Table of Contents

1. Introduction

The 'Yarnball' scripts in this GitHub repository were developed by Rob Stobbe at the University of Alberta. Please cite the 2020 MRM paper, 'Three-dimensional Yarnball k-space acquisition for accelerated MRI' if using this work. Report any errors or problems to rstobbe@ualberta.ca.

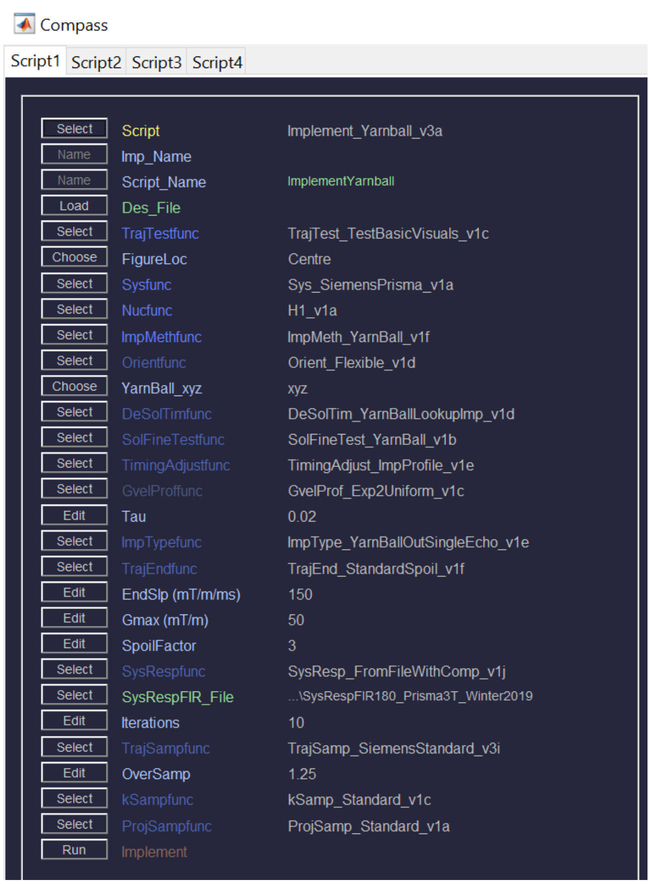
The Yarnball scripts provided are designed to be run within the 'Compass' software tool. This tool is also available for download at (https://github.com/rstobbe/Compass).

The 'CompassScriptsManual' document in the Compass repository describes how to use Compass. Examples for running Yarnball scripts are given in that document.

This document describes the **ImplementYarnball** script in the 'Scripts' folder.

2. The ImplementYarnball Script

The purpose of the **ImplementYarnball** script is to generate gradient waveforms for a Yarnball design previously created with the **DesignYarnball** script. The following information will appear in the associated column when the default (on GitHub) **ImplementYarnball** script is selected.



3. The Overarching Function

The **ImplementYarnball** script currently calls the overarching function 'Implement_Yarnball_v3a'. This function is effectively a container function. The purpose of this overarching function is to load saved output from running a **DesignYarnball** script, and to facilitate alternative selection of TrajTestfunc, Sysfunc, Nucfunc, and ImpMethfunc. These functions are described below.

3.1. Loading the Output from Running a DesignYarnball script

There are 2 ways to load the output from running a **DesignYarnball** script:

1. [Left click] the **Load '**Des_File' button and find the script with Windows Explorer.
2. First select/highlight the appropriate output in the **Loaded** window, the [Shift + right click] the **Load '**Des_File' button. This will load the selected/highlighted file.

Note that a [right click] on the **Load '**Des_File' button will display the information associated with the selection in the **Info** window.

4. The Trajectory Test Function (TrajTestfunc)

The trajectory test function (or TrajTestfunc) controls whether all Yarnball trajectories are to generated ('TrajTest_Imp…' functions), or whether just a representative subset is to be generated for testing ('TrajTest_Test…' functions). The functions provided on GitHub offer varying numbers of figures generated.

