Introduction to the Shinnar-Le Roux algorithm.

Abstract

This was the basis of a talk I gave at GE Corporate Research and Development in 1995. It may need some revamping (and minor typo corrections), but as it is written in (plain) TeX .. I feel too tired and too old for that.

The SU2 representation of rotations.

The *SLR* algorithm uses the SU2 representation rather than the classical O3 (3×3 real matrix which are orthogonal) representation. The SU2 group is equivalent to the quantum mechanics representation of a 2 states system.

A vector in 3D space is generally represented by by a 1 column, 3 rows matrix of its 3 coordinates:

$$V = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

In SU2 we will represent this same vector by a complex 2×2 matrix:

$$V = \begin{pmatrix} z & x - jy \\ x + jy & -z \end{pmatrix} = \begin{pmatrix} z & X^* \\ X & -z \end{pmatrix} \tag{1}$$

where $j = \sqrt{-1}$.

One certainly does not see any simplification in the representation of vectors in SU2. But it is the representation of **rotations** which is simpler in SU2: a rotation is represented by a 2×2 matrix, as vectors are, but with a different structrure:

$$R = \begin{pmatrix} \alpha & -\beta^* \\ \beta & \alpha^* \end{pmatrix} \qquad \text{with} \quad |\,\alpha\,|^2 + |\,\beta\,|^2 = 1 \tag{2}$$

Where α and β are two complex values related to the 3 rotation angles defining the rotation (this relationship will be seen shortly). And the effect of rotation R on a vector V is defined by:

$$V^+ = RVR^* \tag{3}$$

where R^* is the transpose conjugate of R

If we develop (3) with the deinition (1) and (2) we find that V^+ is indeed a matrix of type (1), and is defined by the two quantities:

$$\begin{aligned} X^+ &= 2\alpha^*\beta z + \alpha^{*2}X - \beta^2 X^* \\ z^+ &= (|\,\alpha\,|^2 - |\,\beta\,|^2)z - \alpha^*\beta^* X - \alpha\beta X^* \end{aligned} \tag{3'}$$

This is the Bloch equation without relaxation. Yes. The next section will be devoted to make you accept this assertion.

Rewriting the Bloch Equation.

Equation (3) may seem hard to swallow, as is actually the definition of the density matrix in quantum mechanics which gives the link between the quantum state and the measurable quantities. To convince you, we will seek the expression of α and β for simple (and then general) rotations, starting from the O3 (classical) writting.

•Rotation around z.

Let us recall how we write normally the effect of rotations of a 3D vector, begining by a rotation of angle φ around z.

$$\begin{cases} x^+ = x\,cos\,\varphi + y sin\varphi \\ y^+ = -x\,sin\,\varphi + y\,cos\,\varphi \\ z^+ = z \end{cases}$$

Or, noting $X = x + jy$:

$$\begin{cases} X^+ = e^{j\varphi} X \\ z^+ = z \end{cases} \tag{4}$$

We can verify that (4) can be put into the form of (3) with the matrix R being:

$$R_z(\varphi) = \begin{pmatrix} e^{-j\frac{\varphi}{2}} & 0 \\ 0 & e^{j\frac{\varphi}{2}} \end{pmatrix} \tag{4'}$$

(note that R is always undetermined by an arbitrary phase factor:$e^{j\psi}$, ψ being any value).

•Rotation around y.

Let us continue by a rotation of angle θ around y

$$\begin{cases} x^+ = z\,sin\,\theta + x\,cos\,\theta \\ y^+ = y \\ z^+ = z\,cos\,\theta - x\,sin\,\theta \end{cases}$$

Or, after expressing x,y in terms of X and X^*:

$$\begin{cases} X^+ = z\,sin\,\theta + \dfrac{X}{2}(cos\,\theta + 1) + \dfrac{X^*}{2}(cos\,\theta - 1) \\ z^+ = z\,cos\,\theta - \dfrac{X + X^*}{2}\,sin\,\theta \end{cases}$$

And, here is the important step, we make use of classical trigonometric relations to express the sine and cosine of θ in terms of the sine and cosine of $\theta/2$:

$$c = cos\,\theta/2$$

$$s = sin\,\theta/2$$

we obtain:

$$\begin{cases} X^+ = 2csz + c^2X - s^2X^* \\ z^+ = (c^2 - s^2)z - cs(X + X^*) \end{cases} \tag{5}$$

Once more one verifies that (5) can be put in the form of (3) by taking R to be:

$$R_y(\theta) = \begin{pmatrix} c & -s \\ s & c \end{pmatrix} \tag{5'}$$

Generalization

•Rotation along x.

