Simulation of an OWMS PLL network for clock signal distribution using parallel computing

E. A. Maciel[1], M. S. Batista[1], E. C. Panzo[1], F. H. Dermendjian[2], C. M. Batistela[3], J. R. C. Piqueira[3], J. M. Balthazar[2] and A. M. Bueno[1]
[1]*Institute of Science and Technology, São Paulo State University - UNESP, Sorocaba - SP, Brazil*
[2]*School of Engineering, São Paulo State University - UNESP, Bauru - SP, Brazil*
[3]*Polytechnique School, São Paulo University - USP, São Paulo - SP, Brazil.*

Summary. The development of electronics in general, in particular integrated circuits, allowed accurate and stable clock signal - or phase and frequency - distribution. The PLL is the fundamental component of clock signal distribution networks, and consists of a closed-loop control system that synchronizes a local oscillator, or clock, to a reference signal. The Phase-Locked Loop (PLL) is the basic element of clock signal distribution networks that is a fundamental part of digital communications networks. In many cases One-Way Master-Slave (OWMS) chain networks is used for clock signal distribution due to its reliability and low cost. Due to the nonlinear behavior of PLLs the design of the networks is a difficult problem, therefore, numerical simulations play important role. In this paper, a parallel computing strategy is used do simulate OWMS chain networks aiming to develop a more efficient simulation strategy, and to study the nodes interaction effects on the network due to parallel computing and distribution of the clock signal.

Index Terms — Clock distribution, phase-locked loop, synchronization, OWMS chain networks and parallel computing.

Introduction

PLLs (Phase-Locked Loops) are part of numerous applications in Electrical Engineering, control systems, and especially in communication networks. A time synchronization signal distribution network is defined by the connection of a certain clock oscillators whose purpose is to organize a temporal coordination. Synchronization networks are characterized by the fact that several oscillators (clocks) operate with the same frequency and phase. The basic element in these networks are PLL circuits [1, 2]. The PLL is a control system that synchronizes a local oscillator to an input signal. If properly designed the PLL tracks the input signal and filter phase and frequency fluctuations (jitter and wander) [3, 4, 5, 6].
The PLL block diagram shown in Fig. 1 is composed of a Phase Detector (PD), of a Low-Pass Filter (LPF) and of a Voltage Controlled Oscillator (VCO). The multiplier type PD compares the input signal v_i (Eq. 1) with the output v_o (Eq. 2) of the VCO and generates the phase error signal v_d with the same sign of the phase difference. The LPF filters the phase and frequency fluctuations and also provide the control signal v_c that controls the VCO frequency and phase (Eq. 3) around the free-running frequency ω_M.

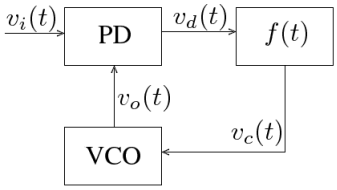


Figure 1: PLL block diagram.

$$v_i(t) = \mathrm{v}_i \sin(\omega_M t + \theta_i(t)) \tag{1}$$
$$v_o(t) = \mathrm{v}_o \cos(\omega_M t + \theta_o(t)) \tag{2}$$

$$\frac{d}{dt}\theta_o(t) = k_o v_c(t), \tag{3}$$

Considering a multiplier type PD in Fig. 1 with the input and output signals in Eqs. 1 and 2, and that the VCO control signal 3, the mathematical model of a PLL is given by

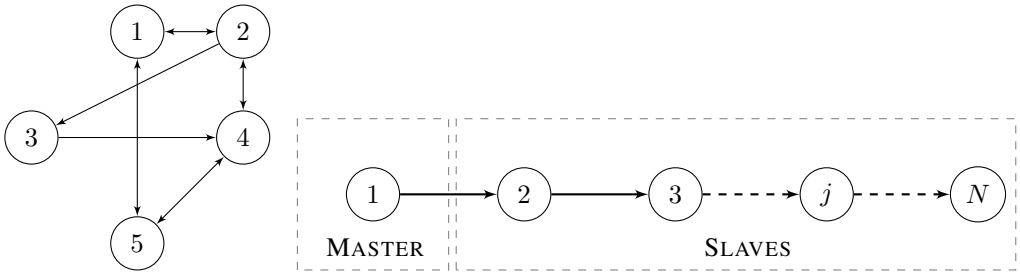
$$L\left[\theta_o(t)\right] + GQ\left[\sin\left(\theta_o(t) - \theta_i(t)\right)\right] = GQ\left[\sin\left(2\omega_M t + \theta_i(t) + \theta_o(t)\right)\right], \tag{4}$$

where $G = \frac{1}{2}k_m k_o \mathrm{v}_i \mathrm{v}_o$ is the loop gain and the operators L and Q depend on the LPF transfer function. In addition, the nonlinear differential equation in 4 is of order $p+1$ given that the LPF order is p [1, 5].

