Neuronal phase locked loops: models and circuits

Tesse Elbert Tiemens
S3652327

July 2023

First examiner: Elisabetta Chicca
Second examiner: Bart Besselink
Daily supervisor: Michele Mastella

Abstract

Recent studies have shown that we do not only sense textures of materials spatially, but also temporally [1, 2]. By sliding our skin past a texture, vibrations are induced, the peaks of which are directly represented by spikes in the brain [3]. To decode the frequencies in these signals, a neuronal version of a phase locked loop has been proposed [4, 5], the Spiking Phased-Locked Loop (sPLL). This sPLL was redesigned using readily available building blocks [6], and later implemented in a CMOS chip. For this thesis, this implementation was tested, both in simulation and on-chip, and it was found that the circuit is capable of both single- and dual-frequency detection under idealized circumstances. A qualitative theory for modelling the behaviour of the Spiking Phased-Locked Loop (sPLL) was developed, tested, and found to accurately predict the behaviour of the circuit. Furthermore, based on this theory, an improved version of the sPLL was designed, which was tested using python simulations and found to have greater phase- and output stability than the original sPLL. Finally, a possible way forward towards a quantitative model of the sPLL, based on event-triggered control theory [7], was discussed.

university of groningen

Contents

1 Introduction

Touch in humans is a surprisingly complex system. Unlike what was previously thought [8], recent studies have shown that the different mechanoreceptors in the skin are not strongly separated by their roles in decoding stimuli. Instead, they all work together on the different aspects of the sense we call touch, such as texture, shape, and vibration [1]. For instance, when a texture is too fine for the receptors 'classically' associated with shape and textures (the Slowly adapting type 1 (SA1) receptors) we slide our finger over the texture, allowing the vibration sensitive Pacinian Corpuscle (PC) receptors to take over, encoding the texture not spatially, but temporally [2]. As the authors of [2] show, the different micro-structures of various textures create different frequencies of vibrations, which can then be used to distinguish materials (see Figure 1). These frequencies then get sent to the cortex, where they have been found to be encoded not in mean firing rates - how often a neuron fires on average - but rather through the phase-locked response of certain groups of cortical neurons [3].

This behaviour agrees with earlier results in research on the whiskers of rodents, which found that cortical neurons can exhibit self-oscillatory behaviour, and that cells in the thalamus - through which the stimuli from the mechanoreceptors pass - can modulate these oscillations [4]. The authors thus propose a Spiking Phased-Locked Loop (sPLL), a neuronal version of the Phased-Locked Loop (PLL), a popular circuit consisting of an oscillator and a phase detector, which modulates the frequency of said oscillator based on the phase difference between it and an external input [9, 10]. In [5], the authors explore this sPLL further, demonstrating how a bank of sPLLs, all tuned to different frequencies, do not only exhibit phase-locking behaviour, but can also help decode the frequencies in the signal, serving as a neuronal Discrete Fourier Transform (DFT). They can do this because the output of an sPLL depends on the frequency difference between the internal and external oscillations, and so the firing rates of this population of neurons can easily be used in further processing.

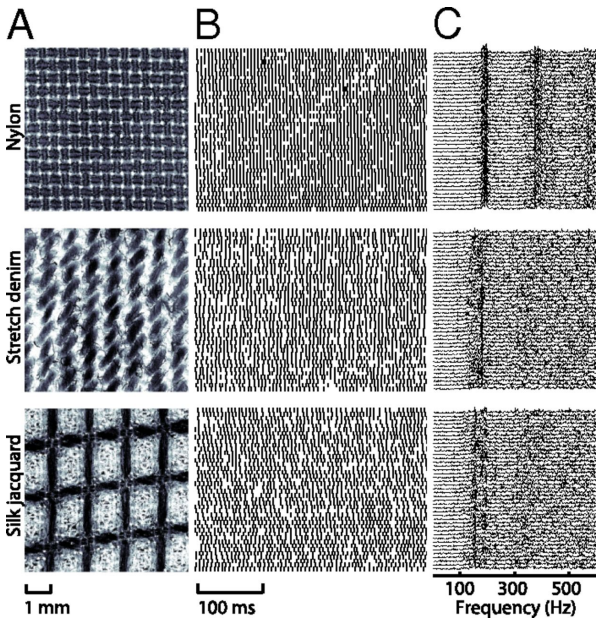

Figure 1: Different fabrics and the PC receptor responses corresponding to them. The receptors were scanned over the fabrics, of which the micro-structures are shown (**A**), 42 times, each time at a different displacement orthogonal to the scanning direction. The spike trains elicited from this are shown under (**B**), with the frequency spectra of these spike trains shown under (**C**). As can be seen, the different fabrics correspond to different frequencies in the spike trains. Image taken from [2].

One of the issues with the sPLL model is that, contrary to the regular PLL, which is well-founded

in control theory, no proper theoretical framework exists to analyze its behaviour, and so most results simply rely on simulation. While control theories of neuronal systems do exist, they either rely heavily on average spike rates, such as the original theory from Carver Mead [11], or model systems in a fully continuous fashion, using complex dynamical equations, such as [12, 13], making it difficult to find analytical solutions. Instead, the dynamics of the sPLL depend on spike timing, not average rate, and, in order to proper model implementations, of interest are the high-level, analytical dynamics of the closed loop. At the same time, due to being a simple, closed-loop system, with a well-understood non-spiking counterpart, the sPLL might be a very good starting point to develop such a theory, which can then be applied to other neuronal systems as well.

Another issue with the sPLL is that, since its inception in the late 90s in [4, 5], it has seen very little further development. Only recently the concept was picked back up in [6], where the authors propose an sPLL based on the TDE. The Time Difference Encoder (TDE) is a structure developed in [14], which allows for precise detection of the time between the arrival of two spikes. This sPLL was designed to be easily realizable using analog Complementary Metal Oxide Semiconductor (CMOS) technology, and so has been built by the Bio-Inspired Circuits and Systems (BICS) group at the University of Groningen. By implementing it in such a way, a road is opened to further development in the direction of artificial touch, as well as more general applications where the decoding of frequencies is important.

The purpose of this thesis is to test this implementation of the sPLL, explore its characteristics, and use it to take a deeper look at the dynamics of sPLLs in general. Specifically, we look to test whether the sPLL, both in simulation and on-chip, is capable of locking to- and detecting frequencies, and whether it can detect multiple frequencies at the same time, essentially functioning as a Discrete Fourier Transform (DFT). To help us do so, we develop and test a qualitative model of the sPLL, allowing us to gain a better understanding of its behaviour in lieu of a control theory. Finally, we discuss a possible improved sPLL, and make the first steps towards creating a control theory of the system.

2 Theory

2.1 Phase locked loops

A fundamental Phased-Locked Loop consists of a closed loop of two elements: a phase detector and a Voltage Controlled Oscillator (VCO). The (ideal) phase detector is an element which has a transfer function such that its output V_{PD}, averaged over one cycle, is linearly related to the phase difference between the two inputs $\overline{V_{\mathrm{PD}}} = \kappa\Delta\phi$ [10]. Assuming further that the response of the VCO is linear, e.g. $f_{\mathrm{VCO}} = f_{\mathrm{in}} + \alpha\overline{V_{\mathrm{PD}}}$, where α is the VCO sensitivity, and f_{in} is the internal, 'free-running' frequency of the oscillator. We can now use these relationships, combined with the fact that the rate of change of the phase of an oscillator is equal to its frequency, to find (where ϕ_{ext} and f_{ext} refer to the external oscillator phase and frequency respectively):

$$\begin{aligned}\frac{d\Delta\phi}{dt} &= \frac{d\phi_{\mathrm{ext}}}{dt} - \frac{d\phi_{\mathrm{VCO}}}{dt} \\ &= f_{\mathrm{ext}} - f_{\mathrm{VCO}} \\ &= f_{\mathrm{ext}} - f_{\mathrm{in}} - \alpha\overline{V_{\mathrm{PD}}} \\ &= \Delta f - \alpha\kappa\Delta\phi\end{aligned}$$

Solving this differential equation for $\Delta\phi$ yields (with c a constant determined by initial conditions):

$$\Delta\phi(t) = \frac{\Delta f}{\alpha\kappa} + ce^{-\alpha\kappa t} \tag{1}$$

This shows us that our ideal PLL converges asymptotically to a stable state where $\Delta\phi$ is proportional to Δf, the intrinsic frequency difference between the two oscillators, and $\frac{d\Delta\phi}{dt} = f_{\mathrm{ext}} - f_{\mathrm{VCO}} = 0$. This state, in which the VCO frequency is equal to the input frequency, is called the 'phase locked' or simply 'locked' state of the PLL.

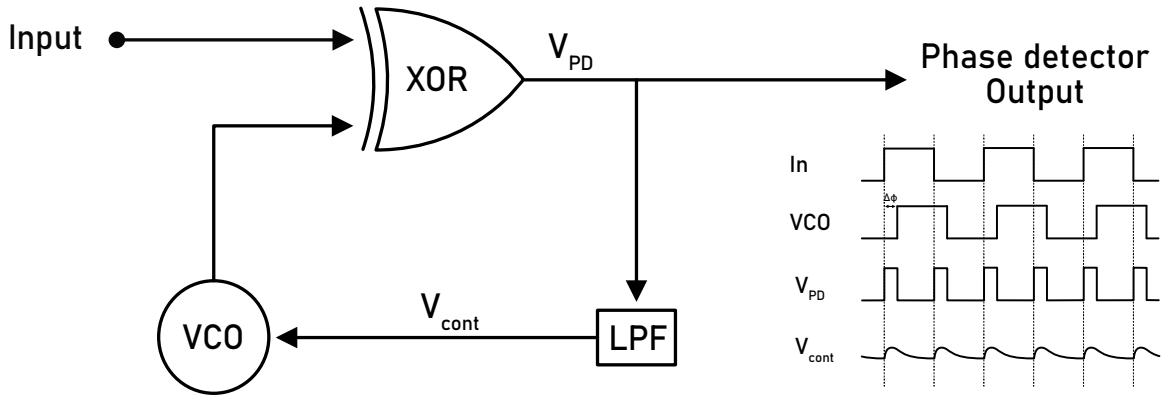

Figure 2: A diagram of a PLL using a XOR gate as a phase detector. The output of the XOR gate is high when the two inputs differ in state, so that the pulse width of its output is directly related to the phase difference $\Delta\phi$. This is then sent to a Low-Pass Filter (LPF) to help with stability of the circuit, as well as suppressing side-bands, before being sent to the VCO. In an ideal case, V_{cont} would be flat, but here it is depicted with some amount of ripple. Adapted from [10].

In order to make sure that the VCO only perceives the average $\overline{V_{\mathrm{PD}}}$, a Low-Pass Filter (LPF) is often included in-between the phase detector and the VCO, such as in the diagram in Figure 2. Here, the phase detector is implemented as a simple XOR gate, which, since its output is high when

the two inputs differ in state, has its average output indeed linearly dependent on the input phase difference (do note that the output can not go negative, so that $\overline{V_{\text{PD}}} \sim \kappa|\Delta\phi|$). If we were to feed this output straight into our VCO however, while it would likely manage to lock to an extend, the rapid modulation would create large sidebands in the signal. By adding a low-pass filter, this signal is smoothed out, reducing sidebands, while still passing on the phase difference information.
Another benefit of having $\overline{V_{\text{PD}}}$ directly represented in the circuit is that it allows us to demodulate frequency modulated (FM) signals. In such a case, the frequency of the input signal f_{ext} is modulated away from a constant 'carrier' frequency by some other signal $g(t)$: $f_{\text{ext}} = f_{\text{carrier}} + g(t)$. If we now tune our VCO such that its internal frequency matches this carrier frequency, the output becomes (assuming the frequency modulation is significantly slower than the time-constant $\alpha\kappa$, so that we can ignore the transient behaviour):

$$\begin{aligned}\overline{V_{\text{PD}}} &= \kappa\Delta\phi(t) \\ &= \kappa\frac{\Delta f}{\alpha\kappa} \\ &= \frac{f_{\text{carrier}} + g(t) - f_{\text{carrier}}}{\alpha} \\ &= \frac{g(t)}{\alpha}\end{aligned}$$

retrieving our original signal.

Of course, the above only holds when the phase difference is known exactly, which is often not possible. Most of the time, the phase difference is approximated using, for instance, signal multipliers for sinusoidal oscillators, or structures like the XOR gate discussed above for square-wave signals. Depending on the signal, these might have a limited lock-in range, such as $\phi \in (-\pi/2, \pi/2)$ for the sinusoidal case. This also means that there is a limited range of frequencies that can be locked to, depending on the phase detector and VCO sensitivities, as well as the LPF characteristics [9].

2.2 Neurons, spikes and synapses

2.2.1 Neurons

In the brain, computation is done using neurons. Neurons are specialized cells, that can move certain ions between the inside and outside of their cell membrane, in order to create a potential difference, the membrane potential (V_{mem}). In the simplified Leaky Integrate-and-Fire (LIF) model of Lapicque [15], once this membrane potential reaches a certain level, the threshold voltage, a positive feedback loop starts, creating a large voltage fluctuation called an action potential, or, more simply, a spike [16] (see Figure 3).

Mathematically, such a neuron is often modelled as a fast-slow system, similar to a Van der Pol oscillator [17]. These models use a pair of differential equations, one 'fast', higher order equation, governing the rapid state changes when the neuron spikes, and one 'slow', lower order equation, modeling the internal dynamics as charge is accumulated. In the well-known FitzHugh-Nagumo model[18, 19], these equations take the form:

$$\begin{aligned}\dot{x} &= c(y + x - x^3/3 + z) \\ \dot{y} &= -(x - a + by)/c\end{aligned} \tag{2}$$

where x is the membrane potential, and y an auxiliary variable. a and b are constants, c the damping coefficient, and z describes an external input. The system can be driven to spike either by a sudden change in z, or by setting z such that the nullclines of x and y intersect in a way to cause a continuous train of impulses.

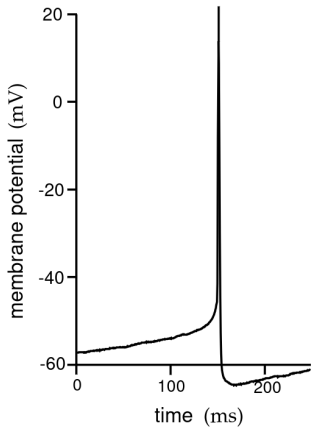

Figure 3: An action potential or spike. Taken from [16]

More recently, another method has gained traction, which uses a concept called mixed feedback. As explained in [12], by viewing the neuron as a control system, in which the controller possesses both a fast, positive feedback element and a slow, negative feedback element, the spiking behaviour can be modelled using principles from control theory. As the membrane potential reaches the threshold needed to spike, the positive feedback activates, starting the spike, before the negative feedback takes over, bringing the neuron back to its resting state.

2.2.2 Spike trains

A number of spikes together form a spike train, which can encode information. Within a single neuron, this information can be encoded in two different ways: through time-based coding or through rate-coding. In time-based coding, the specific time of arrival of each spike encodes information. Following Ahissar in [5], we can express this time information either by using Inter-Spike Intervals (ISIs), simply taking the time difference between two consecutive spikes, or as a cumulative time difference w.r.t. some constant-frequency 'carrier' signal, essentially treating the spike train as an FM signal, sometimes called phase coding.
While they disagree about the specific implementation, both Ahissar [5] and Dayan and Abbott [16] agree that rate-coding refers to a type of neural encoding in which only the average spike rate $r(t)$ in a certain time-frame, the rate bin, matters. Where they differ is that Ahissar defines the time bins as static, essentially creating a histogram of spikes, whereas Dayan and Abbott use a sliding window between time t and $t + \Delta t$ as their time-bin. Such a sliding window representation, using a window of $(t - \Delta t/2, t - \Delta t/2)$, with $\Delta t = 1$ s, is shown in Figure 4, along with an ISI representation of the same signal.

