APPENDIX A
HSS READOUT ALGORITHM

Pseudo-code, describing the algorithm used to generate the spiral trajectories in this work, is shown below. The algorithm was modified from an existing VDS design algorithm\textsuperscript{29} and implemented in the C programming language.

\begin{verbatim}
N ← N;
\bar{k}_r, \bar{\theta}, \dot{k}_r, \dot{\theta}, k_r, \theta ← 0;

while k_r < k_{trans} do
    calculate next \ddot{k}_r, \ddot{\theta}, \ddot{k}_r, \ddot{\theta}, k_r, \theta;
end

\ddot{k}_{trans} ← \dot{k}_r;
\ddot{k}_r, \ddot{\theta}, \dot{k}_r, \dot{\theta}, k_r, \theta ← 0;

N ← 1;

while k_r < k_{trans} do
    calculate next \ddot{k}_r, \ddot{\theta}, \ddot{k}_r, \ddot{\theta}, k_r, \theta;
    output \ddot{k}_r, \ddot{\theta}, \ddot{k}_r, \ddot{\theta}, k_r, \theta;
end

N ← N;
\dot{k}_r ← \dot{k}_{trans};

while k_r < k_{max} do
    calculate next \ddot{k}_r, \ddot{\theta}, \ddot{k}_r, \ddot{\theta}, k_r, \theta;
    output \ddot{k}_r, \ddot{\theta}, \ddot{k}_r, \ddot{\theta}, k_r, \theta;
end
\end{verbatim}

The algorithm begins by calculating $\ddot{k}_{trans}$ (the first while loop), before resetting and calculating the single-shot trajectory up to $k_r = k_{trans}$ (the second while loop), at which point $N$ is set to the number of shots in the multi-shot region and $\dot{k}_r$ is set to $\dot{k}_{trans}$. After which, the algorithm proceeds to calculate the multi-shot portion of the trajectory (the third while loop).