Rev. Roum. Sci. Techn.– Électrotechn. et Énerg.
Vol. **61**, *3*, pp. 283–288, Bucarest, 2016

MULTIPURPOSE TIME RECURSIVE PHASE LOCKED LOOP

DJURDJE PERIŠIĆ[1], MIROSLAV BOJOVIĆ[2]

Key words: Phase locked loop (PLL), Frequency locked loops, Phase shifter, Digital circuits.

This work describes a new model of phase locked loop (PLL), which is based on the measurement and processing of the common time interval of the input and output periods. In the class of the time recursive PLL, this PLL is the simplest one for the realization. PLL is described by three output variables. All mathematical analyses are performed using the Z transform approach. The region of parameters for the stable system is presented. Depending on the choice of the system parameters, PLL can be used as the powerful and precise phase shifter in the noise rejection applications, in the applications with short or adapted transient time and in others. Computer simulation of PLL is made to enable better insight into the PLL properties. The realization of PLL is described.

1. INTRODUCTION

Recently described time recursive PLL and frequency locked loop (FLL) [1–5] as it was stated, represent completely a new approach to PLL and FLL in the literature in theoretical sense and in the way of their implementation and application. All of them are based on the measurement and processing of either the input and output periods or the time differences between them, or on the combination of the periods and the time differences. The scope of the powerful applicability of this approach is demonstrated in [1–5]. PLL described in [1], among other properties, it possesses very high tracking ability. The model described in [2] is the phase shifter. One software predictor is described in [3]. Digital period synthesizer is described in [4]. Although very complex system, consisting of a lot of subsystems, it can also function as a FLL like in [5]. Beside in electronic form, the most of algorithms described in [1–5] are very suitable for the usage in software form, like for example the software predictor described in [3]. The articles and books [6–16] are used as theoretical basis, for the electronics implementation and for the development and analytical necessities.

Let us remember that, according to the classical theory, a FLL generates the output signal whose frequency tends to reach the frequency of the input signal, regardless of the phase difference between the input and output signals. FLL is in the stable state as the output frequency becomes either equal or in certain pre-defined relation to the input frequency. The phase difference of FLL depends on the initial conditions of the variables and FLL parameters. Unlike FLL, in the most of the applications, classical PLL tends to equalize both frequency and phase of the input and the output signals. However, some of the time recursive PLLs possess ability to regulate the phase difference [2]. This is an important feature of the time recursive PLL, which can be designed to generate the phase difference of any value. Note that the phase difference of any PLL cannot depend on the initial conditions of the variables.

In this article a multipurpose time recursive PLL is described. Although defined by two system parameters, it is very simple for the realization. Due to the specific properties, this PLL can be widely used for different applications.

2. MATHEMATICAL DESCRIPTION OF PLL

A general case of the time relation between an input signal Sin and an output signal Sop of PLL, is shown in Fig. 1. The periods TI_0, TI_1,….TI_k, TI_{k+1}, and TO_0, TO_1,….TO_k, TO_{k+1}, as well as the time differences τ_0, τ_1, τ_2…..τ_k, τ_{k+1}, occur at discrete times respectively t_0, t_1, t_2,…..t_k, t_{k+1}. The discrete times t_0, t_1, t_2,…..t_k, t_{k+1} are defined by the falling edges of the pulses of Sop in Fig. 1. Time interval T_k, which is shaded in Fig. 1, is common for the input and output period of the same order i.e. for TI_k and TO_k. Note that the time differences τ_k and τ_{k+1} belong to the input and output periods of the different order. Due to this fact the processing of the common time interval T_k in the calculation of TO_{k+1}, provides the different system properties in comparison with the processing of the time difference τ_k. The natural recursive relations between the variables, yields from Fig. 1 and they are shown in (1)

$$\tau_{k+1} = \tau_k + TI_k - TO_k \text{ and } T_k = TO_k - \tau_k . \qquad (1)$$

