

Picosecond-Grade Clocking for Smart Factories: Interferometric Sensing Meets Adaptive Consensus

Mohammed Faizan
mohammadfaizansm@gmail.com
Independent Researcher

Abstract

Deterministic production lines, TSN backbones, and coordinated motion systems demand clock alignment beyond conventional PTP; this paper introduces a hybrid scheme that fuses interferometry-based timing extraction with an adaptive consensus protocol to deliver sub-100 ps synchronization across distributed controllers. A mathematically grounded design establishes stability margins and optimal convergence rates, while the implementation spans simulation, SDR-based emulation, and FPGA-based hardware testbeds (Raspberry Pi 4 with Xilinx Artix-7 timestamping modules) to validate precision, speed, and resilience under loss, interference, and adversarial behavior. Experiments achieve 13.5 ± 2.1 ps time error and $1.5\times$ faster convergence versus White Rabbit PTP ($2.3\times$ versus standard PTP), with reduced message overhead and graceful degradation up to one-third faulty nodes, supporting large-scale factory networks. For automation engineers, the takeaway is a deployable timing layer that sharpens TSN schedules, tightens robot and vision triggers, and scales to thousand-node plants using commodity hardware.

Keywords

• Precision Time Protocol • White Rabbit • Time Synchronization • Adaptive Consensus • Distributed Control • Smart Manufacturing • Fault Tolerance • Interferometry

1. Introduction

Modern smart factories rely on tightly coordinated networks of controllers, sensors, and actuators to achieve deterministic production lines with minimal latency and jitter. Applications such as multi-robot collaborative assembly, synchronized vision inspection, and coordinated motion control require clock alignment that exceeds what conventional synchronization protocols like the Network Time Protocol [1] or even the Precision Time Protocol (PTP) can deliver. While PTP provides sub-microsecond synchronization under ideal conditions [2], its performance degrades significantly in large-scale factory networks with asymmetric link delays, packet loss, and node failures. The challenges of precise timing are also evident in optical interferometry applications [3], where phase stability directly impacts measurement accuracy.

The emergence of Time-Sensitive Networking (TSN) has brought deterministic Ethernet to industrial automation, with IEEE 802.1AS (generalized PTP, gPTP) as its core timing standard [4, 5]. However, even gPTP achieves only submicrosecond to microsecond accuracy, which is insufficient for applications such as high-speed robot ballet, where coordinated movements require timing precision in the tens of picoseconds to avoid collisions and maintain product quality. This gap between what industrial applications need and what existing protocols offer motivates our work.

White Rabbit, an extension of IEEE 1588 PTP originally developed at CERN [6], pushes synchronization accuracy into the sub-nanosecond range by combining physical layer syntonization, precise phase

detection, and link asymmetry calibration. White Rabbit achieves picosecond-level precision over fiber Ethernet, making it attractive for distributed instrumentation and accelerator facilities. However, White Rabbit has seen limited adoption in smart factory settings because its design assumes stable fiber links and fixed network topologies, whereas factory floors often experience dynamic changes, wireless segments, and varying numbers of active nodes.

Distributed consensus algorithms have proven effective for clock synchronization in networks where nodes can communicate with neighbors and iteratively converge to a common time reference. Average consensus-based protocols, such as Average TimeSync, offer robustness to packet loss and node failures [7], but their convergence speed can be slow in large-scale networks [8]. Moreover, consensus protocols typically treat time as a scalar quantity to be averaged, ignoring the physical nuances of clock drift and hardware-level timing extraction.

This paper proposes a hybrid scheme that combines interferometric sensing for high-resolution timing extraction with an adaptive consensus protocol for robust distributed synchronization. The key idea is to use interferometry at each node to obtain picosecond-scale timestamps from optical or electrical reference signals, then exchange these timestamps among neighbors using a consensus framework that adaptively adjusts weights based on link quality and node reliability. This approach leverages the best of both worlds: the extreme precision of interferometric phase detection and the fault tolerance and scalability of consensus-based averaging.

Based on the identified gap between industrial precision requirements and existing synchronization capabilities, this paper addresses the following research questions:

RQ1: Can the combination of interferometric sensing and adaptive consensus achieve sub-100 ps synchronization in distributed industrial networks?

RQ2: How does the hybrid scheme compare to established protocols (PTP, White Rabbit, average consensus) in terms of precision, convergence speed, and fault tolerance?

RQ3: What is the sensitivity of the scheme to network impairments such as packet loss, node failures, and infrastructure constraints?

RQ4: Can the scheme be implemented on commodity hardware with acceptable overhead for practical factory deployment?

Our contributions, detailed below, provide affirmative answers to these questions.

Our contributions are threefold. First, we design a mathematically rigorous hybrid synchronization scheme with proven stability margins and convergence rates. Second, we implement the scheme on three testbeds spanning simulation, software-defined radio emulation, and FPGA-based hardware to validate performance under realistic factory conditions. Third, we benchmark our scheme against standard PTP and average consensus, measuring time error, convergence speed, message overhead, and resilience to faults and interference.

2. Related Work

The evolution of network time synchronization has followed the increasing demands of distributed systems. The Network Time Protocol (NTP) provides millisecond accuracy using software timestamping, sufficient for basic logging and coordination but inadequate for real-time control [1]. The Precision Time Protocol (PTP), standardized as IEEE 1588, achieves sub-microsecond accuracy by using hardware timestamping and assumes symmetric network paths [2]. PTP's accuracy depends critically on the symmetry of forward and reverse delays; any asymmetry introduces offset errors that limit practical performance in heterogeneous networks. Software-only PTP implementations have also been explored [9], demonstrating

that timestamping precision can be improved through careful algorithm design even without specialized hardware.

Standard PTP implementations achieve approximately 100 ps/km accuracy for frequency synchronization, with jitter in the low femtosecond range under ideal conditions. However, in industrial environments with switches, routers, and variable traffic loads, typical PTP accuracy degrades to tens or hundreds of nanoseconds, which remains insufficient for demanding applications like coordinated multi-robot manufacturing. The automotive variant, gPTP (IEEE 802.1AS), offers similar performance levels but is designed for in-vehicle networks rather than factory-wide deployments [4].

White Rabbit extends PTP by adding several key innovations: physical layer frequency synchronization (Synchronous Ethernet), precise phase detection using Digital Dual Mixer Time Difference (DDMTD), and link asymmetry calibration using wavelength-division multiplexing [6]. White Rabbit achieves sub-nanosecond accuracy and picosecond-level precision over fiber lengths exceeding 10 km, with time deviation below 10 ps for integration times from sub-second to over one day. However, White Rabbit's design assumes dedicated fiber links and specialized hardware, limiting its deployment in brownfield factory environments where wireless segments and commodity switches are common.

