Time and Frequency Measurements from Picosecond-Resolution Time Tags

Edoardo Mornacchi & Mickey Martini, Swabian Instruments

Time & Frequency (T&F)

T&F measurements are central to clock comparison, synchronization monitoring, and stability analysis in metrology, telecommunications, navigation, quantum systems, and fundamental research.

Challenges in T&F Measurements:

- Monitor frequency and phase fluctuations across many timescales.
- Compare several clocks simultaneously.
- Track time offsets with picosecond precision.
- Require specialized low-noise references and dedicated measurement hardware.

Time-Tag-based T&F Analysis

- Frequency-stability, phase-noise, frequency-counter, and 1 PPS signals monitoring from a single time-tag stream.
- Parallel analysis across multiple input channels.
- Merging of synchronized time-tag streams from distributed nodes.
- Software interfaces for Python, MATLAB, LabVIEW, C#, and C++.

Correlating Time Tag Streams from Distributed Nodes

Multiserver-Client Interface

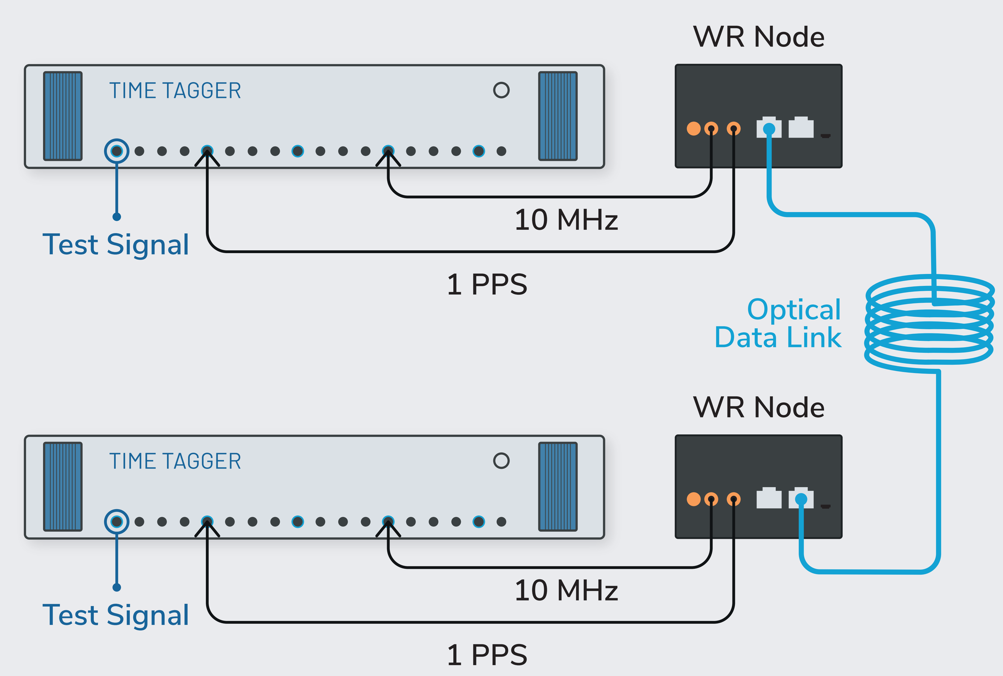
TimeTaggerNetwork merges time-tag streams from distributed nodes for joint processing.

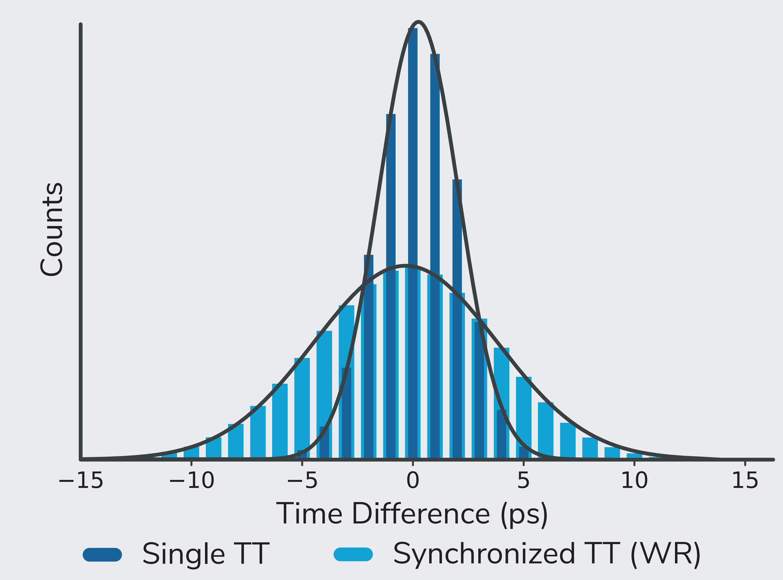
Software Clock

Software-PLL employs frequency and PPS signals for rescaling and aligning the different time bases.

Hardware-Agnostic Synchronization Method

Works with external timing references such as White Rabbit and GNSS-disciplined systems.



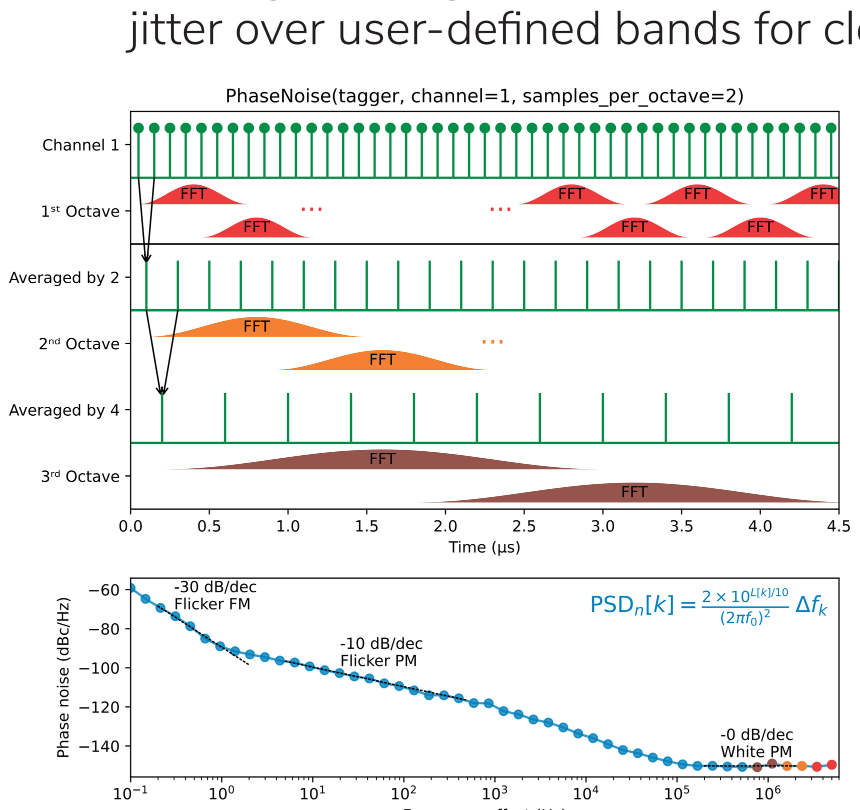


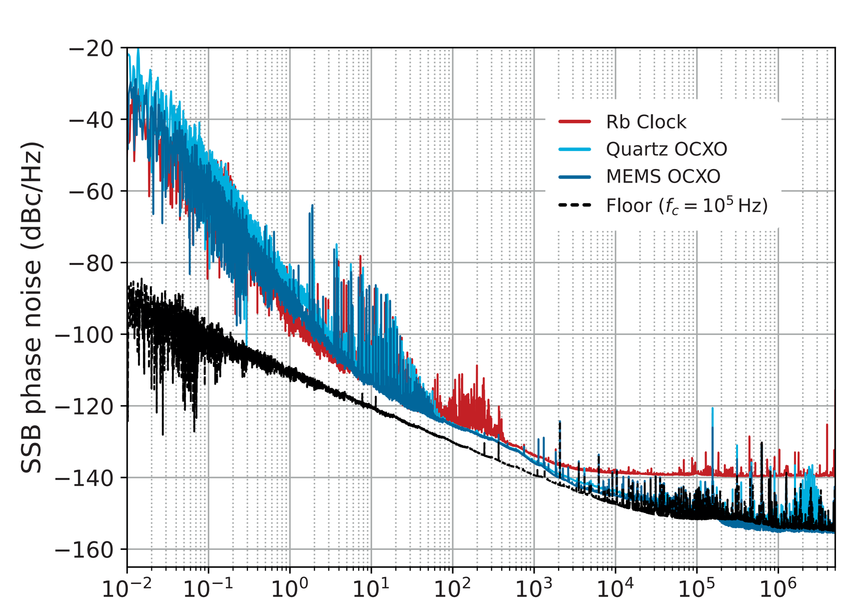
Correlation measurement of a split signal input into two Time Taggers synchronized with White Rabbit (WR) nodes (light blue) vs. same measurement with a single TT (dark blue).

Software Toolkit for T&F Analyses

Time-tag-based T&F analysis can be performed in software from a common acquisition time-tag stream. A software PLL can lock to an external frequency reference. Phase-noise, frequency-stability, frequency-counter, and PPS-monitor analyses run in parallel on the same data.

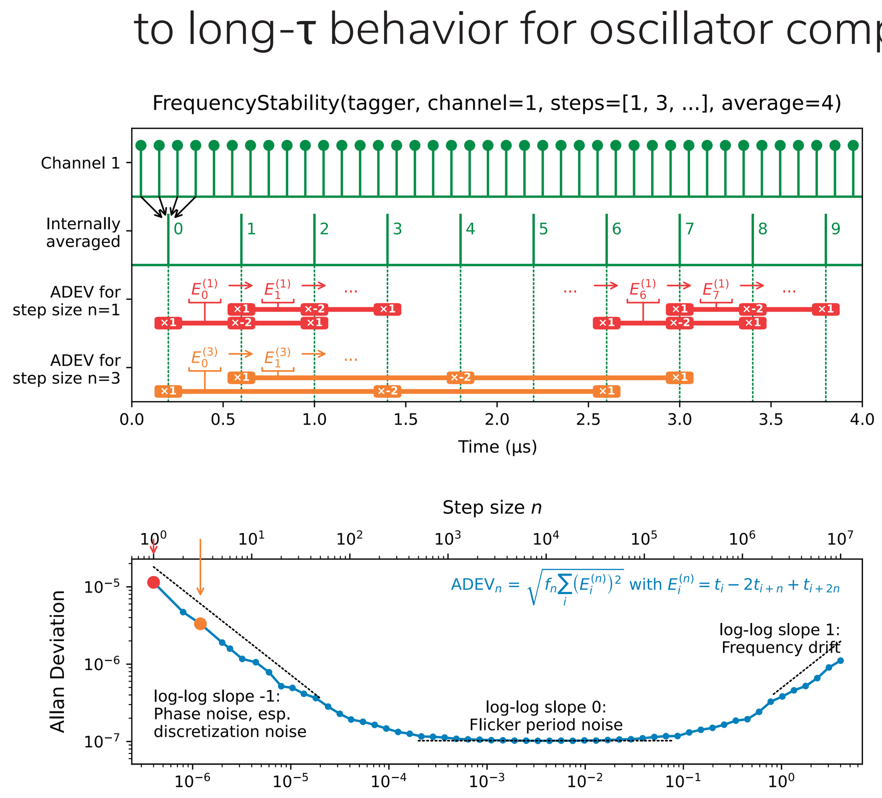
PhaseNoise estimates single-side band (SSB) phase noise on a quasi-log offset grid using the Welch method for PSD estimation and reports integrated jitter over user-defined bands for clock characterization.

