WIreless DEvelopment LABoratory (WIDELAB) Equipment Base

Per Zetterberg

2003-03-16

IR–SB–IR–0316
Abstract

This paper documents some of the hard- and soft-ware used by the signal-processing group at KTH, in its experimental work on wireless systems. We call these items collectively Wireless DEvelopment LABoratory (WIDELAB), and this paper aims to document the most important pieces and some of the knowledge needed to understand, operate and develop them further. To access the hard and software described contact perz@s3.kth.se.

1 Hardware

1.1 Receiver Modules

The most important piece of equipment in WIDELAB are the receiver and transmitter modules. The function of these modules is to downconvert a narrow-band (typically $\approx 11$kHz band-width), to base-band to make it possible to A/D convert and process the data with DSP or PC. The signals are downconvert to a center frequency of 10kHz (a signal of 4kHz bandwidth would occupy the frequency range 8kHz to 12kHz) to make digital I&Q extraction possible (see also Section 1.9.2 and 1.10.1 below).

Figure 1 below shows a receiver module. The receiver module is mountable in a rack as shown in e.g. Figure 7 below. When used in a rack the accessible connectors are those with a number 1-10 in Figure 1. The function of these connectors are described in item 1-10. Item 11-23 describe the internal components of the receiver module. Most of the components are from Mini Circuits (www.minicircuits.com), and the specification are downloadable from their webpage. The signal flow from component input (item 1 below) to the output (item 4 below) goes through the components in the following order 1,11,12,19,23,21,22,20,18,16,17,24,15,14,4.

1. RF input. This is where the input signal enters. The frequency may be in the range 300-3GHz, depending on which filter is used, see item 12 below. The signal shall have bandwidth of less than 14kHz. Typical bandwidths are 3-11kHz. The typical usable signal range is -120dBm to -40dBm. To prevent the receiver from damage, the input level should be kept below -20dBm. (By level we here mean power, 0dBm means 1mW. All impedances are 50 ohm)

2. LO 1 input. Local oscillator input for downconversion to the IF frequency 70MHz. If the frequency of the RF input is $f_c$, then the frequency of this input should be $f_c+70$MHz (high LO) or $f_c-70$MHz (low LO), see Section 3 below. The level shall be +7dBm.

3. LO 2 input. Local oscillator input for downconversion from the IF frequency 70.000MHz to the base-band 10kHz. The frequency of this signal shall be 69.990MHz and the level +7dBm.

4. Base-band output. This base-band signal is the input to A/D converters. The signal is a copy of the RF input translated in frequency to a nominal carrier of 10kHz.
The maximum output swing is +12V. Most A/D converters can not take this high output signal amplitude and need to be protected by resistive dividers or diodes.

5. Attenuator control input 1. When the attenuator control inputs 1 and 2 are both grounded (0V), the gain of the receiver module is at it’s maximum (from RF input to Base-band output). When the attenuator control 1 is set to 5V (TTL), the gain is reduced 10dB. **Warning! if the attenuator control voltage exceeds 5.5V the attenuators are damaged!**

6. Attenuator control input 2. When this input is set to 5V then the gain is reduced 20dB. If both attenuators are active, the gain is reduced 30dB. **Warning! if the attenuator control voltage exceeds 5.5V the attenuators are damaged!**

7. Ground. Connect this with the ground of the computer.

8. Strong DC feed. This connection feeds the downconverted unit with a voltage of 18 to 20V and a current of 0.25A.

9. Low-noise negative DC feed. This feeds the base-band amplifier with -18 to -19V. The current is 6mA. It is critical that the noise level of this feed is low.

10. Low-noise positive DC feed. This feeds the base-band amplifier with +18 to +19V. The current is 6mA. It is critical that the noise level of this feed is low.

11. Amplifier MC ZJL-3G. Sets the achievable noise figure of the receiver, although for some frequencies and set-ups, other noise sources also contribute.

12. Band select filter. For the DCS1800 band the filter MC SHP-1000 is used. This gives no image rejection which increases the noise level. For the NMT450 band, the two filters SHP500 and SLP550 are used.

13. Voltage controllers. This unit creates 5V, 12V and 15V voltage from the strong DC feed, see number 6 above.

14. Base-band amplifier, based on the operational amplifier AMP-01 from analog devices. The amplifier is configured to have an input impedance of 50 ohm. The voltage gain between the input and output is 60dB, in frequency range of about 3kHz to 20kHz.

15. Low-pass filter. Removes any high frequency components (e.g. 70MHz and 140MHz could exist).

16. Fixed attenuator. These fixed attenuators are selected to make the gain of different copies of the receiver modules approximately equal (fine-tuning for instance direction of arrival estimation has to be done in software). Have to be changed when the RF frequency is changed.

17. Channel filter. Crystal filter from TEW. The transfer function of the filter, see Figure 2. Creates the narrow-bandwidth of the receiver.
18. Amplifier MC ZFL-500HLN. Amplifies the 70MHz intermediate IF signal.

19. Mixer MC ZEM-4300. Downconverts the RF input to 70MHz by mixing it with the LO 1 input.

20. MC ZFAT-51020. Digital step attenuator. Attenuates the 70MHz signal based on the attenuator control inputs (item 5-6).

21. Amplifier MC ZFL-500HLN. Amplifies the 70MHz intermediate IF signal.

22. Channel filter. Crystal filter from TEW. The transfer function of the filter, see Figure 2.

23. Filter MC SBP-70. Wide bandpass filter with 70MHz center frequency 70MHz. Used to block remove signals several MHz outside of 70MHz.

24. Mixer MC ZLW-2. Downconverts the 70MHz IF signal to base-band by mixing it with the LO 2 signal (see item 3 above).

Figure 1: Receiver module

1.2 Transmitter Modules

The transmitter modules are used to up-convert a narrow-band base-band signal with a center frequency of 10kHz to a radio frequency. This is done in order to be able to communicate signals over radio that are generated with DSP or PC. The transmitter modules have identical hardware to the receiver modules. In fact, they are obtained by re-configuring the cables inside the receiver modules. The reason why this is possible is since the mixers can be used both for signal up- and down-conversion. To indicate that a module is configured as transmitter the letter “TX” are indicated as is visible on one of the transmitter modules in figure 7. When used in transmit configuration, the front panel connectors are used as follows
1. **RF output!** This is where the output signal leaves the module. The frequency may be in the range 300MHz-3GHz, depending on which filter is used, see item 12 above. The signal will have bandwidth of less than 14kHz. The power can be up to a few dBm.

2. LO 1 input. As in receiver.

3. LO 2 input. As in receiver.

4. **Base-band input!** If the current exceeds 40mA the module is damaged. This condition should be satisfied if the voltage is less than 0.1v.

