Time scale algorithm
~ *basics to applications* ~

* Lecture *

Yuko Hanado（ Retired NICT, Japan）

Bureau
International des
Poids et
Mesures

CCTF Technical Exchange: “Time scale algorithms "
June 25, 2025

Purpose of this tutorial

- **Understand the outline of**
  - **what the *averaged atomic timescale “TA”* is,**
  - **what its characteristics are,**
  - **how to calculate it,**
  - **how to use it.**

※ **“*TA*” : averaged atomic timescale**
Making ***TA*** is averaging many clocks to construct one virtual clock.

Contents

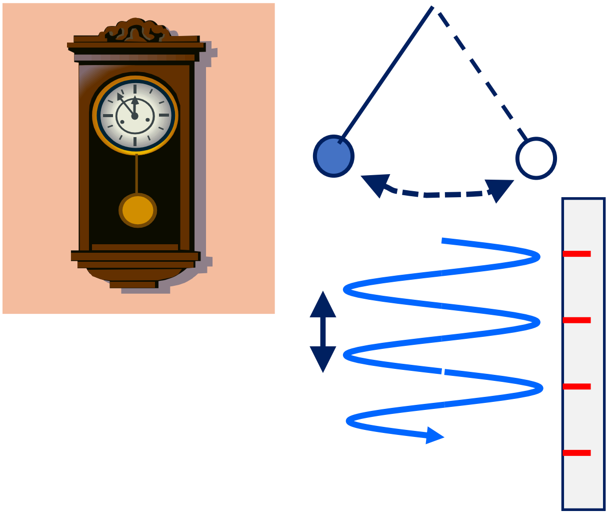
Contents

Bureau
International des
Poids et
Mesures

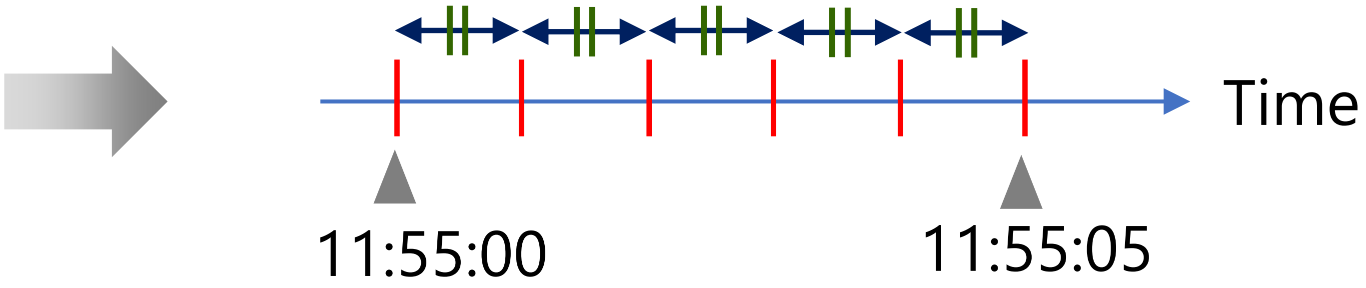
1.1 Why do we make *TA* ?

■ How can we measure the flow of time?

- Measure the time **by counting regularly repeating phenomena**.
- This repeating cycle becomes a "scale of the ruler" to measure the length of time.



- **Clock oscillations＝ Ruler for time**

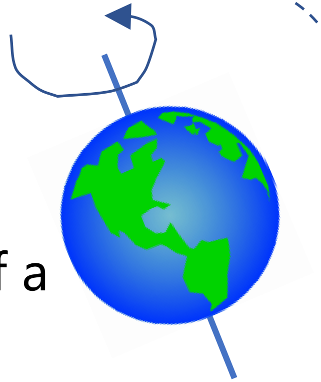


1.1 Why do we make *TA* ?

■ A frequency-stable oscillator is the basis of a clock.

- Previously: Earth’s rotation

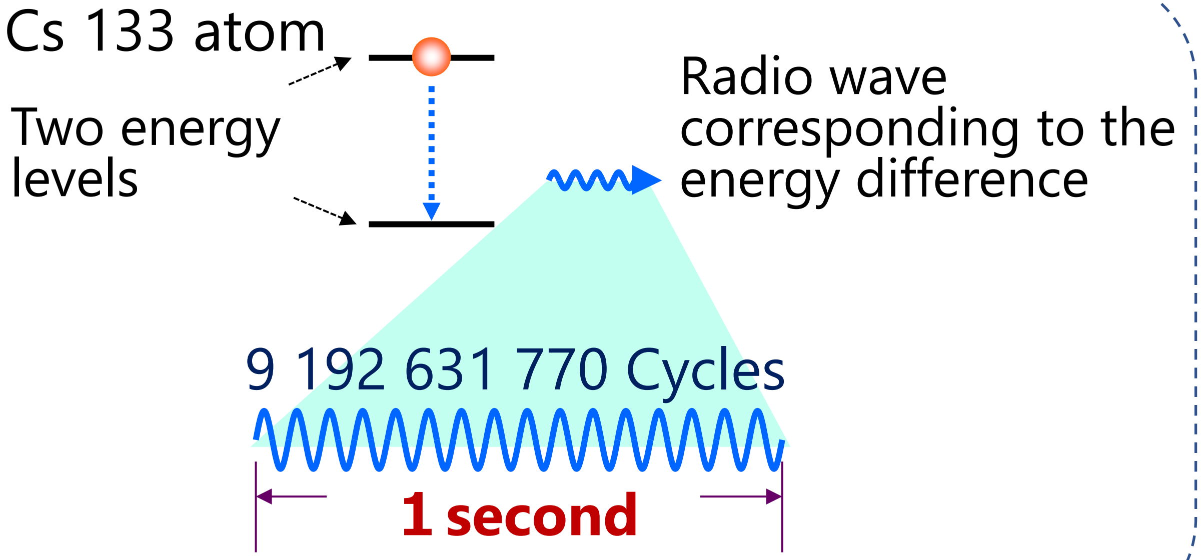
1 day = 1 rotation
1day/24/60/60 = 1sec



- Observation progress has revealed that the length of a day is not constant (fluctuates on the order of milliseconds)

https://www.iers.org/IERS/EN/Science/EarthRotation/LODplot.html?nn=12932

- **Current *SI* second definition:**



Cs 133 atom

Two energy levels

Radio wave corresponding to the energy difference

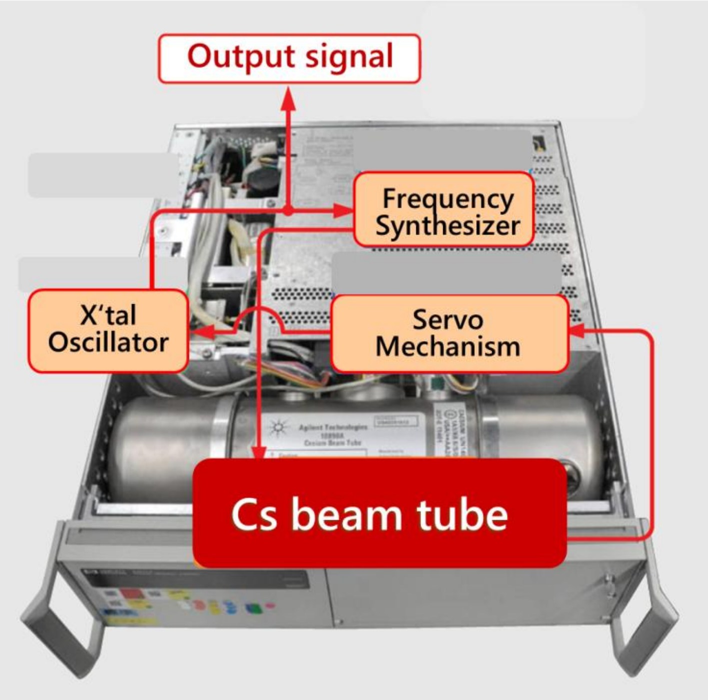
9 192 631 770 Cycles

1 second

1.1 Why do we make *TA* ?

■ Is one “high accuracy Cs atomic clock” enough? NO!

- This is because any atomic clock has individual fluctuation and a limited lifetime. This is the destiny of any device.



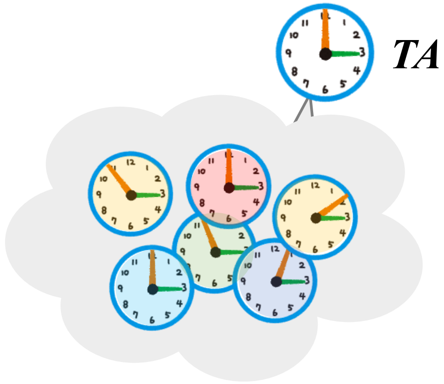
※ Cs atomic clock (Example)

- Atomic clocks are precision devices containing an atomic sample, an oscillator to excite the atomic resonance, and electronic control circuits.
- Instability and offsets in the output frequency are caused by environmental perturbations (pressure, temperature, vibration, etc.)
- The time of this clock will stop as the device ends its lifetime.

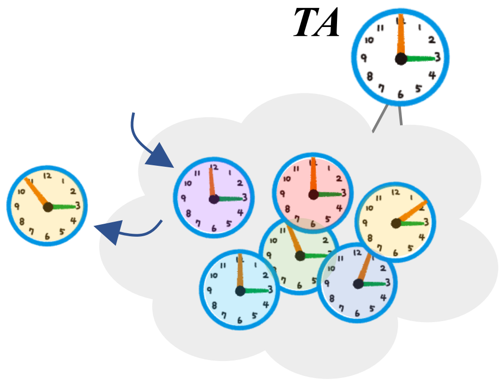
1.1 Why do we make *TA* ?

■ *Averaged atomic time “TA” is* one solution for stable continuous time.

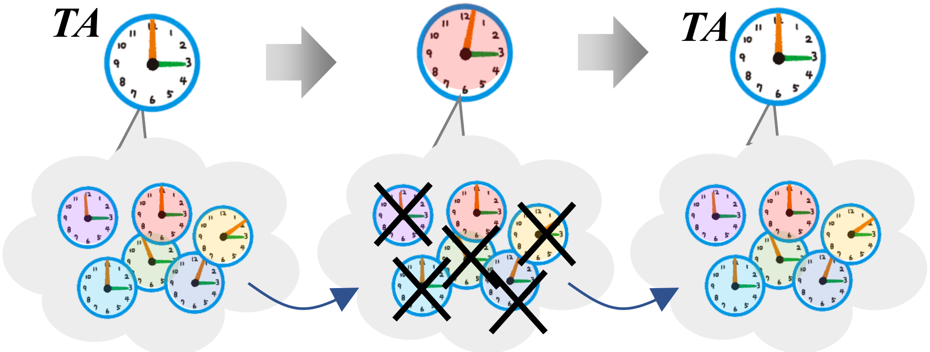
- **Averaging many clocks to construct one virtual clock** is a possible solution to the instrumental difference and the finite life of each clock.
- <u>Strong points of *TA*:</u>



- Individual deviations are smoothed out.



- The effect of a degrading clock can be removed by replacing it with a good one.



- The time scale does not stop if at least one clock is alive.

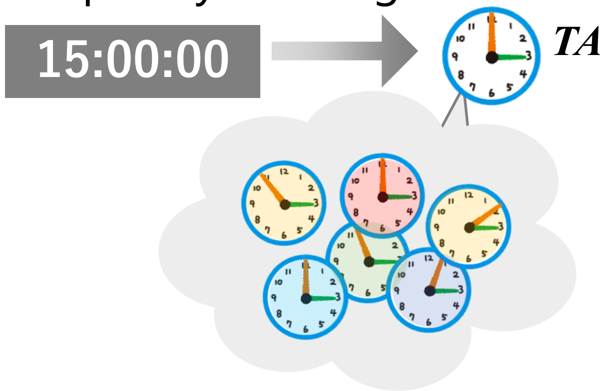
→ **Robust and continuous time can be obtained!**

1.1 Why do we make *TA* ?

■ Consideration points for using *TA* properly and effectively.

Attention!

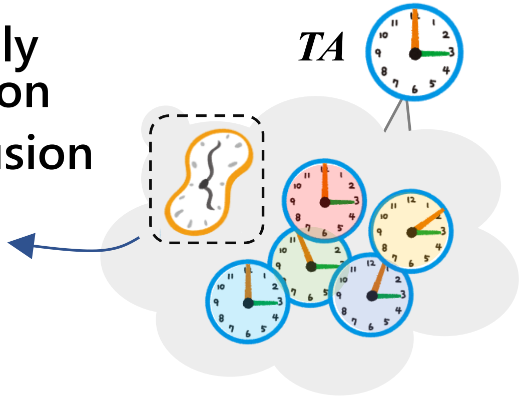
Initial synchronization & frequency steering



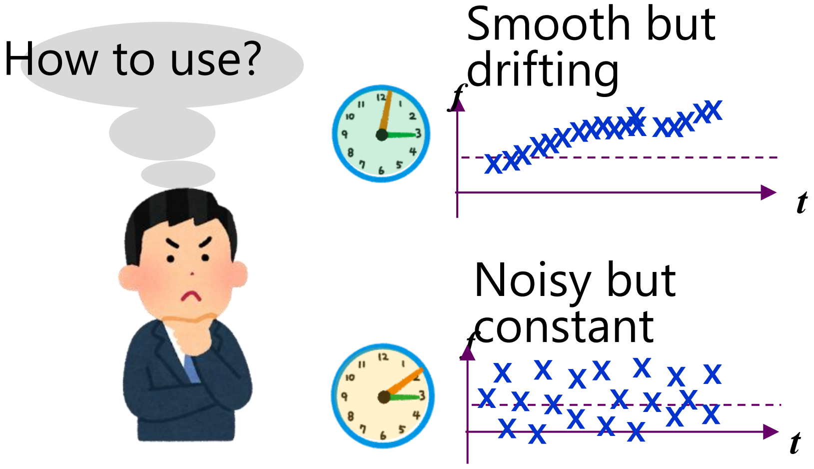
- **Calibration using an external reference time** is required for accuracy.
- Clock average is stable and robust, but accuracy is not guaranteed.

Anomaly detection

& Exclusion



- If clock anomalies are not detected and treated properly, the averaged result will degrade.



- **The proper algorithm** should be applied to achieve the desired outcome.

→ **By addressing these points, we can obtain a high-performance *TA*.**

1.1 Why do we make *TA* ?

Additional notes for making *TA*

- ***TA* as the basis for *UTC/UTC(k):***
  - Proper algorithm depends on the properties of the targeted time scale.
    - *UTC* → long-term stability and reliability → updating monthly
    - *UTC(k)* → usefulness in daily life → updating closer to real-time
- **The aims of *TA* is to achieve high-frequency stability.**
  - Usually, long-term stability is required.
  - There are several variances (Allan variance etc.) used as performance indicator.

The accuracy is not the primary concern.

- **Enhancing robustness is also important.**
  - The point is how it can maintain quality even in various unexpected conditions. (including clocks entry and exit, unpredictable anomalies,,,)

Contents

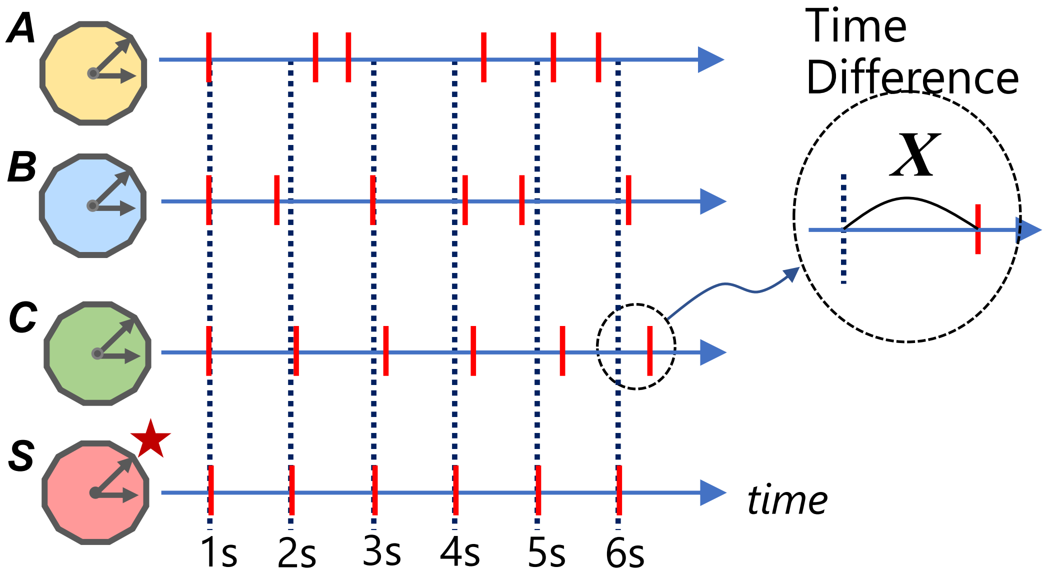
2.1 *TA by basic averaging* : Principle

■ Averaged atomic Time: Constructing a virtual clock by averaging.

→ **Specifically, what should we do?**

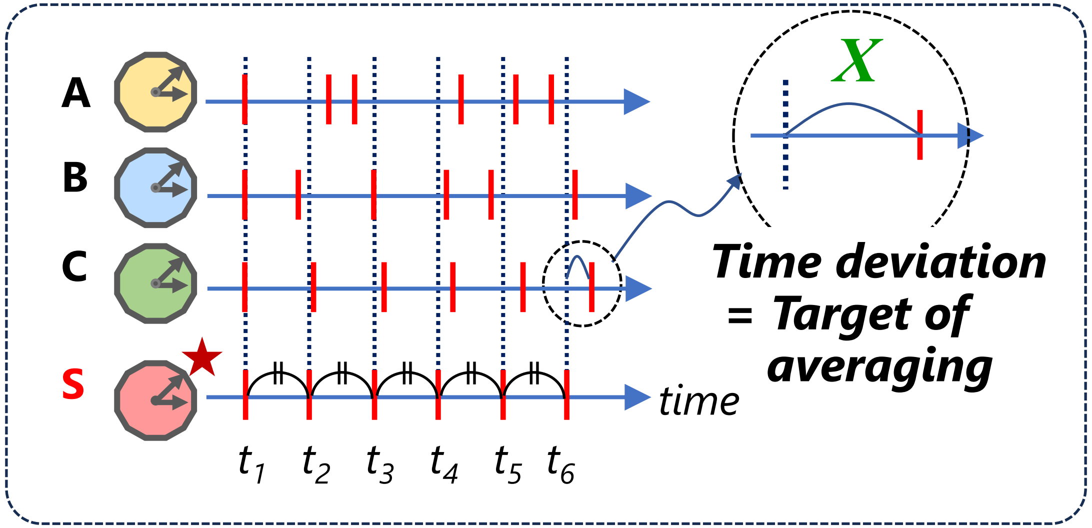
※ Preconditions:

- Each atomic clock generates almost stable output signals.
  (e.g. Pulse-per-second (*PPS*) + 10MHz sine wave, etc.)
- However, the clock's frequencies are not exactly the same. They show fluctuation.
- There is a stable and reliable clock (★) which can be used as reference time.
- The time (phase) differences (X) of all clocks vs this reference clock can be measured simultaneously.

