Telecommunication Electronics

Alberto Tibaldi

March 14, 2011

Contents

Introduction

These text is the transcription of the notes, taken by the author, from the "Telecommunication Electronics" lectures, held by Professor Dante Del Corso in "Politecnico di Torino", academic year 2009/2010.

All the images of this text were taken from the learning material of the course, prepared by the Professor, under his agreement.

Alberto Tibaldi

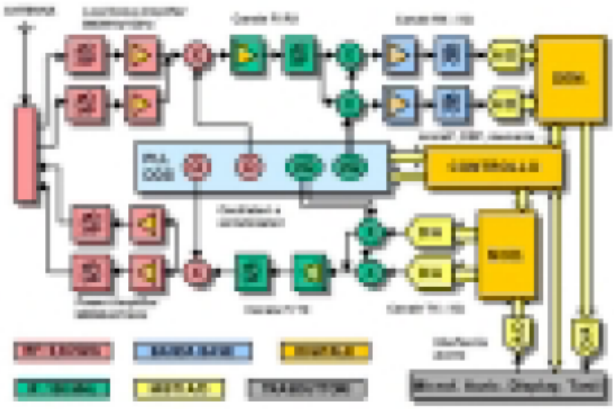
Chapter 1

Architectures of radio systems

The goal of this lecture is expose an overview of a radio system, studying the general architecture in order to identify the function that a radio system must have to work correctly.

The architectures that we will discuss in this text are *heterodyne*: with this word we identify a basic architecture based on a technique that can be used with analog or digital technologies; usually, in an electronic system, there are both analog and digital blocks, but digital parts are the most important: easy to design and realize.

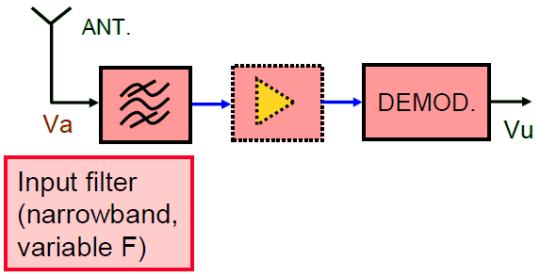
We will use, for this chapter (and generally in all the text) a *top-down approach*; it means studying the functions that the system blocks must realize, then how to realize it with electronic circuits. Which is the top in a radio system? The answer is easy: what is required for the user! With a radio system we must listen to music, speak, or something else. The first thing to do now is define the application and then identify an interesting block to study.



1.1 Receiver

A receiver is a radio system block that can select a channel (a range of frequencies) from the source (the air) and reduce the spectral components of all the other channels, so translate the chosen channel in something usable by human (like sounds!).

A simple architecture for a receiver can be the following one:



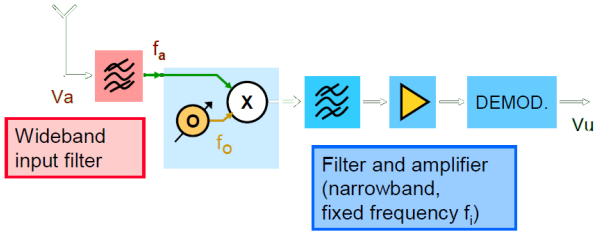
The main blocks are: antenna, narrow-band band-pass filter (that selects the good channel), and demodulator, in order to translate (demodulate) the signal (from AM, FM...).

This architecture works: shifting the response of the band-pass filter we can choose different channels, different signals; using as band-pass filter a resonant circuit.

Changing the reactive parameters of this circuit we can choose one or another signal. Problem: a resonant circuit like this is difficult to use: it can not remove all the other channels, because designing a filter with shifting frequency and narrow band is very very hard. The problem is so the impossibility to obtain a good channel isolation, so to choose only a signal and have possibility of change the channel.

1.1.1 Heterodyne architecture

The basic idea that can resolve our isolation problem is the use of an architecture like this:



This architecture has the antenna, followed by a wideband filter (useful to reject noise); there is a new block, then an amplifier-filter and demodulator.

The new block introduce a new way to work: there is a local oscillator that multiplies the signal filtered by the first wideband filter.

Before the explanations, some remarks: every signal can be identified in the following way: a $x_t(t)$ signal is equal to a $x(t)$ signal multiplied by a sine/cosine, that take account of its translation on the spectral domain:

$$x_t(t) = x(t) \cdot \cos(2\pi f t)$$

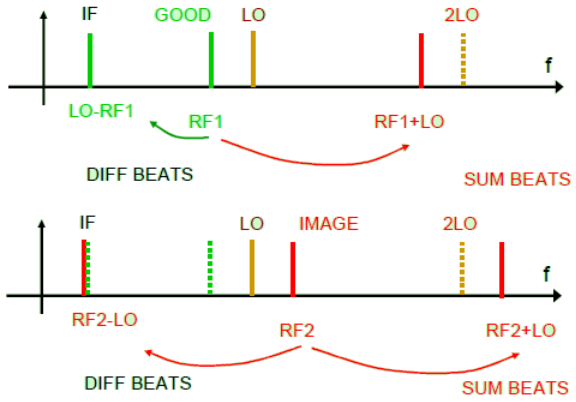
Where f is the middle of the spectrum of the signal. Let's now remark the well-known Werner's formulae:

$$\sin f_1 \cos f_2 = \frac{1}{2}[\sin(f_1 + f_2) + \sin(f_1 - f_2)]$$

$$\cos f_1 \cos f_2 = \frac{1}{2}[\cos(f_1 + f_2) + \cos(f_1 - f_2)]$$

$$\sin f_1 \sin f_2 = \frac{1}{2}[\cos(f_1 - f_2) - \cos(f_1 + f_2)]$$

This formulas are very useful in order to understand what happens when two sine waves are multiplied: the output of the multiplier is composed of two terms: one with frequency $(f_1 + f_2)$, the other with frequency $(f_1 - f_2)$. All this terms are multiplied for $x(t)$, the signal in base-band (centred on 0 Hz).

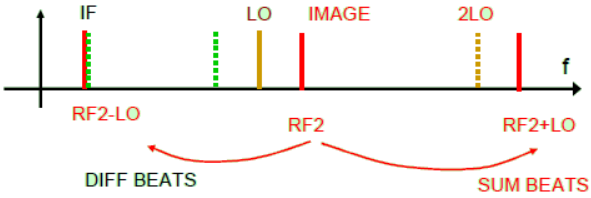


If there is a $x(t)$ signal, by multiplication we can translate its spectrum: ignoring (in all of this text) the $(f_1 + f_2)$ term, the final frequency of the signal it's $(f_1 - f_2)$; tuning the local oscillator to a frequency in order to be, respect to the f_{rf} term (the frequency of the signal out of the antenna filter), in a f_{IF} frequency for what:

$$f_{LO} - f_{rf} = f_{IF}$$

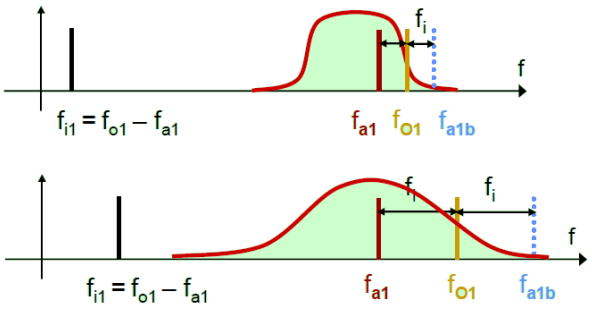
The multiplication will generate a signal translated in f_{IF}. f_{IF} is the Intermediate Frequency: that is a **fixed** frequency set by the designer, where the multiplication block must shift the former signal, naturally only after have well set the f_{LO}, the frequency of the Local Oscillator. The variable parameter is not the final frequency or the frequency of the channel, but only the frequency of the local oscillator, before the multiplication realized by the mixer. The multiplication do the separation, so, in f_{IF} we can design a band-pass filter easily and with better parameters: design a filter in a fixed frequency is better than in variable frequencies, because by this way we can obtain good parameters (like the quality factor, Q, quantifying the selectivity of the filter: with high Q, band is narrower).

Problem: this technology is based on a system that can shift signal spectrum if it is far from the oscillator frequency exactly $f_{RF} - f_{LO}$. There is another critical frequency: a frequency for whom $f_{IF} = f_{LO} - f_{RF,2}$: in this case, if there is some spectral content on this frequency (the symmetric frequency respect of the f_{LO}, the frequencies shifted are two: the good one (f_{RF}) and the bad one ($f_{RF,2}$): the second one is usually called *image*: if the local oscillator is exactly between two signals with the same frequency difference respect to the f_{LO}, the mixer will take two signals instead of one.



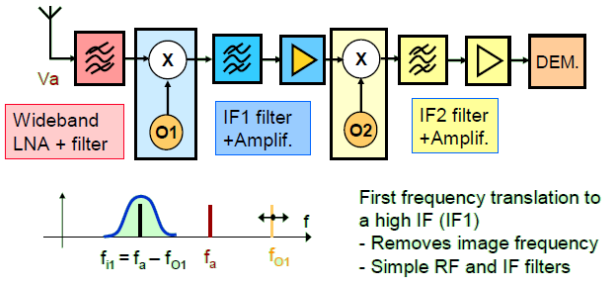
How can we handle this? There are many ideas in order to do this:

- The first idea can be the following one: use a radiofrequency filter (the one which follows the antenna output) with narrow band we can erase part of the image frequency; there are two sub-possibilities at this point:
  - If we use a big f_{IF}, so consider large frequency differences, we can use the effects of the radiofrequency filter and have reduced spectral components for the signal far away from the good one (including in this components the image frequency one!);
  - The previous sub-point is interesting but also has a problem: high values of f_{IF} cause problems in designing of the IF filter: design a narrow-band filter (with good parameters) in high frequencies is very difficult, so we remove images, but we can't obtain an excellent selection of the channel.



This problem appears also in the first part of this idea: this idea is based on introducing a narrow band (not very narrow) in radiofrequency spectrum, and this is very very difficult (and so expensive) to have. This idea is realizable, but not very good. The parameter that quantifies the qualities of a filter is the factor quality parameter, Q: it is very difficult to have high Q (and so narrow band) in high frequencies. A last note: for the radiofrequency, with this idea, the equivalent system is like having a moving filter (IF filter); from the point of view of the filter, it's like to have a shifting spectrum; what really happens is different from both the points of view: the local oscillator with its frequency it's multiplied for the signal, and due to the Werner's formulae we know that there is a shift of the signal.

- Filtering and design filter is hard, expensive, so often it's not the good way to take: from one side we want high IF frequency, from the other side a low IF (in order to increase Q). Can we get something that makes everybody happy? The answer is yes: **do the frequency translation twice: dual converse**.



The first IF is in a high frequency, the second one in a lower frequency respect of the first one, in order to have a narrow band filter easy to design and removing (with the first IF) image effects. There is, after the first translation, another possibility for images; the filter in the middle of the two mixers is useful for this reason: remove components or images before the second translation.

There are systems with three of four conversions; the problems are for the filters, cause give a good shape to the filter can be hard, so expensive.

In order to realize filters, there are different ways:

- LC circuits, so electronic resonators (already seen);
- Mechanical filters: electronic circuits with mechanical elements that can have filtering effects, like LC resonators.

Mechanical filters

Some years ago in high quality audio system the best way in order to realize filtering effects was the use of SAW, so of mechanical filters; SAW (surface acoustic wave) filters are electromechanical devices commonly used in radiofrequency applications. Electrical signals are converted to a mechanical wave in a device constructed of a piezoelectric crystal or ceramic; this wave is delayed as it propagates across the device, before being converted back to an electrical signal by further electrodes. The delayed outputs are recombined to produce a direct analog implementation of a finite impulse response filter. This hybrid filtering technique is also found in an analog sampled filter. SAW filters are limited to frequencies up to 3 GHz.

With quartz filters there are small metallic boxes that contain quartz oscillators: they are the base block for almost every good oscillator/tuned circuit. This can be used in order to realize also narrowband filters.

SAWs use ceramic materials, and are less expensive of the quartz lattice filters, but can produce something similar. The RF filters for the cellular phone use this technology.

1.1.2 Complex filters: SSB

There is another way to relax the specifications of filters, using the theory hidden in the Werner's formulae; as we already remarked before:

$$\sin f_1 \cos f_2 = \frac{1}{2}[\sin(f_1 + f_2) + \sin(f_1 - f_2)]$$

$$\cos f_1 \cos f_2 = \frac{1}{2}[\cos(f_1 + f_2) + \cos(f_1 - f_2)]$$

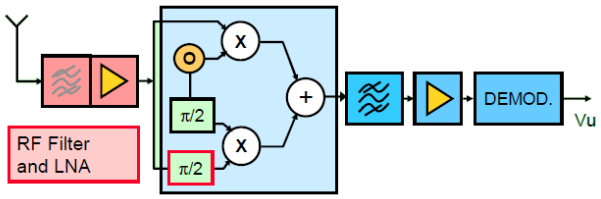
$$\sin f_1 \sin f_2 = \frac{1}{2}[\cos(f_1 - f_2) - \cos(f_1 + f_2)]$$

Now: we know that sine is a cosine shifted by $\frac{\pi}{2}$; what we can do now is use this relations and delete with the maths the bad signal: the image

frequency! If we have on one side the multiplication of two cosines, on the other side the multiplication of two sines, adding the results we obtain:

$$\cos f_1 \cos f_2 + \sin f_1 \sin f_2 = \cos(f_1 - f_2)$$

An architecture that can realize this idea is the same:



With the phase shift device we can change the cosine wave in a sine wave with the same polarity, so remove with the adder the image frequency without using filters. In order to have many channels in the spectrum, we can use SSB modulation (Single Side-Band): with a phase shifter and this technique we can obtain the SSB.

This architecture solves every problem? The answer is yes, but we have yet some problems: this architecture is used in commercial devices only since four-five years, because it requires very tight matching of gain and phase rotation of the signal; if this condition is not satisfied, positive and negative components don't delete by themselves and the system does not work. This can be done now with integrated circuits: although this technique exists since the World War II, it was for many years very expensive to realize, so useless.

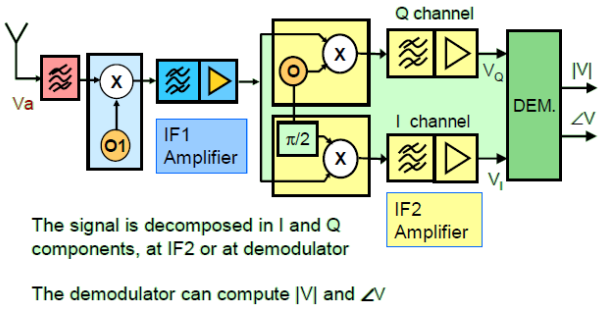
Let's analyse better this architecture: there are two groups of phase shifters: the left ones and the right ones. The left ones are critical: they must change in a wide band but with a great precision; with their, our local oscillator can generate both sine and cosine waves, over a wide range of frequencies; the other left-side phase shifter is connected to the output of the antenna, so it must work in a wide frequency range. The right one is on the difference, on the IF: at IF, so in a fixed frequency, the previous problem does not exists, so we must build a phase shifter that works on a single frequency, in a very narrow band (the filtered one), and this is easy.

Another block present in the architecture is the LNA (Low Noise Amplifier): out of the antenna there is a microvolt (or less) voltage, so noise has a little amplitude; due to this reason, we write about low noise.

I/Q demodulation

We have, after the previous subsection, sine and cosine (thanks to the phase shifter); an idea to realize a radio system architecture is use a phase/quadrature

modulation, so decouple the signal in two components, multiplying it to sine and cosine:



Zero-IF receiver

The idea of heterodyne architecture was shifting the signal spectrum to a frequency range lower than the former one; the idea of zero-IF receivers is moving the spectrum in base-band, using as center of the bandwidth the DC frequency: 0 Hz. The main difference is that we don't need a band-pass filter after this shifting, because it's enough the use of a simple low-pass frequency response; the low-pass filter can be realized easily respect to the band-pass one, and we can use op-amps.

There is also a problem: in zero-IF transmitter, DC becomes the signal, so there are problems with op-amps and other electronics because of the hardware offsets. There is another problem: if there is a noise at the same frequency of the signal, isolation for the local oscillator is impossible.

Another problem: the image frequency can exists, if there is a signal in the other side of the spectrum respect of the 0 Hz: shifting from f_{RF} to 0 Hz the signal, we will shift also the $-f_{RF}$ one, obtaining an overlap of the spectrum. For this reason, zero-IF cannot be realized with only one mixer; however, this is a good technique, because of the filter required: low-pass are very easy and cheap!

The presence of low-pass filters is a characteristic for this architecture: if in a schematic we see a low-pass filter, we can be sure that this is a zero-IF architecture, because it is the only one that can use this kind of filters.

1.2 Digital receivers

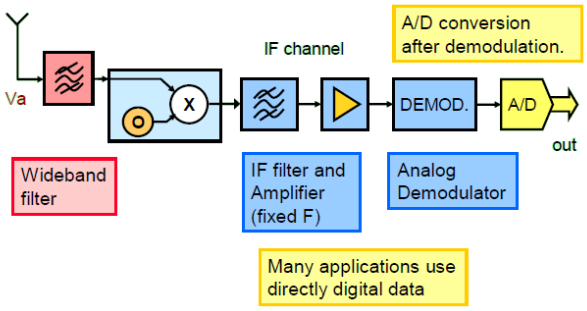
Until now we have studied the heterodyne structure for receivers and transmitters; we have learned what are the image frequencies and how to remove they by filters or some other ways; all the system studied have a common characteristic: they are all analog, so based on analog electronics. In this

years electronics focused their studies and researches on digital realizations: they are cheaper, useful for new media like satellites, and more insensitive respect to noise: due to the fact that there are only two possible levels, noise cannot disturb in a great measure the information and elaboration of the circuits; another reason: with digital electronics is possible to handle hard modulations or other function in an easy way, just by programming a DSP, a processor.

This section will explain how to realize a digital receiver in many ways, describing the differences between the various architectures. The block that realizes the transformation from the analog world to the digital world is the A/D converter (Analog to Digital converter): from the place in the block diagram where the A/D is inserted to the end of the block diagram, all the blocks will become digital.

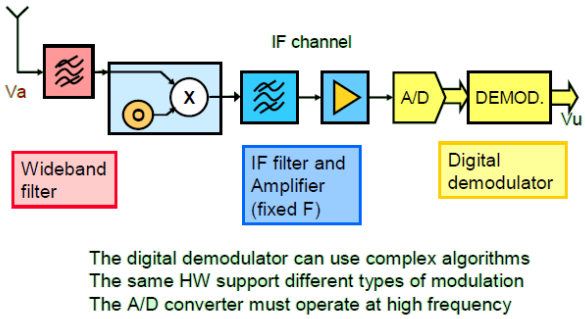
There are many ways to use an A/D converter; let's study all of them:

- First way: put the A/D converter out of the demodulator:



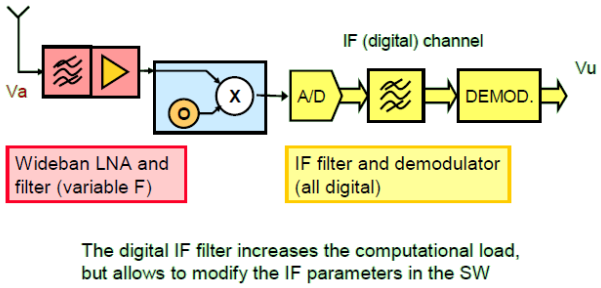
  The demodulated signal will be transformed from analog to digital; on digital we can do easily error correction, encryption or other functions that can be implemented on analog signals, but really hardly.

- Second way: A/D converter between output amplifier and demodulator:



With this architecture we can demodulate the signal with digital systems, so programming the processor! If IF is centred in a low frequency, the sampler can be simple to realize technologically; this is the first step in order to realize a software radio.

- Third way: put the A/D converter after the mixer:

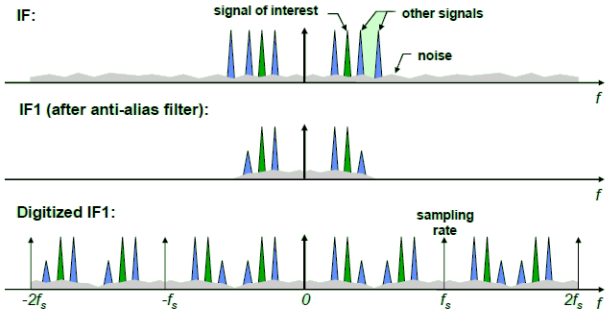


Well, the IF frequency is the same, but now the DSP must realize also digital filtering functions; design a digital filter is more easy that an analog filter: we must only program a processor instead of change values for capacitors or inductors. There is a drawback: we need a processor which requires more power respect to the previous situation: for filtering function we require more computation power, so much more energy to provide to the system.

In order to have a good converter it must represent all the values out of the antenna; in order to do this, there are two ways to design the system: design an A/D converter with many bits, in order to represent correctly also the small values, or use a VGA (Variable Gain Amplifier): if the amplifier amplifies only the small parts of the signal out of the antenna filter, we can describe the smaller signals as the bigger one, with the same precision and without using any more bit.

Remarks about sampling

Let's remark now what happens when we sample a signal: every time we sample a signal, in the time domain we multiply the signal with a train of pulses; as known from the Signal Processing course, a train of pulses in the time domain has, as spectrum, a train of pulses. We have something like this:

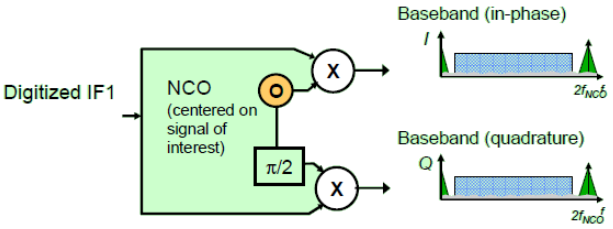


This picture can show the meaning of the Nyquist criteria: sampling generate replicas of the spectrum of the original signal, centred in the double of the base frequency, the triple, and so go on (back and forward); if we sample with a frequency smaller than $2f_B$, where f_B is the bandwidth of the signal we consider, we have aliasing, overlapping of the replicas of the spectrum. For this reason, before of the sampling process, we must use an anti-aliasing filter, realized by a low-pass filter, that erases all the spectral contributes after f_B, reducing aliasing effects.

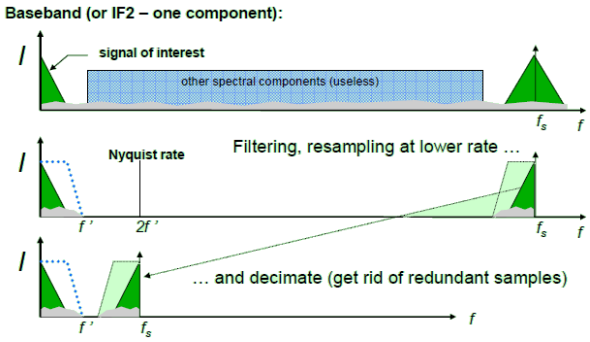
1.2.1 Digital architectures

Basing on this ideas, there are some architectures:

- Second conversion (with phase and quadrature): using two conversions, the digital signal we have can be treated as an image; the digital processor can be used in order to edit or elaborate digital images, using as components the phase and quadrature ones.



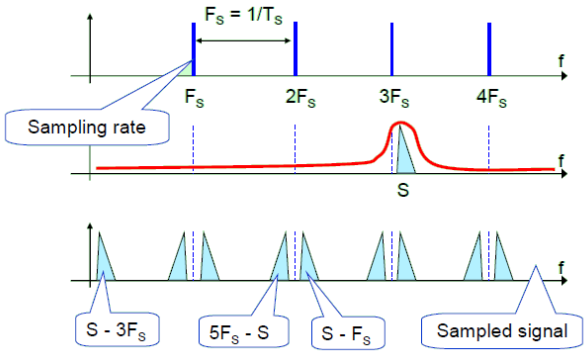
- Decimation: we can reduce sample rate in order to use less power:



If there is a spectral range not interesting for the elaboration, with a filtering process we can reduce the sample rate and obtain a better signal to handle.

Samplers as mixers

The most interesting application merit a dedicated subsection. An introduction: as known, the frequency shifting is realized by multiplying a cosine wave (or sine wave) to a signal, in order to obtain a signal centred in the difference of the two frequencies; we also remarked that a train of pulses in the time domain has as spectrum a train of pulses in the frequency domain; every pulse (Dirac delta signal) is a sine wave, as known from the theory of Signal Processing; we can say that a mixer can be replaced by a sampler, considering a good frequency: depending on the frequency of the spectrum of the delta train, we have different contributes; let's consider this example:



If the only spectral range interesting for the radio elaboration is the one near to $3F_S$, where F_S is defined as:

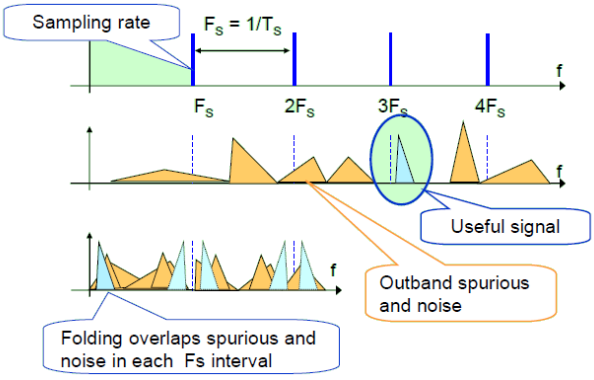
$$F_S \triangleq \frac{1}{T_S}$$

And T_S is the sample time.

The sampling process is so used also as frequency conversion process: changing the sample rate we can change F_S, so change channel and part of translated spectrum.

Obviously, we have a problem: this A/D conversion is not simple to design, because it works in radiofrequency; the only device before it is the anti-alias filter, that removes RF noise (LNA).

Sampling (or oversampling) with respect of Nyquist criteria, we can rebuild the signal. Previously we looked a particular case: the one with the useful signal in the middle of the bandwidth. As we seen, by multiplying for the 3rd delta signal we obtain $S - 3F_S$, so we same thing. A note: we don't sample at Nyquist frequency, because we don't consider the other spectral contents (before and after the useful signal). This case can be very useful if all the channels are in the center of the total bandwidth. By violating Nyquist, we obtain a confused signal, that cannot be treated with any kind of technology:



Here we have the filtering effect: by filtering, so isolating the set of the channels (the useful range of the frequencies) we subsample: we violate the Nyquist criteria, but we don't have problems, because of the elimination of all other elements; the final result is similar to the IF translation; after the translation, we can use a lowpass filter in order to rebuild the signal.

The important thing is the signal bandwidth, so the wideness of the channel bandwidth, **not** the carrier frequency: sampling can move the spectrum back as the mixer + oscillator, by subsampling. Let's understand one thing: the Nyquist criteria must be respected, but only in order to avoid aliasing problems in the final result: we subsample respect to the carrier frequency, not respect to the wideness of the set of channels: if the set of channels has a bandwidth large 300 MHz, we **must** sample at least at 600 MHz, the double of the wideness of the bandwidth. The individual channel so can be isolated with digital sampling, with techniques based on processor programming, using analog electronics only for the radiofrequency filter (the one that deletes

all the non-interesting parts of the spectrum).

The main idea for a radio system must be this: go to digital as soon as possible: to digital, all become easier. The best idea is undersampling (subsampling); with two images, as already wrote, we can do image cancellation with no filtering.

In modern devices or products, the radiofrequency filter it's large enough to keep all the channels; professional receivers have more filters, so depending on the situation they choose one or another; tri-band cellular phones are an example of this thing: three band that they can use, so three radiofrequency filters!

We wrote so much about receivers, but nothing about transmitters; this is not necessary, because the architectures of the transmitters are equal to the ones for the receivers, less then amplifiers: in receivers we use LNA, Low Noise Amplifiers; in transmitters PA, Power Amplifiers, that have, often, non-linearity problems.

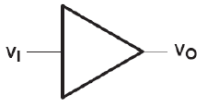
Chapter 2

Linear and non-linear use of bipolar junction transistors

In radio systems an engineer must use very often amplifiers: amplifiers are necessary every time we need to change the amplitude of a signal, increase its power or something similar. There are some ways to design amplifiers: with operational amplifiers, with transistors or with other devices. Operational amplifiers are very easy to use for designing, but they have a big problem: the bandwidth that they can provide is small; over some megahertz an op-amp can not work. Because of this, transistor amplifiers are the best way to design and realize an amplifier for radio systems: transistors can work with signals up to 10 gigahertz frequency, if the designer is skilled (but this boundary is shifting!). Transistor amplifiers are really useful in radiofrequency: near antennas or other radiofrequency/microwave devices, op-amp are totally useless, and the best solution is use transistors.

In this chapter we will study first linear models (small signal models) of the bipolar junction transistor, focusing in analysis and then in design (studying ways to choose amplifier gain, output voltage swing, amplifier bandwidth), so we will use non-linear models, in order to study other applications of the bipolar transistor, based on the Ebers-Moll model (and its exponential relation between voltage and current).

The general symbol of an amplifier is this:



In theory if we put in an amplifier a signal v_i we must have an output v_o like this:

$$v_o = A \cdot v_i$$

v_i and v_o must have the same shape, only re-scaled by a factor A. There are some non-idealities:

- In every electronic system there is **noise**: every block adds additive noise, $n(t)$:

$$v_0 = v_i \cdot A + n(t)$$

- The shape of the signal can change:

  the amplifier can have saturation effects or something similar: slew rate, phase distorsion or something else. In order to consider this effect, can be useful consider the Fourier series of the distorsion[1] of the signal:

$$v_0 = v_i \cdot A + n(t) + (v_e^2, v_e^3 ... v_e^n)$$

  v_e is the error signal in the system.

For amplifiers we need to have only the re-scaled term: we want to keep only this function:

$$v_o = A \cdot v_i$$

To obtain this, we will study ways to have better linearity; another part of this chapter will study applications of non-linearity, but later.

Now, we will have some remarks for some "famous" circuits, and learn to design on they.

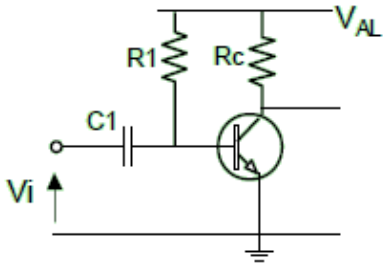
There are many types of transistors: BJT (bipolar junction transistors), MOS, FETs or other; on the small signal model, BJT or MOS are equal; while we move on large signal, there are some differences: for BJT there is the Ebers-Moll model, the well-known exponential relation, easy to analyze and use. For MOS transistors this is not true: there are many equations that describe the behaviour of the MOS in different work regions, or "operating points"; there are different models for every device and for every operating point we want to use. In this text we will study the maths for only BJT, because it's easier: BJT can be handled with maths, MOS only with simulators. There are tricks for control non-linearity and distorsion; we will study this tricks on BJT, but don't worry: they can be applied without problems also in MOS transistors!

[1] A little note: there are some functions, some devices, that need distorsion; mixers are an exemple of this functions.

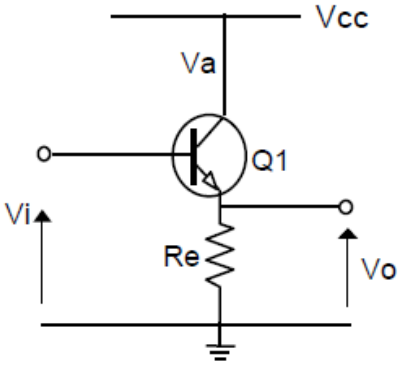
2.1 Topology and biasing selection

There are three topologies to realize single stage amplifiers; for *single stage amplifiers* we mean amplifiers realized with a single transistor; every stage takes its name from the name of the pin connected to ground (to the "common" pin). This three topologies are *common emitter, common collector* (or *emitter follower*), *common base*.

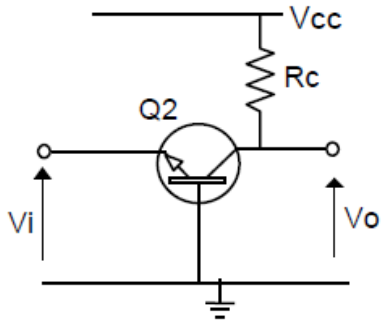
- The common emitter topology is the most important topology of the three: with CE we can amplify both voltage and current (especially voltage).



- The common collector topology is useful in order to realize a voltage buffer: the voltage gain is almost one, but the current gain is higher; this circuit has large input impedance and small output impedance, so it can be used to improve the CE topology, realizing a better voltage amplifier.



- The common base topology can be used in order to amplify very high frequency signals or to realize particular stages.

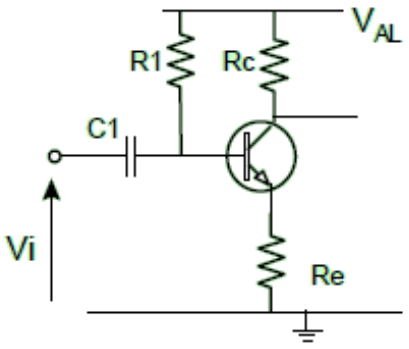


2.1.1 Biasing of the common emitter topology

The best topology to realize single stage voltage amplifier is the first one: common emitter topology. For every amplifier, the designer must choose some parameters; the first parameter is the operating point, or **bias point**. The question is: how can we set the operating point of the amplifier? The answer is quite easy; look at this picture:

The curves represent the different working point that the device (BJT) can use; there is a line, representing the values of voltage and current that can be assumed by a linear net (a circuit composed by resistors, capacitors and inductors: linear devices). Choosing the net we can choose the bias point of the amplifier in order to satisfy specs.

The first step is choose a supply voltage for the circuit; it can be named V_{al} or V_{CC} (the second one is interesting: the capital V means "DC voltage", and the CC means *supply voltage on collector side*). Connecting resistors (in order to have a small current) between the device and the voltage source, we can have a first circuit like this:



This is not a good schematic: the collector current is not fixed, because it depends on the β parameter (current gain); we have something like this:

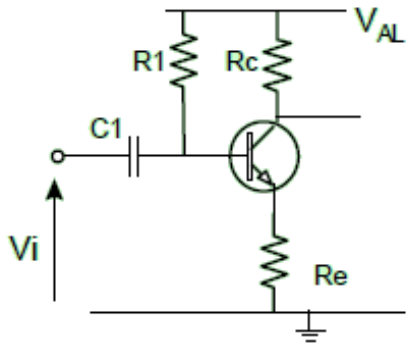
$$I_B = \frac{V_{CC} - V_{BE}}{R_B}$$

But

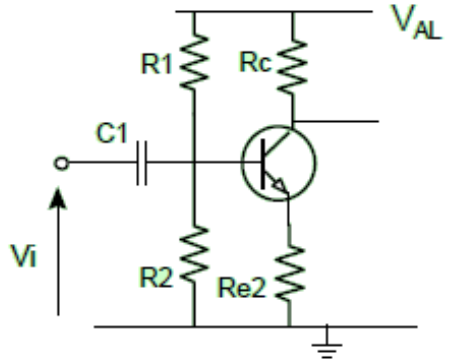
$$I_C = \beta I_B$$

Can we decide the operating point now? No! We don't know the exact value of the current gain β, cause we have a bad tolerance on this parameter.

Second step is introducing a resistor on the emitter, R_E: R_E works like a negative feedback, because if the current on it increases, increases the voltage on it; on the other side, if the voltage in the emitter increases, the other voltage values decreases (the V_{BE} voltage, that controls the operating point of the device), so the current gets stable.



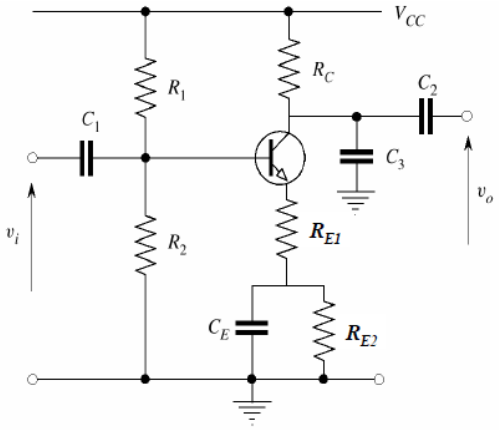
We want a good collector current; with this schematic there is still dependance by β; an idea can be the use of the famous *self-biasing circuit*:



Choosing the good parameters for this circuit, it can be proof that the voltage gain is like:

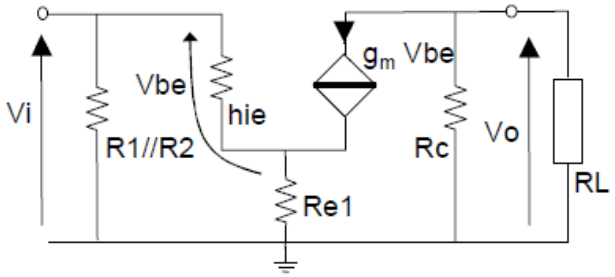
$$A_v \simeq \frac{R_C}{R_E}$$

If we use this schematic we have one more problem: if we want to choose the gain of this stage, we must change the bias poin; that's no good at all. How can we change this? Well, we don't matter if the operating point and the model of the circuit for small signals are different: we only need to realyze amplifiers! What we can do is use circuital (possibly linear) elements that can decouple some resistors from the other component of the circuit, in order to have a behaviour for the DC different to the behaviour for the small signals. The solution can be this:



Does this capacitor modifies the operating point? No! OP depends only on DC, but, after a transient, so in the steady state, the capacitor becomes full of charge, and it can be modelized by an open circuit. If the capacitor is big enough, it introduces a pole in the frequencies near the zero hertz limit, so it won't change the behaviour of the circuit in the good frequency range (where it must work as amplifier): it will be modelized with a short circuit, for frequencies higher than the pole's one.