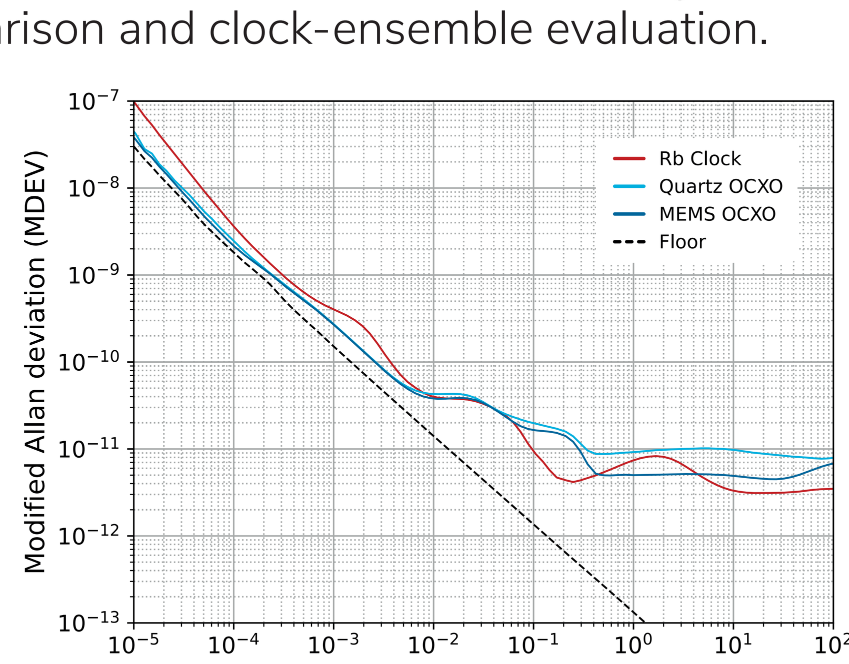




Phase noise spectra of three clocks acquired for one hour. The black line represents the instrument response function.

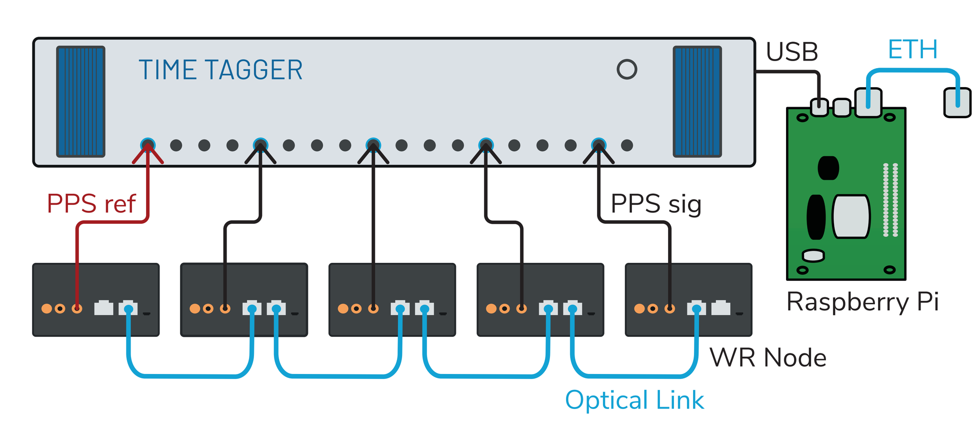
FrequencyStability computes ADEV, MDEV, TDEV, and HDEV, covering short- to long-τ behavior for oscillator comparison and clock-ensemble evaluation.



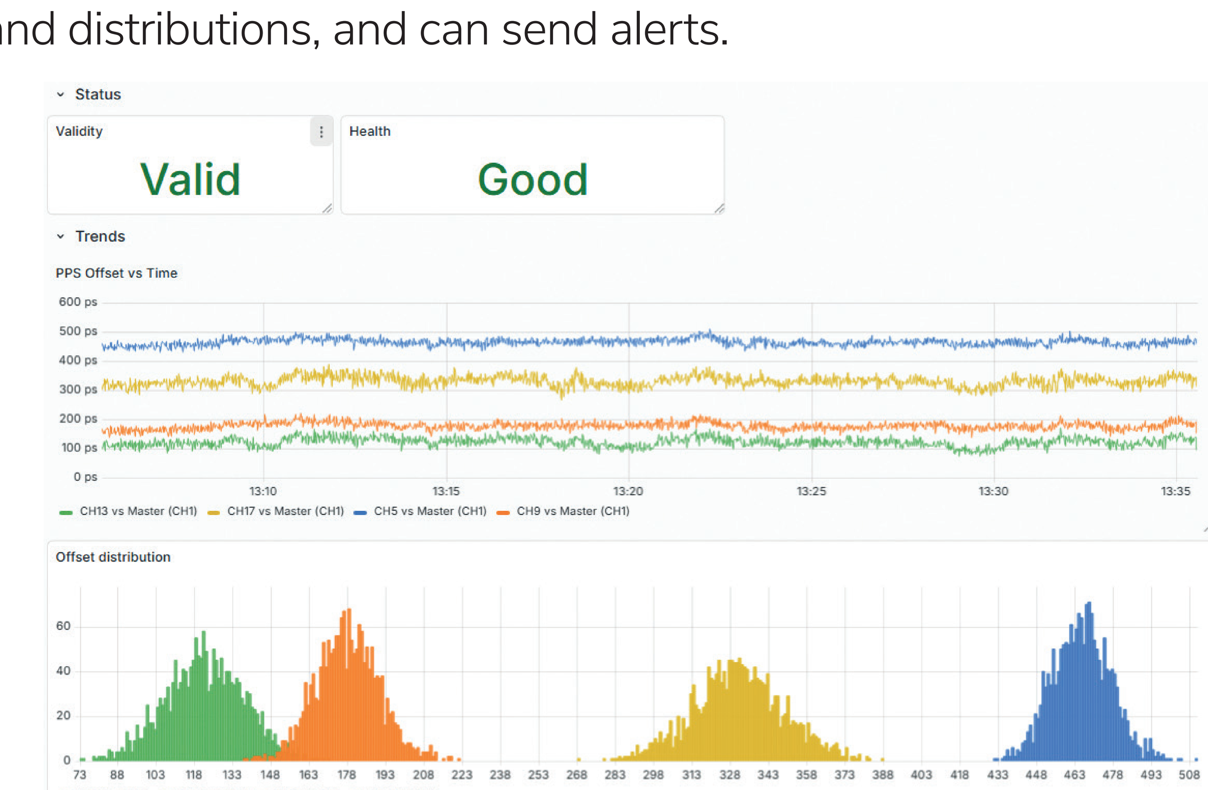


Modified Allan Deviation traces of three clocks acquired for one hour. The black line represents the instrument response function.

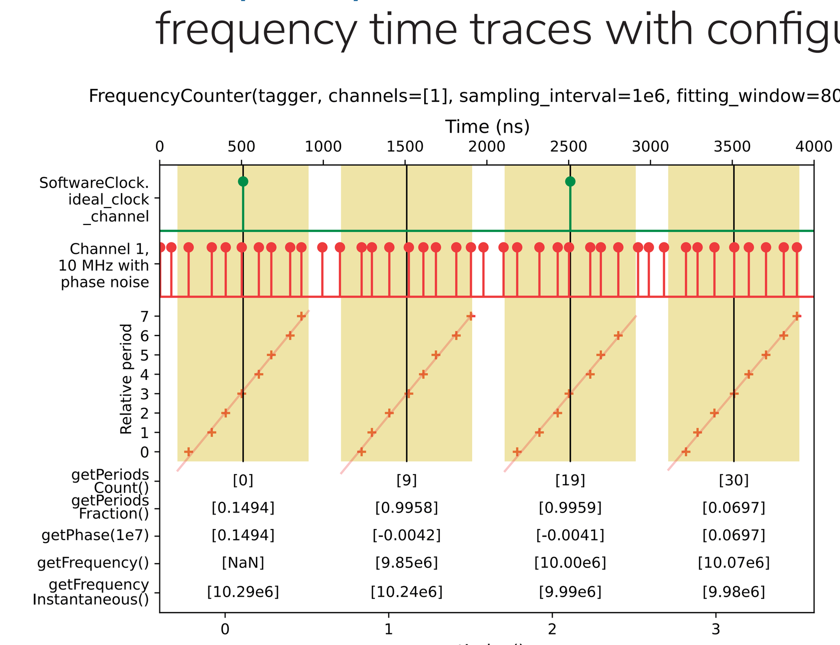
PulsePerSecondMonitor monitors multiple 1 PPS sources against a reference, and records time-offset and jitter to verify synchronization and detect anomalies across sites.

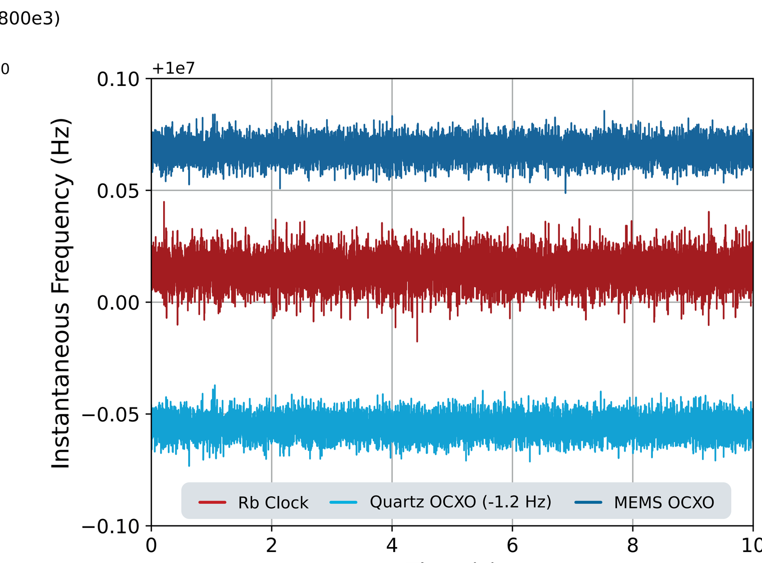


Metrics are streamed to a database. Grafana shows live trends and distributions, and can send alerts.



FrequencyCounter emulates an Ω-type counter to record phase and frequency time traces with configurable sampling and gate/fit window.





Instantaneous frequency traces of three clocks. An offset is applied for better visualization.

