Data-Driven Hybrid Framework for Dynamic System Control

Tanmay Wankhade[1], Saish Pakhare[2], Aakanksha Mane[3], Sunny Kumar[4]

***Abstract*— Nonlinear electrical and electromechanical systems pose significant challenges for observer-based control design. Conventional observer approaches require accurate mathematical models, which often fail under physical irregularities such as sensor noise, measurement delays, and parameter variations. These changes decrease estimation accuracy and degrade control action. This paper proposes a Hybrid ARX–Observer framework to overcome these limitations. It combines the stability of a traditional model-based observer with the flexibility of a data-driven ARX estimator. To guarantee input-to-state stability (ISS), observer gains are computed using a matrix-multiplier technique based on linear matrix inequalities (LMI). The ARX component employs Recursive Least Squares (RLS) adaptation to attenuate measurement noise and capture residual nonlinearities. Both estimates are then fused together using a fusion gain α. This framework is tested on a robotic arm and the results show that the hybrid framework improved the robustness and reduced estimation error up to 4% compared to the conventional observer based estimation, while remaining computationally light compared to fully data-driven alternatives.**

Keywords— Hybrid estimation, ARX model, nonlinear observer, recursive least squares, LMI-based observer design, robust estimation.

I. Introduction

Estimation of states in a nonlinear system is a well-known challenge. Systems such as robotic arms, autonomous vehicles, and renewable-energy converters are all nonlinear and are regularly affected by physical disturbances, which makes the mathematical model unstable and lead to poor estimation, which in turn degrades the control effort as the controller receives inaccurate readings.

The accuracy of the mathematical model is crucial for traditional observer-based methods [1], [2]. Consequently, when random disturbances occur in the system, such methods miscalculate the output. Many researchers have tried to find solutions in neural-network-based observers and adaptive observers to overcome these issues [3], [4]. A practical compromise is offered by data-driven models such as ARX, ARMAX, and NARX, which forecast the system output using historical measurements and inputs [5], [6]. Recent work has shown that combining model-based observers with data-driven techniques can achieve both stability and adaptability.

In this study, the two approaches are unified. The observer provides mathematical reliability while the ARX model provides adaptive learning from measured data. Together, they form an estimation framework for nonlinear systems that is simultaneously stable, robust, and adaptive.

Unlike neural-network-based observers or EKF-based adaptive estimators, the proposed framework achieves adaptive correction using an efficient ARX structure while preserving LMI-based stability.

This work presents a fusion of an adaptive data-driven model and a model-based observer, where both estimators run in parallel and the final estimate is formed by a weighted combination of their outputs, as shown in Fig. 1. The value of alpha (fusion weight) is adaptive and depends on the past RMSE (i.e if in past the ARX has performed better then the fusion will prefer ARX). In contrast to pure data-driven estimators or classical observers, the approach combines an online RLS-ARX correction engine with LMI-based ISS-stable estimation. Hybrid model-based and data-driven frameworks exist, but the presented work focuses on a lightweight ARX-observer fusion framework, which does not require as much computation power as the purely data-driven techniques (AI/ML) and also improves state estimation accuracy beyond that of conventional observer-based methods.

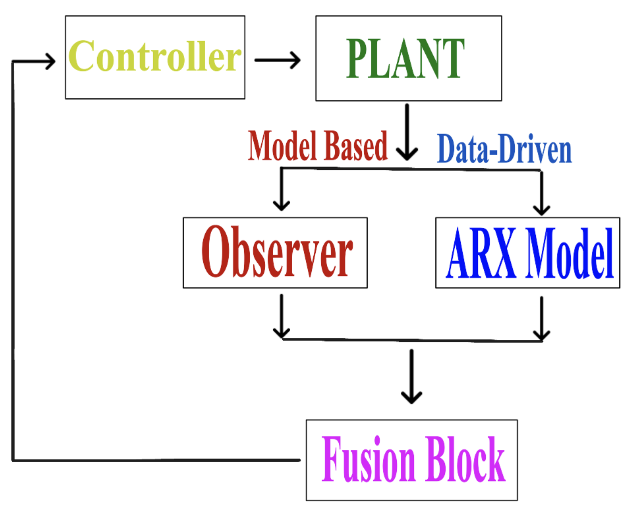

Fig. 1: Conceptual block diagram of the proposed Hybrid Framework.