Having demonstrate the equivalence of (3) and the Bloch equation for the two cases of rotation around z and rotation around y we have normally finished and could argue that (3) is equivalent of the Bloch equation for **any** rotation. Indeed: one can always decompose a rotation in a succession of rotations along two orthogonal axisi, for instance z and y whose expressions we know already. Let us try to express a rotation aroun x of angle θ: it can be performed by a rotation of the object 90° around z (putting the x axis of the object along the y axis of the laboratory frame) followed by a rotation of angle θ around the y axis of the laboratory frame, followed by a $-90°$ rotation around z (putting back the x axis of the object at its original place):

$$\mathcal{R}_x(\theta) = \mathcal{R}_z(-\frac{\pi}{2}) \circ \mathcal{R}_y(\theta) \circ \mathcal{R}_z(\frac{\pi}{2})$$

where $\mathcal{R}$ represent operations, independently of the formalism we choose to write them (O3 or SU2). In the SU2 formalism, using our previous results (4') and (5')i, we obtain the matrix representation:

$$R_x(\theta) = \begin{pmatrix} c & js \\ js & c \end{pmatrix} \tag{7}$$

Being a little more ambitious, one could seek the expression, in O3 or SU2, of a rotation of angle θ around an axis around an axis $\vec{u}$ contained in the $\{x, y\}$ plane making an angle φ with the axis y, $\varphi = \angle(\vec{y}, \vec{u})$) : we would rotate first the object $-\varphi$ around z to put the rotation axis along y, perform the rotation θ around the the y axis of the laboratory frame, then put back the rotation axis $\vec{u}$ at its original place by a rotation φ around z.

$$\mathcal{R}_\varphi(\theta) = \mathcal{R}_z(\varphi) \circ \mathcal{R}_y(\theta) \circ \mathcal{R}_z(-\varphi) \tag{6}$$

Once again, using (4') and (5'), we could easily write the SU2 matrix representation of this rotation. But why not tackcle directly with the expression of a general rotation? It will always be time to specialize the result after that.

•**General rotation** $\mathcal{R}_z(\varphi_2) \circ \mathcal{R}_y(\theta) \circ \mathcal{R}_z(\varphi_1)$.

Indeed any rotation can be put into this form: a rotation of angle φ_1 around z, followed by a nutation θ around y, followed by another precession (rotation around z) of angle φ_2.

Here is the list of operations we have to do, using the definition (3):

$$V^+ = \underbrace{\underbrace{\underbrace{R_z(\varphi_2)\, R_y(\theta)}_{\mathcal{R}_y(\theta)\circ\mathcal{R}_z(\varphi_1)}\, \underbrace{R_z(\varphi_1) V R_z^*(\varphi_1)}_{\mathcal{R}_z(\varphi_1)}}_{\mathcal{R}_z(\varphi_2)\circ\mathcal{R}_y(\theta)\circ\mathcal{R}_z(\varphi_1)}\, R_y^*(\theta)\, R_z^*(\varphi_2)}$$

If we keep the grouping as shown by the braces, we would be using in turn (4) and (5) i.e. the O3 writing. We will instead group the operations this way:

$$V^+ = \underbrace{R_z(\varphi_2) R_y(\theta) R_z(\varphi_1)}_{R}\, V R_z^*(\varphi_1) R_y^*(\theta) R_z^*(\varphi_2)$$

i.e. first obtain the expression of the SU2 general rotation matrix, disregarding all together the initial vector V. Then for any vector V, find the transform vector V^+ by using (3) (if we think this is really useful). The expression of R is easily obtained from (4')(5') :

$$R = \begin{pmatrix} c e^{-j\frac{\varphi_1+\varphi_2}{2}} & -s e^{-j\frac{\varphi_2-\varphi_1}{2}} \\ s e^{j\frac{\varphi_2-\varphi_1}{2}} & c e^{j\frac{\varphi_1+\varphi_2}{2}} \end{pmatrix} \tag{7}$$

i.e. it is the general form of SU2 rotation (2) with:

$$\begin{cases} \alpha = c e^{-j\frac{\varphi_1+\varphi_2}{2}} \\ \beta = s e^{j\frac{\varphi_2-\varphi_1}{2}} \end{cases} \tag{8}$$