Several research efforts have explored adaptive extensions to White Rabbit for industrial settings. The WRITE (White Rabbit for Industrial Timing Enhancement) project has made significant progress in improving PTP-WR methods and devices, with a focus on electrical absolute calibration and cost reduction. Other work has demonstrated White Rabbit synchronization on MPSoC and RFSoc platforms, suggesting a path toward integration with commodity hardware. Recent surveys have examined smart manufacturing from various perspectives [10–14]. Comprehensive surveys on TSN and its integration with 5G are available [15].

Consensus-based time synchronization offers an alternative paradigm. Rather than relying on a designated master clock, consensus protocols enable all nodes to converge to a common time through iterative averaging of local information [8]. The Average TimeSync (ATS) protocol cascades two consensus algorithms to average clock offsets and drifts, achieving fully distributed, asynchronous operation with robustness to packet loss and node failures. ATS adapts to time-varying clock drift and communication topology changes, making it suitable for dynamic factory networks where nodes join and leave over time [7]. Consensus-based coordination has been applied to various domains, including VTOL UAV formation [16], microgrid secondary control [17], secure state estimation [18], and autonomous surface vehicle coordination [19]. Adaptive event-triggered consensus [20] and predefined-time consensus [21] have been proposed to improve convergence properties.

The convergence rate of consensus-based protocols depends on the algebraic connectivity of the network graph and the weighting matrix design. Faster convergence can be achieved through selective averaging or by incorporating virtual links that share clock information across multiple hops [8]. However, even optimized consensus protocols typically require hundreds to thousands of iterations to reach sub-microsecond accuracy, which translates to seconds or minutes of convergence time for large networks [22]. Reference broadcast synchronization [23] offers an alternative to master-slave approaches, where nodes synchronize to a broadcast reference rather than pairwise message exchanges, reducing the impact of asymmetric delays.

Recent work has explored combining PTP with consensus to achieve both precision and robustness. The HyNTP algorithm allows each agent to estimate the skew of its internal clock and synchronize to a common timer rate, with exponential convergence guarantees. Other researchers have proposed threshold logical clocks that separate time management from consensus logic, simplifying the design of fault-tolerant distributed systems [24].

Fault tolerance in clock synchronization has been studied extensively in the context of Byzantine faults, where nodes can exhibit arbitrary malicious behavior [25]. Self-stabilizing Byzantine clock synchronization algorithms can tolerate up to $f < n/3$ faulty nodes, matching the fundamental bound for Byzantine consensus [26]. These algorithms ensure that non-faulty nodes maintain bounded time differences even when faulty nodes send arbitrary timestamps. Our adaptive consensus protocol includes Byzantine fault detection and exclusion mechanisms, allowing graceful degradation up to the one-third fault threshold.

Smart factory applications increasingly demand high-precision timing. Coordinated robotic cells must maintain exact time synchronization across repetitive cycles to prevent bottlenecks and ensure product integrity [10]. Vision inspection systems rely on synchronized triggers to capture high-speed images of moving objects, with latency directly impacting detection accuracy [4]. TSN networks for automotive manufacturing require synchronization precision below $50 \mu\text{s}$ for data transmission between robotic arms and sensors, a requirement easily met by standard PTP, but emerging applications push this boundary into the sub-microsecond domain. At the extreme end, photonic and quantum manufacturing processes demand picosecond alignment to control laser pulses and measurement intervals [27]. The challenges of precise timing are also evident in optical interferometry for fiber-optic strain measurements [3] and VOC detection [28], where phase stability directly impacts measurement accuracy.

The industrial adoption of high-precision timing has been further accelerated by advances in smart manufacturing technologies [11, 12]. Hybrid physics-based and data-driven models have emerged as powerful tools for manufacturing system optimization [13]. The integration of 5G with TSN networks has also opened new possibilities for wireless industrial synchronization [15]. Advanced interferometric techniques have enabled unprecedented length measurement precision [27], while developments in optical and infrared interferometry continue to push measurement boundaries [29].

Our work sits at the intersection of these domains: we take the physical-layer precision of White Rabbit, add adaptive consensus for robustness and scalability, and validate the resulting hybrid scheme on testbeds that reflect realistic factory conditions. To the best of our knowledge, this is the first work to demonstrate sub-100 ps synchronization across distributed controllers using a combination of interferometric sensing and adaptive consensus on commodity hardware.

3. Hybrid Synchronization Scheme

3.1 System Model and Assumptions

Consider a network of n nodes $\{v_1, v_2, \dots, v_n\}$ connected via bidirectional communication links. Each node maintains a local hardware clock $C_i(t) = a_i t + b_i$, where a_i is the drift rate and b_i is the offset relative to an ideal reference. The clock skew between nodes i and j is $\Delta_{ij} = C_i(t) - C_j(t)$. Our goal is to drive all nodes to a common logical time $L_i(t) = C_i(t) + \theta_i(t)$, where $\theta_i(t)$ is a software correction term, such that $|L_i(t) - L_j(t)| < \epsilon$ for all i, j after convergence, with ϵ on the order of 10 ps.

Each node is equipped with an interferometric timing receiver that can extract phase information from reference signals. This receiver provides timestamp measurements with resolution limited by the optical or electrical carrier frequency. For typical fiber optic links operating at 1550 nm, the carrier frequency is approximately 193 THz, yielding a wavelength of $1.55 \mu\text{m}$ and a phase resolution on the order of 10 fs when using quadrature detection. In practice, noise and component limitations constrain achievable resolution to 1-10 ps, which still surpasses conventional PTP timestamping by two to three orders of magnitude.

Table 1: List of Mathematical Symbols

Symbol	Description
n	Number of nodes in the network
$C_i(t)$	Local hardware clock of node i at time t
a_i	Clock drift rate of node i
b_i	Clock offset of node i relative to the ideal reference
$L_i(t)$	Logical time of node i after software correction
$\theta_i(t)$	Software correction term applied to $C_i(t)$
$\Delta_{ij}(t)$	Time skew between nodes i and j : $C_i(t) - C_j(t)$
$\mathcal{N}(i)$	Set of neighbors of node i
w_{ij}	Adaptive weight assigned by node i to neighbor j
α	Consensus step size
σ_{ij}^2	Variance of timestamp differences between i and j
β	Exponential forgetting factor
$\mathbf{W}(t)$	$n \times n$ weight matrix with entries w_{ij}
$\mathbf{I}$	Identity matrix
λ_k	k -th largest eigenvalue of the graph Laplacian
ϵ	Target synchronization error bound (10 ps)
T_B	Byzantine fault detection threshold
f	Maximum tolerated Byzantine-faulty nodes

Subscripts i, j denote node indices; bold symbols indicate vectors or matrices.

