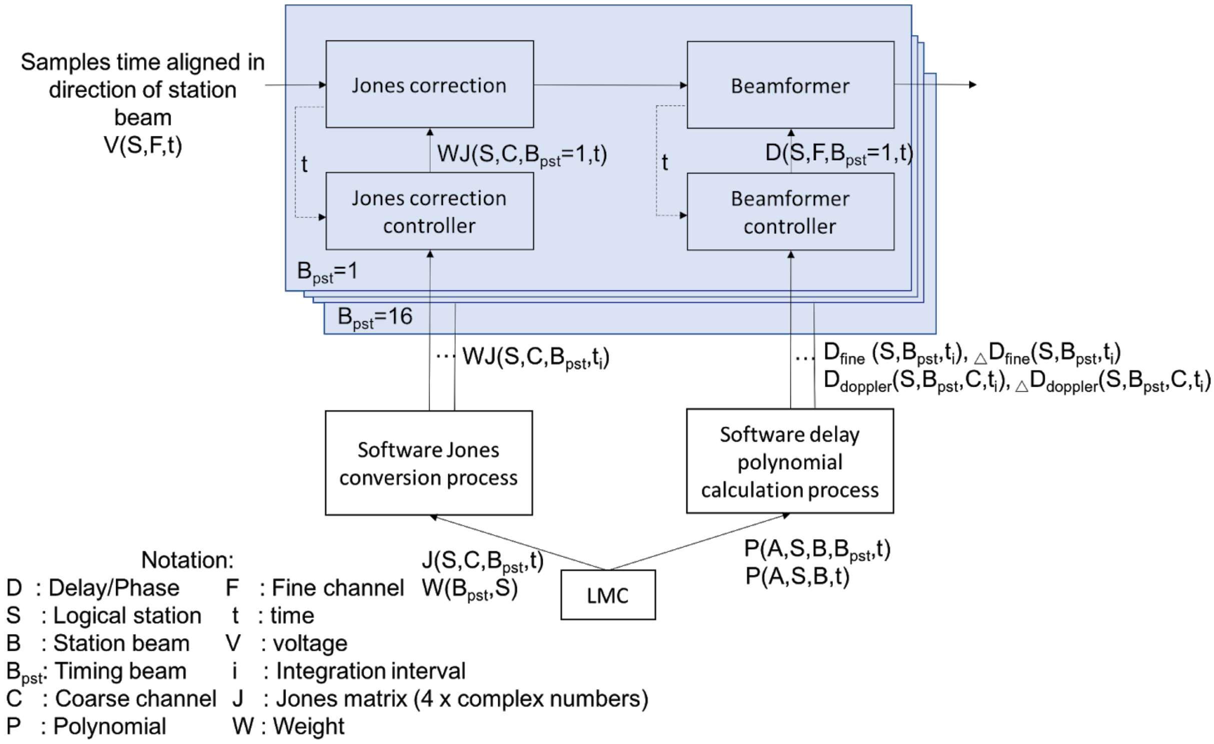
PSS/PST/VLBI? beamforming parameter conversion - Introduction

...

PST Beamforming parameter conversion - Implementation



LMC supplies the following configuration for PST:

1. *Pulsar Timing Jones correction (for each timing beam). Array (dimensions <=512 x <=384) of Jones matrix. Each Jones matrix is provided as 4 x (re,im) where re and im are single precision floats. This correction is supplied for each timing beam to correct station beam polarization errors off boresight. Each timing beam correction is an array of Jones matrix values with one matrix per coarse channel frequency per logical station.*
2. *LogicalStation weights (for each timing beam). Array (dimensions <=2048) of single precision float. A weight must be supplied for each LogicalStation that is a member of the subarray. A weight of 0 excludes the LogicalStation from contributing to the beam output.*
3. *Pulsar Timing Beam Offset Delay polynomials (for each timing beam). Array (dimensions <=512 x 11) of double precision floats. These polynomials describe the offset from StationBeam boresight. They allow the user to control the direction of the TimingBeam within the StationBeam. Delay used by CSP_Low.CBF = Pulsar Timing Beam offset delay polynomial - Station Beam delay polynomial. This is supplied for each Timing Beam. There is one polynomial per Timing Beam per LogicalStation.*
4. *Station beam delay polynomials (for each station beam). Array (dimensions <=2048 x 11) of double precision floats. The delay polynomial corrects for the difference in delay between a station and a reference point within the array. It brings all StationBeams within the subarray to a common wavefront. There is a one delay*

polynomial per LogicalStation. This delay model is for the boresight of the logical station beam and is shared for imaging, pulsar search and pulsar timing.

For 1, one Jones matrix, J(t), is supplied per timing beam per logical station per coarse channel resulting in the array of Jones matrices denoted as J(S,C, B_{pst} ,t).

For 2, one weight, W, is supplied per timing beam per logical station resulting in an array of weights denoted as W(B_{pst} ,S). Weights remain constant throughout the scan so are considered here to not be time variant.

