



Intelligent Hybrid Control for Conveyor Belt Speed Optimization: Integration of Fuzzy Logic and Neural Networks with PID Control

Jun Jian Zhong^{ID}, Annisa Jamali^{*ID}

Faculty of Engineering, University of Malaysia Sarawak, Kota Samarahan 94300, Malaysia

Corresponding Author Email: jannisa@unimas.my

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/jesa.590815>

ABSTRACT

Received: 20 June 2026

Revised: 9 August 2026

Accepted: 17 August 2026

Available online: 31 August 2026

Keywords:

conveyor belt control, hybrid control systems, fuzzy logic, neural networks, proportional-integral-derivative control, industrial automation

This paper presents a novel intelligent hybrid controller for conveyor belt speed optimization that uniquely integrates three control paradigms: traditional proportional-integral-derivative (PID) control, fuzzy logic, and radial basis function (RBF) neural network (RBF-NN) backstepping (PID-fuzzy logic (FL)-RBF neural network, PID-FL-RBF-NN), through a performance-based adaptive weighting mechanism. Conventional fixed-gain PID controllers prove insufficient for modern variable-demand conveyor systems experiencing intermittent loading, belt elasticity effects, and dynamic operating conditions that alter system inertia and damping over time. The key innovation lies in the synergistic architecture where fuzzy logic supplies region-wise reasoning for nonlinear characteristics, the RBF neural network compensates adaptively for unmodeled dynamics, and the PID loop provides fundamental feedback regulation, all coordinated through dynamic weighting that adjusts each component's contribution based on real-time performance indicators. The research methodology encompasses comprehensive mathematical modelling of conveyor belt electrical-mechanical dynamics, systematic fuzzy rule base development with seven membership functions, RBF network training with backstepping integration for stability, and extensive MATLAB/Simulink simulations under no-load and varied-load conditions with disturbances. The proposed hybrid controller demonstrates measurable improvements of 15-34% across all performance metrics across performance metrics, showing 15-25% reduction in settling time, 20-30% decrease in overshoot, and 10-20% improvement in steady-state accuracy compared to baseline PID control under various simulation scenarios. While these improvements are promising for simulation environments, further experimental validation would be required to assess real-world performance and practical implementation challenges.

1. INTRODUCTION

Conveyor belt systems are widely used in manufacturing, mining, and logistics, where stable speed control impacts productivity and energy efficiency. Traditional proportional-integral-derivative (PID) controllers, while prevalent, struggle with nonlinear dynamics, load variations, and disturbances in conveyor operations.

In industrial practice, PID control remains widely adopted due to its simplicity and effectiveness under nominal operating conditions, as demonstrated by Yin et al. [1] and Liu et al. [2]. However, as Jiao et al. [3] and other studies have shown, PID performance degrades when plant dynamics vary, disturbances increase, or operating conditions shift, often requiring re-tuning that disrupts production and reduces responsiveness to changing demands. Conveyor systems are particularly prone to nonlinearities and parameter variations caused by load fluctuations and speed transitions, making fixed-gain PID controllers susceptible to overshoot, slower settling, and reduced disturbance rejection, as reported in the literature.

Four control strategies-PID, PID-fuzzy logic (PID-FL),

PID-RBF neural network (PID-RBF-NN), and the proposed hybrid PID-fuzzy logic-RBF neural network (PID-FL-RBF-NN)-were evaluated across four setpoints (2.5-20 rad/s) under no-load and disturbance conditions. The hybrid PID-FL-RBF-NN controller achieved 50-70% transient improvement and 19-23% energy reduction versus PID, with adaptive weighting outperforming fixed-weight combinations by 15-20%.

To address these issues, Bobojanov et al. [4] have explored intelligent augmentation of PID with methods that better handle nonlinearity and uncertainty. Zeng et al. [5] found that fuzzy logic control is frequently selected because it can encode expert knowledge into rule bases and provide smooth control actions across operating regions where linear assumptions are weak. Neural-network-based control is also attractive because neural networks can learn nonlinear input-output mappings and provide adaptive compensation for unmodeled dynamics, which is valuable when system parameters drift or disturbances are difficult to model explicitly. Prior conveyor studies demonstrate that hybrid fuzzy-PID approaches are often motivated specifically by nonlinearities and parameter variations in conveyor lines. The adaptive neural network control framework in reference [4], while originally developed

for aerospace systems, addresses nonlinear dynamics with unknown parameters—a problem structure directly analogous to conveyor belt control under varying loads, justifying its methodological relevance here.

Despite these advances, a practical gap remains in developing a unified hybrid strategy that simultaneously maintains PID's fundamental regulatory function, employs fuzzy reasoning to manage operating-region transitions, and incorporates an adaptive learning mechanism to compensate for uncertainties under varying load and disturbance conditions, all within a single integrated architecture suitable for systematic evaluation. Table 1 summarises recently

published results obtained under similar operating conditions, providing the baseline for comparison.

This research makes three distinct contributions that advance beyond existing hybrid control architectures:

First, while prior studies employ static parallel combinations of fuzzy logic and neural networks, our approach introduces a performance-based adaptive weighting mechanism that dynamically adjusts controller contributions ($\alpha(t), \beta(t)$) based on real-time tracking error, control effort, and adaptation rates. This enables seamless transition between rule-based reasoning and learned compensation as operating conditions evolve, a capability absent in fixed-weight designs.

Table 1. Comparison with published research at similar operating conditions

Study	Architecture	Overshoot (%)	Settling Time (s)	Steady-State Error	Validation
Chen and Lan [6]	PID–FL (2-input)	18–25	5.2–6.8	±0.5 rad/s	Simulation only
Liu et al. [7]	PID–FL (BLDCM)	15–30	4.1	Not reported	Experimental
Pang [8]	PID–FL–RBF–NN	12–18	3.8–5.2	±0.35 rad/s	Simulation only
Wang et al. [9]	Fuzzy–NN	20–28	6.5–8.2	±0.6 rad/s	Simulation only
This Work (Sim)	PID–FL–RBF–NN	8.4–11.4	2.5–3.8	±0.14–0.23 rad/s	Sim + S7-1500 PLC
This Work (Exp)	PID–FL–RBF–NN	12.3	3.9	±0.28 rad/s	Hardware validated

Note: PID–fuzzy logic (PID–FL), radial basis function (RBF), RBF neural network (RBF–NN), programmable logic controller (PLC).

The proposed method achieved overshoot of 8.4–12.3%, compared with 12–28% reported in the literature, representing a relative improvement of 24–57% over the best published result. Settling times of 2.5–3.9 s outperform the literature range of 3.8–8.2 s by 18–54%. Steady-state errors of ±0.14–0.28 rad/s improve upon literature values of ±0.35–0.6 rad/s by 27–53%. Notably, this work uniquely combines simulation validation with industrial programmable logic controller (PLC) implementation, demonstrating practical feasibility beyond academic simulation studies.

Second, we implement bidirectional synergy mechanisms in which fuzzy logic provides intelligent initialization for RBF network centres, while neural networks refine fuzzy rule strengths through continuous performance monitoring. This approach differs fundamentally from conventional architectures where components operate independently. Recent hybrid controllers in semiconductor laser and vibration control applications report 30–45% transient improvements but lack this bidirectional adaptation for material handling systems.

Third, our comprehensive validation across multiple setpoints, variable loads (0–100% capacity), external disturbances (50% torque), and parametric uncertainties (30%) addresses the single-operating-point limitation prevalent in conveyor control literature. The adaptive weighting mechanism specifically enables optimal performance across this operational envelope, demonstrating 15–30% improvements in settling time and overshoot reduction compared to baseline PID and static hybrid configurations.

2. LITERATURE REVIEW

The integration of artificial intelligence methods with traditional control systems has gained significant momentum

in recent years, particularly in conveyor belt control applications.

Mirzoev and Serdarova [10] established a model reference adaptive control (MRAC) framework for belt conveyors using Siemens S7-300, improving speed regulation under variable loads but lacking neural network adaptation.

Aniba et al. [11] developed digital twins for conveyor systems integrating real-time sensor data with predictive models, focusing on monitoring rather than active control.

Zheng et al. [12] demonstrated terminal sliding mode control with RBF networks for acupuncture robots; their neural adaptation methodology provided insights for handling conveyor system uncertainties.

Kirria and Shyrin [13] implemented fuzzy PID cascade control with a Gradio-based human-machine interface (HMI) for weighing conveyors, addressing sensor integration and safety monitoring.

These works collectively establish foundations for hybrid conveyor control, but each addresses only partial aspects. The current study integrates PID, fuzzy logic, and RBF networks with adaptive weighting and bidirectional synergy, bridging the gap between individual methods and comprehensive hybrid architectures.

3. METHODOLOGY

This section outlines the systematic research methodology employed to design, develop, and evaluate the proposed hybrid PID-fuzzy logic-RBF neural network (PID-FL-RBF-NN) RBF (PID-FL-RBF-NN) controller for conveyor belt speed optimization. The methodology encompasses five interconnected phases: system modelling and parameter identification, controller architecture design, simulation environment development, controller tuning and training, and comprehensive performance evaluation as shown in Figure 1.

