Charge pump phase-locked loop with phase-frequency detector: closed form mathematical model

Kuznetsov N.\textsuperscript{a,b,c}, Yuldashev M.\textsuperscript{a}, Yuldashev R.\textsuperscript{a}, Blagov M.\textsuperscript{a}, Kudryashova E.\textsuperscript{a}, Kuznetsova O.\textsuperscript{a}, Mokaev T\textsuperscript{a}.

Abstract—Charge pump phase-locked loop (CP-PLL) is an electrical circuit, widely used in digital systems for frequency synthesis and synchronization of the clock signals. In this paper a non-linear second-order model of CP-PLL is rigorously derived. The obtained model obviates the shortcomings of previously known second-order models of CP-PLL. Pull-in time is estimated for the obtained second-order CP-PLL.

I. INTRODUCTION

PHASE-locked loops (PLLs) are electronic circuits, designed for generation of an electrical signal (voltage), while the frequency is automatically tuned to the frequency of an input (reference, Ref) signal. Charge pump phase-locked loop with phase-frequency detector (Charge pump PLL, CP-PLL) is widely used in digital systems for frequency synthesis and synchronization of the clock signals \cite{3}. The CP-PLL is able to quickly lock onto the phase of the incoming signal, achieving low steady-state phase error \cite{7}. Important issues in the design of PLL are estimation of the ranges of deviation between oscillators frequencies for which a locked state can be achieved, the stability analysis of the locked states, and study of possible transient processes. The pioneering monographs \cite{11}, \cite{31}, \cite{35} were published in 1966, a rather comprehensive bibliography \cite{24} was published in 1973, and recent surveys \cite{4}, \cite{23}. For the corresponding study of the CP-PLL F. Gardner developed in 1980 a linearized model in vicinity of locked states \cite{10}, \cite{12}. Then an approximate discrete-time linear models of the CP-PLL were suggested in \cite{17}, \cite{25}. However, linear models are essentially limited, since only local behavior near locked states can be studied. For the study of non-local behavior and transient processes some nonlinear second-order models of the CP-PLL were developed in \cite{2}, \cite{7}, \cite{34}.

Examples from section V demonstrate that algorithm and formulas suggested by M. van Paemel in \cite{34} should be used carefully for simulation even inside allowed area (see Fig. 18 and Fig. 22 in original paper \cite{34}). While the

examples are given for the first time, the main idea of Example 1 was already noticed by P. Acco and O. Feely \cite{1}, \cite{9}. P. Acco and O. Feely considered only near-locked state, therefore they didn’t notice problems with out-of-lock behavior. Example 2 and Example 3 demonstrate problems with out-of-lock behavior, which was not discovered before.

Note that while derivation of non-linear mathematical models for high-order CP-PLL requires numerical solution of non-linear algebraic equations or allows to find only approximate solutions (see, e.g. \cite{6}, \cite{8}, \cite{13}, \cite{14}, \cite{16}, \cite{19}, \cite{33}, \cite{36}), non-linear mathematical models for the second order CP-PLL can be found in closed-form. Further, we consider only the second order CP-PLL. Noise performance and simulation of PLL is discussed in \cite{3}, \cite{26}, \cite{28}, \cite{29}, \cite{32}.

II. A MODEL OF CHARGE PUMP PHASE-LOCKED LOOP WITH PHASE-FREQUENCY DETECTOR IN THE SIGNAL SPACE

Consider charge pump phase-locked loop with phase-frequency detector \cite{10}, \cite{12} on Fig. 1. Both reference and output of the voltage controlled oscillator (VCO) are square waveform signals, see Fig. 2.

![Waveforms of the reference and voltage controlled oscillator (VCO) signals are periodic functions with period equal to one. Trailing edges happen at the integer values of corresponding phases.](image)

Without loss of generality we suppose that trailing edges of VCO and reference signals occur when corresponding phase reaches an integer number. The frequency $\omega_{\text{ref}}$ of reference signal (reference frequency) is usually assumed to be constant:

$$\theta_{\text{ref}}(t) = \omega_{\text{ref}}t = \frac{t}{T_{\text{ref}}},$$  \hspace{1cm} (1)

where $T_{\text{ref}}$, $\omega_{\text{ref}}$ and $\theta_{\text{ref}}(t)$ are the period, frequency and phase of reference signal.
The phase-frequency detector (PFD) is a digital circuit, triggered by the trailing (falling) edges of the reference Ref and VCO signals. The output signal of PFD \( i(t) \) can have only three states (Fig. 3): 0, \(+I_p\), and \(-I_p\).

To construct a mathematical model, we wait for first trailing edge of reference signal and define the corresponding time instance as \( t = 0 \). Suppose that before \( t = 0 \) the PFD had a certain constant state \( i(0) \). A trailing edge of the reference signal forces PFD to switch to higher state, unless it is already in state 0. A trailing edge of the VCO signal forces PFD to switch to lower state, unless it is already in state \(-I_p\). If both trailing edges happen at the same time then PFD switches to zero.

Thus, \( i(0) \) is determined by the values \( i(0) \), \( \theta_{vco}(0) \), and \( \theta_{ref}(0) \). Similarly, \( i(t) \) is determined by \( i(t) \), \( \theta_{vco}(t) \), and \( \theta_{ref}(t) \). Thus, \( i(t) \) is piecewise constant and right-continuous\(^1\).

\(^1\)approaching any number from the right yields the same value of \( i(t) \)

The relationship between the input current \( i(t) \) and the output voltage \( v_F(t) \) for a proportionally integrating (perfect PI) filter based on resistor and capacitor

\[
H(s) = R + \frac{1}{Cs}
\]

is as follows

\[
v_F(t) = v_c(0) + Ri(t) + \frac{1}{C} \int_0^t i(\tau)d\tau,
\]

where \( R > 0 \) is a resistance, \( C > 0 \) is a capacitance, and \( v_c(t) = v_c(0) + \frac{1}{C} \int_0^t i(\tau)d\tau \) is a capacitor charge.

The control signal \( v_F(t) \) adjusts the VCO frequency:

\[
\dot{\theta}_{vco}(t) = \omega_{vco}(t) = \omega_{vco}^{\text{free}} + K_{vco}v_F(t),
\]

where \( \omega_{vco}^{\text{free}} \) is the VCO free-running (quiescent) frequency (i.e. for \( v_F(t) = 0 \)), \( K_{vco} \) is the VCO gain (sensitivity), and \( \dot{\theta}_{vco}(t) \) is the VCO phase. Further we assume that\(^2\)

\[
\dot{\theta}_{vco}(t) = \omega_{vco}(t) > 0.
\]

If the frequency of VCO \( \omega_{vco}(t) \) is much higher than the frequency of reference signal \( \omega_{ref} \), then trailing edges of VCO forces PFD to be in the lower state \(-I_p\) most of the time. In this case the output of loop filter \( v_F(t) \) is negative. Negative filter output, in turn, decrease the VCO frequency to match the reference frequency. Similarly, if the VCO frequency is much lower than the reference frequency, the filter output becomes mostly positive, increasing the VCO frequency. If the VCO and reference frequencies are close to each other, then the transient process may be more complicated. In this case the CP-PLL either tends to a locked state or to an unwanted oscillation.

From [1], [2], and [3], for given \( i(0) \) and \( \omega_{ref} \) we obtain a continuous time nonlinear mathematical model of CP-PLL described by differential equations

\[
\dot{v}_c(t) = \frac{1}{C}i(t),
\]

\[
\dot{\theta}_{vco}(t) = \omega_{vco}^{\text{free}} + K_{vco}(Ri(t) + v_c(t)),
\]

\(^2\)In the following sections we suggest equations which allow one to check that the overload is not happened.
with discontinuous piecewise constant nonlinearity

\[ i(t) = i(i(t^-)) + \omega_{ref}, \theta_{vco}(t) \]

and initial conditions \((v_c(0), \theta_{vco}(0))\).

III. DERIVATION OF DISCRETE TIME CP-PLL MODEL

Here we derive discrete time model of the CP-PLL in the following form

\[ \tau_{k+1} = \varphi(\tau_k, v_k), \]

\[ v_{k+1} = v(\tau_k, v_k). \]

First, let us define the state variables \(\tau_k, v_k\).

Let \(t_0 = 0\). Denote by \(t_{0}\) the first instant of such that the PFD output becomes equal to zero\(^3\) if \(i(0) = 0\) then \(t_{0} = 0\). Then we wait until the first trailing edge of the VCO or Ref and denote corresponding moment of time by \(t_1\). Continuing in a similar way, one obtains increasing sequences \(\{t_k\}\) for \(k = 0, 1, 2, \ldots\) (see Fig. 4).

