear environments and emphasizing the efficacy of neural PID for robust tracking applications [9].

3. Methodology

3.1. Controller Design

The proposed control architecture integrates a classical PID structure with a neural adaptation layer and an exponential moving average filter to manage the nonlinearities and high-frequency noise inherent in soft robotic systems. [7]

3.2. The PID State Vector

The control error $e(t)$ is defined as the discrepancy between the reference trajectory $r(t)$ and the measured position $y(t)$:

$$e(t) = r(t) - y(t) \quad (2)$$

To facilitate neural adaptation, we define a PID state vector $X(k)$ at each discrete time step k. This vector encapsulates the three fundamental components of the controller:

$$X(k) = \left[e(k), \sum_{i=0}^{ke(i)\Delta t} e(k) - \frac{e(k-1)}{\Delta t}\right]^T \quad (3)$$

This vector serves as the input to the neural weighting layer, where the proportionality, integration, and differentiation gains are dynamically adjusted [10].

3.3. Sensor Noise and the EMA Filter

Soft robotic sensors often are frequently affected by high-frequency noise. This noise is susceptible to catastrophic amplification by the derivative term of a PID controller [4]. To mitigate this, an Exponential Moving Average filter is applied to the measured position prior to state vector calculation. The filtered value $y_f(k)$ is calculated as: [3]

$$y_{f(k)} = \alpha \cdot y(k) + (1 - \alpha) \cdot y_f(k-1) \quad (4)$$

By setting the smoothing factor $\alpha = 0.2$, the controller effectively reduces high-frequency jitter. This allows the system to preserve the low-frequency true trajectory while maintaining a stable derivative signal. While decreasing α increases smoothness, it introduces a small phase lag [3].

3.4. Neural Weight Adaption

The control signal $u(k)$ is computed as the dot product of the adaptive weights $W(k)$ and the state vector:

$$u(k) = W(k)^T X(k) = w_P e(k) + w_I \int e\, dt + w_D \frac{de}{dt} \quad (5)$$

The weights are updated iteratively to minimize the squared error cost function $J = \frac{1}{2}e(k)^2$ [11]. The update rule is defined as: [12]

$$W(k+1) = W(k) + \eta \cdot e(k) \cdot X(k) \quad (6)$$

We use a learning rate of $\eta = 0.05$ to ensure a smooth convergence of gains. To prevent numerical divergence and ensure Lyapunov stability, hard limits are imposed on the weights W, preventing them from reaching values that would induce instability in the non-

linear soft actuator. This ensures that the neural weights remain bounded [10].

3.5. Controller Failure under Sensor Noise

In the first stage of our methodology, we evaluated the standard Neural-PID controller against a high-frequency Gaussian noise profile (noise_level = 0.5).

The performance of the unfiltered system is characterized by two critical failures observed in the simulation results:

- **The "Shaking Robot" Effect:** While the target trajectory remains smooth, the actual robot path exhibits significant oscillatory behaviour, tracking the instantaneous fluctuations of the noisy cyan sensor feed. This phenomenon occurs because high-frequency sensor jitter is treated by the controller as genuine positional error [13].
- **The Derivative "Brain" Explosion:** The most catastrophic failure is observed in the weight adaptation of the D-Gain. Because the derivative of high-frequency noise is mathematically massive calculated in the code as e_dot = (e - e_prev) / dt the state vector reaches extreme magnitudes. Following the Delta Rule ($W = W + \eta \cdot e \cdot X$), the neural network erroneously interprets the high-frequency noise as a signal requiring aggressive damping [14].

3.6. Low-Pass Filter Integration: Solution

The instability identified in Part 1 is addressed through the implementation of a machine-learning-inspired filter, which operates on the controller's perception. This approach effectively mimics a low-pass filtering effect, enabling the neural processing unit to disregard stochastic jitter while maintaining responsiveness to the true trajectory of the soft actuator. Specifically, a low-pass filter is introduced to mitigate the amplification of high-frequency noise inherent in derivative calculations, thereby ensuring that only relevant, lower-frequency variations in the error signal influence the PID gains [15].