5. Attenuator control input 1. As in receiver.

6. Attenuator control input 2. As in receiver.

7. Ground. Connect this with the ground of the computer.

8. Strong DC feed. This connection feeds the downconverted unit with a voltage of 18 to 20V and a current of 0.25A.

9. Not used. No need to connect.

10. Not used. No need to connect.

The signal flow from base-band to RF flows through the components of the module in the following order: 4,24,23,21,22,20,18,16,17, 19,11,12,1. The most important steps when re-configuring the module from RX to TX are the following:

1. Mount a cable between the front connector, 4, and the ”I” input of the mixer, 24.
2. Mount a cable between the "R" connector of the mixer, 24, and the input of the filter 23.

3. Connect the output of the filter 17 to the "I" connector of the mixer, 19.

4. Mount a cable between the "R" connector of the mixer, 19, and the input of the amplifier, 11.

5. Connect the output of the filter, 12, to the connector, 1.

Note that is easiest to just add cables and not remove any (since they are often glued to the back-plane), this also makes it much easier to change it back to a receiver.

When used in the 2x2 MIMO set-up, see below, the RF output level of the TX module is only about -2dBm. The power can be increased up to 5dBm or so by simply inserting an operational amplifier in the base-band. However, for the laboratory purposes foreseen so far there is no reason to do so.

### 1.3 HP Signal generators

The department has one signal generator HP8648 and two HP8656B. The frequency range covers 100kHz to 990MHz. The generator HP8656B can deliver up to 17dBm while the generator HP8648 can deliver only up to 13dBm. These generators are used as local oscillators (LO 1 or LO 2, see Section 1.1 and 1.2) or as sinusoidal radio sources. The frequency is determined by pressing the "frequency" button followed by the frequency and then pressing one of the four unit buttons for (GHz, MHz, hz, Hz). The amplitude is varied by pressing "level" followed by the amplitude in dBm and the pressing the button with "dBm".

### 1.4 Marconi Signal Generators 2024

There are two Marconi Signal Generators 2024. They have a frequency range of 9kHz to 2.4GHz, and an power range of -137dBm to 13dBm. There exists a manual for these units. The output impedance is 50ohms. The power of 0dBm means that 1mW is fed into a 50ohm resistor. The frequency accuracy is about ±5ppm (part per million). Thus when the generator is programmed to generate a 1MHz signal, the actual frequency can be anywhere in the range of 999995Hz to 1000005Hz. So far, we have only used the generators to produce carrier waves (CW) (by this is meant a sinusoid). These are used either as a local oscillator signal for the receiver/transmitter modules or as a sinusoidal source signal. The frequency is determined by pressing the "carrier freq" button followed by the frequency and then pressing one of the four unit buttons for (GHz, MHz, hz, Hz). The amplitude is varied by pressing "RF Level" followed by the amplitude in dBm and the pressing the button with "dB". The Marconi generators can also be controlled by the software as described below in section 2.1, below.
1.5 69.990MHz local oscillators

Four 69.990MHz generators have been built based on the crystal oscillator VCX02-BV5 from Micro Crystal, see item #1 in Figure 11. When connecting the units to a Mini Circuits SBP 70 filter, the output is an accurate sinusoidal of approximately 69.990MHz frequency. The output power is about +15dBm. The frequency of the units can be tuned using a variable resistor which is accessible with a screw-driver without opening. Typical frequency errors are in the range +500Hz.

1.6 Rohde Schwartz Spectrum Analyzer FSH3

This is hand-held spectrum analyzer with tracking generator. It is capable of estimating the spectrum of signals in the range from a few hz up to 3.0GHz. The resolution bandwidth can be set as low as 1kHz which enables estimation of the spectrum for the narrow-band signals generated by the TX module of Section 1.2. The maximum signal level is +20dBm. The author can recall that it is possible to measure signals down to -90dBm using a resolution band-width of 100kHz, which would imply that -110dBm is feasible with 1kHz band-width. The spectrum analyzer also has an output signal - known as the traffic generator. This output signal enables the instrument to be used as a network analyzer (i.e. it can measure the transfer function phase and amplitude, in the frequency domain, of filters.). The level of the output is around -20dBm. This output signal can also be utilized as a signal source. The FSH3 also has a built in battery. The FSH3 can be connected to the serial port of PC, and there is a handy software called FSH View which enables spectrum measurements to be downloaded to the PC. It is possible to export spectrums in ascii format from FSH View which enables the data to be loaded into matlab (requires a little bit of programming effort as well).

1.7 The EVM Boards

The project course in signal processing and communications 2E1366, has since the spring 2000 been running using so called evaluation module (EVM) DSP boards. These boards has a Texas instruments TMS320C6701 floating point signal processor, a PCI interface, and two A/D and two D/A converters (codec). The boards are mounted inside PC computers called the host. The host downloads program and data to the EVM board. There is a PC program “code composer studio CCS” for developing and debugging programs. Application programs for the DSP can also be developed that communicates with the DSP while it is running. For instance when the DSP is performing tasks such as modulation, beamforming and coding, the PC provide data to be transmitted e.g. for file transfer, and in the receiver receive the bits and re-construct the file. The two A/D and D/A converters are accessible from the front of the EVM board using stereo-plug connectors. An interfacing unit, see Figure xxx, has been built. This unit converts between the stereo-plug connectors and BNC connectors. The unit also contains protection of the A/D and D/A connectors from to high voltages and loads, respectively. The input voltage has the range +-3V, but the diode in the interfacing units starts to introduce non-linearities when the level reaches 3V. The output maximum swing is +-1.4V. The short-circuit protection
in the interfacing unit reduces this switch to ±1.3V. The impedance of the equipment connected to the output must exceed 10kohm. More information about the EVM board can be found in [1].

1.8 The Multi-Channel Daughter Board

One of the EVM boards at the department has a medium acquisition speed multi-channel A/D board, called AED-106. This board is from signalware (www.signalware.com). Tomas Sklerno of signal-processing, S3, programmed the FPGA of this board to enable the use of low bandwidth channels. The software needed to run the multi-channel card is described in Section 2.2, below. The input level of this board must not exceed ±1V. An interfacing unit to be used with the receiver modules is described in Section 1.1 below.

1.9 1x3 (WideBand) Channel Sounder Configuration

This section describes the channel sounder configuration. If the Marconi signal generators, see Section 1.4, are used as local oscillators (LO 1, see Section 1.1), then the operating frequency can, called “fc” in the development in Section 1.9.2 below, can be slowly varied, see Section 2.1 on how to do this. If the environment remains stationary (no moving people), then the measurements at different frequencies can be taken as measurements of the same channel over a large bandwidth. Since the signal generators are running unlocked, a reference signal is also needed as a phase reference.