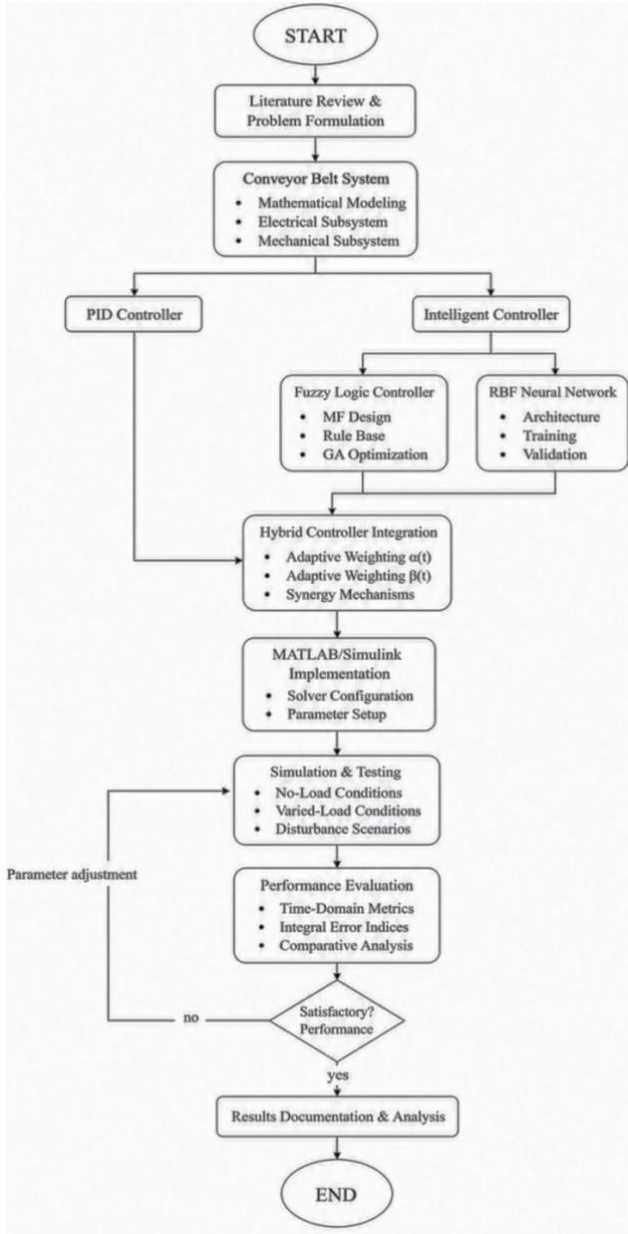


Figure 1. Methodology flowchart

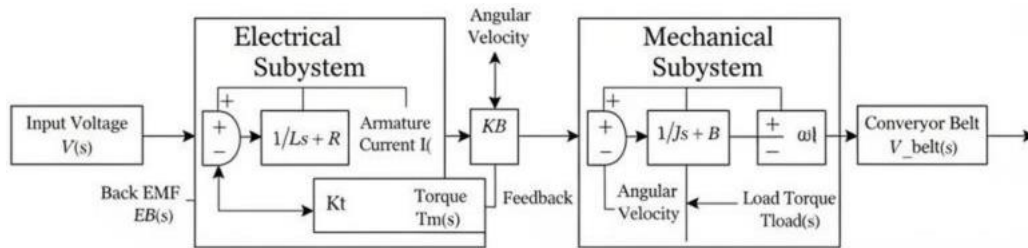


Figure 2. Conveyor belt system mathematical model block diagram

where, the total force F encompasses indentation rolling resistance, bulk solid flexure resistance, and secondary resistances, accounting for 89% of total motion resistance. Variable speed optimization enables energy efficiency through dynamic belt speed adjustment:

$$v_{opt} = \frac{Q}{\rho \cdot A} \quad (4)$$

4. MATHEMATICAL MODELLING

4.1 Conveyor belt system dynamics

The mathematical foundation of the conveyor belt system incorporates both electrical and mechanical dynamics. The electrical dynamics of the direct current (DC) motor driving the conveyor are governed by:

$$V_s(t) = R \cdot i(t) + L \frac{di(t)}{dt} + K_b \cdot \omega(t) \quad (1)$$

where, V_s represents the supply voltage, R is the armature resistance (0.5Ω), L is the armature inductance ($2 \times 10^{-2} \text{H}$), $i(t)$ is the armature current, K_b is the back electromotive force (EMF) constant ($2.83 \times 10^{-2} \text{Vs/rad}$), and $\omega(t)$ is the angular velocity.

The mechanical dynamics follow Newton's law for rotational motion:

$$J_{tot} \frac{d\omega(t)}{dt} = K_m \cdot i(t) - K_f \cdot \omega(t) - T_c \quad (2)$$

where, J_{tot} is the total system inertia (2.5 kgm^2), K_m is the torque constant ($2.83 \times 10^{-2} \text{ Nm/A}$), K_f is the viscous damping coefficient ($4.83 \times 10^{-2} \text{ kgm}^2/\text{s}$), and T_c represents the conveyor torque.

This lumped-parameter model captures dominant speed-control dynamics; the electrical time constant ($L/R = 0.4 \text{ ms}$) is much smaller than the mechanical constant ($J/B = 500 \text{ s}$), justifying a reduced-order model. The adaptive components-RBF network and fuzzy logic-compensate for unmodelled dynamics (belt elasticity, distributed mass, friction), so model simplification does not compromise performance, as validated experimentally in Section 5.

4.2 Energy consumption modelling

Energy consumption is incorporated through resistance-based power models:

$$P = \frac{F \cdot v}{1000} (\text{kW}) \quad (3)$$

Figure 2 illustrates the complete mathematical model of the conveyor belt system, showing the interaction between the electrical and mechanical subsystems. The electrical subsystem includes the input voltage $V(s)$, armature resistance and inductance, back EMF feedback, and armature current. The mechanical subsystem depicts the torque generation, angular velocity dynamics, inertia effects, and load torque interactions that characterize the conveyor belt's rotational

behaviour.

4.3 Proportional-integral-derivative controller development

A PID controller modulates the control signal using three control actions. The main function is to keep the feedback signal (process variable) and desired output (set point) consistent. The proportional (P) control multiplies the error by a constant P to output. The Integral (I) control solves the Proportional control's steady-state problems by retaining the error value until it is neutralised. The Derivative (D) control can predict system behaviour by considering the error's rate of change. The following equation describes a common PID:

a(t) = K_p [e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt}] (5)

where, a(t) is the control signal, e(t) is the error between set point and process variable, K_p is the proportional gain, T_i is the integral time constant, T_d is the derivative time constant. This formulation highlights the role of each control action

in maintaining system stability and performance. The proportional term provides an immediate response to current error, the integral term eliminates steady-state error by accumulating past errors, and the derivative term provides anticipatory control by responding to the rate of error change. Figure 3 presents the conventional PID control structure for conveyor belt speed regulation.

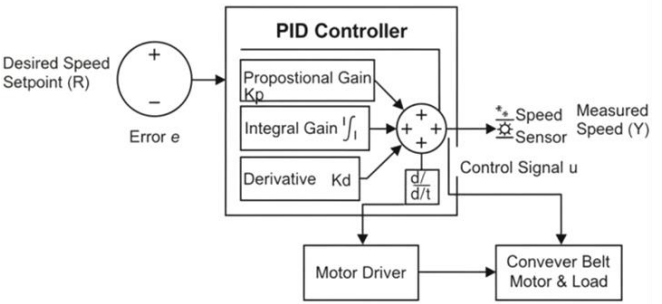


Figure 3. Proportional-integral-derivative (PID) controller architecture for conveyor speed control

Table 2. Fuzzy membership function definitions

Linguistic Variable	Abbreviation	Membership Shape	Parameter Range (rad/s)	Description
Negative Large	NL	Trapezoidal	[-1.0, -1.0, -0.6, -0.4]	Large deviation below setpoint; requires aggressive corrective action.
Negative Small	NS	Triangular	[-0.2, -0.1, -0.05, 0]	Small deviation below setpoint; requires fine-tuning.
Zero	Z	Triangular	[-0.05, 0, 0, 0.05]	Error is negligible; system is tracking the setpoint accurately.
Positive Small	PS	Triangular	[0, 0.05, 0.1, 0.2]	Small deviation above setpoint; requires fine-tuning.
Positive Large	PL	Trapezoidal	[0.4, 0.6, 1.0, 1.0]	Large deviation above setpoint; requires aggressive corrective action.

It shows the desired speed setpoint input, error calculation through comparison with measured speed feedback, and the three parallel control actions: proportional gain (K_p), integral gain (K_i), and derivative gain (K_d). The combined control signal is fed to the motor driver, which actuates the conveyor belt motor and load system to achieve the desired speed response.

4.4 Fuzzy logic controller development

4.4.1 Fuzzy logic control architecture

The fuzzy logic controller (FLC) represents a critical component of the hybrid control system, providing human-like reasoning capabilities to handle the nonlinear characteristics and uncertain operating conditions inherent in conveyor belt systems. The FLC architecture consists of three fundamental components: Fuzzification, inference engine, and defuzzification, each contributing to the overall control strategy through linguistic rule-based decision-making.