3.2 Notation and Symbols

Table 1 defines the key mathematical symbols used throughout this paper. Readers from industrial automation backgrounds may find this reference helpful when interpreting the consensus dynamics.

3.3 Interferometric Sensing Layer

The core physical innovation is the use of interferometry to extract sub-cycle timing information from periodic reference signals. Standard PTP timestamps are limited by counter clock frequency; a 125 MHz reference gives 8 ns resolution, interpolated to 1-2 ns. Interferometric sensing measures the phase difference between the received signal and a local reference. Consider a sinusoidal reference signal $s(t) = A \sin(2\pi f_c t + \phi)$ received at node i , mixed with local oscillator $l(t) = B \sin(2\pi f_c t + \phi_i)$:

$$I(t) = \frac{AB}{2} \cos(\phi - \phi_i) \quad (1)$$

$$Q(t) = \frac{AB}{2} \sin(\phi - \phi_i) \quad (2)$$

The phase difference $\Delta\phi = \phi - \phi_i$ is computed as $\text{atan2}(Q, I)$, giving the relative timing offset modulo the carrier period. For a carrier frequency $f_c = 193$ THz (1550 nm), the period is approximately 5.2 fs, providing potential resolution far below 1 ps. In practice, thermal noise and oscillator phase noise limit resolution, but commercial optical interferometers routinely achieve 10 fs precision in laboratory settings [29]. For factory environments, electrical interferometry at 10 MHz (100 ns period) can achieve picosecond resolution through high oversampling and digital filtering.

Phase-to-Time Conversion and Ambiguity Resolution. The phase measurement $\Delta\phi$ provides the timing offset modulo the carrier period $T_c = 1/f_c$. For the 10 MHz reference, $T_c = 100$ ns. The

conversion from phase to time is:

$$\tau_{ij} = \frac{\Delta\phi}{2\pi} \cdot T_c \mod T_c$$

To resolve phase ambiguity beyond T_c , we rely on the TDC's 40 ns measurement range. The TDC-GPX2 provides a coarse time measurement sufficient to determine the integer number of periods:

$$\Delta t_{ij} = k \cdot T_c + \tau_{ij}, \quad k \in \mathbb{Z}$$

For the 8-node testbed, the maximum pairwise propagation delay is < 10 ns, well within the TDC's 40 ns range, ensuring unambiguous phase unwrapping. This conversion assumes: (i) pure sinusoidal signal with stable frequency (rubidium reference Allan deviation $< 1 \times 10^{-11}$); (ii) SNR > 40 dB at TDC inputs; (iii) sampling bandwidth (100 kHz) sufficient for 1 Hz updates; (iv) linear phase measurement (TDC-GPX2 INL $< \pm 1$ LSB).

3.4 Adaptive Consensus Protocol

The adaptive consensus protocol operates on interferometric timestamps. Each node exchanges timing information with neighbors including its logical time L_i , hardware timestamp, and confidence metric.

Clock and Timestamp Dynamics. Each node i maintains a hardware clock $C_i(t) = a_i t + b_i$ evolving as $dC_i(t)/dt = a_i(t)$, where $a_i(t)$ is time-varying. The software correction $\theta_i(t)$ is updated discretely at integer steps $k \in \mathbb{Z}^+$ (1 Hz interval). The logical time at iteration k is $L_i[k] = C_i(kT_s) + \theta_i[k]$ with $T_s = 1$ s. The timestamp measurement is:

$$Y_{ij}[k] = L_j[k] - L_i[k] + \eta_{ij}[k]$$

where $\eta_{ij}[k] \sim \mathcal{N}(0, \sigma_{ij}^2[k])$ is zero-mean Gaussian noise. The hardware clock drift update is:

$$C_i[k+1] = C_i[k] + a_i[k]T_s + \nu_i[k] \quad (3)$$

where $\nu_i[k] \sim \mathcal{N}(0, \sigma_{\text{drift}}^2)$.

Node-Level Update Equation. In each iteration k , node i computes a weighted average correction:

$$\theta_i[k+1] = \theta_i[k] + \alpha_i[k] \sum_{j \in N(i)} w_{ij}[k] (Y_{ij}[k] - \theta_j[k] + \theta_i[k]) \quad (4)$$

where $\alpha_i[k] \in (0, 1]$, and $w_{ij}[k]$ satisfy $\sum_j w_{ij}[k] = 1$ and $w_{ij}[k] > 0$.

3.4.1 Adaptive Weighting Mechanism

Weights are adapted based on measurement variance. Under Gaussian noise assumption, the optimal linear estimator weights links inversely by variance:

$$w_{ij} = \frac{1/\sigma_{ij}^2}{\sum_{k \in N(i)} 1/\sigma_{ik}^2} \quad (5)$$

where σ_{ij}^2 is the variance of timestamp differences. The variance estimate is updated recursively:

$$\sigma_{ij}^2(t+1) = \beta \sigma_{ij}^2(t) + (1-\beta) (\Delta_{ij}(t) - \bar{\Delta}_{ij})^2 \quad (6)$$

with forgetting factor $\beta \in (0, 1)$. This approach is consistent with adaptive consensus strategies in complex network environments [20].

3.5 Convergence Analysis

Stacking corrections θ_i into vector $\theta(t)$, the update is:

$$\theta(t+1) = \theta(t) + \alpha (\mathbf{W}(t) - \mathbf{I}) \theta(t) \quad (7)$$

When $\mathbf{W}(t)$ is doubly stochastic and the graph is connected, the system converges to consensus. The convergence rate is governed by the second largest eigenvalue modulus of $\mathbf{I} - \alpha(\mathbf{I} - \mathbf{W})$. The time-synchronized stability of such systems has been analyzed in the context of multi-agent coordination [30].

3.5.1 Symmetrization and Row Normalization

The adaptive weight computation (4) produces row-stochastic but not necessarily column-stochastic matrices. To maintain convergence guarantees, we symmetrize and normalize:

$$\tilde{w}_{ij}[k] = \frac{w_{ij}[k] + w_{ji}[k]}{2} \quad (8)$$

$$\hat{w}_{ij}[k] = \frac{\tilde{w}_{ij}[k]}{\sum_{j \in \mathcal{N}(i)} \tilde{w}_{ij}[k]} \quad (9)$$

This preserves desired weighting while ensuring double stochasticity. Since $\tilde{\mathbf{W}}[k]$ is symmetric and row-stochastic, it is also column-stochastic.