Section 1.9.1 below describes the set-up, and then Section 1.9.2 gives the mathematical description of the signal flow. The description is for a DCS1800 band application 1710-1880MHz.

1.9.1 Set-up

The set-up consists of a Marconi signal generator as source, see Section 1.4 above. Four receiver modules are used in the receiver, see Figure 1.1. The RF input of receiver module one, two and three are connected through cables to an array of Huhber-Suhner antennas. The signal from the generator is splitted using a Mini-Circuits splitter ZFSC-2-2500. One of the outputs is connected to a Huhber-Suhner antenna transmitter antenna xxxx. The signal level should be -40dBm or less to avoid disturbing DCS1800 mobile-phones. One of the outputs of the splitter is connected to a cable that runs to one of the receiver units, let us call it receiver number four. The cable should be built up by several short cables, with carefully selected attenuators. The attenuators should be such that the level of the signal on receiver module four is approximately the same as that on receiver number one, two and three. It must also be made sure that the cable does not radiate any signal to receiver antennas.

As local oscillator LO 1, see Section 1.1 above, a Marconi signal generator is used, see Section 1.4 above. The Marconi generator is set to maximum power i.e. +13dBm. This signal is splitted using a Mini-Circuit splitter ZB4PD-42 which is mounted in the rack, see Figure 3. As local oscillator LO 2, see Section 1.1 above, a HP signal generator HP8656B is used which is splitted using a Mini-Circuits splitter ZFSC-8-4-75. This generator is also
set to its maximum output power, in this case +17dBm. The outputs of the splitters are connected to the local oscillator inputs of the four receiver modules. The output of the receiver modules are connected to the conversion box shown in Figure 4. This box ensures that the signal level sent to the multi-channel daughter card, see Section 1.8, does not exceed +-1V. The circuit diagram of the conversion box is shown in Figure 5 below.

All attenuator control inputs are grounded. This selects the lowest signal level range for the receiver modules. If the signal level at the receivers are stronger than about -60dBm then sampled signals starts to get distorted. If this happens, then reduce the output power of the signal generator used as transmitter.

To calibrate the system, the signal source signal generator is set to a low level, say -90dBm, and then splitted using another Mini-Circuits splitter ZFSC-8-4-75. The outputs of the splitters are then connected to the antenna cables of the receiver modules (i.e. we calibrate the receivers + the cables between the antennas and the receiver modules). This ensures that the true phase of all input signals have the same phase and level, and thus enables calibration of the hardware.

The cable between the transmitter and receiver module number four, must also be calibrated, i.e. its phase and attenuation over frequency is needed to be known. How, to accomplish this is left as an exercise for the reader.

![Figure 3: (WideBand) Channel Sounder Rack](image)

1.9.2 Signal flow equations

The signal emitted from the signal generator is given by

\[ S_{emitted} = \text{signal generator output} \]

The four outputs of the splitter can actually differ some tenth of a dB and a few degrees, however by measuring these “errors” using a network analyzer, they are known and can be compensated for in software.
Figure 5: Schematic of multi-channel conversion box. The 300ohm resistor is on the daughter-board

\[
\sqrt{P_i} \Re\{\exp j2\pi(f_c + f_o)t\},
\]
where \(f_c\) is the desired frequency and \(f_o\) is a frequency offset.\(^2\) The signal received on the inputs of the receiver modules (+ cable) \(k = 1, 2, 3, 4\) are given as

\[
x_{m}^{RF-in}(t) = \sqrt{P_i} \Re\{H_k(f_c) \exp j2\pi(f_c + f_o)t\},
\]
where \(H_m(f_c)\) with \(m = 1, 2, 3\) is the transfer function between the transmitter antenna and receiver antenna number 1,2 and 3, respectively. The transfer function of the cable is \(H_4(f_c)\). A simple model of the receiver is shown in Figure 6, where the filters represents the impact of all amplifiers, filters and cables in each stage. As a model of a mixer we use

\(^2\)we here take the local oscillator in the receiver as the reference, alternatively the error can be attributed to the local oscillator.
output(t) = oscillator(t)input(t) + Aoscillator(t). \hfill (3)

The first term is the desired “ideal” mixer, the second term is a local oscillator leakage term. In practice there are several more. In our case the local oscillator leakage is about 30dB weaker than the first desired term. The local oscillator used as Lo 1 in this case is

\[
\text{oscillator}_1 = \Re \{\exp(j2\pi(f_c + 70MHz)t)\}. \hfill (4)
\]

Combining (1-4) and using the property

\[
\cos(\alpha) \cos(\beta) = \frac{1}{2} \cos(\alpha + \beta) + \frac{1}{2} \cos(\alpha - \beta), \hfill (5)
\]

yields

\[
x_{\text{IF}}^m(t) = \sqrt{P_l} |H_{\text{RF}}^m(f_c)||H_{k}^m(f_c)| \cos(2\pi(2f_c + f_o + 70MHz)t + \tilde{\gamma}_m) +
\]

\[
\sqrt{P_l} |H_{\text{RF}}^m(f_c)||H_{k}^m(f_c)| \cos(2\pi(70MHz - f_o)t - \tilde{\gamma}_m) + A \cos(2\pi(f_c + 70MHz)t). \hfill (6)
\]

where

\[
\tilde{\gamma}_m = \angle(H_k^m(f_c)H_{\text{RF}}^m(f_c)). \hfill (7)
\]

After the bandpass filter \(H_{\text{IF}}^m(70MHz - f_o)\) unwanted terms are removed

\[
x_{\text{IF}}^m(t) = \sqrt{P_l} |H_{\text{RF}}^m(f_c)||H_{k}^m(f_c)||H_{\text{IF}}^m(70MHz - f_o)| \cos(2\pi(70MHz - f_o)t - \tilde{\gamma}_m) \hfill (8)
\]

where

\[
\tilde{\gamma}_m = \angle(H_k^m(f_c)H_{\text{RF}}^m(f_c)) - \angle(H_{\text{IF}}^m(70MHz - f_o)). \hfill (9)
\]

Using the mixer model (4) again, with

\[
\text{oscillator}_2 = \Re \{\exp(j2\pi(69.990MHz)t)\}, \hfill (10)
\]

we obtain
\[ x_{m}^{\text{BB-in}}(t) = \sqrt{P_t} |H_m^{\text{RF}}(f_c)||H_k(f_c)||H_m^{\text{IF}}(70\text{MHz} - f_o)| \cos(2\pi(10\text{kHz} - f_o)t - \tilde{\gamma}_{m}) + \sqrt{P_t} |H_m^{\text{RF}}(f_c)||H_k(f_c)||H_m^{\text{IF}}(70\text{MHz} - f_o)| \cos(2\pi(139.990\text{MHz} + f_o)t - \tilde{\gamma}_{m}) A \cos(2\pi 69.990\text{MHz}t) \]