Network Synchronization

The clock signal distribution in synchronous networks requires that the frequencies and phases are common to all network elements. Many methods have been proposed for synchronization of spatially distributed clocks, such as mutually connected networks or master-slave networks. In mutually connected networks, see Fig. 2(a), all the PLLs (clocks) contribute to the phase and frequency scales. On the other hand, in Master-Slave networks, see Fig. 2(b), the master clock dictates

the phase and frequency scales. Due to simplicity and low cost, the Master-Slave strategy is frequently used for clock signal distribution. In OWMS (One-Way Master-Slave) networks, the master node has its own time base and is independent the others while slave nodes have their time base depends on a single node, may come from the master node or from another slave node. The master clock has its own and independent time basis. Slave clocks have their basis depending on a unique node, the master or another slave. Besides, these networks are classified according to the topology in chain and star.



(a) PLL network. (b) OWMS network.

Figure 2: Clock signal distribution networks.

Considering an OWMS network with N nodes, as shown in Fig. 2(b), and the PLL model of the previous section, the OWMS network model of the slave nodes is given by

$$L^{(j)}\left[\theta_o^{(j)}(t)\right] + G^{(j)}Q^{(j)}\left[\sin\left(\theta_o^{(j)}(t) - \theta_o^{(j-1)}(t)\right)\right] = G^{(j)}Q^{(j)}\left[\sin\left(2\omega_M t + \theta_o^{(j-1)}(t) + \theta_o^{(j)}(t)\right)\right] \tag{5}$$

for $j = 2, 3, \cdots, N$. As it can be noticed, the master clock's phase and frequency scales do not instantaneously spread throughout the network given that it depends on the dynamics of each slave node. In addition, noise, jitter and wander impair the quality of the clock signal distribution. The nonlinear and interaction of the slave nodes generate complex dynamics, and in this case numerical simulation is an important tool to study the qualitative behavior of the synchronization.

Parallel Programming

With the development of computational resources and multicore processors, it is possible to buid faster computational systems with higher precision and stability using parallel programming techniques [7]. A program is considered sequential programming when it is viewed as a series of sequential instructions that must be executed on a single processor. A program is considered parallel programming when it is seen as a set of parts that can be solved concurrently. Each part also consists of a series of sequential instructions, which together can be executed simultaneously on several cores of the multicore processor, as depicted in Fig 3. Traditionally, parallel programming was motivated by the resolution of fundamental engineering problems of great scientific and economic relevance, called Grand Challenge Problems (GCPs). Typically, GCPs simulate phenomena that cannot be measured by experimentation, such as: climatic, physical, chemical and biological phenomena and in telecommunications, mainly in the propagation of signals. Two main reasons for using parallel programming are: to reduce the time needed to solve a problem and to solve more complex and larger issues [7].

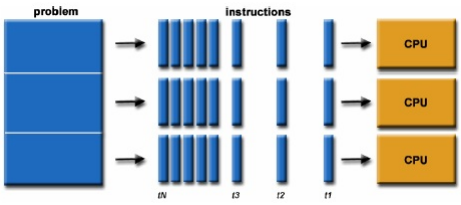


Figure 3: Instructions will be executed simultaneously on multiple processor cores [7].

Today, applications that require the development of faster and faster computers are everywhere. These applications require a lot of computing power or require the processing of large amounts of information. Same idea can be applied for the simulation of synchronization networks. A computer program is considered parallel when it organized as a set of parts that can be solved concurrently, reducing the time needed for complex computational solutions. Each part is also composed of a series of sequential instructions, but as a whole they can be executed simultaneously in several processors or processing cores, as depicted in Fig. 4.

In this paper the parallel programming is used to build and simulate the nodes of OWMS network. The target is developing an efficient simulation strategy, that allows to study the interaction of the nodes running in parallel in a distributed computation system.

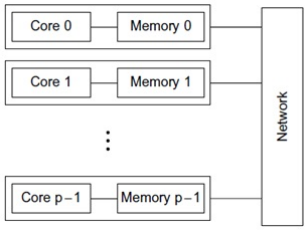


Figure 4: Distributed processing system for a network [7].

