CONFIGURATION OF OFDM DEMODULATOR

Version 1.0.1

Date: October 13, 2023

Imprint

Configuration of OFDM Demodulator

PROCITEC GmbH

Rastatter Str. 41

D-75179 Pforzheim

www.procitec.de

info@procitec.de

Printed: 13-10-2023

Table of Contents

1 Introduction

Orthogonal Frequency Division Multiplex (OFDM) is widely used in today's civil and military transmission techniques. The use of the technique is still continuing to spread worldwide. Because of this, the ability to gather electronic intelligence on them is becoming more and more important. Thus, PROCITEC developed a **universal OFDM demodulator** for production of OFDM signals.

1.1 Purpose of the document

This document details the configuration syntax specific to the software OFDM demodulator implemented in the APC. Throughout of this document it is assumed, that the reader is familiar with concepts of demodulation of an OFDM waveform.

1.2 Abbreviations

Abbreviation	Meaning
APC	Automatic Production Channel (a product of PROCITEC)
DC	Direct Current
DFT	Discrete Fourier Transform
DQPSK	Differential Quadrature Phase Shift Keying
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
MSB	Most Significant Bit
N_{FFT}	FFT length for the demodulation of an OFDM signal
N_G	Length of the guard interval in samples
N_S	OFDM symbol length in samples
N_U	Length of the useful part of an OFDM symbol in samples
OFDM	Orthogonal Frequency Division Multiplex
QAM	Quadrature Amplitude Modulation
T_G	Duration of the guard interval in seconds
T_S	OFDM symbol duration in seconds
T_U	Duration of the useful part of an OFDM symbol in seconds

1.3 Definitions

The following table contains a few important definitions

OFDM symbol	Time domain: All samples contained in an OFDM symbol (samples in guard interval + useful samples)
	Frequency domain: Result of an FFT (DFT) calculation (one complete clock cycle)
Carrier symbol, cell	Symbol of a carrier at the time of the clock cycle
Utilised channel/carrier	Channel/carrier used for transmission (data and pilots).
Pilot	Umbrella term for: pilot tone, pilot carrier, pilot symbol or distributed pilot cells
Pilot tone	Single carrier in the OFDM frequency grid with constant amplitude and constant phase (amplitude and phase are known)
Pilot carrier	Single carrier in the OFDM frequency grid with variable amplitude and variable phase (amplitude and phase at certain points in time are known)
Pilot symbol	Complete OFDM symbol where amplitude and phase of all carriers are known. (Pilot symbols are not used for data transmission, but mainly for equalisation of channels.)
Distributed pilot cells	Known carrier symbols (amplitude and phase), which are distributed over time and carriers (OFDM symbol) according to predefined patterns.
Channel zero (carrier zero)	Carrier of the OFDM signal, which defines the DC-part of the according base-band signal in the transmitter.

2 Configuration of OFDM demodulator

The configuration for the OFDM demodulator is stored in a self-contained XML-formatted file. This file represents part of the modem definition (*.ver file) for use in the APC.

Subsections of the file under the root node `Configuration` are presented below. The description concentrates on meaning and grouping of used data fields. The exact syntax can be seen in examples in section 4.

2.1 Modem

This subsection contains basic parameters of OFDM waveform.

Name	Parameter	Default	Type	Information
Fs (OFDM sampling rate)	dFs	(empty)	Double	OFDM sampling rate
Ts	dTs	(empty)	Double	OFDM symbol duration in seconds
Tu	dTu	(empty)	Double	Duration of the useful part of an OFDM symbol in seconds
F	dF	(empty)	Integer	Number of symbols in time direction specified for frame definition. Usually equals to 1. Needed for matrix size specification in `<Data>` section.
F0	dChannel0Freq	0.0Hz	Double	Optional. Specifies the relative frequency offset to 0th channel. Needed only for complex, multi-config configurations.
Nu	dN	(empty)	Double	Optional and redundant. Duration of the useful part of an OFDM symbols in samples Automatically recomputed according to Tu and Fs.