2.2.3 Synapses

In order for spikes to travel from one neuron to another, they are transmitted through synapses, the places where two neurons meet. In synapses, the arriving spikes from the presynaptic (sending) neuron trigger the release of neurotransmitters into the space between the two neurons. These then get absorbed by the dendrite of the postsynaptic (receiving) neuron, causing ion-channels to open, changing the membrane potential (V_{mem}) of the postsynaptic neuron. If the presynaptic spike causes the membrane potential of the postsynaptic neuron to increase, we speak of an excitatory synapse. If, instead, the membrane potential decreases, we speak of an inhibitory synapse.

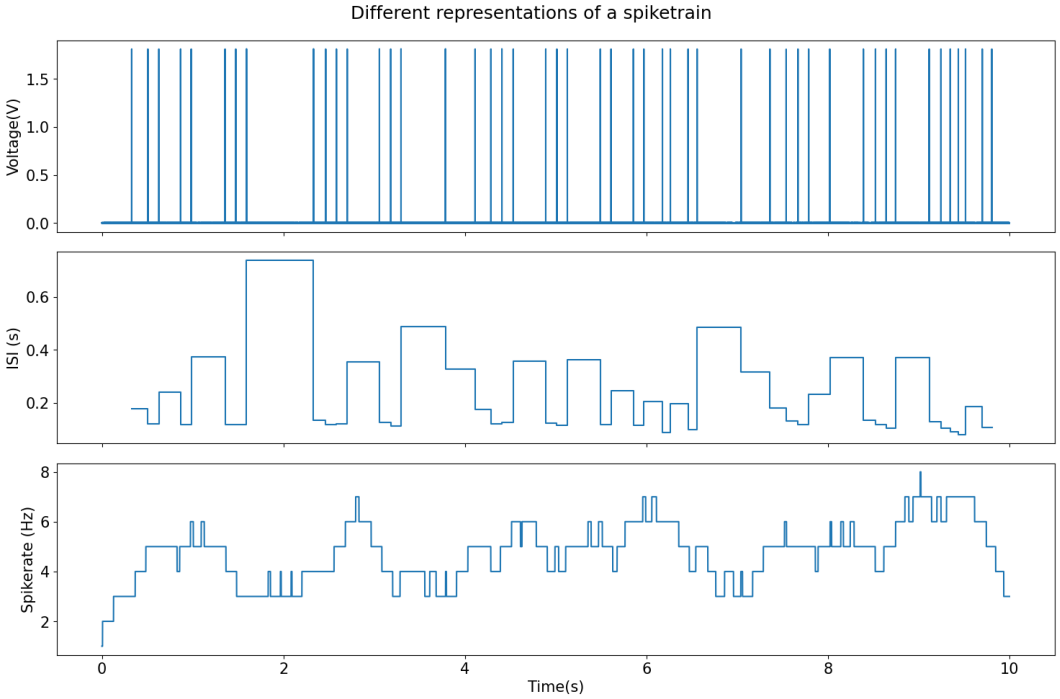

Figure 4: Different representations of a spike train. **Top:** the original spike train, measured from the Cognigr1 chip. **Center:** the ISI representation of the same spike train. **Bottom:** the spike rate representation of the spike train, using a sliding window of $(t - \Delta t/2, t - \Delta t/2)$, with $\Delta t = 1$ s.

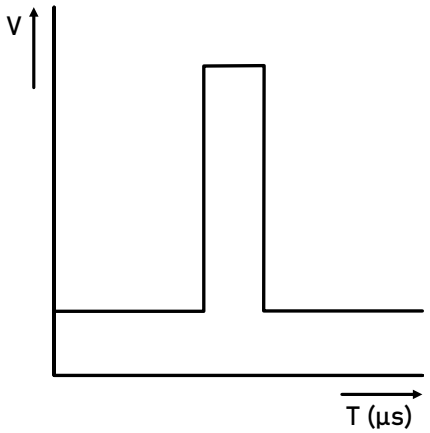

Figure 5: A representation of a 'spike' as transmitted in the analog CMOS circuits used in the sPLL. In our specific implementation, the peak voltage would be close to 1.8 V, and the pulse width would be on the order of 100 μs.

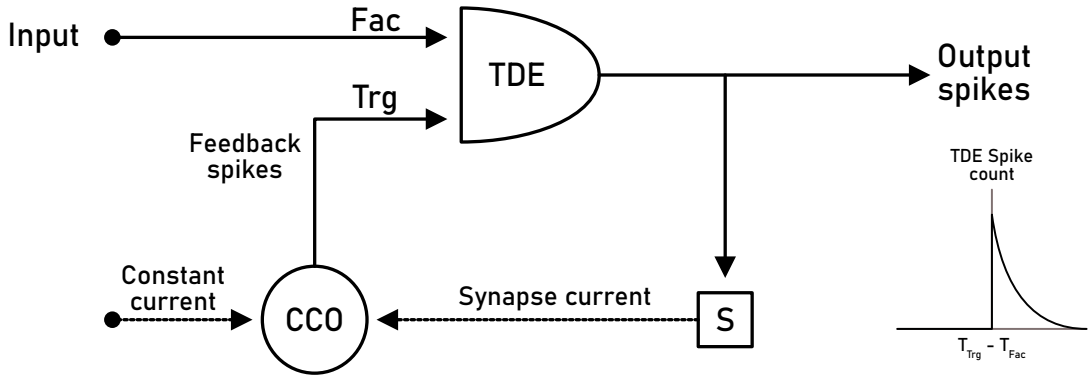

Figure 6: The sPLL. The free-running frequency of the Current Controlled Oscillator (CCO), an integrate-and-fire neuron, is set by an external constant current input, with a higher current corresponding to a higher frequency. To this current is added a current from the synapse (S), depending on the spike rate of the TDE, which serves as the phase detector in the sPLL. Using a gated synapse (see Section 2.3.4), the output spike rate of the TDE falls like $r \sim e^{-\Delta t/\tau}$, where $\Delta t = t_{\mathrm{Trg}} - t_{\mathrm{Fac}}$ and τ a time constant. When $t_{\mathrm{Trg}} < t_{\mathrm{Fac}}$, no spikes occur (see inset).

2.2.4 Artificial neurons, spikes and synapses in CMOS technology

The artificial neurons and synapses used for the sPLL discussed in this thesis are analog CMOS circuits, designed in such a way as to emulate the basic functions of real neurons and synapses. To help with robustness, the spikes transmitted are not the full action potentials, like in Figure 3, but instead simple voltage pulses, as shown in Figure 5. Another difference is that in these circuits, the input to neurons are not neurotransmitters, but currents generated from the synapses directly. The neurons then use these currents to create their own equivalent to a membrane potential, and, faithful to biology, use positive and negative feedback loops to create a voltage spike once a certain threshold is reached. These spikes are then sent to synapses, who in turn generate a current based on this spike.

2.3 The Spiking Phased-Locked Loop (sPLL)

As can be seen from Figure 6, the sPLL [6] has a similar structure to the PLL, but with several differences. The VCO has been replaced with a Current Controlled Oscillator (CCO), in the form of a LIF neuron. The role of the phase detector is taken by a TDE, a neuron with a gated synapse, which is sensitive to the time delay between the spikes on its two inputs, first developed by Milde et al. in [14]. Finally, the LPF is replaced by a single synapse, which also serves the role of converting the voltage signal coming from the TDE into the current signal required for the CCO. The specifics of these subcircuits are discussed in the following subsections, but for those with a lack of time or interest, we will give a quick summary here:

As can be seen in the inset of Figure 6, the TDE is designed such that the output spike rate falls off exponentially with the time difference between the internal CCO spike and the input spike if the CCO spike arrives last. If the CCO spike arrives first however, no spikes occur. These spikes are then integrated by the synapse, which releases a current to the CCO following $\sim e^{-t/\tau_s}$ where t is the time since the arrival of the incoming spike. This current then causes the CCO to spike at a faster rate, making this an excitatory sPLL (ePLL).

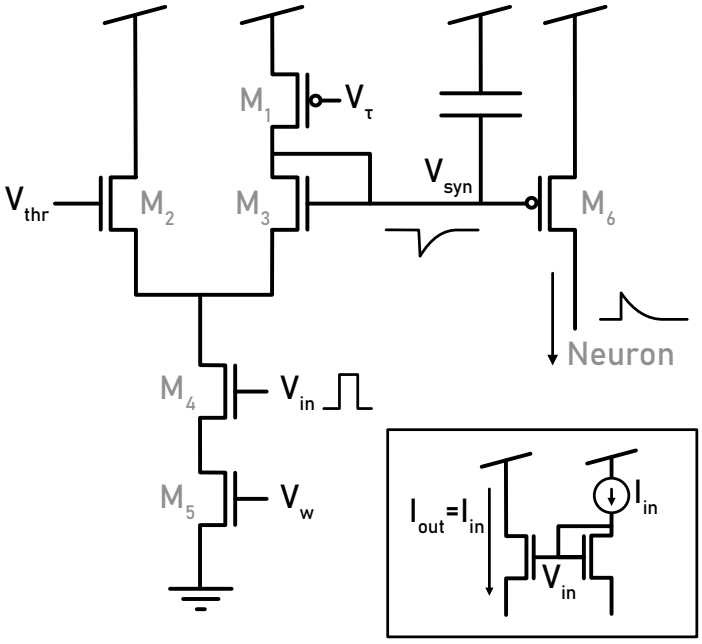

Figure 7: The DPI synapse used in the sPLL, as described in [21]. **Inset**: the current mirror structure used to set V_{thr}, V_w, and V_τ.

2.3.1 The Differential-pair Integrator (DPI)

The CCO synapse and neuron, as well as the neuron inside of the TDE, use (adapted versions) of existing circuits, which were described in [20, 21]. They are based on a structure called the Differential-pair Integrator (DPI), for which the response to an input current follows a non-linear first-order differential equation. However, as long as the input current is sufficiently large, it can be simplified to [21]:

$$\tau \frac{dI_{\text{out}}}{dt} = \frac{I_{\text{thr}}}{I_\tau} I_{\text{in}} - I_{\text{out}} \tag{3}$$

Where I_{thr} and I_τ are currents to be set as parameters using current mirrors (Figure 7 inset), and τ is a time-constant dependent on I_τ, the temperature, the capacitance of the circuit capacitor, and some transistor-dependent constants. Given that this is now a linear first order differential equation, it becomes clear that the behaviour of the Differential-pair Integrator (DPI) follows exponential decay curves.

2.3.2 The DPI synapse

The synapse used in the sPLL, which is based on the DPI, can be seen in Figure 7 [20]. It works as follows: when spikes (see section 2.2.4) arrive from V_{in}, M_4 becomes conductive. This then allows a current, I_{in}, to flow, which is modulated by the gain, V_w. This current is pulled partially through M_2, partially through M_1 and, depending on the state of the circuit, is partially pulled from the capacitor, pulling down V_{syn}. The part coming from M_2 relates to I_{thr} in equation 3, and is regulated by V_{thr}, such that the greater V_{thr}, the less charge gets pulled from the capacitor. The capacitor is then slowly

recharged through M_1 using the current I_τ, set by V_τ. Finally, V_{syn} is connected to a PMOS transistor which regulates the output current I_{out}, such that the lower V_{syn}, the greater the current.

Due to the exponential voltage-current-temperature relationship of transistors, as well as the uncertainty in the actual dimensions of the transistors (called mismatch), it is often problematic to set the parameter voltages directly. Instead, in order to make sure the various currents are set correctly, the voltages V_{thr}, V_w, and V_τ are set using current mirrors, the structure shown in the insert of Figure 7. By connecting a current source I_{in} with the desired current to the drain terminal of a diode-connected (gate connected to drain) transistor, the gate voltage is forced to the point where said current can flow through the transistor. If this gate voltage is then connected to the gate of a similar transistor, the same current will flow.

From [20] we find that the behaviour of the DPI synapse is indeed closely related to the behaviour described in equation 3, but with I_{thr} replaced by I_{gain}:

$$\tau \frac{dI_{\text{syn}}}{dt} = \frac{I_{\text{gain}}}{I_\tau} I_{\text{in}} - I_{\text{syn}} \tag{4}$$

The reason for this replacement is that I_{gain} does not directly represent a current in the circuit, but is instead a fictional p-type CMOS Field Effect Transistor (PMOS) current following

$$I_{\text{gain}} = I_0 e^{-\frac{\kappa(V_{\text{thr}} - V_{dd})}{U_T}} \tag{5}$$

where I_0 is the transistor dark current, κ the subtreshold slope, and U_T the thermal voltage. Solving these dynamics for a spike starting at t_i^-, when $I_{\text{syn}} = I_{\text{syn}}^-$ and ending at t_i^+, as $I_{\text{syn}} = I_{\text{syn}}^+$, gets us:

$$I_{\text{syn}}(t) = \begin{cases} \frac{I_{\text{gain}} I_w}{I_\tau} \left(1 - e^{-\frac{t - t_i^-}{\tau}}\right) + I_{\text{syn}}^- e^{-\frac{t - t_i^-}{\tau}} & \text{(charge phase)} \\ I_{\text{syn}}^+ e^{-\frac{t - t_i^+}{\tau}} & \text{(discharge phase)} \end{cases} \tag{6}$$

Where I_w is the current used to set V_w, such that in the ideal case, the current through M_4 and M_5 would be I_w when a spike is present, and 0 when no spike is present. If we now assume the synapse starts from a resting state ($I_{\text{syn}}^- = 0$) and the spikes have a constant pulse width $t_i^+ - t_i^- = t_w$, we can set $t_i^- = 0$ and integrate to find the total charge delivered per incoming spike:

$$\begin{aligned} Q &= \frac{I_{\text{gain}} I_w}{I_\tau} \left(\int_0^{t_w} 1 - e^{-\frac{t}{\tau}} dt + \left(1 - e^{-\frac{t_w}{\tau}}\right) e^{\frac{t_w}{\tau}} \int_{t_w}^{\infty} e^{-\frac{t}{\tau}} dt \right) \\ &= \frac{I_{\text{gain}} I_w}{I_\tau} \left(t_w + \tau \left(e^{-\frac{t_w}{\tau}} - 1 \right) + \tau \left(1 - e^{-\frac{t_w}{\tau}} \right) \right) \\ &= \frac{I_{\text{gain}} I_w}{I_\tau} t_w \end{aligned} \tag{7}$$

2.3.3 The DPI neuron

The neuron used as CCO and the TDE neuron in the sPLL is also based on the DPI, and can be seen in Figure 8. This neuron is a simplified version of the one presented in [21], with the spike rate adaptation part removed. In the neuron, the input current to the DPI is a sum of I_{in} and I_{rest}, a current used to set the rest level of the voltage over C_{mem}, the membrane potential V_{mem}. The other current governing the membrane potential is the leak current, I_{lk}, set through M_{L3}, which is the neuron's equivalent to I_τ.

V_{mem} is connected to a starved (limited in current) inverter, M_{A2-3}, which flips once the membrane voltage is high enough. As it flips, the current through it, I_{mem}, increases, which, through a current

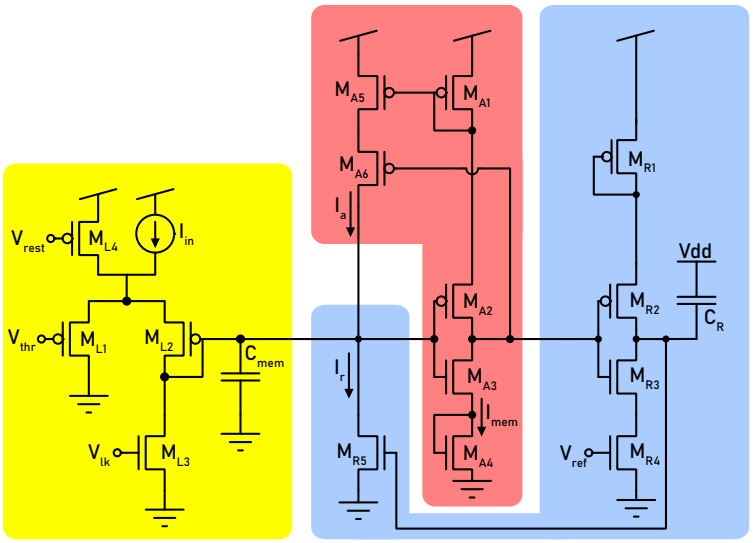

Figure 8: The DPI-based neuron used in the sPLL, adapted from [21]. **Yellow:** the DPI, now using PMOS transistors instead of NMOS like in the synapse. **Red:** the positive feedback circuit, activated once the voltage V_{mem} becomes great enough. **Blue:** the reset mechanism, responsible for returning the neuron to its rest state after the spike.

mirror $M_{A1\&5}$, drives a positive feedback loop, sending the amplification current I_a to further increase V_{mem}.