In the base-band filter, the high frequency terms are removed leaving

\[ x_{m}^{\text{BB-out}}(t) = \sqrt{P_t} |H_m^{\text{RF}}(f_c)||H_k(f_c)||H_m^{\text{IF}}(70\text{MHz} - f_o)||H_m^{\text{BB}}(10\text{kHz} - f_o)| \cos(2\pi(10\text{kHz} - f_o)t - \gamma_{m}) \]

where

\[ \gamma_{m} = \angle(H_k(f_c)H_m^{\text{RF}}(f_c)) - \angle(H_m^{\text{IF}}(70\text{MHz} - f_o)H_m^{\text{BB}}(10\text{kHz} - f_o)) \]

After A/D conversion, we may inside the DSP, create the complex valued signal

\[ x_{m}(t) = h_{\text{LP}}(t) \ast (x_{m}^{\text{BB-out}}(t) \cos(2\pi 10\text{kHz} t)) + j h_{\text{LP}}(t) \ast (x_{m}^{\text{BB-out}}(t) \sin(2\pi 10\text{kHz} t)), \]

where \( \ast \) denotes convolution. Combining (12) and (14) yields

\[ x_{m}(t) = \sqrt{P_t} H_m^{\text{RF}}(f_c) H_k(f_c) H_m^{\text{IF}}(70\text{MHz} - f_o) H_m^{\text{BB}}(10\text{kHz} - f_o) H_{\text{LP}}(-f_o) \exp(-j2\pi f_o t), \]

where \((\cdot)^c\) denotes complex conjugate and \(H_{\text{LP}}(-f_o)\) is the frequency response of the low-pass filter.

### 1.10 2x2 MIMO Set-up

The MIMO set-up is presently used in a configuration of the NMT450 band. S3 has currently a license for NMT450 channel number 115. This correspond to the two center frequencies 455.850MHz (uplink) and 465.850MHz (downlink). Since in the MIMO set-up we transmit simplex (in one direction), two different MIMO set-ups can be operational at the same time if they do not use the same frequency.

As transmitter the MIMO set-up utilizes two transmitter modules, see Figure 7. The transmitter can be distinguished from the receiver by the “TX” marking on the transmitter modules. The RF output, see item 1 in listing in Section 1.2 above, is connected to the filter shown in Figure 8. If the 455.850MHz frequency is used then it is connected to the SMA connector marked “RX”, if the 465.850MHz frequency is used then it is connected to “TX” \(^3\). The 7/16 connector is connected to Allgon RA3132 NMT450 antennas. One such antenna is visible in Figure 7. The local oscillator (LO 1) is obtained by a HP or Marconi signal generator (see above), which is fed to a splitter (item 4 in Figure 11

\(^3\)This seemingly strange marking is used since the filter is intended for NMT450 base-stations.
The generator should be set to +11dBm since the loss in the splitter and cables is about 4dB. The frequency is set to \( f_c - 70\text{MHz} \). The second LO (LO 2) is obtained from a 69.990MHz generator (see also Section 1.5) above, which is filtered and split before being fed to the receiver units, see item #1, #2 and #3, in Figure 11 below. The base-band input (see item #4 in Section 1.2 above) is obtained from the SMA connectors transmitter resistive matching box. (item #1 in Figure 12). The transmitter resistive matching box is connected to the output of the EVM board, see also Section 1.7. The transmitter resistive matching box simply interfaces between the impedance level seen by the EVM board D/A output and the low level required by the transmitter module as its base-band input. This is achieved using only two resistors as shown in Figure 9. The transmitter resistive matching box features the two letters “TX” to distinguish it from the similar receiver resistive matching box, see below.

The attenuator control inputs are obtained from the box which is indicated as item #2 in Figure 12. By operating the two switches the to BNC outputs are set to 0V (position up) or 5V (position down).

The receiver looks almost identical to the transmitter. The differences are that receiver modules are used instead of transmitter modules. Moreover, an external filter are not used and thus the antennas are directly connected to the RF inputs (see item #1 in Section 1.1). Another difference is that the receiver LO 1 should be set to \( f_c + 70\text{MHz} \) (the power is 11dBm as in the transmitter). The base-band outputs are connected with the EVM board A/D input through the receiver resistive matching box. This looks identical to the transmitting box but is not marked with “TX” and is used to reduce the output swing by resistors as indicated in Figure 10.

![Figure 7: Two antenna transmitter](image)

1.10.1 Signal flow equations

Transmitters:
Figure 8: NMT450 filter.

Figure 9: Circuit schematic of the transmitter resistive matching box.

Figure 13 below shows a model of the $k$th transmitter module ($k = 1, 2$). We assume that the modulated complex valued (i.e. with real and imaginary part) base-band signal $s_{in,k}(t)$ has a bandwidth of a few kHz (typically 10kHz). By digital processing inside the DSP the output signal from the the D/A converter is generated as

$$y_{k}^{BB-in}(t) = \Re\{s_{in,k}(t) \exp(j2\pi 10kHz t)\}. \quad (16)$$

The following model is used for mixers in this development

$$output(t) = oscillator(t)input(t) + Aoscillator(t). \quad (17)$$

In practice there are also other terms but these are neglected here, see Section 4. The second term is called “oscillator leakage”, and the constant “A” models the strength of this signal. The first oscillator has 69.990MHz frequency i.e.

$$oscillator_1(t) = \cos(2\pi 69.990MHz t) \quad (18)$$

The relationship

13
Figure 10: Circuit schematic of the receiver resistive matching box.

Figure 11: left-hand side of two antenna transmitter. 1) Local oscillator 69.990MHz (LO 1), 2) Bandpass filter SBP-70, 3) Splitter for LO 2. 4) Splitter for LO 1.

