Pulsed RADAR signal generation and measurements
Educational Note

Products:

- R&S®SMW
- R&S®SMBV
- R&S®FSW
- R&S®FSV
- R&S®RTM
- R&S®BBA150
- R&S®HF907

Target reader group are engineering students who want to perform tests using pulsed or chirped signals.

Current Radar development is focusing the area of signal processing. This is taken into account by this educational note, where the R&S® SMW / SMBV instruments on the transmitter side and R&S®FSW / FSV instruments on the receiver side are combined to a closed loop Radar system, performing radar detection by means of pulse compression and digital signal processing. Appropriate R&S sofware tools for such applications are described as well as the interface between the tools and the test instruments.

Pulsed radar signals ─ 1MA234_0e

Dieter Bues - PulsedRadar_0e

Educational Note

Contents

1 Introduction.. 3

2 Radar application systems...8

3 R&S software solution for radar signal generation 17

4 Signal power discussion...19

5 Appendix..25

6 Rohde & Schwarz...36

1 Introduction

1.1 Motivation

Today's radar systems consist of RF electronics, signal processors and appropriate signal processing software. While the development of RF electronics is mainly focused on increased measurement bandwidth and higher data aquisition speed, the main challenge of the signal processing unit is to find improved algorithms being able to take more and more tasks from the RF electronics.

This is taken into account by the educational note at hand, where the appropriate test instruments are introduced along with internal and external signal processing capabilities. MATLAB scripts are described how to perform signal processing as well as how they can be used as interface between external tools and the test instruments.

Additionally this educational note describes how to setup a complete Radar system by means of appropriate off-the-shelf test instruments available from Rohde & Schwarz along with MATLAB programs demonstrating the signal processing capabilities of modern Radar systems. All experiments described in this document can be easily reproduced and easily adapted to specific requirements by means of the MATLAB source code provided as attachment. Because the shown Radar experiments are independent from each other, only parts of the application can be extracted and used for special purpose.

Target reader group of this Educational Note are engineering students who want to perform tests using pulsed or chirped signals.

1.2 Product abbreviations

Following product abbreviations are used in this Educational Note:

R&S®SMW200A Vector Signal Generator: SMW

R&S®SMBV100A Vector Signal Generator: SMBV

R&S®FSW Signal and Spectrum Analyzer: FSW

R&S®FSV Signal and Spectrum Analyzer: FSV

R&S®RTM Digital Oscilloscope: RTM

R&S®BBA150 Broadband Amplifier: BBA150

R&S®HF907 Double-Ridged Waveguide Horn Antenna: HF907

1.3 Demonstrating pulse compression

Pulse compression is a mathematical method to reduce the effective pulse width at constant energy of the transmitted signal at the receiver. Short pulses are needed for good range resolution, i.e. the capability to separate (resolve) multiple targets moving in a close distance with same radial velocity to each other. Therefore pulse compression is an important method to improve the performance of radar systems.

Figure 1-1 shows a typical example.

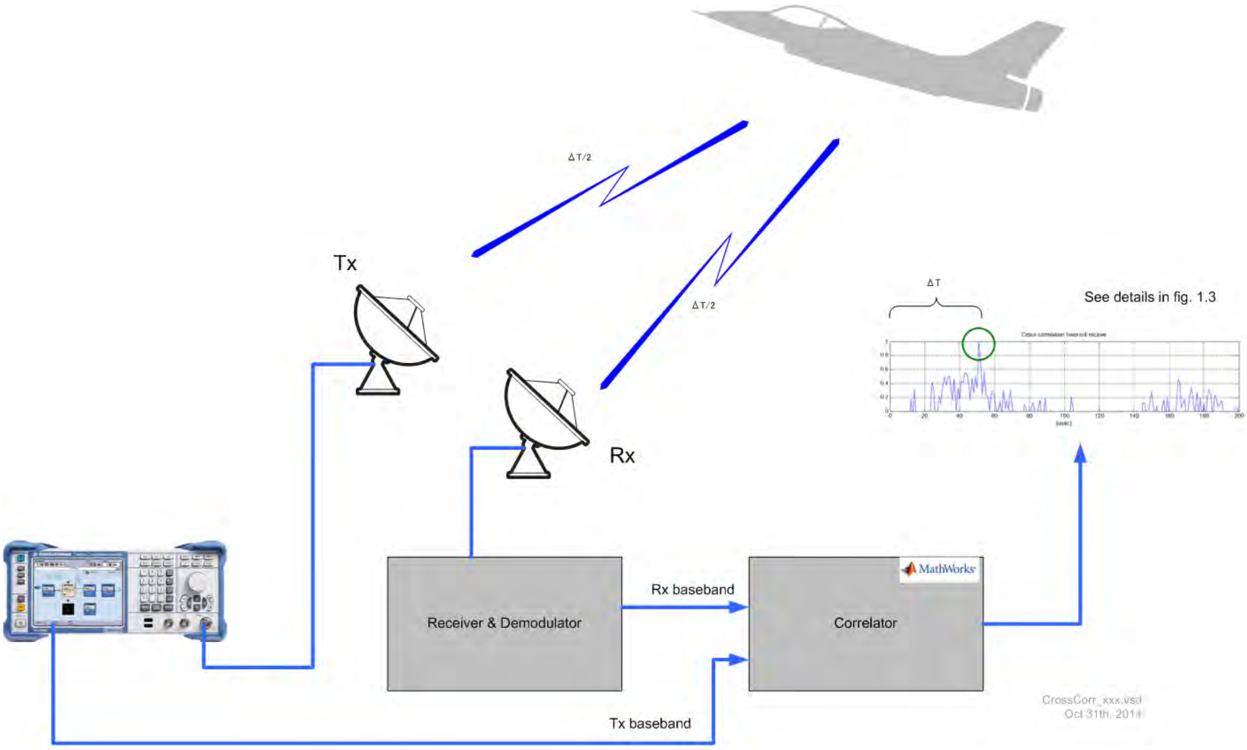


Fig. 1-1: Example of pulse compression

A generator provides a signal to the transmit antenna which radiates an appropriate wave to a flying object. The reflected wave Rx, which is delayed and attenuated, sometimes down close to the noise floor of the system, is received, demodulated and then lead to the "correlator", which compares the transmitted and the received baseband signal. Because we are talking about a coherent system, the transmitter and the receiver are supplied by a common reference frequency, not shown in Fig. 1-1. The correlator is a mathematical algorithm, which is able to detect the time position of the known Tx baseband signal within the noisy Rx baseband signal. The output of the correlator thus delivers a single pulse providing the distance information of the plane. The correlator is normally implemented in the digital signal processing (DSP) unit of the receiver. In our case it is implemented in a small piece of MATLAB code, as shown in appendix 5.1.

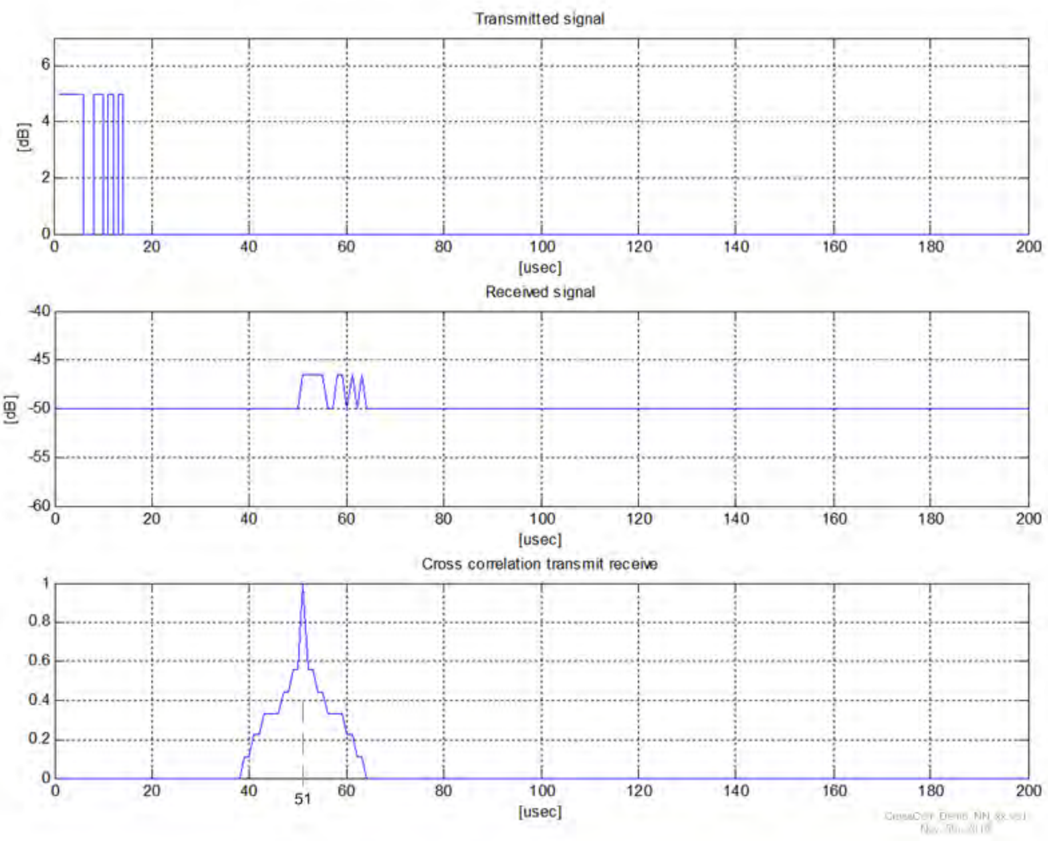


Fig. 1-2: Tx / Rx baseband cross correlation using Barker code, no noise

Fig. 1-2 shows the ideal situation without noise. The top diagram shows the 13 bit Barker code, which is used for BPSK-modulation in the transmitter, for example signal generator SMBV as shown in Fig. 1-1. The diagram in the center of Fig. 1-2 shows the received and BPSK-demodulated signal. The lower diagram finally shows the output from the correlator, showing a peak at 51 usecs, which equates the simulated signal delay.

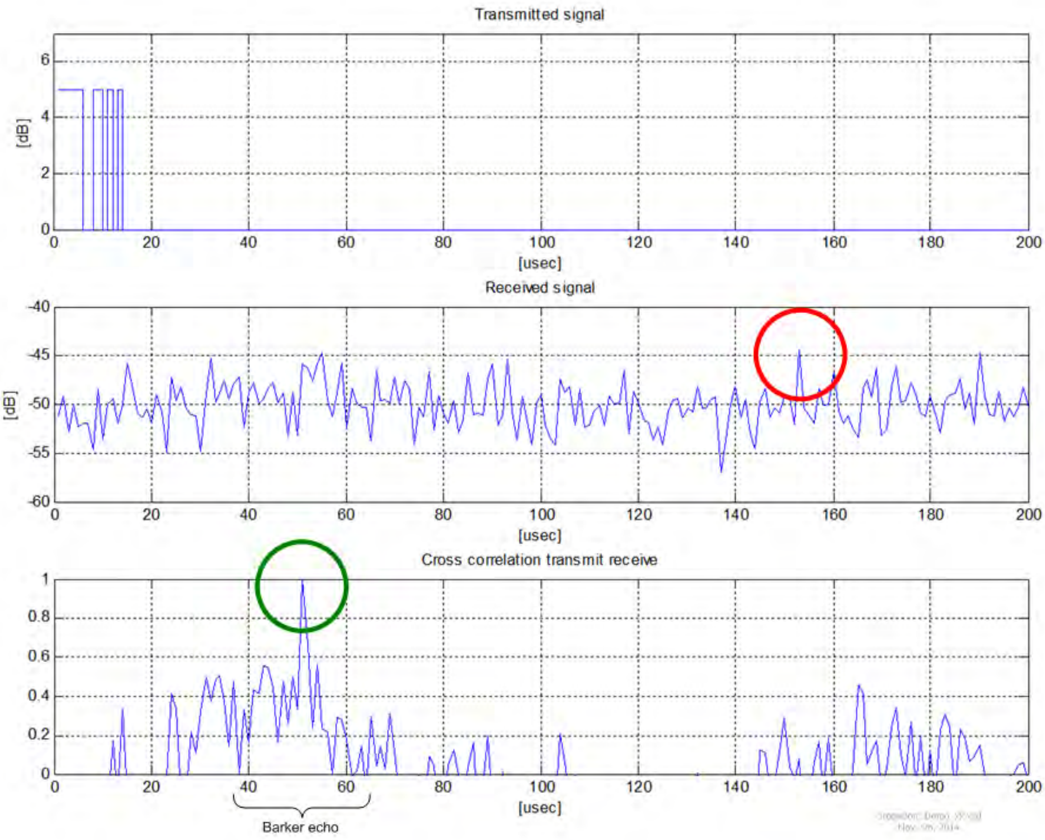


Fig. 1-3: Tx / Rx baseband cross correlation using Barker code, with noise

Fig. 1-3 shows the real situation, which includes noise in the received signal. Even though there is a high peak in the received signal at roughly 155 usec, marked by the red circle, the true signal is detected clearly at 51 usec as indicated by the green circle. This example shows that by means of pulse compression the signal to noise ratio in a transmit receive chain can be increased dramatically. Higher transmission power or more efforts in receiver electronics would be required to obtain similar results by hardware in order to increase the sensitivity. Hence the example shows that the sensitivity of a system can be increased by means of digital signal processing, which would need more efforts and causes much higher costs when being implemented in hardware. The example also demonstrates the term "Pulse Compression". Even though the Barker signal is reveived in full length it appears as a small peak because of the correlation as indicated by the green circle in Fig. 1-2. Thus avoiding the drawbacks of long pulses and using the benefits of short pulses at the same time. Barker pulses have a special bit sequence in order to keep sidelobes minimal when doing the correlation. In other words they have an autocorrelation function with a strong center and very low side lobes, as shown in Fig. 1-2. Additional information is available in [17].

The complete MATLAB simulation program of this example is included in appendix 5.1. The signal shapes will vary over each run because of the random noise generator, but the proper pulse delay detection can be veryfied for each new run of the software. Additional experiments like the variation of Barker code lengths, increasing/decreasing noise or using codes different from Barker codes are possible using the provided MATLAB program.

