Clock-Integrity, Jitter-Bounded, and Coherence-Monitored Commutation Networks for Multi-Phase Drive Systems

Docket: NTL-PROV-005

Inventors: [Inventor Names]

Assignee: [Assignee / Organization]

Date: February 1, 2026

Abstract

Disclosed are apparatus, systems, methods, and non-transitory computer-readable media for generating, distributing, measuring, and enforcing clock integrity in multi-phase drive systems that synthesize rotating fields or commutate multi-channel actuators. The disclosure provides: (i) a clock distribution architecture with bounded inter-channel skew and jitter; (ii) channel-level time-stamping and time-to-digital conversion (TDC) or equivalent measurement to estimate per-channel phase error; (iii) a coherence metric that compares expected commutation events to observed electrical waveforms (e.g., current and voltage), enabling closed-loop correction and drift compensation; and (iv) fault detection and mitigation including detuning, shutdown, or safe-mode commutation when coherence falls below threshold.

In various embodiments, a multi-phase schedule is represented as an 8-tick kernel repeated over time; the disclosed clock-integrity layer ensures that the physical commutation events remain aligned with the intended tick grid to within a programmable tolerance. The disclosed techniques increase reproducibility, reduce artifact risk, and enable deterministic replay across experimental runs and hardware platforms.

Technical Field

The present disclosure relates to timing integrity in multi-channel drive systems, and more particularly to clock distribution, jitter/skew measurement, coherence monitoring, and closed-loop correction for multi-phase commutation and rotating-field synthesis in electromagnetic and electromechanical devices.

Background

In high-speed multi-phase systems (e.g., commutated coil arrays, brushless motor drives, and phased electromagnetic arrays), system behavior depends not only on the nominal schedule but also on timing integrity. Clock jitter, channel skew, and schedule drift can cause:

- phase errors that destroy intended commutation order;
- increased harmonic content and unintended EMI;
- inconsistent results across hardware builds and runs;
- unsafe overshoot when a controller loses phase lock;
- false positives in sensitive metrology due to timebase/sensor misalignment.

Conventional timing solutions (e.g., a nominal PWM clock) often do not provide end-to-end verification that commutation events occurred when intended, nor do they provide a system-level coherence metric that ties the timing system to measured electrical behavior. Accordingly, there is a need for a clock-integrity framework that provides both (i) bounded timing distribution and (ii) measured coherence between intended schedules and observed waveforms.

Summary

This disclosure provides an end-to-end clock-integrity and coherence-monitoring stack for multi-phase commutation networks.

In one aspect, a system distributes a reference clock to C drive channels with bounded skew and jitter, optionally using a PLL, DLL, or digitally controlled delay (DCD) per channel.

In another aspect, the system measures actual commutation event times (e.g., driver edge times, sensed current zero-crossings, or gate-drive transitions) and computes per-channel phase error relative to expected tick times.

In another aspect, the system computes a coherence score that quantifies alignment between intended commutation events and observed waveforms; the score is used for closed-loop correction and for fault gating.

In another aspect, the system provides deterministic replay by storing a canonical schedule representation together with clock calibration parameters and measured jitter/skew statistics.

Brief Description of the Drawings

Drawings may be provided later. For purposes of this specification:

- **FIG. 1** depicts a reference clock distributed to multiple channels, including per-channel delay adjustment and measurement.
- **FIG. 2** depicts definitions of jitter, skew, and phase error relative to a tick grid.
- **FIG. 3** depicts a time-to-digital converter (TDC) measuring edge timestamps and computing phase errors.
- **FIG. 4** depicts a coherence score computed from expected event times and observed current/voltage waveforms.
- **FIG. 5** depicts a closed-loop correction system that updates channel delays and/or schedule phase to maintain lock.
- **FIG. 6** depicts fault gating and safe-mode behaviors triggered by low coherence or excessive jitter.
- **FIG. 7** depicts deterministic replay packaging including schedule, calibration, and integrity hashes.

Definitions and Notation

Unless otherwise indicated:

- $C \in \mathbb{N}$ is the number of drive channels.

- $t \in \mathbb{N}$ is a discrete tick index.
- $\tau_{\text{nom}} > 0$ is the nominal tick period (seconds).
- $T_{\text{ref}}(t) := t \cdot \tau_{\text{nom}}$ is the ideal time of tick t under the reference clock.
- $T_c(t)$ is the measured (actual) time of a commutation event associated with tick t on channel c.
- The *phase error* (time-domain) for channel c at tick t is:
$$e_c(t) := T_c(t) - T_{\text{ref}}(t).$$
- *Skew* refers to differences in phase error between channels:
$$\Delta_{c,c'}(t) := e_c(t) - e_{c'}(t).$$
- *Jitter* refers to time variation of a channel's phase error over time. One convenient proxy is:
$$J_c(t) := e_c(t) - e_c(t-1).$$
- A *coherence score* is a scalar that quantifies agreement between expected commutation events and observed signals.