\[ \Re\{z_1\} \Re\{z_2\} = \frac{1}{2} \Re\{z_1z_2 + z_1z_2^c\} \]  
(19)

between the complex numbers \(z_1\) and \(z_2\) will be frequently utilized in the following derivations. Using the model indicated in Figure 13 together with the equations above (and neglecting the influence of the base-band filter) yields

\[ y_k^{IF-in}(t) = \Re\{\exp(j2\pi 70MHz t)s_{in,k}(t) \}
+ \exp(j2\pi 69.980MHZ t)s_{in,k}^c(t)\} + A \cos(69.990MHz t) \]  
(20)

where \(s_{in,k}^c(t)\) is the complex conjugate of \(s_{in,k}(t)\). After the bandpass filter \(G_k^{IF}(f)\) the input signal to the second filter is given by
\begin{align}
g_k^{IF-out}(t) & = \Re\{\exp(j2\pi70MHz t)g_{70MHz}^{IF}(t) * s_{in,k}(t) + \exp(j2\pi69.980MHz t)g_{69.980MHz}^{IF}(t) * s_{in,k}^c(t) + AG^{IF}(69.990MHz)\exp(j2\pi69.990MHz t)\} , \tag{21}\end{align}

where \(g_{70MHz}^{IF}(t)\) and \(g_{69.980MHz}^{IF}(t)\) are frequency shifted versions of the impulse response of the IF band-pass filter \(g^{IF}(t)\). More precisely they are given by

\begin{align}g_{70MHz}^{IF}(t) & = g^{IF}(t) \exp(-j2\pi70MHz t) \tag{22}\end{align}

and

\begin{align}g_{69.980MHz}^{IF}(t) & = g^{IF}(t) \exp(-j2\pi69.980MHz t) , \tag{23}\end{align}

respectively. Since the center frequency of the filter is 70.000MHz the first of these two frequency translated filters will act essentially like a low-pass filter. The second filter will attenuate the \(s_{in,k}^c(t)\) signal since its pass-band will be at 20kHz while the signal \(s_{in,k}(t)\) and its conjugate \(s_{in,k}^c(t)\) have their spectrum centered a zero with most of its power within half its power-bandwidth. The second oscillator is set to

\begin{align}\text{oscillator}_1(t) & = \cos(2\pi(f_c - 70MHz)t). \tag{24}\end{align}

in order to transmit the desired signal component at the frequency \(f_c = 455.850MHz\). This selection of frequency is called low-LO. Another solution is to use the frequency \(f_c + 70\) which is called high-LO. Combining the mixer model (18) with (21) and (24) yields
The last filter in the transmitter $G_{RF}^k(f_c)$ is intended to remove all undesired terms i.e. all but the first term in (25). It is almost impossible to remove the second and third terms since they are very close in frequency (only 20kHz and 10kHz offset from the desired term respectively). We will keep them in our analysis below. These terms are small compared to the desired signal. However, the remaining terms are not small but can be removed using the filter shown in Figure 8, if we are operating the NMT450 up- or down-link band.  

Thus after the filter $G_{RF}^k(f_c)$ the output signal is given by

$$y_{RF-out}^k(t) = \Re\{\exp(j2\pi f_c t)g_{70MHz}^k(t) * s_{in,k}(t) + \exp(j2\pi(f_c-20kHz)t)g_{69.980MHz}^k(t) * s_{in,k}(t) + AG_{RF}^k(69.990MHz) \exp(j2\pi((f_c-10kHz)t) + \Re\{\exp(j2\pi(f_c-140MHz)t)g_{70.000MHz}^k(t) * s_{in,k}(t) + \exp(j2\pi(f_c-139.980MHz)MHz t)g_{69.980MHz,c}^k(t) * s_{in,k}(t) + AG_{RF,c}^k(69.990MHz) \exp(j2\pi((f_c-10kHz)t) + A\cos(2\pi(f_c-70MHz)t)). \ (25)$$

note that the filter $G_{RF}^k(f_c)$ is flat-enough that its influence can be modeled using only a scalar gain and a phase shift.

$$y_{RF-out}^k(t) = \Re\{G_{RF}^k(f_c) \exp(j2\pi f_c t)g_{70MHz}^k(t) * s_{in,k}(t) + G_{RF}^k(f_c) \exp(j2\pi(f_c-20kHz)t)g_{69.980MHz}^k(t) * s_{in,k}(t) + AG_{RF}^k(f_c)G_{RF,c}^k(69.990MHz) \exp(j2\pi((f_c-10kHz)t)\} \right), \ (26)$$

 Receivers:

After the signal $y_{RF-out}^k(t)$ has been transmitted by the transmitter antenna and received in the mth receiving antenna ($m = 1, 2$), the received signal is given by

$$x_{RF-in}^m(t) = \Re\{H_{m,1}(f_c)y_{RF-out}^1(t) + H_{m,2}(f_c)y_{RF-out}^2(t)\} \ (27)$$

where $H_{m,k}(f_c), k = 1, 2, m = 1, 2$ is the channel between transmitter antenna element $k$ and receiver antenna element $m$. This channel can certainly be considered flat i.e. it


---

4For other frequencies, e.g. GSM900 or DCS1800, S3 does not have appropriate filters today. However, it can be shown that if the receiver uses a high-LO (see below) then these terms does not affect the receiver and only the radio license is a problem.
can be modeled by a single complex constant over the frequency range of the transmitted signal. From now on, we will only consider the contribution from transmitter antenna \( k \) to receiver antenna element \( m \), while the reader should keep in mind that the actual received signal will be a superposition of the contribution from both transmitter antenna elements. A model of the receiver is shown in Figure (6). Using our mixer model (18) with the first oscillator 1 frequency set to

\[
\text{oscillator}_1(t) = \cos(2\pi(f_0 + 70\text{MHz})t),
\]  

(28)

where we have also introduced an error in the frequency, as such are exist and are not negligible see Section 1.3 and 1.4. In fact, all local oscillators will have frequency errors. A model with a frequency error only in oscillator\(_1\) will in most be sufficient. The more general case with errors in frequency of other oscillators is left to the reader as an exercise. Combining the receiver model, the oscillator mode (17), and equation (28) yields

\[
x_{m,\text{IF-in}}(t) = \Re\{H_{m,k}(f_c)G_{k}^{\text{RF,c}}(f_c)\exp(j2\pi(f_0 + 70\text{MHz})t)g_{70.000\text{MHz}}(t) * s_{n,k}(t) +
\]

\[
H_{m,k}(f_c)G_{k}^{\text{RF,c}}(f_c)\exp(j2\pi(f_0 + 70.020\text{MHz})t)g_{69.980\text{MHz}}(t) * s_{n,k}(t) +
\]

\[
AH_{m,k}(f_c)G_{k}^{\text{RF,c}}(f_c)G_{k}^{\text{IF,c}}(69.990\text{MHz})\exp j2\pi(f_0 + 70.010\text{MHz})t\} +
\]

\[
\Re\{H_{m,k}(f_c)G_{k}^{\text{RF,c}}(f_c)\exp(j2\pi(2f_c + f_0 + 70\text{MHz})t)g_{70\text{MHz}}(t) * s_{n,k}(t) +
\]

\[
H_{m,k}(f_c)G_{k}^{\text{RF,c}}(f_c)\exp(j2\pi(2f_c + f_0 + 69.980\text{MHz})t)g_{69.980\text{MHz}}(t) * s_{n,k}(t) +
\]

\[
AH_{m,k}(f_c)G_{k}^{\text{RF,c}}(f_c)G_{k}^{\text{IF,c}}(69.990\text{MHz})\exp j2\pi((2f_c + f_0 + 69.990\text{MHz})t) +
\]

\[
A\cos(2\pi(f_c + f_0 + 70\text{MHz})t).
\]  