As the output of this first inverter goes low, the output of the second inverter, $M_{R2\&3}$, starts charging the capacitor C_R. The voltage on C_R causes M_{R5} to start conducting, which starts discharging C_{mem} with a current I_r, eventually resetting the neuron, but since the capacitor takes some time to charge, this negative feedback is slower than the positive feedback, creating the spike. After V_{mem} has discharged back to the point of the output of the first inverter going high again, C_r will be slowly discharged through M_{R4}, with a current set by V_{ref}. This creates a so-called refractory period, during which C_{mem} is continuously discharged through M_{R5}, making it essentially impossible for the neuron to spike.

Sadly, the DPI neuron is too complex to model analytically, but, based on [22], we can use some rough approximations to find a general relationship for the spike rate. We start with the differential equation describing I_{mem} (where I_{gain} is similar to what it was in the DPI synapse):

$$\tau \frac{d}{dt} I_{\text{mem}} \approx \frac{I_{\text{gain}} I_{\text{in}}}{I_{\text{lk}}} - I_{\text{mem}} + f(I_{\text{mem}}) \tag{8}$$

In this equation, the term $f(I_{\text{mem}})$ relates to the positive feedback, and relates to a term $e^{I_{\text{mem}} - I_{\text{th}}}$, where I_{th} is related to the inverters switching point. This allows us to instead think of the circuit as a LIF neuron, which spikes as I_{mem} reaches I_{th}. Solving the ODE without the $f(I_{\text{mem}})$ term gets us simply:

$$I_{\text{mem}} = \frac{I_{\text{gain}} I_{\text{in}}}{I_{\text{lk}}} \left(1 - e^{-\frac{t}{\tau}}\right) + I_0 \tag{9}$$

Which means that, for $I_0 = 0$, the time to reach $I_{\text{mem}} = I_{\text{th}}$ follows:

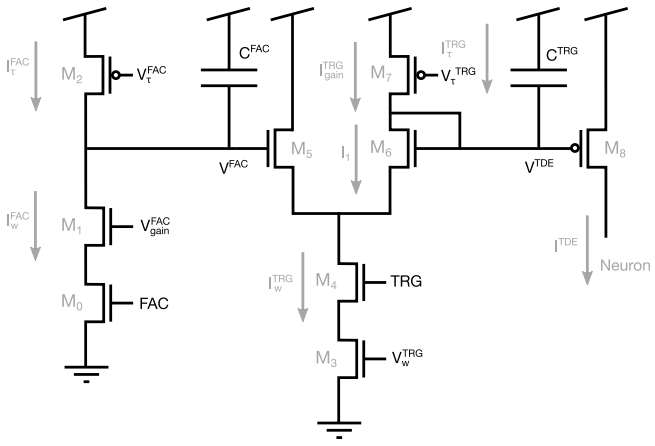

Figure 9: The schematic of the TDE synapse, as developed by Milde et al. [14]. Figure adapted with permission from [23].

$$t_{\text{th}} = -\tau \ln\left(1 - \frac{I_{\text{th}} I_{\text{lk}}}{I_{\text{gain}} I_{\text{in}}}\right) \tag{10}$$

and so, assuming that the neuron spikes immediately when I_{th} is reached, we can find the frequency and simplify using a first order Taylor expansion near ln(1):

$$f = -\frac{1}{\tau \ln\left(1 - \frac{I_{\text{th}} I_{\text{lk}}}{I_{\text{gain}} I_{\text{in}}}\right)} \approx \frac{I_{\text{gain}} I_{\text{in}}}{I_{\text{th}} I_{\text{lk}}} \qquad \frac{I_{\text{th}} I_{\text{lk}}}{I_{\text{gain}} I_{\text{in}}} \ll 1 \tag{11}$$

2.3.4 The TDE Synapse

Finally, as can be seen from Figure 9, the synapse of the TDE is essentially an extended DPI synapse, where V_{thr}, instead of being set as a parameter, is set by another trace, V_{Fac}. V_{Fac} is normally high, meaning that I_{gain} (see eq. 5) is normally low, such that the arrival of a spike does not change I_{syn} (see eq: 4 and 6). If a spike arrives at the Facilitatory (Fac) input however, it creates a current set by the gain I_w^{Fac}, which lowers the capacitor charge, and thus V_{Fac}. After the spike, the capacitor resets linearly through I_τ^{Fac}. This gives us (keeping in mind that $0 < V_C < V_{\text{dd}}$ and ignoring nonlinearities):

$$V_C(t) = V_{\text{dd}} - V_{\text{thr}}(t) = \begin{cases} \frac{1}{C}(I_w^{\text{Fac}} - I_\tau^{\text{Fac}})t & \text{(charge phase)} \\ \frac{1}{C}(I_w^{\text{Fac}} t_w^f - I_\tau^{\text{Fac}} t) & \text{(discharge phase)} \end{cases} \tag{12}$$

where t is the time since the start of the facilitatory spike and t_w^f its pulse width. Assuming the trigger spike arrives during the discharge phase and substituting into eq. 5 gives:

$$I_{\text{gain}} = I_0 e^{\frac{\kappa V_C}{U_T}} = I_0 e^{\frac{\kappa(I_w^{\text{Fac}} t_w^f - I_\tau^{\text{Fac}} t)}{C U_T}} = \alpha_f e^{-\frac{\kappa(I_\tau^{\text{Fac}} t)}{C U_T}} = \alpha_f e^{-\frac{t}{\tau_f}} \tag{13}$$

Where $\alpha_f = I_0 e^{\frac{\kappa(I_w^{\text{Fac}} t_w^f)}{C U_T}}$ and $\tau_f = \frac{C U_T}{\kappa I_\tau^{\text{Fac}}}$ are the facilitatory scaling and time constants respectively. Filling this result in in equation 6, with t_δ the time difference between the facilitatory and the trigger spikes, and once again assuming $I_{\text{syn}}^- = 0$, $t_i^- = 0$ and t_w^t the trigger pulse width, we get:

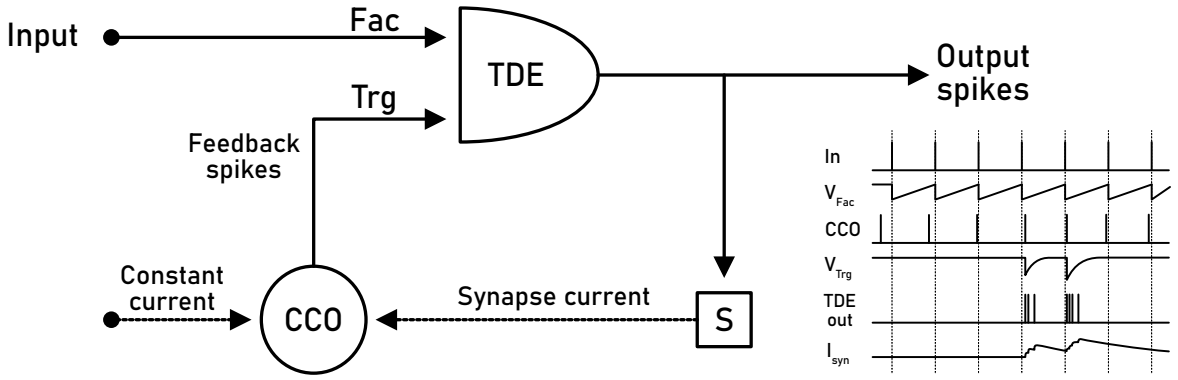

Figure 10: The sPLL, now with the internal traces shown in the inset

$$I_{\text{tde}}(t) = \begin{cases} \frac{\alpha_f e^{-\frac{t_\delta}{\tau_f}} I_w^{\text{Trg}}}{I_\tau^{\text{Trg}}} \left(1 - e^{-\frac{t}{\tau_t}}\right) & \text{(charge phase)} \\ \frac{\alpha_f e^{-\frac{t_\delta}{\tau_f}} I_w^{\text{Trg}}}{I_\tau^{\text{Trg}}} \left(e^{\frac{t_w^t}{\tau_t}} - 1\right) e^{-\frac{t}{\tau_t}} & \text{(discharge phase)} \end{cases} \tag{14}$$

We now recall that this synapse is connected to a DPI neuron, and so since I_{tde} scales with $e^{-\frac{t_\delta}{\tau_f}}$, we can use equation 11 to find (in strong approximation):

$$f_{\text{TDE}}(t) \sim e^{-\frac{t_\delta}{\tau_f}} e^{-\frac{t}{\tau_t}} \tag{15}$$

and so the total spike count scales with:

$$N_{\text{TDE}} \sim \int_0^\infty e^{-\frac{t_\delta}{\tau_f}} e^{-\frac{t}{\tau_t}} dt = \tau_t e^{-\frac{t_\delta}{\tau_f}} \tag{16}$$

2.3.5 Returning to the full sPLL model

Now we have explored the various subcircuits, we can return to the full sPLL model. In Figure 10 the sPLL is shown again, but now with the internal traces drawn for a small number of spikes. We see here how the CCO frequency is initially lower than the external frequency, until the Trg spike arrives after the Fac spike, causing the TDE to spike. This then causes I_{syn} to increase, increasing the frequency of the CCO, until $t_{\text{Trg}} < t_{\text{Fac}}$ again. As the time difference does not continue increasing, and instead the frequency of the CCO is increased to match the external frequency, we call this state *locking*.

As we have seen in the previous subsections, each subcircuit has a number of parameters governing its behaviour, and so the global circuit behaviour is also dependent on these parameters. It is hard to predict the global effect of each parameter, but some of the local, approximate effects can be listed. To give an overview, all parameters and their observed effects are listed in Table 1.

2.3.6 Potential well model

While we can mathematically describe (in approximation) the behaviour of the subcircuits of the sPLL, we do not have a good description of the behaviour of the full model. In order to still be able to make predictions on the behaviour of the system, we introduce a qualitative model: the potential well model. The basic idea behind this model is that the phase difference between the internal and external oscillators can be seen as a massive 'phase particle', which gets moved around by various

Parameter	On chip	Effect when increased
TDE synapse		
I_τ^{Fac}	Fac_tau_PI	increase in TDE time selectivity
I_w^{Fac}	Fac_w_NI	decrease in TDE time selectivity
I_τ^{Trg}	Trg_tau_PI	linear decrease TDE spikes
I_w^{Trg}	Trg_w_NI	exponential increase TDE spikes
t_w^f	external	exponential increase TDE spikes
t_w^t	pw_NI	exponential increase TDE spikes
TDE neuron		
I_{lk}	leak_tde_NI	linear decrease TDE spikes
I_{gain}	thr_tde_NI	linear increase TDE spikes
I_{rest}	rest_tde_PI	linear increase TDE spikes (adds to I_{in})
I_{refr}	refr_tde_NI	increase minimum time between TDE spikes
Osc. Synapse		
I_τ	tau_syn_PI	linear decrease charge to neuron; decrease timescale
I_w	w_syn_NI	linear increase charge to neuron
I_{gain}	thr_syn_PI	linear increase charge to neuron
t_w	pw_NI	linear increase charge to neuron
Osc. Neuron		
I_{lk}	leak_osc_NI	linear decrease osc. frequency
I_{gain}	thr_osc_NI	linear increase osc. frequency
I_{rest}	input_osc_PI	linear increase osc. frequency
I_{refr}	refr_osc_NI	increase minimum time between osc. spikes

Table 1: All circuit parameters, along with the names used in simulation and on the chip, as well as their approximate effects, as gathered from the equations above. It should be noted that the interplay between a lot of these parameters is a lot more subtle than is shown in this table.

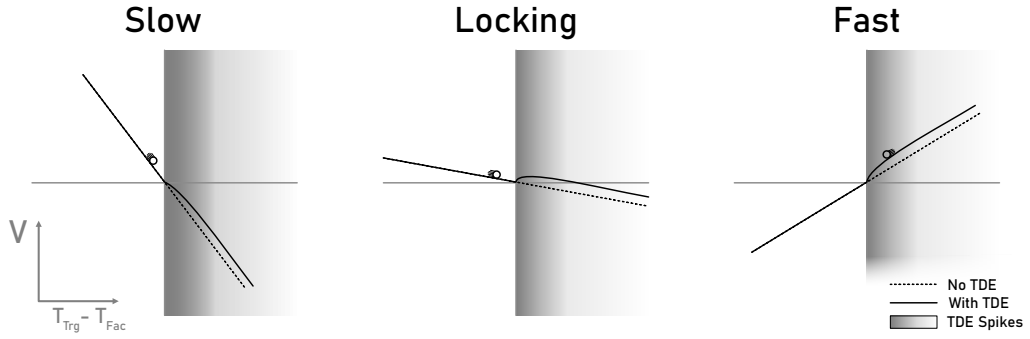

Figure 11: The three scenarios in the potential well model: internal oscillator too slow, locking, and internal oscillator too fast. On the x-axis of the plots is the time difference $\phi = t_{\mathrm{in}} - t_{\mathrm{ext}}$, taken periodically with respect to the period of the external spike, so that if $\phi > 0$, $t_{\mathrm{Trg}} > t_{\mathrm{Fac}}$, and the TDE can spike. The y-axis represents the 'potential'. The dashed lines represent the behaviour created by the 'pure' free-running frequency difference, whereas the solid line has the TDE behaviour added. The amount of TDE spikes is shown as the gray shading, where darker shading represents more spikes. For ease of visualization, the imaginary 'phase particles' are also drawn. Note that, since this is phase space, and the behaviour of the system is periodic, the particles can be thought of as 'wrapping around'; after running off one side of the plot, they are reset on the potential surface on the other side.

fictional forces, given by the circuit dynamics. If we then integrate the conservative forces over phase space, we can find a potential landscape, which can allow us to visualize the behaviour of the circuit. The first 'force' we model is that of the frequency difference between the internal and external oscillators, as such a frequency difference will continuously drive the phase difference to change. We are tempted to write down the equation of motion like this:

$$m\frac{d^2\phi}{dt^2} = F_{\text{freq}} \tag{17}$$

where $\phi = t_{\text{in}} - t_{\text{ext}}$. The problem with this however is that this would cause the rate of change of the phase difference to go to infinity. Instead we introduce a friction term, dependent on $\frac{d\phi}{dt}$, to create an equilibrium point:

$$m\frac{d^2\phi}{dt^2} = F_{\text{freq}} - C_f\frac{d\phi}{dt} \tag{18}$$

$$m\frac{d^2\phi}{dt^2} = 0 \Rightarrow \frac{d\phi}{dt} = \frac{1}{C_f}F_{\text{freq}} \tag{19}$$

such that the rate of change of the phase difference scales linearly with the frequency difference 'force', so that $F_{\text{freq}} \sim -\alpha\Delta f = -\alpha(f_{\text{in}} - f_{\text{ext}})$, where α is some constant of proportionality. The minus sign here appears due to the fact that if, for instance, $f_{\text{in}} > f_{\text{ext}}$, the internal spike would come earlier than the corresponding external spike, and so ϕ would decrease, whereas a positive F_{freq} causes ϕ to increase. The potential landscape connected to these first forces is found by integrating the conservative force, so F_{freq}, w.r.t. phase:

$$V = -\int F_{\text{freq}}d\phi = -F_{\text{freq}}\phi = \alpha\Delta f\phi \tag{20}$$

which is a straight line, as shown in Figure 11, dotted lines. In the figure, only one cycle is shown, but as we are dealing with oscillations, the phase space is cyclic, and so this can be extended to all of space.