Swabian instruments

edoardo@swabianinstruments.com

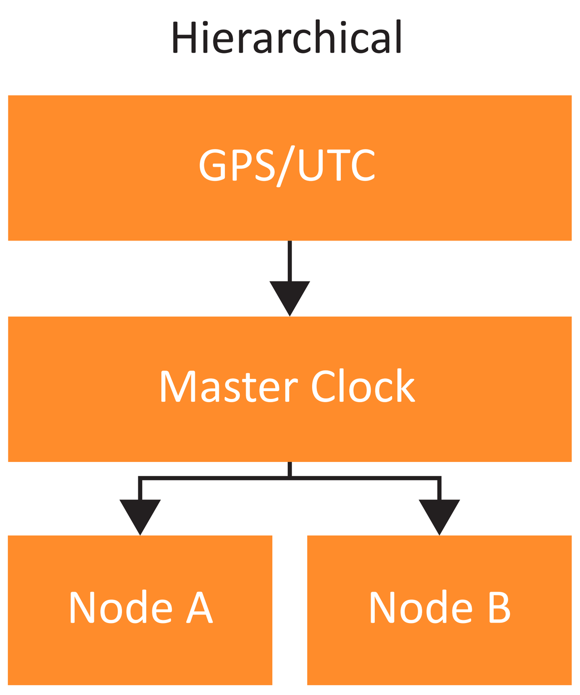
Asynchronous Two-Way Time Transfer (ATWTT)

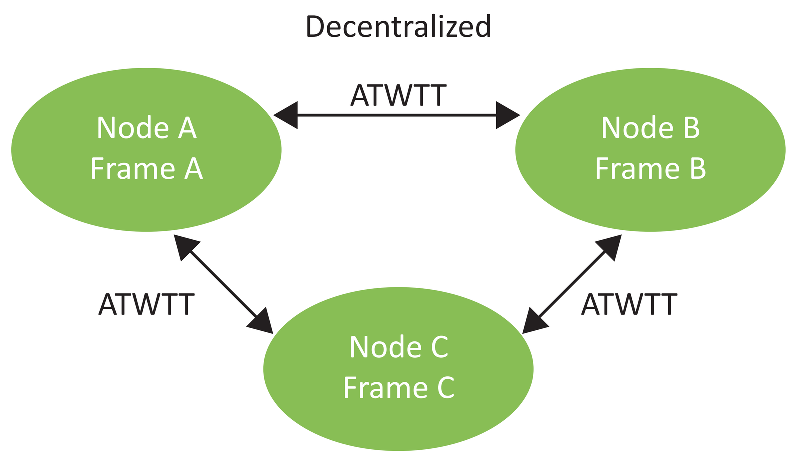
Time Transfer for Decentralized PNT Networks

Evan Alexander | Microchip Technology Inc. Frequency and Time Systems | Workshop on Synchronization and Timing Systems (WSTS) 2026

ATWTT enables asynchronous nodes to exchange PNT information without a shared reference frame. Each node maintains its own local timescale and ATWTT measures the offset between frames and translates timestamps at each hop as time is transferred across the network. A network of nodes sharing PNT data this way can achieve sub 100 picosecond inter-node coherence without GPS, a master clock and without precision distribution infrastructure. This technology is ideal for the strict requirements of phased array radar and coherent sensing systems.

1 – The Problem: Global Reference Frames





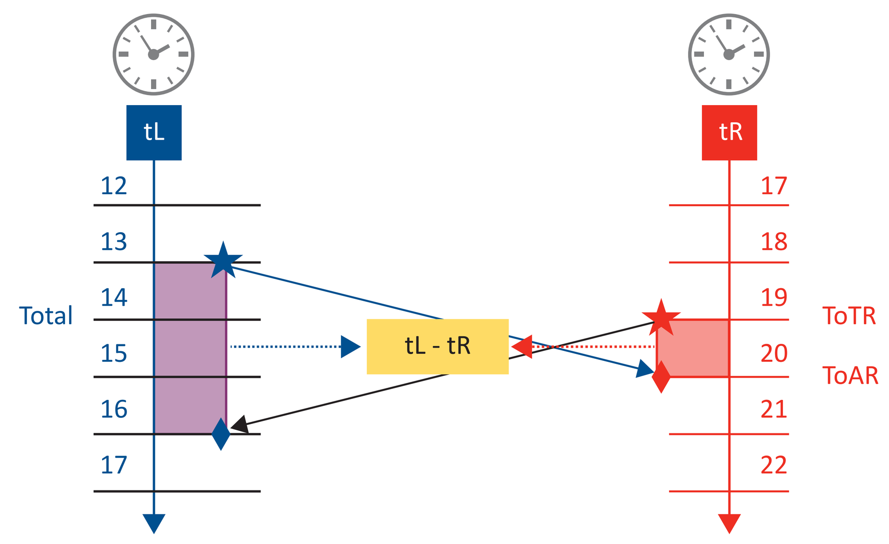
Traditional (Global Coherency)

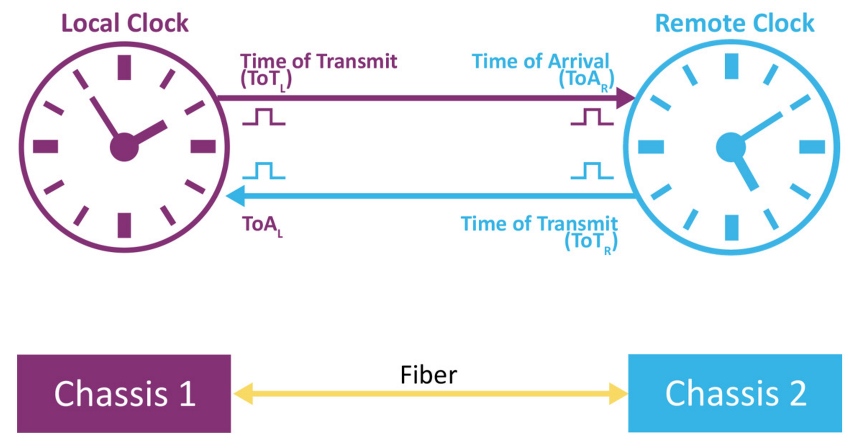
- Requires universal reference frame (10 MHz, 1 PPS, GPS time)
- Hierarchical node organization that is fixed at design time
- All nodes must reference a common authoritative source
- Vulnerable to single-point GPS denial

Decentralized (Local Reference Frames)

- Each node maintains its own local reference frame/timescale
- GPS-capable but not GPS-dependent
- Hierarchy defined at runtime, not design time and can be dynamically reconfigured
- PNT data is translated between reference frames at each hop

2 – How ATWTT Works: Asynchronous Event Pairing



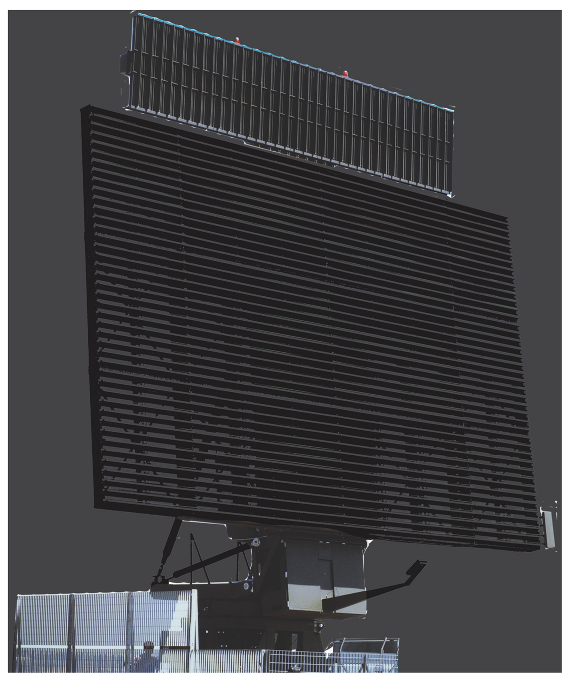


Asynchronous Two-Way Time Transfer

- ATWTT transmits signals between clocks asynchronously and measurement pairing is performed independently by each platform
- A simultaneity equation determines which measurements to pair. Clocks can be arbitrarily offset in time and transmit at different rates
- Insensitive to asymmetric path delay. The mean group delay calculation captures the midpoint of the exchange, which is invariant to how propagation delay is distributed between the two directions.

- Each chassis maintains its own independent local timescale
- Timing data is translated into the receiving chassis' local reference frame at each hop
- Any chassis can utilize PNT data from any other chassis in the network. If GPS is available on one chassis, that reference can be progogated to all others via ATWTT

3 – CONOPs and Use Cases



CONOP: Decentralized PNT on a Platform

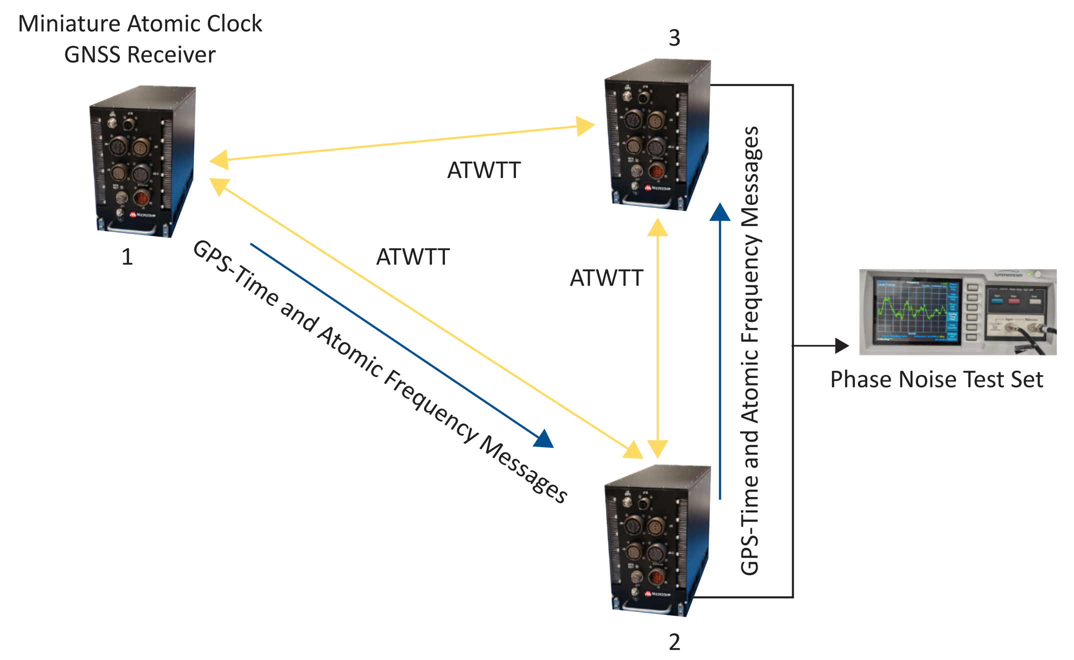
- Three SOSA-aligned chassis distributed across a platform, connected via optical ATWTT links. Each chassis maintains its own local reference frame. PNT data (GPS time, atomic frequency) is shared and translated across the network without requiring co-location of PNT sources
- No master node – any chassis can source or consume PNT data
- Dispersed PNT assets – any node on the platform can benefit from assets anywhere in the network
- Resilience – loss of one node does not disable timing for the others

CONOP: Phased Array Radar

- ATWTT provides frequency stability and phase coherence between all elements in a distributed phased array radar system without a shared timing backbone or GPS dependency. Each antenna element node maintains its own local timescale, synchronized to all others via ATWTT links.
- Sub-50ps coherence between nodes enables coherent beamforming at high frequencies
- GNSS independent – relative coherence across the array matters more than absolute UTC alignment
- Distributed aperture synthesis – elements can be physically separated across platforms