2.1.1 Example

```
<Modem>
   <dTs>0.0225</dTs>
   <dTu>0.01777777777777777</dTu>
   <dFs>3600</dFs>
   <dF>1</dF>
   <dN>64</dN>
</Modem>
```

2.2 Pilots

This subsection is the container for specification of pilots. Pilot carriers and pilot cells are supported. Please see example (2.2.3) for syntax details.

2.2.1 Pilot cells

Pilot cells can be specified in form of matrix of size $N_u \times F$. Cells not used for pilots must contain the value of 0. The I/Q value for the pilot cell is to be specified as a complex number in the form $x + yi$.

2.2.2 Pilot carrier

To specify a pilot carrier one needs its I/Q value and the assigned channel number. Carriers are grouped under node named `PilotCarriers`, each carrier gets its own subnode `Carrier`.

The channel number is stored under `dChIdx`. Channels are numbered in the convention with 0th channel in the middle. Assuming DFT length of N, channels receive numbers in range

$$\left[-l\frac{N}{2}J, \ldots, 0, \ldots, l\frac{N-1}{2}J \right]$$

The I/Q value corresponding to the magnitude and phase of the carrier as sampled at each OFDM symbol is to be specified as a complex number in the form $x + yi$.

In case symbol differentiation before equaliser (see 2.5.3) is in use, then all phase values for pilot carriers are interpreted as differential.

2.2.3 Example

Pilot definition for $N_u = 8$ and $F = 15$ could be entered as follows:

```
<Pilot>
  <PilotCells>
    <Cell>
      <dIQ>[0 0        0         0 0 0 0  0 0 0  0 0 0 0 0;
            0 1.2+0.3i 0         0 0 0 0  0 0 0  0 0 0 0 0;
            0 0        0         0 0 0 1  0 0 0 -1 0 0 0 0;
            0 0        0         0 0 0 0  0 0 0  0 0 0 0 0;
            0 0        0.2+2.1i  0 0 0 0  0 0 0  0 0 0 0 0;
            0 0        0         0 0 0 0 -1 0 0  1 0 0 0 0;
            0 0        0         0 0 0 0  0 0 0  0 0 0 0 0;
            0 0        0         0 0 0 0  0 0 0  0 0 0 0 0;]</dIQ>
    </Cell>
  </PilotCells>
  <PilotCarriers>
    <Carrier>
      <dChIdx>-2</dChIdx>
      <dIQ>-1.4142135623730951-1.4142135623730951i</dIQ>
    </Carrier>
  </PilotCarriers>
</Pilot>
```

2.3 Data

This subsection specifies content of each OFDM cell. A constellation of I/Q points is defined for each channel (frequency direction) and each symbol (time direction). We specify a list of constellations by their names (labels). The assignment of constellations is represented as matrix of indexes to the list of labels

In case symbol differentiation before equaliser (see 2.5.3) is in use, then all constellations modified by scale factors (see 2.3.2) and phase offsets (see 2.3.3) are interpreted as differential.

2.3.1 Constellations

This subsection specifies I/Q constellations assigned to each cell. The constellations are listed by name, where the first one (referenced by 0) is to be empty (-). The sequence number of named constellation (starting with 0) is used in the following matrix of size $N_u \times F$. The definition of points creating a constellation is described in detail in section 2.4

2.3.2 Scale factors

There are usually only a few constellations used in an OFDM waveform. With the definition of scale factors (`ScaleFactors`) for each cell it is possible to reuse predefined constellations scaled to the needs of the particular type of signal.

It is common practice to define constellations on a integer grid of coordinates and then scale each constellation by a factor providing equal energy distribution over all used I/Q constellations.

2.3.3 Phase offsets

Predefined constellations can be not only scaled, but also rotated. The matrix containing phase offsets (in radians) (`PhaseOffsets`) defines rotation of constellation assigned for given cell.

Assigning different rotations to predefined constellations on different channels is sometimes used to reduce usually high crest factor of OFDM signal.

