1 Development of the 3D Stack of Stars Sequence

1.1 General Design

Proposed Pulse Diagram An optimized version of a generic pulse diagram was implemented (see Fig. 1.1). The generic version was modified as follows:

1. The slab-rephasing gradient (A) is combined with the partition-encoding gradients (B), resulting in gradients (F).

2. The prephasing readout gradients are applied simultaneously with the gradient (F).

3. The spoiler gradient (E) in the readout direction was eliminated, and instead, the flat time of the readout gradients was prolonged to account for the desired phase dispersion in the readout direction.

4. The spoiler gradient in the Z direction (D) was combined with the partition-rephasing gradients resulting in gradients (H).

The first two aspects aimed to reduce TE, whereas the last two aimed to reduce TR.

Software Architecture Since both sequences have some sequence events in common (e.g., the slab selection events), an object-oriented programming paradigm was used. It makes the code reusable, extensible, and easy to maintain. The class diagram is presented in Fig. 1.2, where the class hierarchies are visualized. The software comprises two superclasses, one called `protocol` to validate all the input parameters and make user-friendly changes in contrast and resolution, and the superclass `kernel` which creates all the sequence events and the final .seq file to be executed on the scanner. Subclasses are created for code inherent to each sequence.

Available Features The sequence has a versatile design that allows the user to choose among different options. An overview of the currently available features is given in Tab. 1.1.

Depending on the combination of selected features, the sequence can be modified to accomplish specific goals; the following guidelines can assist future users to tailor the features to their needs:

- To be time efficient use non-selective RF excitation, a small number of dummy scans, and no gradient spoilers; in this setting, spoiling is achieved by RF-spoiling and, inherently by half of the readout gradient.

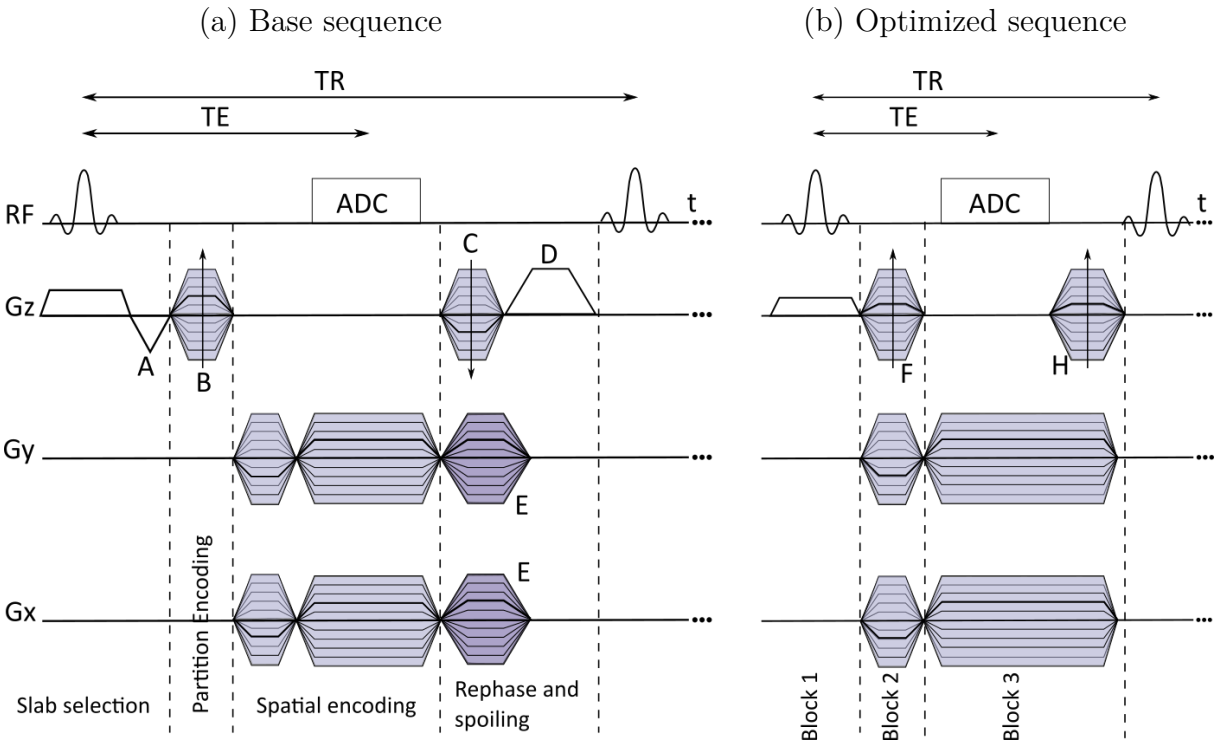

Figure 1.1: Pulse sequence diagrams for a generic (a) and an optimized (b) Stack of Star sequence. The optimized version was implemented in this project.

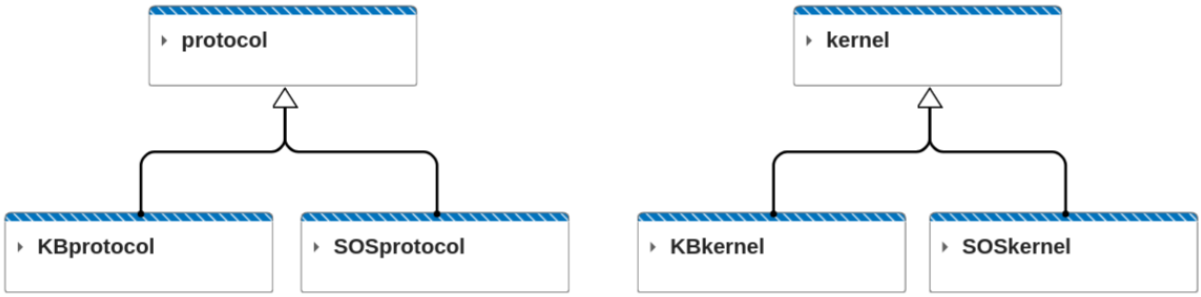

Figure 1.2: Overview of the resulted system architecture for the creation of the sequences.

.

Table 1.1: Available features in the Stack of Stars Sequence

Available Features	Options
2x readout oversampling	yes/no
Spoiling in readout	desired phase dispersion [rad]
Spoiling in Z	desired phase dispersion [rad]
Angular ordering	uniform, alternating, golden angle
Angle range	[0,360°)/[0,180°)
Partition Rotation	aligned, linear, golden angle
View Order	partitionsInInnerLoop, partitionsInOuterLoop
RF excitation	selective, non-selective
Include dummy scans	desired number of dummy scans
Suggest number of dummy scans	ideal number of dummy scans
Adaptable TE/TR and resolution	

- Apply spoiler gradients in the readout (prolonged the readout gradient), Z, or both directions to better eliminate residual transverse magnetization.
- Increase the number of dummy scans to be sure to start the measurements after reaching the steady-state.
- Use golden angle ordering in the in-plane and z-direction to improve the undersampling capabilities of the sequence
- Avoid wrap-around artifacts by using 2x readout oversampling.

1.2 Detailed Design Process

1.2.1 Protocol Validation

Input data consistency MR imaging is known to have many user-selectable acquisition parameters. Usually, only the most important ones are given by the user, and the rest are calculated under the hood by the program. To avoid errors or unexpected behavior in the resulting sequence, it is necessary to check the data consistency as a first step after the user enters the desired parameters and features. This is implemented in the `properties` of the superclass `protocol`. For instance, the number of radial spokes is expected to be a non-negative integer; the RF excitation option is expected to be 'selectiveSinc' or 'nonSelective'; any other input for these parameters will cause an error. However, some values will be automatically changed by the program, as in the case of resolution and contrast parameters, which is described next.

Changes in Contrast and Resolution The sequence was created with a flexible design, which allows for user-friendly changes in contrast and resolution. The priority is given to the resolution, followed by the contrast. It was necessary to impose certain limits on the system and resolution parameters, all of them presented in Tab. 1.2 and implemented as `constant properties` of the superclass `protocol`.

