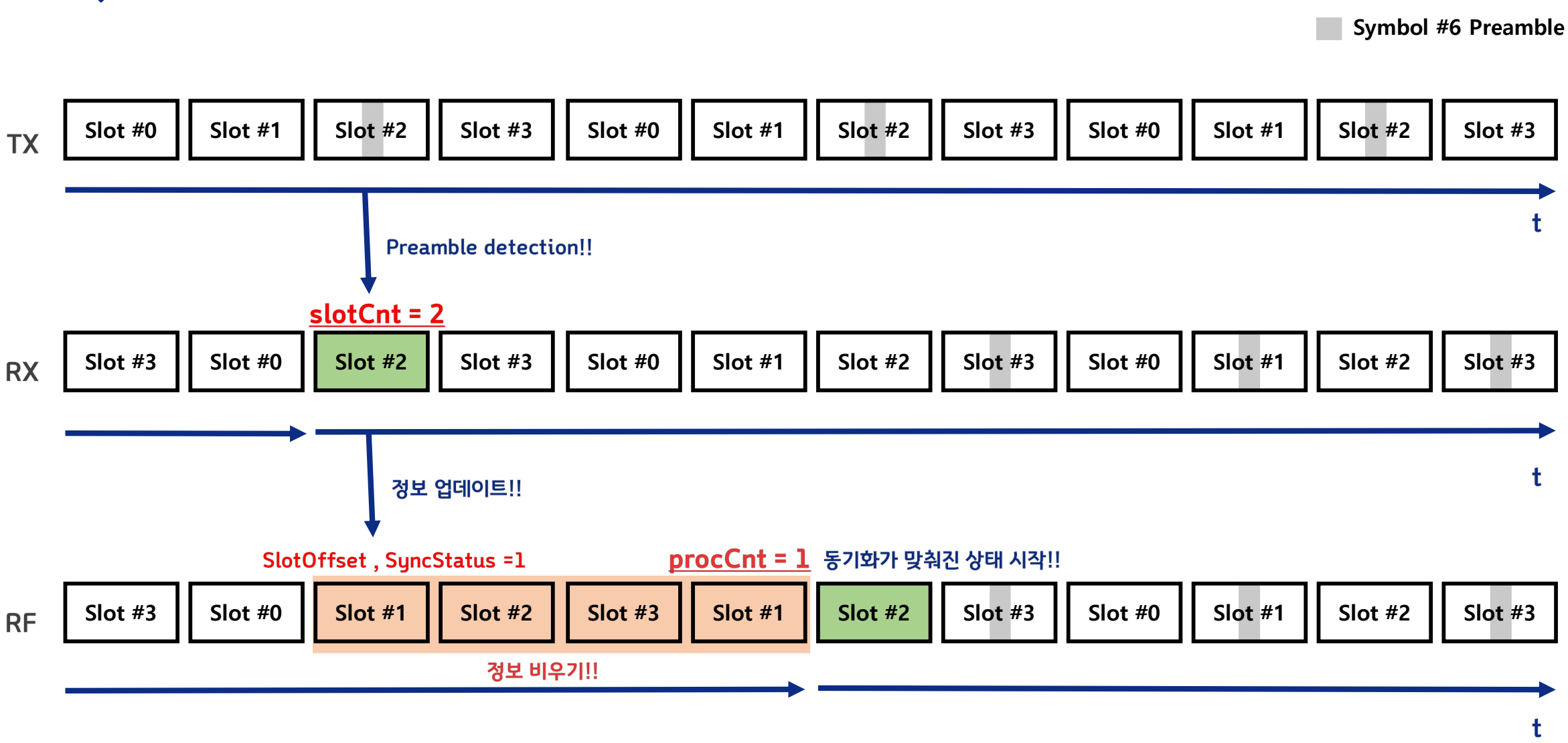
STLC channel estimation을 위한 Feedback channel 구현

발표자 : 김상은

CONTENTS


Symbol #6 Preamble
TX
Slot #0 Slot #1 Slot #2 Slot #3 Slot #0 Slot #1 Slot #2 Slot #3 Slot #0 Slot #1 Slot #2 Slot #3
t
Preamble detection!!
slotCnt = 2
RX
Slot #3 Slot #0 Slot #2 Slot #3 Slot #0 Slot #1 Slot #2 Slot #3 Slot #0 Slot #1 Slot #2 Slot #3
t
정보 업데이트!!
SlotOffset , SyncStatus =1
procCnt = 1
동기화가 맞춰진 상태 시작!!
RF
Slot #3 Slot #0 Slot #1 Slot #2 Slot #3 Slot #1 Slot #2 Slot #3 Slot #0 Slot #1 Slot #2 Slot #3
정보 비우기!!
t

1. 코드 구현

1-1) TX와 RX의 slot번호 맞추기(동기화)

[RX단] -> procCnt, slotCnt, status_synch

```
                if (status_synch == 0)
                {
                    LOG_E(PHY, "preamble SNR0 : %f\n", snr[0]);
                    LOG_E(PHY, "preamble SNR1 : %f\n", snr[1]);
                    frameCnt = 0;
#ifdef MOD_STLC     status_synch = 1;

                    vrx_commandSlotSync(procCnt, slotCnt, status_synch);
                    for (int i=0;i<VHW_NB_SAMPLEBANK;i++)
                        vrx_clear_processBitmap(procCnt+i);
                    procCnt = VHW_PREAMBLE_SLOT-1;
 #endif
                }
                else
                {
                    frameCnt++;
                }


            } « end if (time_offset=vrx_dete... »
            else //detection failed
            {
#ifdef MOD_STLC
                if (status_synch == 1)
                {
                    vrx_commandSlotSync(procCnt, procCnt, 0);
                }
 #endif

                status_synch = 0;
                snr[0] = VRX_SNR_NOSYNC;

                outSyncCnt++;
                #ifdef VRX_OUTSYNCDUMP
                if (outSyncCnt > VRX_MAX_OUTSYNC)
                {
                    vhw_dumpDataToFile(&vrx_rfRegPtr->roReg.rxData[procCnt][0][0], vrx_rfRegPtr->roReg.slotSize, "rxdata.txt");
                    LOG_E(PHY, "max count has reached! bye bye\n");
                    exit(-1);
                }
                #endif
            } « end else »
        } « end if slotOffsetStatus==0&&... »
```

1) Sync 0 → 1 (동기화 초기)

2) Sync 1 → 0 (동기화를 잃었을 때)

```
#ifdef MOD_STLC
static void vrx_commandSlotSync(int proc_nr, int slot_nr, int syncStatus)
{
    uint8_t res = 0;
    AssertFatal ( 0== pthread_mutex_lock(&(vrx_rfRegPtr->sharedReg.sharedMutex)), "");
    vrx_rfRegPtr->sharedReg.syncStatus = syncStatus;
    if (syncStatus == 1)
    {
        vrx_rfRegPtr->sharedReg.slotOffset = (slot_nr-proc_nr+VHW_NB_SAMPLEBANK)%VHW_NB_SAMPLEBANK;
    }
    else
    {
        vrx_rfRegPtr->sharedReg.slotOffset = 0;
    }
    AssertFatal ( 0== pthread_mutex_unlock(&(vrx_rfRegPtr->sharedReg.sharedMutex)), "");
}
#endif
```

1. 코드 구현

1-1) TX와 RX의 slot번호 맞추기(동기화)

[RX단]

1) Sync 0 → 1 (동기화 잡을 때)

TX	RX	
slot_nr	procCnt	slotCnt
0	2	2
1	3	3
2	0	~~0~~ 2
3		3
0		0
1	1	1
2	2	2
3	3	3

Preamble detection!!

