Angewandte
Eine Zeitschrift der Gesellschaft Deutscher Chemiker
Chemie

Supporting Information

Enhancing the Sensitivity of CPMG Relaxation Dispersion to Conformational Exchange Processes by Multiple-Quantum Spectroscopy

*Tairan Yuwen, Pramodh Vallurupalli, and Lewis E. Kay**

anie_201605843_sm_miscellaneous_information.pdf

Supporting Information

Table of Contents

Materials and Methods

Details of TQ-CPMG pulse sequence. 90° (180°) rectangular pulses, denoted by narrow (wide) bars, are applied with a 25 kHz field between points A and B, and at maximum power elsewhere. All pulses are applied along the x-axis unless otherwise indicated. The hatched bars denote composite 180° pulses[1], as described below, while the water-selective shaped pulse marked with "w" (∼7 ms) is implemented using the EBURP-1 profile[2]. Note that a single 180_{ϕ_2} pulse is applied in the center of the CPMG scheme that decreases the effects of pulse imperfections, as described previously[3]. Between points A and B the ^{1}H carrier is placed in the center of the methyl region (∼1 ppm), while at point B the carrier is positioned at the water resonance (∼4.7 ppm); ^{13}C and ^{2}H carriers are at ∼20 ppm and ∼1 ppm, respectively. ^{13}C and ^{2}H WALTZ-16 decoupling elements[4] are applied with fields of 2 kHz and 500 Hz, respectively. The delays used are: $\tau_a = 1.8$ ms, $\tau_b = 1.8$–2 ms, $\tau_{cp} = T_{relax}/(4N)$, N is an integer. The phase cycle is $\phi_1 = (0°, 60°, 120°, 180°, 240°, 300°)$; $\phi_2 = y, -y$; $\phi_3 = 6(x), 6(-x)$; $\phi_4 = x$; $\phi_5 = x, -x$; $\phi_{rec} = 3(x,-x), 3(-x,x)$. Typically we record the complete phase cycle for H_2O samples, as the ^{13}C 90° pulse of phase ϕ_3 significantly decreases the residual water signal, while for D_2O samples a 6-step cycle is used ($\phi_3 = x$) so long as sensitivity is not limiting. The phases ψ_1/ψ_2 are used to implement the XY-16 scheme[5]. Phase ψ_1 is incremented with the cycle $(x,y,x,y,y,x,y,x,-x,-y,-x,-y,-y,-x,-y,-x)$ for each successive N value, with ψ_2 decremented in the same manner, but inverted. Thus, the CPMG element becomes

$$\tau_{cp}\Pi_1\tau_{cp}\tau_{cp}\Pi_2\tau_{cp}\ldots\tau_{cp}\Pi_{N-1}\tau_{cp}\tau_{cp}\Pi_N\tau_{cp}180_{\phi_2}\tau_{cp}\overline{\Pi}_N\tau_{cp}\tau_{cp}\overline{\Pi}_{N-1}\tau_{cp}\ldots\tau_{cp}\overline{\Pi}_2\tau_{cp}\tau_{cp}\overline{\Pi}_1\tau_{cp}$$

where $\Pi_1 = \Pi_x, \Pi_2 = \Pi_y, \Pi_3 = \Pi_x$ and so on following the XY-16 scheme, with $\Pi_x = 90_{-y}180_x90_{-y}$, $\Pi_y = 90_x180_y90_x$, $\Pi_{-x} = 90_y180_{-x}90_y$, $\Pi_{-y} = 90_{-x}180_{-y}90_{-x}$. $\overline{\Pi}_j$ values are calculated from Π_j by interchanging $y(-y)$ with $-y(y)$ in expressions for Π_j, that follows from the 180_{ϕ_2} pulse in the center of the pulse scheme as discussed by Hansen et al.[3] Quadrature detection in F_1 is achieved by STATES-TPPI of ϕ_4[6]. Gradients are applied with the following durations (ms) and strengths (G/cm): g_0: $(1.0, 4)$, g_1: $(0.4, 5)$, g_2: $(0.6, 6)$, g_3: $(0.3, -5)$, g_4: $(0.5, 10)$, g_5: $(0.3, -8)$.

Sample preparation. All proteins were expressed in *E. coli* BL21(DE3) cells grown in M9 minimal media ($\sim$100% D_2O), containing ^{15}N-ammonium chloride (1 g/L) and [^{2}H,^{12}C]-glucose (3 g/L) as the sole nitrogen and carbon sources, respectively. Specifically labeled precursors[7] were added 1 h prior to the induction of protein overexpression with 1 mM IPTG, as described previously[7,8]. Purification of samples followed literature protocols. In brief, the following samples were used: (i) 1.5 mM [U-^{2}H; Ileδ1-$^{13}CH_3$; Leu,Val-$^{13}CH_3$/$^{12}CD_3$; Met-$^{13}CH_3$]-labeled B1 domain of immunoglobulin binding protein G[9] in 50 mM potassium phosphate pH 7.5, 100 mM NaCl, 0.1 mM NaN_3, 100% D_2O; (ii) 1.5 mM [U-^{2}H; Ileδ1-$^{13}CH_3$; Leu,Val-$^{13}CH_3$/$^{12}CD_3$; Met-$^{13}CH_3$]- and 1.0 mM [U-^{2}H; Ileδ1-$^{13}CHD_2$; Leu,Val-$^{13}CHD_2$/$^{13}CHD_2$; Met-$^{13}CHD_2$]-labeled FF domain from human HYPA/FBP11[10] in 25 mM potassium phosphate pH 6.8, 50 mM NaCl, 1 mM EDTA, 100% D_2O; (iii) 0.9 mM (monomer concentration) [U-^{2}H; Ileδ1-$^{13}CH_3$; Leu,Val-$^{13}CH_3$/$^{12}CD_3$; Met-$^{13}CH_3$]-labeled $\alpha_7\alpha_7$, from *T. acidophilum*[11], 25 mM potassium phosphate pH 7.4, 50 mM NaCl, 4.6 mM NaN_3, 1 mM EDTA, 100% D_2O. The level of deuteration in all samples was $> 95\%$ at all carbon sites.