Some limits are set due to safety reasons; for instance, in [1], the authors of the Pulseq environment recommend avoiding using high gradient amplitudes when creating the sequences; therefore, even when the maximum gradient amplitude of the scanner is 72 mT/m, this was restricted to 50 mT/m in all the measurements. Some other limits are based on physical constraints; for instance, the maximum FOV is limited by the scanner aperture, which is 600 mm. The rest of the limits were taken from the datasheet of a commercial 7T Siemens scanner [2] available on the internet.

Table 1.2: Limits of the system and resolution parameters

Parameter	Minimum	Maximum
Gradient amplitude [mT/m]	0	50
Gradient slew rate [mT/m/s]	0	150
Base rsolution n	64	1024
Field of view FOV [mm]	10	500
Bandwidth per pixel BW_{pixel} [Hz]	100	2000
In-plane resolution Δx [mm]	0.2	8
Number of partitions n_z	5	1024
Slab thickness [mm]	10	500
Transmitter bandwidth Δf [KHz]	1.5	250
Partition thickness Δz [mm]	0.2	20
RF pulse duration T_{rf} [ms]	0.02	12
Time bandwidth product $T_{rf}\Delta f$	2	10

In-plane Resolution The procedure to check and change the in-plane resolution is presented in Fig. 1.3 and described next.

The primary individual parameters for the in-plane resolution given by the user are n, FOV, and BW_{pixel}. First, the algorithm checks if those parameters are within limits, forcing them to be at least at the extremes if they do not fall within limits. For instance, a FOV of 700 mm will be set to 500 mm. Second, the parameters that depend on two or more single parameters are checked, starting with the in-plane resolution $\Delta x = FOV/n$ and finalizing with the gradient amplitude $G_{max} = n \cdot BW_{pixel}/FOV$ (the term $\frac{\gamma}{2\pi}$ has been absorbed into G_{max} for convenience here, then the units of the gradients are in [Hz/m]). When fixing the in-plane resolution, the number of samples is decreased until a valid resolution is achieved. To fix the maximum gradient amplitude, the BW_{pixel} is reduced until a valid gradient amplitude is reached.

Through-plane Resolution The procedure to check and change the through-plane resolution is presented in Fig. 1.4 and described next.

The primary individual parameters for the through-plane resolution given by the user are: the number of partitions n_z, the slab thickness, the RF pulse duration T_{rf} and, the time-bandwidth product TBW. Here again, the algorithm checks if those parameters are within limits, fixing them if necessary. Then, the parameters that depend on two or more single parameters are checked, starting with the partition thickness $\Delta z = slabThickness/n_z$, followed by the transmitter bandwidth $\Delta f = T_{rf}/TBW$ and finally, checking the gradient amplitude $G_{max} = \Delta f/(n_z \cdot \Delta z)$. To fix Δz, n_z is increased or decreased until Δz is within the limits. To fix Δf, TBW is increased or decreased until a valid Δf is reached. Finally, the maximum gradient amplitude in Z direction is fixed by increasing the RF pulse duration until a valid gradient amplitude is achieved.

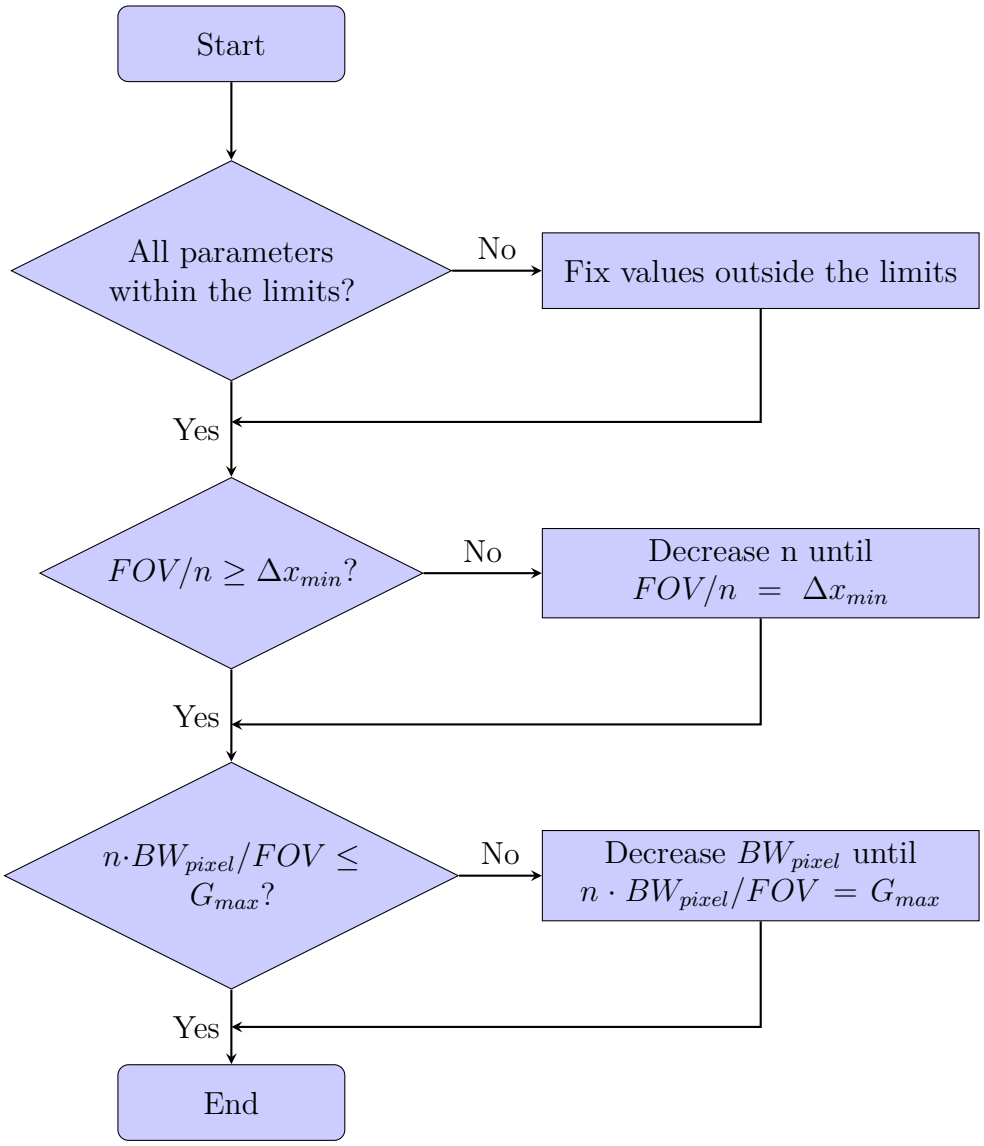

Figure 1.3: Flow chart describing the procedure to check and change the in-plane resolution parameters.

