

```

AZ = '0 0 1 0 0 0 1'; % BPSK - 0011001
BAZ = '1 1 1 0 0 0 1'; % BPSK - 1001001
EL = '1 0 0 0 0 0 1'; % BPSK - 1100001

%a = [1, 1, 0, 1, 1, 0, 1, 1; 1, 0, 1, 0, 0, 0, 1, 0];

% Preliminary go through to get functionIDs
a = totalMatrix;
BPSK = a(:, 2);
DPSK = bpskdpsk64(BPSK);

TX_enable_prev = 0;
FID_flag = 0;
FunctionIDs = zeros(50 , 25);
FID_count = 1;
FID_bit = 1;
SBS_prev = 0;

% for i = 1:64:length(a)
%     TX_enable = a(i, 1);
%     DPSKitr = DPSK(i);
%     SBS = a(i, 4);
%
%     %start of the preamble; assumes we are on the leading edge of a
%     %Transmit enable but not the leading edge of Scan Beam Start
%     if (TX_enable_prev == 0 && TX_enable == 1) && ~(SBS == 1 && SBSprev == 0)
%         FID_flag = 1;
%     end
%
%     %
%     if (FID_flag == 1 && FID_bit <= 25) % We hit leading edge of TX_enable
%         FunctionIDs(FID_count, FID_bit) = DPSKitr;
%         FID_bit = FID_bit + 1;
%     end
%
%     if FID_bit > 25
%         FID_flag = 0;
%         functionID = num2str(FunctionIDs(FID_count,19:end));
%         if strcmp(num2str(functionID), AZ) || strcmp(num2str(functionID), EL) || strcmp(num2str(functionID), BAZ)
%             FID_count = FID_count + 1; % writes the most recent functionID.
%         else
%             FunctionIDs(FID_count,:) = zeros(1,25);
%         end
%         FID_bit = 1;
%     end
%
%     if SBS == 1 && SBSprev == 0 && ~(TX_enable_prev == 0 && TX_enable == 1)
%         FID_count = FID_count + 1; % writes the most recent functionID.
%         FID_bit = 1;
%     end
%     TX_enable_prev = TX_enable;
%     SBSprev = SBS;
% end

%for the actual amount of IDs
FID_count = FID_count-1;

outArray = zeros(length(a(:, 1)), 1);

thetaR = 0;
thetaBW = 0;
thetaMIN = 0;
thetaMax = 0;
FID_itr = 1;

omega = 156250;

prevSBS = 0;

y = 0;

BPSK = a(:, 2);
DPSK = bpskdpsk64(BPSK);
%DPSK = BPSK;
for i = 1: length(a(:,1)) % file hasn't ended yet (400,000+ iterations)
    TX_enable = a(i, 1);
    DPSKitr = DPSK(i);
    TO_FR0 = a(i, 3);
    SBS = a(i, 4);
    ANT_pos = a(i, 5:7);
    ANTread = a(i, 8);

    if ANTread == 1
        ANT = num2str(ANT_pos);
        % Only check if antenna read is enabl

```

```

switch ANT
case '0 0 0'
    A = 1;
case '0 0 1'
    A = 0.5;
case '0 1 0'
    A = 0.25;
case '0 1 1'
    A = 0.125;
case '1 0 0'
    A = -1 %supposed to be unused
case '1 0 1'
    A = 10;
case '1 1 0'
    A = 0;
case '1 1 1'
    A = 0;
end
end

if TX_enable == 1

    if SBS == 1 && prevSBS == 0

        % Check function ID. Change angles based on function ID.
        if(FID_itr <= 50)
            functionID = FunctionIDs(FID_itr, 19:end);
            FID_itr = FID_itr + 1;
        end
        switch(stationNumber)
        case 0
            functionID = AZ;
        case 1
            functionID = BAZ;
        case 2
            functionID = EL;
        end

        if strcmp(num2str(functionID), AZ)
            thetaR = -5;
            thetaBW = 2;
            thetaMIN = -62;
            thetaMAX = 62;

        elseif strcmp(num2str(functionID), BAZ)
            thetaR = -5;
            thetaBW = 2;
            thetaMIN = -42;
            thetaMAX = 42;

        elseif strcmp(num2str(functionID), EL)
            thetaR = 3;
            thetaBW = 1.5;
            thetaMIN = -2;
            thetaMAX = 30;

        else
            thetaR = 0;
            thetaBW = 0;
            thetaMIN = 0;
            thetaMax = 0;
        end
        scanStart = i;
        while TX_enable ~= 0
            TX_enable = a(i, 1);
            TO_FRO = a(i, 3);
            SBS = a(i, 4);

            if TO_FRO == 1 % Scanning TO thetaMIN + t/50
                thetaT = thetaMIN + (i-scanStart)/50;
            end
            if TO_FRO == 0 % Scanning FRO thetaMIN - t/50
                thetaT = thetaMIN - (i-scanStart)/50;
            end

            outArray(i) = A * (sin(pi*(thetaT - thetaR)/(1.15*thetaBW)) / (pi * (thetaT - thetaR) / (1.15 * thetaBW))) * sin(omega * i));
            i = i + 1;
        end
        i = i - 1;
    else
        outArray(i) = A * sin(i * omega + DPSKitr * pi);
    end
end

else
    outArray(i) = 0;
end
end

```

```
    prevSBS = SBS;
end

figure(1)
plot(1:length(outArray), outArray)
xlabel('time');
ylabel('Amplitude');
title('TCU out');
figure(2)
plot(1.09 * 10^4:1.2 * 10^4, outArray(1.09 * 10^4:1.2 * 10^4))
```

---