NMR spectroscopy. All NMR experiments were recorded on Varian 600 MHz and 800 MHz spectrometers equipped with triple-axis gradient room temperature probes. Data sets were measured at 25°C with the exception of $\alpha_7\alpha_7$ where dispersion profiles were recorded at 50°C. ^{1}H SQ $^{13}CH_3$ and $^{13}CHD_2$ relaxation dispersion data sets were recorded as described in the literature[12,13], while TQ profiles were measured using the pulse scheme described in the text. ^{1}H relaxation dispersion experiments have been recorded using a constant-time CPMG element[14,15] $T_{relax} = 40$ ms (GB1) or 20 ms (FF domain and $\alpha_7\alpha_7$). Experiments were recorded as pseudo-3D datasets by varying the number of CPMG pulses N during T_{relax} for each 2D plane, with the value of T_{relax} chosen such that peak intensities decay to $\sim$50% of their values for $T_{relax} = 0$. Each 2D data set was recorded with 6 scans/FID, a relaxation delay of 2.0 s and (576, 64) complex points in (t_2, t_1) to give a net acquisition time of $\sim$30 min/spectrum. A series of 2D maps was obtained with ν_{CPMG} values varying between 25 Hz ($T_{relax} = 40$ ms)

or 50 Hz ($T_{relax} = 20$ ms) and 2000 Hz. Approximately 30–40 planes were recorded for each dispersion series, including duplicates for error analysis[16]. The net measurement time for each experiment was ~15–20 h.

Data analysis. All NMR spectra were processed and analyzed using the NMRPipe suite of programs[17]. Effective transverse relaxation rates, $R_{2,eff}$, were calculated based on peak intensities according to the relation $R_{2,eff}(\nu_{CPMG}) = -\ln(I(\nu_{CPMG})/I_0)/T_{relax}$, where I_0 is the peak intensity in a reference spectrum recorded without the relaxation delay, T_{relax}[15]. Fitting of cross-peaks in CPMG data sets was carried out using the software package ChemEx (https://github.com/gbouvignies/chemex); a separate module is required for fitting TQ ^{1}H relaxation dispersion data, which is available upon request. Only residues with TQ(SQ) dispersion profiles such that $R_{ex} = R_{2,eff}(\nu_{CPMG} = 50\text{ Hz}) - R_{2,eff}(\nu_{CPMG} = 2000\text{ Hz}) > 7\text{ s}^{-1}(3\text{ s}^{-1})$ at 800 MHz, were included in the analyses. Exchange parameters were extracted from fits of dispersion data to a two-site exchange model using the Bloch–McConnell equations[18], as described below. The fitting parameters include the global values p_E and k_{ex} (or alternatively $k_{EG} = (1-p_E)k_{ex}$ and $k_{GE} = p_E k_{ex}$, see below) and residue specific chemical shift differences, $\Delta\varpi$ (ppm), and intrinsic relaxation rates for the anti-phase TQ coherence $3C_zI^1_+I^2_+I^3_+ + 3C_zI^1_-I^2_-I^3_-$.

As described in the main text, the resulting imbalance between the relaxation rates of in-phase and anti-phase coherences[19], that interconvert during intervals between ^{1}H CPMG refocusing pulses, can lead to small contributions to $R_{2,eff}$ values in dispersion profiles that can be corrected so long as the difference in transverse relaxation rates of these coherences is taken into account. The differential relaxation between in-phase and anti-phase TQ coherences, $\Delta R_2 = R_{2,I} - R_{2,A}$, can be measured with the scheme shown in Figure S1 (see below) with ΔR_2 subsequently inserted as a fixed parameter in the fits or, alternatively, ΔR_2 can be treated as a fitting parameter for each residue. The two approaches yield essentially identical results for the FF domain and we prefer, therefore, treating ΔR_2 as a fitting parameter as this involves less measurements.

Further discussion of the sensitivity of the TQ CPMG pulse scheme. In the text we have presented a brief explanation as to why $3/4$ of the signal is retained in the TQ experiment. It is important to note that the terms of interest at the start of the CPMG element ($8C_zI_x^1I_y^2I_y^3 + 8C_zI_y^1I_x^2I_y^3 + 8C_zI_y^1I_y^2I_x^3 = 8C_zQ$) are subsequently selected at the end of T_{relax} and transferred back to observable ^{1}H magnetization. This gives rise to the factor of 3/4 discussed in the text. Instead, imagine an experiment where the magnetization of interest at the start of the CPMG element is given by $8C_zI_x^1I_x^2I_x^3$. As with $8C_zQ$ it is straightforward to show that $8C_zI_x^1I_x^2I_x^3$ comprises both TQ and SQ coherences. Selection of the TQ elements results in the following terms, $8C_zI_x^1I_x^2I_x^3 \xrightarrow{\text{TQ selection}} C_zI_+^1I_+^2I_+^3 + C_zI_-^1I_-^2I_-^3$. If the pulse scheme subsequently selects for elements proportional to $8C_zI_x^1I_x^2I_x^3$ at end of the CPMG interval then only $1/4$ of the starting coherences will ultimately be detected by observing ^{1}H. Thus, the 'nature' of the coherences from which the TQ elements are selected and subsequently returned to is critical in defining the sensitivity of the experiment.

Differential relaxation between anti-phase and in-phase ^{1}H TQ elements, ΔR_2. It is of interest to establish the magnitude of $\Delta R_2 = R_{2,I} - R_{2,A}$ and in particular how ΔR_2 relates to the methyl ^{13}C R_1 rate, as ΔR_2 derives from differences in relaxation rates of coherences that are in and out of phase with respect to ^{13}C. A Redfield density matrix calculation[20] shows that

$$\Delta R_2 = 0.3 d_{CH}^2 [3J(\omega_C) + 6K(\omega_C)] \qquad \text{[S1]}$$

where $d_{CH}^2 = \left(\frac{\mu_0}{4\pi}\right)^2 \frac{\gamma_H^2 \gamma_C^2 \hbar^2}{r_{HC}^6}$, γ_j is the gyromagnetic ratio of spin j, h is Planck's constant, and r_{HC} is the one-bond ^{1}H–^{13}C distance. In Eq. [S1] $J(\omega)$ and $K(\omega)$ are auto- and cross-correlation spectral densities for the ^{13}C–^{1}H dipolar interactions, defined as[21,22]:

$$J(\omega) = \frac{1}{9} S_{axis}^2 \frac{\tau_c}{1 + (\omega\tau_c)^2} + \left(1 - \frac{1}{9} S_{axis}^2\right) \frac{\tau'}{1 + (\omega\tau')^2}, \qquad \text{[S2]}$$

$$K(\omega) = \frac{1}{9} S_{axis}^2 \frac{\tau_c}{1 + (\omega\tau_c)^2} + \left(-\frac{1}{3} - \frac{1}{9} S_{axis}^2\right) \frac{\tau'}{1 + (\omega\tau')^2} \qquad \text{[S3]}$$