(29)

The passband filter in the IF-stage of the receiver easily removes all but three first terms yielding

\[
x_{m,\text{IF-out}}(t) =
\]

\[
\Re\{H_{m,k}(f_c)G_{k}^{\text{RF,c}}(f_c)\exp(j2\pi(f_0 + 70\text{MHz})t)g_{70\text{MHz}}(t) * h_{70\text{MHz}+f_0}^{\text{IF,c}}(t) * s_{n,k}(t) +
\]

\[
H_{m,k}(f_c)G_{k}^{\text{RF,c}}(f_c)\exp(j2\pi(f_0 + 70.020\text{MHz})t)g_{69.980\text{MHz}}(t) * h_{70.020\text{MHz}+f_0}^{\text{IF,c}}(t) * s_{n,k}(t) +
\]

\[
AH_{m,k}(f_c)G_{k}^{\text{RF,c}}(f_c)G_{k}^{\text{IF,c}}(69.990\text{MHz})H_{70.010\text{MHz}}^{\text{IF,c}}\exp(j2\pi(f_0 + 70.010\text{MHz})t)\}. \]

(30)

In the last mixer, the signal is multiplied by the oscillator signal

\[
\text{oscillator}_2(t) = \cos(2\pi(69.990\text{MHz})t).
\]  

(31)

Which, combined with the mixer model (18), yields
The base-band filter easily removes all but the three first terms leaving

\[ x_{m}^{\text{BB-in}}(t) = \Re \{ H_{m,k}^{c}(f_c)G_{k}^{RF,c}(f_c) \exp(j2\pi(f_o+10\text{kHz})t)g_{70\text{MHz},k}^{\text{IF,c}}(t)\ast h_{70\text{MHz},f_o,m}^{\text{IF,c}}(t)\ast s_{m,k}^{c}(t) + H_{m,k}^{c}(f_c)G_{k}^{RF,c}(f_c) \exp(j2\pi(f_o+30\text{kHz})t)g_{69.980\text{MHz},k}^{\text{IF,c}}(t)\ast h_{69.980\text{MHz},f_o,m}^{\text{IF,c}}(t)\ast s_{m,k}^{c}(t) + A H_{m,k}^{c}(f_c)G_{k}^{RF,c}(f_c)G_{k}^{IF,c}(69.990\text{MHz})H_{70.010\text{MHz}}^{IF}(j2\pi(f_o+20\text{kHz})t) + \Re \{ H_{m,k}^{c}(f_c)G_{k}^{RF,c}(f_c) \exp(j2\pi(f_o+139.90\text{MHz})t)g_{70\text{MHz},k}^{\text{IF,c}}(t)\ast h_{70\text{MHz},f_o,m}^{\text{IF,c}}(t)\ast s_{m,k}^{c}(t) + H_{m,k}^{c}(f_c)G_{k}^{RF,c}(f_c) \exp(j2\pi(f_o+140.01\text{MHz})t)g_{69.980\text{MHz},k}^{\text{IF,c}}(t)\ast h_{69.980\text{MHz},f_o,m}^{\text{IF,c}}(t)\ast s_{m,k}^{c}(t) + A H_{m,k}^{c}(f_c)G_{k}^{RF,c}(f_c)G_{k}^{IF,c}(69.990\text{MHz})H_{70.010\text{MHz}}^{IF}(j2\pi(f_o+140\text{MHz})t) + \right. \]

\]

The base-band filter easily removes all but the three first terms leaving

\[ x_{m}^{\text{BB-out}}(t) = \Re \{ \]

\[ H_{m,k}^{c}(f_c)G_{k}^{RF,c}(f_c) \exp(j2\pi(f_o+10\text{kHz})t)g_{70\text{MHz},k}^{\text{IF,c}}(t)\ast h_{10\text{kHz},f_o,m}^{\text{IF,c}}(t)\ast s_{m,k}^{c}(t) + H_{m,k}^{c}(f_c)G_{k}^{RF,c}(f_c) \exp(j2\pi(f_o+30\text{kHz})t)g_{69.980\text{MHz},k}^{\text{IF,c}}(t)\ast h_{30\text{kHz},f_o,m}^{\text{IF,c}}(t)\ast s_{m,k}^{c}(t) + A H_{m,k}^{c}(f_c)G_{k}^{RF,c}(f_c)G_{k}^{IF,c}(69.990\text{MHz})H_{70.010\text{MHz}}^{IF}(j2\pi(f_o+20\text{kHz})t) \].

This signal is sampled by the DSP. Inside the DSP the following complex-valued signal is created

\[ x_{m}(t) = (\cos(2\pi10\text{kHz}t)x_{m}^{\text{BB-out}}(t) + j\sin(2\pi10\text{kHz}t)x_{m}^{\text{BB-out}}(t)) \ast h_{LP}^{\text{IF}}(t), \] (34)

where \( h_{LP}^{\text{IF}}(t) \) is a digital low-pass filter with a bandwidth comparable to \( s_{m,k}(t) \). Combining (30) and (31) yields

\[ x_{m}(t) = H_{m,k}(f_c)G_{k}^{RF}(f_c) \exp(-j2\pi f_o t) \times \]

\[ g_{70\text{MHz},k}^{\text{IF,c}}(t) \ast h_{70\text{MHz},f_o,m}^{\text{IF,c}}(t) \ast h_{LP}^{\text{IF}}(t) \ast s_{m,k}^{c}(t) \times H_{m,k}(f_c)G_{k}^{RF}(f_c) \exp(j2\pi(f_o+20\text{kHz})t) \times \]

\[ g_{69.980\text{MHz},k}^{\text{IF,c}}(t) \ast h_{69.980\text{MHz},f_o,m}^{\text{IF,c}}(t) \ast h_{LP}^{\text{IF}}(t) \ast s_{m,k}^{c}(t) \times H_{m,k}(f_c)G_{k}^{RF}(f_c) \exp(j2\pi(f_o+40\text{kHz})t) \times \]

\[ g_{69.980\text{MHz},k}^{\text{IF,c}}(t) \ast h_{69.980\text{MHz},f_o,m}^{\text{IF,c}}(t) \ast h_{LP}^{\text{IF}}(t) \ast s_{m,k}^{c}(t) \times \]

\[ H_{m,k}(f_c)G_{k}^{RF}(f_c)G_{k}^{IF}(69.990\text{MHz}) \ast H_{70.010\text{MHz}}^{IF,c}(20\text{kHz}) \exp(j2\pi(f_o-10\text{kHz})t) + A H_{m,k}(f_c)G_{k}^{RF,c}(f_c)G_{k}^{IF,c}(69.990\text{MHz}) \ast \]

\[ H_{70.010\text{MHz}}^{IF,c}(20\text{kHz}) \exp(j2\pi(f_o+30\text{kHz})t) \] (35)
Interpreting the result