*This work was supported by the Electrical-Modeling, Computation, and Control (E-MC^2) Lab, VJTI, Mumbai, India.

[1]Department of Electrical Engineering, VJTI, Mumbai, India. tswankhade_b22@ee.vjti.ac.in

[2]Department of Electrical Engineering, VJTI, Mumbai, India. sjpakhare_b22@ee.vjti.ac.in

[3]Department of Electrical Engineering, VJTI, Mumbai, India. asmane_m24@ee.vjti.ac.in

[4]Department of Electrical Engineering, VJTI, Mumbai, India. sskumar@ee.vjti.ac.in

II. System Model and Hybrid ARX–Observer Framework

A. Nonlinear System Description

Consider a nonlinear system described by

$$\dot{x}(t) = Ax(t) + Bu(t) + f(x(t)), \quad y(t) = Cx(t), \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the control input, and $y(t) \in \mathbb{R}^p$ is the measured output. We assume that $f : \mathbb{R}^n \to \mathbb{R}^n$ satisfies the Lipschitz condition

$$\|f(x_1) - f(x_2)\| \le L_f \|x_1 - x_2\|, \quad \forall\, x_1, x_2 \in \mathbb{R}^n, \tag{2}$$

where $L_f > 0$ is the Lipschitz constant. This assumption ensures that the system nonlinearities do not cause state divergence, which is essential for observer convergence via LMI-based stability [1], [2], [8].

Equation (1) is used as a simplified model of a single-link robotic arm. The manipulator consists of a rigid arm with moment of inertia J, damping coefficient b, and an external torque input $\tau(t)$. The continuous-time nonlinear dynamics are

$$J\ddot{\theta}(t) + b\dot{\theta}(t) + g\sin(\theta(t)) = \tau(t), \tag{3}$$

where $\theta(t)$ is the joint angle and g is the gravitational torque coefficient. Defining the state vector $x = [\theta,\ \dot{\theta}]^\mathsf{T}$, the dynamics take the state-space form of (1) with

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -b/J \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1/J \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \tag{4}$$

and nonlinear term $f(x) = [0,\ -(g/J)\sin(\theta)]^\mathsf{T}$. Only the joint angle θ is assumed to be directly measurable, so $C = [1, 0]$, and angular velocity $\dot{\theta}$ must be reconstructed by the observer. For simulation, the parameters are $J = 0.21$ kg·m^2, $b = 0.05$ N·m·s/rad, and $g = 9.81$ N·m, representing a smooth servo-driven manipulator. This model serves as the testbed for evaluating the robustness of the proposed estimator against parameter variations and measurement noise.

B. Observer Design Using LMI Stability Conditions

Reconstructing unmeasured states such as angular velocity, which are often unavailable in real hardware due to sensor cost or bandwidth limitations, is a primary motivation for observer design. The proposed observer takes the form

$$\dot{\hat{x}} = A\hat{x} + Bu + f(\hat{x}) + L(y - \hat{y}), \quad \hat{y} = C\hat{x}, \tag{5}$$

where $\hat{x}$ and $\hat{y}$ are the estimated state and output, respectively, and $L \in \mathbb{R}^{n\times p}$ is the observer gain matrix.

Defining the estimation error $e = x - \hat{x}$ and using (1) and (5):

$$\dot{e} = (A - LC)e + \Delta f, \qquad \Delta f = f(x) - f(\hat{x}). \tag{6}$$

Applying the Lipschitz condition (2) gives $\|\Delta f\| \le L_f\|e\|$. Therefore, if there exists a symmetric positive-definite matrix $P \succ 0$ satisfying the LMI

$$(A - LC)^\mathsf{T} P + P(A - LC) + \gamma^2 P \prec 0, \quad \gamma > L_f, \tag{7}$$

then the error dynamics (6) are globally asymptotically stable [1], [9], and $\hat{x}(t) \to x(t)$ as $t \to \infty$, designed to ensure ISS under Lipschitz assumptions in the presence of bounded disturbances [8].