For 3, one delay polynomial, P(t), is supplied per timing beam per station beam per logical station resulting in the array of polynomials denoted as $P(A, S, B, B_{pst}, t)$

For 4, one delay polynomial, P(t), is supplied per subarray per station beam per logical station resulting in the array of polynomials denoted as P(A,S,B,t)

LMC provides delay polynomials before the scan starts and at regular intervals during the scan. Polynomials are expected to be valid for minutes. LMC provides Jones matrices before the scan starts and at regular intervals during the scan. Each Jones matrix is valid for T_{jones} seconds (this is expected to be of the order of 1 to 10 seconds)

We use the notation, i, to denote a scan integration period where i=0 is the start of a scan. t_i is the time at the start of integration period i.

Software delay polynomial calculation Process

The MACE server uses the LMC supplied delay polynomials to calculate configuration for the beamformer controller. The software performs the following calculation for integration period i:

for B_{pst} in 1 to 16:
  A = Subarray that timing beam B_{pst} belongs to
  B = station beam assigned to timing beam B_{pst}
  for S in the set of logical stations that are members of timing beam B_{pst} :
    $P_{offset}(S) = P(A, S, B, B_{pst}, t_i) - P(A, S, B, t_i)$
    $\Delta D_{fine}(S, B_{pst}, i) = 2\pi\ x\ \Delta f_{pst} x\ P_{offset}(S)$
    $D_{fine}(S, B_{pst}, i) = -216/2\ x\ \Delta D_{fine}(S, B_{pst}, i)$
    for C in 1 to 384:
      if C is a member of (A,B):
        $D_{doppler}(S, B_{pst}, C, i) = 2\pi f_c P_{offset}(S)$
        $\Delta D_{doppler}(S, B_{pst}, C, i) = (\ D_{doppler}(S, B_{pst}, C, i) - D_{doppler}(S, B_{pst}, C, i+1)\ )\ /\ N$

Notes:

1. The integration period is 0.9 seconds.

2. All software calculations are performed in double floating point precision.
3. Calculation uses LMC supplied polynomials that are valid for time t_i
4. D_{pst} values are phases with units of radians.
5. FPGA configuration for integration period i must be calculated prior to t_i giving enough time for it to be transferred to the FPGA.
6. f_c is the coarse channel centre frequency in Hz for coarse channel C
7. Δf_{pst} is the frequency difference between adjacent PST fine channels. This is 3.6kHz; calculated as the width of a coarse frequency channel (926kHz) divided by 256 fine channels.
8. N is the number of coarse time samples in the integration period
9. Delays are normalised before sending to FPGAs so as not to generate negative values.

Beamformer controller and beamformer

The beamformer and beamformer controller are FPGA firmware modules. The basic operation for the beamformer is a complex multiply-accumulate (CMAC) operation as shown in figure X.

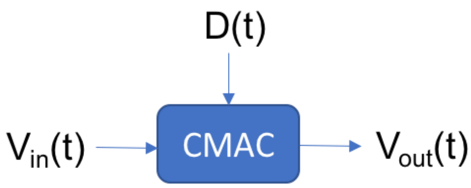


Figure X: A single beamformer CMAC module

A separate CMAC operation is performed for each logical station and fine channel data stream. Generalising for all data streams then:

$$V_{out}(S,F,t) \leftarrow V_{out}(S,F,t) + V_{in}(S,F,t)\ x\ D(S,F,t)$$

Calculation of D(S,F,t) is performed by the beamformer controller block from the software supplied values. The calculation is a multiple stage process as shown in figure X.

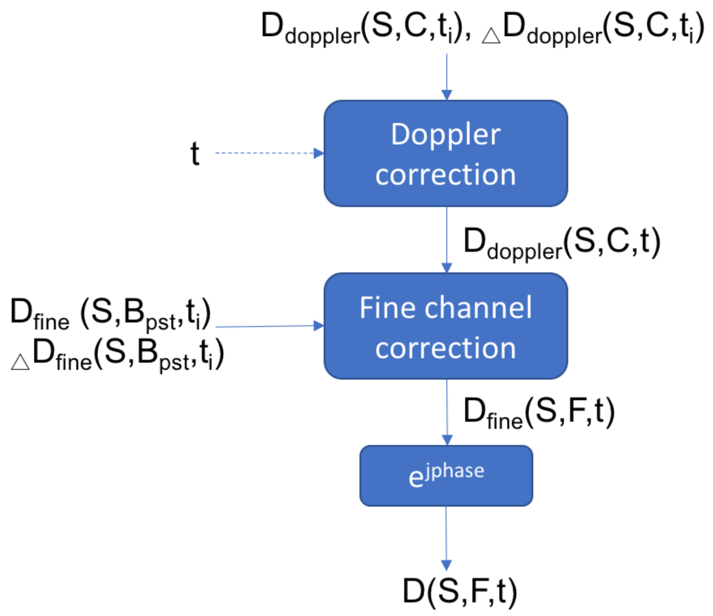


Figure X: Beamformer controller calculations

The software supplies an initial value, $D_{doppler}(S,C,i)$, and an increment, $\Delta D_{doppler}(S,C,i)$, for each integration period iteration (i). The doppler correction block performs a linear interpolation for the n=1..N coarse time sampling points within this period as shown in figure XX.