Slotoffset = (slotCnt –procCnt +4)%4 = 2

2) Sync 1 → 0 (동기화를 잃었을 때)

TX	RX	
slot_nr	procCnt	slotCnt
0	0	0
1	1	1
2	2	2
3	3	3
0	0	0
1	1	1
2	2	2
3	3	3

Sync Loss!!

다른 보정없이
동기화 했을 때와
마찬가지로
SLOT은 그대로 흐름

Slotoffset = (slotCnt –procCnt +4)%4 = 0

1. 코드 구현

1-1) TX와 RX의 slot번호 맞추기(동기화)

[RF단] -> slot_nr, syncStatus

RX_DL

```
else  //normal DL readslot (BS to UE : slot decoding)
#endif
{
    LOG_I(PHY, "Downlink in slot : %i\n", slot_nr);
    //register reading
    AssertFatal ( 0== pthread_mutex_lock(&(vrf_rfIxReg.sharedReg.sharedMutex)), "");
    int syncTime = vrf_rfIxReg.sharedReg.syncOffset;
    int timeDrift = vrf_rfIxReg.sharedReg.timeDrift;
    int freqDrift = vrf_rfIxReg.sharedReg.freqDrift;

#ifdef MOD_STLC
    int slotOffset = vrf_rfIxReg.sharedReg.slotOffset;
#endif

    AssertFatal ( 0== pthread_mutex_unlock(&(vrf_rfIxReg.sharedReg.sharedMutex)), "");

    //synch time command
    if (syncTime != 0)
    {
        vrf_syncInSlot(&timestamp, syncTime);
    }
    else
    {
        //time compensation command
        if (timeDrift >= VRF_TIMEDRIFT_DELTA)
        {
            deltaDrift = VRF_TIMEDRIFT_DELTA;
        }
        else if (timeDrift <= -VRF_TIMEDRIFT_DELTA)
        {
            deltaDrift = -VRF_TIMEDRIFT_DELTA;
        }

        if (deltaDrift != 0)
            LOG_D(PHY, "[VRF] ------- time compensation - %i (command:%i)\n", deltaDrift, timeDrift);
    }

#ifdef MOD_STLC
    if (slotOffset != 0)
    {
        LOG_E(VRF, "SLOT offset command %i -> slot nr : %i -> %i\n", slotOffset, slot_nr, (slot_nr+slotOffset)%VHW_NB_SAMPLEBANK);
        slot_nr = (slot_nr+slotOffset)%VHW_NB_SAMPLEBANK;
    }
#endif

    //reset register commands
    AssertFatal ( 0== pthread_mutex_lock(&(vrf_rfIxReg.sharedReg.sharedMutex)), "");
    vrf_rfIxReg.sharedReg.syncOffset = 0;
    vrf_rfIxReg.sharedReg.timeDrift = 0;

#ifdef MOD_STLC
    vrf_rfIxReg.sharedReg.slotOffset = 0;
#endif

    AssertFatal ( 0== pthread_mutex_unlock(&(vrf_rfIxReg.sharedReg.sharedMutex)), "");
```

slotOffset 정보 RF에 업데이트!!

slotOffset 초기화!!

보정한 slot대로 slot이 돌아가도록 RF에도 설정!!

```
//slot decoding --------
AssertFatal ( 0== pthread_mutex_lock(&(vrf_rfIxReg.roReg.regMutex)), "");
vrf_rfIxReg.roReg.slot_nr = slot_nr;
AssertFatal ( 0== pthread_mutex_unlock(&(vrf_rfIxReg.roReg.regMutex)), "");

if (vrf_checkRxStatus(slot_nr) == 0)
{
    vrf_readSlot(&timestamp, slot_nr, deltaDrift);
    vrf_irqRx(slot_nr);
}
else
{
    LOG_E(PHY, "[WARNING] RX is in busy, so passing the slot %i\n", slot_nr);
    vrf_bypassSlot(&timestamp);
}
```

DL에서 slotOffset으로 slot_nr보정!!

1. 코드 구현

1-1) TX와 RX의 slot번호 맞추기(동기화)

[RF단] -> slot_nr, sched_slot_nr, syncStatus

RX_UL

```
case rfmode_rx:

    //slot operation (TX or RX)
    #ifdef MOD_STLC
    AssertFatal ( 0== pthread_mutex_lock(&(vrf_rfIxReg.sharedReg.sharedMutex)), "");
    int syncStatus = vrf_rfIxReg.sharedReg.syncStatus;
    AssertFatal ( 0== pthread_mutex_unlock(&(vrf_rfIxReg.sharedReg.sharedMutex)), "");

    if(syncStatus == 1 && vrf_fdbkConfig.slotBmp & (0x01<<slot_nr))  //UL sendslot (UE to BS : pilot transmission)
    {
        LOG_I(PHY, "Uplink in slot : %i\n", slot_nr);

        vrf_bypassSample(&timestamp, vrf_samples_in_slot-vrf_IQSendMargin-vrf_SchedTimeSample);  //Guard Band

        vrf_irqRx(slot_nr);
        usleep(vrf_SchedTime);

        //sending fdbk slot (preamble & pilot)
        vrf_sendFdbk(&timestamp, slot_nr);
        //bypass the rest
        vrf_bypassSample(&timestamp, vrf_IQSendMargin+vrf_SchedTimeSample);
    }
```

sycnStatus 정보 RF에 업데이트!!

sycnStatus = 1
즉 싱크가 맞춰진 상태에서
보정된 slot_nr을 사용하여 fdbk채널 시작!!