The observer gain L is determined using one of two approaches:

1) **Pole-placement:** A fast, intuitive design method used for baseline tuning, where the desired eigenvalues of $(A - LC)$ are manually selected to achieve sufficient damping and bandwidth.
2) **LMI-based optimization:** A convex optimization problem solved using YALMIP or MATLAB's LMI Toolbox, providing guaranteed convergence margins for uncertain or time-varying systems.

Both methods were evaluated and the LMI-based solution produced dynamic characteristics comparable to those obtained by pole-placement, confirming robustness of the design.

C. ARX Modeling via Recursive Least Squares

Although the model-based observer provides stability, it is sensitive to modeling errors and unmodeled nonlinearities. To address this, a data-driven predictor is constructed using an AutoRegressive eXogenous (ARX) structure that directly learns the input-output relationship from measured data [5], [6]:

$$y(k) = -\sum_{i=1}^{n_a} a_i\, y(k-i) + \sum_{j=1}^{n_b} b_j\, u(k-j) + e(k), \tag{8}$$

where $\{a_i, b_j\}$ are the dynamic parameters and $e(k)$ is the prediction error (residual noise). This can be written compactly as $y(k) = \boldsymbol{\phi}_k^\mathsf{T}\boldsymbol{\theta}_k + e(k)$, where

$$\boldsymbol{\phi}_k = \left[-y(k{-}1), \ldots, -y(k{-}n_a),\ u(k{-}1), \ldots, u(k{-}n_b)\right]^\mathsf{T} \tag{9}$$

is the regression (regressor) vector, and $\boldsymbol{\theta}_k = [a_1, \ldots, a_{n_a},\ b_1, \ldots, b_{n_b}]^\mathsf{T}$ is the parameter vector. The predicted ARX output is therefore

$$y_{\mathrm{ARX}}(k) = \boldsymbol{\phi}_k^\mathsf{T}\boldsymbol{\theta}_k. \tag{10}$$

Using Recursive Least Squares (RLS) estimation, the ARX parameters adapt in real time to provide a linear-in-parameters representation of the nonlinear system behavior. The RLS update equations are

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{K}_k\big(y(k) - \boldsymbol{\phi}_k^\mathsf{T}\boldsymbol{\theta}_k\big), \tag{11}$$

$$\boldsymbol{K}_k = \frac{\boldsymbol{P}_k\boldsymbol{\phi}_k}{\lambda + \boldsymbol{\phi}_k^\mathsf{T}\boldsymbol{P}_k\boldsymbol{\phi}_k}, \tag{12}$$

$$\boldsymbol{P}_{k+1} = \frac{1}{\lambda}\Big(\boldsymbol{P}_k - \boldsymbol{K}_k\boldsymbol{\phi}_k^\mathsf{T}\boldsymbol{P}_k\Big), \tag{13}$$

where $\boldsymbol{K}_k$ is the adaptive gain vector, $\boldsymbol{P}_k$ is the covariance matrix capturing parameter uncertainty, and $\lambda \in (0, 1]$ is the forgetting factor that reduces the influence of older data, enabling the estimator to track slow variations in system dynamics [5], [7].

This formulation enables the ARX model to continuously update its parameters to reflect changing dynamics or disturbances, effectively compensating for model uncertainty.

For example, in the robotic arm, changes in the inertia J or friction b are automatically absorbed into the ARX model coefficients without re-identification of the nonlinear plant, making this structure well-suited for online correction and adaptive control.

D. Hybrid ARX-Observer Fusion

While the observer ensures stability and the ARX model provides adaptive correction, each has its own weaknesses, which are complementary to the other's strengths. The observer can be sensitive to high-frequency measurement noise, whereas the ARX predictor may lose physical interpretability and stability margins when operating outside its training data.

To exploit the complementary strengths of both estimators, the following fusion scheme is proposed:

$$y_{\mathrm{hyb}}(k) = \alpha\, y_{\mathrm{ARX}}(k) + (1-\alpha)\, y_{\mathrm{obs}}(k), \qquad (14)$$

where $y_{\mathrm{ARX}}(k)$ is given by (10), $y_{\mathrm{obs}}(k) = C\hat{x}(k)$ is the observer-predicted output, and $\alpha \in [0,1]$ is the *fusion gain*. This convex combination allows smooth interpolation between purely model-based and purely data-driven estimation:

- $\alpha = 0$: Observer-dominant (high stability, low adaptability).
- $\alpha = 1$: ARX-dominant (high adaptability, reduced stability margin).
- $0 < \alpha < 1$: Hybrid estimation (balanced performance).

