COMPUTER SIMULATION AND HARDWARE IMPLEMENTATION OF A FPLL

I. Bogdan*, H. N. Teodorescu*, F. Grigoras**

* Polytechnic Institute of Iasi, Iasi, Romania
** Institute for Information Theory, Romanian Academy, Iasi Branch, Iasi, Romania

Abstract. In previous papers, it was shown that the fuzzy-controlled PLL (FPLL) demonstrates a much better dynamical behavior and a better phase noise rejection than its crisp counterpart. The results obtained by computer simulation are verified by means of a hardware implementation. This is a brief review of the simulation and of the implementation results.

Keywords: fuzzy control, phase locked loop, computer simulation, micro processor development system

Introduction

In order to prove the benefic influence of the fuzzy control on a PLL parameters, a high performant crisp PLL was chosen as a starting point. This is the all digital PLL reported in [1] and presented in figure no. 1, that was claimed to have a small

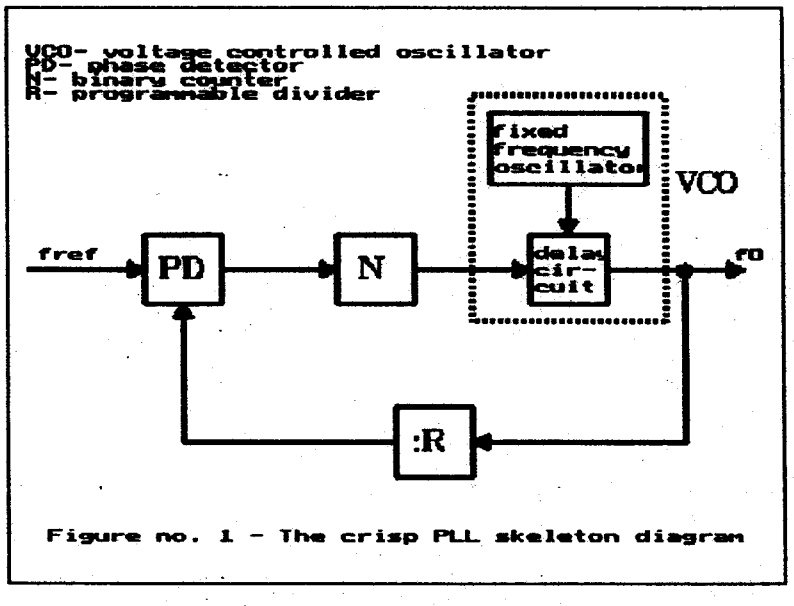

Figure no. 1 - The crisp PLL skeleton diagram

Figure 1

phase-locking time (there is not any corresponding figure or curve provided in [1]; we performed appropriater computer simulation that gave a time equal to 85 periods of the reference periods of the clock, for a 2 radians step change in input phase), to regain frequency lock in 6000 periods of the reference periods for a 100% increase of the input frequency and to reduce the input phase noise with about 15 dB.

These excellent parameters are mainly due to the non-linear PD transfer characteristic (figure no. 2) that is continuously adapted to the actual value of the PD input signals phase error by modifying the value of phr.