where τ_c is the rotational correlation time for the assumed spherical molecule to which the methyl group is attached, S^2_{axis} is the generalized order parameter of the methyl 3-fold symmetry axis, $1/\tau' = 1/\tau_c + 1/\tau_e$, where τ_e is the time constant for rapid rotation of the methyl averaging axis. Assuming that $(\omega_C\tau_c)^2 \gg 1, (\omega_C\tau')^2 \ll 1$ it follows that

$$\Delta R_2 = 0.3d^2_{CH}\left[\frac{S^2_{axis}}{\omega^2_C\tau_c} + 3\left(1-\frac{S^2_{axis}}{9}\right)\tau' + 6\left(-\frac{1}{3}-\frac{S^2_{axis}}{9}\right)\tau'\right] \approx 0.3d^2_{CH}\left[\frac{S^2_{axis}}{\omega^2_C\tau_c} + (1-S^2_{axis})\tau_e\right]. \quad \text{[S4]}$$

Eq. [S4] can be compared with the corresponding expression for the ^{13}C longitudinal relaxation rate, $R_1(^{13}\text{C})$, for a $^{13}CH_3$ spin system (neglecting CSA) as measured using a pulse scheme that suppresses cross-correlation between ^{13}C–^{1}H dipolar interactions[23]

$$R_1(^{13}\text{C}) \approx 0.3d^2_{CH}\left[\frac{3.35}{9}\frac{S^2_{axis}}{\omega^2_C\tau_c} + 10\left(1-\frac{S^2_{axis}}{9}\right)\tau_e\right]. \quad \text{[S5]}$$

It is noteworthy that although $J(0) \approx K(0)$ for intra-methyl ^{13}C–^{1}H dipolar interactions, $J(\omega_C)$ is typically different from $K(\omega_C)$ due to the large contribution from τ_e[24], leading to differences between $R_1(^{13}\text{C})$ and ΔR_2. For example, assuming $S^2_{axis} = 0.5$, $\tau_e = 20$ ps and $\tau_c = 5$ ns, parameters that are similar to those for GB1 (25°C), the calculated $R_1(^{13}\text{C})/\Delta R_2$ ratio is 1.94 at 600 MHz, and the average measured value for GB1 at 600 MHz is 1.94, as well. Average values for ΔR_2 are 1.08 s^{-1} for GB1 and 1.62 s^{-1} for the FF domain (600 MHz).

Evolution of ^{1}H TQ coherences due to chemical exchange. In the simplest implementation the evolution of ^{1}H TQ coherences due to chemical shift, intra-methyl ^{1}H–^{13}C scalar couplings, $^1J_{HC}$, and two-site chemical exchange, $G \underset{k_{EG}}{\overset{k_{GE}}{\rightleftharpoons}} E$, is given by

$$\frac{d}{dt}\begin{pmatrix} M^G_A \\ M^G_I \\ M^E_A \\ M^E_I \end{pmatrix} = -\begin{pmatrix} 3i\omega_G + R^G_{2,A} + k_{GE} & 3i\pi^1 J_{HC} & -k_{EG} & 0 \\ 3i\pi^1 J_{HC} & 3i\omega_G + R^G_{2,I} + k_{GE} & 0 & -k_{EG} \\ -k_{GE} & 0 & 3i\omega_E + R^E_{2,A} + k_{EG} & 3i\pi^1 J_{HC} \\ 0 & -k_{GE} & 3i\pi^1 J_{HC} & 3i\omega_E + R^E_{2,I} + k_{EG} \end{pmatrix}\begin{pmatrix} M^G_A \\ M^G_I \\ M^E_A \\ M^E_I \end{pmatrix} \quad \text{[S6]}$$

where $M_A = 3C_zI^1_+I^2_+I^3_+$, $M_I = 1.5I^1_+I^2_+I^3_+$, the subscripts or superscripts G and E denote ground and excited states, respectively, and $R_{2,A}$ and $R_{2,I}$ are intrinsic transverse relaxation rates for TQ anti- and in-phase coherences. A similar set of equations holds for the evolution of $M_A = 3C_zI^1_-I^2_-I^3_-$, $M_I = 1.5I^1_-I^2_-I^3_-$ as well, with i replaced by $-i$. All analyses assume equal relaxation rates in ground and excited states. It should be noted that Eq. [S6] is a simplification, in general, as the number of basis set elements that are needed to account for offset effects, B_1 inhomogeneity and pulse miscalibration must be larger. Since TQ coherences can only be created within the subspace of the $\mathrm{I} = 3/2$ manifold and pulses do not couple coherences belonging to different manifolds[25] a more rigorous treatment would be to include a pair of 32 basis set elements that span the $3/2$ manifold. Each basis set includes both in-phase and anti-phase elements to account for scalar coupled ($^1J_{HC}$) evolution, one set for spins in the ground state and a second for spins in the excited state. We have found that the simple approach of Eq. [S6] produces essentially identical results to those obtained with the more complex treatment and have therefore used Eq. [S6] in all analyses of dispersion data.

As described above the fitting parameters include global values (k_{GE}, k_{EG}) as well as per-residue rates, $R_{2,A}$ and $\Delta R_2 = R_{2,I} - R_{2,A}$. In the case where flat dispersions are obtained, such as for GB1 (Figure 1), these are 'corrected' for differential relaxation between in-phase and anti-phase TQ coherences using the relation[26]

$$R_{2,eff}(\nu_{CPMG}, \text{corrected}) = R_{2,eff}(\nu_{CPMG}) - \Delta R_2 \left[\frac{1}{2} - \frac{\sin(6\pi^1 J_{HC}\tau_{cp})}{12\pi^1 J_{HC}\tau_{cp}}\right] \qquad [S7]$$

and no attempt at fitting the data has been made. By contrast, ΔR_2 is used as an explicit fitting parameter when dispersion profiles are fitted.