2.3.4 Example

Let's assume $F=1$ and $N_u=8$

```
<Data>
  <Constellations>
    <caConstellationName>
      <item>-</item>
      <item>QAM64</item>
      <item>QPSK</item>
    </caConstellationName>
    <dIdx>[0; 0; 2; 1; 1; 2; 0; 0;]</dIdx>
  </Constellations>
  <ScaleFactors>
    <dIQ>[0; 0; 0.5; 1.2; 1.2; 0.5; 0; 0;]</dIQ>
  </ScaleFactors>
  <PhaseOffsets>
    <dIQ>[0; 0; 0; 0.785398163; 0.785398163; 0; 0; 0;]</dIQ>
  </PhaseOffsets>
</Data>
```

In the definition above we have 4 unused channels: two on the right and two on the left. Two channels in the centre use constellation labelled as QPSK scaled by linear factor of 1.2 and rotated counter clockwise by $45°$ ($\pi/4 \approx 0.785$). Channels -2 and 1 contain constellation labelled as QAM64 scaled by 0.5 and not rotated.

2.4 Constellations

The definition of points creating a constellation is stored under node `Constellations`. The contents of this node reuses definitions imported from other software package and therefore may break some conventions used in the rest of the OFDM configuration.

Under the `Constellations` node there is `contree` node with attribute `name="Constellations"`. The node `Constellations/contree/constel` stores in the attribute `name` the label of the constellation is referenced in cell definition (see 2.3.1).

The node `Constellations/contree/constel/points` contains white space separated I and Q values for all points. Real numbers composing I and Q coordinates (interpreted usually as a complex number) are specified interleaved one after another. The node `Constellations/contree/constel/numberings/numbering` with attribute `name="Base 0"` describes mapping from I/Q values to symbol number (or bit sequence). The bit sequence is interpreted as MSB first, e.g. "2" is "10" in binary (for two bits per symbol), where "1" is first and "0" is second in the sequence of bits.

In case more than one constellation is necessary, further nodes `Constellations/contree/constel` with unique labels have to be enclosed.

2.4.1 Example

```
<contree name="Constellations" >
  <constel name="QPSK">
    <points>
      1 1  -1 1  -1 -1  1 -1
    </points>
    <numberings>
      <numbering name="Base 0">
        0 1 2 3
      </numbering>
    </numberings>
  </constel>
  <constel name="QAM16">
    <!-- ... points and numberings ... -->
  </constel>
</contree>
```

2.5 Equaliser

Parameter specified in this section (`Equaliser`) control processing of I/Q values after DFT step in OFDM demodulator.

2.5.1 Channel phase increment

Field name: `dChannelPhaseIncrement`

Default: 0.0

Channel phase increment specifies phase accumulatively added to the demodulated I/Q value for each symbol clock. The most common use is to compensate symbol-to-symbol rotation for waveforms generated as multichannel PSK which can also be interpreted as OFDM signals.

2.5.2 Channel offset correction

Field name: `lChannelOffsetCorrection`

Default: false

If set to true, then automatic detection and correction of integer channel offset is performed. In case number of transmitted channels is not identical with information specified for data (2.3) and pilots (2.2) the automatic detection may fail.

2.5.3 Differential mode

Field name: `lDiferentiateSymbols`

Default: false

If set to true then I/Q values directly after DFT are multiplied with conjugated complex value of previous symbol in time direction. This operation is equivalent to computation of differential phase for PSK constellations.

2.5.4 Differential decision

Field name: `lDiferentiateAfterEQ`

Default: false

If set to true then the differential computation as described above is performed as the last step before I/Q to symbol mapping (symbol decision). The differentiation is computed only once.

2.5.5 Pilot tone equalisation

Field name: `lPilotToneEqualisation`

Default: false

If set to true then pilot carriers as specified in 2.2.2 are used as the first step in phase/magnitude equalisation.

2.5.6 Automatic gain control (AGC)

Field name: `lAGC`

Default: false

Automatic gain control is turned on if the parameter is set to true. The correction is based on average energy of constellations specified in data section under the assumption of uniform symbol distribution.

2.5.7 AGC filter length

Field name: `lAGCFilterLength`

Default: 0

Specify length of moving average filter used in AGC. The length is given in the number of symbols in time direction that are to be considered. The filter length should be odd. If the specified value is 0 (default), then the filter length is computed automatically based on the constellation size.