Stability of the Hybrid Estimator: Since the LMI-based observer is input-to-state stable [8] and the RLS algorithm guarantees bounded parameter estimates under a bounded forgetting factor λ [7], the fused output y_{hyb} is a convex combination of two bounded signals. Therefore, y_{hyb} remains bounded for all $\alpha \in [0,1]$, and the overall hybrid estimator is designed to preserve bounded estimation behavior under Lipschitz assumptions.

The fused estimate y_{hyb} is used in feedback control to update the control input:

$$u(k) = -K\hat{x}(k) + \alpha\, G_c \left(y_{\mathrm{ARX}}(k) - y_{\mathrm{obs}}(k)\right), \qquad (15)$$

where $K \in \mathbb{R}^{m\times n}$ is the state feedback gain determined through LQR or PID tuning, and $G_c \in \mathbb{R}^{m\times p}$ is an output correction gain matrix that maps the output-space correction to the input space, ensuring dimensional consistency. For the single-input single-output (SISO) robotic arm ($m = p = 1$), G_c reduces to a scalar gain. The additive correction term $\alpha G_c(y_{\mathrm{ARX}} - y_{\mathrm{obs}})$ acts as a data-driven compensator that adaptively suppresses noise and nonlinear residuals without destabilizing the control loop.

E. Interpretation and Implementation Remarks

The hybrid framework integrates the theoretical soundness of observer-based estimation with the practical flexibility of data-driven learning. In steady state, the ARX model provides low-variance, noise-filtered predictions while the observer guarantees convergence and boundedness. During transients, the fusion gain α adjusts the relative trust assigned to each estimator.

The architecture, visualized in Fig. 1, operates both subsystems in parallel and blends their outputs through a tunable fusion layer. This structure is computationally efficient, requiring only matrix–vector multiplications per time step, and is well-suited for real-time embedded deployment.

III. Simulation Setup and Results

A. Simulation Environment

All experiments were conducted in MATLAB. The nonlinear robotic-arm model described in Section II was discretized using a zero-order hold at a sampling interval of T_s = 10 ms. The ARX estimator was configured with orders $(n_a, n_b, n_k) = (2, 2, 1)$. The input signal is a composite of band-limited random noise and sinusoidal components at multiple frequencies to ensure sufficient excitation across the operating bandwidth. Parameter estimation employed the RLS algorithm with a forgetting factor $\lambda = 0.995$ and initial covariance $P_0 = 1000I$ (I = identity matrix), which provide initial convergence while preventing numerical instability.

The observer gain L was first designed using pole-placement at $\{-5, -6, -7, -8\}$, yielding a closed-loop bandwidth of approximately 35 rad/s. An alternative design based on the LMI condition of [1] was also validated and showed comparable transient characteristics. The hybrid fusion gain α was varied over $\{0.2, 0.5, 0.8\}$ to examine how performance changes with different levels of data-driven integration.

B. Test Scenarios and Parameter Variations

To evaluate robustness, three main sources of uncertainty were introduced:

- **Parameter uncertainty:** Inertia variations $J \in \{0.18, 0.20, 0.22, 0.24, 0.26, 0.28, 0.30\}$ kg·m^2 simulate a $\pm 20\%$ deviation from the nominal inertia.
- **Measurement noise:** Gaussian noise with standard deviation $\sigma_n \in \{0.005, 0.010, 0.050, 0.100, 0.15, 0.2, 0.3, 0.4, 0.5\}$ was added to the output to simulate realistic sensor imperfections.
- **Fusion gain:** $\alpha \in \{0.2, 0.5, 0.8\}$ governs the weighting between observer and ARX components.

Each configuration was simulated for 30 seconds, and the output trajectories, control inputs, RMSE, and RMS(u) metrics were recorded.