[RX단]

insertPreamble 함수를 이용하여 vtx_freqBuffer 1, 3 slot에 preamble 넣어주기!!

```
void vrx_insertPreamble(vrx_cfgInfo_t* cfgInfo, vrx_rtCfg_t* rtCfgInfo)
{
    uint16_t symbol_bitmap = (0x01 << VHW_PREAMBLE_SYMBOL);
    int k = cfgInfo->fftSize - cfgInfo->preamble_length/2;

    for (int i=0;i<cfgInfo->preamble_length;i++)
    {
        vtx_freqBuffer[0][VHW_PREAMBLE_SLOT_UL_1][VHW_PREAMBLE_SYMBOL][k] = cfgInfo->preambleSeq[i];
        vtx_freqBuffer[0][VHW_PREAMBLE_SLOT_UL_3][VHW_PREAMBLE_SYMBOL][k] = cfgInfo->preambleSeq[i];
        vtx_freqBufferBp[0][VHW_PREAMBLE_SLOT_UL_1][VHW_PREAMBLE_SYMBOL] = 1;
        vtx_freqBufferBp[0][VHW_PREAMBLE_SLOT_UL_3][VHW_PREAMBLE_SYMBOL] = 1;

        k = (k+1)%cfgInfo->fftSize;
    }

    //clear the time buffer first
    vrx_clearBuf(cfgInfo, (0x01 << VHW_PREAMBLE_SLOT_UL_1)); //make vtx_clearBuf
    vrx_clearBuf(cfgInfo, (0x01 << VHW_PREAMBLE_SLOT_UL_3));

    //RX ANT 0 only have preamble, ANT2 preamble x

    vrx_modSlot(cfgInfo, 0, VHW_PREAMBLE_SLOT_UL_1, symbol_bitmap);
    vrx_modSlot(cfgInfo, 0, VHW_PREAMBLE_SLOT_UL_3, symbol_bitmap);
} « end vrx_insertPreamble »
```

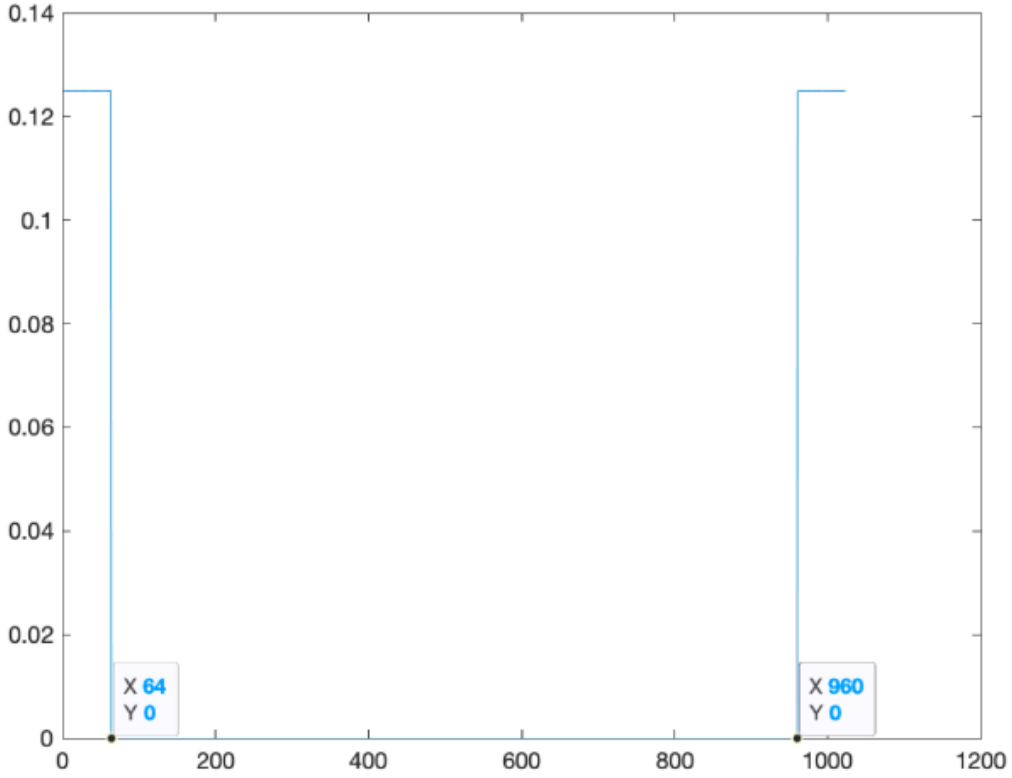
RX Ant_0에만 preamble 넣어주기!!
(SIC(Successive Interference Cancellation)를 위해서)

```
#ifdef MOD_STLC
                else
                {
                    // At 0,1 slot, preamble insert
                    for(int RXANT =0; RXANT<2; RXANT++)
                    {
                        //Send to RF (Data)
                        AssertFatal ( 0== pthread_mutex_lock(&(vrx_rfRegPtr->sharedReg.sharedMutex)), "");
                        memcpy(&vrx_rfRegPtr->sharedReg.txData[procCnt][RXANT][0],&vtx_buffer[RXANT][procCnt][0], vrx_rfRegPtr->roReg.slotSize*sizeof(int));
                        AssertFatal ( 0== pthread_mutex_unlock(&(vrx_rfRegPtr->sharedReg.sharedMutex)), "");

                    }

                }
#endif
```

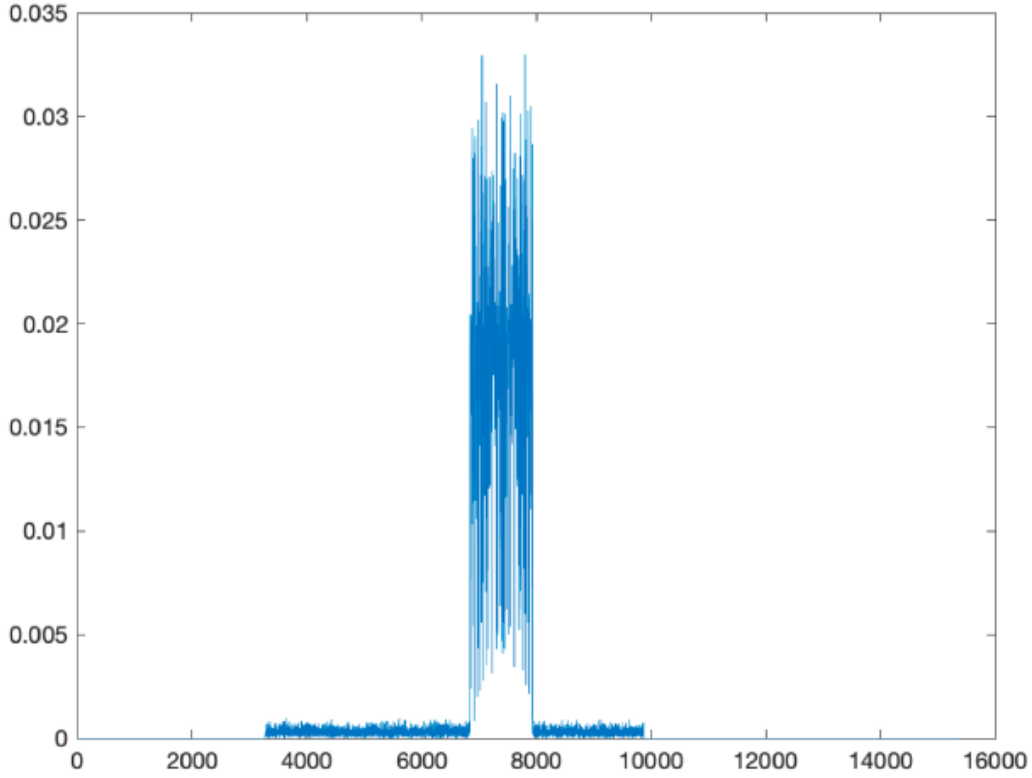
안테나 별로 slot번호에 맞는 buffer를 RF에 업로드하기!!