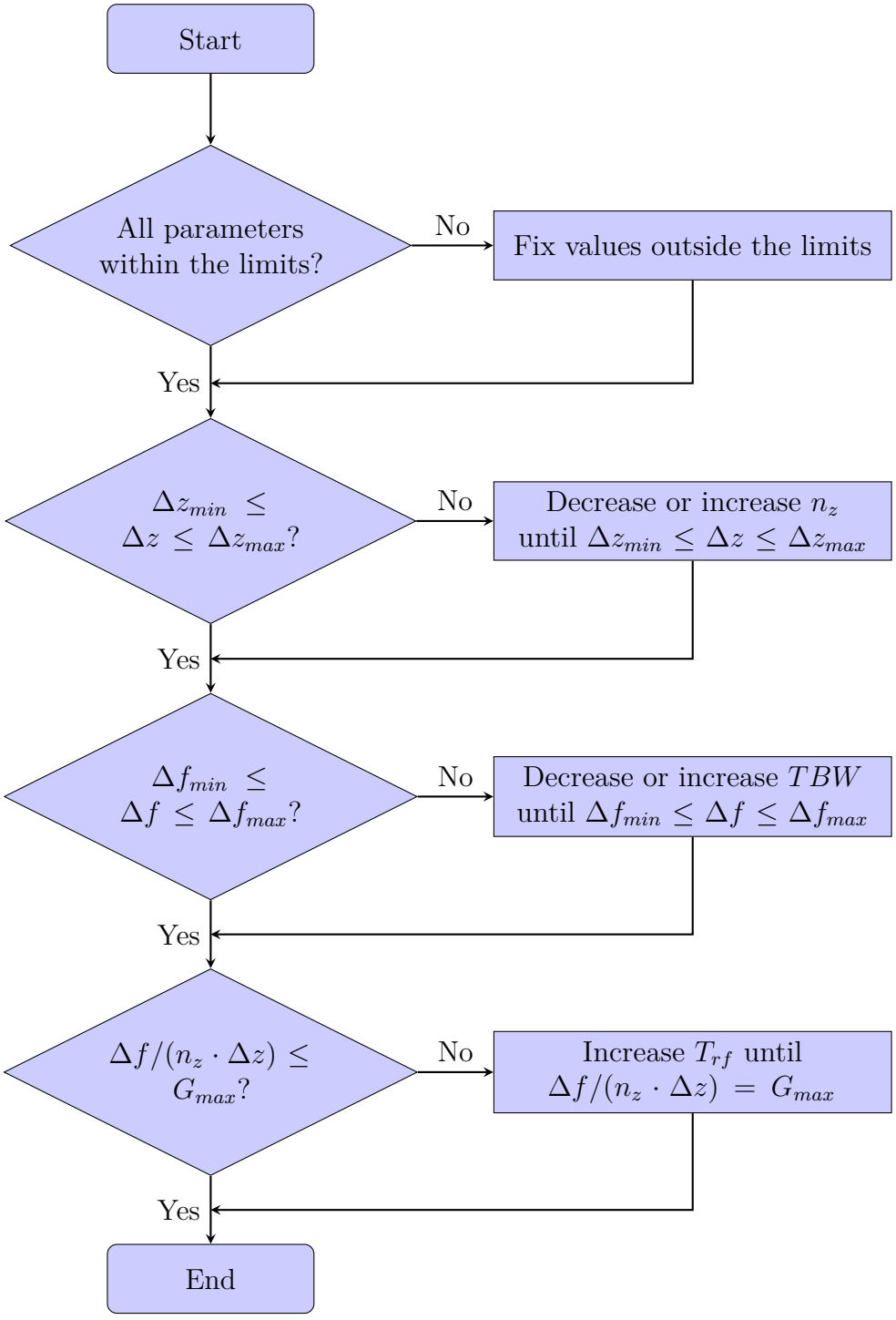

Figure 1.4: Flow chart describing the procedure to check and change the through-plane resolution parameters.

Contrast The procedure to check and change the TE and TR values is presented in Fig. 1.5 and described in the following.

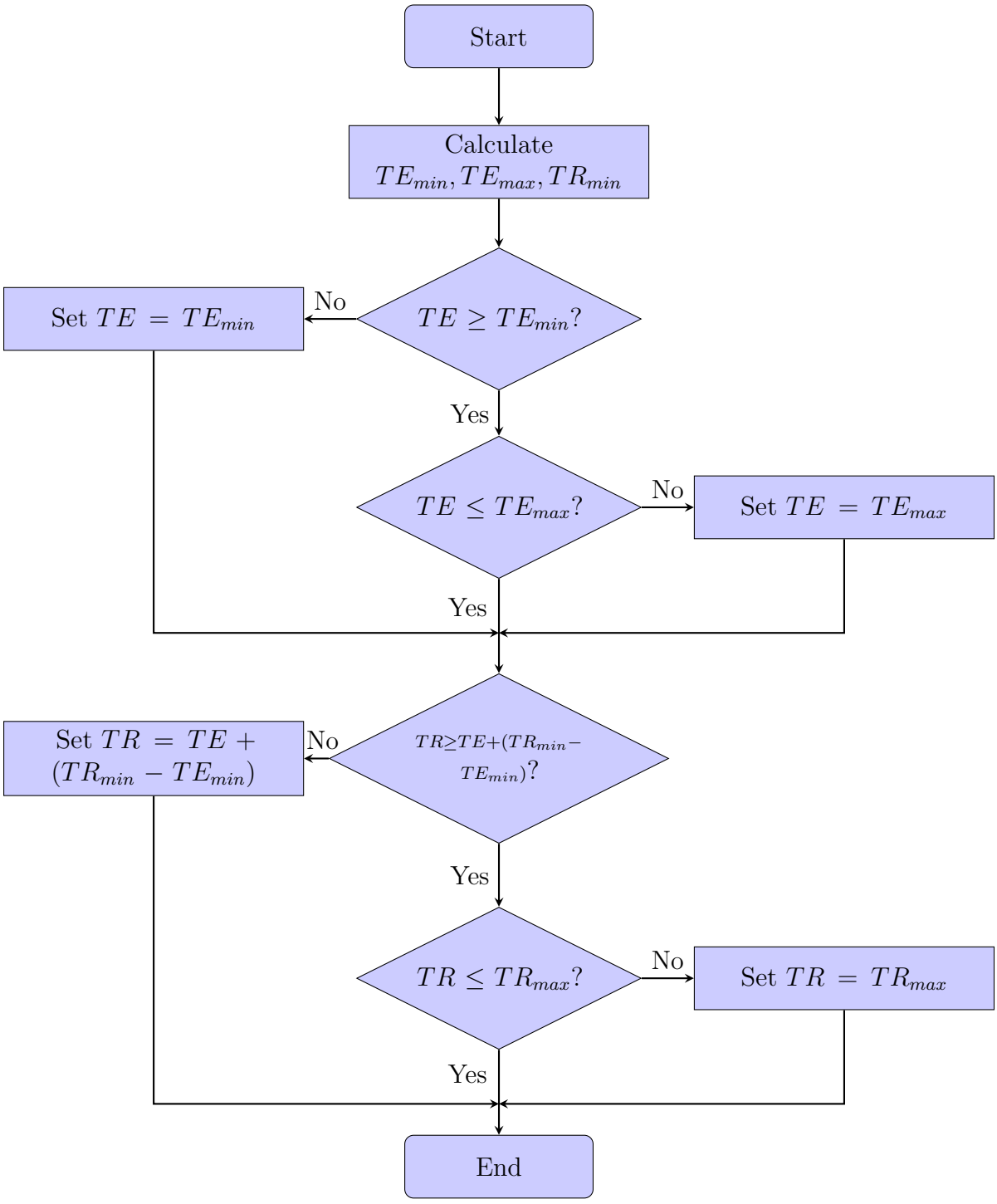

Figure 1.5: Flow chart describing the procedure to check and change the TE and TR values.

After having fixed all the resolution parameters, the algorithm proceeds to pre-calculate the duration of all the sequence events and with that information at hand, the minimum achievable TE and TR values illustrated in Fig. 1.1b can be calculated. The TR_{max} is set

arbitrarily to 100 ms and, the minimum TE-TR-interval is calculated as $(TR_{min} - TE_{min})$. The TE_{max} is then calculated as TR_{max} minus the minimum TE-TR-interval. Having those limits for TE and TR, the algorithm proceeds to fix TE, followed by TR. Specifically, if the intended TE is less than TE_{min}, TE is set to TE_{min}, or if the intended TE is greater than the TE_{max}, it is set to TE_{max}. Similarly, if the intended TR is less than $TE_{fixed} + (TR_{min} - TE_{min})$, then TR is set to $TE_{fixed} + (TR_{min} - TE_{min})$, or if the intended TR is greater than the TR_{max}, it is set to TR_{max}.