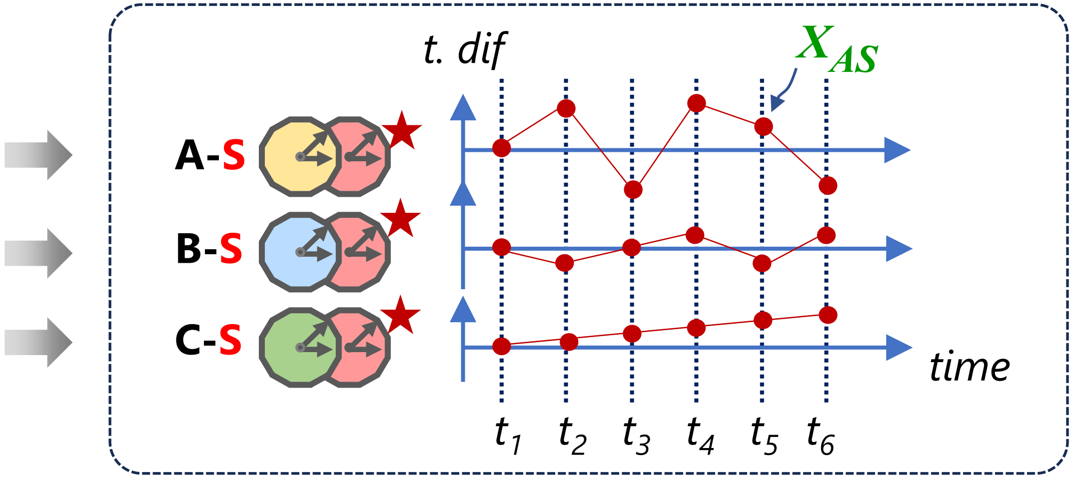


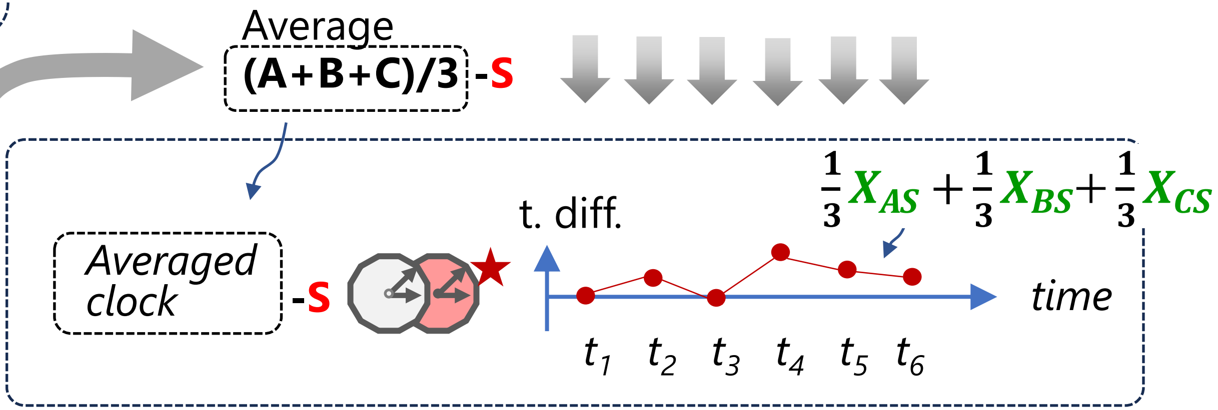
2.1 *TA by basic averaging* : Principle

[1] Averaging the random fluctuations of each clock leads to a stable virtual clock.



Time deviation = Target of averaging





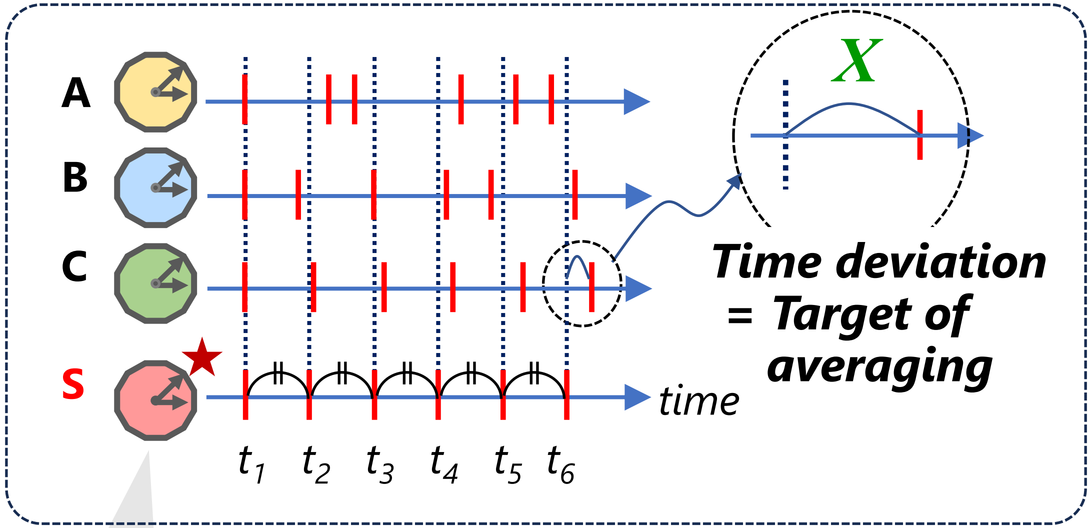
$$\frac{1}{3}X_{AS}+\frac{1}{3}X_{BS}+\frac{1}{3}X_{CS}$$

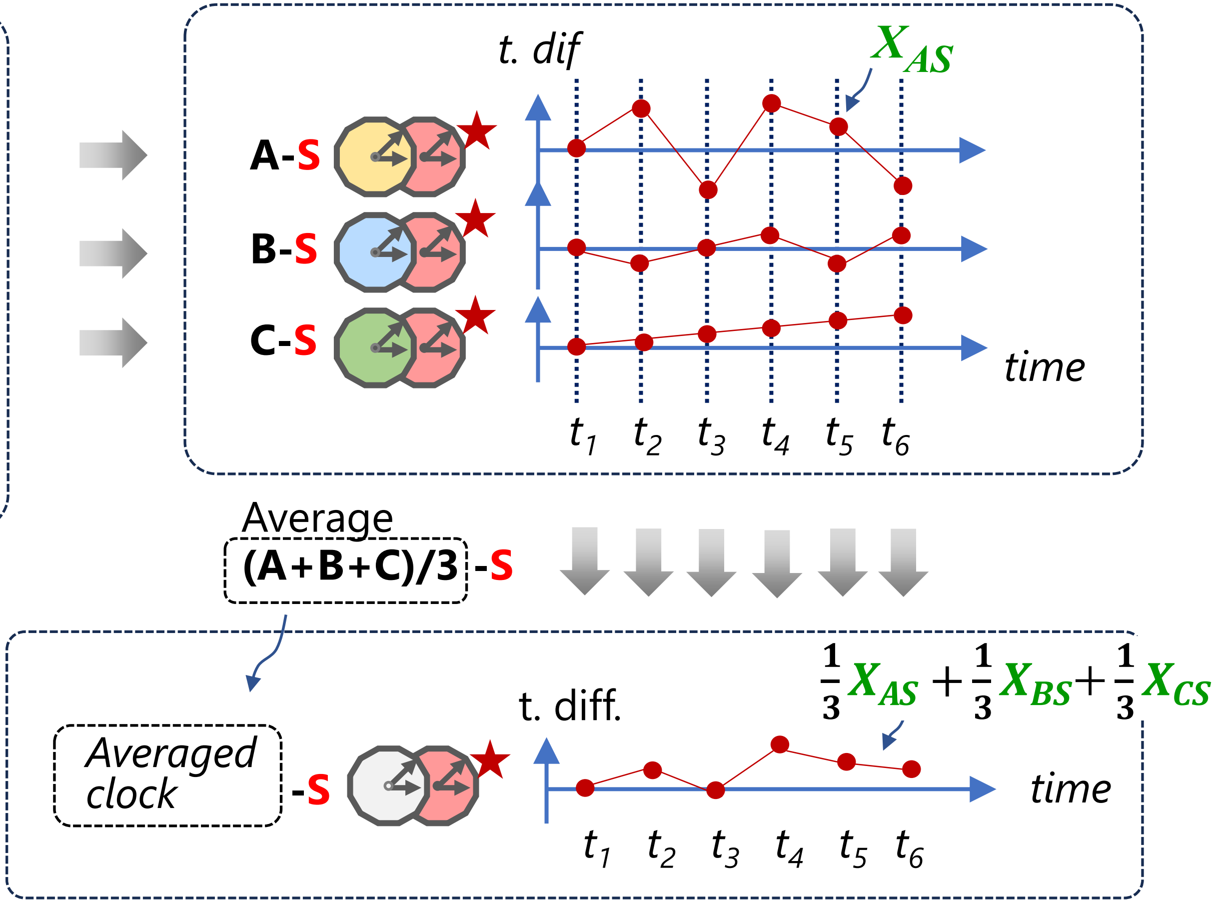
Basic idea

- Averaging the fluctuating time deviation of each clock can make a smooth and stable time scale.

2.1 *TA by basic averaging* : Principle

[1] Averaging the random fluctuations of each clock leads to a stable virtual clock.



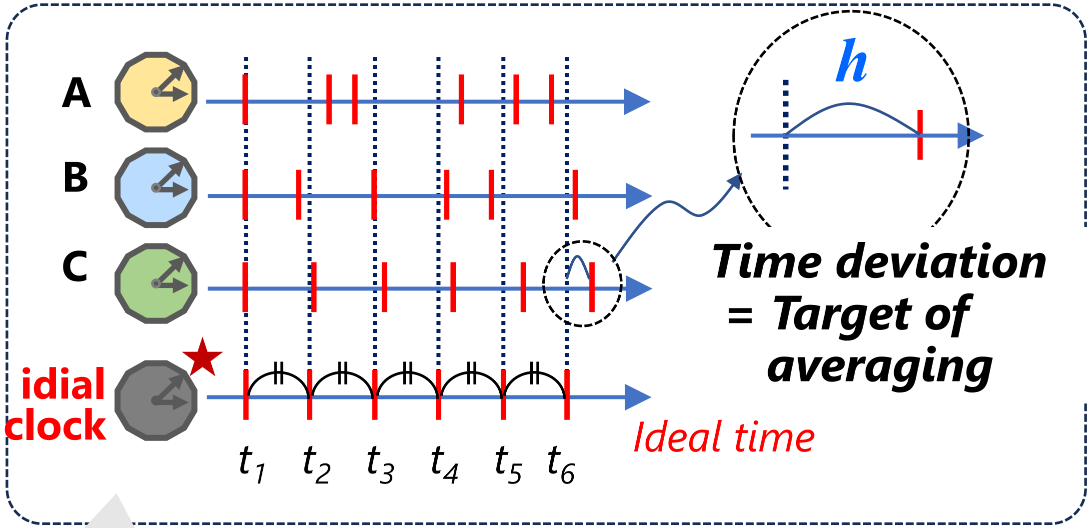


- Reference clock should be as stable as possible!

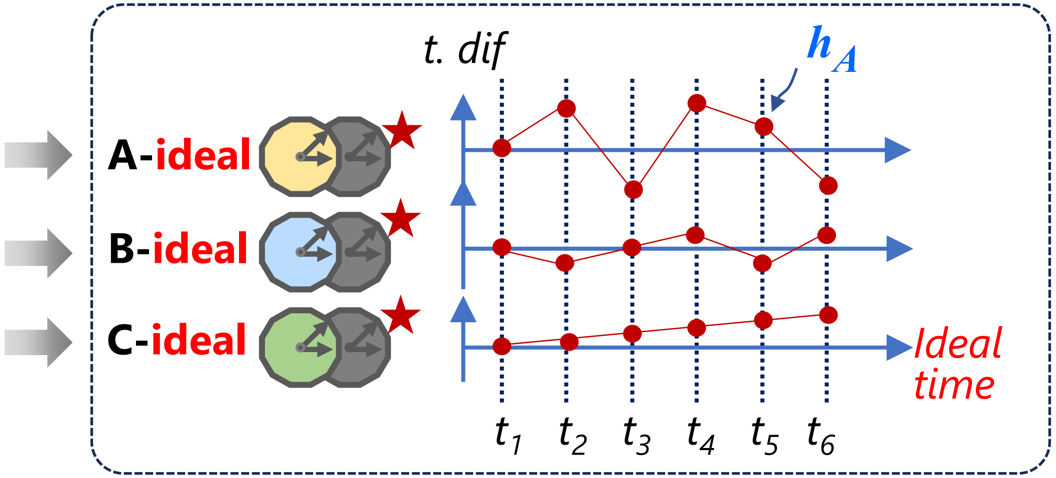
➔ *UTC(k)* or the most stable clock is used.

2.1 *TA by basic averaging* : Formulation

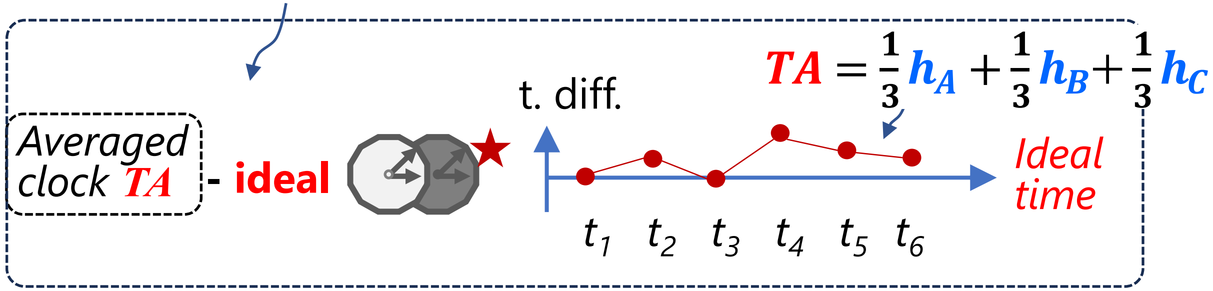
[2] Introduce conceptual values for getting algorithm equations.



Time deviation = Target of averaging



Average
(A+B+C)/3 - **ideal**



$$TA = \frac{1}{3}h_A + \frac{1}{3}h_B + \frac{1}{3}h_C$$

- Introduce the concept of ideal time as the perfectly accurate reference time, and h; time difference from the ideal time.

X_{iS} : Time difference of [*Clock-i* - Ref. clock] → h_i : Time difference of [*Clock-i* - ideal time]

2.1 *TA by basic averaging* : Formulation

■ Formulation of *TA by basic averaging* *(named "TAb")*

Eq. (1)

$$TAb(t) \equiv \sum_{i=1}^{N} w_i \cdot h_i(t)$$

Here

Eq. (2)

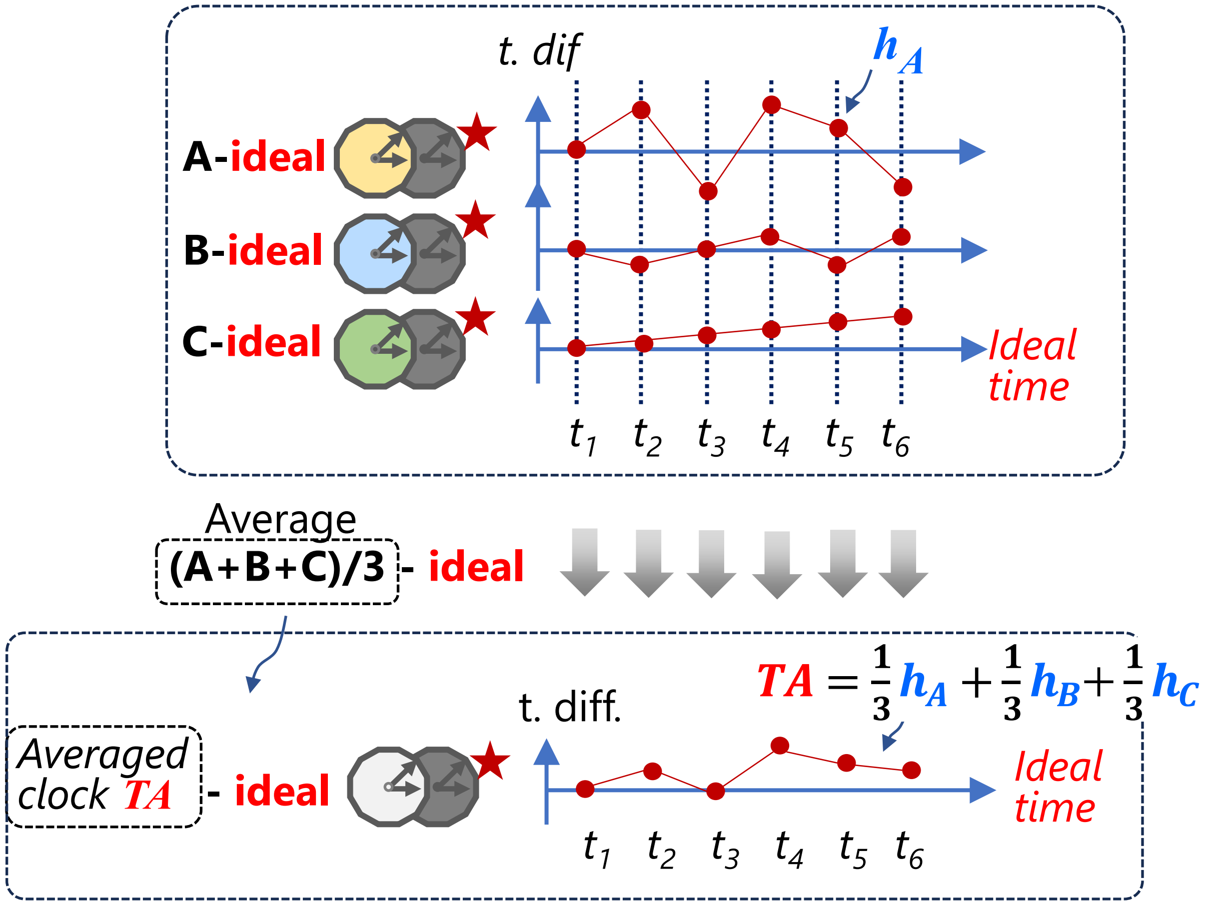
$$\sum_{i=1}^{N} w_i = 1$$

$TAb(t)$: *average atomic time*

$h_i(t)$: *time(phase) difference of Clock* i *from the ideal time*

w_i : *weight of Clock* i

- **Proper weighting** of the atomic clocks **makes the *TA* smoother** than each individual clock.



