(Not so) brief memo on the Camadia tuning procedures

1 Introduction

Kinetic Inductance Detectors are superconducting resonators whose resonance frequency shifts depending on the incoming optical power. The measure of such frequency shift is what allows us to use KID as mm-wave detectors, which is great. Yet at the same time it poses a problem for the optimization of the detector sensitivity. Any change in the background optical load (due, for example, to changes in the atmospheric transmission or in the elevation) will in fact also contribute to the shift of the resonances. In order to maximize the sensitivity of a KID, the excitation signal used to read it out must always be near to the its resonance frequency. Therefore, if the background load changes significantly, one must be able to evaluate the new resonance frequencies of the detectors and adapt the frequencies of the excitation tones accordingly. This counteracts the effect of the variation in the background load and allows to always keep the detectors at their ideal working point.

The Camadia® software developed for NIKA2 includes a series of procedures that are dedicated to this scope. These are the object of this memo. In particular, in section 2 we describe the two different kind of methods used for tuning the KID detectors, while in section 3 we outline the associated flagging system that allows to keep track of them.

2 Tuning vs Decalage

The two methods used for finding the optimal excitation tones are called *Tuning* and *Decalage*. This is somewhat confusing, as the term 'tuning' is often used for the whole procedure of optimizing the excitation tones and not for the specific method Tuning. To try and make things clearer, we'll always use capital letters for the specific methods described in this section. Basically, we can say that:

- **Tuning** is the final step of tones optimization, and is the method that really fine-tunes the excitation tone of each single pixel.
- **Decalage** can be seen as a 'rough tuning', which allows to make a first global compensation of a change in the background load by acting at the level of an entire feedline rather than on a pixel per pixel basis.

2.1 The KID resonance circle

In order to understand how the two methods work, it is necessary to have in mind the picture of how a KID affects an excitation signal sent on the feedline. Without giving all the details, for which one can make reference to many useful papers, we will outline the here only the main results. We will represent an idealized KID, not affected by distortion or parasitic effects, as these are not in any case of interest for the tuning process in itself.

For the KID readout, an excitation signal is sent into the cryostat on the feedline coupled to the KID. The transmitted signal can be described by its amplitude and phase, or, as is common practice for KID, by its components that are In-phase (I) and in Quadrature Q with respect to the excitation signal. When a frequency sweep is carried around a KID resonance, the transmitted signal makes a sort of circle in the *IQ plane*, as shown in figure 1. One of the fundamental advantages of the NIKA2 electronics and readout system is that, for each acquisition point, it gives as an output both the (I, Q) values, and the (dI, dQ) ones, which represent the tangent to the resonance circle. The way in which the (dI, dQ) values are calculated is not described here. If interested in this, some (cryptic) details can be found in Calvo et al., A&A 551, L12 (2013).

The tuning methods are essentially based on the measurement of the angle ϕ between the (I, Q) and the (dI, dQ) vectors, as shown in figure 1. This angle for convenience translated to $\vartheta = \phi - \pi/2$ as defined in the figure. If we plot its variation during a scan around a resonance, we find a curve $\vartheta(f)$ similar to the one shown in figure 2. Therefore,

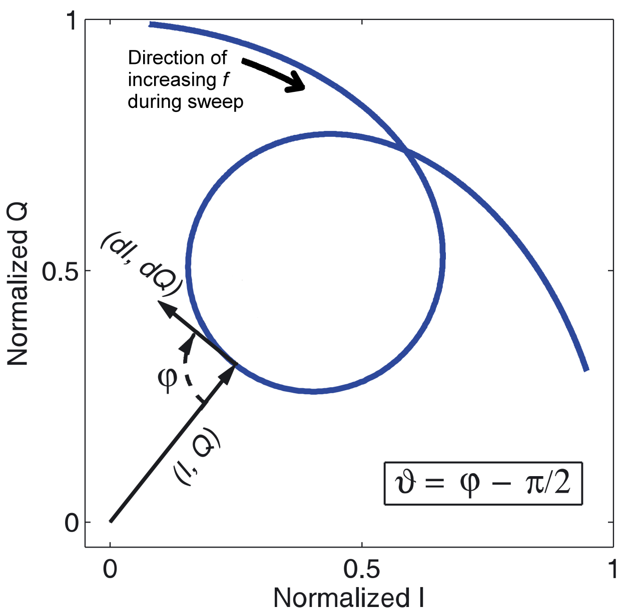


Figure 1: Sample resonance circle in the IQ plane. Both the tuning procedures are based on the measurement of the angle ϑ between the two vectors (I, Q) and (dI, dQ)