Estimation of the Number of Dummy Scans Usually, some dummy scans have to be performed in GRE sequences to start the actual measurements in a steady state condition. The normalized convergence error between the longitudinal magnetization before the nth pulse $M_{z'}^{n}(0_{-})$, and the steady state longitudinal magnetization $M_{z'}^{ss}(0_{-})$ is given by [3]:

$$\frac{M_{z'}^{n}(0_{-}) - M_{z'}^{ss}(0_{-})}{M_{z'}^{ss}(0_{-})} = \frac{(cos\alpha E_1)^n E_1(1 - cos\alpha)}{1 - E_1} \tag{1.1}$$

where $E_1 = e^{-TR/T_1}$.

Once the flip angle α and repetition time TR are chosen, the user receives feedback on the number of dummy scans required to have a longitudinal magnetization within a particular error relative to the steady-state value. For that purpose, a function was created that takes α, TR, the desired relative error, and a particular T_1 value as inputs; and uses Eq. 1.1 to estimate the required number of dummy scans. The user can change the desired relative error, and the T_1 value depending on the part of the body to scan; for instance, for $\alpha = 5°$, $TR = 3$ ms, $error = 10\%$ and T_1 for the white matter at 7T of 1184 ms, the estimated and suggested number of dummy scans will be 454, which correspond to approximately 1.3 seconds of waiting time before starting the data acquisition. However, the program only suggests the number of dummy scans, and it is ought to the user whether or not to change the current value.

1.2.2 Kernel

Creation of Sequence Events After all the input parameters have been validated, the algorithm proceeds to calculate all the sequence events. The custom toolbox provided by the Pulseq environment is used for that purpose. The use of the functions provided by the Pulseq toolbox simplifies the creation of sequence events. For instance, when creating a gradient for which only the total area is important, such area is given as an input, and a function creates the gradient using the shortest possible duration by creating a triangular or trapezoidal waveform.

Slab Selection Part For a non-selective RF excitation, a rectangular RF pulse is used

without a slab selection gradient. Whereas for a slab-selective RF excitation, a SINC RF pulse is applied together with a slab selection gradient. The SINC pulse can be apodized using a Hanning or Hamming function, and the user makes this choice. Possible remanent Eddy currents from the spoiler gradients in previous repetitions are trying to be avoided by setting a dead time of 100 μs before applying the RF pulse and the flat region of the slab selection gradient. This dead time may also help to avoid overheating of the gradient coils and amplifiers and to keep the specific absorption rate (SAR) levels within the limits [4]. An RF ring-down time [5] of 20 μs was added at the end of the RF pulse; if not, an error occurs when executing the sequence in the IDEA environment.

Partition Encoding Part The partition encoding step size in k-space is calculated as

$$\Delta k_z = 1/(slab\ Thickness) \tag{1.2}$$

to satisfy the Nyquist criterion. The partitions are acquired asymmetrically with respect to $k_z = 0$ and starting at the bottom of the k-space at

$$-k_{z,max} = -(n_z/2)\Delta k_z \tag{1.3}$$

The different gradient partition areas are calculated as

$$area(m) = (m - n_z/2)\Delta k_z, \quad m = 0, 1, ..., n_z - 1 \tag{1.4}$$

Spatial Encoding and Data Sampling Part At this point, one generic prephasing $G_{x,pre}$, and one readout gradient G_x in the x-direction are created. These two gradients can be thought of as the necessary ones to generate the first spoke trajectory in k-space at 0° [6, p. 27]. As explained in a later stage (stage 3), these two gradient waveforms are used as the blueprints to generate the spokes' different trajectories, which will depend on the desired angular ordering.
The duration of the flat region of the readout gradient is made equal to the period of data acquisition, which is related to the user input BW_{pixel} as:

$$T_{acq} = 1/BW_{pixel} \tag{1.5}$$

To avoid timing errors in the sequence, T_{acq} is rounded to the nearest multiple greater than or equal to 10 μs, which is the gradient raster time used. The flat area of the readout gradient is then calculated as the product of T_{acq} and the gradient amplitude, which results in:

$$flat\ area\ readout = n\Delta k_x \tag{1.6}$$

The flat area and the flat time are enough to calculate the readout gradient with Pulseq.

The prephasing gradient is then created to have a total area of one half of the total area of the readout gradient and opposite polarity. The data is acquired asymmetrically with respect to $k_x = 0$; the starting sampling point is

$$-k_{x,max} = -(n/2)\Delta k_x \tag{1.7}$$

and at the echo time, the zero-frequency component of the spoke is measured.
The duration of the ADC event is made equal to Tacq, and the number of samples is:

$$number\ of\ samples = n \cdot readoutOversampling \tag{1.8}$$

The input parameter *readoutOversampling* can have the value of one or two. When it is two, the sampling rate of the ADC is doubled, reducing the k-space step Δk_x and increasing the final FOV by a factor of two. The intended spatial resolution and the gradient amplitude of the readout gradient do not change.

Spoiling The aim of the two sequences is to produce T_1-weighted contrast in the final image. This is achieved by making the transverse magnetization zero before each RF excitation. [7]. There are two mechanisms to achieve spoiling, the first one is to use spoiler gradients. The effect of a spoiler gradient is to add a phase dispersion within a voxel. After applying the gradient with the correct amplitude, the vectorial sum of the transverse magnetization of different isochromats within the voxel is zero. An isochromat is a group of spins with the same resonance frequency in a voxel. It is found that the phase dispersion across a voxel is directly proportional to the spoiler gradient area A_{sp} and to the voxel dimension along the spoiler gradient direction Δr [8, p. 353]:

$$\Delta\phi(r) = \gamma A_{sp} \Delta r \tag{1.9}$$

However, in GRE sequences for which $T_2 >> TR$, a steady state is reached after multiple RF excitations [9, p. 500], and even in the presence of spoiler gradients, some coherent steady state transverse magnetization will appear before each RF excitation. In [7] an RF spoiling strategy was proposed to further eliminate this transverse magnetization and consists of linearly incrementing the phase offset of the RF pulses in each RF excitation. Both strategies, gradient spoiling and RF spoiling, can be used synergistically.

Spoiling Part The spoiler gradient area in the readout direction is calculated based on the input parameter *phaseDispersionReadout*. The part of the readout gradient after TE already adds a phase dispersion within a voxel of π radians [10], therefore when the intended phase dispersion is less than or equal to π, no extra spoiler area is added in the

readout gradient. When the intended phase dispersion within a voxel is greater than π, the extra spoiler gradient area is calculated using Eq. 1.9 as:

$$A_{sp,extra} = (phaseDispersionReadout - \pi)/(\gamma \Delta x) \tag{1.10}$$

Then, the plateau of the readout gradient is prolonged to account for this extra spoiler gradient area.
The spoiler gradient area in the z-direction is calculated based on the input parameter $phaseDispersionZ$, and two options are considered.

1. When the intended phase dispersion within a voxel in the z-direction is zero, the partition encoding gradient is refocused using gradients with the same areas as the partition encoding gradients but with opposite polarity (gradients H in Fig. 1.1b). With this, we counteract the phase dispersion for the partition encoding gradient in each TR to avoid banding artifacts [8].

2. When the intended phase dispersion is greater than zero, then the extra phase dispersion needed to get the total intended phase dispersion in each TR is calculated by subtracting the absolute value of the phase dispersion due to the partition encoding gradient from the total intended phase dispersion. With that extra phase dispersion, the necessary additional spoiler area is calculated using Eq. 2.15. Here again, a dummy gradient with the largest absolute spoiler area is calculated to get a fixed duration for all the gradients. Finally, spoiler gradients with the desired spoiler area and fixed duration are created (gradients H in Fig 1.1b), taking care of pairing its polarities with the partition encoding gradients, so its phase dispersions add constructively and its sum is equal to the desired phase dispersion in the z-direction. With this approach, the partitions located in k-space with $k_z < 0$ will have a negative phase dispersion, and partitions with non-negative k_z will have positive phase dispersion.