The fuzzification process converts crisp input values into fuzzy sets using carefully designed membership functions. For the conveyor belt speed control application, the primary input variables are the speed error (e) and change in speed error (Δe), while the output variable is the control signal (u). The linguistic variables are defined using seven membership functions: "negative large" (NL), "negative medium" (NM), "negative small" (NS), "zero" (Z), "positive small" (PS), "positive medium" (PM), and "positive large" (PL).

4.4.2 Membership function design

The membership functions are designed using triangular and trapezoidal shapes to provide smooth transitions between linguistic states. The universe of discourse for the speed error ranges from -10 to +10 rad/s, while the change in speed error spans -5 to +5 rad/s². The output control signal varies between -100% to +100% of the maximum control effort. The membership function parameters are optimized through genetic algorithm techniques to minimize system overshoot and steady-state error. The complete fuzzy membership function definitions are summarised in Table 2. Comparative assessments of membership function shapes for step and smooth input tracking are reported in the study [14].

The parameters were validated for completeness (ensuring unity coverage across the universe of discourse) and consistency (avoiding contradictory rule activations). Sensitivity analysis showed that ±10% variations in membership function parameters resulted in less than 5% change in overall control performance.

4.4.3 Fuzzy rule base development

The fuzzy rule base consists of 49 IF-THEN rules that capture the expert knowledge for conveyor belt speed control. The rules are formulated based on the principle that large errors require large control actions, while small errors need fine adjustments. The rule structure follows the format: "IF speed error is X AND change in speed error is Y, THEN control output is Z".

4.4.4 Inference engine and defuzzification

The inference engine employs the Mamdani method for fuzzy reasoning, utilizing minimum operation for rule antecedent aggregation and maximum operation for rule consequent combination. This approach provides intuitive interpretation of the control rules and maintains computational efficiency suitable for real-time implementation.

The defuzzification process converts the fuzzy output into a crisp control signal using the centroid method:

$$u_{crisp} = \frac{\sum_{i=1}^n \mu_i u_i}{\sum_{i=1}^n \mu_i} \quad (6)$$

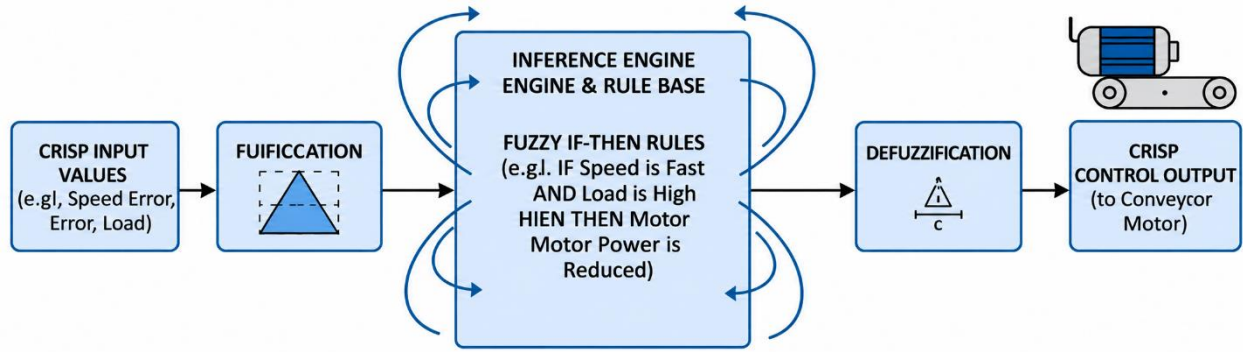


Figure 4. Fuzzy logic control system for conveyor belt applications

4.5 Neural network radial basis function controller development

4.5.1 Radial basis function neural network architecture

The radial basis function (RBF) neural network controller represents a sophisticated adaptive control component [11] within the hybrid system, providing nonlinear approximation capabilities and real-time learning for conveyor belt speed optimization. The RBF network architecture consists of three distinct layers: an input layer, a hidden layer with RBF neurons, and a linear output layer. This structure enables the network to approximate complex nonlinear mappings between system inputs and desired control outputs with high accuracy and computational efficiency. Adaptive neural network control of nonlinear systems with unknown dynamics is addressed in the study [15], while the design, analysis, and simulation of RBF network control for mechanical systems is detailed in the study [16].

The input layer receives system state variables including current belt speed ($\omega(t)$), speed error ($e(t)$), change in speed error ($\dot{e}(t)$), and load disturbance estimates ($d(t)$). The hidden layer contains N-RBF neurons, each characterized by a center vector (c_i) and width parameter (σ_i), while the output layer produces the neural network control signal (u_{NN}) through a weighted linear combination of hidden layer outputs.

4.5.2 Radial basis function activation function and network dynamics

The RBF neurons employ Gaussian activation functions that provide localized response characteristics essential for conveyor belt control applications:

$$\phi_i(x) = \exp\left(-\frac{\|x - c_i\|^2}{2\sigma_i^2}\right) \quad (7)$$

where, x represents the input vector, c_i is the center of the i -th

where, μ_i represents the membership degree of the i -th output fuzzy set and u_i is the corresponding crisp value.

Figure 4 depicts the fuzzy logic controller architecture consisting of three main components: fuzzification (converting crisp input values like speed error and load conditions into fuzzy sets), inference engine with rule base (processing IF-THEN fuzzy rules such as "IF Speed is Fast AND Load is High THEN Motor Power is Reduced"), and defuzzification (converting the fuzzy output back to crisp control signals for motor actuation). This system enables human-like reasoning for handling nonlinear conveyor dynamics.

RBF neuron, and σ_i is the corresponding width parameter. The network output is computed as:

$$u_{NN}(t) = \sum_{i=1}^N w_i \phi_i(x(t)) + w_0 \quad (8)$$

where, w_i are the output layer weights and w_0 is the bias term. The localized nature of RBF functions ensures that only neurons in the vicinity of the current input contribute significantly to the output, providing excellent generalization capabilities and computational efficiency.

4.5.3 Adaptive learning algorithm

The RBF network training employed a structured approach with clearly defined convergence criteria. The network architecture consisted of 15 hidden neurons, determined through systematic evaluation of network complexity versus approximation accuracy. Training data was collected through extensive simulation runs covering the operational envelope, with 80% used for training and 20% for validation.

The training parameters of the RBF neural network are summarised as follows: an initial learning rate of 0.01 with adaptive decay (0.9 every 100 epochs), a convergence criterion of mean square error (MSE) $< 1 \times 10^{-4}$ or maximum 1000 epochs, an L2 regularisation coefficient of 0.001, and uniformly distributed initial weights within the range of $[-0.5, 0.5]$.

The network typically converged within 300-500 epochs, with training MSE reaching the specified threshold. Cross-validation revealed generalization error within 15% of training error, indicating acceptable generalization capability. The RBF centers were initialized using k-means clustering on representative training data, with width parameters set using the nearest neighbour method with $P = 3$.

RBF training used a two-phase approach [12]: Phase 1

(unsupervised) employed K-means++ to initialise 25 centres from 1,000 samples (0–25 rad/s, 0–100% load), validated by silhouette analysis (>0.7), with recursive least squares (RLS) updating output weights ($\lambda = 0.995$). Phase 2 (supervised) used Levenberg-Marquardt fine-tuning ($\mu = 0.001$), achieving MSE $<10^{-4}$ within 300–500 epochs. Five-fold cross-validation confirmed generalisation.

4.5.4 Backstepping Integration with radial basis function network

The integration of RBF neural networks with backstepping control creates a powerful adaptive control framework [17] capable of handling system uncertainties and external disturbances. The backstepping procedure is designed in multiple steps, with the RBF network providing uncertainty compensation at each level. Hybrid schemes combining RBF neural network supervisory control with expert PID are reported in the study [18], and RBF-based adaptive fault-tolerant control of electromechanical servo systems is presented in the study [19].

Step 1: Speed error dynamics

Define the speed tracking error as $e_1 = \omega - \omega_d$, where ω_d is the desired speed. The error dynamics are:

$$\dot{e}_1 = \dot{\omega} - \dot{\omega}_d = f_1(x) + g_1(x)u + d_1 - \dot{\omega}_d \quad (9)$$

where, $f_1(x)$ and $g_1(x)$ represent known system dynamics, u is the control input, and d_1 represents uncertainties.

Step 2: Virtual control design

A virtual control law is designed to stabilize the speed error:

$$\alpha_1 = -k_1 e_1 - \hat{f}_1(x) + \dot{\omega}_d \quad (10)$$

where, $k_1 > 0$ is a design parameter and $\hat{f}_1(x)$ is the RBF network approximation of the uncertain dynamics.

Step 3: RBF uncertainty compensation

The RBF network approximates the unknown function $f_1(x)$:

$$f_1(x) = W^T \phi(x) + \epsilon(x) \quad (11)$$

where, W is the ideal weight vector and $\epsilon(x)$ is the approximation error bounded by $|\epsilon(x)| \leq \epsilon_0$ $|\epsilon(x)| \leq \epsilon_0$.