Parallel Programming in Matlab

With parallel computing, Matlab can help solve large computing problems in different ways. The software provides an interactive programming environment where it can vectorize tasks and use parallel computing support through distributed matrices, allowing you to perform multiple calculations simultaneously. Big problems can often be broken down into smaller ones, which are then solved at the same time (in sync) The main reasons to consider parallel computing in Matlab are: optimization of simulation time when distributing tasks with simultaneous execution, solving big data problems and simulating signal propagation in telecommunications and electronics [8].
Workers in Matlab are the cores of the multicore processor and are thought of as computational engines that automatically execute smaller tasks (threads) in the background when triggered by parallelism commands. Here, the PARFOR loop was used, which executes the sequence of instructions of the loop in parallel, analogously synchronous counter.

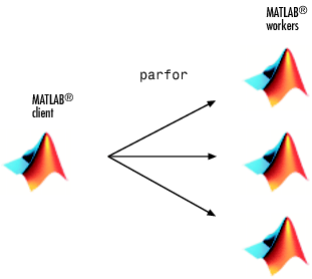


Figure 5: Interactively run a loop in parallel using PARFOR [9].

Each loop execution is an iteration where the workers perform in any order and independently of each other. If the number of workers is equal to the number of iterations of the loop, each one performs one iteration, as depicted in Fig. 5. If there are more iterations than cores, some will run more than one, receiving multiple threads at once to reduce processing time. When using parallelism commands, consideration must be given to the parallel overhead, which includes the parameterization of the processor cores, the time required for communication, coordination and data transfer (sending and receiving data) [9].

Steps to simulate a OWMS chain network

To design the mathematical simulation some steps were followed:
1 - Simulation parameters: number of PLLs, start and end time, integration math step, time vector and number of integration points;
2 - PLL parameters: input and output sinusoids amplitudes, phase detector and VCO gains, free-running angular frequency and excitation signal input;
3 - Declaration of 2 auxiliary variables to handle parallel processing. Its necessary cause the processing output of each core is given by a matrix that indexes each line corresponds to a processor core;
4 – Designing the OWMS network: making a selector to identify the condition of PLL1 (Master) and the others PLLs (Slaves), that identify the blocks that should receive feedback signal (previous PLL);
5 – Run the Network in parallel processing (PARFOR Loop);
6 - Receive the calculated values from the simulating, compile results and call the graphics function.

Simulation graphics

The simulation was performed with 4 PLLs OWMS chain network and phase angle step excitation input θ_i(t) = u(t).
Let vi(t): input signal, vo(t): output signal; vd(t): output of detector phase; vc(t): control signal; θ_i(t): reference phase and θ_o(t): ouput phase (or lagged phase).

First PLL node (Master)

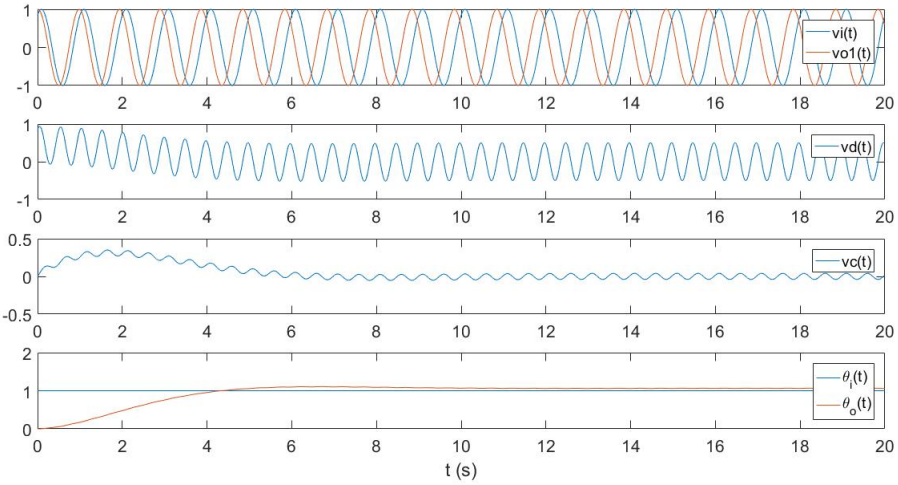


Figure 6: The output signal is lagged 90 degree ($\pi/2$ rad) of input signal, according to Eqs. 1 and 2.

