

# Frequency Locked Loop Based on the Time Nonrecursive Processing

Djurdje M. Perisic

Faculty of Information Technology,  
Slobomir P University (SPU),  
Slobomir, Republic of Srpska, BIH  
djurdje@beotel.rs

Vera Petrovic

School of Electrical and Computer  
Engineering of Applied Studies,  
Belgrade, Serbia  
vera.petrovic@viser.edu.rs

Branko Kovacevic

School of Electrical Engineering,  
University of Belgrade,  
Beograd, Serbia  
kovacevic\_b@etf.rs

**Abstract**—This paper describes one new approach to frequency locked loop (FLL), which is based on the time non-recursive processing of input periods. System parameters are defined by the ratio of frequencies. The conditions, under which the described system can have the properties of a FLL, are analyzed. All math analyses were made by the use of Z transform approach. It was shown that FLL is extremely fast and that it is suitable for usage in different predicting, tracking and modulation applications, for the measurements of frequency and for other applications. The FLL realization in the technique of standard integrated circuits is described. The oscilloscope picture, made on the realized 8 bit model, is presented. Analysis in frequency domain was made using matlab tools.

**Keywords**—digital circuits; FLL; PLL; pulse circuits

## I. INTRODUCTION

Time non-recursive time processing FLL (TNP FLL) described in this paper calculates and generates the output period using measurements and processing of the input periods only. The terms “non-recursive” and “recursive” are borrowed from the theory of digital filters. Actually, finite impulse response (FIR) digital filters calculate the next output using only input samples, without taking in account previous filter outputs. Such processing was called “non-recursive”. Infinite impulse response (IIR) digital filters use in calculations both input samples and previous outputs as well. This kind of processing was called “recursive”. In [1-10], to describe different applications, recursive processing was used. This means that the output period is calculated using input and output periods. Some use time differences between them, but they obviously include partly both periods in the processing. This actually means that all described algorithms in [1-10] function as feedback systems and they can possess the properties either of phase locked loop (PLL) or FLL. Unlike the described algorithms in [1-10], this approach of FLL is an open loop system. From the aspect of the algorithm used, it functions similarly to an FIR digital filter. Although digital filters process amplitude samples of the input signal, and TNP FLL uses periods of an input signal, it is very useful to understand the relationship between these two physically different systems in order to utilize, as much as possible, power digital filter theory and software tools in further development

and application of TNP FLL. The theory and techniques for developing TNP FLL are basically very similar to the demonstrated ones in [1-10]. The applicability of this approach is very wide. Frequency multiplier is described in [1]. Time shifters are described in [2, 3] and time/phase shifting in [4]. TRP PLL and TRP FLL for noise rejection are described in [5-7]. A wide range of tracking and prediction applications is described in [5, 6, 8]. Most of the algorithms described in [1-10] are suitable for usage in software form. Such a software predictor is described in [9]. Very complex systems, consisting of a lot of subsystems, can also function as FLLs, whose realization is based on the same technique [10]. Articles and books [11-17] are used as theoretical base, for electronics implementation and for the development necessities.

## II. MATHEMATICAL DESCRIPTION OF FLL

The general case of an input signal  $S_{in}$  and an output signal  $S_{op}$  of FLL is shown in Figure 1. The time difference  $\tau_k$  is used instead of the phase difference. The periods  $TI_k$  and  $TO_k$ , as well as the time difference  $\tau_k$ , occur at discrete times  $t_0, t_1, \dots, t_k, t_{k+1}$ , which are defined by the falling edges of the pulses of  $S_{op}$  in Figure 1. The main recursive equation describing the functioning of FLL is presented by (1), where  $a$  and  $b$  are the system parameters. The natural relations between variables, which come out from Figure 1, are given by (2) and (3):

$$TO_{k+2} = a \cdot TI_k + b \cdot TI_{k+1} \quad (1)$$

$$\tau_{k+1} = \tau_k + TO_k - TI_k \quad (2)$$

$$\tau_{k+2} = \tau_{k+1} + TO_{k+1} - TI_{k+1} \quad (3)$$

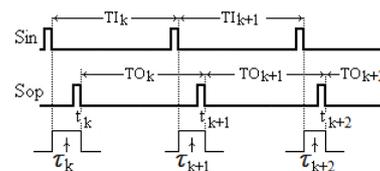


Fig. 1. Time relation between variables.