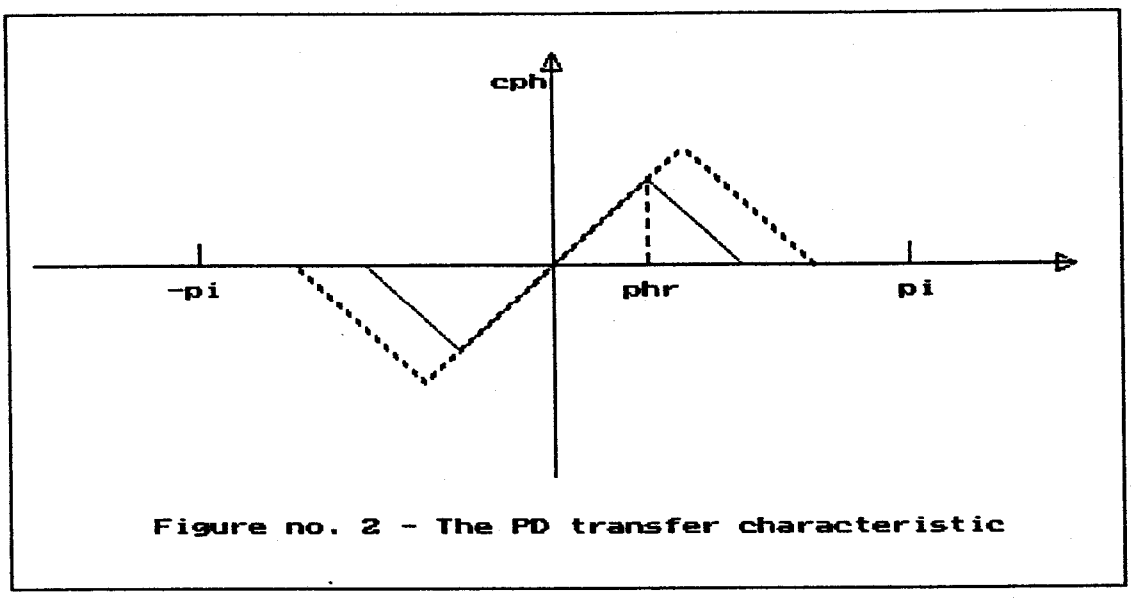

Figure 2

Fuzzy-controlled PLL

This crisp PLL operation was computer simulated and the results reported in [1] were re-obtained by running the PASCAL program several times. Then, the procedure simulating the PD operation was replaced by another one, which computes the necessary correction of the VCO frequency by using a logical fuzzy model. Thus, it were taken into consideration the values of the PD input signals phase error at the moment of the correction computing pha, and at a reference period ago phb. These crisp values were fuzzified by using a 5 degree (NB - negative big, NS - negative small, Z - zero, PS - positive small and PB - positive big) linguistic variable. The necessary fuzzy correction cph, was chosen as a 11 degree (NVB - negative very big, NB - negative big, NM - negative medium, NS - negative small, NVS - negative very small, Z - zero, PVS - positive very small, PS - positive small, PM - positive medium, PB - positive big, PVB - positive

very big) linguistic variable in order to get a fine control. All the membership functions were unequal base triangles, whose base lengths were dependent on a parameters. The value of s was chosen as a compromise between getting the smallest possible locking-time and maintaining a non-oscillating loop response [2], [3], [4].

The fuzzy inferences were done in accordance with the table presented in figure no. 3. The center of gravity method was used for defuzzification. The simulated FPLL showed a smaller phase-locking time (only 42 reference frequency periods for a 2 radians step in input phase), rather the same frequency acquisition time and a better phase noise rejection (about 20 dB) for an amplitude of the input phase noise that significantly differs from that of the input signal [2], [3], [4].

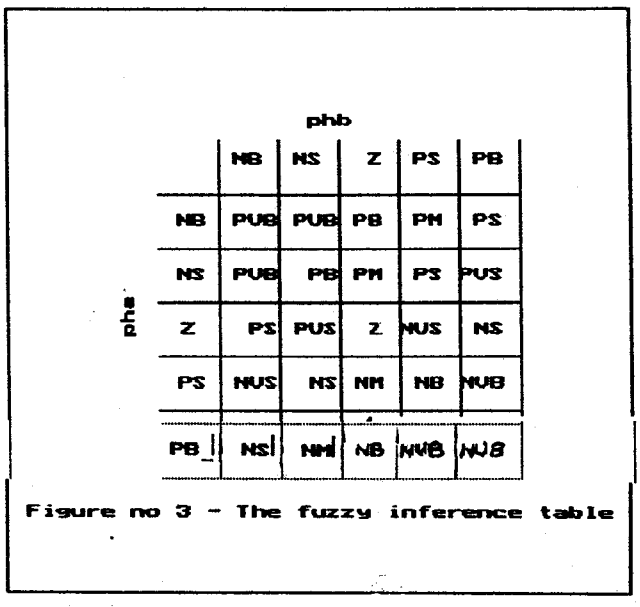

Figure 3

Hardware implementation

In order to prove the results given by the computer simulation a 8086 microprocessor based system was built and the FPLL model was transferred into its memory. For the first stage of the implementation an external reference oscillator was built and its output was used by the microsystem to get the VCO frequency correction computing time by detecting the leading edges of the reference frequency (as in the actual crisp PLL). An initial phase error is interactively introduced. A D/A interface was provided to the system. The DA converter, with a settling time of 1 micro-second, was enough fast to elliminate any time problems: the computation time of the system was much higher (about 10 ms).

The microsystem implemented model was run several times and the results obtained are in good agreement with those yielded by the PC. In the following steps of the hardware implementation the

VCO will be constructed and the real time fuzzy controlled PLL will be tested. Due to the great amount of calculus needed, the input reference frequency could not be greater than some hundred Hertz. For audio band frequencies one must choose a faster dedicated microprocessor. As for the radio frequencies band the Yamakawa's fuzzy microcontroller is practically the only choice.

Conclusions

A fuzzy controlled PLL was computer simulated, showing a better behavior than the similar crisp one. The simulation results were further checked by implementing the fuzzy control on a microprocessor based development system. The implementation proved the predicted good performances of the fuzzy control of the PLL.

References

1. O. Nakajima, H. Hikawa, S. Mori -- Performance Improvement of an All Digital PLL with Adaptive Multilevel Quantized Phase Comparator. Proceedings ISCAS 1988, IEEE, pp 603 - 606.

2. I. Bogdan, H. N. Teodorescu, D. Galea, E. Sofron -- Analysis of Fuzzy Control Techniques. AMSE Int. 90 Conference on Signals and Systems, Cetinje, Yugoslavia.

3. H. N. Teodorescu, I. Bogdan, D. Galea, -- Fuzzy PLLs, 1990 International Symposium on Fuzzy Systems and Applications, Iasi, Romania.

4. I. Bogdan, H. N. Teodorescu, D. Galea, -- Further Analysis of the FPLL Dynamical Behavior. RSFS Magazine, no. 3 - 4 / 1991, Iasi, Romania.

Nota Bene: A version of this paper is included in the volume: H.N. Teodorescu, T. Yamakawa, A. Rascanu (Editors): Fuzzy Systems and Artificial Intelligence, Iasi and Iizuka, 1992.