If we now want to add the effect of the TDE we can combine equations 6, 11, and 16 to find that the frequency of the oscillator increases with $e^{-\frac{\phi}{\tau_f}}$, giving us an extra force

$$F_{\text{tde}} = \begin{cases} 0 & \phi < 0 \\ -\beta e^{-\frac{t_\delta}{\tau_f}} & \phi > 0 \end{cases} \tag{21}$$

where β is some positive constant of proportionality. This gives us the equation of motion:

$$m\frac{d^2\phi}{dt^2} = F_{\text{freq}} - C_f\frac{d\phi}{dt} + F_{\text{tde}} = \begin{cases} -\alpha\Delta f - C_f\frac{d\phi}{dt} & \phi < 0 \\ -\alpha\Delta f - C_f\frac{d\phi}{dt} - \beta e^{-\frac{t_\delta}{\tau_f}} & \phi > 0 \end{cases} \tag{22}$$

and the potential:

$$V = -\int F_{\text{freq}} + F_{\text{tde}}\ d\phi = \begin{cases} \alpha\Delta f\phi & \phi < 0 \\ \alpha\Delta f\phi - \beta\tau_f e^{-\frac{t_\delta}{\tau_f}} + \beta\tau_f & \phi > 0 \end{cases} \tag{23}$$

where the final $\beta\tau_f$ is added as a constant of integration to prevent a discontinuity at $\phi = 0$. This potential can be seen as the solid line in Figure 11. It should be noted that this integral is also equal to:

$$V = -\int F_{\text{freq}} + F_{\text{tde}}d\phi = -\int m\frac{d^2\phi}{dt^2} + C_f\frac{d\phi}{dt}d\phi \tag{24}$$

which is useful since we can directly measure ϕ, and from that find its derivatives. m and C_f are constants which, since this is a qualitative model, can be assumed to be equal to 1, allowing us to test

the model from data.

Having now described the potentials, we can start using them to make some predictions of the sPLL behaviour. To do so, we imagine again the phase particle, now 'rolling down' the slopes of the potentials. Looking at Figure 11, we can split the behaviour up into three cases. The first case is where the CCO is too slow, and so the slope $\nabla V = -F_{\text{freq}} - F_{\text{tde}}$ is always less than 0. This means that, while the phase particle might temporarily 'slow down' when it reaches the spiking regime of the TDE due to the friction term, it will never truly stop and lock (see Section 2.3.5). In case this slowdown is small, the amount of spikes we expect to see from the TDE is close to that of a 'free-running' system; it will spend an equal amount of time in all portions of the graph. If the slowdown is large enough however, the system will spend a larger than usual amount of time in the region just beyond $\phi = 0$, where the TDE spikes a lot, meaning we expect to see more total spikes than the 'free running' spike count.

On the other hand, if Δf is positive, so the internal oscillator is faster than the external one, we cannot see any locking, as the TDE spikes only serve to increase the oscillator frequency. Indeed, if we look on the right side of Figure 11, we see that if this is the case, the only effect for a particle rolling down the slope would be a slight increase in speed as it passes through the TDE spiking regime. This also means that we expect to see total TDE spike counts slightly less than or approximately equal to the 'free-running' counts, as the speed-up causes the particle to spend less time in the spiking regime.

Finally, looking at the center of Figure 11, we see that if Δf is negative, but close enough to 0, a local minimum appears at $\phi = 0$. This means a stable state exists here, and thus this state allows for locking behaviour. In order to now analyze the expected output of such a state, we need to keep in mind that our spikes are discrete in time and count, and that the TDE spike count is maximum at $\phi = 0$. What this means is that instead of a perfectly stable point existing at $\phi = 0$, the oscillator will spike at, say, $\phi = 0.01$, and find itself higher up on the potential curve. It will then 'bounce' back, until it has slowed down enough to start moving back towards $\phi = 0$. These bounces move further and further away from $\phi = 0$ as the $\Delta f \to 0$, and so the closer the frequencies, the more time the system will spend in the non- or low-spiking regime of the TDE, meaning we expect to see less total TDE spikes than in a 'free running' state.

Through this 'bouncing' behaviour, as a smaller frequency difference yields further bounces, we also expect to see a relationship between the frequency difference and the average ϕ of the locked sPLLs, similar to the regular PLL (see equation 1), but with a large amount of variance around this average. Another result of this behaviour is that during the 'bounce', we get relatively little information from the system, and so if these are very long, for instance when τ_{syn} is very long, it could increase the time before an accurate judgement of locking can be made. Besides this, if, for instance, a mixed frequency signal is presented, where another facilitatory spike might be close to the one the system is 'bouncing' off, a long bounce can cause the phase particle to interact with this other spike, breaking the lock.

2.3.7 The inhibitory sPLL (iPLL)

We theorize that the issues with the 'bouncing' behaviour mentioned in the previous section can be solved through the change to an inhibitory sPLL (iPLL), where the excitatory synapse is replaced by an inhibitory one. This would allow only sPLLs with fast free-running frequencies to lock, as the spikes from the TDE now slow the CCO down instead of speeding it up. Looking at Figure 12, where a potential landscape for this setup is drawn, we can see that this allows for a more gradual minimum, the position of which depends directly on the free-running frequency difference Δf. What we can thus expect is a system which does not exhibit this bouncing behaviour, but instead locks stably. We also expect to see a minimum in the amount of TDE spikes when Δf is very small, which then gradually increases to a maximum as the increasing Δf forces the locking point closer and closer to $\phi = 0$

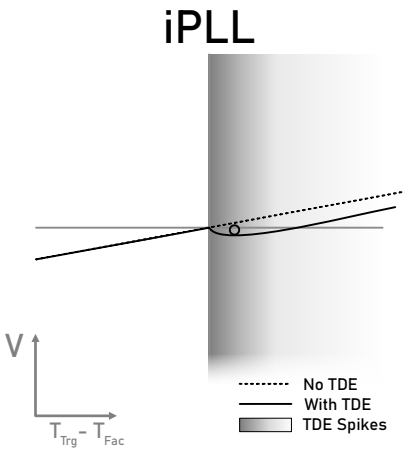

Figure 12: The theorized potential landscape of a locked 'inverted' sPLL, with an inhibitory instead of excitatory synapse. Note how in this case the well is gradual, allowing the minimum to depend on the original slope.

3 Methods

3.1 Overview

Although the locking behaviour of the sPLL has been shown in [6], a real-world circuit implementation had yet to be tested. In order to do this, a Very Large Scale Integration (VLSI) layout of the circuit was designed by the BICS group on the X-FAB 180 nm node using Cadence® Virtuoso®. This was done as part of a larger chip, Cognigr1, with a number of other test circuits on it (see Figure 13). While the actual circuit design is not part of this thesis, the testing of both the designed schematic and the finalized physical circuit is.

As a reminder, the questions we set out to answer were whether the sPLL can detect frequencies, and whether it can act as a DFT. We also set out to develop and test a qualitative model, the Potential Well Model, as described in section 2.3.6, and to test a possible improved sPLL, the inhibitory sPLL (iPLL) (section 2.3.7). To do these tests, three different platforms were used. The behaviour of the sPLL was tested both in Cadence® ADE Explorer®, the simulation component of Cadence® Virtuoso®, as well as directly on the finished chip, whereas the iPLL was simulated using PyTorch. The potential well model was tested using data from all three. These platforms are described in more detail in section 3.2, whereas the specifics of the tests, as well as the data processing, are described in section 3.3. In general however, the tests performed involved parameter sweeps of the internal CCO constant current (and thus frequency) versus the external input frequency or frequencies, and observing the system behaviour for each combination.

3.2 Setups

3.2.1 Cadence® simulation

To help speed up testing, all tests performed on the chip were first performed in simulation using Cadence® ADE Explorer®. This tests the circuit behaviour in idealized conditions, without the potential noise coming from the physical layout in silicon. Most circuit parameters, except the CCO constant current, were kept constant at the values listed in Table 2, which were obtained from M. Mastella, who tuned these during the chip design process. The simulations were ran for 1 second of internal time each, and outputs such as the TDE spike rate or the time difference ϕ were preprocessed using the internal calculator of Cadence®, so that less data needed to be stored. The parameter

Figure 13: A closeup of the Cognigr1 silicon

currents were set directly using ideal current sources connected to current mirrors, as shown in the inset of Figure 7, with the transistor parameters set such that the currents were mirrored 1-to-1.

For the single-frequency measurements, the input spikes were generated using a pulse wave voltage source, with a pulse width of 100 μs, and a rise- and fall time of 1 μs. For the multi-frequency measurements, this was replaced by an arbitrary waveform voltage source, for which the waveform was generated using python.

3.2.2 Chip tests

Testing the chip was done using a custom circuit board, allowing us to send signals in and out of the chip. Communication with the chips internal circuitry was done using a Teensy® 4.1, running custom firmware developed by M. Mastella, H. Greatorex, O.J. Richter and B. Hucko of the BICS group before work on this thesis began. The Teensy® could be controlled from a PC using a set of python scripts, developed by the same team, allowing us to send control signals and spikes to the chip. The chip was powered using a Keysight® E36312A programmable DC power supply, and the output traces were measured using an Agilent Technologies® InfiniiVision™ DSO7054B oscilloscope. For most of the tests, the data from the oscilloscope was gathered using Rigol® BenchVue™ software, from which it was saved as `.mat` files, such that it could be loaded into python later. For the multi-frequency tests however, we developed some python code to interface with the oscilloscope directly. As the chip exhibited some strong temperature dependence, the temperature in the room was kept constant using air conditioning.

The parameters on the chip were set by sending it digital instructions using the Teensy®, which were then decoded using an internal First-In-First-Out structure (FIFO) and a Digital Analog Converter (DAC), turning the instructions into currents. These currents were then brought into the circuit using current mirrors, but unlike in the Cadence® simulations, the current mirrors for the n-type CMOS Field Effect Transistor (NMOS) transistors were 1-to-1, whereas the currents for the PMOS transistors followed a 3-to-1 ratio, meaning we expect the PMOS current parameters to be 3 times larger in the chip than in the simulation to reach the same effect. The actual parameter set used was obtained by starting off from the parameters used in the Cadence® simulations, and then hand-tuning based on the observed behaviour until locking was observed. The reason this tuning was necessary was due to a difference in the temperature at which the circuit was tested, some non-linear behaviour in the DAC, as well as process variation and mismatch in the transistors, meaning that the transistor dimensions might not be exactly as in the design. Finally, to make sure the system remained in its optimal operating regime during the frequency sweeps done, before each sweep, the parameters were checked for both the highest and the lowest observed internal and external oscillator frequencies, and adjusted

Name	Simulation	Chip single-frequency	Chip dual-frequency	Type
Fac τ	10p	40p	80p	P
Fac W	10n	2n	2.5n	N
Trg τ	20p	20p	40p	P
Trg W	50p	1n	50p	N
TDE refr	5n	250n	15n	N
TDE rest	1p	30p	0.1p	P
TDE thr	10p	50p	500p	N
Syn τ	5p	3p	15p	P
Syn thr	100p	100p	300p	P
Syn W	100p	300p	100p	N
Osc refr	1n	50n	1n	N
Osc leak	1p	50p	1p	N
Osc thr	100p	5n	100p	N
Temperature	27° C	20°	20°	–

Table 2: A subset of the parameters used in the simulations and measurements. The parameters are all currents, which are set using current mirrors. On the chip, the P-type (referring to P-channel MOSFETs) current mirrors have a 3-to-1 ratio, such that one would expect those currents to be 3 times higher on the chip than in Cadence®.

if necessary.

The Teensy® was also used to send spikes, which had an approximate pulse width of 100 μs, and were defined as a list of inter-spike intervals. Besides this, one pin was used to send a longer pulse to the external trigger input of the oscilloscope to signify that the input spikes were about to be sent, allowing us to synchronize the measurement with the input.

3.2.3 PyTorch simulations

Finally, the iPLL was tested by running python simulations, which were originally developed by M. Mastella using PyTorch, and were edited to fit the iPLL model. In these simulations, the various subsystems were represented using simple ODEs, similar to those described in sections 2.3.2, 2.3.3, and 2.3.4, but with the neuron modeled as a simple LIF neuron, meaning the positive feedback is replaced by a spiking condition $V_{\text{mem}} \geq V_{\text{th}}$. These differential equations were then solved numerically using a forward Euler method, where the different subsystems communicated at each time-step to each other whether a spike took place or not, such that the spike information can be processed during the next time-step. For each simulation, the time-step was set to 10^{-4} s, and the total internal run-time was set to 10 seconds. For each run, the total number of TDE spikes, the total number of CCO spikes, as well as an array containing the CCO spike timings, were saved. The parameters used in these simulations were once again obtained from M. Mastella, and, since the simulations are unrelated to any real-world system, their exact values are arbitrary.

3.3 Tests performed

3.3.1 Single frequency discrimination

In order to test single frequency detection, we could make use of the predictions outlined in section 2.3.6, by varying the internal and external oscillator frequencies, and recording the average TDE spike rate. Recalling now that the CCO frequency has an approximately linear relationship with its input current (equation 11), if the sPLL is capable of detecting frequencies, we should see a linear relationship between the input frequency and the CCO constant current for which the TDE spike rate is minimum.

In the Cadence® simulation, this was tested by sweeping the input period from 15 ms to 100 ms in 20 steps using a logarithmic step size, while linearly varying the CCO constant current from 0 to 10 nA in 50 steps. These ranges were chosen because this was where the circuit appeared most stable. After gathering the TDE frequencies from Cadence®, the data was loaded into python to be processed. During this process, a few data-points were lost due issues with the way Cadence® saves its data, causing the blank spots in Figure 14.

It should be noted that, since the TDE only spikes after it receives some input, if the input and CCO frequencies are higher, the TDE spike rate also tends to be higher. To help combat this from skewing the data too much, for the data from Cadence®, the spike rates were averaged over all input periods to create an 'average profile', which was then normalized and used to adjust the spike rates w.r.t. CCO frequency. The input periods were then converted to frequencies, and the adjusted data was plotted. Since the system failed to show any change in spike rate for the highest 5 frequencies, likely due to the TDE synapse being overwhelmed (see section 5.2), it was decided to leave the top 4 frequency rows off of the final plot to aid with visualization of the other input frequencies.

In the physical chip testing, it was found that the chip behaved well in the frequency range around 50 Hz, and so the sweep was performed with input frequencies ranging from 40 Hz to 60 Hz in 1 Hz increments, and the corresponding CCO constant current range, 20 to 28 nA, in a 20 step linear sweep. Since the output of the oscilloscope consisted simply a set of voltage arrays, the spikes needed to be found before any further processing could be done. As the spikes were pulses of a few samples wide, this was done by taking the sample-to-sample difference for each trace, and then finding the peaks of this difference array using `scipy.signal.find_peaks`. From this, the TDE spike count could simply be found by taking the length of the array of peak times. This data was then plotted, but unlike the results from Cadence®, we did not re-scale the data w.r.t. the average over all runs, as the range of CCO frequencies was not large enough to seriously skew the data, and instead the smaller number of datapoints might have caused this step to introduce unwanted artifacts.

3.3.2 Dual frequency discrimination

For testing whether the circuit can detect multiple frequencies at the same time, a similar method was employed. In this case, the external signal was a mixed signal, where the frequency of the first signal was kept at 25 Hz, and the frequency of the second signal was swept from 35 to 44 Hz in 1 Hz increments. The signals were created by first creating the separate signals, and then mixing them in-phase, meaning that the first spike for both signals occurred at $t = 0$.

For the Cadence® simulation, the parameters were kept the same as for the single-frequency tests, and the input current was swept from 0.5 nA to 5 nA in 100 steps. For the on-chip testing on the other hand, the chip had to be re-tuned, as the locking behaviour was not close enough (see section 4.2), and the parameters obtained from this tuning can be found in Table 2. Besides the parameters in Table 2, the pulse width of the spikes coming from the CCO was increased as well, to match the pulses in the Cadence® simulation. This was done by changing the `pw_NI` parameter, the input to the pulse-extender, which can be found in Table 3. Finally, on the chip, the currents were swept from 5 nA to 50 nA in 120 steps. The data processing for both methods was similar to that used for the single-frequency tests, but in this case, neither of the sets were re-scaled, as this would cause issues with detecting the constant 25 Hz term.