Let \(t_k \leq t_{k+1}\). Then for \(t \in [t_{k}, t_{k+1})\) sign \((i(t))\) is a non zero constant \((\pm 1)\). Denote by \(\tau_k\) the PFD pulse width (length of time interval, where PFD output is non-zero constant), multiplied by the sign of PFD output (see Fig. 5):

\[ \tau_k = (t_{k+1} - t_k) \text{sign} (i(t)), \quad t \in [t_k, t_{k+1}), \]

\[ \tau_k = 0 \quad \text{if} \quad t_k = t_{k+1}. \]  

Finally, from (2) we get

\[ v_F(t) = \begin{cases} v_k, & t \in [t_{k+1}, t_{k+2}), \\ v_k \pm RL_p \pm I_p (t - t_{k+1}), & t \in [t_{k+2}, t_{k+3}). \end{cases} \]

Combining (3) and (8) we obtain

\[ \omega_{vco}(t) = \begin{cases} \omega_{vco}^{free} + K_{vco} v_k, & t \in [t_{k+1}, t_{k+2}), \\ \omega_{vco}^{free} + K_{vco} (v_k \pm RL_p \pm I_p (t - t_{k+1})), & t \in [t_{k+2}, t_{k+3}). \end{cases} \]

where the sign \(- \text{or} +\) in the last equation corresponds to \(\text{sign}(i(t))\).

By (8) and (6) the value \(v_{k+1}\) can be expressed via \(\tau_{k+1}\) and \(v_k\):

\[ v_{k+1} = v_k + I_p C \tau_{k+1}. \]

To find \(\tau_{k+1}\) one needs to consider four possible cases of PFD transitions (see Fig. 6):

1) \(\tau_k \geq 0, \tau_{k+1} \geq 0\);

2) \(\tau_k \geq 0, \tau_{k+1} < 0\);

3) \(\tau_k < 0, \tau_{k+1} \leq 0\);

4) \(\tau_k < 0, \tau_{k+1} > 0\).

Remark that the PFD output \(i(t)\) always returns to zero from non-zero state at certain time. If \(i(t_0) = -1\), then the first Ref trailing edge returns the PFD output to zero. If \(i(t_0) = 1\), then the VCO frequency is increasing until the first VCO trailing edge returns the PFD output to zero.

Fig. 4: Explanation of \(t_k\) and \(t_{k+1}\).

Fig. 5: Definition of discrete states \(\tau_k\) and \(v_k\). \(l_k\) is a PFD pulse width.

Fig. 6: Four cases of PFD states transitions

Case 1: \(\tau_k \geq 0\) and \(\tau_{k+1} \geq 0\) (see Fig. 28).
(a) Case 1–1: general case, Reference cycle slipping; \( \tau_k \geq 0, \tau_{k+1} \geq 0 \).

Fig. 7: Subcases of the Case 1. Integral of the VCO frequency \( \omega_{vco} \) over the VCO period \( T_{vco} \) is pink subgraph area (grey in black/white). The integral is equal to 1 according to the PFD switching law and definition of time intervals.

First, rewrite requirement \( \tau_{k+1} \geq 0 \) in terms of \( \tau_k \) and \( v_k \). Since \( \omega_{vco}(t) > 0 \), condition \( \tau_{k+1} > 0 \) takes the form

\[
\theta_{vco}(t_{k+1}) - \theta_{vco}(t_k) \leq 1. \tag{11}
\]

By \( [6] \) we get

\[
\theta_{vco}(t_{k+1}) - \theta_{vco}(t_k) = \int_{t_k}^{t_{k+1}} \omega_{vco}(\tau) \, d\tau = (T_{ref} - (\tau_k \mod T_{ref})) \left( \omega_{vco}^{free} + K_{vco}v_k \right) \tag{12}
\]

Thus, condition \( \tau_{k+1} \geq 0 \) can be expressed via \( v_k \) and \( \tau_k \) in the following way:

\[
(T_{ref} - (\tau_k \mod T_{ref})) \left( \omega_{vco}^{free} + K_{vco}v_k \right) \leq 1. \tag{13}
\]

Now find \( \tau_{k+1} \). VCO trailing edges appeared twice on time interval \( [t_k, t_{k+2}) \): at \( t = t_k \) and \( t = t_{k+1} \). Thus, we get

\[
\theta_{vco}(t_k) - \theta_{vco}(t_k^{middle}) = 1. \tag{14}
\]

By \( [3] \) it is equivalent to

\[
\int_{t_k}^{t_k^{middle}} \omega_{vco}(t) \, dt + \int_{t_k^{middle}}^{t_{k+1}} \omega_{vco}(t) \, dt = 1. \tag{15}
\]

By definition of \( \tau_k \) \( [6] \) we get (see Fig. 28a)

\[
t_{k+1} = t_k^{middle} - (\tau_k \mod T_{ref}) + T_{ref}, \quad t_k^{middle} = t_k + \tau_k. \tag{16}
\]

4 Here \( (a \mod b) \) is the remainder after division of \( a \) by \( b \), where \( a \) is the dividend and \( b \) is the divisor. This function is often called the modulo operation.

Substituting \( [16] \) and \( [9] \) into \( [15] \) and calculating the integral (i.e., shaded area in Fig. 28a) we get the following relation for \( \tau_{k+1} \):

\[
(T_{ref} - (\tau_k \mod T_{ref}) + \tau_{k+1}) \left( \omega_{vco}^{free} + K_{vco}v_k \right) + K_{vco}I_p \left( R\tau_{k+1} + \frac{1}{2C} \omega_{vco}^{2} \right) = 1. \tag{17}
\]

Equation \( [VIII] \) is quadratic with discriminant

\[
b(v(k))^2 - 4ac(\tau_k, v_k), \tag{18}
\]

where

\[
a = \frac{K_{vco}I_p}{2C}, \quad b = b(v_k) = \omega_{vco}^{free} + K_{vco}v_k + K_{vco}I_p R, \quad c = c(\tau_k, v_k) = (T_{ref} - (\tau_k \mod T_{ref})) \left( \omega_{vco}^{free} + K_{vco}v_k \right) - 1. \tag{19}
\]

Relation \( [13] \) is equivalent to \( c \leq 0 \), which means that the discriminant is non-negative. Therefore, equation \( [VIII] \) has exactly one positive solution:

\[
b + \sqrt{b^2 - 4ac} \frac{2a}{2a}. \tag{20}
\]

Case 2: \( \tau_k \geq 0 \) and \( \tau_{k+1} < 0 \) (Fig. 29). The first VCO edge appears at \( t = t_k^{middle} \), and the second VCO edge appears at \( t = t_{k+1} \).

Fig. 8: Subcases of the Case 2. Integral of the VCO frequency \( \omega_{vco} \) over the VCO period \( T_{vco} \) is pink subgraph area (grey in black/white). The integral is equal to 1 according to the PFD switching law and definition of time intervals.

Similarly to the previous case, we can derive a relation for \( \tau_{k+1} \) by integrating VCO frequency \( \omega_{vco}(t) \) over its period \( [t_k^{middle}, t_{k+1}] \):

\[
(T_{ref} + \tau_{k+1} - (\tau_k \mod T_{ref})) \left( \omega_{vco}^{free} + K_{vco}v_k \right) = 1. \tag{17}
\]
Therefore
\[ \tau_{k+1} = \frac{1}{\omega_{vco} + K_{vco}v_k} - T_{ref} + (\tau_k \mod T_{ref}). \] (21)

**Case 3:** \( \tau_k < 0 \) and \( \tau_{k+1} \leq 0 \) (see Fig. 30).

First, we determine the sign of \( \tau_{k+1} \). To do that we need to find out which of the signals (VCO or reference) reaches its period first after \( t_k \) middle, i.e. to check if \( l_b = t_{k+1} - t_k \) middle \( \leq T_{ref} \). By (9) for the time interval \( l_b \) we get (see Fig. 30)

\[ l_b = l_b(v_k) = \frac{S_{l_b}}{K_{vco}v_k + \omega_{vco}}, \] (22)

where

\[ S_{l_b} = \int_{t_k \text{ middle}}^{t_{k+1} \text{ middle}} \omega_{vco}(t)dt = 1 - S_{l_a}. \] (23)

Here \( S_{l_a} \) (green area in Fig. 30) is computed as a fractional part of the subgraph area corresponding to \( l_k = -\tau_k \):

\[ S_{l_a} = S_{l_a}(\tau_k, v_k) = S_{l_k} \mod 1, \]
\[ S_{l_k} = S_{l_k}(\tau_k, v_k) = \]
\[ = (K_{vco}v_k - I_p R)K_{vco} + \omega_{vco})l_k + K_{vco}I_p \frac{l_k^2}{2C}. \] (24)

Finally, we get

\[ \tau_{k+1} = \frac{1 - S_{l_a}}{K_{vco}v_k + \omega_{vco}} - T_{ref}. \] (25)

**Case 4:** \( \tau_k < 0 \) and \( \tau_{k+1} > 0 \) (see Fig. 31).

(a) Case 4 1: general case, VCO and Reference cycle slipping; \( \tau_k < 0, \quad \tau_{k+1} > 0 \)

Fig. 10: Subcases of the Case 4. Integral of the VCO frequency \( \omega_{vco} \) over the VCO period \( T_{vco} \) is pink subgraph area (grey in black/white). The integral is equal to 1 according to the PFD switching law and definition of time intervals.