2.5.8 Automatic frequency and phase control

Field name: `lAFCAPC`

Default: false

If set to true, then blind, decision directed combined frequency and phase correction is performed (see also 2.5.9 and 2.5.10).

2.5.9 Feedback ratio AFC

Field name: `dAFCAPCDPhiAlpha`

Default: 0.1

This parameter specifies feedback ratio for AFC (frequency control loop). The value is in range between 0.0 and 1.0.

2.5.10 Feedback ratio APC

Field name: `dAFCAPCPhiAlpha`

Default: 0.1

This parameter specifies feedback ratio for APC (phase control loop). The value is in range between 0.0 and 1.0.

2.5.11 Block length for AFC/APC initialisation

Field name: `dAFCAPCFirstBlockLength`

Default: 16

This block length specified the number of symbols at the beginning of energy bust that are used for the determination of initial state of APC/AFC control loop.

2.5.12 Frame sequence

Modems that require reconfiguration of I/Q constellations at runtime can be often described using sequence of preconfigured sets of channel assignments. The definition how constellations are allocated in channels is specified in the `Data` section (see 2.3). Each column in the $N_u \times F$ matrix is one configuration set. It means we have F configuration sets. The frame sequence stored under `FrameSequence` allows navigation between predefined sets of channel assignments.

The specification consists of a list of items (`item`), where each item contains length (`dLength`), index (`dSymbolInFrame`) of the column in `Data` section (zero based), and number (`dNext`) of the next item within the frame sequence (zero based). For the exact syntax see the following example (2.5.12.1).

2.5.12.1 Example

The frame sequence included below specifies that first 47 symbols are to be demodulated with constellations as in column 0. The successor is at index 1 on the list, where constellations from column 1 are to be used. The demodulator stays up to the end of the energy burst using column 1. The length of 0 is to be interpreted as infinity. If the length were specified as positive number then the functionality would not change because the item points to itself as its successor.

```
<FrameSequence>
  <item>
    <dLength>47</dLength>
    <dSymbolInFrame>0</dSymbolInFrame>
    <dNext>1</dNext>
  </item>
  <item>
    <dLength>0</dLength>
    <dSymbolInFrame>1</dSymbolInFrame>
    <dNext>1</dNext>
  </item>
</FrameSequence>
```

2.6 Synchroniser

Settings for timing and frequency synchronisation and the sampling rate correction can be found here. The configuration is stored under the node `Synchroniser`.

2.6.1 Processing block size

Field name: `dBlockSize`

Default: 20

Synchroniser block size in symbols for the calculation of timing and frequency offset. The larger the value, the more symbols are averaged for the running estimation of sampling position and coarse frequency offset.

2.6.2 Sampling frequency error

Field name: `dFsMeaDurationSize`

Default: 0.0

This setting specifies the amount of signal in seconds used for initial estimation of sampling frequency error. If the value is set to 0 then an equivalent of 500 symbols (or the whole energy burst, if shorter) is taken for the measurement.

2.6.3 Skip symbols

Field name: `dSkipSymbols`

Default: 0

This parameter specifies the number of symbols (integer) which should be skipped at the beginning of an energy burst. It is necessary to skip signal parts that would interfere with initial OFDM synchronisation and demodulation.

2.6.4 Signal length

Field name: `dSignalLength`

Default: 0.0

Signal length needs to be specified in case waveform changes after given duration and other OFDM configuration needs to be used after that. The value is given in seconds. If the duration is omitted or set to 0, then infinite length is assumed.

3 Reparametrization at runtime

There are use cases, where OFDM signal parameters change over time. The demodulator supports two basic modes for modification of waveform parameters. One of them can be realised when OFDM base parameters, such as symbol rate and channel distance do not change, only contents of cells is modified in predefined pattern. The other one is needed when all parameters but the OFDM sampling rate does change. It is assumed that the processing time counts from the beginning of energy burst (energy detection).