Here is the equivalent circuit of the amplifier (in linearity):



Remember that now we are only interested about changes of voltage and current (signals): since we are analyzing changes, the DCs are not important.

On the emitter there is, as resistance, $R_1//R_2$; on the collector there is only R_C. The voltage gain can be calculated as:

$$A_v = \frac{v_o}{v_i}$$

Where

$$v_o = -g_m \cdot v_{BE} \cdot R_C$$

Because, as we can see in the equivalent model of the amplifier, $V_{BE} = v_i$. So:

$$A_v = -g_m \cdot R_C$$

This is the gain in this circuit, and this is not so good: gain depends on g_m, that depends on the operating point chosen for the circuit.

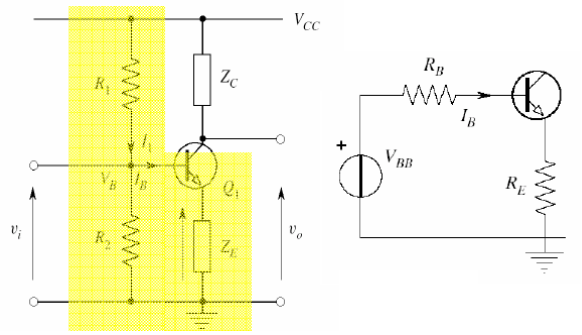
In order to end this section, a little remark: in DC, the capacitor is full of charge, so it's impedance is higher than all other impedances in the circuit, and it's modelizable as an open circuit; if the frequency of the signal is high, capacitors have an impedance smaller than the others in the circuit, so are modelizable as short circuits. This can be useful for selecting the gain without touching the bias point: due to linearity, we can decouple DCs and signals and their effects on the circuit, so, using capacitors, we can "show" to different signals different circuits, obtaining a frequency-dependent voltage gain.

2.2 Analysis of the circuit

At first, we want to analyze the circuit, in order to get formulas useful to design with it. We are interested only on the resistive part of the impedances Z_E and Z_C, so we will consider only R_E and R_C. h_{ie} and h_{fe} are some of the small signal parameters we are interested to study; in order to not use the circuit in a non-linear zone, we want $V_{CE} > 0,2$ volt: if we have a smaller collector to emitter voltage, the transistor works as a switch, not as an amplifier, and distorsion effects change the shape of the signal.

2.2.1 Analysis of the bias point

Beginning with the already shown circuit, we can use the Thvenin equivalent circuit, obtaining:



Where:

$$V_{BB} = V_{CC} \cdot \frac{R_2}{R_1 + R_2}$$

$$R_B = R_1 // R_2$$

Now, let's write an equation for the mesh of the circuit that don't go through the collector, obtaining:

$$V_{BB} = R_1 // R_2 \cdot \frac{I_E}{\beta + 1} + V_{BE} + R_E I_E$$

If the circuit is well designed, the first term (depending by β) will be near to 0; as known, a typical value for V_{BE} is 0,6 volt. I_E can be evaluated with this:

$$V_{BB} \simeq 0,6\text{V} + R_E I_E$$

With this and the other mesh we can evaluated the V_{CE} voltage:

$$V_{CE} = V_{CC} - R_C I_C - R_E I_E$$

If $V_{CE} > 0,2$ volt, we know that the circuit works in a good zone, so that we are far from the saturation area.

2.2.2 Bandwidth

In this text there was nothing written about amplifier bandwidth. A good designer must limit bandwidth for a reason: many bandwidth, many noise: the stochastic process more often used in order to modelize noise in electronic systems is the white gaussian noise, that exists in every zone of the spectrum; limiting bandwidth we can limit the incoming noise, increasing the performances of our system.

How can we control bandwidth? Somewhere this question was already answered: with capacitors! Capacitors (like inductors, elements no more used because very difficult to integrate) can show, as previously written, show to the signal different circuits depending on the its frequency, hiding or showing other elements like resistors or something else. Where must we put capacitors? Well, there are substantially two ways to use capacitors:

- Putting they on series to other elements: putting a capacitor in series introduces a transmission zero for this component, so a high-pass response; this can determine the low frequency response of the circuit or of a part of it;
- Putting capacitors on parallel (connected to ground) have a dual effect: the capacitor for low frequency is hidden by his big impedance, but for high frequencies he becomes a short circuit, so connect a pin to ground, and stop the signals on its way. This is a low-pass response: this way to use capacitors permit to determine and set the high frequency response of the circuit.

Sometimes we can see capacitances between base and collector of the transistor; they are not capacitors, but parassite capacitances; for now, we don't consider this on circuit.

2.3 A design example

Design an amplifier with the following specifics and schematic:

- Voltage gain $A_v = 15$ (nominal);
- - 3 dB bandwidth from 200 Hz to 20 kHz (minimum);
- output dynamic at least 4 Vpp on 10 kiloohm load (or higher);
- supply voltage 15 V (nominal);
- 2n2222A Transistor.

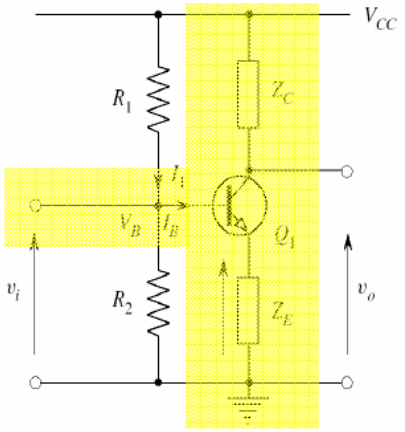
2.3.1 Resolution

Bias point

The operating point of the amplifier must be chosen in order to choose the output swing. There are many ways to start design, here is proposed one of

these. We want at least 4 volt peak-to-peak of output voltage swing with a load of 10 kiloohm. This indication is useless: bias point must be decided without introducing load in the circuit (in the beginning).

For the signal, the equivalent circuit is this:



When we connect the load to the circuit, we have a voltage divider. The idea can be the next (it's NOT the only way to follow!): V_u must be obviously higher than 4 volt, so let's suppose 8 volt, in order to have $R_L = R_C$: the divider becomes equal, and on the load we obtain half of 8 volt, so the 4 volt of the specs.

Now: with 8 volt of swing on the collector, remembering that the highest voltage on the collector can be 15 volt (the supply voltage); the minimum voltage on the collector is the difference of 15 and 8: 7 volt! Due to avoid saturation, we must impose a voltage between collector and emitter of almost 0,2 volt; we choose 1 volt in order to keep us far away from the saturation zone. Imposing 6 volt on the emitter we can guarantee that the threshold will be respected. What is the value of the current I_C ?

$$I_C = \frac{15 - 11}{10 \cdot 10^3} = 0,4\text{mA}$$

Voltage divider

What do we need now? Well, we now can calculate the ratio of the voltage divider on the base: we have the voltage on the emitter and the V_{BE} voltage drop, so:

$$V_B = 6 + 0,6 = 6,6\text{V}$$

This is one constraint of the problem; the other constraint is that we need the base current: we have to take account of the value of I_B in order

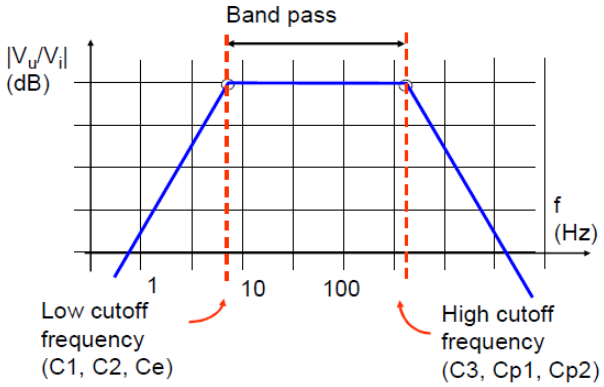
to calculated good values for the base resistances. Don't forget one detail: we don't know β, the current gain! Let's use a β_{min}, which can be found on datasheets; with $\beta_{min} = 50$:

$$I_{B,MAX} = 8\mu\text{A}$$

In order to make the voltage here not related with this current, if we make the current on the R_2 resistor much higher than 8 microampre, we are ok; let's not have a huge current: this can dissipate power on the resistors due to Joule effect. Choosing a current 10 or 50 times greater than the other, we are ok.

Voltage gain

For the voltage gain, found the bias point, we must study gain and its frequency response. What must we do now? Well, the frequency response of this system is something like this:



In the frequency range where the circuit works as amplifier, we must impose a voltage gain. This can be done easily:

- From the collector, we see a resistance of $R_C//R_L$;
- From the emitter, we see a resistance of R_{e1}.

So:

$$|A_v| \sim \frac{R_C//R_L}{R_{e1}}$$

Because C_4 is closed, C_3 is open.
There is a more precise formula:

$$A_v = \frac{v_o}{v_i} = -\frac{Z_C h_{fe}}{h_{ie} + Z_E(h_{fe}+1)}$$

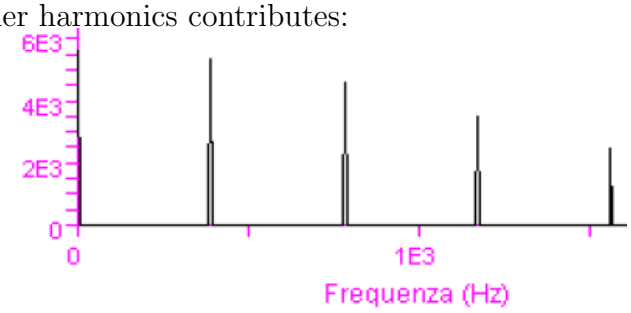
For the frequency limits, C_3 and C_4 must be calculated in order to put poles in the position requested by the specifics of the circuit.

2.4 Non-linear issues on transistor amplifiers

Until now we have analysed a basic transistor circuit to realize amplifiers, in a condition: the linearity. Having signals that can not go out of the output voltage dynamic; in linearity we use the small signal model:

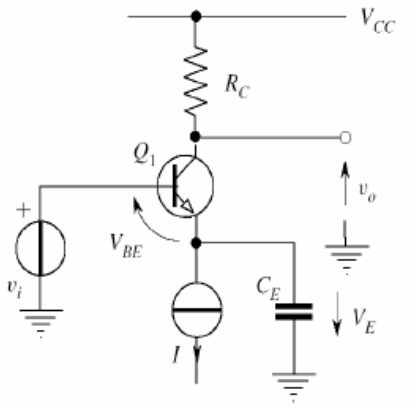
Out of the linearity voltage amplitude range, the analysis cannot be done with the linear model, but there is another good model to use: the Ebers-Moll model, taking account of non-linear effects:

If we put a sine wave in a system with this model, due to non-linearity out of the system we will not have a sine wave, because non-linearity generates other harmonics contributes:



Problems of non-linearity in radio systems are in power amplifiers, because signals treated with them is not small; in telecommuncation electronics, non-linearity effects can be good: some functions useful in this context (like mixers) can be realized with there effects.

The basic circuit that we will consider is the next one:



Biasing is fixed by fixing the current source on the emitter; supposing that we are interested on signals, variations, we will consider C_1 and C_4 closed,

C_3 open; we need something that put to ground the current for the signal but not for the bias, and this is the capacitor; choosing the convention for the voltage sign positive in the lower pin of the capacitor, supposing that the current generator is connected to a voltage reference (not necessary zero, because often we use negative voltages in the emitter). We introduce a sine wave on base (signal with no offset, because it has an average equal to zero), we know that on the emitter there are -0,6 volt (due to the v_{BE} voltage and because on base there are zero volt of DC). Supposing so that:

$$v_i = V_i \cos(\omega_i t)$$

$$I_C \simeq I_E$$

Then, using the Ebers-Moll equation:

$$I_E = I_S \mathrm{e}^{\frac{v_{BE}}{\eta V_T}}$$

Where I_S is the reverse saturation current, and $\eta = 1$ in transistors (due to technological reasons). Using down current convention, we can write:

$$v_i = v_{BE} - V_E \Longrightarrow v_{BE} = v_i + V_E$$

V_E is a DC voltage, so it's written with the capital. We can substitute and find:

$$I_C = I_E = I_S \mathrm{e}^{\frac{v_i + V_E}{V_T}} = I_S \cdot \mathrm{e}^{\frac{V_E}{V_T}} \cdot \mathrm{e}^{\frac{V_i}{V_T}\cos(\omega_i t)}$$

Defining the normalized voltage value x as:

$$x \triangleq \frac{V_i}{V_T}$$

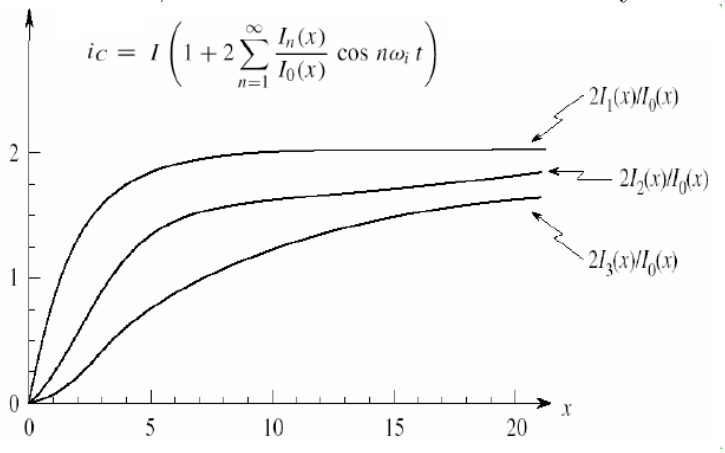
We can try to obtain a better equation to study. The critical term is the exponential of cosine; can be proof that:

$$\mathrm{e}^{x\cos(\omega t)} = I_0(x) + 2\sum_{n=1}^{\infty} I_n(x)\cos(n\omega_i t)$$

This means that voltage can be decomposed in different contributes depending by harmonics, where every harmonic contribute has a frequency equal to the ω_i frequency multiplied by an integer factor n. The coefficients of this series expansion depend on the modified Bessel function of first specie. In order to compute this values, we will use tables. Putting this expression into the old equation, we obtain:

$$i_C \triangleq I_C = I_S e^{\frac{V_E}{V_T}} \left[I_0(\frac{V_E}{V_T}) + 2\sum_{n=1}^{\infty} I_n(\frac{V_E}{V_T}) \cos(n\omega_i t) \right]$$

The first term (depending by $I_0(\frac{V_E}{V_T})$) is a DC term, so an offset term; with $n = 1$ we have the fundamental harmonic contribute, so with $n \geq 2$ the distorted terms, came out due to the non-linearity of the system.



Taking off of the parenthesis the I_0 term, we obtain the following relation:

$$i_C = I_C = I_S I_0(x) e^{\frac{V_E}{V_T}} \left[1 + 2\sum_{n=1}^{\infty} I_n(x) \cos(n\omega_i t) \right]$$

So the plot shows the various contributes related to $\frac{2I_n(x)}{I_0(x)}$; a little remark: DC term is I times 1; I is the DC current. By now we will use the following equation in order to represent the total collector current:

$$i_C = I \left[2\sum_{n=1}^{\infty} I_n(x) \cos(n\omega_i t) \right]$$

ω_1 is the frequency (pulsation) of the fundamental harmonic, and it depends on $I_1(x)$; $\omega_2 = 2\omega_1$ depends on $I_2(x)$, and is the frequency of the second harmonic, the first generated by the non-linearity of the system. With changing of n, the graphs show many contributes for the harmonics.

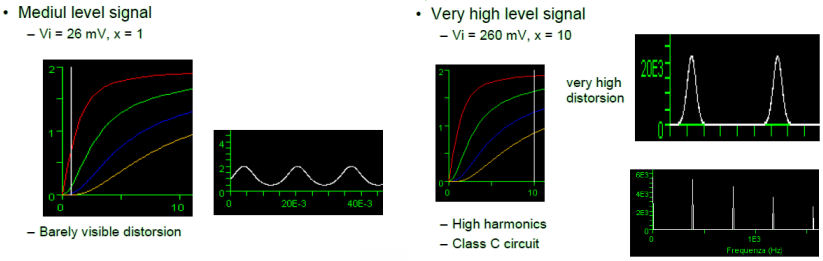
Let us assume for a moment that we are using a linear model; in small signal model we wrote:

$$i_C = g_m \cdot V_i \cos(\omega_i t)$$

Let's define k as

$$k = g_m \cdot V_i$$

We can write, in the linear model, that $i_C = k\cos(\omega_i t)$, where k is the gain of the system; in this case the gain is constant, so the characteristic of the system is a line; this equation represents only the coefficient for $n = 1$, so the linear coefficient: proportional increase of the amplitude of the output with increasing of the input amplitude.



The plot represents the non-linear model of the system; this drawing shows that if we increase the input level output (so increasing x, supposing that V_T is constant), we obtain for low amplitudes of x a linear response, so a constant gain (we can think that gain is the slope of this curve), and the proportional increase of the amplitude; going to higher level of amplitude of the input signal (so by increasing x), we have no more a linear increase, due to saturation effects of the transistor.

The first part of the model is almost linear: this is the zone where linear model can be used; note that with $x = 1$, $V_i = xV_T \sim 26$ mV (about 26 millivolt, but it depends on the temperature); 26 millivolt is a very small signal: if $V_i = 260$ mV, we are surely in saturation zone! The small signal model works only with small signals, so few millivolt of amplitude! The linear model can't predict the saturation effect because linearity does not generate other harmonics (as known from Signal Processing Theory): a linear model can produce only signals with fundamental component.

Now, considering (as already written) that

$$I \triangleq I_S \mathrm{e}^{\frac{V_E}{V_T}} I_0(x)$$

We can observe something: I is a DC current (so fixed), but it depends on V_i: if we change V_i (and so x), we can change I_0 output value; V_E is a hidden function of x, so I is really fixed, but some particulars were missing. There is a logarithmic dependence between V_E and x:

$$V_E = V_T \ln\left(\frac{l}{I_S I_0(x)}\right)$$

There is a fact: DC value depends on the amplitude of the input signal: this can happen because we are using a non-linear device, so DC can be modified by signals (this is obviously impossible in linear devices/models!).

We are interested on the output voltage:

$$v_C = R_C i_C$$

If we look only part of spectrum that don't consider DCs we can ignore them; this can be done by studying only the second term of the equation:

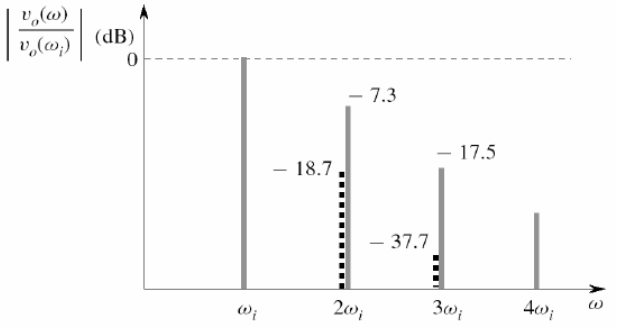
$$i_C = I \left[2 \sum_{n=1}^{\infty} I_n(x) \cos(n\omega_i t) \right]$$

So we don't consider the 1 in the parenthesis, ignoring DC terms and looking only at variable terms.

If we look at $\frac{I_1}{I_0}$ curves, we can see that gain decreases as the input amplitude increases; there is a phenomena called *gain compression*: gain is (as already written) the slope of V_o on V_i; this slope decreases as V_i increases; if V_i is high, spurious harmonics have a greater contribute respect to the main one, so gain decreases because the system becomes less linear!

Little exercise

How can we know how many non-linearity is in the system? In other words, how can we quantify the contributes of spurious harmonics respect to the fundamental one? Let's understand it with the following exercise:



Given a transistor amplifier with input V_i, output V_o:

$$V_i = 13\text{mV}; \quad Z_C = R_C$$

The second hypothesis is useful because we don't have to consider the dependence of the output respect to frequency; the question is: draw spectrum of V_o in dB_c.

Introduction: decibel (dB) is a measure unit for ratio; dB_c is a *carrier dB* unit: we calculate the ratio respect to the carrier, so respect to the harmonic with the fundamental frequency.

Our goal is to calculate the second and third harmonics contribute respect to fundamental one; we want:

$$\left.\frac{v_o(\omega_2)}{v_o(\omega_1)}\right|_{dB} \Longrightarrow \left.\frac{I_2(x)}{I_1(x)}\right|_{\mathrm{dB}}$$

$$\left.\frac{v_o(\omega_3)}{v_o(\omega_1)}\right|_{dB} \Longrightarrow \left.\frac{I_3(x)}{I_1(x)}\right|_{\mathrm{dB}}$$

So:

$$x = \frac{13}{26} = 0,5$$

$$I_1(0,5) = 0,4850 \Longrightarrow 20\log(0,4850) = -6,283\mathrm{dB}$$

$$I_2(0,5) = 0,06 \Longrightarrow 20\log(0,06) = -24.44\mathrm{dB}$$

$$I_3(0,5) = 0,005 \Longrightarrow 20\log(0,005) = -46.02\mathrm{dB}$$

So:

$$\left.\frac{v_o(\omega_2)}{v_o(\omega_1)}\right|_{dB} = 20\log(0,06) - 20\log(0,4850) = -18,15\mathrm{dB}_c$$

$$\left.\frac{v_o(\omega_3)}{v_o(\omega_1)}\right|_{dB} = 20\log(0,005) - 20\log(0,4850) = -39.74\mathrm{dB}_c$$

If values are not good, like $x = 1,54$, we must do linear interpolation between the two near values (in the Bessel function's table).

There are two ways, two approaches in order to treat non-linearity:

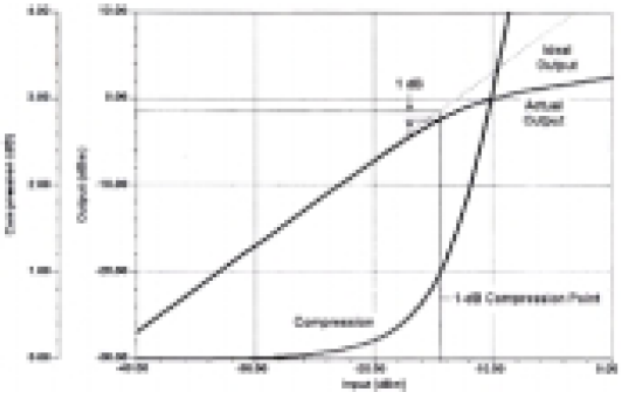
- Fight it: we can remove harmonics using resonant circuits or tuned amplifiers; by removing the harmonics, we have the same gain; of the linear-zone one.
- Use it: we can use harmonics in order to obtain frequency multipliers, VGAs or other particular devices (realizing particular functions).

2.4.1 Fight non-linearity : Compression

In order to fight non-linearity, we have to know it, so we must introduce some definitions that can be useful to study and avoid problems. The two terms we will introduce are useful because that can be found on datasheets or docs:

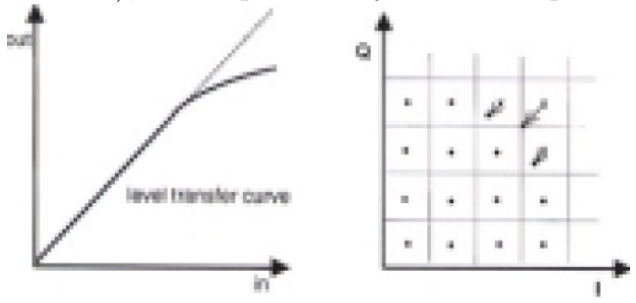
- 1 dB compression level
- IP (Intercept Point)

What is 1 dB compression level? Well, as already written, as the amplitude of the input signal increases, gain decreases;



This graph show how the difference of the linear model (small signal model) and non-linear model brings to have a 1 dB difference; 1 dB compression level is the voltage level that gives 1 dB of difference between the ideal output and the non-linear model output. At the begin, compression is zero, because for small signals contributes of 2nd, 3rd and other harmonics is almost zero; by increasing the voltage, that becomes important, so must be quantified (for example with this parameter).

One of the effects of compression related to radio issues can be studied on modulations: QAM is a Quadrature and Angle modulation (digital modulation), that represents symbols on a phase plane:



Compression can be critical because it changes the expected value of the amplitude of the signal, so informations are lost; if the modulation is only

on angle, like PSK (Phase Shift Modulation), compression is no problematic, because all the information is in the angle. The same thing can be written for analog modulations (FM or AM).

How to correct this kind of problems? Well, an idea can be predistortion: a unit P_A with distorsion can be compensated by introducing before it another block, with a non-linearity opposite to the one of P_A (example, put quadratic block before square-root block, logarithmic before exponential, and other...). We can put non-linear corrections in the digital part of the system:

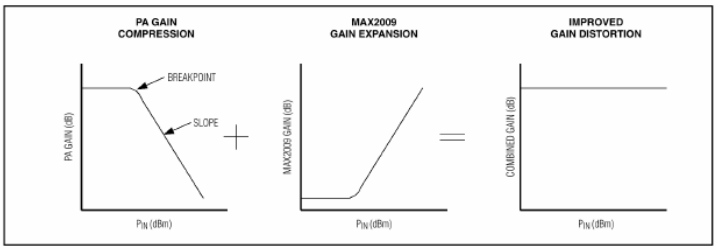


Figure 3. PA Gain Compression Canceled by MAX2009 Gain Expansion

A LUT can implement something like this. The idea is measure the power in the antenna, in order to introduce the non-linear correction, and modify the predistortion information.

2.4.2 Fight non-linearity : Intermodulation

We written about two parameters; one was the 1 dB compression level, already described; now we are going to define another phenomena very bad for circuits, phenomena that can not be fight: intermodulation.

We are reasoning on non-linear circuits, non-linear blocks; there are many ways to express the non linear output; a way can be, given a v_i input signal, consider the linear term, the quadratic term, the cubic term and so on:

$$v_i \longrightarrow Av_i + Bv_i^2 + Cv_i^3$$

An idea can be use power series expression. If we have a signal with frequency f_i, the linear term will be the signal with f_i frequency; the quadratic term will have a contribute with frequency $2f_i$, the cubic $3f_i$ etcetera.

What if our signal is composed by two parts, one with frequency f_a and one with frequency f_b ? Well, let's study it:

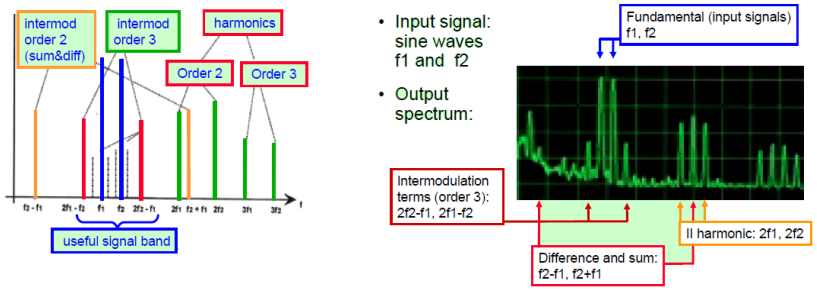
$$V_i = v_a(f_a) + v_b(f_b)$$

- The linear term will be a linear combination of the two terms:

$$v_{o,1}(f_a) + v_{o,2}(f_b)$$

- The quadratic term, so the second order term, will have signals with frequencies $2f_a$, $2f_b$, $f_a - f_b$, $f_a + f_b$.
- The cubic term, so the third order term, will be obtained by developing the power of three.

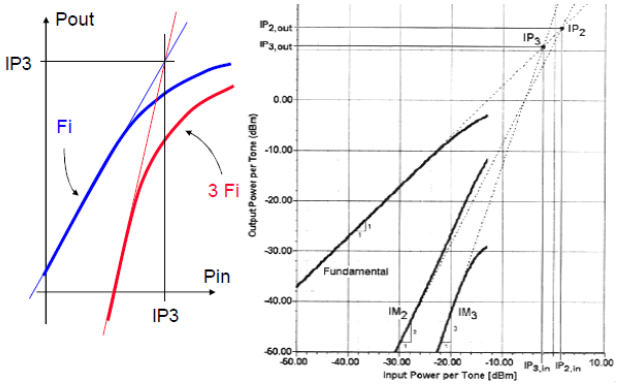
The problem is, with multi-component signals, that spurious harmonics go not only out of the original spectrum, but also **into** the spectrum, so cannot be filtered (without damaging the useful component of the signal). Inband terms can **not** be filtered, so we cannot get rid of them.



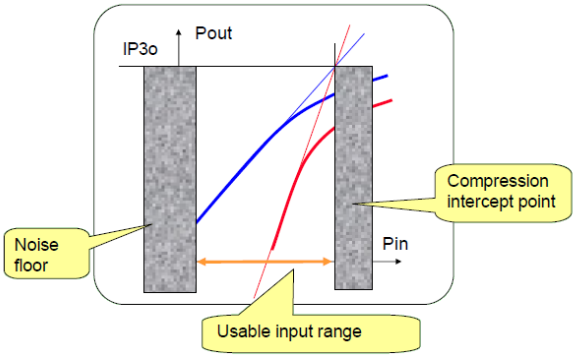
What is IP, and how is it related with intermodulation? Well, let's try to show it: if we have v_a and v_b with amplitudes V_a and V_b, if we increase their values (they are input signals!) until they have values $2V_a$ and $2V_b$, we will have something like this:

- The first order term output will be multiplied by two, like we can expect;
- The third term, so the cubic term, will be multiplied by 8: eight is the third power of two, so the third term harmonic will increase faster than the fundamental: if we increase input level, we can have problems like this, problems that can not be resolved.

The IP3, Intercept Point related to 3rd harmonic, is the intercept point of the linear prosecution of the small signal model, and of the line that can show how third harmonic increases it's level respect to the other.



Second order terms are not important, but third order terms are very dangerous, critical. IP3 can be used to study the dynamic of an amplifier:



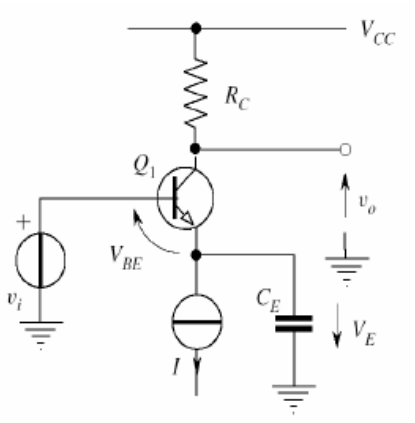
Going too low with amplitudes we confuse them with noise; too high, noise it's generated due to compression and intercept point.

In receivers (or transmitters) which is the effect for this phenomena? Well, in transmitters, we generate interference to other channels! In LNA (receivers) we separate channel with IF, but we have too strong components in a part of the spectrum, so due to very strong transmitters (for example, too near to the receiver) we have problems like this.

Chapter 3

Applications of non-linearity

Let's consider the following circuit:



With small signal analysis, we had:

$$v_o = -g_m R_C v_i$$

Now, we want something similar for large signals:

$$v_o(\omega_i) = -G_m(x) R_C v_i$$

This is referred **only** to the carrier v_i; this analysis can be used also on tuned amplifiers.

Instead of g_m we use G_m: large signal transconductance; it depends on operating point and value of signal amplitude. As known, for the large signal model, the output voltage is:

$$v_o(\omega_i) = -R_C I 2 \frac{I_1(x)}{I_0(x)} \cos(\omega_i t)$$

We want something formally equivalent to $G_m(x)$, so we can observe that:

$$v_i = V_i \cos(\omega_i t) = xV_T \cos(\omega_i t)$$

$$\implies \cos(\omega_i t) = \frac{V_i}{xV_T}$$

So:

$$v_o(\omega_i) = -R_C I2 \frac{I_1(x)}{I_0(x)} \frac{V_i}{xV_T}$$

Let's observe, now, that:

$$\frac{I}{V_T} = g_m$$

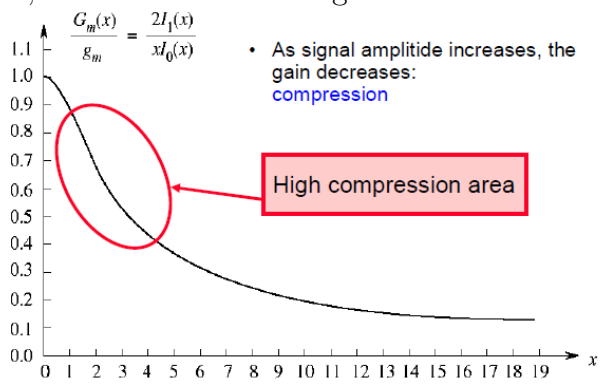
So we found a transconductance; now:

$$v_o(\omega_i) = -g_m R_C 2 \frac{I_1(x)}{I_0(x)} \frac{V_i}{x}$$

So

$$G_m(x) = g_m \cdot 2 \cdot \frac{I_1(x)}{x \cdot I_0(x)}$$

We can calculate and measure the gain for different signal amplitude values, in order to find that gain decreases as amplitude increases:



Where to use this? Well, the first idea can be... VGA: Variable Gain Amplifiers! If transconductance changes with signal level, we change gain.

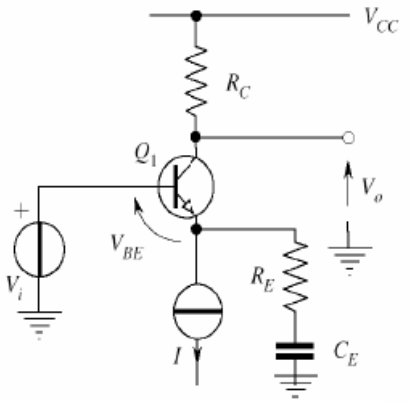
How can we use it? In a FM receiver: we don't want that amplitude change in the receiver, so VGA can be useful! In FM receivers there is a chain of amplifiers that works on this idea. If the receiver is moving, we can obtain some thing with almost the same amplitude.

These are compressing amplifiers (for compressing receivers).

And for AM? Compression can be useful because information is in the amplitude; we important part of the amplitude is the relative amplitude respect of different times, but amplitude can change due to movements of the receiver respect of the transmitter (like in cellular phones!). We have something good: signals we consider are in audio band, so 20 kHz; by using high time constant amplifiers, we can measure the power of the signal for a long time, so see changes due to movements and introduce average corrections: changes of averages are very slow respect to the frequency of the signal, so amplifying with high τ can be great in order to realize AGC (Automatic Gain Control).

3.0.3 Amplifier with emitter resistance

We already showed a model that can represent the gain behaviour with large signal, for the *circuit without* R_{e1}: a circuit where emitter capacitor hides for the signal both the emitter resistors. In order to increase stability of the amplifier (where stability is paid with gain decrease and with a harder mathematical model) we can use and study the next circuit instead of the previous one:



Now we will analyse this circuit. In the previous circuit we had that $v_{BE} = -v_i$, because there were no voltage drops (capacitor short-circuited emitter to ground, for the signal, so the only voltage drop was the junction one); now, in this circuit, there is current on $R_{e1} = R_E$ (the only emitter resistance that signal can see), so we must write an equation in order to identify only the useful part, the one that modifies the output voltage. This part, this variable, is the v_{BE} voltage. Let's consider the following hypothesis: we are considering large signals, but only the fundamental component, not the other harmonics; so, for the ω_i harmonic, we have:

$$v_i(\omega_i) = v_{BE} + v_C = v_{BE} + i_C(\omega_i)R_E$$

i_C can be written using the large signal transconductance; we formerly defined x as the V_i normalized by the equivalent voltage for temperature, V_T; this old relation is no more useful, because now there is R_E resistance: all the input voltage is no longer applied on the base-emitter junction.