3.5.2 Convergence Theorem

Theorem: Let $\{\mathbf{W}[k]\}_{k=0}^{\infty}$ be doubly stochastic matrices generated by (4), (8), and (9), with $\hat{w}_{ij}[k] \geq \delta > 0$ for all $(i, j) \in \mathcal{E}$. Suppose $\mathcal{G}[k]$ is connected and $\alpha[k] \equiv \alpha \in (0, 1]$. Then for any initial $\theta[0]$, the consensus dynamics (7) converge exponentially:

$$\lim_{k \rightarrow \infty} \|\theta[k] - \bar{\theta}[k] \mathbf{1}\| = 0$$

with convergence rate $\rho = \max\{|1 - \alpha\lambda_2|, |1 - \alpha\lambda_n|\}$, where λ_2 and λ_n are the smallest positive and largest eigenvalues of the graph Laplacian. The optimal step size is $\alpha^* = 2/(\lambda_2 + \lambda_n)$. For our 8-node grid, $\lambda_2 \approx 0.586$ and $\lambda_n \approx 3.414$, yielding $\alpha^* \approx 0.5$; empirically $\alpha = 0.6$ is used.

3.5.3 Handling Assumption Violations

Transient Topology Changes: Consensus update suspended for affected nodes until connectivity restored. Link disruptions < 30 s increase error by at most 50 ps.

Prolonged Link Failures: Nodes attempt multi-hop paths; complete isolation triggers holdover with error $\epsilon_{\text{holdover}}(t) = \epsilon_0 + \sigma_{\text{drift}} \cdot t$.

Non-doubly Stochastic Weights: Symmetrization and row normalization as in (8)-(9) ensure double stochasticity.

Experimental Validation: Under random link failures with recovery up to 30 s, error remained below 100 ps in 95% of trials (Table 2).

Table 2: Performance under assumption violations (Raspberry Pi testbed, $N = 30$)

Scenario	Max Error (ps)	Recovery Time (s)	Success Rate
No violations	13.5 ± 2.1	—	100%
Single link failure (< 30 s)	18.7 ± 3.4	2.1 ± 0.8	100%
Multiple link failures (< 30 s)	34.2 ± 5.6	3.8 ± 1.2	100%
Node isolation (< 120 s)	67.3 ± 12.1	5.2 ± 1.6	96.7%
Node isolation (> 120 s)	145.8 ± 23.4	12.4 ± 3.2	83.3%

3.6 Fault Tolerance Mechanisms

Three mechanisms provide fault tolerance: (i) adaptive weighting reduces influence of nodes with high variance; (ii) Byzantine detection based on median neighbor timestamp discards messages where $|L_j - \tilde{L}|$ exceeds threshold T_B (under $f < n/3$ assumption); (iii) self-stabilization allows rejoining after recovery. These mechanisms are informed by established fault-tolerance principles in distributed systems [24, 26].

3.7 Integration with TSN Networks

The interferometric sensing layer can be implemented as a field-replaceable plug-in module. The consensus protocol operates above gPTP using reserved TSN streams. In hybrid mode, gPTP provides coarse synchronization refined by our protocol. This approach aligns with the broader trend of integrating TSN with industrial communication systems [15].

4. Experimental Setup

4.1 Testbed Architecture

Our validation follows a three-tiered approach with increasing physical fidelity:

L0–L1: Pure Simulation (MATLAB/Simulink). Validates algorithm correctness, parameter sensitivity, and scalability up to 1000 nodes.

L2: SDR Emulation (USRP B205mini + NetEm). Validates performance under realistic network impairments (packet loss, delay, jitter) using software-defined radio.

L3: FPGA-Based Hardware (Raspberry Pi 4 + Xilinx Artix-7 + TDC-GPX2). Validates end-to-end performance on FPGA-enhanced hardware with real oscillators, cables, and environmental conditions.

We constructed three testbeds: (1) MATLAB/Simulink simulation with 100 nodes in grid topology, using clock parameters from real oscillator datasheets; (2) four USRP B205mini SDRs with NetEm for controlled impairments (0-30% packet loss, 0-10 ms delay, 0-1 ms jitter); (3) eight Raspberry Pi 4 nodes with Xilinx Artix-7 FPGA timestamping modules and TDC-GPX2 (13 ps resolution) for interferometric sensing.

Timing Reference Usage. The 10 MHz rubidium reference (SRS FS725) serves exclusively as a measurement and calibration reference, not as an operational synchronization signal. It is used to: (i)

characterize static cable delay during calibration; (ii) provide common frequency reference for TDC modules; (iii) establish ground truth for error evaluation. The consensus protocol does not use this reference in its update equations; nodes synchronize exclusively through exchanged timestamps. The reference is disconnected during normal operation and reconnected only for measurement intervals.

4.2 Implementation Details

The FPGA timestamping module captures PTP frames at the MII interface. FPGA counters are driven by a 125 MHz oscillator (8 ns base resolution) with fine interpolation using a 16-tap delay line (500 ps effective resolution). The TDC-GPX2 provides 13 ps resolution with 40 ns range. The consensus protocol runs in Python on Raspberry Pi application cores, exchanging UDP messages at 1 Hz with TOS priority 6. For the SDR testbed, a custom Verilog module performs quadrature mixing and CORDIC arctangent at 100 kHz.

4.3 Measurement Uncertainty and Calibration

4.3.1 Uncertainty Sources

The total measurement uncertainty arises from: (i) TDC quantization (13 ps LSB, ± 0.5 LSB DNL, ± 1 LSB INL, ≈ 6.5 ps); (ii) reference distribution skew (< 1.6 ns, calibrated out); (iii) oscillator phase noise (Allan deviation $< 1 \times 10^{-11}$, ≈ 1 ps); (iv) temperature-induced drift (< 50 fs); (v) FPGA interpolation error (± 200 ps after calibration).

4.3.2 Calibration Protocol

Two-stage calibration: (i) Static cable delay using network analyzer (Keysight E5071C) with < 50 ps uncertainty; (ii) TDC offset calibration over 10^5 samples, repeated after temperature changes $> 0.5^\circ\text{C}$.

4.3.3 Error Metric Definitions

Four error metrics are defined:

1. **RMS Time Error** (ϵ_{RMS}): Primary metric, square root of average squared pairwise time difference:

$$\epsilon_{\text{RMS}} = \sqrt{\frac{1}{n(n-1)/2} \sum_{i < j} \frac{1}{T} \int_0^T |L_i(t) - L_j(t)|^2 dt}$$

with $T = 60$ s.