The main recursive equation describing PLL is given by the (2). Note that, because of simplicity, instead of *TI(k)* it is used simple TI_k and instead of *TO(k)* it is used TO_k. This simplified notation is used throughout the article for all variables

$$TO_{k+1} = b \cdot TI_k + a \cdot T_k . \qquad (2)$$

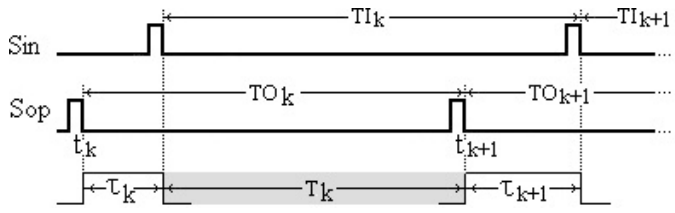

Fig. 1 – The real time relation between the input and output variables TI_k, TO_k, T_k and τ_k.

[1] Slobomir P University, Faculty of Information Technologies, Str. Pavlovića put 76, 76300 Slobomir, Republic of Srpska, Bosnia and Herzegovina, E-mail: djurdje@beotel.rs
[2] University of Belgrade, Faculty of Electrical Engineering, Serbia, E-mail: mbojovic@etf.rs

For the complete analysis of the system properties, it is now necessary to consider three output variables *TO(k)*, $\tau(k)$ and T(k), depending on *TI(k)*. Z transforms equations (1) and (2) given by (3) and (4), respectively, where τ_0 and TO_0 are the initial values of the variables τ_k and TO_k respectively;

$$z\tau(z) - z\tau_0 = \tau(z) + TI(z) - TO(z)\,,\ T(z) = TO(z) - \tau(z)\,, \tag{3}$$

$$zTO(z) - zTO_0 = b \cdot TI(z) + a \cdot T(z)\,. \tag{4}$$

It is now necessary to calculate $\tau(z)$ from the first expression of (3) and to substitute $\tau(z)$ into the second

$$TO(z) = TI(z)\frac{bz-(a+b)}{z(z-a-1)} + \frac{TO_0 z(z-1) - a\tau_0 z}{z(z-a-1)}, \tag{5}$$

$$\tau(z) = TI(z)\frac{z-(a+b)}{z(z-a-1)} - \frac{TO_0 z - \tau_0 z(z-a)}{z(z-a-1)}, \tag{6}$$

$$T(z) = TI(z)\frac{b-1}{z-a-1} + \frac{z(TO_0 - \tau_0)}{z-a-1}, \tag{7}$$

of (3) with $T(z)$. After the substitution of new expression for $T(z)$ into (4), *TO(z)* is calculated and shown by (5). After the substitution of *TO(z)* from (5) into the first expression of (3), $\tau(z)$ is calculated and shown by (6). At last, after the substitution of $\tau(z)$ from (6) into the second expression of (3), $T(z)$ is calculated and shown by (7).

Providing that the step function is applied to the input *TI(k)* = *TI* = const., it is necessary to substitute the Z transform $TI(z) = TI \cdot z/(z-1)$ into (5)–(7). By substituting *TI(z)* into (5) and using the final value theorem, it is possible to find the final value of the output period $TO_\infty = \lim TO(k)$ if $k\to\infty$, using *TO(z)*, which is shown by (8).

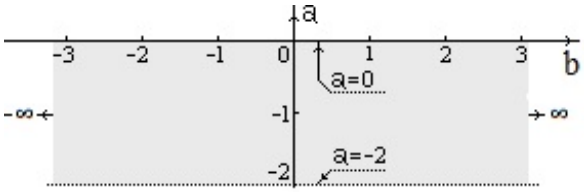

Fig. 2 – The region of parameters a and b for the stable PLL.