•**The special case of rotation with axis in the xy plane.**

The special case of a rotation around an axis in the (x, y) plane (6) is obtained by putting in (8) $\varphi_2 = -\varphi_1 = \varphi$. We have then:

$$\alpha = c$$

$$\beta = s e^{j\varphi}$$

Note that the phase (argument) of α is null. This is the characteristic of a rotation in the x, y plane. The argument of β is simply the angle between the y axis and the axis of the rotation. As this kind of rotation is very special in MR (B_1 field is in the xy plane, or rather its longitudinal components has no substancial effect) we will use a special notation which is a generalization of the rotation around y (5')

$$\begin{aligned} R_\varphi(\theta) &= \begin{pmatrix} c & -s^* \\ s & c \end{pmatrix} \\ s &= e^{j\varphi}\, cos\, \frac{\theta}{2} \\ c &= cos\, \frac{\theta}{2} \end{aligned} \tag{10}$$

•**Sequence of rotations** Let us now write the combination of two general rotations. Note that doing that in O3 (multiplication of two 3x3 real matrices) begins to lead to an untractable result.

So let us suppose we are given two rotations $\mathcal{R}_1$,$\mathcal{R}_2$ whose SU2 matrices R_1 and R_2 are characterized by the couples α_1, β_1 and α_2, β_2 respectively. By definition (3), the list of operations to perform is:

$$V^+ = R_2 \overbrace{\overbrace{R_1 V R_1^*}^{\mathcal{R}_1}}^{\mathcal{R}_2 \circ \mathcal{R}_1} R_2^*$$

Which we group into:

$$V^+ = \underbrace{R_2 R_1}_{R} V R_1^* R_2^* \tag{9}$$

The SU2 result matrix R is:

$$R = \begin{pmatrix} \alpha_2 & -\beta_2^* \\ \beta_2 & \alpha_2^* \end{pmatrix} \times \begin{pmatrix} \alpha_1 & -\beta_1^* \\ \beta_1 & \alpha_1^* \end{pmatrix} = \begin{pmatrix} \alpha_2\alpha_1 - \beta_2^*\beta_1 & -\alpha_2\beta_1^* - \beta_2^*\alpha_1^* \\ \beta_2\alpha_1 + \alpha_2^*\beta_1 & -\beta_2\beta_1^* + \alpha_2^*\alpha_1^* \end{pmatrix} \tag{9'}$$

where we verify that R is indeed in the form of (2) which was expected as we proclaimed from the start the SU2 matrices formed a group, and is determined by the two components:

$$\begin{aligned} \alpha &= \alpha_2\alpha_1 - \beta_2^*\beta_1 \\ \beta &= \beta_2\alpha_1 + \alpha_2^*\beta_1 \end{aligned} \tag{9''}$$

Some will even prefer to speak of *Spinors* rather than SU2 rotation matrices, as these matrices are defined by only two complex numbers. A spinor is a mathematical being composed of two complex numbers (α, β) with the constrains $\mid \alpha \mid^2 + \mid \beta \mid^2 = 1$. We define the multiplication between two spinors as:

$$(\alpha_2, \beta_2) \times (\alpha_1, \beta_1) = (\alpha_2\alpha_1 - \beta_2^*\beta_1, \beta_2\alpha_1 + \alpha_2^*\beta_1)$$

The SLR algorithm

The direct SLR recursion

Consider a train of hard pulses $(\theta_0, \varphi_0), (\theta_1, \varphi_1)$ ....where θ_i is the nutation angle of the pulse number i, φ_i the phase angle of the rotation axis in the x_1, y_1 plane. The period separating two pulses is of length T, and during each of these periods, a given proton experiences a constant precession (i.e rotation around z) of angle $-\omega T$ (the minus sign is here to take into account that protons rotate in the negative direction and that ω generally stands for the angular resonance frequency).