2.1 *TA by basic averaging* : Notes

- ***TAb* is enough for practical use?**

Consideration points!

  - **Discontinuity tends to occur at a change of ensemble clocks**
    - When the composition of clock ensemble changes, ***TAb*** tends to show a gap. (See p.33.)
  - ***TAb* needs a usable stable reference time for actual calculation**
    - For the calculation, a usable reference time is required instead of the conceptual ideal time. ➔ If there is no stable reference time available, ***TAb*** cannot be stable.

↓

- **“*TA with prediction*” provides a solution!**
  - **It can suppress a negative effect when the ensemble clocks change.**
  - **It can maintain stability by itself without external reference time.**

Contents

2.2 *TA with prediction* : Concept

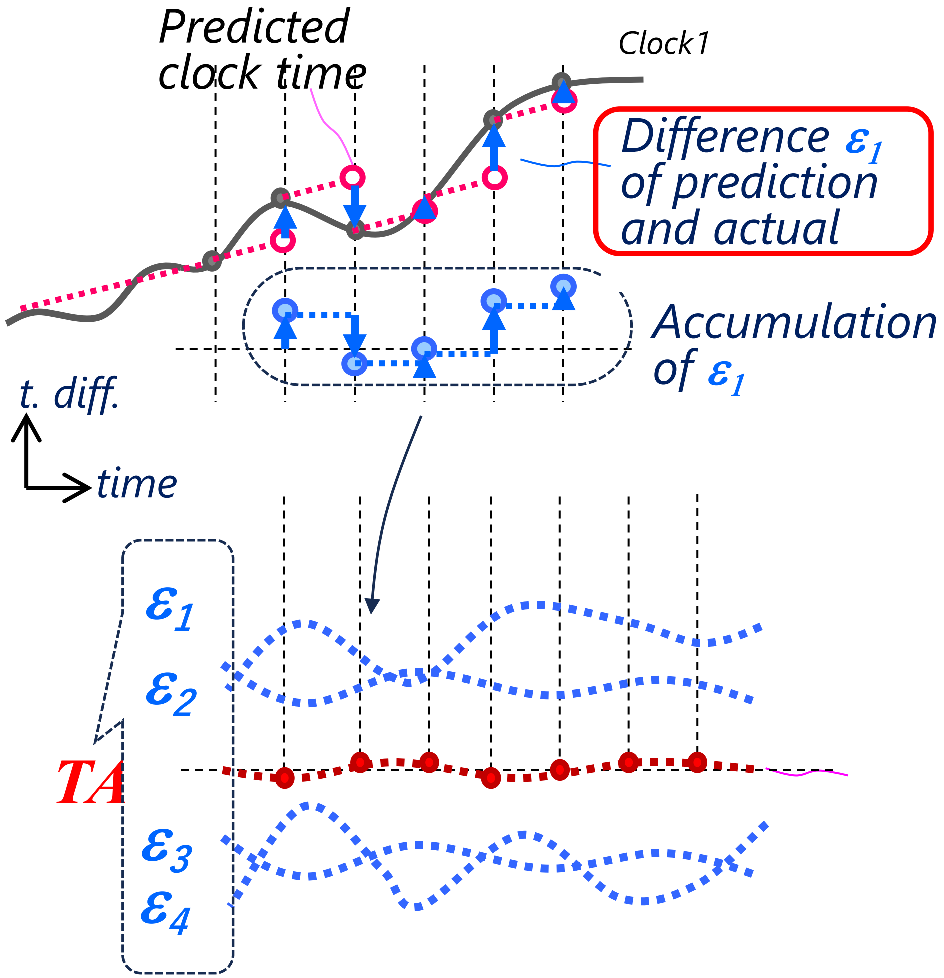
What is *TA with prediction*?

- It is the **weighted average of the residuals,** which **are the differences between the predicted and actual phases of atomic clocks**.

Basic idea

- The purpose of *TA* is to smooth out the random fluctuations of clocks.
- If so, it should also be effective to average un-modeled errors after removing the predictable offset.

※ The time scale "*EAL*", the base of *TAI*, is a kind of the *TA with prediction*.



2.2 *TA with prediction* : Formulation

■ Formulation of the *TA with prediction* *(named "TAp")*

$$TAp(t_1) \equiv \sum_i w_i(t_1) \cdot \{h_i(t_1) \boxed{- \hat{x}_i(t_1)\}}$$ Eq. (3)

w_i: *Weight of Clock-**i***

$$\sum_i w_i(t_1) = 1$$ Eq. (4)

h_i : *Time difference of Clock-**i** vs ideal time*

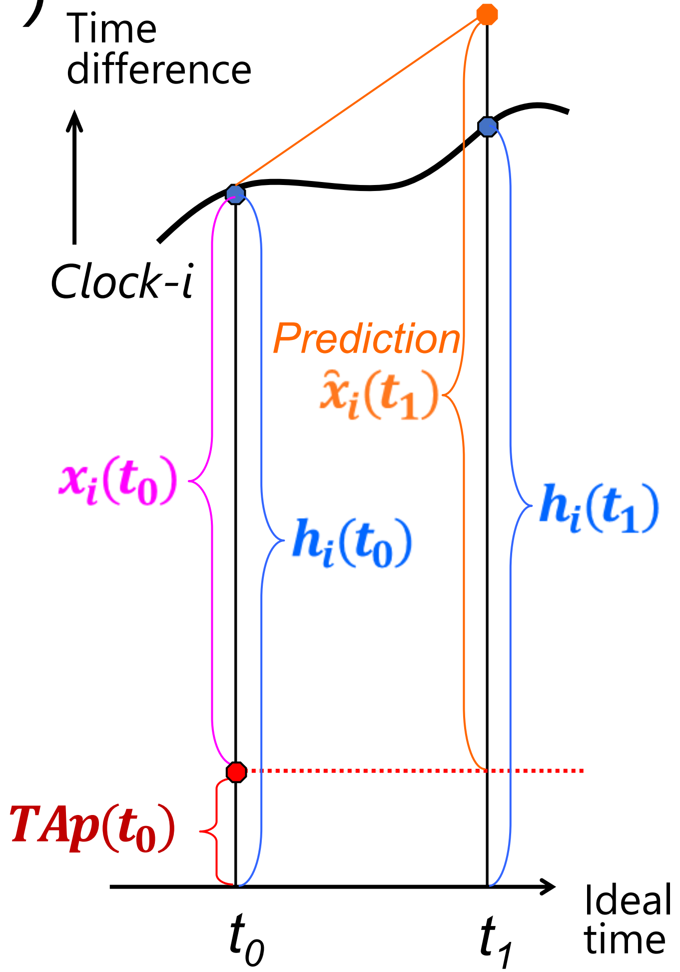
x_i : *Time difference of Clock-**i** vs* TAp

$$x_i(t) \equiv h_i(t) - \underset{\uparrow}{TAp(t)}$$ Eq. (5)

(NOT ideal time) Attention

$\hat{x}_i$: *Prediction of* x_i

$$\hat{x}_i(t_1) = x_i(t_0) + \hat{y}_i(t_0) \cdot (t_1 - t_0)$$ Eq. (6)



2.2 *TA with prediction* : Formulation

■ Confirm the meaning of Eq. (3)

- $TAp(t)$ is the accumulation of $\varepsilon(t)$.

  ε_i : *Difference between predicted and real*

This relation follows the figure to the right:

$$TAp(t_{k-1}) + \hat{x}_i(t_k) = h_i(t_k) + \varepsilon_i(t_k) \dashrightarrow \boxed{TAp(t_{k-1}) - \varepsilon_i(t_k) = h_i(t_k) - \hat{x}_i(t_k)}$$

Equation (3) :

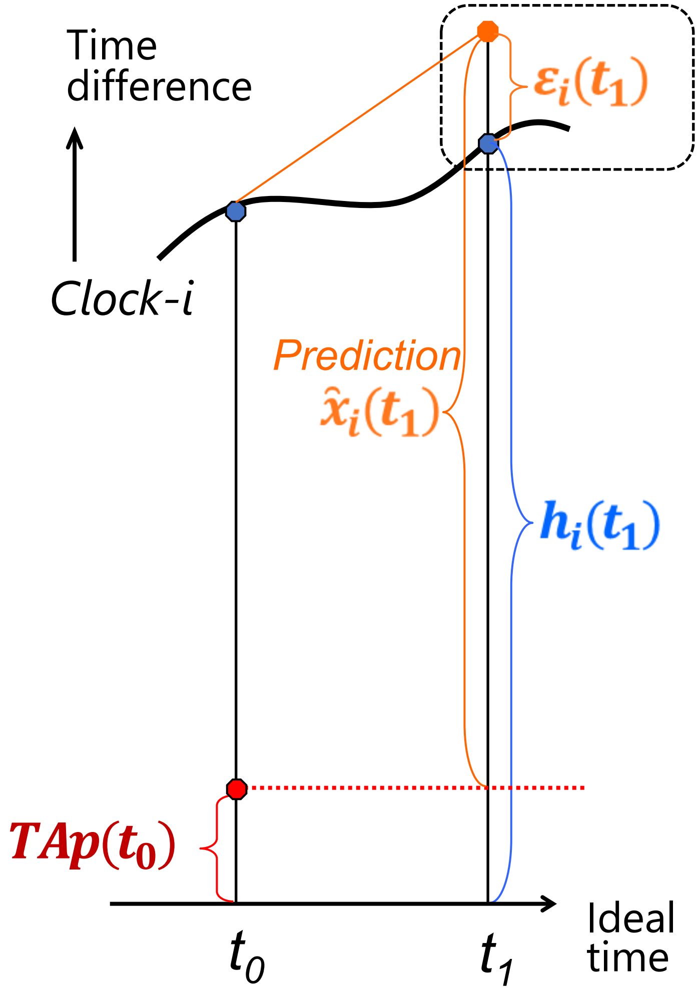
$$TAp(t_k) \equiv \sum_i w_i(t_k) \cdot \{h_i(t_k) - \hat{x}_i(t_k)\} = \sum_i w_i(t_k) \cdot \{TAp(t_{k-1}) - \varepsilon_i(t_k)\}$$

$$= TAp(t_{k-1}) - \sum_i w_i(t_k)\varepsilon_i(t_k)$$

$$= TAp(t_{k-2}) - \sum_i w_i(t_{k-1})\varepsilon_i(t_{k-1}) - \sum_i w_i(t_k)\varepsilon_i(t_k)$$

$$\vdots$$

$$\therefore TAp(t_k) = TAp(t_0) - \sum_{j=1,k} \sum_i w_i(t_j)\varepsilon_i(t_j) \qquad \boxed{\text{Eq. (7)}}$$



2.2 *TA with prediction* : Merits

■ Merits of *TA with prediction*

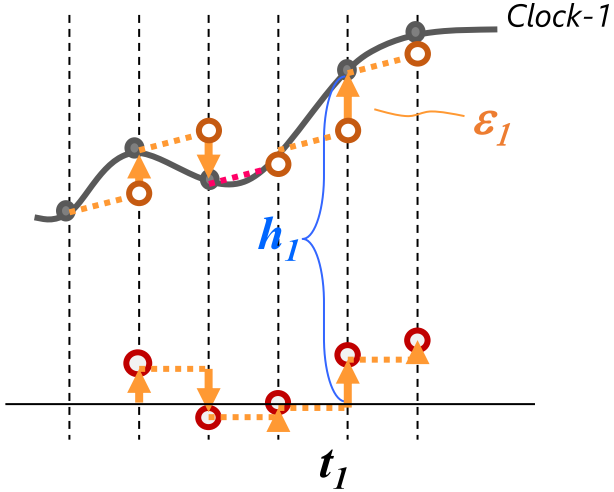
- **Continuity**
  - Even if the composition of the clock ensemble changes, the impact is smaller than that of ***TAb***.

  ➔ Continuity of the time scale can be maintained.

- **No need of external reference time**
  - As ***TAp*** refers to its own past values, no external reference time is required for the ***TA*** calculation (except at the starting of the calculation).

※ If accuracy is required, ***TA*** requires calibration with an external reference.

Let's see the details of the calculation!



- If *Clock-1* exits the ensemble after t_1, a large gap will occur in ***TAb*** due to the offset h_1.

$$TAb(t_1) \equiv \sum_i w_i(t_1) \cdot h_i(t_1)$$

- In the case of ***TAp***, however, the effect can be smaller because the residual ε_1 is smaller than h_1.

$$TAp(t_1) \equiv \sum_i w_i(t_1) \cdot \{h_i(t_1) - \hat{x}_i(t_1)\}$$

Bureau International des Poids et Mesures

2.2 *TA with prediction* : Calculation

- How can we calculate TAp ?

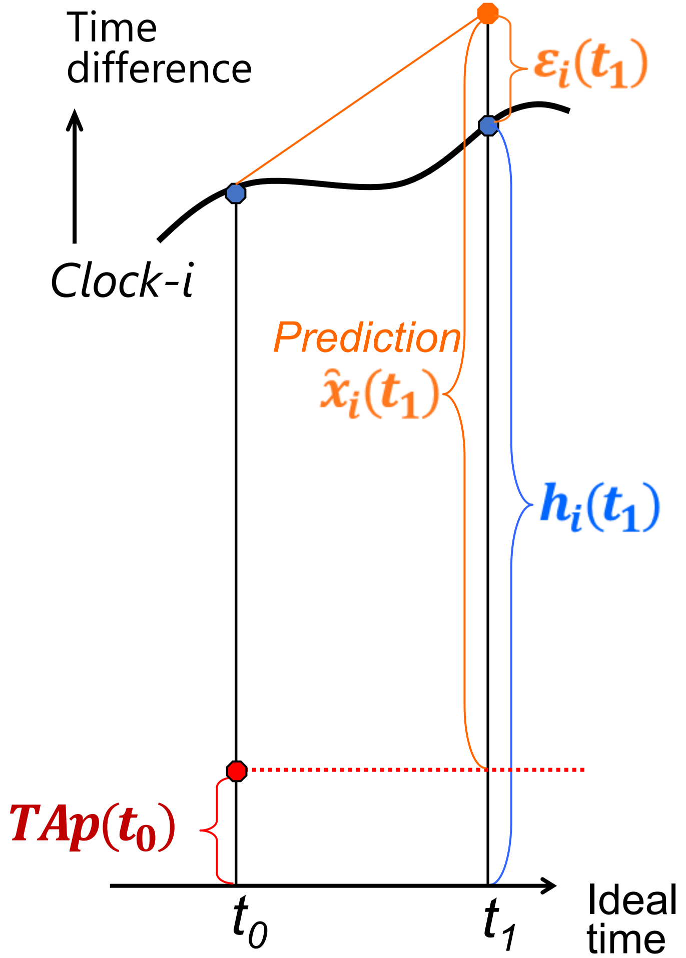
$$TAp(t_1) \equiv \sum_i w_i(t_1) \cdot \{h_i(t_1) - \hat{x}_i(t_1)\}$$ Eq. (3)

- In this expression,
  the numerical value of $TAp(t)$ cannot be obtained because the value of $h_i(t)$ is unknown.

TAp = 123.456…
Impossible expression !

Since the Ideal time is conceptual, $h_i(t)$ cannot be obtained as a numerical value.

- **What should we do to calculate TAp?**



2.2 *TA with prediction* : Calculation

- **Let's introduce the measurement value, and transform the equation as calculable form!**

1. $X_{is}(t)$ is actually measured time difference between two clocks.

$$X_{is}(t) \equiv h_i(t) - h_s(t) \quad \text{Eq. (8)}$$

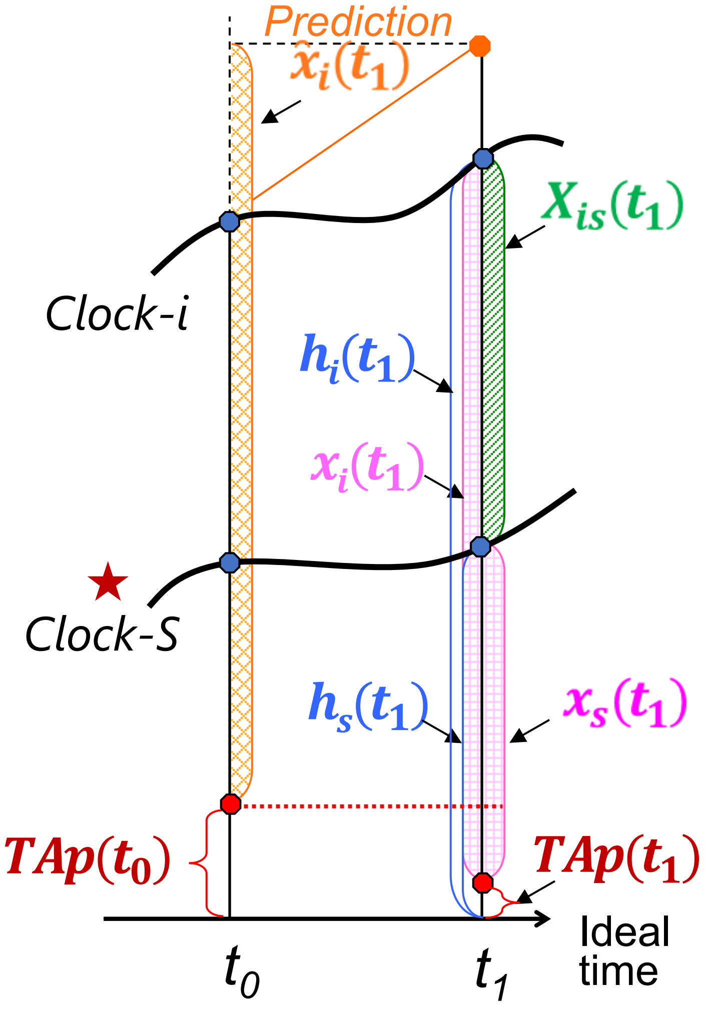
2. By using Eq.(8) and the $x_i(t)$ defined as Eq.(5), Eq.(3) is modified to become Eq. (9).

$$x_i(t) \equiv h_i(t) - TAp(t) \quad \text{Eq. (5)}$$

$$x_s(t) = \sum_i w_i(t)\cdot\{\hat{x}_i(t) - X_{is}(t)\} \quad \text{Eq. (9)}$$

Eq. (3) → $TAp(t)\boxed{-h_s(t)} \equiv \sum_i w_i(t)\cdot\{h_i(t)\boxed{-h_s(t)} - \hat{x}_i(t)\}$

→ $\boxed{h_s(t) - TAp(t)} \equiv \sum_i w_i(t)\cdot\{\hat{x}_i(t) - X_{is}(t)_s\}$ → Eq.(9)



2.2 *TA with prediction* : Calculation

- How can we get TAp? ➜ "x_i" represents TAp.