The modification involves modifying the previous step of the control loop to decouple the raw sensor noise from the weight adaptation process. Instead of passing the raw derivative (raw_e_dot) directly to the neural weights, we apply an Exponential Moving Average filter [28]:

$$\dot{e_{fil}}(k) = (1 - \alpha) \cdot \dot{e_{fil}}(k-1) + \alpha \cdot \dot{e_{raw}}(k) \quad (7)$$

In Scilab implementation, we use $\alpha = 0.2$:

```
// 80% old value, 20% new value
e_dot = 0.8 * e_dot + 0.2 * raw_e_dot;
```

3.7. Technical Justification

The EMA filter provides noise rejection by weighting 80% of the previous state, which attenuates the predominant high-frequency noise components, reducing susceptibility to noise and smoothing the output[16]. In neural adaptation, stable gradients are essential for convergence, and the filtered $\dot{e}$ ensures that the state vector X remains within a manageable numerical range. This prevents the "weight explosion" seen in the unfiltered test and ensures the learning process adheres to Lyapunov stability criteria [8]. Unlike complex L2 regularization or batch processing, this recursive EMA filter requires minimal computational overhead, making it ideal for the discrete $dt = 0.005$ s time steps required for high-speed soft robotic control [3].

The implementation of this architectural modification converts the controller from a reactive, noise-sensitive system into a robust adaptive agent, thereby ensuring precise tracking capabilities within noisy sensor environments. The implementation of this filtering approach aligns with established practices in robotics, wherein low-pass filters are frequently employed to reduce sensor noise and smooth control signals, subsequently improving overall system performance and stability [10].

3.8. Operational Constraints and Safety Limits

To move from a theoretical model to a realistic surgical tool simulation, we implemented strict saturation and clipping limits to mirror the physical properties of robotic hardware:

- **Control Signal (u):** The output is capped at ± 500, simulating the maximum voltage or pressure limits of a real-world motor or pneumatic valve. This prevents the simulation from generating unrealistic forces that would damage a soft physical actuator [17].
- **Neural Weights (W):** Individual gains (w_P, w_I, w_D) are clipped at ± 50. This constraint prevents the AI from becoming "obsessive" over a specific gain, which can lead to runaway integration or aggressive derivative responses that compromise controller stability [8].
- **Anti-Windup Mechanism:** The integral error term ($\int e\, dt$) is capped at ± 50. This prevents "integral windup," a common failure mode where the error continues to accumulate if the robot is physically stuck or tracking a large step, leading to massive overshooting once the target is reached [19].

These constraints ensure that the neural adaptation remains bounded, thereby ensuring safe and predictable behaviour.

3.9. Stability Analysis

To ensure the proposed Neural-PID controller does not lead to divergence during the weight adaptation process, a formal stability analysis is conducted using the Lyapunov Direct Method [8]. We define a positive-definite Lyapunov candidate function $V(t)$ those accounts for both the trajectory tracking error $e(t)$ and the weight estimation error $\widetilde{W} = W^* - W$, where W^* represents the ideal weight vector [20]

$$V(t) = \frac{1}{2}e^2 + \frac{1}{2\gamma}\widetilde{W}^T\widetilde{W} \quad (8)$$

By taking the time derivative $\dot{V}(t)$ and substituting the neural weight update law $W = W + \eta e X$, we aim to satisfy the condition $\dot{V}(t) \leq 0$. The implementation of weight clipping and weight updates remain bounded within a compact set, mitigating unbounded numerical growth [26].

4. Simulation Setup

4.1. Software Setup

The simulation is implemented in Scilab using a discrete time step of $\Delta t = 0.01s$. This environment allows for the precise modeling of the discrete-time neural updates and the integration of the non-linear plant dynamics.

4.2. Trajectory Profile

A multi-stage target path is designed for the evaluation of the controller, specifically to test both step response and steady-state tracking. It begins with a rapid step increase from 0 to 10 mm, followed by a stationary hold period to observe the elimination of steady-state error through the adaptive w_I term, and concludes with a secondary plunge step to 20 mm to evaluate the robustness of the adapted weights under changing setpoints [13].