5. The System Function (Sysfunc)

The system function (or Sysfunc) contains information pertinent to the MRI system. Currently just the Siemens Prisma is supported.

6. The Nucleus Function (Nucfunc)

The nucleus function (or Nucfunc) simply determines what gyromagnetic ratio for trajectory implementation.

7. The Implement Method Function (ImpMethfunc)

The implement method function (or ImpMethfunc) is primarily a container function. The only function currently available for selection on GitHub ('ImpMeth_YarnBall_v1e') calls 10 underlying design functions. Alternative implement method functions may call greater or fewer functions or different types of functions.

7.1. ImpMeth_YarnBall_v1f

The ‘ImpMeth_YarnBall_v1e’ function is used for basic Yarnball implementation and the functions it calls are described below.

8. Underlying Functions

The function underlying Yarnball implementation are described below. The implement method function (ImpMethfunc) may call any number of these functions.

8.1. The Orientation Function (Orientfunc)

The orientation function (or Orientfunc) describes how the Yarnball axes are to be oriented within k-space.

8.1.1. Orient_Flexible_v1d

‘Orient_Flexible_v1d’ is currently the only function available for selection on GitHub. This function offers a direct mapping between Yarnball ‘xyz’ and physical ‘xyz’ (i.e. Yarnball ‘x’ maps to the first axis/letter in the selection, and so on).

8.2. The Differential Equation Solution Timing Function (DeSolTimfunc)

The differential equation solution timing function (or DeSolTimfunc) describes how finely/coarsely (in time) the differential equations are solved. Only functions that include the term ‘Imp’ can be selected within the context of Yarnball implementation. These functions should be selected as described for Yarnball design.

8.3. The Solution Fineness Test Function (SolFineTestfunc)

The solution fineness test function (or SolFineTestfunc) is tests whether the trajectory has been sampled finely enough for implementation. There is currently only one function to select: ‘SolFineTest_YarnBall_v1b’.

8.4. The Timing Adjust Function (TimingAdjustfunc)

The timing adjust function (or TimingAdjustfunc) describes how the Yarnball trajectory solution time segments will be modified to achieve a desired gradient slew rate (or level of peripheral nerve stimulation). Functions that include the term ‘DesignTest’ cannot be used within the context of Yarnball implementation.

8.4.1. TimingAdjust_ImpProfile_v1e

The 'TimingAdjust_ImpProfile_v1e' function will constrain the Yarnball waveform to a gradient velocity profile determined by the GvelProffunc (or to approximately this profile – the constraint function may not be able to exactly match the profile).

8.4.1.1. GvelProf_Exp2Uniform_v1c

This 'GvelProf_Exp2Uniform_v1c' function defines a gradient velocity profile that rises to a constant value at $\{1 - \exp(-t/\tau)\}$, where t is relative time from 0 to 1 (at the end of the trajectory), and τ (**Tau**) is a fraction of that relative time. A larger **Tau** results in a slower rise to full value.

8.4.1.2. GvelProf_Exp2LinearDecay_v1a

This 'GvelProf_Exp2LinearDeacy_v1a' function defines a gradient velocity profile that rises to a constant value at $\{1 - \exp(-t/\tau)\}$, where t is relative time from 0 to 1 (at the end of the trajectory), and τ (**Tau**) is a fraction of that relative time. A larger **Tau** results in a slower rise to full value. At relative time **DecayStart**, the profile will begin to decay at the rate **DecayRate**.