[RX단 - 결과]

Preamble을 M-sequence를 사용함
⇒ Preamble length : 128
⇒ fftSize : 1024



Frequency domain에서 본 insert preamble



Time domain에서 본 insert preamble

1. 코드 구현

1-2) RX preamble insert & TX preamble detection

[TX단]

```
//need to be verified
static int vtx_detectPreamble (vtx_cfgInfo_t* cfgInfo,
                               uint8_t proc_nr,
                               int *freqOffset,
                               int detect_offset,
                               int state_sync)

{
    AssertFatal ( 0== pthread_mutex_lock(&(vtx_rfRegPtr->roReg.regMutex)), "");
    int slotSize = vtx_rfRegPtr->roReg.slotSize;
    AssertFatal ( 0== pthread_mutex_unlock(&(vtx_rfRegPtr->roReg.regMutex)), "");

    //modify window Size
    int start = 6400;
    int end = 7200;
    int64_t result = 0, max_result = 0;
    int64_t av_result = 0;
    int max_offset = 0;
    int cnt = 0;
    int16_t maxval = 0;
    int shift;
    int16_t *ptr16;
    double f_off;
#ifdef VRX_CORHISTDUMP
    int64_t corr_hist[10000];
#endif


    LOG_D(PHY, "preamble detection trial in state %i and slot %i\n", state_sync, proc_nr);

    short vrx_preambleSeqBuf[2048];
    int vrx_preambleLen = 1024;

    memcpy(vrx_preambleSeqBuf,(short *)(&vtx_buffer[0][2][cfgInfo->tOffset_preamble]), cfgInfo->fftSize*sizeof(short)*2);

    //----------------------------------------------------------------------

    //amplitude estimation ------------------------------------------------
    ptr16 = &(vrx_preambleSeqBuf[0]);
    for (int i=0; i<2*vrx_preambleLen; i++)
    {
        maxval = max(maxval,ptr16[i]);
        maxval = max(maxval,-ptr16[i]);
    }
    shift = vhw_log2Approx(maxval);

    //window search command
    if (detect_offset > 0)
    {
        start = detect_offset - VTX_SYNC_WINDOW_SIZE/2;
        end = detect_offset + VTX_SYNC_WINDOW_SIZE/2;
    }

    //----------------------------------------------------------------------
```

RX에서 동기화를 한 후에, preamble을 보냈기 때문에, window detection 진행!!

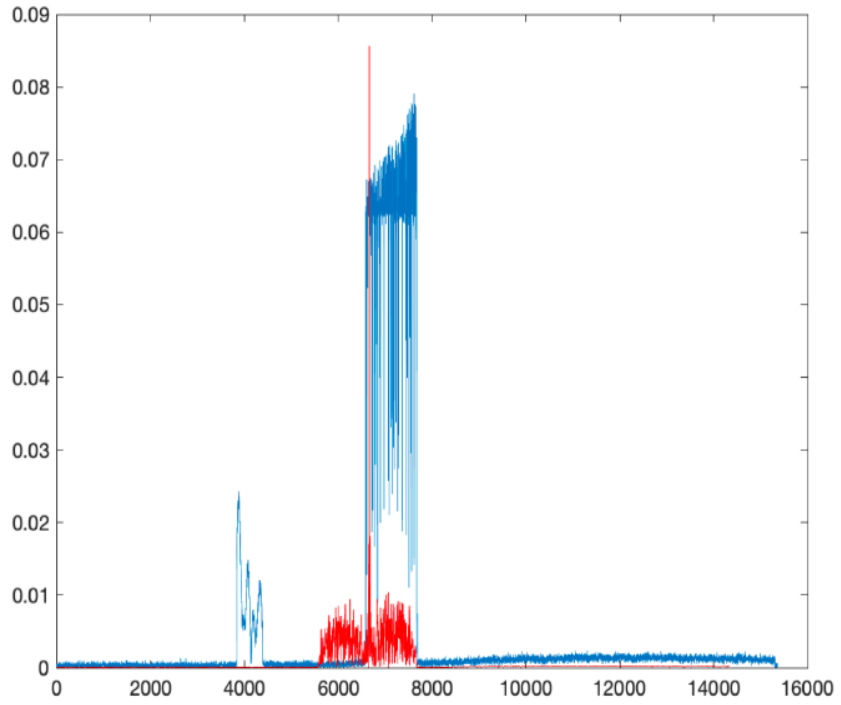
```
#ifdef VTX_DBGPROC
LOG_I(PHY, "preamble detection start :::::: start : %i, end : %i\n", start, end);
#endif
for (int n=start; n < end; n+=4) // n= 0 ~ end-4 (slotSize)
{
    if ( n < (slotSize - vrx_preambleLen) )
    {
        int64_t dotResult   = dot_product64(&(vrx_preambleSeqBuf[0]),
                              (short*) &(vtx_rfRegPtr->roReg.rxData[proc_nr][0][n]),
                                vrx_preambleLen,
                                shift);

        result = vhw_abs64(dotResult);
        av_result += result;

        /* calculate the absolute value of sync_corr[n] */
        if (result > max_result)
        {
            max_result = result;
            max_offset = n;
        }
        cnt++;
    }
} « end for int n=start;n<end;n+=4 »

av_result /= cnt;

LOG_D(VTX, "[detection result] max_offset : %i \n" , max_offset);


if ((detect_offset == 0 && max_result < VTX_DEFAULT_THRES_PREAMBLEDETECTION*av_result) ||
    (detect_offset > 0 && max_result < VTX_ONLINE_THRES_PREAMBLEDETECTION*av_result))
{
    LOG_D(VRX, "peak-to-average is below threshold! (max:%li av:%li) (start:%i, end:%i) end with assertion!\n", max_result, av_result, start, end);
    return(-1);
}

//f_off estmation :: angle, digital freq ----------------------------------
f_off = vtx_estFreqOffsetFromPss(&(vrx_preambleSeqBuf[0]), (short*) &(vtx_rfRegPtr->roReg.rxData[proc_nr][0][max_offset]), vrx_preambleLen, shift);

//freqOffset = analog freq
*freqOffset = (int)(f_off*(15360000.0/vrx_preambleLen));

if (start == 0) // if it is full search
    LOG_I(PHY, "[VRX] Found preamble - offset : %i, max result : %li, av_result : %li, frequency offset : %lf (%lf)\n",
                               max_offset, max_result, av_result, f_off, f_off*(15360000.0/vrx_preambleLen));

if (*freqOffset > 300 || *freqOffset < -300)
    LOG_E(PHY, "[VRX] WARNING ::: too large frequency offset : %i, needs to be compensated immediately\n", *freqOffset);


return(max_offset);
```

