Predictive Control Algorithm for Phase-Locked Loops

Angel Abusleme and Boris Murmann

***Abstract*— Phase-locked loops (PLLs) exhibit a tradeoff between settling time and noise rejection, due to the fact that a low noise PLL requires a narrow bandwidth (BW) loop filter, which degrades settling time. However, the moments when fast settling or good noise rejection is required are clearly identified in a PLL, and this can be used to overcome this tradeoff. A recent technique – PLL gear shifting – exploits this fact by modifying the loop filter BW according to the PLL current objective. In this work, a similar solution based on predictive control techniques is presented. Through a very simple digital loop filter an optimal response is obtained, where a single parameter controls the bandwidth to improve either settling time or noise rejection.**

I. Introduction

Frequency synthesis is essential in many communication circuits, and particularly in Frequency Hopping Spread Spectrum (FHSS) [1], where a signal is modulated in a rapidly changing carrier frequency. In these systems, the circuit frequency hops between different values according to strict settling specifications. Obviously, fast-settling PLLs are highly valued in these applications. The price that a fast settling, large BW PLL pays is a decreased noise rejection.

Many solutions have been proposed to overcome this limitation, mostly as adaptive techniques. One example is the use of adaptive phase comparators [2]. Adaptive loop filters have also been proposed, as hot-standby PLL architecture [3], as optimum loop gain gear-shifting techniques [4], as error-dependent loop filter bandwidth [5], and as continuously adaptive loop filters [6]. Adaptive techniques have also been proposed for the reference frequency and frequency divide ratio [7].

PLLs are usually treated – analyzed and designed – as circuits, whereas they could also be thought of as control systems, as in the work of Abramovitch [8]. This approach should be more relevant nowadays, as digital signal processing is becoming more important in PLLs. As a consequence of CMOS technology scaling, some traditionally analog PLL blocks have been implemented as digital circuits, making possible all-digital PLLs (ADPLL) in CMOS circuits [9] [10] [11]. In this concept, a high resolution time-to-digital converter (TDC) detects the difference between reference and output phases and provides a digital output suitable for any of the numerous digital control systems available.

In this paper, another adaptive technique for PLLs is presented. Relying on predictive control algorithms, the optimal loop filter (controller) transfer function is computed in order to minimize the settling time, by adjusting the relative weights of RMS phase error and VCO action (manipulated variable) in a cost function.

Model-based predictive control (MPC) [12] is a powerful control technique that computes the optimal manipulated variable necessary to steer the controlled variable closest to the reference. The algorithm requires an objective function and a model of the plant.

MPC techniques use the receding horizon concept to overcome noise and modeling errors. The predictive controller computes a sequence of values for the manipulated variable that minimizes the objective function during a defined control horizon (number of time steps in the future), but only uses the first value to control the system during the next time step. Then it updates the measured variables and computes the sequence again, considering the same control horizon as before (receding horizon). This way, noise and modeling errors are not integrated in time.

MPC is optimal by definition, and can be optimized for figures of merit such as RMS error and manipulated variable changes. As predictive controllers are implemented digitally, they are more versatile to adapt. This feature permits using PLL gear shifting by just changing a single parameter, as will be shown later.

II. PLL Model

For predictive controller design purposes, detailed models will produce better responses, but they will require more computational power to obtain the control sequence. As in a high frequency, fast settling PLL there is little time between time steps to perform these computations, the simplest PLL model that captures all first-order features will be used. As the controller deals with phase errors, the model's main signals are phases.

In this model [13], the phase detector yields a signal proportional to the difference between reference and output phases; the VCO is an integrator with a known proportionality constant, and the frequency divider is just a mathematical divider. Fig. 1 shows the block diagram of the model for a generic PLL.

