A Design of Bang-Bang PLL in Low Jitter and Wide Pull-in Range

Xihong Chen, Qiang Liu*, Denghua Hu
Air and Missile Defense College, Air Force Engineering University,
No. 1 Changle Road, Xi'an, 710051, China
*Corresponding author, e-mail: dreamlq@163.com

Abstract
As bang-bang PLL (BBPLL) could resume clock data rapidly, its application in clock data recovery has become increasingly abroad. Aiming at the contrary demand of lower jitter and wider pull-in range of BBPLL, the issue puts forth a method to choose the most appropriate gain of digitally controlled oscillator (DCO) to settle. A judged and modified model has been added to the 2nd order traditional BBPLL, which modified the gain of DCO dynamically by step forward method. Meanwhile it proposes pull-in jitter function (PJF) to judge the modified results. Then it takes a gradual comparison means to search the max PJF. It could be concluded from the simulations that the algorithm of the issue could get a compromise DCO gain in view of BBPLL’s jitter and pull-in range.

Keywords: bang-bang PLL, DCO, pull-in range, jitter, pull-in jitter function

1. Introduction
High precision atomic clock has been widely used in satellites’ communication, like navigation, orientation and timing. The clocks’ performances are the base of satellites’ services. While the clocks errors will worsen the services precision, even interrupt satellites normal operations. In order to get ideal satellites services, we should establish proper PLL circuits to calibrate the clock errors [1-3].

Bang-bang PLL (BBPLL) has been taken in high speed clock data recovery area widely for the advantages of easy design and high speed process. The main disadvantage is the system’s jitter brought by phase detection, which could be depressed by reducing the gain of digitally controlled oscillator (DCO), but also leading to a narrower pull-in range. Some scholars have made researches about this problem [4-6].

In 2005, Dalt designed a nonlinear dynamics digital BBPLL and studied the effects of loop delays of the first- and second-order BBPLL. And he summarized the design method of such PLL and gave optimal parameters in low jitter [7]. Aiming at neglecting the effect of the BBPLL dynamics on the effective jitter in the traditional calculation of binary phase detector (BPD), in 2006, Dalt put forth a method to model BBPLL dynamics by Markov chain, which got more accurate precision [8]. While converting the phase error into digital value by Bang-bang Phase Detector (BBPD) will brought serious nonlinearity into the loop, which will limit the traditional linear analysis. In 2008, Dalt analyzed the deficiency of the linearized loop and applied it to jitter transfer and the jitter generation in computation [9]. It was validated by analysis of actual circuits.

Besides, Chan et al designed a new PLL by dynamically scales its gain, which achieved fast lock times and great jitter performance in lock [10]. Chun and Kennedy studied the stationary state probability based on a delayed Markov chain model and a state flow diagram, which improved capture range and lock time [11].

Based on the idea of dynamically scales PLL’s gain brought by Chun [11], aiming at the contrary of jitter and pull-in range for DCO’s gain, in this issue, we use a function to evaluate the compounded characteristics of jitter and pull-in range, and get a compromise result between them. In section 2 and 3, traditional and improved BBPLL models will be presented. In section 4, we will validate our improved model by examples. At last, the conclusion will be arrived in section 5.

Received June 18, 2014; Revised September 24, 2014; Accepted October 19, 2014
2. Traditional BBPLL

A 2nd order BBPLL is composed of a BBPD, a DCO and a frequency divider, which could be illustrated in Figure 1 [9, 12].

In the Digital Loop Filter (DLF) module, we could get:

\[
\begin{align*}
\epsilon_k &= \text{sgn}(tr_k - td_k) \\
\psi_k &= \alpha \psi_{k-1} + \epsilon_k \\
\omega_k &= \beta \omega_{k-D} + \alpha \psi_{k-D}
\end{align*}
\] (1)

Where \( \epsilon_k \), \( tr_k \), \( td_k \) and \( \omega_k \) are denoted for output of BBPD, referred clock, feedback clock and DLF’s output separately. \( \text{sgn}(\cdot) \) is sign function, \( D \), \( \alpha \) and \( \beta \) are the filter latency, integral and proportional gains respectively. Based on (1), we will get the transfer function \( F \) in \( z \) domain as:

\[
F(z) = \frac{\omega(z)}{\epsilon(z)} = \left[ \beta + \frac{\alpha}{1 - \alpha z^{-1}} \right] z^{-i\omega}
\] (2)

Substitute \( z = e^{i\omega T} \) into (2), we will get:

\[
F(\omega) = \left[ \beta + \frac{\alpha}{1 - \alpha e^{-i\omega T}} \right] e^{-i\omega T}
\] (3)

From thumb rules, we could get the pull-in range of the BBPLL as:

\[
\Delta \omega_p = \sqrt{\frac{K_r}{N} \Re \left[ F(\frac{\Delta \omega_p}{2}) \right] F(j0)}
\] (4)

Where \( K_r \) is the gain of DCO, \( N \) is the divider coefficient.