Even though the SMBV can basically provide all signals and modulations as discussed in this paragraph, its maximum output power of 19 dBm is normally not enough to detect planes in a long range. Detailed power calulations are outlined in chapter 4, where the receiving capabilites are also taken into account. A real system based upon this simulation would need the reference signal to be connected between transmitter and receiver. However, phase coherent coupling between transmitter and receiver is not necessarily needed. In the next chapter we will first introduce a radar system consisting of R&S test instruments only, including transmitter and receiver.

2 Radar application systems

2.1 Barker sequence TxRx system

Figure 2-1 shows a typical radar transmit receive system. The appropriate R&S products are shown in the lower part. The software is indicated in the grey boxes in the upper part. Starting with the baseband signal, for instance a Barker Sequence or a FM chirp, Fig. 2-1 upper left, the IQ-modulation signal will be calculated and then transferred to the radar transmitter, shown in the lower left.

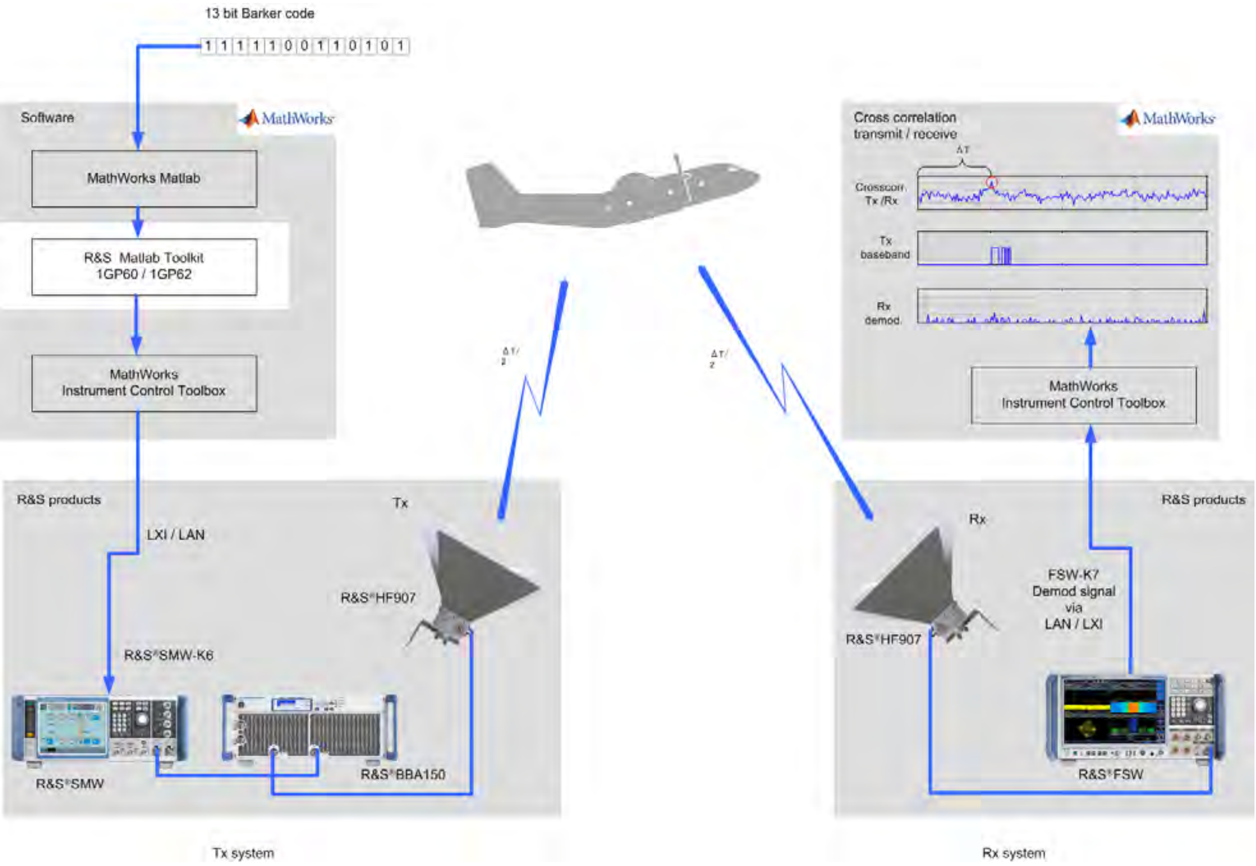


Fig. 2-1: Generation and evaluation of Barker code modulated signals

Because we are talking about a coherent monostatic system, receiver and transmitter are operated by a common reference frequency, which is not shown in Fig. 2-1 and Fig. 2-2 for the sake of simplicity. However, phase locking is not necessarily needed.

The radio signal can be transmitted / received via the air interface or can be directly applied to a hardware device being tested, as detailed for example in [4]. As shown in the introduction, pulse compression is applied by means of cross correlation between the transmitted and received/demodulated baseband signal. Due to the properties of Barker sequences there is basically a very sharp correlation function with low side lobes which is only disturbed by noise coming from the transmit / receive path.

So far this example does not consider whether the received power is high enough to trigger the receiving unit, the FSW in our example. The power issues are detailed in chapter 4 "signal power discussion" and need to be taken into account when designing such systems.

2.2 Closed loop radar application

While in the previous chapter a constant baseband signal was used for transmission, this chapter shows a different situation where the transmitted baseband signal is derived from a received signal, after an initial signal has been sent first. Retransmitting modified signals is a common practice in Electronic Counter Measure (ECM) applications in order to confuse hostile airborne devices. Figure 2-2 shows a simple example where only two bits of the received bit sequence are modified. However, because of MATLAB being behind the scenery the algorithm's complexity is rather unlimited. Complex cyphering/decyphering algorithmns can be developed, tested and can be burned finally into a FPGA using the appropriate Mathworks toolboxes.

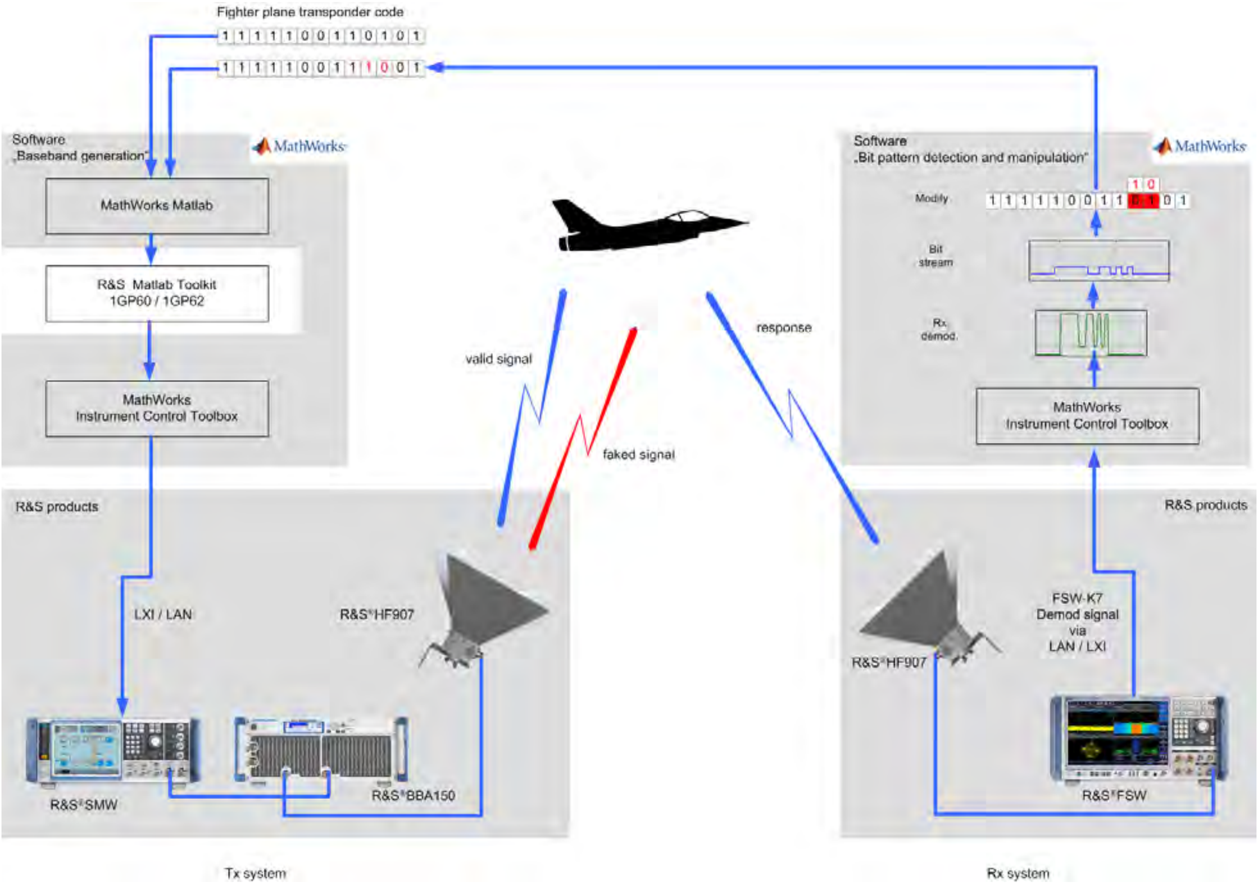


Fig. 2-2: Closed loop radar system

In this example an arbitray bit sequence is used for BPSK modulation. Fig. 2-2 shows the evaluation proceedings as performed with MATLAB, Fig. 2-3 shows the results. The program is explained in appendix 5.1.3. and is available along with this educational note [16]. In order to ensure proper function of the software it is required to operate the signal generator and the spectrum analyzer from a single reference. Additionally the spectrum analyzer is triggered by a marker from the signal generator. For the sake of clarity the reference cable and the trigger cable are not shown in Fig. 2-2.

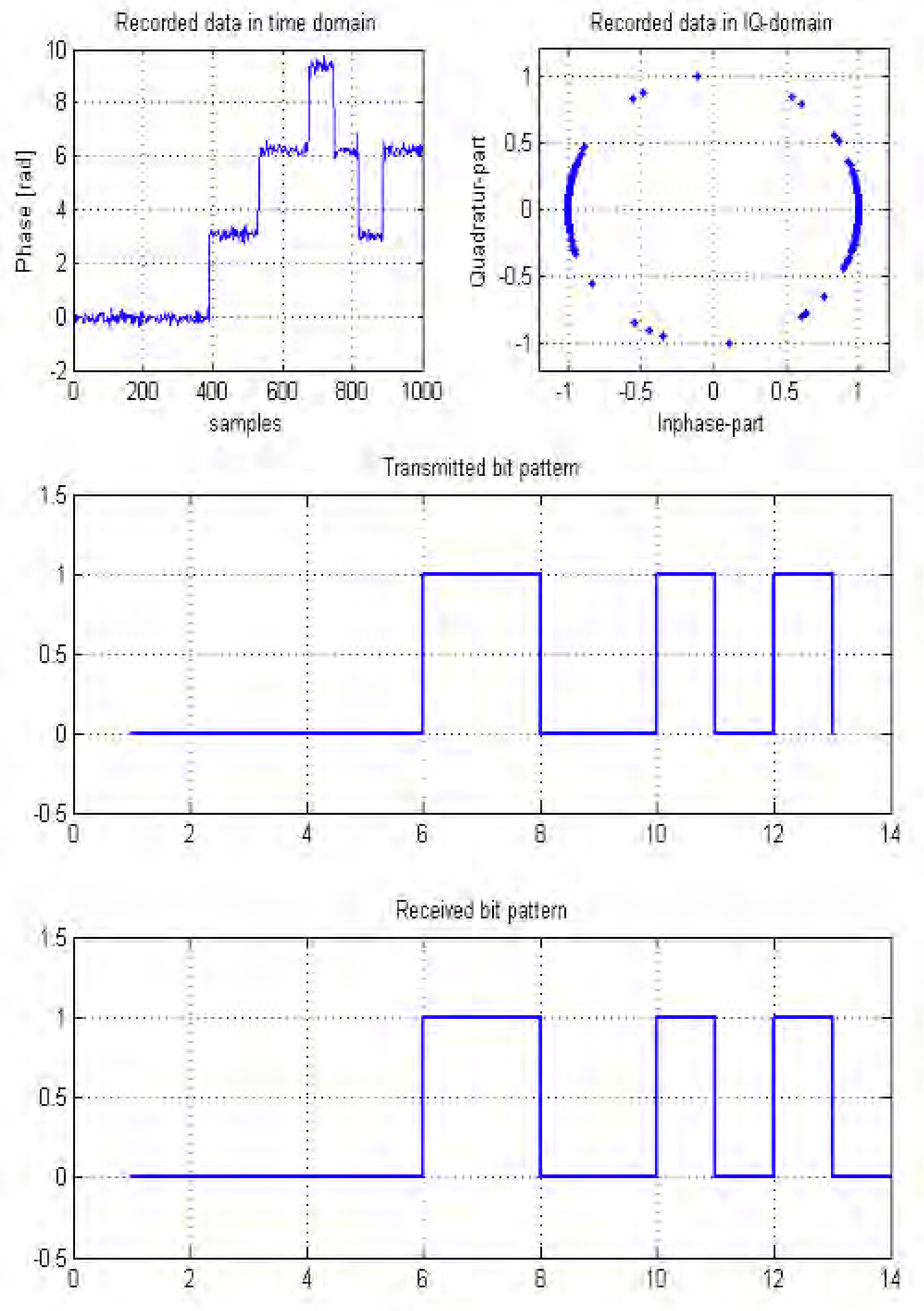
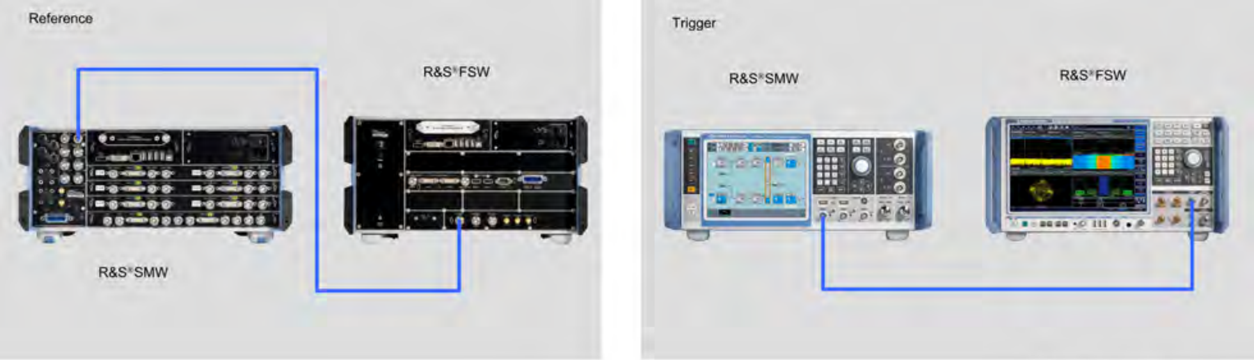


Fig. 2-3: Regaining bit information from the demodulated signal

The diagram in the upper left of Fig. 2-3 shows the demodulated signal as captured from the signal analyzer FSW. An FSV can also be used. By means of MATLAB commands, refer to appendix 5.1.3, the I/Q- constellation diagram as shown in the upper right can be created. By means of the I/Q constellation vector the bit sequence can be easily determined. The lower two diagrams in Fig. 2-3 are showing the original and the received bit sequence. The received bit sequence can be used to calculate a jamming signal, which can be used to confuse hostile airborne devices when being sent back where the incoming signal was received from. The determination of the bitsequence via the I/Q constellation diagram -/vector can be extended to other digital modulations like QPSK by just slightly extending the MATLAB code as listed in the Appendix.

