libns_pTx : a SBB for pTx pulses in VE12U

Contributors	Aurélien Massire[1], Redouane Jamil[2], Franck Mauconduit[2]

[1] *Siemens Healthinners, Saint-Denis, France*

[2] *Université Paris-Saclay, CEA, CNRS, BAOBAB, NeuroSpin, Gif-sur-Yvette, France*

Last update	February 11, 2022
Description	The SBB pTx helps to include external pTx pulses into sequences for Terra systems.
Contacts	Franck Mauconduit, CEA NeuroSpin

Contents

1 Introduction

The "libns_pTx" library contains a SBB class called "SeqBuildBlockExcitation_pTx" that can be used to easily insert a pTx pulse into any sequence. It uses the standard Siemens framework for pTx pulses that needs an ini file. This ini file contains the gradient and RF informations, as well as the pulse duration, the nominal flip angle and several other informations.

It was initially developped to be used in the PASTeUR package which currently contains different anatomical sequences such as 3D GRE, MPRAGE, MP2RAGE and SPACE sequences. This package is also available for VE12U systems on the Siemens C2P platform [Mauconduit F (5)].

1.1 Features of the "libns_pTx"

This library handles different features such as:

- The gradient waveforms defined in the ini file can be played in the gradient frame of reference (XYZ frame of reference) independently of the acquisition orientation. It makes it easier to tilt the volume of interest without requiring the change the ini file. A RotateGradientsPTX() method is available for this purpose.
- The flip angle can be changed by using the setFlipAngle() method. It will rescale the pTX RF pulse according the nominal flip angle available in the ini file.
- In addition, a setFlipAngleArray() mechanism is available as standard Siemens RF pulses are providing.
- The RFInfo is correctly computed to account for changes in flip angles from the nominal flip angle.
- The SBB can provide a pointer to the RF pulse object. It can be passed on other sequence SBBs to be handled like any other RF objects.
- There is no need to define a given PPD config file.

2 Using the SBB library

Library binaries: The binary files must be copied into IDEA in the following locations:

```
n4
├── x86
│   └── prod
│       └── lib
│           ├── libns_pTxLib.lib
│           └── libns_pTxdLib.lib
└── linux
    └── prod
        └── lib
            └── liblibns_pTx.a
```

Makefile: The following should be added into the sequence makefile. For linux compilation, the order in which the libraries are added is important. It should be kept as follow:

```
// Required libraries for pTx SBB
STATICLIBS   (libns_pTx)
LDLIBS       (AdjUIData)
LDLIBS       (libINIAccess)
```

header: An SBB object should be added in the class definition:

```
// Adding a SBB object
SeqBuildBlockExcitation_pTx m_SBBpTx;
```

Source file inclusions: The following set of codes can be used in order to integrate a pTx RF pulse.

```
// Adding a class contruct in the sequence class declaration list
MySeq::MySeq(void)
, ...
, m_SBBpTx (NULL, "PTX_PULSE_NAME") // here ini file is defined
, ...
```

In prepare method: Preparing the SBB in the prepare method of the sequence. In addition, use the duration and RFInfo informations.

```
    // In sequence prepare() method, prepare the SBB
    if ( !m_SBBpTx.prep(rMrProt, rSeqExpo ) )
    {return SEQU_ERROR;}

    // Get the SBB duration for timing purpose
    lSBBDuration = m_SBBpTx.getDurationPerRequest();

    // Get the SBB RFInfo for SAR management
    m_RFInfo += m_SBBpTx.getRFInfoPerRequest();
```

Adding the SBB into sequence calculatePTX() to prepare the pTx pulse into the Matlab environement:

```
NLSStatus MySeq::calculatePTX(...)
{
    // Condition to activate pTx pulse preparation such as :
    if ( rMrProt.getsTXSPEC().getaPTXRFPulse().size() == 1 )
    {
        if ( !m_SBBpTx.calculatePTX(rMrProt, pSeqLim) )
        {
            UTRACE(Error,0,"Could not prepare the pTx pulse.");
            return SEQU_ERROR;
        }
    }
    return SEQU_NORMAL;
}
```

Adding the run of the SBB in the runKernel of the sequence:

```
    // Apply a gradient rotation of the pTx blips to be imune
    // to orientation of the acquisition
    if ( rMrProt.getsTXSPEC().getaPTXRFPulse().size() == 1 )
    {
        if (! m_SBBpTx.RotateGradientsPTX(m_asSLC[lSlice]->getROT_MATRIX()) )
        {
            UTRACE(Error,0,"m_SBBpTx.RotateGradientsPTX() failed.");
            return SEQU_ERROR;
        }
    }

    // Run the pulse via the SBB using a start time
    if ( ! m_SBBpTx.run( lStartTime ) )
```

```
{
    UTRACE(Error,0,"m_SBBpTx.run( lStartTime ) failed.");
    return SEQU_ERROR;
}
```

3 Advanced possibilities

3.1 Flip angle

By default, the nominal flip angle is used by the pTx RF pulse. However, it can be changed by the sequence, for instance by using a protocol parameter.

```
m_SBBpTx.setFlipAngle ( rMrProt.flipAngle() );
```