Detailed Description

1. System Overview

In one embodiment, a commutation network comprises:

- a reference clock source (e.g., oscillator or recovered clock);
- a clock distribution network delivering timing to C channels;
- per-channel timing adjustment elements (optional) such as delay lines;
- per-channel drivers (power stages) generating physical commutation events;
- measurement circuitry (e.g., TDC, comparators, ADCs) to timestamp events and measure waveforms;
- a controller computing phase errors, coherence scores, and corrective actions.

2. Clock Distribution with Bounded Skew and Jitter

2.1 Reference clock. In one embodiment, the system uses a reference clock with specified phase noise. The reference clock may be temperature-compensated, oven-controlled, or disciplined by an external reference.

2.2 Distribution network. In one embodiment, the reference clock is distributed over:

- matched-length PCB traces,
- differential pairs (e.g., LVDS),
- shielded coax,
- optical links for galvanic isolation.

2.3 Per-channel calibration (delay alignment). In one embodiment, a per-channel delay d_c is applied so that:

$$T_c(t) \approx T_{\text{ref}}(t) + d_c,$$

and the controller chooses d_c to minimize $|e_c(t)|$ and/or to minimize inter-channel skew $|\Delta_{c,c'}(t)|$.

2.4 Acceptance thresholds. In one embodiment, the system enforces thresholds:

- $\max_c |e_c(t)| \leq e_{\max}$ (absolute phase error bound),
- $\max_{c,c'} |\Delta_{c,c'}(t)| \leq \Delta_{\max}$ (skew bound),
- $\text{RMS}(J_c) \leq J_{\max}$ (jitter bound).

Thresholds may be configured per operating regime (e.g., higher frequency requires tighter bounds).

3. Event Time Measurement and Phase Error Estimation

3.1 Event sources (non-limiting). An "event" may refer to:

- a gate-drive rising edge,
- a driver output edge,
- a sensed current threshold crossing,
- a voltage threshold crossing,
- a digital marker produced by the driver stage.

3.2 Time stamping. In one embodiment, each channel provides a timestamp $T_c(t)$ for at least one event per tick. Timestamps may be acquired using:

- a TDC referenced to the master clock,
- a counter capture unit (timer capture),
- a high-speed ADC with interpolated crossing time.

3.3 Phase error estimation. Given timestamps $T_c(t)$ and ideal times $T_{\text{ref}}(t)$, compute:

$$e_c(t) = T_c(t) - T_{\text{ref}}(t).$$

In one embodiment, the controller estimates and subtracts systematic offsets (e.g., fixed driver propagation delay) to isolate variable timing error.

4. Coherence Monitoring

4.1 Motivation. Even if edge timestamps are within tolerance, a driver may fail to produce the intended physical waveform due to load changes, saturation, supply droop, EMI, or component faults. Therefore, timing integrity is strengthened by monitoring coherence between intended commutation and observed waveforms.

4.2 Observed signals. Let $x_c(t)$ be a measured signal per channel such as current magnitude, current sign, voltage, or a demodulated sensor output sampled synchronously with ticks.

4.3 Expected sign sequence (example). In one embodiment, a schedule specifies an expected sign $s_c(t) \in \{-1, +1\}$ for a bipolar drive. The observed sign $\hat{s}_c(t)$ may be inferred from current polarity or comparator outputs.

4.4 Per-channel coherence score (example). Define a coherence score over a window of length N:

$$\mathrm{Coher}_c(t_0; N) := \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{1}\{\hat{s}_c(t_0 + k) = s_c(t_0 + k)\}, \tag{1}$$

where $\mathbf{1}\{\cdot\}$ is an indicator function. This yields a value in $[0, 1]$.

4.5 Correlation-based coherence (example). For real-valued signals, define:

$$\mathrm{Coher}_c(t_0; N) := \frac{\sum_{k=0}^{N-1} x_c(t_0 + k)\, u_c(t_0 + k)}{\sqrt{\sum_{k=0}^{N-1} x_c(t_0 + k)^2} \sqrt{\sum_{k=0}^{N-1} u_c(t_0 + k)^2}}, \tag{2}$$

where $u_c(t)$ is an expected reference waveform derived from the schedule (e.g., ±1 values). This yields a normalized correlation in $[-1, 1]$.

4.6 Aggregate coherence. In one embodiment, the controller computes:

$$\mathrm{Coher}_{\mathrm{agg}} := \min_c \mathrm{Coher}_c \quad \text{or} \quad \mathrm{Coher}_{\mathrm{agg}} := \frac{1}{C} \sum_c \mathrm{Coher}_c,$$

and triggers mitigation if $\mathrm{Coher}_{\mathrm{agg}} < \gamma$ for a threshold $\gamma \in (0, 1)$.