The evaluation of the received signal requires a common reference signal as shown in Fig. 2-4 on the left. The SMW provides a trigger signal to FSW. The required cabling is shown in Fig. 2-4 on the right.



TxRx_ecm_cablingxxx.vsd
Aug. 27th, 2014

Fig. 2-4: Cable links for reference and trigger

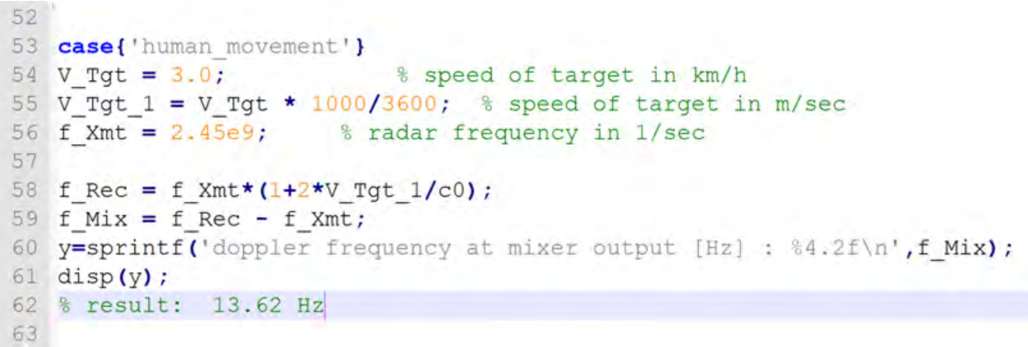
Beyond the receiver/transmitter feedback as indicated in Fig. 2-2, this example shows also how to handle customer specific pulse signals. Any arbitrary digital sequence, respecitvely any baseband signal can be fed into the transmission system using this configuration. The sample MATLAB code to feed the digitial signal into the baseband generator is provided in appendix 5.1.3.

Please take into account also the power requirements as discussed in chapter 4 "signal power discussion" when designing such systems. The power scenario "Military-Plane" comes closest to Fig. 2-2

The jamming example in this text assumes a single RF frequency. However modern radar systems use frequency hopping to minimize interference caused by other systems or hostile signals. Reference [14] provides detailed information on how to evaluate frequency hopping signals.

2.3 Detecting and simulating moving objects

State-of-the-art radar systems not only detect the distance of an object but also its radial velocity. Both, radial velocity and range, are essential especially in the car industry for instance for pedestrian detection. While the range information in radar technology is derived from the time delay between the transmitted and the received signal, the radial velocity is obtained from the frequency shift of the received compared to the transmitted signal, the so called "Doppler Shift". Figure 2-5 shows a MATLAB calculation of the Doppler shift of a pedestrian moving with a radial velocity of 3 km/h. When being detected with a transmitted radar signal of f_Xmt= 2.45 GHz (S-band) the frequency deviation of the received signal is 13.62 Hz depending on the radial direction the pedestrian is moving. If the pedestrian is moving towards the antennas the frequency is increased by 13.62 Hz, if the pedestrian is turning, the Doppler frequency is negative. In radar technology this effect is used to detect the radial velocity of objects.



```

case{'human_movement'}
V_Tgt = 3.0;            % speed of target in km/h
V_Tgt_1 = V_Tgt * 1000/3600;  % speed of target in m/sec
f_Xmt = 2.45e9;     % radar frequency in 1/sec

f_Rec = f_Xmt*(1+2*V_Tgt_1/c0);
f_Mix = f_Rec - f_Xmt;
y=sprintf('doppler frequency at mixer output [Hz] : %4.2f\n',f_Mix);
disp(y);
% result:  13.62 Hz

```

Fig. 2-5: Doppler shift calculation of moving objects

Figure 2-6 shows a test setup to determine the Doppler shift. The output signal of SMBV is divided into one path feeding the transmit antenna and a second path going to the local oscillator input of a mixer. The received signal is routed to the RF input of the mixer. The mixer output signal is displayed on an oscilloscope showing directly the Doppler signal according to the movement of the pedestrian as calulated similar to Fig. 2-5.

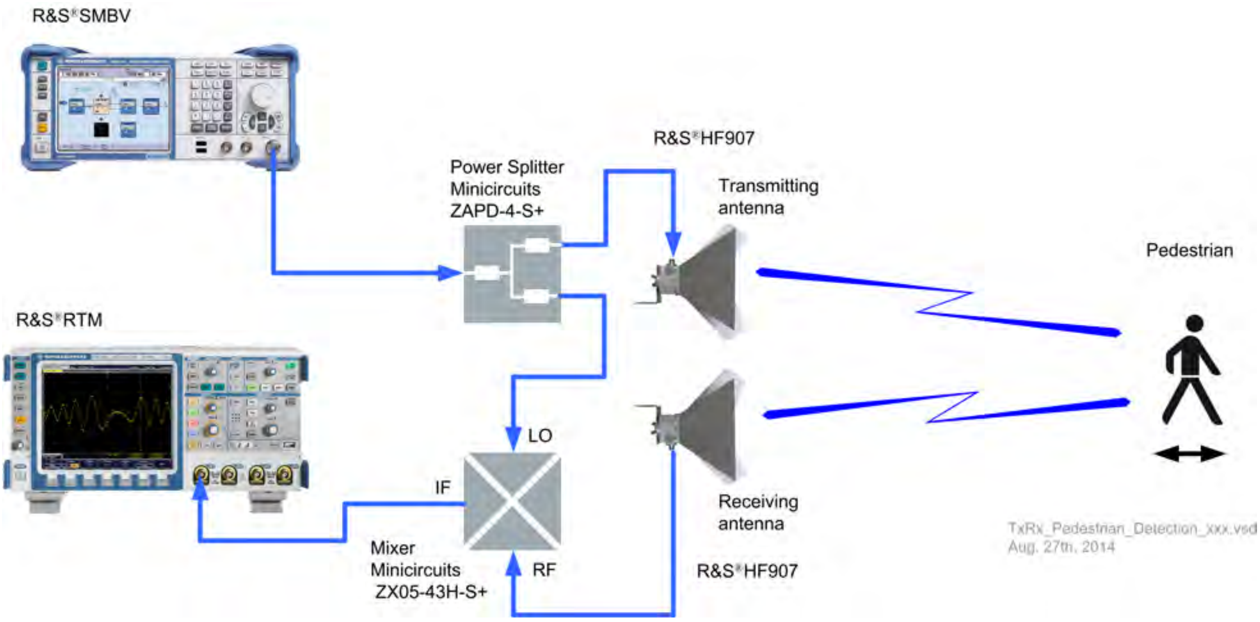
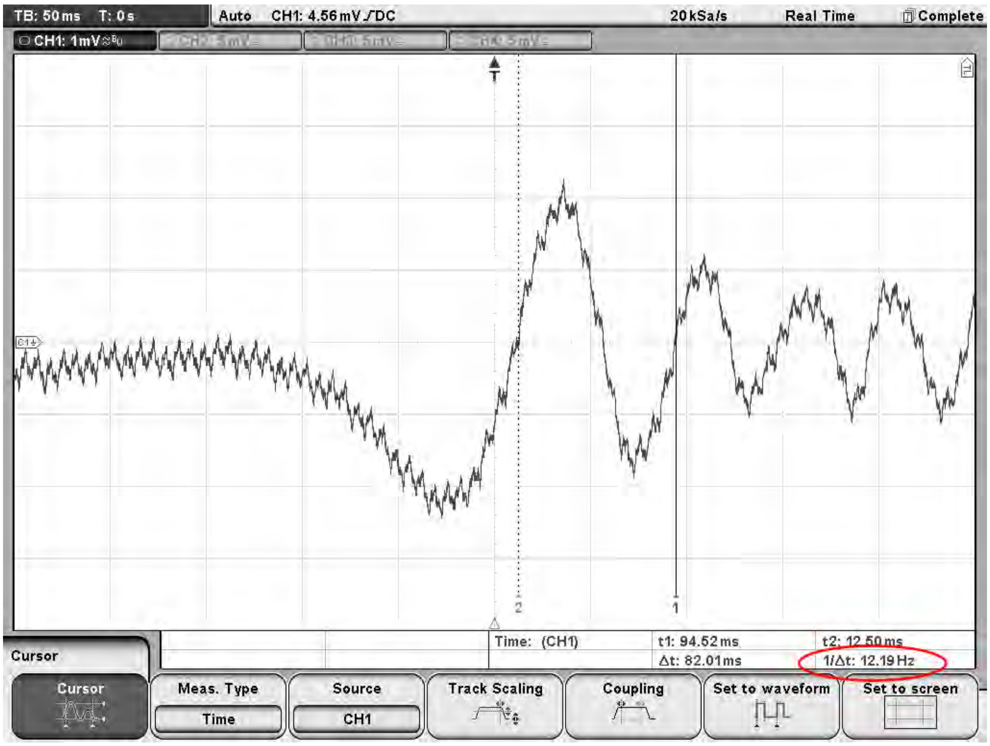


Fig. 2-6: Radar speed detection of moving objects

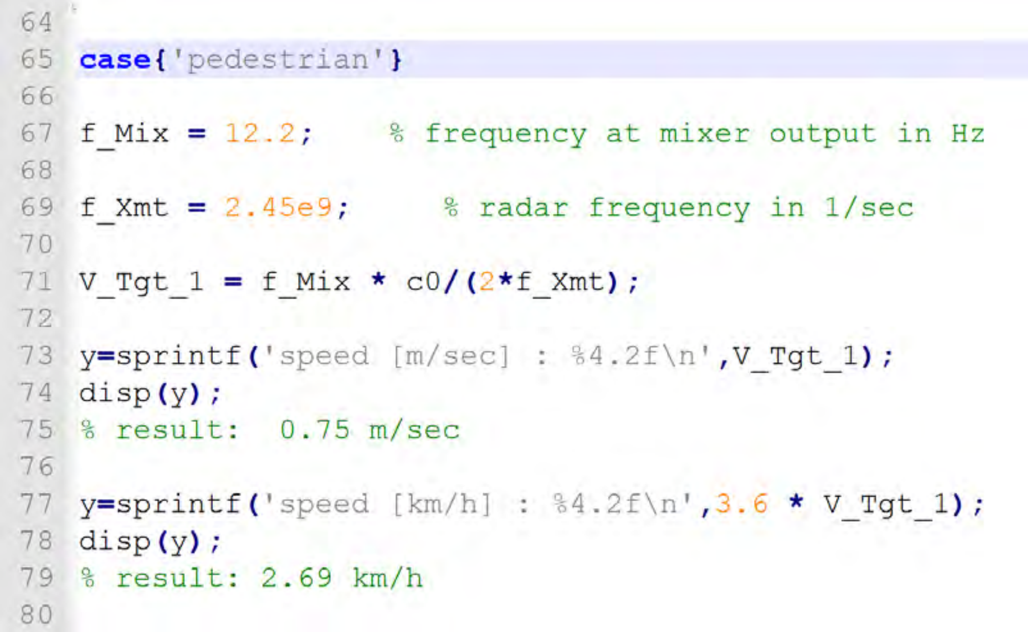
Using continuous wave signals this setup is suitable to perform speed measurements. The results are shown in Fig. 2-7. Using the cursor function of the oscilloscope the frequency of one signal period can be determined as shown in the red marked field of Fig. 2-7. For constant radial velocity the FFT function of the oscilloscope can be also used to determine the frequency.


TB: 50 ms T: 0 s
Auto CH1: 4.56 mV ⁄DC
20 kSa/s
Real Time
Complete
CH1: 1 mV

Time: (CH1)
t1: 94.52 ms
Δt: 82.01 ms
t2: 12.50 ms
1/Δt: 12.19 Hz

Fig. 2-7: Oscilloscope to determine the mixer output frequency

A MATLAB program helps to calculate the speed of the object as shown in Fig. 2-8. The result is 2.69 km/h in this example. The whole application can be completely remote controlled using MATLAB. In this case the time evaluation is done using half a period of the mixer output signal in contrast to Fig. 2-7, where a full period was used for convenience. The MATLAB program can also perform speed evaluation over time and display the acceleration in a diagram. The actual experiment was performed using a 2.45 GHz signal. Higher frequencies for instance at 24 GHz can provide more accurate measurements as the Doppler shift is higher and easier to measure. The formulas in Fig. 2-5, lines 58 and 59 show the relationship between the RF frequency and the mixer output signal.