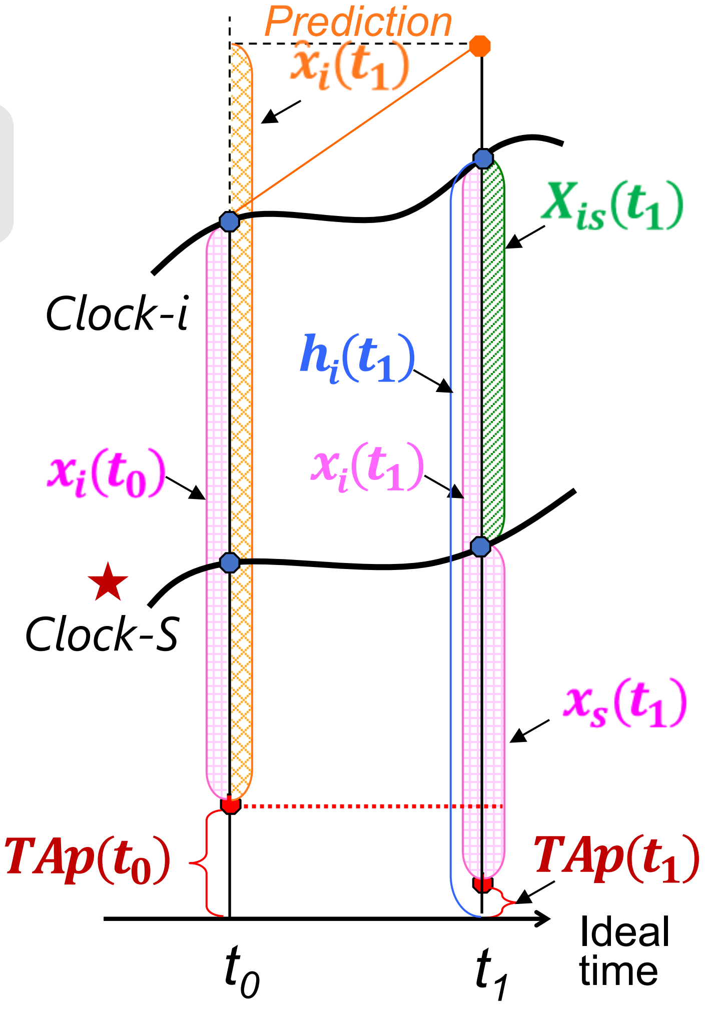
$$x_i(t) \equiv h_i(t) - TAp(t)$$ Eq. (5)

$$x_s(t) = \sum_i w_i(t) \cdot \{\hat{x}_i(t) - X_{is}(t)\}$$ Eq. (9)

Interpretation of Eq.(9) is described in Appendix-1.

Important

- **TAp can be defined as "the time shifted by x_i from the time of *Clock-i* ".**
- $x_i(t)$ can be obtaind from the measurement values and calculable values.
- **The algorithm of TAp is the procedure that computes the time series of $x(t)$.**



2.2 *TA with prediction* : Calculation

■ Practical way to calculate x

Predicted rate $\hat{y}_i$:
See p.28-29.

$$\hat{x}_i(t_k) = x_i(t_{k-1}) + \hat{y}_i(t_{k-1}) \cdot (t_k - t_{k-1})$$ Eq. (10)

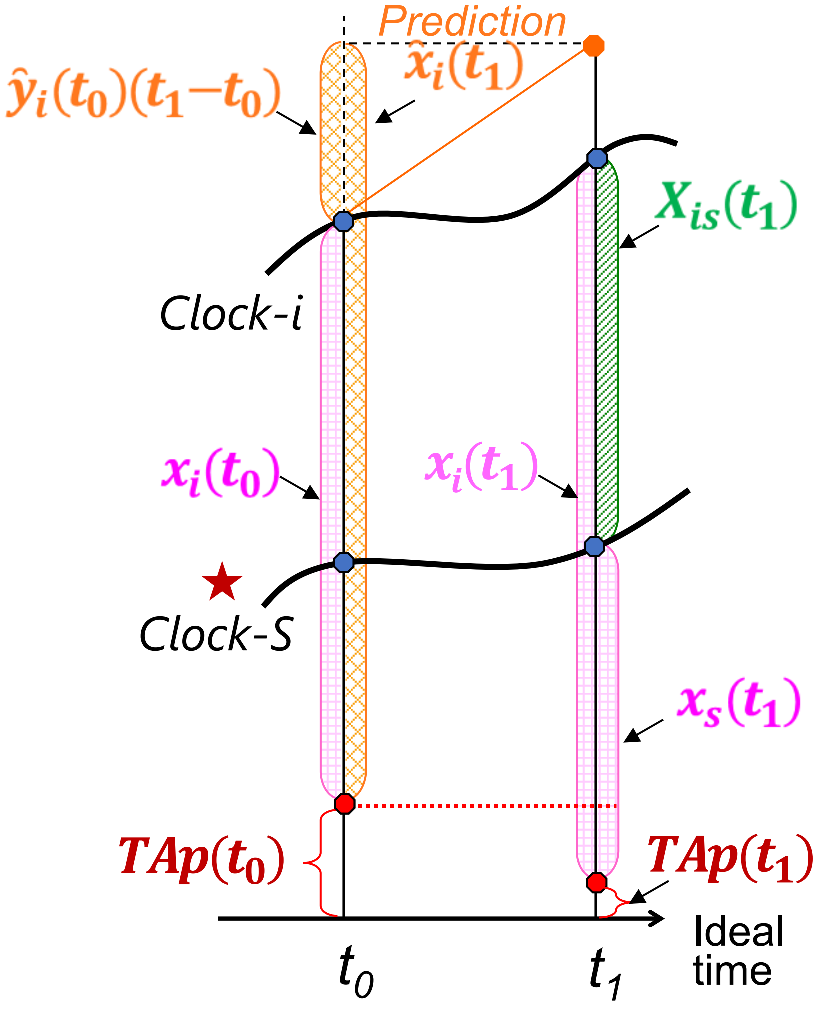
$$x_s(t_k) = \sum_i w_i(t_k)\{\hat{x}_i(t_k) - X_{is}(t_k)\}$$ Eq. (9)

$$x_i(t_k) = x_s(t_k) + X_{is}(t_k)$$ Eq. (11)

🟩 *:Measured* 🟧 *:Predicted* 🟪 *:Output*

🟦 *:Calculated from previous x_i*

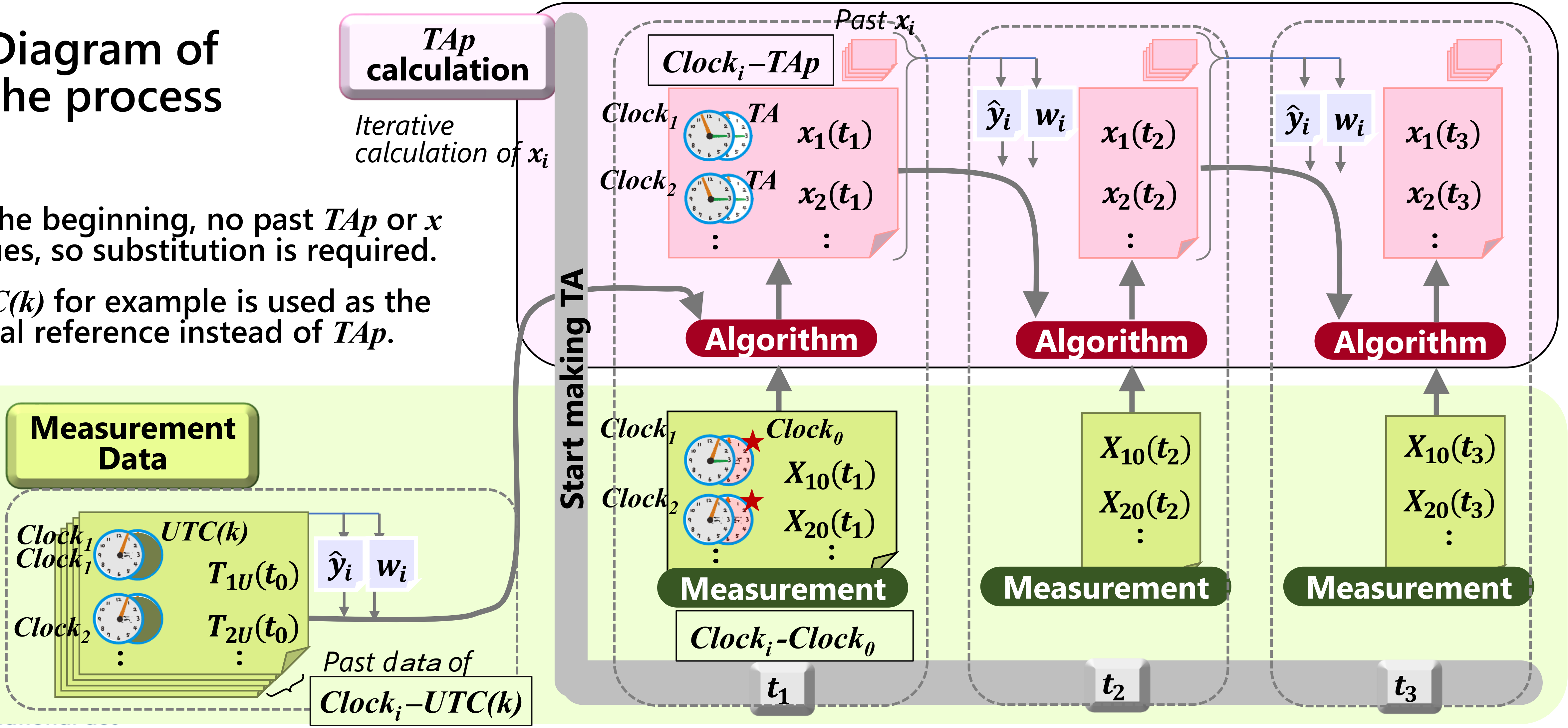
- Iterative calculation provides a time series of $x_i(t_k)$.



2.2 *TA with prediction* : Diagram of the process

■ Diagram of the process

- At the beginning, no past *TAp* or *x* values, so substitution is required.
- *UTC(k)* for example is used as the initial reference instead of *TAp*.



2.2 *TA with prediction* : Important parameters

■ $\hat{y}_i$ (predicted clock rate) and w_i (weight)

- **Predicted clock rate $\hat{y}_i$ and weight w_i are calculated from the past x_i values.**

➡ *TAp* itself becomes the reference.

Attention!

- Proper calculation of de-trending and weighting depends on the situation.

  (Properties of clocks, priorities in the design of *TAp*...)

- Variation of de-trending ➔ p.29
- Optimizing of weighting ➔ p.30

Predicted clock rate

$$\hat{y}_i(t_k) = \frac{x_i(t_k) - x_i(t_k - T_{rate})}{T_{rate}} \quad \text{Eq. (12)}$$

Predicted clock rate $\hat{y}_i$ is estimated at the latest interval T_{rate}.

weight

$w_i(t) \propto$ stability evaluated from past x_i values

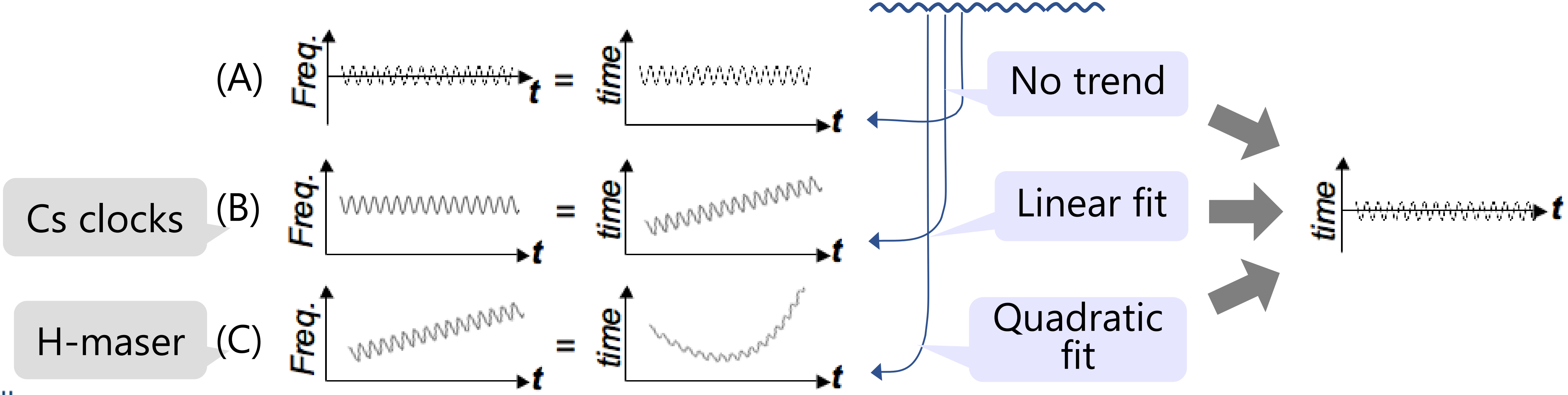
$$\sum_i w_i(t) = 1$$

2.2 *TA with prediction* : Important parameters

※ Variation of de-trending

- The optimal way to detrend depends on the clock's behavior.

$$\hat{x}_i(t_k) = x_i(t_{k-1}) + \text{detrend term}$$



2.2 *TA with prediction* : Important parameters

※ Optimizing of weighting

- **Which variance is chosen?**

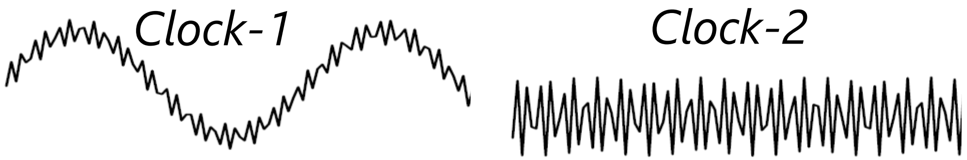
$$w_i(t) \propto \frac{1}{\sigma(t)_i^2}$$

- classical variance,
- Allan variance,
- predicted error² ,,

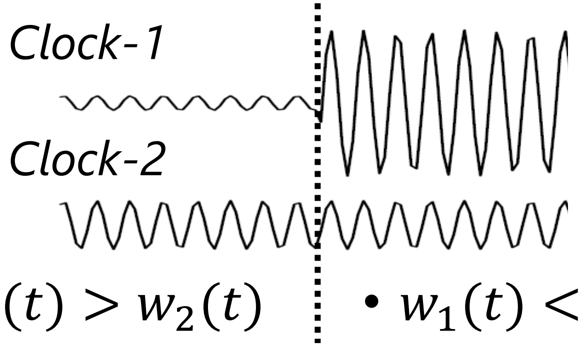
- **Maximum limit for $\boldsymbol{w_i(t)}$**

- Setting a maximum limit can prevent too much concentration on particular clocks.

- **Proper weighting depends on purpose and clock's behavior.**



- For Short-term $w_1(t) > w_2(t)$
- For long-term $w_1(t) < w_2(t)$



- $w_1(t) > w_2(t)$
- $w_1(t) < w_2(t)$

2.2 *TA with prediction* : Important parameters

$\hat{y}_i$ (predicted clock rate) and w_i (weight)

- In this way, the best setting of the parameters is not unique and will vary depending on the situation.

- Let's try out various cases using the tutorial Python program!