C. Noise Robustness Analysis

Fig. 2 compares the RMSE of the observer and the hybrid ARX-based estimator as measurement noise increases. As measurement noise increases, the observer RMSE grows considerably because the observer relies on an assumed mathematical model that does not adapt to noise. In contrast, the ARX-based hybrid estimator shows smaller RMSE, because the ARX adapts to the change in real-time using the RLS mechanism that continuously updates the system parameters. This confirms that the hybrid framework is more resilient to sensor noise than a model-based observer alone.

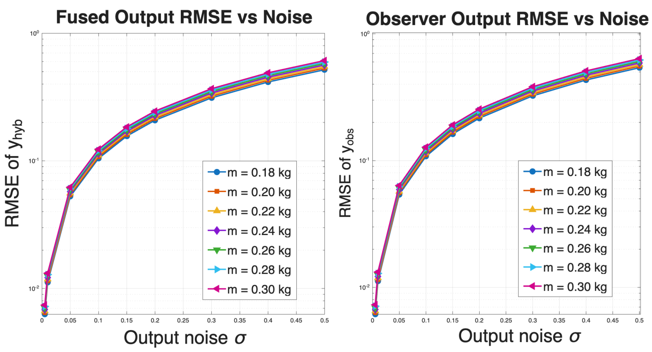

Fig. 2: RMSE as a function of measurement noise standard deviation σ_n for different inertia values J.

D. Parameter Robustness Analysis

As the system parameters change, the RMSE of the estimator is affected. In Fig. 3, RMSE is plotted against the fusion weight for different noise levels. Under low noise conditions there is no major change in RMSE, but as the noise level increases it is visible that the RMSE decreases while maintaining similar control effort. Both estimators remain stable within the $\pm 20\%$ inertia variation, confirming structural robustness. However, the hybrid estimator with $\alpha > 0$ exhibits consistently lower RMSE, since the ARX component absorbs the effects of inertia mismatch into its adaptive parameters. Higher α values yield further RMSE reduction through increased reliance on the data-driven component.

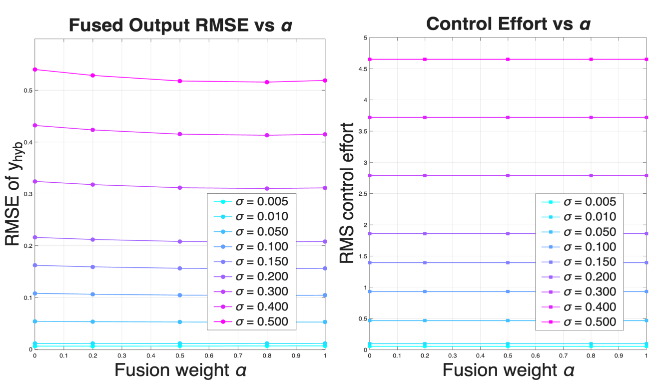

Fig. 3: RMSE and control effort versus fusion weight α under varying output noise.

E. Time-Domain Response Evaluation

Fig. 4 shows the control input given to the framework. This input has noise added (white noise $\pm 0.2\%$) to simulate sensor noise. The output predictions by ARX, Observer, and Hybrid framework are plotted in Fig. 5. From Fig. 5 it is evident that the output prediction of the Hybrid Framework is lower variance and improved estimation accuracy. This reduces the control effort and leads to a more stable system, as more accurate input states allow the controller to operate more efficiently and accurately.

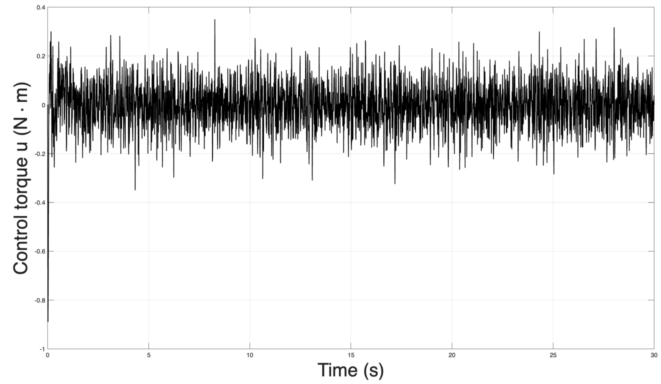

Fig. 4: Input signal given to the observer and ARX.