For a more advanced possibility to prevent any clipping of the voltages defined in the ini file, one can checked that the maximum flip angle is not exceeded as follow:

```
// Prevent flip angle higher than as defined in INI file
if ( rMrProt.flipAngle() > m_SBBpTx.getMaximalFlipAngleIni() )
{
    if ( rSeqLim.isContextPrepForMrProtUpdate() )
    {// This condition helps to repair the protocol if the ini file
     // has a lower maximal flip angle than the one set in the protocol
        rMrProt.flipAngle(m_SBBpTx.getMaximalFlipAngleIni());
    }
    else
    {return SEQU_ERROR;}
}

m_SBBpTx.setFlipAngle ( rMrProt.flipAngle() );
```

3.2 Flip angle array

The setFlipAngleArray method is available in this class. In this case, setRunIndex(int) can be used to change array index during the runKernel. In addition, the getRFInfo method is based on the standard behavior of Siemens pulses, i.e. without argument it returns a mean of the RFInfo of the flipangle array. It is also possible to get a given index with RFInfo(int).

```
// Using a flipangle array of size 2:
```

```
double *adFA = new double[2];
adFA[0] = 5; adFA[0] = 10;
m_SBBpTx.setFlipAngleArray(adFA, 2);

// In runKernel, use the flipangle index with:
m_SBBpTx.setRunIndex(1);
m_SBBpTx.run(lStartTime); // will use the 2nd flip angle in the array
```

3.3 Pointer to pTx pulse object

In some Siemens sequences, it is useful to give the RF object address so that the pulse is completely handled by an given SBB in the sequence. The "libns_pTx" library was prepared to handled this behaviour. For this reason, all standard RF object features were rewritten to be available in this custom pTx object. One can retrieve the pTx pulse address and then pass it on to other SBB that will prepare() and run() the pulse as any other.

```
// Option : the IRF_PULSE object can be obtained via:
m_pSRF = &m_SBBpTx.m_SRFpTX;
```

4 Your feedback

We hope to enhance the features of this package. Your feedback is welcome regarding the use of this library. In particular, we are interested in having feedbacks on the following topics:

- if you encounter unexpected behavior, or bugs
- if you have suggestions for improvments
- if you think of specific features or wanted sequences
- if you have any tips to share

Please contact the authors of this package.

References

1 Gras, V., Boland, M., Vignaud, A., Ferrand, G., Amadon, A., Mauconduit, F., Le Bihan, D., Stöcker, T. and Boulant, N. (2017), 'Homogeneous non-selective and slice-selective parallel-transmit excitations at 7 Tesla with universal pulses: A validation study on two commercial RF coils', *PLOS ONE* **12**(8), e0183562.
URL: *https://dx.plos.org/10.1371/journal.pone.0183562*

2 Gras, V., Mauconduit, F., Vignaud, A., Amadon, A., Le Bihan, D., Stöcker, T. and Boulant, N. (2018), 'Design of universal parallel-transmit refocusing k_t-point pulses and application to 3D T_2-weighted imaging at 7T: Universal Pulse Design of 3D Refocusing Pulses', *Magnetic Resonance in Medicine* **80**(1), 53–65.
URL: *http://doi.wiley.com/10.1002/mrm.27001*

3 Gras, V., Pracht, E. D., Mauconduit, F., Le Bihan, D., Stöcker, T. and Boulant, N. (2019), 'Robust nonadiabatic T_2 preparation using universal parallel-transmit k_t-point pulses for 3D FLAIR imaging at 7 T', *Magnetic Resonance in Medicine* **81**(5), 3202–3208.
URL: *http://doi.wiley.com/10.1002/mrm.27645*

4 Gras, V., Vignaud, A., Amadon, A., Bihan, D. and Boulant, N. (2017), 'Universal pulses: A new concept for calibration free parallel transmission', *Magnetic Resonance in Medicine* **77**(2), 635–643.
URL: *https://onlinelibrary.wiley.com/doi/10.1002/mrm.26148*

5 Mauconduit F, e. a. (2022), Traveling pulses visit 7t terra sites: Getting ready for parallel transmission in routine use, *in* 'Proceedings of the International Society for Magnetic Resonance in Medicine', London, United Kindom.

6 Van Damme, L., Mauconduit, F., Chambrion, T., Boulant, N. and Gras, V. (2021), 'Universal nonselective excitation and refocusing pulses with improved robustness to off resonance for Magnetic Resonance Imaging at 7 Tesla with parallel transmission', *Magnetic Resonance in Medicine* **85**(2), 678–693.
URL: *https://onlinelibrary.wiley.com/doi/10.1002/mrm.28441*

7 Wu, X., Gras, V., Vignaud, A., Mauconduit, F., Boland, M., Stoecker, T., Ugurbil, K. and Boulant, N. (2018), The travelling pulses: multicenter evaluation of universal pulses at 7T., *in* 'Proceedings of the International Society for Magnetic Resonance in Medicine', Paris, France.