받은 preamble과 원본 preamble을
dot_product를 사용하여 Preamble detection을 함

max_offset을 찾아서 preamble로 timing을 찾음!!

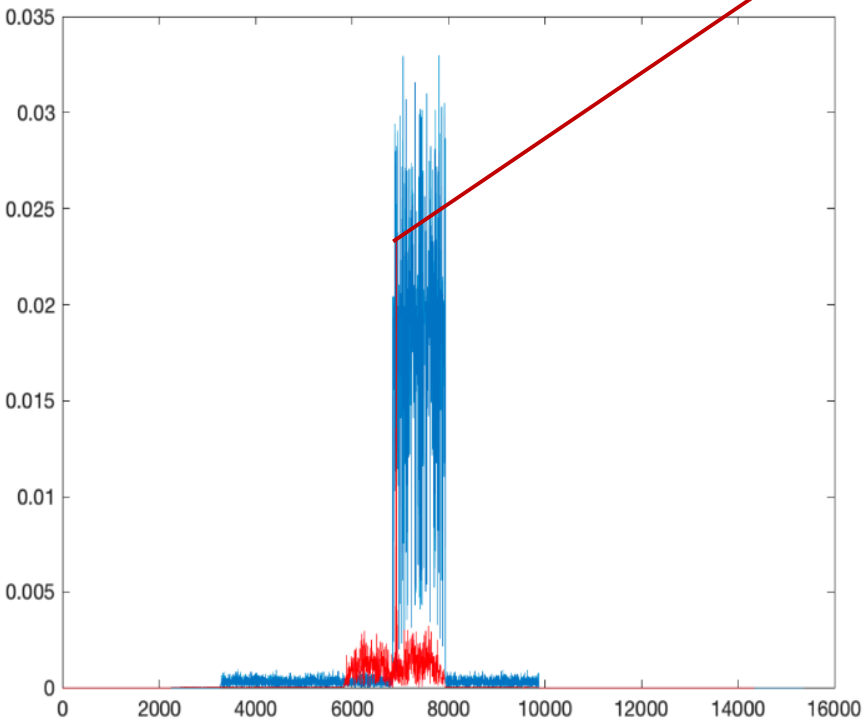
[TX, RX단에서 추출한 preamble 결과]

RX에서 preamble이 detection은 6664가 나옴!!



DL(Slot #2)

TX에서 preamble이 detection된 시간은 6840 또는 6844로 일정한 결과가 나옴!!



UL(Slot #3)

```
[PHY] [VRX] WARNING ::: too large frequency off
sated immediately
[PHY] detection success (time offset : 6844)
slot_nr : 2 ,count1024: 907
[PHY] [VRX] WARNING ::: too large frequency off
sated immediately
[PHY] detection success (time offset : 6844)
slot_nr : 0 ,count1024: 908
[PHY] [VRX] WARNING ::: too large frequency off
sated immediately
[PHY] detection success (time offset : 6844)
slot_nr : 2 ,count1024: 909
[PHY] [VRX] WARNING ::: too large frequency off
sated immediately
[PHY] detection success (time offset : 6844)
slot_nr : 0 ,count1024: 910
```

```
[PHY] [VRX] WARNING ::: too large frequency offs
sated immediately
[PHY] detection success (time offset : 6840)
slot_nr : 2 ,count1024: 1013
[PHY] [VRX] WARNING ::: too large frequency offs
sated immediately
[PHY] detection success (time offset : 6840)
slot_nr : 0 ,count1024: 1014
[PHY] [VRX] WARNING ::: too large frequency offs
ated immediately
[PHY] detection success (time offset : 6840)
slot_nr : 2 ,count1024: 1015
[PHY] [VRX] WARNING ::: too large frequency offs
sated immediately
[PHY] detection success (time offset : 6840)
slot_nr : 0 ,count1024: 1016
```

```
//modify window Size
int start = 6400;
int end = 7200;
```

디버깅할 때, 계산시간을 보고 부족하면
Window detection을 더 작은 범위로 할 수 있음!!

[TX, RX단에서 추출한 preamble 결과]

```
[PHY] [VRF] 1 slot processing......
[PHY] [VRF] schedule IRQ for slot 2 (fdbk), rest:3280
[VHW]  >>>> 3/4 rx :   : 697 us
[PHY] [vTX] slot IRQ from RF (2)
[VHW] >>>> signal to L1 : 1 us
slot_nr : 2 ,count1024: 1019
[VHW] >>>> L1 schedule indication : 4 us
[VHW] >>>>>> VTX start preamble detection : 0 us
[PHY] preamble detection start :::::: start : 6400, end : 7200
[PHY] [VRX] WARNING ::: too large frequency offset : 1174, needs to be compen
sated immediately
[VHW] >>>>>> VTX end preamble detection : 65 us
[PHY] detection success (time offset : 6844)
[VHW] >>>> VTX start modulation : 1 us
[VHW] >>>> VTX end modulation : 15 us
[PHY] [vTX] waiting scheduling IRQ
[VHW] >>>>>> return back to origin : 5 us
[VHW]  >>>> sendUSRP end :   : 125 us
[VHW]  >>>> receiveUSRP end :   : 207 us
[VHW] --> slot interval : 1035 us
```

```
//modify window Size
int start = 6400;
int end = 7200;
```

vThread_tx , vThread_RF에서 디버깅할 때 사용
#define VTX_DBGPROC
#define VRF_DBGPROC

Window detection을 진행하는데 65us 걸림!!