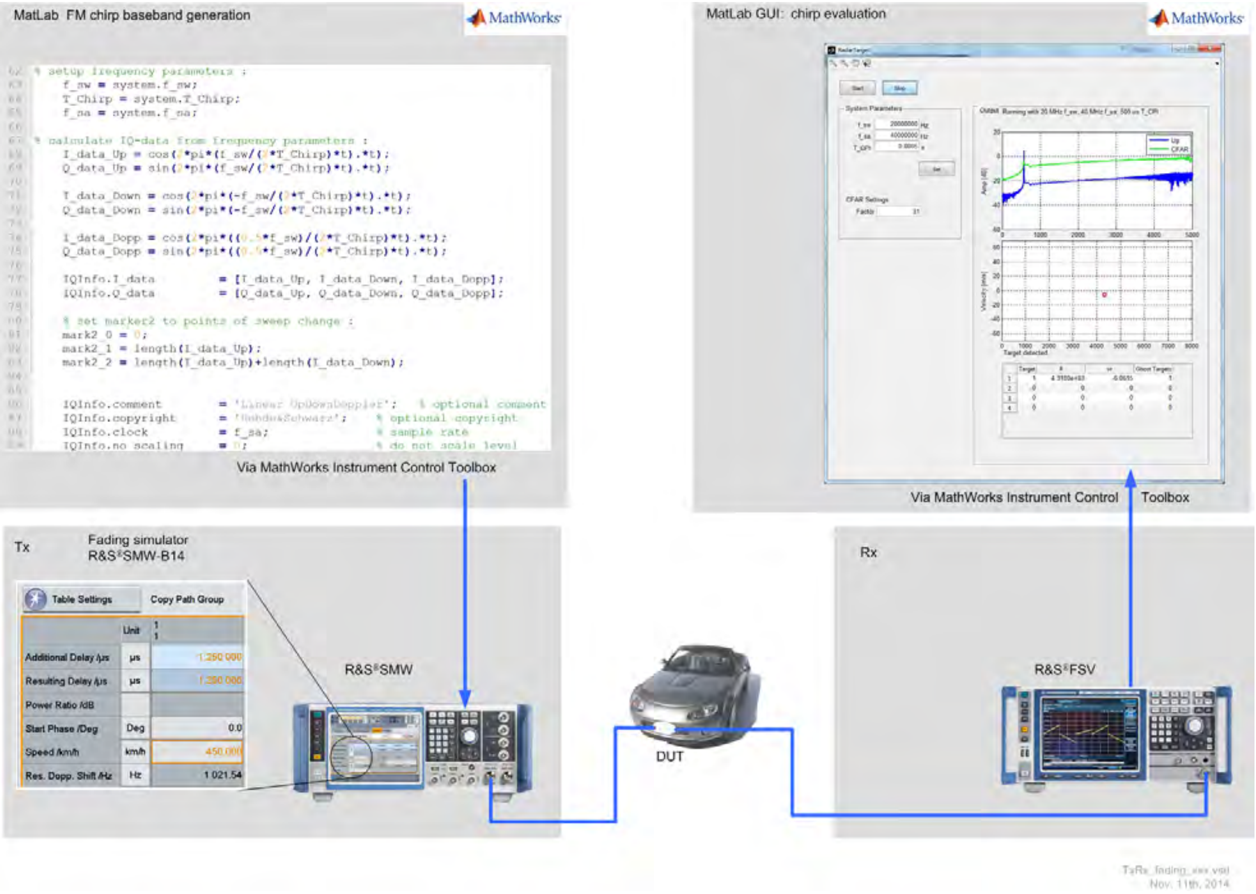


```

case{'pedestrian'}

f_Mix = 12.2;     % frequency at mixer output in Hz

f_Xmt = 2.45e9;      % radar frequency in 1/sec

V_Tgt_1 = f_Mix * c0/(2*f_Xmt);

y=sprintf('speed [m/sec] : %4.2f\n',V_Tgt_1);
disp(y);
% result:  0.75 m/sec

y=sprintf('speed [km/h] : %4.2f\n',3.6 * V_Tgt_1);
disp(y);
% result: 2.69 km/h

```

Fig. 2-8: MATLAB program to calculate the speed from the mixer frequency

This example has been tested under lab conditions with short ranges of 2 -3 m and devices with a size of 0.25 sqm. In order to create such systems for open air applications with real pedestrians the power requirements have to be taken into account also. The power requirements are detailed in chapter 4 "signal power discussion" and provide hints on the requirements of amplfiers, antennas and the size of objects being detected.

When the so far discussed experimental radar systems are reaching their technical limits in terms of radial velocity or distance, another method, called "target simulation" can be used. As shown in Fig. 2-9 the problem can be solved by means of the off-the-shelf fading simulator option as for example available for signal generators of the SMW family. The transmitted signal can be Doppler shifted using the fading simulator included in the generator. The configuration makes the resulting signal appear like a signal being reflected from a moving object. At the receiving side, this is detected by the signal analyzer accordingly.



```
% setup frequency parameters :
    f_sw = system.f_sw;
    T_Chirp = system.T_Chirp;
    f_sa = system.f_sa;

% calculate IQ-data from frequency parameters :
    I_data_Up = cos(2*pi*(f_sw/(2*T_Chirp)*t).*t);
    Q_data_Up = sin(2*pi*(f_sw/(2*T_Chirp)*t).*t);

    I_data_Down = cos(2*pi*(-f_sw/(2*T_Chirp)*t).*t);
    Q_data_Down = sin(2*pi*(-f_sw/(2*T_Chirp)*t).*t);

    I_data_Dopp = cos(2*pi*((0.5*f_sw)/(2*T_Chirp)*t).*t);
    Q_data_Dopp = sin(2*pi*((0.5*f_sw)/(2*T_Chirp)*t).*t);

    IQInfo.I_data          = [I_data_Up, I_data_Down, I_data_Dopp];
    IQInfo.Q_data          = [Q_data_Up, Q_data_Down, Q_data_Dopp];

    % set marker2 to points of sweep change :
    mark2_0 = 0;
    mark2_1 = length(I_data_Up);
    mark2_2 = length(I_data_Up)+length(I_data_Down);


    IQInfo.comment         = 'Linear UpDownDoppler';   % optional comment
    IQInfo.copyright       = 'Rohde&Schwarz';     % optional copyright
    IQInfo.clock           = f_sa;                % sample rate
    IQInfo.no_scaling      = 0;                   % do not scale level
```

	Unit	1 1
Additional Delay /µs	µs	1.250 000
Resulting Delay /µs	µs	1.250 000
Power Ratio /dB		
Start Phase /Deg	Deg	0.0
Speed /km/h	km/h	450.000
Res. Dopp. Shift /Hz	Hz	1 021.54

Fig. 2-9: Fading option to simulate range and speed in radar systems (*)

In contrast to Fig. 2-9 where an object was simulated at a range of 4.3 km, sometimes the simulation of high speed objects is needed, passing the RADAR in a close range, up to 200 m. Supposed we have an arbitrary baseband signal, for example an FM chirp signal created by MATLAB, we can set a delay, eg. 1.25 usec and speed, eg. 450 km/h, which both are applied to the modulated signal and is led to the output of SMW. The signal appears to the DUT as being reflected from an object with a distance range of 187 m and a speed of 450 km/h. For an experimental system It would be hard to implement such a fast object in a area close to the test equipment. The basic principle of the signal generator fading options are best described in [10]. According to the fading parameter specifications [15] the maximum frequency shift is 4000 Hz, which equates an object speed of more than 2.000 km/h for 1 GHz radar carrier frequency. Various MATLAB calculation examples are given in the file "doppler_3.m" of the programs attached to this document. Regarding distance simulation the appropriate delay can be set up to 0.5 sec, which equates a two way distance of more than 75.000 km. The range setting is very low because of the superposition with a fine-pitch delay. Therefore the fading simulator can be considered as a widerange simulation tool in terms of speed and distance, applicable for the simulation of moving shipborne and airborne devices.

This example is using FM chirp signals as it is common in state-of-the-art radar based measurements for speed and distance. The video "Analysis of FMCW radar signals" [11] provides a comprehensive introduction

Please take into account also the power issues as discussed in chapter 4 "signal power discussion" when designing such systems.

(*) Hint: the M code in the upper left of Fig. 2-9 can be read by zooming into the PDF document, additionally the full code to create FM chirp signals is available in the file attachment of the educational note.

3 R&S software solution for radar signal generation

In state-of-the-art radar applications there are high signal processing requirements as outlined in the previous text. There are some vendor specific software model libraries for development and verification of radar systems available. The R&S software solution for radar signal generation is based on an open model environment such as MATLAB, C++ and VHDL. The design goals for the R&S radar test software have been: vendor independence, low entry level in terms of investments and staff education and easy implementation into target systems, eg. airborne devices or customer environments such as production plants.

Fig. 3-1 shows a simplified block diagram of the R&S software solution for radar signal generation. In order to feed the signal generator with a radar baseband signal there are basically two alternatives as indicated by the switch.

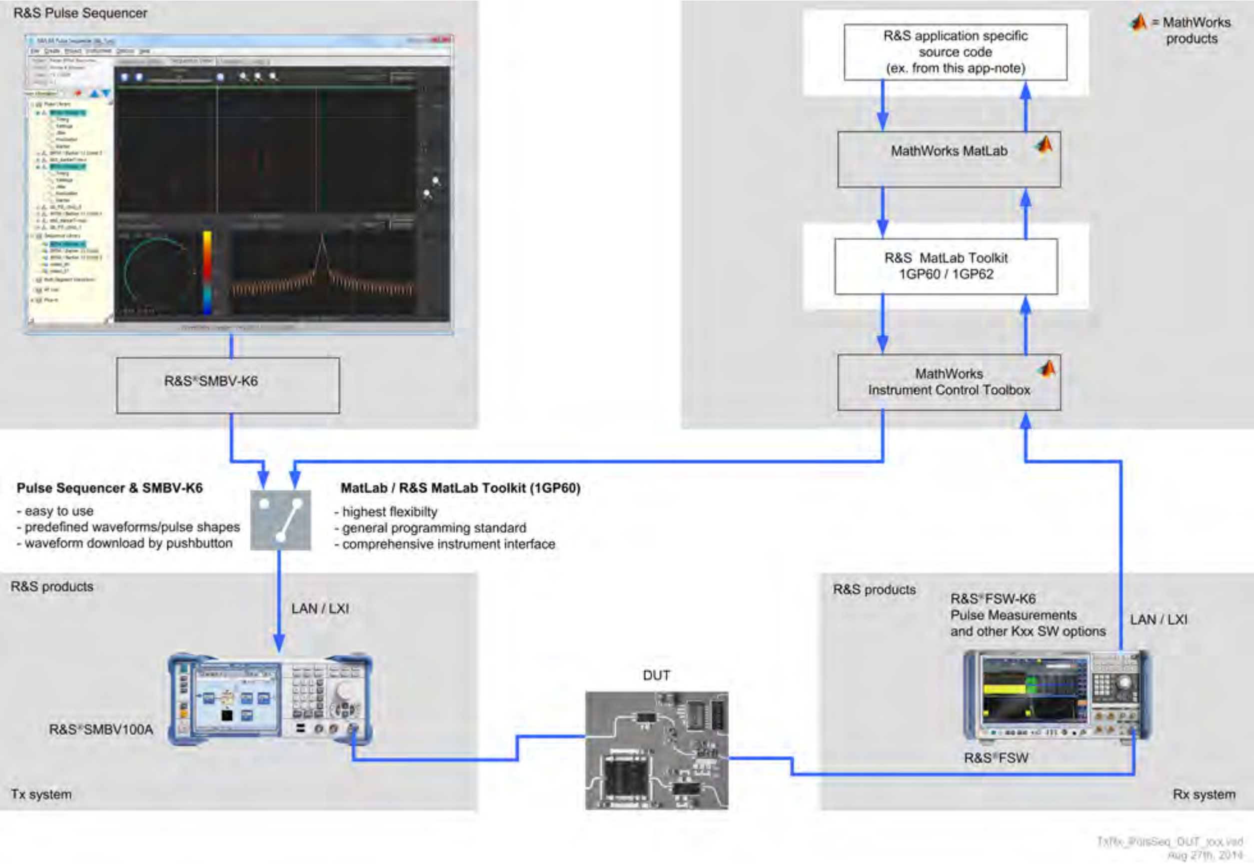
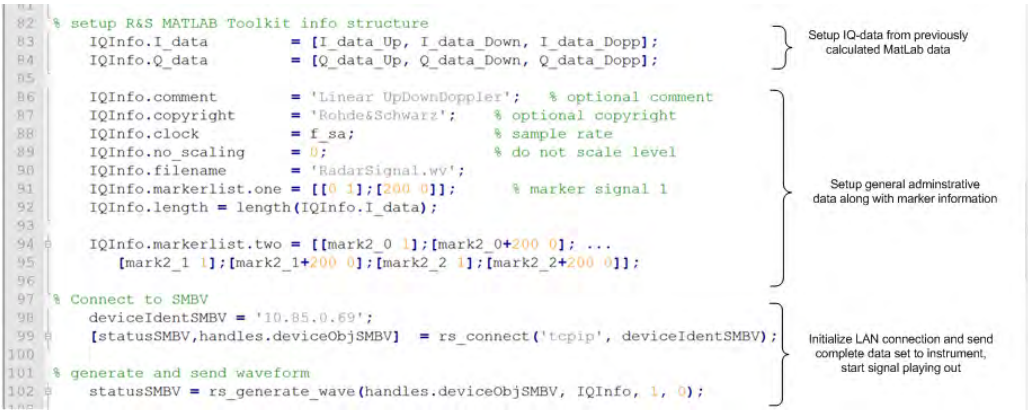


Fig. 3-1: Easy choices in the R&S radar software concept

The more convenient method, shown on the left, is using the R&S Pulse Sequencer software, which is applicable for various R&S signal generator families. It can be freely downloaded from the R&S homepage [12]. The free software is useful to perform first trials and to get an impression about the user interface. Additionally the final waveform file can be analyzed and displayed as shown in Fig. 3-1. However, in order to download the final waveform via the LAN interface the option SMBV-K6 must be installed on the Signal Generator where the radar baseband signal will be used. The pulse sequencer consists of a pulse library and a sequencer library. Both together are providing a complete set of common radar baseband signals, like BPSK modulated Barker Sequence, multiphase codes like Frank, P1, P2 or FM chirp signals. After entering the parameters or after selecting a predefined pulse pattern and modulation, the basband signal can be reviewed graphically as shown in Fig.3-1. By means of the R&S Pulse Sequencer software the so called "markers" can be easily defined. The "marker" signals are led to external connectors of the signal generator and are useful to trigger external devices such as a signal analyzer or an oscilloscope. When the waveform is being defined and calculated, the waveform can be downloaded and started by push button sending appropriate commands directly via LAN.