Let us apply now the principles outlined in the preceeding paragraph.

The effect of such a sequence of pulses writes formally:

$$\mathcal{R}_n = \mathcal{R}_{\varphi_n}(\theta_n) \circ \mathcal{R}_z(-\omega T) \circ \mathcal{R}_{\varphi_{n-1}}(\theta_{n-1}) \circ \mathcal{R}_z(-\omega T) \cdots \circ \mathcal{R}_{\varphi_1}(\theta_1) \circ \mathcal{R}_z(-\omega T) \circ \mathcal{R}_{\varphi_0}(\theta_0)$$

Writing now explicitly the product of rotation matrix in the SU2 representation:

$$\begin{aligned} R_n &= \begin{pmatrix} c_n & -s_n^* \\ s_n & c_n \end{pmatrix} \begin{pmatrix} Z^{1/2} & 0 \\ 0 & Z^{-1/2} \end{pmatrix} \underbrace{\begin{pmatrix} c_{n-1} & -s_{n-1}^* \\ s_{n-1} & c_{n-1} \end{pmatrix} \cdots \begin{pmatrix} Z^{1/2} & 0 \\ 0 & Z^{-1/2} \end{pmatrix} \begin{pmatrix} c_0 & -s_0^* \\ s_0 & c_0 \end{pmatrix}}_{R_{n-1}} \\ \begin{pmatrix} \alpha_n & -\beta_n^* \\ \beta_n & \alpha_n^* \end{pmatrix} &= \begin{pmatrix} c_n & -s_n^* \\ s_n & c_n \end{pmatrix} \begin{pmatrix} Z^{1/2} & 0 \\ 0 & Z^{-1/2} \end{pmatrix} \begin{pmatrix} \alpha_{n-1} & -\beta_{n-1}^* \\ \beta_{n-1} & \alpha_{n-1}^* \end{pmatrix} \end{aligned} \tag{11}$$

Where the notation conforms to the precedding paragraph and with :

$$Z = e^{j\omega T}$$

From (11), by keeping only the two components of the first column α and β of the rotation matrices, one deduces the recursive equations :

$$\begin{aligned} \alpha_n &= c_n Z^{1/2} \alpha_{n-1} - s_n^* Z^{-1/2} \beta_{n-1} \\ \beta_n &= s_n Z^{1/2} \alpha_{n-1} + c_n Z^{-1/2} \beta_{n-1} \end{aligned} \tag{12}$$

with:

$$\alpha_0 = c_0$$

$$\beta_0 = s_0$$

One can obtain a recursive equations where only integer power of Z appear, by letting:

$$\begin{aligned} \alpha_n &= Z^{n/2} a_n \\ \beta_n &= Z^{n/2} b_n \end{aligned} \tag{13}$$

Then (12) writes:

$$\begin{aligned} a_n &= c_n a_{n-1} - s_n^* Z^{-1} b_{n-1} \\ b_n &= s_n a_{n-1} + c_n Z^{-1} b_{n-1} \\ a_0 &= c0 \\ b_0 &= s_0 \end{aligned} \tag{14}$$

Or, in matrix form:

$$\begin{bmatrix} a_n \\ b_n \end{bmatrix} = \begin{pmatrix} c_n & -s_n^* \\ s_n & c_n \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & Z^{-1} \end{pmatrix} \begin{bmatrix} a_{n-1} \\ b_{n-1} \end{bmatrix} \tag{14'}$$

It is easily seen that after n+1 pulses the quantities a_n and b_n are polynomials of order n in the variable Z^{-1}. For instance after the pulse number 1:

$$a_1 = c_1 c_0 - s_1^* s_0 Z^{-1}$$

$$b_1 = s_1 c_0 + c_1 s_0 Z^{-1}$$

After the pulse number 2:

$$a_2 = c_2 c_1 c_0 - (s_1^* s_0 - s_1 s_2^*) Z^{-1} - c_1 s_0 s_2^* Z^{-2}$$

$$b_2 = c_1 c_0 s_2 + (c_0 c2 s_1 - s_2 s_1 * s_0) Z^{-1} + c_2 c_1 s_0 Z^{-2}$$

and so on...with expressions of the coefficients more and more complicated, but the recursion can still be tackled numerically on a computer.