In the same way the final value τ_∞ of $\tau(k)$ and the final value T_∞ of *T(k)* are calculated using (6) and (7) and shown by, (9) and (10), respectively. According to (8)–(10), no one of the output final values depends on the initial conditions of variables. According to (8), $TO_\infty = TI$ for the stable system. It can be concluded that the system possesses PLL properties. This conclusion is valid only if the system is stable *i.e.* if $|z_1| < 1$ and $|z_2| < 1$, where z_1 and z_2 are the poles of PLL, *i.e.* the zeros of the polynomial $z(z-a-1)$ in (5) and (6). According to the given polynomial, $z_1 = a+1$ and $z_2 = 0$. The transfer function $H_T(z)=T(z)/TI(z)=(b-1)/(z-a-1)$, given by (7), is of the first order. It possesses only one pole $z_1 = a+1$. PLL is the stable system if $-2 < a < 0$. Note that parameter b does not affect the system stability. That means b can take any

$$TO_\infty = \lim[TO(k)]_{k\to\infty} = \lim[(z-1)\cdot TO(z)]_{z\to 1} = TI\,, \tag{8}$$

$$\tau_\infty = \lim[\tau(k)]_{k\to\infty} = \lim[(z-1)\cdot\tau(z)]_{z\to 1} = \frac{TI(a+b-1)}{a}, \tag{9}$$

$$T_\infty = \lim[T(k)]_{k\to\infty} = \lim[(z-1)\cdot T(z)]_{z\to 1} = \frac{TI(1-b)}{a} \tag{10}$$

value. The regain of parameters a and b which satisfy the condition for the system stability is shown in Fig. 2. On the other hand, parameter b takes part in the determination of τ_∞ and T_∞, given by (9) and (10).

3. REALIZATION OF PLL

This algorithm provides ability to keep the same sign of the common time interval T during one locking procedure. This enormously facilitates the realization of PLL. To analyze the behavior of T in time domain, let us find the inverse Z transform of $T(z)$ given by (7). Providing that the step function $TI(z) = TI \cdot z/(z-1)$ is fed to PLL, (7) can be transformed into more suitable form, which is shown in (11). The inverse Z transform of (11) is shown by (12). It represents time interval T in time domain. In order to verify some previous results, let us find "T_∞" using (12). According to (12), $T_\infty = \lim T(k)_{k\to\infty} = TI(1-b)/a$, which is in accordance with the expression shown by (10). The expression (12) will be also used for the checking of the simulation results. The first part of (12) never changes sign during the locking, because $[1-(a+1)^k]$ is always greater than zero. If $b<1$, the first part is always negative, and if $b>1$, this part is always positive. The second part of (12), disappears during the locking for $k\to\infty$. For some special initial conditions, the second part might change the sign of T, but only at the very beginning of the locking procedure, which will not affect the final result of locking. Note, if it is necessary for

$$T(z) = \frac{TI(1-b)/a}{z-1} - \frac{TI(1-b)\cdot(a+1)/a}{z-(a+1)} + \frac{z(TO_0-\tau_0)}{z-(a+1)}, \tag{11}$$

$$T(k) = \frac{TI(1-b)}{a}[1-(a+1)^k] + (TO_0 - \tau_0)\cdot(a+1)^k \tag{12}$$

any specific application, the initial conditions TO_0-τ_0 can be reduced to zero, either electronically at the very beginning, if the realization of PLL is performed by the standard IC circuits, or by software if the algorithm is used in software form.

In order to realize PLL with the standard IC circuits, let us modify (2) into (13) substituting $a = -f_a/f_c$ and $b = f_b/f_c$, where f_a, f_b and f_c are the frequencies of three clock signals. Note that parameter a is preferably presented with the negative value, because it is always negative for the stable PLL. Parameter b can be either positive or negative. One should take care of the sign of parameter b and use the proper sign in (13). If now the frequencies f_c, f_b and f_a in (13) are changed by theirs corresponding periods respectively $t_c = 1/f_c$, $t_b = 1/f_b$ and $t_a = 1/f_a$, (13) will be modified into (14). (14) is more suitable to discover how to realize PLL using the standard digital circuits. According to (14), the input period TI_k is measured by clock periods t_b, time interval T_k is measured by clock periods t_a, and the output period TO_k is to be realized by clock periods t_c. The simplified model of PLL,

$$TO_{k+1}\cdot f_c = TI_k \cdot f_b - T_k \cdot f_a\,, \tag{13}$$

$$TO_{k+1}/t_c = TI_k/t_b - T_k/t_a, \tag{14}$$

corresponding to eq. (14), is presented in Fig. 3. Recursive calculation model (RCM) measures and calculates the sum of the right side of (14) at discrete time t_k. In the next step, programmable period generator (PPG) generates the output period TO_{k+1} at the discrete time t_{k+1}.