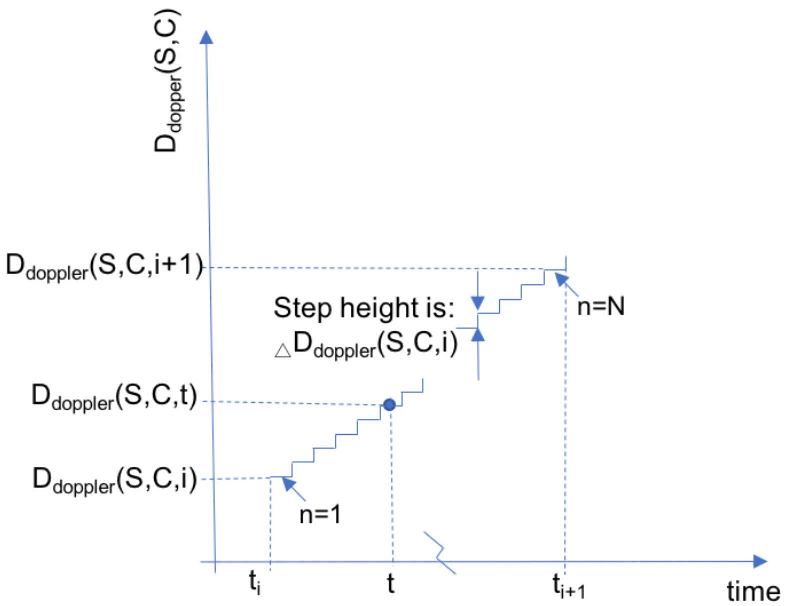


Figure X: Doppler correction for coarse channel C

The doppler correction block calculates the $D_{doppler}(S,C,t)$ value for time, t. This is then adjusted for each fine channel within the coarse channel frequency range by adding a fine channel correction. The fine channel correction is calculated as shown in figure XX. Fine channels within the coarse channel frequency range are numbered 1 to 216, with channel

108 being the center channel. Software supplies the adjustment for this center channel, $D_{fine}(S,i)$, and a fine channel increment, $\Delta D_{fine}(S,i)$. Note that these are valid for the whole integration period, i, such that $t_i \le t < t_{i+1}$.

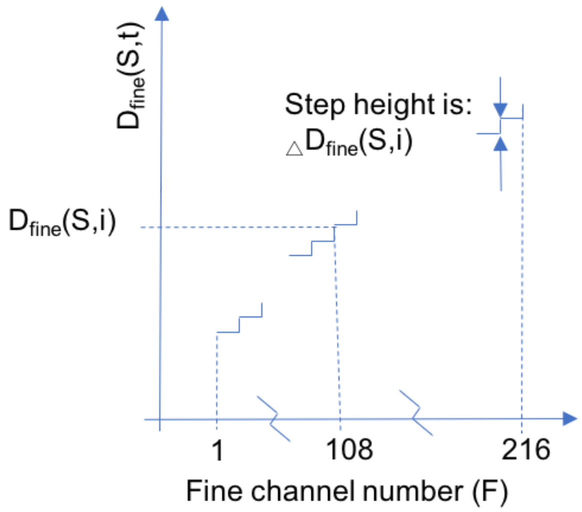


Figure X: Fine channel correction

The fine channel correction block outputs a value of $D_{fine}(S,F,t)$ which is used to lookup a value of e^{jphase} for the beamformer CMAC.

Software Jones conversion process

LMC provides the software Jones conversion process with an array of Jones matrices, $J(S,C,B_{pst},t)$ every T_{jones} seconds (where T_{jones} is expected to be about 1 to 10 seconds) and also an array of weights W(B_{pst} ,S). The weights are considered to be constant during a scan. The software Jones conversion process performs the following:

- Value conversion from floating point to fixed point with precision TBD.
- Scaling of Jones values by weight
- Mapping of Jones array to integration period

Software sends the weighted Jones array that is valid for integration period, i, as $WJ(S,C,B_{pst},t_i)$, to the FPGA Jones correction controller modules.

Jones correction controller

The basic operation for the Jones correction FPGA firmware block is shown in figure X where J(t) is a single Jones matrix consisting of 4 complex numbers

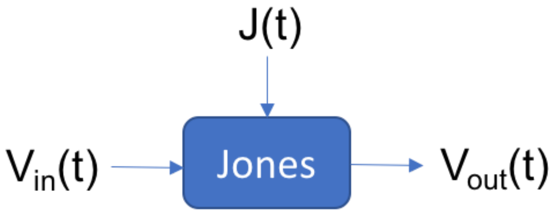


Figure X: Jones correction

There is one Jones correction block for each fine channel and for each logical station data stream. The Jones correction controller unpacks the software supplied array $WJ(S,C,B_{pst},t_i)$. The same Jones matrix is used for all of the fine channels within the coarse channel frequency range.