III. Predictive Control Basics

There are many MPC techniques, differing on the system model format, the objective function, the optimization algorithm, among others [12]. However, all of them operate by predicting the system response and computing the manipulated variable to optimize it according to some objective. In this paper, transfer function models are used. The most basic transfer function expresses a single-input, single-output (SISO) system dynamics as follows:

$$y_{k+1} + \sum_{i=0}^{n} A_{i+1} \cdot y_{k-i} = \sum_{j=1}^{n} c_j \cdot u_{k-j+1} \tag{1}$$

where A_i, c_j are constants, and y_k and u_k are the plant output and input for current time k, respectively. For MPC purposes, it is advantageous to express the system response as a function of *changes* in the manipulated variable, instead of the manipulated variable itself. This modification allows to reach zero steady-state error, even when the model is not accurate or noise is present. Thus the new model is

$$y_{k+1} + \sum_{i=0}^{n} A_{i+1} \cdot y_{k-i} = \sum_{j=1}^{n} b_j \cdot \Delta u_{k-j+1}, \tag{2}$$

where b_j are constant coefficients. Let n_y be the prediction horizon, this is, the number of steps to be predicted. Based on (2) it is possible to write the output prediction values for $k+1$ to $k+n_y$ as follows:

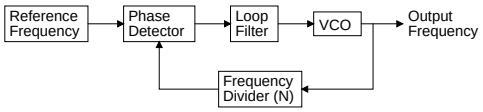


Fig. 1. Block diagram of a simple PLL model.

$$\begin{matrix} y_{k+1} + \sum_{i=0}^{n} A_{i+1} \cdot y_{k-i} = \sum_{j=1}^{n} b_j \cdot \Delta u_{k-j+1} \\ \vdots \\ y_{k+n_y} + \sum_{i=0}^{n} A_{i+n_y} \cdot y_{k+n_y-i-1} = \sum_{j=1}^{n} b_j \cdot \Delta u_{k-j+n_y} \end{matrix} \quad (3)$$

or, expressed as matrix operations,

$$C_A \underset{\rightarrow k}{y} + H_A \underset{\leftarrow k}{y} = C_{zb} \underset{\rightarrow k-1}{\Delta u} + H_{zb} \underset{\leftarrow k-1}{\Delta u} \quad (4)$$

where $\rightarrow k$ indicates future n_y values from $k+1$, and $\leftarrow k$ indicates previous n_y values up to instant k. This notation has been taken from the work due to Rossiter [12], where more comprehensive information on these derivations is available. Let $H = C_A^{-1} C_{zb}$, $P = C_A^{-1} H_{zb}$, $Q = -C_A^{-1} H_A$. Then, the predicted sequence of values for y is

$$\underset{\rightarrow k}{y} = H \underset{\rightarrow k-1}{\Delta u} + P \underset{\leftarrow k-1}{\Delta u} \; Q \underset{\leftarrow k}{y} \quad (5)$$

Now the optimal sequence for the manipulated variable will be computed. Let the objective function be

$$J = \left\| \underset{\rightarrow}{r} - H \underset{\rightarrow}{\Delta u} - P \underset{\leftarrow}{\Delta u} - Q \underset{\leftarrow}{y} \right\|_2^2 + \lambda \left\| \underset{\rightarrow}{\Delta u} \right\|_2^2 \quad (6)$$

where $r_{\rightarrow}$ is a vector of future reference values, and λ is a weighing coefficient for changes in u; if λ is large, Δu is strongly penalized, so the system response is smooth and slow (small loop filter BW); if λ is small, the response is faster (large loop filter BW). If this problem is unconstrained, the objective function has a minimum at

$$\underset{\rightarrow}{\Delta u} = \left(H^T H + \lambda I \right)^{-1} H^T \left(\underset{\rightarrow}{r} - P \underset{\leftarrow}{\Delta u} - Q \underset{\leftarrow}{y} \right). \quad (7)$$

This represents the sequence of next n_y values of Δu, necessary to minimize the objective function in the next n_y steps.

IV. Predictive Control of a PLL

The phase detector yields a signal proportional to the difference of phases, but there is no simple way to find each operand. The PLL locking process can be visualized in the phase trajectory plane (Fig. 2). Here, the reference phase is a ramp, whereas the output phase moves along straight lines when output frequency is constant, and the slope varies when changes in the manipulated variable are effected (times t_1 and t_2 in Fig. 2). Whenever N changes, the slope changes too, so when the loop is locked, changes in N are simply known disturbances. The error is the difference between reference and output phases.