The first four terms in (35) are filtered, frequency shifted, and scaled versions of the input signal \( s_{\text{in},k}(t) \). The transmitted signal can in principle be estimated from all of them, however the first one will typically be much stronger than the other three. This is because they have passed various filters outside of their pass-band. If the bandwidth of \( s_{\text{in},k}(t) \) is less than say 6kHz, then these terms are probably negligible for most purposes. If the bandwidth is higher than 20kHz their will certainly be interference from the second, third and fourth term when trying to demodulate based on the first. The two last terms are cisoids that could be eliminated with notch-filters or similar techniques, if they cause degradation. To emphasize on our interpretation of (35) we may re-write it as

\[
x_m(t) = \exp(-j2\pi f_o t)h_{m,1}(f_o, t) * s_{\text{in},1}(t) + \exp(-j2\pi f_o t)h_{m,2}(t) * s_{\text{in},2}(t) + e_m(t) + n_m(t),
\]

where \( h_{m,k}(t) \), \( s_{\text{in},k} \), \( e_m(t) \) and \( n_k(t) \) are the channel impulse responses, the transmitted signals, the errors and the thermal noise. Here we have made explicit that the signal receiver will be a superposition of the two transmitted signals and thermal noise is present (as always). From (35) we have that the channel is a function of the frequency offset \( f_o \) namely

\[
h_{m,1}(f_o, t) = H_{m,k}(f_c)G_{RF}^c(f_c)G_{RF}^{\text{IF}}(f_{\text{RF},0.000MHz}, k)(t) * h_{70.000MHz+f_o,m}(t) * h_{10kHz+f_o,m}(t) * h_{\text{LP}}(f_o, t).
\]

The error term, \( e_m(t) \), consists of all but the first term in (35) where we again note that this term will be negligible if the bandwidth of \( s_{\text{in},k}(t) \) is small, and increases when the bandwidth of \( s_{\text{in},k}(t) \) increases. The channel impulse responses \( h_{m,k}(f_o, t) \) changes rapidly with time due to changes in the RF propagation channel which enters the equation in the complex factor \( H_{m,k}(f_c) \). The remaining terms changes slowly due frequency and temperature drifts.

2 Software

2.1 Software for Marconi 2024

Some functions for controlling two Marconi 2024 from the two COM ports of PC has been developed. These functions are given by the two files "marconi.cpp" and "marconi.h". To embed these functions in a C++ program, the "marconi.h" program should be included using #include in the main program, and marconi.cpp added to the project using Microsoft Visual C++. By looking at the code in marconi.c and the documentation in the manual of the Marconi signal generator, it is relatively simple to use more features of the Marconi signal generator than those already implemented.

2.2 Software for the multi-channel daughter board

TBD.

\footnote{In term number two and three \( s_{\text{in},k}(t) \) is also conjugated but that is a minor issue.}
2.3 Chan and Lek’s receive.m

The "receive.m" matlab program can be downloaded from www.s3.kth.se/signal/edu/projekt/DSPsupport/receive.shtml. Using this program it is possible to use the DSP as data acquisition or sound receiving card. When this program is called from matlab it in turn calls a PC program receive_host.exe which downloads a program receive_DSP.out to the DSP. This program is extremely useful in the early stages of DSP code development. However, the implementation of the DSP code is implemented such that it takes one sample at a time (rather than taking a whole buffer) and interrupts the DSP for each sample. Therefore the code is usually not a good starting point for code development.

2.4 Chan and Lek’s transmit.m

The "transmit.m" matlab program can be downloaded from www.s3.kth.se/signal/edu/projekt/DSPsupport/transmit.shtml. Using this program it is possible to use the DSP as function generator or sound creating card. When this program is called from matlab it in turn calls a PC program transmit_host.exe which downloads a program transmit_DSP.out to the DSP. This program is extremely useful in the early stages of DSP code development. However, the implementation of the DSP code is implemented such that it takes one sample at a time (rather than taking a whole buffer) and interrupts the DSP for each sample. Therefore the code is usually not a good starting point for code development.

2.5 Chan and Lek’s DMA_sendrec.m

The ”DMA_sendrec.m” program can be downloaded from www.s3.kth.se/signal/edu/projekt/DSPsupport/dma_sendrec.shtml. This program performs the tasks of the two previous i.e. ”receive.m” and ”transmit.m”, simultaneously. This is very useful when in the early stages of DSP code development for systems where the input and output are closely related - or when examining if such a relation exists (cross-talk). The implementation of the DSP code is usually a good starting point for implementation since it implements transfers of A/D and D/A in blocks of data (buffers) and interrupts the DSP only when the block transfer is finished.

2.6 How to write linear assembler programs for TMS320C6000

When profiling DSP code, one may find that one function dominates in terms of the number of cycles needed to execute it. It may even prevent the application to be successfully implemented, as the DSP just doesn’t execute the code fast enough, even after trying to optimize the C-code. One option is to find code for instance on Texas instruments ftp site. The disadvantage with this approach is that this code might not take advantage of all the particular properties of the problem at hand. For instance, if two matrices are to be multiplied where one alway have zeros in a number of positions, then these multiplications never need to be done.
Another option for optimization is to use linear assembler. The authors experience is that it takes about a week to implement a function like filtering in linear assembler. Start by reading chapter four of [3]. Then look at the comments below.

### 2.6.1 Basics

Open a new file with extension “.sa” and load it into the project, say the program is called mylincode.sa. Assume we implement a function mylinroutine in the file mylincode.asm. In the C-program add the instruction

```c
extern void mylinroutine(void);
```

Thus in this case we pass no parameters directly to or from the function. The in mylincode.sa there must be the instructions

```assembly
.def _mylineroutine ; On line one of the code.
_mylinroutine: .cproc; After .global instructions
.return ; last but one instruction in file
.endproc; last instruction in file
```

which states that we implement a C callable routine “mylinroutine”. Indeed in the C-program we may call the function using

```c
mylineroutine();
```

As second line of the file mylincode.sa the following instruction can be used to define in which area the code should be allocated

```assembly
.sect "text",
```

To access the global c-variable "var1" from the linear assembler routine the following line may be inserted between .sect directives and the .cproc directive,

```assembly
.global _var1;
```

Then if we want to load the value of _var (assumed to be a 32bit data type) into a register r, we may insert the following code line

```assembly
LDW **DP(_var1),r
```

Here we assume that a “near data model has been selected” and that the data page pointer (DP) has been defined as

```assembly
.asg B14, DPx
```

just after the .cproc directive. If a “far data model” is used then the author believes that the line should read

```assembly
LDW **DP(_var1),r
```
MVK _var1,r2
LDW *r2, r

without any need for a “.asg” assignment. If you find this out, please let me know the true answer.
All variables you can play with are the 32 core 32-bit registers (A0...A15,B0...B15). You can assign any number of variable names you want to use as variable names as.