Note that (3) is just (2), shifted for one step. Equation (3)

will be taken in account for the simulation of FLL. According to (1) and (2), FLL has two output discrete variables, which describe the behaviour of FLL in terms of  $TI_k$ . The output variables are  $TO(k+1)=f[TI(k)]$  and  $\tau(k+1)=f[TI(k)]$ . To analyze the conditions under which the described system possesses the properties of FLL, the Z transforms of (1) and (2) are presented in (4) and (5) respectively, where  $TO_1$ ,  $TO_0$  and  $\tau_0$  are the initial values of  $TO_k$  and  $\tau_k$ . Note that, according to (1),  $TO_1=bTI_0$ . Substituting  $TO_1=bTI_0$  into (4),  $TO(z)$  was found and presented in (6). Substituting  $TO(z)$  from (6) into (5),  $\tau(z)$  was calculated and presented in (7). Note that, in order to shorten the mathematical procedure, (7) is reached taking in account that the relation between the system parameters must be  $a+b=1$ , shown later on, in (12). Two transfer functions, which describe FLL, can be now recognized. The first one is  $H_{TO}(z)$ , shown in (8), which describes the behaviour of the output period in terms of the input period. The second one  $H_\tau(z)$ , shown in (9), describes the behaviour of time difference in terms of the input period.

$$z^2TO(z) - zTO_1 - z^2TO_0 = aTI(z) + zbTI(z) - zbTI_0 \quad (4)$$

$$z\tau(z) - z\tau_0 = \tau(z) + TO(z) - TI(z) \quad (5)$$

$$TO(z) = TI(z) \frac{(a+zb)}{z^2} + TO_0 \quad (6)$$

$$\tau(z) = -TI(z) \frac{z-(b-1)}{z^2} + \frac{TO_0 + z\tau_0}{z-1} \quad (7)$$

$$H_{TO}(z) = \frac{TO(z)}{TI(z)} = \frac{(a+zb)}{z^2} \quad (8)$$

$$H_\tau(z) = \frac{\tau(z)}{TI(z)} = -\frac{z-(b-1)}{z^2} \quad (9)$$

### III. ANALYSIS OF FLL PROPERTIES

It is necessary now to investigate the conditions under which the described system possess the properties of FLL or PLL. Let us remember that a FLL generates the output pulse rate whose frequency tends to reach the frequency of the input pulse rate during the transient state. FLL is in the stable state when the output frequency becomes either equal or in certain pre-defined relation to the input frequency. FLL does not care about the phase difference between input and output signals. FLL regulates only the frequency of the output signal. The phase difference of FLL depends on the initial conditions and FLL parameters. Unlike FLL, PLL regulates both frequency and phase difference between input and output signals at the same time. In most applications, classical PLL tends to equalize both frequency and phase of input and output signals. However, taking into account results in [2, 3], the phase difference of time recursive PLL between input and output signals can be regulated by the system parameters. This phase difference can take any value, but it does not depend on the initial conditions of the variables. Generally, some of time recursive PLLs can control the phase difference between input and output by the system parameters. Note that thereby, the phase difference does not depend on the initial conditions.

Otherwise the system would represent FLL. The step analysis is the most suitable approach for the investigation of the properties of the system described. Let us suppose that the step input is  $TI(k)=TI=\text{constant}$ . Substituting the Z transform of  $TI(k)$  i.e.  $TI(z)=TIz/(z-1)$  into (6) and using the final value theorem, it is possible to find the final value of the output period  $TO_\infty=\lim TO(k)$  if  $k \rightarrow \infty$ , using  $TO(z)$ :

$$TO_\infty = \lim[(z-1)TO(z)]_{z \rightarrow 1} = TI(a+b) \quad (10)$$

Substituting  $TI(z)=TIz/(z-1)$  into (7) and using the final value theorem, it is possible to find the final value of the time difference  $\tau_\infty=\lim \tau(k)$  if  $k \rightarrow \infty$ , using  $\tau(z)$ :

$$\tau_\infty = \lim[(z-1)\tau(z)]_{z \rightarrow 1} = TI(b-2) + TO_0 + \tau_0 \quad (11)$$

