# Pulse – Coupled Clocks

## VC Sensor Networks

**Alexandra Wissiak**

**Nikita Soldatov**

**Bernhard Rinner**

**June 9, 2022**

## I. Introduction

*Clock* is a time measurement device consisting of an oscillator and an accumulator. Each node in a sensor system has a local clock, which is used for various purposes such as registering events or interacting with other nodes. Thus, it is crucial that time correlation between the events in real – time coincide with the one registered by the sensors. However, due to various factors (e.g. environment temperature) clocks' measurements tend deviate from their initial state. Therefore, *synchronization* refers to the process of achieving and maintaining coordination among independent local clocks via the exchange of local time information. There are three types of synchronization: phase, frequency and full synchronization. They refer to processes where clocks' offset, rate or both are being adjusted respectively.

There are also two types of clocks:

- Analog clock is characterized by an oscillator: $s(t) = cos\varphi(t)$
- Discrete clock can be seen as a sequence of significant time instants of an analog clock (e.g., upward zero crossing points): $t(n) = n * 2\pi$

In our research, we are going to focus on the latter one.

## II. Synchronization Process

*a) Packet Coupling*

Traditional methods based on packet coupling prescribe the periodic exchange of packets carrying time stamps that contain the local time $t_j(n)$ at the sender, through either point-to-point or broadcast connections. The main sources of errors for packet-based techniques are the random delays associated with the construction of a packet, queuing at the MAC layer, propagation, and processing of the packet at the receiver side. In fact, these delays imply that node i actually receives the timing packet from a node j at time $t_j(n) + q_{ij}$, where $q_{ij}$ is the random delay between the two nodes, thus making the time information on $t_j(n)$ contained in the packet outdated. Different techniques have been designed to mitigate the effects of these random factors according to diverse principles, such as synchronization between receivers of the same packet rather than between transmitter and receiver. The state of the art in packet-based techniques reports synchronization accuracies of the order of milliseconds to microseconds. Moreover, the need for exchanging of a large number of packets is common to all packet-based methods. This in turn entails large computational complexity, energy expenditure, and poor scalability.

*b) Pulse Coupling*

Physical layer-based schemes, where the local timing information in encoded directly in the trans mission times of given waveforms g(t). In particular, each node radiates a periodic train of waveforms according to its local clock, on either a dedicated bandwidth or on an overlay system such as ultra-wideband (UWB). The update of each local clock is then carried out by processing the received signal, which is a combination of waveforms transmitted by neighboring nodes. Possible processing techniques include time-of arrival estimators but efficient synchronization techniques can be devised that do not need to explicitly perform such operation. Pulse-coupled synchronization is naturally scalable, since the operations performed at each node are independent of the number of nodes available in the network and has limited complexity, requiring only simple processing at the baseband level.

## III. Pulse Coupling Approaches

*a) Integrate-and-fire oscillators*

According to the model, each node is equipped with an integrate- and-fire oscillator, as sketched in Figure 1(a). This oscillator is described, when isolated, by a state variable $\boldsymbol{x_i(t)} = g(\varphi_i(t))$, where g(·) is a periodic function (with period 2π) such that in each period it is smooth, monotonically increasing from zero to one, and concave. As before, the ticks $t_i$ (n) of the clock correspond to the time instants when the phase returns, after one period, to $2\pi$, or equivalently when the state variables charges up to its maximum value $x_i\ (t_i\ (\text{n}))\ =\ 1$ and then returns to zero. The model of integrate-and-fire oscillators prescribes the following coupling mechanism among clocks, illustrated in Figure 1(b). Upon detection of the pulse sent by any node j a time $t_j$ (n) (propagation delays are neglected in this model), the i th clock modifies the state function by adding a value ε towards the goal of selecting a firing instant that is closer to that of clock and adjusts the phase $\varphi_i$ (t ) accordingly. The main drawbacks of the model of integrate-and-fire oscillators when applied to wireless networks are: i) it is hard to extend the analysis to realistic and complex scenarios with inaccurate clocks, propagation delays, or time-varying channels; ii) the system design is not flexible enough to grant degrees of freedom for the achievement of additional relevant goals, such as trading complexity for accuracy, security, etc.

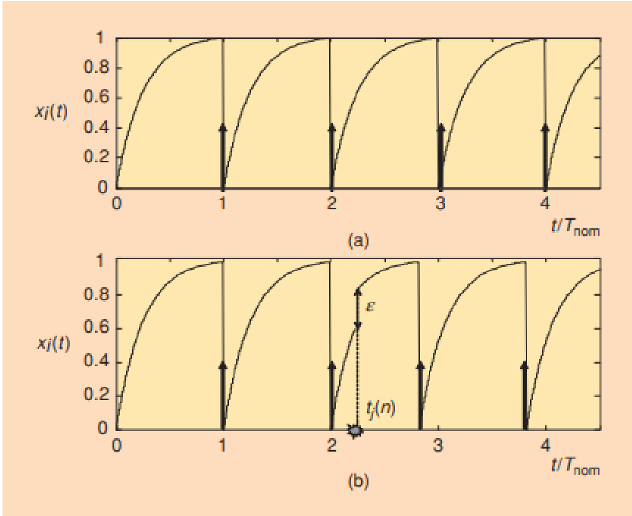

*Figure 1*

*b) Distributed discrete time Phase Locked Loops (PLLs)*

It is a feedback control system, which is depicted on Figure 2. Each node calculates a convex combination of the time differences using a time difference detector (TD). The time is exchanged between neighbouring nodes. The loop filter brings significant benefits regarding the stability of the system. Adding a pole in the LF will reduce the mismatch phases.