4 – Performance Characterization and Results

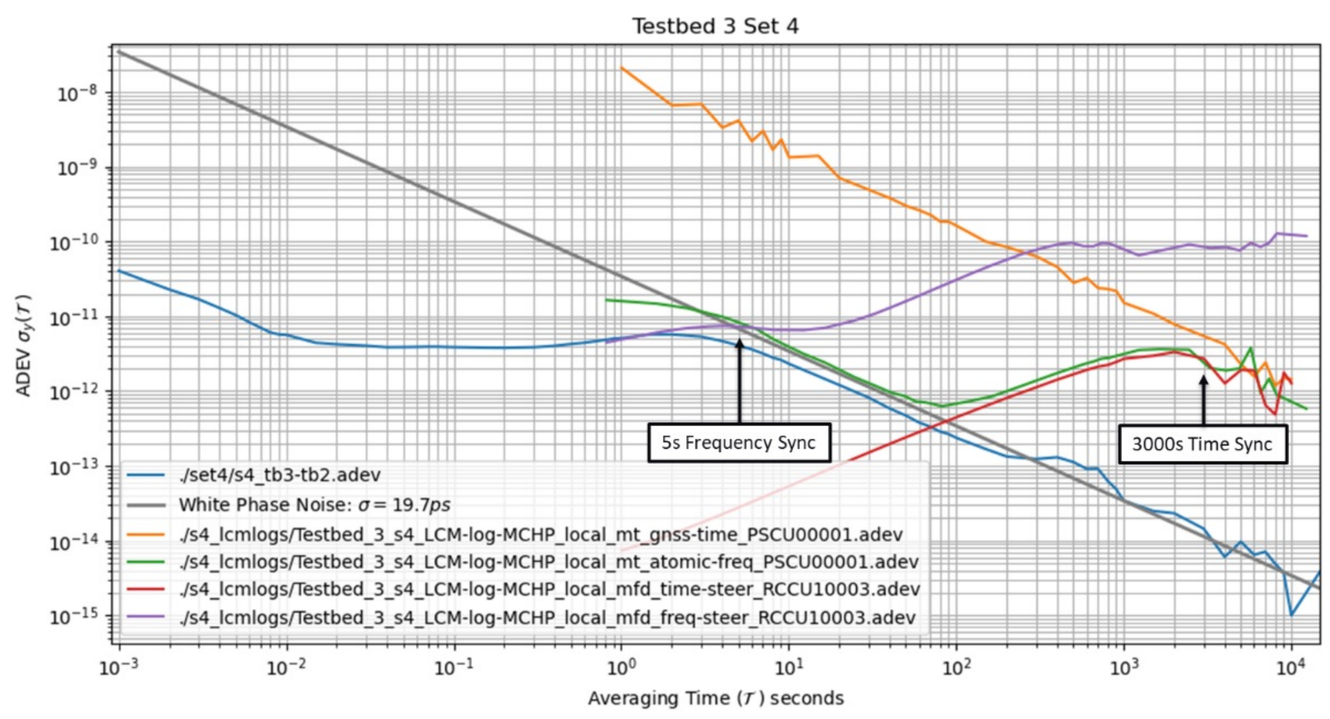
Lab Test Setup – Three Node Chain Configuration



What We Measured: Virtual Reference Fidelity

Each chassis independently measures its view of the GPS and atomic frequency references. Translation of the GPS measurements and MAC frequency are below the noise such that they appear invisible

Source	Chassis 1 (physical PNT sources)	Chassis 2 (1 Hop)	Chassis 3 (2 Hops)	Spread
GPS Bias	558 ps	562 ps	560 ps	4 ps
MAC Frequency	-3.175767E-10	-3.175746E-10	-3.175779E-10	3E-15



Local Coherence:

- Blue trace shows ~20 ps relative frequency stability between chassis 2 and 3, measured directly by 5120A phase noise test set on the 10 MHz outputs
- Control loops behave as designed, OCXO noise dominates below 5s, atomic clock between 5s and 3000s, GPS above 3000s
- Absolute phase coherence requires calibration to remove static hardware path offsets. We are currently investigating to achieve consistent sub-50 ps performance

MICROCHIP

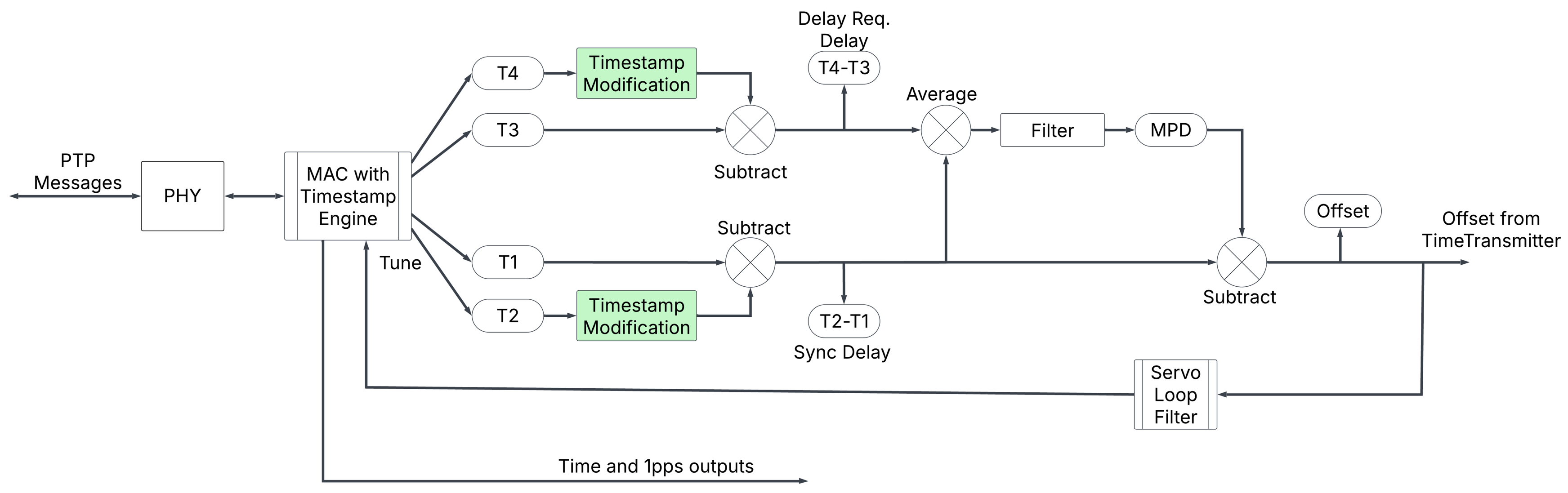
Simulating Network Impairments via Modifications to PTP Follower Timestamp Values

Devices to create PTP system impairments such as jitter and network delays are available from several vendors but are expensive . This paper explains some development work to simulate these two impairments as straightforward modifications to the timestamps after they are received in the follower. An explanation of how the code was implemented and results from the simulation and comparison to other methods are shown below.

Authors and presenters: Mike Overton and Ansul Kumar, Telestream

The block diagram for PTP4L is shown here. Timestamps are processed to calculate the mean path delay (MPD) and the Offset. The offset is filtered then used to control the clock in the MAC to track the clock in the grandmaster (GM)

Modification of the timestamps at the location indicated on the diagram can emulate real delay in the network. This project implemented two separate functions, one for packet delay variation (PDV) aka "jitter" and one for route switching.



TimeReceiver Block using PTP4L with Additions

Network PDV is simulated by randomly adding delay to selected PTP T2 and T4 message timestamps. The delay is applied within a configurable range and only to a percentage of messages, allowing controlled emulation of packet delay variation. Delay is only added since traffic only increases delay, it does not reduce delay.

For example, considering the PDV configuration below, PDV is applied to Sync and Delay Request messages. For each message, there is a 50% chance that a delay will be added. If selected, a random value between 100 ns and 100 µs is generated. For example, a Sync message may receive an additional 12 µs, while the next message may receive no delay, creating a random jitter pattern similar to real network conditions.

Route switching is generated using a deterministic step-based sequence. A set of predefined steps defines the simulated network path conditions. Each step specifies the duration (in seconds) and the additional delay applied to the T2 (Sync receive) and T4 (Delay Response receive) timestamps. When route switching is enabled, the system begins with Step 1 and applies the configured delays (in nanoseconds) for the specified duration. After the duration expires, the system automatically transitions to the next step and updates the delay values. Once the final step is reached, the sequence loops back to the first step and repeats. This creates a repeating delay pattern that mimics route changes where packets begin traveling through paths with different propagation delays.

Configuration commands to generate PDV and how they work

enable_delay_variation 3
delay_variation_probability 50
min_delay_variation 100
max_delay_variation 100000

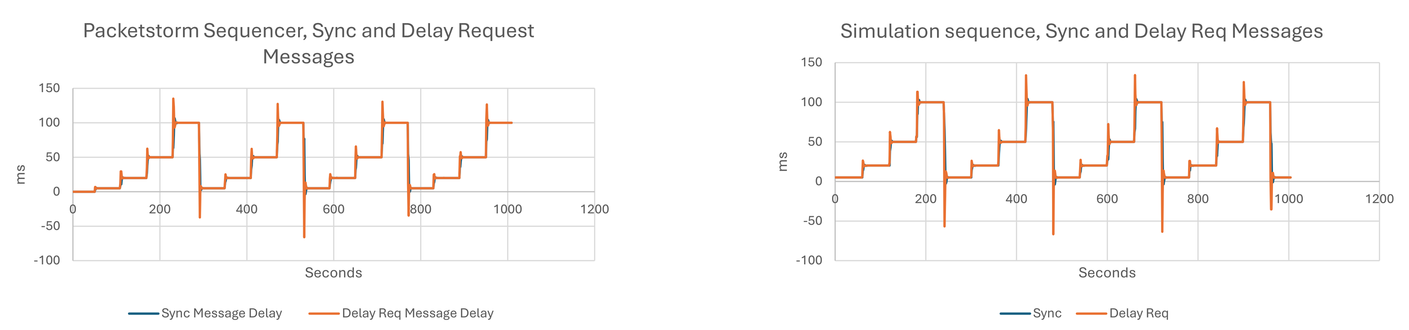
These are the configuration parameters added to PTP4L to implement PDV.

- "enable delay variation" allows adding PDV to the sync message timestamps, the delay request timestamps, both, or none.
- "delay variation probability" sets the percentage of the messages to which the PDV is applied.
- The min and max delay variation set the range of PDV in nano-seconds.

This set of configuration commands will apply the PDV to both messages, with a 50% probability over a uniform range of 100ns to 100us.