We can do that:

$$x' = \frac{v_{BE}}{V_T}$$

Remarking that the voltage that controls the output, so the gain, is v_{BE}, not the v_C term. We can write:

$$v_i = V_i \cos(\omega_i t) = V_T x' \cos(\omega_i t)$$

but:

$$i_C = G_m(x') \cdot v_{BE}$$

Remember: transconductance must be multiplied by the only interesting part, so v_{BE}; the argument of the transconductance function will be x' instead of x, no more useful. We can write that:

$$V_i \cos(\omega_i t) = V_T x' \cos(\omega_i t) \left[1 + G_m(x')R_E\right]$$

We started from the fundamental component of signal on the mesh, so we defined how input and i_C are related. Remembering that:

$$V_i = xV_T$$

We have:

$$x = x' \left(1 + G_m(x')R_E\right)$$

This relation puts the normalized amplitude of v_{BE} (so x') in relation with other terms; the non-linear behaviour of the circuit is in $G_m(x')$, but respect to the previous equation, we have different argument to the non-linear term.

Now: what is the unknown in this problem? Well, x is the normalized value of V_i respect to V_T, x' is unknown! We can not set the voltage source to get x', and x' is in G_m, so we need it in order to have a good model for the amplifier. We need an equation in x', in order to obtain the normalized value of the base-emitter normalized voltage.

Can we invert the equation? Mathematically maybe it's possible, but it is quite complex; usually, we use a recursive approximation, like the same:

$$x' = \frac{x}{1 + G_m(x')R_E}$$

We have to solve it recursively:

1. The first step is consider $x' = x$: substituting x in G_m, we watch how much the equation is wrong;
2. The second step is: change values, trying to obtain something better;
3. Continue until the two terms have small differences!

The output (and so the gain) can be easily evaluated remembering that:

$$v_o = -G_m(x') \cdot v_{BE} \cdot R_C$$

So:

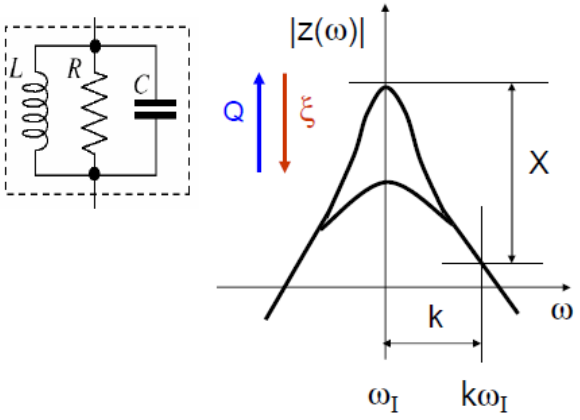
$$V_o = \frac{-G_m(x')R_E v_i}{1 + G_m(x')R_E}$$

This takes account for the fundamental of the non-linear behaviour of the circuit with R_{e1}.

3.1 Tuned amplifiers

For *tuned amplifiers* we mean amplifiers with a resonant (tuned) circuit as load. Keeping the circuit without R_{e1}, but putting a tuned circuit instead of the R_C, we have a load which impedance depends on the frequency of the signal we introduce in the circuit.

Some refresh on tuned circuits; considering a LC circuit, we now that:



Let's observe that, for low frequencies, the capacitor can be substituted with an open circuit, the inductor with a short circuit, so impedance is 0; for very high frequencies, the capacitor is like a short circuit, the inductor like an open circuit, so again impedance is 0. For the ω_0 frequency, where:

$$\omega_0 = \frac{1}{\sqrt{LC}}$$

We have infinite impedance, because positive and negative contributes of the impedances are equal, so they delete themselves.

This was ideal; in real world, capacitors have leakage resistance and inductance; inductors have resistance problems; the real equivalent circuit for a tuned circuit can be the following one:

This is the standard model used in order to represent resonant circuits.

The maximum value of resistance in the circuit is obviously R: when the reactances delete their contributes for the resonance frequency, remains only the R contribute.

Something else: if we have a load, the impedance goes in parallel with the tuned circuit, so R depends also on the load; if we plot the logarithmic graph, we have something like this:

With changing R, changes the shape, but not the frequency position of the peak.

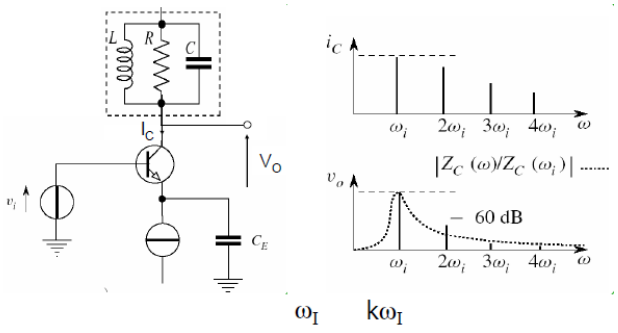
Now, what is important to know? Well, the ratio of impedance in a point respect to the peak; considering the resonance frequency ω_0, we will consider a frequency $k \cdot \omega_0$; remarking that there is symmetry only in the logarithmic scale, we can remember that:

$$Q = \frac{1}{2\xi}$$

Referring to k as a multiplication factor for the ω_0 frequency, there is an approximation for the attenuation of the impedance:

$$X = Q \left| k - \frac{1}{k} \right|$$

Now, let's put this circuit into our amplifier:



The collector current is controlled by base-emitter current; what is on collector does not modify i_C. We look at the collector voltage ($i_C \cdot Z_C$): voltage depends on collector's load. Every contribute for every harmonic is multiplied for the contribute of impedance in harmonic's frequency, obtaining something like this. Let's observe that shape is not symmetric. We have:

$$v_o(k\omega_i) = I \cdot |Z_C(n\omega_i)| \frac{2I_k(x)}{I_0(x)} \cos(k\omega_i t)$$

For this reason, there are two contributes to the non-linear behaviour: the well-known one, so the one previously calculated with Bessel's functions, and the new one, depending by $|Z_C(n\omega_i)|$; we have to quantify this term, in order to multiply it to the previous term. From now, let's suppose that the resonant circuit is tuned with $\omega_0 = \omega_i$, having so as resonance frequency the fundamental harmonic.

We can identify X as the ratio between values of impedance in ω_i frequency and in $k\omega_i$; we can write that:

$$X(k\omega_i) = \left| \frac{Z(\omega_i)}{Z(k\omega_i)} \right| = Q \left| k - \frac{1}{k} \right|$$

Having this, we can evaluate X in dB, so sum it to the Bessel's functions contribute and obtain the output voltage theoretical value considering the tuned circuit.

Let's try with an example; given the old 13 mV amplifier, let's calculate the attenuation (respect to carrier) of the tuned amplifier. We know that, with a resistive load (R_C), we had:

$$v_o(2\omega_i) = -18,3\text{dB}_c$$

Now, let's calculate $X_2 = X(2\omega_i)$, so the ratio between impedance on peak frequency and its double frequency:

$$X_{2,dB} = 20 \cdot \log_{10} \left(100 \left| 2 - \frac{1}{2} \right| \right) = -43,52\text{dB}$$

This term must be added to the original non-linear term, the $v_o(2\omega_i)$ previously calculated with the amplifier without R_{e1} and with resistive load on collector. We obtain:

$$v_o(2\omega_i) = -18,3 - 43,3 \sim -61\text{dB}$$

Why are we doing this calculations? What they mean? Well, we formally studied a circuit (the one *without* R_{e1}) that had only resistive load; we introduced a model of this circuit in order to define a frequency-dependent gain and to study the dependence from the amplitude of the input of the harmonic content of the output voltage. Now, we have a tuned circuit as load of the former amplifier; the reference condition for studying the amplifier is the old one, so the resistive load case: when our circuit works in resonance frequency (supposed equal to the fundamental harmonic, so to the input signal harmonic), inductive and capacitive reactances have equal contributes, so erase by themselves, and the equivalent load is already resistive; for this reason, at the resonance frequency, the two circuits are exactly the same. Introducing a ratio between the n-th harmonic impedance and the fundamental one is important due to use the old model: we already have techniques to calculate the reference harmonic values, so by introducing this ratio we can take account of the attenuation introduced by the resonant circuit.

The effect of introducing a tuned circuit instead of the collector resistive load is increasing the attenuation of harmonics, reducing the non-linear effects thanks to this increased attenuation: we keep intact the fundamental harmonic, and reduce the others!

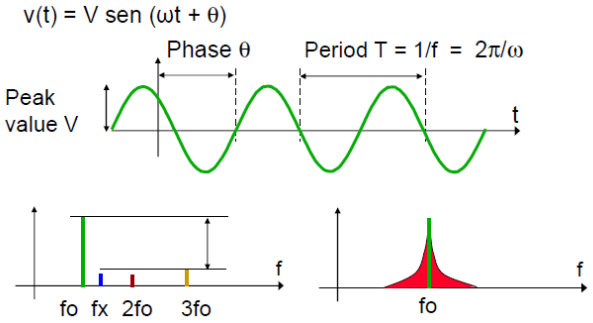
There are at least two ways to use this idea:

- Tuned amplifiers: amplifiers with a resonant circuit that can reduce spurious harmonics contributes;

- Frequency multipliers: given an input signal, if we put a resonant circuit tuned on $3\omega_i$ instead of ω_i, we have the greater level for the third harmonic, not for the first, because the harmonics are attenuated. This is a way to realize frequency multipliers. It can be used in order to transmit 1 GHz frequencies, by realizing a 200 MHz oscillator and multiplying its value. Another way to obtain the same effect is using phase-lock loops. The quality of multiplication depends on Q factor: there are sub-harmonics and super-harmonics, so harmonics in lower and higher frequencies respect to the resonance frequency: the quality of filtering process depends on Q factor (how much the filter band is narrow).

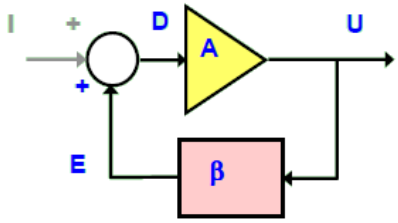
3.2 Oscillators

Now will be exposed an overview to a special exploit of non-linearity: the realization of oscillators. We will write about sine oscillators, taking account of some parameters: peak voltage, period, phase, spectral purity.



Spectral purity is the distance (evaluated in decibel) from the signal fundamental harmonic to the most high of the spurious harmonics.

If we want to build a sine generator we need to obtain a pulse, a δ in the frequency domain; this can be done with a general block schematic like this:

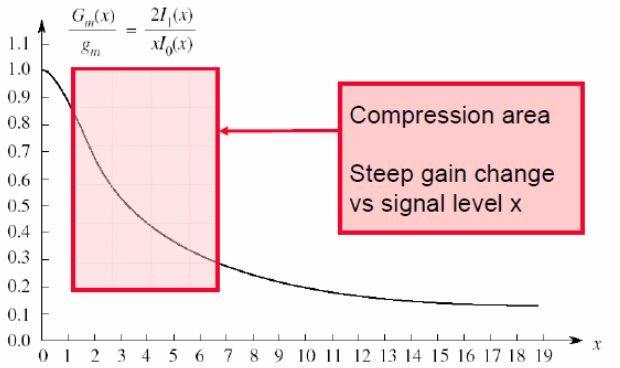


A remark: this is a positive feedback system, so a system where feedback signal keeps the same polarity of the output voltage signal. This system can oscillate if the Barkhausen hypothesis are verified: given loop gain $A\beta$, we need that $|A\beta| = 1$, $\angle A\beta = 0°$; with this condition, the signal which enters the loop remain exactly the same. We don't want the same signal, because we want a sine generator; in order to obtain out of the system a sine wave, we can make the conditions satisfied only for a specified value of frequency.

Problem: we obviously can not have exactly unitary module (or phase rotation of 0 degrees): components have tolerances, not exact values, so we need techniques that can realize this problem: if the gain is less then 1, the signal will decrease after a transient, if the gain is higher than one the system will saturate; we need some way to resolve this issue.

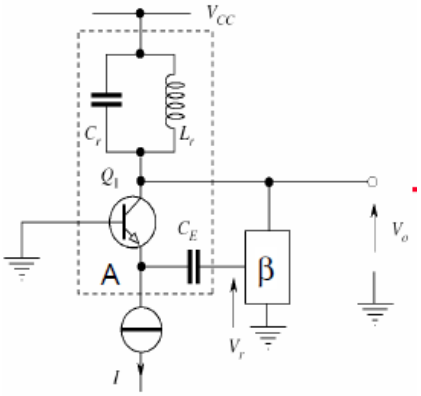
The idea is using gain compression: using the transistor compression area (**not** other areas), we can automatically make the gain stable: if amplitude is too high gain decreases, if amplitude is too low gain increases, and so go on); the only useful area is the compression one, in order to obtain a good

effect of this type.



Issues where two: the first one, already solved, was the modulus one; and phase rotation? Well, if phase shift is related with frequency, we can find a technique to obtain in some frequency values 0 phase rotation. This technique is based on resonant circuits: tuning a resonant circuit on ω_0, we can obtain a null phase rotation for our system. Changing the quality factor Q of the resonant circuit we change the slope of the phase rotation: higher Q, higher slope.

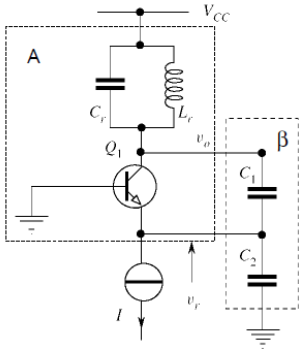
This system so can be realized with a tuned amplifier connected to a β feedback block:



The first idea cannot be used, because connecting the feedback block to the input we obtain a negative feedback, so our system does not work as an oscillator; the good idea is connect to ground the base of the transistor, and to the emitter the β block: this is a common base configuration! There is a gain similar to the common emitter configuration one, but no inversion of phase. By using capacitors we can decouple bias and signal, as usually done.

This is the fundamental scheme for oscillators based on transistor amplifiers: tuned circuits + common base amplifier. What can we use for β block? There are some ideas:

- Colpitts oscillator: by using a capacitive voltage divider as β, we obtain an oscillator.



Capacitances do not change frequency behaviour, because:

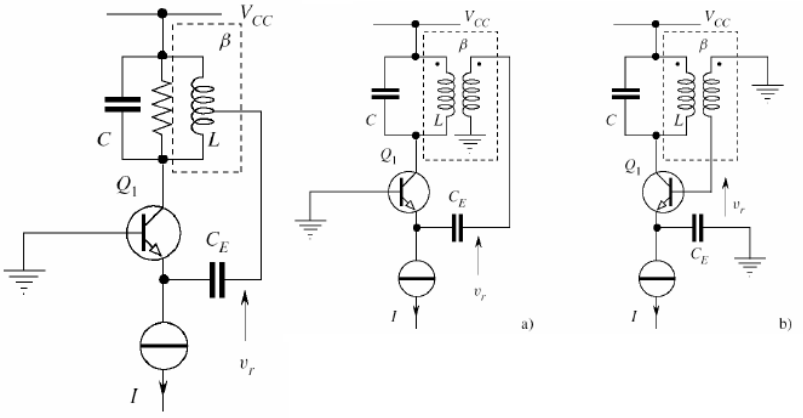
$$v_R = \frac{\frac{1}{sC_2}}{\frac{1}{sC_2}} + \frac{1}{sC_1}$$

All s terms are simplified, so there is no frequency dependence for the voltage divider capacitor. This is not true at all, because into the transistor we see, from the emitter, a resistance equal to $\frac{1}{g_m}$, where g_m is the transconductance; often we can ignore this fact, but we can solve also this problem (in fact it introduces, as we can proof, a dependence on frequency to gain) by introducing between β and emitter a voltage buffer circuit.

This can be seen as a two-stage system, or as a differential pair.

- Hartley oscillator: same circuit, same observations, with inductive voltage divider instead of capacitive voltage divider.

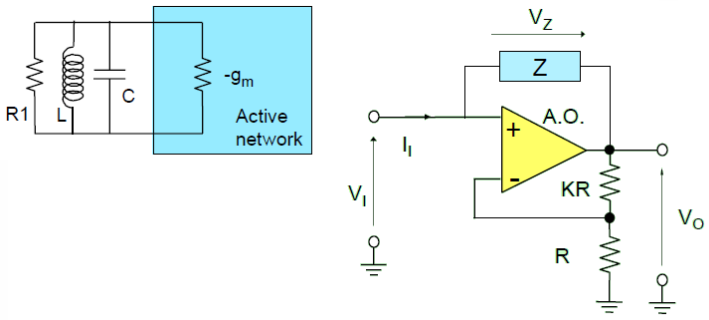
- Meissner oscillator: uses a transformer as feedback block.



3.2.1 Another technique for realizing oscillators

There is another way to obtain oscillators: RLC circuits. We can know, by calculating the transfer function of a resonant circuit, that if we introduce a current pulse (so a Dirac delta in time domain) we obtain as output a sine wave; if there is no resistance, there is no power loss in the circuit, so voltage continues to oscillate, obtaining a sine wave. In real circuits there is always some resistance term, so the output voltage has a transient that brings voltage to zero.

The idea can be the next one: if we cancel the resistor contribute with a negative conductance (realized by an active network), we can obtain an equivalent RC circuit, so an *ideal resonator*. This can be done with NIC (Negative Impedance Converter):



Using the well-known equations, we can write that in v_- there is the same voltage as v_+, so V_i; V_o and V_i are related by the voltage divider:

$$V_i = V_o \cdot \frac{R}{R + KR} = V_o \frac{1}{1 + K}$$

So:

$$V_o = V_i(1 + K)$$

By studying the mesh, we can write that:

$$V_Z = V_o - V_i = KV_i$$

We can quantify the input current I_i as the current on V_z, considering satisfied the hypothesis of null currents on the pins of the active device:

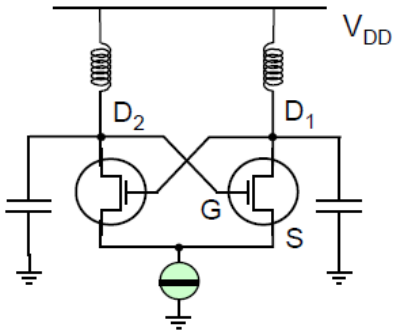
$$I_i = -\frac{KV_i}{Z}$$

We can so write that the impedance seen in the pins of this device is:

$$Z_i = \frac{V_i}{I_i} = -\frac{Z}{K}$$

We can do this trick, because on V_o we have a higher voltage of V_i. There is still a problem: we need to adapt the two resistances to the same values; we can use the non-linear behaviour of the system: if we make the V_o voltage decrease we can make Z decrease, so changes of voltage make gain and impedance compensate; saturation and other non-linear effects of this circuits can so be used in order to tune the two resistances (or conductances), in order to solve the problem.

Usually, this technique is realized with only transistors, in order to obtain high frequency oscillators, like this:



3.3 Logarithmic Amplifier

Now, we want to obtain specific non-linear behaviours, from some circuits. Given a non-linear transfer function shape, we want to obtain something similar to it, with some approximation. Now, what we will study is a circuit that can realize logarithmic transfer functions. Our transfer function must be as close as possible to the ideal one; obviously, using the circuit approximation, we will obtain some differences respect to the original case. There are two ways to obtain non-linear behaviours:

- Use a continuous approximation: obtain something similar to the original transfer function, approximating it with a continuous function.

  We have an *actual* transfer function, different from the *ideal* one.

- What we can also do is approximate the ideal transfer function with a set of lines:

  This is called piecewise approximation: there are many lines that approximate the shape of the original transfer function.

In our case, the transfer function behaviour is the logarithmic behaviour; with piecewise approximation we can obtain something better respect to continuous approximation, but with very complex circuits; we will use the worst approximation, but with simple circuits.

We will follow as usually a top-down approach: we will begin from the function, from the mathematical model, and then come to a circuit realization.

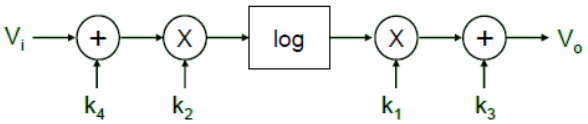
The beginning function is the next one:

$$V_o = \log(V_i)$$

So, we want to realize a system that has as output the logarithm of the input voltage. This is the simplest type of logarithmic expression, but we can introduce some other terms:

$$V_o = k_1 \log\left[k_2(V_i + k_4)\right] + k_3$$

We can have many effects, gains, offsets on input, offsets on output; k_4 is a critical (as we will see) term, because it introduces a shift on V_i. This operation can be represented with the following block diagram:



When we design the circuit we must understand what kind of circuit must be used, in order to treat every parameter correctly; parameters are four, so we can think to have 4 degrees of freedom. This is not true: we know that:

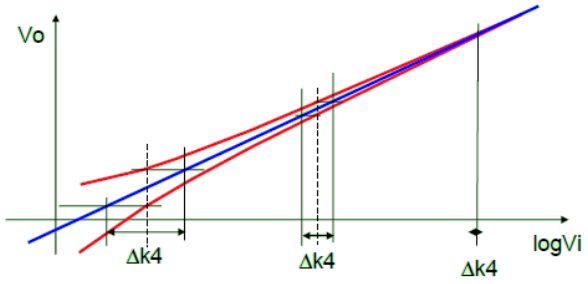
$$\log(AB) = \log(A) + \log(B)$$

k_2 introduce the same effect on transfer function that k_3, so the two are interacting.

In order to represent well a logarithmic behaviour, a good idea can be this: we can represent the x-axis with a logarithmic scale, so obtaining the same ratio with equal intervals. In logarithmic scale, a logarithmic function will be, obviously, a line:

If we plot a generic transfer function and we find how it changes when we change parameters, we have a line. When we change k_1, we change the slope of the line, we *rotate* it; k_2 shifts it horizontally, k_3 vertically. If we change k_4, we have something bad: changes of the shape are not equal in every zone, because if we change for example of 50 mV the amplitude of the input, it will become important for the low values (like 0,1 to 1 V), but

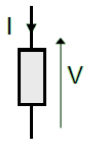
almost zero for higher values (like 10 to 100 V decay). We don't shift or rotate the characteristic, but we change its shape, depending on the point we consider! We will have an error very important for low values, and almost zero for high values.



Blue line represents an ideal function (with no offset or input changes), and Δk_4 is the effect of an offset: red lines are *actual* characteristics.

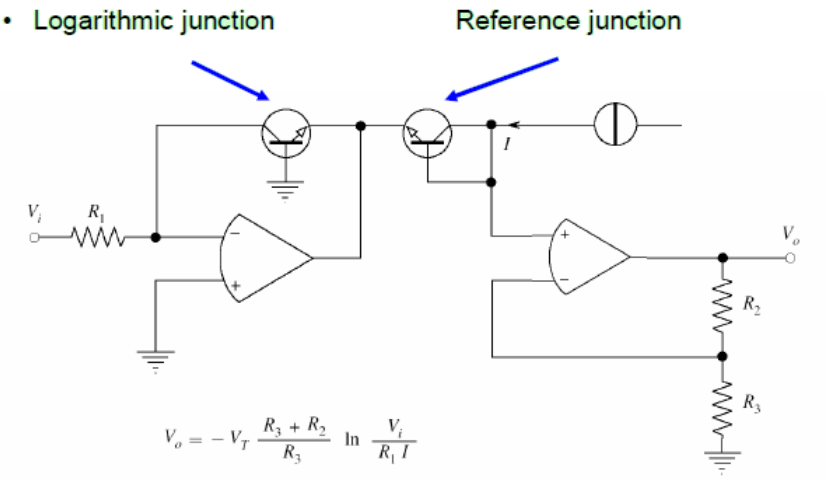
Let's now try to build a circuit with this transfer function; we need a *logarithmic core*, so something that can realize a logarithmic relation; this can be simply a junction, a diode:

$$V_{BE} = \eta V_T \ln \frac{I_E}{I_S}$$



$$V_d = \eta V_T \ln\left(\frac{I_E}{I_S}\right)$$

We have to send in junction a voltage that can bring the right I_E; we can force this current with this circuit:



The ideal operation amplifier can provide this thing; in fact:

$$I = \frac{V_i}{R} \Longrightarrow V_o = -V_D$$

Due to 0 V in the minus pin, and so in the plus pin; we have that:

$$V_o = -V_D = -\eta V_T \ln\left(\frac{I_E}{I_S}\right)$$

Let's compare this with the maths; we have that:

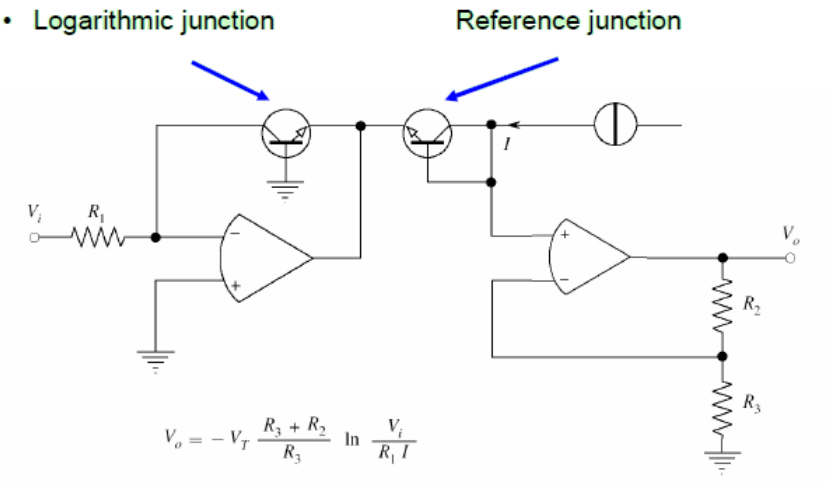
$$k_1 = -\eta V_T$$

With this circuit we can add an amplifier, in order to have a variable gain. k_2 is the coefficient of V_i, so:

$$k_2 = \frac{1}{RI_S}$$

Changing R we can change k_2, but it's no good: I_S has a strong dependence on temperature! k_3 can be realized by adding something, so by an adder. k_4 is an offset of V_i, so we can control or compensate it with any circuit.

As already written, we have strong dependences with temperature: there is η, V_T, I_S; can we improve this circuit by making some changes? The answer obviously is yes: first approach uses the following circuit:



We have that:

$$v_B = v_A + v_{b2} = v_A + V_T \eta \ln\left(\frac{I_E}{I_S}\right) =$$

$$= -\eta V_T \left(\frac{V_i}{RI_S}\right) + V_T \eta \ln\left(\frac{I_E}{I_S}\right) =$$

Due to the properties of the logarithmic function:

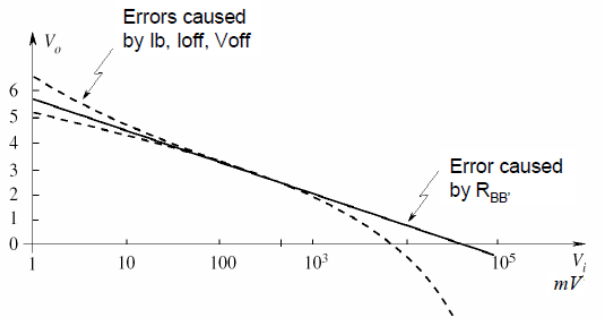
$$= \eta V_T \ln\left(\frac{V_i}{RI}\right)$$

This, obviously, can be written if the two junctions are identical. We have that:

$$v_o = A v_B$$

Note that V_T is still present; we can compensate V_T by making the gain of the amplifier related with temperature ϑ; this thing can be done with bolometers, with temperature-dependent resistors.

Let's remark another thing: curve has this behaviour:



If we don't change A with ϑ, current will depend on ϑ; change of ϑ implies change of k_1, so of the rotation of the characteristic. The center of this rotation is the point where there is no change of position due to the rotation; known that:

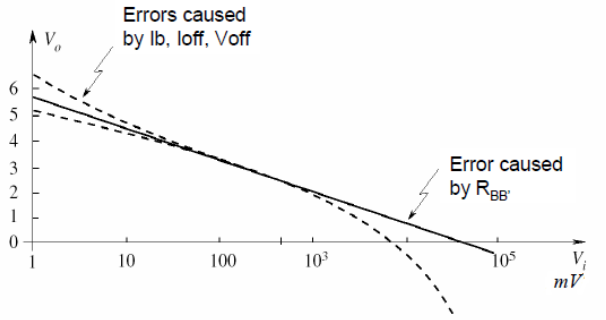
$$Y = V_T X$$

Where V_T is the changing term, the only point where V_T don't change Y is $X = 0$. Rotation is around the point corresponding to zero volt output; when the argument of logarithm is one, so

$$V_i = RI$$

We have $v_o = 0$, so there is the center of the transcharacteristic. This center should be placed in the middle of the characteristic, in order to minimize errors! Phase rotation are in fact more critical if rotation center is in the middle.

One more point: we said that k_4 is critical because it is connected to V_i; this is not the only issue regarding V_i value: with low values, we already watch that there are shape problems due to the logarithmic scale and input offsets; for high values, we need to use a better module of the junction (considering diodes as junctions): Giacoletto's model considers that semiconductors have a resistive behaviour far away from the doped part, so between the base pin B and the diode zone we have an $R_{BB'}$ resistance, that have values between the 10 ohms to the 100 ohms. The voltage drop error becomes important for high values of V_i, because by increasing V_i we increase the current on the junction, so introduce a linear term (resistive term) that will be added to the logarithmic one, introducing another voltage drop for high amplitude values of the input. This can be the real characteristic function:



Good thing: with transistors instead of diodes, this problem is not really important, because of this:

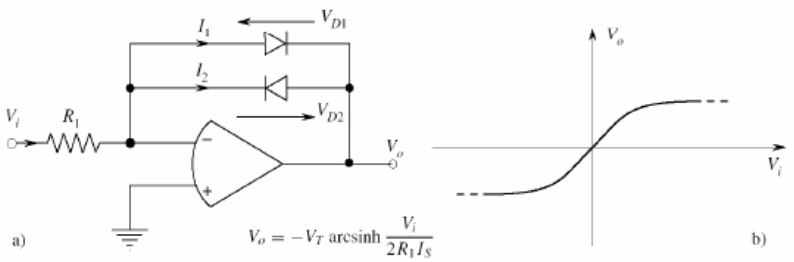
Current does not enter in $R_{BB'}$ because the base of the transistor is connected to ground, so we have less offset errors for high input voltages.

Another information: between the emitter of the logarithmic junction (left transistor) can be a good idea to introduce a resistance R; this transistor configuration is an amplifier, with common base configuration, respect to the feedback signal; it can increase the voltage, because the β parameter of the loop gain can be higher than 1, so bring up the op-amp transfer function.

The circuit can oscillate: op-amps are compensated in order to put the second pole in the point where there is unitary gain: the voltage follower configuration. Voltage follower configuration for an operation amplifier is the worst case: the most wide frequency spectrum, with the least voltage gain (unitary); if we shift vertically this transfer function, we don't respect the margin of stability introduced by the producer of the op-amp, so we can have a phase rotation of 180 degrees, and obtain a positive feedback instead of a negative feedback. Putting R we decrease the gain of the amplifier, obtaining the original second pole position.

3.3.1 Bipolar logarithmic amplifier

A problem of the circuit presented is the following one: it can handle only positive voltages, due to bipolar junction transistor saturation zone; we actually can have inversion of polarity, so handle negative signals, simply by changing the circuit this way:



Let's pay attention to this thing: we have a characteristic of the circuit that jumps the origin and has, for the other value (far enough from the origin), a logarithmic behaviour.

This amplifier is really useful: it can be used in many cases in order to treat signals with compression; in a receiver, for example, we need to compress signal due to avoid bad effects like saturation or frequency poles; after the treatment in a digital part of the receiver, it must be de-compressed; in order to de-compress, we must apply an opposite transformation, so an opposite non-linear compression; if the logarithmic amplifier applies a logarithmic non-linearity, with an exponential we can obtain the opposite result!

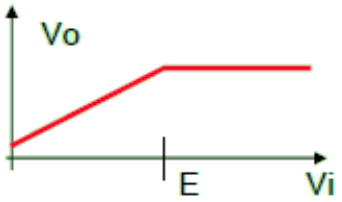
To realize an exponential transfer function we can use again diodes, but connected to the input instead of in the previous place, the feedback; current

is converted by the diodes, and feedback resistor converts current into voltage, solving our problem!

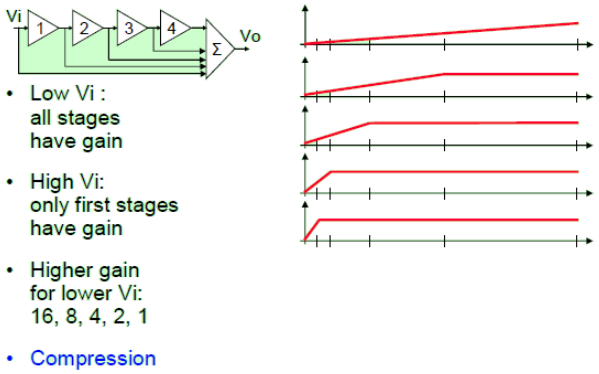
3.3.2 Piecewise Approximation

As already written, there is another approach in order to realize logarithmic non-linearity with electronic circuits: piecewise approximation. This type of approximation consists of realizing different shapes with lines, whose slopes depend on input voltage amplitude. This is the mostly used realization for the electronic logarithmic approximation, for example in RSSI (Received Signal Strenght Indicator: a device that measure power in antenna output); by using the logarithmic amplifier, we can measure and treat with the same resolution very low variations and very high variations, because we are in a logarithmic scale; in linear scale, this is not possible. In integrated circuits, like in mobile phones or radios, these are the most common techniques.

Given a chain of amplifiers, where every one has a characteristic like this:

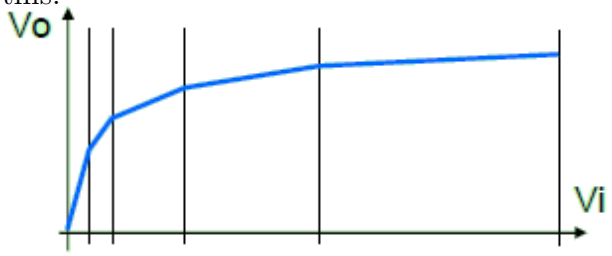


Every amplifier has this dynamics: for an input voltage from 0 to 100 mV there is a voltage gain of two, then only saturation; we have something like this:



In the first amplifier we introduce a signal v_i; out of it, there will be the already seen characteristic: from 0 to 100 mV gain of two, then saturation; the output of the first amplifier will be the input of the second one; when the first amplifier reaches 50 mV, out of it we will have 100 mV (due to the voltage gain), so the second amplifier will saturate; like before, the output

of the second amplifier will be the input of the third amplifier; when the second amplifier reaches 50 mV, so that the first amplifier reaches 25 mV (we are obviously supposing that all it's linear, before saturation!), the third amplifier will saturate. Adding this three signals, we will obtain something like this:



Between 0 and 25 mV we will have a total gain of 14: at the third stage the gain is equal to 8, at the second equal to 4, at the first only 2; adding the three gains, we obtain for the first piece a gain equal to 14; between 25 and 50 mV, with the same operation, we can compute a gain equal to 6; final piece, from 50 to 100 mV, gain equal to 2.

This is an approximation of the logarithmic function; with enough amplifiers (all equal!) in the chain, we will approximate better the logarithmic shape.

Every amplification stage is a differential amplifier:

Let's remark that with this technique we can not only realize logarithmic functions, but **every** function: changing the V' voltage we can change the shape of the piecewise approximation, and obtain almost every shape!

3.4 Mixers and Multipliers

In the description of a general radio receiver/transmitter architecture, we already used a lot of mixers: every time we needed to do multiplications, there were a mixer to do it. Actually, mixers are useful in a lot of other applications. Every mixers has at least two inputs and one output; the function that it must realize electronically is:

$$V_o = K_m \cdot v_x \cdot v_y$$

The symbol with the wider pins means that it works with differential signals.

We will focus our study in frequency domain, so in the effects that mixers have in the frequency spectrum: time domain is really hard to study with multiplication operations, so we prefer this type of analysis.

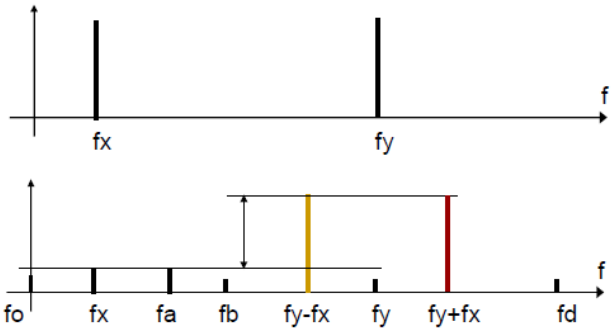
The ideal multiplier realize a function like the already seen one; the *actual* multiplier has terms like these:

$$V_o = K_m \cdot (v_x + \Delta v_x) \cdot (v_y + \Delta y_y)$$

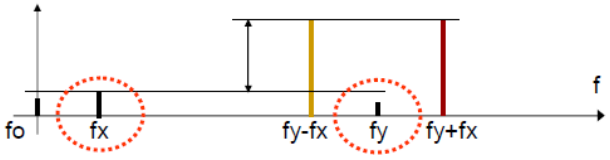
This gives many spurious terms respect to the previous function:

$$V_o = K_m \cdot V_x \cdot V_y + E_x \cdot V_x + E_y \cdot V_y + E_o + ...$$

There are many terms of second and higher order, depending on the spurious harmonics.



