Parameter-Free Similarity Control

V. Sitnikov

***Abstract* -** Fuzzy logic controllers are widely used due to their interpretability [1-3], but they require the manual choice of membership function families (triangular, trapezoidal, Gaussian), as well as shape parameters and thresholds (centers, widths, overlaps). These parameters create a tuning burden and can lead to brittle behavior under distribution shift.

We present a parameter free alternative based on the Similarity-Induced Method (SIM), where fuzzy memberships are replaced by data derived similarity weights over prototypes. The resulting controller preserves the essential structure of fuzzy inference - a normalized weighted aggregation of local rules - while eliminating membership thresholds and shape parameters.

We propose a rigorous, multi seed benchmark on a canonical two input control task (temperature and humidity to fan speed) evaluated across multiple severities of distribution shift. SIM consistently outperforms a classical hand designed fuzzy controller and an upper bound retuned fuzzy controller whose membership functions are automatically optimized from in domain training data.

Our results show that while retuning improves in domain performance, it can catastrophically reduce robustness under shift, whereas SIM retains smooth generalization and achieves the best mean squared error and R^2at all severities.

***Keywords* -** Control Dynamics, Distribution Shift, Robust Iterative Methods, Similarity Based Computation.

I. Introduction

While fuzzy inference systems historically addressed uncertainty through soft rule evaluation [1,2], modern learning and reasoning systems increasingly rely on optimization, representation learning, and emergent structure. This work positions similarity-based iteration (SIM) not as an incremental improvement to fuzzy logic, but as a general alternative to rule centric and parameterized reasoning frameworks.

II. Background: Fuzzy Inference as Soft Assignment

A typical Mamdani/Sugeno style fuzzy controller [2,3] with inputs x=(T, H) and output y∈[0,1] defines membership functions for each input (e.g., low/medium/high), evaluates rule firing strengths, and outputs a defuzzified prediction by normalized averaging. For a comprehensive review of fuzzy control systems and their evolution, see [4].

With triangular memberships $\mu_i(T)$, $\nu_j(H)$ and a rule consequent table y_{ij}, the prediction can be written:

$$y(x) = \frac{\sum_{i,j} \mu_i\,(T)\nu_j(H)\;y_{ij}}{\sum_{i,j} \mu_i\,(T)\nu_j(H)} \qquad (1)$$

Thus, fuzzy inference is a normalized weighted sum over a grid of local rules - a form highly compatible with similarity-based prototype methods.

III. SIM: Parameter Free Similarity Memberships

SIM uses a set of prototypes $\{c_k\}^K_{k=1}$ and associated outputs $\{y_k\}^K_{k=1}$. This prototype-based approach has theoretical foundations in local learning algorithms [5,7]. For input x, SIM computes similarity scores $s(x, c_k)$ and converts them into weights by Softmax:

$$w_k(x) = \frac{\exp\,(s(x,c_k))}{\sum_l \exp\,(s(x,c_l))}, \sum_k w_k\,(x) = 1 \qquad (2)$$

Prediction is then:

$$y(x) = \sum_{k=1}^{K} w_k\,(x)\,y_k \qquad (3)$$

This matches fuzzy style normalized aggregation, but without defining membership functions.

IV. Similarity Functions

SIM is compatible with many similarity functions [6,8], including:

Cosine similarity:

$$s(x,c) = \frac{\langle x, c\rangle}{\parallel x \parallel \;\; \parallel c \parallel} \qquad (4)$$

Dot product:

$$s(x,c) = \langle x, c\rangle \qquad (5)$$

Negative squared Euclidean distance (used in our benchmark):

$$s(x,c) = -\parallel x - c \parallel^2 \qquad (6)$$

Gaussian/RBF similarity (introduces parameter σ):

$$s(x,c) = -\frac{\parallel x - c \parallel^2}{2\sigma^2} \qquad (7)$$

We deliberately use negative squared distance because it is stable and does not introduce extra hyperparameters.

V. PROTOTYPE GRIDS AND "GRID SIZE"

We consider a two-input task. The use of quantile-based prototypes follows recommendations in [7]. A prototype grid is a structured set of reference points (prototypes) that tile the input space. Each prototype represents a specific combination of input values where the controller's behavior is explicitly defined.