Alternatively the baseband signal can be calculated using the MATLAB software along with R&S MATLAB Toolkit as shown at the right of the switch of Fig. 3-1. The R&S MATLAB Toolkit is freely available via AppNote 1GP60 [13]. It provides an comprehensive and easy to handle interface between standard MATLAB "number crunching" software and the instrument underneath. For example the IQ-data of a three-slope FM chirp signal can be easily calculated. By means of the versatile MATLAB Toolkit, the calculated IQ-data is downloaded to the instrument and is directly started along with trigger markers previously defined in MATLAB.



```
 82  % setup R&S MATLAB Toolkit info structure
 83      IQInfo.I_data          = [I_data_Up, I_data_Down, I_data_Dopp];
 84      IQInfo.Q_data          = [Q_data_Up, Q_data_Down, Q_data_Dopp];
 85
 86      IQInfo.comment         = 'Linear UpDownDoppler';   % optional comment
 87      IQInfo.copyright       = 'Rohde&Schwarz';    % optional copyright
 88      IQInfo.clock           = f_sa;               % sample rate
 89      IQInfo.no_scaling      = 0;                  % do not scale level
 90      IQInfo.filename        = 'RadarSignal.wv';
 91      IQInfo.markerlist.one = [[0 1];[200 0]];       % marker signal 1
 92      IQInfo.length = length(IQInfo.I_data);
 93
 94      IQInfo.markerlist.two = [[mark2_0 1];[mark2_0+200 0]; ...
 95         [mark2_1 1];[mark2_1+200 0];[mark2_2 1];[mark2_2+200 0]];
 96
 97  % Connect to SMBV
 98      deviceIdentSMBV = '10.85.0.69';
 99      [statusSMBV,handles.deviceObjSMBV]  = rs_connect('tcpip', deviceIdentSMBV);
100
101  % generate and send waveform
102      statusSMBV = rs_generate_wave(handles.deviceObjSMBV, IQInfo, 1, 0);
```

Fig. 3-2: Performing instrument waveform download using R&S MATLAB toolkit

Fig. 3-2 shows the easy setup of the three stage FM signal to be downloaded to SMBV. Lines 83 / 84 show preparation of I/Q data from previously performed MATLAB code, lines 86 - 95 show some administrative data setup. Via lines 97 - 102 the data is finally send to the instrument and playing out the signal is directly started. The MATLAB code attached to the educational note is providing the full set of lines of this software. In addition to arbitrary waveform generation the toolkit provides direct instrument programming commands such as "rs_send_command" and "rs_send_query" which both also can be used for other instrument control beyond signal generators. The commands for instance have been used also to remote control the signal analyzers of the FSV- or FSW -family as shown in the lower right of Fig. 3-1. Using standard software like MATLAB for radar signal processing tasks has many advantages over vendor specific software tools. In radar literature many signal processing examples are provided in MATLAB code, sometimes also simply referred to as "m" code, reference [8] provides a full set of MATLAB source code examples for radar applications.

4 Signal power discussion

This chapter is addressing power requirements which have to be taken into account when designing radar systems based on test instruments. Power requirements have already been discussed in [5] and [6]. Additionally the radar tutorial video [9] provides an introduction to the radar equation, which is essential for power calculations as provided in this chapter. In appendix 5.1.2 the receiving power of some practical scenarios is calculated by means of a small MATLAB program. Best impedance matching with negligible reflection losses is assumed for all examples herein discussed.

Besides the antenna gain there are two main parameters to be considered when designing radar systems. First, the size of the object to be detected and second the range. The size of reflecting objects is represented by the parameter "Radar Cross Section" (RCS). The parameter is expressed in square meters (sqm) and depends on shape, material, frequency and viewing angle. Following table shows some RCS values [1] applicable for microwave frequencies (*) :

Target	6 (sqm)
Bird	0.01
Missile	0.5
Person	1.0
Small plane	1 - 2
Bicycle	2
Small boat	2
Fighter plane	3 - 8
Bomber	30 - 40
Large airliner	100
Truck	200

rcs_1.vsd
July 28th, 2014

Fig. 4-1: Radar Cross Section (RCS) simulation objects and typical values [1]

(*) These values are provided just to give a rough idea on RCS values only. Normally RCS values are specified along with their determining conditions for instance frequency, viewing angle and environment (sea/air).

Picture 4-2 gives a visual impression of typical RCS values for microwave frequencies.

Fig. 4-2: Comparing typical examples of radar cross section (RCS)

The values according to Fig. 4-1 can be used to estimate the RCS values for various objects to be detected by radar. For example, when using the R&S power amplifier BBA150 output power of 56 dBm is available. Fig. 4-3 shows an appropriate example of an experimental radar system.

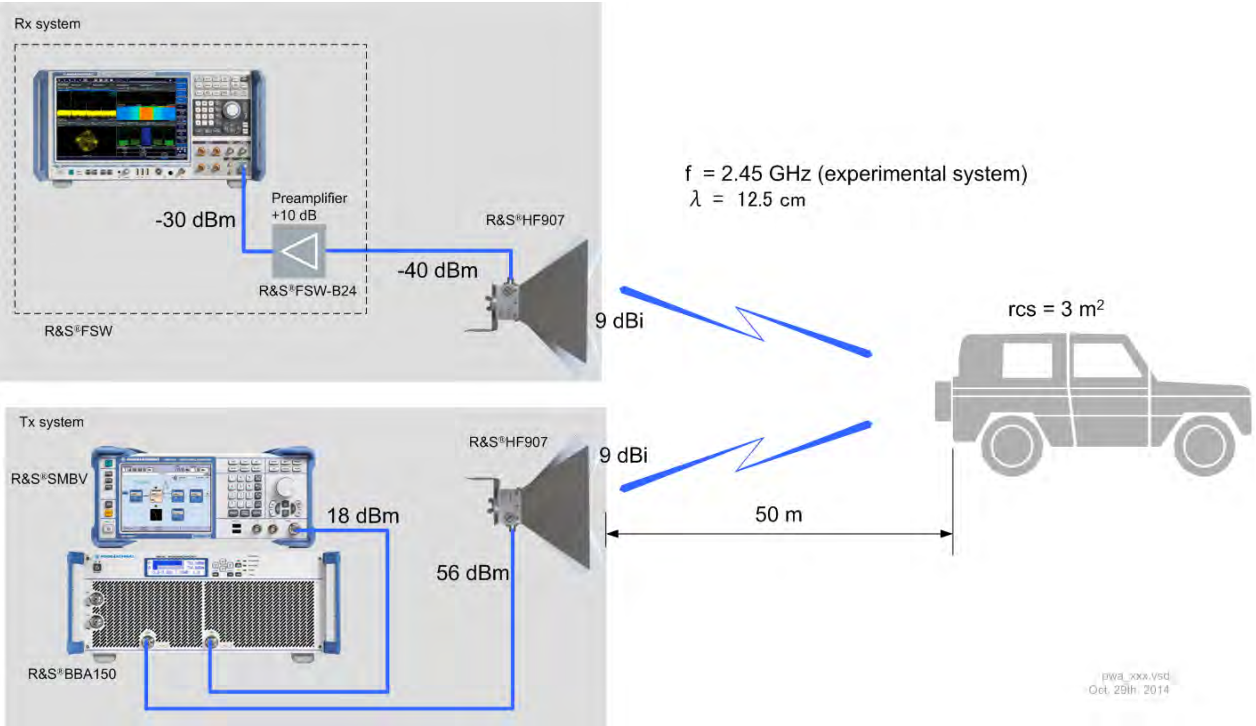


Fig. 4-3: Power values within typical experimental radar system

For the R&S horn antenna HF907 the gain at 2.4 GHz is specified to 9 dBi. If we want to detect an SUV now (RCS = 3 sqm) in a distance of 50 m we can expect a receiving power of -40 dBm. When designing transmission systems according to fig. 4-3 the maximum feed power for the antenna has to be taken into account also. However, the maxim PEP feed power of the HF907 is specified to 57 dBm, therefore the antenna can be used for this purpose.

In order to evaluate the power situation within the entire system we have to take into account the receiving performance also. The FSW "IF power trigger sensitivity" is specified to -60 dBm [19]. The receiving sensitivity of FSW can be improved by a preamplifier, option FSW-B24, increasing the effective trigger sensitivity by at least 10 dB. Fig. 2-7 shows the receiving input levels for various radar scenarios. The trigger sensitivity for FSW standard configuration is shown as a blue dashed line. When the preamplifier option FSW-B24 is installed the 10 dB improved red dashed line is valid. The black diamonds show the input receiving level for following radar scenarios:

- Scenario "Pozar" stems from reading [1], page 662, signal pulse power of 63 dBm (2 kW) using a high gain antenna of 28 dBi, identifying an object in over 8.000m with Radar cross section 12 sqm, i.e. a medium sized airplane. The power at the receiving antenna is calulated to -90 dBm. Radar frequency is 10 GHz, X-band in this case, while for the upcoming scenarios a frequency of 2.45 GHz, S-band is used.

- "SUV_no_amp", a scenario using a SMBV signal generator (18 dBm) directly connected to R&S horn antenna HF907 (9 dBi), detecting a SUV (RCS = 3 sqm) in a distance of 50 m. This is similar to Fig. 4-3 with the exception that the power amplifier BBA150 is replaced by short cut. Resulting receiving power is -78 dBm.

- "MilitaryPlane", using a BBA150, 56 dBm, 400 W, power amplifier connected to antenna HF907, detecting a big plane (RCS = 60 sqm) in a distance of only 500 m. Receiving power -67 dBm, this could be internally triggered when the FSW-B24 is available.

- "Lab_Cond_no_amp", SMBV , 18 dBm, directly connected to HF907 horn antenna, 9dBi, detecting a small item according to Fig. 2-2, located in a distance of 8 m. Receiving power of -54 dBm, which could be internally triggered even without FSW-B24.This "Laboratory condition" scenario can be operated in a small area. However, clutter reflections from other devices like walls or furniture have to be taken into account in this case. Clutter reflections can also be kept small in an open air environment or focusing the target object by high gain antennas.

- "SUV_with_amp" is shown in Fig. 4-3, receiving gain is - 40 dBm.

- "Lab_Cond_with_amp" is similar to "Lab_Cond_no_amp" but using the 56 dBm power amplifier BBA150, receiving power - 23 dBm. This example shows how experimental radar systems can be created directly under laboratory conditions.

The calculations of the receving input levels is based upon a small MATLAB program, refer to appendix 5.1.2. This progam can be used to perform additional power calculations and it shows also all details on the different radar scenarios as herein stated.

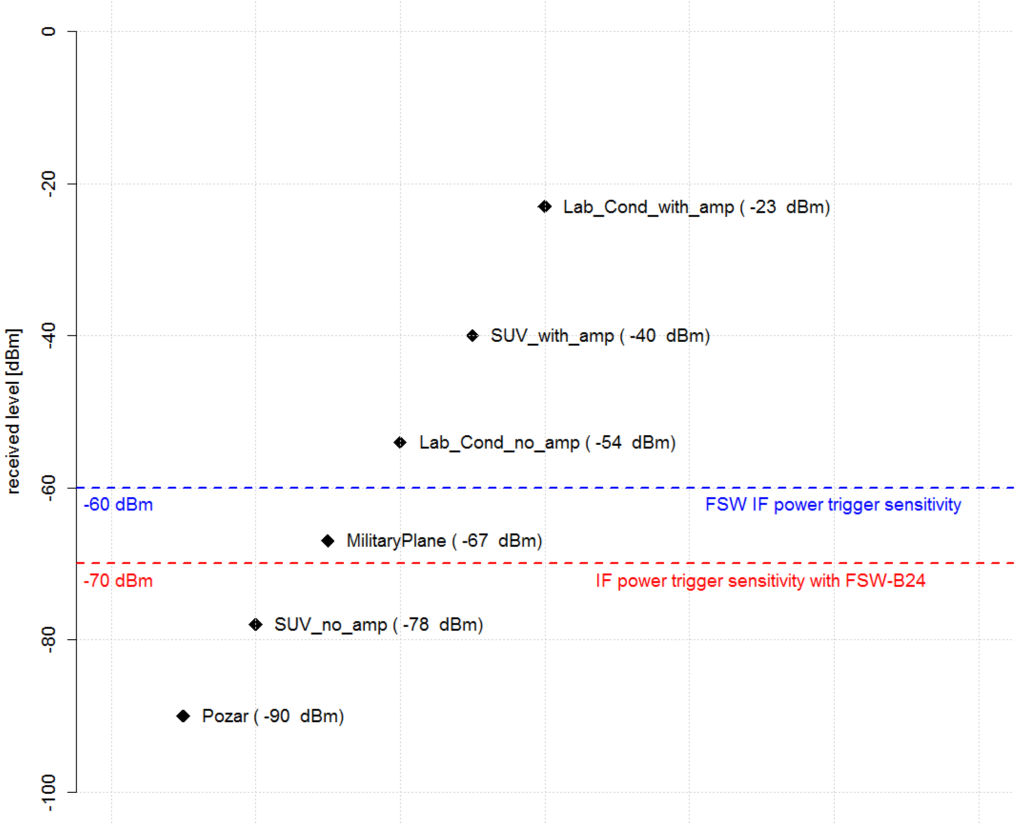


Fig. 4-4: Reveiving input levels for various radar scenarios (typical radar bands)

We can expect suitable results for all scenarios located above the dashed lines. For scenarios below the dashed lines additonal measures are needed. For instance by increasing the transmitter amplifier pulse power, increasing antenna gain, for instance using small horn antennas as shown in Fig. 4-5. In contrast to the broad band HF907 antenna horn antennas shown in Fig. 4-5 are specified for a dedicated frequency band but are providing gain of more than 20 dBi.