Comparison of PDV generation

Comparison of sequence generation

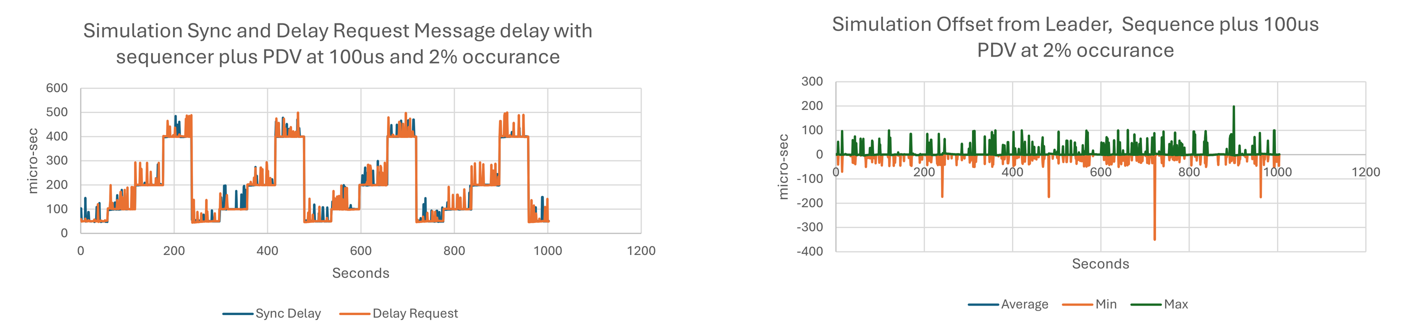


Configuration commands to generate route switching sequence

The configuration shown here has four steps which each last for 60 seconds. Each step increases the delay. The delay value is in nano-seconds so in this case the delays are 5 ms → 20 ms → 50 ms → 100 ms. This simulates scenarios where traffic is routed through progressively longer paths. Once the final step is reached, the sequence loops back to Step 1, continuously cycling through the delay profile until the simulation is stopped. The values for T2 and T4 may be the same to simulate symmetric networks or set to different values to simulate asymmetric networks.

enable_route_switching 1
step1_duration 60
step1_T2_delay 5000000
step1_T4_delay 5000000
step2_duration 60
step2_T2_delay 20000000
step2_T4_delay 20000000
step3_duration 60
step3_T2_delay 50000000
step3_T4_delay 50000000
step4_duration 60
step4_T2_delay 100000000
step4_T4_delay 100000000

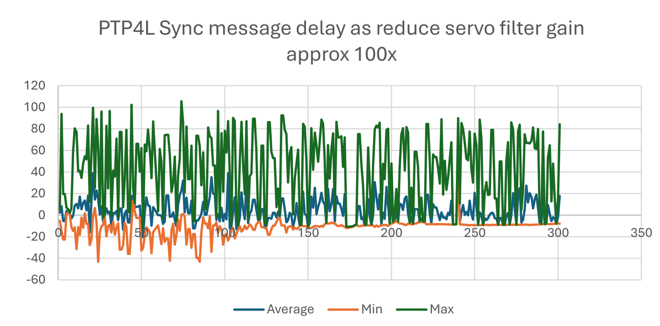
Simultaneous PDV and sequence generation via the Simulation



The PDV and route switching can both be applied simultaneously as demonstrated on these two plots. In this case there 100us PDV at a 2% occurrence rate, in conjunction with a step sequence of 50, 100, 200 and 400us.

Future Steps: The simulation could be improved by adding other PDV profiles such as exponential or gamma contours to better replicate common traffic induced PDV. The PDV could also be linked to the sequence process to allow the PDV to vary over time.

Measurement Methodology: The graphs in this poster were created by capturing the sync and delay request times from PTP4L at 8 messages a sec. If the servo loop filter gain is high, then the timestamp clock will be tuning rapidly in response to the offset from leader. This makes it appear that the minimum delay is unstable. To prevent this, the servo loop filter gain was set low so then the delay measurements show the true variations. This plot starts a servo gain of 0.7 and steps down to 0.25, then 0.05, the 0.005. So about 100 to 1 reduction

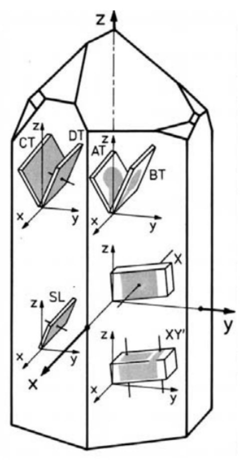


telestream

Conclusion: The straightforward SW effort described in this paper provides an effective alternative to conventional network impairment equipment. The results are quite similar to what is available from conventional devices, very flexible and capable of emulating live networks, and allows every team member simultaneous access to this capability.

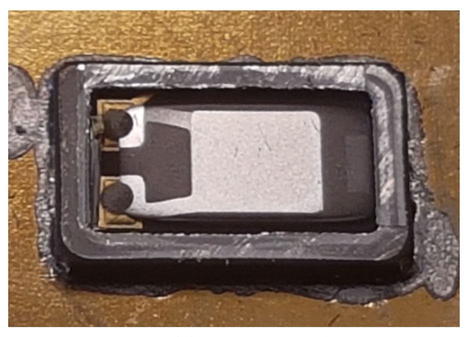
Crystals and their use in MultiMode resonance

Quartz resonator technology is mature and well understood



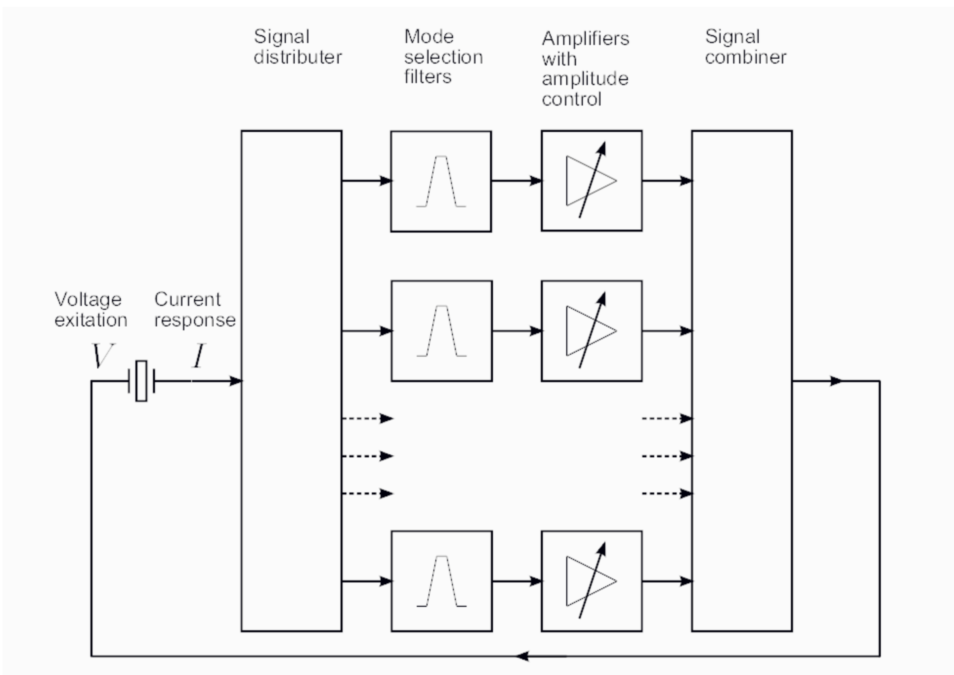
Observations:
- 100 plus year old technologies.
- Piezoelectric so simple to use.
- Extremely high Q factor, 10^6 and beyond for frequencies >10 MHz.
- Some cuts (AT) simpler than other cuts (SC).
- Hence SC cut expensive.

Major improvements in those years:
- Hydrothermal synthesis in vertical autoclaves.
- Packaging issues solved (Motorola, Epson).
- SC cut (Jack Kusters).
- Only miniaturization as current development.
- MEMS is not so much differentiation at all.
- Cleaner dicing, lapping, packaging.



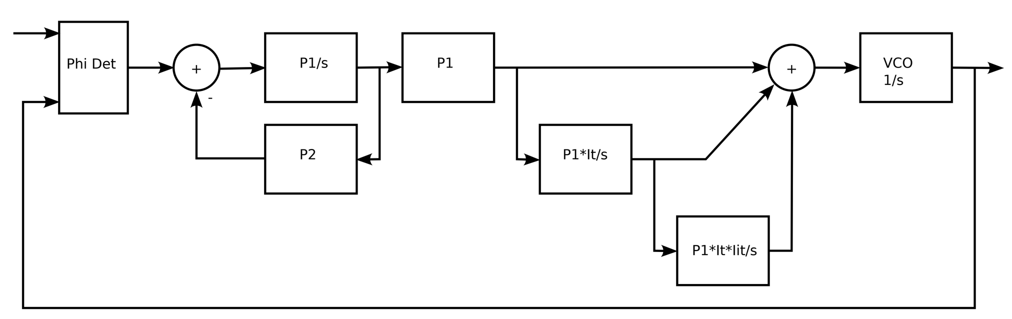
our crystal in 5032 package

On amplifier based oscillators in MultiMode



Major issue:
- The relative detuning contribution by the filters is mitigated by a factor of $Q_{filter}/Q_{crystal}$.
- An inductor may bring a detuning of 10%, a crystal maybe 100 ppm.
- The inductor is required to have a high Q to add decent separation of the resonances. An example inductor Q would be 100.
- So the filter can kill the desired 'crystal only' behaviour.