Optional

There are two possible visions of the recursion (14). The first immediate vision it to see in the frequency domain: pointwise, at the angular frequency ω the two complex values a,b (directly related to α, β by (13)) evolve from pulse to pulse according to a rotation given by (14). This rotation depends on the RF pulse (by c_n,s_n) and on the frequency offset (by $Z = e^{j\omega T}$). Note that for a frequency $\omega' T = \omega T + 2\pi$ the evolution is the same. That means

that $a(\omega)$,$\underline{(}\omega)$ are periodic functiuns of ω. The second vision corresponds to take the Fourier transform of of $a(\omega)$,$b(\omega)$ (this is similar in MR imaging where $\omega \leftrightarrow x$ to take the k space representation).

$$a(\tau) = \frac{1}{2\pi}\int a(\omega)d\omega$$
$$b(\tau) = \frac{1}{2\pi}\int b(\omega)d\omega$$

The fact that a_n and b_n are polynomials of order n in the variable Z^{-1} is equivalent to say that $a_n(\tau)$,$b_n(\tau)$ are comb filters (composed of Dirac pulses) with period T and whose Dirac time-integrals are the coefficients of the polynomial. In short $a_n b_n$ are digital FIR (Finite Impulse Response) filters of length $n+1$. Although not essential, I think interesting to look at the recursion (14) in the time (τ) domain, i.e. to follow the individual coefficients of the polynomial.

After pulse n-1:

$$a^0_{n-1} \quad a^1_{n-1} \quad \cdots a^{n-2}_{n-1} \quad a^{n-1}_{n-1}$$
$$b^0_{n-1} \quad b^1_{n-1} \quad \cdots b^{n-2}_{n-1} \quad b^{n-1}_{n-1}$$

Just before pulse n: $b(\omega)$ has been multiplied by Z^{-1}, i.e. in the time domain, its coefficients have been shifted one place to the right:

$$a^0_{n-1} \quad a^1_{n-1} \quad \cdots a^{n-2}_{n-1} \quad a^{n-1}_{n-1}$$
$$0 \quad b^0_{n-1} \quad \cdots b^{n-3}_{n-1} \quad b^{n-2}_{n-1} \quad b^{n-1}_{n-1}$$

After pulse n: the pulse have combined the two previous signal according to c_n,s_n

$$c_n a^0_{n-1} \quad (c_n a^1_{n-1} - s^*_n b^0_{n-1}) \cdots\cdots (c_n a^{n-1}_{n-1} - s^*_n b^{n-2}_{n-1}) \quad - s^*_n b^{n-1}_{n-1}$$
$$s_n a^0_{n-1} \quad (s_n a^1_{n-1} + c_n b^0_{n-1}) \cdots\cdots (s_n a^{n-1}_{n-1} + c_n b^{n-2}_{n-1}) \quad c_n b^{n-1}_{n-1}$$

Note how the leading coefficients a^0_n,b^0_n and trailing coefficients a^n_n,b^n_n are simple:

$$\begin{aligned} a^0_n &= c_n a^0_{n-1} \quad a^n_n = -s^*_n b^{n-1}_{n-1} \\ b^0_n &= s_n a^0_{n-1} \quad b^n_n = c_n b^{n-1}_{n-1} \end{aligned} \tag{15}$$

The SLR inverse recursion

Equation (12) (or (13)(14)) permits to find the α, β (hence to characterize fully the rotation) for any sequence of RF pulses, and that for all angular frequency offset ω. In reverse we would like to know in what condition, two functions of frequency $\alpha(\omega)$,$\beta(\omega)$ could be generated by a train $n+1$ hard pulses, and what is this pulse train.

According to the preceeding section we know the necessary conditions, we have just to hope that these conditions are sufficient.