As there is no access to reference or output phases, the system output is the phase error, expressed as a function of previous phase errors and *changes* of the VCO voltage. Let y_k be the phase error in time k. By definition,

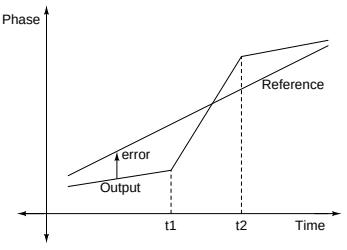


Fig. 2. Trajectories for reference and output phases. The only measurable variable is the difference between both, which must be ideally zero.

$$y_{k+1} = y_k + \Delta y_{k+1}. \quad (8)$$

Therefore, the error change Δy_{k+1} must be estimated in order to predict the error in the next time step y_{k+1}. The phase error y_k is the difference between reference phase ϕ_r and output phase ϕ. The reference phase is the integral of the reference frequency f_o. As f_o is constant, the phase is the product of f_o and time:

$$\phi_{rk} = 2\pi f_o \cdot T \cdot k \quad (9)$$

In this equation, k is the discrete time index and T is the sampling period. The instantaneous output frequency at time step k is proportional to the VCO input voltage u_k and the VCO constant K_o (expressed in $\mathrm{rad/Vs}$), and inversely proportional to the divider constant N. Its integral over time yields the output phase, which is subtracted from (9) to obtain the phase error:

$$y_k = 2\pi f_o \cdot T \cdot k - \sum_{i}^{k} \frac{K_o u_i T}{N_i}. \quad (10)$$

Substituting equation (10) in equation (8) and using the Δ operator in u, the change in the phase error Δy_k for a single time step is

$$\Delta y_k = y_k - y_{k-1} = 2\pi f_o \cdot T - \frac{K_o \left(u_{k-1} + \Delta u_k \right) T}{N_k}. \quad (11)$$

As N changes only for frequency hops, here it is assumed constant. Equation (11) can be then manipulated to yield

$$\Delta y_k = \Delta y_{k-1} - \frac{K_o \Delta u_k T}{N}. \quad (12)$$

Shifting expression (12) one time step ahead and substituting into equation (8) yields a suitable PLL model for MPC:

$$y_{k+1} = 2y_k - y_{k-1} - \frac{K_o \Delta u_{k+1} T}{N} \quad (13)$$

Equation (13) expresses phase error y_{k+1} in terms of previous phase errors and next change in VCO voltage Δu_{k+1}.

Now the model found in (13) will be used to obtain a control law according to Section III. Using the notation in (3), the following coefficients are readily obtained: $A_1 = -2$, $A_2 = 1$, and $b_1 = -K_o T/N$. All other coefficients are zero. This simple result characterizes most parameters of a PLL; whether a different PLL were used, only b_1 would change. As most other coefficients are zero, matrix products become really simple; in fact, prediction matrix P is zero.

Using the result from previous section and realizing that r (reference for phase error) is also zero, the predictive control law becomes

$$\underset{\rightarrow}{\Delta u} = - \left(H^T H + \lambda I \right)^{-1} H^T Q y, \quad (14)$$

which is just a product of a periodically constant matrix defined as $- \left(H^T H + \lambda I \right)^{-1} H^T Q$, and a variable vector y. The matrix has many zeros and will be constant for a fixed λ and N. If any of both is changed, the matrix must be recomputed. The product yields a vector consisting of the future changes in the manipulated variable.

Finally, as MPC relies on receding horizon, only the first value of vector Δu is used, so the control law can be reduced to the dot product of two vectors of length n_y, which are the first row of matrix $C_x = - \left(H^T H + \lambda I \right)^{-1} H^T Q$, and vector y. Moreover, only a few terms of matrix C_x are nonzero; particularly, only the first two columns of matrix Q have nonzero elements, so only the first two

terms of the first row of matrix C_x are nonzero, which means that the control law is reduced to

$$\Delta u_{k+1} = C_1 y_k + C_2 y_{k-1}, \tag{15}$$

where C_1 and C_2 are the two nonzero terms mentioned above and can be computed for different values of λ and N using equation (14).