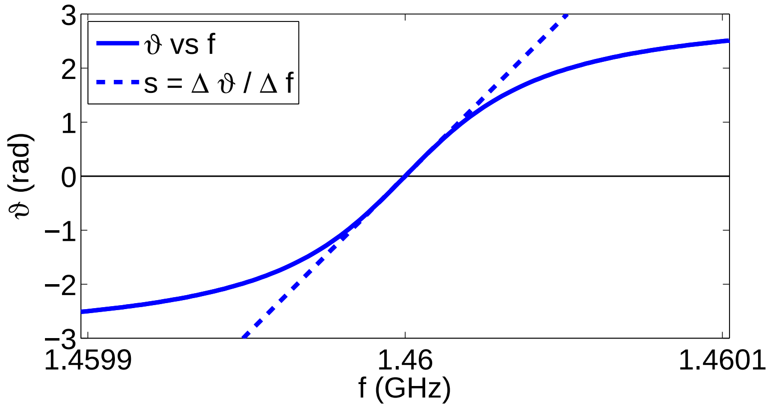


Figure 2: Sample $\vartheta(f)$ curve observed when making a frequency sweep around a resonance. The ideal operating point of a KID is where $\vartheta = 0$

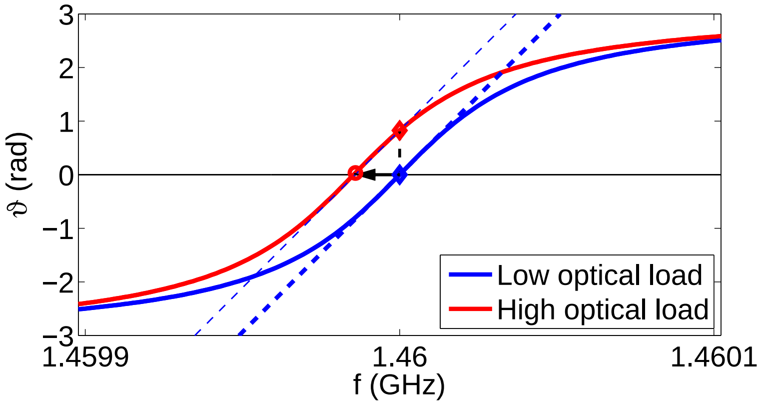
we can define the resonance point, and thus the ideal KID working point, as the one where $\vartheta = 0$.

If a change in optical load is observed, the resonance frequency will shift. As we will see later, the resonance circle shape will also be a little bit affected, but let us start with a simplified case in which this extra effect is neglected. Then, the change in f_0 will just result in a shift of the $\vartheta(f)$ curve. Since the excitation tone remains fixed (until we do a tuning procedure, clearly!), what will happen is that we will observe a variation in the angle ϑ between (I, Q) and (dI, dQ). If we know the slope s of the $\vartheta(f)$ curve around the resonance, $s = \Delta\vartheta/\Delta f$, expressed in radians per Hz, then we can easily estimate the new position of the resonance frequency, which will be given by:

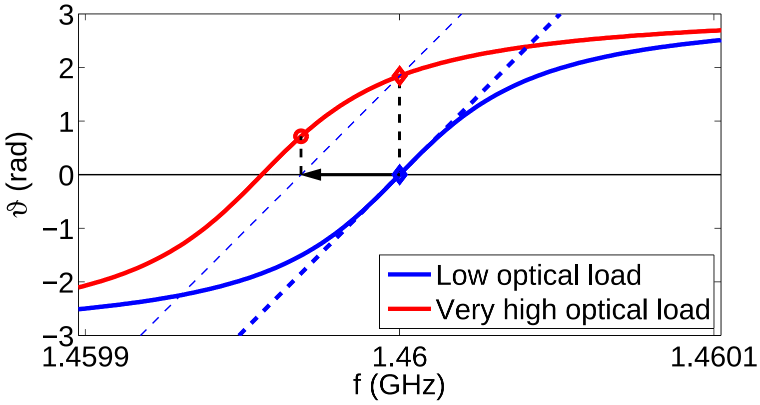
$$f_0^{new} = f_0 - \vartheta^{new}/s \tag{1}$$

The slope s is calculated a first time by means of a frequency sweep around the resonances at the beginning of a campaign, and is saved in the 'default parameter' of the array. Then, as we will see, can be automatically updated by Camadia during the tunings to take into account its variations. Note that in the Camadia jargon it is common to speak in terms of resonance width (or *largeur*), calculated as the width between the points at +0.5rad and -0.5rad in the $\vartheta(f)$ curve. In this case one simply has $s = 1/w$.

The formula (1) is valid only for small variations of the resonance frequency, as in this case one remains in the region where $\vartheta(f)$ is approximately linear. To illustrate this, in figure 3a and 3b we show two examples of correction of an excitation tone done using (1). In both cases, the optical load increased, causing a shift towards lower frequencies of the $\vartheta(f)$ curve. In the first case a small change of the optical load has been assumed, leading to a small δf_0. This is the condition in which the estimated correction is almost exact. On the other hand, if a much larger shift is assumed, one single correction step is not enough to find the new resonant frequency, although it allows to get nearer to it. The error can nonetheless be still relatively large. In this case, the correction can be



(a)



(b)

Figure 3: Two example of tone frequency corrections starting from the measure of the angle ϑ. In these images, the blue diamonds represent the original positions of the resonance frequencies, where the original excitation tones are set. When the load increases, before the tones are corrected the observed angle ϑ will increase (red diamonds). The estimated correction is indicated by the black arrows, and the red circles are the actual position on the $\vartheta(f)$ curve after the tone has been corrected. As it can be seen, in the case of small frequency variations the correction is almost perfect (a),whereas if the change in optical load is strong one single step is not sufficient to find the new resonance point (b). The iteration of the process allows in any case to rapidly converge to the correct frequency.

reiterated multiple times in order to reach the correct working point.

Now that we have a clear picture of the basic principle on which the tuning procedures are based, we can describe them in more detail to understand the difference between the Decalage and the Tuning methods.

2.2 Decalage

The Decalage method is essentially a correction carried out on a per feedline basis. The steps done during a Decalage are the following:

1. The angle ϑ is measured for each KID of a feedline. To reduce the noise of the measurement, this is repeated for chosen number of data points that are all averaged together. This number is defined by the *moyenne* ('average' in french!) variable of in the *controle* program of Camadia. Typical values are 20 to 40 depending on the sampling rate of data acquisition.

2. All the measured angles are averaged together to get $< \vartheta >$.

3. The median resonance width of the KIDs of the feedline is evaluated, and hence their median slope s_{med}. Note that KIDs having a resonance width much different from the expected one are excluded from the estimation of s_{med}.

4. If the absolute value of $< \vartheta >$ is above a chosen threshold (typically of order of 400mrad, or ∼22deg), then a correction $\Delta f_{feedline} = - < \vartheta > / s_{med}$ is applied. The same correction is used for all the tones of the considered feedline.

5. If the excitation tones have been changed, a number of acquisition sample are ignored. This waiting time is necessary: it assures that all the commands sent to the electronic boards in order for them to change the frequencies are well received and that the new frequencies have effectively been set. The maximum waiting time is defined by the *attente* ('waiting') parameter of the controle application. Note that as soon as a message is received from the electronic boards telling to the system that the frequencies have been updated (corresponding to a particular flag, see section 3.1), then only 2 extra samples are ignored, then the waiting is interrupted and the program proceeds to the next step. This is true for all the waiting periods.