Creation of K-space Trajectory The base spoke angles ϕ_j for one partition are created depending on the input parameters: number of spokes per partition N_r, $angularOrdering$, and $angleRange$. If the angular ordering is set to 'uniform,' the spokes angles are calculated as:

$$\phi_{j,U} = \frac{\pi}{N_r} \cdot (j-1) \quad j = 1, 2, .., N_r \tag{1.11}$$

If the angular ordering is set to 'goldenAngle,' the spokes angles are calculated as:

$$\phi_{j,G} = (j-1)\pi/(\tau + N - 1), \quad j = 1, 2, .., N_r \tag{1.12}$$

where, $\tau = (\sqrt{5}+1)/2$ is the golden ratio [11] and N is another input parameter called $goldenAngleSequence$. When $N = 1$, we get the golden angle sequence [12], when $N = 2$,

we get the small golden angle sequence [13], and for $N > 2$, we get the tiny golden angle sequences [11, 14].

The user input *angleRange* dictates whether the spoke angles can vary from 0 to 180° ($angleRange = halfCircle$) or from 0 to 360° ($angleRange = fullCircle$).
Finally, the angle offset across different partitions is calculated according to the input parameters *partitionRotation* and the number of partitions n_z. For the aligned case, the partition rotation angles are set to zero. For the linear and golden angle case, the partition rotation angles are calculated respectively as [15]:

$$\varphi_{j,L} = \frac{\pi}{N_r} \cdot \frac{(j-1)}{n_z}, \quad j = 1, 2, .., n_z \tag{1.13}$$

$$\varphi_{j,G} = mod\left((j-1) \cdot \frac{\pi}{N_r} \cdot \frac{\sqrt{5}-1}{2}, \frac{\pi}{N_r}\right), \quad j = 1, 2, .., n_z \tag{1.14}$$

View Order For an explanation about why the *partitionsInInnerLoop* option is a good idea please refer to [16]. In that paper *partitionsInInnerLoop* is called $P - in - L$.

RF Phase Cycling Scheme For the RF spoiling, the RF phases were calculated using the phase-cycling scheme described in [8, p. 585] as:

$$\Phi_j = \Phi_{j-1} + \Phi_0, \quad j = 1, 2, 3, ... \tag{1.15}$$

There are many suitable starting values Φ_0 for this strategy, like 117°, 123°, 84°, 96.5° and 150° [17]. Therefore Φ_0 is set to be a user input parameter, with $\Phi_0 = 117°$ being the default value. Finally, the phases of the ADC events varied in the same manner as the RF pulses in each TR.

Gathering of Information for Reconstruction The algorithm collects and saves all the information required to reconstruct the images in a variable called `info4Reco`. For instance, it is essential to know whether or not readout oversampling was used, the view order used to collect the spokes, and the spoke angles to reproduce the k-space trajectory. Having this information at hand during the reconstruction allows us to reconstruct the images automatically.

2 Development of the 3D Koosh-Ball Sequence

2.1 General Design

Proposed Pulse Diagram Fig. 2.1b shows the final pulse diagram of the implemented 3D Koosh-ball sequence. It consists of seven gradient waveforms, one RF pulse, and one ADC event. This optimized version differs from the generic pulse diagram (see Fig. 2.1a) in two aspects:

1. The slab-rephasing gradient (A) is combined with the prephasing readout gradients in the z-direction.

2. The spoiler gradient (E) in the readout direction was eliminated, and instead, the flat time of the readout gradients was prolonged to account for the desired phase dispersion in the readout direction.

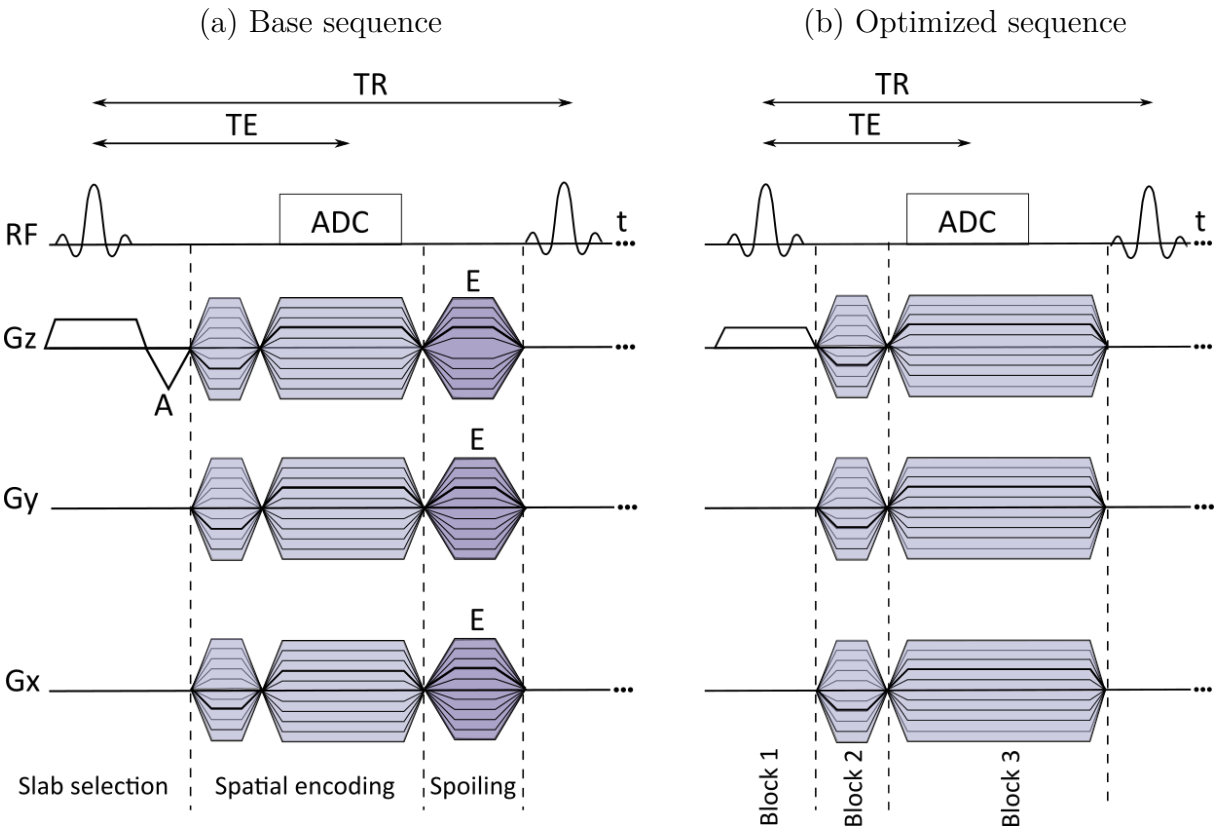

Figure 2.1: Pulse sequence diagrams for a generic (a) and an optimized (b) Koosh-Ball sequence. The optimized version was implemented in this thesis.

.

Software Architecture The architecture for this sequence is similar to the SOS counterpart as shown in Fig. 1.2.

Available Features 2x readout oversampling is available for this sequence. The gradient spoiling is achieved by extending the flat region of the readout gradient. The angular ordering can be uniform [18, 19] or pseudo-random using the 2D golden angle means [20, 21]. The RF excitation can be selective or nonselective.

2.2 Detailed Design Process

No validation of resolution in the z-direction is required as the resolution and FOV of this Koosh-ball sequence are isotropic and determined primarily by the readout resolution [22]. The rest of the protocol validation is exactly the same as the Stack of Stars (SOS) sequence. There are two major differences with respect to the SOS sequence implementation; the creation of the k-space trajectory and the design of calibration scans to calculate the gradient delay errors.