To get a better understanding of the behaviour of the sPLLs with a dual-frequency input, another analysis was done on the data, where the frequencies of the internal and external oscillators were compared. For the Cadence® simulation, the frequency of the internal oscillator was simply found using the built-in preprocessor, like with the TDE frequencies. For the data from the chip testing, we first found the CCO spikes using the same method as used for finding the TDE spikes. This then allowed us to compute the ISIs for each spike train, and from that find the average CCO frequency. These frequencies were then subtracted from the second (swept) input frequency, and the norm of this value was plotted.

3.3.3 Inhibitory sPLL

In order to be able to make a fair comparison, the torch simulations were run for both the ePLL and iPLL cases. Both were run with the same set of 50 input frequencies (50 to 99 Hz) and 40 CCO currents. The TDE spike output counts were then plotted for both to compare the behaviour. From these plots, one input frequency and a sub-range of input currents were chosen for closer analysis, where the frequencies were 80 Hz for the ePLL and 60 Hz for the iPLL. The reason for this difference was that the minimum frequency of the CCO in the simulation was found to be around 50 Hz, and since for the ePLL, only the 'slower' sPLLs can lock, a higher frequency needed to be chosen for that. In order to test whether the iPLL did indeed have a better relationship between the average time difference ϕ and the free-running frequency difference, the time difference was calculated by comparing the timings of the internal and external spikes, and then taking the modulo with respect to the input period to map it onto $\phi_{\text{ex}} \in (-\pi, \pi)$, where ϕ_{ext} is the phase of the external oscillator. This was then averaged, and the standard deviation was found in order to show the effect of the 'bouncing' behaviour discussed in section 2.3.6.

3.3.4 Potential well models and phase portraits

Finally, the potential well model was tested by analyzing some of the data gathered for the previous tests. By finding the time difference between the internal and external spikes, we can find ϕ. We can then find the difference between consecutive values of ϕ to find a measure of $\frac{d\phi}{dt}$, and repeat this to also find $\frac{d^2\phi}{dt^2}$. We then took the modulo of ϕ with respect to the input period to map it onto $\phi_{\text{ex}} \in (-\pi, \pi)$, like was done to compute the average phase difference for the iPLL, and sorted the two derivatives along with it. This then allowed us to perform the integral listed in equation 24 by using trapezoid integration (`scipy.integrate.cumulative_trapezoid`) to get approximate plots of the potential landscapes associated with the behaviour seen.

As no spiking information was available for the original data gathered with the Cadence® simulations, a new sweep was performed at an input period of 30 ms, or a frequency of 33.33 Hz. The time difference for this data was then calculated using the internal calculator of Cadence®. For the chip, the data used was from the single-frequency measurements, where a central frequency, close to 50 Hz, was chosen to get a good spread of behaviours. Finally, for the iPLL, the data used was that gathered for the the plots comparing the average time difference and the frequency difference.

While analyzing some of the data, we noticed some large discrepancies between the behaviours shown in the chip and the Cadence® simulations, and so to help discuss this, we plotted $d\phi$, the difference between consecutive measures of ϕ, against ϕ, as can be seen in Figure 16. For these figures, the 'maximally locking' samples in the data for the potential plots were used, which were found by finding the traces with the lowest amount of TDE spikes. The methods for then finding ϕ and $d\phi$ were the same as those described above for the potential plots.

4 Results

4.1 Single frequency discrimination

The results from the current-frequency sweep performed in Cadence® can be seen in Figure 14. Notable is the mostly linear relationship between the input frequency and the CCO constant current described by the locations of the TDE spike minima, as expected by the approximation found in equation 11. Besides this, another feature of interest is the locking of PLLs to frequencies corresponding to (sub)harmonics of their natural frequencies. Finally, it should also be noted that above an input frequency of 45 Hz, no PLLs locked, potentially due to the facilitatory trace of the TDE being depressed past its normal working range.

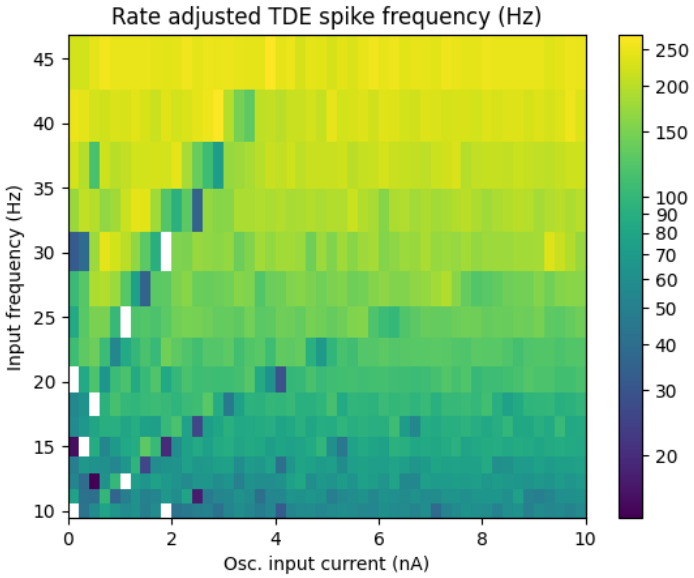

Figure 14: Results from the current-frequency sweep in Cadence®, the colors correspond to the TDE spike rates. The results are adjusted for internal oscillator frequency as discussed in section 3.3.1. The white spots correspond to runs that were not properly exported by Cadence®. As can be seen, for each input frequency, there are one or more sPLLs that have a locally minimal amount of TDE output spikes. What can also be seen are the various (sub)harmonics to which the sPLLs lock besides their 'main' frequency. Due to issues with the way Cadence® saves its data, some results were lost, causing the white pixels.

In Figure 15, similar results can be found, from the chip instead of the simulation. As can be seen, here too we see the linear relationship between the input current and the external frequency described by the TDE spike minima, showing that the TDE circuit is in fact capable of differentiating between frequencies. As the frequency range here was not large enough, no harmonics can be seen, given that we expect these at either twice or half the frequencies used. Also note the higher input currents needed to reach similar internal frequencies to the Cadence® simulations.

In both experiments it can be seen that the PLLs with a slightly lower input current, so a slightly lower internal frequency, have a higher TDE output frequency than the surrounding PLLs. This makes sense; as was discussed in section 2.3.6, we expect sPLLs that are slightly too slow to spend a larger amount of time in the high spiking regime of the TDE. It could also be that an sPLL that locks, but poorly, will spend a lot of time in the high spiking area of the TDE curve, more so than those that

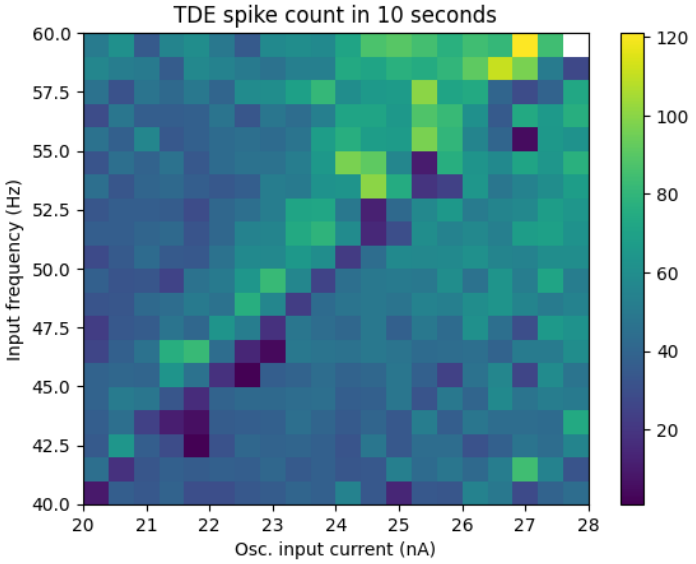

Figure 15: Results from the current-frequency sweep on the chip. The white spot in the top-right corner corresponds to a run that, due to the higher frequencies, saw its sPLL enter a 'run-away' state. The higher spike counts towards the top-left corner can be contributed to the fact that at the higher frequencies, the sPLL completes more 'cycles', as well as some potential 'integrating' behaviour of the TDE synapse, where the synapse currents do not fully reset to their resting level between spikes.

are simply running free. This does however create the issue that when attempting to detect frequency by looking for the lowest spike rate, poor locking is worse than no locking at all.

4.2 Phase behaviour

What the figures above fail to show however, is the large difference between the locking behaviours of the sPLLs on the chip and those in the simulations, due to the different parameter sets chosen. As can be seen from figure 16, in which phase portraits of two maximally locking PLLs are shown, the PLL in the simulation results spends almost all its time near 0 phase difference, and is slow before it and fast behind it, indicating that it likely alternates cycles being in the non-spiking regime of the TDE and the strong spiking regime of the TDE. This is due to its low synapse gain and high synapse I_τ, resulting in a tight, but high-output coupling.

The PLL on the chip on the other hand, was configured with a low synapse I_τ but a high synapse gain, along with low TDE gain settings, mainly to preserve stability in the TDE circuit. What this means is that for each TDE spike, the synapse releases a lot of current, and keeps doing so for a relatively long time, allowing the CCO to accumulate a lot of time difference before its frequency drops below the external one again. The result is that the sPLL spends a large amount of its time in the low-spike regions of the TDE, with the CCO sometimes being over half a cycle ahead of the input spikes. This creates a far less tight of a lock, making it less responsive to potential changes in frequency, but allows for sparsity in the output signal. In fact, returning to Figures 14 and 15, we can see that the total TDE spike count for most sPLLs is between 40 and 80 for the whole 10 second run, which, given an input frequency between 40 and 60 Hz, corresponds to having one spike per 10 input cycles on average.

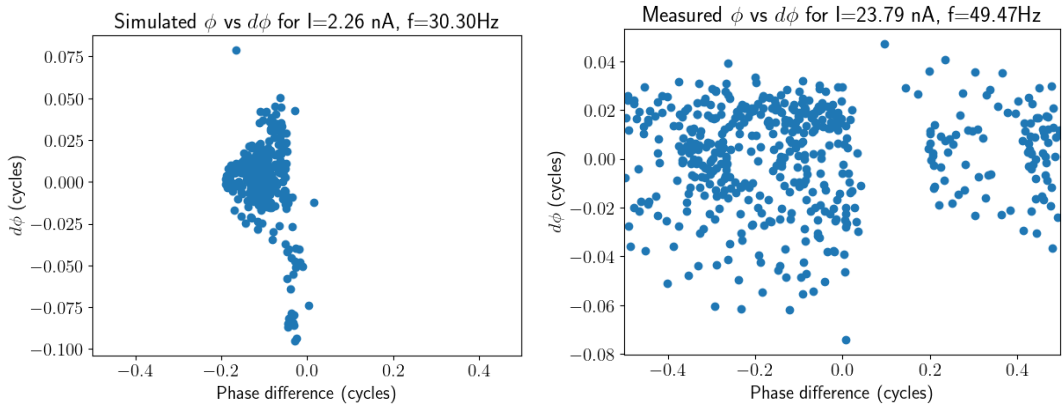

Figure 16: $d\phi$ vs ϕ plots for maximally locking states in both the Cadence® simulations and the chip measurements. Both were obtained from 10 seconds worth of activity. In both plots, the phase difference is defined as the timing of the internal oscillator spike w.r.t. the input spike, and so the TDE is maximally spiking in the region where ϕ is just above 0.

4.3 Dual frequency discrimination

To test the capability of the circuit to detect multiple frequencies at once, tests were done where the circuit was presented with a mixed-frequency signal, consisting of a constant 25 Hz signal and a signal that was swept from 35 to 44 Hz. The results of these experiments are shown in Figure 17. As can be seen, for both the Cadence® simulations and the chip measurements, the 25 Hz signal is visible as a vertical line of low TDE spike-counts, and the swept frequencies are visible as a diagonal line, similar to what was observed in the single-frequency experiments, showing that both frequencies were successfully detected. On both plots, there is also a faint set of low-spiking sPLLs visible on the left side of the 25 Hz line, showing the sub-harmonics of the swept frequencies.

A second analysis was done on the data from the dual-frequency experiments, where the frequency of the CCOs was compared to the frequencies of the second (swept) input signals. The results of this analysis can be found in Figure 18. As this figure shows, while the results from the Cadence® simulations show clear regions where the frequencies are equal, and thus the sPLLs are locked, the chip data does not show this, instead behaving closer to what would be expected from comparing 'free-running' CCOs to the inputs. Since we did see however from Figure 17 that we can detect both input frequencies in both cases, the question needs to be raised whether the frequency locking is important, or whether simply comparing the two signals with a TDE is sufficient.

4.4 Inhibitory sPLL

The results of the python simulations ran to test the iPLL can be found in Figures 19 and 20. Turning our attention first to Figure 19, we can see that, in simulation, both the ePLL and the iPLL show linear relationships between the CCO constant current and the input frequency. This means that, here too, the TDE output spikes can be used to detect frequencies. Looking closer, we can see that, as expected from the potential well model (section 2.3.7), we see that the behaviour of the iPLL starts with a dip in TDE spikes for the lowest frequency locking sPLL, after which the spike counts climb to a maximum for the highest frequency locking sPLL.

Another thing to note in Figure 19 is the fact that the 'strongest locking' states for the iPLL have higher CCO currents than their ePLL counterparts. This is likely due to the fact that the TDEs in these models were configured such that there is almost always some amount of TDE spikes, meaning that the minimum current the CCO sees throughout the cycle is not the pure constant input current. Instead, it is a combination between the constant current and a 'background' synapse current, which

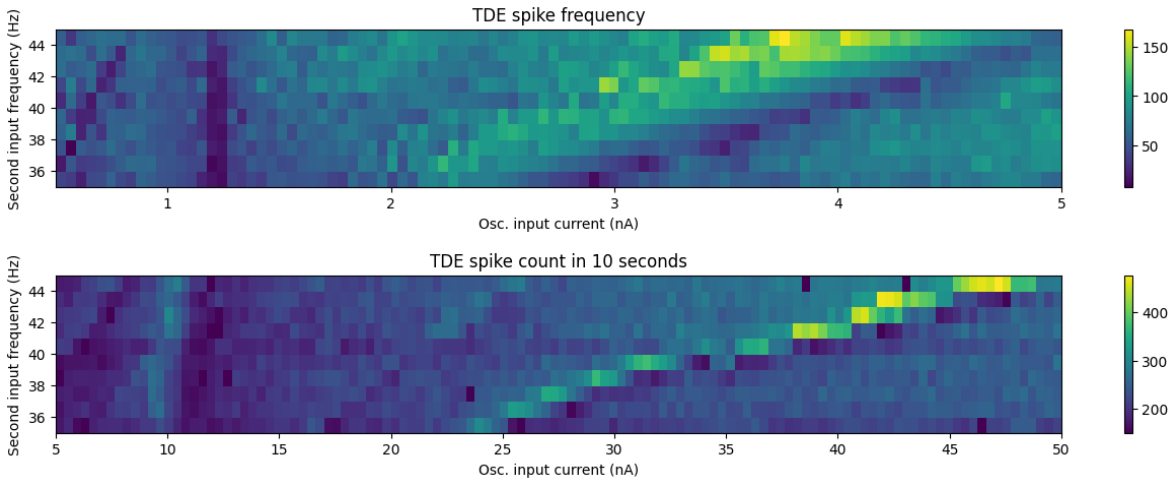

Figure 17: Results of the dual-frequency experiment for both the Cadence® simulations (**above**) and chip measurements (**below**). The colors represent TDE spike rates. In both cases, the input was a mixed signal consisting of a constant 25 Hz part and a part that was swept from 35 to 44 Hz (y-axis). Note that both frequencies are clearly visible as minima in the TDE spike counts for both the Cadence® and the chip results.