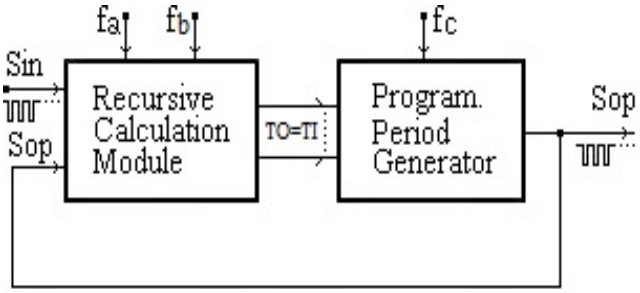

Fig. 3 – The simplified model of PLL described.

The technique for the realization of this type of PLL and their real time functioning are presented in [1, 2, 4]. Clock signals are presented by their frequencies f_a, f_b and f_c in Fig. 3.

4. APPLICATIONS OF PLL

In this chapter, some of the applications of PLL will be considered. Beside the usage of the previous analyses, these considerations will be supported by the simulation of PLL operations. The simulations are to prove the mathematical analyses, to discover additional properties and possible efficient applications of PLL and to enable better insight into the procedure and the physical meaning of the variables described. All discrete values in simulations were merged to form continuous curves. The variables in the following diagrams were presented in time units. The time unit can be, μs, ms or any other, but assuming the same time units for *TI*, *TO*, *T* and τ. 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 initial values of the variables are selected randomly and they are indicated in the diagrams.

The simulations were performed using (2) and (1).

Let us first consider how PLL can operate as the phase shifter. Note that, the final values τ_∞ and T_∞, according to (9) and (10), are linearly dependent on parameter *b*. If PLL is in the stable state, $\tau_\infty + T_\infty = TO_\infty = TI$. From the application point of view, this means that, choosing the corresponding parameter *b*, PLL is able to provide shifting of the input pulse rate, without disarranging the system stability. Since *b* can take any value, PLL can provide any time shifting from minus infinity to plus infinity. The limitations are the capacity and required resolution of PLL designed.

The simulation of *TO(k)*, *TI(k)*, τ(*k*) and *T*(*k*) for the input pulse rate Sin with the period $TI = 10$ t.u, than for $a = -0.75$ and $b = 1.45$ is shown in Fig. 4a. The initial conditions are $\tau_0 = 4$ t.u and $TO_0 = 8$ t.u. It can be seen that $TO_\infty = TI$ in Fig 4a, proving (8). According to Fig. 1, it was supposed that the output signal was backward shifted for the time difference τ. When PLL is in the stable state, according to (9), the backward shifting for this case is $\tau_\infty = TI\cdot(a+b-1)/a = 10\cdot(-0.75+1.45-1)/(-0.75) = 4$ t.u. This result agrees with the simulated τ_∞ in Fig. 4a. The backward phase shifting for this case can be calculated as $Ph = 2\pi\cdot(\tau_\infty/TO_\infty)$ rad $= 2\pi\cdot 4/10$ rad $= 144$ degrees. According to (10), the common time interval $T_\infty = 6$ t.u. To illustrate the real time transient state of the shifting procedure, the values of Sin, Sop, τ_k and T_k, are presented in their real time relations. Substituting *TI*, the initial conditions and the values for the system parameters *a* and *b* in (12), it was found out that $T(k) = 6-2\cdot(0.25)^k$. It can be calculated that $T(0) = 4$ t.u, $T(1) = 5.5$ t.u, $T(2) = 5.875$ t.u and $T(3) = 5.96875$ t.u. All these calculated values agree with the simulated *T*(*k*), shown in Fig. 4b. The identity of the analytical and simulation results proves the correctness of the entire theoretical approach, as well as the validity of the results obtained.