Creation of 3D K-space Trajectory Finding suitable 3D trajectories for full-echo koosh-ball sequences can be thought of as distributing points (the tips of the spokes) on a hemisphere in k-space [18], and each point location is determined by a polar θ, and azimuthal ϕ angle [19]. For the uniform case, the algorithm described in [19] was implemented to achieve regular equidistribution of points. The idea is that each point will be surrounded by a surface area equal to the average area per point $a = 2\pi/N_r$, assuming a hemisphere of radius 1 and $N_r = number\ of\ points = number\ of\ spokes$. This approach gives similar results compared with the equal-solid-angle strategy proposed in [18].
For the golden angle case, the polar and azimuthal angles are obtained using the 2D golden means $\varphi_1 = 0.4656$ and $\varphi_2 = 0.6823$ according to:

$$\begin{aligned} \theta_m &= \arccos\left(\ mod(m \cdot \varphi_1, 1)\ \right), \quad m = 1, 2, .., N_r \\ \phi_m &= 2\pi \cdot mod(m \cdot \varphi_2, 1), \quad m = 1, 2, .., N_r \end{aligned} \tag{2.1}$$

Fig. 2.2 shows the distribution of 1000 points corresponding to the starting and ending points of 500 spokes for the uniform (top) and 2D golden means (bottom) scenarios. The algorithm equidistributes the points through circles of constant latitudes in the uniform case, whereas the distribution is anisotropic in the 2D golden means case.

Gradient Delay Calibration Scans Calibration scans to retrospectively calculate the gradient delays were necessary as no openly available auto-calibrated methods exist for koosh-ball sequences. We acquired 120 spokes in each of three different orthogonal planes, x-y, z-x, and z-y. The first 40 spokes for each plane followed the same direction as the

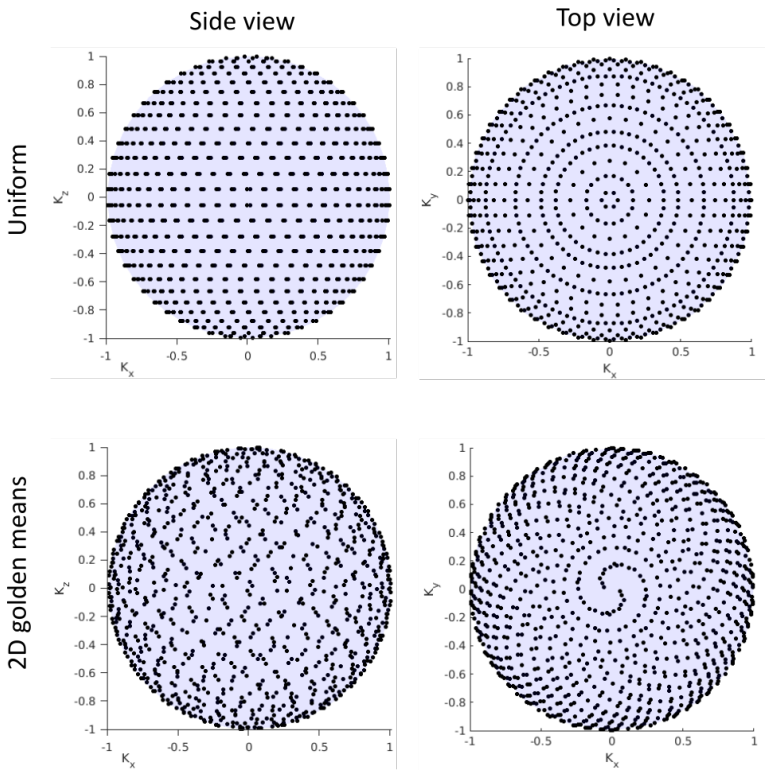

Figure 2.2: Distribution of 1000 points on the surface of a sphere of radius 1 for the uniform (top) and 2D golden means (bottom) strategies.

.

actual spokes, then the next 40 spokes were made antiparallel (+180°) to the first ones, and the last set of 40 spokes were made orthogonal (+90°) to the first ones.
The idea behind this configuration was that robust auto-calibrated methods exist to calculate the delay errors for 2D partitions. The AC-Adaptive method requires antiparallel (or almost antiparallel) spokes to calculate the shift in k-space [23], and the RING method requires orthogonal (or almost orthogonal) spokes to do the same [24] . Therefore, To enable the compliance with both methods, parallel and antiparallel spokes are acquired.

3 Image Reconstruction

BART provides a MATLAB wrapper, an addon that allows the user to call BART commands directly from MATLAB; therefore, the complete reconstruction pipelines were written in MATLAB. This approach simplifies the implementation, especially for MATLAB users.

3.1 Reconstruction Pipeline for the 3D Stack of Stars Sequence

The reconstruction pipeline is shown in Fig. 3.1. It was meant to perform partition-by-partition reconstructions of the measured data, and it consists of three stages explained next.

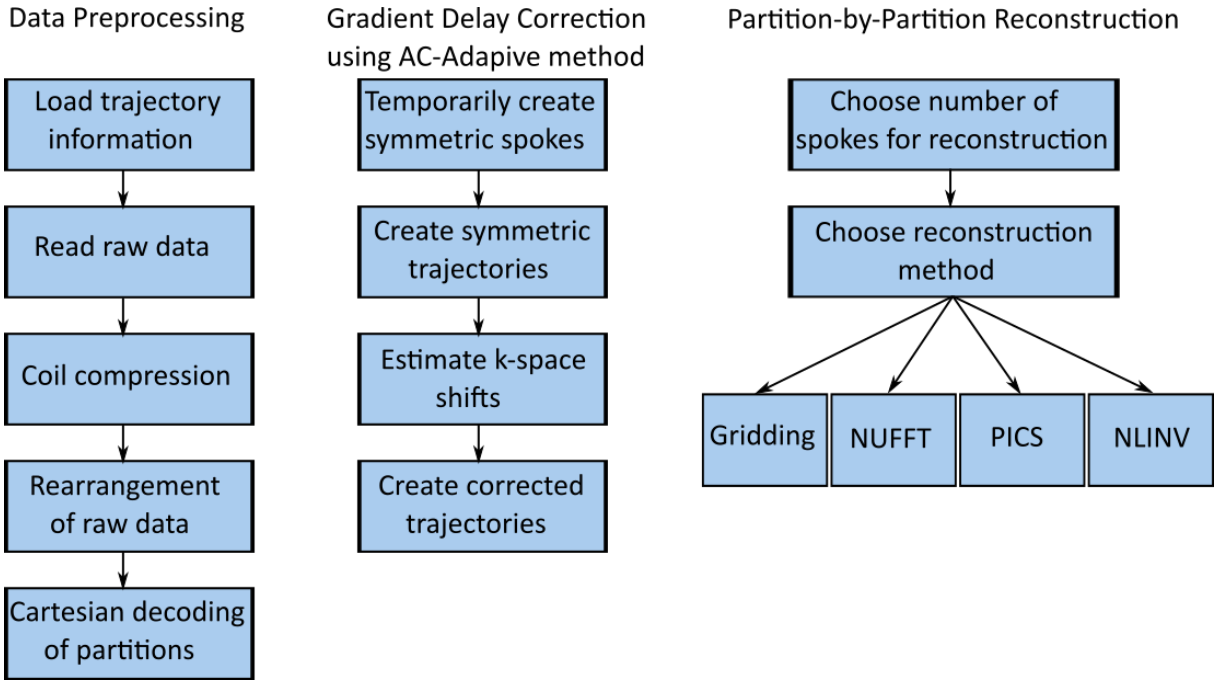

Figure 3.1: Reconstruction Pipeline for the Stack of Stars Sequence

.

Data preprocessing This stage is composed of five steps:

1. The necessary sequence and trajectory information saved during the sequence creation is loaded.

2. The raw data from the scanner in .dat format is read using the BART command `twixread`.

3. Coil compression is performed to reduce the amount of input data and computation time. The principal component analysis method [25] as implemented in BART through the `cc` command is used for this purpose. The final number of coils is set by the user.

4. The input data is a 4-dimensional (4-D) array. The first two dimensions correspond to a row vector containing the readout samples for one spoke. All the spokes ($N_r \cdot n_z$) are stacked in the third dimension, and the fourth dimension corresponds to the number of coils. It is necessary to reshape the third dimension so that the final multidimensional array is a 5-D array with the third dimension containing the spokes per partition, N_r, and the fifth dimension the different partitions, n_z. This step also takes into account the view order used during the acquisition of the spokes. For instance, if the view order was such that the partition-encoding axis was in the inner loop, then, before reshaping the third dimension, the spokes have to be ordered so that the spokes for each partition appear consecutively.