Both types of parameter reconfiguration are orthogonal to each other and can be combined.

3.1 Reconfiguration of channel assignments

The reconfiguration of cells is based on definition of multiple symbols in time direction ($F \geq 1$). To make use of this definition one must also specify which column of the $N_u \times F$ matrix is to be used for which symbol as time passes. The specification of data constellations has been described in 2.3. With the use of frame sequence (see 2.5.12) it can be described how long should a specific column be used and which column comes next after predefined number of symbols. It is, for example, possible to say: process data as specified in column 0 for the first 5 symbols, then use the column 1 for next 10 symbols and finally use column 0 again for the rest of the energy burst.

3.2 Modification of OFDM configuration at runtime

The modem description stores multiple configurations. Each of them has its signal length specified (see 2.6.4). After processing of the given amount of seconds the demodulator reconfigures itself and switches to next configuration in the order of appearance in the file.

As already noted the OFDM sampling frequency must not change between work modes. The offset between work modes for the frequency of channel 0th can be explicitly specified with `dChannel0Freq` (see 2.1)

The modem configuration file needs to contain multiple sub-configurations of OFDM waveform. In the single configuration case the nodes `Modem`, `Data`, … come directly under root node `Configuration`. To allow multiple configurations one additional node `SubConfig` has been inserted to facilitate easy separation of OFDM waveform descriptions. See the following example for details.

3.2.1 Example

The following example contains only XML nodes relevant for the described functionality

```
<Configuration>
  <SubConfig>   <!-- first configuration -->
    <Modem></Modem>
    <Data></Data>
    <Constellations></Constellations>
    <Equaliser></Equaliser>
    <Synchroniser>
      <!-- first config use for first 3.5 seconds of signal -->
      <dSignalLength>3.50</dSignalLength>
    </Synchroniser>
  </SubConfig>
  <SubConfig>   <!-- second configuration -->
    <Modem></Modem>
    <Data></Data>
    <Constellations></Constellations>
```

```
    <Equaliser></Equaliser>
    <Synchroniser>
      <!-- use this configuration for the rest of the energy burst -->
      <dSignalLength>0</dSignalLength>
    </Synchroniser>
  </SubConfig>
</Configuration>
```

4 Examples

This section contains examples of complete and working OFDM configurations

4.1 Differential PSK with pilot tone as OFDM

```
<?xml version="1.0" encoding="UTF-8"?>
<!--This is generated-->
<Configuration>
  <Modem>
    <dTs>0.0225</dTs>
    <dTu>0.01777777777777778</dTu>
    <dFs>3600</dFs>
    <dF>1</dF>
    <dN>64</dN>
  </Modem>
  <Pilot>
    <PilotCarriers>
      <Carrier>
       <dChIdx>-24</dChIdx>
       <dIQ>-1.4142135623730951-1.4142135623730951i</dIQ>
      </Carrier>
    </PilotCarriers>
  </Pilot>
  <Data>
    <Constellations>
      <caConstellationName>
        <item>-</item>
        <item>DPSK4B</item>
      </caConstellationName>

<dIdx>0;0;0;0;0;0;0;0;0;0;0;0;0;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;
1;1;1;1;1;1;1;1;1;1;1;1;1;0;0;0;0;0;0;0;0;0;0;0;0]</dIdx>
    </Constellations>
    <ScaleFactors>

<dIQ>0;0;0;0;0;0;0;0;0;0;0;0;0;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1
;1;1;1;1;1;1;1;1;1;1;1;1;0;0;0;0;0;0;0;0;0;0;0;0]</dIQ>
    </ScaleFactors>
  </Data>
  <Constellations>
    <contree name="Constellations">
      <constel name="DPSK4B">
        <points>0.70710677 0.70710677 -0.70710677 0.70710677 -0.70710677 -
0.70710677 0.70710677 -0.70710677</points>
        <numberings>
          <numbering name="Base 0">0 2 3 1</numbering>
        </numberings>
      </constel>
    </contree>
  </Constellations>
```

```
   <Equaliser>
     <lChannelOffsetCorrection>true</lChannelOffsetCorrection>
     <lTgPhaseIncrementCorrection>false</lTgPhaseIncrementCorrection>
     <lFrameSyncTracking>true</lFrameSyncTracking>
     <lDifferentiateSymbols>true</lDifferentiateSymbols>
     <lDifferentiateAfterEQ>false</lDifferentiateAfterEQ>
     <lPilotToneEqualisation>false</lPilotToneEqualisation>
     <lAGC>true</lAGC>
     <lAFCAPC>true</lAFCAPC>
     <dAGCFilterLength>0</dAGCFilterLength>
     <dAFCAPCDPhiAlpha>0.1</dAFCAPCDPhiAlpha>
     <dAFCAPCPhiAlpha>0.1</dAFCAPCPhiAlpha>
     <dAFCAPCFirstBlockLength>16</dAFCAPCFirstBlockLength>
   </Equaliser>
   <Synchroniser>
     <dBlocksize>20</dBlocksize>
     <lFsErrorEstimation>true</lFsErrorEstimation>
     <dFsMeaDuration>11.25</dFsMeaDuration>
     <dSkipSymbols>21</dSkipSymbols>
   </Synchroniser>
</Configuration>
```