Theorem (sort of)

If two polynomials of oder m, $a_m(Z^{-1})$,$b_m(Z^{-1})$, $Z = e^{j\omega}$ are such that, for all ω

$$\mid a_m \mid^2 + \mid b_m \mid^2 = 1$$

then these polynomials can be generated by a recursion (14) with $n = 0..m$

So let us suppose we are given such two comb filters (polynomials of order m in Z^{-1}). We will try some recursive peeling off and ask the question: could a_m and b_m be the result of one pass of recursion (14)? i.e. can we find two polynomials of order $m-1$,a_{m-1},b_{m-1} and the parameter of one pulse (c_m,s_m) such that (14) becomes an equality?. As (14) is the expression of a rotation it is easily invertible: we can express a_{m-1},b_{m-1} in term of a_m,b_m.

$$\begin{bmatrix} a_{m-1} \\ b_{m-1} \end{bmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & Z \end{pmatrix} \begin{pmatrix} c_m & s_m^* \\ -s_m & c_m \end{pmatrix} \begin{bmatrix} a_m \\ b_m \end{bmatrix} \qquad (14'')$$

For general values of c_m,s_m, the quantities a_{m-1}, b_{m-1} will **not** be polynomials of order $m-1$: a_{m-1} will indeed contain a coefficient of order m:

$$a_{m-1}^m = c_m a_m^m + s_m^* b_m^m$$

and b_{m-1} will contain a coefficient of order -1 in Z^{-1}:

$$b_{m-1}^{-1} = -s_m a_m^0 + c_m b_m^0$$

Trying first to cancel this last coefficient b_{m-1}^{-1}, we find the last RF pulse by:

$$t_m = \frac{s_m}{c_m} = \frac{b_m^0}{a_m^0} \tag{16}$$

But we would get a problem if this value of t_m cannot cancel the coefficient a_{m-1}^m:

$$t_m^* = \frac{s_m^*}{c_m} = \frac{-a_m^m}{b_m^m}$$

So we must absolutely have:

$$b_m^0 b_m^{n*} + a_m^0 a_m^{n*} = 0$$

or the problem has no solution. But recall that the frequency spectrum of $f(t)$, $\mid F(\omega) \mid^2$ is the Fourier transform of the autocorrelation function of $f(t)$.

Translating $\mid a_m(\omega) \mid^2 + \mid b_m(\omega) \mid^2 = 1$ in the time domain, we get:

$$\sum_{i=0}^{i=m} \mid a_m^i \mid^2 + \mid b_m^i \mid^2 = 1$$

For the autocorrelation coefficients of lag 0; then, some more complicated expressions for lag 1,2,3.. But the interesting lag is m:

$$b_m^0 b_m^{m*} + a_m^0 a_m^{m*} = 0$$

So there is one and one only solution t_m; the two expressions of t_m given above are indeed equivalent (theorytically, numerically we will use the first expression (16). More on that later).

Having determined t_m we use the inverse recursion (14″) and obtain two polynomials of order $m-1$. As (14") correspond to a rotation for any ω we have:

$$\mid a_{m-1}(\omega) \mid^2 + \mid b_{m-1}(\omega) \mid^2 = 1$$

So we can reiterate the same reasonning after having change m to $m-1$. And so on, down to $m = 0$. The sequence of pulses $\{t_n; n = m..0\}$ is thus determined.

The minimum phase a polynomial gives the RF train with minimum energy

We have now to find a way to design polynomial of a given order (for a given pulse train length) which give the desired action on the proton. For instance for a 90° flip selective pulse we would like the rotation $(\alpha(\omega), \beta(\omega))$ to be as close as possible to 90° along y, $\alpha = 1/\sqrt{(2)}, \beta = 1/\sqrt{(2)}$ in a given band (pass-band) and a null rotation $\alpha = 1, \beta = 0$ in a given stop-band. One could think to try to fit FIR filters to both desired function $\alpha(\omega), \beta(\omega)$. But one would face, then, two stumbling blocks:

• If there exist classical FIR filter design algorithm, they are aimed at the design of one filter, and none can take into account the nonlinear constraint $\mid a_m(\omega) \mid^2 + \mid b_m(\omega) \mid^2 = 1$ which we know to be crucial for the inverse SLR recursion.

• If one succeeds one way or the other to have two polynomials matching the the desired functions α, β, chances are great that the pulse train one obtains will contain very large pulses (possibly some at 180°).