5. Finally, an inverse Fourier transform along the partition-encoding direction is performed to decode the different partitions and allow a partition-by-partition reconstruction.

It is important to note that the partition-by-partition reconstruction approach was chosen to reduce memory consumption and enable the reconstruction of single partitions. This allows, for instance, a quick assessment of one partition image to detect possible artifacts derived from the incorrect implementation of the sequence or the reconstruction pipeline. A full 3D reconstruction is possible for data sets derived from SOS sequences but at a higher cost in memory demand and reconstruction speed.

Gradient Delays Correction BART provides two methods to estimate the gradient-delay-induced k-space shifts, the RING, and the AC-Adaptive method. The latter method is used for the delay estimations since it brought good results even when large k-space shifts are observed. The gradient delay correction is done partition-by-partition, and it is composed of the following steps:

1. One partition is extracted from the data. As explained before Eq. 1.7, the acquired measurements are asymmetric concerning the center of the k-space, but the AC-Adaptive method requires symmetric spokes. Therefore, and for delays calculation purposes only, each spoke's first sample is taken off to make the spoke symmetric around the DC sample (i.e., to have the same number of samples to the left and to the right of the DC sample).

2. A nominal asymmetric radial trajectory for the current partition is created using the `traj` tool from BART and using the *custom angle file* option `-C` to provide

the spoke angles calculated during the sequence creation. As in step one, the first element in each spoke's trajectory is eliminated to make the nominal trajectory symmetric.

3. With the information from steps one and two, the gradient-delay-induced k-space shifts S_x, S_y and S_{xy} are estimated using the `estdelay` command from BART and which default method is the AC-Adaptive one.

4. Finally, a new asymmetric trajectory is created as in step 2, but this time using the $-q$ option to provide the estimated shifts, and the $-O$ option in the `traj` tool to correct the gradient delay errors in the radial trajectory. With this corrected trajectory, which contains all data points, the subsequent image reconstruction is performed.

Partition-by-Partition Reconstruction In the case of golden angle angular ordering, the user can select a custom number of spokes to reconstruct the data with a certain temporal resolution and at different time points [12]. For the reconstruction of each partition, four different methods can be used.

- **Gridding:** a Ram-Lak filter is used (see, [6, p.34]) to weight the data and compensate for its varying sample density. After that, the adjoint NUFFT as implemented in BART is applied to complete the gridding procedure. The resulting coil images are combined using the root-of-sum-of-squares approach [26]. Finally, the final image is cropped to eliminate the excess of FOV due to the 2x oversampling if necessary.

- **NUFFT:**: The procedure is the same as the gridding procedure, but with no density compensation and applying the inverse NUFFT instead of the adjoint NUFFT.

- **Pics:** The first step is to calculate the coil sensitivity maps, and this is done using the ESPIRiT calibration [27]. For this approach, the radially sampled k-space has to be converted into cartesian k-space; this is done by applying an inverse NUFFT to the data and then applying a forward Fourier transformation to the resulting coil images to get the gridded k-space data. After that, the sensitivity maps are calculated using the `ecalib` command, which performs the ESPIRiT calibration. The *soft-weighting of the singular vectors* option, `-W`, in the `ecalib` command is enabled to create tighter sensitivity maps [28].
  A density compensation function is also calculated here using the Ram Lak filter (see, [6, p.34]). Still, different from the gridding approach, the squared root of those weights is used as a preconditioner to accelerate the convergence in the pics reconstruction [29] instead of directly weighting the data samples.
  With all that information at hand, the `pics` command is called, using an in-plane

l1-wavelet regularization. Finally, the images are cropped to eliminate the excess of FOV due to the 2x oversampling if necessary.

- **Nlinv:** Unlike the pics method for which the sensitivity maps are required prior to the start of the iterative linear reconstruction process, nlinv jointly calculates the sensitivity maps and the images in a nonlinear iterative reconstruction process [30]. Therefore only the input data and the trajectory are needed to call the `nlinv` command to reconstruct the images. No density compensation of the data is necessary for this method.

3.2 Reconstruction Pipeline for the 3D Koosh-Ball Sequence

The reconstruction pipeline is shown in Fig. 3.2. It performs a full 3D reconstruction of the measured data, and it consists of three stages explained next.

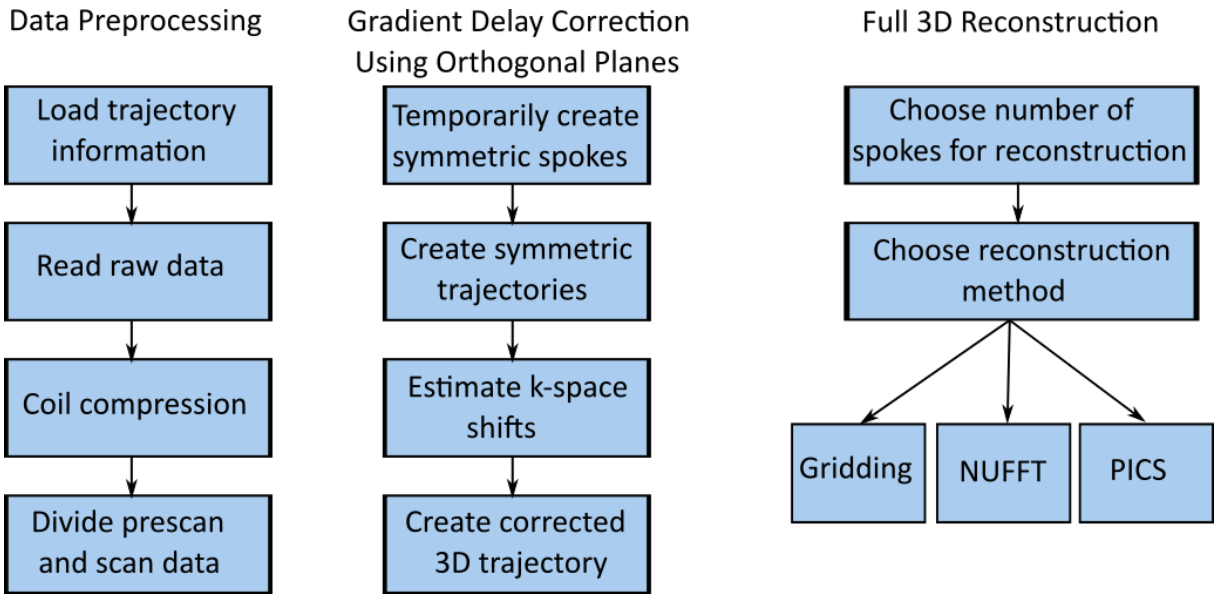

Figure 3.2: Reconstruction Pipeline for the Koosh-Ball Sequence

.

Data preprocessing Similar to the first three steps of the previous pipeline, the necessary sequence and trajectory information that was collected during the sequence definition is loaded, and the raw data in .dat format is read using the `twixread` command. If desired, coil compression can be performed as well. There is no need to rearrange the raw data in this case, but the prescan data is separated from the actual scan data, as only the prescan data is needed for the next stage to calculate the gradient delay errors.

Gradient Delays Correction The AC-Adaptive method was used to estimate the gradient-delay-induced k-space shifts. The three orthogonal planes acquired during the

prescan can be treated as partitions. Therefore the k-space shifts are calculated following steps one to three for the gradient-delays-correction stage from the previous pipeline. From the plane x-y, S_x, S_y and S_{xy} are obtained. From the plane z-x, S_z, S_x and S_{xz} are obtained. From the plane z-y, S_z, S_y and S_{yz} are obtained. Since two estimated values for S_x, S_y and S_z are obtained, those are averaged to get a better result. Finally, with those six values at hand, a 3D asymmetric trajectory is created using the `traj` command with the -q option to provide the S_x, S_y and S_{xy} values, -Q option to provide the S_z, S_{xz} and S_{yz} values, and the -O option to correct the gradient delay errors in the 3D radial trajectory.