It can be concluded, according to (10), that the described system can possess the properties of a FLL, if the system parameters satisfy (12). Note that if (12) is satisfied,  $TO_\infty=TI$ , i.e. for stable FLL, the output frequency is equal to the input frequency. Equation (11) confirms that the system possesses the properties of FLL, since  $\tau_\infty$  depends on the initial conditions. It comes out that the system does not possess the properties of a PLL.

$$a+b=1 \quad (12)$$

It is interesting to analyze, whether FLL is able to track the ramp input. To estimate this, it is necessary to determine well known velocity error  $K_V$ , providing that the input period is the ramp function  $TI(k)=TI_V(k)=ck$ , where  $c$  is a time constant. Note that  $TI(z)=TI_V(z)=Z(c \cdot k)=cz/(z-1)^2$ . Generally, velocity error  $K_V=\lim[TO_V(k)-TI_V(k)]$  for  $k \rightarrow \infty$ . One more suitable expression for velocity error is  $K_V=\lim TI_V(k)[H_{TO}(k)-1]$  for  $k \rightarrow \infty$ . Using the condition (12), the final value theorem and  $H_{TO}(z)$  given by (8),  $K_V$  is calculated and shown in (13). According to (13), FLL is able to track the velocity input with the constant error. However, if  $b=2$  ( $a=1-b=-1$ ),  $K_V=0$ , i.e. FLL tracks the velocity input without any error.

$$K_V = \lim\{(z-1)TI_V(z)[H_{TO}(z)-1]\}_{z \rightarrow 1} = c(b-2) \quad (13)$$

Let us now determine the behaviour of  $\tau_V(k)$  for the velocity input, if  $k \rightarrow \infty$ . Taking in account  $b=2$  and  $a=-1$ , and using the final value theorem,  $\tau_{V\infty}=\lim \tau_V(k)_{k \rightarrow \infty}$  is calculated using  $\tau_V(z)$  and shown in (14). The expression  $\tau_V(z)$  is found out by substitution of  $TI(z)=TI_V(z)=cz/(z-1)^2$  in (7):

$$\tau_{V\infty} = \lim[(z-1)\tau_V(z)]_{z \rightarrow 1} = -c + TO_0 + \tau_0 \quad (14)$$

According to (14),  $\tau_{V\infty}$  is a constant. Besides the initial conditions  $TO_0$  and  $\tau_0$ ,  $\tau_{V\infty}$  depend on the time constant  $c$ , which is the slope of the ramp input function. It is worth checking whether FLL is able to track the acceleration input function  $TI(k)=TI_A(k)=ck^2$ . Note that, in this case,  $TI(z)=TI_A(z)=Z(ck^2)=cz(z+1)/(z-1)^3$ . It is necessary to calculate now the acceleration error  $K_A=\lim[TO_A(k)-TI_A(k)]$ , for  $k \rightarrow \infty$ . One more suitable expression for  $K_A$  is  $K_A=\lim TI_A(k)[H_{TO}(k)-1]$  for  $k \rightarrow \infty$ . Taking in account the values  $b=2$  and  $a=-1$ , then using the final value theorem and  $H_{TO}(z)$  given by (8),  $K_A$  is calculated in (15). According to (15), FLL is able to track the acceleration input, but with the constant time error  $K_A=-2c$ :

$$K_A = \lim_{z \rightarrow 1} (z-1) T_{IA}(z) [H_{TO}(z) - 1] = -2 \cdot c \quad (15)$$

IV. REALIZATION OF FLL

According to the above analysis, FLL possesses powerful tracking performances for  $\alpha=-1$  and  $b=2$ . If we substitute these values into (1), it will be transformed into (16). If we now multiply, at the same time, all of its members by clock frequency  $f_c$ , (16) will be transformed into (17). The functional scheme of FLL, which comes out from (17), is presented in Figure 2. According to (17), the input period  $TI_k$  is measured by clock with frequency  $f_c$ , the input period  $TI_{k+1}$  is measured by clock with frequency  $2f_c$  and the output period  $TO_{k+2}$  is generated by clock with frequency  $f_c$ . FLL consists of a recursive calculation model (RCM) and a programmable period generator (PPG). RCM calculates  $N_b$  in binary form, according to the right part of (17), and PPG generates the output period in the next step. PPG is described in [2-4].