Similar to Case 3 condition \( \tau_{k+1} > 0 \) is equivalent to

\[ l_b > T_{ref}, \] (26)

where \( l_b = t_{k+1} \text{ middle} - t_k \text{ middle} \) can be computed in the following way

\[ l_b = l_b(\tau_k, v_k) = \frac{1 - S_{l_a}}{K_{vco}v_k + \omega_{vco}}, \]
\[ S_{l_a} = S_{l_a}(\tau_k, v_k) = S_{l_k} \mod 1, \]
\[ S_{l_k} = S_{l_k}(\tau_k, v_k) = \]
\[ = (K_{vco}v_k - I_p R)K_{vco} + \omega_{vco})l_k + K_{vco}I_p \frac{l_k^2}{2C}. \] (27)

Now we can compute the VCO phase corresponding to its full period:

\[ S_{l_a} + S_{l_{ref}} + S_{l_{k+1}} = 1, \]
\[ S_{l_{ref}} = S_{l_{ref}}(v_k) = T_{ref}(K_{vco}v_k + \omega_{vco}), \]
\[ S_{l_{k+1}} = S_{l_{k+1}}(\tau_{k+1}, v_k) = \tau_{k+1} \left( K_{vco}v_k + \omega_{vco} + I_p R K_{vco} \right) + \]
\[ + \frac{I_p K_{vco} r^2(k+1)}{2C}. \] (28)

From (26) there is only one positive solution \( \tau_{k+1} \) of quadratic equation

\[ \tau_{k+1} = \frac{-b + \sqrt{b^2 - 4ad}}{2a}, \]
\[ a = \frac{K_{vco} I_p}{2C}, \]
\[ b = b(v_k) = \omega_{vco} + K_{vco}v_k + K_{vco} I_p R, \]
\[ d = b(\tau_k, v_k) = S_{l_a} + S_{l_{ref}} - 1, \] (29)
IV. Corrected Full Discrete Model of CP-PLL

Combining equations for all four cases we obtain a discrete time nonlinear mathematical model of CP-PLL

\[ v_{k+1} = v_k + \frac{I_p}{C} \tau_{k+1}(\tau_k, v_k), \]

\[ k = 0, 1, 2, \ldots \]

with initial conditions \( v_0 \) and \( \tau_0 \). Function \( \tau_{k+1}(\tau_k, v_k) \) is defined by the following equation

\[
\tau_{k+1}(\tau_k, v_k) = \begin{cases} 
\frac{-b + \sqrt{b^2 - 4ac}}{2a}, & \tau_k \geq 0, \quad c \leq 0, \\
\frac{1}{\omega_{\text{vco}}^2 + K_{\text{vco}} v_k} - T_{\text{ref}} + (\tau_k \mod T_{\text{ref}}), & \tau_k \geq 0, \quad c > 0, \\
l_b - T_{\text{ref}}, & \tau_k < 0, \quad l_b \leq T_{\text{ref}}, \\
\frac{-b + \sqrt{b^2 - 4ad}}{2a}, & \tau_k < 0, \quad l_b > T_{\text{ref}}, 
\end{cases}
\]

(31)

Here the initial conditions are the following: \( v_0 \) is the initial phase difference of \( \tau_0 \) which is determined by \( \theta_{\text{vco}}(0) \) and \( i(0) \).

If at some point VCO becomes overloaded (frequency approaches zero) one should stop simulation or use another set of equations, based on ideas similar to (34) and (35) in [34], see section VIII.

A. Locked states, hold-in range, and pull-in range

After the synchronization is achieved, i.e. transient process is over, the loop is said to be in a locked state. For practical purposes, only asymptotically stable locked states, in which the loop returns after small perturbations of its state, are of interest. CP-PLL is in a locked state if trailing edges of the VCO signal happens near the trailing edges of the reference signal:

\[
\left| \frac{\tau_k}{T_{\text{ref}}} \right| \leq \tau_{\text{lock}}, \quad k = 0, 1, 2, \ldots
\]

(33)

where \( \tau_{\text{lock}} \) is sufficiently small, and the loop returns in this state after small perturbations of \( (v_k, \tau_k) \).

In a locked state the output of PFD \( I(t) \) can be non-zero only on short time intervals (shorter than \( \tau_{\text{lock}} \)). An allowed residual phase difference \( \tau_{\text{lock}} \) should be in agreement with engineering requirements for a particular application. The ideal case is \( \tau_{\text{lock}} = 0 \).

There is a hypothesis, which has not yet been proven rigorously, that the hold-in and pull-in ranges[5] coincide and both are infinite if the model has a locked state (see, e.g. [2], [7], [18]).

B. Pull-in time

One of the important characteristics of CP-PLL is how fast it acquires a locked state during the pull-in process. Suppose that the CP-PLL is in a locked state with input frequency \( \omega_{\text{ref}1} \). Then the reference frequency changes to new frequency \( \omega_{\text{ref}2} \) from fixed range \([\omega_{\text{ref}1}^\text{min}, \omega_{\text{ref}1}^\text{max}]\). Since CP-PLL lost it’s locked state, the feedback loop of CP-PLL tunes the VCO to the new input frequency to acquire new locked state. Transient process takes time \( T_{\text{ref}1} \rightarrow \text{ref}2 \), and maximum of such possible time intervals is called a pull-in time \( T_{\text{pull-in}} \). The pull-in time can be measured in seconds or the number of cycles of the input frequency.

Suppose that one needs to design CP-PLL for frequency range from 10MHz to 25MHz with pull-in time less than 10000 cycles. System [54] allows to estimate accurately maximum pull-in time for considered range. An estimate of pull-in time is presented in Fig. 11. As we can see, to decreases the pull-in time we can either increase \( K_{\text{vco}} I_p \) or decrease filter capacitance \( C \). If \( K_{\text{vco}} I_p \) is less than \( 2 \cdot 10^{10} \), then pull-in time is greater than 10000 cycles. Similar analysis can be done for any frequency ranges.

V. Comparison with Original Model by M. van Paemel

In this section using original notation [34] we demonstrate why existing model has problems and how proposed model fixes them.

A. Example 1

Consider the following set of parameters and initial state:

\[ R_2 = 0.2; C = 0.01; K_v = 20; I_p = 0.1; T = 0.125; \]

\[ \tau(0) = 0.0125; v(0) = 1. \]

(34)

5 The largest symmetric interval (continuous range, without holes) of frequencies around free-running frequency of the VCO is called a hold-in range if the loop has a locked state. If, in addition, the loop acquires a locked state for any initial state of the loop and any reference frequency from the interval then the interval is called a pull-in range [4], [20], [21], [23].