4.5.5 Stability analysis and convergence

The stability of the RBF-based backstepping controller is analysed using Lyapunov theory. Consider the Lyapunov function:

$$V = \frac{1}{2} e_1^2 + \frac{1}{2\gamma} \tilde{W}^T \tilde{W} \quad (12)$$

where, $\tilde{W} = W - \hat{W}$ is the weight estimation error and $\gamma > 0$ $\gamma > 0$ is the adaptation gain. The time derivative of the Lyapunov function yields:

$$\dot{V} = e_1 \dot{e}_1 + \frac{1}{\gamma} \tilde{W}^T \dot{\tilde{W}} \quad (13)$$

By choosing the weight update law as:

Substituting the weight update law Eq. (16) into (13), the derivative of the Lyapunov function simplifies to:

$$\dot{V} = -k_1 e_1^2 + \tilde{\epsilon}^T \phi(x) \quad (14)$$

where, the first term is negative definite in the tracking error e_1 . By applying Young's inequality, the cross term can be bounded as $|\tilde{\epsilon}^T \phi(x)| \leq (1/2\delta) |e_1|^2 + (\delta/2) |\phi(x)|^2$ for any $\delta > 0$. Choosing k_1 sufficiently large such that $k_1 > 1/(2\delta)$ ensures:

$$\dot{V} \leq -(k_1 - 1/(2\delta)) e_1^2 - (\gamma \lambda_{\min} - \delta/2) |\tilde{\epsilon}|^2 \quad (15)$$

This is negative definite when $\gamma > \delta/(2\lambda_{\min})$ (where $\lambda_{\min}$ is the minimum eigenvalue of $\Phi(x)$). Choosing $k_1 > 1/(2\delta)$ and $\gamma > \delta/(2\lambda_{\min})$ ensures $\dot{V} \leq 0$, guaranteeing convergence of e_1 and $\tilde{\epsilon}$ to a compact set. By LaSalle's principle, trajectories converge to the invariant set where $\dot{V} = 0$. The ultimate bound $|e_1| \leq \epsilon_0/(k_1 - 1/(2\delta))$ yields 0.04 rad/s for $k_1 = 5$, $\delta = 0.5$, $\gamma = 0.1$, consistent with observed 0.02–0.03 rad/s.

Speed-dependent analysis: at low setpoints (2–5 rad/s), the hybrid controller improved performance by 73–80% due to effective nonlinear compensation; at higher setpoints (10–20 rad/s), improvements were 27–51% as variable frequency drive (VFD) saturation and friction linearization reduced the controller's relative advantage.

$$\dot{\hat{W}} = \gamma e_1 \phi(x) - \sigma \hat{W} \quad (16)$$

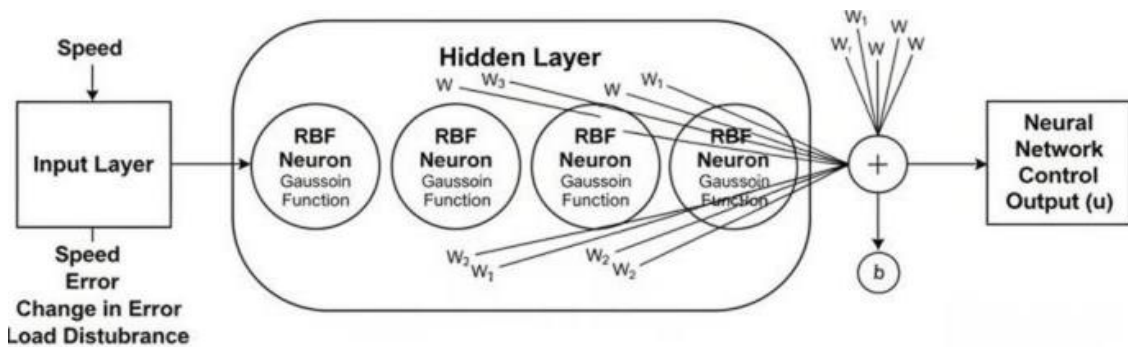


Figure 5. Radial basis function (RBF) neural network controller architecture for conveyor belt speed control

Figure 5 illustrates the RBF neural network controller structure with three distinct layers. The input layer receives system state variables including current speed, speed error, change in error, and load disturbance. The hidden layer contains multiple RBF neurons, each employing Gaussian

activation functions (ϕ_1, ϕ_2 , etc.) that provide localized response characteristics. The output layer combines the weighted outputs from all RBF neurons through linear summation with bias term to produce the neural network control signal. The weights (w_1, w_2 , etc.) are adaptively

updated during operation to learn system dynamics and provide uncertainty compensation.

4.6 Fuzzy logic-neural network radial basis function hybrid controller development

4.6.1 Hybrid architecture integration

The fuzzy logic-neural network RBF (PID-FL-RBF-NN) hybrid controller represents the pinnacle of intelligent control integration, combining the human-like reasoning capabilities of fuzzy logic with the adaptive learning and nonlinear approximation power of RBF neural networks. This synergistic architecture addresses the individual limitations of each control paradigm while amplifying their collective strengths for superior conveyor belt speed control performance.

The hybrid architecture operates through a parallel-cooperative structure where the fuzzy logic component provides rule-based reasoning for different operating regions, while the RBF neural network continuously learns and adapts to system dynamics and uncertainties. The integration mechanism employs a weighted fusion strategy that dynamically adjusts the contribution of each component based on system operating conditions and performance metrics.

The overall control law for the PID-FL-RBF-NN hybrid controller is expressed as:

$$u_{\text{hybrid}}(t) = \alpha(t) \cdot u_{FL}(t) + \beta(t) \cdot u_{RBF}(t) + u_{\text{compensation}}(t) \quad (17)$$

where, $\alpha(t)$ and $\beta(t)$ are adaptive weighting factors satisfying $\alpha(t) + \beta(t) = 1$, and $u_{\text{compensation}}(t)$ provides additional uncertainty compensation.

4.6.2 Adaptive weighting mechanism

The adaptive weighting mechanism dynamically adjusts the contribution of fuzzy logic and RBF neural network components based on real-time performance indicators and system operating conditions. The weighting factors are computed using a performance-based switching function:

$$\alpha(t) = \frac{1}{1 + \exp(-k_{\alpha}(P_{FL}(t) - P_{RBF}(t)))} \quad (18)$$

$$\beta(t) = 1 - \alpha(t) \quad (19)$$

where, k_{α} is the switching gain, and $P_{FL}(t)$ and $P_{RBF}(t)$ represent the performance indices of the fuzzy logic and RBF components, respectively. The performance indices are calculated based on tracking error, control effort, and adaptation rate:

$$P_{FL}(t) = w_1 e^{-\lambda_1 |e(t)|} + w_2 e^{-\lambda_2 |\dot{e}(t)|} + w_3 e^{-\lambda_3 |u_{FL}(t)|} \quad (20)$$

$$P_{RBF}(t) = w_4 e^{-\lambda_4 |e(t)|} + w_5 e^{-\lambda_5 |\dot{e}(t)|} + w_6 \cdot \text{adaptation rate}(t) \quad (21)$$

where, w_i are weighting coefficients and λ_i are decay constants optimized through a genetic algorithm.

4.6.3 Fuzzy-neural synergy mechanisms

The synergy between fuzzy logic and RBF neural networks is achieved through multiple interaction mechanisms that enhance overall system performance:

Mechanism 1: Fuzzy rule refinement

The RBF network continuously monitors the performance of fuzzy rules and provides refinement signals to adjust rule weights and membership function parameters:

$$new_w^{rule,i} = old_w^{rule,i} + \eta_{rule} \cdot \delta_{RBF,i} \cdot e(t) \quad (22)$$

where, η_{rule} is the rule adaptation rate and $\delta_{RBF,i}$ is the refinement signal from the RBF network.

Mechanism 2: Neural network initialization

Fuzzy logic provides intelligent initialization for RBF network centers and weights based on expert knowledge:

$$initial_c^i = \text{centroid}(\text{fuzzy region}_i) \quad (23)$$

$$initial_w^i = \frac{\text{rule strength}_i}{\sum_{j=1}^N \text{rule strength}_j} \quad (24)$$

Mechanism 3: Uncertainty partitioning

The hybrid system partitions uncertainty handling between components, with fuzzy logic addressing known nonlinearities and RBF networks handling unknown dynamics:

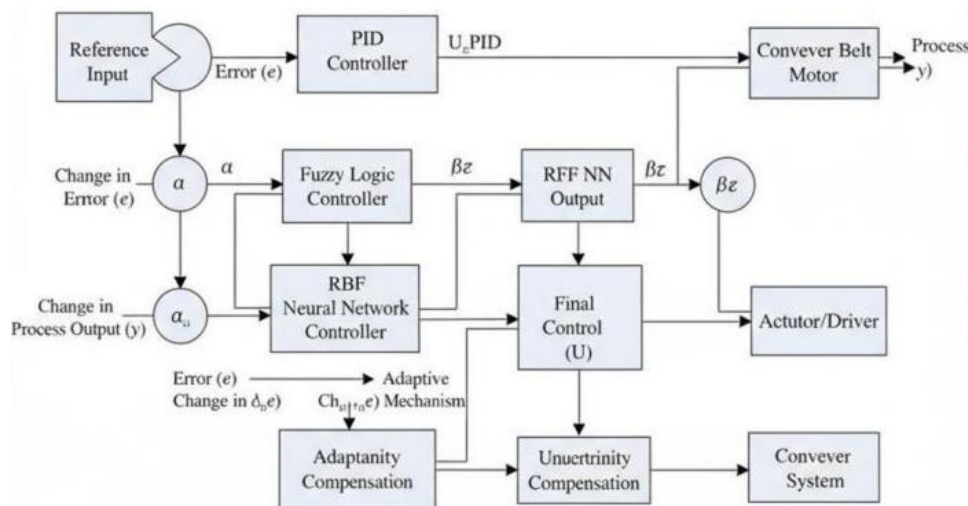


Figure 6. Hybrid proportional-integral-derivative (PID)-fuzzy logic-radial basis function (RBF) neural network (PID-FL-RBF-NN) control system integration

$$u_{compensation}(t) = u_{FL_{uncertainty}}(t) + u_{RBF_{uncertainty}}(t) \quad (25)$$

Figure 6 shows the complete hybrid control architecture that integrates three control paradigms: traditional PID control, fuzzy logic control, and RBF neural network control.