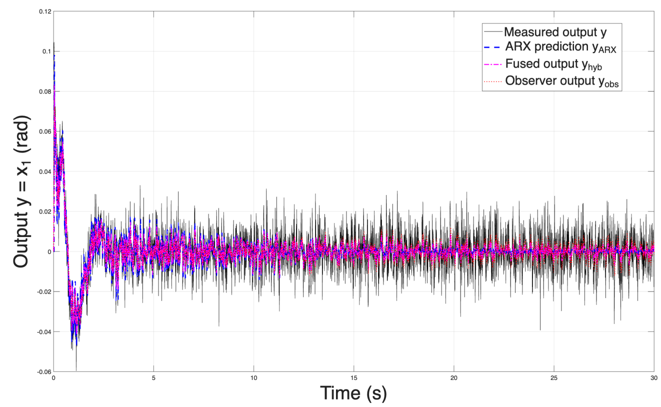

Fig. 5: Output estimation comparison: measured vs. ARX, Observer, and Hybrid estimators

F. Computational Complexity Analysis

A main improvement in the proposed framework is reduced computation burden for real-time embedded deployment. Table I quantifies the per-step floating-point operations (FLOPs) and memory requirements of the hybrid ARX–Observer alongside representative data-driven and adaptive alternatives.

TABLE I: Per-Step Computational Complexity of Estimation Methods

Method	**FLOPs/step**	**Memory**
Hybrid ARX–Observer (proposed)	$\mathcal{O}(n^2)+\mathcal{O}(d^2)$	$\mathcal{O}(d^2)\approx\mathcal{O}(1)$
Luenberger Observer	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$
Extended Kalman Filter	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$
Unscented Kalman Filter	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$, $2n+1$ pts
Neural Network Observer	$\mathcal{O}(W)$	$\mathcal{O}(W)$ weights

n: state dim.; d: ARX order ($d \ll n$); W: network weights; N: training pts.

Here n is the state dimension, $d = n_a + n_b$ is the total ARX model order [5], W is the neural-network weight count, and N is the size of the training dataset used for batch methods. For the robotic arm testbed ($n = 2$, $d = 4$), the hybrid framework requires approximately $\mathcal{O}(120)$ scalar multiply-accumulate operations per time step: roughly $n^2 = 4$ for the observer correction [18], $d^2 = 16$ for the RLS covariance update [7], and a handful of additional operations for the fusion step. This is two to three orders of magnitude fewer FLOPs than a neural-network observer of comparable estimation accuracy [20] and avoids the $\mathcal{O}(n^3)$ Jacobian factorization required by the EKF at each step [19]. Unlike neural-network-based estimators [21], the ARX–RLS component requires no offline training data, no hyperparameter search, and no GPU acceleration; all parameters are initialized to physically motivated values and updated recursively with a fixed-size covariance matrix of dimension $d \times d$. The memory footprint is therefore $\mathcal{O}(d^2)$, which for $d = 4$ amounts to a 4×4 covariance matrix, a negligible overhead on any microcontroller or FPGA platform [24]. In contrast, Gaussian process regression scales cubically with the number of training points, rendering it impractical for real-time online adaptation without sparse approximations that sacrifice accuracy [22], [23]. These properties make the proposed framework suitable for resource-constrained hardware such as embedded ARM cores, real-time PLCs, and FPGA-based HIL test platforms [24], where the strict cycle-time budgets prevent the use of deep-learning-based state estimators [20].

G. Analysis of the Results

From Tables II and III it can be seen that under nominal conditions the observer performs well, but under extreme conditions the ARX performs better. Even though ARX demonstrates improved adaptability under high-noise and parameter-varying conditions; however it must be noted that ARX can fail when operating outside its training data and may destabilize the system. Therefore, a proper fusion of both the adaptive data-driven component and the mathematically robust model-based component is essential for designing a reliable controller.