$$TO_{k+2} = -TI_k + 2TI_{k+1} \quad (16)$$

$$f_c \cdot TO_{k+2} = -f_c \cdot TI_k + 2f_c \cdot TI_{k+1} \quad (17)$$

Let us remember that PPG is based on the up-down counter and that the output period  $TO$  of Sop is  $TO=N_d \cdot t_c$ .  $N_d$  is the decimal value of binary code  $N_b$  and  $t_c$  is the clock period, i.e.  $t_c=1/f_c$ . For the realization of RCM, presented in Figure 2, the technique in [1-10] was used. However, there are always differences in realization of any two RCMs. The differences are conditioned by the dissimilarity of the algorithms used. The real time functioning of FLL is presented in Figure 3.

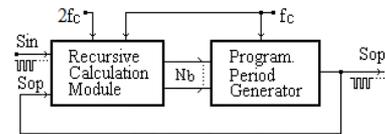


Fig. 2. Functional scheme of FLL.



Fig. 3. The loop is in the stable state and  $TO$  is equal to  $TI$ .

The oscilloscope picture is made on the realized 8-bit FLL. The voltage waveforms were taken when the FLL was in stable state. For this purpose, step input was chosen  $TI=0.1ms$  ( $f_{in}=1/TI=10kHz$ ),  $\alpha=-1$  and  $b=2$ . Clock frequency corresponding to parameter  $a$  was  $f_c=110kHz$ . Clock frequency corresponding to parameter  $b$  was  $2f_c=220kHz$ . The ratio  $TI/t_c = f_c/f_{in}=110kHz/10kHz=11$ . This ratio can be noticed in Figure 3. As it can be seen, the output period is locked to the input period. The precision of locking depends on the ratio between the input period  $TI$  and the clock period  $t_c=1/f_c$ . Greater ratio  $TI/t_c$  provides more precise locking. However, greater ratio of  $TI/t_c$  requires greater capacity of all FLL parts. Note also that, in case that a fast dynamic input function is fed into FLL, like for instance a ramp function or an acceleration function, the

capacity of FLL must be great enough to calculate the greatest output period which may appear within the transient state. Time difference  $\tau_{\infty}$ , which corresponds to stable FLL in Figure 3, is not zero. This is consistent with the properties of any FLL. Time difference  $\tau_{\infty}$  has random value, because it depends on the initial conditions, according to (11).

V. APPLICATIONS OF FLL

In this chapter some of the applications of FLL will be considered. Besides the above analysis, these considerations will be supported by the simulation of FLL operations. The simulations are to demonstrate the tracking performances of FLL, to prove the mathematical analysis, to discover additional properties and possible efficient applications of FLL, and to enable better insight into the procedure and the physical meaning of the described variables. All discrete values in simulations were merged to form continuous curves. Note that all variables in the following figures were presented in time units. The time unit can be,  $\mu sec$ ,  $msec$  or any other, but assuming the same time units for  $TI$ ,  $TO$ ,  $\tau$  and  $c$ , it was more suitable to use just "time unit" or abbreviated "t.u" in the text. It was more convenient to omit the indication t.u in diagrams. All simulations were performed using (1)-(3). The simulations of  $TO(k)$  and  $\tau(k)$  for the step input  $TI_k=10t.u$ , are shown in Figure 4(a). All values for three cases of different parameters  $\alpha$  and  $b$ , initial conditions and final values are shown in Figure 4(a). The system parameters satisfy (12) in all cases and consequently, the output periods reached input periods. According to (11), using the parameter values and the initial conditions presented in Figure 4(a),  $\tau_{1\infty}$  can be calculated as:  $\tau_{1\infty}=TI(b-2)+TO_0+\tau_0=10(0.1-2)+3+0=-16t.u$ . This result agrees with the simulated  $\tau_{1\infty}$ . In the same way, it can be calculated that  $\tau_{2\infty}=-4t.u$ , and  $\tau_{3\infty}=6t.u$ . Note that calculated values  $\tau_{2\infty}$  and  $\tau_{3\infty}$  also agree with the simulated  $\tau_{2\infty}$  and  $\tau_{3\infty}$  presented in Figure 4(a). These simulation results prove the correctness of the mathematical description and step analysis. The real time relation between  $Sin$ ,  $Sop$  and  $\tau_k$ , for simulated case Nr. 2, is shown in Figure 4(b). For the stable FLL, period  $TO_{\infty}=TI=10t.u$  and  $\tau_{\infty}=-4t.u$ . Note that FLL is very fast. It takes only two steps to reach the stable state.