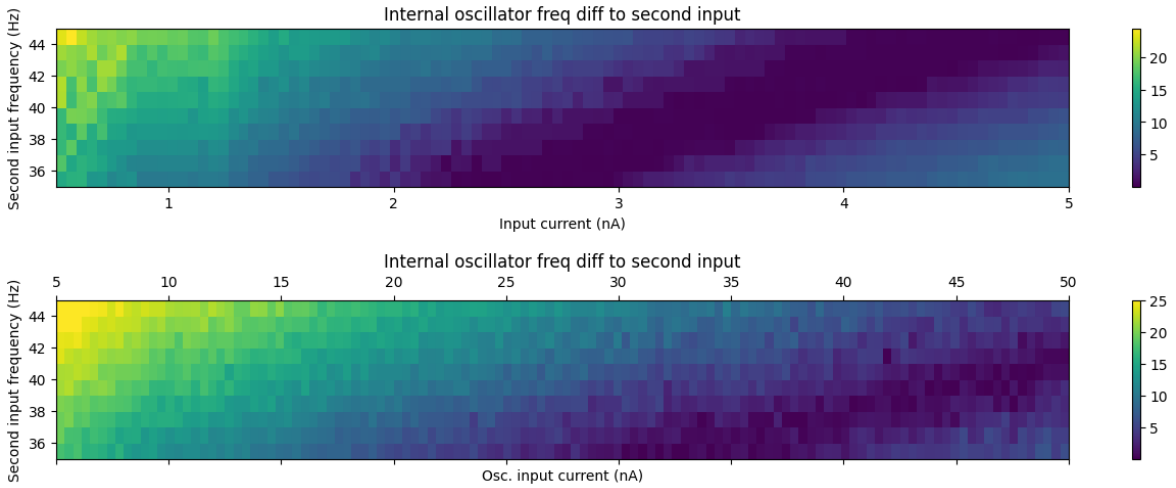

Figure 18: Frequency differences for the dual-frequency experiment for both the Cadence® simulations (**above**) and chip measurements (**below**). The colors represent the absolute value of the frequency difference between the CCO and the second (swept) external signal. As can be seen, while the simulated sPLLs show a broad range where the frequency difference is essentially 0, the chip data shows no such behaviour, instead following a pattern closer to what would be expected of a free-running CCO.

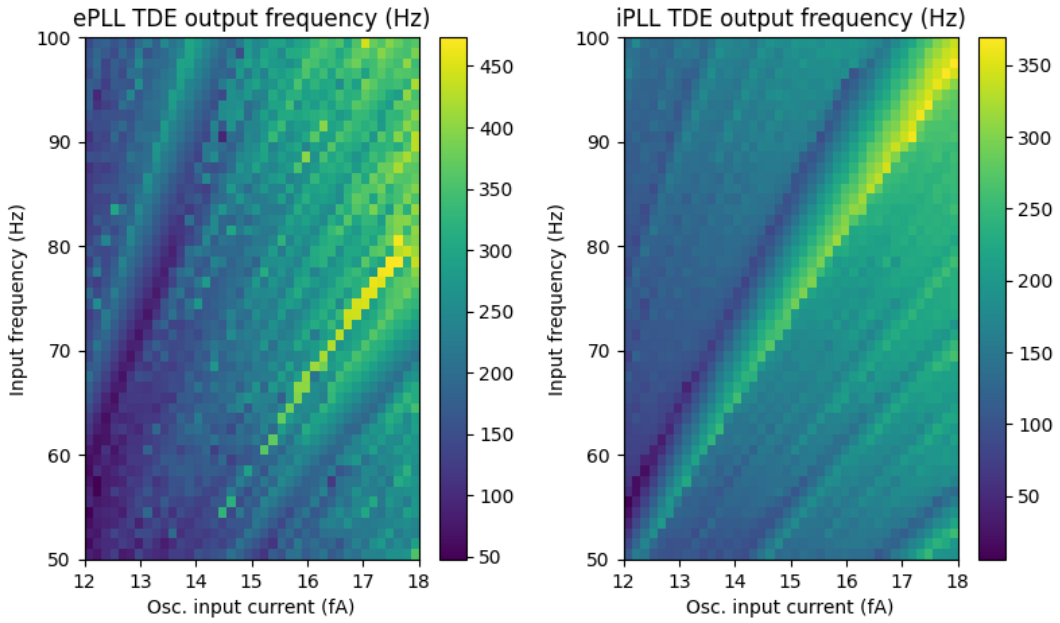

Figure 19: TDE spike rates from the python simulations of the ePLL (or 'regular' sPLL) (**left**) and the iPLL (**right**). As can be seen, both the ePLL and iPLL are capable of detecting frequencies. Note however the disagreement between them on the 'strongest locking' states, which have higher constant currents in the iPLL than in the ePLL.

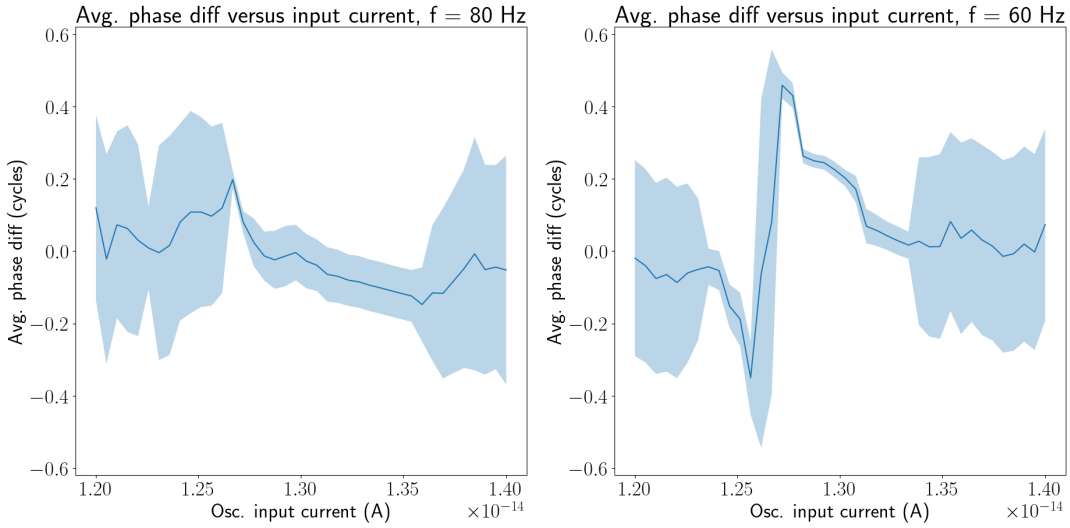

Figure 20: Average phase differences between the CCO and the external oscillator for both the simulated ePLL (**left**) and the iPLL (**right**). The shaded area represent the standard deviations. The phase difference is expressed w.r.t. the external oscillator, in number of external oscillator cycles, mapped onto ± 0.5 cycles. This also means that while it looks from the plot that the average iPLL phase crosses through 0 around $1.27 \cdot 10^{-14}$ A, it actually 'wraps around' from -0.5 to $+0.5$ cycles.

is negative in the case of the iPLL, making it so that a higher constant current is required to reach the same minimum frequency.

Finally, we note that the TDE spike rates for the ePLL are far more 'noisy' than those of the iPLL. This might be due to the fact that, in the ePLL, as the CCO spikes more, the TDE has more chances to spike, which in turn speeds up the CCO more, creating a postivie feedback loop. In the iPLL on the other hand, since the TDE spikes cause the CCO to spike less, the behaviour is closer to a negative feedback loop, which might be what 'smoothes out' the behaviour.

Moving on now to Figure 20, we can see that not only does the iPLL show a stronger relationship between the average phase and the frequency difference, the standard deviation of the phase is far lower than that of the ePLL, which agrees with the expectations set out in section 2.3.7. It should be noted here that this extremely strong phase-frequency relationship is made possible in this case by the fact that the TDE was active practically across the range of time-differences, whereas a shorter Facilitatory time constant τ_f would create a weaker relationship.

4.5 Potential well model

Finally, while we have tested a number of the predictions made using the potential well model, we have yet to directly test whether the actual potential landscapes can be found. By performing the integral described in equation 24, and plotting the results for multiple CCO currents, and thus multiple frequency differences, we can get an idea of whether or not these potential landscapes show up in the sPLL data.

These integrals can be found in Figures 21, 22, and 23 for the chip measurements, the Cadence® simulations, and the iPLL simulations respectively. Starting with Figure 21, the chip measurements, we see that the data seems to agree quite well with the theory. As the frequency difference decreases, the slope becomes less and less steep, and although the potential bump created by the TDE is rather small, it does eventually appear, and is even visible in some of the sPLLs with an internal frequency that is too high.

As for the potential landscapes from the Cadence® simulations, found in Figure 22, instead of forming a 'bump' on the positive side of $\phi = 0$, there appears to be a potential well on the negative side. While counter-intuitive at first, this is caused by a combination of long TDE time constants, high TDE spike rate, and a low synapse time constant. This makes it so that the TDE is spiking throughout most of the phase space, except in the region just below $\phi = 0$, such that the internal oscillators drop in frequency as soon as they reach that space, causing the sharp drop in the integral.

Finally, for the data from the iPLL simulations, found in Figure 23, we again see potential landscapes similar to what we expected. While not fully visible in the locking cases, as those simply followed a transient towards the phase difference they locked at, in both the slow and the fast iPLLs, the valley above $\phi = 0$ is clearly visible. We also see how the slow iPLLs fail to lock as they are only further slowed down, and at the other end, the iPLLs that were too fast do not see the slopes become positive and so fail to form a well. Finally, the potential plots of the locking iPLLs show very clearly how the state migrates to a locking point and stays there, showing far less variation than the sPLLs shown in the previous two figures.

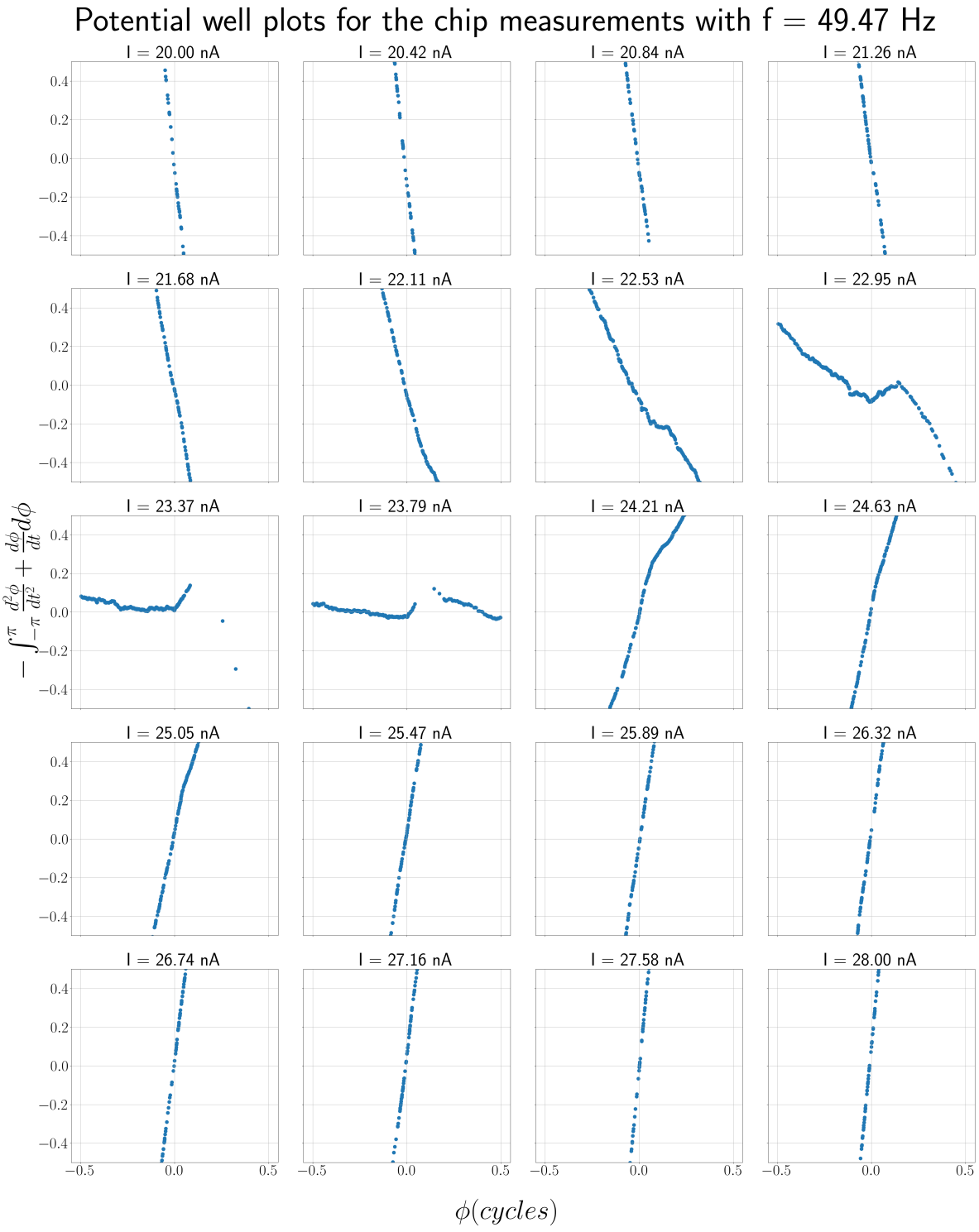

Figure 21: Potential landscapes calculated from the chip measurements. The integrals were shifted be 0 on average, and then limited to ± 0.5 for clarity. The currents I in the titles refer to the constant input current to the internal oscillator. As in figure 16, the strongest locking occurred at $I = 23.97$ nA.

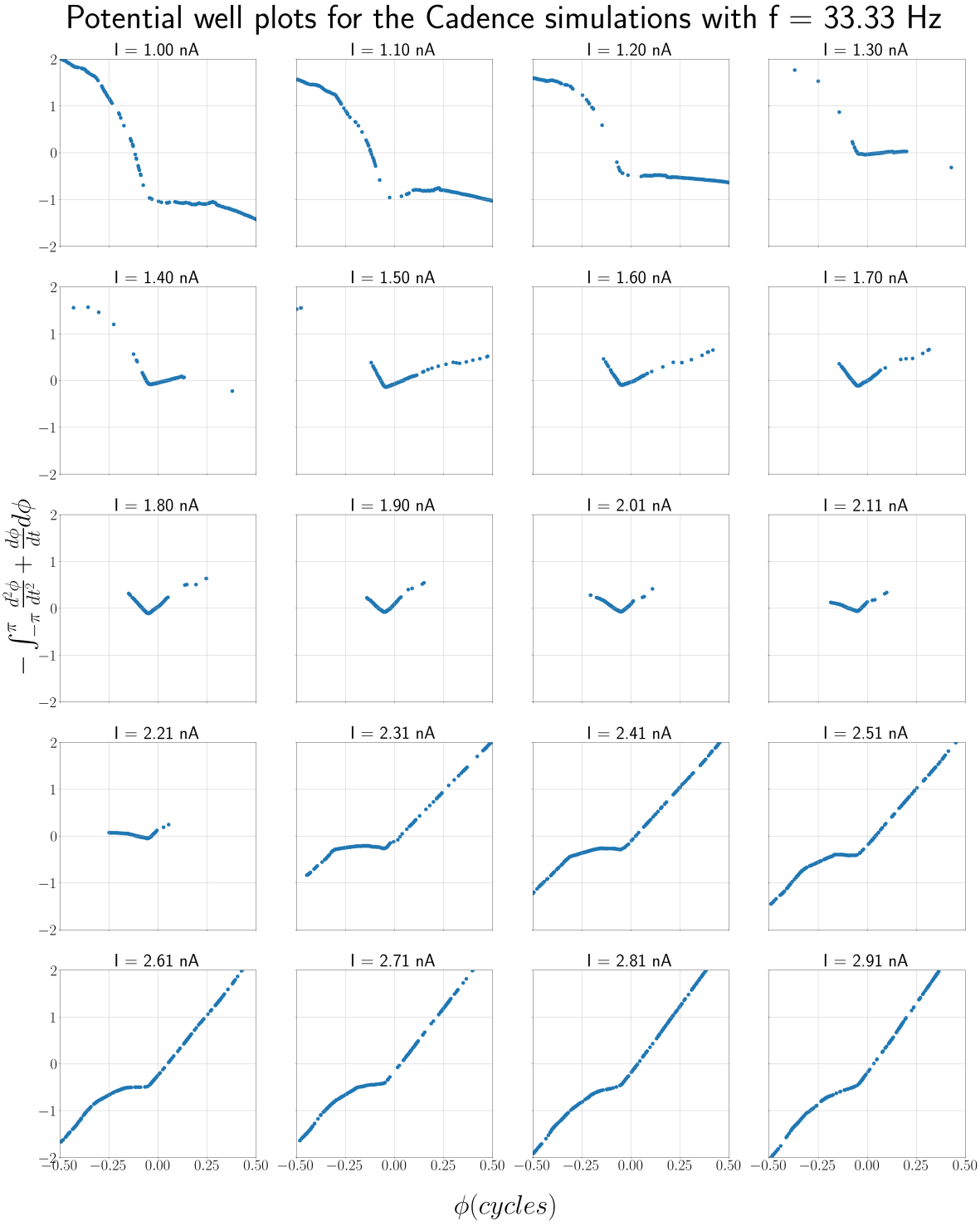

Figure 22: Potential landscapes calculated from the Cadence® simulations. The integrals were shifted to be 0 on average. The currents I in the titles refer to the constant input current to the internal oscillator. As in Figure 16, the strongest locking occurred at $I = 2.26$ nA (not included in this plot). Due to the rather 'tight' locking of the simulations, the sPLL never left the area around $\phi = 0$ in the locking PLLs, causing the lack of of data outside that region.