6 To estimate pull-in time we loop throughput values \( \alpha = 0.2, \ldots, 10 \) and \( \beta = 100, 10^{10}, 2 \cdot 10^{10}, \ldots, 2.5 \cdot 10^{7} \) (see Appendix) and estimate corresponding pull-in time in the following way: loop through \( \omega_{\text{vco}} \) and \( \omega_{\text{ref}} \) in \( 10^7, 10^8 + 2 \cdot 10^7, 10^7 + 4 \cdot 10^6, \ldots, 2.5 \cdot 10^7 \). \( \tau_{\alpha\beta} \) in \(-0.99, -0.97, \ldots, 0.99 \) and estimate maximum pull-in time if it happens in less than 10000 cycles.
Calculation of normalized parameters (equations (27)-(28) and (44)-(45) in [34]):
\[ K_N = I_p R_2 K_v T = 0.05, \]
\[ \tau_{2N} = \frac{R_2 C}{T} = 0.016, \]
\[ F_N = \frac{1}{2\pi} \frac{\sqrt{K_N}}{\tau_{2N}} \approx 0.2813, \]
\[ \zeta = \frac{\sqrt{K_N \tau_{2N}}}{2} \approx 0.0141, \]
shows that parameters [34] correspond to allowed area in Fig. 14 (equations (46)-(47), Fig 18 and Fig. 22 in [34]):
\[ F_N < \frac{\sqrt{1+\zeta^2} - \zeta}{\pi} \approx 0.3138, \]
\[ F_N < \frac{1}{4\pi\zeta} \approx 5.6438. \]

Now we use the flowchart in Fig. 13 (Fig. 10 in [34]) to compute \( \tau(1) \) and \( v(1) \): since \( \tau(0) > 0 \) and \( \tau(0) < T \), we proceed to case 1) and corresponding relation for \( \tau(k+1) \) (equation (7) in [34]):
\[
\tau(k+1) = \frac{-I_p R_2 - v(k)}{I_p} + \sqrt{\left(\frac{I_p R_2 + v(k)}{I_p}\right)^2 - \frac{2I_p}{C} \left(\frac{v(k)(T - \tau(k)) - \frac{1}{K_v}}{I_p}\right)},
\]
However, the expression under the square root in (37) is negative:
\[
(I_p R_2 + v(0))^2 - \frac{2I_p}{C} (v(0)(T - \tau(0)) - \frac{1}{K_v}) = -0.2096 < 0.
\]

Therefore the algorithm is terminated with error. From [34] for our model we have:
\[
\omega_{\text{free}} = 0, \quad c = (T - (\tau(0) \mod T)) K_v v(0) - 1 = 1.2500,
\]
\[
\tau(1) = \frac{1}{K_v v(0)} - T + (\tau(0) \mod T) = -0.0625, \]
\[
v(1) = v(0) + \frac{I_p}{C} \tau(1) = 0.3750.
\]

B. Example 2

Consider the same parameters as in Example 1, but \( \tau(0) = -0.098 \):
\[
R_2 = 0.2; C = 0.01; K_v = 20; I_p = 0.1; T = 0.125; \]
\[
\tau(0) = -0.098; \quad v(0) = 1.
\]
In this case (35) and Fig. 14 are the same as in Example 1, i.e. we are in the “allowed area”. Now we compute \( \tau(1) \) and \( v(1) \) following the flowchart in Fig. 13 (since \( \tau(0) < 0 \) we proceed to case 2) and corresponding equation of \( \tau(k+1) \) (equation (9) in [34]):
\[
\tau(1) = \frac{1}{K_v} - I_p R_2 \tau(0) - \frac{I_p \tau(0)^2}{K_v} - T + \tau(0) = -0.21906, \]
\[
-0.2191 < -T = -0.125.
\]
This fact indicates cycle-slipping (out of lock). According to the flowchart in Fig. 13 (see Fig. 10 in [34]), we should proceed to case 6) and recalculate \( \tau(1) \). First step of case 6) is to calculate \( t_1, t_2, t_3, \ldots \) (equations (16) and (17) in [34]):
\[
t_n = \frac{v_{n-1} - I_p R_2 - \sqrt{(v_{n-1} - I_p R_2)^2 - 2\frac{I_p}{C} \frac{1}{K_v}}}{\frac{I_p}{C}},
\]
\[
v_n = v_{n-1} - \frac{I_p}{C} t_n,
\]
\[
v_0 = v(k-1).
\]
Since \( k = 0, \) then
\[
v_0 = I_p R_2 - \sqrt{(v_0 - I_p R_2)^2 - 2\frac{I_p}{C} \frac{1}{K_v}},
\]
\[
v_1 = v_0 - \frac{I_p}{C} t_1,
\]
\[
v_0 = v(-1).
\]
However, \( v(-1) \) doesn’t make sense and algorithm terminates with error. Even if we suppose that it is a typo and \( v_0 = v(0) \), then relation under the square root become negative:
\[
(v_0 - I_p R_2)^2 - 2\frac{I_p}{C K_v} = -0.0396 < 0.
\]
In both cases the algorithm is terminated with error. Note, that modification of case 2) corresponding to VCO overload (equation (35) in [34]) can not be applied here, since \( v(0) > I_p R_2 \) (no overload) and \( v(1) \) is not computed yet because of the error.

Corrected model successfully detects overload without any error:

\[
v(1) + \frac{v_{\text{free}}}{K_{\text{vco}}} - I_p R_2 \approx -0.2106 < 0 \tag{45}
\]

**C. Example 3**

Consider parameters:

\[
\tau(0) = -0.123; \quad v(0) = 0.6, \\
R_2 = 0.2; C = 0.02; K_v = 20; I_p = 0.1; T = 0.125.
\tag{46}
\]

Similar to (35) and (36)

\[
K_N = 0.05, \quad \tau_2 N = 0.032, \\
F_N \approx 0.1989, \quad \zeta = 0.02.
\tag{47}
\]

\[
F_N < \sqrt{1 + \zeta^2} - \zeta \approx 0.3120, \\
F_N < \frac{1}{4 \pi \zeta} \approx 3.9789,
\tag{48}
\]

parameters (46) correspond to allowed area in Fig. 14 (equations (46)-(47), Fig. 18 and Fig. 22 in [34]).

Now we compute \( \tau(1) \) and \( v(1) \) following the flowchart in Fig. 13 since \( \tau(0) < 0 \) one proceeds to case 2) and corresponding equation for computing \( \tau(k+1) \) (equation (9) in [34]):

\[
\tau(1) = \frac{1}{K_v} - I_p R_2 \tau(0) - I_p \left(\frac{\tau(0)^2}{2C} - T + \tau(0) \right) \\
= \frac{1}{K_v} - \frac{\tau(0)^2}{2C} - T + \tau(0) \approx -0.224, \\
-0.224 < -T = -0.125.
\tag{49}
\]

The last inequality indicates cycle-slipping (out of lock). According to the flowchart in Fig. 13 (see Fig. 10 in [34]), one proceeds to case 6) and recalculates \( \tau(1) \). First step of case 6) is to calculate \( t_1, t_2, t_3, \ldots \) using (42) (see equations (16) and (17) in [34]) until \( t_1 + t_2 + \ldots + t_n > |\tau(0)| \). Even if we suppose \( v(-1) = v(0) - I_p C \tau(0) \), we get:

\[
t_1 = 0.0463, \quad v_1 = 1.215; \\
t_2 = 0.0618, \quad v_2 = 0.983; \\
t_1 + t_2 = 0.1081 < |\tau(0)| = 0.123.
\tag{50}
\]

However, \( t_3 \) can not be computed, because the relation under the square root in (42) is negative:

\[
(v_2 - I_p R_2)^2 - 2 \frac{I_p}{C} \cdot \frac{1}{K_v} \approx -0.0726.
\tag{51}
\]

Corrected model gives:

\[
\tau(1) = -0.0569, \quad v(1) = 0.3153.
\tag{52}
\]
VI. MATHEMATICAL SIMPLIFICATION OF THE DISCRETE MODEL OF CP-PLL

Follow the ideas from [1, 7], the number of parameters in (31) can be reduced to just two (α and β)

\[ \tau_k^{αβ} = \frac{\tau_k}{T_{ref}}, \]
\[ \omega_k^{αβ} = T_{ref} \left( \omega_{vco}^{free} + K_{vco} v_k \right) - 1, \]
\[ \alpha = K_{vco} I_p T_{ref} R, \]
\[ \beta = \frac{K_{vco} I_p T_{ref}^2}{2C}. \]

Then

\[ \omega_k^{αβ} = \omega_k^{αβ} + 2\beta \tau_k^{αβ} (\tau_k^{αβ}, \omega_k^{αβ}), \]
\[ k = 0, 1, 2, \ldots \]

with initial conditions \( \tau_k^{αβ}(0) \) and \( \omega_k^{αβ}(0) \). Function \( \tau_k^{αβ}(\tau_k^{αβ}, \omega_k^{αβ}) \) is defined by the following equation

\[ \tau_k^{αβ}(\tau_k^{αβ}, \omega_k^{αβ}) = \begin{cases} \frac{-b_n + \sqrt{b_n^2 - 4bc}}{2c}, & \tau_k^{αβ} \geq 0, \quad c \leq 0, \\ \frac{1}{\omega_k^{αβ} + 1} - 1 + (\tau_k^{αβ} \mod 1), & \tau_k^{αβ} < 0, \quad c > 0, \\ l_{bn} - 1, & \tau_k^{αβ} < 0, \quad l_{bn} \leq 1, \\ \frac{-b_n + \sqrt{b_n^2 - 4bc}}{2c}, & \tau_k^{αβ} < 0, \quad l_{bn} > 1, \end{cases} \]

\[ b_n = b_n(\omega_k^{αβ}) = \omega_k^{αβ} + \alpha + 1, \]
\[ c = c(\tau_k^{αβ}, \omega_k^{αβ}) = (1 - (\tau_k^{αβ} \mod 1))(\omega_k^{αβ} + 1) - 1, \]
\[ l_{bn} = l_{bn}(\tau_k^{αβ}, \omega_k^{αβ}) = \frac{1 - (S_l \mod 1)}{\omega_k^{αβ} + 1}, \]
\[ S_l = S_l(\tau_k^{αβ}, \omega_k^{αβ}) = \omega_k^{αβ} - \alpha + 1, \]
\[ d = d(\tau_k^{αβ}, \omega_k^{αβ}) = (S_l \mod 1) + \omega_k^{αβ}. \]

This change of variables allows to reduce parameter space and simplify design of PFD PLLs.