The control law derived in this section can be used in any PLL whose behavior is close to the model of Fig. 1 by computing coefficients C_1 and C_2 for all values of N and λ, provided that T is sufficiently small. The only tuning parameters are n_y and λ. Interestingly, the MPC for the model presented computes u as a linear combination of y (delayed) and its integral. This simple PI-like result is linear, and optimal for the model and objective function specified. However, the real potential of MPC is best appreciated when deadtime is considered in the model. The control algorithm for this case is explained next.

Control algorithm for PLL with delay in the loop

If the loop delay is small compared to settling time, the controller as described should overcome it; if this assumption does not hold and the system becomes unstable, there is a model-based solution. If the delay can be estimated (e.g., the average delay from a change in u to an effective VCO action), it can be included in the PLL model. In this case, matrix P is no longer zero and must be considered. Intuitively, the system must remember previous actions when computing next action, as previous actions take some time to act. This additional term allows controlling systems with large delays in the loop.

V. Simulation Results

This section presents simulation results of a PLL using MPC. Part V-A shows how the PLL controller responds to changes in its three parameters. Part V-B presents behavioral PLL simulations in time.

For proof-of-concept simulations, generic ω_o and T will be used:

- Step time: 1
- Reference frequency ω_o: 0.01
- Initial and final N: 75 and 125
- Initial and final output frequency: 0.75 and 1.25
- Min and max VCO freq. (behavioral simulation): 0.7 and 1.3
- VCO and phase detector gain: 1

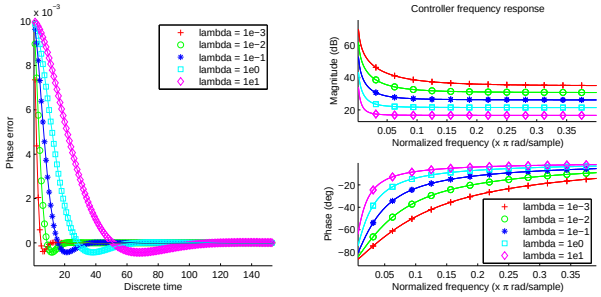
A. Prediction calculations

A MATLAB® script was written to compute the prediction matrices and plot y for different parameters, using the ideal PLL model presented earlier. The step in N occurs at $t = 0$, and the PLL is perfectly locked for $t < 0$. A λ-sweep was done, with λ between 10^{-3} and 10. The results are shown in Fig. 3(a). The prediction horizon was 200 time steps. As λ increases, changes in Δu are more penalized and the response becomes slower. This effect can also be confirmed in the controller frequency response, shown in Fig. 3(b), where higher λ implies smaller bandwidth.

B. Time-domain Phase Simulations

Now the results of time-domain simulations for phase variables are shown. Different values of λ, n_y and loop delay are used. N steps at $t = 100$ for the next two plots, and at $t = 500$ for the last three. Here u is limited between 0.7 and 1.3.

It was found that settling time is longer than that expected from VCO operation limits, when n_y is shorter than the non-slewing settling time; on the other hand, if n_y is long enough, settling time is only slightly sensitive or totally insensitive to changes on n_y. This agrees with the MPC operation, as the sequence of u will be



(a) Phase error time response. (b) Controller frequency response.

Fig. 3. PLL response for different values of λ.

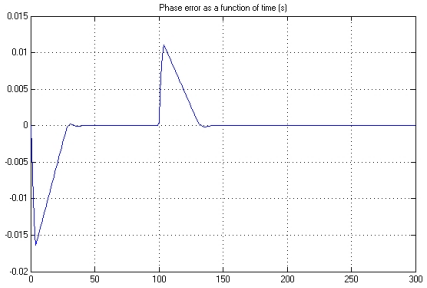


Fig. 4. Phase error in a PLL time-domain simulation for phase variables.

the same for different values of n_y as long as it covers the entire settling process. For the numbers in the example, the minimum n_y that ensures fast settling is 8. Longer n_y does not help, but shorter n_y increases settling time.

