A Quasi-periodic System is one where every process $P$ is periodic with a nominal period $T^n_P$ and a jitter of $\epsilon$. The time between two ticks may thus vary between ‘small margins’ during an execution:

$$T^n_P - \epsilon \leq \kappa_i - \kappa_{i-1} \leq T^n_P + \epsilon.$$

Signal values are sent across a bus to one-place buffers at a receiver, whence they are sampled periodically.

In his ‘cooking book’ [2], Paul Caspi showed how to build abstractions for implementing discrete systems on top of this architecture. These discrete abstractions can be expressed in a synchronous language and used to simulate quasi-synchronous systems [4]. In later work, with Albert Benveniste and others [1, 3, 5], he proposed communication protocols for preserving the discrete semantics of signal flows.

We present a brief survey of this work. In particular, we explain the simple relations between the periods and jitters of real-time tasks, and overwriting and oversamplings of values between writers and readers (it’s all a matter of fence posts). We generalize (slightly) the idea of quasi-synchronous traces. We also clarify one of the communication protocols by modelling it in the hybrid synchronous language Zélus (see Figure 1 and the corresponding code below).

---

**Figure 1**: Time-Based LTSA Protocol
References