Moreover, checking requirement [33] for all k is impossible in practice during simulation. Instead one can check that values of \( \omega_{k+1}^{αβ} \) and \( \tau_{k+1}^{αβ} \) are small, which indicate that frequencies of VCO and input signal are close and trailing edges happen almost at the same time. It means that the loop is in a locked state.

A. Locked states and periodic solutions

By (53), there is only one stationary point

\[ \omega_k^{αβ} = \omega_{k+1}^{αβ} = 0, \]
\[ \tau_k^{αβ} = \tau_{k+1}^{αβ} = 0. \]

which is a locked state if it is locally stable. Period-2 points

\[ \omega_{k+2}^{αβ} = \omega_k^{αβ}, \]
\[ \tau_{k+2}^{αβ} = \tau_k^{αβ}, \]

also can be found by [53]:

\[ \omega_k^{αβ} + 2 = \omega_k^{αβ} + 2\beta \tau_{k+2}^{αβ}, \]
\[ = \omega_k^{αβ} + 2 + 2\beta \tau_{k+2}^{αβ} + 2\beta \tau_{k+1}^{αβ}, \]
\[ \tau_{k+2}^{αβ} = -\tau_{k+1}^{αβ}. \]

From [58] period-2 points correspond to the alternating between Case 2 and Case 4. In general, period-P points satisfy equation

\[ \sum_{i=k+1}^{k+p} \tau_i = 0. \]

VII. COMPARISON OF SIMULINK VS. V. PAEMEL’S MODEL VS CORRECTED MODEL

Correctness of proposed model was verified by extensive simulation in Matlab Simulink. Circuit level model in Matlab Simulink was compared with original model by V. Paemel and proposed model. Full code also can be found in Github repository [https://github.com/mir/PFD-simulation]. Here we consider only few examples. Based on simulation for this set of parameters all three models produce almost the same results.

A. Example 5

\[ \tau(0) = 0; \quad v(0) = 10; \]
\[ R_2 = 1000; C = 10^{-6}; K_v = 500; I_p = 10^{-3}; T = 10^{-3}. \]
\[ \tau_{2N} = 1; K_N = 0.5; F_N = 0.1125; \zeta = 0.3536. \]
**VIII. Algorithm for VCO Overload**

At step $k$ VCO overload can be detected by checking the following conditions:

\[
\begin{align*}
\tau_k &> 0 \text{ and } v_k + \frac{\omega_{\text{free}}}{K_{\text{vco}}} - \frac{I_p}{C} \tau_k < 0, \\
\tau_k &< 0 \text{ and } v_k + \frac{\omega_{\text{free}}}{K_{\text{vco}}} - I_p R < 0.
\end{align*}
\] (62)

Following subsections describe how to deal with VCO overload.

- **Case O1.** $\tau_k < 0$, $K_{\text{vco}} v_k + \omega_{\text{free}} > 0$, $\tau_{k+1} < 0$
- **Case O2.** $\tau_k < 0$, $K_{\text{vco}} v_k + \omega_{\text{free}} > 0$, $\tau_{k+1} \geq 0$
- **Case O3.** $\tau_k < 0$, $K_{\text{vco}} v_k + \omega_{\text{free}} \leq 0$, $v_k + \frac{\omega_{\text{free}}}{K_{\text{vco}}} + I_p R < 0$
- **Case O4.** $\tau_k < 0$, $K_{\text{vco}} v_k + \omega_{\text{free}} \leq 0$, $v_k + \frac{\omega_{\text{free}}}{K_{\text{vco}}} + I_p R \geq 0$
- **Case O5.** $\tau_k \geq 0$, $K_{\text{vco}} v_k + \omega_{\text{free}} \leq 0$, $\tau_{k+1} > 0$
- **Case O5*. Impossible. $\tau_k \geq 0$, $K_{\text{vco}} v_k + \omega_{\text{free}} \leq 0$, $\tau_{k+1} < 0$. 

\[
\tau(0) = 0; v(0) = 100, \quad R_2 = 1000; C = 4 \cdot 10^{-6}; K_v = 500; I_p = 10^{-3}; T = 10^{-3}; \tau_{2N} = 4; K_N = 0.5; F_N = 0.0563; \zeta = 0.7017. 
\] (61)
• Case O6. \( \tau_k \geq 0, K_{vco} v_k + \omega_{vco}^{\text{free}} > 0, \tau_{k+1} > 0, \)
• Case O7. \( \tau_k \geq 0, K_{vco} v_k + \omega_{vco}^{\text{free}} > 0, \tau_{k+1} \leq 0, \)

Case O1 \( [34] \). \( \tau_k < 0, K_{vco} v_k + \omega_{vco}^{\text{free}} > 0, \tau_{k+1} < 0 \) (equivalently \( l_b < T_{\text{ref}} \))

Note, that values \( \tau_k \) and \( v_k \) can be correctly computed by \( [31] \).

In order to compute \( \tau_{k+1} \) taking into account VCO overload, one can compute phase of VCO before it’s frequency hit zero. The VCO phase at that moment corresponds to the area \( S_{la} \) of the green triangle (see Fig. 17). In order to find the triangle area one can find \( l_x \) — time interval corresponding to zero VCO frequency:

\[
l_x = \min \left\{ -C, \frac{1}{I_p} (v_k + \omega_{vco}^{\text{free}} - I_p R), -\tau_k \right\}. \quad (63)
\]

Since the triangle in question \( S_{la} \) is part of the larger triangle \( S \), we get

\[
S = K_{vco} (\tau_k + l_x)^2 \frac{I_p}{2C},
\]

\[
S_{la} = S \mod 1. \quad (64)
\]

Condition \( \tau_{k+1} < 0 \) means that VCO edge triggered change of PFD state from 0 to \(-I_p\). Since phase change between two consecutive falling edges of VCO is 1, we get

\[
S_{la} + l_b (K_{vco} v_k + \omega_{vco}^{\text{free}}) = 1,
\]

\[
l_b = \frac{1 - S_{la}}{K_{vco} v_k + \omega_{vco}^{\text{free}}}. \quad (65)
\]

where \( l_b \) is time during which PFD was in zero state. If \( l_b < T_{\text{ref}} \), then

\[
\tau_{k+1} = T_{\text{ref}} - l_b. \quad (66)
\]

If \( l_b \geq T_{\text{ref}} \) then we should proceed to the next case (O2).