Predicted clock rate

$$\hat{y}_i(t_k) = \frac{x_i(t_k) - x_i(t_k - T_{rate})}{T_{rate}} \quad \text{Eq. (12)}$$

Predicted clock rate $\hat{y}_i$ is estimated at the latest interval T_{rate}.

weight

$w_i(t) \propto$ stability evaluated from past x_i values

$$\sum_i w_i(t) = 1$$

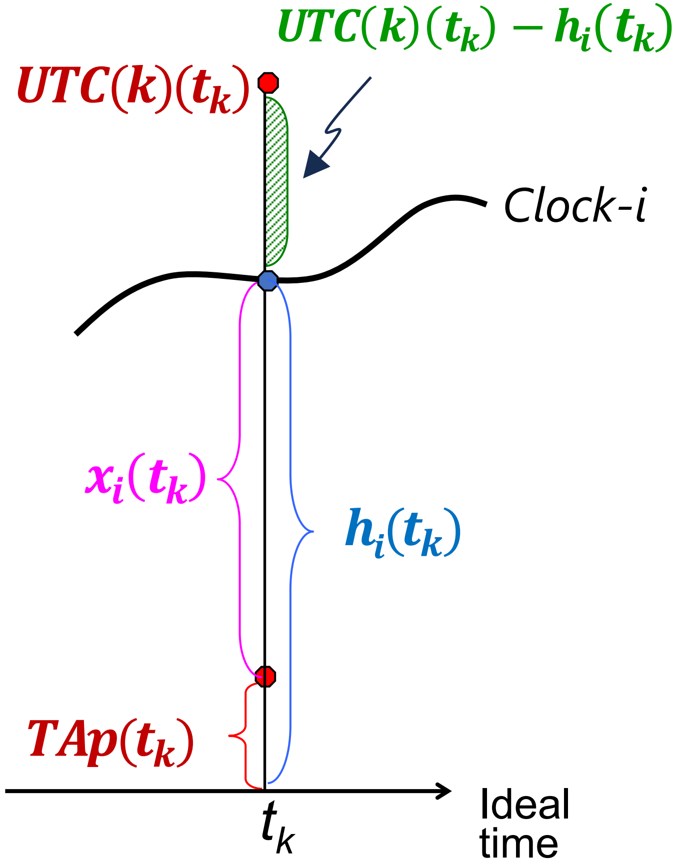
2.2 *TA with prediction* : Evaluation

- We know that we can indirectly obtain TAp through x_i… But, How can we evaluate TAp itself ?
  - For this purpose, the time difference against an external reference time, such as ***UTC(k)***, is usually used.

$$UTC(k)(t_k) - TAp(t_k)$$

$$= \underbrace{[UTC(k)(t_k) - h_i(t_k)]}_{\text{Given by measurement}} + \underbrace{[h_i(t_k) - TAp(t_k)]}_{\text{Calculated } x_i} \quad \boxed{\text{Eq. (13)}}$$

UTC(k) -TAp becomes calculable by dividing it into these two terms.



2.2 *TA with prediction* : Simulation

※ **Examples of**
TAb-UTC(k)* vs. *TAp-UTC(k)

- Fig.1 is ***TAb*** and Fig.2 is ***TAp***.
- Both ***TA***s were the average of *Clock-1*, *Clock-5*, and *Clock-7* with fixed equal weights.
- (a) is the time difference vs ***UTC(k)***, and (b) is the Allan deviation of (a).
- In ***TAp***, each clock rate was daily calculated from the latest 30 days.

- In Fig.2, there is **no apparent jump in *TAp*** when *Clock-1* data are missing.
- Allan deviation of ***TAp*** shows better stability than each clock. (The stability of ***TAb*** has degraded by the jump when *Clock-3* is missing.)

Fig.1 : ***TAb***

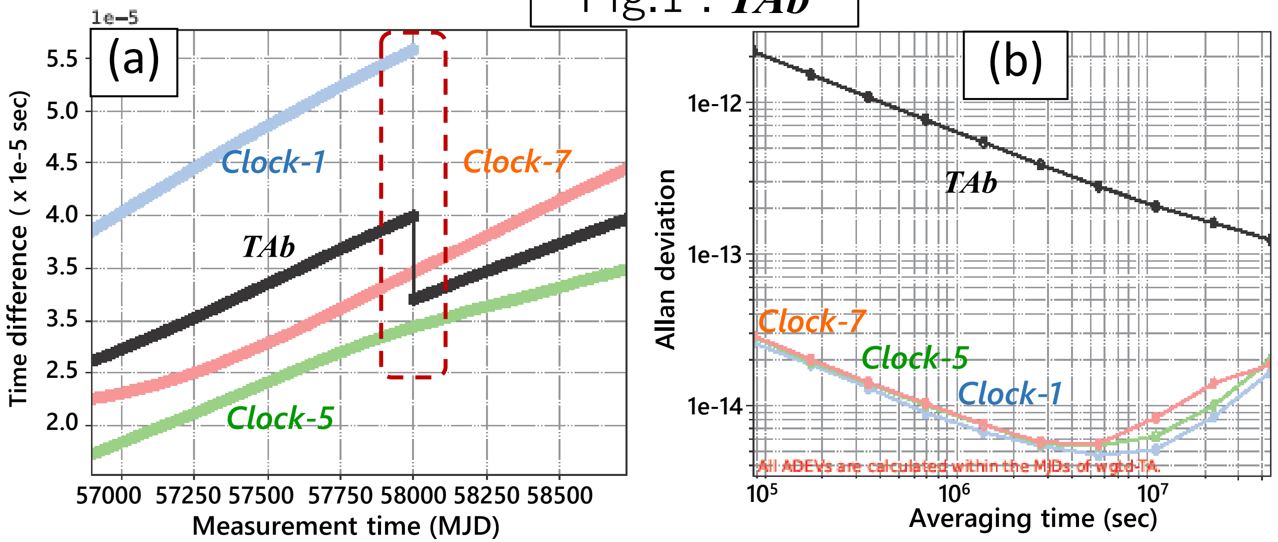
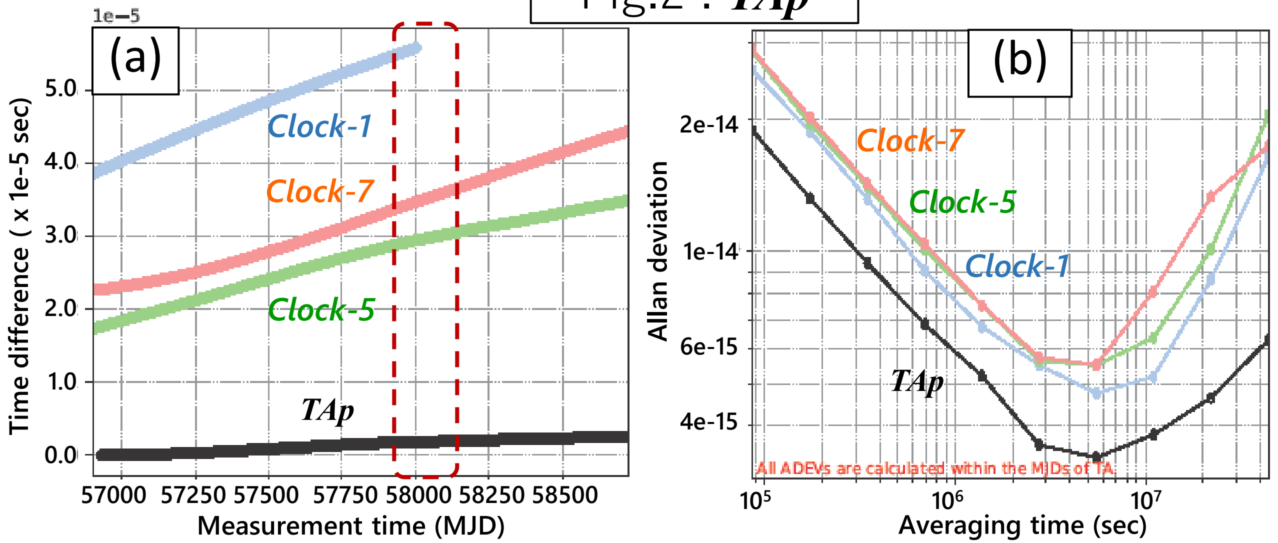


Fig.2 : ***TAp***



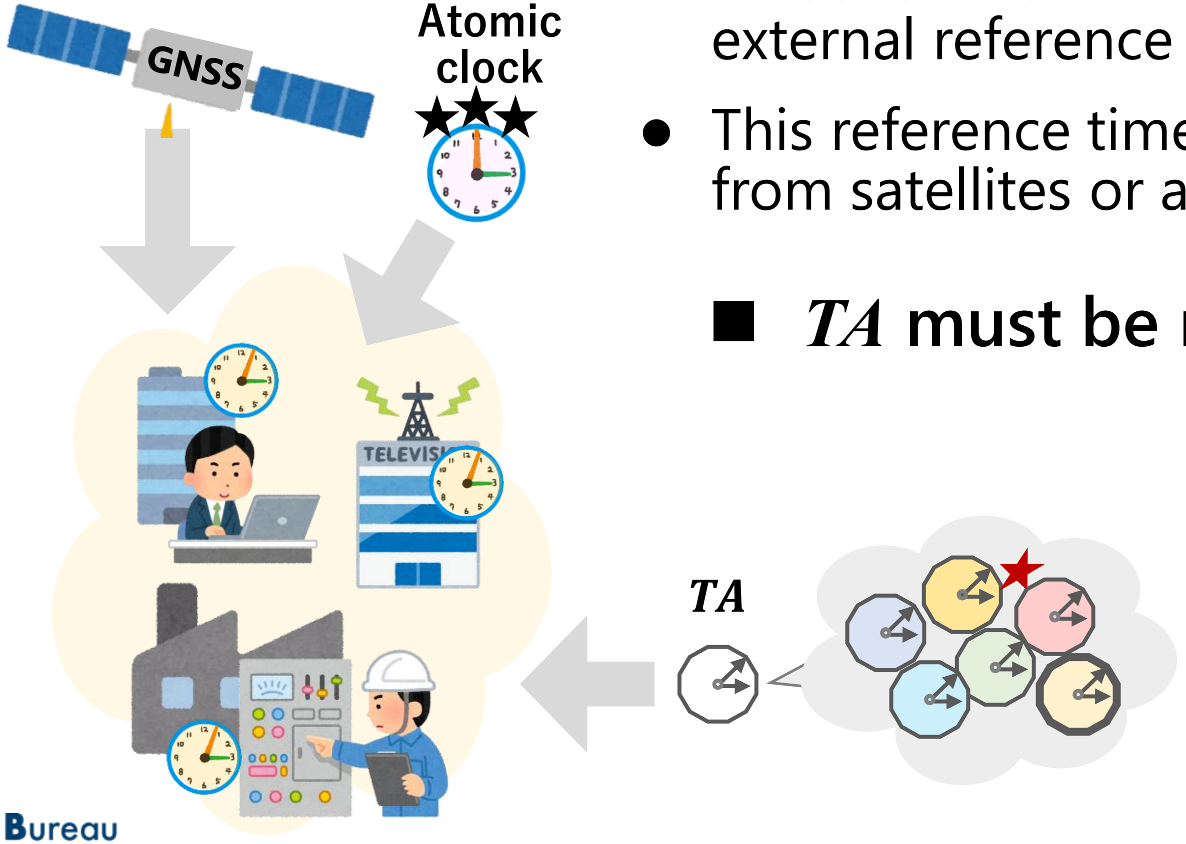
2.2 *TA with prediction* : Notes

- **The reference time for the calculation is *TAp* itself.**
  - This is reasonable because we consider the most stable time scale to be *TAp*.
  - However, it also suggests a risk of self-divergence within the process.
- ***TAp* cannot be obtained in real time.**
  - The time we can obtain $TAp(t_k)$ should always be later than t_k .
- **There is a hidden assumption in the calculation.**
  - The assumption is that --- Current clock status can be roughly predicted based on its past behavior, and that the fluctuations after excluding this predicted trend can be smoothed out by averaging.
  - There is no guarantee that *TA* will be stable when this assumption does not hold.
- **Initial phase & frequency offset of *TA* compared to the reference:**
  - Since the initial offsets do not affect the stability, they can be set arbitrarily.
  - They are also preserved throughout the calculation.

Contents

3.1 How to implement *TA* for practical use

■ Actual clocks used in daily lives usually need an external reference time.



- Actual clocks in daily life tend to drift, so adjustments to an external reference time are necessary.
- This reference time can be obtained using the ***UTC*** time code from satellites or a stable atomic clock. ***TA*** is also an option.

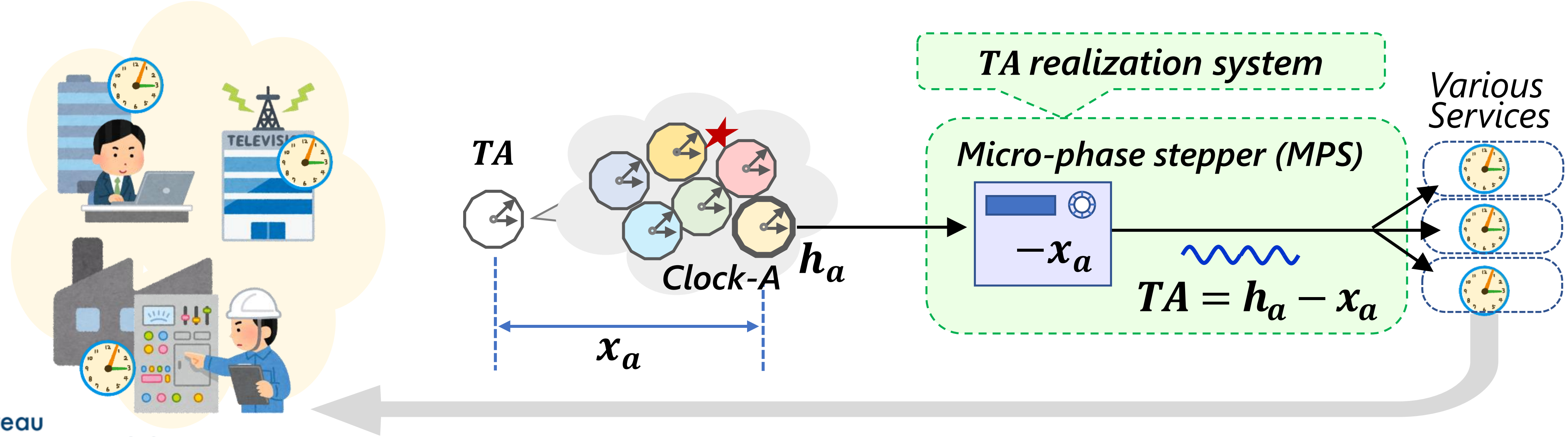
■ *TA* must be realized for its practical use.

- ***TA*** has many advantages (robustness, long-term stability) and shows potential as the reference time.
- ***TA*** itself is numerical data, however, so physically realization of ***TA*** is required for practical use.

3.1 How to implement *TA* for practical use

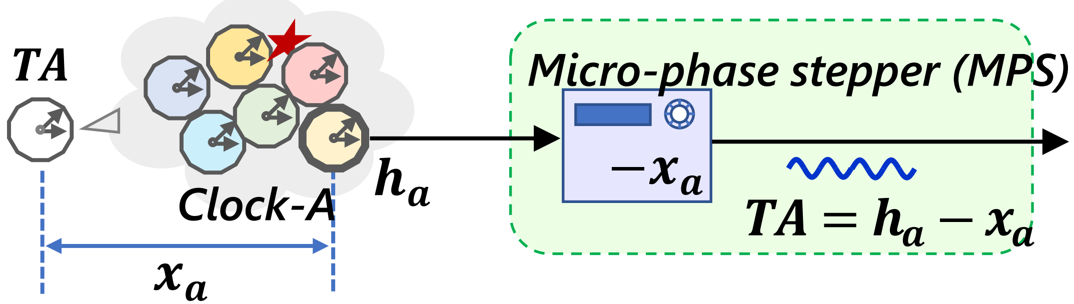
Outline of *TA* realization:

- Prepare a **micro-phase stepper** ***(MPS)*** which shifts the external input phase/frequency, and connect this *MPS* to *Clock-A* (🡐 *component of the TA calculation)*.
- *Clock-A* has the time offset $\boldsymbol{x_a}$ vs ***TA*.** This value is provided in the ***TA*** calculation.
- **If *MPS* compensates for this offset $\boldsymbol{x_a}$, *TA* is realized as the output signal of *MPS*.**



3.1 How to implement *TA* for practical use

Principle :



- **Fig.1 is a principle.**
  The adjustment is done at t_1, and *MPS* only compensates for the offset $x_a(t_1)$.

- **Fig.2 is an actual process.**
  <u>*MPS* aims to synchronize at t_2</u>. For that, *MPS* compensates not only x_a but also the time shift due to the rate of *Clock-A* vs *TA*.

Not realistic, because

1. There is usually a time lag in adjustments, and
2. moderate adjustments are better for stability.

Fig. 1

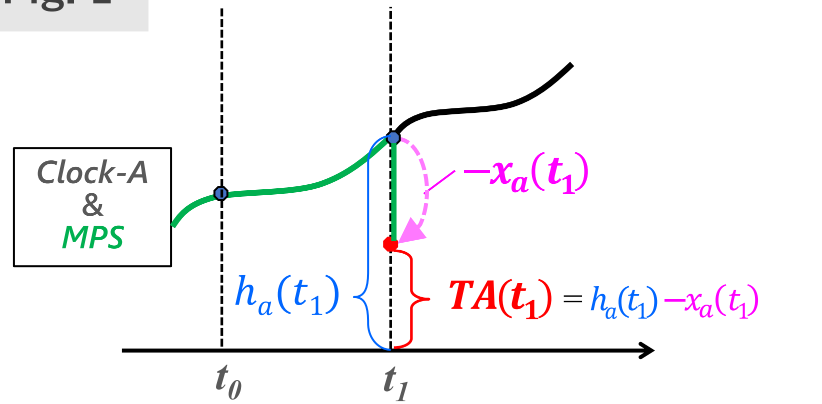
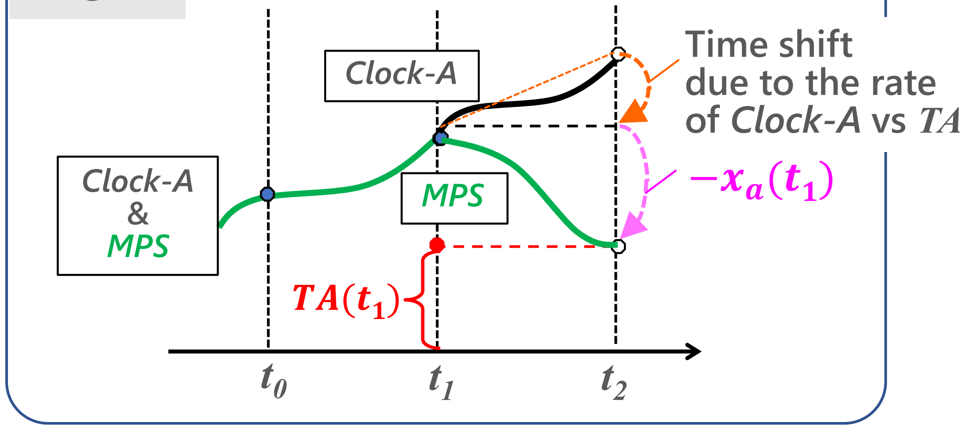