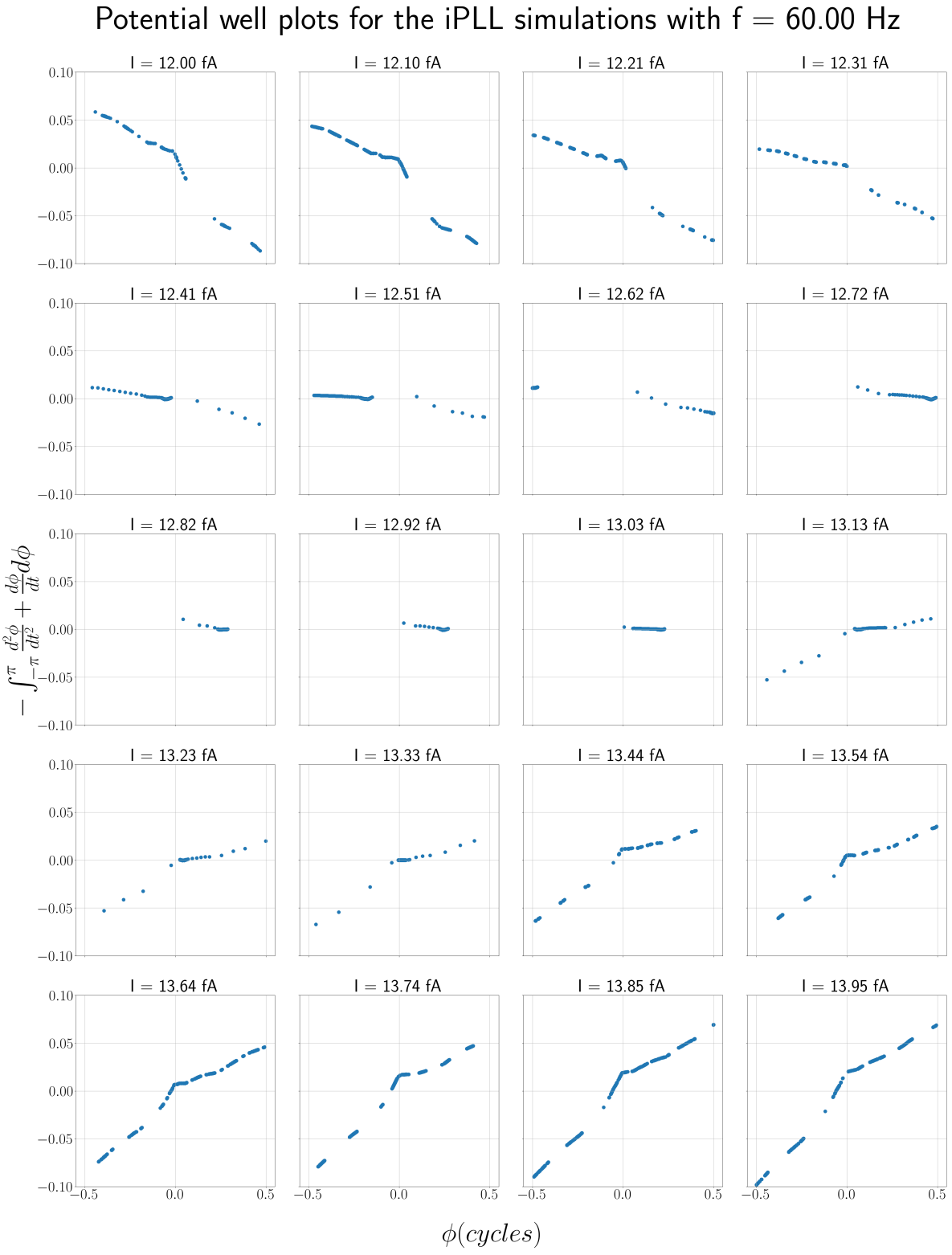

Figure 23: Potential landscapes calculated from the iPLL simulations. The integrals were shifted to be 0 on average. The currents I in the titles refer to the constant input current to the internal oscillator. In this case, the strongest locking (the lowest count of TDE spikes) happened at $I = 12.35$ fA. Note how here, for the locking sPLLs, there are clear trajectories to a final locking point, dependent on the input current.

5 Discussion

5.1 Repeatability of chip measurements

During testing, stability of the chip was a major issue. 'Stability' here refers to two things: repeatability of measurements, and robustness of the behaviour. Repeatability of the measurements was mainly impacted by temperature. Due to the temperature dependence in the I-V characteristic of transistors [24], and the specific design of the neuron circuit, a number of the currents in the circuit are exponentially dependent on temperature [21]. This manifested during testing as a variance in the internal oscillator free-running frequency, from which changes in excess of 30Hz were seen due to a few degrees of heating or cooling. In the lab setting, this could be countered by controlling the temperature of the ambient air, but this issue would need to be addressed before any real-world implementation of the circuit can be done.

Besides temperature, there also seemed to be other influences on the internal frequency. For instance, during an attempt to characterise the relationship between the internal oscillator constant current and its free-running frequency, for which the circuit had been running for a considerable amount of time, the chip needed to be reset, as it had stopped responding to inputs from the computer. This was done by temporarily cutting the power to the chip, after which power was restored and the measurements continued. By doing this however, something had caused the frequency to increase by around 5% (a 3 Hz increase around 60 Hz), but this frequency also seemed to be slowly decreasing down to its original value. After leaving the circuit running for approximately half an hour, the values had returned to their previous state. We currently do not know what caused this.

5.2 Robustness of chip behaviour

Robustness of the chip behaviour refers mainly to the potential for what we call 'runaway scenarios', where the output of the TDE stops spiking and ends up in a permanently excited state. These runaway scenarios had two possible causes. The first one, which is mostly expected behaviour, has to do with the trigger time-constant of the TDE synapse, τ_t, as found in equation 14. If the oscillator frequency is too high for the trigger time-constant, it might enter an 'integrating' regime, where the trace does not return to its resting point before the next spike arrives. This in turn can cause the TDE to spike more than usual, causing the oscillator to speed up, entering a positive feedback loop. This can eventually end up with the trigger trace fully depressed, and so the synapse current as high as it can be. This current is so large that it can overwhelm the TDE neuron to the point where the output is permanently excited, due to the negative feedback not being strong enough to reset V_{mem}.

In a worst-case scenario, when the output of the TDE enters this permanently excited state, it forces the CCO synapse to release a large amount of current as well, which in turn overwhelms the CCO neuron. The CCO neuron then also enters a permanently excited state, creating a scenario where the TDE synapse might not be able recover, even if no more spikes are sent to the circuit. In this case, the circuit needs to be reset, either by resetting the whole chip, or by sending new instructions to the internal DAC, as that sends out reset pulses as well.

This behaviour can be mostly guarded against by careful tuning of the gains and time-constants of the circuit, but if the input spike rate increases past what the facilitatory time-constant is tuned for, it can still cause uncontrolled TDE spiking, potentially kick-starting such a loop. If, on the other hand, the gains and time-constants are set too conservatively, the TDE might fail to spike at all, or only be able to spike very little, decreasing the time-resolution of the circuit.

While the behaviour described above is partially within the expected behaviour of the TDE, we found that it can also be caused by something that isn't. Specifically, it was found that in some cases, as the TDE started spiking, it prevented the TDE synapse voltages from recovering. However, if the synapse voltages do not recover, the TDE neuron keeps receiving current, possibly causing it to spike

more, which would then again stop the synapse from recovering more, creating a positive feedback loop. Often this behaviour would end before the next spike arrived, but due to remaining low for longer, the synapse would not always be able to reset fully, causing it to enter the integrating regime, and potentially creating a feedback loop as described above. This behaviour became especially pronounced in cases where the TDE would already be spiking a lot, meaning we had to tune the circuit to avoid this, as described in section 3.3.1.

The reason for this is not fully clear, but is likely related to the specific layout of the circuit on the chip, or even the circuit itself. Specifically, there might be issues with the power delivery to the synapse, where the TDE neuron draws so much power that not enough current remains available for the synapse to recover. It could also be due to the choice to use MOS-capacitors, which are essentially CMOS transistors with their drain and source terminals connected to the bulk, instead of regular capacitors. One upside is that, while the sPLL uses an older version of the TDE as developed by Milde et al. in [14], the Cognigr1 chip also contains test circuits for an improved version of the TDE, which might not face these issues.

5.3 Parameters

As can be seen in Table 2, there was quite a discrepancy between the parameters used in the Cadence® simulation and those used in the chip measurements, especially for the single-frequency experiments. Some of these are within the expected ranges of mismatch (the variation in transistor parameters) and temperature, while others are far outside this. A number them were set to avoid the instabilities described in the previous section, but some are remnants of the various stages of trial-and-error the parameter set went through. It could also be that some of the parameter changes, especially for the CCO constant current, which was also vastly different between the simulation and the chip, were necessary to overcome the temperature difference. As listed in Table 2, the simulation was run at 27° C, while the chip was tested at 20° C, and as discussed above, temperature seems to be a large factor in the behaviour of the chip. A first step towards 'unifying' these parameter sets would thus be to run the simulations closer to the chip testing temperature, which, due to time constraints, was sadly outside the scope of this thesis.

It should also be noted that after the re-tune of the chip parameters for the dual-frequency experiment, a number of parameters were significantly closer to those used in the simulation. Since the goal of this retuning was to create a behaviour closer to the tight locking behaviour described in section 4.2, and this goal was mostly achieved, we can see how the different parameter sets do indeed create different behaviours in the circuit. It would be interesting to further explore which parameters influence this, and what the benefits and drawbacks of each mode are.

5.4 Single and dual frequency detection

In sections 4.1 and 4.3, we saw that we can use TDE spike rates to describe a relationship between the input frequency and the CCO constant current. This means that if we were to have a 'bank' of sPLLs, all with different constant currents, we can detect frequencies by presenting the input signal to all of them, and seeing which one spikes least. By then employing an inverted winner-take-all network, which detects the lowest spiking sPLL, like described in [6], we can detect frequencies without ever leaving the spiking domain.

One thing that needs to be taken into account however, is that the signals presented to the sPLLs were clean, pure-frequency signals. This means that, unlike what we would expect in a real-world setting, the only spikes in the signal were from the frequency on test, and this frequency was perfectly constant, with no significant fluctuations in the ISIs. While the dual-frequency results do show that it is still possible to detect a frequency when a signal of another frequency is present, if the TDE parameters are tuned correctly, there should be very little interactions between the sPLLs locking to

one signal and the other signal. On the other hand, frequency noise in a single signal has a higher likelihood of interacting poorly with the locking sPLLs and breaking the lock, as the 'noisy' spikes will inherently be close to the spike the sPLLs are locking to. Another issue could come up when a dual-frequency signal is presented where the two signals are not perfectly in-phase (remember that the signals were generated in such a way that they both spiked at $t = 0$), or with a small frequency difference, meaning they could spike in relatively short succession. Both of these cases should be further investigated.

The data presented in this thesis also completely ignores the time it takes for the frequency detection to accurately take place. For each dataset, the sPLLs were run for either 1 or 10 seconds, while in an ideal world, it would be possible to detect a frequency within a small number of cycles, as the signal might not be presented for very long, or might shift in frequency over time. It would therefore be useful to further investigate this, either by performing more experiments, or by performing further analysis on the data gathered for this thesis to see how the average spike rates evolve over time.

Finally, as was mentioned in section 4.3, given the data presented, the question needs to be raised whether the closed-loop nature of the sPLL is important or not. While of course further experiments need to be performed, we can theorize that, in case the system is run in an open-loop fashion, frequency detection becomes far more dependent on the exact frequency of the CCO, as well as the initial condition of the time-difference ϕ and the total measurement time. We can demonstrate this by imagining a case where the CCO is very slightly slower (order $< 1\%$) than the external signal. If the system now starts with $\phi = 0$, the state of the sPLL will immediately drift into the high-spiking regime of the TDE, and stay there for a long time, making it seem like the system spikes more if frequencies are close. If we now leave the system to evolve for longer, or simply start it at $\phi = 0.5$ cycles, the sPLL state will be in the low-spiking regime of the TDE, making it seem like the system instead spikes less if the frequencies are close. From this contradiction, we can draw the conclusion that an open-loop version of the sPLL is not likely to perform very well at frequency detection.

5.5 Inhibitory sPLL

The data from the python simulations for the iPLL show that it is a viable alternative to the current ePLL-style sPLL, and could very well behave better than the current setup. However, besides the fact that we currently do not have a circuit implementation of the iPLL, which can be solved by simply replacing the synapse subcircuit by an existing inhibitory synapse circuit, the iPLL also has some other drawbacks. Specifically, in the current implementation, we noticed that the locking iPLLs had rather slow transients before reaching their locking states. This behaviour was especially pronounced in the 'strongest locking' iPLLs, so the ones which had their 'free-running' CCO frequencies closest to the external ones. What this creates is a situation where one would need to wait a minimum amount of time for all the transients to settle before a confident readout of the frequency could be made.

We theorize however, that this behaviour can be limited by setting a more aggressive time constant τ_f on the TDE facilitatory trace, making it so that the locking states are less spread out in phase-space, and so less space needs to be traversed to get to the final state. This does reduce the strength of the frequency-phase relationship however, and might introduce more phase noise, making it so that an extra trade-off needs to made when tuning the circuit.

On the other hand, due to the 'smoother' TDE response near most locking points, once locked, the iPLL might be far more robust against frequency noise, and should also be more effective at demodulating frequency-modulated or phase coded signals. There might also be merit in a 'combined' sPLL, where the iPLL is combined with an ePLL with the inputs of its TDE swapped. This 'swapped' ePLL, which is similar to the one used in [6], would have the CCO connected to the facilitatory input of the TDE, meaning that it is only active when the CCO spikes before the external signal does. The combined sPLL would then have one CCO, neuron, but two TDEs and two synapses, one inhibitory and one excitatory. If tuned correctly, this would allow the 'slow' sPLLs to lock with the 'swapped' TDE in the region $-0.5 < \phi < 0$ cycles and the 'fast' sPLLs to lock with the inhibitory TDE in the region $0 < \phi < 0.5$ cycles.

5.6 Potential well model and further modeling steps

Comparing the expectations laid out in sections 2.3.6 and 2.3.7 to the results found, we can see that the results follow the predictions from the potential well model very well. Furthermore, in section 4.5, we see that we can also recover potential landscapes similar to those expected from the actual data. Even the results from Cadence® (Figure 22), while at first seeming to disagree due to the long time-constants of the TDE, can be adequately explained, and show locking behaviour as long as a well is present. The stronger well created by this might actually be something worth exploring more in the future.