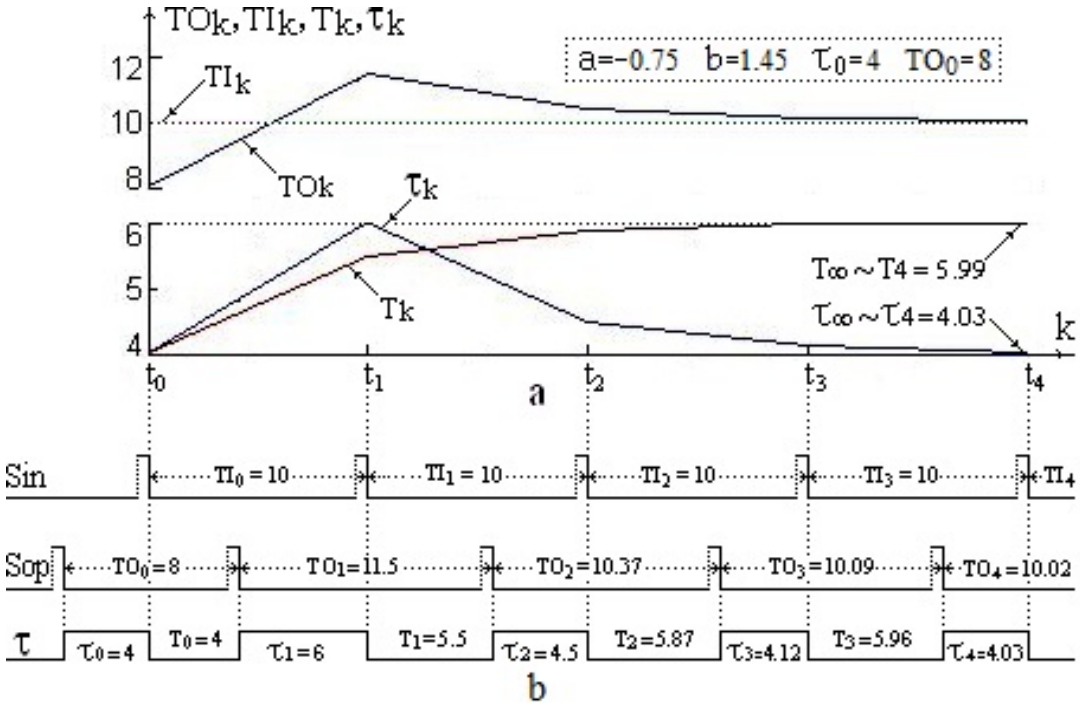

Fig. 4 – a) Simulation of the transient state of PLL backward shifting for 144 degrees;
b) real time presentation of the simulated variables Sin, Sop, τ_k and T_k for every step of the transient state.