Fig. 2



3.1 How to implement *TA* for practical use

Steering of *MPS*

- The first given frequency : $\Delta y(t_1)$

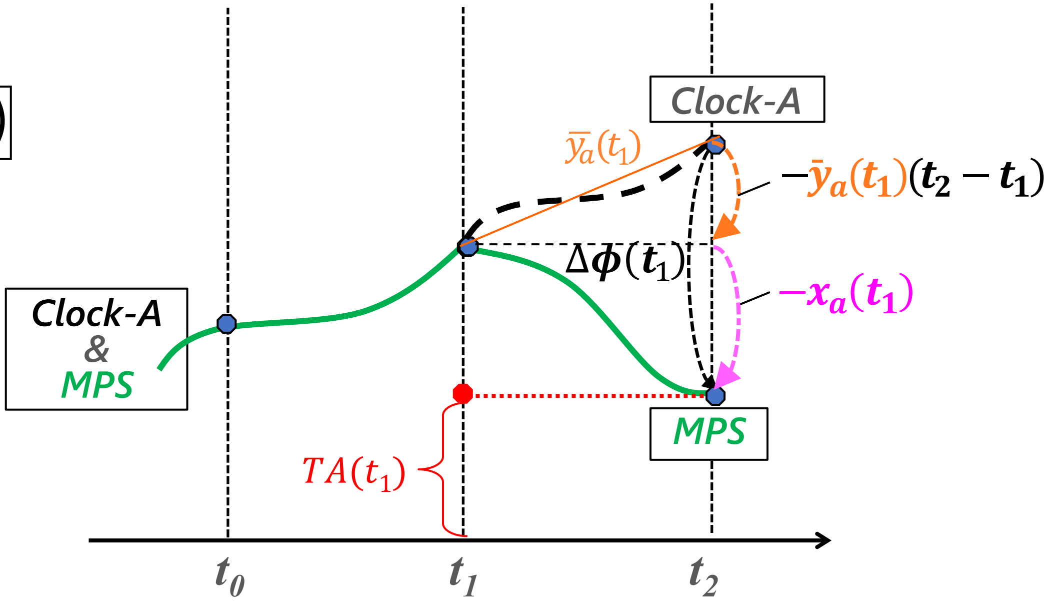
$$\Delta y(t_1) = \frac{\boxed{-\Delta\phi(t_1)}}{t_2 - t_1} = -\left(\frac{x_a(t_1)}{t_2 - t_1} + \bar{y}_a(t_1)\right) \quad \boxed{\text{Eq. (14a)}}$$

Such a frequency adjustment to trace *TA* is called "***Steering***".

$$\boxed{\Delta\phi(t_1) = \underbrace{x_a(t_1)}_{\text{Compensate for time difference}} + \underbrace{\bar{y}_a(t_1)\cdot(t_2 - t_1)}_{\text{Compensate for frequency difference}}} \quad \boxed{\text{Eq. (14b)}}$$

※ $\bar{y}_a(t_1) = \{x_a(t_1) - x_a(t_1 - T)\}/T$

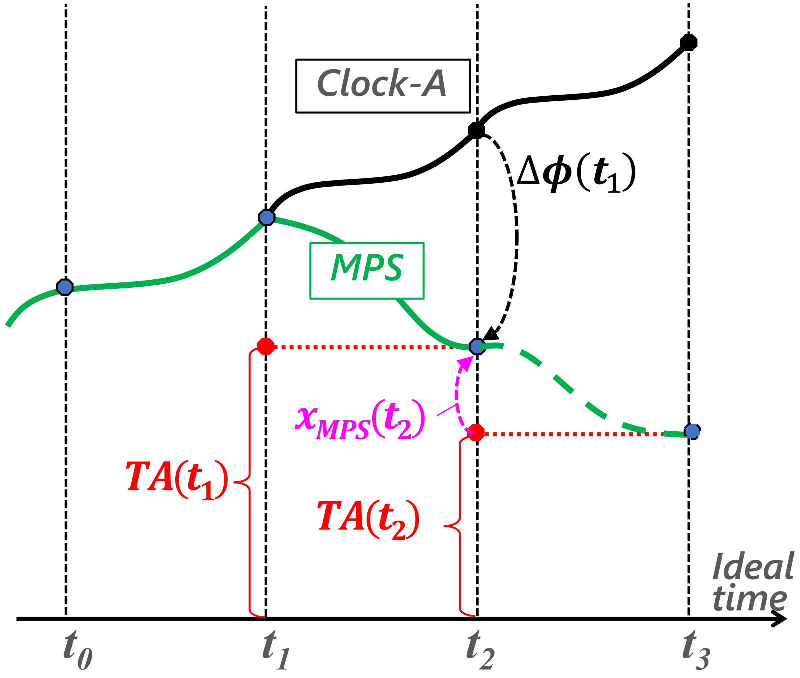
※ Proper prediction of the time difference at a future time t_2 is so important.



3.1 How to implement *TA* for practical use

Continuous steering of *MPS*

- After the first adjustment, *TA* is realized as the output of *MPS*.
- The target of the next steering at t_2 is to synchronize *MPS* with *TA* at t_3.



3.1 How to implement *TA* for practical use

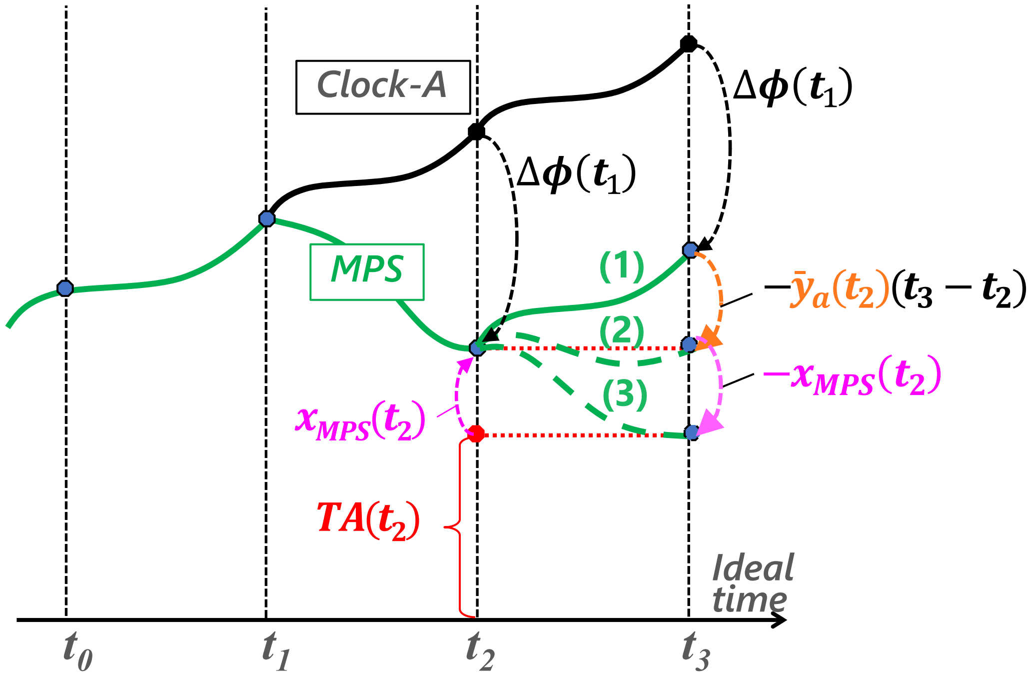
■ Continuous steering of *MPS*

- Three cases with different adjustment at t_2 :
  1. (1) The case with no adjustments
  2. (2) The case with only an adjustment for the shift due to the rate of *clock-A*
  3. (3) The desired case:

$$\Delta y(t_2) = \frac{-\Delta\phi(t_2)}{t_3 - t_2} = -\left(\frac{\boxed{x_{MPS}(t_2)}}{t_3 - t_2} + \bar{y}_a(t_2)\right)$$

$$\Delta\phi(t_2) = x_{MPS}(t_2) + \bar{y}_a(t_2)\cdot(t_3 - t_2)$$

$$x_{MPS}(t_2) = x_a(t_2) - X_{a\,MPS}(t_2)$$



※ Generalization are in *Appendix-2*.

※ Confirm the adjustment method of the *MPS*!

- If adjustment frequency is added to the external reference frequency of *MPS*, $\Delta y(t_k)$ should be given.
- If adjustment frequency is added to the current frequency of *MPS*, $\Delta y(t_k)$- $\Delta y(t_{k-1})$ should be given.

3.1 How to implement *TA* for practical use

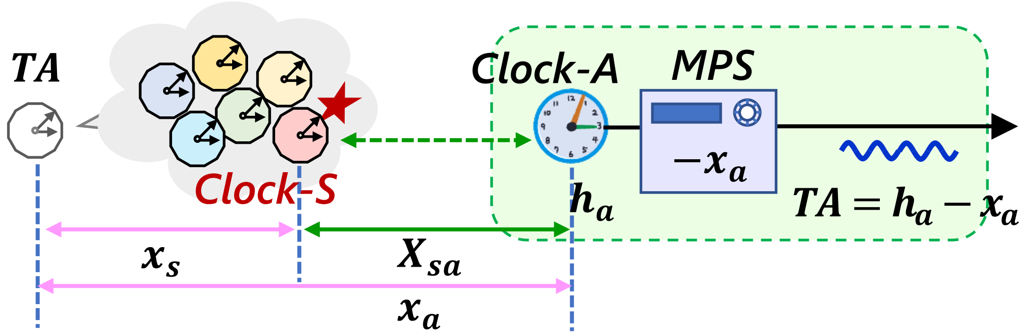
■ Application : The case that *Clock-A* is not included in the *TA* calculation

- **If *clock-A* can be regularly compared with a clock in the *TA* ensemble, *MPS* can output the realized *TA*.**

  - The steering equations (Eq.14) can be easily modified using the following relation.

$$x_a(t_k) = x_s(t_k) - X_{sa}(t_k)$$

  - In order to compare *clock-A* with the *TA* component clocks correctly, simultaneous measurement of all clocks is required.



Nice application

- If a clock with high short-term stability (e.g. *H-maser*) is used as *clock-A*, the output of *MPS* can achieve both the short-term stability of *clock-A* and the long-term stability of *TA*.

3.1 How to implement *TA* for practical use

Notes for the Steering:

- **Steering is recommended to be applied to *MPS*, not directly to the clock.**
  - If the adjustment is applied to *clock-A* directly, the prediction of the true rate of *clock-A* becomes difficult.

- **Steering is also a kind of disturbance.**
  - Stronger adjustments can lead to faster synchronization, but they also cause larger frequency noise.
  - Proper steering should be adopted depending on the purpose and situation.

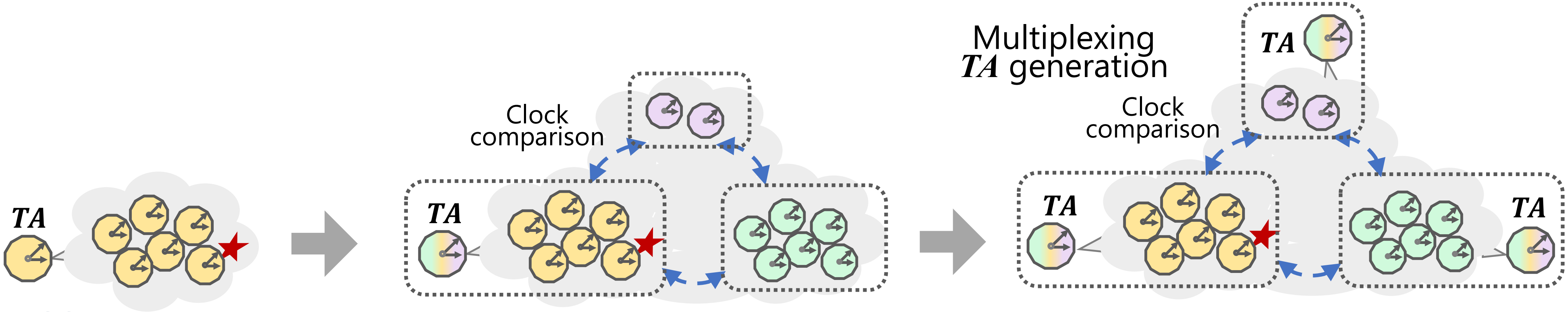
- **Simple method is enough in some cases.**
  - Quadratic fitting is typically appropriate if an *H-maser* is used as *clock-A*. However, a simple linear fitting is enough if we adopt a short-term prediction interval.

Contents

3.2 Application ~ Multiplexing of *TA* operation

■ Concept of multiplexing *TA* generation

- *TA* calculation requires the comparison data between clocks. Basically, the clocks installed in the same place are used for this comparison.
- However, clocks located at a distant station can also join the *TA* calculation, if such a mutual comparison is feasible.
- This condition is the same for any station, that is, every station can calculate *TA* if mutual clock comparison data can be shared each other.



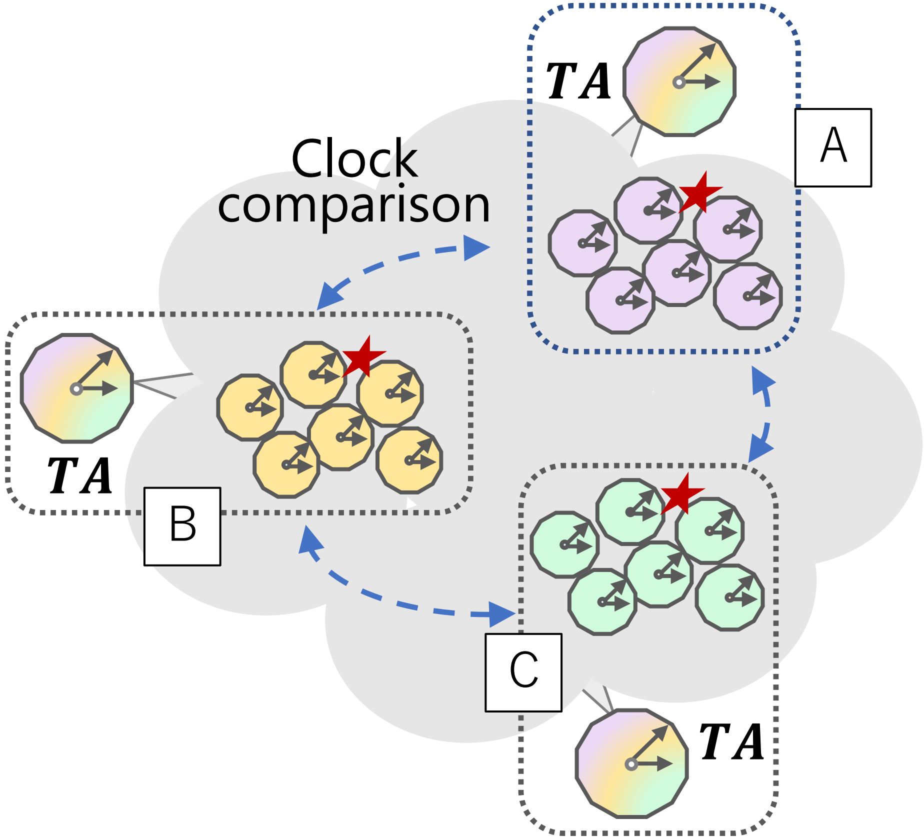
3.2 Application ~ Multiplexing of *TA* operation

■ Operation process :

1. Each station compares its clock with other clocks inside and outside.
2. All stations share the common clock data by exchanging the comparison data with each other.
3. Each station independently calculates *TA* using all the clock data.
4. All *TA*s are expected to be of a similar quality. They are regularly compared to each other for monitoring purpose.

※ Each *TA* is not exactly the same because the comparison data with remote clocks includes transfer noise.

※ Three remote stations linked with satellites or optical fiber.

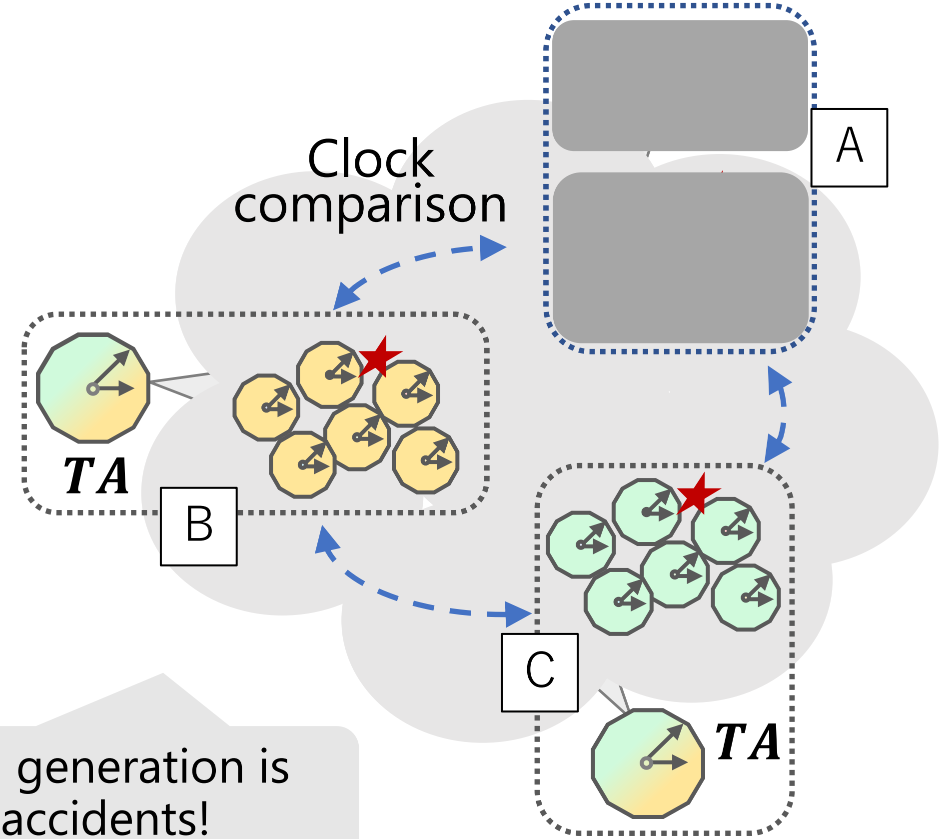


3.2 Application ~ Multiplexing of *TA* operation

■ Merit-1 : Redundancy for *TA* generation

※ Let *station-A* be the usual master station.

- Even if the clock data at *station-A* cannot be obtained, ***TA* at *station-A* can still be calculated using the clock data at *station-B* and C**.
- If *station-A* stops working and cannot calculate *TA*, ***station-B* or C can assume the role of the master station.**
- Even in such an emergency, **seamless switching of the master station** is possible if all the necessary data has been shared in advance.