Meanwhile we could get the PLL’s jitter as [13]:

\[
\sigma_{\omega} \approx \sqrt{\frac{\pi}{8} \sigma_{\omega_p}^2 + \frac{1+D}{\beta K_f} N \beta K_r}
\] (5)
3. Improved Design of BBPLL

Aiming at the conflict of jitter and pull-in range, in order to compromise them, we add a judged and modified module in DCO part, which could be interpreted in Figure 2.

![Figure 2. Improved sketch diagram of 2nd order BBPLL with modified DCO's gain.](image)

As the increase of DCO’s gain $K_T$ will better the pull-in range but worsen the jitter performance, we bring out a track of DCO’s output to a judgment and modification module (JMM), and modulate $K_T$ based on the result of judgment. Detailed steps are as follows.

Before we start the process of JMM, we should wait the whole PLL to locked the input signal, which is called big feedback loop. When the big loop becomes locked and stable, the jitter or the pull-in range of the PLL is not ideal at the moment, even though the loop is locked. As a result, it needs the small feedback loop JMM to modify the DCO’s gain to improve the performance of BBPLL.

Step 1: Initialization

Initialize DCO’s original gain $K_T$, and other parameters of BBPLL based on the referred clock, including $\alpha$, $\beta$, $D$ and $N$ etc.

Step 2: Particles training

After the BBPLL has locked, we calculate the pull-in range and jitter in a certain $T_K$. The process is called a training and the $T_K$ a particle. In the Step 3, we will adjust the DCO’s gain $K_T$ gradually. When we get a new particle, i.e. $K_T$, we will get the corresponding pull-in range and jitter.

Step 3: Judgment and Modification

Pull-in range and jitter are the functions of $K_T$. In order to get an ideal pull-in range and jitter, we bring a pull-in jitter function (PJF) to judge the modified results and take step forward method to search the best particle. PJF is defined as:

$$PJF = \frac{\Delta \sigma_p}{\sigma_{i,u}}$$

(6)

When the $i$th particle and PJF are $K_T$ and $PJF_i$, we increase a positive step length to $K_T$, then we will get $K_{T(i+1)}$ and $PJF_{i+1}$. If $PJF_{i+1} > PJF_i$, we will continue increase $K_{T(i+1)}$ positively and repeat the process above. If $PJF_{i+1} < PJF_i$, we will increase a negative step length to $K_T$ and continue make comparisons. If $PJF_{i+1} = PJF_i$, we will make $K_{T(i+1)}$ fixed. During the comparisons, we will get the maximum of $PJF$, and the corresponding best particle $K_T$, which is the compromise gain for jitter and pull-in range. After we get the best particle, we will send it to DCO and modify its gain to the particle we get. As the input of PLL is dynamic, so the process is not stable. When the input changes, the big loop will work to lock the input frequency. After the process, the small loop starts to work, in order to get a compromise pull-in range and jitter performances against the input signal.
4. Examples Calculations and Analyses

The initial parameters are set as follows: $N = 10$, $\sigma_{\nu_0} = 1$ps, $\alpha = 0.5$, $\beta = 16$, $K_{r_0} = 10^{-15}$s, $f_{ref} = 10$MHz and step length $l_{step} = 10^{-14}$s. When $D = 2$, we will get curves of BBPLL’s pull-in range and jitter against $K_T$ in Figure 3.

![Graph](image)

(a) Pull-in range against DCO’s gain $K_T$, (b) Jitter against DCO’s gain $K_T$

Figure 3. BBPLL’s partial parameters curves with DCO’s gain $K_T$

From Figure 3, we could get that the pull-in range is proportional to $K_T$ approximately, while jitter experiences decreasing former and increasing later. When $D = 2$, we will get $P_{JF}$ against $K_T$ in Figure 4. Meanwhile $P_{JF}$ in other latencies, like $D = 0, 1$ are also presented.

![Graph](image)

Figure 4. $P_{JF}$ against DCO’s gain $K_T$ in different latency $D$

From Figure 4, we will get that $P_{JF}$ increases former and decreases later with the increase of $K_T$. We will achieve the best particle when the $P_{JF}$ reaches maximum. When latency $D$ increases, the $P_{JF}$ decreases, indicating that the pull-in range and jitter compound performances deteriorate.
5. Conclusion

Aiming at the contrary demands of DCO’s gain while considering jitter and pull-in range in BBPLL, we have added a judged and modified module in DCO part based on traditional BBPLL. In order to get a compromise DCO’s gain, we put forth the pull-in jitter function. Besides, We have made comparisons of BBPLL’s pull-in range & jitter characteristics and PJF characteristics in different latencies. From examples calculations and analysis, we could conclude that with the help of PJF, we will get a compromise DCO gain while considering jitter and pull-in range. And with the increase of latency, PJF experiences worse performances.

References