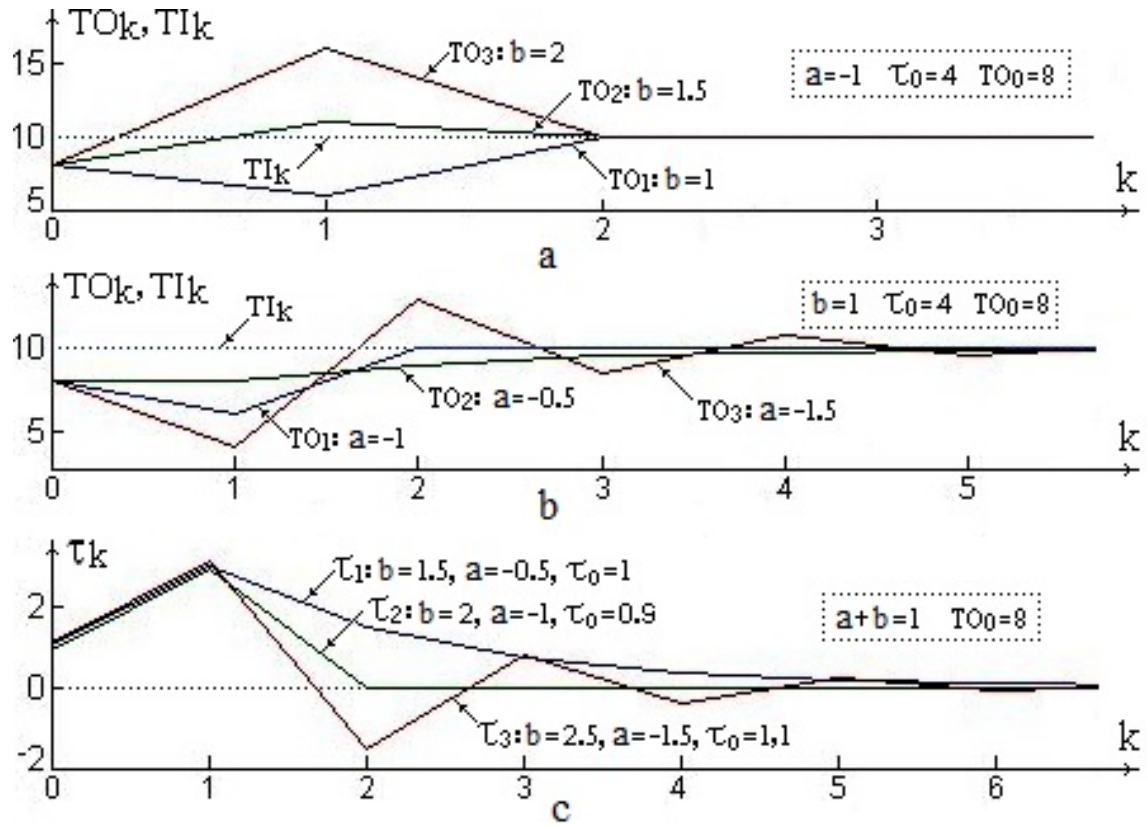

Fig. 5 – a) Simulation of *TO(k)* for $a = -1$ and different b ; b) simulation of *TO(k)* for $b = 1$ and different a; c) simulation of the time interval $\tau(k)$ for $a+b = 1$.

PLL described is suitable in those applications where extremely fast locking is required. The simulation in Fig. 5a has shown, that for $a = -1$, all locking procedures are finished only for two steps. All initial conditions and the values of parameters are presented in Fig 5a. It can be seen that for the different values of parameters b, the number of locking steps is always two. For the different values of the parameter b, PLL will possess the various properties. Due to this specific characteristic, PLL can provide the fast locking procedure and, at the same time, reach the different application requirements. Fig. 5b demonstrates how the speed of the locking procedure depends on the parameter a, for $b = 1 = \text{constant}$.

PLL described can be used as a classical PLL, which always generates the phase difference equal to zero in the stable state. According to (9), providing that $a+b = 1$, the time difference τ_∞ is always zero. This property of PLL is shown in Fig. 5c. For the different values of the parameters a and b, which satisfy the condition $a+b = 1$, the phase difference $Ph=2\pi\cdot(\tau_\infty/TO_\infty)$ rad is always zero. Note that for any required value of the time difference, a wide range of combinations of the parameters a and b is offered. The wider choice of parameters offers the better adapting of PPL properties to the application requirements.

PLL described is also suitable for the application in the noisy environments. In dependence of the chosen values of the parameters, it can provide the powerful noise rejection. To demonstrate this property of PLL, the simulation of its noise suppression ability is presented in Fig. 6. The input step, consisting of 10 time-units, which is strongly corrupted by uniformly distributed noise, is shown in Fig. 6a.