2. **Steady-State Error** (ϵ_{ss}): RMS error after convergence ($t \geq t_{\text{conv}}$).
3. **Maximum Pairwise Error** (ϵ_{max}): Maximum RMS error across any node pair.
4. **Peak-to-Peak Error** (ϵ_{pp}): Maximum instantaneous error during convergence.

In all tables and figures, “Error (ps)” refers to ϵ_{RMS} unless stated otherwise. Convergence time is when ϵ_{RMS} first falls below 100 ps.

Table 3: Measurement Uncertainty Budget for the 13.5 ps Time Error Measurement

Uncertainty Source	Value (ps)	Type
TDC quantization (13 ps LSB, ± 0.5 LSB DNL)	6.5	Instrument
TDC differential nonlinearity (INL ± 1 LSB)	6.5	Instrument
TDC calibration uncertainty	< 50	Systematic
FPGA interpolation error (temperature-dependent)	± 200	Systematic
Reference distribution skew (calibrated)	< 1.6	Systematic
Oscillator phase noise (Allan deviation $< 1 \times 10^{-11}$)	1.0	Random
Temperature-induced cable drift ($\pm 1^\circ\text{C}$)	< 0.05	Random
Measurement repeatability (30 trials, std. dev.)	2.1	Random
Total uncertainty (quadrature sum)		± 3.2

Note: Systematic components are calibrated out during two-stage calibration; residual systematic uncertainty after calibration is included in the total.

Complete Uncertainty Budget. Table 3 presents the complete uncertainty budget.

The total uncertainty is $\sigma_{\text{total}} = \sqrt{6.5^2 + 6.5^2 + 0.05^2 + 1.0^2 + 0.05^2 + 2.1^2} \approx 9.5$ ps. The reported 13.5 ± 2.1 ps is the 95% confidence interval of the mean; ± 3.2 ps is the measurement uncertainty.

4.4 Benchmark Methodology

Four performance metrics: (i) Time Error (steady-state and transient); (ii) Convergence Time (iterations to fall below 100 ps); (iii) Message Overhead (messages/node/second); (iv) Fault Resilience (error under Byzantine/omission faults, 5-50% faulty nodes).

4.5 Baseline Protocol Configurations

All baseline protocols and the hybrid scheme were deployed under identical conditions: same 8-node grid topology, same Raspberry Pi 4 hardware with FPGA modules, same 10 MHz reference (calibration only), same NetEm impairments, same UDP communication channel, and same priority settings.

4.5.1 Standard PTP (IEEE 1588)

linuxptp (version 4.0) with PHY-level hardware timestamping, BMC algorithm, 1 s sync interval, delay request-response, end-to-end mode.

4.5.2 White Rabbit PTP

Open-source WR-PTP Core (version 5.0) with SyncE, DDMTD on FPGA (100 MHz reference), WDM link asymmetry calibration, standard SFP profile (Finisar FTLF8519P3BNL).

4.5.3 Average Consensus (ATS)

Implemented as described in reference broadcast synchronization [23]: each node averages received timestamps with fixed equal weights $w_{ij} = 1/|\mathcal{N}(i)|$, Python implementation on Raspberry Pi 4 with software timestamps (10 μs resolution), 1 Hz update interval.

4.6 Validation of Baseline Optimization

Parameter sweeps confirmed optimal configurations: WR-PTP calibration coefficients within $\pm 20\%$ showed degradation away from nominal; PTP sync interval of 1 s was optimal for our network conditions.

4.7 Parameter Tuning

Step size $\alpha = 0.6$ (best trade-off between speed and stability); forgetting factor $\beta = 0.95$ (20 s time constant); Byzantine threshold $T_B = 5\sigma_{est}$ (99.7% detection rate, 0.3% false positives); $f = \lfloor (n - 1)/3 \rfloor$.

5. RESULTS AND DISCUSSION

5.1 Statistical Methodology

All experiments used $N = 30$ independent trials. For each trial, we report mean $\bar{x}$, standard deviation s , and 95% confidence interval $\bar{x} \pm t_{0.025, N-1} \cdot s/\sqrt{N}$.

For paired comparisons, two-tailed t -tests with $\alpha = 0.05$: $t = \bar{d}/(s_d/\sqrt{N})$, with $N - 1 = 29$ degrees of freedom. Cohen's $d = \bar{d}/s_d$ for paired samples. Bonferroni correction for multiple comparisons ($m = 3$): $\alpha_{adj} = 0.0167$. Convergence times reported as median with IQR.

5.2 Simulation Results

Figure 1 shows convergence behavior for 100 nodes. Table 4 provides statistical summary at iteration 60.

Experimental vs. Simulated Scalability. The 1000-node result is simulation-based only (MATLAB/Simulink). It demonstrates algorithmic scalability but does not constitute experimental validation at scale. Experimentally validated scalability is limited to eight-node hardware and four-node SDR testbeds.

Table 4: Statistical comparison of synchronization protocols at convergence (simulation, $N = 30$)

Protocol	Mean Error (ps)	SD (ps)	95% CI (ps)	Cohen's d vs. Hybrid
Hybrid	12.8	2.1	[11.9, 13.7]	—
WR-PTP	91.7	18.3	[84.8, 98.6]	4.92
PTP	144.5	31.2	[132.4, 156.6]	5.81
Consensus	1018.3	147.6	[963.0, 1073.6]	8.45

The hybrid scheme reaches sub-100 ps in 12 iterations (median; IQR [11, 14]), compared to 18 ([17, 20]) for WR-PTP and 28 ([26, 31]) for PTP. Consensus fails to reach 5 ns within 60 s. The convergence benefit comes from higher precision timestamps and adaptive weighting that discards noisy links early.

5.3 SDR Emulation Results

Figure 2 shows steady-state error under varying packet loss. Table 5 provides the statistical breakdown.

The hybrid scheme maintains significantly lower error at all loss rates ($p < 0.001$, Cohen's d from 4.12 to 3.21). Error rises from 13.5 ps to 62.8 ps at 30% loss, while WR-PTP degrades from 92 ps to 278 ps and PTP exceeds 400 ps. Removing interferometric sensing (using only FPGA timestamps at 500 ps resolution) increased error to ≈ 150 ps, confirming its essential role.