6. After the waiting time, the new average angle is measured for the feedline (once again, averaged over multiple samples). If this is below the chosen threshold, or if it changed sign with respect to the previous measurement (meaning the we passed on the other side of the average resonance position), then the Decalage is considered completed. Otherwise, steps 1 to 4 are repeated.

The Decalage method is only an approximate correction. Each tone might in fact need a slightly different adjustment to be able to be perfectly matched to the corresponding f_0. Nonetheless, the Decalage helps keeping track the global shift of the feedline, so that in any case all the tones will remain near to their correct position. Plus, it has a very important advantage with respect to the more precise correction done using the Tuning: when the optical load is changing rapidly, for example when the telescope's elevation is changing, it would be difficult to make iterated measurements of ϑ and the consequent corrections without each time risking to miscalculate the good position of one or more tones. Thus, after many iterations, one would risk loosing track of a lot of pixels. Since the Decalage shifts all the tones of a feedline rigidly, this risk is strongly reduced, as the global shape of the frequency comb is preserved. Only once the telescope has reached its final position, and all the Decalage iterations have been done to assure a small value of $< \vartheta >$, a final fine-tune step is carried with the Tuning method to really optimize every individual pixel.

When the option 'Decalage continu' is active, the Decalage method is applied repeatedly, until the status of the system change. This option gets for example automatically activated when no observation are being performed by the telescope, and is deactivated as soon as a scan is started.

2.3 Tuning

Contrary to the Decalage method, a Tuning makes all the calculation on a pixel by pixel basis, and allows to optimize each individual tone. I will not repeat it every time, but all the measurements are averaged over a certain number of samples and are followed by a waiting period, as describe for the Decalage method. The steps of a Tuning are the following:

1. The angle ϑ is measured for each KID of a feedline.
2. First of all, if the average angle $< \vartheta >$ is above a fixed threshold (typically the same used in the Decalage method), then a Decalage is launched. The Decalage is iterated a maximum of three times, or less if the conditions described in point 5) of the Decalage method is met.

At this point, we can be sure that all the tones are reasonably near to the corresponding f_0 and we can proceed to the final adjustment step

3. When we arrive at this point, a measurement of ϑ will have just taken place for each KID. Let us call these values ϑ^{i-1}. Based on this measurement, a shift is applied to each tone. This time each tone has its proper shift. This given by

$$\Delta f_{tone} = -\left(\frac{\vartheta_{tone}^{i-1} + 0.5rad \cdot sign(\vartheta_{tone}^{i-1})}{s_{tone}}\right)$$

   The goal of this shift is therefore not that of setting the tone on the resonance, but rather on the opposite side of the resonance with respect to the side we were before. On the $\vartheta(f)$ curve, this corresponds to try to fix the tone at a frequency such that ϑ is negative if it was positive during the last measurement, or vice versa. Consider now the values of ϑ involved: in the last measurement, before this shift is applied, it was on average less than .4rad away from 0. After this tone per tone shift is applied, we expect ϑ to be on the opposite side of the 0, at a distance of about .5rad. By looking at figure 4, it is clear that the two points should be conveniently positioned in the zone where $\vartheta(f)$ is linear.