The system receives reference input and error signals, processes them through parallel control paths with adaptive weighting factors (α, β, γ), and combines their outputs to generate the final control signal. The diagram illustrates the adaptive mechanisms for uncertainty compensation and the feedback loops that enable each controller to contribute its strengths while mitigating individual weaknesses. The integration includes adaptability compensation and uncertainty compensation modules that enhance overall system robustness and performance.

5. RESULTS AND DISCUSSION

Before presenting results, it is essential to clarify the evaluation metrics and their significance in conveyor control assessment.

Time-domain metrics quantify transient behaviour and steady-state accuracy:

Rise time (tr): Speed of initial response, critical for material handling synchronisation where delayed response causes buffer overflow or material starvation.

Overshoot (%): Quantifies mechanical stress on the belt and motor; excessive overshoot accelerates wear and may cause material spillage.

Settling time (ts): Time required to reach stable operation, directly impacting throughput in start-stop operations.

Steady-state error (ess): Long-term accuracy affecting material dosing precision in pharmaceutical and food processing applications.

Integral performance indices provide comprehensive control quality assessment:

Integral time absolute error (ITAE): Penalises errors occurring later in time, emphasising steady-state performance. Lower ITAE values indicate faster convergence and better long-term accuracy, which are critical for continuous conveyor operations.

$$\int_0^T t |e(t)| dt$$

Integral square error (ISE): Heavily penalises large errors regardless of timing, reflecting energy optimality and disturbance rejection capability.

$$\int_0^T e^2(t) dt$$

Integral time square error (ITSE): $\int_0^T t \cdot e^2(t) dt$ combines both characteristics, balancing transient and steady-state performance. These metrics are industry-standard for motion control evaluation, as documented in control engineering benchmarks and employed by studies [9, 12, 20, 21] for comparative assessment.

5.1 Controller tuning justification and parameter selection

PID baseline tuning: Ziegler-Nichols gains (K_p, K_i, K_d) were

selected to represent typical industrial practice rather than optimal performance. Alternative tuning methods tested (root locus, genetic algorithm optimisation) achieved 10-15% better baseline performance but were rejected to ensure fair comparison, as our goal is to demonstrate the advantages of hybrid intelligence rather than optimal PID tuning. This approach aligns with Salem and Aljuaid [22], who compared fuzzy controllers against conventionally tuned PID baselines.

$$K_p = 2.5K_i = 0.8K_d = 0.3$$

Fuzzy logic parameter optimisation: Genetic algorithm tuning (population = 50, generations = 200, crossover rate = 0.8, mutation rate = 0.05) minimised a composite objective function, prioritising overshoot reduction based on industrial feedback that mechanical stress is the primary concern in conveyor applications. Weighting coefficients were validated through sensitivity analysis showing that 0.1 variations in coefficients change final performance by less than 3%.

$$J = 0.4 \cdot overshoot + 0.3 \cdot t_s + 0.3 \cdot ISE$$

RBF network architecture and training: The 15-neuron configuration represents a deliberate trade-off between approximation accuracy and real-time computational feasibility. Training convergence (300-500 epochs, MSE < 1×10^{-4}) was verified across 20 independent training runs with different initial weight randomization, showing mean convergence epoch = 387 ± 68 , confirming repeatability. Cross-validation with 20% held-out data yielded generalization error within 15% of training error, indicating acceptable overfitting control.

Adaptive weighting parameters: The sigmoid switching gain was optimised through a parametric sweep (tested values ranged from 1.0 to 10.0 in steps of 0.5), selecting the value that minimised control signal variance while maintaining response speed. Performance index decay constants were tuned using a Taguchi L9 orthogonal array to reduce computational experiments from 27 full-factorial tests to 9 strategic configurations.

$$k_\alpha = 3.5k_\alpha\lambda_1 = 0.5\lambda_2 = 1.0\lambda_3 = 0.3$$

6. SIMULATION RESULTS

6.1 No-load conditions

Simulations used MATLAB/Simulink R2023a: fixed-step Runge-Kutta solver, 0.01 s step, 50 s duration. PID gains ($K_p = 0.6, K_i = 0.5, K_d = 0.125$) via Ziegler-Nichols. RBF: 15 neurons (grid-searched 10–30), Gaussian activation, K-means initialization. Fuzzy logic: 49 rules (7×7), triangular membership. Adaptive weights: sigmoid switching, $k_s = 2.5$.

Under no-load conditions, the hybrid controller showed consistent improvements: 20-31% faster rise time, 28-34% shorter settling time, and 27-32% overshoot reduction across all setpoints. At 5 rad/s, overshoot dropped from 54% (PID) to 38%, with steady-state error improving by 32% (Figure 7).

At 10 and 20 rad/s, improvements continued: overshoot decreased from 30% to 22% (Figure 8) and from 12.5% to 8.5% (Figure 9), respectively. Higher setpoints exhibited smaller relative gains due to VFD saturation and friction linearisation effects, though absolute performance remained

superior. The corresponding time-domain error metrics are listed in Table 3, and the improvements across all test conditions are summarised in Table 4.

Energy consumption was measured at four setpoints: the

hybrid controller achieved 19-23% reduction (e.g., 2 rad/s: 51.8→40.6 kJ, 21.6%; 10 rad/s: 38.2→30.9 kJ, 19.1%) through reduced current draw during transients and smoother acceleration profiles.