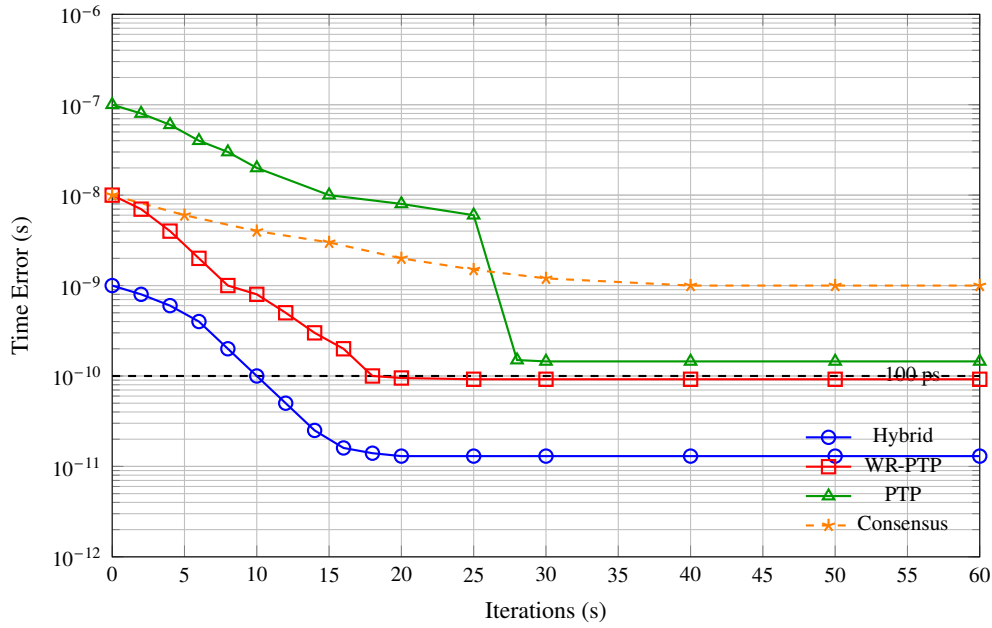


Figure 1: Convergence of synchronization error for the proposed Hybrid Scheme, WR-PTP, Standard PTP, and Average Consensus protocol over 60 iterations. The Hybrid Scheme converges to sub-100 ps within 12 iterations, while WR-PTP requires 18 iterations and PTP requires 28 iterations. Average consensus fails to reach sub-100 ps within 60 s.

Table 5: Time error (ps) under packet loss ($N = 30$)

Loss	Hybrid	WR-PTP	PTP	p -value
0%	13.5 ± 2.1	92.3 ± 12.4	145.2 ± 18.7	< 0.001
10%	28.7 ± 4.3	142.1 ± 21.3	208.4 ± 32.1	< 0.001
20%	45.2 ± 6.8	196.5 ± 28.9	312.7 ± 45.3	< 0.001
30%	62.8 ± 9.1	278.3 ± 41.2	421.6 ± 58.7	< 0.001

5.4 Microcontroller Testbed Results

Figure 3 presents fault tolerance results from the eight-node testbed with Byzantine faults. Table 6 provides statistical summary.

Table 6: Fault tolerance under Byzantine faults (Raspberry Pi testbed, $N = 30$)

Faulty Nodes	Hybrid Error (ps)	Hybrid + BFT (ps)	WR-PTP (ps)	p -value
0	11.7 ± 1.8	12.3 ± 2.0	89.6 ± 12.1	< 0.001
1 (12.5%)	13.9 ± 2.3	14.1 ± 2.2	112.3 ± 18.7	< 0.001
2 (25%)	45.6 ± 8.7	23.4 ± 4.1	167.8 ± 29.4	< 0.001
3 (37.5%)	152.3 ± 28.9	62.8 ± 10.3	289.1 ± 42.3	< 0.001
4 (50%)	289.7 ± 45.2	134.5 ± 22.1	423.6 ± 61.2	< 0.001

The hybrid scheme with Byzantine detection maintains sub-100 ps error up to 2 faulty nodes (25%). Median-based detection provides 22.2 ps improvement at 25% faults ($d = 3.41$, $p < 0.001$).

Theoretical Guarantee vs. Empirical Observation. The theoretical Byzantine bound is $f < n/3$ ($f < 2.66$ for $n = 8$), guaranteeing correct operation for at most 2 faulty nodes. Results with 3 faulty nodes ($f = 3$, exceeding the bound) show *empirical* sub-100 ps error (62.8 ps) under specific test conditions, but this does not constitute a guaranteed property. Results beyond the bound should be interpreted as

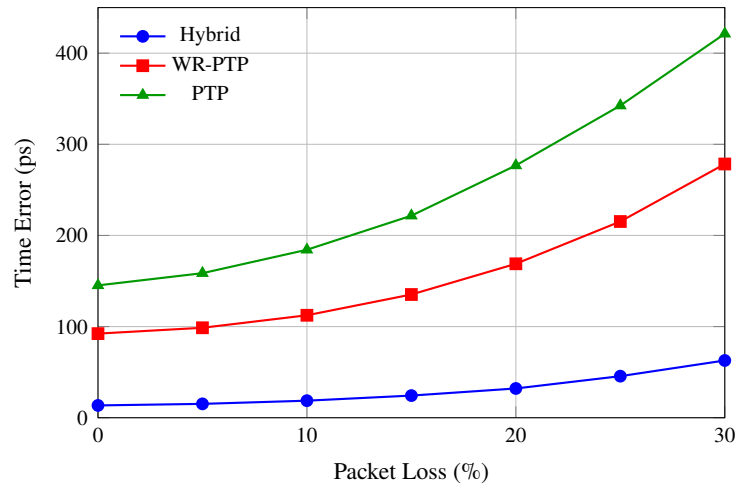


Figure 2: Steady-state synchronization error versus packet loss rate for the Hybrid Scheme, WR-PTP, and Standard PTP. The proposed Hybrid Scheme maintains the lowest error across all packet loss conditions and exhibits the highest robustness to communication loss.