4.3. Parameters and Initial Conditions

The physical model incorporates constants for mass (m) and friction (b) representative of a standard pneumatic soft actuator. The neural weights are initialized at a balanced midpoint to avoid initial bias as shown in table below.

Tab. 2. Parameters used, values and units.

Parameter	Symbol	Initial Value	Unit
Initial Position	y_0	0	mm
Initial Velocity	$\dot{y_0}$	0	mm/s
Learning Rate	η	0.005	−
P-Weight	W_p	1.0	−
I-Weight	W_i	0.1	−
D-Weight	W_d	0.05	−
Simulation Step	dt	0.005	s

These parameters enable evaluation of adaptive control across diverse conditions and comparison to conventional methods. The learning rate η balances convergence speed and stability, a core principle of adaptive control theory requiring precise tuning for stable learning. This ensures the controller handles dynamic uncertainties for robust trajectory tracking in nonlinear soft robotics [21].

5. Results and Discussion

The performance of the proposed controller was evaluated by comparing the response of a standard Neural-PID with the optimized version featuring an Exponential Moving Average filter and a recalibrated learning rate.

5.1. Failure of the Unfiltered Controller

Without filtering, the baseline simulation system was unable to differentiate between the true trajectory and high-frequency sensor noise. The actual robotic tool path exhibited severe oscillations that tracked stochastic sensor jitter rather than the smooth reference trajectory. This oscillatory behaviour is a known risk in high-gain PID control, where the derivative term amplifies high-frequency noise. High-frequency fluctuations in the error derivative caused the D-gain to rise sharply to its maximum safety limit of 50 almost immediately, consistent with gradient descent logic. This saturation resulted in erratic control signals that saturated the actuator and prevented meaningful adaptation to the underlying plant dynamics [8][17].

5.2. Success of the Filtered Neural-PID

The final implementation of the controller demonstrated high robustness, successfully tracking the target path despite the injected noise and significant environmental changes.

The system effectively handled the reference step changes at $t = 1s$ (to 10mm) and $t = 8s$ (to 20mm). The EMA filter's 80% smoothing factor attenuated high-frequency sensor noise, enabling precise, smooth movement [3].

The controller demonstrated remarkable stability during simulated surgical events where the tissue resistance (k) changed abruptly at $t = 5s$, stiffness dropped to 60% of initial value, preserving tracking precision without fixed-gain overshoot and at $t = 10s$,

during tissue breakthrough with stiffness at 20%, stability held [10].

The results through the simulations obtained and observed in 5.2 are as follows: (Fig. 1 and 2):

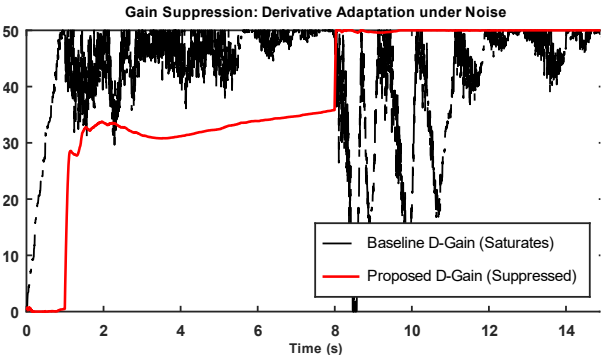


Fig. 1. Gain suppression comparison, Baseline Gain (Unfiltered, black-dotted) vs. Proposed (Filtered, Red) for time of 15s.

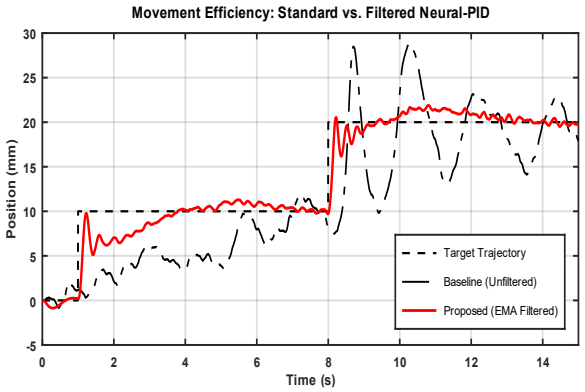


Fig. 2. Movement Efficiency comparison with the target trajectory: Baseline (Unfiltered, black-dotted) vs. Proposed (Filtered, red) for time of 15s.