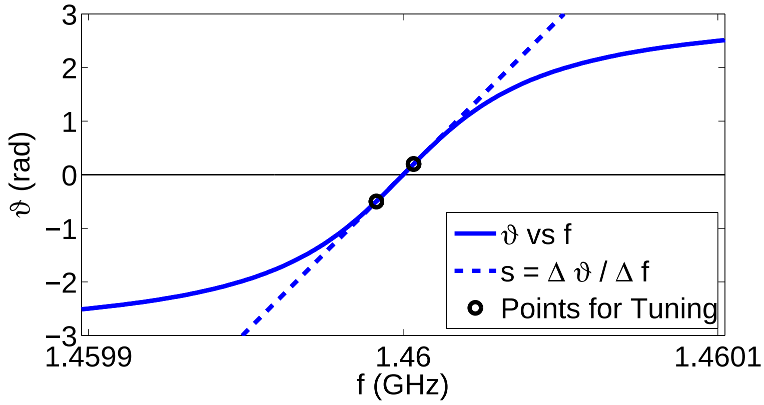


Figure 4: Example of position of the points during a Tuning step in which the tuning is going well. The first point has in this case $\vartheta^{>}0$ but lower than .4rad. The second point is shifted, as desired, to values near $\vartheta \sim -0.5rad$. It can be seen that in this case the two points lie well within the linear region of $\vartheta(f)$.

4. A measurement of theta is carried out with the 'overshifted' tones. Let us call the values found in this configuration ϑ^i. At this point, two extra checks are done:
   - for a number of reasons, it can sometimes happen that ϑ^i and ϑ^{i-1} have the same sign, contrary to what one would expect. In this case, the new frequency f^i_{tone} (supposedly nearer to f_0 in any case) is kept the for the KID showing this anomalous behaviour tone, without making the extra steps 5) to 7), and the corresponding KID is flagged as having a 'bad tuning' (see section 3.2). In many of such cases the excitation tone is actually quite near to the actual resonance frequency, and the associated timeline can be good, but this is not granted.
   - if the Δf_{tone} calculated for a pixel turns out to be much smaller or much larger than expected (with respect to the average resonance width on the array), then again the flag 'bad tuning' is activated. The tone frequency for this KID is set to the average between f^i_{tone} and f^{i-1}_{tone}, and the steps 5) to 7) are once more disregarded.
5. If on the other hand the tests in step 4) indicate that everything is going as desired, then using the last two measurements of ϑ , and knowing the frequencies of the tones that are associated to them, we can calculate the current slope of $\vartheta(f)$ for each tone as
$$s_{tone} = (\vartheta^i_{tone} - \vartheta^{i-1}_{tone})/(f^i_{tone} - f^{i-1}_{tone})$$
6. Using this value, and the measurement ϑ^i, we can calculate with very good accuracy the appropriate excitation frequency of each tone:
$$f^{final}_{tone} = f^i_{tone} - \vartheta^i_{tone}/s_{tone}$$

7. As we pointed out before, the change in optical load also changes the shape and size of the resonance circle. The curve $\vartheta(f)$ therefore changes in turn, basically become wider, and thus less steep around the resonance, if the optical load increase. During the Tuning method, a direct measurement of current value of the $\vartheta(f)$ slope is done. This new value is used to update the slope and width of each resonance. As a consequence, the new shape of the resonance circle is taken into account, and all the following Decalage calculation are based on the new values of s. This makes the whole system very stable even for extreme variations in the optical load. Basically, it can follow and optimize the KID working point even switching directly from a 300K optical load to very good sky conditions!

3 Data flagging

The tuning procedure represent a critical step for the whole data acquisition system. They are needed to correctly use the KID detectors. At the same time, every time that the excitation frequency of a tone is changed, this will lead to a jump in the corresponding timeline. Therefore, a flagging system has been introduced. The two fundamental flags associated to the tuning procedures are saved in the k_flag value. Each array has one single k_flag value associated to it (it is not individual for each pixel!). The k_flag is sampled at the same rate of the data acquisition (ie, 23.8Hz or 47.6Hz), and is obtained summing the following values:

- **1** if a frequency sweep is ongoing (not very useful as this happens only when no observations are being done)
- **2** again, if a frequency sweep is ongoing (it is actually basically a double of the previous flag..)
- **4** if the tone frequencies have been changed
- **8** if a Decalage or a Tuning is ongoing
- **32** if the tuning was not successful ('bad tuning')
- **64** it the tone seems to have really wandered away from its expected position ('bad placement')
- [**128** the resonance corresponding to this tone is lost (!obsolete flag!)]