.reg length,width,height,weight,area

The compiler will do the mapping of the names to the physical registers (A0...A15,B0...B15). The .reg instructions can be put just below the .asg instructions. Now you are ready to start to play with instructions like

MPYSP length,width,area

which multiplies the values of the floating points numbers in registers length and width and puts the results in area. All available instructions can be found in [2]. A well documented (?) sample program is given below,

2.6.2 Efficiency
The key of efficiency is to execute many instructions simultaneously, i.e. on the same cycle. Thus one may think of how all (or as many as possible) of the available units can be used. Two load or store instructions can be executed in the same cycle. However, this requires that there are no memory conflicts. This is the case if the two 32-bit words are adjacent. So if p1 and p2 are pointers offset one word, the the two instructions “LDW *p1++[2],register1” and “LDW *p2++[2],register2” can be performed at the same time. Thus it is a good practice to have two pointers to the same vector offset four bytes, and load values into registers as indicated. For complex-valued data this also results in quite readable code. This compiler is instructed on this feature by using the instructions

.mptr p1,b,8
.mptr p2,b+4,8,

which tells the processor that the pointers p1 and p2 are incremented or decremented with a multiple of eight and that they are offset four bytes with respect to each other.

2.6.3 Debugging
You can debug linear assembler code by compiling it in debug mode (just as you do with C code), and inserting break points. Then you can watch the value of variables using the watch window.
When debugging linear assembler programs in CCS2, floating point values are a bit tricky to debug using watch window, as they will presented as the decimal values their bit-pattern correspond to (there is drag-down menu where float can be selected but it doesn’t work).
The solution is to take the integer number and apply it to the matlab function Int2float.m which returns the correct floating point number.

Sometimes watch-window does not seem to update the presented register values after a multiply, however the results are correct anyway. An idea may be to look through the core registers of the DSP and see if any of these shows the correct value. Another tricky thing is that some lines of the linear assembler code are never executed even in "debug" mode. If a breakpoint is set on one of these lines, CCS will move the breakpoint to the next line that exist. This may happen if CCS decides that one code-line unnecessary, for instance a register is written to and then never used. A solution can be to add some “dummy” line where the register is used.

### 3 Principles of high- and low- local oscillators

A mixer basically multiplies it’s two ingoing signals. In practice there are more terms in the outgoing signal than this (see next section), however for the purposes of the section they may be ignored. Assume that the frequency of the ingoing modulated signal is $f_1$ and that wanted signal is $f_2$. Based on the relationship

$$\cos(\alpha) \cos(\beta) = \frac{1}{2} \cos(\alpha + \beta) + \frac{1}{2} \cos(\alpha - \beta),$$

we conclude that there are two possible choices for the frequency of the local-oscillator namely $f_2 - f_1$ or $f_2 + f_1$, the first choice is known as low-LO while the latter is called high-LO. For the two cases we get

$$\cos(2\pi f_1 t) \cos(2\pi (f_2 - f_1) t) = \cos(2\pi f_2 t) + \cos(2\pi (2f_1 - f_2) t)$$

and

$$\cos(2\pi f_1 t) \cos(2\pi (f_2 + f_1) t) = \cos(2\pi f_2 t) + \cos(2\pi (f_2 + f_1) t),$$

respectively. Since only the first term is desired the second term has to be removed by filtering. The difficulty of removing this term depends on its frequency, thus the choice between high or low-LO is often based on the frequency of the unwanted term. Another issue to consider in receivers is there may be other signals impinging on the receiver other than the desired. Assuming for instance that there is an interfering signal of frequency $f_3$. If $f_3 = f_2 - (f_1 - f_2)$ then this signal will be downconverted to the frequency $f_2$ in a low-LO setting. The same happens with a signal of frequency $f_3 = f_1 + (f_1 - f_2)$ in the low high-LO case. These frequencies are known as “image” frequencies. In order to protect the receiver from these signals an image reject filter is required. Again, the ease of implementing such a filter may determine the choice between high and low-LO.

As will be shown in the next section there are usually many “unideal” effects of the mixer, this is described in the next section.
4 Diode-Ring Mixers

In Figure 14 a model of so-called diode-ring mixers are shown. The mixers in the RX and TX modules (see Section 1.2 and 1.1) are of this type. We will analyze this circuit for a receiving application i.e. the local oscillator (L) and radio (R) terminals are inputs, and the I terminal is an output. Assuming ideal transformers, and that the local oscillator is much stronger than the radio input, the model of Figure 15 and 16, results when the local oscillator is positive and negative, respectively. In these models, the diodes are modelled as resistors. For positive local oscillator the model yields

\[ u_I(t) = Ce_L(t) + e_R(t) \]  \tag{41} 

where

\[ C = \frac{r_2 - r_1}{r_1 + r_2} \]  \tag{42} 

and for negative

\[ u_I(t) = De_L(t) - e_R(t) \]  \tag{43} 

where

\[ D = \frac{r_4 - r_3}{r_1 + r_2} \]  \tag{44} 

These equations can be combined into one, namely

\[ u_I(t) = Ae_L(t) + P(t)e_R(t) + P(t)Be(t), \]  \tag{45} 

where \( P(t) \) is given by

\[ P(t) = \text{sign}\{u_L(t)\}, \]  \tag{46} 

while \( A \) and \( B \) are defined by

\[ A = \frac{C + D}{2} \]  \tag{47} 

and

\[ B = \frac{C - D}{2}, \]  \tag{48} 

respectively. In almost all cases, \( e_L(t) \), is a sinusoid e.g.

\[ e_L(t) = \sin(2\pi f_L t), \]  \tag{49} 

and \( P(t) \) will obviously become a square-wave, as such it can be expanded into a Fourier series.
\[ P(t) = \frac{4}{\pi} \sum_{n=0}^{\infty} \frac{\sin(2\pi(2n+1)fLt)}{2n+1}. \]  

Equation (50)

Obviously, only the first term in (50) is desired. The other terms in (50) and also the unwanted terms in (45) will lead to undesired so-called “spurious” signals which has to be removed by filtering. A more thorough analysis of diode-ring mixers which takes into account some additional factors may be found in [4].

Figure 14: Schematic of diod-ring mixer

Figure 15: Model of diod-ring mixer when local oscillator is positive, i.e. \( e_L(t) > 0 \).

Figure 16: Model of diod-ring mixer when local oscillator is negative, i.e. \( e_L(t) < 0 \).

References