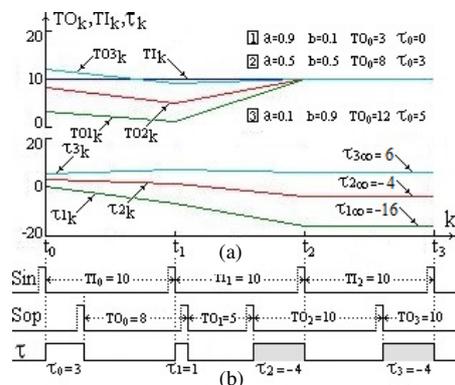


Fig. 4. (a) Transition states of FLL for the step input and different system parameters. (b) Real time presentation of  $Sin$ ,  $Sop$  and  $\tau_k$  for the simulated case Nr 2. All transient states are finished in two steps.

The simulations of  $TO(k)$ ,  $K_V$  and  $\tau(k)$  for the velocity input  $TI_k=(10+4\cdot k)t.u$ , are shown in Figure 5. The values for two cases of different parameters  $\alpha$  and  $b$ , initial conditions and final values are shown in Figure 5. The case Nr. 1 corresponds to the condition  $\alpha+b=1$ . According to (13),  $K_{V1}=c(b-2)=4(0.2-2)=-7.2t.u$ . This calculated result agrees with  $K_{V1}$ . The case Nr. 2 matches both, the previous one  $\alpha+b=1$  and the additional conditions  $b=2$  and  $\alpha=-1$ . According to (13),  $K_{V2}=c(b-2)=0.0t.u$ . This result also agrees with  $K_{V2}$  simulation result, proving the previous conclusion. So if  $b=2$  ( $\alpha=1-b=-1$ ),  $K_V=0$ , i.e. FLL tracks the velocity input without any error. Finally, according to (14),  $\tau_{2\infty}=-c+TO_0+\tau_0=-4+8+3=7t.u$ . This result agrees with the simulated  $\tau_{2\infty}$ . All simulated values  $K_{V1}$ ,  $K_{V2}$ ,  $\tau_{2\infty}$  agree with the calculated ones by (13) and (14). The identity of analytical and simulation results proves the correctness of the entire theoretical approach, as well as the validity of the obtained results. Even for the ramp input, demonstrated in Figure 5, FLL takes only two steps to reach stable state.

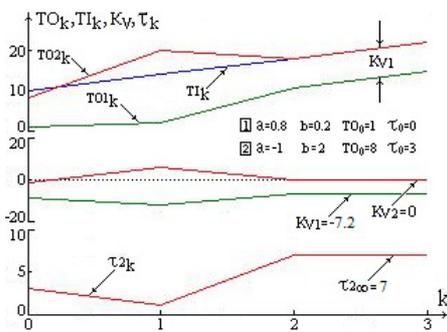


Fig. 5. Simulation of the input and output variables in the tracking of the velocity function. All transient states are finished in two steps.

The simulations of  $TO(k)$ ,  $K_A$  and  $\tau(k)$  for the acceleration input  $TI_k=(10+4\cdot k^2) t.u$ , are shown in Figure 6. The values for two cases of different parameters  $\alpha$  and  $b$ , initial conditions and the final values are shown. Note that the parameters  $\alpha$  and  $b$  and the initial conditions are the same as those in Figure 5, so comparison between them is available. The case Nr. 1 corresponds to the condition  $\alpha+b=1$  and the case Nr. 2, satisfies both conditions,  $\alpha+b=1$  and  $b=2$ ,  $\alpha=-1$ . It can be seen that, for the case Nr. 2, the output period  $TO_2$  is able to track even the acceleration function, but with constant error  $K_{A2}$ .