Second PLL node (1º Slave)

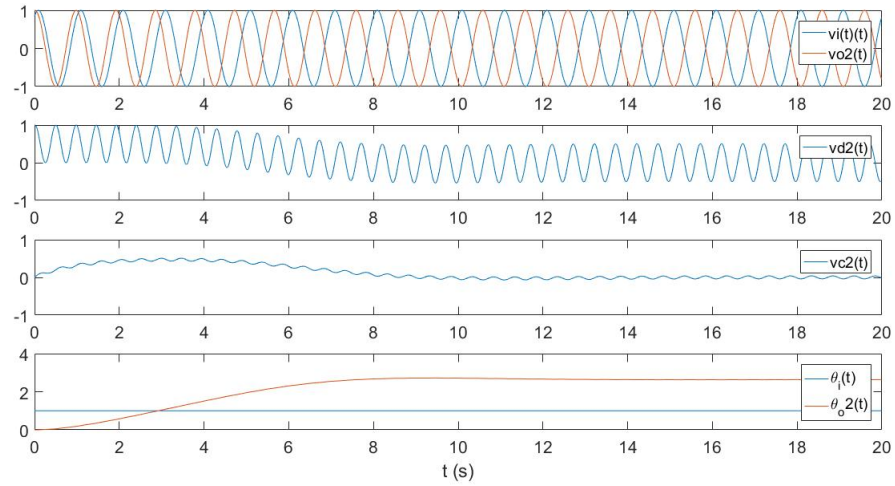


Figure 7: The output signal is lagged 180 degree (π rad) of input signal

Third PLL node (2º Slave)

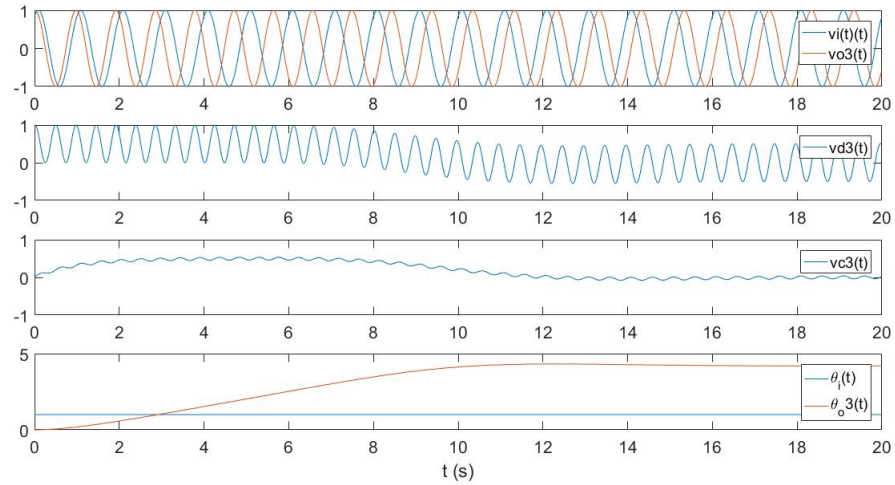


Figure 8: The output signal is lagged 270 degree ($3\pi/2$ rad) of input signal

Fourth PLL node (3º Slave)

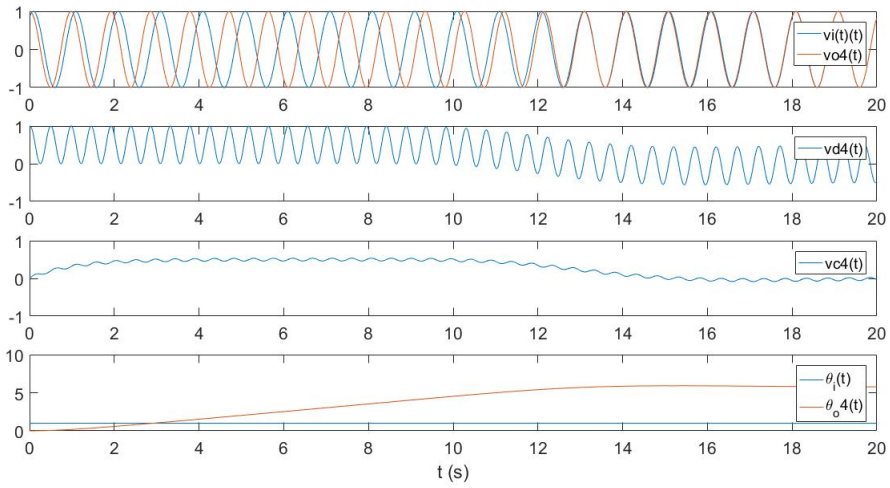


Figure 9: The output signal is lagged 360 degree (2π rad) of input signal (1 delay cicle, in phase).