Case O2. \( \tau_k < 0, K_{vco} v_k + \omega_{vco}^{\text{free}} > 0, \tau_{k+1} \geq 0 \) (equivalently \( l_b \geq T_{\text{ref}} \))

Value of \( S_{la} \) is computed by \( [64] \). Now we can compute \( \tau_{k+1} \) (see Fig. 18) similarly to case 4, \( [29] \)

\[
\tau_{k+1} = -b + \sqrt{b^2 - 4ad},
\]

\[
a = \frac{K_{vco} I_p}{2C}, \quad (67)
\]

\[
b = b(v_k) = \omega_{vco} + K_{vco} v_k + K_{vco} I_p R, \quad (64)
\]

\[
d = b(\tau_k, v_k) = S_{la} + S_{T_{\text{ref}}} - 1,
\]

\[
S_{T_{\text{ref}}} = S_{T_{\text{ref}}} (v_k) = T_{\text{ref}} (K_{vco} v_k + \omega_{vco}^{\text{free}}).
\]

Case O3. \( \tau_k < 0, K_{vco} v_k + \omega_{vco}^{\text{free}} \leq 0, v_k + \omega_{vco}^{\text{free}} K_{vco} + I_p R < 0 \)

Fig. 17: VCO overload case O1. \( \tau_k < 0, K_{vco} v_k + \omega_{vco}^{\text{free}} > 0, \tau_{k+1} < 0 \) (equivalently \( l_b < T_{\text{ref}} \))

Fig. 18: VCO overload case O2. \( \tau_k < 0, K_{vco} v_k + \omega_{vco}^{\text{free}} > 0, \tau_{k+1} \geq 0 \) (equivalently \( l_b \geq T_{\text{ref}} \))

Fig. 19: Overload limit, VCO cycle slipping; \( \tau_k < 0, \tau_{k+1} < 0 \)
Consider timing diagram on Fig. [19]. One can split interval \( \tau_{k+1} \) into subintervals \( l_{b0} \), \( l_{b+} \) such that PFD output is zero on \( l_{b0} \) and positive on \( l_{b+} \):

\[
\tau_{k+1} = l_{b0} + l_{b+},
\]

\[
\omega_{\text{free}} + K_{\text{vco}}(I_p R + v_k + \frac{I_p}{C} l_{b0}) = 0,
\]

\[l_{b0} = \frac{I_p}{C} \left( -\frac{\omega_{\text{free}}}{K_{\text{vco}}} - I_p R - v_k \right). \tag{68}\]

Since phase difference between two consecutive falling edges of VCO is 1, we get

\[
S_{l_a} + S_{l_{b+}} = 1,
\]

\[
S_{l_{b+}} = \frac{K_{\text{vco}} I_p \tau_{k+1}^2}{2C l_{b+}},
\]

\[l_{b+} = \sqrt{(1 - S_{l_a}) \frac{2C}{K_{\text{vco}} I_p}}, \tag{69}\]

where \( S_{l_a} \) and \( S_{l_b} \) are areas of corresponding triangles (phase difference of VCO for corresponding time intervals).

**Case O4.** \( \tau_k < 0, K_{\text{vco}} v_k + \omega_{\text{free}} \leq 0, v_k + \frac{\omega_{\text{free}}}{K_{\text{vco}}} + I_p R \geq 0 \)

Consider timing diagram on Fig. [20]. Since phase difference between two consecutive falling edges of VCO is 1, we get

\[
S_{l_a} + S_{\tau_{k+1}} = 1,
\]

where \( S_{l_a} \) can be computed using [64], and \( S_{\tau_{k+1}} \) is phase of VCO corresponding to time interval \( \tau_{k+1} \). Then

\[
S_{\tau_{k+1}} = \frac{K_{\text{vco}} I_p \tau_{k+1}^2}{2C} + \tau_{k+1} (\omega_{\text{free}} + K_{\text{vco}} (v_k + I_p R)). \tag{70}\]

Finally,

\[
\tau_{k+1} = \frac{-b + \sqrt{b^2 - 4ad_o}}{2a},
\]

\[a = \frac{K_{\text{vco}} I_p}{2C},
\]

\[b = \omega_{\text{free}} + K_{\text{vco}} v_k + K_{\text{vco}} I_p R,
\]

\[d_o = -S_{\tau_{k+1}}. \tag{71}\]

**Case O5.** \( \tau_k \geq 0, K_{\text{vco}} v_k + \omega_{\text{free}} \leq 0, \tau_{k+1} > 0 \)

Consider timing diagram on Fig. [21]. Since phase difference between two consecutive falling edges of VCO is 1, we get

\[
\tau_{k+1} (\omega_{\text{free}} + K_{\text{vco}} (v_k + I_p R)) + \tau_{k+1} \frac{K_{\text{vco}} I_p}{2C} = 1 \tag{72}
\]

Therefore

\[
\tau_{k+1} = \frac{-b + \sqrt{b^2 - 4ad_o}}{2a},
\]

\[a = \frac{K_{\text{vco}} I_p}{2C},
\]

\[b = \omega_{\text{free}} + K_{\text{vco}} v_k + K_{\text{vco}} I_p R. \tag{73}\]

**Case O6.** \( \tau_k \geq 0, K_{\text{vco}} v_k + \omega_{\text{free}} > 0, \tau_{k+1} > 0 \)

Consider timing diagram on Fig. [22]. Here \( \tau_{k+1} \) can be computed using case 1 (without overload, see Fig. [28a]).

**Case O7.** \( \tau_k \geq 0, K_{\text{vco}} v_k + \omega_{\text{free}} > 0, \tau_{k+1} \leq 0 \)

Consider timing diagram on Fig. [23]. Here \( \tau_{k+1} \) can be computed using [21] from case 2 (without overload).
In this paper a non-linear mathematical model of CP-PLL is rigorously derived. The obtained model obviates the shortcomings in previously known mathematical models of CP-PLL. The VCO overload case initially noted in [34] is extended to take into account new cases. Analysis of local stability with respect to coordinate \( \tau_k \) (partial stability, see condition (33)) gives us the estimation of the hold-in range. There is a hypothesis, which has not yet been proven rigorously, that the hold-in and pull-in ranges are coincide. For some parameters we estimate the pull-in time. There were many attempts to generalize equations derived in [34] for higher-order loops (see, e.g. [5], [14], [15], [27], [30], [33]), but the resulting transcendental equations can not be solved analytically without using approximations.

**CONCLUSION**

The work is supported by the Russian Science Foundation project 19-41-02002 and Grant for Leading Scientific Schools of Russia (2018-2019).

Authors would like to thank Mark Van Paemel for valuable comments. In private communication with Mark Van Paemel it turned out that some of the considered shortcoming were avoided in FORTRAN code that he used for the simulation of the circuit. Also Mark Van Paemel emphasized importance of VCO overload and motivated us to develop corresponding algorithm.

**ACKNOWLEDGEMENTS**

The work is supported by the Russian Science Foundation project 19-41-02002 and Grant for Leading Scientific Schools of Russia (2018-2019).

Authors would like to thank Mark Van Paemel for valuable comments. In private communication with Mark Van Paemel it turned out that some of the considered shortcoming were avoided in FORTRAN code that he used for the simulation of the circuit. Also Mark Van Paemel emphasized importance of VCO overload and motivated us to develop corresponding algorithm.

**REFERENCES**

[1] P. Acco. *Study of the loop ‘a phase lock: Hybrid aspects taken into account*. PhD thesis, Toulouse, INSA, 2003.

[2] P. Acco, J. Daafouz, D. Fournier-Prunaret, and A.-K. Taha. Approche hybride de la stabilité locale de la boucle à verrouillage de phase par impulsions de charge. *Revue e-STA*, 1(4), 2004.

[3] R.E. Best. *Phase locked loops: design, simulation, and applications*. McGraw-Hill Professional, 2007.

[4] R.E. Best, N.V. Kuznetsov, G.A. Leonov, M.V. Yuldashev, and R.V. Yuldashev. Tutorial on dynamic analysis of the Costas loop. *Annual Reviews in Control*, 42:27–49, 2016.

[5] Chuang Bi, Paul F. Curran, and Orla Feely. Linearized discrete-time model of higher order charge-pump pll. In *Circuit Theory and Design (ECCTD)*, 2011 20th European Conference on, pages 457–460. IEEE, 2011.

[6] F. Bizzarri, A. Brambilla, and G.S. Gajani. Periodic small signal analysis of a wide class of type-ii phase locked loops through an exhaustive variational model. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 59(10):2221–2231, 2012.

[7] P.F. Curran, Ch. Bi, and O. Feely. Dynamics of charge-pump phase-locked loops. *International Journal of Circuit Theory and Applications*, 41(11):1109–1135, 2013.

[8] B. Daniels. *Analysis and design of high order digital phase locked loops*. PhD thesis, National University of Ireland Maynooth, 2008.

[9] Orla Feely, Paul F. Curran, and Chuang Bi. Dynamics of charge-pump phase-locked loops. *International Journal of Circuit Theory and Applications*, 2012.

[10] F. Gardner. Charge-pump phase-lock loops. *IEEE Transactions on Communications*, 28(11):1849–1858, 1980.

[11] F.M. Gardner. *Phase lock techniques*. John Wiley & Sons, New York, 1966.

[12] F.M. Gardner. *Phase lock techniques*. John Wiley & Sons, 2005.