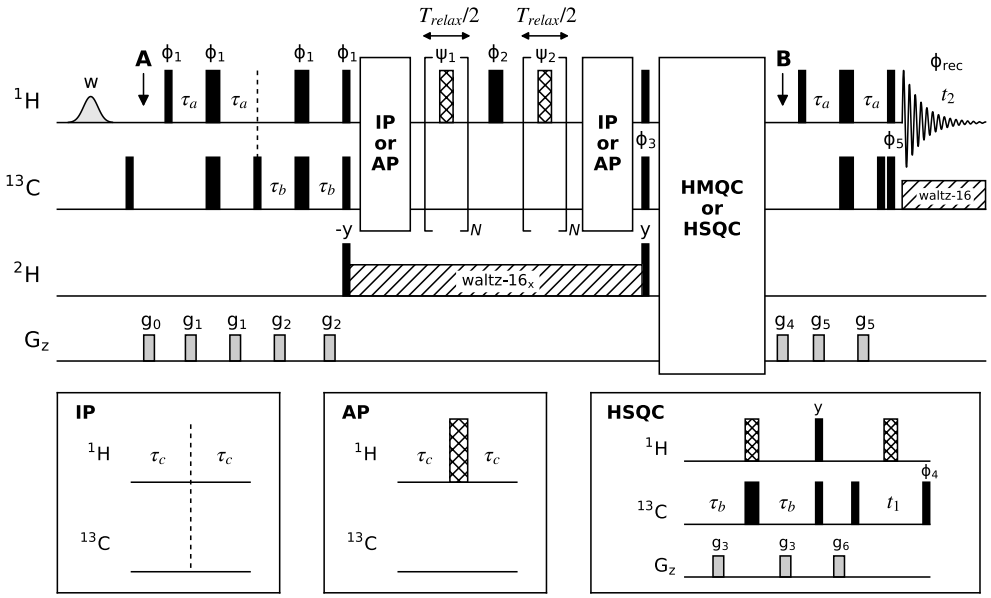

Figure S1. Pulse scheme for measurement of ^{1}H TQ CPMG relaxation dispersion profiles (see Figure 1) with ^{13}C chemical shifts recorded via either HMQC (Figure 1) or HSQC elements. Details are as per Figure 1, with $\tau_c = 1/(12^1J_{HC})$ ($\simeq 0.65$ ms) and g_6: $(0.5, -15)$. Insets labeled IP and AP are schemes for measuring the difference between transverse relaxation rates of anti-phase (AP) $3C_zI^1_+I^2_+I^3_+ + 3C_zI^1_-I^2_-I^3_-$, and in-phase (IP) $-1.5iI^1_+I^2_+I^3_+ + 1.5iI^1_-I^2_-I^3_-$, TQ elements. By recording a pair of planes (one for in-phase and one for anti-phase) at high ^{1}H pulsing rates ($\nu_{CPMG} = 2$ kHz), so that the interconversion between in-phase and anti-phase magnetization is prevented, the difference in R_2 rates is obtained from the relation $\Delta R_2 = R_{2,I} - R_{2,A} = -\dfrac{1}{T_{relax}} \ln \dfrac{I_I}{I_A}$ where I_I and I_A are intensities of corresponding peaks in spectra recording the evolution of in-phase and anti-phase coherences, respectively. Note that the IP and AP elements are removed when CPMG dispersion profiles are measured.

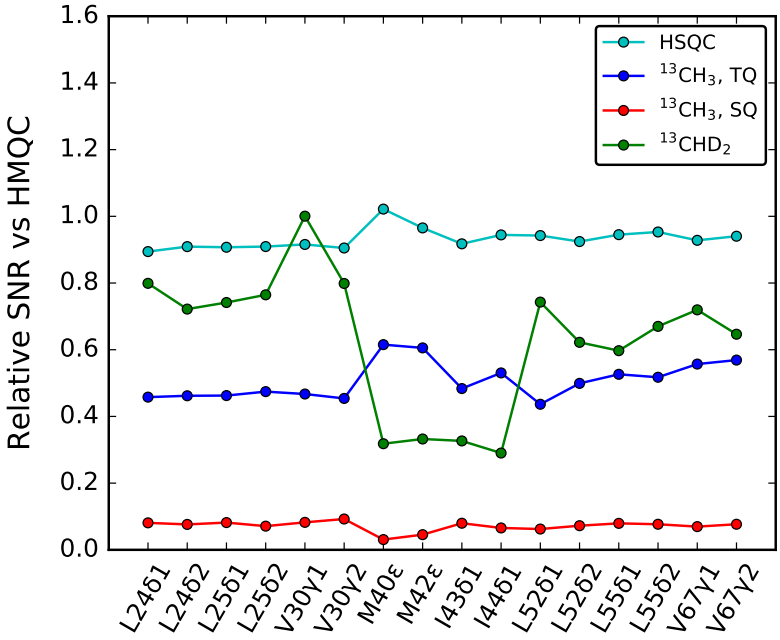

Figure S2. Per-residue signal to noise ratios (SNRs) as measured from ^{13}C–^{1}H HSQC (cyan), ^{1}H TQ CPMG RD (blue), ^{1}H SQ $^{13}CHD_2$ RD (green) and ^{1}H SQ $^{13}CH_3$ RD (red) schemes, recorded on ILVM-$^{13}CH_3$ or ILVM-$^{13}CHD_2$ FF domain samples, 600 MHz, 25°C. Intensities of correlations were measured in 2D spectra recorded with the CPMG element removed. All intensities are normalized to those obtained in a ^{13}C–^{1}H HMQC reference plane. Average SNRs are 0.93 ± 0.03, $0.73 \pm 0.10 (0.32 \pm 0.02)$, 0.51 ± 0.05 and 0.07 ± 0.01 for HSQC, L,V residues(I,M residues) in $^{13}CHD_2$ SQ CPMG, $^{13}CH_3$ TQ CPMG, and $^{13}CH_3$ SQ CPMG, respectively. These values are in reasonably good agreement with calculations. For example, the relative SNR of L,V ($^{13}CH_3$) cross peaks in the TQ vs SQ ($^{13}CHD_2$) data sets is predicted to be $\frac{3 \times 0.75}{2\sqrt{2}} = 0.80$ (measured value of 0.68) where the factor of 3 takes into account that magnetization from 3 methyl protons contributes to the TQ signal, while 0.75 includes the loss from generation of TQ coherences. The factor of $2\sqrt{2}$ in the denominator reflects the fact that the effective concentration of NMR active L,V methyl groups is twice as high with the $^{13}CHD_2$ labeling scheme relative to $^{13}CH_3$ (both isopropyl methyls are $^{13}CHD_2$, while only one is $^{13}CH_3$) and an enhanced sensitivity pulse scheme is used that preserves both cosine and sine modulated t_1 components (factor of $\sqrt{2}$). In the case of I,M methyl groups the SNR in TQ and $^{13}CHD_2$ SQ data sets is predicted to be $\frac{3 \times 0.75}{\sqrt{2}} = 1.59$ and a value of 1.77 is measured.