Comparison between input and output phase

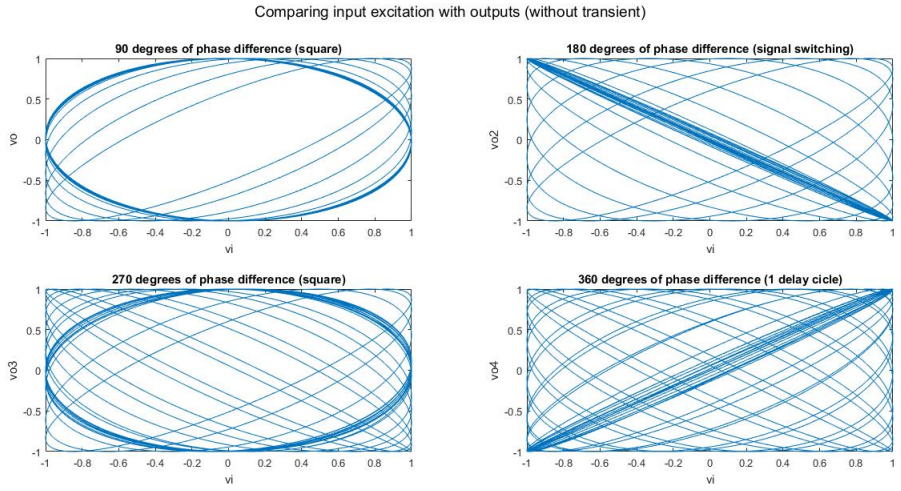


Figure 10: After the transient the trajectory of the system presents a closed orbit around of a point in lissajous.

Conclusions

Dynamic Systems Simulation is a very useful tool for this type of problem, providing conditions of existence and stability for the synchronous state of networks, relating circuit parameters, delays and deviations. An OWMS chain network was modeled with order P + 1 PLLs as slave nodes, taking into account the jitter from the phase detector. Notes that if it is not properly mitigated, network performance can be seriously degraded. It was shown that amplitude on the control signal depends on the node gain and on the filter frequency response. Using several processing cores to build a synchronization network gives flexibility to the simulation, however, parallel interactions of the nonlinear nodes, including jitter and wander dynamics, can lead to complex dynamics and, probably, to a more realistic model of synchronization networks. While parallel computing can be more complex and have a higher upfront cost, the advantage of being able to solve a problem more quickly outweighs any difficulties.

References

[1] W. C. Lindsey, F. Ghazvinian, W. C. Hagmann, and K. Dessouky. Network synchronization. *Proceedings of the IEEE*, 73(10):1445–1467, 1985.

[2] G. C. Lopes, Á. M. Bueno, and J. M. Balthazar. Elastic beam vibration control with phase-locked loop. *Volume 4B: Dynamics, Vibration, and Control*, Nov 2014.

[3] ITU-T. *Timing Characteristics of Primary Clocks - Recomendation G.811 ITU-T*, 1997.

[4] ITU-T. *Timing Requirements of Slave Clocks Suitable for Use as Node Clocks in Synchronization networks - Recomendation G.812 ITU-T*, 1997.

[5] A. M. Bueno, A. A. Ferreira, and J. R. C. Piqueira. Modeling and filtering double-frequency jitter in one-way master slave chain networks. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 57(12):3104–3111, Dec. 2010.

[6] R. O. Felix, A. M. Bueno, D. P. F. Correa, and J. M. Balthazar. Analysis of double-frequency jitter on time-delayed oscillators networks and perturbations adjustment in clock. *The European Physical Journal Special Topics*, 230(18):3603–3608, 2021.

[7] P. Pacheco. *An Introduction to Parallel Programming*. Elsevier Science, 2011.

[8] A. Zaid. *Introduction to Parallel Computing using Matlab*. Lambert Academic Publishing, 2015.

[9] MATHWORKS: Parallel Computing Fundamentals web site, https://www.mathworks.com/help/parallel-computing/parallel-computing-fundamentals.html

[10] M. Szymczyk, and P. Szymczyk. Matlab and parallel computing. *Image Processing and Communications*, 17(4), 207, 2012.

[11] P. Luszczek. Parallel programming in MATLAB. *The International Journal of High Performance Computing Applications*, 23(3), 277-283, 2009.

[12] J. R. C.Piqueira, S. A. Castillo-Vargas, L. H. A. MONTEIRO. *Two-way master-slave double-chain networks: limitations imposed by linear master drift for second order PLLs as slave nodes.* IEEE Communications Letters, v. 9, n. 9, p. 829-831, 2005.

[13] J. R. C.Piqueira. *Master-Slave Topologies with Phase-Locked Loops.* Wireless Communications and Mobile Computing, v. 2020, 2020.

[14] J. R. C.Piqueira and A. C. B. de Godoi. *Clock signal distribution with second order nodes: Design hints.* ISA transactions, v. 115, p. 124-142, 2021.