Vengono inizialmente caricati nel workspace i dati delle acquisizioni, salvati precedentemente con estensione .mat.

Questi file contengono diversi segnali ricevuti dalla SDR, in formato matriciale e come timeseries.

Ogni matrice contiene frame da 12500 campioni complessi, per un totale di 801 frame.

Sono presenti due acquisizioni di segnale senza imbalance con diversi valori di ampiezza e tre segnali ai quali è stata applicata una iq imbalance prima della trasmissione.

\begin{itemize}

\item{signal\_1}

\item{signal\_2}

\item{signal\_IQ10}

\item{signal\_IQ20}

\item{signal\_IQ30}

\end{itemize}

La prima parte dello script ha l’obbiettivo di modificare uno dei segnali originali applicando un imbalance arbitrario a “signal1”, una delle acquisizioni alla quale non era stato imposto un IQ imbalance in trasmissione.

La funzione apply\_IQ\_imbal, tramite iqimbal consente di applicare un imbalance di ampiezza e fase ad un segnale qualsiasi.

Nel nostro caso siamo interessati solo a quello di ampiezza, inseriamo quindi come parametro di ingresso alla funzione il valore in dB desiderato: $20\*log10(1.3)$ per un imbalance del 30%.

A questo punto la funzione aumenta la parte reale e riduce quella immaginaria del valore inserito, per ogni campione.

Si può osservare con un plot l’effetto della funzione sugli ultimi x campioni.

La fase successiva consiste del correggere il segnale precedentemente modificato.

L’approccio utilizzato è quello generico, considerando che ci sia un imbalance sia di ampiezza che di fase.

Il segnale originale può essere recuperato tramite il prodotto tra il segnale con imbalance e una matrice di correzione R.

\[

Y=RX

\]

Il segnale X dovrà essere una matrice 2xN dove N è il numero di campioni e le due righe sono rispettivamente parte immaginaria e reale.

La matrice R invece è espressa come:

\[

R =

\begin{bmatrix}

1+Re{C} & Im{C}\\

Im{C} & 1-Re{C}

\end{bmatrix}

\]

Dove il coefficiente C viene ricavato dalla funzione iqimbal2coeff inserendo come ingresso i valori di imbalance di ampiezza e fase.

La funzione imbalance\_estimation infine, utilizza l’oggetto comm.IQImbalanceCompensator per ricavare la stima dell’imbalance di ampiezza e fase.

L’algoritmo di stima è efficace per il segnale modificato dallo script mentre non fornisce risultati soddisfacenti per i segnali ai quali è stato applicato l’imbalance prima della trasmissione.

La causa è probabilmente l’effetto di rotazione della nuvola di campioni dovuta all’ambiguità di fase che si rileva al ricevitore che porta la nuvola di campioni ad essere indistinguibile da una nuvola senza imbalance.

Per questi segnali è stato quindi utilizzato un approccio diverso.

In seguito al processamento su simulink è possibile osservare che le nuvole di campioni dei segnali con imbalance si addensano in quattro punti che rappresentano i vertici di un rettangolo.

Il “rettangolo” però ruota nel tempo assumendo una posizione verticale o orizzontale.

L’idea è quindi quella di raggruppare i campioni in intervalli di 1000 nel quale è visibile una forma coerente della costellazione e stimare la lunghezza dei lati calcolando la deviazione standard della parte reale e della parte immaginaria dei punti.

Facendo il rapporto fra questi due valori si ottiene quindi il rapporto fra i lati. Per tenere conto della posizione verticale che può assumere il “rettangolo” il rapporto viene invertito quando risulta minore di uno.

Conseriamo un quadrato AxA i cui vertici rappresentano I punti di una costellazione senza imbalance.

L’effetto di un imbalance in ampiezza di valore K produce una costellazione rettangolare $A’B’$ dove $A’=AK$ e $B’=A/K$. Il rapporto fra i lati sarà quindi $A’/B’=K^2$.

Per ottenere il coefficiente di imbalance quindi, è necessario considerare la radice quadrata del rapporto fra i lati.