1 SLOT의 데이터를 처리하고 보내는데
총 1ms 정도 소요되는 것을 확인가능!!
⇒ 이후 STLC modulation 걸리는 시간을 고려하여 범위를 조정할 계획

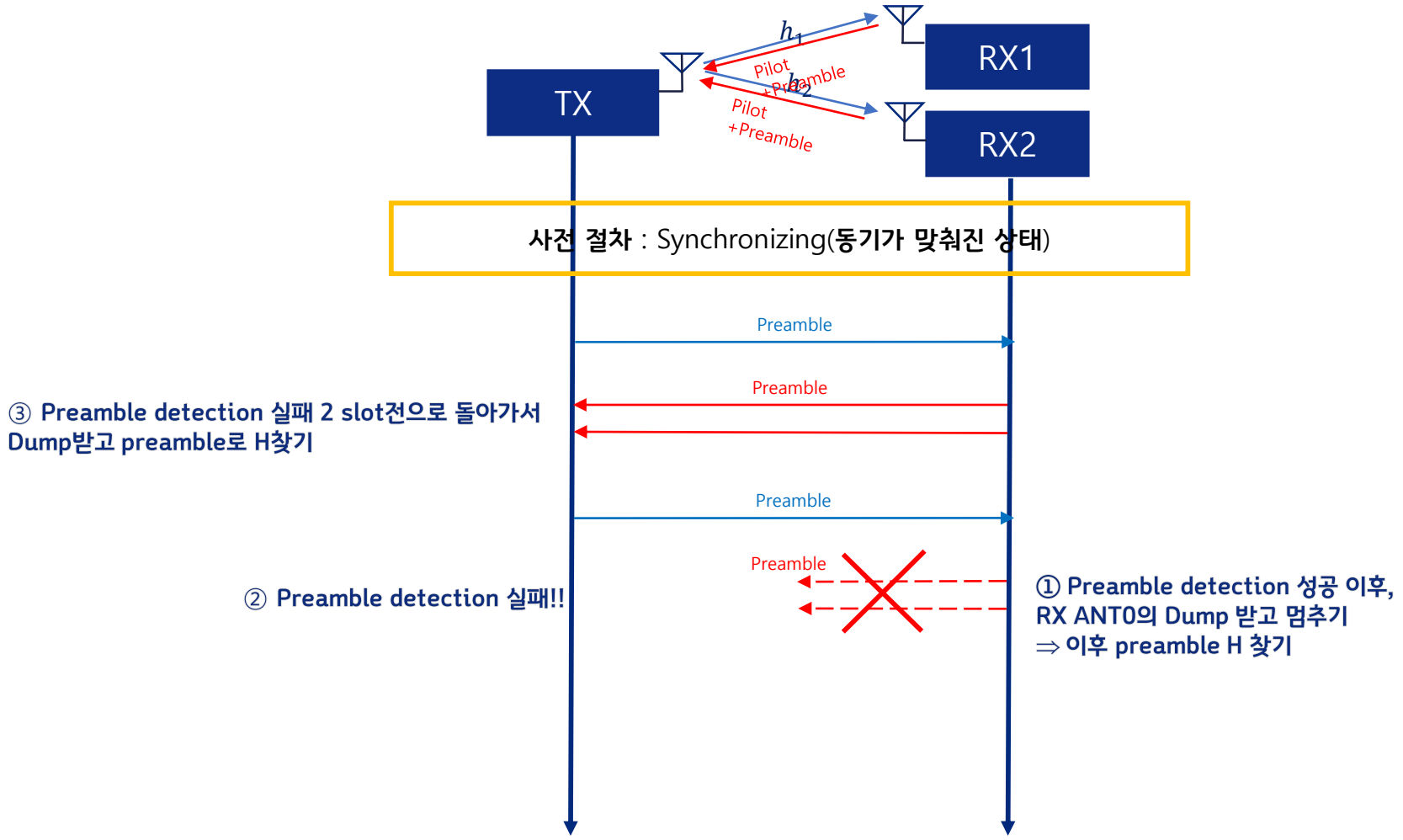
```
#ifdef MOD_STLC
        if (rtCfgInfo.slot_nr_sch%2 == 0)
        {
#endif

#ifdef MOD_STLC
        if(cnt > 300) //after DL Sync
        {
            slot_nr = (rtCfgInfo.slot_nr_sch+3)%4 ;

            #ifdef VTX_DBGPROC
            vhw_timeStamp(&timer, ">>>>>> VTX start preamble detection");
            #endif

            time_offset = vtx_detectPreamble(&cfgInfo, slot_nr , &freqOffset, detect_offset, syncStatus);

            #ifdef VTX_DBGPROC
            vhw_timeStamp(&timer, ">>>>>> VTX end preamble detection");
            #endif
```

Slot_nr_sch : 미래의 slot (현재 slot+1)
Slot_nr : 현재의 slot

Slot이 1개씩 넘어갈 때마다 cnt가 올라감
⇒ 현재 cnt>300 이면 detection 시작!!
⇒ sync를 잡은 이후부터 preamble detection 하도록 코딩해 놓지는 않음 (임의로 300으로 잡아둠)

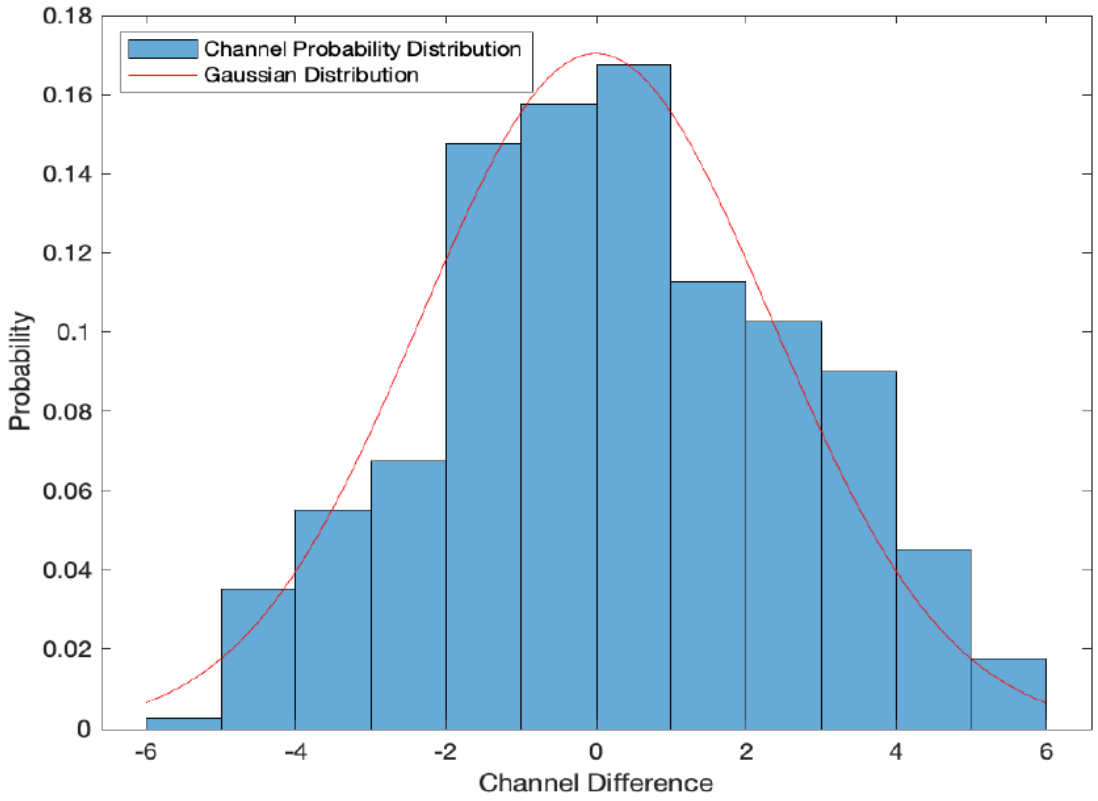
2. Preamble 을 이용한 Channel estimation

[channel estimation 방법1]