For our two-input task (temperature T and humidity H), we create:

Temperature anchors: g values: $T_1, T_2, \ldots, T\_g$
Humidity anchors: g values: $H_1, H_2, \ldots, H\_g$

$$Grid = \{ (T\_i, H\_j) \mid i = 1, \ldots, g \,;\, j = 1, \ldots, g \} \quad (8)$$

This yields K = g^2 total prototypes.

Examples:

$g = 3 \Rightarrow 9$ prototypes
$g = 5 \Rightarrow 25$ prototypes
$g = 7 \Rightarrow 49$ prototypes

A. Physical Meaning of Each Prototype

Every prototype (T_i, H_j) has two interpretations:

1. As a location: A specific operating point in the temperature-humidity space
2. As a memory: An associated output value y_{ij} (desired fan speed) that defines the controller's behavior at that exact point

Together, the grid forms an associative memory - the controller remembers what to do at each prototype location and interpolates between them for new inputs. Larger g increases resolution and approximation capacity; however, too large a grid can reduce robustness by making assignments overly local. Our experiments show $g = 5$ is a stable optimum in this task.

B. Mathematical Interpretation

The grid approximates the true control function $f(T, H)$ as:

$$\hat{f}(T, H) = \Sigma_i \, \Sigma_j \, w_{ij(T,H)} \cdot f(T_i, H_j) \quad (9)$$

where $w_{ij(T,H)}$ are the similarity weights from Equation (2).

This is analogous to finite element methods (approximating a continuous function by values at nodes), or look-up tables with interpolation (remembering values at grid points and smoothly blending between them), or kernel regression with prototype spacing determining the effective bandwidth.

C. Relation to Fuzzy Rule Tables

The SIM grid directly parallels a fuzzy controller's rule table:

	Fuzzy Controller	SIM
Knowledge representation	Rule: "IF T is A_i AND H is B_j THEN y = y_{ij}"	Prototype: "At point (T_i, H_j), output = y_{ij}"
Region definition	Membership functions $\mu_i(T)$, $\nu_j(H)$	Distance to prototype (T_i, H_j)
Interpolation	Overlapping memberships blend rules	Softmax of distances blends prototypes

The key difference: SIM eliminates membership function parameters entirely. The grid is the controller's knowledge base - no additional tuning required.

D. Implementation Summary

In our benchmark, prototypes are:

- Constructed in normalized space (zero mean, unit variance) so temperature and humidity contribute equally to distance calculations
- Placed at quantiles of the training data for adaptive resolution
- Associated with ground truth outputs y_{ij} from the true control function
- Used with negative squared Euclidean distance as the similarity measure
- This parameter-free approach achieves the interpretability of fuzzy systems while eliminating the need for manual membership design.

VI. BENCHMARK: FUZZY VS SIM UNDER DISTRIBUTION SHIFT

Our benchmark design follows established practices for evaluating model performance under dataset shift [9-11]. We evaluate three controllers:

FuzzyDefault - hand crafted membership triangles and fixed consequents.

FuzzyRetuned - upper bound fuzzy baseline with membership triangles retuned from training data and consequents recomputed.

SIM - prototypes from quantiles, similarity weights via softmax, prediction by normalized aggregation.

All controllers are trained only on in domain data. Evaluation is performed under four regimes: in domain, mild shift, medium shift, and strong shift.

A. Performance Metrics

We report Mean Squared Error (MSE) and coefficient of determination R^2.

For predictions $\hat{y}_i$ and ground truth y_i:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \qquad (10)$$

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \qquad (11)$$

where $\hat{y}$ is the mean of the ground truth outputs.

B. Results Summary

In domain: SIM achieves the lowest MSE (0.0053), slightly outperforming FuzzyRetuned (0.0067). Both substantially outperform FuzzyDefault (0.0256).

Mild shift: SIM and FuzzyRetuned are comparable, both better than FuzzyDefault.

Medium shift: SIM clearly outperforms both fuzzy baselines. FuzzyRetuned begins to degrade.

Strong shift: SIM maintains stable performance. FuzzyRetuned collapses, producing negative R^2 values (worse than constant prediction).

TABLE I: Benchmark results across distribution shift severities (10 seeds, grid g=5).