The amplitude of noise is 10 time-units peak to peak. According to (2), the noisy input period and the common time interval T take part in the calculation of the output period. According to (1), T is also corrupted by noise, because it depends on TI. The suppression of a noise in every next output period will be stronger if the participations of the noisy input period and the noisy time interval T are smaller in the calculation. According to (2), the participation of the noisy input period and noisy time interval are proportional to the values of the parameters respectively b and a. It comes out that the noise suppression at the output periods will be better for the smaller values of the parameters b and a. However the parameters must belong to the regain of the parameters for the stable PLL, shown in Fig. 2. Following this conclusion, two cases of the output periods are simulated for the different values of the parameter b and for $a = -0.01 = \text{const}$. For the output period TO_1, $b = 0.3$. For the output period TO_2, $b = 0.01$. Periods TO_1 and TO_2 are presented respectively in Fig. 6b and in Fig. 6c. The noise suppression is considerable for TO_1. However if both parameters are very small, like in case for TO_2, the noise suppression is very powerful. The simulation results agree with the previous conclusions. However, it can also be seen from Fig. 6b and Fig 6c, that in case of better noise suppression, the transition time of PLL is longer. This conclusion is also in agreement with the previous one, because if the parameters are smaller, PLL takes longer time to reach the stable state.

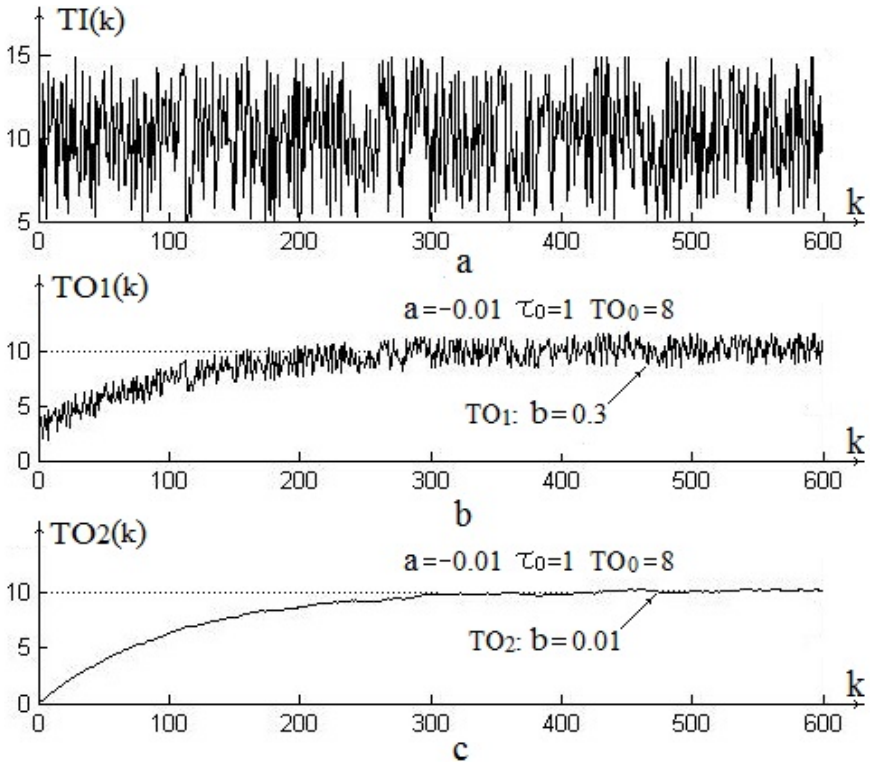

Fig. 6 – Simulation of the noise rejection performance of PLL: a) The input step is strongly corrupted by noise; b) The noise rejection is considerable at the output; c) The noise rejection is very powerful.

5. **CONCLUSIONS**

The description and illustrations of the realized PLL represent a new approach to the design of time recursive PLL. Unlike to the approaches in [1–5], this approach is based on the processing of the common time interval of the input and output period. Due to this approach, PLL provides some new properties in comparison with those described in [1–5]. Accordingly, PLL is suitable for a wide range of applications. It was shown that it could be used more efficiently for the phase shifting applications, than the phase shifter described in [2]. The phase shifter described in [2], uses, for the phase shifting, one system parameter and one outside parameter, which does not affect the PLL properties. This PLL uses two system parameters. For the corresponding shifting, this PLL provides a large number of the combination of the system parameters, enabling additional adjusting of PLL properties to the specific requirements. It was also shown, that this PLL is considerable simpler for realization than those described in [1–5]. The transient time can be regulated by parameter a. For $a = -1$, no matter of the value of the parameter b, PLL takes only two steps to reach the stable state. At the same time, the wide choice of parameter b can be used independently, to adjust the other properties of PLL. Because of that, PLL is suitable for the applications in which fast locking procedure is required. If $a+b = 1$, PLL operate as the classical PLL. For the corresponding values of the system parameters, PLL provides the powerful noise rejection.