Pro: This method is much more flexible and can be easily implemented using standard tools.

Contra: LF increases the computational complexity of the system.

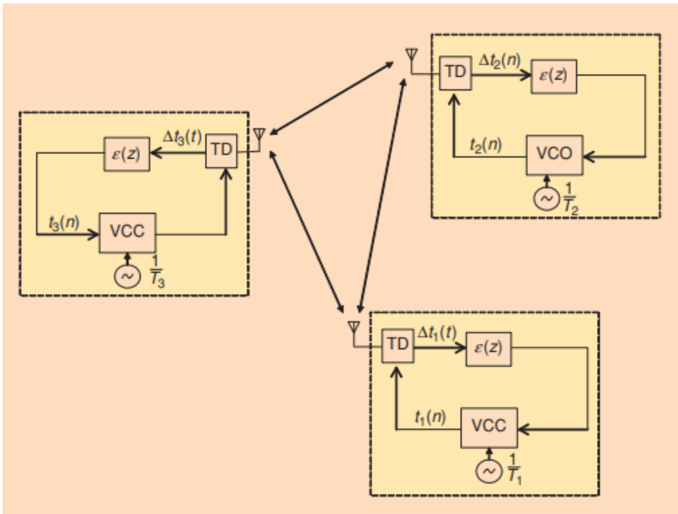

*Figure 2*

## 4. Further Analysis:

*a) Impact of topology and small – world effects of shadowing*

Increasing the amount of shadowing (i.e. standard deviation σ) improves the convergence speed and the asymptotic phase error of the system of the distributed PLLs (Figure 3).

A small-world network is characterized by the existence of paths made of a small number of edges between any two nodes. Shadowing breaks a few close connections and creates a few long links.

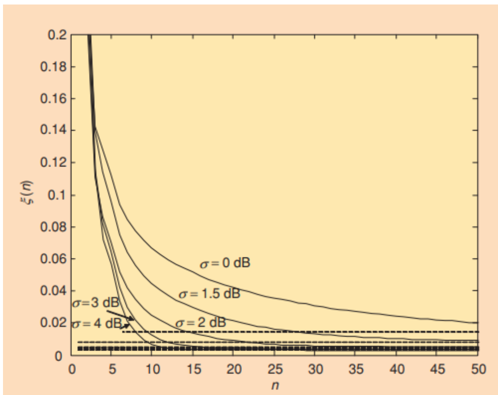

*Figure 3*

*b) Fault - Tolerance and Security*

Signal processing techniques can enhance resilience of distributed synchronization.

In a basic discrete-time PLL the time difference detector evaluates the convex combination $\Delta t_i(n)$ of the clock errors. Using only this measure, it is not possible for nodes to recognize outliers that may disrupt the synchronization process.

A threshold $\beta$ which is small enough manages to maintain a constant error $\xi(n)$ over n, thus shadowing it is able to approximately achieve full synchronization within a limited timing error (Figure 4).

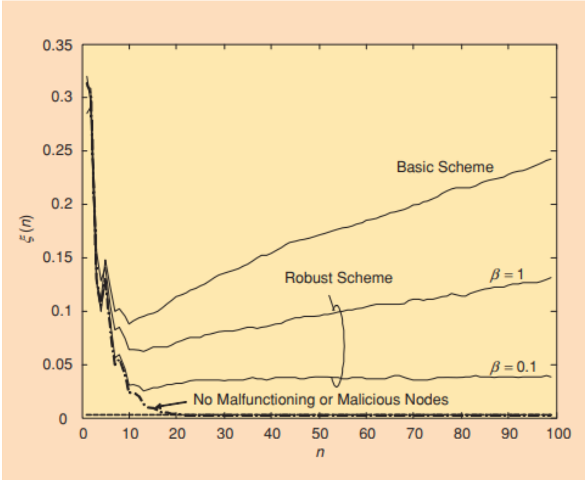


*Figure 4*

*c) Applications: Distributed consensus for multiagent coordination & Distributed Estimation/Detection in Wireless Sensor Networks*

One basic problem in multiagent coordination is achieving consensus (or agreement) on a given quantity as to yield a common decision. The basic signal and the steady-state solution depend on the initial conditions. The common value on which the agents achieve consensus is then a convex combination of the initial values t(0) with given weights $\alpha_{ij}$. The control of these weights can lead to an average or maximum likelihood estimate of a given parameter.

Goal of a WSN without fusion center is to achieve a global estimation/detection task through localized processing so that the final estimation/decision value is available for each node. Applications are monitoring, tracking and localization. Therefor a basic operation performed by the network is the calculation of the global weighted average of some local measurements. Two distinct approaches obtain global average of local measures:

Similarly to the consensus problem, local measurements are mapped into the initial values, respectively with zero frequencies. The final outcome is given at each node by the steady-state value of the phase of the local oscillator.

One could map the local measurement into free-running frequencies of the local oscillators. The frequencies eventually synchronize to the common value.

**5. Conclusion:**

To sum up with, we once again observe two aforementioned approaches.

Packet-coupling: periodic exchanging of packets with time stamp that contain the local time at the sender node, it requires large computational resources.

Pulse-coupling: updates the node's clock by a periodic train of pulses corresponding to the time instants $t_i(n)$, it does not require high computational power, but only simple processing.

## References


- Osvaldo Simeone, Umberto Spagnolini, Yeheskel Bar-Ness, and Steven H. Strogatz. Distributed Synchronization in Wireless Networks, IEEE SIGNAL PROCESSING MAGAZINE, 2008