4.2 Multimode OFDM

This configuration combines both mode switching techniques described in 3

```
<?xml version="1.0" encoding="UTF-8"?>
<!--This is generated-->
<Configuration>
  <SubConfig>
    <Modem>
      <dTs>0.01333333333333333</dTs>
      <dTu>0.008888888888888889</dTu>
      <dFs>3600</dFs>
      <dF>2</dF>
      <dN>32</dN>
      <dChannel0Freq>56.25</dChannel0Freq>
    </Modem>
    <Data>
      <Constellations>
        <caConstellationName>
          <item>-</item>
          <item>DPSK2B</item>
          <item>DPSK4B</item>
        </caConstellationName>
        <dIdx>[0 0;
0 0;
0 0;
0 0;
0 0;
0 0;
0 0;
0 0;
1 2;
1 2;
1 2;
1 2;
1 2;
1 2;
```

```
1 2;
1 2;
1 2;
1 2;
1 2;
1 2;
1 2;
1 2;
1 2;
0 0;
0 0;
0 0;
0 0;
0 0;
0 0;
0 0;
0 0;]</dIdx>
        </Constellations>
        <ScaleFactors>
          <dIQ>[0 0;
0 0;
0 0;
0 0;
0 0;
0 0;
0 0;
0 0;
1 1;
1 1;
1 1;
1 1;
1 1;
1 1;
1 1;
1 1;
1 1;
1 1;
1 1;
1 1;
1 1;
1 1;
1 1;
0 0;
0 0;
0 0;
0 0;
0 0;
0 0;
0 0;
0 0;]</dIQ>
      </ScaleFactors>
    </Data>
    <Constellations>
      <contree name="Constellations">
        <constel name="DPSK2B">
          <points>0 1 0 -1</points>
          <numberings>
            <numbering name="Base 0">0 1</numbering>
```

```
        </numberings>
      </constel>
      <constel name="DPSK4B">
        <points>0.70710677 0.70710677 -0.70710677 0.70710677 -0.70710677 -
0.70710677 0.70710677 -0.70710677</points>
        <numberings>
         <numbering name="Base 0">0 1 3 2</numbering>
        </numberings>
      </constel>
    </contree>
  </Constellations>
  <Equaliser>
  <lTgPhaseIncrementCorrection>false</lTgPhaseIncrementCorrection>
  <lFrameSyncTracking>true</lFrameSyncTracking>
  <lDifferentiateSymbols>true</lDifferentiateSymbols>
  <lDifferentiateAfterEQ>false</lDifferentiateAfterEQ>
  <lPilotToneEqualisation>false</lPilotToneEqualisation>
  <lAGC>true</lAGC>
  <lAFCAPC>true</lAFCAPC>
  <dAGCFilterLength>9</dAGCFilterLength>
  <dAFCAPCDPhiAlpha>0.1</dAFCAPCDPhiAlpha>
  <dAFCAPCPhiAlpha>0.1</dAFCAPCPhiAlpha>
  <dAFCAPCFirstBlockLength>16</dAFCAPCFirstBlockLength>
  <FrameSequence>
    <item>
     <dLength>47</dLength>
     <dSymbolInFrame>0</dSymbolInFrame>
     <dNext>1</dNext>
    </item>
    <item>
      <dLength>0</dLength>
      <dSymbolInFrame>1</dSymbolInFrame>
      <dNext>1</dNext>
    </item>
    </FrameSequence>
  </Equaliser>
  <Synchroniser>
    <dBlocksize>20</dBlocksize>
    <lFsErrorEstimation>true</lFsErrorEstimation>
    <dFsMeaDuration>6.666666666666667</dFsMeaDuration>
    <lCfoTrack>true</lCfoTrack>
    <dSignalLength>1.513333333333333</dSignalLength>
    <dSkipSymbols>32</dSkipSymbols>