References

[1] M. H. Levitt, R. Freeman, *J. Magn. Reson.* **1979**, *33*, 473–476.

[2] H. Geen, R. Freeman, *J. Magn. Reson.* **1991**, *93*, 93–141.

[3] D. F. Hansen, P. Vallurupalli, L. E. Kay, *J. Phys. Chem. B* **2008**, *112*, 5898–5904.

[4] A. J. Shaka, J. Keeler, T. Frenkiel, R. Freeman, *J. Magn. Reson.* **1983**, *52*, 335–338.

[5] T. Gullion, D. B. Baker, M. S. Conradi, *J. Magn. Reson.* **1990**, *89*, 479–484.

[6] D. Marion, M. Ikura, R. Tschudin, A. Bax, *J. Magn. Reson.* **1989**, *85*, 393–399.

[7] V. Tugarinov, L. E. Kay, *J. Biomol. NMR* **2004**, *28*, 165–172.

[8] N. K. Goto, K. H. Gardner, G. A. Mueller, R. C. Willis, L. E. Kay, *J. Biomol. NMR* **1999**, *13*, 369–374.

[9] J. R. Huth, C. A. Bewley, B. M. Jackson, A. G. Hinnebusch, G. M. Clore, A. M. Gronenborn, *Prot. Sci.* **1997**, *6*, 2359–2364.

[10] D. M. Korzhnev, T. L. Religa, W. Banachewicz, A. R. Fersht, L. E. Kay, *Science* **2010**, *329*, 1312–1316.

[11] R. Sprangers, L. E. Kay, *Nature* **2007**, *445*, 618–622.

[12] V. Tugarinov, L. E. Kay, *J. Am. Chem. Soc.* **2007**, *129*, 9514–9521.

[13] A. J. Baldwin, T. L. Religa, D. F. Hansen, G. Bouvignies, L. E. Kay, *J. Am. Chem. Soc.* **2010**, *132*, 10992–10995.

[14] M. Tollinger, N. R. Skrynnikov, F. A. A. Mulder, J. D. Forman-Kay, L. E. Kay, *J. Am. Chem. Soc.* **2001**, *123*, 11341–11352.

[15] F. A. A. Mulder, N. R. Skrynnikov, B. Hon, F. W. Dahlquist, L. E. Kay, *J. Am. Chem. Soc.* **2001**, *123*, 967–975.

[16] D. M. Korzhnev, X. Salvatella, M. Vendruscolo, A. A. Di Nardo, A. R. Davidson, C. M. Dobson, L. E. Kay, *Nature* **2004**, *430*, 586–590.

[17] F. Delaglio, S. Grzesiek, G. W. Vuister, G. Zhu, J. Pfeifer, A. Bax, *J. Biomol. NMR* **1995**, *6*, 277–293.

[18] H. M. McConnell, *J. Chem. Phys.* **1958**, *28*, 430–431.

[19] J. P. Loria, M. Rance, A. G. Palmer, *J. Am. Chem. Soc.* **1999**, *121*, 2331–2332.

[20] A. G. Redfield, *IBM J. Res. Dev.* **1957**, *1*, 19–31.

[21] L. E. Kay, D. A. Torchia, *J. Magn. Reson.* **1991**, *95*, 536–547.

[22] G. Lipari, A. Szabo, *J. Am. Chem. Soc.* **1982**, *104*, 4559–4570.

[23] L. E. Kay, T. E. Bull, L. K. Nicholson, C. Griesinger, H. Schwalbe, A. Bax, D. A. Torchia, *J. Magn. Reson.* **1992**, *100*, 538–558.

[24] E. Rennella, R. Huang, A. Velyvis, L. E. Kay, *J. Biomol. NMR* **2015**, *63*, 187–199.

[25] P. L. Corio, *Structure of High-Resolution NMR Spectra*, Academic Press, New York, **1966**.