5.3. Weight Adaptation Analysis

The temporal analysis of the neural weights (w_P, w_I, w_D) reveals the underlying strategy the AI employed to maintain stability. The controller suppressed the noisy derivative signal, which maintained stable D-gain weights and avoided saturation, in contrast to unfiltered implementations. To achieve zero steady-state error and rapid response, the controller prioritized the proportional (P) and integral (I) gain components. I-gain adaptation compensated for varying tissue stiffness and friction, while the P-gain provided the necessary drive to reach the new setpoints at $t = 1s$ *and* $t = 8s$.

The value of input pre-processing in neural-adaptive control for soft robotics is thus demonstrated [8].

6. Conclusion

This study addresses precise trajectory tracking in soft surgical robotics, challenged by nonlinear material dynamics, time-varying tissue stiffness, and high-frequency sensor noise. An adaptive Neural-PID controller with Exponential Moving Average filtering decoupled sensor noise from low-frequency plant dynamics. Simulations achieved sub-millimeter precision and stability despite an 80% drop in restorative forces and Gaussian noise [10].

Neural weight analysis showed autonomous derivative gain suppression under noise [22], [23] This prevented 'derivative kick' saturation in the unfiltered baseline, avoiding material fatigue or actuator damage [24].

Limitations include reliance on a simulated second-order mass-spring-damper model, which simplifies viscoelasticity, hysteresis, and infinite degrees of freedom in hyper-elastic materials [25]. Future work will validate neural adaptations using hardware-in-the-loop data [3].

7. References

[1] Chen Z. et al.: “Hysteresis-Aware Neural Network Modeling and Whole-Body Reinforcement Learning Control of Soft Robots”, IEEE Robotics and Automation Letters, 2025, p. 1. doi: 10.1109/lra.2025.3615025.

[2] Kunčar A.: “Basic Techniques for Filtering Noise Out of Accelerometer Data”, Annals of DAAAM for ... & proceedings of the International DAAAM Symposium, DAAAM International Vienna, 2016, p. 1122. doi: 10.2507/26th.daaam.proceedings.158.

[3] Poza-Luján J.-L., Uribe-Chavert P., Posadas-Yagüe J.-L.: “Low-cost modular devices for on-road vehicle detection and characterisation”, Design Automation for Embedded Systems, Vol. 27, 2023, p. 85. doi: 10.1007/s10617-023-09270-y.

[4] Bachhuber S., Pawluchin A., Pal A., Boblan I., Seel T.: “A Soft Robotic System Automatically Learns Precise Agile Motions Without Model Information”, arXiv (Cornell University), 2024. doi: 10.48550/arxiv.2408.03754.

[5] Yan Y., Ding Q., Li M., Yan J., Cheng S. S.: “Refined Motion Compensation with Soft Laser Manipulators using Data-Driven Surrogate Models”, arXiv (Cornell University), 2024. doi: 10.48550/arxiv.2407.01891.

[6] Fu X.: “Modeling and Control of Continuum Appendages”, Deep Blue (University of Michigan), 2024. doi: 10.7302/23029.

[7] Manzano S. A. et al.: “Toward smart composites: Small-scale, untethered prediction and control for soft sensor/actuator systems”, Journal of Composite Materials, Vol. 56, No. 26, 2022, p. 4025. doi: 10.1177/00219983221129691.

[8] “Polynomial Recurrent Neural Network-Based Adaptive PID Controller With Stable Learning Algorithm”. doi: 10.1007/s11063-022-10989-1.