2. Preamble 을 이용한 Channel estimation

[channel estimation 방법1 결과 및 문제점]



평균 : 0.2767
표준편차 : 2.3425
표본 : 400개

결론
: UL, DL 채널의 각도 차이가 평균을 0으로 갖는 가우스 분포를 띔
: 높은 확률로 UL, DL의 각도가 일치함
: 인접한 시간 내 UL에서 추정한 CSI를 사용한다면, TX에서 인코딩 과정을 수행하는 STLC에서도 우수한 성능 보장이 가능할 것으로 기대됨

But, 분포가 완벽히 일치하지는 않는다는 문제점이 있음.

문제점. 데이터를 추출하는 방식
⇒ 앞에서 생각한 channel estimation이 부정확할 가능성이 존재함
⇒ 연속 slot이 아닌 데이터를 뽑아서 채널을 추정했을 가능성이 존재함
⇒ 확실한 방법으로 채널을 추정할 필요성이 있음
⇒ 다른 방법으로 연속된 채널을 추정해보기로 함

Data 값을 이용하여 연속인지 판별한 다음
Dump를 받아 channel estimation을 하기로 함!!

2. Preamble 을 이용한 Channel estimation

[channel estimation 방법2]

NPHY_nr-ue.c

```
#ifdef bitsend
void mktxdatabuf(void){

    int slotCnt10 = 0;
    int p0 = 0;
    int p1 = 0;
    int p2 = 0;
    int p3 = 0;
    int p4 = 0;
    int p5 = 0;
    int p6 = 0;
    int p7 = 0;
    int p8 = 0;
    int p9 = 0;

    for(int i=0; i<1024; i++)
    {

        p0 = slotCnt10/512 ;
        p1 = (slotCnt10 %512)/256;
        p2 = ((slotCnt10 %512)%256)/128;
        p3 = (((slotCnt10 %512)%256)%128)/64;
        p4 = ((((slotCnt10 %512)%256)%128)%64)/32;
        p5 = (((((slotCnt10 %512)%256)%128)%64)%32)/16;
        p6= ((((((slotCnt10 %512)%256)%128)%64)%32)%16)/8;
        p7= ((((((((slotCnt10 %512)%256)%128)%64)%32)%16)%8)/4;
        p8= (((((((((slotCnt10 %512)%256)%128)%64)%32)%16)%8)%4)/2;
        p9= (((((((((slotCnt10 %512)%256)%128)%64)%32)%16)%8)%4)%2;

        txdatabuf10[i][0] = p0;
        txdatabuf10[i][1] = p1;
        txdatabuf10[i][2] = p2;
        txdatabuf10[i][3] = p3;
        txdatabuf10[i][4] = p4;
        txdatabuf10[i][5] = p5;
        txdatabuf10[i][6] = p6;
        txdatabuf10[i][7] = p7;
        txdatabuf10[i][8] = p8;
        txdatabuf10[i][9] = p9;

        slotCnt10 = slotCnt10 +1;

        } « end for inti=0;i<1024;i++ »
} « end mktxdatabuf »
#endif
```

txdatabuf[1024][10] 채우기!!

0 0 0 0 0 0 0 0 0 0 ⇒ 0

10 bit Data

1 1 1 1 1 1 1 1 1 1 ⇒ 1023

Data scaling !!

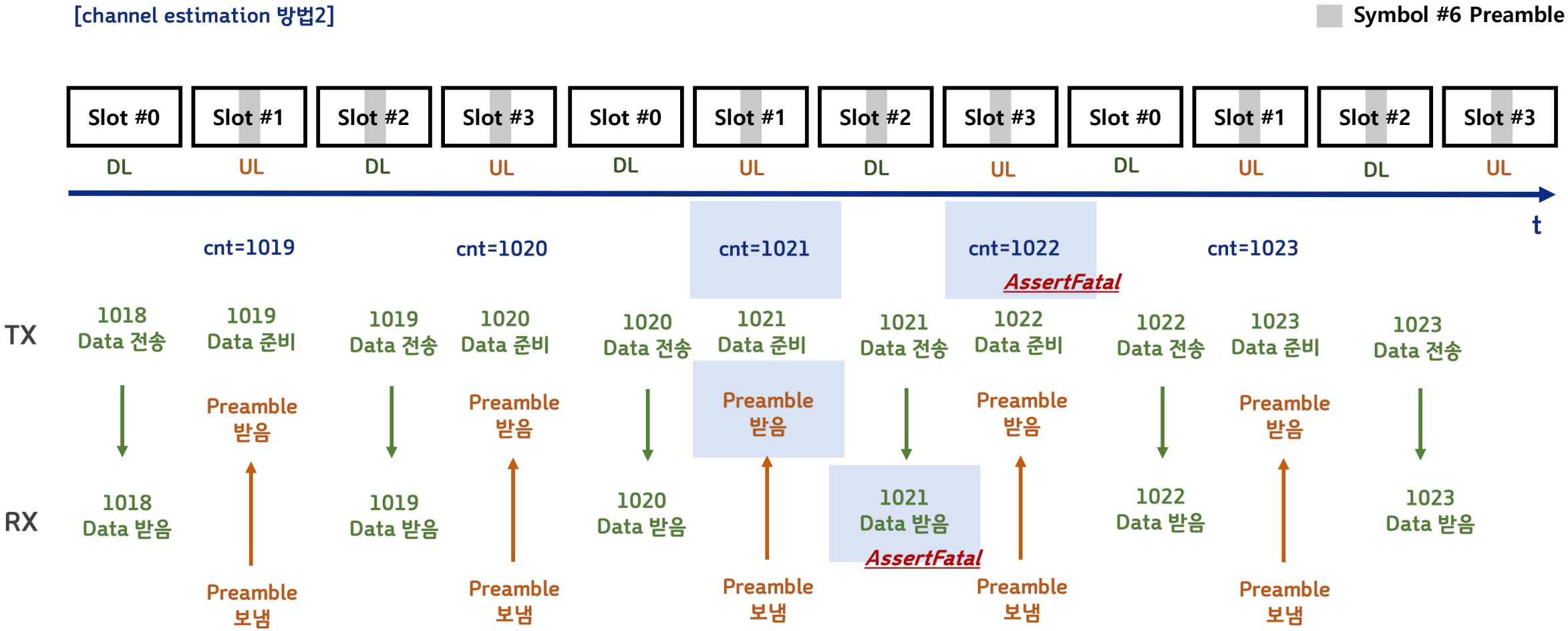
```
#ifdef bitsend
    //When bit send, scaling
    for(int j=0 ; j<1024; j++)
    {
        for(int i=0; i<10; i++)
        {
            if(txdatabuf10[j][i]==1)
            {
                txdatabuf10[j][i] = 5792;
            }
            else
            {
                txdatabuf10[j][i] = -5792;
            }
        }
    }

#endif
```