5. Closed-Loop Correction and Drift Compensation

5.1 Delay correction. In one embodiment, the system updates d_c using:

$$d_c \leftarrow d_c - K_e \cdot \overline{e_c},$$

where $\overline{e_c}$ is an averaged phase error and $K_e > 0$ is a gain.

5.2 Schedule phase correction. In one embodiment, the system applies a global phase shift to the schedule (equivalently, shifts the tick alignment) when common-mode error is detected across channels.

5.3 Combined objective. In one embodiment, the controller optimizes a combined objective:

$$\min \left(\alpha \max_c |e_c| + \beta \max_{c,c'} |\Delta_{c,c'}| \right) \quad \text{subject to} \quad \mathrm{Coher}_{\mathrm{agg}} \geq \gamma.$$

6. Fault Detection and Safe-Mode Behaviors

6.1 Timing faults. If $|e_c| > e_{\max}$ or $|\Delta_{c,c'}| > \Delta_{\max}$, the system declares a timing fault.

6.2 Coherence faults. If $\mathrm{Coher}_{\mathrm{agg}} < \gamma$ for a persistence interval, the system declares a coherence fault.

6.3 Mitigation actions. Mitigation actions include (non-limiting):

- detune/phase slip injection (soft stop),
- reduction of drive amplitude or duty,
- switching to a safe neutral kernel,
- full shutdown with latching fault state.

7. Deterministic Replay and Provenance (Optional)

In one embodiment, the system stores a replay bundle containing:

- a canonical schedule representation,
- calibration parameters $\{d_c\}$,
- measured jitter/skew statistics and thresholds,
- hashes and signatures for integrity.

The bundle enables reproduction of timing and coherence behavior across runs.

Claims (Draft)

Note: The following claims are an initial, non-limiting claim set intended to preserve multiple fallback positions. Final claim strategy should be reviewed by counsel.

Independent Claims

1. **(System)** A multi-phase commutation system comprising: a reference clock source; a clock distribution network configured to provide a clock signal to a plurality of drive channels; measurement circuitry configured to obtain timing measurements of commutation events associated with the plurality of drive channels; and a controller configured to compute a phase error based on the timing measurements and to output a corrective action to reduce at least one of clock jitter or inter-channel skew.

2. **(Method)** A method of operating a multi-channel drive system, the method comprising: distributing a reference clock to a plurality of channels; generating commutation events according to a schedule; measuring actual event times for at least one commutation event per channel; computing a phase error between the actual event times and expected event times; computing a coherence score comparing observed electrical behavior to expected behavior; and modifying at least one of a per-channel delay, a schedule phase, or a drive amplitude based on the phase error and the coherence score.

3. **(Non-transitory medium)** A non-transitory computer-readable medium storing instructions that, when executed by one or more processors, cause the one or more processors to: compute per-channel phase errors from captured event timestamps; compute an aggregate coherence score from measured current or voltage waveforms and an expected reference waveform; and trigger a fault mitigation action when at least one of a phase error bound or a coherence threshold is violated.

Dependent Claims (Examples; Non-Limiting)

4. The system of claim 1, wherein the measurement circuitry comprises a time-to-digital converter referenced to the reference clock source.
5. The system of claim 1, wherein the controller is configured to apply a per-channel digitally controlled delay to reduce inter-channel skew.
6. The method of claim 2, wherein measuring actual event times comprises capturing gate-drive edge timestamps.
7. The method of claim 2, wherein computing the coherence score comprises computing a normalized correlation between a measured signal and an expected schedule-derived waveform.
8. The method of claim 2, wherein modifying comprises switching to a safe neutral kernel when the coherence score falls below a threshold.
9. The non-transitory medium of claim 3, wherein the fault mitigation action comprises detuning or phase slip injection.
10. The system of claim 1, wherein the clock distribution network comprises differential signaling over matched-length traces.
11. The system of claim 1, further comprising storing a deterministic replay bundle including the schedule and calibration parameters.

Additional Embodiments and Fallback Positions (Non-Limiting)

- Timing measurements may be obtained from any combination of edge capture, comparator crossing, ADC interpolation, or digital markers.
- Coherence monitoring may use sign agreement, correlation, spectral coherence, lock-in techniques, or combinations thereof.
- Corrective actions may include delay calibration, schedule phase shifting, frequency nudging, amplitude/duty reduction, detuning, or shutdown.
- The system may enforce different timing bounds depending on operating mode or frequency.
- The system may report timing metrics and coherence metrics in real time and store them for forensic analysis.

End of Specification (Draft)