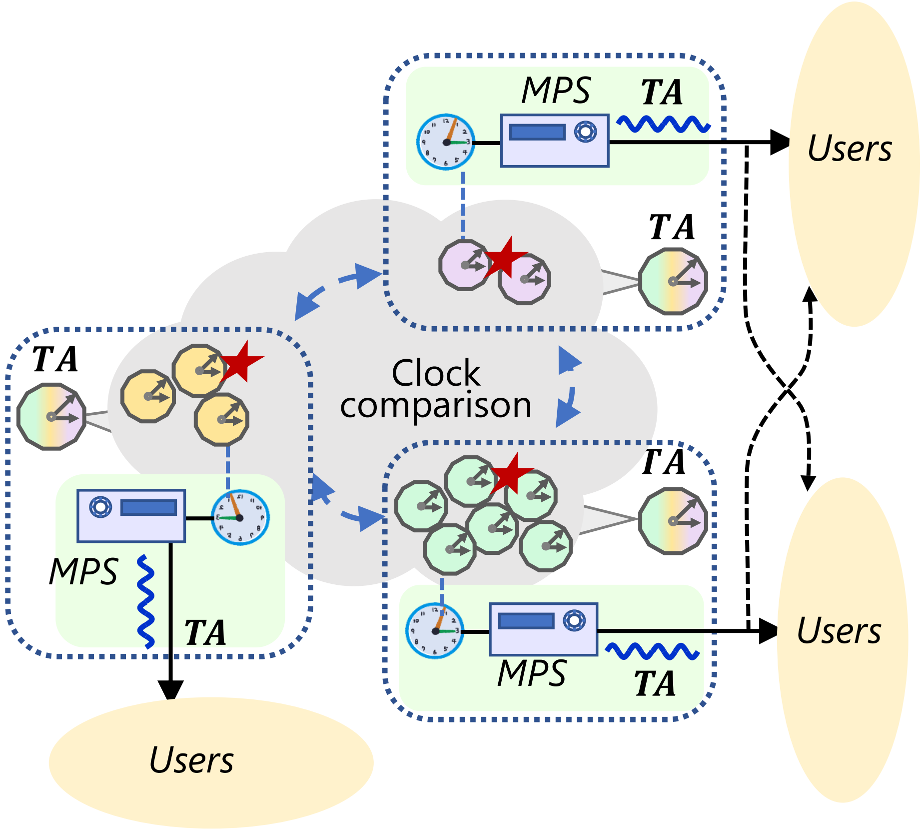
Multiplexing of *TA* generation is robust to physical accidents!

3.2 Application ~ Multiplexing of *TA* operation

■ Merit-2 : Enhancement of accessibility

- If a *TA* realization system is also installed at this remote station, it will have the ability to provide *TA* on its own.
- This indicates that even **a station with few clocks can provide *TA*** if its realization system is connected to a clock included in *TA* calculation.
- Expanding the *TA* providing station in this way enhances operational robustness and user accessibility.

※ Getting time from a distant station may be affected by delays in the transmission line. Such delays must be calibrated and compensated for, if necessary.



Contents

Bureau International des Poids et Mesures

Summary of Section 1

- **TA is a time scale of a virtual clock by averaging many atomic clocks.**
  - TA gives a solution to instrumental weakness and finite life of physical clock.
- **TA aims frequency stability and continuity.**
  - The accuracy is not the primary concern.
- **Take the consideration points into account for proper and effective use.**
  - E.g. Calibration for accuracy if needed, anomaly detection, proper algorithm.
  - Especially, clock anomaly detection in advance is so important to obtain the original quality of TA.

Summary of Section 2 (1)

- **The basic principle of *TA* is that “Averaging the fluctuating time deviation of each clock makes a smooth and stable time scale.”**

⬇

- **First, the most basic form *“TA by basic averaging (TAb)”* was considered.**
  - While ***TAb*** is useful for understanding the concept of ***TA***, it is not enough for actual use.

⬇

- **For a more advanced and practical approach, *“TA with prediction (TAp)”* was introduced.**
  - ***TAp*** provides a robust, continuous, and highly stable atomic time scale.

Summary of Section 2 (2)

- *"TA with prediction (TAp)"* is defined as follows:

$$\boldsymbol{TAp(t_1) \equiv \sum_i w_i(t_1) \cdot \{h_i(t_1) - \hat{x}_i(t_1)\}}, \quad \boxed{\text{Eq. (3)}}$$

It means that "$\boldsymbol{TAp}$ is the accumulation of prediction-errors of clocks".

- As Eq.3 is not calculable due to the conceptual parameters, $\boldsymbol{TAp}$ is obtained via the parameter "$\boldsymbol{x_i}$".

$$\boldsymbol{x_i(t) \equiv h_i(t) - TAp(t)} \quad \boxed{\text{Eq. (5)}}$$

  - $\boldsymbol{x_i}$ is the time difference between each clock and $\boldsymbol{TAp}$.
  - $\boldsymbol{TAp}$ can be defined as "the time shifted by $\boldsymbol{x_i}$ from the time of *Clock_**i***".
  - The practical $\boldsymbol{TAp}$ algorithm is the procedure for computing the time series of $\boldsymbol{x_i}$.

Summary of Section 2 (3)

- **The calculation of time series of x_i is iterative.**

$$\begin{cases} \hat{x}_i(t_k) = x_i(t_{k-1}) + \hat{y}_i(t_{k-1}) \cdot (t_k - t_{k-1}) & \boxed{\text{Eq. (10)}} \\ x_s(t_k) = \sum_i w_i(t_k)\{\hat{x}_i(t_k) - X_{is}(t_k)\} & \boxed{\text{Eq. (9)}} \\ x_i(t_k) = x_s(t_k) + X_{is}(t_k) & \boxed{\text{Eq. (11)}} \end{cases}$$

- These equations consist of measurable values and calculable values from previous calculation.
- The equations show that ***TAp*** refers to its own past values.
- At the start of the calculation only, an external reference time (for example ***UTC(k)***) is required instead of the past ***TAp***.

Summary of Section 2 (4)

- **Appropriate parameters depend on the target or situation of the calculation.**
  - Clock rate estimation and weighting should be optimized according to the condition of the clocks.
- ***TA* algorithm is not unique and includes the possibility of modification.**
  - The example used in this lecture is just a basic one. There are variations in the actually used algorithms.
- **Please understand the principle well, and find the best way of calculation.**

Summary of Section 3

Principle of ***TA*** realization:

- The numerical ***TA*** is realized as the output of a micro-phase stepper (*MPS*) driven by a frequency from *Clock-A* that is included or linked with ***TA*** calculation.
- *MPS* is continuously adjusted (="*steering*") to compensate for the time difference between *Clock-A* and ***TA***, so that the *MPS* output signal will trace ***TA***.
- This adjustment parameter includes both frequency and phase offset components.
- This adjustment cannot be made in real-time, so the process includes a prediction.

Multiplexing ***TA*** operation:

- By sharing the clock comparison data, remote stations can generate ***TA*** with almost the same quality in parallel.
- This system improves both the robustness of ***TA*** generation and the accessibility for users.

Related Python tools

■ Let's try a simulation using the tutorial Python tools.

- "*TAgen_basic.py*" : for ***TAb* by basic averaging**
  - This is good for the first item to learn the principle of how to calculate ***TA*** and the role of parameters.
- "***TAgen_pred.py***" : for ***TAp* with daily prediction & fixed weights**
  - To learn step by step the process of "***TA*** *with prediction*", this program adopts fixed weighting.
  - You can investigate how the ***TA*** changes by various combinations of clocks, parameters for the prediction, and the initial weights for each clock.
- "***TAgen_auto.py***" : for ***TAp* with daily prediction & dynamical weighting**
  - This program is almost the same as "*TAgen_pred.py*", except that it adopts a dynamical weighting $\propto 1/\sigma^2_y(\tau)$.
  - In "*TAgen_pred.py*" the weights are fixed, so that automatic optimization function, a big merit of ***TA***, is not enough. "*TAgen_auto.py*" optimizes ***TA*** by dynamical weighting according to the clock stability renewed at every calculation.
  - This ***TA*** is very practical and you can find several important attention points here.

※ Detailed explanations are denoted in user manuals. [Ref.4-5]

References

[1] Audoin C and Guinot B 2001 *The Measurement of Time* (Cambridge: Cambridge University Press)

[2] Thomas C Wolf P and Tavella P 1994 Time Scales *BIPM Monographie 94/1*

[3] P. Tavella, C. Thomas, 1991 *Metrologia* 28 57 DOI 10.1088/0026-1394/28/2/001

[4] User’s manual of Python program : “User’s Manual of “TAgen_basic.py”

[5] User’s manual of Python program : “User’s Manual of “TAgen_pred.py”

[6] User’s manual of Python program : “User’s Manual of “TAgen_auto.py”

Thank you very much for your kind attention!

Appendix-1

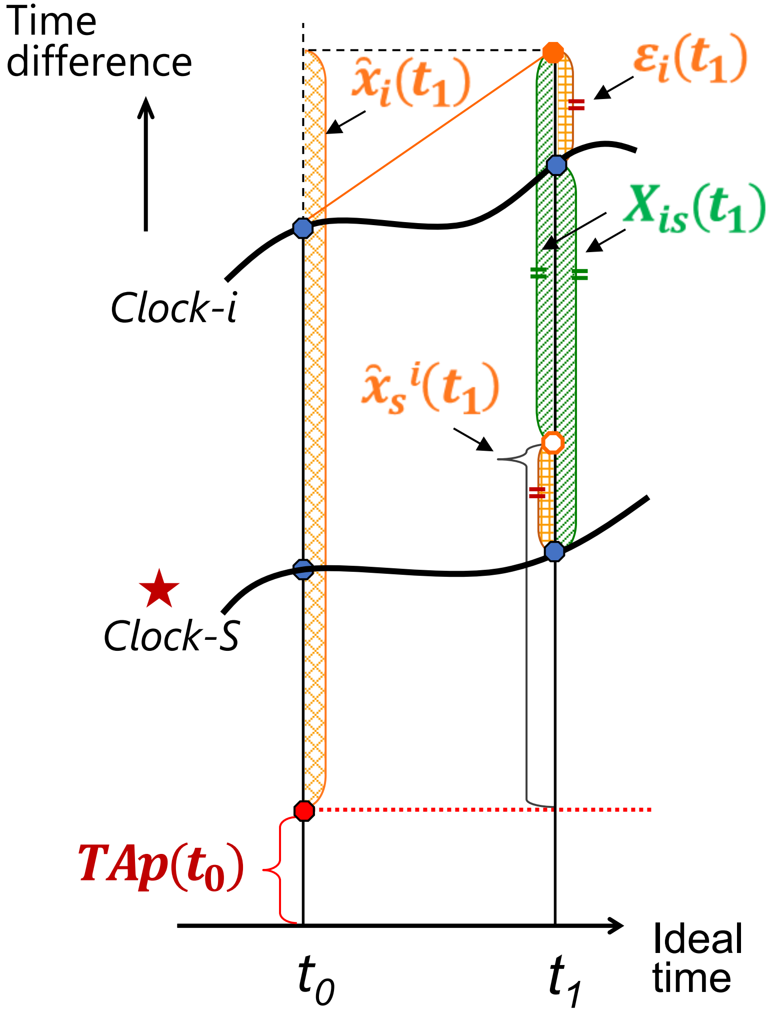
Interpretation of Eq. (9)

App-1 Interpretation of Eq. (9)

Meaning of Eq. (9)

$$x_s(t) = \sum_i w_i(t) \cdot \{\hat{x}_i(t) - X_{is}(t)\} \qquad \text{Eq. (9)}$$

- $\hat{x}_s{}^i(t) \equiv \hat{x}_i(t) - X_{is}(t)$ means the estimated time of *Clock-S* according to *Clock-i*.

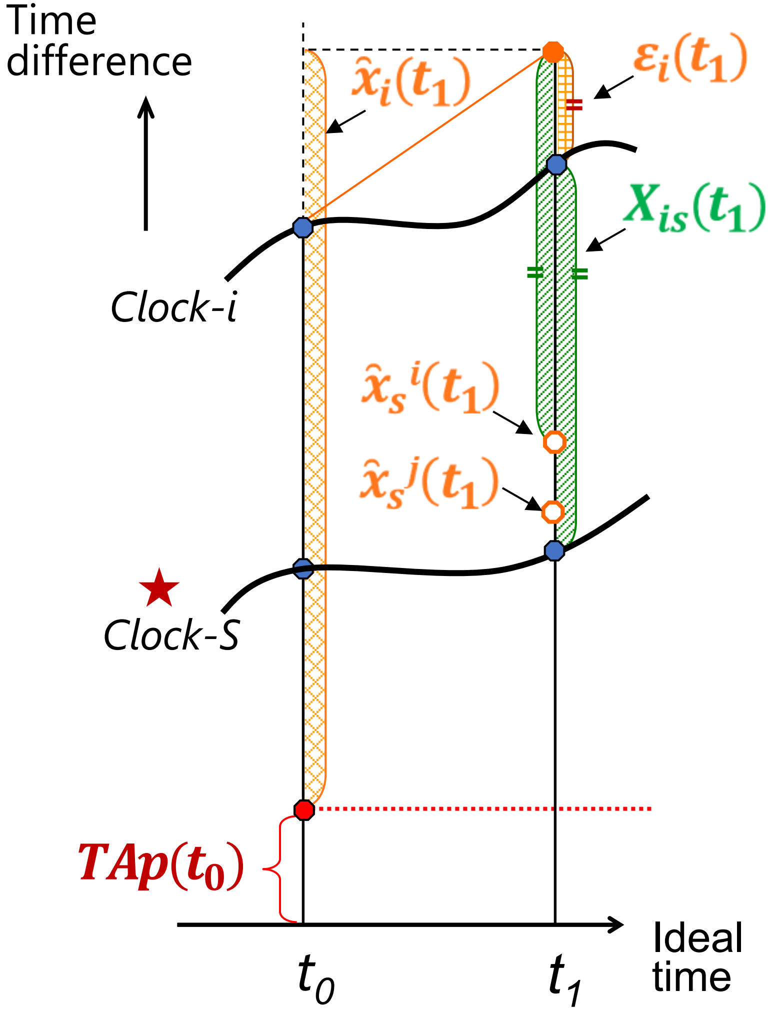


App-1 Interpretation of Eq. (9)

Meaning of Eq. (9)

$$\boldsymbol{x_s(t) = \sum_i w_i(t) \cdot \{\hat{x}_i(t) - X_{is}(t)\}}$$ Eq. (9)

- $\hat{x}_s{}^i(t) \equiv \hat{x}_i(t) - X_{is}(t)$ means the estimated time of *Clock-S* according to *Clock-i*.
- $\hat{x}_s{}^j(t)$, might be different from $\hat{x}_s{}^i(t)$, because the prediction errors ε_i and ε_j are not the same.
- However, we cannot judge which $\hat{x}_s(t)$ is truly correct.

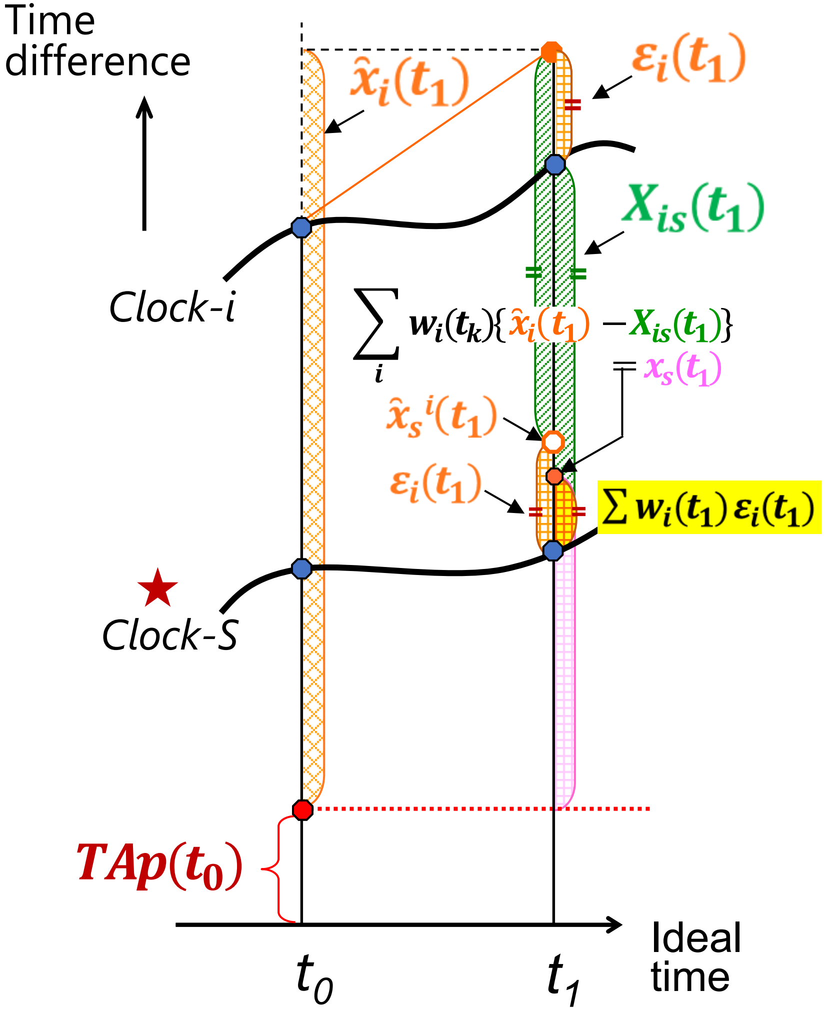


App-1 Interpretation of Eq. (9)

■ Meaning of Eq. (9)

$$x_s(t) = \sum_i w_i(t) \cdot \{\hat{x}_i(t) - X_{is}(t)\}$$ Eq. (9)

- $\hat{x}_s{}^i(t) \equiv \hat{x}_i(t) - X_{is}(t)$ means the estimated time of *Clock-S* according to *Clock-i*.
- $\hat{x}_s{}^j(t)$, might be different from $\hat{x}_s{}^i(t)$, because the prediction errors ε_i and ε_j are not the same.
- However, we cannot judge which $\hat{x}_s(t)$ is truly correct.
- Therefore, the weighted average of estimated values according to all clocks is adopted as the estimated time of *Clock-S*. → Eq.(9).