It was demonstrated in this article, that any compositional time interval of the input and output periods could be analyzed independently in Z domain, as well as in the time domain. The identity of the analytical and simulation results, in every step, proves the correctness of the entire theoretical approach, presented in the article. This provides an idea, that besides PLL, this theoretical approach might be used for the development of some other kind of systems, based on the time recursive processing.

ACKNOWLEDGEMENTS

This article was supported by the Ministry of Science and Technology of the Republic of Serbia within the project TR 32047.

Received on September 5, 2015

REFERENCES

1. Dj.M. Perisic, A. Zoric, M. Perisic, V. Arsenovic, Lj. Lazic, *Recursive PLL based on the Measurement and Processing of Time*, Electronics and Electrical Engineering, **20**, 5, pp. 33–36 (2014).
2. Dj.M. Perisic, M. Perisic, S. Rankov, *Phase Shifter based on a recursive Phase Locked Loop of the second order,* Rev. Roum. Sci. Techn. – Électrotechn. et Énerg., **59**, *4*, pp. 391–400 (2014).
3. Dj.M. Perisic, A. Zoric, Dj. Babic, Dj. Dj. Perisic, *Decoding and Prediction of Energy State in Consumption Control*, Rev. Roum. Sci. Techn. – Electrotechn. et Energ., **58**, *3*, pp. 263–272 (2013).
4. Dj.M. Perisic, A. Zoric, S. Obradovic, Dj. Dj. Perisic, *FLL as Digital Period Synthesizer based on Binary Rate Multiplier Control*, Electrical Review, **R.89**,*1a*, pp. 145–148 (2013).
5. Dj. M. Perisic, A. Zoric, S. Obradovic, P. Spalevic, *Application of Frequency Locked Loop in Consumption Peak Load Control*, Electrical Review, **R.88**, *1b*, pp. 264–267 (2012).
6. D. Jovcic, *Phase locked loop system for FACTS*, IEEE Transaction on Power System, **18**, pp. 2185–2192 (2003).
7. A.S.N. Mokhtar, B.B.I. Reaz, M. Maruffuzaman, M.A.M. Ali, *Inverse park transformation using Cordic and Phase-Locked Loop,* Rev. Roum. Sci.Techn. – Electrotechn. et Energ., **57**, *4*, pp. 422–431 (2012).

8. C.C. Chung, *An all-digital phase-locked loop for high speed clock generation*, IEEE Journal of Solid-State Circuits, **38**, *2*, pp. 347–359 (2003).
9. S. Arezki, M. Boudour, *Study and regulation of DC Bus Voltagesof Wind-Photovoltaic system,* Rev. Roum. Sci. Techn. – Electrotechn. et Energ., **59**, *1*, pp. 35–46 (2014).
10. L. Joonsuk, B. Kim, *A low noise fast-lock phase-locked loop with adaptive bandwith control-Solid-State Circuit*, IEEE Journal, **35**, *8*, pp. 1137–1145 (2000).
11. D. Abramovitch, *Phase-locked loops: a control centric tutorial*, Amecan Control Conference-2002, Proceedings of 2002, Vol. 1, pp. 1–15 (2002).
12. R. Vich, *Z Transform Theory and Application (Mathematics and Applications)*, First Edition Springer, 1987.
13. G. Bianchi, *Phase-Locked Loop Synthesizer Simulation*, Mc Grow-Hill, Inc. New York, USA, 2005.
14. B.D. Talbot, *Frequency Acquisition Techniques for PLL,* Wiley, IEEE Press, 2012.
15. C.B. Fledderman, *Introduction to Electrical and Computer Engineering*, Prentis Hall, 2002.
16. M. Gardner, *Phase lock techniques*, Hoboken, Wiley-Interscience, 2005.