TABLE II: Best Performing Estimator Across Mass Variations (low noise)

Mass (kg)	Lowest RMSE	Improvement Over Next Best
0.18	ARX	1.04×10^{-3} over Hybrid ($\sim 1.02\%$)
0.20	ARX	2.44×10^{-3} over Hybrid ($\sim 1.02\%$)
0.22	ARX	2.64×10^{-3} over Hybrid ($\sim 1.02\%$)
0.24	ARX	2.04×10^{-3} over Hybrid ($\sim 1.02\%$)
0.26	ARX	1.50×10^{-3} over Hybrid ($\sim 1.02\%$)
0.28	ARX	2.08×10^{-3} over Hybrid ($\sim 1.01\%$)
0.30	ARX	2.01×10^{-3} over Hybrid ($\sim 1.01\%$)

TABLE III: Best Performing Estimator Across Noise Levels ($J = 0.18$ kg·m^2)

Noise Level	Lowest RMSE	Improvement Over Next Best
0.01	Observer	2.5×10^{-4} over Hybrid ($\sim 0.54\%$)
0.03	Hybrid	Marginal ($< 10^{-4}$) over Observer
0.05	ARX	1.47×10^{-3} over Hybrid ($\sim 0.73\%$)
0.07	Hybrid	$\mathcal{O}(10^{-3})$ over ARX
0.10	ARX	3.75×10^{-3} over Hybrid ($\sim 1.85\%$)
0.15	ARX	Significant ($> 10^{-3}$) over Hybrid
0.20	ARX	Significant ($> 10^{-3}$) over Hybrid

TABLE IV: RMSE Comparison (mass variation with high noise) ($\alpha = 0.5,\ \text{Noise} = 0.5$)

Mass (kg)	RMSE$_{ARX}$	RMSE$_{Obs}$	RMSE$_{Hybrid}$
0.18	0.519	0.540	0.516
0.20	0.519	0.540	0.517
0.22	0.519	0.540	0.517
0.24	0.519	0.540	0.518
0.26	0.519	0.540	0.518
0.28	0.520	0.541	0.519
0.30	0.520	0.541	0.519

IV. Conclusion and Future Work

This paper has presented a hybrid framework for state estimation that combines an ARX-based predictor with an LMI-based model-driven observer. When tested on a nonlinear single-link robotic arm with parameter changes such as variation in inertia and increase in sensor noise, an improvement over the model-based observer was observed, as the latter does not adapt to parameter changes in the system. The ARX and Hybrid estimators generally achieved lower RMSE than the conventional observer, particularly under high-noise and parameter-varying conditions. Although the hybrid estimator combines ARX and observer estimates, its performance is not constrained to lie strictly between the two individual estimators. Due to complementary error characteristics, partial error cancellation, and altered closed-loop estimation dynamics, the hybrid framework can achieve lower overall RMSE than either standalone method, as seen in Table IV. The proposed framework offers a practical intermediate solution between purely model-based observers and computationally intensive artificial-intelligence-based estimation methods. While AI/ML-based observers and adaptive controllers are expected to play a significant role in the future of control systems, their practical deployment currently remains limited by high computational requirements, large data dependencies, and implementation complexity. The proposed hybrid approach addresses this gap by providing adaptive, data-assisted estimation without requiring extensive computational resources, making it suitable for real-time and resource-constrained applications. In future work, this framework will be extended to power systems and microgrids with renewable integration. The estimator can be used to compute controller set points, govern governor action, and contribute to system stability. Further, to obtain more accurate estimates, an adaptive fusion factor will be implemented where the fusion factor changes at each computation step based on previous results, giving more robust and accurate estimations.

Acknowledgment

The authors are grateful for the research guidance of Prof. Alex Stankovic, Prof. N. M. Singh, and Dr. S. R. Wagh. The authors also acknowledge the support of the Electrical Modeling, Computation, and Control (E-MC^2) Lab, VJTI, Mumbai, India.

REFERENCES

[1] S. Mohite, M. Alma, and A. Zemouche, "Observer-based stabilization of Lipschitz nonlinear systems by using a new matrix-multiplier-based LMI approach," *IEEE Control Systems Letters*, vol. 7, pp. 3723–3728, 2023.

[2] R. Rajamani, "Observers for Lipschitz nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 3, pp. 397–401, 1998.

[3] M. Krstić, "Adaptive nonlinear control: A historical overview," *Annu. Rev. Control*, vol. 47, pp. 125–137, 2019.