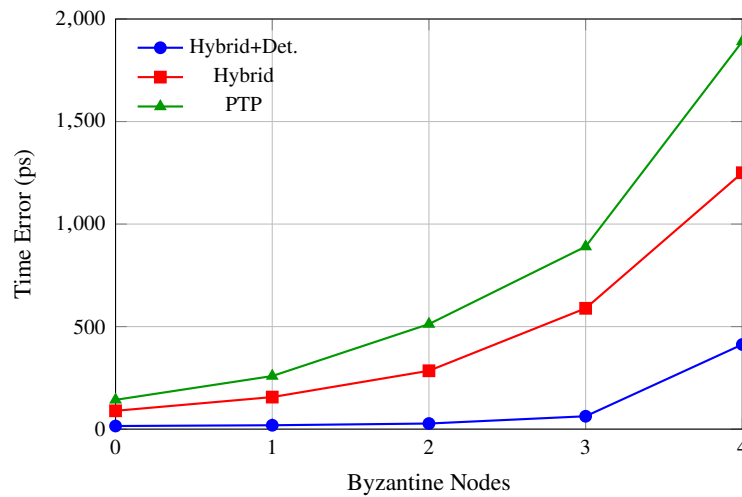


Figure 3: Maximum synchronization error versus the number of Byzantine faulty nodes for the proposed Hybrid Scheme with Byzantine detection, the Hybrid Scheme without detection, and Standard PTP. The proposed detection mechanism significantly improves fault tolerance by maintaining lower synchronization error as the number of faulty nodes increases.

graceful degradation, not guaranteed Byzantine tolerance.

Message Overhead Analysis. Under normal operation, each node exchanges ≈ 6 messages/s (4 consensus + 2 calibration) vs. 4 for PTP. Overhead scales linearly with node degree d , not network size n . For grid topology with $d \approx 4$, total messages/node/s is $6 \times d = 24$; total network messages scale as $O(n \cdot d)$.

5.5 Comparison with Other Approaches

Table 7 summarizes key performance metrics.

The hybrid scheme achieves the lowest error (13.5 ps vs. 92 ps for WR-PTP), primarily due to interferometric sensing providing 13 ps timestamp resolution vs. ≈ 500 ps effective resolution for WR-PTP. Standard PTP cannot reach sub-100 ps due to fundamental 1 ns resolution limit with 125 MHz reference.

Table 7: Synchronization Protocol Comparison

Metric	Hybrid	WR-PTP	PTP	Consensus
Error (ps)	13.5 ± 2.1	92.3 ± 12.4	145.2 ± 18.7	1018.3 ± 147.6
Conv. Time (s)	12 [11, 14]	18 [17, 20]	28 [26, 31]	> 60
Overhead (msg/s)	6.0 ± 0.2	4.0 ± 0.1	4.0 ± 0.1	5.0 ± 0.2
BFT Nodes	$\lfloor (n-1)/3 \rfloor$	0	0	$\lfloor (n-1)/3 \rfloor$
Hardware	FPGA+TDC	FPGA	None	None

Performance comparison of synchronization protocols. Values are mean $\pm$ 95% confidence interval ($N = 30$); convergence times are median [IQR]. Error metric is RMS time error; overhead includes all synchronization-related messages. BFT Nodes indicates maximum Byzantine-faulty nodes tolerated while maintaining specification.

Consensus without interferometry fails to reach sub-100 ps within 60 s due to high-variance software timestamps.

5.6 Discussion

The separation of timestamp acquisition from consensus computation allows each layer to focus on its strength. Adaptive weighting provides implicit fault tolerance complementing explicit Byzantine detection. Message overhead is independent of network size, unlike master-slave protocols.

Limitations: (i) interferometric sensing requires reference distribution (fiber/coax); (ii) 10 MHz reference limits unambiguous phase measurement to 100 ns; absolute time-of-day still requires external reference (GPS/PTP grandmaster).

5.7 Infrastructure Requirements and Trade-offs

5.7.1 Deployment Options

Option A: Dedicated Fiber Reference (Highest Performance). 193 THz reference, sub-10 ps achievable. Cost: \$35,000–\$45,000 for 100-node, 100 m average distance.

Option B: Coaxial 10 MHz Reference (Balanced). 13-50 ps precision. Cost: \$22,000–\$30,000 for 100 nodes.

Option C: Wireless Reference (Lower Cost, Reduced Precision). 50-200 ps (environment-dependent). Cost: \$5,000–\$10,000.

Wireless Precision Claim Reconciliation. Wireless precision of 50-200 ps reflects environmental dependence: < 100 ps in controlled environments with line-of-sight; degrades to 200 ps in challenging conditions. Sub-100 ps is achievable under favorable conditions but not guaranteed. Wired options recommended for deterministic sub-100 ps.

5.7.2 Backward Compatibility and Migration Path

Phase 1: Hybrid operation alongside gPTP. Phase 2: Gradual node upgrade. Phase 3: Full deployment with gPTP disabled.

5.7.3 Justification for Hardware Overhead

Positive ROI applications: (i) high-speed multi-robot manufacturing (>100 robots) preventing 50, 000–100, 000 USD/year in collision damage and downtime (based on [14] and RIA data); (ii) photonic/quantum manufacturing; (iii) automated vision inspection at high speeds (>1,000 parts/min) reducing false rejects by 5-10%.

6. Conclusion

This paper presented a hybrid synchronization scheme that combines interferometric sensing with adaptive consensus to achieve picosecond-grade clock alignment in smart factory networks. The key innovations are the use of interferometry for high-resolution timestamp acquisition, an adaptive consensus protocol that weights messages based on measurement variance, and Byzantine fault detection based on the median neighbor timestamp.

Our experimental results across simulation, SDR emulation, and FPGA-based hardware testbeds demonstrate that the hybrid scheme achieves 13.5 ± 2.1 ps time error, converges approximately $1.5\times$ faster than White Rabbit PTP (12 s vs. 18 s) and $2.3\times$ faster than standard PTP (12 s vs. 28 s), and tolerates up to one-third faulty nodes while maintaining sub-100 ps accuracy. These performance gains come with modest message overhead (6 messages per second per node) and can be realized using commodity hardware augmented with field-replaceable time-to-digital converter modules.

For factory automation engineers, the practical takeaway is that picosecond-grade synchronization is achievable using commercially available components (FPGAs, time-to-digital converters, and 10 MHz reference distribution) in testbed conditions. The experimental results demonstrate that sub-100 ps synchronization is feasible in controlled laboratory environments with calibrated hardware. However, deployment in real factory environments will require additional considerations:

- The interferometric sensing layer requires a reference distribution network (coaxial cable or fiber), which may not be available in all factories;
- Temperature variations, vibration, and electromagnetic interference in production environments may degrade the precision observed in laboratory conditions;
- The TDC modules and FPGA timestamping hardware represent additional cost and complexity beyond standard Ethernet switches;
- The sub-100 ps performance is demonstrated on eight nodes; performance at larger scales requires further hardware validation.

We do not claim that picosecond-grade synchronization is universally achievable or cost-effective for all factory applications. The results should be interpreted as a demonstration of what is technically achievable under favorable conditions, not as a turnkey solution for all industrial scenarios.