The Δ terms are offset or other kind of spurious terms and non-idealities. An interesting spurious term is the *feedthrough*: if mixer has unbalance contributes, so can not balance correctly inputs, we have something like this:



$$V_o = K_m V_x (V_y + V_{yo}) = K_m V_x V_y + K_m V_x V_{yo}$$

The first member of the equation has, in the parenthesis, an offset (DC) error on V_y); **this** term causes the V_x feedthrough error: the V_x signal comes in the output, but in the ideal world it don't comes! The unbalanced stages causes this type of errors.

Another note: if we write something like:

$$v_o = k_m v_x(t) v_y(t) + k_m v_x(t) V_{yo}$$

There are higher order terms: almost every harmonic with frequency equal to a linear combination of the basic two comes out:

$$M f_x \pm N f_x, \quad \forall M, N \in \mathbb{N}$$

We usually carry out only for lower product terms ($M, N = 2, 3, 4$), cause higher terms are less influent; the most dangerous terms are the intermodulation ones: other terms can be removed with filtering, these **no**.

Amplifier mixers

The first idea can be the following one: given a linear amplifier, given two signals with different frequencies, f_x and f_y, we need an adder; as long at it is linear, we will have out of it two contributes: f_x and f_y. Non-linear terms are of two species: **harmonics** and **intermodulation terms**.

The multiplication of the two signals can be obtained as the second order distorsion term of the circuit: considering the power series of the signal, we can take the $(v_x + v_y)^2$ term: with $v_x \cdot v_y$.

In order to realize this type of mixers, it's better to use transistors: op-amps are too linear, transistors have more non-idealities. The drawback of this approach is that we are interested only on $v_x \cdot v_y$, not on every term; here, he have a lot of other terms, that we need to filter, and filtering is a hard process.

Real multipliers

Now, let's try to design a real multiplier, not one where multiplication is a side effect of the usual operations; as we know, for small signal, we have that:

$$v_o = -g_m R_C v_x$$

We know that:

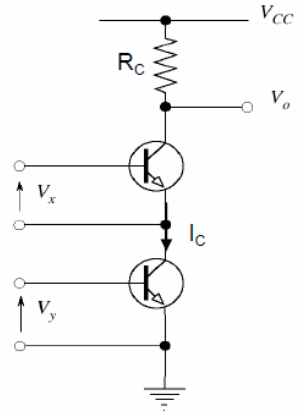
$$g_m = \frac{I_C}{V_T}$$

Changing I_C, we can multiply v_x for the changing I_C; I_C can be changed by putting a second transistor, where a v_y signal can control I_C and so g_m; this is named *transconductance amplifier*, because it works by controlling g_m.

$$v_o = -\frac{V_y R_C V_x}{R_E V_T}$$

Because:

$$I_C \simeq I_E = \frac{v_y}{R_E}$$



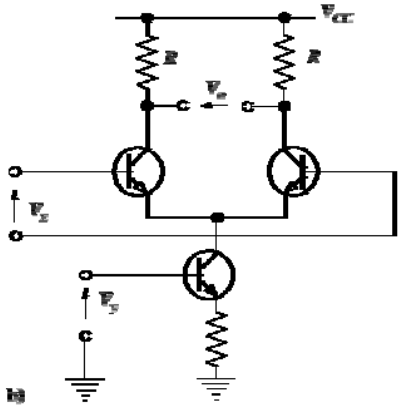
Pay attention: this is the small signal model, so this amplifier can work only with small signals; because of this, we have no feedthrough terms: small signal terms are only linear, so there are no higher terms (or with very small amplitudes), and no feedthrough!

Now: can we apply a sine wave? The answer is yes, but only with one condition: in order to don't turn off one of the transistors, we need to have a sine wave with DC component, so with an offset:

$$v_x = \sin(\omega t) + V_x$$

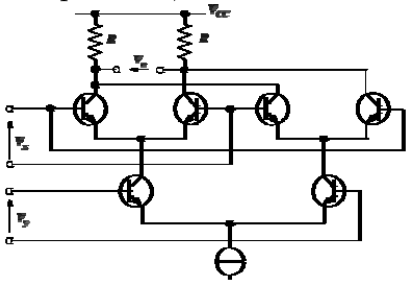
Where V_x is a DC voltage. Both v_x and v_y must have an offset, so this multiplier is named *single quarter multiplier*: it can work only in the first quadrant, so v_x and v_y both positive. We can also add a filtering circuit: by introducing $Z_c(\omega)$ instead of R_C, so a tuned circuit, we can filter unwanted spectral components.

This circuit can be modified by adding another transistor, obtaining instead of the higher transistor a differential pair:



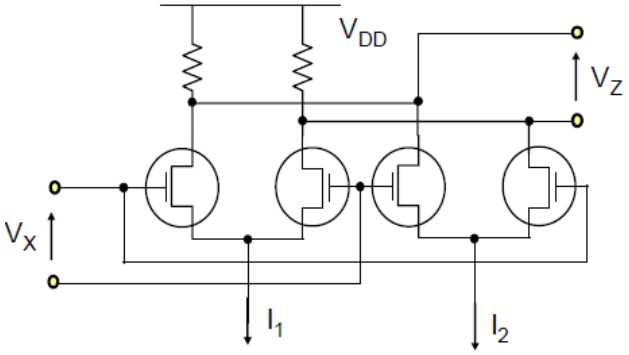
Now, the relation previously introduced is valid, but we can have positive or negative v_x values; this is the first type of *balanced mixer*.

Problem for v_x is solved, but not for v_y, that must be positive; the solution for this problem, is the double-balanced stage:



Now, both inputs are differential, but we need two differential stages, and we can have CMRR problems. This can be built with bipolar or MOS transistors; these are known as *Gilbert cells*, from the electronic engineer that invented they in the '60s.

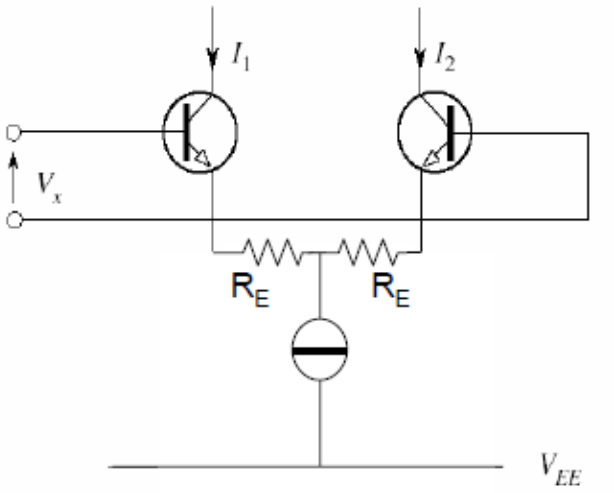
Which is the limit of the Gilbert cell? Basically, the problem is that it works using the voltage-to-current transfer for the differential stage; we can only use small signal functions, in order to have linearity and so avoid the feedthrough terms.



We have that

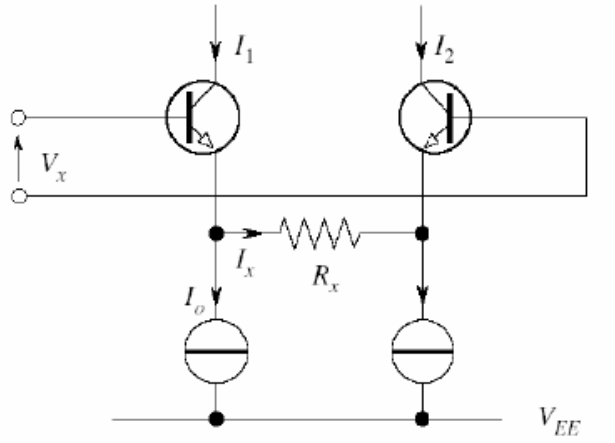
$$\Delta I = kV_x$$

We want to have linearity for more amplitude values of the small signal ones, without non-linear terms. In order to do this, we need to improve the linear dynamics range. An idea can be the one to use *negative feedback*; there are at least two ideas:



A trick can be introducing these two resistances, in order to have the same effect of the R_E on the single transistor: decrease the gain and increase stability and linearity. With matched resistances, we can obtain something very good; if resistances are not matched, we can have many problems, like common mode issues.

Something better can be the following circuit:



This can provide even more linearity than the previous circuit: balanced stages! We have in the two stages v_x, so we can say that base to emitter voltage drops are the same, and:

$$i_x = \frac{v_x}{R_x}$$

and obtain a voltage to current converter for very high amplitude dynamics range. This can be done only in integrated circuits, with matched current sources.

Now: the differential pair on the bottom can be replaced with the *wide-*

amplitude-range one, but the upper one no; we can do something else: our problem is that base-emitter junctions are not linear; in order to linearize the system, we can put something with opposite non-linearity behaviour, to obtain a linear behaviour. Introducing a current to voltage conversion with logarithmic behaviour (diodes), so a voltage to current exponential characteristic, we can obtain linearity!

Chapter 4

Phase-lock loop

A phase-lock loop (or PLL) is a system, a functional unit with an input and an output, that can realize many functions; the first one can be the filtering one: given an input with a lot of noise, in the output it provides a very clean signal with a nicer shape.

In frequency domain this means that a spectrum with many spurious harmonics will be filtered, so out of the block there will be only a line.

A PLL so can seem similar to a filter; there are many differences between this two kind of circuits: a basic difference can be the fact that this is a filter with very very precise parameters that we can control better respect to a resonant circuit; with a PLL we can automatically change the frequency of the signal in input, and it will synchronize its parameters to it, without re-tune any circuit: it can be automatically do a frequency line shifting and filtering. The filtering function is not the only one that it can realize, as we will see later: in every frequency synthesizer we have PLL, in order to obtain very precise frequency of the harmonics.

There are many ways to see, study and use a PLL: synchronizers, filters, synthesizers are the basic functions that it can provide.

4.1 Mathematical model of PLL

Phase-lock loops can be useful every time we want to obtain, out of it, a signal with a well defined phase relation with the input; in this part of the text the keyword will be **phase**: usually in electronics the fundamental variable in this contest is frequency, but now it is not very important (only in the side effects of the study), because of the very important studies on phase.

A little note, about phase synchronization: using (in order to have simple formulas) a sine wave as input of our signal, we have that:

$$v_i = V_i \sin(\omega_i t + \vartheta_i)$$

There is an explicit term of the phase; in input, we consider to have a cosine wave (in order to simplify the following expression), with another explicit term of the phase:

$$v_o = V_o \cos(\omega_o t + \vartheta_o)$$

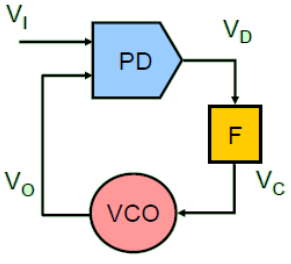
Now, if we look at the signal with the scope, we can have two cases:

- If $\omega_i \neq \omega_o$, one signal is fixed, because the oscilloscope trigger synchronizes itself on the the signal, but the other signal will shift respect to the first; this happens because there is a time-dependent phase difference function between the two signals, due to the difference of frequency: as we will see later, frequency can be thought as the derivative of the phase, so have a different frequency means have a different velocity of changing phase between the two signals, so one signal will shift respect to the other because there is not a constant relation of phase; this condition is named **not-locked condition**: there is a continuous phase shift on the screen of the scope.

- If $\omega_o = \omega_i$, but with random precision, **exactly**, we expect to see this: the phase difference between the two signal exists, but is constant, so it will remain the same, cycle after cycle. This is named **locked condition**.

Only through a PLL we can synchronize **exactly** the frequencies of the two signals; every time we have generators apparently matched, but independent, there **must** be some difference, so minimum variations of the two frequencies generate a variation in the difference of the phases, so phase shift.

This is the block diagram for a phase-lock loop:



There are some blocks:

- Phase detector (PD) : something that can measure the phase of a signal;

- Filter (F) : the loop filter;
- Voltage Controlled Oscillator (VCO): given as input a DC voltage, the output of this block will be a signal whose frequency depends on V_c.

We have that:

$$v_i = V_i \sin(\omega_i t + \vartheta_i)$$

$$v_o = V_o \cos(\omega_o t + \vartheta_o)$$

Let's remark that a PLL can work also with other types of signals like square waves, but for this analysis sine waves are the easiest way to obtain results and generalize them.

First step: the v_d voltage, out of the phase detector, can be a non-linear relation; we suppose to linearise it, so to consider only in a part of the characteristic and approximate it to the tangent line, in order to have this relation:

$$v_d = K_d(\vartheta_i - \vartheta_o)$$

Often we define a **phase error** as the difference of the input and output phase:

$$\vartheta_e \triangleq \vartheta_i - \vartheta_o$$

Second step: when we apply a control signal, there is a $\Delta\omega_o$ function of the control voltage v_c; again, this is not linear; by linearisation we approximate the characteristic as a line and get the change of frequency of the VCO as proportional to the v_c:

$$\Delta\omega_o = K_o v_c$$

Finally, the F filter has a transfer function dependent by s, so something like this:

$$v_c(s) = v_d(s) \cdot F(s)$$

K_d is known as the **gain of the phase detector**, K_o as the **VCO gain**, so the loop gain can be calculated as the product of the three gains: PD, filter and VCO:

$$G_L = K_d \cdot K_o \cdot F(s)$$

A remark: K_d and K_o are not depending on frequency; if we want to consider DC gain, we can evaluate this function in $s = 0$ (frequency zero).

Remember: our circuit, our system, works on phase, not on voltage, so the next step will be introducing a well-defined transfer function of the entire system, depending not on internal voltages, but on input and output phase; we will consider all this stuff in the Laplace transform domain, so our $H(s)$ transfer function will be:

$$H(s) = \frac{\vartheta_o(s)}{\vartheta_i(s)}$$

Now, we have some relationship between frequencies, voltages and... phase. We need to remove all dependences on every variable but phase, in order to quantify only phase variations. Assuming linear relations between voltage and phase, we know that frequency is the time derivative of phase, so:

$$\Delta\omega_o(t) = \frac{\mathrm{d}\vartheta_o(t)}{\mathrm{d}t}$$

In the Laplace domain, this becomes this:

$$\Delta\omega_o(s) = s\vartheta_o(s)$$

We already know that:

$$\Delta\omega_o(s) = K_o v_c$$

But:

$$v_c = F(s) \cdot v_d$$

And:

$$v_d = K_d(\vartheta_i(s) - \vartheta_o(s)) = K_d\vartheta_e$$

By substituting, we obtain:

$$s\vartheta_o(s) = K_o K_d F(s)\left[\vartheta_i(s) - \vartheta_o(s)\right]$$

Here we can find every loop parameter: K_o, K_d, $F(s)$: these are circuit parameters, that designer knows and can modify; ϑ_i and ϑ_o, function of the frequency, are the input and output of the system; all the transfer function can so be written as:

$$H(s) = \frac{\vartheta_o(s)}{\vartheta_i(s)} = \frac{K_o \cdot K_d \cdot F(s)}{s + K_o \cdot K_d \cdot F(s)}$$

This is similar to the transfer function formula of a generic feedback system; only difference is the s term to the denominator, instead of 1; this can be explained easily, because this circuit contains a kind of implicit integrator, in order to realize the conversion from the frequency ω to the phase ϑ. The term *lock* derives by the fact that frequency is different less then a constant; mathematically, we can say that, by maintaining constant the phase difference, we can maintain equal the derivatives of the frequencies, because the difference of frequencies has no time variation, so the derivative of the phase difference is null; due to linearity, the two frequencies of the input and output signals are the same). Filter must be a low-pass filter, as we will see later, so if we change the frequency, ω_i, we will have different frequencies, and v_d changes; if change is small, v_d change can go through the low-pass filter, v_c so change and so changes also ω_o due to the local oscillator, and ω_o becomes equal to ω_i.

The only stable condition in this loop is the lock one.

We can define a **phase error** related to the transfer function as:

$$\frac{\vartheta_e(s)}{\vartheta_i(s)} = \frac{s}{s + K_o \cdot K_d \cdot F(s)}$$

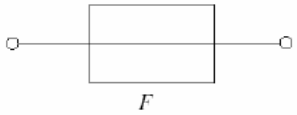
As usually happens in feedback systems, when we study part of the transfer function, we obtain the same denominator every time, as known from the theory of the Automatic Controls. This can be useful when we analyze the function, in known cases as the second order one, because we already known the behaviour of the system with variations of paramters like ξ or ω_n.

4.2 Loop filters

Let's try to begin with circuits, beginning with the most known part of the system: the loop filtering. Trying to change filters characteristics and parameters, we will see how they can change loop parameters, and so the behaviour of the system.

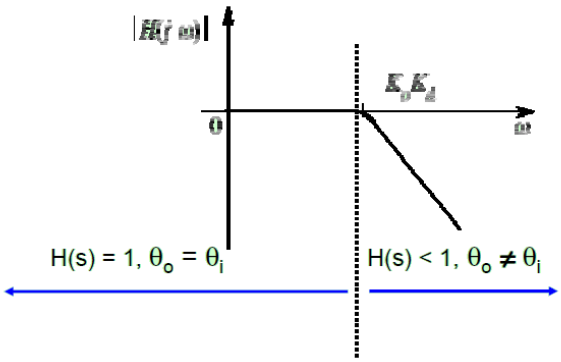
Short circuit

The first type of filter that we can try is the simplest: a direct connection:



The transfer function of this circuit is 1: it is independent by s, and unitary, because there is no leakage in the short circuit. Substituting this in the original transfer function, we will obtain:

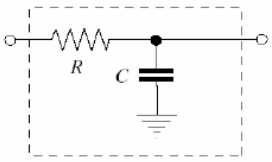
$$H(s) = \frac{K_o K_d}{s + K_o K_d}$$



Let's remark that filter has order 0, the loop transfer function order 1; this plot can tell us many informations about the behaviour of the system; first question: ω is in some way related to ω_i and/or ω_o ? The answer is **no**, and if there is some relation we don't have to care about it: we are studying phase, not frequency, so let's forget this problems! ω is in fact related to the phase changes in the system. We can divide the plot in two parts: the first, constant part, and the decreasing part: in the first one, as long as we are in the constant zone, we have that $\vartheta_o = \vartheta_i$. In the second part, ϑ_i is different by ϑ_o; this plots says that the phase-lock loop can work, but only if the signal does not change phase too fast! In fact if we have a **phase step**, so a very fast variation of phase in the input, after a transient ϑ_i may or may not be equal to ϑ_o.

RC cell filter

Another way to realize the filter can be the following one:



We have that:

$$F(s) = \frac{1}{1 + sRC}$$

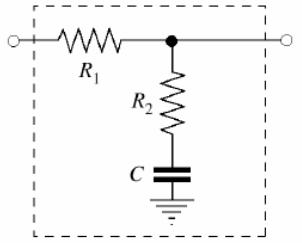
Given ξ and ω_n, we know that, now:

$$H(s) = \frac{K_o K_d}{s^2 RC + s + K_o K_d}$$

We have only two parameters, and four variables: we can change only ξ and ω_n by changing R, C, K_o and K_c, but changes of a parameter cause changes in the other one, so this circuit cannot be controlled well; in a more general contest, we can change three parameters; in a general $H_g(s)$, in fact:

$$H_g(s) = \frac{H(0)}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

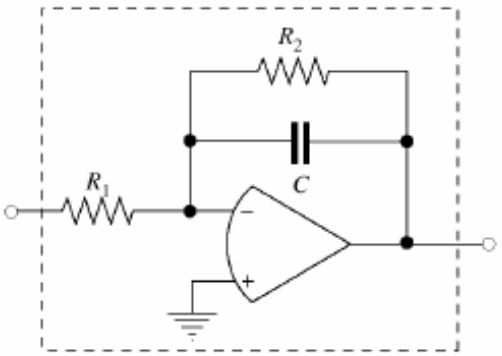
We cannot modify every parameter of the system; a better circuit can be the following one:



Now there is a low-pass filter, with a better control of the parameters: with this resistor, now, we can modify in an independent way the two parameters.

Active RC cell

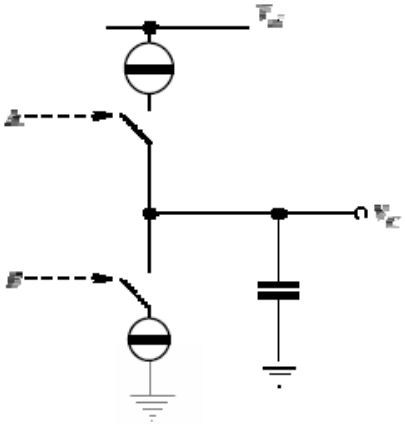
For what concerns DC gain, there is another way to realize low-pass filters: we can introduce, if we want to increase $H(0)$, a low-pass filter realized with an op-amp, with a kind of integrator:



With active components so we can easily modify gain; by having high gain, we can have a wide frequency range with a very small phase error in

the loop input (close to zero), and this can be useful every time we need wide dynamic. Ideally, if we put no resistances in the feedback block of the amplifier, we can obtain infinite gain (because we have, as gain, $-\frac{Z_2}{Z_1}$, but if $Z_2 \to \infty$ as happens in DC, the gain becomes infinite).

This was one of the two approaches to realize high gain circuits; the other approach is based on charge pumps, so circuits that can, with null phase error of the signals that pilot the switches, increase voltages:



4.2.1 Steady state phase error

An important definition, very useful for the study of the phase-lock loops, is the **steady state phase error**: it is the value of the already defined phase error, with very high values of t; for *very high values* we mean that transient must be ended, and that the system is in the steady state. The steady state phase error is so the value of the phase error after the transient; this value will be very useful in order to understand if the system locks or not.

To evaluate this expression, we can evaluate $\vartheta_e(s)$ in the Laplace domain, using the final value theorem (known from the Analysis courses):

$$\vartheta_{e,r} = \lim_{t\to\infty} \vartheta_e(t) = \lim_{s\to 0} s\vartheta_e(s)$$

Let's start from the lock condition, we are interested to study the steady state phase error after some changes of the input; as already said, there will be a transient, and then... ? Well, in order to start with this subsection, we need to calculate a better expression that quantifies the steady state phase error, $\vartheta_{e,r}$; given the $\vartheta_e(s)$, previously calculated, we can obtain:

$$\vartheta_{e,r} = \lim_{s\to 0} s\vartheta_e(s) = \lim_{s\to 0} \frac{s^2\vartheta_i(s)}{s + K_o K_d F(s)} =$$

$$= \lim_{s \to 0} \frac{s^2 \vartheta_i(s)}{s + K_o K_d F(0)}$$

We are interested only on DC terms, so instead of the entire $F(s)$ we can keep by now only the $F(0)$ term, so the DC gain of the filter. This value changes with the type of filter:

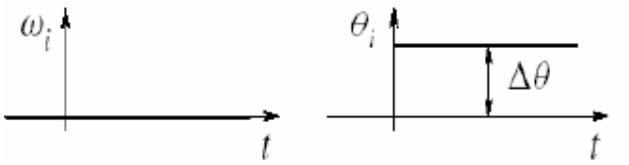
- The RC cells have gain, in $s = 0$, equal to 1;
- The integrator has infinite gain (so it goes to saturation);
- There is an intermediate solution, using amplifiers with a feedback resistor; the DC gain becomes:

$$F(0) = -\frac{R_2}{R_1}$$

Distinguish the two cases of finite and infinite gain values can be useful for the analysis that we are going to do.

Phase step response

As already said many, many times, the fundamental variable when we are studying phase-lock loops is **phase**. Let's consider an input signal with this phase behaviour:



There is a discontinuity on the origin, because there is a rough change of phase. This type of behaviour can be found in many situations, like for example PSK (Phase Shift Keying, a notorious numeric modulation). What happens to ω_i, with a signal like this? Well... nothing! Frequency, if we don't consider the transient, remains the same! This can be simply proof: considering the fact that a step, in the Laplace domain, can be represented with this:

$$\vartheta_i(s) = \frac{\Delta\vartheta_i}{s}$$

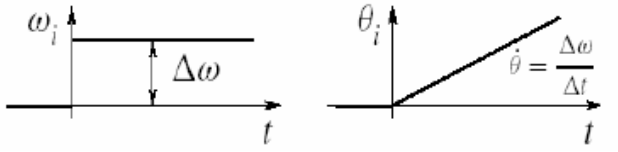
Where $\Delta\vartheta_i$ is the difference of the values of the phase before and after the discontinuity, we can replace this expression in the limit, obtaining:

$$\vartheta_{e,r} = \lim_{s\to 0} \frac{s^2 \cdot \frac{\Delta\vartheta_i}{s}}{1 + K_o K_d F(s)} = 0$$

This limit goes to 0, because $F(s)$ has always a non-zero value. This result is real, as we can say considering the following idea: if we need to keep the same frequency, due to a constant phase difference variation, we don't have to change v_c; if v_c does not change, v_d does not change, so neither the phase difference, independently by the loop filter.

Frequency step response

Now let's consider, instead of a signal that has a phase step, a signal with a step in frequency. A frequency step causes a linear change of the phase between the original signal (without frequency step) and the new one.



In mathematical model, respect to the previous signal, we have to divide for another s:

$$\lim_{s\to\infty} \frac{s^2 \frac{\Delta\omega_i}{s^2}}{s + K_o K_d F(s)} = \frac{\Delta\omega_i}{K_o K_d F(0)}$$

So, we have two cases:

- If $F(0) = A < \infty$, the limit is a finite value, so a constant term;
- If $F(0) \to \infty$, the limit tends to 0.

We are studying the phase error, so our system in both cases will guarantee lock condition: fixed phase difference in fact means that frequencies are equal. Considering this model and this two types of signals, we will have all the times lock condition respected, independently from the amplitude of the frequency step.

4.3 Phase detectors

We are going to look inside the blocks of the PLL system, beginning from the phase detector. Circuits that we must use as PD depend on the type of signals we have to use: analog or digital signals request different hardware implementations. For analog circuits we will use op-amps or transistors, for digital circuits logic gates and flip flops.

4.3.1 Analog phase detectors

The most common way to realize analog phase detectors uses multipliers as phase detectors. Let's understand why: given v_i and v_o signal of this type:

$$v_i = V_i \sin(\omega_i t + \varphi_i)$$

$$v_o = V_o \cos(\omega_o t + \varphi_o)$$

Using an analog multiplier, we will produce two harmonic terms with two frequencies; considering that we have $\omega_i = \omega_o$ in every PLL system, we will have: zero frequency (DC), and $2\omega_i$. Supposing so that multiplier is ideal, and that there is a low-pass filter that keeps only the zero frequency term we will have:

$$v_d = \frac{K_m V_i V_o}{2} \sin(\vartheta_i - \vartheta_o)$$

Where K_m is the gain of the multiplier.

We want to obtain, beginning from here, the following expression:

$$v_d = K_d \vartheta_e$$

We need to make some assumptions, that can make this system work. First step: let's remember that we are thinking about phase shift: if we have a phase shift longer that a period, we can not determine how long it is really, because sine wave is periodic of 2π: we have to consider a limit range of applications, so to consider only $[-\pi; \pi]$ as zone where can be calculated. Another observation: if we go too far from origin, we can introduce an inversion of sign for the slope: gain can be defined, but with negative values, so introducing a minus in the loop gain transforms the negative feedback in positive feedback, making the PLL system unstable; due to this observation, we will consider the phase evaluation in $\left[-\frac{\pi}{2}; +\frac{\pi}{2}\right]$.

Last observation: we want a relation between ϑ_e and v_d linear, proportional with a gain K_d; this can be obtained only if we consider little phase variations, in order to consider a linear behaviour of the sine wave next to the origin of the x and y axis. So, we have this *actual* relation:

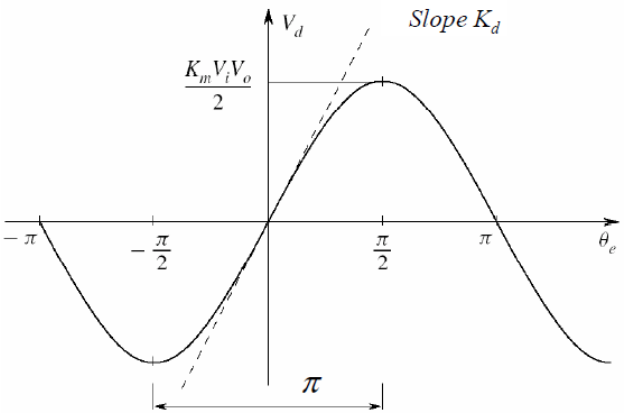
$$v_d = \frac{K_m V_i V_o}{2} \sin(\vartheta_e)$$

So a non-linear function; for small ϑ_e, we can say that:

$$v_d \simeq \frac{K_m V_i V_o}{2} \vartheta_e$$

Obviously:

$$K_d = \frac{K_m V_i V_o}{2}$$



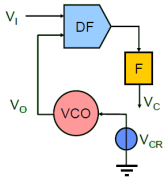
This expression says us something bad: can we, as designers, choose the phase detector gain? The answer is no: it depends on the amplitude of the signal, so we must know and fix V_i and V_o in order to define a gain in some way.

4.3.2 Butterfly characteristic

We are going to ask some questions, and give the answers, in order to realize some kind of *theoretical lab experiment*: we will try to find what happens starting from our actual knowledge, and understand the following questions:

- **When** the loop **locks**?
- **When** the loop **stays locked**?

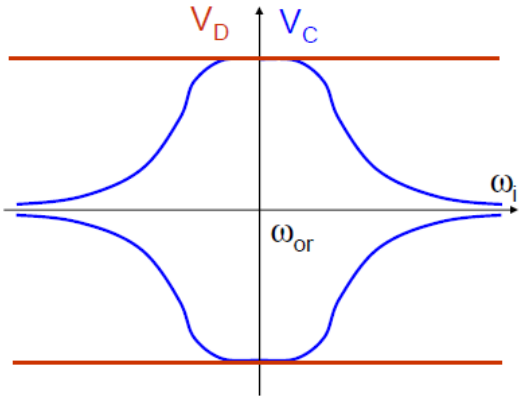
In order to understand and introduce this problems, we will consider a traditional study of a feedback system: we will open the loop, dividing the VCO block from the others, considering so a fixed output frequency, with rest value ω_{or}, and we will consider it as the center of our reference system; we will plot the diagrams of v_d and v_c depending by the input signal frequency, ω_i, considering it variable and ω_{or} fixed. Let's remark that there is no relation between ω_i and ω_o, because now the system has the loop open, so there is no correction to the v_c, connected to a fixed V_{cr} value.



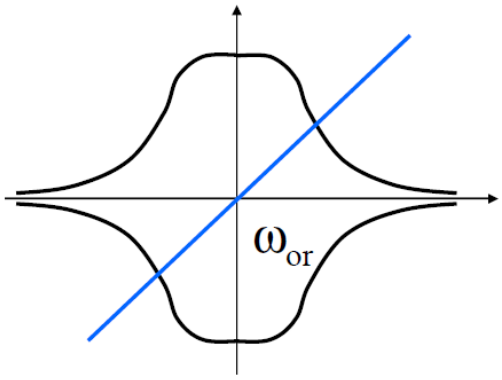
Assuming that F is a low-pass filter transfer function, we can immediately write that v_d is the voltage out of the phase detector, and v_c is simply the v_d voltage filtered through F. In phase detector, that is analogically realized with a multiplier, we obtain an harmonic contribute with frequency equal to the difference of the ω_i and the ω_{or}: considering (as already written) ω_{or} as the center of the reference, we will compute the various ω_i terms considering the difference between this center. For v_d, does not matter how many frequency difference we have between input and output signals: gain is only defined by K_d, equal to:

$$K_d = \frac{K_m V_i V_o}{2}$$

Keeping the same amplitudes of input and output signals, we will not have any frequency difference, so v_d spectrum is constant. Considering the same origins for the v_c spectrum and the filter frequency response, we can see that if difference between ω_i and ω_o is high, terms are very attenuated; if we consider similar values of ω_i and ω_o, filter let more contributes pass. v_c is simply a sine wave filtered with F: moving to high frequencies respect to ω_{or}, we have more attenuation; with lower difference, lower attenuation!

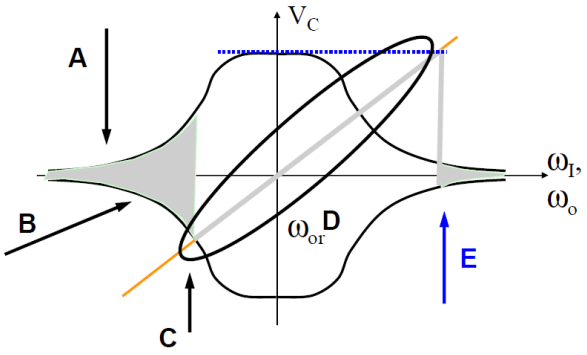


This, was the open gain behaviour; now, let's close the loop, and try to understand what happens with closed loop: we suppose that the voltage controlled oscillator (VCO) characteristic is linear, so that with a v_c change we have a ω_o change proportional to the v_c change; if $V_{CR} = 0$, and $v_c = V_{CR}$ there is no change of frequency respect to the rest point.



Let's apply a $\omega_{i,1}$ that has a very high difference respect to the ω_{or}: in our system we will have a v_c term, due to this $\omega_{i,1}$, very attenuated, and so that can not correct the VCO output in order to modify ω_{or}; moving to right with the ω_i, v_c increases its amplitude, because the filter's attenuation becomes lower and let more voltage pass through; when there is an interception between the VCO linear characteristic and the v_c, we have that v_c is high enough to modify the VCO output frequency, so to correct the ω_o to ω_i, obtaining the lock condition. From here, all becomes easier: we are moving with only DC terms, because we are only correcting with low values the ω_o respect to new values of ω_i, and we are maintaining the lock condition; we are only working with DC values of v_c, now. Now, let's increase ω_i, and go over the symmetric filter response; do we lose the lock condition? The answer is **no**! We are already moving on the VCO characteristic and using only little DC terms to obtain ω_o variations, so keeping our condition safe! The critical point, after them we don't have any locking, is the interception between VCO linear characteristic and $F(0)$ gain: after this point, we have to give to the VCO a voltage higher than the one that filter can let pass through itself, because of the DC gain; after this point we will lose the lock condition, and only going back until the interception we will regain it! We can define two frequency ranges:

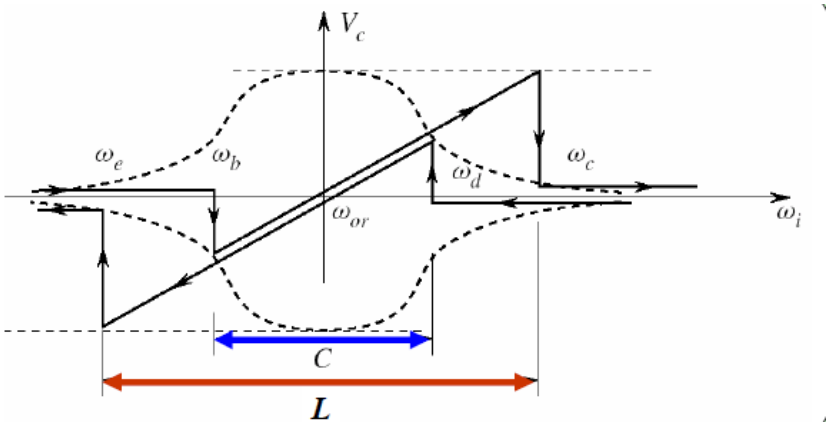
- Capture range: from unlock to lock condition: if we have no lock condition and we go through the interception of the filter frequency response and the VCO characteristic, we obtain the lock condition; the range of values (symmetric) where we can obtain the lock condition without having it before, is known as **capture range**.
- Lock range: with having lock condition, there is a wider range of frequencies that can guarantee the maintaining of the lock condition; the set of this frequencies is known as **lock range**.



Now we can also understand why we need a low-pass filter as loop filter: when loop provides corrections to the VCO, corrections are DC values; the filter must so be a low-pass, because we need to have DC and some other harmonics; the wideness of the frequency response tells us informations about the transient: if we have many harmonics, we will have a more disturbed transient; moreover, if we filter too much harmonics, we risk to obtain a too-insensible PLL, that can not react to fast frequency shifts.

Now, what happens to this characteristic, known also as **butterfly characteristic** (so named due to its shape), if we change some parameters, like the pole position (or it's τ_p), the K_d or the K_o ? Well, let's observe it!