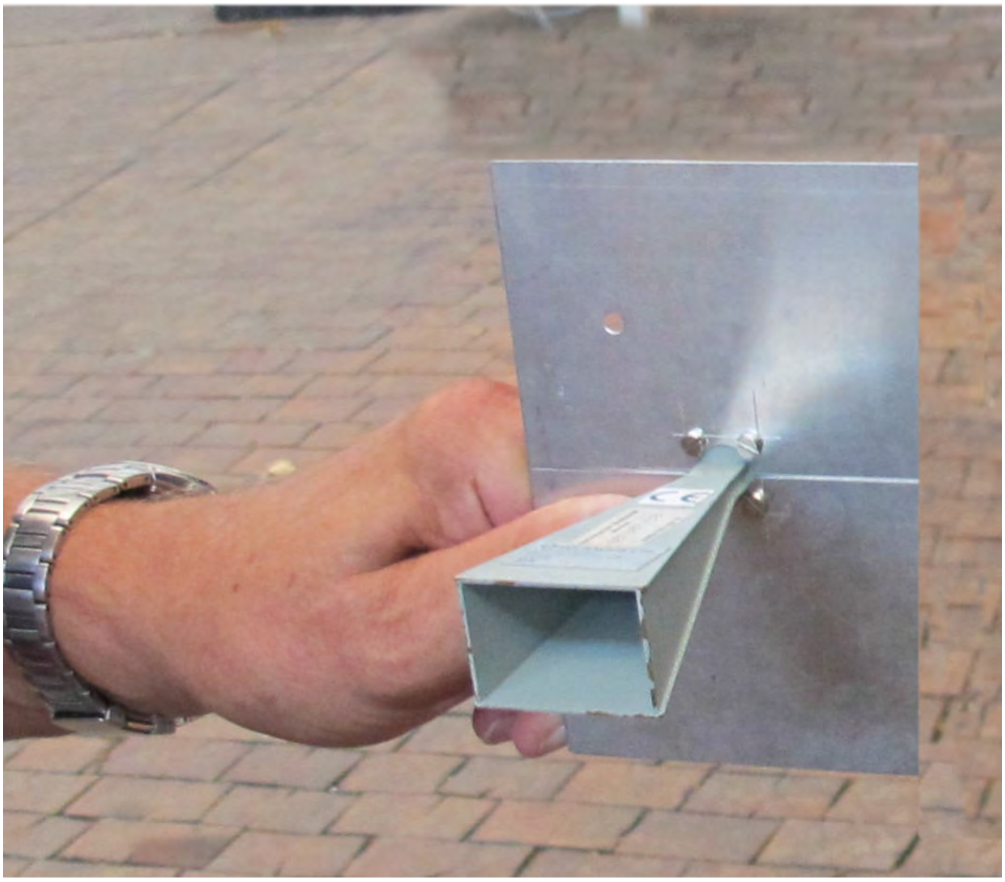


Fig. 4-5: Small horn antenna providing more than 20 dBi gain

Additionally external trigger of FSW can be used at least for active radar systems where transmitter and receiver are operated close together. This is possible when the transmitting SMBV and the receiving FSW are located close together, which is normally the case for active radar systems. For passive radar systems [6], where transmitter and receiver can be a long way away from each other, two cases have to be taken into account:

(1) A common time reference is needed, because the round trip delay of the radar signal has to be determined. In this case receiver and transmitter have to be synchronized using a GPS system

(2) No common time reference is needed, eg. when performing pure Doppler measurements. In this case the receiving system, i.e. the signal / spectrum analyzer, can be triggered directly by the received signal. In this case the preamplifier option FSW-B24 is recommended.

The equipment introduced so far is suitable for laboratory conditions or small area experiments, as demonstrated for example by the "SUV" scenarios. The BBA150 can provide 56 dBm (400 W) and thus can make tests up to some kilometers along with sensitivity increasing signal pocessing. However, if experiments need to be extended beyond, eg. up to some hundred kilometers, more powerful amplifiers are needed to overcome the path attenuation. Fig. 4-6 shows a typical commercial device for maritime applications, where the power requirements for real world radar systems can be seen.

Raytheon R41

X-Band :	9445 +/-30 MHz
Pulse power :	**3 KW =~ 64 dBm**
Pulse length :	0,12 / 0,5µs
Pulse repetition rate :	920 / 920 Hz
Intermediate frequency :	60 MHz
Polarisation :	Horizontal
Distance resolution :	< 22m
Azimut resolution :	<= 2°
Minimum distance :	< 32m
Maximum distance :	**32 NM =~ 60 km**

Fig. 4-6: Commercial maritime device, pulse power and maximum distance

The technical data shows that in order to overcome distances of 60 km a pulse power of 64 dBm is needed, neglecting radio cross section, most likely supposing big ships with RCS values more than 30 sqm.

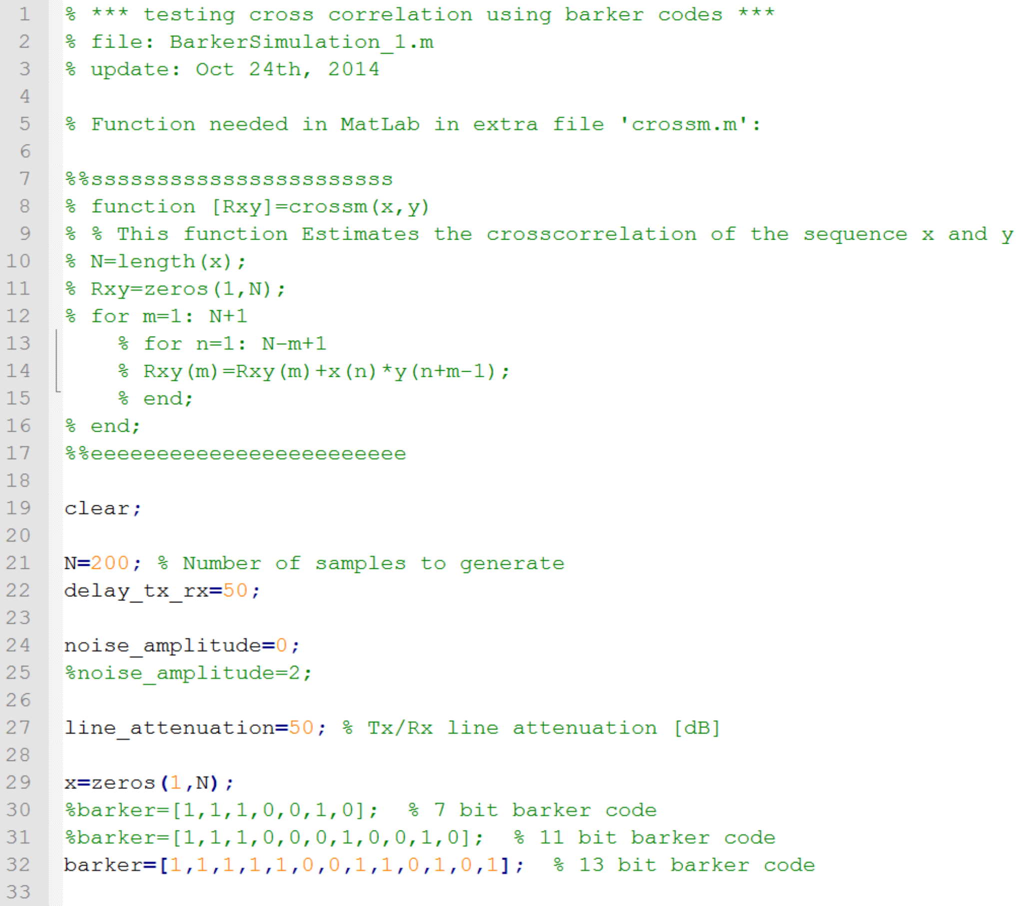
5 Appendix

5.1 MATLAB programs

This appendix provides description of MATLAB programs going beyond descriptions already included in the source files. The source files are provided with line numbers in order to support the descriptions herein given. Therefore it is not possible to get the source code from this document. However the sources are provided as attachment to this educational note, refer to [16] for this purpose.

5.1.1 Cross correlation using Barker codes

This program simulates a transmission and reception of a Barker code signal. The received signal is correlated with the transmitted one in order to find the time delay between transmit and receive. Fig. 5-1 and Fig. 5-2 show the program.



```
% *** testing cross correlation using barker codes ***
% file: BarkerSimulation_1.m
% update: Oct 24th, 2014

% Function needed in MatLab in extra file 'crossm.m':

%%sssssssssssssssssssssss
% function [Rxy]=crossm(x,y)
% % This function Estimates the crosscorrelation of the sequence x and y
% N=length(x);
% Rxy=zeros(1,N);
% for m=1: N+1
    % for n=1: N-m+1
    % Rxy(m)=Rxy(m)+x(n)*y(n+m-1);
    % end;
% end;
%%eeeeeeeeeeeeeeeeeeeeeeee

clear;

N=200; % Number of samples to generate
delay_tx_rx=50;

noise_amplitude=0;
%noise_amplitude=2;

line_attenuation=50; % Tx/Rx line attenuation [dB]

x=zeros(1,N);
%barker=[1,1,1,0,0,1,0];  % 7 bit barker code
%barker=[1,1,1,0,0,0,1,0,0,1,0];  % 11 bit barker code
barker=[1,1,1,1,1,0,0,1,1,0,1,0,1];  % 13 bit barker code
```

Fig. 5-1: Crosscorrelation using Barker codes, part 1 of 2

Lines 8 to 16 provide a replacement for the cross correlation which in MATLAB is normally available only along with the Signal Processing Toolbox. In order to run the program in standard MATLAB without the need of additional tool boxes, lines 8 to 16 need to be stored in an extra file "crossm.m" .

```
34  x(1:length(barker))=5*barker;
35  subplot(3,1,1);
36  stairs(x);
37  title('Transmitted signal');
38  ylim([0,7]);
39  xlabel('[usec]');
40  ylabel('[dB]');
41  grid();
42
43  xd=[zeros(1,delay_tx_rx), x(1:N-delay_tx_rx)];  % signal delay
44  xd=0.7*xd(1:N);   % signal attenuation
45
46  w=noise_amplitude*randn(1,N);   % create noise signal
47
48  r=xd+w; % create the sum signal
49  subplot(3,1,2);
50  plot(r-50);
51  title('Received signal');
52  xlabel('[usec]');
53  ylim([-60,-40]);
54  ylabel('[dB]');
55  grid();
56
57  Rxy=crossm(x,r); % Estimate the cross correlation
58  Rxy=Rxy/max(Rxy);
59  subplot(3,1,3);
60  plot(Rxy);
61  title('Cross correlation transmit receive');
62  xlabel('[usec]');
63  ylim([0,1]);
64  grid;
65
66  disp('maximum found at sample : ');
67  disp(find(Rxy==max(Rxy)));
```

Fig. 5-2: Crosscorrelation using Barker codes, part 2 of 2

The appropriate MATLAB function "crossm" is called in line 57 of the main program. Lines 29 to 34 create and plot the transmitted signal. Lines 34 to 41 simulate the reception with attenuation (line 44) and noise (46) . The crosscorrelation finally is calculated and plotted in lines 57 to 64. Finally the maximum is determined and printed (lines 66, 67). Fig. 1-2 of this document has been created using this MATLAB code. The curve shapes are changing from call to call because of the noise of the receive function. (line 46).

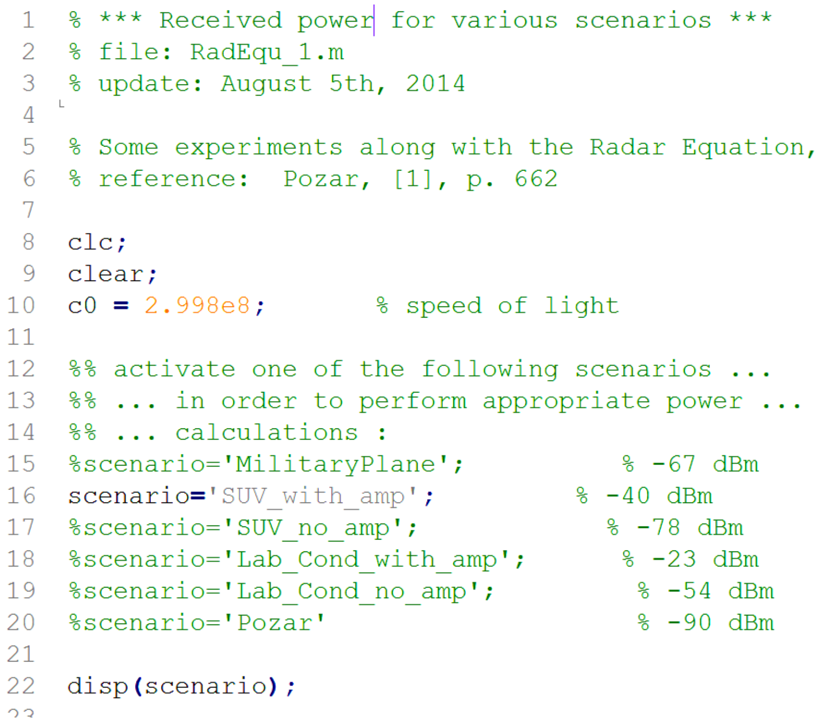
In order to implement this method with instruments this example provides basic hints how to implement an appropriate system for instance according to Fig. 1-1. Up to line 41 the base band signal is calculated for a signal generator, eg. SMBV, where it is BPSK modulated and transmitted. After reception of the reflected signal it is demodulated, eg. by FSW and FSW-B7 and then stored into the vector "r". Finally it is postprocessed according to lines 57 to 67, yielding the time delay and thus the distance to the object detected by radar. Because lines 43 to 55 are simulating the transmit/receive path, this code section is not used anymore when the system is implemented with instruments, similar to Fig. 1-1.

Chapter 5.1.3 of this educational note provides a complete MATLAB example, showing how to perform modulation and demodulation of the signals, including instrument programming.

The crosscorrelation function according to Fig. 5-1, lines 8 to 16 provides a straightforward time domain based method to demonstrate the principle of pulse compression. There is another method described in the literature, eg. [8], pg. 299 using a "correlation processor" based on frequency domain calculations. This method performs a FFT on both input signals and then retransforms the multiplied signal back into the time domain by means of Inverse FFT (IFFT). This can be important when implementing pulse compression systems. However, a technical evaluation with a detailed comparison of advantages and drawbacks of both methods is going beyond the scope of this document.