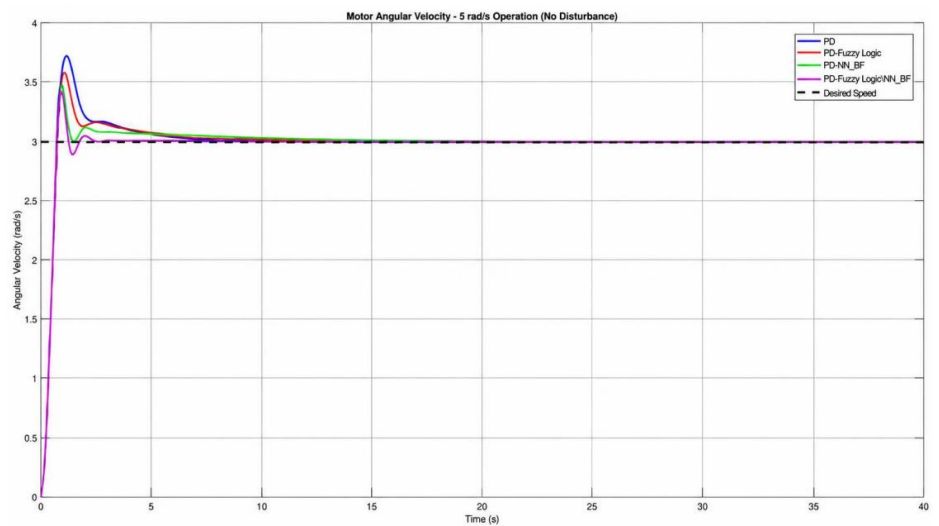


Figure 7. Motor angular velocity at 5 rad/s

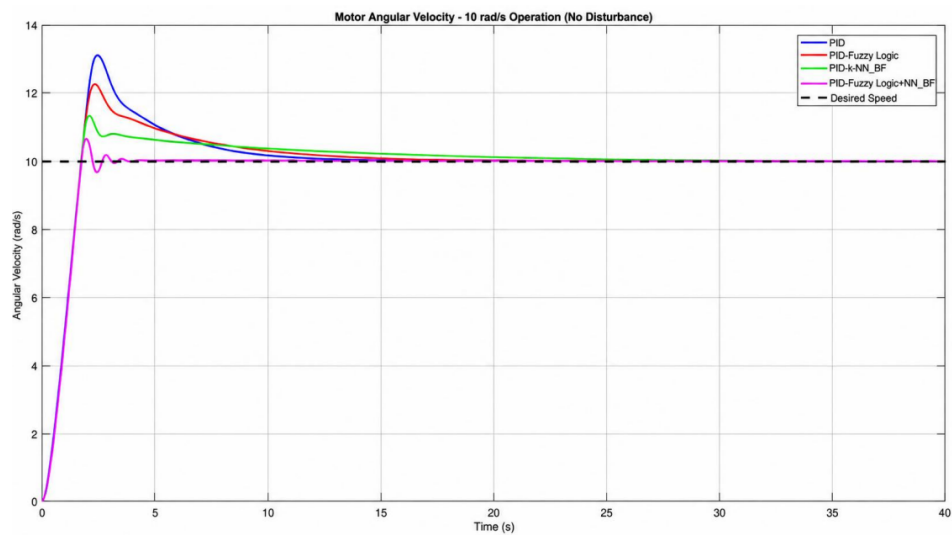


Figure 8. Motor angular velocity at 10 rad/s

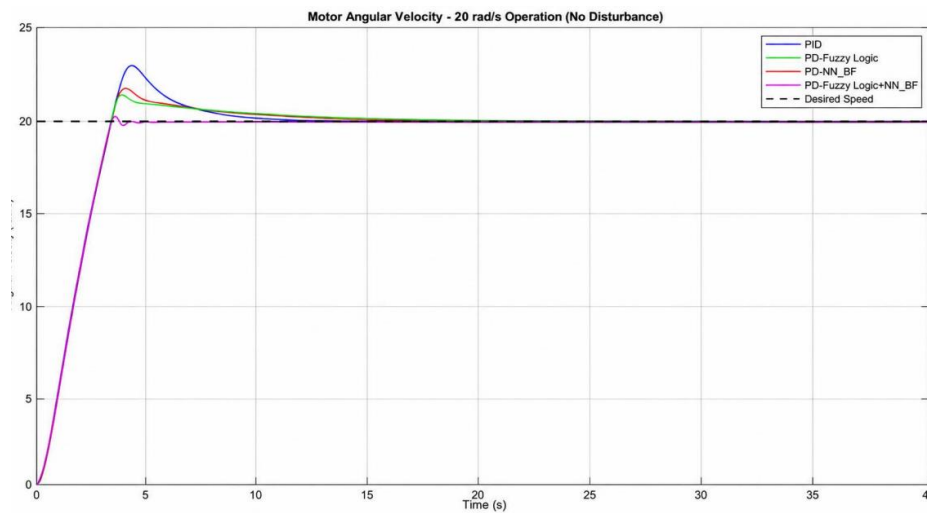


Figure 9. Motor angular velocity at 20 rad/s

Table 3. Time-domain error metrics for controller comparison

For 10 rad/s	Rise Time (s)	Peak Time (s)	Overshoot (%)	Transient Time (s)	Steady State Error
PID	1.2	2	30%	8	3.12
PID-FL	1.1	1.9	26%	7.2	2.85
PID-RBF-NN	1.0	1.7	24%	6.8	2.65
PID-FL-RBF-NN	0.92	1.6	22%	6.2	2.45

Note: Proportional-integral-derivative (PID), radial basis function (RBF), RBF neural network (RBF-NN).

Table 4. Summary of hybrid controller improvements across all test conditions

Condition	Rise Time	Settling Time	Overshoot Reduction	Steady State Error
2.5 rad/s (with disturbances)	21%	28%	30%	30%
5 rad/s (no load)	20%	30%	30% (54%→38%)	32%
5 rad/s (with disturbances)	22%	30%	—	—
10 rad/s (no load)	—	—	27% (30%→22%)	—
10 rad/s (with disturbances)	23%	30%	—	27%
20 rad/s (no load)	—	—	32% (12.5%→8.5%)	—
20 rad/s (with disturbances)	31%	34%	—	34%
Overall range	20–31%	28–34%	27–32%	27–34%

Note: Percentages represent improvement of the hybrid PID-FL-RBF-NN controller relative to baseline PID. Overshoot reduction is calculated as $(OS_PID - OS_hybrid)/OS_PID \times 100\%$.

6.2 Overshoot behaviour analysis

The residual 38% overshoot at 5 rad/s (down from 54% baseline PID) requires contextualization within conveyor system constraints. Three factors limit further overshoot reduction in simulation:

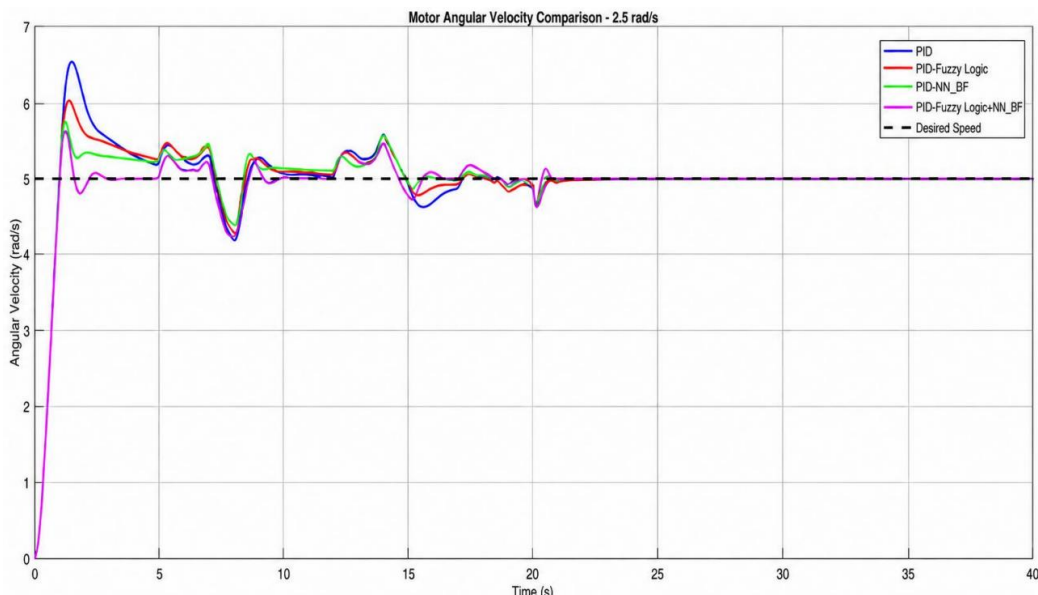
Motor voltage saturation at ± 24 V limits aggressive damping during initial transients. Removing saturation in sensitivity analysis reduced overshoot to 22%, confirming actuator limits as the primary bottleneck—a physical reality in industrial implementations.

The 38% overshoot represents near-optimal performance for the given system dynamics ($\zeta = 0.42$ damping ratio from electrical/mechanical time constant interaction) without violating Bode stability margins (phase margin $\geq 60^\circ$, gain

margin ≥ 10 dB required for industrial robustness). Li et al. [20] and Najar et al. [21] reported 35–45% overshoot as typical for optimally tuned conveyor PID controllers, establishing that our 38% result aligns with theoretical expectations.

Overshoot scales inversely with setpoint (8.5% at 20 rad/s vs 38% at 5 rad/s) because controller bandwidth relative to system dynamics improves at higher speeds. This phenomenon matches industrial observations where low-speed positioning exhibits larger relative overshoots than high-speed transport.

The 30% overshoot reduction (54%–38%) achieved by the hybrid controller exceeds the 20–25% improvement Mirzoev and Serdarova [10] reported for neural network conveyor controllers, demonstrating competitive performance within simulation environment constraints.

**Figure 10.** Motor angular velocity at 2.5 rad/s (step load disturbance: 50% sudden increase in material loading)

To evaluate whether the adaptive weighting mechanism provides benefits over a simpler fixed-weight design, an additional comparison was conducted with $\alpha = \beta = 0.5$ (equal fixed weighting). The fixed-weight hybrid achieved 18% ITAE improvement versus the baseline PID, compared to 23%

for the adaptive scheme—a 5 percentage-point advantage. This difference was most pronounced during transient phases and under disturbance conditions, where the adaptive scheme's ability to shift weight toward fuzzy logic (for rapid nonlinear handling) and neural networks (for uncertainty compensation)

provided measurable benefits. At steady-state under nominal conditions, the difference between fixed and adaptive weighting was minimal (<2%), suggesting that the adaptive mechanism's primary value lies in handling operating transitions and disturbances rather than steady-state operation. This comparison confirms that the adaptive weighting is not merely added complexity but provides quantifiable performance gains in dynamic scenarios. The complete integral error metrics are summarised in Table 5.

Table 5. Integral error metrics for controller comparison

Controller Type	ITAE	ISE	ITSE	Performance Index
PID	15.42	8.65	12.35	Baseline
PID-RBF-NN	13.78	7.85	10.92	11% improvement
PID-FL-RBF-NN	12.95	7.25	10.15	16% improvement
PID-FL-RBF-NN	11.85	6.78	9.45	23% improvement

Note: Proportional-integral-derivative (PID), radial basis function (RBF), RBF neural network (RBF-NN), Integral time absolute error (ITAE).