The explanation of this, lies in the fact that one cannot choose freely the a polynomial, or one has to be prepared to pay the price in RF power deposition and maximum amplitude.

Thus, we will suppose that we have designed the b polynomial. Then, the power spectrum of $a(\omega)$ is given. The only degree of freedom left is the phase of $a(\omega)$. We will show, that among all pulse trains giving $b(\omega)$ the one with minimum energy is the one for which $a(\omega)$ is minimum phase. That is an happy result because in order to determine the phase of a FIR filter of given power spectrum one has to seek for its zeroes (a very lengthy, imprecise procedure), **except** in the case of the minimum-phase solution. In this case a very fast, direct procedure exists. We recall also that the minimum-phase polynomial has a leading term in time a_n^0 maximum.

Optional

The minimum phase polynomial has all its zeroes inside the unit-circle:

$$a_n(Z) = \prod_{i=0}^{i=n}(1 - z_i Z^{-1}) \qquad | \ z_i \ | < 1$$

If we take the *log* of a, each term is decomposable into a convergent series in Z^{-1}:

$$log(1 - z_i Z^{-1}) = -z_i/Z + 1/2(zi/Z)^2 \cdots$$

(reversely if all zeros were outside the unit circle, $log(a)$ would be decomposable in a convergent series in Z).

Thus the Inverse Fourier transform of $log(a_n(\omega))$ (called the cepstrum!) is a causal signal.

We are given $a_(\omega)a^*(\omega)$, we take the *log*. We take the Inverse Fourier transform. The minimum phase part $a_(\omega)$ gives a causal signal, whereas the maximum phase part gives an anticausal signal. We can thus isolate the minimum-phase part on the cepstrum by nulling the negative time part. We do the Fourier transform and and inverse the *log* to obtain $a(\omega)$

a_n^0 is maximum:

For each zero we have two choice for the corresponding monomial: wether the zero is inside the unit circle

$$1 - z_i Z^{-1}$$

or it is outside the unit circle:

$$(z_i^* - Z^{-1})$$

Switching from the first choice to the second choice we multiply the constant term, and hence the 0 order coefficient of the polynomial a_n^0, by z_i^*. As $| \ z_i \ | < 1$, a_n^0 diminishes. Thus the minimum phase polynomial is also the one with the largest leading term.

Using the recursion (14), it is immediate to find that:

$$a_n^0 = c_n c_{n-1} c_{n-2} \cdot \cdots \cdot c_2 c_1 c_0$$

Note that $-\sum log(cos(\theta_i/2))$ is a good loss function on its own right (it is positive, increasing). As the minimum-phase polynomial has a a_n^0 maximum, it is the one which minimizes that loss function. But in the case one can suppose that each nutation angle θ_i small we can relate this loss-function to the RF power.

$$\prod_{i=0}^{i=n} c_i \approx \prod_{i=0}^{i=n} (1 - \frac{\theta_i^2}{8}) \approx 1 - \sum_{i=0}^{i=n} \frac{\theta_i^2}{8} \tag{18}$$

QED

One notes that the fact that a_n^0 is maximum is benificial also for numerical reasons when using (16).

There are some instance when one will use non minimum-phase a_n polynomial. Well, there is at least one such instance: it is when designing self-refocusing pulse. Self-refocusing (flip) pulses try to compensate the phase of b by the phase of a. One then has to switch the zeroes of a. At any rate, even in this case one would start from the minimum-phase solution and access the cost in power of switching one zero by (18).

The b polynomials and real world specifications

From now on the b polynomial is our only degree of freedom. We have now to specify b or β out of what we want the RF pulse to perform. We only scan rapidly this subject, because it is exposed with much details in [1].

One can distinguish, in usual applications: flip pulses, with low flip angle, or 90°, inversion pulses, saturation pulses, and refocusing pulses.