[13] M. Guermandi, E. Franchi, and A. Gnudi. On the simulation of fast settling charge pump PLLs up to fourth order. *International Journal of Circuit Theory and Applications*, 39(12):1257–1273, 2011.

[14] C. Hangmann, C. Hedayat, and U. Hilleringmann. Stability analysis of a charge pump phase-locked loop using autonomous difference equations. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 61(9):2569–2577, 2014.

[15] Pavan Kumar Hannumolu, Merrick Brownlee, Kartikeya Maharam, and Un-Ku Moon. Analysis of charge-pump phase-locked loops. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51(9):1665–1764, 2004.

[16] C.D. Hedayat, A. Hachem, Y. Leduc, and G. Benbassat. Modeling and characterization of the 3rd order charge-pump PLL: a fully event-driven approach. *Analog Integrated Circuits and Signal Processing*, 19(1):25–45, 1999.

[17] J.P. Hein and J.W. Scott. Z-domain model for discrete-time PLL’s. *IEEE Transactions on Circuits and Systems*, 35(11):1393–1400, 1988.

[18] A. Homayoun and B. Razavi. On the stability of charge-pump phase-locked loops. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(6):741–750, 2016.

[19] T. Johnson and J. Holmberg. Nonlinear state-space model of charge-pump based frequency synthesizers. In *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 4469–4472. IEEE, 2005.

[20] N.V. Kuznetsov, G.A. Leonov, M.V. Yuldashev, and R.V. Yuldashev. Rigorous mathematical definitions of the hold-in and pull-in ranges for phase-locked loops. *HFAC-PapersOnLine*, 48(11):710–713, 2015.

[21] N.V. Kuznetsov, G.A. Leonov, M.V. Yuldashev, and R.V. Yuldashev. Solution of the Gardner problem on the lock-in range of phase-locked loop. *ArXiv e-prints*, 2017. 1705.05013.

[22] N.V. Kuznetsov, M.V. Yuldashev, R.V. Yuldashev, M.V. Blagov, E.V. Kudryashova, O.A. Kuznetsova, and T.N. Mokaev. Comment on “Analysis of a charge-pump PLL: A new model” by M. van Paemel. arXiv preprint arXiv:1810.02698, 2018.

[23] G.A. Leonov, N.V. Kuznetsov, M.V. Yuldashev, and R.V. Yuldashev. Hold-in, pull-in, and lock-in ranges of PLL circuits: rigorous mathematical definitions and limitations of classical theory. *IEEE Transactions on Circuits and Systems–I: Regular Papers*, 62(10):2454–2464, 2015.

[24] W.C. Lindsey and R.C. Tausworthe. A Bibliography of the Theory and Application of the Phase-lock Principle. *JPL technical report*. Jet Propulsion Laboratory, California Institute of Technology, 1973.

[25] J. Lu, B. Grung, S. Anderson, and S. Rokhsaz. Discrete z-domain analysis of high order phase locked loops. In *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, volume 1, pages 260–263. IEEE, 2001.

[26] G Mészáros, Cs Fúzy, and J Ladvánszky. Does amplitude or phase noise arise first? *Electronics Letters*, 48(12):1, 2012.

[27] Siniša Milicic and Leonard MacEarchern. Time evolution of the voltage-controlled signal in charge pump pll applications. In *Microelectronics, 2008. ICM 2008. International Conference on*, pages 413–416. IEEE, 2008.

[28] B. Razavi. *Monolithic phase-locking loops and clock recovery circuits: theory and design*. John Wiley & Sons, 1996.

[29] Werner Rosenkranz. Phase-locked loops with limiter phase detectors in the presence of noise. *IEEE Transactions on communications*, COM-30(10):805–809, 1982.
[30] Sergio Sancho, Almudena Suárez, and Jeffrey Chuan. General envelope-transient formulation of phase-locked loops using three time scales. *IEEE Transactions on Microwave Theory and Techniques*, 52(4):1310–1320, 2004.

[31] V.V. Shakhgildyany and A.A. Lyakhovkin. *Fazovaya avtopodstroika chastoty (in Russian)*. Svyaz', Moscow, 1966.

[32] V.V. Shakhgildyany and A.A. Lyakhovkin. *Sistemy fazovoi avtopodstroiki chastoty (in Russian)*. Svyaz', Moscow, 1972.

[33] B.I. Shakhtarin, A.A. Timofeev, and V.V. Sizykh. Mathematical model of the phase-locked loop with a current detector. *Journal of Communications Technology and Electronics*, 59(10):1061–1068, 2014.

[34] M. van Paemel. Analysis of a charge-pump PLL: A new model. *IEEE Transactions on Communications*, 42(7):2490–2498, 1994.

[35] A. Viterbi. *Principles of coherent communications*. McGraw-Hill, New York, 1966.

[36] Z. Wang. An analysis of charge-pump phase-locked loops. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 52(10):2128–2138, 2005.
Appendix: MATLAB lock-in code

```matlab
1 a_range = 0.001:2:10;
2 b_range = 100:10^10:2*10^11;
3 ref_range = 10^7:(2*10^6) :(2.5*10^7) ;
4 5 diagram = zeros(length(a_range),length(b_range));
6 x = 1;
7 y = 1;
8 for alpha = a_range
9   for beta = b_range
10      lock_in_t = find_lock_in(alpha,beta,ref_range);
11      diagram(x,y) = lock_in_t;
12      y = y + 1;
13      beta ;
14   end
15   x = x + 1;
16   y = 1;
17 end
18 19 figure
20 surface(b_range,a_range,diagram);
21 zlabel('pull-in time ');
22 xlabel('K_{vco}I_{p}/2C ');
23 ylabel('K_{vco}I_{p}R ');
```

Function `find_lock_in`:

```matlab
1 function [lockin_t] = find_lock_in(alpha,beta,...
2   ref_range)
3 4 lockin_t = 0;
5 for omega_ref = ref_range
6   if omega_ref == omega_ref
7     continue
8   end
9   omega_ab0 = omega_ref / omega_ref - 1;
10   tau_ab0 = -0.99:0.02:0.99
11   for tau_ab0 = -0.99:0.02:0.99
12      cur_lockin_t = lockin_time(omega_ab0,tau_ab0,...
13          alpha,beta,omega_ref,ref_range);
14   end
15   if (cur_lockin_t > lockin_t)
16      lockin_t = cur_lockin_t;
17   end
18 end
19 20 function [lockin_time] = lockin_time(omega_ab,...
21   tau_ab,omega_ref,...
22   alpha,beta,omega_ref,...
23   max_lockin_t)
24 25 % maximum phase shift in locked state
26   eps = 0.1;
27   tau_ab = tau_ab0;
28   omega_ab = omega_ab0;
29   lockin_time = 0;
30 while (lockin_time < max_lockin_t)
31      if (abs(omega_ab < eps))...
32         & (abs(tau_ab < eps))
33         return;
34     end
35     omega_ab1 = omega_ab + 2*beta*tau_ab1;
36     tau_ab1 = tau_ab + 2*beta*tau_ab0;
37     omega_ab_zero = omega_ab1;
38     tau_ab_zero = tau_ab1;
39     lockin_time = lockin_time + abs(tau_ab) + abs(tau_ab_zero);
40 end
41 42 function [omega_ab1,tau_ab1,tau_ab_zero] = ...
43   righthand_ab(omega_ab,tau_ab,omega_ab_zero,...
44       alpha,beta,omega_ref);
45   tau_ab = tau_ab1;
46   omega_ab = omega_ab1;
47   lockin_time = lockin_time + abs(tau_ab) + abs(tau_ab_zero);
48 end
```