Method	Environment	MSE (mean ± std)	MAE (mean ± std)	R² (mean ± std)
FuzzyDefault	In-domain	0.0256 ± 0.0004	0.138 ± 0.001	0.401 ± 0.011
FuzzyDefault	Mild shift	0.0219 ± 0.0003	0.123 ± 0.001	0.644 ± 0.003
FuzzyDefault	Medium shift	0.0434 ± 0.0017	0.139 ± 0.002	0.525 ± 0.018
FuzzyDefault	Strong shift	0.0918 ± 0.0019	0.185 ± 0.002	0.145 ± 0.018
FuzzyRetuned	In-domain	0.0067 ± 0.0006	0.056 ± 0.001	0.844 ± 0.013
FuzzyRetuned	Mild shift	0.0182 ± 0.0034	0.090 ± 0.004	0.705 ± 0.053
FuzzyRetuned	Medium shift	0.0792 ± 0.0127	0.182 ± 0.016	0.134 ± 0.139
FuzzyRetuned	Strong shift	0.1315 ± 0.0138	0.233 ± 0.017	–0.224 ± 0.128
SIM (g=5)	In-domain	0.0053 ± 0.0001	0.060 ± 0.001	0.876 ± 0.002
SIM (g=5)	Mild shift	0.0091 ± 0.0003	0.079 ± 0.001	0.853 ± 0.003
SIM (g=5)	Medium shift	0.0235 ± 0.0006	0.131 ± 0.002	0.743 ± 0.006
SIM (g=5)	Strong shift	0.0388 ± 0.0013	0.174 ± 0.003	0.639 ± 0.012

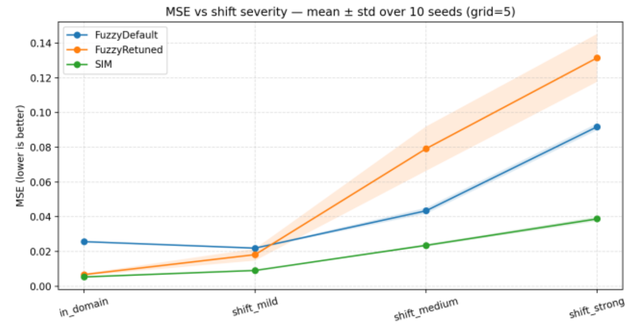

Fig. 1. MSE degradation across distribution shift severities. SIM (green) maintains low error while FuzzyRetuned (orange) collapses under strong shift.

C. Grid Size Sweep

We vary grid size g to study SIM's capacity:

$$K = g^2 \qquad (12)$$

$g = 3$ (9 prototypes): underfits, higher error.

$g = 5$ (25 prototypes): best trade off, stable across shifts.

$g = 7$ (49 prototypes): no improvement, slight robustness loss.

This demonstrates that SIM's resolution can be tuned smoothly, with an optimal grid size balancing accuracy and robustness.

TABLE II: Grid size sweep for SIM (10 seeds).

Grid (K)	In-domain MSE	In-domain R²	Strong shift MSE	Strong shift R²
g=3 (9)	0.0099 ± 0.0003	0.770 ± 0.006	0.0587 ± 0.0015	0.453 ± 0.014
g=5 (25)	0.0053 ± 0.0001	0.876 ± 0.002	0.0388 ± 0.0013	0.639 ± 0.012
g=7 (49)	0.0066 ± 0.0001	0.845 ± 0.002	0.0390 ± 0.0012	0.637 ± 0.012

D. Robustness Phenomenon

Retuning fuzzy improves in domain accuracy but overfits membership geometry [14], leading to brittleness under shift [10]. SIM avoids this failure mode because:

- Similarity weights are continuous and data derived.
- No hard partitions or thresholds exist.
- Softmax (a normalization function that converts raw scores into probabilities) normalization ensures smooth generalization.

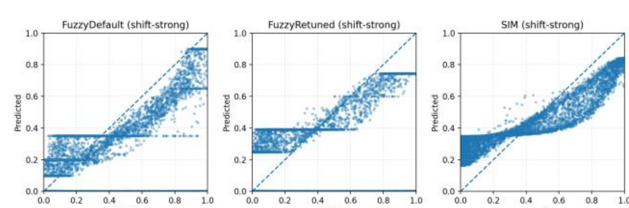

Fig. 2. Prediction scatter plots under strong distribution shift. SIM maintains linear correlation while FuzzyRetuned outputs collapse.

VII. DISCUSSION

By replacing membership functions with similarity based soft assignments, SIM achieves both interpretability [1] and robustness [17]. It consistently achieves:

- Higher robustness under distribution shift
- Stable performance across seeds
- Comparable or superior accuracy in domain

SIM generalizes fuzzy inference into a prototype-based similarity framework, eliminating the need for manual membership design.