Non-slewing settling time grows roughly with the logarithm of λ, but the PLL becomes unstable for large values of λ. This is because, for large λ, the objective function privileges Δu minimization over y, thus settling time is increased and exceeds n_y. This shows that short term plans in long time constant systems lead to wrong decisions.

To show the mentioned behavior, two simulation results are shown. Fig. 4 is for $\lambda = 0.01$, $n_y = 8$. This presents a stable, smooth response. The difference between this response and that computed using prediction calculations (Fig. 3(a), for $\lambda = 0.01$) is mainly due to the effect of the VCO output frequency limits, not modeled before.

Fig. 5 is for $\lambda = 5$, $n_y = 10$. Here, as λ is large, the expected settling time is large too; although n_y is larger than that in Fig. 4, it is small compared to the expected settling time, thus the PLL oscillates.

Other simulations showed that the effect of longer T is not critical as long as the expected settling time is at least $8\times$ longer than T.

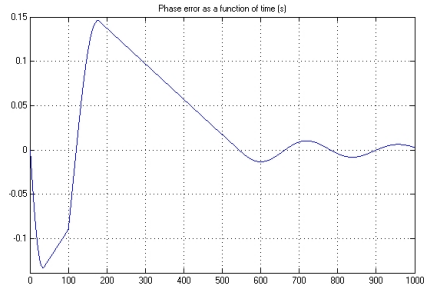


Fig. 5. Phase error in a PLL time-domain simulation for phase variables using a relatively short prediction horizon.

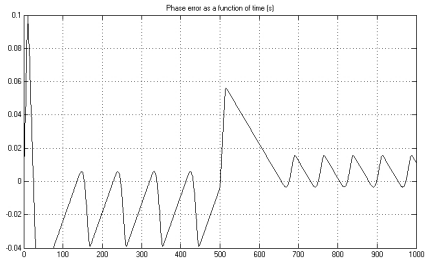


Fig. 6. Phase error in a PLL time-domain simulation, 10-step loop delay.

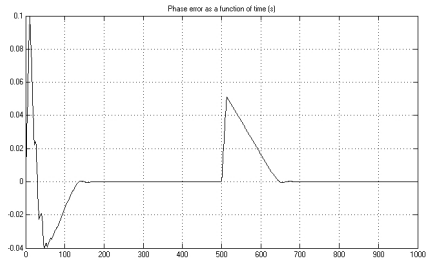


Fig. 7. Phase error in a PLL time-domain simulation, with 10-step loop delay and a proper PLL model.

C. Long delay simulations

In order to test the model with a delayed loop, two simulations were run including a 10-step delay: Fig. 6 without delay compensation and Fig. 7 with delay compensation. Both use $\lambda = 0.1$, $n_y = 40$. The first one shows how the controller fails to lock with such a long delay, while the second one presents an optimal response.

From these and other simulation results, it was found that stability is more sensitive to loop delay for small λ, whereas larger λ controllers are more stable when the delay is not modeled. However, better responses are attained when the delay is included in the model.

D. Comparison against a traditional PID

As mentioned earlier, for the PLL model and objective function presented, the optimal controller is a PI. Thus the advantage of MPC over PID in this case is the tuning methodology, which ensures optimal results when the model is appropriate. MPC becomes more interesting, however, when a delay in the loop is considered in the model. In this case, the optimal controller is no longer a simple PI controller.

For comparison purposes, the same simulation setup used for the 10-step delay on the MPC was used for the PID controller evaluation. The PID controller, tuned by trial-and-error, was designed to provide a fast, stable response. Its tuning was considerably more tedious than simply adjusting λ for the MPC.

Fig. 8 presents the PID controller response. The MPC response (Fig. 7) is smoother and faster than the PID's. Settling time, arbitrarily defined as the number of time steps required to reach 1% error, was computed for both, yielding 151 time steps for MPC and 223 time steps for the PID. Other simulations not presented in this work showed that the PID controller is barely stable for a 20-step delay, whereas the MPC still performs optimally.