  </Synchroniser>
  </SubConfig>
  <SubConfig>
    <Modem>
      <dTs>0.0225</dTs>
      <dTu>0.01777777777777778</dTu>
      <dFs>3600</dFs>
      <dF>1</dF>
      <dN>64</dN>
      <dChannel0Freq>56.25</dChannel0Freq>
    </Modem>
    <Data>
      <Constellations>
        <caConstellationName>
          <item>-</item>
          <item>DPSK4B</item>
        </caConstellationName>
```

```
<dIdx>[0;0;0;0;0;0;0;0;0;0;0;0;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1
;1;1;1;1;1;1;1;1;1;1;1;0;0;0;0;0;0;0;0;0;0;0;0;0]</dIdx>
            </Constellations>
            <ScaleFactors>

<dIQ>[0;0;0;0;0;0;0;0;0;0;0;0;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;
1;1;1;1;1;1;1;1;1;1;1;1;0;0;0;0;0;0;0;0;0;0;0;0;0]</dIQ>
            </ScaleFactors>
        </Data>
        <Constellations>
          <contree name="Constellations">
            <constel name="DPSK4B">
              <points>0.70710677 0.70710677 -0.70710677 0.70710677 -0.70710677 -
0.70710677 0.70710677 -0.70710677</points>
            <numberings>
              <numbering name="Base 0">0 1 3 2</numbering>
            </numberings>
          </constel>
        </contree>
      </Constellations>
      <Equaliser>
        <dChannelPhaseIncrement>1.57079632679489656</dChannelPhaseIncrement>
        <lTgPhaseIncrementCorrection>false</lTgPhaseIncrementCorrection>
        <lFrameSyncTracking>true</lFrameSyncTracking>
        <lDifferentiateSymbols>true</lDifferentiateSymbols>
        <lDifferentiateAfterEQ>false</lDifferentiateAfterEQ>
        <lPilotToneEqualisation>false</lPilotToneEqualisation>
        <lAGC>true</lAGC>
        <lAFCAPC>true</lAFCAPC>
        <dAGCFilterLength>9</dAGCFilterLength>
        <dAFCAPCDPhiAlpha>0.1</dAFCAPCDPhiAlpha>
        <dAFCAPCPhiAlpha>0.1</dAFCAPCPhiAlpha>
        <dAFCAPCFirstBlockLength>16</dAFCAPCFirstBlockLength>
      </Equaliser>
      <Synchroniser>
        <dBlocksize>20</dBlocksize>
        <lFsErrorEstimation>true</lFsErrorEstimation>
        <dFsMeaStartTime>0</dFsMeaStartTime>
        <dFsMeaDuration>11.25</dFsMeaDuration>
        <lCfoTrack>true</lCfoTrack>
      </Synchroniser>
    </SubConfig>
</Configuration>
```

5   Note on supported number formats

Vectors and matrices are serialised as text within XML node. Numbers, real and complex values are separated by white space. End of row is marked additionally with semicolon. Complex numbers are written as $x+yi$ where x and y are real numbers. There must be no white space between x, y and + and i. In case white space character occurs within complex number, the number will be misinterpreted as multiple numbers.