[26] A. G. Palmer, N. J. Skelton, W. J. Chazin, P. E. Wright, M. Rance, *Mol. Phys.* **1992**, *75*, 699–711.

```
/*  CH3_H_TQ_CPMG_600_lek.c

    This pulse sequence will allow one to perform the following experiment:

    2D C13,1H correlation map to observe exchange with variable CPMG spacing
    Used to record 1H dispersions of 13CH3 methyls

    Uses three channels:
         1)  1H             - 4.73 ppm or methyl region
         2) 13C             - 20 ppm (methyl,ile region)
         3)  2H             - 1 ppm (methyl,ile region)

    Set dm = 'nnny', dmm = 'cccp' [13C decoupling during acquisition].
    Set dm2 = 'nnnn', dmm2 = 'cccp'.
    Set dm3 = 'n', dmm3 = 'c'.

    Set water_flg = 'y' to apply water selective pulse at the beginning

    Set mq_flg = 'y'/'n' to record as HMQC/HSQC

    Set cal_flg = 'y' and AP_flg = 'y'/'n' to measure differential relaxation
    between IP and AP TQ coherence

    Set D_flg = 'y' to turn on 2H decoupling (not necessary for only
    measuring nu_cpmg > 100 Hz or only studying Met)

    Set comp180_flg = 'y' to apply composite 180 CPMG pulses,
    "composite_1"/"composite_2" refers to 90-y180x90-y/90y180x90y

    Must set phase = 1,2 for States-TPPI acquisition in t1 [13C].

    At tpwr_cp: pw_cp for CPMG region

*/

#include <standard.h>

static int  phi1[6] = {0,1,2,3,4,5},
            phi2[2] = {1,3},
            phi3[12] = {0,0,0,0,0,0,2,2,2,2,2,2},
            phi4[24] = {0,0,0,0,0,0,0,0,0,0,0,0,2,2,2,2,2,2,2,2,2,2,2,2},
            phi5[2] = {0,2},
            psi1[16] = {0,1,0,1,1,0,1,0,2,3,2,3,3,2,3,2},
            psi2[16] = {0,3,0,3,3,0,3,0,2,1,2,1,1,2,1,2},
            psi3[16] = {1,0,1,0,0,1,0,1,3,2,3,2,2,3,2,3},
            psi4[16] = {3,0,3,0,0,3,0,3,1,2,1,2,2,1,2,1},
            psi5[16] = {0,2,2,0,2,0,0,2,2,0,0,2,0,2,2,0},
            rec[24] = {0,2,0,2,0,2,2,0,2,0,2,0,2,0,2,0,2,0,0,2,0,2,0,2};

static double d2_init=0.0;

pulsesequence()
{
/* DECLARE VARIABLES */

 char       fsat[MAXSTR],
            fscuba[MAXSTR],
            f1180[MAXSTR],    /* Flag to start t1 @ halfdwell            */
            water_flg[MAXSTR],
            shape_sl[MAXSTR],
            mq_flg[MAXSTR],
            cal_flg[MAXSTR],
            AP_flg[MAXSTR],
            D_flg[MAXSTR],
            comp180_flg[MAXSTR];

 int         phase,
             t1_counter;   /* used for states tppi in t1          */

 double      tau1,         /*  t1 delay */
             taua,         /*  set to < 1/4JCH  */
             taub,         /*  set to < 1/4JCH  */
             tauc,         /*  set to exactly 1/12JCH  */
```

```
              tsatpwr,      /* low level 1H trans.power for presat  */
              sw1,          /* sweep width in f1                    */

              tpwr_cp,      /* power level for 1H CPMG    */
              pw_cp,        /* 1H pw for CPMG     */

              ncyc_cp,      /* number of CPMG cycles */
              time_T2,      /* total time for CPMG trains  */
              tau_cpmg,

              tof_me,

              dhpwr,
              pwc,

              dpwr3,
              dpwr3_D,

              pwd,
              pwd1,

              gt0,
              gt1,
              gt2,
              gt3,
              gt4,
              gt5,
              gt6,

              gzlvl0,
              gzlvl1,
              gzlvl2,
              gzlvl3,
              gzlvl4,
              gzlvl5,
              gzlvl6,

              tpwrsl,
              pw_sl;

/* LOAD VARIABLES */

  getstr("fsat",fsat);
  getstr("f1180",f1180);
  getstr("fscuba",fscuba);

  getstr("shape_sl",shape_sl);
  getstr("water_flg",water_flg);
  getstr("mq_flg",mq_flg);
  getstr("cal_flg",cal_flg);
  getstr("AP_flg",AP_flg);
  getstr("D_flg",D_flg);
  getstr("comp180_flg",comp180_flg);

  taua   = getval("taua");
  taub   = getval("taub");
  tauc   = getval("tauc");
  tpwr = getval("tpwr");
  tsatpwr = getval("tsatpwr");
  dpwr = getval("dpwr");
  phase = (int) ( getval("phase") + 0.5);
  sw1 = getval("sw1");

  tpwr_cp = getval("tpwr_cp");
  pw_cp = getval("pw_cp");
  ncyc_cp = getval("ncyc_cp");
  time_T2 = getval("time_T2");

  tof_me = getval("tof_me");

  dpwr3 = getval("dpwr3");
  dpwr3_D = getval("dpwr3_D");
```

```
  pwd = getval("pwd");
  pwd1 = getval("pwd1");

  dhpwr = getval("dhpwr");
  pwc = getval("pwc");

  gt0 = getval("gt0");
  gt1 = getval("gt1");
  gt2 = getval("gt2");
  gt3 = getval("gt3");
  gt4 = getval("gt4");
  gt5 = getval("gt5");
  gt6 = getval("gt6");

  gzlvl0 = getval("gzlvl0");
  gzlvl1 = getval("gzlvl1");
  gzlvl2 = getval("gzlvl2");
  gzlvl3 = getval("gzlvl3");
  gzlvl4 = getval("gzlvl4");
  gzlvl5 = getval("gzlvl5");
  gzlvl6 = getval("gzlvl6");

  tpwrsl = getval("tpwrsl");
  pw_sl = getval("pw_sl");

/* LOAD PHASE TABLE */

  settable(t1,6,phi1);
  settable(t2,2,phi2);
  settable(t3,12,phi3);
  settable(t4,24,phi4);
  settable(t5,2,phi5);
  settable(t11,16,psi1);
  settable(t12,16,psi2);
  settable(t13,16,psi3);
  settable(t14,16,psi4);
  settable(t15,16,psi5);
  settable(t16,16,psi5);
  settable(t8,24,rec);

/* CHECK VALIDITY OF PARAMETER RANGES */


    if((dm[A] == 'y' || dm[B] == 'y' || dm[C] == 'y' ))
    {
        printf("incorrect dec1 decoupler flags!  ");
        abort();
    }

    if((dm2[A] == 'y' || dm2[B] == 'y' || dm2[C] == 'y' || dm2[D] == 'y'))
    {
        printf("incorrect dec2 decoupler flags!  ");
        abort();
    }

    if( tsatpwr > 6 )