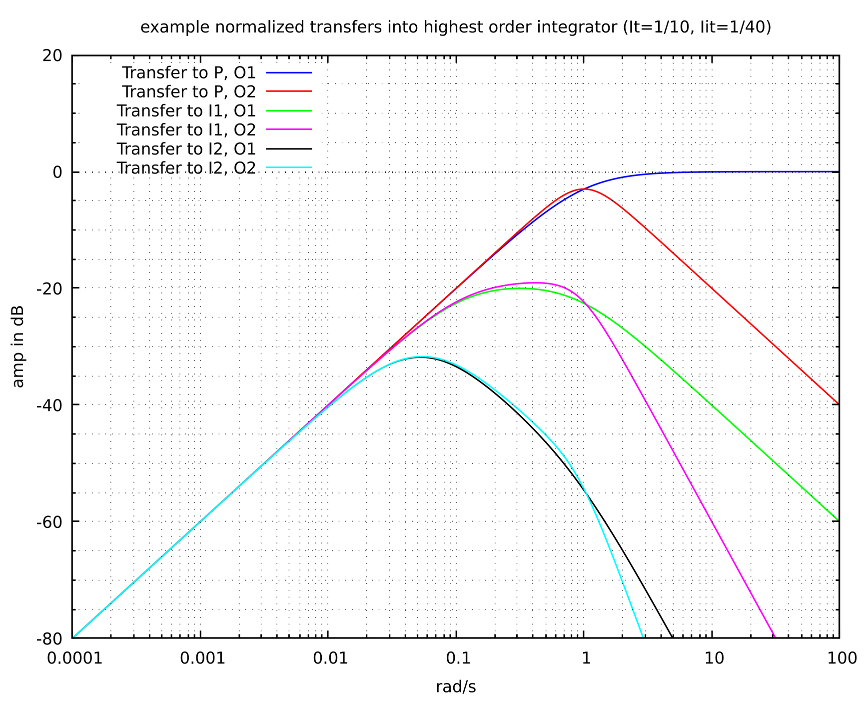
PLL orthogonalization

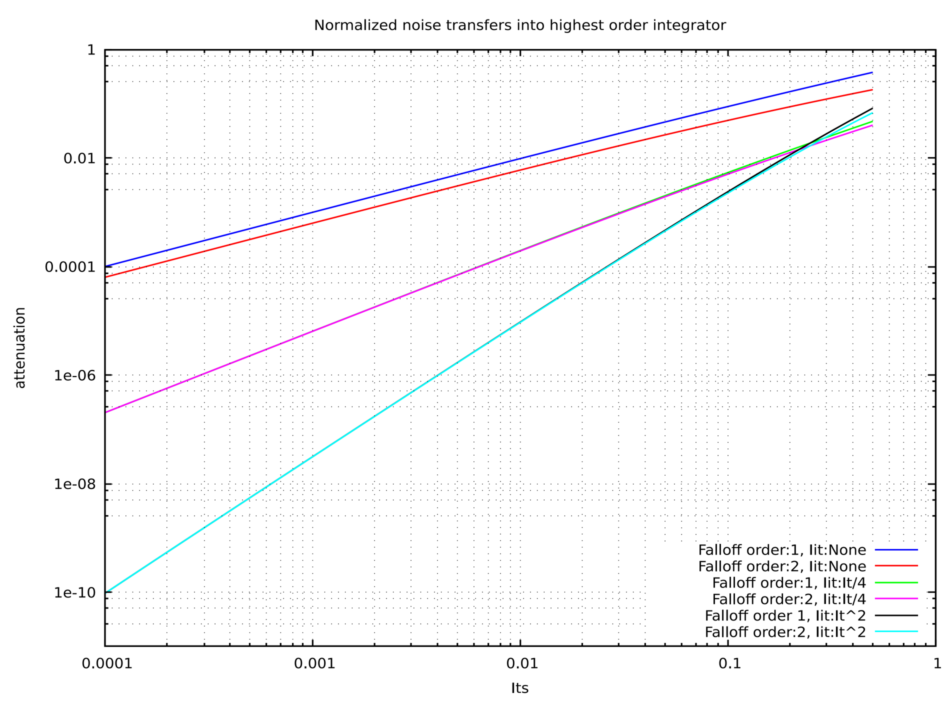


Observations:
- Shown block diagram is complete and mathematically expandable to any PLL type/order.
- But the structure allows orthogonal behaviour of P1, It (I tick), Iit etc.
- For instance: substituting P1*s with s makes P1 disappear (normalization).
- Or choosing P2=sqrt(2) sets the transfer on Butterworth, P2=sqrt(3) on Bessel.

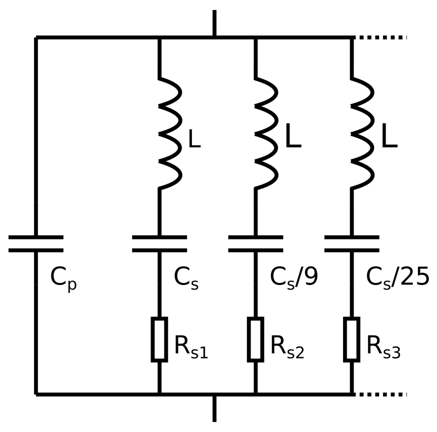
Effect:
- Complete analysis possible, generation of nomograms simple.

Further remarks:
- The larger loopgain path 'takes' the larger signal carrying. Hence the highest order integrator at low frequencies carries the set frequency, at higher frequencies this moves upwards through lower order integrators towards the P branch.
- It becomes easier to break up noise propagation, see below.
- The examples use the P1 noise attenuation as reference.





On crystals in oscillators



Observations:
- All resonance structures sort of support overtones, but in physical crystal shape 'anti' measures can be taken.
- The electric model is formed by actual capacitor and electrical equivalents of actually springs and masses.
- Physics dictates that the overtones show for temperature and stress and 'common mode' and 'differential mode' behaviour.
- This differential behaviour is never zero (thus our equivalent model has very limited accuracy).

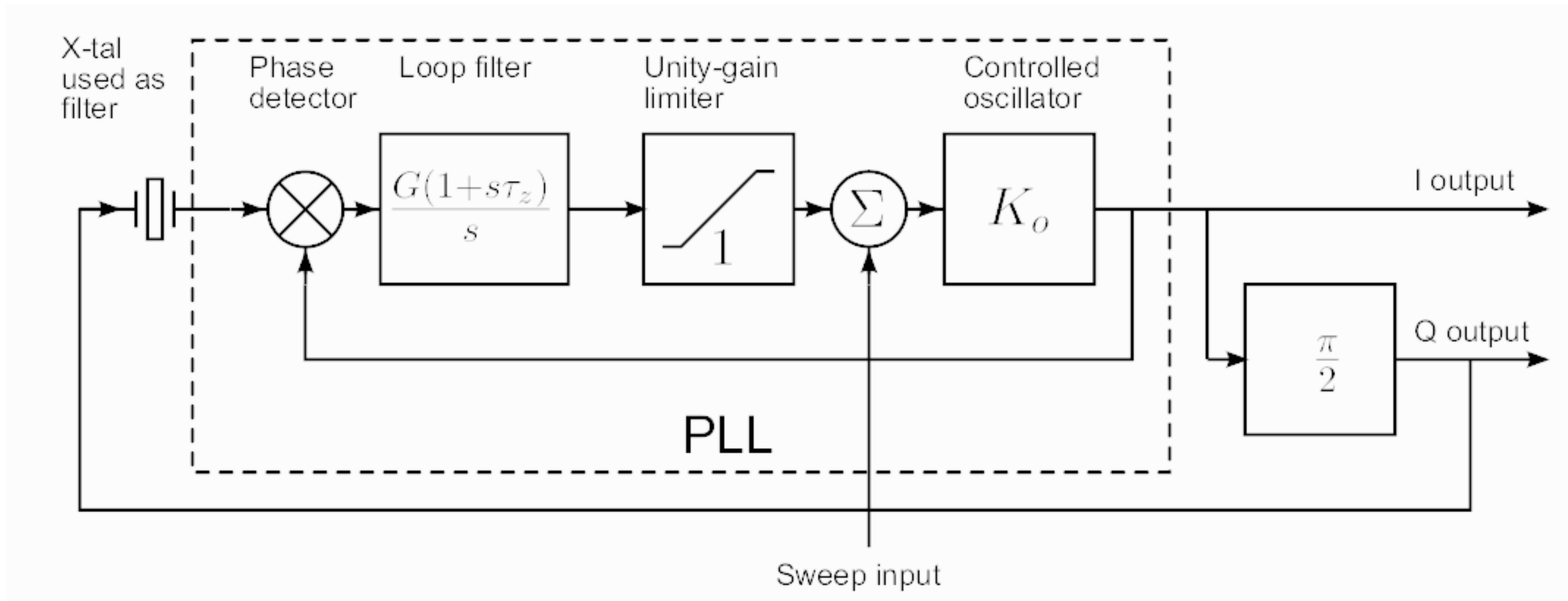
Our use:
- For MultiMode we use multiple modes
- In MultiMode the smaller portion of the 'differential' is used to compensate the larger total of 'common plus differential mode'

On amplifier based oscillators

Observations:
- Most oscillators combine in a single device both sensing and actuation of some resonance element. This is a bad idea for noise (Westra, 1998).
- An oscillator is here taken as strictly the circuit where the Barkhausen condition is obeyed (loopgain=1).

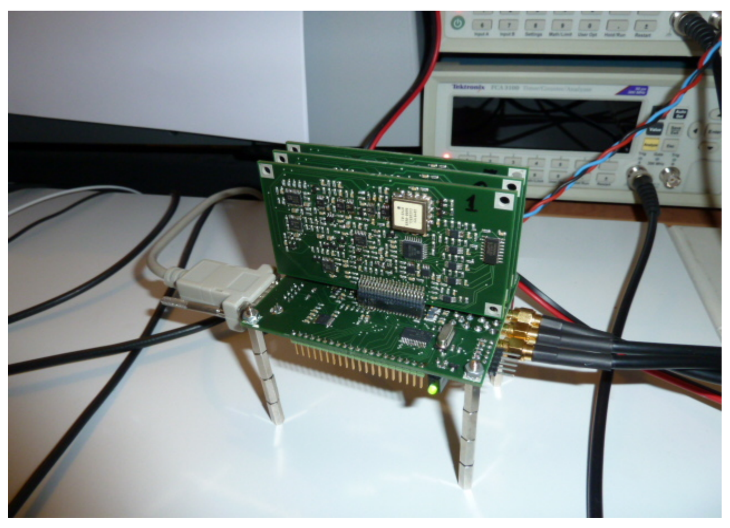
Alternative:
- Have Barkhausen on another location, typically with a much higher resonance frequency.
- Have crystal energized without Barkhausen condition, sensed with a receiver like structure.

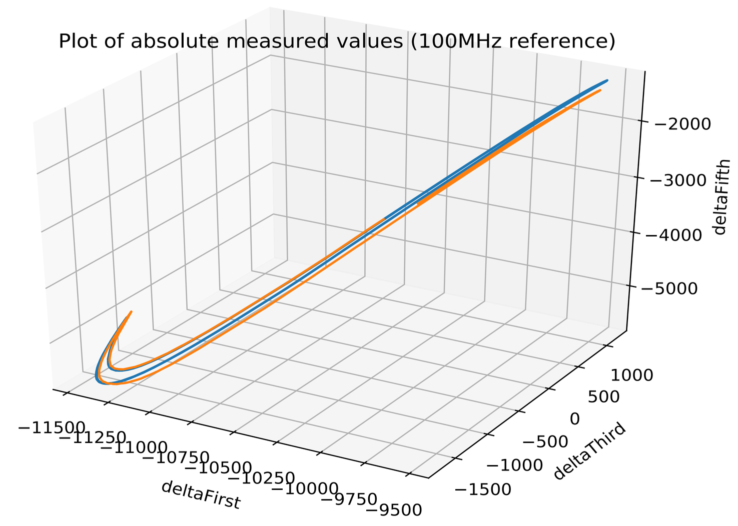
On PLL based oscillators

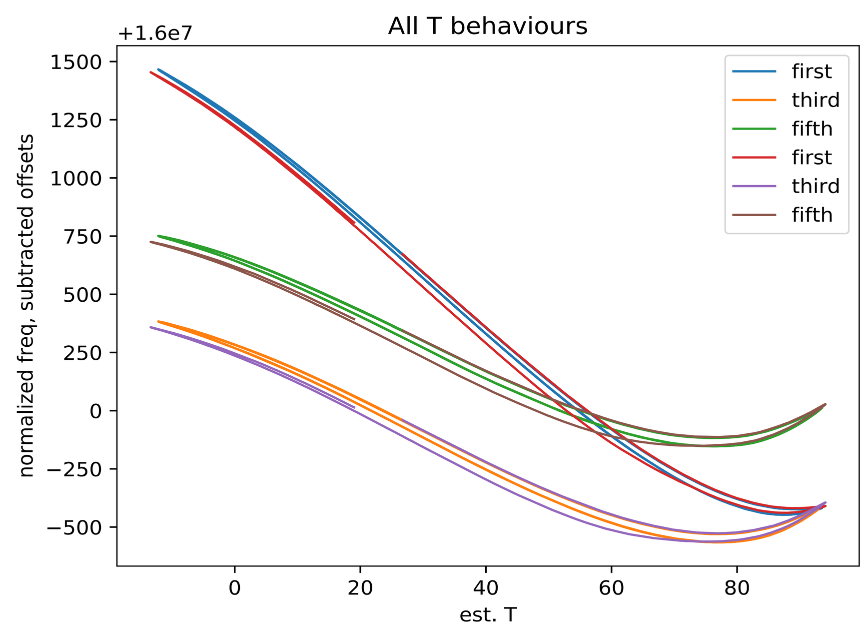


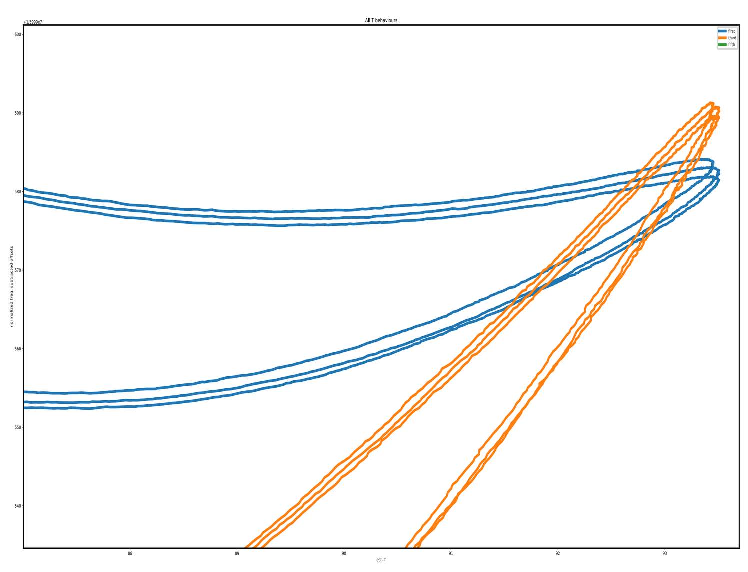
Observations:
- In this block diagram only a single mode is drawn, with a simple additive network one can support multi resonances.
- The high Q need for amplifier based MultiMode is here replaced with a synchronous detection system. Effectively a 'small' bandwidth is a simple design effort.
- Temperature coefficients all taken into DC..