- If we change K_o, we change the slope of the line, so both the capture range and the lock range, that are both related to the slope of the VCO linear characteristic;
- With changing τ_p, so the frequency response of the filter, we change only the capture range; changing τ_p in fact we don't change the DC gain, that provides, by intersection with the VCO characteristic, the lock range, so only the capture range is sensible to this type of variations;
- With changing V_i, V_o or K_m (so, with changing K_d), we change both capture and lock range, because we act on the y-axis scale, rescaling so the vertical variable; this changes both DC gain of the filter and intersection between the filter response and the VCO characteristic.



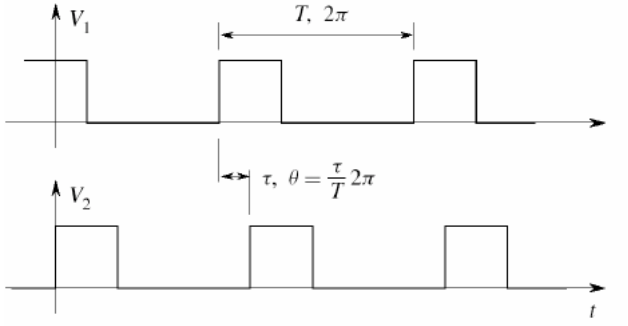
4.3.3 Digital phase detectors

We ended the study of the analog phase detector, realized with an analog multiplier; now we will study some way to realize digital phase detectors, now the most used in PLL systems.

The first step is define the phase shift also with digital signals: until this subsection we studied monochromatic signals, so with only one frequency; now we must study digital signals, so signals similar to square waves, with only two levels. In order to define phase in this case, we have to study the total period T of this signal, and the time when it remains to the upper state; it will be defined as τ; τ is so related to the phase shift in the time domain; in order to make the phase shift definition similar to the old one, we introduce a de-normalization of 2π, obtaining:

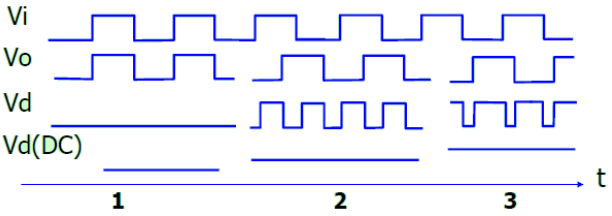
$$\vartheta_e \triangleq \frac{\tau}{T} \cdot 2\pi$$

There will be a problem, like before: with $\tau > T$, we will have more than a complete phase rotation, so it will be recognized as something more than 0 (because circuits can not recognize/discriminate more than 2π phase shift).

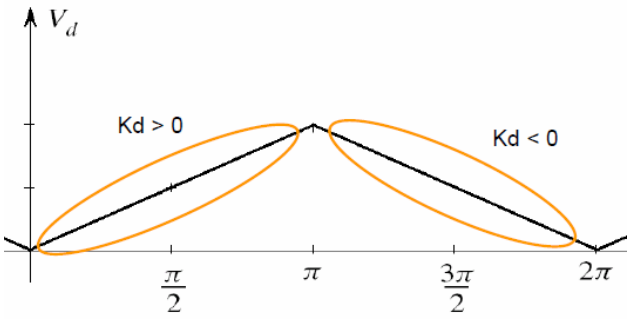


XOR phase detector

The first logic system we can use in order to realize a digital phase detector is just a XOR gate: XOR, or *exclusive-OR*, produce 1 value on the output when inputs are different, and 0 value when inputs are the same:



What we can do is consider the equivalent DC component, proportional to the amplitude of the 1 area; area is rectangular, where the height is constant (1), and the width depends on τ, so is very related to the phase shift. When $\vartheta_e = \pi$, so τ is equal to 0,5 , there is the maximum value that the XOR gate can measure:



This is the characteristic of the XOR gate respect to phase shift measurements: when ϑ_e increases, v_d (the DC component derived by the rectangular areas, depending by τ) increases, since it becomes 1 for $\vartheta_e = \pi$; after that situation, if we increase ϑ_e the situation becomes the opposite: our system sees that the DC component decreases, so the v_d; this type of phase detector consider only the lesser of the two delays.

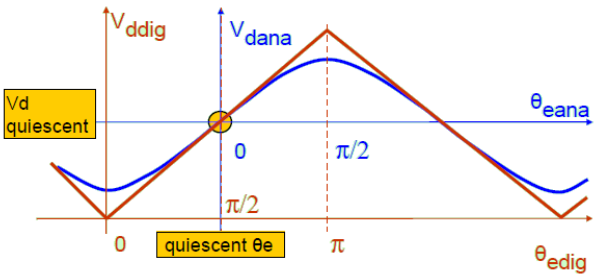
The triangular form of this characteristic is quite good: we know that a phase detector must be this characteristic:

$$v_d = K_d \vartheta_e$$

Due to the XOR's linear behaviour of v_d respect to ϑ_e, K_d is exactly the slope of our triangle; what we can do now is evaluate K_d as ratio between v_d and ϑ_e, in a well known point; we know that in $\vartheta_e = \pi$ v_d is V_H, where V_H is a parameter depending by the technology of the XOR gate, so:

$$K_d = \frac{V_H}{\pi}$$

Now, let's try to find some relations and differences between the two characteristics, of analog and digital phase detectors:

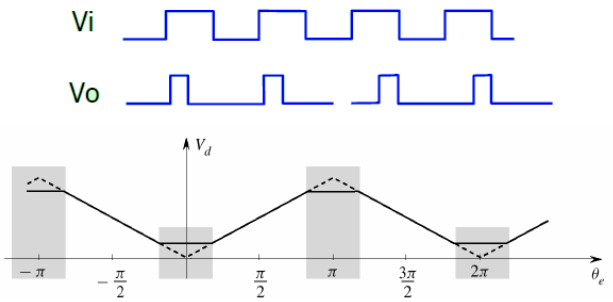


First relation: both the characteristics are periodic; a difference is that with $\vartheta_e = \pi$, the analog has negative slope, so positive feedback; the positive slope zone of the analog phase detector is narrower as the other. Another important difference regards the operating point of the circuits, so the rest point, the work point that circuits has if there is no signal in. For the analog phase detector, it is $\vartheta_e = 0$: this is the best point in order to obtain, with both increase and decrease of ϑ_e, increase and decrease of the v_d voltage, so the best situation for a signal; we can not have that also in the digital XOR phase detector, because $\vartheta_e = 0$ is not a point where v_d increases or decreases for increases or decreases of ϑ_e; the rest point for this circuit will be $\vartheta_e = \frac{\pi}{2}$, so a quarter of the total period of the characteristic.

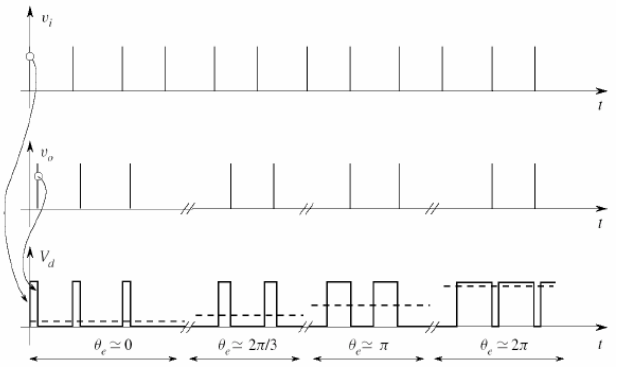
Let's pay attention to one thing: the ϑ_e phase shift definition is defined, for analog systems, from the sin/cos relation, so considering on the input a sine wave, on the output a cosine wave. The $\frac{\pi}{2}$ shift on the operating point can be explained with this convention: sine on the input, cosine on the output; for the digital phase detector (with XOR), there is a $\frac{\pi}{2}$ shift, because of the obligated choice of the operating point.

Sequential digital phase detectors

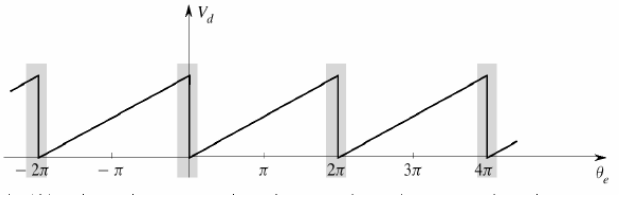
Until now, we analysed only signals like square waves, with 50 % duty cycle, in order to measure its phase error; another possibility is study digital signals, with no 50 % duty cycle.



Can this be studied with a XOR gate? **No**: the XOR will work as logic gate, but it can not measure the phase shift: if we move the v_o position respect to the previous one, the DC component does not change, because this type of detector can not determine pulse (or not-50% duty cycle) phase shifts. We need something different, something sequential, like a SR-flip-flop (set-reset flip-flop): it will obtain something like this:



The DC component is proportional to the pulse, measured with sampling the pulses; pulses change the state of the flip-flop and make it have a DC value; every pulse make the flip flop begin and end it's output: if pulses are very close the output DC will be very small, and if they are far, very shifted, DC will be greater.



Now DC depends on shift, until we reach 2π; when we go over $\vartheta_e = 2\pi$, shift can not be recognize as greater of 2π, and the characteristic starts over from 0; the good fact is that this phase detector can handle a full period of 2π, without changing its slope, so without risks of change the polarity. This can be good or bad: we lose the negative slope zone, that can be useful if we don't know a priori the polarity of the signal, and risk to make unstable the system. Rest point now can be put in $\vartheta_e = \pi$, in order to put it in the center of the characteristic; if we go over 2π, we jump down and lose lock condition.

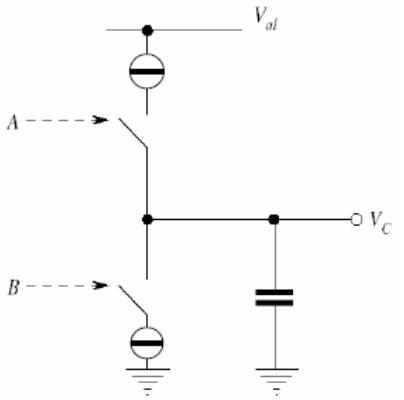
Problems for this type of circuit: pulses must be narrow and separate, because if there is overlapping between the pulses in the two inputs, we can have problems: depending on the technology of the flip-flops, they can or can not recognize the real phase shift, and propose different DC values in output. This teaches us that **there is no universal phase detector**, so that a good circuit that can realize phase shift measures in every condition

does not exists.

For overlapped signals, there are two ways:

- We can try to change the signal in something with only pulses, by deriving; in order to have 50% duty cycle, for example, we can use frequency dividers, that change period and phase shift, but maintains constant the relations between they!
- We can use special phase detectors that can operate directly on these signals, using fine state machines that can predict the signal's problems and, if they are *edge sensitive*, resolve them.

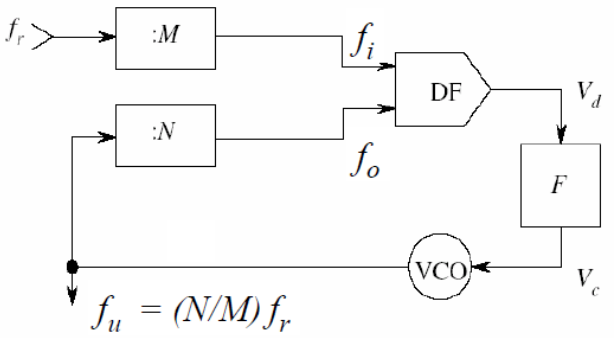
Often we use charge pump phase detectors:



There are two inputs, so v_c is the voltage already filtered; if there is only A closed, the capacitor gets charged; if only B is open, the capacitor gets discharged; if the two switches are both open or closed, we can imagine that charge in the capacitor will not change. This circuit works as an ideal integrator: if phase difference remains the same, gain increases, so like the circuit with the op-amp.

4.4 Signal synthesizers

Now, we will introduce a PLL description as fundamental block for frequency synthesis; this is one of the possible application of the PLL (like the filtering one, previously introduced); this function can be realized, simply by introducing on the input signal and on the output pin two dividers, respectively by M and N:



We have that $f_i = f_o$; if we consider that:

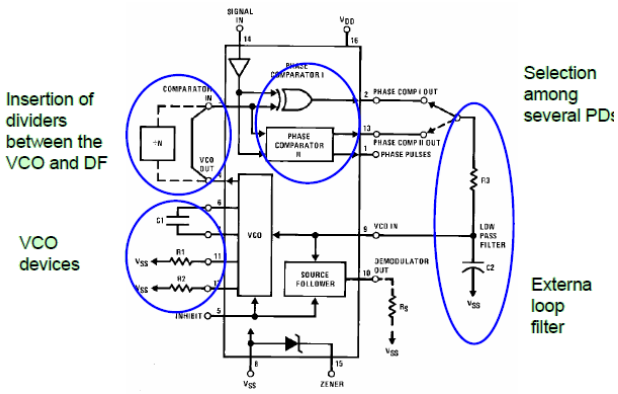
$$f_i = \frac{f_r}{M} \quad f_o = \frac{f_u}{N}$$

We have that:

$$\frac{f_r}{M} = \frac{f_u}{N} \Longrightarrow f_u = \frac{M}{N} f_i$$

If there are pins that can take f_u, we obtain a frequency synthesizer. This is a very stable and precise frequency generator, so we can obtain a very high frequency range.

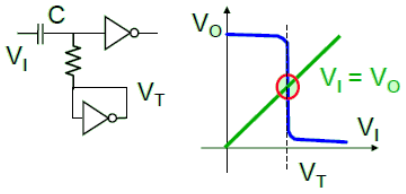
Let's consider this block diagram of a PLL system:



As we can see, there are many blocks: two phase comparators, one realized with a XOR gate and one with a charge pump, closed on a tri-state output; out of the tri-state we have to introduce the filter block, realized for example with a RC cell; out of this there is a source follower: it can be useful in order to use probes, that can change the load out of the tri-state circuit; if we have an impedance decoupler like this, we can feel free to use every probe without problems. This system must work with digital and analog signals; in order to treat both of the types of signals, there are input buffers, that can transform the sine wave (most used analog signal) into a square wave,

simply with a voltage comparator; this block is not simple to realize, because the threshold must be selected in order to be positive, in the middle value of the signal; this is difficult, with integrated circuits, because cheap integrating processes cannot guarantee precision; a trick which guarantees that DC level is equal to the threshold capacitor is to use many inverters like these, in chain (obviously, in even number, in order to maintain the original polarity of the signal).

V_T is define with another inverter that must be *special*: it can be connected in this way:



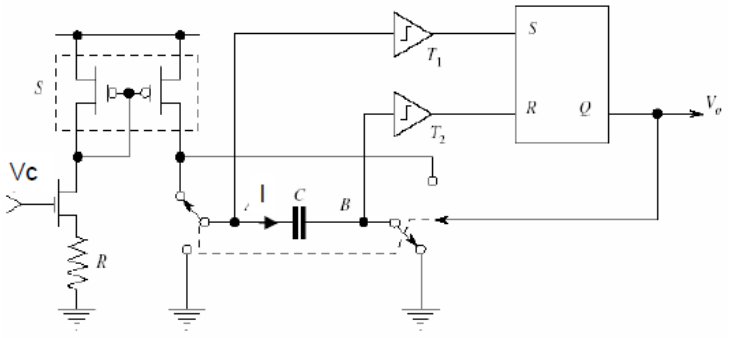
If we put this V_T resistor and this threshold inverter, with decoupling capacitor on the input, we can obtain the V_T simply with connecting input to output of the inverter; if the inverter is well define, the only possible operating point that it can have is the one previously plotted.

All these inverters must be equal, in the same thermal state and with same parameters; they must be well designed, because the problem of the CMOS inverters is that, if they conduct, there can be a big current on the channels, current that can destroy the devices; if the inverters are designed in order to have low currents, Joule effect will not destroy anything.

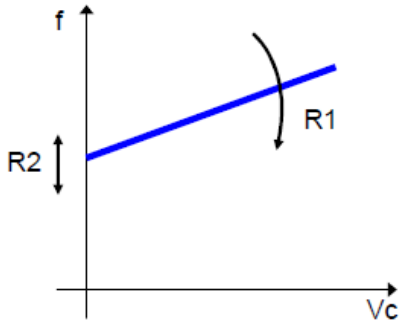
4.4.1 Voltage Controlled Oscillators

The VCO circuit is based on this principle:

This is the general way to realize square wave generators: due to constant current, there is a line (capacitor work as integrator, but it integrate just a constant, so the output voltage will be a line); when we reach the threshold, the comparator will invert the voltage of the capacitor; current begins to re-recharge the capacitor, and so go on, starting over. The frequency of this process depends on C, I, V_R: by changing C or I we change the slope, and by changing V_R we change the inverting point of the comparator. A circuit suited for integrated circuit design can be the following one:



Instead of the single comparator there are two, and the output is a flip-flop; when the threshold is reached, switches switch, and capacitor changes it's polarity; switches are controlled by the output, so by the flip-flop. The current is generated by a current mirror, realized in CMOS technology: it creates a replica of the current on the control MOS. Current is realized as sum of two contributes: one, depending on a resistance directly connected to V_{DD}, so constant (by keeping constant R_2 and V_{DD}), and the one on R_1, that depends on V_C voltage; from V_C we can change the circuit's I_1 current, and since we change I_1 we change I, so the replica, so the slope of the triangle, and the frequency of the output signal; the current on R_2 sets only an offset:

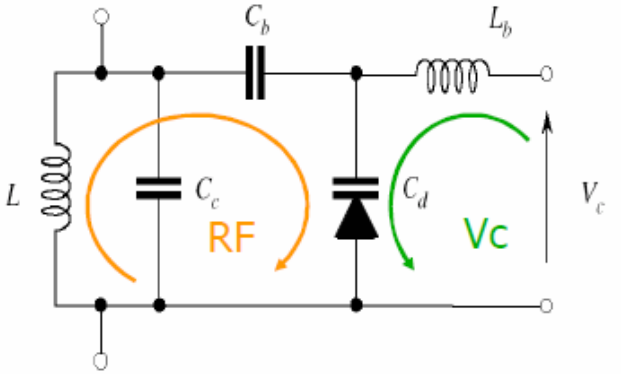


R_2 controls the starting point of the characteristic; as we apply V_C, there is also I_1, so a linear increase of the frequency. As we increase I_1, capacitor in fact is recharged quickly and frequency becomes higher, linearly! It will increase, until there is some voltage on the current mirror, keeping on the MOS.

In order to introduce more details about VCOs, let's start again, from the functional description of the block: given a control voltage v_c, with DC and variable components, out of this block there is a square wave with a f_o frequency; K_o is a proportional term that creates a relation between the change of angular frequency and the control voltage input:

$$K_o = \frac{\Delta\omega_o}{\Delta v_c}$$

It can generally be a non-linear relation, but we can consider only a small variation zone, and approximate to a line the characteristic; we can handle, with this system, both increase and decrease of frequency, identifying an opportune V_{CR} voltage control rest point and its f_{or} output frequency, from where we will move by changing v_c. K_o can quantify frequency change for v_c change, so it must be chosen well, depending on the applications we want to realize. Let's remark the following idea: the frequency range of changing must be compared with the **central frequency**: setting the VCO change from 1 kHz to 10 kHz is very different that setting it from 100 MHz to 101 MHz: even if *absolute* range between the second pair of bounds is very wider respect to the first one, we have to remark that frequency change is compared with center frequency, so the *relative* widest range is the first one (and we are interested about the relative range, not the absolute!); another parameter important, when we are about to setting a VCO, is the frequency zone where we want to work: we can need small frequency change around 1 kHz, or wide ranges around 10 GHz; this two are very different conditions, and need very different realizations. When we work in low frequency ranges, like from 100 kHz to few MHz, we can use very easy devices, like op-amps; with 10 GHz, op-amps are not working. Since op-amps circuits are easy to realize, let's focus on a high frequency realizations based on something we already know: one way to realize VCOs is based on oscillators, with tuned circuits:

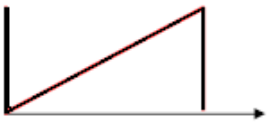


Realizing a positive feedback circuit based on this circuit, we can realize an oscillator; how can we change the resonance frequency? Well, a way is using **varistors**: with changing the voltage (in this case, the v_c control voltage) we can change the capacitance of the device, changing the depletion area; we need to isolate DC and radiofrequency, in order to modify varistor capacitance without touching any signal; in lock condition v_c does not change, so we can maintain constant also the output frequency. This realization is the most common used in consumer devices, like TVs or cellular phones.

If we want to realize a VCO in radiofrequency with a very wide range, we

need to use an heterodyne approach: multiplicating with a cosine with the right frequency we can obtain sum and difference beats, so generate terms with GHz frequencies.

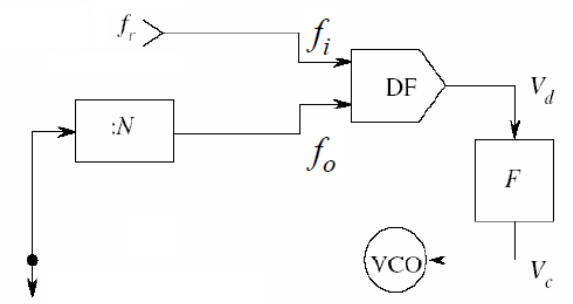
We use, in order to have small capacitances, charge and discharge capacitor circuits with fixed sources or resistances; in the VCO there is something like this:



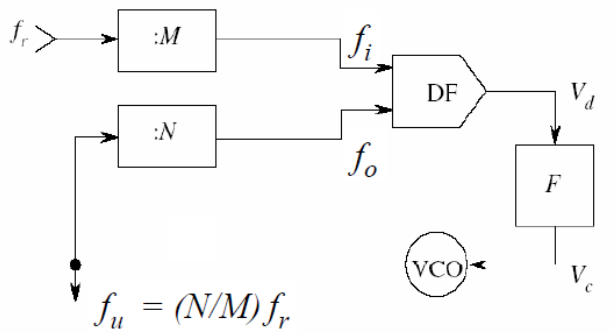
With the same circuit we can change frequency by changing the threshold of the components, instead of the current; there are circuits that can change the threshold: if capacitor is charged and discharged through a resistor instead of a fixed current source, we have a similar effect, with an exponential shape: current is not constant anymore, so there is a behaviour like this.

4.4.2 Fractional synthesizers

In order to introduce this new type of synthesizer, we have to introduce or remember the definition of **resolution**: let's consider the following scheme:



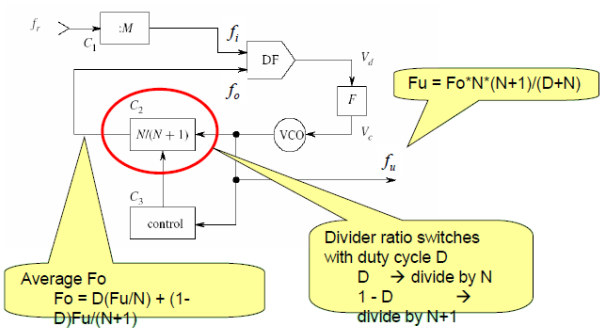
Supposing that out VCO works in 400 kHz range, and $N = 4$, if we have fixed reference frequency equal to 100 kHz, and we can only change N, what is the resolution of the output frequency? Well, if $N = 3$, output frequency will be equal to 300 kHz, if $N = 5$ output frequency will be equal to 500 kHz, so we can say that resolution, so the minimum change that we can obtain by changing the N parameter, is 100 kHz. Our question is: how to obtain a well define resolution, with a block diagram like this (or something similar) ? For example: if we ask a 1 kHz resolution, in order to obtain 401, 402, 403... kHz as possible values, what can we do?



If we want this resolution keeping the same reference frequency, we have to modify the circuit, introducing in the input a divider by 100; 100 means 7 bit (like 128), so we have to use 7 flip flops. With this circuit, if $M = 100$, N should be equal to 401, if we want 401 kHz as output frequency, and so on.

If we want a better resolution, can we use this circuit? For example, if we want a 1 Hz resolution, can we use this circuit? Well, there is a problem: every time we divide the input frequency with a flip-flop chain, we need to change the loop filter characteristics; have very precise dividers is not a problem, because they are very simple to realize (we have only to use very long chains of flip-flops), but we must remember that the loop gain filter must be realized in order to have the cut-off frequency equal to the input frequency, divider by 10; if we introduce a divider that reduces to 1 Hz the input frequency, we must use a low-pass filter with 0,1 Hz cut-off frequency, and this is impossible: the cut-off frequency cuts almost every variation term, so v_c is very similar to a DC; if v_c can not change, the PLL can not lock the signal, so it becomes useless (or very hard to use), because variations of the VCO are too slow.

How to get high resolution with fast response? This can be an idea:



We use a divider which can change the ratio, and the change is controlled by another digital circuit; there is a periodic change of the division factor N, controlled by this circuit; this circuit creates a relation between the duty

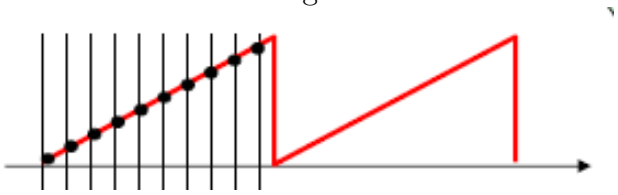
cycle D and the division factor N, in order to permits the change from N to $N+1$. Considering the average frequency, we will obtain a value of frequency that can be between f_u divided by N and f_u divided by $N+1$; it depends on the duty cycle measured by the additional circuit: the relation between the average f_o and f_u is:

$$f_u = f_o \cdot \frac{N \cdot (N+1)}{D+N}$$

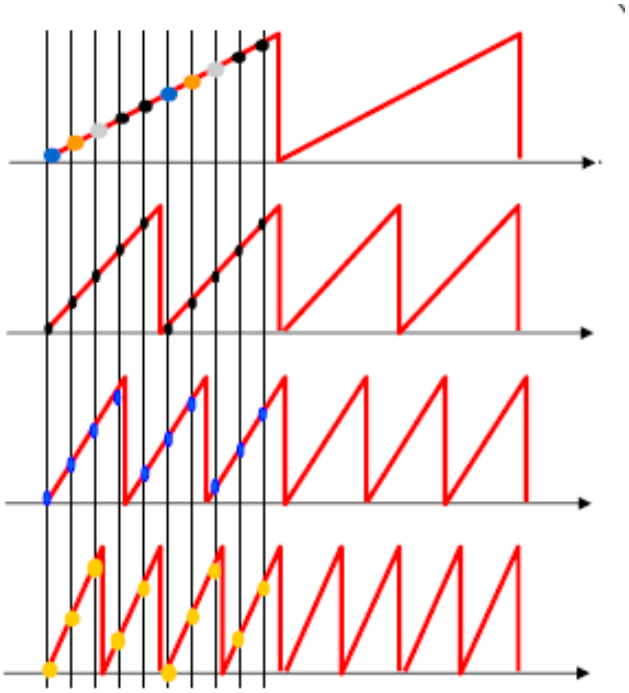
This means that if N is 0 we divide for $N+1$, so we increase the value of N of 1, and if D is 1 we maintain the same N; depending on D, evaluated on the average frequency, we can increase (or can not increase) to the following value of N. The filter does not have problems, because the division part is realized after the filter: the division term in the input of the system is designed in order to don't introduce critical divisions, so, with this technology, we have resolved the previous problem.

4.4.3 Direct digital synthesis

Let's consider the following situation:



If we consider only some of this samples, like one sample every two, and we put they in a sequence with the same sampling rate (like if they were one after the other), we obtain something very interesting:



From the beginning frequency, we obtain a signal with a frequency equal to the double of the previous one.

The idea is: if we have a list of samples of a signal, for example of 100 samples, if we read this table and take out all the values with an f_{ck} frequency, we will have a signal with frequency equal to:

$$\frac{f_{ck}}{100}$$

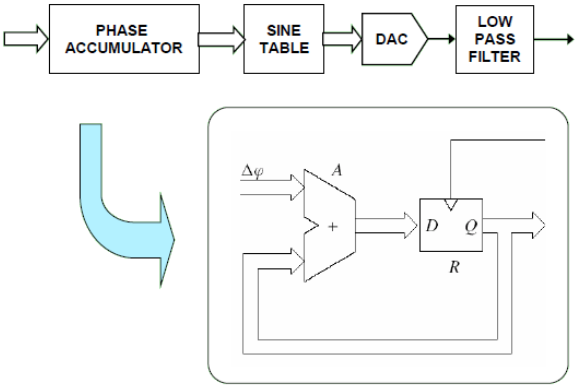
Now, if we skip half of the values, taking only one every two of them, we have:

$$\frac{f_{ck}}{50}$$

So a frequency equal to the double of the previous one, because we complete one period in half of the time.

This is only an idea, but it can be realized with any signal shape (this was a sawtooth, but we can use sine waves or every kind of shape and behaviour). Obviously, if we take one every three samples, we have a frequency equal to the triple of the previous one, and so go on. This technique is very used in many applications, like in sound processing or music synthesizers: if we need to increase the sound velocity, we can use this technique, in order to increase reproduction frequency, and generate sounds with different frequencies, just by changing the step of scanning.

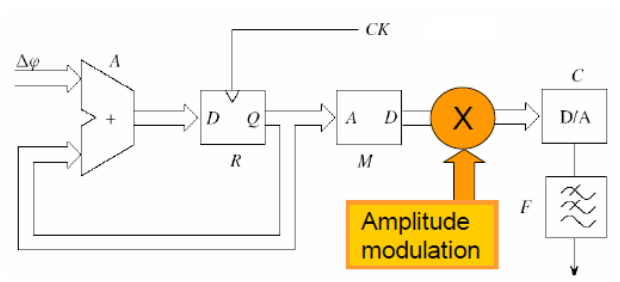
To realize this idea, we can use a structure like the following one:



In order to access to the memory we need a pointer, that must be generated with a circuit called *phase accumulator*: there is an adder with a register, and for every clock we add something to the register, in order to obtain, out of it, a new address that points to a memory block. Every $\Delta\varphi$ corresponds to a step to the next sample; out of the phase accumulator there is so a memory that contains the samples of the wave that we want to represent: a sine wave, a triangular wave, a sawtooth wave, or something else; with a DAC we move from the digital domain to the analog domain, and then introduce a low-pass filter in order to erase all the replicas generated from the sampling process.

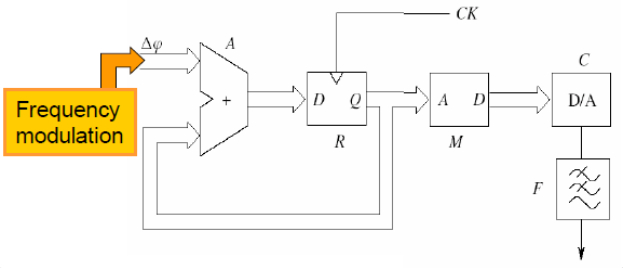
This technique can be used in order to realize the three more notorious modulations:

- In order to realize amplitude modulation, we can simply introduce, out of the phase accumulator, a multiplier; by multiplying the samples amplitudes, we obtain exactly what we want, so a modulation of the amplitude of the output signal;

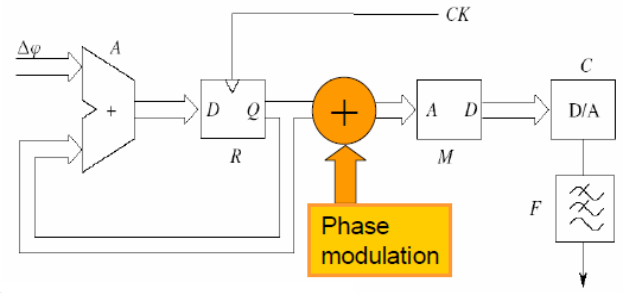


Multiplying must be an easy operation, so we choose to multiply for *simple factors*, like the powers of 2, in order to realize products simply by shifting the values of the samples of a position.

- Frequency modulation can be realized by changing the scan step: if the memory out of the phase accumulator contains the sine samples, if we change the scan step we take different samples, increasing or decreasing the frequency of the sine, obtaining the FM!



- Phase modulation can be realized simply by adding a constant to the phase detector: phase, respect to a sine wave, is simply a time-domain shift, so by adding a constant we change the first sample we consider, and obtain a shift of the entire sine wave (by taking different samples respect to the expected ones).



Until now, we considered only ideal situations; now, we need to consider the parameters which can or can not permit to realize the idea presented. We have, as already said, all the samples of the basic waveform in a memory; we assume that, in this memory, there are K samples, and a scan frequency of S; if there is , for every scan, a T_S time needed in order to realize scanning, we have that the equivalent period of the wave is:

$$T_o = T_S \cdot \frac{K}{S}$$

In frequency:

$$F_o = F_S \cdot \frac{S}{K}$$

The output frequency depends on the scan frequency S (depending on phase step, that we can change); resolution depends on S, that is a digital

number; the minimum change that we can observe is one LSB (of S), so the least significant bit of every output; 1 LSB is an absolute value: the weight of one LSB depends on how many samples are in memory; in order to appreciate small changes, we need many samples; many samples means that the phase accumulator must generate addresses with many bits; we can find systems with addresses with up to 32 bits, but often only some of this are used; for example, only 12 of 32 bits can be used to realize address, and so will go into the memory (into the table); if we have many samples, we have more resolution.

Resolution does not only depends on the number of samples memorized into the table: there are many other errors, like:

- Aliasing: we need that the spectrum of the signals satisfies Nyquist's criteria, so that sampling process is realized with a high enough frequency. It can be removed by filtering, before starting the sampling process.
- Quantization: there are errors related to the quantization process, as we will study later, that make resolution become worst. As we will see, quantization error depends on the number of bit of the sampler.
- Distorsion: the table of samples is discrete, so does not describe every phenomena; we have only approximations, so there can be distorsion, that can be quantified by studying the spectrum of the output signal.

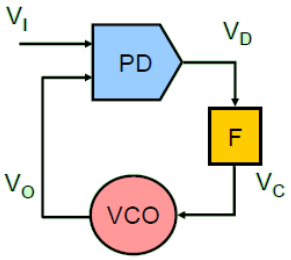
Numeric Controlled Oscillator

There is one more special application about DDS: we have seen that with DDS we can synthesize many signals but... we did not write anything about square waves; there is one good reason: obviously, with DDS we can synthesize square waves, but we don't need any memory: just one bit! In fact, a square wave can be thought as a binary number that can assume 1 or 0 values. If we want a square wave, so, we need to take only the MSB of the entire number, obtaining an NCO: an oscillator (generator of square waves) that depends only on a number.

4.5 PLL as filter

As we already studied, PLL can be seen as a very good filter; now, we want to study the parameters that can describe a PLL as a filter; we said that, if we set correctly the PLL, we can obtain a band-pass filter with a very narrow

band; the width of the bandwidth of the PLL filter depends, as we will see later, on the bandwidth of the loop filter.



In order to analyse this effect we will consider an input with noise. Noise rejection will be the fundamental study of this section: the effect of noise is related to the bandwidth of the filter, because if we have a wideband noise and a band-pass filtering system, we have a noise contribute whose width depends on the characteristics of the filtering device.

If we can find a method to evaluate the noise parameters (amplitude) and the output noise power, we could find the equivalent bandwidth of our system, seen as a filter. This is not enough: we want to compute the value of the power of v_d, with only noise, in order to obtain a relation between an only-input noise and it's output, over the phase detector; we need relations between noise and phase, because we know that:

$$\vartheta_o = H(s) \cdot \vartheta_i$$

We introduce an *equivalent phase noise* and evaluate the equivalent bandwidth; this will be done by analysing sine wave signals, so with an analog phase detector, considering for hypothesis lock condition satisfied, with $f = f_{or}$ as the resting point of the system. Another hypothesis: before the input pin of the PLL we consider a band-pass filter, that limits the incoming noise; signal has a limited bandwidth, so our filter will have bandwidth B_i, centred on f_{or}.

Finally, last hypothesis: we consider the noise value in phase and quadrature representation:

$$n(t) = n_c(t)\cos(\omega_o t) + n_s(t)\sin(\omega_o t)$$

So, first step: in order to evaluate the power after the phase detector, we have to consider an input with only noise; we know that an analog phase detector is simply a multiplier, so the output of the multiplication will contain the v_o signal, and the noise contribute, multiplied by K_m:

$$V_{dn}(t) = K_m V_o \cos(\omega_o t)\left[n_c(t)\cos(\omega_o t) + n_s(t)\sin(\omega_o t)\right]$$

By multiplying the cosines and the sine with the cosine, we obtain the sum and difference beats; obviously, only the difference beat will be useful, because the sum one is going to be filtered; we have something like:

$$V_{dn}(t) = \frac{K_m V_o}{2} \left[n_c(t) \cos(\vartheta_o(t)) - n_s(t) \sin(\vartheta_o(t)) \right]$$

Now, we are interested on power; power can be evaluated as the square of the mean of this signal; we have:

$$< V_{dn}^2(t) >=< \left(\frac{K_m V_o}{2} \left[n_c(t) \cos(\vartheta_o(t)) - n_s(t) \sin(\vartheta_o(t)) \right] \right)^2 >=$$

$$=< n_c^2(t) \cos^2(\vartheta_o(t)) > -$$

$$+2 < n_c^2(t) n_s^2(t) >< \cos(\vartheta_o(t)) >< \sin(\vartheta_o(t)) > + < n_s(t) >^2 \sin^2(\vartheta_o(t)) >$$

We consider, as hypothesis, the fact that the two statistic variables are independent, so we can separate them from the sine waves, and observe that the mean value of a sine/cosine is 0; so, the middle term get erased; by considering $n(t) = n_c(t) = n_s(t)$ (every variable with the same distribution), we have that:

$$< n^2(t) > \cdot \left[\cos^2(\vartheta_o(t)) + \sin^2(\vartheta_o(t)) \right] =< n^2(t) >$$

So:

$$< V_{dn}^2 >= \left(\frac{K_m V_o}{2} \right)^2 < n^2(t) >$$

End of the first step: we obtained an average value of the v_d noise signal.