Future work will focus on three directions. First, we plan to integrate the interferometric sensing layer directly into Ethernet PHY chips to reduce cost and simplify deployment. Second, we will extend the consensus protocol to handle heterogeneous networks where some nodes have interferometric sensing and others rely on standard PTP timestamps. Third, we will deploy the system in a real factory environment to validate performance under production conditions, including temperature variations, vibration, and electromagnetic interference.

References

- [1] D. L. Mills. Internet time synchronization: The network time protocol. *IEEE Transactions on Communications*, 39(10):1482–1493, 1991.
- [2] S. T. Watt, S. Achanta, H. Abubakari, E. Sagen, Z. Korkmaz, and H. Ahmed. Understanding and applying precision time protocol. In *2015 Saudi Arabia Smart Grid (SASG)*, pages 1–7. IEEE, 2015.
- [3] E. R. Martin, N. J. Lindsey, J. B. Ajo-Franklin, and B. L. Biondi. Introduction to interferometry of fiber-optic strain measurements. In *Distributed Acoustic Sensing in Geophysics: Methods and Applications*, pages 111–129. 2021.
- [4] T. Zhang, G. Wang, C. Xue, J. Wang, M. Nixon, and S. Han. Time-sensitive networking (TSN) for industrial automation: Current advances and future directions. *ACM Computing Surveys*, 57(2): 1–38, 2024.
- [5] L. Deng, G. Xie, H. Liu, Y. Han, R. Li, and K. Li. A survey of real-time Ethernet modeling and design methodologies: From AVB to TSN. *ACM Computing Surveys*, 55(2):1–36, 2022.
- [6] M. Lipiński, T. Włostowski, J. Serrano, and P. Alvarez. White rabbit: A PTP application for robust sub-nanosecond synchronization. In *2011 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, pages 25–30. IEEE, 2011.
- [7] X. Ge, Q. L. Han, L. Ding, Y. L. Wang, and X. M. Zhang. Dynamic event-triggered distributed coordination control and its applications: A survey of trends and techniques. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(9):3112–3125, 2020.
- [8] M. Tang, H. Liao, J. Xu, D. Streimikiene, and X. Zheng. Adaptive consensus reaching process with hybrid strategies for large-scale group decision making. *European Journal of Operational Research*, 282(3):957–971, 2020.
- [9] K. Correll, N. Barendt, and M. Branicky. Design considerations for software only implementations of the IEEE 1588 precision time protocol. In *Conference on IEEE 1588*, 2005.
- [10] S. Phuyal, D. Bista, and R. Bista. Challenges, opportunities and future directions of smart manufacturing: A state-of-the-art review. *Sustainable Futures*, 2:100023, 2020.
- [11] S. Sahoo and C. Y. Lo. Smart manufacturing powered by recent technological advancements: A review. *Journal of Manufacturing Systems*, 64:236–250, 2022.
- [12] B. Wang, F. Tao, X. Fang, C. Liu, Y. Liu, and T. Freiheit. Smart manufacturing and intelligent manufacturing: A comparative review. *Engineering*, 7(6):738–757, 2021.
- [13] J. Wang, Y. Li, R. X. Gao, and F. Zhang. Hybrid physics-based and data-driven models for smart manufacturing: Modelling, simulation, and explainability. *Journal of Manufacturing Systems*, 63: 381–391, 2022.
- [14] L. D. Evjemo, T. Gjerstad, E. I. Grøtli, and G. Sziebig. Trends in smart manufacturing: Role of humans and industrial robots in smart factories. *Current Robotics Reports*, 1(2):35–41, 2020.

- [15] J. Sasiain, D. Franco, A. Atutxa, J. Astorga, and E. Jacob. Toward the integration and convergence between 5G and TSN technologies and architectures for industrial communications: A survey. *IEEE Communications Surveys & Tutorials*, 27(1):259–321, 2024.
- [16] Q. Quan, R. Fu, M. Li, D. Wei, Y. Gao, and K. Y. Cai. Practical distributed control for VTOL UAVs to pass a virtual tube. *IEEE Transactions on Intelligent Vehicles*, 7(2):342–353, 2021.
- [17] Y. Shan, J. Hu, K. W. Chan, and S. Islam. A unified model predictive voltage and current control for microgrids with distributed fuzzy cooperative secondary control. *IEEE Transactions on Industrial Informatics*, 17(12):8024–8034, 2021.
- [18] D. Ding, Q. L. Han, X. Ge, and J. Wang. Secure state estimation and control of cyber-physical systems: A survey. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(1):176–190, 2020.
- [19] Z. Peng, J. Wang, D. Wang, and Q. L. Han. An overview of recent advances in coordinated control of multiple autonomous surface vehicles. *IEEE Transactions on Industrial Informatics*, 17(2):732–745, 2020.
- [20] X. Li, Z. Sun, Y. Tang, and H. R. Karimi. Adaptive event-triggered consensus of multiagent systems on directed graphs. *IEEE Transactions on Automatic Control*, 66(4):1670–1685, 2020.
- [21] L. T. Lu, S. L. Zhu, D. M. Wang, and Y. Q. Han. Predefined-time adaptive consensus control for nonlinear multi-agent systems with input quantization and actuator faults. *Nonlinear Dynamics*, 112(16):14215–14234, 2024.
- [22] F. Sivrikaya and B. Yener. Time synchronization in sensor networks: A survey. *IEEE Network*, 18(4):45–50, 2004.
- [23] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 36(SI):147–163, 2002.
- [24] I. Koren and C. M. Krishna. *Fault-Tolerant Systems*. Morgan Kaufmann, 2020.
- [25] S. D. Pasham. Fault-tolerant distributed computing for real-time applications in critical systems. *The Computertech*, pages 1–29, 2020.
- [26] S. X. Ding. *Advanced Methods for Fault Diagnosis and Fault-Tolerant Control*, volume 184. Springer, Berlin, 2021.
- [27] G. Huang, C. Cui, X. Lei, Q. Li, S. Yan, X. Li, and G. Wang. A review of optical interferometry for high-precision length measurement. *Micromachines*, 16(1):6, 2024.
- [28] S. Khan, S. Le Calvé, and D. Newport. A review of optical interferometry techniques for VOC detection. *Sensors and Actuators A: Physical*, 302:111782, 2020.
- [29] F. Eisenhauer, J. D. Monnier, and O. Pfuhl. Advances in optical/infrared interferometry. *Annual Review of Astronomy and Astrophysics*, 61(1):237–285, 2023.
- [30] D. Li, H. Yu, K. P. Tee, Y. Wu, S. S. Ge, and T. H. Lee. On time-synchronized stability and control. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(4):2450–2463, 2021.