3.1 k_flag 4 and 8

4 and **8** are the two most important values included in k_flag. It is important to understand the difference between the two. *If the flag 8 is active, this does not necessarily mean that the data are affected.* A flag of 8 just means that some calculations or measurements related to the tuning procedures are ongoing. If, as a result of the calculations, the system decides that a change of excitation tones is not needed, then nothing will happen, and the data will keep being valid. No jumps or strange effects on the timelines will be observed.

On the other hand, if the system decides that some or all the tones of a feedline need to be adjusted, then it will send the command 'change frequencies', with the new frequencies list, to the electronic boards. As soon as the boards receive the command and change the excitation tones, the value of 4 will be added to the k_flag of the corresponding array, *only for the precise sample in which the new tones have been set by the electronic boards!* In this case, the timeline of the KIDs of the considered feedline **will be perturbed**. Notice that typically not only the sample in which the tones have been changed will be affected, but a certain number of samples before and after that. This is due to the fact the *RFdIdQ* values are calculated averaging a certain number of samples, so one has to wait enough samples to be sure that this average is not affected by the change of the tones. **To stay on the safe side, for the RFdIdQ data you should consider 60 samples before and after the k_flag = 4 one as being bad, and disregard them during data analysis!** This is true only for the RFdIdQ data: the I, Q, dI, dQ data should be good already about 2 to 3 samples after the one having k_flag = 4, because they don't use the averaging on multiple samples. To have some margin, **for the I, Q, dI, dQ data consider 5 samples before and after the k_flag = 4 one as bad.**

One can easily find if a particular flag n (4, 8, etc..) was present in k_flag for a given sample by making:

$$IsActive(n) = mod(floor(k\_flag/n), 2)$$

To conclude, **extreme care** must be taken when treating data at and near to the samples where $mod(floor(k\_flag/4), 2) = 1$!

It is important to note that, while the k_flag is sampled synchronously with the acquired data, the values of the tone frequencies, f_{tones}, are not: to avoid an excessive data rate, this information is stored in the 'global settings' of the array, which are recorded once per 'block of data' received. This corresponds to 36 samples. Thus, a delay of up to 36 samples can be noticed between the sample with k_flag = 4 and the first sample for which the new values of the tone frequencies are taken into account in the array settings. If one is a bit unlucky, and the tones are changed near the end of a data block, the delay can actually even slightly exceed the 36 samples.

3.2 k_flag 32 and 64

These flags give a first indication on the quality of the tuning of each tone. The cases corresponding to this flags have already been described in section 2.3.

32 is added to k_flag in the case of a 'bad tuning', as described in point 4) of section 2.3. As pointed out, a bad tuning doesn't automatically imply that the excitation tone was very far from the resonance frequency. Our experience teaches us that many KIDs with the flag 32 activated are actually well behaving. Nonetheless, the activation of such a flag must be considered as a serious alarm bell, and *if you want to keep on the safe* *side we recommend to discard the timelines of the corresponding pixels from your data analysis.*

64 is a different kind of check, based on the position of a tone with respect to all the other ones of the same feedline. To first order, the resonance frequencies move around rigidly when the optical load changes. Therefore, the relative distances between the tones shouldn't change by much. Based on this fact, the following check is done: for each KID, its current distance from all the other well-tuned KID of the feedline (ie, without the flag 32 active) is calculated, and is compared to the distance to the same pixels found in the default parameters of the feedline, which can considered as the 'reference comb' (they are never touched once the configuration of the feedline has been done, typically at the beginning of the run!). The median of such differences is calculated: this basically tells us how much this tone shift is different from the global shift of the whole frequency comb. The nearer to zero this difference is, the better the considered pixels fits in the expected frequency comb. If the difference, on the other hand, is very large, it means that this pixel went far away from its expected position. In this case, the value 64 is added to k_flag. Pixels with this flag are extremely likely to be bad (we can say, surely..), or their tone might have got attached to the resonance of a pixel which was not the original one ('jumping pixel'). Thus, **all pixels with k_flag 64 active must be disregarded** during the data analysis!