Second step: let's consider a sine signal with phase noise, where for *phase noise* we mean that there is a casual shift in the signal:

$$v_i(t) = V_i \sin(\omega_i + \vartheta_{in}(t))$$

This model can be useful in order to relate the characteristic of the phase detector with the formula previously obtained; we are supposing, like already written, that lock condition is satisfied, so that $\omega_i = \omega_o$; both $\vartheta_{in}(t)$ and $\vartheta_o(t)$ are present; VCO is controlled by the voltage out of the filter, so it can change frequency slowly; this hypothesis is very useful, because we can imagine that

ϑ_o changes very slowly respect to $\vartheta_{in}(t)$, so we consider ϑ_o as the reference of the phase, respect to $\vartheta_{in}(t)$, and we evaluate the other, considering so:

$$v_i(t) = V_i \sin(\vartheta_{in}(t))$$

We have seen that, for small phase differences, we can approximate all with only $\vartheta_{in}(t)$ (Taylor expansion):

$$V_{dn}(t) \simeq \frac{K_m V_o V_i}{2} \vartheta_{in}(t)$$

Now we can compute the square average of this function, and compare with the one previously obtained:

$$\left(\frac{K_m V_o V_i}{2}\right)^2 < \vartheta_{in}^2(t) >= \left(\frac{K_m V_o}{2}\right)^2 < n^2(t) >$$

Let's remark that $n(t)$ is a voltage, so this equation is dimensionally correct. We have that:

$$< \vartheta_{in}^2(t) >= \frac{< n^2(t) >}{V_i^2}$$

This is an expression that can quantify the equivalent phase noise. Now, we know that the RMS (root mean square) value of V_i is $\frac{V_i}{\sqrt{2}}$, so:

$$P_S = \frac{V_i^2}{2}$$

So, considering that noise power is the square average of the noise signal:

$$\vartheta_{in}^2(t) = \frac{P_n}{2P_s}$$

Now: given the noise power, the spectral density of noise is related! If we know the spectral density of noise N_i (normally given as parameter of the transmission channel), and the noise power, we can compute the equivalent bandwidth B_i:

$$N_i = \frac{< n^2(t) >}{B_i}$$

We can introduce something similar to this idea, for the equivalent phase noise:

$$\Phi = \frac{< \vartheta_{in}^2(t) >}{\frac{B_i}{2}}$$

Warning: the bandwidth of the $\vartheta_{in}^2(t)$ it's half of the original B_i; this, because we multiply the $\vartheta_{in}(t)$ signal for a signal with the same frequency (in the phase detector), so we bring it back to the base band; here, useful bandwidth is only half of the total bandwidth (out of the base band); by substituting the first formula in the second one, we can obtain:

$$\Phi = 2\frac{N_i}{V_i^2}$$

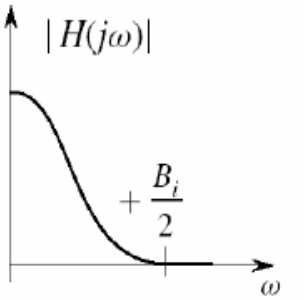
Here we know almost everything: N_i is known, V_i is known: the spectral density of equivalent phase noise can be easily calculated, with the last formula. Why do we need this formula? Well, from Electrical Communications, we know how to calculate the output power of a signal, as the integral on the equivalent bandwidth of the signal of the spectral power density, multiplied by the square absolute value of the transfer function:

$$\vartheta_{in}^2(t) = \int_0^{\frac{B_i}{2}} \Phi \cdot |H(j2\pi f)|^2 \, \mathrm{d}f$$

This is the output noise power; but... We already know it's value! So, we can say that:

$$P_o = \Phi \cdot \int_0^{\frac{B_i}{2}} |H(j2\pi f)|^2 \, \mathrm{d}f$$

This is not good: we can not put in relations this functions with the PLL parameters, because these parameters depends on B_i, that depends on the band-pass filter put before the PLL block; we can introduce another (good) hypothesis: we can suppose that $|H(j2\pi f)|$ is zero (or negligible) above $\frac{B_i}{2}$; this is an easy hypothesis to satisfy, because we can think that PLL must filter more than the input band-pass filter, because introducing a wide filter after a narrow filter is a non-sense. We can say that parameters are equal to ∞, so that:



$$P_o = \Phi \cdot \int_0^\infty |H(j2\pi f)|^2 \, \mathrm{d}f$$

Now this is the equivalent bandwidth of the PLL! In fact, Φ is the spectral power density, P_o is a power, so dimensionally the integral is a spectrum range: the spectrum range where we consider the filtering properties of the PLL!

$H(j2\pi f)$ depends on the loop filter, and on the other PLL parameters (like K_o, K_m). If we have wide bandwidth, there is more noise; with narrow bandwidth, there is less noise, but low frequency response, and the PLL has a narrower capture range (in order to lock a signal it must be have a frequency nearer to the ω_{or}).

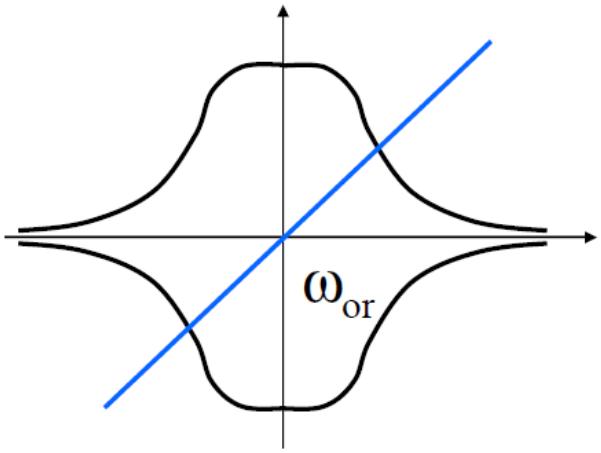
Final consideration: exists a formula that can put in relation the SNR of the input signal, and the SNR of the output signal:

$$\mathrm{SNR}_o = \mathrm{SNR}_i \cdot \frac{2B_i}{B_L}$$

Where B_L is the just defined equivalent bandwidth, B_i is the bandwidth of the filter, and the other are the input and output signal to noise ratios. This formula is strange: obviously, if we have a narrow B_L we reduce noise, so we increase the output SNR; if we increase B_i, theoretically we increase the output SNR; it does not make sense: if we get wider the filter's bandwidth, we let more noise come in; there is not a detail in this formula: the input SNR depends on B_i, so, if from one side it seems that output SNR increases, this is not true because the input SNR decreases.

4.6 PLL as frequency demodulator

Can we use a PLL as frequency demodulator? Well... the answer is yes! It can be used as demodulator for FM and AM, but also for digital modulations; let's start from the FM:

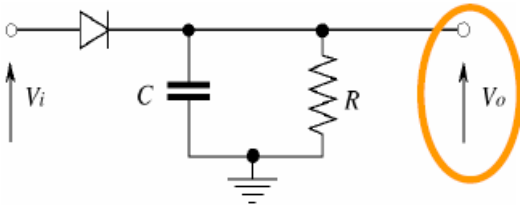


If we remember the butterfly characteristic, changing frequency we have different v_i, so we can use this characteristic to understand how frequency can be translated to voltage.

There are other techniques to do that: given a filter with cut-off frequency nearby, when we change frequency we change amplitude, so we convert a frequency modulation into an amplitude modulation: as we change frequency, the frequency response of the filter will change amplitude!

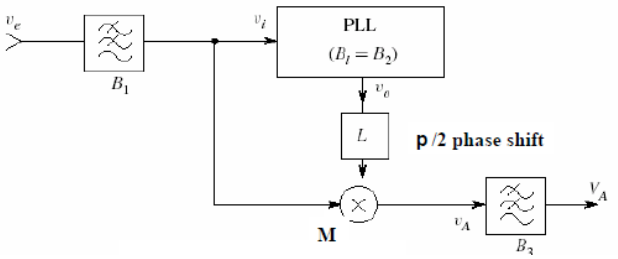
4.6.1 Coherent demodulation

The simplest modulation we know is the AM (amplitude modulation): the information, for this type of modulation, is hidden into the amplitude changing of the signal. The simplest way (but also the worst one) to demodulate AM signals is by using a diode, in series to a low-pass filter:



This is known as *envelope demodulator*.

There are more elaborated techniques, that use PLL properties to obtain a better result: in frequency domain we know that an amplitude-modulated signal is a signal shifted for a ω_c frequency; if we can measure/obtain in some way ω_c, then multiply this signal for a signal with the same frequency, as we well-know we can shift the signal in base-band, in order to obtain a simple signal to analyse/process. How can we obtain ω_c ? Well... from a PLL!



We introduce a band-pass filter, in order to remove noise; if we obtain ω_c with a PLL, we shift its output of $\frac{\pi}{2}$, so we multiply the two sine terms, in order to have a base-band signal, with a very high precision. In base-band, we can apply another low-pass filter, that cleans the other parts of the spectrum, maintaining only the useful ones. The $\frac{\pi}{2}$ phase shift is very important: from the multiplier we can get K_m, V_i and V_o; we know K_m and V_o, and we want to obtain informations about the input signal, V_i, which comes from an AM receiver; if we multiply sine wave by cosine wave (remember that a PLL produces, out if it, cosine waves), we introduce a DC component equal to 0, so useless in order to obtain V_i; with the phase shift of 90°, we can transform the cosine out of the PLL in sine, and get a DC component after the multiplier, so the V_i information.

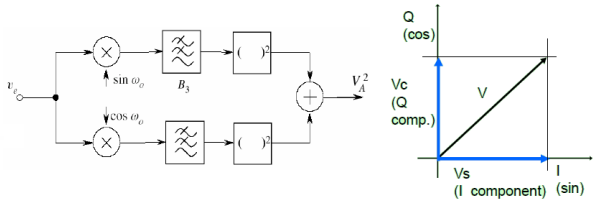
There is a problem, about this observation: output generally does not depend only on the input amplitude of the signal (as we expect, in an AM demodulator), but only on frequency changes: when PLL must lock the signal, there is a shift between the input carrier frequency and the VCO one; this shift will not give an exact product of two cosines, but a product of a cosine and something different, that is less of the expected value. There are two ways to reduce this problem:

- Increase the loop gain of the system, reducing the problem: ϑ_e in fact depends on the amplitude of the loop gain, and if we increase it we reduce the problem; this solution can be realized, but it isn't very good, because, as we know from the Automatic Controls theory, increasing the loop gain in a feedback system increase the bandwidth of the system, so the noise components that can disturb the PLL. How much the feedback noise rejection properties and the bandwidth increase change the sensitivity of the system respect to noise can be computed only with a detailed analysis.

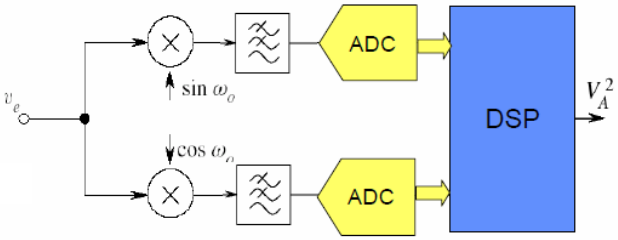
- Use an I/Q demodulator (phase and quadrature demodulator): by taking both phase and quadrature components of a signal, we consider the modulus instead of the single parameter, reducing the problem.

This second observation is interesting, in order to introduce another ap-

plication of the PLL:

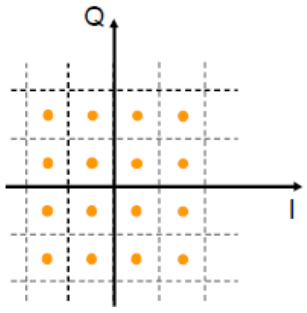


By realizing this schematic, multiplying by cosine to obtain phase component and by sine to obtain quadrature component, then adding, output will depend only on modulus, as already written. An additional note: adding and other operations can be handled in a good way with DSP:



If we use a schematic like this one, we can realize the difficult operations in a processor, that can handle they in a simpler way; we can do something more radical, introducing the A/D conversion before this blocks, but we don't do it in order to have a system that can be used also with high frequency signals.

Recovering phase and quadrature components can be useful in order to realize a QAM modulation, so a modulation that handles both phase and quadrature components, to realize a digital modulation.



PLL systems can be realized with analog electronics, with digital electronics or with software; depending on the application (obviously, near an antenna we can't use software realizations, unless we have very good radiofrequency filters and samplers!), a PLL can be implemented on a core, in wired logic, by software; the only important thing to remark if we don't want to use analog electronics is: **sample**: after the sampling process, by realizing a

digital control loop, we can resolve every other problem.

4.6.2 Tone decoders

An application for the PLL is the **tone decoder**. Tone decoders are usually integrated circuits that contain analog PLL systems setted by coherent demodulators; the VCO output for these systems is a square wave, that goes back in the loop multiplied by an analog multiplier (that realizes the phase detector block). These systems were used for phones, in order to decode tones for controls (like telephone numbers).

There are many specifics about this systems; let's study they one by one.

Bandwidth

Generally, these devices, in order to decode a well defined tone, work only on a frequency line, or in a very narrow band around it. Usually, bandwidth is equal to the 15 % of the center frequency, around it, in order to avoid decoding of unwanted tones; noise is a problem, so reducing bandwidth will be a very useful way to reduce this types of errors.

Input amplitude dynamics

There are specifications about the input voltage amplitudes, in order to specific which are the minimum and maximum values that can be recognized as valid tones. These limits are useful, because if signal amplitude is too low, it becomes similar to noise, so noise becomes very important respect to signal. There are minimum and maximum absolute input levels that guarantee to recognize a tone in order to avoid noise problems and saturation problems; these specifics are similar to the $V_{IL}/V_{OL}/V_{IH}/V_{OH}$ specifics known for logic gates: by introducing these bounds, we are quite sure that the device will recognize the tone signal.

Noise and interference rejection

There are other specifications about noise and interference terms: PLL systems have a very good immunity to noise or interference terms; for noise we mean the usual additive stochastic process that changes the equivalent signal; for interference term, we mean a signal with very great amplitude out of the band of the signal, that can decrease the performances of the system.

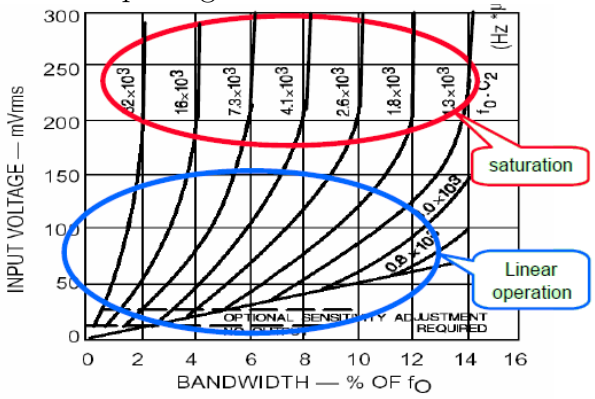
These circuit can handle these specifications (we are talking about the NE567 family, for example):

- Greatest simultaneous out-band signal to in-band signal ratio of 6 dB : this means that if we have an interference signal term out of the bandwidth with power equal to 4 times the power of the good signal, the system can recognize and reject it;

- Minimum input signal to wide-band noise ratio of - 6 dB : this means that if there is a wide-band noise term with power equal to 4 times the power of the signal, the system can recognize and reject it.

Capture and lock ranges

As already known from the previous theory, there exist for the PLL-based systems a capture range and a lock range; these terms depend on the amplitude of the input signal. There are two fundamental regions:

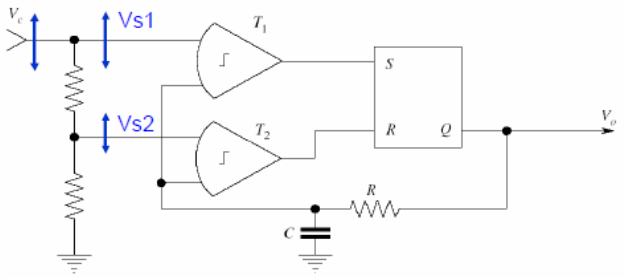


- Linear region: capture range depends on V_i, so by changing the input signal will change the capture range for the signals;

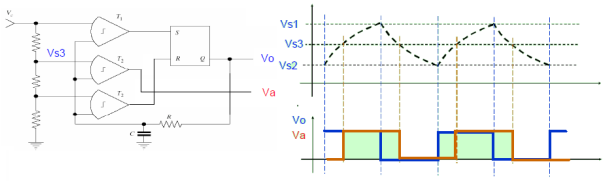
- Saturation region: if the V_i parameter is too high, system will work in saturation zone, that means that changing of V_i (over the lower bound that separates linear and saturation regions) will not determine changes of the capture range.

I-C fixed τ VCO

In order to end this subsection, will be presented a VCO implementation based on fixing the $\tau = RC$ time constant.



Here we don't change the time constant, because frequency is changed by changing the threshold; threshold is changed with the V_c control voltage: the voltage divider changes the amplitude of the signals connected to the operational amplifiers inputs, and realize the threshold change. There is another version of this circuit:



Here there is a third threshold, that introduces the possibility of a second output, which has a phase shift respect to the other; if resistances are equal, there is a $\frac{1}{3}$ exact voltage division for each resistance, and the third threshold will be exactly in the middle of the other two's distance. With this hypothesis, the output signal will be shifted by 90°: from every threshold to the third one there is the same distance, so the signal derived by the third op-amp will start and end with a delay correspondent to $\frac{\pi}{2}$ respect to the original one.

Chapter 5

Analog to Digital and Digital to Analog conversion

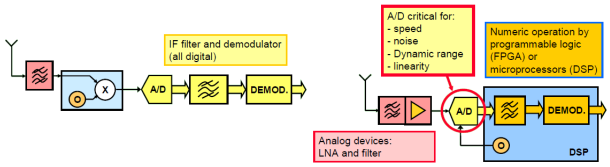
5.1 Introduction

Like already done, in this chapter we will focus first on a structured description of A/D and D/A converters, in order to understand how they must work, then we will analyse circuit implementations and other stuff; in other words, we will focus first on a structured description, looking at parameters we must know and how we can modify it in order to process information, then realizing they with electronics.

There are two fundamental applications for A/D and D/A converters:

- Radiofrequency systems, like radio (near antennas) or something similar (systems that must work with very high frequency signals);
- Audio applications (which work with tens of kHz, so low frequencies).

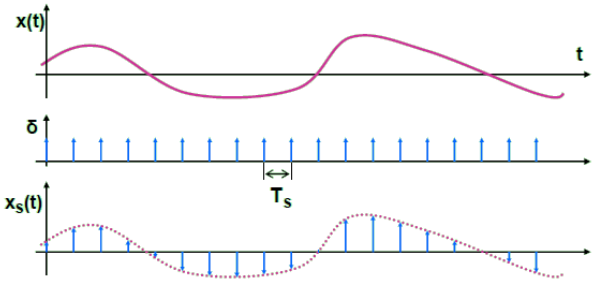
We are going to look to A/D and D/A conversion in radio architectures, in many situations: out of the IF, or before it (hard to realize).



These are two situations: the left one, which is simple (working after the IF conversion means working after a frequency rescaling), and the right one, which is the hardest situation: working after the antenna filter is very hard, because there is many noise, and converters must work with very high frequencies (and low amplitude signals).

Our systems must be **linear**: the worst non-linearity effect is intermodulation, so a set of terms that can introduce interference in our signal, and can't be filtrated; if A/D has an intrinsic non-linear transfer function, conversion introduces intermodulation, and we have many problems that cannot be resolved.

Some refresh, about analog and digital domains:



A digital signal is just a sequence of numbers, that can be represented geometrically with a set of point, assuming various amplitudes. There is a relation between the voltage values in the input of the converter and the numbers out of it; time domain is discrete, so we know informations only about **these** points. Input information can variate with continuity, on an interval (bounded): for every time value the analog signal can have every value inside the interval; output information is numeric, digital, so it can represent only some values: there is an approximation at this level, so a loss of information; the two steps, considering the **discrete time** characterization and the **discrete amplitudes** characterization are realized with two processes: **sampling** and **quantization**.

5.1.1 Sampling

Sampling can be described, in the time domain, as the multiplication of the analog signal for a sequences of pulses (Dirac deltas); we can define a sampling frequency F_S as:

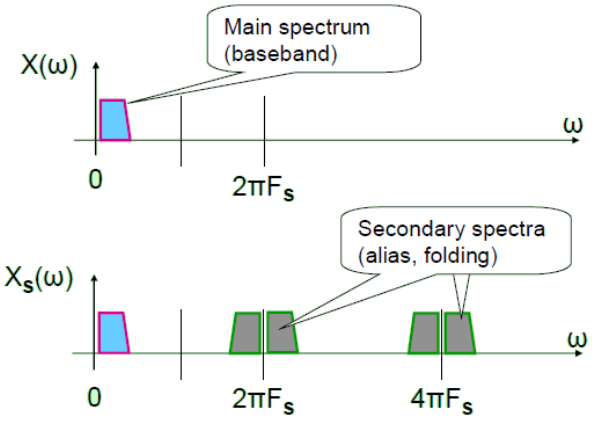
$$F_S = \frac{1}{T_S}$$

Where T_S is the distance (considered equal from every point to the next/previous one), in time domain.

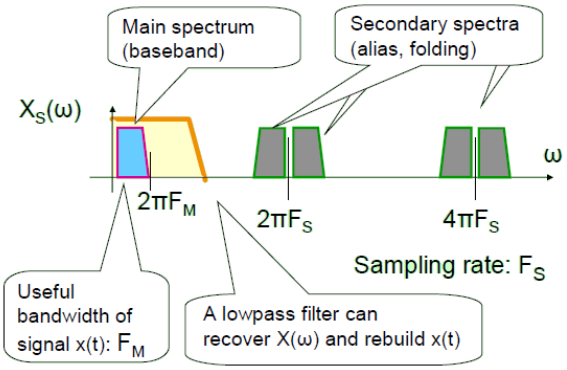
What are we losing, with this processing? Let's see: as known from the Signal Processing Theory, using the Discrete Fourier Transform, product becomes convolution:

$$x(t)\cdot\sum_{n=-\infty}^{+\infty}\delta(t-nT_S)\xrightarrow{\mathcal{F}}X(f)\otimes\sum_{n=-\infty}^{+\infty}\delta(f-nF_S)=\sum_{n=-\infty}^{+\infty}\delta(t-nT_S)X(f-nF_S)$$

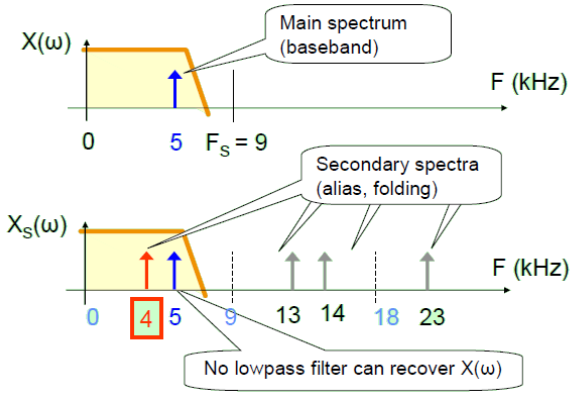
There is the base-band spectrum, and other aliases, shifted in the spectral domain:



Until now, there is no loss: this process creates replicas of the original spectrum, without introducing problems. If we want to get back the original signal, we can simply put a low-pass filter which removes the aliases and keeps only the interesting part of it.

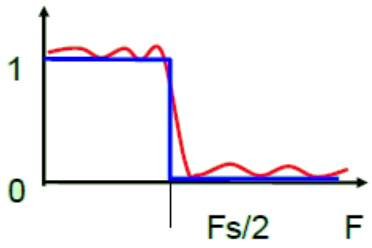


There are two bad situations: the first one is about aliasing; in order to sample correctly, we must satisfy the Nyquist theorem, so sample with a frequency equal or higher to the double of the original spectrum bandwidth; if we don't respect this criteria, in fact, we obtain something bad:



Aliasing creates harmonic contributes with frequency lower than the original spectrum one; this means that these harmonics cannot be filtered, because filtering means erase the good signal; let's remark that these are examples based on very simple spectrums, with a single harmonic; every interesting signal has spectral contents more complicated than this, so there is no way to filter they.

Aliasing is less easy to avoid as we can think: real signals hasn't a spectrum like the one showed, because of different reasons: one is the fact that filters are not perfect, ideal as we think, so they will attenuate many harmonics, but not every, keeping some disturbing interference/noise; the main reason is: as known from the Signal Processing Theory, every signal we handle before sampling, **must have a non-bounded bandwidth**: in fact, if the signal has bounded bandwidth, it must exists from $t = -\infty$ to $t = +\infty$, and this is not possible, in the real world!



Now we will lose informations: in order to delete the aliasing contributes, we must pre-filter the signal, before the sampling process, in order to avoid the aliasing effects; these effects derives from the presence of high frequency harmonics that are replicated by the sampling process. By filtering we remove some informations from the original signal so **now** we are losing informations (but if we are skilled, we can reduce this problem: we must filter only the bad part!).

Aliasing can be reduced by two ways:

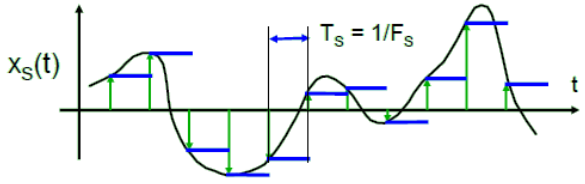
- Improving filtering: a real filter has ripple, and a slow transition from the passing zone to the attenuating zone; if we design good filters, we can reduce these problems, and reduce aliasing;

- Increasing sampling frequency: if we sample with a frequency much more higher than the Nyquist's one, we obtain much more samples and much less aliasing, obviously after a filter process: increasing sampling frequency means increasing the distance between the replicas, so if filtering is well done, much more we increase the distance between the replicas much more we will obtain interference from the other replicas, resolving (or reducing to negligible) our problem.

This last technique is known as **oversampling**: Nyquist's criteria says that one must sample **at least** at $2f_B$, where f_B is the bandwidth of the signal; techniques like **delta-sigma** or other advanced conversion are based on sampling with very high frequency! Let's remark that this is not so good: sampling with high frequency means that we generate many samples per second, so we need many power, many memory, many computation capacity, good DSPs; there are operations that can reduce bitrate: with digital filters (known as **decimation filters**) we can reduce the bitrate keeping only the good information out of the sampler; now bitrate is lower, so we can handle every sample with simpler analog electronics.

If we study this operations with a block chain, we must introduce an LPF (low-pass filter) before the sampler, in order to erase bad harmonics contributes;

A little remark: in the input of the A/D converter we need the sampler, so the electronic circuits inside the ADC will produce numbers; A/D converters are **slow**, so they need the samples for a time not short; we have to hold the value for a time, in order to realize the conversion.



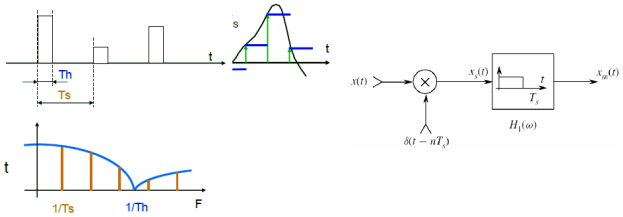
These situation is like having steps; this means that we have no more a signal like the one previously seen, but something different: if we have a pulses sequence in time domain, in frequency domain we will have again a pulses sequence in frequency domain; if in time domain instead of pulses we have steps, there will be a modification of the spectrum, because it will be multiplied by a $\text{sinc}(x)$, where:

$$\mathrm{sinc}(x) = \frac{\sin(x)}{x}$$

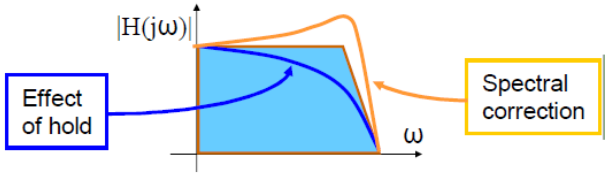
The hold operator introduces a transfer function like:

$$H_h(s) = \mathrm{e}^{-j\omega\frac{T_S}{2}} \frac{\sin\left(\omega\frac{T_S}{2}\right)}{\omega}$$

In block schematic there is so the multiplication for this new function, which changes the characteristics of the system:



This problem can be resolved by introducing, in every position of the block schematic, a filtering block with a transfer function like this (for example, it can be the input filter):



$$H_2(\omega) = H_0(\omega)/H_1(\omega) = e^{j\omega T/2}\left[\frac{\omega}{2\sin(\omega T/2)}\right]$$

The multiplication of these graphs (which means adding), means having a flat behaviour of the frequency response of the system, solving our problem. If we see filters with this shape, this **correction** reason can explain why.

These examples were related to an A/D chain; the opposite process (which requires another chain) is the D/A process (digital to analog); a digital to analog converter has the same chain, mirrored (symmetric):

After the filtering and the A/D process, some DSP will process the information and then feed the D/A, in order to produce an analog signal from the digitally-processed one; out of the DSP we will have a sampled spectrum, so a spectrum with many replicas; in order to remove replicas, we need, out of the D/A converter, another filter, which removes every replica and make the signal treatable with simple analog electronics. There are two filters in our system:

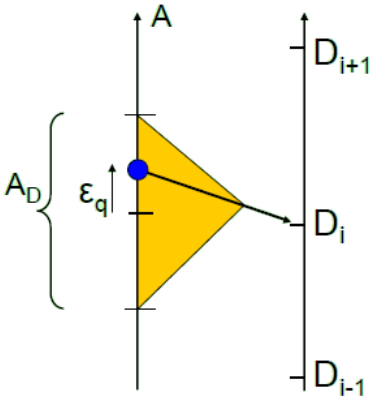
- The input filter, put before the A/D converter, which deletes unwanted contributes of the harmonics (it must have a cutoff frequency minor of

$\frac{F_S}{2}$, in order to filter; the right frequency must be designed considering that signal must not be damaged); this filter is known as **antialiasing filter**;

- The output filter, put after the D/A converter, which deletes unwanted replicas, reconstructing the signal; this filter must have the **same** bandwidth of the antialiasing filter, and it's called **reconstruction filter**.

5.1.2 Quantization

The operation previously studied was, in a few words, the discretization of the time domain, so the process which permits to consider, of all the continuous time domain, only some values; the operation we will study now is the second step of the A/D conversion previously introduced: the conversion from the continuous amplitude interval to numbers: the conversion from samples to numbers.



Which is the idea? A continuous signal can have any value in the input range S; out of our system we must have numbers in D, where D is the discrete domain. Considering an enumeration with N bits, and M levels, where $M = 2^N$, we can introduce our representation. Given a value in A, we want to translate it into D; the rule to follow is simply: divide A by M equal parts (we are considering **linear quantization**), so every value into one of the M subintervals must be mapped to a number representing the center value of this subinterval. Every number will be represented with N bits.

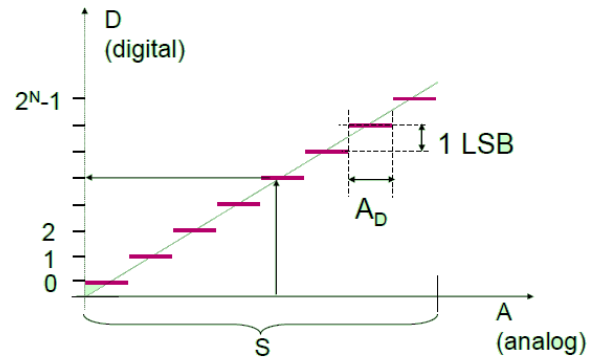
Now it's evident that we are losing information: before quantization we know the exact value (not considering measuring tools non-idealities) of the amplitude of the signal; after quantization, we know only **in which subinterval our analog signal was**, introducing an approximation, and an approximation error. The quantization error is define as the difference from

the middle of the interval and the input value; the maximum error can be introduced is the distance from the center to the bound of the subinterval, so:

$$|\varepsilon_q| \leq \frac{A_d}{2}$$

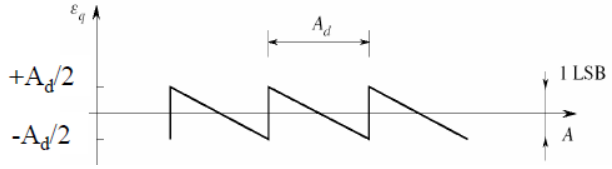
The amplitude A_d depends on N, so on the number of bits that can be used to represent every number of the sub-interval.

This is the standard transfer function for the A/D conversion:



We see steps, and their center value on a line; this line represents the *conversion gain* between input and output: there is some kind of gain which relates the position of the steps respect to numbers (some numeric relation), but this is not important. The important thing to remark is the step shape: we have zero level and, after an interval equal to $\frac{S}{M}$, a discontinuity and another step. This is obvious: until we produce a signal higher than $\frac{S}{M}$, we can not discriminate a signal from another, so every signal from 0 to $\frac{S}{M}$ is, for our converter, equal; this is the loss of information: we change input value, but can't change output value, due to quantization error.

Can we represent quantization error? Of course:



Before the center of the step, the difference between the real value and the step value is negative, so we will have a negative value; it will increase until become 0, when center is equal to real value; error will increase, because difference becomes positive, and then becomes negative for the new interval.

The parameter we want to use in order to quantify the quantization error is a relation between the signal power and the noise power: the signal to

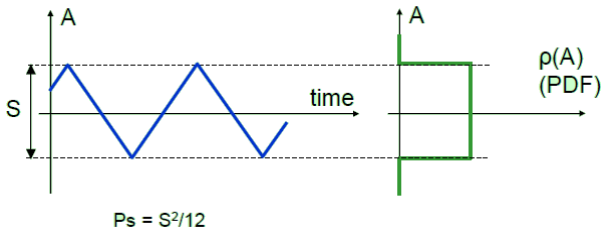
quantization error noise ratio, SNR_q. This can put, as we will see later, signal, quantization and other parameters, like N.

In order to introduce these ideas, let's introduce another idea: the **amplitude distribution**.

Amplitude distribution

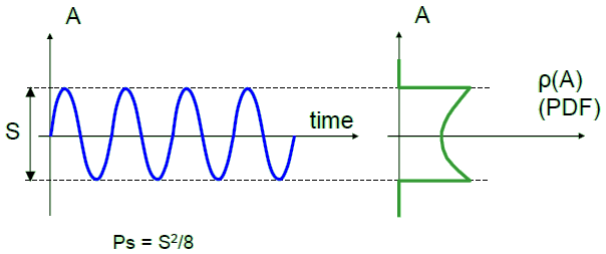
The amplitude distribution is the probability to sample the same level of the signal. This distribution is represented with a diagram *rotated of 90 degrees* respect the usual ones: this is useful in order to understand, comparing these graphs with the original ones, which amplitudes are more probable than the other. In order to understand this idea, let's show some examples:

- Triangular wave: as we can see, in triangular waves, every amplitude of the signal, each value that signal can have, has the same probability than the other values: there is always the same slope, so the same distribution, because there are no flatter zones respect to other; the amplitude distribution of this signal will be a line:



- Square wave: as we can see, square waves have only two values that signal can have: the sampler will get or the high or the low level. This means that amplitude distribution will simply be represented by two pulses:

- Sine wave: sine wave is different of the previous signals: it has zones with higher probability (the flat zones, zones were signal has similar values) respect to others; :



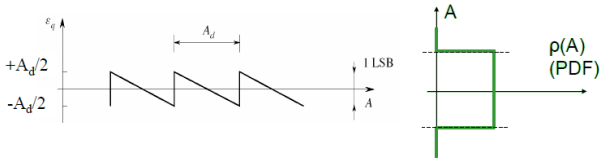
For this definition, let's remark a fact: amplitude distribution **does not depend on frequency**: an offset can change the offset of the curve (logical: if we introduce an offset, we introduce a component that will be added every time to every value of the signal, so a vertical offset on the graph.