    {
        printf("TSATPWR too large !!!  ");
        abort();
    }

    if( pw > 200.0e-6 )
    {
        printf("dont fry the probe, pw too high ! ");
        abort();
    }

    if( tpwr_cp > 61 )
    {
        printf("don't fry the probe, tpwr_cp too large: < 62!  ");
        abort();
    }
```

```
    if( pw_cp < 9.0e-6 ) {
        printf("pw_cp is too low; > 9.0us\n");
        abort();
    }

    if( dpwr > 48 )
    {
        printf("don't fry the probe, DPWR too large!  ");
        abort();
    }

    if( dpwr2 > -16 )
    {
        printf("don't fry the probe, DPWR2 too large!  ");
        abort();
    }

    if( gt0 > 3e-3 ||  gt1 > 3e-3 || gt2 > 3e-3 || gt3 > 3e-3 || gt4 > 3e-3 || gt5 > 3e-3
                || gt6 > 3e-3 )
    {
        printf("gradients on for too long. Must be < 3e-3 \n");
        abort();
    }

    if( ncyc_cp > 80 ) {
        printf("ncyc_cp is too large; must be less than 81\n");
        abort();
    }

    if( time_T2 > 0.080 ) {
        printf("time_T2 is too large; must be less than 80 ms\n");
        abort();
    }

    if( ncyc_cp > 0 ) {
        tau_cpmg = time_T2/(4.0*ncyc_cp);
        if(ix == 1)
          printf("nuCPMG for curent experiment is (Hz): %5.3f\n",1/(4.0*tau_cpmg));
    }else {
        tau_cpmg = time_T2/(4.0);
        if(ix == 1)
          printf("minimum nuCPMG for curent experiment is (Hz): %5.3f\n",1./(4.0*tau_cpmg));
    }

    if( tau_cpmg < 50e-6 ) {
       printf("tau_cpmg is too small; decrease ncyc_cp\n");
       abort();
    }

    if( dpwr3 > 59 ) {
       printf("dpwr3 is too high; < 60\n");
       abort();
   }

    if( dpwr3_D > 51 ) {
       printf("dpwr3 is too high; < 52\n");
       abort();
   }

/*  Phase incrementation for hypercomplex 2D data */

    if (phase == 2){
        tsadd(t4,1,4);
    }

/*  Set up f1180  tau1 = t1                */

    tau1 = d2;

    if(f1180[A] == 'y') {
        if (mq_flg[A] == 'y'){
```

```
            tau1 += ( 1.0 / (2.0*sw1)) - 4.0*pw - 4.0e-6;
        }else{
            tau1 += ( 1.0 / (2.0*sw1)) - 4.0*pwc/PI - 4.0*pw - 4.0e-6;
        }
        if(tau1 < 0.4e-6) tau1 = 0.4e-6;
    }

    if(tau1 < 0.4e-6)
        tau1 = 0.4e-6;

     tau1 = tau1/2.0;

/* Calculate modifications to phases for States-TPPI acquisition          */

   if(ix == 1) d2_init = d2 ;
   t1_counter = (int) ( (d2-d2_init)*sw1 + 0.5 );
   if(t1_counter % 2) {
      tsadd(t4,2,4);
      tsadd(t8,2,4);
   }

/* BEGIN ACTUAL PULSE SEQUENCE */

status(A);

   rlpower(tsatpwr,TODEV);      /* Set transmitter power for 1H presaturation */
   rlpower(dhpwr,DODEV);        /* Set Dec1 power for 13C CPMG pulses         */
   rlpower(dpwr2,DO2DEV);       /* Set Dec2 power for 15N decoupling       */

   obsoffset(tof);

/* Presaturation Period */

status(B);

   lk_sample();

   if (fsat[0] == 'y')
   {
        delay(2.0e-5);
        rgpulse(d1,zero,2.0e-6,2.0e-6);  /* presaturation */
        rlpower(tpwr,TODEV);    /* Set transmitter power for hard 1H pulses */
        delay(2.0e-5);
        if(fscuba[0] == 'y')
        {
                delay(2.2e-2);
                rgpulse(pw,zero,2.0e-6,0.0);
                rgpulse(2*pw,one,2.0e-6,0.0);
                rgpulse(pw,zero,2.0e-6,0.0);
                delay(2.2e-2);
        }
   }
   else
   {
     delay(d1);
   }

   rlpower(tpwr_cp,TODEV);          /* Set transmitter power for 1H CPMG pulses */
   txphase(zero);
   dec2phase(zero);
   decphase(zero);
   delay(1.0e-5);

/* Begin Pulses */

status(C);

   rcvroff();
   lk_hold();
   delay(20.0e-6);

   if(water_flg[A] == 'y') {
     obspower(tpwrsl);
```

```
      shaped_pulse(shape_sl,pw_sl,zero,4.0e-6,0.0);
      obspower(tpwr_cp);
    }

    decrgpulse(pwc,zero,4.0e-6,0.0);

    obsoffset(tof_me);
    obsstepsize(60.0);
    xmtrphase(t1);

    delay(2.0e-6);
    rgradient('z',gzlvl0);
    delay(gt0);
    rgradient('z',0.0);
    delay(250.0e-6);

    rgpulse(pw_cp,zero,0.0,0.0);

    delay(2.0e-6);
    rgradient('z',gzlvl1);
    delay(gt1);
    rgradient('z',0.0);
    delay(250.0e-6);

    delay(taua - gt1 - 252.0e-6 - (2.0/PI)*pw_cp);

    simpulse(2.0*pw_cp,2.0*pwc,zero,zero,0.0,0.0);

    delay(2.0e-6);
    rgradient('z',gzlvl1);
    delay(gt1);
    rgradient('z',0.0);
    delay(250.0e-6);

    delay(taua - gt1 - 252.0e-6);

    decrgpulse(pwc,zero,0.0,0.0);

    delay(2.0e-6);
    rgradient('z',gzlvl2);
    delay(gt2);
    rgradient('z',0.0);
    delay(250.0e-6);

    delay(taub - gt2 - 252.0e-6);

    simpulse(2.0*pw_cp,2.0*pwc,zero,zero,0.0,0.0);

    delay(taub - gt2 - 252.0e-6 - POWER_DELAY - pwd1 - 4.0e-6 - POWER_DELAY - PRG_STA
RT_DELAY);

    delay(2.0e-6);
    rgradient('z',gzlvl2);
    delay(gt2);
    rgradient('z',0.0);
    delay(250.0e-6);

    if (D_flg[A] == 'y'){
        dec3power(dpwr3);
        dec3rgpulse(pwd1,three,4.0e-6,0.0);
        dec3phase(zero);
        dec3power(dpwr3_D);
        dec3unblank();
        dec3prgon(dseq3,pwd,dres3);
        dec3on();
    }else{
        delay(POWER_DELAY + pwd1 + 4.0e-6 + POWER_DELAY + PRG_START_DELAY);
    }

    decrgpulse(pwc,zero,0.0,0.0);
    rgpulse(pw_cp,zero,0.0,4.0e-6);

    xmtrphase(zero);
```

```
    decphase(t3);

    if (cal_flg[A] == 'y'){
        if (AP_flg[A] == 'y'){
            delay(tauc - 2.0*pw_cp - WFG_START_DELAY - 4.0e-6);
            shaped_pulse("composite_1",4.0*pw_cp,zero,4.0e-6,0.0);
            delay(tauc - 2.0*pw_cp - WFG_STOP_DELAY);
        }