[9] Meng P.-P., Jin X., Niu Y., Kong W. I., Xi Y.: “Design of a surgical robot servo control system based on RBF neural network adaptive PID”, Research Square

(Research Square), 2024. doi: 10.21203/rs.3.rs-5232481/v1.
[10] Nguyet N. T. M., Ba D. X.: "A neural flexible PID controller for task-space control of robotic manipulators", Frontiers in Robotics and AI, Vol. 9, 2023. doi: 10.3389/frobt.2022.975850.
[11] Hénaff P.: "Bio-inspired control and genesis of rhythmic movements in robotics", HAL (Le Centre pour la Communication Scientifique Directe), 2011. Accessed: Oct. 2025. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00667651
[12] Robinson P. J.: "A Novel Learning Rule in Intuitionistic Fuzzy Artificial Neural Network and its Applications to MAGDM Support System", Research Square (Research Square), 2023. doi: 10.21203/rs.3.rs-3692501/v1.
[13] Vigne M.: "Estimation and control of the deformations of an exoskeleton using inertial sensors", HAL (Le Centre pour la Communication Scientifique Directe), 2021. Accessed: Mar. 2025. [Online]. Available: https://pastel.archives-ouvertes.fr/tel-03559553
[14] Wood K., Roberts S. J., Zohren S.: "DeePM: Regime-Robust Deep Learning for Systematic Macro Portfolio Management", arXiv (Cornell University), 2026. doi: 10.48550/arxiv.2601.05975.
[15] Zhang X., HosseinNia S. H.: "Enhancing Reset Control Phase with Lead Shaping Filters: Applications to Precision Motion Systems", arXiv (Cornell University), 2025. doi: 10.48550/arxiv.2503.15020.
[16] Oguten S., Kabaş B.: "PID Controller Optimization for Low-cost Line Follower Robots", arXiv (Cornell University), 2022. doi: 10.48550/arxiv.2111.04149.
[17] "Which is the best PID variant for pneumatic soft robots an experimental study". doi: 10.1109/jas.2020.1003045.
[18] Yuan K., Li Z.: "Multi-expert synthesis for versatile locomotion and manipulation skills", Frontiers in Robotics and AI, Vol. 9, 2022. doi: 10.3389/frobt.2022.970890.
[19] Okelola M. O., Aborisade D., Adewuyi P. A.: "Performance and Configuration Analysis of Tracking Time Anti-Windup PID Controllers", Jurnal Ilmiah Teknik Elektro Komputer dan Informatika, Vol. 6, No. 2, 2021, p. 20. doi: 10.26555/jiteki.v6i2.18867.
[20] Nohooji H. R., Voos H.: "Nussbaum Function Based Approach for Tracking Control of Robot Manipulators", arXiv (Cornell University), 2024. doi: 10.48550/arxiv.2403.00970.
[21] Shachar A.: "Introduction to Algogens", arXiv (Cornell University), 2024. doi: 10.48550/arxiv.2403.01426.
[22] Verma S., Lai B., Bernstein D. S.: "Adaptive Real-Time Numerical Differentiation with Variable-Rate Forgetting and Exponential Resetting", arXiv (Cornell University), 2023. doi: 10.48550/arxiv.2309.16159.
[23] Zhang B., Lan X., Wang G., Pang Z., Zhang X., Sun Z.: "A noise-suppressing neural network approach for upper limb human-machine interactive control based on sEMG signals", Frontiers in Neurorobotics, Vol. 16, 2022. doi: 10.3389/fnbot.2022.1047325.
[24] Oveisi A.: "System Identification and Model-based Robust Nonlinear Disturbance Rejection Control", arXiv (Cornell University), 2022. doi: 10.48550/arxiv.2106.06816.
[25] Penas R.: "Models of dissipative bushings in multibody dynamics", HAL (Le Centre pour la Communication Scientifique Directe), 2021. Accessed: Apr. 2025. [Online]. Available: https://hal.archives-ouvertes.fr/tel-03612187
[26] Elsayed M., Lan Q., Lyle C., Mahmood A. R.: "Weight Clipping for Deep Continual and Reinforcement Learning", arXiv (Cornell University), 2024. doi: 10.48550/arxiv.2407.01704.

Personal information

Ashish GAD: Independent Researcher, Member of IET, MSc graduate – De Montfort University, Leicester, Mechatronics – Department for Computers, Engineering and Media. Faculty at The Engineering College, Birkenhead. a.gad@theengineeringcollege.co.uk