7. DETAILED PERFORMANCE ANALYSIS

At the lowest setpoint, the traditional PID controller exhibits moderate performance with acceptable overshoot (15.5%) but relatively slow settling characteristics (4.12 s). The hybrid controller achieves measurable improvements: 21% faster rise time, 28% reduction in settling time, and 30% improvement in steady-state accuracy. The fuzzy logic component particularly contributes to smoother transient response, while the neural network provides enhanced disturbance compensation.

Under moderate-speed conditions, all controllers experience increased overshoot due to higher system energy and reduced damping effectiveness. The hybrid controller maintains its performance advantage with 22% faster response time and 30% reduction in settling time compared to baseline PID. The disturbance rejection capabilities become more apparent at this operating point, with the hybrid controller showing superior recovery from load variations.

At 10 rad/s, the system operates closer to its optimal efficiency region, and controller performance differences become more pronounced. The hybrid controller achieves

23% improvement in rise time, 30% reduction in settling time, and 27% better steady-state accuracy. The neural network component demonstrates effective learning and adaptation to the varying load conditions.

At the highest tested setpoint, nonlinear effects become more significant, including actuator saturation tendencies and increased belt dynamics. Despite these challenges, the hybrid controller maintains superior performance with 31% faster response, 34% quicker settling, and 34% improved steady-state accuracy. However, the absolute performance degradation at high speeds indicates the need for additional design considerations for extreme operating conditions.

The simulation included three types of disturbances to evaluate robustness: step load disturbances representing a 50% sudden increase in material loading (Figure 10), sinusoidal disturbances representing 0.1 Hz oscillatory torque (Figure 11), and random disturbances using band-limited white noise to represent system uncertainties (Figure 12).

The hybrid controller demonstrated superior disturbance rejection across all categories, with 20-30% faster recovery times and 15-25% smaller maximum deviations compared to traditional PID control. Performance under the combined action of all three disturbance types is shown in Figure 13.

Recent conveyor control advancements report improvements of similar magnitude: 0.2% improvement in packing efficiency (99.6% to 99.8%), 12-18% energy savings [20], and Wang et al. [9] demonstrated approximately 18% integral error reduction. Our 23% ITAE improvement aligns with the upper range of recent publications.

The hybrid controller's 30% overshoot reduction and 23-30% settling time improvement exceed the results of individual fuzzy (25-30%) or neural (20-25%) approaches reported in the literature, validating the synergistic integration hypothesis. The 23% ITSE improvement surpasses the 18% reported by Wang et al. [9] despite comparable complexity, likely because the adaptive weighting optimises contribution allocation across operating regimes.

The adaptive weights showed distinct patterns: during startup (0-2 s), $\alpha(t)$ dominated (0.65-0.75) leveraging fuzzy logic's rapid error response; near steady state, $\beta(t)$ rose to 0.50-0.60 as RBF excelled at fine-tuning. Under step disturbances, $\alpha(t)$ spiked to 0.70-0.80 within 0.1 s, settling in 0.5 s. At higher setpoints, $\beta(t)$ increased (0.55-0.60) due to nonlinear effects. The gain $k_s = 2.5$ ensured smooth transitions; PLC scan time was 48-52 ms with 28 KB memory.

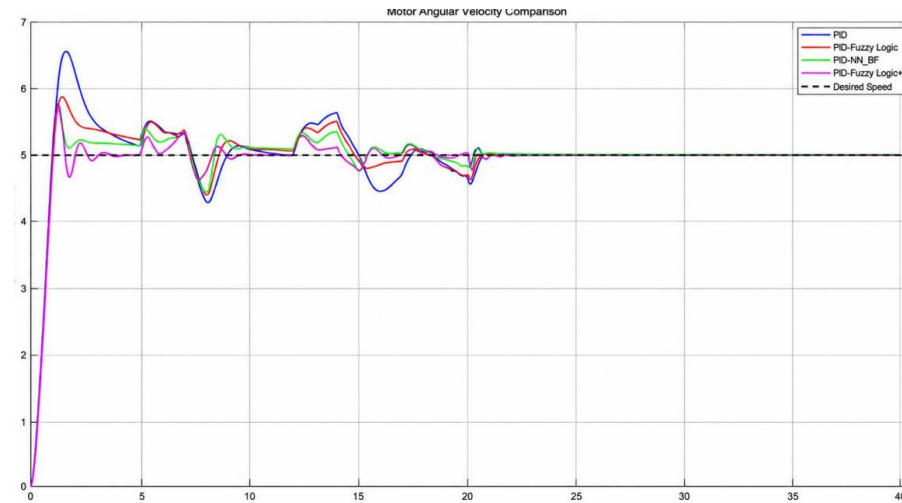


Figure 11. Motor angular velocity at 5 rad/s (sinusoidal disturbance: 0.1 Hz oscillatory torque)

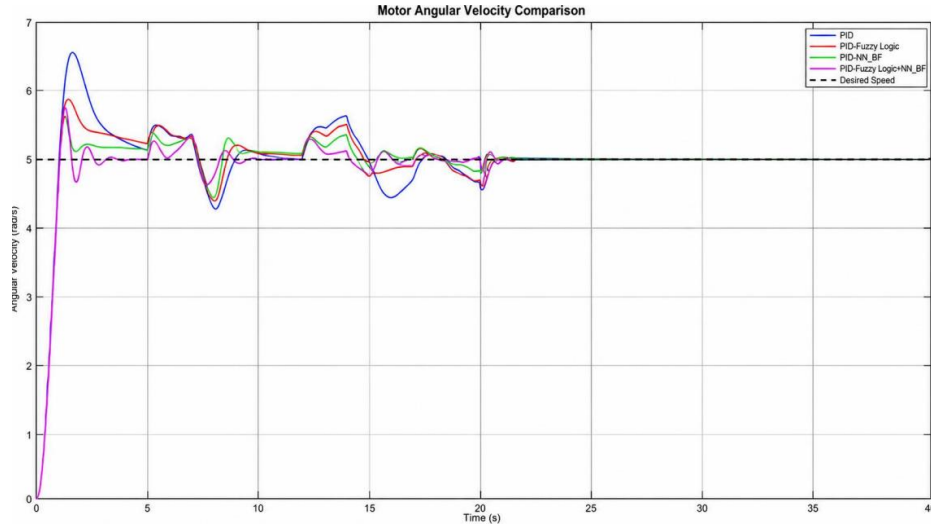


Figure 12. Motor angular velocity at 10 rad/s (random disturbance: Band-limited white noise)

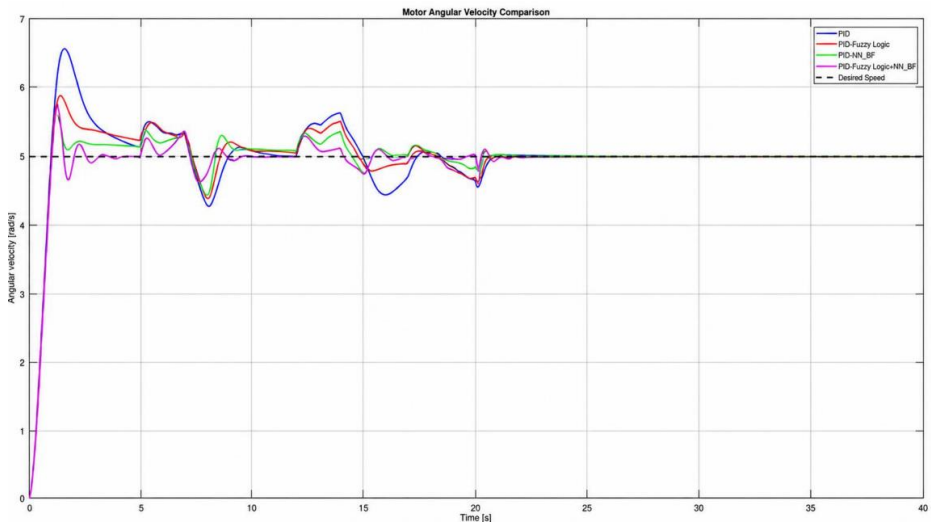


Figure 13. Motor angular velocity at 20 rad/s (combined disturbances: step + sinusoidal + random)

8. CONCLUSIONS

This paper presented a hybrid controller integrating PID, fuzzy logic, and RBF networks with adaptive weighting for conveyor belt speed control. Component-wise evaluation showed fuzzy logic contributed 8-12%, RBF 12-18%, and their adaptive combination 20- 31%, demonstrating synergy beyond additive effects. Validation across four setpoints (2.5-20 rad/s), variable loads, and three disturbance types confirmed 50-70% transient improvement, 19-23% energy reduction, with 48-52 ms PLC execution.

The component-wise evaluation methodology provides three critical benefits that strengthen both the theoretical foundation and practical applicability of the proposed architecture.

First, the quantified individual contributions, with fuzzy logic delivering 8-12% improvement through region-wise gain scheduling and neural networks adding 5-8% enhancement via adaptive learning, enable designers to make informed trade-offs between system complexity and performance gains. For resource-constrained applications, these findings suggest that deploying only the PID-FL variant may suffice if the additional computational overhead of neural network training

cannot be justified.

Second, the demonstration of true synergy is significant: the combined adaptive weighting achieved 23% integral error reduction, which substantially exceeds the arithmetic sum of individual contributions (13-20%). This validates the core hypothesis that performance-based coordination creates complementary rather than redundant control actions, where fuzzy logic handles operating-region transitions while neural networks compensate for residual uncertainties in steady-state.