Signal to quantization error noise

Now we will study quantization error in with the SNR_q parameter:

$$\text{SNR}_q \triangleq \frac{P_S}{P_{\varepsilon_q}}$$

We previously introduced the idea of amplitude distribution, in order to obtain some help from it to quantify this parameter; we know that error has a behaviour like this:



The amplitude distribution is equal to the triangular wave one, because sawtooth wave is very similar to a triangular wave. Since the integral of the amplitude distribution is equal to the probability to find the error in an interval, the value must be normalized to 1, so amplitude of the distribution is $\frac{1}{A_d}$. The power of the quantization noise can be evaluated as the variance of the quantization noise (due to ergodicity):

$$P_{\varepsilon_q} = \sigma^2_{\varepsilon_q} = \int_{-\frac{A_d}{2}}^{+\frac{A_d}{2}} \varepsilon_q^2 \cdot \rho(\varepsilon_q) \mathrm{d}\varepsilon_q$$

The integration bounds are $\pm\frac{A_d}{2}$ because as bounds of the interval we must consider (in order to simplify) the maximum value that the quantization error ε_q can have, so half of the distance from the center to the bound of each interval. We have said that our amplitude distribution is $\frac{1}{A_d}$, so:

$$P_{\varepsilon_q} = \int_{-\frac{A_d}{2}}^{+\frac{A_d}{2}} \varepsilon_q^2 \cdot \frac{1}{A_d} \mathrm{d}\varepsilon_q =$$

$$= \frac{\varepsilon_q^3}{3A_d}\Bigg|_{-\frac{A_d}{2}}^{+\frac{A_d}{2}} =$$

$$= \frac{1}{A_d}\left[\frac{A_d^3}{2}\cdot\frac{1}{3} - \frac{(-A_d)^3}{2}\cdot\frac{1}{3}\right] =$$

$$= \frac{A_d^2}{12}$$

This is the power of the quantization error noise; A_d as known is the quantization step, and it depends on S (amplitude of the continuous signal input range) and 2^N (number of discrete values that can be represented with N bits):

$$A_d = \frac{S}{2^N}$$

So:

$$P_{\varepsilon_q} = \frac{S^2}{12 \cdot 2^{2N}}$$

Now: we defined the signal to quantization error noise ratio, and we computed the quantization error ratio; we can compute the signal to quantization error noise ratio, for some types of signals: the expression of SNR_q in fact depends on the shape of the signal: if we put triangular, sine or square wave we will have behaviours very different (as we can see from their amplitude distributions, parameter we can use to compute also signal's power). Let's analyse three cases.

SNR_q with sine wave signal

If we introduce a sine wave signal that fill the quantizer's dynamic range we can say that the peak value of the sine wave is equal to $\frac{S}{2}$ (peak-to-peak amplitude equal to S); the power of the sine wave can be computed with it's root mean square:

$$V_p = \frac{S}{2} \Longrightarrow P_S = \left(\frac{S}{\frac{2}{\sqrt{2}}} \right)^2 = \frac{S^2}{8}$$

So:

$$\text{SNR}_q = \frac{S^2}{8} \cdot \frac{12 \cdot 2^{2N}}{S^2} = 1,5 \cdot 2^{2N}$$

Usually, SNR_q is measured/quantified in decibel (dB):

$$10 \log_{10} \left(1,5 \cdot 2^{2N}\right) = 10 \cdot \left[\log_{10}(1,5) + \log_{10}(2^{2N})\right] =$$

$$= 10 \cdot [\log_{10}(1,5) + 2N \cdot \log_{10}(2))] =$$

$$= (1,76 + 6N)\text{dB}$$

If we add 1 bit to our representation, to our device, we add six decibel of signal to quantization noise ratio; this is obvious: if we add one bit, it means that the LSB will value half then the previous one, so that we are dividing the subinterval by 2; this means that power of the ε_q is divided by 4, so the signal to noise ratio multiplied by 4, and has 6 dB more!

Opposite observation: and if... we consider a sine signal with peak-to-peak value equal to half of the dynamics? Well, if we decrease the amplitude of the signal, P_S decreases, so the signal to noise ratio decreases, of 6 dB ! Not using the entire dynamic range is like having less bits on the quantizer; as we will see later, there are a conditioning part of the system that must adapt every signal to the quantizer's dynamic, in order to obtain the best performances.

SNR_q with triangular wave signal

If we have a triangular wave, we can do the following observation: the power of the signal is known from the amplitude distribution; surely, it will be less of the sine wave power, because sine *stays more time to the higher level*; because amplitude distribution of quantization noise and triangular waves are equal we can say that:

$$P_S = \frac{S^2}{12}$$

Like the previous maths suggest! So:

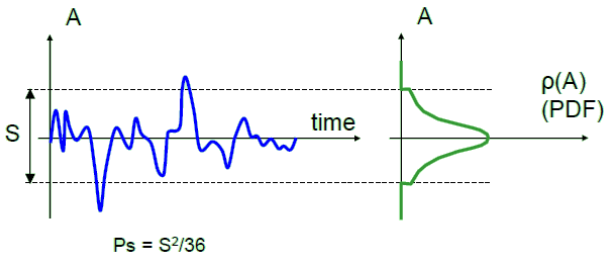
$$\text{SNR}_q = \frac{S^2}{12} \cdot \frac{12 \cdot 2^{2N}}{S^2 A_d^2} = 6N\text{dB}$$

We just removed the constant from the previous relation.

An observation: if we have no information about the signal we must handle, triangular wave is a good model: we assume, if we have no information, that every level has the same probability to be sampled, so to have a triangular wave.

SNR_q with voice signal

A real signal, like a voice signal, has this characteristics:



For the most time, it assumes low levels; for the remaining time, high level; the behaviour is quite similar to a gaussian behaviour. How must we treat this signal? Usually, like in phone systems, we cut-off part of the signal, after 3σ level, so near three times the variance of the signal; the remaining range is the useful one for A/D conversion; we obtain that:

$$S = 6\sigma, \sigma = \frac{S}{6}$$

So:

$$P_S = \sigma^2 = \frac{S^2}{36}$$

This will be very worse respect to the previous signals, because we have less signal power, and when signal power is low SNR is worst:

$$\text{SNR}_q = (6N - 4,77)\text{dB}$$

An observation: all the times there is the $6N$ term, but with a different constant.

5.1.3 Signal conditioning

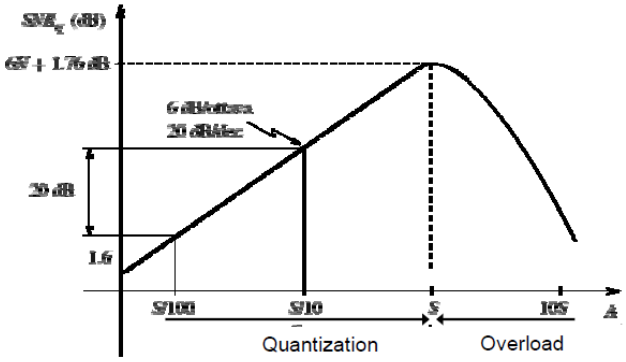
Starting from the previous signal, we introduced a cut-off of all contributes after a bound (in that case, 6σ); the cut must be done **before** the quantizer, in order to compress characteristic and to not go out of the dynamics, keeping the SNR level to acceptable values.

Treatments that **must be done** in order to obtain the best performances of the conversion systems are of two types:

- Amplitude conditioning: the amplitude dynamics of the signal must be equal to the amplitude, so it must be adapted with a conditioning amplifier (normally can be realized with op-amp circuits; it is a good idea to insert as soon as possible the amplifier, in order to reduce noise and treat signal with high amplitudes, amplitudes that makes noise negligible.

- Frequency conditioning: the frequency response of the signal, in order to avoid aliasing and other problems, must be adapted; this can be obtained with the antialiasing filter.

The blocks that realize this operations are called **signal conditioning** blocks.

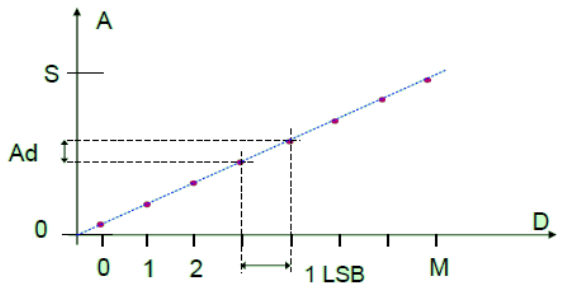


Changing the amplitude of the signal, from very low to high, we have an increase of the signal to quantization error noise ratio of 6 dB every time we duplicate the amplitude; when we have dynamics and signal amplitude equal, there is the adaptation point: the maximum value of the signal to noise ratio that the system can provide with the same number of bits for representation; if we increase the amplitude, overloading the system, there is another effect: the ε_q increases without having discontinuities, because difference between the last center of the interval and the real analog value continue to increase, so continues to increase also the error, and the signal to noise ratio decreases very fast.

Someone talks about ENOB (Effective Number Of Bits): if noise is too high, signal to noise ratio is small, so if there are error sources before quantizer, it is not useful to have a good resolution, because only few of all bits are significant; this is the effective number of bits: the number of bits useful in the representation.

5.2 Digital to Analog Converters

In order to study the first block of our system, the digital to analog converter, we will introduce the external behaviour of the system, so how it work when excited by some external event (in this case, the external event is the introduction of **numbers**):



No lines, no step, no strange things: in the device there is a number, out of it a **point**, that must ideally stay on a line. This theory will be explained for signals on the first quadrant (only positive), but is really easy to extend on every signal with every sign. The x-axis domain, so the D domain, exists only in some points; out of the block we will have A, bounded in S, so the analog range of values. The A domain is considered from 0 to S, like done before. The resolution of this system is related to the LSB, so to the least significant bit: for every LSB there is a corresponding change of A_d (like previous written, remembering every definition already introduced). If N is high, the set of points seems to be a continuous line, but it **is not**: we have to think, remember and remark that the input of this signal, and it's characteristic is not a line, but a set of points.

This device has errors; they can be classified in many ways, for example:

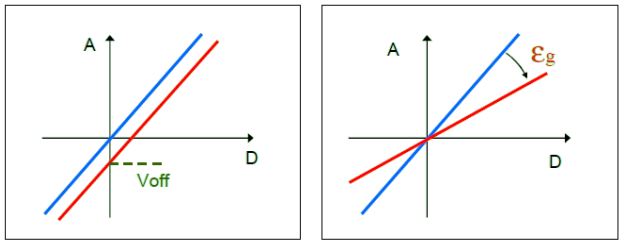
- Static errors: errors considering when there is **no change** of the values, so without introducing sequences of bits or considering transients;
- Dynamic errors: errors that take account of transient or studied with application of a sequence of numbers.

In order to characterize the static errors, we will consider three *types* of characteristics:

- Ideal characteristic: a line that starts from 000 - 0 V, ends to 111 - S V.
- Best approximated characteristic: characteristic, as we will see soon, is not linear; a way to linearize it is by using the **ordinary least square**: we evaluate the best line as the line that has the minimum least square from the real points.
- Real characteristic: the union of the various points that represents the characteristic of the device.

These three lines make us think that there are two types of errors:

- From the ideal line (ideal characteristic) to the best approximation one there are two types of errors: errors of slope, so differences between the two slopes of the two signals, and errors of offset, so shifting between the two characteristics; these are errors that can be fixed easily, simply with op-amps circuits (by introducing gain or offset terms); because errors can be fixed simply with modifying line parameters, they are known as **linear errors**;



- As we written few rows ago, the best approximated line is obtained with the ordinary least square, so a method which minimize the quadratic difference between an hypothetic line and the real point positions; in order to obtain a **linear approximation**, we trash every contribute which is not linear, so from quadratic to upper, and from here the name **ordinary least square method**; the errors committed in this operation are **not linear**, because are committed approximated a non-linear function into a linear one, so they cannot be corrected simply with operational amplifiers, feedback and linear corrections: offset and gain are not enough to provide a non-linear correction (parabolic, cubic or more, depending on the **real characteristic** of the block); these, are errors that can not be fixed easily, and are known, for the already explained reasons, as **non-linear errors**.

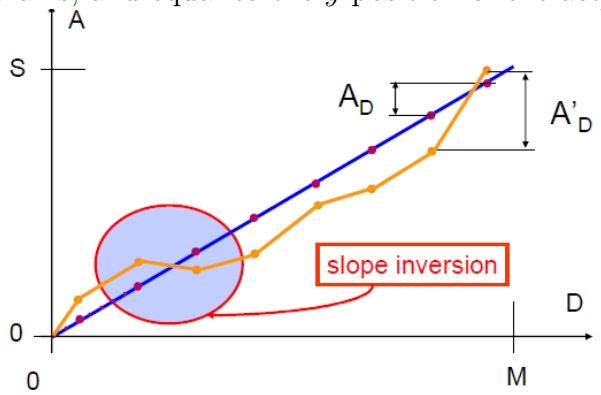
5.2.1 Quantifying of non-linear errors

Non-linear errors cannot be corrected, unless we introduce a circuit with a non-linear characteristic opposite to the previous one; it means that the characteristic of the D/A converter must be measured, approximated to a well-known function, and inverted, in order to **try** to realize this last function with electronics devices and circuits. This is never done, unless we need a very precise circuit.

What is do in every system is **quantify** non-linear errors, by using two parameters: **differential nonlinearity** and **integral nonlinearity**.

Differential nonlinearity

The differential nonlinearity is a **local** parameter, that permits to quantify the non-linear error between one point and the next one. Given a point, we expect that it is on the line (we *hope* it, because we want to have an ideal device); it can assume some position, and we must accept it. The next point, if the characteristic is ideal must be in the next point of the D domain, for the x-axis, and equal to the y position of the actual value, plus A_d;



Our characteristic (we are writing about the *best approximated one*) is only an approximation of the real one, which is non linear; we can not expect (unless we are very lucky) that the next increase respect to the actual value, A'_d, will be equal to A_d, because it is not sure: it should be sure in a linear characteristic, not in a non-linear one, where increases are not constant (definition of linearity: all increases are equal, with equal shift in the x-axis). The difference between the A_d expected and the actual A'_d is the differential nonlinearity. As already said, this is a **local** parameter, because it is computed only on two parameters: the A_d expected and the real one.

Why can be this situation critical? Well, there is a very very critical situation: if $A_d - A'_d$ is higher of 1 LSB, the converter will be **non-monotone**: by increasing the input numbers, we will have equal or less voltage out. This can be very critical for control systems, which want to increase some values and actually decreases, without possibility to change.

Integral nonlinearity

If differential nonlinearity is a local parameter, integral nonlinearity is a global parameter: it can be computer simply as the sum of every differential nonlinearity contribute, until the point we want to consider.

There is a simple way to compute the integral nonlinearity: it can be evaluated by taking the lines with the same slope of the best approximating one, passing through the most far point from the linear characteristic;

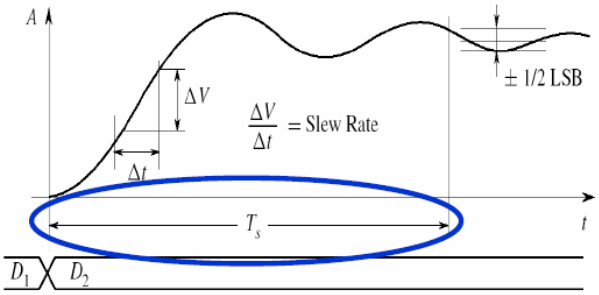
the distance between the two lines will be the integral characteristic: **the maximum distance between the differential nonlinearities**.

5.2.2 Dynamic errors

In the last subsection there was an introduction about **static** parameters of the D/A converters; as already said, there are also **dynamic** error parameters, which can introduce and study some other bad situations.

Settling time

When we introduce a sequence of numbers, we change from a sequence of inputs to another, or modify some inputs, jumping from a step value to another, there is a transient, so an actual change: the jump is not immediate, and does not have the behaviour that we expect:



Transient **never changes**, in our mathematical model: solving the differential equations that model the circuit, the system, we can see that our signal does not converge to 0: it goes close to 0, but never reaches it!

In every system there is noise, as we know; one noise source is the quantization error, already described; we can consider the transient finished after that the output value becomes, respect to the expected one, so close to can not distinguish dynamic error from noise. The time that actually quantifies the *end of the transient* is known as **settling time**.

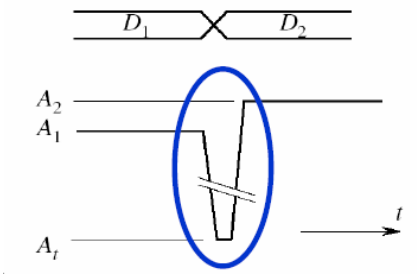
For example, if our converter has 8 bit for conversion, quantization error ε_q is equal to:

$$\varepsilon_q = \frac{1}{2^8} = \frac{1}{256} \simeq 0,4\%$$

Settling time, so, can be define as the time that system needs to reach an error of 0,4 % respect to the steady state value (static value).

Glitches

Another error related with the dynamic behaviour of the system is this one: considering two values composed of many bits (for example, four bits), like 0111 and 1000, so different from 1 bit, if some event makes 0999 change to 1000, ideally the shape of the signal changes and after a transient there is no problem. What may happen is that bits changes in different times: maybe a 0 to 1 transition is faster than a 1 to 0 transition, so there is a moment where signal becomes 1111, so 1000; when signal is 1111, the converter in its output introduces the maximum amplitude that output dynamics permits, obtaining a peak; can happen the opposite thing: if 0 to 1 transition is slower then the other one, we have something like this:



The non-sense values that derive from this phenomena are known as **glitches**: they are states of the transient where there are these effects. These phenomena are very difficult to control: there are filters that can limit the slew rate of the system, removing these variations.

5.2.3 Circuits for DAC

In order to realize digital to analog conversion we need a reference, so a quantity that remains constant, from whom we can build our circuits. There are two ideas to realize digital to analog converters:

- Uniform converters: once generated the reference quantity, we add it many times, in order to obtain the final value; this means that we need many time the same quantity to obtain a big value, because we need to add the same thing many many times.

  For example:

$$1101 = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1$$

- Weighted converters: we add quantities with **different weights**: if in the previous system we added all the time the same values, now we want to add values with different waves, in order to add less values in

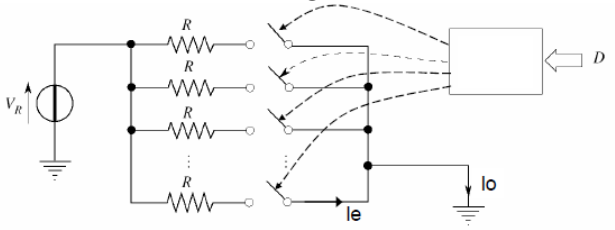
order to obtain the same result. The idea is: add many powers of 2, with multiplying by 0 or 1, depending on the state of the circuit; for example:

$$1101 = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3$$

We will study, for now, converters based on adding uniform of weighted **currents**, and then introduce some techniques to add voltages; our base quantity, for now, are **currents**.

Uniform current converters

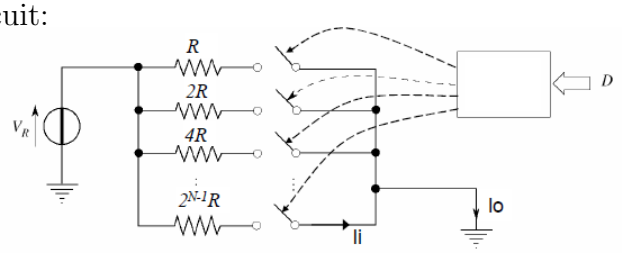
Let's consider the following circuit:



What is the idea? Well, easy: every resistor has same voltage; it may or may not be connected to the ground, so it can provide it's current or not; this means that if switch connects the resistor, current will pass on it, so out of our circuit we will sum, for every closed switch, a current contribute. Every current is equal to the others: resistors are at the same voltage level and respect to the same voltage level (if closed), and equal, so every time we close a switch we will add in parallel another resistance, obtaining on each one the same current. This is a uniform current converter, because fundamental current is the same all the times: if we close a switch, we add another current, equal to the fundamental one.

Weighted current converters

We put the same resistances in the circuit, and obtained a uniform converter; what happens if we put weighted resistances? Well, let's see the following circuit:



Topology is the same, but contributes are very different: according to the power of 2, we have different currents! Depending on the mesh we choose, we will have:

$$I_1 = \frac{V_R}{R}$$

$$I_2 = \frac{V_R}{2R}$$

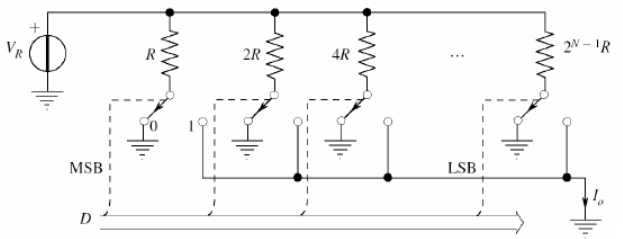
$$I_3 = \frac{V_R}{4R}$$

$$I_N = \frac{V_R}{2^N}$$

These currents are weighted, with ratio equal to 2 between one and the previous / next one.

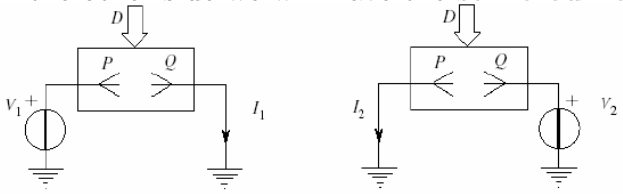
Now, let's focus on a big problem of these two circuits (uniform and weighted, as we have studied they): if the generator has a resistance, also a very small resistance, it introduces a non-linear error: this resistance creates a voltage divider, which weight is different for every configuration of the switches: every time we close a switch, with this topology, we add a resistance from V_R to ground, which is **different**: the problem of these circuits is that resistance changes when we close the switches, because if a switch is open, the resistance near it isn't referred to any voltage; when closing, we introduce a new resistance, which will change the parameters of the divider, introducing a voltage divider that **changes with the number of switch closed**, so **non-linear error** (there is **also** an error about gain, but not only it), taking our ratios out of the set of powers of 2; this error is not *systematic*, like the gain or offset ones, because it change for every configuration in a different way, not obtaining only different slopes or offsets. We need precision of the ratio, in order to reduce non-linear effects, which are not controllable. This effect can be bad: if the converter has many bits, like 16, we have a quantization error of 2^{-16}, so we need that other parameters of the circuit are negligible respect to this, and also with small resistances we can not fight this problem: we need another topology.

Our solution is the following one:

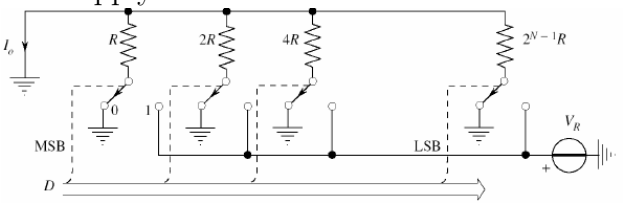


Now every resistance is **always** referred to ground: all the times, from V_R to ground, we see the **same** resistance; output current changes, because our *current collector* is referred to ground, but only if switches are connected in a right way we can obtain the good current. There is a logic circuit which controls switches, in order to realize, with its states and outputs, the weighted sum of the required power of 2.

Now, let's consider a fact: we know that passive circuit networks has **reciprocity** property: this means that if we change the input of the system, from the other side we will have the **same current**.



Let's apply it to our circuit:



There is a fundamental frequency between these two circuits: in the first one, we needed **current switches**: switches were referred from ground to ground, so more difficult to realize. Now, switches are referred from V_R to ground, so they are **voltage switches**; voltage switches are very easy to realize, for example with CMOS logic circuits.

A current switch is a circuit that works by stealing currents, like a **differential pair**.

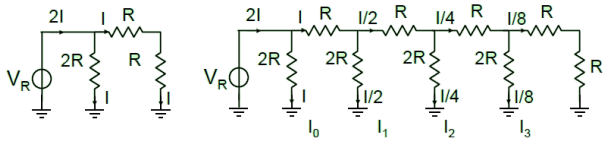
A remark: in this last circuit, we have again the non-linear problem about input resistance: resistance must be computed as seen from left to right (in this case, from where I_o is); this means that, depending on which switch is closed, resistance changes, so this circuit is again bad from the point of view of non-linearity.

The drawback of this structure are resistors: we need resistances with high spreading values, thing very difficult to realize, from the point of view

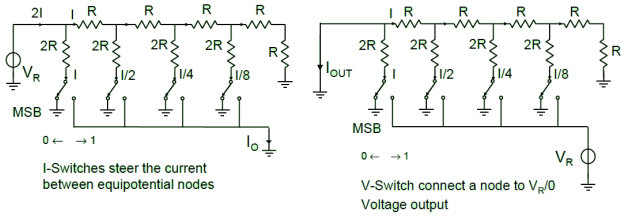
of integrated circuits.

Ladder networks

The solution of the last introduced problem is: use a ladder network. A ladder network is a resistive network with a topology like this:

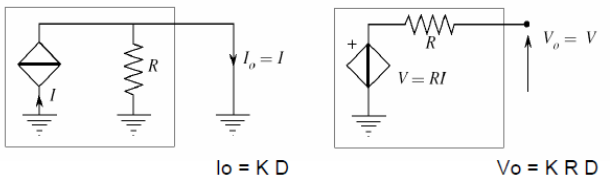


We increase the number of resistors in the ladder, but the equivalent resistance seen from the input pin is always the same, as we can calculate easily; by iterating the first equivalence many times, we divide the current for a power of two that increases with the number of branches. Depending on which type of switches we want to use, we can do something like this:

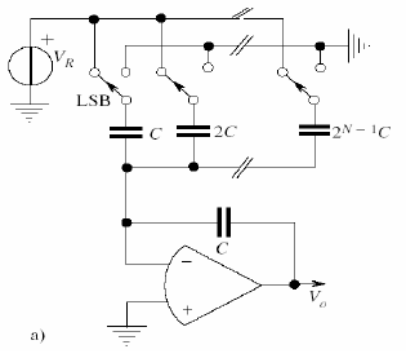


This circuit realizes the equal function of the previous one, but solving the high resistance values spreading: we need only two values of resistances, in order to realize this circuit. The circuit which uses current switches does not have the generator resistance problem; the one with voltage switches, like before, has the already seen problem, because there are, for every switch configuration, different current values, so different contributes and non-linear issues.

The *voltage switches circuit* has an advantage: it can realize not only a current output converter, but also a voltage output converter: by taking the open-circuit output voltage, we can translate input numbers into output voltages without a special circuit (or op-amps).



Final observation: instead resistances, we can use capacitances (and capacitors):



We know that resistances have a well-define linear relation between current and voltage; with capacitances, there is a similar relation between voltage and charge; using this, we can realize capacitive weight networks instead of resistive weight networks; here, weights depend on capacitance ratios; these circuits are good, because better suited for CMOS technology, because they can provide high impedances and low currents (and... realize low capacitances is quite easy, resistances is very bad), and they need **no power** for static states.

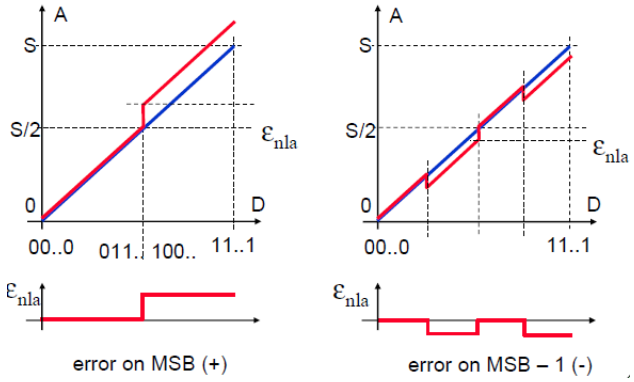
Final considerations

As already said many times, there are two types of errors: the *good ones* are linear errors, like reference voltage errors (which cause **gain errors**, because **every point of the characteristic will translate of some level linearly**), and some offset errors.

Non-linear errors are very bad: if only one resistance for example is bad, its contribute will modify only **a part of the characteristic**: linear errors are errors which are global, which exist for all the characteristic in the same way (offset), or in a proportional way depending on the position of the point considered. Non-linear errors are, by this way, very different: some zones of the characteristic can be normal, some others very wrong; this is worse than have all the characteristic modified by a same (or proportional) factor, because it can be corrected with simple additive circuitry, and non-linearities no.

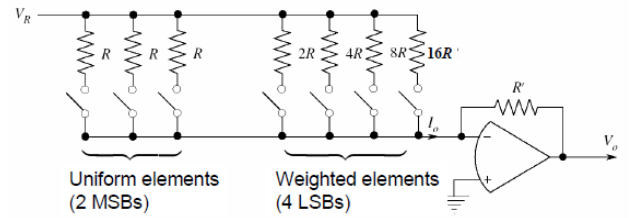
If we have, for example, an error only on MSB, in every characteristic there are two zones: one with MSB equal to zero, another with MSB equal to 1; this means that for the first part of the characteristic, an error on zero will be zero, so characteristic is normal; when MSB becomes 1, there will be error for all the second part of the characteristic, so there will be a **non-linear error**: something that can not be fixed with just offset or gain changes.

The parameter which quantifies the contribute of the non-linearity, for cases like these, is the **integral non-linearity**: we can see that how less important is our bit (most significant is MSB), less significant will be the non-linear error: if bit is near to LSB, it will change many times, so its non-linearity will be distributed in many places in the characteristic, obtaining *more linearity*. Obviously, error of these types can transform our converter from monotone to non-monotone.



Now, a little observation: with weighted structures, we can obtain non-monotonicity. Can we obtain it with uniform structures? **No**! Uniform structures are based on the paradigm "to obtain a value, let's continue to add": every time we **add**, so we **increase** our value; there is no way to obtain a non-monotone converter, with uniform structures! Why can't we use all the times these structures? Well... Let's remark that, for N bits, we need 2^N switches, and with weighted converters only N.

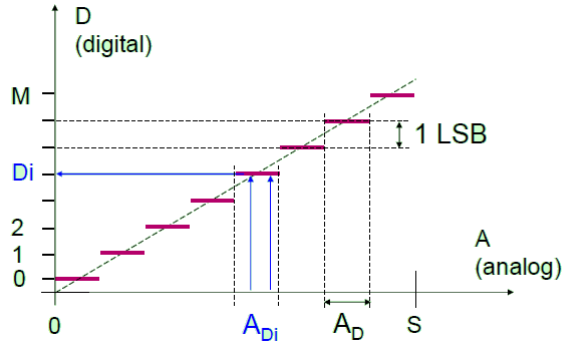
What can we do? Easy: mixed structures! For MSB and other significant bits, let's use **uniform** conversion, in order to handle in a good way the most important bits and do not have errors on they; for less significant bits, weighted conversion!



5.3 Analog to Digital Converters

As we usually have done, we will introduce the functional behaviour of ADC systems, before introducing circuital realizations; these observations are very similar to the DAC ones, so we will introduce quickly many arguments.

The ADC characteristic is dual respect of the DAC one:



Now, in the x-axis we have the analog domain (input domain), in the y-axis the digital domain. All the values in each horizontal interval are represented with digital numbers. In the x-axis signal can have any value, in y-axis only some values, because out of ADC systems there are **numbers**.

Each step size is 1 LSB, and A_d is defined, as known, as:

$$A_d = \frac{S}{2^N}$$

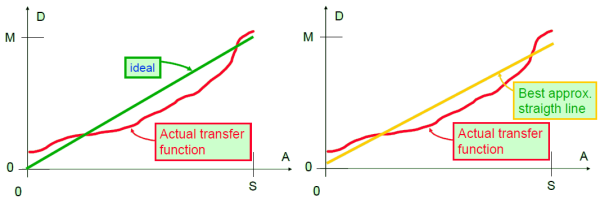
Now: if we have many steps, their width will become small.

5.3.1 Static and Dynamic errors

Static errors

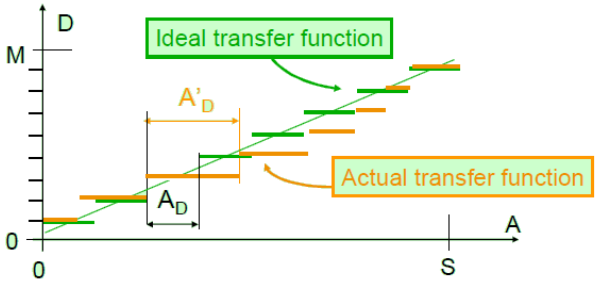
As previously done, all the analysis is based on:

- Ideal characteristic of the A/D converter, supposing that every interval is equal, and the center of each interval is connected by a line, which starts from the origin of the system;
- Linear approximation of the actual characteristic;
- Actual characteristic.



As already done, there are two static non-linear errors: **differential non-linearities** and **integral non-linearities**. Non-linearity can be corrected, but with very hard ways (like introducing ad-hoc pre-distorsions).

There is only one thing to remark: as we had, for the D/A converters, the non-monotonicity error, there is a dual error with these types of converters: missing code errors. These errors happen when three intervals, representing three steps (with different amplitude, so associated to different values in the y-axis), are overlapped and cover the middle one; in this case, the middle output code cannot be obtained, because the big non-linearity error will produce a **skip** of the middle value, causing a **missing code error**.



Dynamic errors

There is only one error to introduce and remark (because it will be very important in our following analysis): the conversion time. As we know, into the device we introduce an analog signal, sampled and hold; this second word is very important: as already said, hold operation is very important, because the conversion process of the device is not instantaneous. The device will need a time equal to T_c from when the *conversion start* signal is received to the *end of conversion* answer; this T_c is known as **conversion time**, and it introduce a limit on the maximum sampling rate (or, to the maximum useful rate which must come into the A/D converter).

5.4 Circuital implementations

Until now, we said that all static errors depend on the number of bits, and dynamic errors on conversion time: increasing N, we decrease the amplitude of the intervals, obtaining a better resolution; if we increase N, on the other side, our risk is to obtain an architecture with many blocks, so with a very high conversion time: these two parameters, N and T_c, are **concurrents**. Now, we are looking inside the box, studying some circuits and characterizing them by two parameters:

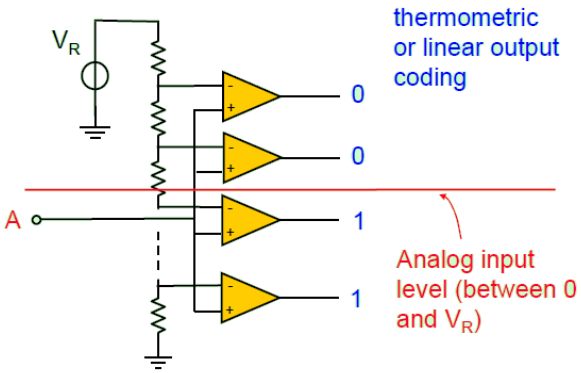
- Complexity: how many MOSFET/BJT or other components are used. We will search for a relation between N, so the number of bit (related to resolution), and the number of **comparators** needed to realize the

circuit. Comparators are not the only *bad* elements in our circuit, but surely the best ones, so in our first approximation models we will consider only their contributes;

- Conversion time: we will try to study, for each topology, its conversion time, evaluating how many stages, how many levels must work in order to realize the conversion (so how many latency exists in the system).

Flash converters (parallel converters)

Let's consider the following schematic:

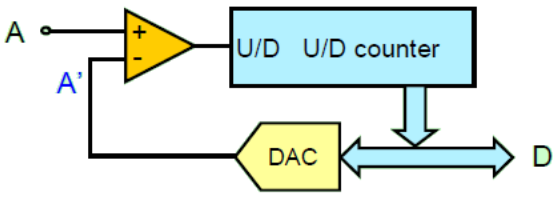


On every resistor there is, in first approximation, the same current, so every resistor defines a different threshold for every comparator, with a thermometric code. There are N bits, $2^N - 1$ thresholds (so the same number of comparators).

This is a very fast converter, because delay is equal to T_c: there is only **one** stage to use, because all decisions happen in parallel.

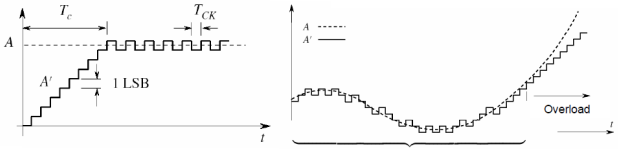
Tracking converters

Now we will introduce the opposite idea, respect to the previous one; the feedback converter:



As input we have an analog signal, which is processed by a threshold comparator; out of the comparator there is a UP/DOWN counter: if A' is less than A, then counter goes UP; with opposite situation, it goes down:

voltage comparator changes its state until steady state is reached, and with situation it continues to switch.



If signal changes, our system will follow it (so, the name **tracking converter**).