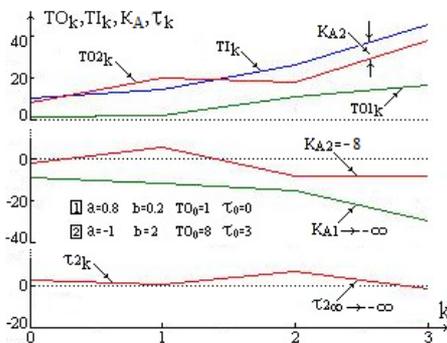


Fig. 6. Simulation of the input and output variables in the tracking of the acceleration function. All transient states are finished in two steps.

According to (15),  $K_{A2}=-2\cdot c=-2\cdot 4=-8t.u$ . This result agrees with the simulated  $K_{A2}$  in Figure 6. The acceleration error  $K_{A1}$ , for case Nr. 1, tends to minus infinity in Figure 6. That means FLL is not able to track the acceleration function if the parameters do not satisfy the conditions  $b=2$ ,  $\alpha=-1$ . Time difference  $\tau_{2\infty}$  also tends to minus infinity in Figure 7. Simulations showed that, providing that  $\alpha+b=1$ , FLL is very fast. It always reaches stable state after only two steps, no matter if the input function was a step function, a velocity function or an acceleration function. No FLL or PLL, described in [1-10], possesses this property. From a practical point of view, this means that this FLL is very suitable for tracking and predicting applications, as well as for other applications which require very short transition time.

VI. ANALYSIS OF FLL IN FREQUENCY DOMAIN

Analysis of FLL in frequency domain should expend knowledge of its properties. For this purpose, Matlab commands, devoted to digital filter design, were used. Since FLL is described by two transfer functions  $H_{TO}(z)$  and  $H_\tau(z)$ , shown in (8) and (9), using Matlab command freqz, frequency responses in the regain (0, pi) [rad], from both transfer functions, are generated for  $\alpha=-1$  and  $b=2$  and presented in Figures 7(a) and 7(b) respectively. The frequency responses consist of magnitude and phase responses. The sampling frequency  $f_s=200Hz$  corresponds to the region (0, 2pi) [rad], so that  $f_s/2=100Hz$ , covers the region (0, pi). Regarding Figure 7, time presentations of  $TI$  and  $TO$  as well as the spectrums of  $TI$ ,  $TO$  and  $\tau$ , shown in Figure 8, will be used. The input signal is the input period  $TI$ , presented in Figure 8(a), as  $TI(k+1)=10+6\cdot\sin[(2\pi/f_s)\cdot f_m\cdot k]$ . It can be considered as a constant period of 10t.u., which is modulated by samples of sinus signal, whose amplitude is 6t.u. and frequency  $f_m=10Hz$ .

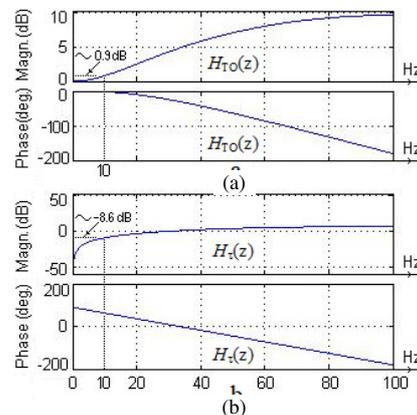


Fig. 7. (a) Frequency responses -  $H_{TO}(z)$ . (b) Frequency responses -  $H_\tau(z)$

The number of time steps is chosen to be  $k=200=f_s$ . The angular sampling step is  $w_s=2\pi/200$  [rad]. Since 200 frequency sampled steps covers region of one period, (0, 2pi), it means the sampling rate will be 20 samples/period. This provides sufficiently good resolution of  $TI$  in Figure 8(a) for this analysis. FLL generates the output  $TO$ , which is calculated according to (1), (2) and (3). Note that  $TO$  exactly tracks  $TI$  with delay of one step. Matlab commands fft and stem are used for the generation of the spectrums of  $TI$ ,  $TO$  and  $\tau$  in Figure