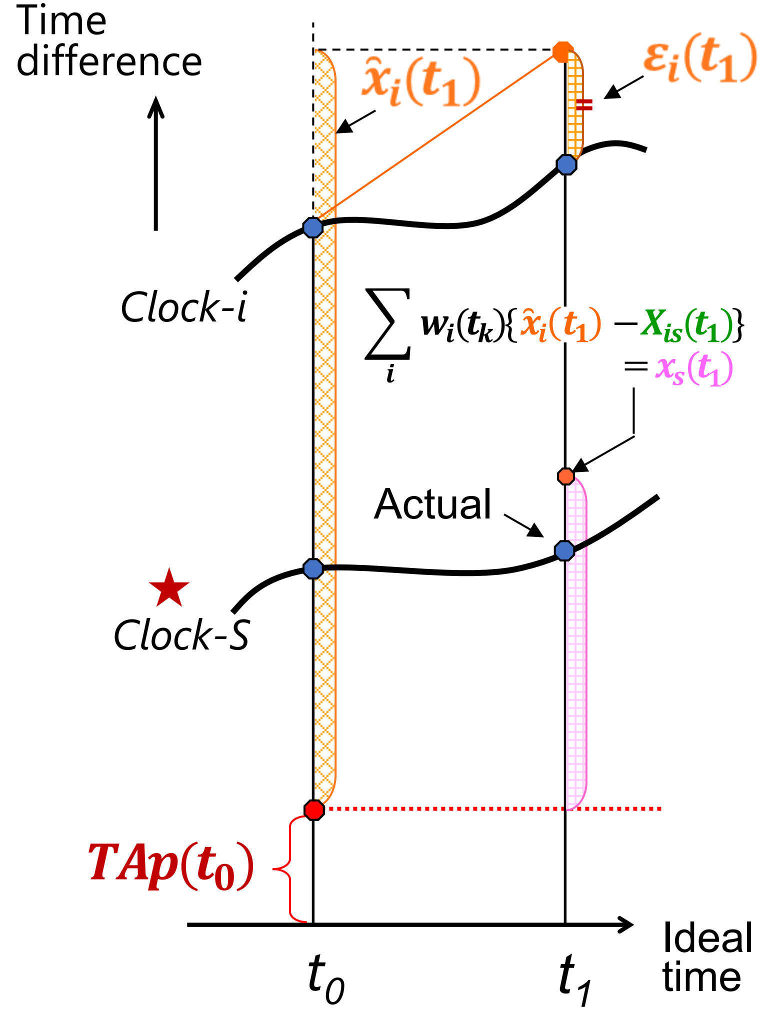
App-1 Interpretation of Eq. (9)

Meaning of Eq. (9)

$$\boldsymbol{x_s(t) = \sum_i w_i(t) \cdot \{\hat{x}_i(t) - X_{is}(t)\}} \quad \boxed{\text{Eq. (9)}}$$

- This estimated time of *Clock-S* usually differs from the actual time of *Clock-S*.

※ If all prediction are perfect with no errors, both points will be the same.
However, this is not a realistic scenario.



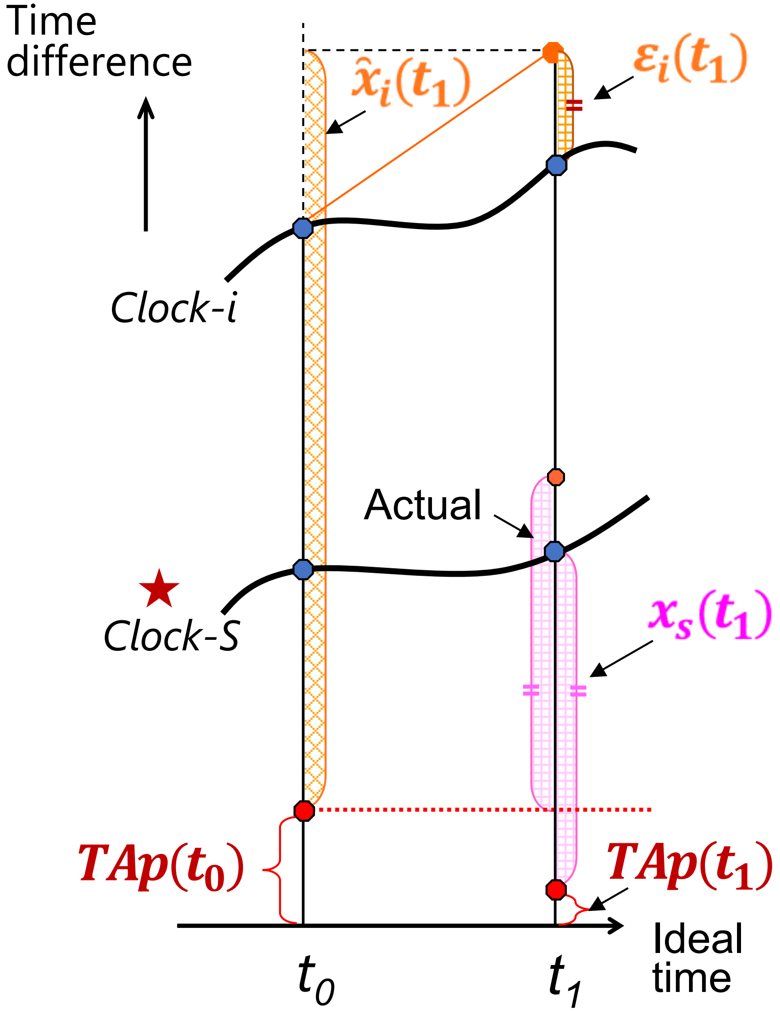
App-1 Interpretation of Eq. (9)

Meaning of Eq. (9)

$$x_s(t) = \sum_i w_i(t) \cdot \{\hat{x}_i(t) - X_{is}(t)\}$$ Eq. (9)

- This estimated time of *Clock-S* usually differs from the actual time of *Clock-S*.
- To solve this inconsistency, $TAp(t_1)$ will shift from the extended value of $TAp(t_0)$.

※There is no constraint to keep TA as the same value with the previous value.



App-1 Interpretation of Eq. (9)

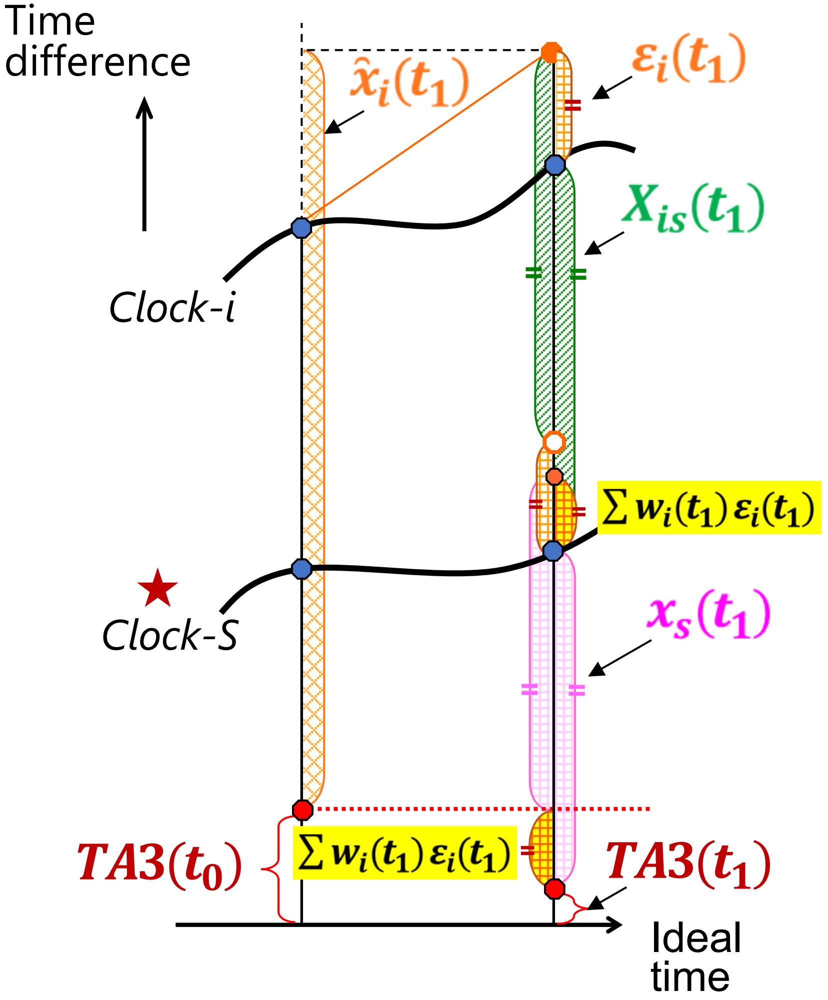
■ Meaning of Eq. (9)

$$x_s(t) = \sum_i w_i(t) \cdot \{\hat{x}_i(t) - X_{is}(t)\} \quad \boxed{\text{Eq. (9)}}$$

- This estimated time of *Clock-S* usually differs from the actual time of *Clock-S*.
- To solve this inconsistency, $TAp(t_1)$ will shift from the extended value of $TAp(t_0)$.

  ※There is no constraint to keep TA as the same value with the previous value.
- From a closer look at the right figure, this shift corresponds to the error accumulation.

  ➔ This is consistent with Eq.(7).



Appendix-2

Continuous steering of *MPS*

App-2 Continuous steering of *MPS*

■ Continuous steering of *MPS*

- After the first adjustment, $\boldsymbol{TA}$ is realized as the output of *MPS*.
- The next step is to make $\boldsymbol{x_{MPS}}$, the time difference between the *MPS* and $\boldsymbol{TA}$, equal to zero.

⬇

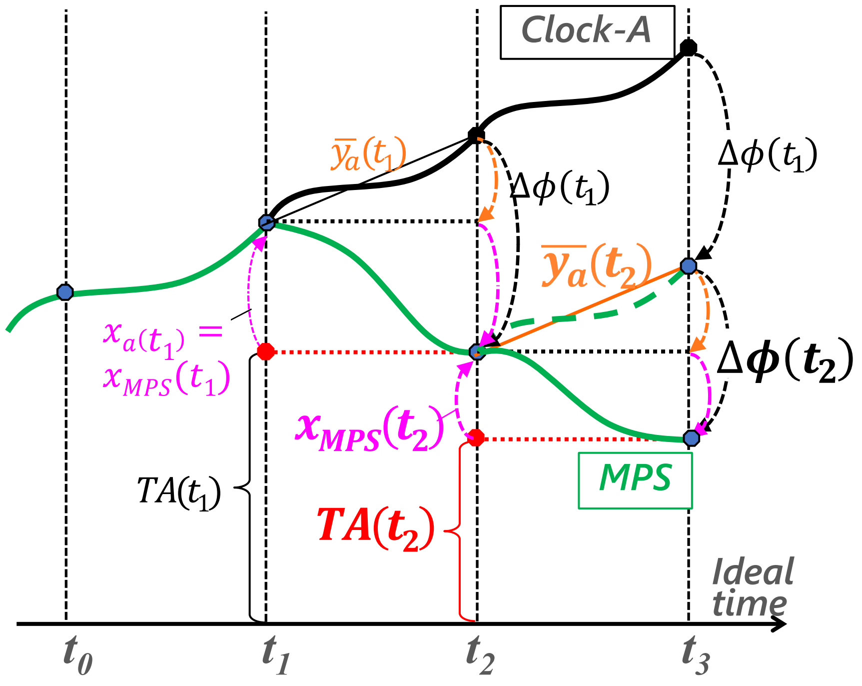
- **The adjustment frequency to *MPS* :**

$$\Delta \boldsymbol{y}(\boldsymbol{t_k}) = -\Delta \boldsymbol{\phi}(\boldsymbol{t_k}) / (\boldsymbol{t_{k+1}} - \boldsymbol{t_k}) \quad \text{Eq. (14a)}$$

$$\Delta \boldsymbol{\phi}(\boldsymbol{t_k}) = \boxed{\boldsymbol{x_{MPS}(t_k)}} + \bar{\boldsymbol{y}}_{\boldsymbol{a}}(\boldsymbol{t_k}) \cdot (\boldsymbol{t_{k+1}} - \boldsymbol{t_k}) \quad \text{(14b)}$$

※ How to get it? ➔ Next page!

※ $\bar{\boldsymbol{y}}_{\boldsymbol{a}}(\boldsymbol{t_k}) = \{\boldsymbol{x_a}(\boldsymbol{t_k}) - \boldsymbol{x_a}(\boldsymbol{t_k} - \boldsymbol{T})\} / \boldsymbol{T}$



※ **Confirm the adjustment method of the *MPS*!**

- If adjustment frequency is added to the external reference frequency of *MPS*, $\Delta \boldsymbol{y}(\boldsymbol{t_k})$ should be given.
- If adjustment frequency is added to the current frequency of *MPS*, $\Delta \boldsymbol{y}(\boldsymbol{t_k})$- $\Delta \boldsymbol{y}(\boldsymbol{t_{k-1}})$ should be given.

App-2 Continuous steering of *MPS*

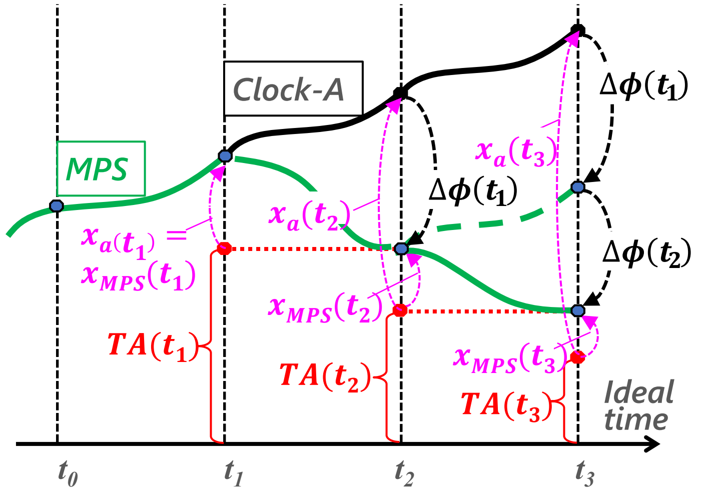
■ Continuous steering of *MPS*

- How to get $x_{MPS}(t_k)$?

  - Using the accumulation of past adjustments:

$$x_{MPS}(t_1) = x_a(t_1) \qquad \boxed{\text{Eq. (15a)}}$$

$$x_{MPS}(t_k) = x_a(t_k) - \sum_{n=1}^{k-1} \Delta\phi(t_n) \quad (k > 1) \qquad \boxed{\text{(15b)}}$$

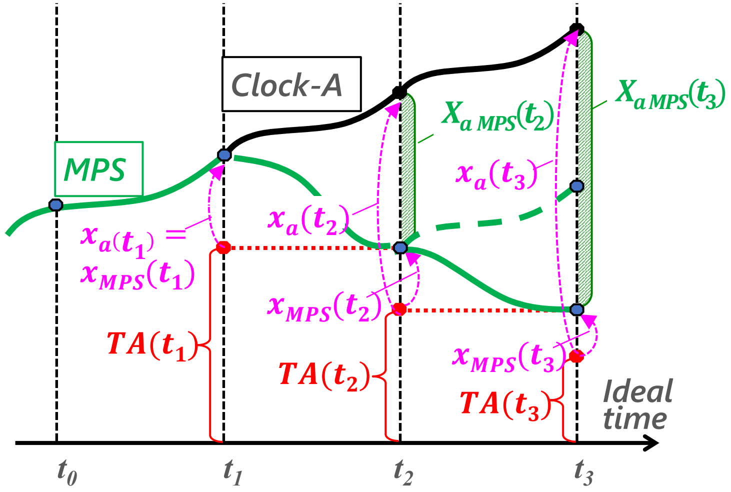


  - Using the measured time differences:

$$x_{MPS}(t_k) = x_a(t_k) - X_{a\,MPS}(t_k) \qquad \boxed{\text{Eq. (16)}}$$

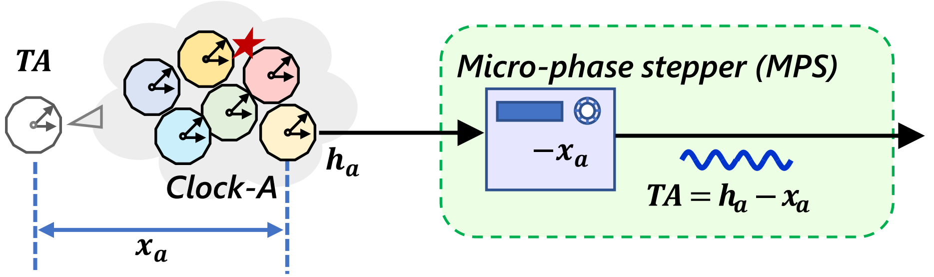
$$x_{MPS}(t_1) = x_a(t_1) \qquad (\because X_{a\,MPS}(t_1) = 0)$$

recommended.



App-2 Continuous steering of *MPS*

Summary of steering:



- **Output signal of *MPS* becomes the "realized TA".**
  - TA is realized by compensating for x_a of *Clock-A*.
  - *MPS* driven by *Clock-A* is the tool for this compensation.
- ***MPS* is continuously adjusted to trace TA.**
  - At the first epoch t_1, the *MPS* is synchronized with *Clock-A*, so $x_{MPS}(t_1)=x_a(t_1)$ will be compensated at t_2.
  - After t_2, to make x_{MPS} zero, the adjustment frequency to the *MPS* should be as follows:

$$\Delta y(t_k) = -\frac{\Delta\phi(t_k)}{t_{k+1}-t_k} = -\left\{\frac{x_{MPS}(t_k)}{t_{k+1}-t_k} + \bar{y}_a(t_k)\right\} = -\left\{\frac{x_{MPS}(t_k)}{t_{k+1}-t_k} + \frac{\{x_a(t_k) - x_a(t_k - T)\}}{T}\right\}$$

Here, $x_{MPS}(t_k) = x_a(t_k) - X_{a\,MPS}(t_k)$

or, $x_{MPS}(t_k) = x_a(t_k) - \sum_{n=1}^{k-1} \Delta\phi(t_n) \quad (k > 1)$