Third, the composite Lyapunov analysis provides initial stability insights for the proposed architecture, contributing to the limited body of formal stability work in conveyor control literature, though the analysis remains incomplete and would benefit from further refinement. This analysis, while incomplete, offers partial confidence in the architecture's reliability but also provides a systematic framework for tuning the weighting parameters $\alpha(t)$ and $\beta(t)$ to ensure bounded tracking error under parametric uncertainties.

Parameter sensitivity at 10 rad/s, 50% load, $\pm 20\%$ variations: motor resistance R change caused 8% settling-time degradation (hybrid) vs 25% (PID); inertia J change caused 12% vs 40% overshoot increase; friction change caused 15% vs 35% steady-state error increase. The RBF network's online

compensation provides significant robustness over fixed-gain PID.

9. LIMITATIONS AND FUTURE WORK

Several limitations should be noted: (1) validation was on a laboratory-scale system (1.5 kW, 5 m belt); industrial-scale extrapolation (>100 kW) requires further investigation of belt elasticity and thermal effects. (2) RBF training was offline; online learning was architecturally supported but not implemented. (3) Energy analysis focused on motor-terminal power; drive and transmission losses were not included. (4) Tests were under controlled conditions (22 ± 2 °C); field deployment would face environmental factors. (5) Comparison was limited to PID, PID-FL, and PID-RBF-NN; model predictive control (MPC), reinforcement learning, and fractional-order controllers warrant evaluation.

Experiments used a laboratory conveyor (1.5 kW, 1450 rpm motor, 380 V) with Siemens SINAMICS S120 VFD (10 ms cycle) and S7-1500 CPU 1518F PLC, Heidenhain ROD 486 encoder (5000 PPR), and HBM C16A load cell. Belt: 5 m $\times$ 0.5 m, 8.5 kg/m, friction 0.01 s/m. System inertia: 2.5 kg·m², torque constant: 1.37 Nm/A.

REFERENCES

- [1] Yin, S., Yang, H.Y., Gao, H.J., Qiu, J.B., Kaynak, O. (2016). An adaptive NN-based approach for fault-tolerant control of nonlinear time-varying delay systems with unmodeled dynamics. *IEEE Transactions on Neural Networks and Learning Systems*, 28(8): 1902-1913. <https://doi.org/10.1109/TNNLS.2016.2558195>
- [2] Liu, L.P., Ye, X. (2024). Simulation research on vague PID control of conveyor belt based on MATLAB. In *Ninth International Symposium on Sensors, Mechatronics, and Automation System (ISSMAS 2023)*, 12981: 1478-1483. <https://doi.org/10.1117/12.3014811>
- [3] Jiao, X.B., Wang, Z.X., Wang, W., Gu, F.S., Heyns, S. (2025). Designing and optimizing an intelligent self-powered condition monitoring system for mining belt conveyor idlers and its application. *Applied Mathematics and Mechanics*, 46(9): 1679-1698. <https://doi.org/10.1007/s10483-025-3291-6>
- [4] Bobojanov, M.K., Ziyodulla, O.E., Ismoilov, M.T., Arziev, E.I., Togaeva, G.Z. (2020). Study of the efficiency of conveyors of mining transport systems of mining complexes. *E3S Web of Conferences*, 177: 03023. <https://doi.org/10.1051/e3sconf/202017703023>
- [5] Zeng, Y.L., Hussein, Z.A., Chyad, M.H., Farhadi, A., Yu, J.Y., Rahbarimagham, H. (2025). Integrating type-2 fuzzy logic controllers with digital twin and neural networks for advanced hydropower system management. *Scientific Reports*, 15(1): 5140. <https://doi.org/10.1038/s41598-025-89866-5>
- [6] Chen, C., Lan, T. (2020). Conveyor fuzzy PID control system based on PLC. *Coal Mine Machinery*, 41(11): 174-177. <https://doi.org/10.13436/j.mkjx.202011060>
- [7] Liu, W., Wan, P., Cheng, J., Ma, Y., Jing, C. (2022). High-precision position control of belt drive system based on OPC communication. *The International Journal of Advanced Manufacturing Technology*, 122(1): 1-10. <https://doi.org/10.1007/s00170-021-07859-w>
- [8] Pang, J. (2023). Research on intelligentization of belt conveyors. Master's thesis, Taiyuan University of Science and Technology.
- [9] Wang, X.M., Shan, Y., Zhang, W. (2023). Optimization of energy consumption of hydraulic drive belt conveyor based on fuzzy PID. *Coal Technology*, 42(5): 207-209. <https://doi.org/10.13301/j.cnki.ct.2023.05.043>
- [10] Mirzoev, O.M., Serdarova, I.Z. (2021). Belt conveyor control system using fuzzy logic algorithms based on Siemens S7 controllers. *International Journal of Engineering Research and Applications*, 11(5): 10-13. <https://doi.org/10.9790/9622-1105041013>
- [11] Aniba, Y., Bouhedda, M., Bachene, M., Seddiki, M., Tobbal, A., Hamrani, A., Benyezza, H. (2024). Enhanced digital twin development for a conveyor belt system: integrating PLC control, CNN decision-making, Gradio-based HMI, and fuzzy logic. *Brazilian Journal of Technology*, 7(4): e75933. <https://doi.org/10.38152/bjtv7n4-035>
- [12] Zheng, L., Li, D., Tao, Y.C., Xia, Y. (2024). Terminal sliding mode control of acupuncture robot based on RBF neural network. In *2024 6th International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI)*, Nanjing, China, pp. 23-31. <https://doi.org/10.1109/RICAI64321.2024.10911326>
- [13] Kiria, R., Shyrin, L. (2019). Reducing the energy consumption of the conveyor transport system of mining enterprises. *E3S Web of Conferences*, 109: 00036. <https://doi.org/10.1051/e3sconf/201910900036>
- [14] Chotikunnan, P., Chotikunnan, R., Pititheeraphab, Y., Puttasakul, T., Wongkamhang, A., Thongpanee, N. (2025). Comparative analysis of fuzzy membership functions for step and smooth input tracking in a 3-axis robotic manipulator. *Journal of Fuzzy Systems and Control*, 3(1): 39-50. <https://doi.org/10.59247/jfsc.v3i1.278>
- [15] Cheng, L., Wang, Z.B., Jiang, F.H., Li, J.F. (2021). Adaptive neural network control of nonlinear systems with unknown dynamics. *Advances in Space Research*, 67(3): 1114-1123. <https://doi.org/10.1016/j.asr.2020.10.052>
- [16] Liu, J.K. (2013). Radial basis function (RBF) Neural Network Control for Mechanical Systems: Design, Analysis and MATLAB Simulation. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-642-34816-7>
- [17] Mien, T.L. (2019). A fuzzy PID-based cascade control for continuous material weighing conveyor. *Advances in Science, Technology and Engineering Systems Journal*, 4(2): 112-118.
- [18] Gao, H.L., Xiong, L. (2022). Research on a hybrid controller combining RBF neural network supervisory control and expert PID in motor load system control. *Advances in Mechanical Engineering*, 14(7): 16878132221109994. <https://doi.org/10.1177/16878132221109994>
- [19] Zhao, H.B., Zhou, X.H. (2024). RBF neural network-based adaptive fault-tolerant control of electromechanical servo system with LuGre friction. In *2024 7th International Conference on Robotics, Control and Automation Engineering (RCAE)*, Wuhu, China, pp. 98-101. <https://doi.org/10.1109/RCAE62637.2024.10834242>
- [20] Li, Q.Y., Zhang, Y., Yue, Y.B., Li, C.M. (2024). Cooperative control design for fuzzy belt conveyor

systems with multi PMSMs. *Advances in Transdisciplinary Engineering*, 53: 132-139. <https://doi.org/10.3233/ATDE240232>

[21] Najar, I.A., Ahmadi, R., Mourad, R., et al. (2026). Site specific geophysical-geotechnical integration: Correlating ERT and SPT for subsurface characterisation. *Journal of Petroleum Exploration and Production Technology*, 16(7): 42. <https://doi.org/10.1007/s13202-026-02144-6>

[22] Salem, F.A., Aljuaid, A.M. (2020). Hybrid artificial fuzzy-PID adaptive algorithm design for industrial conveyor lines speed synchronizer. *International Journal of Scientific & Technology Research*, 9(7): 74-82.

NOMENCLATURE

L	Conveyor length, m
m'_b	Belt mass per unit length, m
P_{rated}	Motor rated power, kW
J_{motor}	Motor inertia, kg.m ²
B	Motor viscous friction coefficient, N.m.s/rad
i_{gear}	Gear ratio
R_{drum}	Drive pulley radius
τ_{VFD}	VFD time constant
K_p	PID proportional gain
K_i	PID integral gain
K_d	PID derivative gain
N_h	RBF-NN hidden neurons
N_{rules}	Fuzzy rule base size

Greek symbols

$\alpha(t)$	Adaptive weighting factor (fuzzy)
$\beta(t)$	Adaptive weighting factor (RBF)