8(b). These spectrums present the absolute values of amplitudes, covering the whole region  $(0, 2\pi)$  [rad]. They appear as positive values in the symmetric second half  $(\pi, 2\pi)$  [rad]. The constant of  $10t.u$  as a part of the input signal  $TI(k+1)$ , corresponds to zero frequency component. This constant appears as very strong amplitude with the frequency of 0Hz in  $TI$  and  $TO$  spectrums. Besides the constant of  $10t.u.$ ,  $TI(k+1)$  consist of the sinusoidal signal, whose frequency amplitude can be seen in the spectrums of  $TI$ ,  $TO$  and  $\tau$  at 10Hz. Time amplitude of  $TO$  spectrum at 0Hz is practically the same as in spectrum of  $TI$ , because FLL attenuation, shown in Figure 7(a) for  $H_{TO}$ , is 0 [dB] at 0Hz. However the time amplitude at 10Hz is slightly amplified at the output  $TO$ , because the magnitude, at Figure 7(a), for  $H_{TO}$ , is about 0.9[dB]. According to Figure 8(b), the amplitude of the input sinus signal at 10Hz is about 560t.u. and in the spectrum of  $\tau$ , this component amounts about 200t.u. If we express this attenuation in dB, it gives  $20\log(200/560) \sim -8.6\text{dB}$ . This corresponds to attenuation of -8.6dB, as shown in Figure 7(b). Unlike the very strong amplitude at 0Hz in the spectrum of  $TO$ , this component completely disappeared in the spectrum of  $\tau$  in Figure 8(b), because FLL attenuation, shown in Figure 7(b), for  $H_{\tau}(z)$ , is about -42dB at 0Hz. This fact means that the width of time difference  $\tau$  contains the input sinus signal, but without the zero frequency component. However, according to (11),  $\tau$  depends on the initial conditions too. The difference between two adjacent  $\tau$ , i.e.  $\tau_{k+1}-\tau_k$ , will eliminate the initial conditions, since their influence is the same in every  $\tau$ , if FLL is in the stable state. The demodulated input sinus signal, without input period and all other influences is shown as  $\tau_{k+1}-\tau_k$  in Figure 9. The agreements between time and frequency presentations of FLL input and output signals, confirm the correctness of the presented analysis.

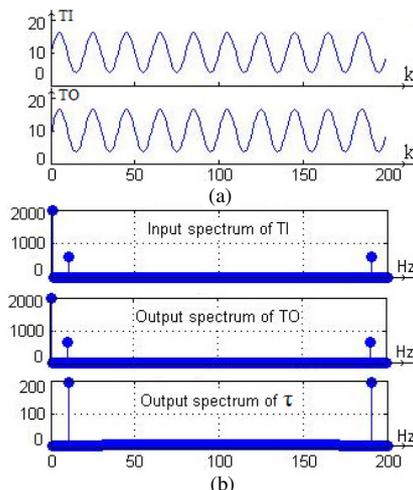


Fig. 8. (a) Time presentation of  $TI$  and  $TO$ . (b) Spectrums of  $TI$ ,  $TO$  and  $\tau$

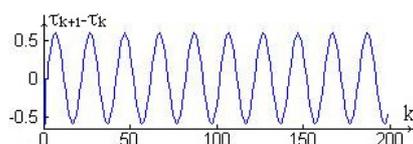


Fig. 9. Demodulated input sinus signal in time domain

## VII. CONCLUSION

This paper is closely related to the recently published articles in [1-10]. Due to the fact that this FLL version is based on measurement and processing of input periods only, it is simpler for the realization in comparison to those described in [1-10]. At the same time, it takes FLL only two steps to reach stable state for any kind of input. It was shown that this FLL can be very efficiently used for step tracking, ramp and acceleration functions. It is especially suitable for applications which require fast FLL with very short transient time. This FLL is scalable to the very strict requirements in the fields of tracking and predicting. Although TNP FLL and the digital filter represent different types of systems, since the former is based on time processing and the latter is based on processing of amplitudes, this study showed that Matlab tools devoted to the design of FIR digital filters, can be used to analyze the TNP FLL in the frequency domain. All it takes is to understand the physical aspects of the whole process and to identify the meanings of TNP FLL variables in Matlab. Using Matlab, wide options for new analyses and new applications of TNP FLL emerged. One as such is described in this article, where TNP FLL is used as a demodulator.