5.1.2 Pulse power scenarios

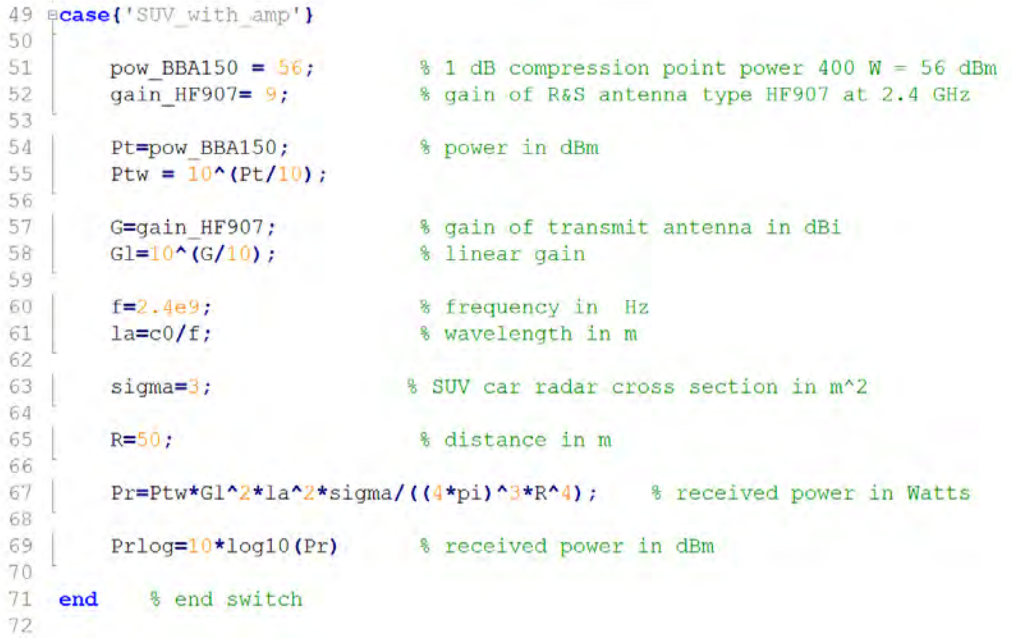
The various power scenarios from chapter 4 have been calculated with a MATLAB program shown as excerpt in figures 5-3 and 5-4 below. One scenario out of six in total is shown in Fig. 5-4. Fig. 5-3 shows the header of the program where the specific scenario is selected. The scenario to be calculated is chosen in the lines 15 to 20 by uncommenting the appropriate line. Actually 'SUV_with_amp' is active.



```
 1  % *** Received power for various scenarios ***
 2  % file: RadEqu_1.m
 3  % update: August 5th, 2014
 4
 5  % Some experiments along with the Radar Equation,
 6  % reference:  Pozar, [1], p. 662
 7
 8  clc;
 9  clear;
10  c0 = 2.998e8;       % speed of light
11
12  %% activate one of the following scenarios ...
13  %% ... in order to perform appropriate power ...
14  %% ... calculations :
15  %scenario='MilitaryPlane';          % -67 dBm
16  scenario='SUV_with_amp';         % -40 dBm
17  %scenario='SUV_no_amp';            % -78 dBm
18  %scenario='Lab_Cond_with_amp';      % -23 dBm
19  %scenario='Lab_Cond_no_amp';         % -54 dBm
20  %scenario='Pozar'                    % -90 dBm
21
22  disp(scenario);
```

Fig. 5-3: Excerpt of power scenarios calculation, part 1 of 2

This scenario is calculated in the lines 51 to 69. Lines 51 to 65 set up all parameters according to the scenario. The radar equation is coded in line 67. Line 69 is used to print the result to the console, because there is *no* semicolon at the end of the line. The entire program including all six scenarios is available in [16].



```
49  case('SUV_with_amp')
50
51      pow_BBA150 = 56;         % 1 dB compression point power 400 W = 56 dBm
52      gain_HF907= 9;           % gain of R&S antenna type HF907 at 2.4 GHz
53
54      Pt=pow_BBA150;           % power in dBm
55      Ptw = 10^(Pt/10);
56
57      G=gain_HF907;            % gain of transmit antenna in dBi
58      Gl=10^(G/10);            % linear gain
59
60      f=2.4e9;                 % frequency in  Hz
61      la=c0/f;                 % wavelength in m
62
63      sigma=3;                % SUV car radar cross section in m^2
64
65      R=50;                    % distance in m
66
67      Pr=Ptw*Gl^2*la^2*sigma/((4*pi)^3*R^4);    % received power in Watts
68
69      Prlog=10*log10(Pr)       % received power in dBm
70
71  end    % end switch
72
```

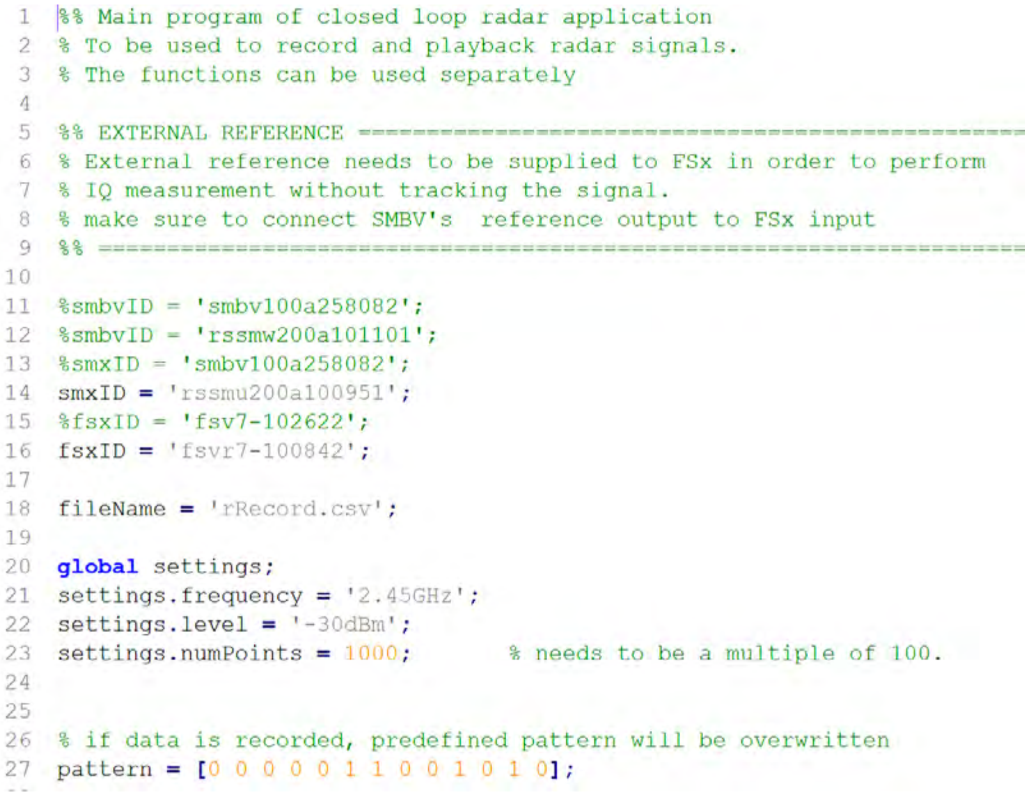
Fig. 5-4: Excerpt of power scenarios calculation, part 2 of 2

Further scenarios can be added by just copying additional 'case' clauses into the switch / end boundary as shown in the figure.

5.1.3 Closed loop radar

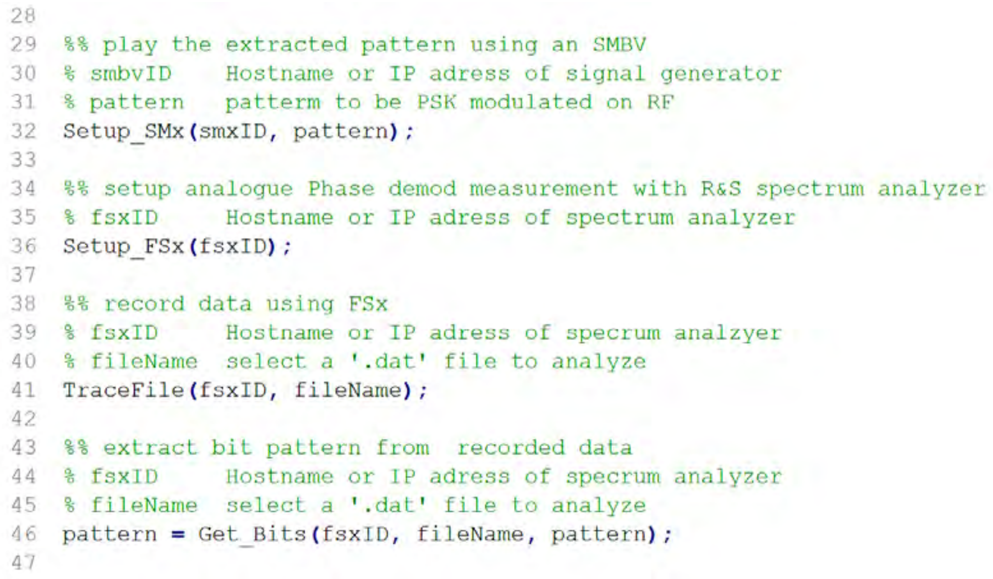
According to chapter 2.2 a simple loop of transmitter, test path / test device and receiver has been implemented. For special cases the received and demodulated signal can be used as modulation base for followup transmissions, either in original or jammed version. Fig. 5-5 shows the main part of the appropriate MATLAB code. In addition to the software main part, there are four additional MATLAB functions needed as called in lines 32, 36, 41 and 46. This chapter provides a look into the main part and the function "Setup_SMx". The whole set of programs is available along with the educational note at hand and can be downloaded from the internet [16] . As indicated in Fig. 2-2 one MATLAB Instrument Control toolbox license is needed to control the instruments in the Tx and the Rx paths. After having installed the system according to Fig. 2.2 and 2.4 the software can be directly started and is expected to provide results simliar to Fig. 2.3.

Lines 11 to 16 of the main part provide remote control IDs for various instruments the software has been tested with. Lines 20 to 23 define three global parameters which are used throughout the entire program, functions included. By means of the parameters the operating frequency and the transmit level of the generator can be defined globally. Line 27 defines the bit pattern for the BPSK modulation of the signal, which is transmitted and received. The function "Get_Bits" is used for demodulation. The transmitted bit sequence as well as the received one are both displayed in Fig. 2-3.



```
%% Main program of closed loop radar application
% To be used to record and playback radar signals.
% The functions can be used separately

%% EXTERNAL REFERENCE ================================================
% External reference needs to be supplied to FSx in order to perform
% IQ measurement without tracking the signal.
% make sure to connect SMBV's  reference output to FSx input
%% ====================================================================

%smbvID = 'smbv100a258082';
%smbvID = 'rssmw200a101101';
%smxID = 'smbv100a258082';
smxID = 'rssmu200a100951';
%fsxID = 'fsv7-102622';
fsxID = 'fsvr7-100842';

fileName = 'rRecord.csv';

global settings;
settings.frequency = '2.45GHz';
settings.level = '-30dBm';
settings.numPoints = 1000;       % needs to be a multiple of 100.


% if data is recorded, predefined pattern will be overwritten
pattern = [0 0 0 0 0 1 1 0 0 1 0 1 0];
```

Fig. 5-5: Main program of the closed loop radar application, part 1 of 2



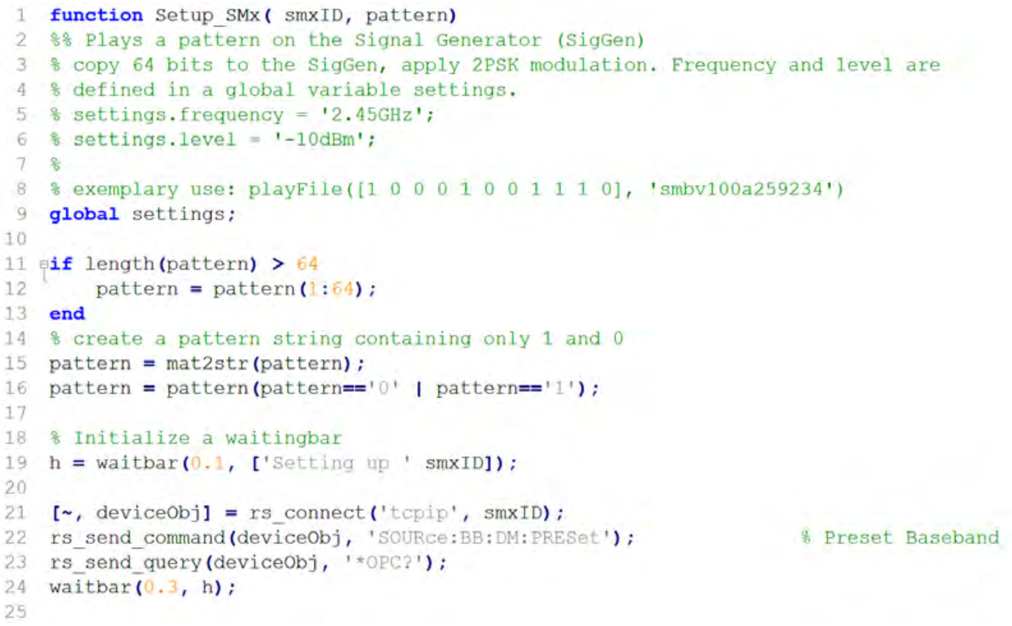
```

%% play the extracted pattern using an SMBV
% smbvID    Hostname or IP adress of signal generator
% pattern   pattern to be PSK modulated on RF
Setup_SMx(smxID, pattern);

%% setup analogue Phase demod measurement with R&S spectrum analyzer
% fsxID     Hostname or IP adress of spectrum analyzer
Setup_FSx(fsxID);

%% record data using FSx
% fsxID     Hostname or IP adress of specrum analzyer
% fileName  select a '.dat' file to analyze
TraceFile(fsxID, fileName);

%% extract bit pattern from  recorded data
% fsxID     Hostname or IP adress of specrum analyzer
% fileName  select a '.dat' file to analyze
pattern = Get_Bits(fsxID, fileName, pattern);

```

Fig. 5-6: Main program of the closed loop radar application, part 2 of 2

Lines 29 to 46 provide the calls to the four functions being detailed in the upcoming text.

5.1.3.1 Setup_SMx

This function provides the setup of the transmitting device. Various types of R&S generators have been tested, eg. from the SMW and SMBV familiy. Fig. 5-7 and 5-8 show the complete listing of the function's MATLAB code.



```
function Setup_SMx( smxID, pattern)
%% Plays a pattern on the Signal Generator (SigGen)
% copy 64 bits to the SigGen, apply 2PSK modulation. Frequency and level are
% defined in a global variable settings.
% settings.frequency = '2.45GHz';
% settings.level = '-10dBm';
%
% exemplary use: playFile([1 0 0 0 1 0 0 1 1 1 0], 'smbv100a259234')
global settings;

if length(pattern) > 64
    pattern = pattern(1:64);
end
% create a pattern string containing only 1 and 0
pattern = mat2str(pattern);
pattern = pattern(pattern=='0' | pattern=='1');

% Initialize a waitingbar
h = waitbar(0.1, ['Setting up ' smxID]);

[~, deviceObj] = rs_connect('tcpip', smxID);
rs_send_command(deviceObj, 'SOURce:BB:DM:PRESet');             % Preset Baseband
rs_send_query(deviceObj, '*OPC?');
waitbar(0.3, h);

```

Fig. 5-7: Generator setup for the closed loop radar application, part 1 of 2

Lines 26 and 27 show the setup of generator frequency and power according to the appropriate global variables defined in the main program. The MATLAB instrument programming via "rs_send_command" and "rs_send_query" is typical for "MATLAB Toolkit" instrument programming according to Fig. 3-1. The MATLAB Toolkit is explained in detail in [13].