Appendix: Simulink model

Fig. 24: Simulink model of CP-PLL

Fig. 25: Simulink model of PFD

Fig. 26: Simulink model of VCO
```
% Block Parameters: Loop filter

State Space
State-space model: C = D * A + B

Parameters
A:
B:
C:
D:
Initial conditions:
initial_filter_state = v_1 - I_p * tau_1 /C;
initial_ref_phase = 1 + tau_1 / T_ref ;
initial_vco_phase = 0;

Absolute tolerance:
auto

State Name (e.g., position)

OK  Cancel  Help  Apply

Fig. 27: Simulink model of Loop filter

1
% parameters
2 omega_free = 0;
3 T_ref = 10^-5;  
4 K = 1000;
5 C = 10^-6;
6 K_vco = 500;
7 I_p = 10^-3;
8
9 % calculated values
10 tau_2N = R*C/ T_ref 
11 % recalculated values
12 tau_k_zero = T_ref ;
13 tau_k1 = -l_k1 ;
14 if lb <= T_ref
15 lb = S_lb /( K_vco * v_k + omega_free );
16 S_lb = 1- S_la ;
17 S_la = rem(S_lk ,1) ;
18 S_lk = ( K_vco *v_k - I_p *R* K_vco + omega_free )*lk +( K_vco * I_p *lk ˆ2) /(2* C);
19 lk = -tau_k 
20 else
21 tau_k_zero = 1/( omega_free + K_vco * v_k );
22 tau_k1 = 1/( omega_free + K_vco * v_k ) - T_ref + rem ( tau_k , T_ref );
23 % tau (k +1) < 0, case 2)
24 else
25 tau_k_zero = T_ref - rem ( tau_k , T_ref );
26 tau_k1 = (-b + sqrt (bˆ2 - 4*a*c)) /(2* a);
27 end
28
29
30
31 % check that omega_free is zero
32 if (omega_free > 0)
33 errordlg('Please set omega_free to zero');
34 end
35
36 % initialize PFD output with initial data
37 pfd_output = zeros((max_step-1)*4,2);
38 pfd_output(index,:) = [0 0];
39 if (tau_k1 >= 0)
40 pfd_output(1,:) = [0 I_p ];
41 pfd_output(3,:) = [tau_k1 -I_p ];
42 pfd_output(4,:) = [tau_k1 +I_p ];
43 if( tau_k1 == 0)
44 pfd_output(index,:) = [tau_k1 0];
45 end
46 else
47 pfd_output(2,:) = [0-I_p ];
48 pfd_output(4,:) = [-tau_k1 0];
49 pfd_output(5,:) = [-tau_k1 +I_p ];
50 t_k = t_k1 ;
51 end
52 for step = 2:(max_step - 1)
53 check for VCO overload
54 if (tau_k < 0)
55 if (K_vco + omega_free*K_vco - 1/pC < 0)
56 errordlg('VCO overload detected at k = %d, step-%d');
57 break;
58 end
59 end
60 [tau_k, v_k, tau_k_zero] = righthand(tau_k, v_k,...
61 K_vco, T_ref, L_p, C, omega_free);
62 t_k = t_k + abs(tau_k);
63 index = index + 1;
64
65 ```
if (tau_k1 < -T_ref)
44    break;
45  end
46
47  if (step <= 2)
48
49    errordlg('ERROR in Paemel algo case 6) , no v( -1) ');
50  end
51
52  t_sum = 0;
53
54  while (t_sum < abs(tau_k))
55    t_n = ( v_n_1 - I_p *R - sqrt(( v_n_1 - I_p *R)^2 - 2* I_p /C/ K_vco) ...)
56    / (I_p /C);
57  
58    t_sum = t_sum + t_n;
59    v_n_prev = v_n_1;
60
61    v_n_1 = v_n_1 - I_p /C* t_n;
62
63  end
64
65  t_sum = t_sum - t_n;
66  
67  t_a = -tau_k - t_sum;
68  
69  t_b = (1/ K_vco - t_a *( v_n_prev - I_p *R) + I_p /C* t_a ˆ2/2) / v_k;
70  
71  tau_k1 = t_b - T_ref;
72
73  v_k1 = v_k + tau_k1 * I_p /C;
74  
75  tau_k_zero = t_b;
76
77  end
78
79  end
80
81  end
82
83  end
84
85  end
86
87  end
88
89  end
90
91  (tau_k1, v_k1, tau_k_zero) = righthand(tau_k, v_k, ...)
92
93  function [tau_k1, v_k1, tau_k_zero] = paemel_righthand(...
94
95  if (tau_k > 0)
96    % case 1
97    a = I_p /(2* C);
98    b = v_k + I_p *R;
99    c = (T_ref - tau_k)*v_k -1/ K_vco;
100   tau_k1 = (-b + sqrt(bˆ2 - 4*a*c)) /(2* a);
101   
102   tau_k_zero = T_ref - tau_k;
103   
104   else
105     % case 2 Paemel
106     tau_k1 = (...
107       1/ K_vco - I_p *R* tau_k - I_p * tau_k ˆ2/(2* C) ...)
108       / v_k - T_ref + tau_k;
109       
110       lk = -tau_k;
111       
112       S_lk = ( K_vco *v_k - I_p *R* K_vco)*lk +( K_vco * I_p *lk ˆ2) /(2* C);
113       
114       tau_k_zero = (1 - rem(S_lk ,1) )/( K_vco * v_k);
115       
116       if( tau_k1 > 0)
117         % case 4 Paemel
118         tau_k1 = (...
119           -I_p *R -v_k + sqrt(( I_p *R + v_k )ˆ2 + (2* I_p /C)* v_k * tau_k1 ) ...)
120           /( I_p /C);
121         
122         tau_k_zero = T_ref;
123         
124       end
125
126       v_k1 = v_k + tau_k1 * I_p /C;
127
128       end
129  end
130
131  % truncate trailing zeros
132  last_non_zero = find (paemel_output (: ,1) ,1,'last ');
133  
134  if (last_non_zero > 1)
135    paemel_output = paemel_output (1: last_non_zero ,:);
136  end
137
138  end
139
140  % plot (pfd_output (: ,1) , pfd_output (: ,2) );
141  % ylim([-1.1* I_p 1.1* I_p]);

APPENDIX: ADDITIONAL PICS
Ref | VCO | PFD
\( \omega_{vco}(t) \) | \( \omega_{vco} \) | \( \omega_{vco} \)
\( \tau_k \) | 0 | 0
\( t_k, t_{k+1} \) | 0 | 0
\( t_{middle, k}, t_{middle, k+1} \) | 0 | 0

(a) Case 1: general case, Reference cycle slipping; \( \tau_k \geq 0, \tau_{k+1} \geq 0 \).

(b) Case 1: all VCO and Reference trailing edges happen at different time instances; \( \tau_k > 0, \tau_{k+1} > 0 \).

(c) Case 1: VCO and Reference trailing edges both happen at the same time instance \( t_{k+1}; \tau_k > 0, \tau_{k+1} = 0 \).

(d) Case 1: VCO and Reference trailing edges both happen at the same time instance \( t_{k+1}; \tau_k \mod T_{ref} = 0, \tau_{k+1} > 0 \).

(e) Case 1: VCO and Reference trailing edges both happen at the same time instance \( t_k; \tau_k = 0, \tau_{k+1} > 0 \).

Fig. 28: Subcases of the Case 1. Integral of the VCO frequency \( \omega_{vco} \) over the VCO period \( T_{vco} \) is pink subgraph area (grey in black/white). The integral is equal to 1 according to the PFD switching law and definition of time intervals.
(a) Case 2-1: general case, Reference and VCO cycle slipping; \( \tau_k \geq 0, \tau_{k+1} < 0 \)

(b) Case 2-2: all VCO and Reference trailing edges happen at different time instances; \( \tau_k > 0, \tau_{k+1} < 0 \)

(c) Case 2-3: VCO and Reference trailing edges both happen at the same time instance \( t_{k+1}^{middle} \); \( \tau_k \mod T_{ref} = 0, \tau_{k+1} < 0 \)

(d) Case 2-4: VCO and Reference trailing edges both happen at the same time instance \( t_k \); \( \tau_k = 0, \tau_{k+1} < 0 \)

Fig. 29: Subcases of the Case 2. Integral of the VCO frequency \( \omega_{vco} \) over the VCO period \( T_{vco} \) is pink subgraph area (grey in black/white). The integral is equal to 1 according to the PFD switching law and definition of time intervals.
Fig. 30: Subcases of the Case 3. Integral of the VCO frequency $\omega_{\text{vco}}$ over the VCO period $T_{\text{vco}}$ is pink subgraph area (grey in black/white). The integral is equal to 1 according to the PFD switching law and definition of time intervals.
Fig. 31: Subcases of the Case 4. Integral of the VCO frequency $\omega_{vco}$ over the VCO period $T_{vco}$ is pink subgraph area (grey in black/white). The integral is equal to 1 according to the PFD switching law and definition of time intervals.

(a) Case 4 1: general case, VCO and Reference cycle slipping; $\tau_k < 0$, $\tau_{k+1} > 0$

(b) Case 4 2: Reference cycle slipping; Refer all VCO and Reference trailing edges happen at different time instances; $\tau_k < 0$, $\tau_{k+1} > 0$

(c) Case 4 3: VCO and Reference trailing edges both happen at the same time instance $t_{k+1}^{\text{middle}}$; $\tau_k < 0$, $\tau_{k+1} > 0$