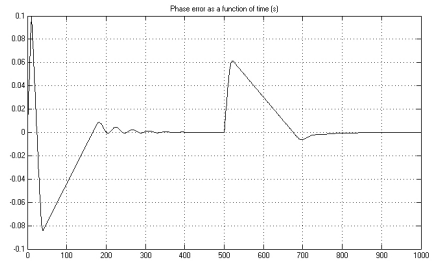


Fig. 8. Phase error in a PLL time-domain simulation with 10-step loop delay, PID controller with $K_P = 10$, $K_I = 0.15$, $K_D = 12$.

VI. Conclusion

The results shown in this paper support the idea of using MPC in PLLs. For PLLs where the loop delay is negligible, the control law presented is linear, easy to tune and presents optimal results. It can be considered as a new design methodology for optimal PI controller for PLLs; it is a digital loop filter product of an optimization process instead of a traditional filter design from frequency response considerations. It also allows gear shifting by scheduling the value of a single parameter. As a corollary, it can be stated that a PI controller is indeed the optimal solution for the problem reflected by the objective function presented, and the method shown computes the optimal controller parameters as a function of λ. If loop delay is large and has small variance, it can be included in the model and the controller will work optimally, but the control algorithm will no longer be a PI, requiring more computations per step. Simulations show that, under these conditions, the MPC outperforms the PID controller in both settling speed and stability.

References

[1] M. Ristenbatt and J. Daws, J., "Performance criteria for spread spectrum communications," *IEEE Trans. Commun.*, vol. 25, no. 8, pp. 756–763, 1977.

[2] J. Eijselendoorn and R. Den Dulk, "Improved phase-locked loop performance with adaptive phase comparators," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-18, no. 3, pp. 323–332, 1982.

[3] M. Mizuno *et al.*, "A 0.18 μm CMOS hot-standby phase-locked loop using a noise-immune adaptive-gain voltage-controlled oscillator," in *ISSCC Dig. Tech. Papers*, 1995, pp. 268–269, 377.

[4] G.-T. Roh, Y. H. Lee, and B. Kim, "Optimum phase-acquisition technique for charge-pump PLL," *IEEE Trans. Circuits Syst. II*, vol. 44, no. 9, pp. 729–740, 1997.

[5] J. Lee and B. Kim, "A low-noise fast-lock phase-locked loop with adaptive bandwidth control," *IEEE J. Solid-State Circuits*, vol. 35, no. 8, pp. 1137–1145, 2000.

[6] C. Vaucher, "An adaptive PLL tuning system architecture combining high spectral purity and fast settling time," *IEEE J. Solid-State Circuits*, vol. 35, no. 4, pp. 490–502, 2000.

[7] Y. Tang, Y. Zhou, S. Bibyk, and M. Ismail, "A low-noise fast-settling PLL with extended loop bandwidth enhancement by new adaptation technique," in *Proc. 14^{th} Annual IEEE Intl. ASIC/SOC Conference*, 2001, pp. 93–97.

[8] D. Abramovitch, "Phase-locked loops: a control centric tutorial," in *Proc. of the 2002 American Control Conference*, vol. 1, 2002, pp. 1–15.

[9] R. B. Staszewski *et al.*, "All-digital PLL and transmitter for mobile phones," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2469–2482, 2005.

[10] R. Staszewski, S. Vemulapalli, P. Vallur, J. Wallberg, and P. Balsara, "1.3 V 20 ps time-to-digital converter for frequency synthesis in 90-nm CMOS," *IEEE Trans. Circuits Syst. II*, vol. 53, no. 3, pp. 220–224, 2006.

[11] R. B. Staszewski and P. T. Balsara, "All-digital PLL with ultra fast settling," *IEEE Trans. Circuits Syst. II*, vol. 54, no. 2, pp. 181–185, 2007.

[12] J. Rossiter, *Model-Based Predictive Control: A Practical Approach*. CRC Press, 2003.

[13] R. Best, *Phase-Locked Loops: Design, Simulation, and Applications*. McGraw-Hill Professional, 2003.