Full 3D Reconstruction Here again, if the 2D golden means strategy is chosen as the angular ordering, the user can select a custom number of spokes to reconstruct the data with certain temporal resolution and at different time points [12]. A full 3D reconstruction can then be performed using three methods: gridding, NUFFT, and Pics.

- **Gridding:** a squared version of the Ram-Lak filter (see, [6, p.34]) is used in 3D gridding to density compensates the data and avoid blurred images. After that, a 3D adjoint NUFFT is applied to the data. Finally, the final image is cropped in the three dimensions to eliminate the excess of FOV due to the 2x oversampling.

- **NUFFT:** no density compensation is used here. A 3D inverse NUFFT is performed. The final image is cropped in the three dimensions to eliminate the excess of FOV due to the 2x oversampling.

- **Pics:** The procedure is the same as in the previous pipeline. For pics reconstruction, providing the squared root of the weights using the Ram Lak (see, [6, p.34]) as a preconditioner is still sufficient to get images with good quality. Also the l1-wavelet regularization is applied in the three spatial dimensions.

Although it is possible to use the nlinv method for 3D reconstructions, the memory demand for nlinv may be too high.

Bibliography

[1] Maxim Zaitsev. Pulseq interpreter module, July 2019.

[2] Siemens Healthineers. Magnetom terrathe first 7t for clinical use. `https://www.siemens-healthineers.com/en-us/magnetic-resonance-imaging/7t-mri-scanner/magnetom-terra`. (Visited on 1/21/2021).

[3] Zhi-Pei Liang and Paul C Lauterbur. *Principles of magnetic resonance imaging: a signal processing perspective*. SPIE Optical Engineering Press, 2000.

[4] Evgeniy N Ivanov, Alexander Y Pogromsky, Johan S Van Den Brink, and Jacobus E Rooda. Optimization of duty cycles for mri scanners. *Concepts in Magnetic Resonance Part B: Magnetic Resonance Engineering*, 37(3):180–192, 2010.

[5] Alexey S Peshkovsky, J Forguez, L Cerioni, and Daniel Jose Pusiol. Rf probe recovery time reduction with a novel active ringing suppression circuit. *Journal of Magnetic Resonance*, 177(1):67–73, 2005.

[6] Kai Tobias Block. Advanced methods for radial data sampling in magnetic resonance imaging. *SUB University of Goettingen*, 2008.

[7] Y Zur, ML Wood, and LJ Neuringer. Spoiling of transverse magnetization in steady-state sequences. *Magnetic resonance in medicine*, 21(2):251–263, 1991.

[8] Matt A Bernstein, Kevin F King, and Xiaohong Joe Zhou. *Handbook of MRI pulse sequences*. Elsevier, 2004.

[9] Robert W Brown, Y-C Norman Cheng, E Mark Haacke, Michael R Thompson, and Ramesh Venkatesan. *Magnetic resonance imaging: physical principles and sequence design*. John Wiley & Sons, 2014.

[10] Jochen Leupold, Jürgen Hennig, and Klaus Scheffler. Moment and direction of the spoiler gradient for effective artifact suppression in rf-spoiled gradient echo imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 60(1):119–127, 2008.

[11] Stefan Wundrak, Jan Paul, Johannes Ulrici, Erich Hell, and Volker Rasche. A small surrogate for the golden angle in time-resolved radial mri based on generalized fibonacci sequences. *IEEE transactions on medical imaging*, 34(6):1262–1269, 2014.

[12] Stefanie Winkelmann, Tobias Schaeffter, Thomas Koehler, Holger Eggers, and Olaf Doessel. An optimal radial profile order based on the golden ratio for time-resolved mri. *IEEE transactions on medical imaging*, 26(1):68–76, 2006.

[13] H Speier and Michael S Hansen. A golden angle of 68.75 improves gradient spoiling in radial gre. In *Proceedings of the 22nd Annual Meeting of ISMRM, Milan, Italy*, page 4245, 2014.

[14] Stefan Wundrak, Jan Paul, Johannes Ulrici, Erich Hell, Margrit-Ann Geibel, Peter Bernhardt, Wolfgang Rottbauer, and Volker Rasche. Golden ratio sparse mri using tiny golden angles. *Magnetic resonance in medicine*, 75(6):2372–2378, 2016.

[15] Ziwu Zhou, Fei Han, Lirong Yan, Danny JJ Wang, and Peng Hu. Golden-ratio rotated stack-of-stars acquisition for improved volumetric mri. *Magnetic resonance in medicine*, 78(6):2290–2298, 2017.

[16] Li Feng. Golden-angle radial mri: Basics, advances, and applications. *Journal of magnetic resonance imaging : JMRI*, 2022.

[17] C Preibisch and R Deichmann. Influence of rf spoiling on the stability and accuracy of t1 mapping based on spoiled flash with varying flip angles. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 61(1):125–135, 2009.

[18] Gary H Glover, John M Pauly, and Kenneth M Bradshaw. Boron-11 imaging with a three-dimensional reconstruction method. *Journal of Magnetic Resonance Imaging*, 2(1):47–52, 1992.

[19] Markus Deserno. How to generate equidistributed points on the surface of a sphere. *If Polymerforshung (Ed.)*, 99, 2004.

[20] Rachel W Chan, Elizabeth A Ramsay, Charles H Cunningham, and Donald B Plewes. Temporal stability of adaptive 3d radial mri using multidimensional golden means. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 61(2):354–363, 2009.

[21] Peter G Anderson. Linear pixel shuffling for image processing: an introduction. *Journal of Electronic Imaging*, 2(2):147–155, 1993.

[22] Andrew V Barger, Walter F Block, Yuriy Toropov, Thomas M Grist, and Charles A Mistretta. Time-resolved contrast-enhanced imaging with isotropic resolution and broad coverage using an undersampled 3d projection trajectory. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 48(2):297–305, 2002.

[23] Markus Untenberger, Zhengguo Tan, Dirk Voit, Arun A Joseph, Volkert Roeloffs, K Dietmar Merboldt, Sebastian Schätz, and Jens Frahm. Advances in real-time phase-contrast flow mri using asymmetric radial gradient echoes. *Magnetic resonance in medicine*, 75(5):1901–1908, 2016.

[24] Sebastian Rosenzweig, H Christian M Holme, and Martin Uecker. Simple auto-calibrated gradient delay estimation from few spokes using radial intersections (ring). *Magnetic resonance in medicine*, 81(3):1898–1906, 2019.

[25] Feng Huang, Sathya Vijayakumar, Yu Li, Sarah Hertel, and George R Duensing. A software channel compression technique for faster reconstruction with many channels. *Magnetic resonance imaging*, 26(1):133–141, 2008.

[26] Peter B Roemer, William A Edelstein, Cecil E Hayes, Steven P Souza, and Otward M Mueller. The nmr phased array. *Magnetic resonance in medicine*, 16(2):192–225, 1990.

[27] Martin Uecker, Peng Lai, Mark J Murphy, Patrick Virtue, Michael Elad, John M Pauly, Shreyas S Vasanawala, and Michael Lustig. Espirit—an eigenvalue approach to autocalibrating parallel mri: where sense meets grappa. *Magnetic resonance in medicine*, 71(3):990–1001, 2014.

[28] SS Iyer, F Ong, and M Lustig. Towards a parameter free espirit: Soft weighting for robust coil sensitivity estimation. In *Proc. Intl. Soc. Mag. Reson. Med*, volume 24, page 1093, 2016.

[29] Frank Ong, Martin Uecker, and Michael Lustig. Accelerating non-cartesian mri reconstruction convergence using k-space preconditioning. *IEEE transactions on medical imaging*, 39(5):1646–1654, 2019.

[30] Martin Uecker, Thorsten Hohage, Kai Tobias Block, and Jens Frahm. Image reconstruction by regularized nonlinear inversion—joint estimation of coil sensitivities and image content. *Magnetic Resonance in Medicine*, 60(3):674–682, 2008.