There is only a critical situation: the change speed of the signal. If signal changes too fast, our system can follow it only unless if signal's slope is lower than the maximum slew rate permitted by the system:

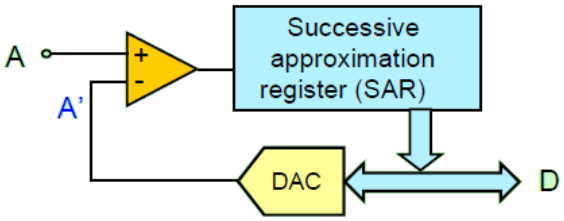
$$\mathrm{SR}_{max} \leq \frac{A_d}{T_{ck}} = \frac{1\mathrm{LSB}}{T_{ck}}$$

If the signal is faster, our converter will track it up to its maximum speed.

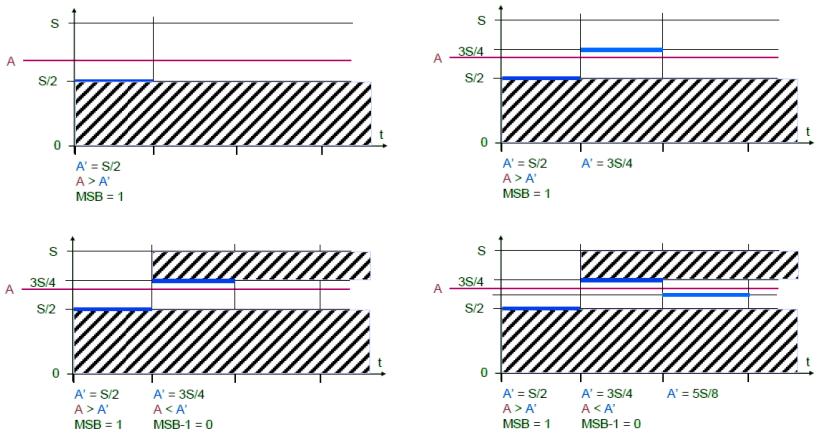
Now: which are the parameters of this system? Well, in order to go through the full scale, the full dynamic range, our system needs 2^N elaboration times, so 2^N times T_c; from the other side, it is very simple: only 1 comparator!

Successive approximation converter

Now, starting from the previous topology (useful only for didaptics), we will introduce some better solutions. Let's consider the following idea:



We are introducing a SAR, so a Successive Approximation Register: it start to go near to the solution, with large steps, so reduce by dividing by two the interval's width, excluding every time half of the remaining interval; excluding with every clock beat half of an interval means **decide the value of a bit**: starting from MSB, the SAR decides if its value is 0 or 1; decided it, it goes on MSB-1 value, and so on, using N clock beats (so N times T_c) to convert our value. Let's remark that in our first-approximation model we are **not** considering logic gates delays or issues of every type but comparators.



First time, circuit decides if excluding half of the entire interval, so the upper or lower interval respect to the entire dynamic range, $\frac{S}{2}$; decided that values cannot be in the lower part of the interval (so, that MSB = 1), we find the center of the second step interval (from $\frac{S}{2}$ to S, the average is $\frac{3S}{4}$); once done this operation, we decide the value of MSB-1, excluding values higher or lower to $\frac{3S}{4}$, and so on, finishing with the LSB determination. There is only 1 comparator, and N times the conversion time, because the only comparator in the circuit will decide for N bits with N clock beats.

Residue converting

Now, we will study the previous converter type (successive approximation converter), with a different approach, in order to obtain some improvements.

When our previous converter tried to determine the value of the first MSB, what had it done? Well, it asked to himself... *is our signal higher or lower than* $\frac{S}{2}$*, so to half of the dynamic range?* Verified that the signal is higher (like in the previous example) of half of the interval, it continued with the second step: *is our signal higher or lower than* $\frac{3S}{4}$*, so to* $\frac{S}{4}$*, plus the previous* $\frac{S}{2}$*?*. Now, let's generalize this idea: we had MSB = 1, but... who says to us that it is all the time true? Well, we can do something better: we can consider that, for the second step:

$$A > \frac{S}{4} + \frac{S}{2}\text{MSB} \quad ?$$

Obvious: if MSB is 1, the system adds $\frac{S}{2}$ to the actual parameter (for the MSB-1): $\frac{S}{4}$; if MSB is 0, we have only $\frac{S}{4}$, as expected. Now, a little algebraic manipulation: given A our number:

$$\left(A - \frac{S}{2}\text{MSB}\right) > \frac{S}{4} \quad ?$$

So:

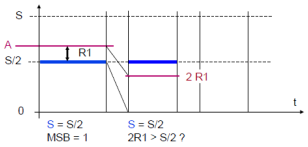
$$2\left(A - \frac{S}{2}\text{MSB}\right) > \frac{S}{2} \quad ?$$

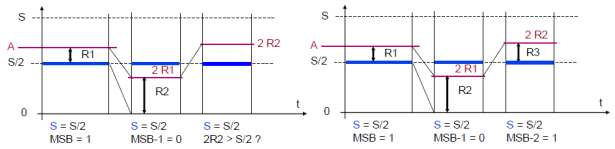
We can define this term:

$$R_1 = A - \frac{S}{2}\text{MSB}$$

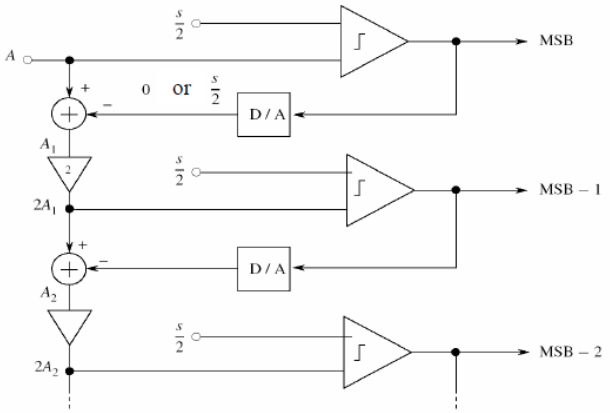
And R_1 is known as **MSB residue**.

These definitions are very useful, because they permit to introduce a way to compute, with the same operation each time, the code associated to the value. Each time we can compute the $(i+1)$-bit simply by comparing the double of the previous residue, R_i, with $\frac{S}{2}$. Each time the **same** residue is amplified by two, and compared with $\frac{S}{2}$, until it goes over the signal; when it goes over, the residue becomes computed as the difference from the high value (the old residue) and the signal value (A), so approximation is done from up to down. All the times we do the same operation.





This is the block diagram of this system:

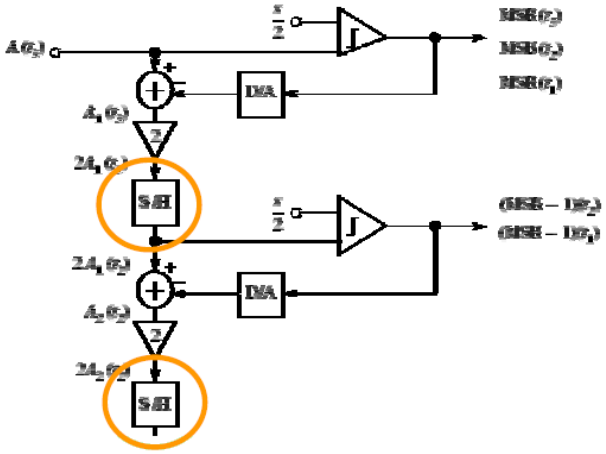


With this architecture, for N bits we need N comparators, and we will have N levels to go through; 'till now, this is not very interesting, but we

are going to introduce an idea which will transform this idea to a very smart one.

Pipeline structure

Considering as known the idea of **pipelining**, so the technique with whom we can increase the throughput of the system, we can do something like this:



Every time we end elaboration in each single stadium, there is an analog memory (so a Hold circuit) which maintains the signal ready for the next stadium, and loads a value from the previous; beat after beat we will process in each stadium a different value, so the throughput of the system will result very increased.

This system is very interesting: skipping the first/last value, it has the same speed of the flash converter, with N comparators instead of 2^N.

Now, we can complete a benchmark between the various architectures, and show the results in the following table:

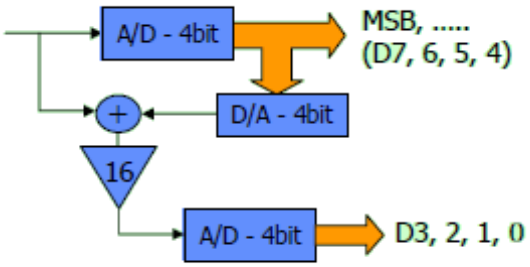
	Complex	Conv time	Latency
Parallel (flash)	2^N	1	1
Pipeline	N	1	N
Residue	N	N	N
Successive Approx	1	N	N
Tracking	1	2^N	2^N

Complexity: proportional to the number of comparators.
Conversion time: the maximum number of comparator delay (clock periods) to complete a conversion
Latency: delay to get the result

Mixed architectures

Can we obtain something better of the two best architectures, by introducing some trade-off architecture, which mix the benefits of the two previous? Well, the answer is yes, and solution is: *build residue systems on multiple bits*, instead of using bit-to-bit conversion.

An example of what we can do, is the following one: in order to realize a 8 bit A/D converter, we can do this:

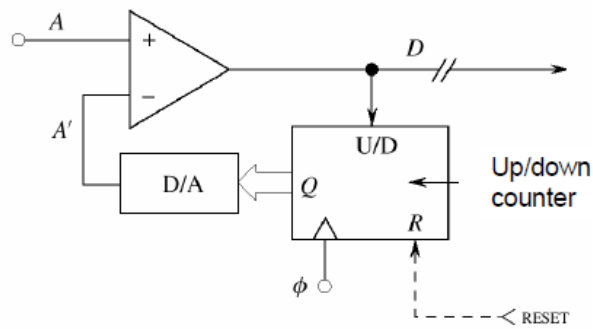


Using two cells of 4 bits, we can obtain directly the four MSB, so turn back to analog with a D/A converter, add the input signal (which was analog), and, multiplying by 16 (in order to shift of four positions our signal), use the same A/D cell previously used. This is very interesting: we need $2^4 - 1$ instead of $2^N - 1$ comparators; conversion time is $2T_c$, so, without exploiting this idea (this is a basic example), we can obtain very very good results. This architecture can be improved, as previously done, with **pipelining**, increasing the throughput of the system.

5.5 Differential converters

Differential converters are a family of special conversion circuits; formerly, these circuits were very useful for voice signals, but now are widely used for almost every type of application. As we will see, one of our goals will be the one to shift the complexity from the analog domain to the digital domain, where everything is simpler.

The idea is this: let's start from the old tracking converter:

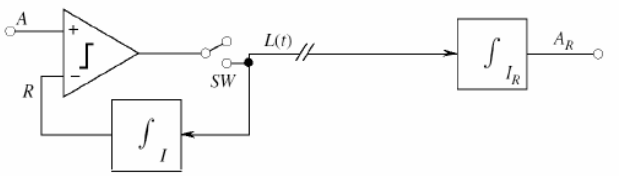


This converter was studied in the previous section, so we are not going to talk about it. As we can see, the output of the comparator is sent to the U/D counter, which sends another signal to the comparator, and the output will show if the new value is higher of lower of the signal. Differential converters are based on another idea: if we take as output directly the stream of values which controls the U/D counter, and we send it into a D/A converter, we have as output of the system a sequence of non-weighted values. Why this idea is useful? Well, non-weighted values are interesting because they provide informations about the behaviour of the signal: by reading "0" we know that signal is decreasing, by reading "1" we know that signal is increasing, so we can compare each value to the previous one, obtaining a differential information, a local information: we know, at every bit, if signal is increasing or decreasing respect to the previous one.

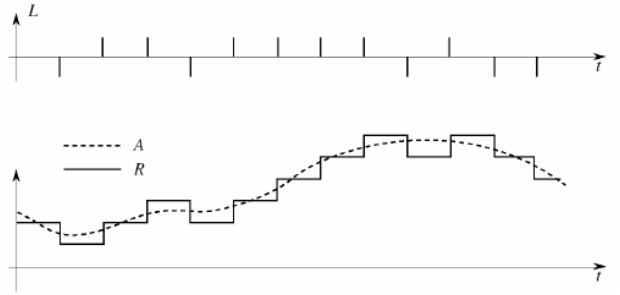
5.5.1 Δ converters

Now, let's try to simplify our circuit, without using D/A converters, which are a critical block (slow, and which must be precise): if we consider the U/D counter block and the D/A converter, we know that depending on the output of the U/D we will have a signal that goes up or down; we want a circuit which realizes a similar behaviour, with simple elements.

Now, let's try to simplify our circuit: if we put a switch out of the comparator instead of the U/D counter, and an integrator instead of the D/A converter (let's remark that the **combination** of the two blocks is equivalent to the previous one: each block has different behaviour, but by taking they together we obtain something similar), we have this behaviour: out of the switch we will have positive or negative pulses, depending on the output of the comparator; these pulses are sent to an integrator, which keeps in memory the previous state of the system, and produces an increasing ladder or a decreasing ladder, depending on the state of the switch.



Pulses are so obtained with the switch; they are produced at fixed rate, so with a clock CK, with frequency F_{ck} and period T_{ck} equal to the inverse of the frequency. The output of the converter can be high or low; out of the switch, in the instants when switch is closed, there are the values catched out of the comparator, so positive or negative pulses. The integrator is just a decoder of these pulses: it proposes, given as input a sequence of pulses, some signal shape.



This is still a tracking converter, but it uses a switch instead of the old circuitry; let's remark that this converter is used directly with changing signals, so it does not need any sampler: sampling process is internal of the system. Converters of this family are known as **Delta converters**, or Δ converters.

Parameters of the Δ converter

Let's try to characterize Δ converters with their main characteristics; we define γ as the absolute amplitude of a step. We want to identify the parameters (and the limits) of this system, in order to put it in relation with previous converters. Previously, we used two parameters, in order to characterize converters: the number of bits N, and the conversion time T_c; both N and T_c are not interesting, because Δ converters are 1-bit converters, which work at very high frequencies, realizing so a similar result respect to the old converters, with only 1-bit conversion; T_c is not necessary anymore, because we are not using any D/A in our new system, so we must find new parameters which can characterize and put in relation these systems.

We know that N was related to other parameters, in old systems, with this relation:

$$N \longrightarrow A_d = \frac{S}{1^N}$$

Increasing N, we reduce step size, increasing the resolution.

In Δ converters, the full-scale of the differential converter depends on **slew rate**: if signal changes too fast, converter cannot follow it; the maximum slew rate is defined as the maximum voltage change in a time, so:

$$\left.\frac{\Delta V}{\Delta t}\right|_{max} = \frac{\gamma}{T_{ck}}$$

Previously, γ was A_d, and it was related with 1 LSB; now, let's try to find something other. For sine waves, we have that:

$$\left.\frac{\mathrm{d}}{\mathrm{d}t} V_i \sin(\omega_i t)\right|_{max} = V_i \cdot \omega_i \sin(\omega_i t)|_{max} = V_i \cdot \omega_i$$

Converter can track only signals with a *full-scale equivalent* like this:

$$\frac{\gamma}{T_{ck}} = \omega_i V_i$$

So:

$$V_i = \frac{\gamma}{T_{ck} \cdot \omega_i}$$

This is the maximum amplitude that converter can handle; this corresponds, in old systems, to S.

We found the maximum value which can be handled by this system; and the minimum one? Well, with old converters, the minimum signal amplitudes that converters handled were the ones which might be recognized, respect to quantization noise; now, if signal has amplitude lower than $\frac{\gamma}{2}$, it is recognized as **idle noise**, so:

$$\frac{\gamma}{2} < V < \frac{\gamma}{\omega T_{ck}}$$

Numerical example and oversampling

Let's try to solve the following problem: we want a differential converter which provides the same performances in **dynamic range** of an 8-bits converter with 3 kHz signal, and sampling rate of 3 kS/s.

Before starting, a definition: we define the **dynamic range** as:

$$D_R = \frac{S_{MAX}}{S_{min}} = \frac{V_{MAX}}{V_{min}}$$

So as the ratio between the input voltage range bounds which guarantees that system works.

For the special converter, we have that:

$$M = 2^N = 2^8 = 256$$

Other specifications are not interesting: the sampling rate one guarantees only that sampling process satisfies the Nyquist's criteria, and other informations are not useful at all.

For Δ converters:

$$\frac{V_{MAX}}{V_{min}} = \frac{\frac{\gamma}{\omega T_{ck}}}{\frac{\gamma}{2}} = \frac{2}{\omega T_{ck}} = 256$$

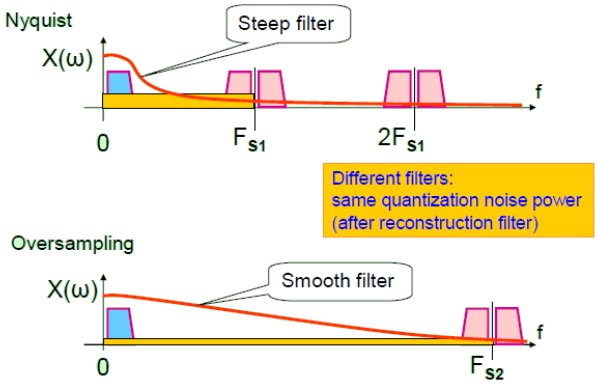
So the minimum sampling frequency must be equal to:

$$F_{ck} = \omega \cdot 128 = 2\pi \cdot 3 \text{ kHz } \cdot 128 \simeq 2,5 \text{MS/s}$$

This is a very high sampling rate.

Previously, converters needed a 64 kS/s sampling rate, now 2,5 MS/s: differential converters **must** work with **oversampling**, so with sampling rates very very higher respect to the Nyquist's frequency. There are bad and good news:

- Bad news: with sampling rate so high, we must handle many samples; many samples means that we must introduce memories, or something else;
- Good news: we are operating with very high sampling rates; this means that the aliases which derive from the the sampling process are very far, as known from the Signal Processing Theory; this means that like always done we had to remove with filters aliases, but now, supposing (as always done) that signals have a low-pass frequency behaviour, there are less contributes with the same filter; this means that we can relax the specifications of the filter, obtaining a less expensive device.

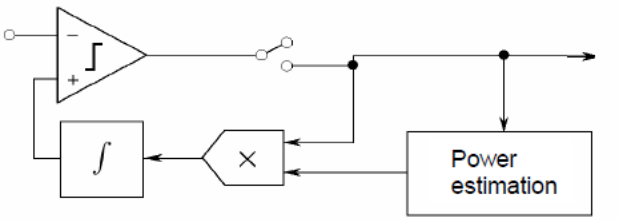


Another benefit: in '40s was proof that quantization noise has a spectral density that goes from 0 to the sampling rate frequency; if we increase the sampling rate, noise power becomes lower, because it is normalized with a higher divider; by increasing sampling rate, we reduce the spectral density amplitude, increasing performances.

Oversampling so has benefits respect to noise, to filtering, etc. , benefits paid with a very high amount of information to handle; this issue can be solved by putting a **decimation filter** after of the differential converter: a decimation filter is a digital filter which takes many bits to realize only one very precision bit, reducing the bit-rate; other filters are easier, because they can be realized, thanks to oversampling, also with simple RC cells!

Adaptive converters

There are some techniques which can increase the dynamic range without increasing too much the sampling rate; they are based on the idea to have different values of γ, depending on the output amplitude of the system. Let's consider a circuit like this:



Before introducing the integrator there is analog multiplier, controlled by a power estimator which introduces some gain (varying the input signal of the multiplier), obtaining pulses with variable amplitudes.

As we have already written, these converters are **differential**; if our system sees that the polarity of the comparator is changing all the times, means that signal is very low; the power estimator so sends a signal which

reduces the γ, in order to increase the resolution of our system; if there is a unipolar signal (so a sequence of ones or zeros), γ will increase, in order to handle fast signals.

$\Sigma - \Delta$ converters

Let's consider another solution, widely used in modern systems, and probably the most interesting of all ideas since 'till now. If we take sine signals, like:

$$v(t) = V \sin(\omega t)$$

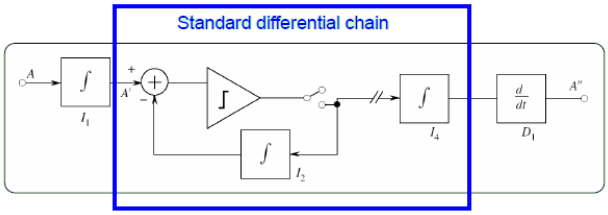
If we introduce it in an integrator, out of it we will have:

$$\int V \sin(\omega t) \mathrm{d}t = \frac{V}{\omega} \cos(\omega t)$$

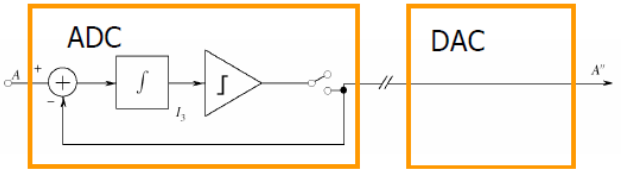
Let's consider the slew rate expression, with a signal treated with this operation:

$$S_R = \omega \cdot \frac{V}{\omega} = V$$

Very very interesting: the maximum slew rate of this signal is no longer depending on its frequency, but only on its amplitude, V! This idea can be implemented in a differential converter:



In this feedback loop, there is an integrator before every branch of the +; this means that we can simplify our chain, considering a system like this:



In fact, integration is a linear operation, so we can substitute the sum of two integrals with the integral of the sum of the terms, because of linearity; on the decoder side, we previously needed a differentiator out of the last integrator, that compensate the integrator. In this schematic there are no filters: this converter realizes some kind of sampling, so we must define the bandwidth before processing our signal, and cut-off every alias after the

process. Because of oversampling, these filters will be simple (RC-cells), so nothing hard.

Converters of this type are called **Sigma-Delta converters**, or $\Sigma - \Delta$ converters, and their main property is the fact that they have no dependence of dynamic range from frequency!

Now, let's take a look on quantization noise: we can show that this model has this transfer function:

$$\frac{Y(s)}{N(s)} = \frac{1}{1 + \frac{1}{s}} = \frac{s}{1 + s}$$

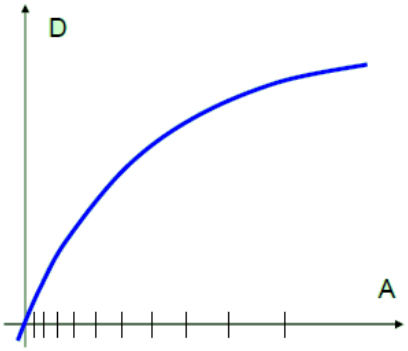
This can be proof with Automatic Controls theory. This transfer function has a **high-pass** frequency behaviour: noise is no longer flat, so we have **shaped** the spectrum of noise. This high-pass behaviour **reduces** the quantization noise in base-band, increasing the performances respect to the Δ converters.

Final considerations

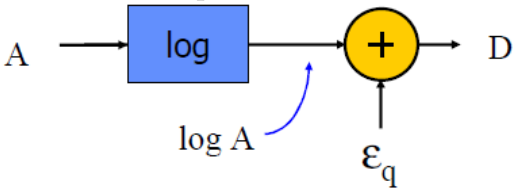
Usually, when we need high performances with only 1-bit converters, $\Sigma - \Delta$ converters are the best choice; this is a **first order** differential converter, because there is only one integrator; often there are **second order** converters, based on using two integrators instead of one; this increases the precision of noise rejection, obtaining performances similar to 24-bit converters, so very very high! These systems are very precise, with non-precise devices, so there are good results by starting from not excellent components.

5.5.2 Logarithmic converters

The analog to digital converters studied before can be used for almost every application; there are techniques which can be used for one of the most important signal processing application: voice signals processing. As known, voice signals are signals which have values near to zero for almost every time, and in some times have values close to the full-scale ones. For signal like these, the usual linear conversion is not the best one. In order to have a better resolution for signals with amplitude close to zero, an idea can be the one to introduce a non-linear transfer function, so a transfer function where, depending on the amplitude of the signal that must be converted, there is a different width of the intervals. For voice signals, we want narrower intervals close to zero, so the best shape can be the logarithmic one.



This is a weak reason to use logarithmic quantization instead of linear quantization: there is, in fact, a very important reason, which can drives us to use this technique instead of the other. Let's consider this block scheme:



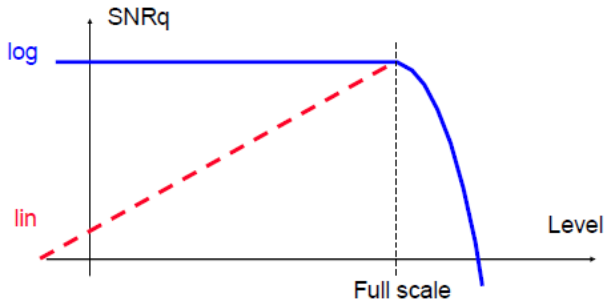
Out of our scheme we will have D, which is:

$$D = \log(A) + \varepsilon_q$$

The quantization error ε_q can be represented as the logarithm of some other value, K_q; we have that ε_q is a value similar to zero, so we must expect that K_q is similar to 1; we can express the previous expression with:

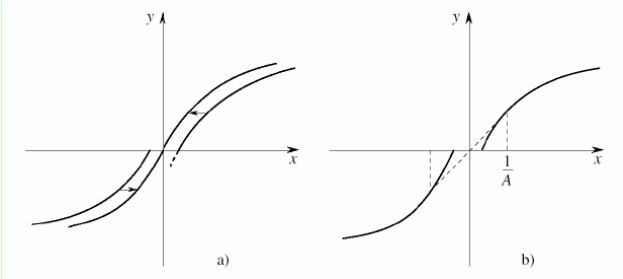
$$D = \log(A) + \log(K_q) = \log(A \cdot K_q)$$

This is very very interesting: with logarithms, we can see the additive error as a multiplication error; this is very good, because multiplicative errors change **linearly** the output of the system: if we have a constant ε_q, but a variable signal, the quantization error will affect the signal every time in a different way, so non-linearly, because there is no relation between quantization error and signal value; now, we found a relation, and this thanks to logarithms! In other word, now error is **relative** to the value of the signal. This can be explained with the *weak* reason: if we work with low values of the amplitude of the signal, we have a reduced step A_d, so we have a quantization noise reduced (due to the properties of the logarithmic function) in a way that can maintain constant the signal to noise ratio:



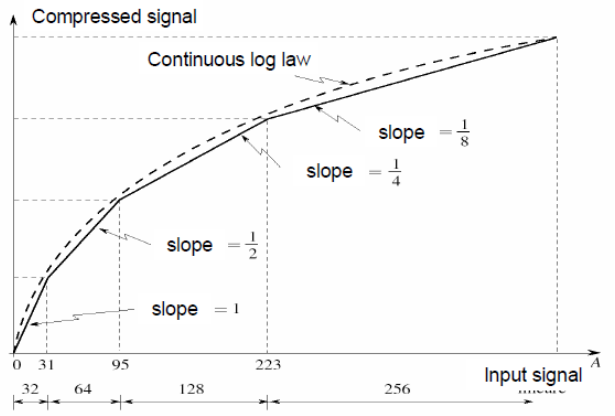
Now, let's think about circuits: the first thing we have to handle is that our signal is not only positive, but is **bipolar**: can assume both negative and positive values; we must introduce some variation from the original logarithmic function, in order to take account of this fact. Another fact: as we move close to zero, theoretically the logarithm's value must go close to $-\infty$. There are two solutions for this issue:

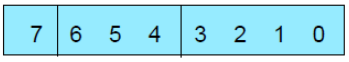
- Translation: we introduce a distance from the two characteristics, avoiding problems related to their original behaviour and obtaining bipolar support; this is done with μ-law encoders;
- Linearisation: we introduce, from a point identified by $\frac{1}{A}$, a linearisation of the behaviour of the characteristic near the zero amplitude, so an approximation of the characteristic with a line.



Since in both cases we do something different, something approximated respect to real logarithms, we can expect that what is predicted from the maths is different from what we will obtain actually.

There is another really bad approximation: logarithms are not represented, obtained with their continuous shape, but with a piecewise approximation of the theoretical function. Piecewise approximation, as previously studied, means that the shape of the function is approximated with a set of lines, one put after the other. With logarithmic functions, we have, for each point, that the same distance represent the same ratio; in piecewise approximation we can imagine that this behaviour is *globally* satisfied, but not locally: we divide the function in **segments**, which are divided in **levels**:

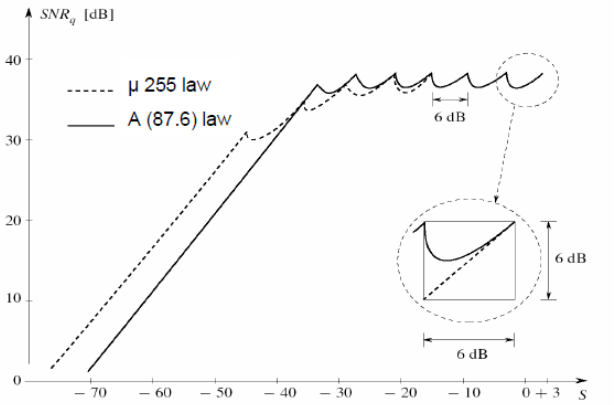




There is a **sign** parameters, so a bit which can identify the polarity of the signal, a **segment** parameter, which can identify which line we are considering, and a **level** parameter, that identifies which point of the considered segment are we considering.

This approximation introduces a very bad issue: on each segment, we have a **linear approximation** of the logarithm; this means that, on each segment, **the quantization error is constant**: on every line there is a non-linear error! In fact, signal changes its value, but quantization error no, because this is a linear approximation of a logarithm: from two segments there is a change of the quantization error, but inside the segment we can not have this benefit. Another issue: if we consider points near to the bound of the segment, with points which are right respect to the bound we will have a quantization error, with points which are left respect to the bound a lower quantization error, and almost the same signal amplitude! If from each interval to the previous/next one there is a ratio of 2, we will increase or decrease the signal to quantization noise ratio of 6 dB, because maintaining the same amplitude of the signal, and reducing to half the quantization's noise one. From the other side, if we go from the left bound to the right bound of the segment, we will decrease the signal to noise ratio of 6 dB, because we are saying that each segment is realized in order to duplicate (or reducing to the half of) the quantization error, increasing or decreasing the width of the A_d intervals: **into the intervals, the error is not relative, but constant**.

The actual behaviour of the signal to noise ratio will be the following one:



There are ripples where we expected to have a flat behaviour; these ripples are wide 6 dB, because of the considerations previously done.

With the new technologies, logarithmic quantization is realized with digital circuits, using an ADC before the quantizer and processing with a DSP.

5.5.3 Waveform and model encoding

In this subsection we want to introduce a different way to encode signals, without introducing any formal idea. Until now, we have sampled a signal, so processed it and sent with some transmission circuit; given a sequence of numbers, we sent they in order to obtain the signal our of the receiver.

This is a good idea, but not the only one: every sine signal can be characterized with three parameters: amplitude, frequency, phase. If we send these three parameters, instead of the samples of the sine wave, we can obtain, out of the receiver, the same information. This can be done only if we have in both receivers a **model** of the signal: send these three parameters can be useless, if from the other side of the telecommunication system there is no way to understand what they mean!

Model must be well-known in each system, in order to avoid misunderstandings; we wish to obtain also parameters with a good correctness and resolution, in order to apply they on the model and obtain correct informations.

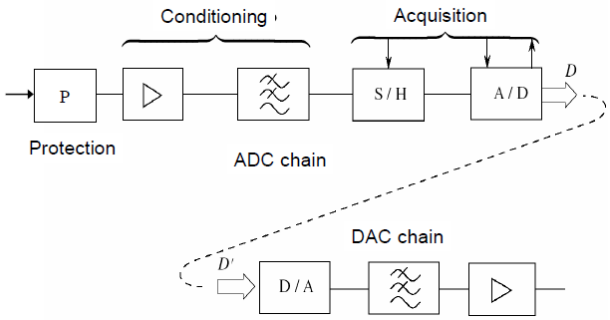
With this idea can be realized models of signal very interesting, like the voice ones: by modelling the larinx, we can obtain a model on whom we can apply our parameters, and obtain the voice transmission/reception. With these ideas, there are two *quality standards*:

- Recognize the interlocutor: in application like the telephonic ones, a good result is permit to the users to recognize the interlocutor;

- Understand the speech: if we ask less information, we can only want to understand the speech; this can be useful with alerts or some situations like these.

5.5.4 A/D/A systems

A complete analog to digital to analog system (the most used chain, in order to take a signal, process it with DSP and take it out with a usable format) can have a block schematic like the following one:



About this chain, we will study only two things:

- How to design the characteristics of the antialiasing filter;
- How to quantify the global errors of the chain.

Antialiasing filters

Every signal has a useful bandwidth f_B; for **useful bandwidth** we mean that portion of the spectrum of the signal which contains useful informations to process; every signal exists for a finite time, so its spectrum has no bound: this means that we have aliasing effect. Supposing to not know anything about the signal but its useful bandwidth, we can assume that it is flat, in the frequency domain; filter must be designed in order to limit bandwidth as specifications say, but how much it must filter must be defined starting from other specifications: if our filter is badly designed,it will filter, but keeping high noise amplitudes. The goal of this subsection is to introduce a way to understand how much our filter must filter, quantifying the number of poles which it must introduce.

Let's start from an observation: a 1-order filter, so an RC-cell, has a decreasing behaviour, with a slope (in the Bode plot) of - 20 dB/dec ; this means that the amplitude-bandwidth product is constant, so that with a

frequency increase there is a linear amplitude decrease, in decibel. For P-order filters, there will be something similar to this: now the amplitude-bandwidth product is not constant anymore, because it will decrease more quickly: the amplitude and the frequency are still related, but with a different slope: instead of - 20 dB/dec, there will be a slope of - $P\cdot$ 20 dB/dec, so a difference of P.

This idea can be very useful when we want to have a well-specified signal to noise ratio: if we want for example a signal to noise ratio of 40 dB from 12 to 38 kHz, we must:

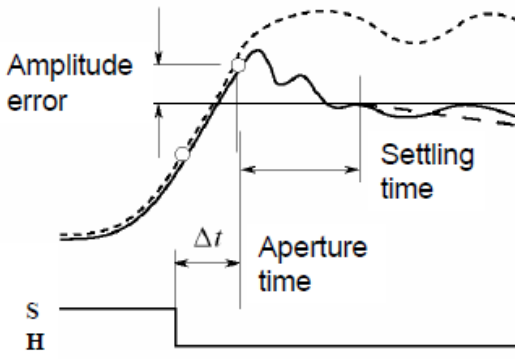
1. Evaluate in decibel the ratio of the frequencies;
2. Determine the number of poles necessary to obtain this signal to noise ratio, by doing:

$$P = \frac{\text{Desired SNR}}{\text{Frequency range ratio in decibel}}$$

Sample and hold circuits

As known, the sample and hold circuit must sample a value, and hold it in the output until conversion process is ended; these circuits are not ideal, so there are many error sources.

The worst error in this block of the system is obtained when the circuit goes from the **sample state** to the **hold state**: the **sampling jitter**: from the instant when the sampling and hold circuit receives the **open** command (in order to end to charge the capacitor and hold this voltage level to the output), and the instant when switch opens, there is a non-zero time. In this time, that we will call T_A, the circuit continues to charge the capacitor, changing the voltage at its pin, obtaining a different output voltage respect to the expected one.



Now, there is a problem: T_A is not a constant time, because it is affected by noise; due to this reason, we have to consider T_j, so the **jitter time**. It can provide, with a sine wave, a voltage change of ΔV:

$$\Delta V = T_j \cdot \mathrm{SR} = T_j \cdot \omega \cdot V_i$$

Effective Number Of Bits

We are going to end this chapter, introducing a parameter which can quantify the total error in the A/D/A system, with the already seen considerations and approximations. This chapter can be derived with an idea deriving by one already seen in these notes: the quantization error one.

For our system, we consider many error sources; we can take account of all of them, by computing a signal to noise ratio for the whole chain; every error source is stochastically independent from the others, so we can evaluate the total noise power by adding the noise power of each block.

What we are interested to define is something similar to the number of bits of a quantizer: previously we proofed a formula which put in relation the number of bit of the quantizer and the quantization error; what we are going to do is to define an **equivalent number of bits**, so not a real number of bits, but only a parameter which says how many of the output bits of the system are significant. Given the total signal to noise ratio SNR_T, we can compute it as:

$$\mathrm{SNR}_T = 6 \cdot \mathrm{ENOB} + 1,76\mathrm{dB}$$

So:

$$\mathrm{ENOB} = \frac{\mathrm{SNR}_T - 1,76}{6}$$

Surely, this number will be lower than N, so of the number of bits of the quantizer: quantization error is only one of the many errors in our system, so when we are taking into account all of them, we have a decrease of the signal to noise ratio respect to the quantization noise one, and the equivalent number of bits will be lower respect to the available one.