At the same time, the potential well model has a fatal flaw in the fact that we cannot directly relate the model parameters m, C_f, α, and β (see equation 22) to the parameters set in the sPLL, as outlined in Table 1. This means that, while the potential well model might be good for understanding and explaining the behaviour of the sPLL, it cannot help us in directly modeling the behaviour under specific parameters. If such a model were to exist, it would be of great use in tuning, as it would allow us to compute the behaviour directly without simulating or running tests.

One possibility for constructing such a model is by using Event-Triggered Control theory, as described in [7]. This field of control theory deals with systems in which the control signals are not continuously updated, but are instead computed based on unscheduled 'events', triggered by the state of the system. In our case, these events would be CCO spikes, as those are what cause the TDE to evaluate the time difference and send out a control signal. The theory laid out in [7] then treats the control signal in a 'sample-and-hold' manner, where it is expected that the control signal is constant untill the next event arrives. This at first seems to pose an issue, since the output of the CCO synapse is not constant in time, but we can make some simplifications and assumptions to get around this issue and arrive at a first viable model.

Specifically, if we assume that the CCO synapse I_τ is set high enough that the synapse current resets before the next CCO spike arrives, we can instead view this current as a 'charge packet' being delivered to the neuron during the time between the two spikes. Using now equations 7 and 11, we find that the ISI of the neuron is linearly dependent on the amount of spikes produced by the TDE, and thus follows the relationship laid out in equation 16. This leaves the TDE response as the only non-linear element in the model, and thus, together with actually implementing the Event-Triggered control theory, the final hurdle of building such a model, which is sadly outside the scope of this thesis.

5.7 Outlook

Although the current implementation of the sPLL is not ready to be used in applications, and a lot more testing needs to be done, looking forward, we can see various ways in which it might be applied. The first one that comes to mind links back to what inspired the sPLL in the first place: touch. While artificial vision, hearing, and even olfaction have all seen significant work in the recent decades [25, 26, 27], neuromorphic touch has been lagging behind. Improvements in artificial touch are not just useful in autonomous robotics, but might also improve remote-operated machinery, such as surgical robots, as well as greatly improve the abilities and experiences of patients with upper-limb prosthetics [28]. By employing neuromorphic methods to implement this sense of touch, we do not only save in cost and energy, but also open up the way to integration with the human nervous system directly, as was demonstrated in [29].

Besides touch, we also see other areas where the sPLL might be useful. For instance, as support circuitry for other neuromorphic circuits, where efficient frequency detection of spikes might be important. It could also be that they find a use in Oscilatory Neural Networks (ONNs). ONNs are neural networks in which the computation is performed by coupled oscillators, and, such as in [30], sometimes these oscillators are coupled using PLLs. By instead implementing the oscillators using CCO neurons, and using sPLLs, more energy- and space efficient networks might be feasible.

Finally, the theoretical understanding gained from the sPLL might help model and create future

closed-loop spiking systems, as well as answer questions about current ones. We can also envision this understanding feed back into neuroscience, for instance in understanding phase-locking in the brain, or in helping explain the phenomenon of anticipating synchronization [13, 31].

6 Conclusion

We have successfully tested the Spiking Phased-Locked Loop (sPLL) circuit, both on-chip and in simulation, and shown that is capable of detecting the frequencies of both single- and dual-frequency signals. However, this frequency detection was done with idealized input signals, and significant discrepancies existed between the parameter sets used to tune the simulated and on-chip circuit behaviours, warranting further investigation.

We have also developed and tested a qualitative model for understanding the sPLL behaviour, the potential well model, and shown the predictions made with the model to hold true. We have shown that the potential landscapes used to make these predictions can be recovered back from the experimental data. Furthermore, we have discussed a way forward towards a more quantitative model, based on event-triggered control theory, which would be able to help with modeling and design of future circuits.

Finally, we used the potential well model to design an improvement to the sPLL, the inhibitory sPLL, and tested this in python. This iPLL has shown improvements over the regular sPLL in both phase response and output stability, and could open up usecases beyond those of the regular sPLL. Although no circuit for the iPLL has been designed yet, the necessary building blocks exist, and we strongly recommend this for further testing.

Acronyms

$V_{\mathbf{mem}}$ membrane potential. 5, 6, 11, 31

BICS Bio-Inspired Circuits and Systems. 3, 17, 18

CCO Current Controlled Oscillator. 8–10, 13, 16, 17, 19–27, 31–34

CMOS Complementary Metal Oxide Semiconductor. 1, 3, 7, 8, 10, 18, 32, 36

DAC Digital Analog Converter. 18, 31

DFT Discrete Fourier Transform. 2, 3, 17

DPI Differential-pair Integrator. 1, 9–11, 13

ePLL excitatory sPLL. 8, 21, 24, 26, 27, 33

FIFO First-In-First-Out structure. 18

iPLL inhibitory sPLL. 1, 16, 17, 19, 21, 24, 26, 27, 30, 33, 35

ISI Inter-Spike Interval. 6, 20, 32, 34

LIF Leaky Integrate-and-Fire. 5, 8, 11, 19

LPF Low-Pass Filter. 4, 5, 8

NMOS n-type CMOS Field Effect Transistor. 18

ONN Oscilatory Neural Network. 34

PC Pacinian Corpuscle. 2

PLL Phased-Locked Loop. 2, 4, 8, 16, 23, 34

PMOS p-type CMOS Field Effect Transistor. 10, 18

SA1 Slowly adapting type 1. 2

sPLL Spiking Phased-Locked Loop. 1–3, 7–10, 13, 16, 17, 19–27, 29, 30, 32–36

TDE Time Difference Encoder. 1, 3, 8–10, 12–17, 19–27, 30–34

VCO Voltage Controlled Oscillator. 4, 5, 8

VLSI Very Large Scale Integration. 17

References

[1] Hannes P. Saal and Sliman J. Bensmaia. "Touch is a team effort: interplay of submodalities in cutaneous sensibility". In: *Trends in Neurosciences* 37.12 (Dec. 2014), pp. 689–697. ISSN: 0166-2236. DOI: 10.1016/J.TINS.2014.08.012.

[2] Alison I. Weber et al. "Spatial and temporal codes mediate the tactile perception of natural textures". In: *Proceedings of the National Academy of Sciences of the United States of America* 110.42 (Oct. 2013), pp. 17107–17112. ISSN: 00278424. DOI: 10.1073/PNAS.1305509110/SUPPL{_}FILE/PNAS.201305509SI.PDF. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1305509110.

[3] Michael A. Harvey et al. "Multiplexing Stimulus Information through Rate and Temporal Codes in Primate Somatosensory Cortex". In: *PLOS Biology* 11.5 (2013), e1001558. ISSN: 1545-7885. DOI: 10.1371/JOURNAL.PBIO.1001558. URL: https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.1001558.

[4] Ehud Ahissar, Sebastian Haidarliu, and Miriam Zacksenhouse. "Decoding temporally encoded sensory input by cortical oscillations and thalamic phase comparators". In: *Proceedings of the National Academy of Sciences of the United States of America* 94.21 (Oct. 1997), pp. 11633–11638. ISSN: 00278424. DOI: 10.1073/PNAS.94.21.11633/ASSET/50EA9CA7-44DA-4D94-BA99-0A927474CEAA/ASSETS/GRAPHIC/PQ2070416005.JPEG. URL: https://www.pnas.org/doi/abs/10.1073/pnas.94.21.11633.

[5] Ehud Ahissar. "Temporal-Code to Rate-Code Conversion by Neuronal Phase-Locked Loops". In: *Neural Computation* 10.3 (Apr. 1998), pp. 597–650. ISSN: 08997667. DOI: 10.1162/089976698300017683.

[6] Michele Mastella and Elisabetta Chicca. "A hardware-friendly neuromorphic spiking neural network for frequency detection and fine texture decoding". In: *Proceedings - IEEE International Symposium on Circuits and Systems* 2021-May (2021). ISSN: 02714310. DOI: 10.1109/ISCAS51556.2021.9401377.

[7] W P M H Heemels, K H Johansson, and P Tabuada. "An Introduction to Event-triggered and Self-triggered Control". In: (2012). DOI: 10.0/Linux-x86{_}64.

[8] K. O. Johnson. "The roles and functions of cutaneous mechanoreceptors". In: *Current Opinion in Neurobiology* 11.4 (Aug. 2001), pp. 455–461. ISSN: 0959-4388. DOI: 10.1016/S0959-4388(00)00234-8.

[9] Guan Chyun Hsieh and James C. Hung. "Phase-locked loop techniques - A survey". In: *IEEE Transactions on Industrial Electronics* 43.6 (1996), pp. 609–615. ISSN: 02780046. DOI: 10.1109/41.544547.

[10] Behzad Razavi. *RF microelectronics*. Second Edition. 2011. ISBN: 978-0137134731.

[11] Steve De Weerth et al. "A neuron-based pulse servo for motion control". In: (1990), pp. 1698–1703. DOI: 10.1109/ROBOT.1990.126254.

[12] Rodolphe Sepulchre. "Spiking Control Systems". In: *Proceedings of the IEEE* 110.5 (May 2022), pp. 577–589. ISSN: 15582256. DOI: 10.1109/JPROC.2022.3163926.

[13] Elif Köksal Ersöz et al. "Anticipation via canards in excitable systems". In: *Chaos* 29.1 (Jan. 2019), p. 13111. ISSN: 10541500. DOI: 10.1063/1.5050018/341696. URL: /aip/cha/article/29/1/013111/341696/Anticipation-via-canards-in-excitable-systems.

[14] M. B. Milde et al. "Spiking Elementary Motion Detector in Neuromorphic Systems". In: *Neural computation* 30.9 (Sept. 2018), pp. 2384–2417. ISSN: 1530-888X. DOI: 10.1162/NECO{_}A{_}01112. URL: https://pubmed.ncbi.nlm.nih.gov/30021082/.

[15] Louis Lapicque. "Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation". In: *Journal of Physiol Pathol Générale* (1907), pp. 620–635. URL: https://fr.wikisource.org/wiki/Recherches_quantitatives_sur_l%27excitation_%C3%A9lectrique_des_nerfs_trait%C3%A9e_comme_une_polarisation.

[16] Laurence F. Abbott and Peter Dayan. *Theorethical Neuroscience*. 2001. ISBN: 9780262041997.

[17] Balthasar Van der Pol. "On "relaxation-oscillations"". In: *http://dx.doi.org/10.1080/14786442608564127* 2.11 (Nov. 1926). ISSN: 1941-5982. DOI: 10.1080/14786442608564127. URL: https://www.tandfonline.com/doi/abs/10.1080/14786442608564127.

[18] Richard FitzHugh. "Impulses and Physiological States in Theoretical Models of Nerve Membrane". In: *Biophysical Journal* 1.6 (1961), p. 445. ISSN: 00063495. DOI: 10.1016/S0006-3495(61)86902-6. URL: /pmc/articles/PMC1366333/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1366333/.

[19] J. Nagumo, S. Arimoto, and S. Yoshizawa. "An Active Pulse Transmission Line Simulating Nerve Axon*". In: *Proceedings of the IRE* 50.10 (1962), pp. 2061–2070. ISSN: 00968390. DOI: 10.1109/JRPROC.1962.288235.

[20] Chiara Bartolozzi and Giacomo Indiveri. "Synaptic dynamics in analog VLSI". In: *Neural computation* 19.10 (Oct. 2007), pp. 2581–2603. ISSN: 0899-7667. DOI: 10.1162/NECO.2007.19.10.2581. URL: https://pubmed.ncbi.nlm.nih.gov/17716003/%20https://pubmed.ncbi.nlm.nih.gov/17716003/?myncbishare=CMB&otool=inlugrmlib.

[21] Elisabetta Chicca et al. "Neuromorphic electronic circuits for building autonomous cognitive systems". In: *PROCEEDINGS OF THE IEEE* 1 (Mar. 2014). DOI: 10.1109/JPROC.2014.2313954. URL: http://arxiv.org/abs/1403.6428%20http://dx.doi.org/10.1109/JPROC.2014.2313954.

[22] Giacomo Indiveri, Fabio Stefanini, and Elisabetta Chicca. "Spike-based learning with a generalized integrate and fire silicon neuron". In: *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems* (2010), pp. 1951–1954. DOI: 10.1109/ISCAS.2010.5536980.

[23] Hugh Greatorex, Michele Mastella, and Elisabetta Chicca. "Visual Event Based Ego Motion Estimation with a Neuromoriphic Time Difference Encoder Circuit, Unpublished". In: ().

[24] Shih-Chii Liu et al. "Analog VLSI: Circuits and Principles". In: *Analog VLSI* (Dec. 2002). DOI: 10.7551/MITPRESS/1250.001.0001. URL: https://direct.mit.edu/books/book/1826/Analog-VLSICircuits-and-Principles.

[25] Mattias Nilsson et al. "A Comparison of Temporal Encoders for Neuromorphic Keyword Spotting with Few Neurons". In: (Jan. 2023). URL: http://arxiv.org/abs/2301.09962.

[26] Giacomo Indiveri and Rodney Douglas. "Neuromorphic Vision Sensors". In: *Science* 288.5469 (May 2000), pp. 1189–1190. ISSN: 0036-8075. DOI: 10.1126/science.288.5469.1189.

[27] Anup Vanarse, Adam Osseiran, and Alexander Rassau. "An Investigation into Spike-Based Neuromorphic Approaches for Artificial Olfactory Systems". In: *Sensors* 17.11 (Nov. 2017), p. 2591. ISSN: 1424-8220. DOI: 10.3390/s17112591.

[28] Chiara Bartolozzi. "Neuromorphic circuits impart a sense of touch". In: *Science* 360.6392 (June 2018), pp. 966–967. ISSN: 0036-8075. DOI: 10.1126/science.aat3125.

[29] Yeongin Kim et al. "A bioinspired flexible organic artificial afferent nerve". In: *Science* 360.6392 (June 2018), pp. 998–1003. ISSN: 0036-8075. DOI: 10.1126/science.aao0098.

[30] Frank C. Hoppensteadt and Eugene M. Izhikevich. "Pattern recognition via synchronization in phase-locked loop neural networks". In: *IEEE Transactions on Neural Networks* 11.3 (May 2000), pp. 734–738. ISSN: 10459227. DOI: 10.1109/72.846744.

[31] Leonardo Dalla Porta et al. “Exploring the Phase-Locking Mechanisms Yielding Delayed and Anticipated Synchronization in Neuronal Circuits”. In: *Frontiers in Systems Neuroscience* 13 (Aug. 2019), p. 41. ISSN: 16625137. DOI: 10.3389/FNSYS.2019.00041/BIBTEX.

A Parameters used

Name	Simulation	Chip single-frequency	Chip dual-frequency
Fac_tau_PI	10p	40p	80p
Fac_w_NI	10n	2n	2.5n
leak_osc_NI	1p	50p	1p
leak_tde_NI	10p	10p	3p
pw_NI	100p	100p	7n
refr_tde_NI	5n	250n	15n
rest_tde_PI	1p	30p	0.1p
tau_syn_PI	5p	3p	15p
thr_osc_NI	100p	5n	100p
thr_syn_PI	100p	100p	300p
thr_tde_NI	10p	50p	500p
Trg_tau_PI	20p	20p	40p
Trg_w_NI	50p	1n	50p
w_syn_NI	100p	300p	100p
st_NI	100n	100n	100n
refr_osc_NI	1n	50n	1n
st_PI	100n	100n	300n
Temperature	27° C	20°	20°

Table 3: The full parameters used in the Cadence® simulations and chip measurements, with the names as used in Cadence®. The pw_NI parameter governs the pulse width of the pulse-extender situated between the internal oscillator and the input of the TDE, whereas the st_NI and st_PI parameters refer to the input parameters for a starved inverter on the input of said pulse-extender. The other parameters are all described in the Methods section.

B Data and code availability

In the short term, all raw data and code used for this work will be available through my personal server, please email me at t.e.tiemens@student.rug.nl for access. In the long term ($>$ 1 month from publication), the data and code will be available through the BICS group of the University of Groningen, please contact Michele Mastella for more info.