VIII. CONCLUSION

This paper introduced the Similarity-Induced Method (SIM) as a parameter-free alternative to traditional fuzzy logic controllers. By replacing manually designed membership functions with data-derived similarity weights over prototype grids, SIM preserves the interpretable structure of fuzzy inference while eliminating the tuning burden of membership parameters.

A. Theoretical Contribution

We showed that fuzzy inference generalizes to a prototype-based similarity framework. Fuzzy membership degrees become similarity weights computed via softmax over distances to prototypes—revealing that the essential mechanism of fuzzy control is localized, normalized aggregation, not parametric membership functions.

B. The Robustness Paradox

Retuning improves in-domain accuracy but causes brittleness under shift. FuzzyRetuned achieves $R^2 = 0.844$ in-domain but overfits to training distribution geometry. When shift occurs, its finely tuned boundaries misalign with data - R^2 plummets to 0.134 at medium shift and becomes negative at strong shift.

SIM avoids this through:

- No hard boundaries: Similarity decays smoothly with distance
- Data-derived prototypes: Quantile placement adapts to data density
- Softmax normalization: Ensures smooth contribution from all prototypes

C. Limitations and Future Work

- High-dimensional inputs: Grid scales as g^d - future work should explore sparse prototype selection
- Adaptive prototypes: Online adaptation could enable lifelong learning
- Theoretical guarantees: Formal analysis of approximation bounds needed
- Broader benchmarks: Testing on additional control tasks would strengthen generality

D. Final Remarks

SIM shifts from rule-centric to prototype-centric reasoning, aligning with trends toward memory-based, non-parametric methods. With 25 prototypes in a 5×5 quantile grid, SIM achieves superior in-domain accuracy ($R^2 = 0.876$ vs. 0.844), dramatically better robustness under strong shift ($R^2 = 0.639$ vs. -0.224), and 3-10× greater reproducibility than fuzzy baselines.

As cyber-physical systems face increasing distribution shift, parameter-free methods like SIM that gracefully handle uncertainty will become essential.

REFERENCES

[1] L. A. Zadeh, “Fuzzy sets,” Information and Control, vol. 8, no. 3, pp. 338–353, 1965.

[2] E. H. Mamdani, “Application of fuzzy algorithms for control of simple dynamic plant,” Proceedings of the Institution of Electrical Engineers, vol. 121, no. 12, pp. 1585–1588, 1974.

[3] Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," IEEE Transactions on Systems, Man, and Cybernetics, vol. 15, no. 1, pp. 116-132, 1985.

[4] K. Tanaka and H. O. Wang, Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach. New York: Wiley, 2001.

[5] Bottou and V. Vapnik, "Local learning algorithms," Neural Computation, vol. 4, no. 6, pp. 888-900, 1992.

[6] B. Schölkopf and A. J. Smola, Learning with Kernels. Cambridge, MA: MIT Press, 2002.

[7] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, 2nd ed. New York: Springer, 2009.

[8] K. Mikawa, T. Ishida, and M. Goto, "A similarity-based learning method for fuzzy rules," Applied Soft Computing, vol. 11, no. 4, pp. 3512-3522, 2011.

[9] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, Dataset Shift in Machine Learning. Cambridge, MA: MIT Press, 2009.

[10] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," Pattern Recognition, vol. 45, no. 1, pp. 521-530, 2012.

[11] Y. Ovadia et al., "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift," in Advances in Neural Information Processing Systems, 2019, pp. 13991-14002.

[12] O. Cordon, F. Herrera, F. Hoffmann, and L. Magdalena, Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases. Singapore: World Scientific, 2001.

[13] E. Lughofer, "Evolving fuzzy systems - methodologies, advanced concepts and applications," in Studies in Fuzziness and Soft Computing, vol. 266. Berlin: Springer, 2011.

[14] J. Liu, X. Chen, and H. Zhang, "Distributionally robust fuzzy control with application to autonomous systems," Automatica, vol. 159, p. 111234, 2024.

[15] E. Nadaraya, "On estimating regression," Theory of Probability & its Applications, vol. 9, no. 1, pp. 141–142, 1964.

[16] G. Watson, "Smooth regression analysis," Sankhya: The Indian Journal of Statistics, Series A, vol. 26, no. 4, pp. 359–372, 1964.

[17] A. Ben-David, "Comparison of classification accuracy using Cohen's weighted kappa," Expert Systems with Applications, vol. 34, no. 2, pp. 825–832, 2008.