The common characteristic of the three first categories (flip,saturation,inversion) is that the magnetization before the pulse is aligned with the z axis. One can thus readily write, from

(3'), the magetization vector at the end of the RF pulse:

$$X^+ = 2\alpha^*\beta$$

$$z^+ = \mid \alpha \mid^2 - \mid \beta \mid^2$$

In the case of inversion pulse or saturation pulses one is interested only by the longitudinal magetization z, which can be written:

$$z^+ = 1 - 2\mid \beta \mid^2$$

This gives directly the specifications for designing β from the desired response $z^+(\omega)$. We use for that the *Parks McClellan (P.M.)* algorithm which is able to design equiripple optimum FIR filters. The ripples in the stop band and the pass-band can be adjusted, which is happy: if for instance one wants the same ripple in the pass-band and stop-band of z^+, the ripples of β must be different in the two bands. For the details of setting the ripples of β according to the ripples of the desired response see the article of Mach 91 [1].

In inversion, or saturation one does not care about the phase of β (because one does not care about the longitudinal magnetization X^+), so one could design by *P.M.* $\mid \beta \mid^2$: this permits better selectivity to be achieved. One determined arbitrarily the phase of β afterwards, making it for instance minimum or maximum phase.

For flip pulses we are interested by X^+. The amplitude specfication and ripples are deduced from:

$$\mid X^+ \mid = 2\sqrt{1 - \mid b \mid^2} \mid b \mid$$

We cannot really master the phase of X^+ as we dont master the phase of a. But the phase of a happens to be very close to linear, with a small slope and one can put in first approximation:

$$\angle X^+ = \angle \beta$$

If one design a symmetrical FIR β the RF pulse will be symmetrical (this can be deduced from (11), take the transpose – not conjugate transpose – of both sides); the phase of X^+ will be quasi-linear with an equivalent focus time sligthly before the center of the pulse (the

slight advance comes from the factor α^* in the expression of X^+). Here also one can make the pulse more selective by designing first $\mid \beta \mid^2$ and taking its phase as minimum-phase. For a given pulse length, the equivalent focus time is then retarded greatly, reducing the refocusing gradient period. Conversely, if one supposes it is this refocusing time which counts one can keep it constant and lengthen the pulse, obtaiming sharper profile. Be warned though, minimum-phase pulses may have subtanstially non-linear phase for high selectivity, and are more sensitive to flow.

The case of refocusing pulses is more complicated because we are not starting from one initial condition but from a set of initial conditions (a 'pancake' of magnetzation X). But if this would be a problem if one used the Bloch equation, it is not really one for SU2, which works directly with the rotation parameters. Nethertheless writing the Bloch equation equivalent (3') for an, unspecified, initial condition X, one gets:

$$X^+ = \alpha^{*2} X - \beta^2 X^*$$

$$z^+ = -\alpha^* \beta^* X - \alpha \beta X^*$$

Recall what we want to do with a refocusing pulse. We would like to change X into X^* (change the sign of the phase of the transverse magetization, in order to have the faster 'proton' become the most retarded and the slower one the more advanced). So the useful part of the rotation is β^2 which should be ideally 1 in amplitude (then, α is zero) and of constant phase along the axis ω if we want the whole slice to come into echo at the same time. We nethertheless have to take care of the parasistic term $\alpha^{*2} X$ which cannot be zero in the transition band of the slice because, there, β will shink from 1 (its ideal value in the pass-band) to 0 (its ideal value in the stop-band). This is the role of the crusher gradients. Lets make this remark first: looking at (8) one sees that if one adds a same value ψ to the 'predephasing' phase φ_1 and to the 'postdephasing' phase φ_2, β **does not change**, and the phase of α is increased by ψ. So given an RF refocusing pulse, one will, by adding equal crusher gradient on each side, increase the slope of the phase of α (and thus α^{*2}). If the crushers are large enough the phase variation of α^{*2} can correspond to several 2π

dephasing across the slice, nulling the parasistic term. The design of a refocusing pulse is then straigthforward: β is a symmetrical FIR filter that we design by the *P.M.* algorithm with ripples specifications adapted to the ripples we accept on β^2.

References

1 . J. PAULY, P. LE ROUX, D. NISHIMURA, A. MACOVSKI, *IEEE Trans. Med. Imaging* **10**, 53 (1991)

2 . French Patent 8610179,1986

3 . L.I.SCHIFF, "Quantum Mechanics,"McGraw-Hill,ISBN 0-07-085643-5.