Measurement results from an early functional model









SEMI BLOCKS

Endpoint Clocks : Role of the Oscillator

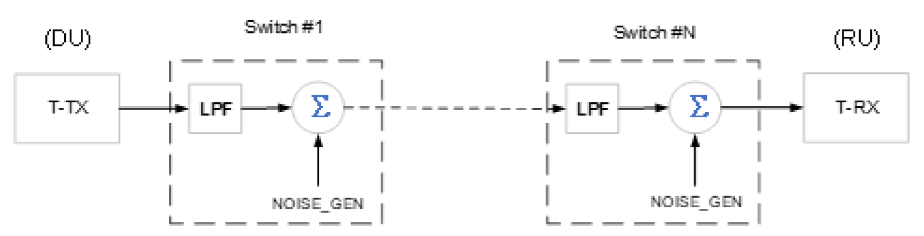
Oscillator Time Uncertainty Crucial in Time Synch Applications

Kishan Shenoi (kshenoi@sitime.com)

Example Design of Base Station Clock : For RU need $|TE_{max}| < 65$ns or $\sigma_E < 16.25$ ns

Fronthaul network : N T-BC switches

STEP 1: Develop noise model of Fronthaul network



Noise Generation spectrum : $S_\eta(f)$

Low Pass Filter : $H_L(f)$

Noise generation TDEV : $\sigma_\eta(\tau)$

Table 7-5 – Dynamic time error low-pass filtered noise generation (TDEV) for T-BC/T-TSC with constant temperature (within ±1K)

T-BC/T-TSC Class	TDEV limit [ns]	Observation interval τ [s]
A	4	$m < \tau \le 1\,000$ (Note)
B	4	$m < \tau \le 1\,000$ (Note)
C	2	$m \le \tau \le 1\,000$ (Note)
D	For further study	For further study

NOTE – The minimum τ value m is determined by packet rate of 16 packet per second (m = 1/16) or 1 PPS signal (m = 1).

$dTE_H < 70$ns (A/B); < 30ns (C)

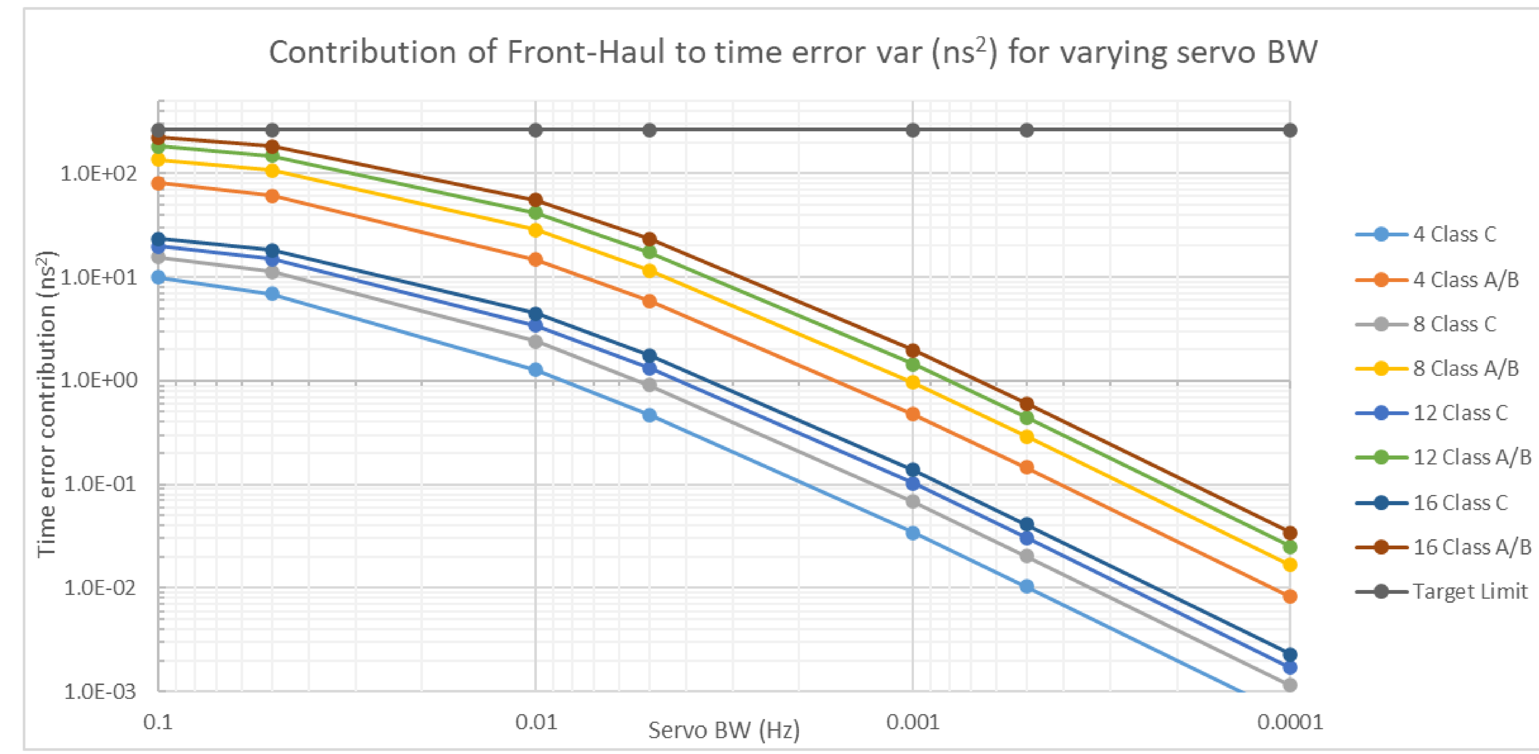
Extracted from G.8273.2

- Output noise spectrum as function of input (for each switch):
  $$X_n(f) = |H_L(f)|^2 \cdot X_{(n-1)}(f) + S_\eta(f)$$
- Low Pass Filter (LPF) characteristic (2nd order) assuming maximally-flat equivalent:
  $$|H_L(f)|^2 = \frac{1}{1+\left(\frac{f}{f_C}\right)^4}$$
- Spectrum from TDEV (approximation):
  $$S_\eta(f) = \left(\frac{0.75}{f}\right)\cdot\left(\sigma_\eta\left(\tau = \frac{0.3}{f}\right)\right)^2 \quad (ns)^2/Hz$$
- For Class C T-BC ($f < 16$):
  $$S_\eta(f) = |H_C(f)|^2 * \left(\frac{0.75}{f}\right)\cdot(2)^2 \quad (ns)^2/Hz$$
- For Class A/B T-BC ($f < 16$):
  $$S_\eta(f) = |H_A(f)|^2 \cdot \left(\frac{0.75}{f}\right)\cdot(4)^2 \quad (ns)^2/Hz$$

Note: $H_A(f)$ and $H_C(f)$ needed to meet dTE_H requirement

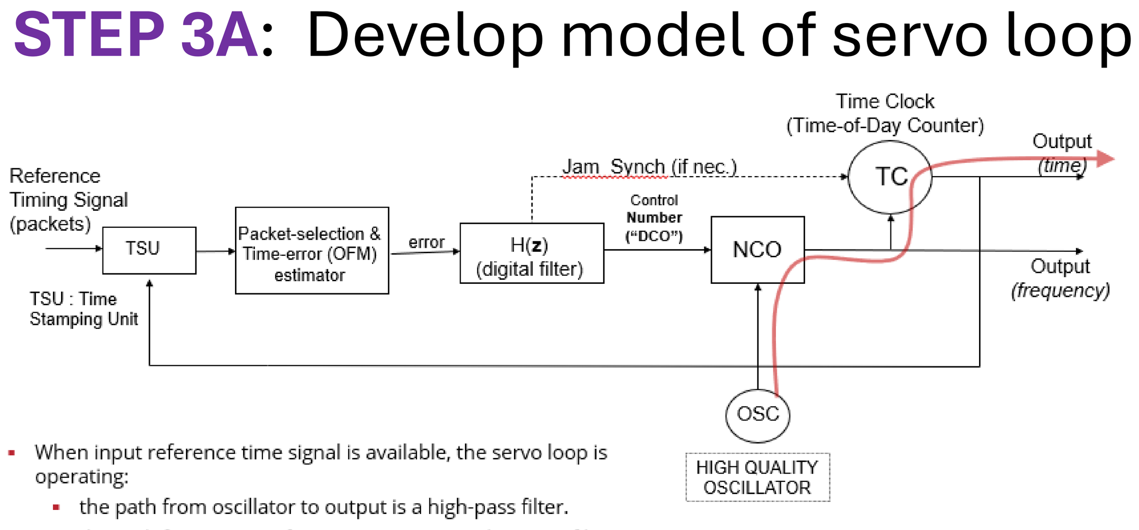
Power Spectral View

STEP 2: Establish range of servo bandwidth and allowable oscillator contribution



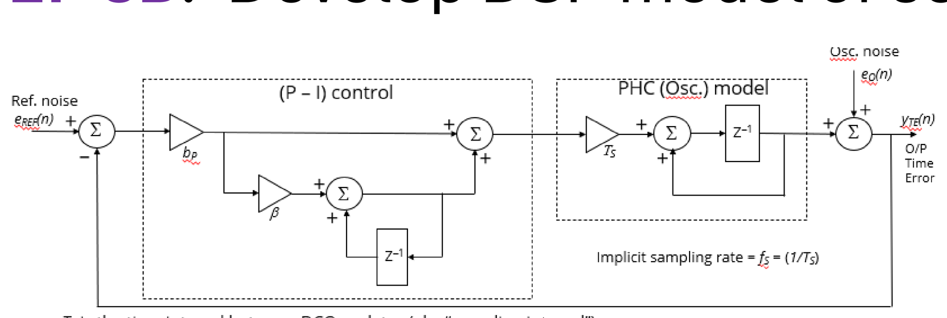
STEP 3: Estimate oscillator contribution to clock time error

STEP 3A: Develop model of servo loop



- When input reference time signal is available, the servo loop is operating:
  - the path from oscillator to output is a high-pass filter.
  - the path from input reference to output is a low-pass filter
- When input is unavailable, the servo is not operating and the path from oscillator to output is direct (aka "holdover").