## ACKNOWLEDGEMENT

This article was supported by the Ministry of Science and Technology of the Republic of Serbia within the project III 47016.

## REFERENCES

- [1] D. M. Perisic, A. Zoric, Z. Gavric, "A frequency multiplier based on the Time Recursive Processing", Engineering, Technology & Applied Science Research, Vol. 7, No. 6, pp. 2104-2108, 2017
- [2] D. M. Perisic, M. Perisic, S. Rankov, "Phase shifter based on a Recursive Phase Locked Loop of the second order", Revue Roumaine Des Sciences Techniques–Serie Electrotechnique et Energetique, Vol. 59, No. 4, pp. 391-400, 2014
- [3] D. M. Perisic, A. Zoric, D. Babic, D. D. Perisic, "Recursive PLL of the First Order", Przegląd Elektrotechniczny (Electrical Review), Vol. 89, No. 7, pp. 50-53, 2013
- [4] D. M. Perisic, M. Bojovic, "Application of Time Recursive Processing for the development of the time/phase shifter", Engineering, Technology & Applied Science Research, Vol. 7, No. 3, pp. 1582-1587, 2017
- [5] D. M. Perisic, M. Bojovic, "Multipurpose Time Recursive PLL", Revue Roumaine Des Sciences Techniques–Serie Electrotechnique et Energetique, Vol. 61, No. 3, pp. 283-288, 2016
- [6] D. M. Perisic, A. Zoric, M. Perisic, D. Mitic, "Analysis and Application of FLL based on the Processing of the Input and Output Periods", Automatika, Vol. 57, pp. 230-238, 2016
- [7] D. M. Perisic, M. Perisic, D. Mitic, M. Vasic, "Time Recursive Frequency Locked Loop for the tracking applications", Revue Roumaine Des Sciences Techniques–Serie Electrotechnique et Energetique, Vol. 60, No. 2, pp. 195-203, 2015
- [8] D. M. Perisic, A. Zoric, M. Perisic, V. Arsenovic, L. Lazic, "Recursive PLL based on the Measurement and Processing of Time", Electronics and Electrical Engineering, Vol. 20, No. 5, pp. 33-36, 2014
- [9] D. M. Perisic, A. Zoric, D. Babic, D. D. Perisic, "Decoding and Prediction of Energy state in Consumption control", Revue Roumaine Des Sciences Techniques–Serie Electrotechnique et Energetique, Vol. 58, No. 3, pp. 263-272, 2013
- [10] D. M. Perisic, A. Zoric, S. Obradovic, S. Spalevic, "Application of Frequency Locked Loop in Consumption Peak Load Control", Przegląd

- Elektrotechniczny (Electrical Review), Vol. 88, No. 1b, pp. 264-267, 2012
- [11] R. N. Dean, A. K. Rane, "A Digital Frequency-Locked Loop System for Capacitance Measurement", *IEEE Transactions On Instrumentation and Measurement*, Vol. 62, No. 4, pp. 777-784, 2013
- [12] H. E. Taheri, "A 4-4.8GHz Adaptive Bandwidth, Adaptive Jitter Phase Locked Loop", *Engineering, Technology & Applied Science Research*, Vol. 7, No. 2, pp. 1473-1477, 2017
- [13] D. Jovicic, "Phase locked loop system for FACTS", *IEEE Transactions on Power Systems*, Vol. 18, No. 3, pp. 2116-2124, 2003
- [14] G. Bianchi, *Phase-Locked Loop Synthesizer Simulation*, McGraw Hill Professional, 2005
- [15] D. Abramovich, "Phase-locked loops: a control centric tutorial", *IEEE 2002 American Control Conference*, Anchorage, USA, May 8-10, 2002
- [16] R. Vich, *Z Transform Theory and Application (Mathematics and Applications)*, Springer, 1987
- [17] D. B. Tablot, *Frequency Acquisition Techniques for Phase Locked Loops*, Wiley-IEEE Press, 2012