8.4.1.3. GvelProf_Exp2OvershootDecay_v1c

This 'GvelProf_Exp2LinearDeacy_v1a' function defines a gradient velocity profile that rises to a constant '**Overshoot**' value at $\{1 - \exp(-t/\tau)\}$, where t is relative time from 0 to 1 (at the end of the trajectory), and τ (**Tau**) is a fraction of that relative time. Note that the profile is still displayed in a relative sense with a maximum value of 1. A larger **Tau** results in a slower rise to the '**Overshoot**' value. At relative time **DecayShift**, the overshoot will decay according to **OvershootDecay** (a larger number results in a faster decay). Beyond the relative time **DecayShift** the gradient velocity profile will continue to decay at the rate **EndDrop**.

8.4.2. TimingAdjust_Dummy_v1a

The 'TimingAdjust_Dummy_v1a' function does not constrain the gradient velocity of the Yarnball waveform.

8.5. The Implementation Type Function (ImpTypefunc)

The implementation type function (or ImpTypefunc) facilitates different types of Yarnball sampling. In the future, multi-echo spin-in/out functions will be added here as these papers are published.

8.5.1. ImpType_YarnBallOutSingleEcho_v1e

'ImpType_YarnBallOutSingleEcho_v1e' is currently the only function available for Yarnball design selection on GitHub. This function facilitates standard Yarnball implementation from the centre of k-space outward and ends at the edge of k-space.

8.6. The Trajectory End Function (TrajEndfunc)

The trajectory end function (or TrajEndfunc) determines how the Yarnball trajectory will end.

8.6.1. TrajEnd_StandardSpoil_v1e

'TrajEnd_StandardSpoil_v1e' ends the trajectory with constant spoiling to the value **SpoilFactor** (as a factor of kmax). The maximum value of the spoiler (**Gmax**) and the maximum gradient slope (**EndSlp**) are also selectable.

8.6.2. TrajEnd_StandardRephase_v1d

'TrajEnd_StandardRephase_v1d' ends the trajectory with rephasing. The maximum gradient slope (**EndSlp**) is selectable.

8.6.3. TrajEnd_Spoil1ChanZeroOthers_v1e

'TrajEnd_Spoil1ChanZeroOthers_v1e' ends the trajectory with constant spoiling to the value **SpoilFactor** (as a factor of kmax) on the channel specified by **Dir**. The other channels are rephased. The maximum value of the spoiler (**Gmax**) and the maximum gradient slope (**EndSlp**) are also selectable.

8.7. The System Response Function (SysRespfunc)

The system response function (or SysRespfunc) determines compensates for the gradient system response of the scanner.

8.7.1. SysResp_FromFileWithComp_v1j

'SysResp_FromFileWithComp_v1i' uses the loaded **SysRespFIR_File** to compensate for the gradient system response of the scanner. A **SysRespFIR_File** file associated with the Siemens Prisma scanner (at the University of Alberta, Edmonton, Canada) is located in the Yarnball\Supporting folder. **Iterations** identifies how many passes the compensation algorithm should use to reduce trajectory error.

8.7.2. SysResp_noResp_v1a

'SysResp_NoResp_v1a' does not perform any compensation.

8.8. The Trajectory Sample Function (TrajSampfunc)

The trajectory sample function (or TrajSampfunc) determines the rate of sampling (or sampling dwell time) along the trajectory.

8.8.1. SysResp_noResp_v1a

'TrajSamp_SiemensStandard_v3i' is currently the only function available for selection on GitHub. The **OverSamp** parameter defines Siemens specific oversampling.

8.9. The k-Space Sample Function (kSampfunc)

The k-space sample function (or kSampfunc) determines the final k-space value of each sampled point. The function 'kSamp_Standard_v1c' is currently the only function available on GitHub.

8.10. The Projection Distribution Function (ProjSampfunc)

The projection distribution function (or ProjSampfunc) governs how the Yarnball trajectories are distributed. The function 'ProjSamp_Standard_v1a' is currently the only function available on GitHub.

9. Output

Yarnball trajectory information (including gradient waveforms and k-space locations) from running the ImplementYarnball script is stored in structure labelled 'IMP' inside a variable labelled 'saveData' within the saved output file.