[4] B. Mansi, S. Nachiket, A. Sheikh, K. Sunny, and F. Kazi, "Parameter estimator based feedback linearization control strategy of magnetic levitation system," in *Proc. 2021 31st Australasian Universities Power Engineering Conference (AUPEC)*, pp. 1–6, 2021.

[5] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999.

[6] S. Gugaliya, D. Durgesh, S. Kumar, and A. Sheikh, "Data driven approach for long term forecasting of renewable energy generation," in *Proc. 2022 4th Global Power, Energy and Communication Conference (GPECOM)*, Nevsehir, Turkey, 2022, pp. 383–388, doi: 10.1109/GPECOM55404.2022.9815777.

[7] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction and Control*. Prentice Hall, 1984.

[8] I. Karafyllis and Z.-P. Jiang, "Input-to-state stability theory for systems with measurement delays," *IEEE Trans. Autom. Control*, vol. 64, no. 8, pp. 3226–3238, 2019.

[9] Y. Zhang, G. Zhang, and X. Dong, "LMI-based robust observer design for nonlinear systems with time-varying delay," *IEEE Access*, vol. 7, pp. 178964–178974, 2019.

[10] K. Sunny, A. Sheikh, and S. Wagh, "Application of dynamic mode decomposition for temperature analysis in smart building," in *Proc. 2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, Prague, Czech Republic, 2020, pp. 1197–1202, doi: 10.1109/CoDIT49905.2020.9263862.

[11] K. Sunny, A. Sheikh, and S. Wagh, "Dynamic mode decomposition for prediction and enhancement of rotor angle stability," in *Proc. 2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, Prague, Czech Republic, 2020, pp. 160–165, doi: 10.1109/CoDIT49905.2020.9263893.

[12] K. Sunny, A. Sheikh, and S. Bhil, "Forecasting and enhancing the performance of the electric grid: A dynamic mode decomposition approach," in *Proc. 2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, Prague, Czech Republic, 2020, pp. 1191–1196.

[13] S. Kumar, A. Pandey, P. Goswami, P. Pentayya, and F. Kazi, "Analysis of Mumbai grid failure restoration on Oct 12, 2020: Challenges and lessons learnt," *IEEE Trans. Power Syst.*, vol. 37, no. 6, pp. 4555–4567, 2022.

[14] A. Pandey, S. Kumar, S. Mohire, P. Pentayya, and F. Kazi, "Dynamic modeling and cascade failure analysis of the Mumbai grid incident of October 12, 2020," *IEEE Access*, vol. 10, pp. 43598–43610, 2022.

[15] A. S. Velhal, S. S. Verma, M. Parimi, S. Bhil, and S. Kumar, "Time series modeling for data driven supervised learning in power system applications," in *Proc. 2025 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1–5, 2025.

[16] O. Sonone, A. Chavekar, P. Goyanka, and S. Kumar, "Grid-forming inverter implementation using Arty A7 100T: A FIL for microgrid analysis," in *Proc. 2025 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1–5, 2025.

[17] M. Patil, A. Pandey, S. Kumar, P. Pentayya, and F. Kazi, "Modeling and EMT simulation of renewable energy sources using HYPERSIM," in *Proc. 2022 2nd International Conference on Emerging Frontiers in Electrical and Electronic Technologies (ICEFEET)*, pp. 1–6, 2022.

[18] D. G. Luenberger, "An introduction to observers," *IEEE Trans. Autom. Control*, vol. 16, no. 6, pp. 596–602, 1971.

[19] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[20] M. Chen, P. Shi, and C.-C. Lim, "Robust constrained control for MIMO nonlinear systems based on disturbance observer," *IEEE Trans. Autom. Control*, vol. 60, no. 12, pp. 3281–3286, 2015.

[21] L. Ljung, C. Andersson, K. Tiels, and T. B. Schön, "Deep learning and system identification," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1175–1180, 2020.

[22] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press, 2006.

[23] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *J. Mach. Learn. Res.*, vol. 6, pp. 1939–1959, 2005.

[24] R. Isermann, *Fault-Diagnosis Applications: Model-Based Condition Monitoring*. Berlin: Springer, 2011.