STEP 3B: Develop DSP model of servo



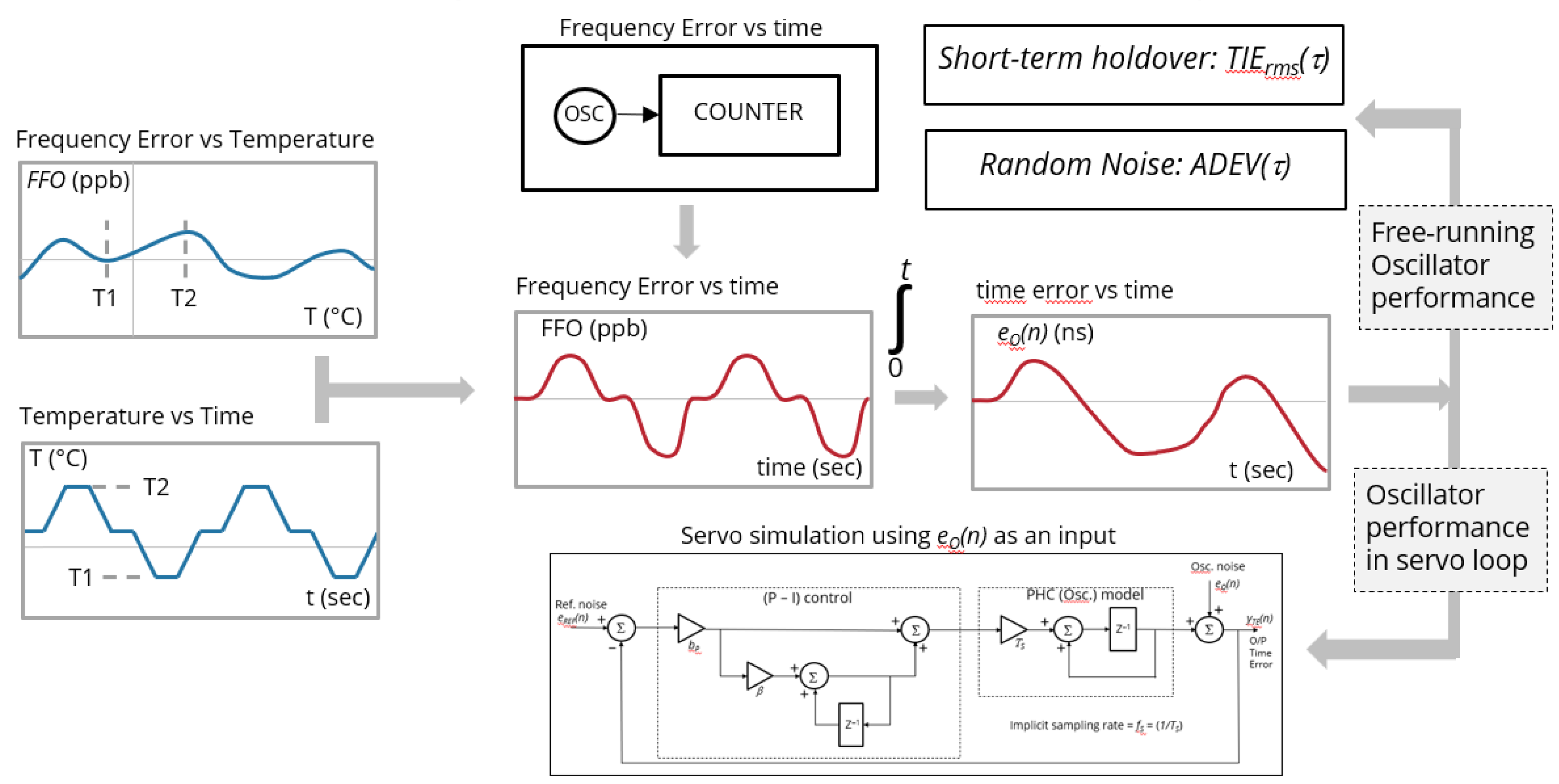
- T_S is the time interval between DCO updates (aka "sampling interval")
- For considering oscillator contribution, we can set $e_{REF}(n)$ to zero
- The (HPF) transfer function, $H_{H1}(z)$, from oscillator as input to output time error and the equivalent frequency response is easily established
- Measurement data provides samples of $\{e_O(n)\}$ allowing a sample-by-sample simulation based on this Z-plane model

$$H_{H1}(z) = \frac{(z-1)^2}{z^2 - z\left(2 - T_S\cdot b_P\cdot(\beta+1)\right) + (1 - T_S\cdot b_P)}$$

The coefficients b_P and β govern the desired passband ripple (usually < 0.1dB) and BW as a fraction of the sampling frequency f_S.

STEP 3C: Estimate variance of oscillator contribution by simulation

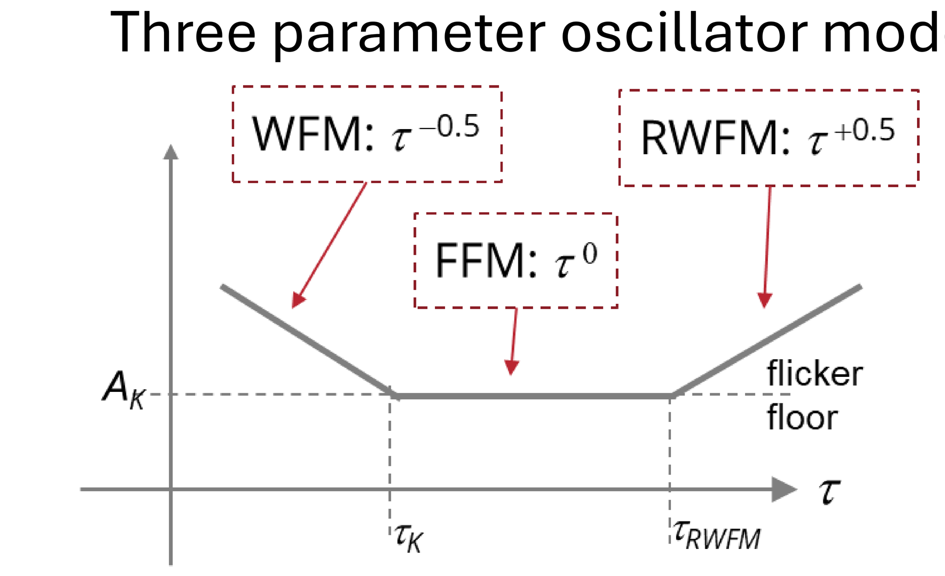
Time Uncertainty estimates from measurements and thermal profile



Step 4: Verify that total time error = { contributions of input + oscillator (constant temp.) + oscillator (variable temp)} satisfies target limit (with margin)

Estimating oscillator contribution using ADEV (e.g., OCXO)

Three parameter oscillator model

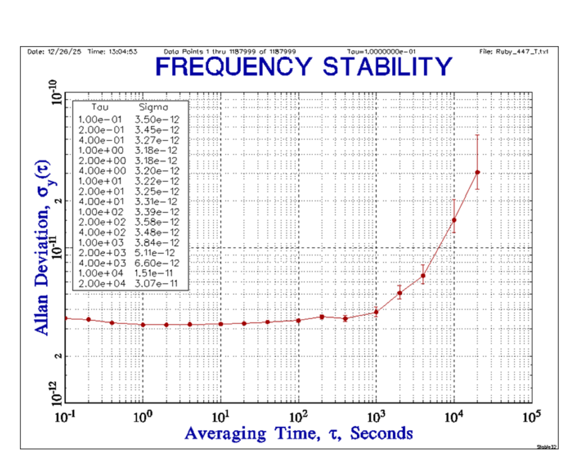


Two break-points: τ_K and τ_{RWFM}

ADEV(τ_K) : A_K ; TDEV(τ_K) : B_K ; $B_K \approx \frac{\tau_K \cdot A_K}{\sqrt{3}}$

$$X_{OSC}(f) = \begin{cases} {B_K}^2\cdot\left(\frac{0.75}{f}\right)\cdot\left(\frac{0.3}{f\cdot\tau_K}\right) \;;\; f > \frac{0.3}{\tau_K} & (WFM) \\ {B_K}^2\cdot\left(\frac{0.75}{f}\right)\cdot\left(\frac{0.3}{f\cdot\tau_K}\right)^2 \;;\; \frac{0.3}{\tau_K} \ge f > \frac{0.3}{\tau_{RWFM}} & (FFM) \\ {B_K}^2\cdot\left(\frac{0.75}{f}\right)\cdot\left(\frac{\tau_{RWFM}}{\tau_K}\right)^2\cdot\left(\frac{0.3}{f\cdot\tau_{RWFM}}\right)^3 \;;\; \frac{0.3}{\tau_{RWFM}} \ge f & (RWFM) \end{cases}$$

$$Y_{OSC}(f) = (|H_{H1}(f)|)^2 \cdot X_{OSC}(f)$$



3-parameter model :
$\tau_K \approx 1$s; $A_K \approx 3.4E{-12}$; $\tau_{RWFM} \approx 400$s

standard deviation (rms)

Osc. ADEV model		Servo Bandwidth			
τ_K, τ_{RWFM}	A_K (ppb)	0.1Hz	0.01Hz	0.001Hz	0.0001Hz
1s, 400s	3.4E-3	0.17E-1 ns	0.28E0 ns	0.84E+1 ns	0.26E+2 ns
1s, 400s	3.4E-2	0.17E0 ns	0.39E0 ns	0.84E+2 ns	0.26E+4 ns
1s, 100s	3.4E-3	0.21E-1 ns	0.5E0 ns	0.17E+2 ns	0.53E+3 ns
1s, 100s	3.4E-2	0.20E0 ns	0.53E+1 ns	0.17E+3 ns	0.53E+4 ns

Observations:

1. Time error standard dev. is proportional to oscillator flicker floor ADEV
2. For narrower bandwidths, error sigma inversely proportional to square root of flicker-random-walk breakpoint

Useful data sheet information regarding oscillator (e.g., TCXO)

Datasheet Specification	Servo Bandwidth		
	0.1 Hz	0.01 Hz	0.001 Hz
Constant temperature 1	12.0 ps rms	265.0 ps rms	10 ns rms
Constant temperature 2	XX ps rms	XX ps rms	XX ps rms
Constant temperature 3	XX ps rms	XX ps rms	XX ps rms
Constant temperature 4	XX ps rms	XX ps rms	XX ps rms

standard deviation (rms)

	Servo Bandwidth		
	0.1 Hz	0.01 Hz	0.001 Hz
Temperature Profile 1	44.1 ns p-p	608.5 ns p-p	4254 ns p-p
Temperature Profile 2	XX ns p-p	XX ns p-p	XX ns p-p
Temperature Profile 3	XX ns p-p	XX ns p-p	XX ns p-p
Temperature Profile 4	XX ns p-p	XX ns p-p	XX ns p-p

Good information to have in the oscillator data sheet