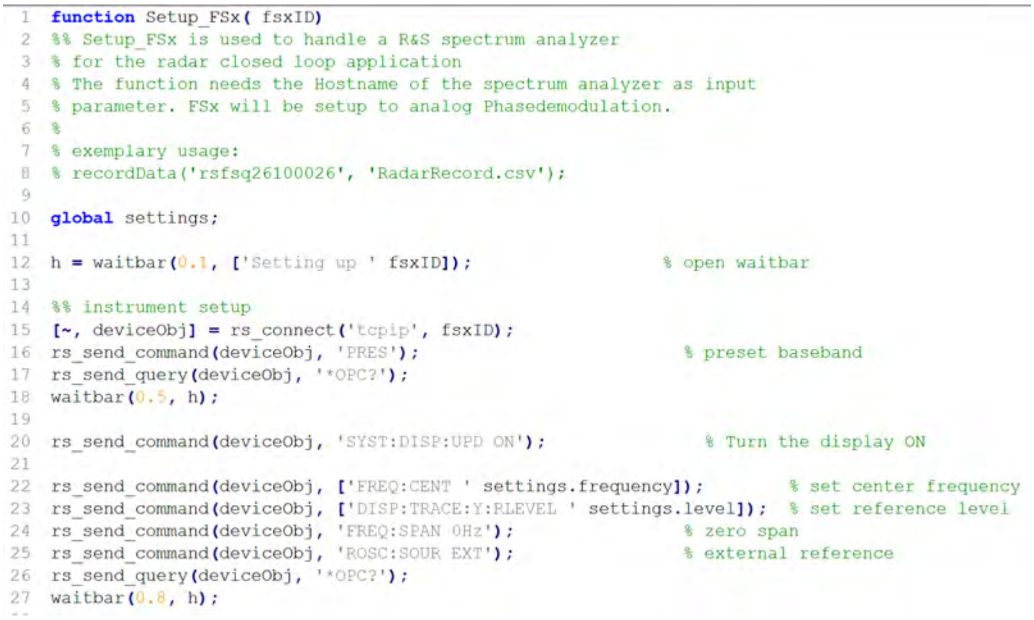
```
26  rs_send_command(deviceObj, ['SOURce:FREQuency ' settings.frequency]);   % set frequency
27  rs_send_command(deviceObj, ['SOURce:POW ' settings.level]);              % set level
28
29  rs_send_command(deviceObj, 'SOURce:BB:DM:CODing OFF');             % Coding OFF
30  rs_send_command(deviceObj, 'SOURce:BB:DM:FORMat BPSK' );            % BPSK modulation
31
32  rs_send_command(deviceObj, 'SOURce:BB:DM:SOURce PATTern');         % Data Source is pattern
33  rs_send_command(deviceObj, ['SOURce:BB:DM:PATTern #B' ...          % copying pattern to SMBV
34      , pattern, ',' num2str(length(pattern))]);
35  rs_send_command(deviceObj, 'SOURce:BB:DM:Filt:TYPE RECT');         % rectangular pulse shaping
36
37  rs_send_command(deviceObj, 'BB:DM:TRIG:OUTP1:MODE PATT');          % Create Marker 1 Output for FSx
38  rs_send_command(deviceObj, ...
39      ['BB:DM:TRIG:OUTP1:PATT #B0000000000001,13']);
40
41  rs_send_command(deviceObj, 'BB:DM:TRIG:OUTP2:MODE PATT');          % Create Marker 2 Output for FSx
42  rs_send_command(deviceObj, ...
43      ['BB:DM:TRIG:OUTP2:PATT #B0000000000001,13']);
44
45  waitbar(0.7, h);
46  rs_send_command(deviceObj, 'SOURce:BB:DM:STATe ON');               % switch BB signal ON
47
48  rs_send_command(deviceObj, 'OUTPut ON');                           % switch RF signal ON
49  rs_send_query(deviceObj, '*OPC?');
50  clear  deviceObj;
51  close(h);
52  end
```

Fig. 5-8: Generator setup for the closed loop radar application, part 2 of 2

Lines 26 to 49 show the SCPI commands in lite grey text color as being sent to the instrument. They are described in the operating manual of the instrument. Lines 37 to 44 show the programming of the Markers output as available on the "User1" and "User2" output connectors. Both outputs provide the same signal, one is intended to trigger the test receiver, the other can be used to trigger an oscilloscope in order to observe the baseband IQ-signals of the generator.

5.1.3.2 Setup_FSx

The receiver side is programmed in a similar way using the MATLAB Toolkit [13]. Fig. 5-9 and 5-10 show the complete listing of the MATLAB code.



```
 1  function Setup_FSx( fsxID)
 2  %% Setup_FSx is used to handle a R&S spectrum analyzer
 3  % for the radar closed loop application
 4  % The function needs the Hostname of the spectrum analyzer as input
 5  % parameter. FSx will be setup to analog Phasedemodulation.
 6  %
 7  % exemplary usage:
 8  % recordData('rsfsq26100026', 'RadarRecord.csv');
 9
10  global settings;
11
12  h = waitbar(0.1, ['Setting up ' fsxID]);                    % open waitbar
13
14  %% instrument setup
15  [~, deviceObj] = rs_connect('tcpip', fsxID);
16  rs_send_command(deviceObj, 'PRES');                           % preset baseband
17  rs_send_query(deviceObj, '*OPC?');
18  waitbar(0.5, h);
19
20  rs_send_command(deviceObj, 'SYST:DISP:UPD ON');                 % Turn the display ON
21
22  rs_send_command(deviceObj, ['FREQ:CENT ' settings.frequency]);          % set center frequency
23  rs_send_command(deviceObj, ['DISP:TRACE:Y:RLEVEL ' settings.level]);  % set reference level
24  rs_send_command(deviceObj, 'FREQ:SPAN 0Hz');                  % zero span
25  rs_send_command(deviceObj, 'ROSC:SOUR EXT');                  % external reference
26  rs_send_query(deviceObj, '*OPC?');
27  waitbar(0.8, h);
```

Fig. 5-9: Receiver setup for the closed loop radar application, part 1 of 2

```
29  rs_send_command(deviceObj, 'ADEM ON');                        % enter analog demodulation mode
30  rs_send_command(deviceObj, 'CALC:FEED ''XTIM:PM''');          % PM mode
31  rs_send_command(deviceObj, 'ADEM:MTIMe 52us');               % measurement time is 60us
32  rs_send_command(deviceObj, 'ADEMod1:AF:COUPling DC');         % AF coupling DC
33  % number of data points in measurement :
34  rs_send_command(deviceObj, ['SWEEP:POINTS ' num2str(settings.numPoints)]);
35
36  rs_send_command(deviceObj, 'INIT:CONT ON');                   % sweep mode to collect samples
37  rs_send_query(deviceObj, '*OPC?');
38
39  rs_send_command(deviceObj, 'INIT:CONT OFF');                  % single sweep mode
40
41  rs_send_command(deviceObj, 'TRIG:SOUR EXT');                  % trigger Setup
42  rs_send_query(deviceObj, '*OPC?');
43
44  clear  deviceObj;
45
46  close(h);                                                     % close waitbar
47
```

Fig. 5-10: Receiver setup for the closed loop radar application, part 2 of 2

Even though [13] is called "Toolkit ... *for Signal Generators*", the basic instrument programming calls "rs_send_command" and "rs_send_query" can be used for other instruments also. Therefore the same instrument programming interface is also used for the receiving side, i.e. for instruments from the FSxx families as shown in the listing in Fig. 5-9 and 5-10. In lines 22 and 23 frequency and level are set according to the appropriate global parameters. The instrument operation is performed based upon the light grey SCPI commands. Based on line 41 the instrument will perform a trace on the next upcoming trigger signal appearing at the trigger input connector of the device. The function ends with one valid demodulated signal in the display memory which is read by function "TraceFile.m".

5.1.3.3 TraceFile

This function reads the trace data from the receiver and stores it in a MATLAB compatible format. The filename is specified via an input parameter of the function, refer to Fig. 5-5 and 5-6, line 18 and line 41.

5.1.3.4 GetBits

This function finally retrieves the bit pattern using the IQ-constellation diagram and plots the results into a MATLAB stairs diagram. The function is called in the main program, Fig. 5-6, line 46. The function returns the resulting bit pattern, which can be used to calculate a jamming code or any other sequence to be retransmitted to the flying object.

5.2 References

[1] Pozar David M., 2005, Microwave Engineering, third edition, WILEY, ISBN 978-0-471-44878-5

[2] Merrill I. Skolnik (Editor in Chief), 1990, radar Handbook, Second Edition McGraw-Hill, ISBN 0-07-057913-X

[3] Heuel, 2013, "Radar Waveforms for A&D and Automotive Radar", White Paper, R&S Application Note Nr. 239, available from http://www. rohde-schwarz.com/appnote/1MA239

[4] Naseef, Minihold, Bednorz, 2013, "Testing S-Parameters on Pulsed Radar Power Amplifier Modules", R&S Application Note Nr. 126, available from http://www.rohde-schwarz.com/appnote/1MA126, video available for download

[5] Bues, Minihold, 2012, "Overview of Tests on Radar Systems and Components", R&S Application Note Nr. 127, available from http://www.rohde-schwarz.com/appnote/1MA127

[6] Minihold, Bues, 2012, "Introduction to Radar System and Component Tests", R&S Application Note Nr. 207, available from http://www.rohde-schwarz.com/appnote/1MA207

[7] R&S®ZVA network analyzer basics part 4: Amplifier gain and matching, YouTube Video, 21.2.2012, available from https://www.youtube.com/watch?v=6B7tS-Mj3BU , retrieved at July 24th, 2014

[8] Mahafza, Bassem R., 2005, Radar Systems Analysis and Design Using MATLAB, Second Edition, Chapman & Hall, ISBN 1-58488-532-7

[9] Radar Tutorial #1: Demonstrating radar principles using a vector network analyzer, YouTube Video Series / 1 of 5 items, 12.6.2014, available from https://www.youtube.com/watch?v=ISkPcv7C_Uk&list=UUoJfm2BU72j699FH3IUr3mg, retrieved at July 24th, 2014

[10] Tröster-Schmid C.,2013, Simulating Fading with R&S® Vector Signal Generators, R&S Application Note Nr. 1GP99, available from http://www.rohde-schwarz.com/appnote/1GP99

[11] Heuel, Schmitt, 2014, Rohde & Schwarz webinar: Analysis of FMCW radar signals in automotive applications, available from https://www.youtube.com/watch?v=8qaCSQ83ZyU&list=UUoJfm2BU72j699FH3IUr3mg

[12] R&S® Pulse Sequencer Software, Rohde & Schwarz, 2012, available from http://www.rohde-schwarz.com/en/software/smbvk6/

[13] T.Röder, C.Tröster, 2013, MATLAB Toolkit for R&S® Signal Generators, R&S Application Note Nr. 1GP60, available from http://www.rohde-schwarz.com/appnote/1GP60

[14] Wendler, Wolfgang, 2014, Seamless realtime analysis of frequency hopping with the R&S®FSW, R&S NEWS 210/14, pg. 28, available from http://cdn.rohde-

schwarz.com/pws/dl_downloads/dl_common_library/dl_news_from_rs/magazin/NEWS_210_150dpi_english.pdf

[15] R&S®SMW200A, June 2014, Vector Signal Generator, Specifications 02.02, available from http://www.rohde-schwarz.com/en/ , search tag "SMW200A", Brochures and Data Sheets

[16] Bues, 2014, Generation and Analysis of Pulsed Radar Signals, R&S Application Note Nr. 1MA234, available from October 2014, http://www.rohde-schwarz.com/appnote/1MA234

[17] Wikipedia, "Barker code", available on "http://en.wikipedia.org/wiki/Barker_code", retrieved at August 5th, 2014

[18] Dr. Steffen Heuel, 11.2014, "Real-time Radar Target Generation", R&S Application Note Nr. 1MA256, available from http://www.rohde-schwarz.com/appnote/1MA256

[19] R&S® FSW Signal and Spectrum Analyzer Specifications, Rohde & Schwarz, 2014, available from http://cdn.rohde-schwarz.com/pws/dl_downloads/dl_common_library/dl_brochures_and_datasheets/pdf_1/FSW_dat-sw_en_5214-5984-22_v1000.pdf, retrieved at August 14th, 2014

6 Rohde & Schwarz

Rohde & Schwarz is an independent group of companies specializing in electronics. It is a leading supplier of solutions in the fields of test and measurement, broadcasting, radiomonitoring and radiolocation, as well as secure communications. Established more than 80 years ago, Rohde & Schwarz has a global presence and a dedicated service network in over 70 countries. Company headquarters are in Munich, Germany.

Sustainable product design

- Environmental compatibility and eco-footprint
- Energy efficiency and low emissions
- Longevity and optimized total cost of ownership

Certified Quality Management
ISO 9001

Certified Environmental Management
ISO 14001

Regional contact

- Europe, Africa, Middle East
  Phone +49 89 4129 12345
  customersupport@rohde-schwarz.com
- North America
  Phone 1-888-TEST-RSA (1-888-837-8772)
  customer.support@rsa.rohde-schwarz.com
- Latin America
  Phone +1-410-910-7988
  customersupport.la@rohde-schwarz.com
- Asia/Pacific
  Phone +65 65 13 04 88
  customersupport.asia@rohde-schwarz.com
- China
  Phone +86-800-810-8228 / +86-400-650-5896
  customersupport.china@rohde-schwarz.com

Headquarters

Rohde & Schwarz GmbH & Co. KG

Mühldorfstraße 15 | D - 81671 München

+ 49 89 4129 - 0 | Fax + 49 89 4129 – 13777

www.rohde-schwarz.com