        else{
            delay(2.0*tauc);
        }
    }

    if (ncyc_cp == 1){
        delay(tau_cpmg - 2.0*pw_cp - WFG_START_DELAY - 4.0e-6);
        shaped_pulse("composite_1",4.0*pw_cp,zero,4.0e-6,0.0);
        delay(tau_cpmg - 2.0*pw_cp - WFG_STOP_DELAY);
    }

    if (ncyc_cp > 1) {
        assign(zero,v1);

        initval(ncyc_cp,v4);
        loop(v4,v5);
            if (comp180_flg[A] == 'y'){
                getelem(t11,v1,v2);
                delay(tau_cpmg - 2.0*pw_cp - WFG_START_DELAY - 4.0e-6);
                shaped_pulse("composite_1",4.0*pw_cp,v2,4.0e-6,0.0);
                delay(tau_cpmg - 2.0*pw_cp - WFG_STOP_DELAY);
            }else{
                if((int) ncyc_cp % 8 == 3 || (int)ncyc_cp % 8 == 5) {
                    getelem(t11,v1,v2);
                }else if((int) ncyc_cp % 8 == 1 || (int)ncyc_cp % 8 == 7) {
                    getelem(t13,v1,v2);
                }else{
                    getelem(t15,v1,v2);
                }
                delay(tau_cpmg - pw_cp - 4.0e-6);
                rgpulse(2.0*pw_cp,v2,4.0e-6,0.0);
                delay(tau_cpmg - pw_cp);
            }
            incr(v1);
        endloop(v5);
    }

    rgpulse(2.0*pw_cp,t2,4.0e-6,4.0e-6);

    if (ncyc_cp == 1){
        delay(tau_cpmg - 2.0*pw_cp - WFG_START_DELAY - 4.0e-6);
        shaped_pulse("composite_2",4.0*pw_cp,zero,4.0e-6,0.0);
        delay(tau_cpmg - 2.0*pw_cp - WFG_STOP_DELAY);
    }

    if (ncyc_cp > 1) {
        decr(v1);

        initval(ncyc_cp,v4);
        loop(v4,v5);
            if (comp180_flg[A] == 'y'){
                getelem(t12,v1,v2);
                delay(tau_cpmg - 2.0*pw_cp - WFG_START_DELAY - 4.0e-6);
                shaped_pulse("composite_2",4.0*pw_cp,v2,4.0e-6,0.0);
                delay(tau_cpmg - 2.0*pw_cp - WFG_STOP_DELAY);
            }else{
                if((int) ncyc_cp % 8 == 3 || (int)ncyc_cp % 8 == 5) {
                    getelem(t12,v1,v2);
                }else if((int) ncyc_cp % 8 == 1 || (int)ncyc_cp % 8 == 7) {
                    getelem(t14,v1,v2);
                }else{
                    getelem(t16,v1,v2);
                }
                delay(tau_cpmg - pw_cp - 4.0e-6);
                rgpulse(2.0*pw_cp,v2,4.0e-6,0.0);
```

```
                delay(tau_cpmg - pw_cp);
            }
            decr(v1);
        endloop(v5);
    }

    if (cal_flg[A] == 'y'){
        if (AP_flg[A] == 'y'){
            delay(tauc - 2.0*pw_cp - WFG_START_DELAY - 4.0e-6);
            shaped_pulse("composite_2",4.0*pw_cp,zero,4.0e-6,0.0);
            delay(tauc - 2.0*pw_cp - WFG_STOP_DELAY);
        }
        else{
            delay(2.0*tauc);
        }
    }

    rgpulse(pw_cp,zero,4.0e-6 + SAPS_DELAY,0.0);
    decrgpulse(pwc,t3,0.0,0.0);

    if (D_flg[A] == 'y'){
        dec3off();
        dec3prgoff();
        dec3blank();
        dec3power(dpwr3);
        dec3rgpulse(pwd1,one,4.0e-6,0.0);
    }else{
        delay(POWER_DELAY + PRG_STOP_DELAY + pwd1 + 4.0e-6);
    }

    delay(2.0e-6);
    rgradient('z',gzlvl3);
    delay(gt3);
    rgradient('z',0.0);
    delay(250.0e-6);

    decphase(zero);

    if (mq_flg[A] == 'y') {
        delay(taub - gt3 - 252.0e-6 + WFG_START_DELAY - POWER_DELAY - PRG_STOP_DELAY
- pwd1 - 4.0e-6);

        decrgpulse(2.0*pwc,zero,0.0,0.0);

        delay(tau1);

        shaped_pulse("composite_1",4.0*pw_cp,zero,4.0e-6,0.0);

        delay(tau1);

        txphase(one);
        decphase(t4);

        delay(2.0e-6);
        rgradient('z',gzlvl3);
        delay(gt3);
        rgradient('z',0.0);
        delay(250.0e-6);

        delay(taub - gt3 - 252.0e-6 - WFG_STOP_DELAY);

        decrgpulse(pwc,t4,0.0,0.0);

        delay(2.0*pwc + 2.0*WFG_START_DELAY);

        rgpulse(pw_cp,one,0.0,0.0);

    }else{
        delay(taub - gt3 - 252.0e-6 - WFG2_START_DELAY - POWER_DELAY - PRG_STOP_DELAY
 - pwd1 - 8.0e-6);

        simshaped_pulse("composite_1","hard",4.0*pw_cp,2.0*pwc,zero,zero,4.0e-6,0.0);
        txphase(one);
```

```
        delay(2.0e-6);
        rgradient('z',gzlvl3);
        delay(gt3);
        rgradient('z',0.0);
        delay(250.0e-6);

        delay(taub - gt3 - 252.0e-6 - WFG2_STOP_DELAY);

        decrgpulse(pwc,zero,0.0,0.0);
        rgpulse(pw_cp,one,0.0,0.0);

        delay(2.0e-6);
        rgradient('z',gzlvl6);
        delay(gt6);
        rgradient('z',0.0);
        delay(250.0e-6);

        decrgpulse(pwc,zero,0.0,0.0);
        decphase(t4);

        delay(tau1);

        shaped_pulse("composite_1",4.0*pw_cp,zero,4.0e-6,0.0);

        delay(tau1);

        decrgpulse(pwc,t4,0.0,0.0);
    }

    obsoffset(tof);
    decphase(zero);
    obspower(tpwr);

    delay(2.0e-6);
    rgradient('z',gzlvl4);
    delay(gt4);
    rgradient('z',0.0);
    delay(250.0e-6);

    rgpulse(pw,zero,2.0e-6,0.0);

    delay(2.0e-6);
    rgradient('z',gzlvl5);
    delay(gt5);
    rgradient('z',0.0);
    delay(150.0e-6);

    delay(taua - gt5 - 152.0e-6);

    simpulse(2.0*pw,2.0*pwc,zero,zero,0.0,0.0);

    delay(2.0e-6);
    rgradient('z',gzlvl5);
    delay(gt5);
    rgradient('z',0.0);
    delay(150.0e-6);

    delay(taua - gt5 - 152.0e-6 - pwc - 2.0e-6 - pwc - 2.0*POWER_DELAY);

    decrgpulse(pwc,zero,0.0,0.0);
    decrgpulse(pwc,t5,2.0e-6,0.0);

    rlpower(dpwr,DODEV);   /* Set power for decoupling */
    rlpower(dpwr2,DO2DEV);   /* Set power for decoupling */

    rgpulse(pw,zero,0.0,0.0);

/* BEGIN ACQUISITION */

status(D);
    setreceiver(t8);
}
```