Tx에서 data를 1번 보낼 때마다 count1024 커짐!!
⇒ count1024를 보고 몇 번 buffer를 보내는지 확인가능

```
#ifdef bitsend
                printf("slot_nr : %d ,count1024: %d \n",slot_nr, count1024);

                if((slot_nr%2)==0)
                {
                    memcpy(&txdata10[0], &txdatabuf10[count1024][0], (10)*sizeof(short));

                    count1024 = (count1024+1)%1024;
                }

                NHAL_TXcmd_cfgTxData(dataLength,txdata10);
#endif

                NHAL_TXcmd_startTx();

                break;
```

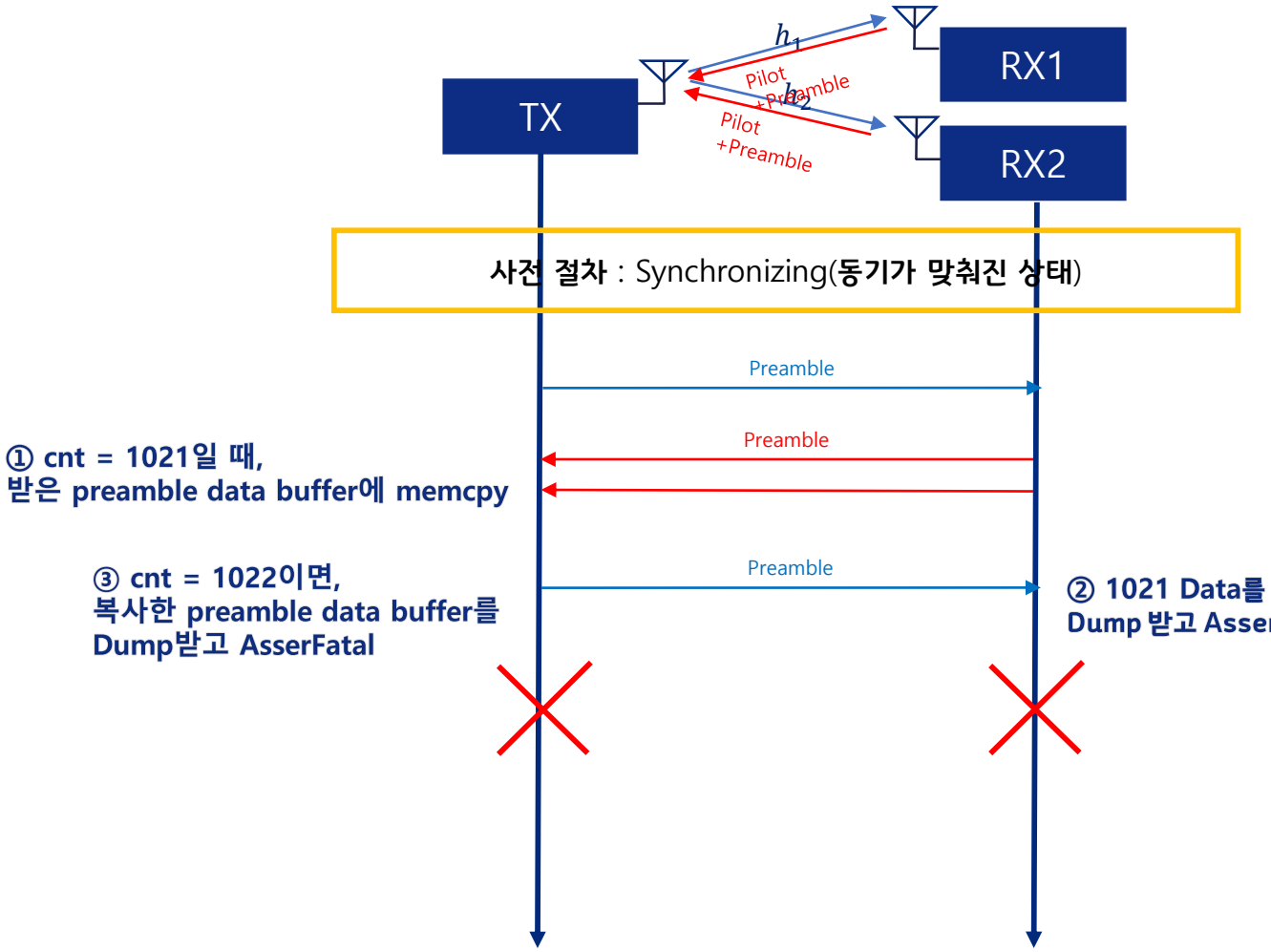
2. Preamble 을 이용한 Channel estimation

[channel estimation 방법2]



2. Preamble 을 이용한 Channel estimation

[channel estimation 방법2]



1 1 1 1 1 1 1 1 0 1

1021 Data decode

2. Preamble 을 이용한 Channel estimation

[channel estimation 방법2 구현 설명]

```
#define bitsend
#define symbolsend
```

⇒ NPHY_nr-ue.c , vThreadtx.c, vThreadrx.c

⇒ Bit 또는 symbol로 보낸 데이터를 확인하고 싶을 때 사용하는 변수

TX

```
#ifdef bitsend
if(cnt == 1021)
{
    memcpy(&vrx_preambleDump[0][0][0], &vtx_rfRegPtr->roReg.rxData[slot_nr][0][0], rtCfgInfo.slotSize*sizeof(int));
}

if(cnt == 1022)
{
    vhw_dumpDataToFile(&vrx_preambleDump[0][0][0], rtCfgInfo.slotSize, "tx_preamble.txt");
    AssertFatal(0,"Dump for Channel estimation by preamble! bye bye\n");
    #endif
}
#endif
```

cnt=1021에서 buffer를 복사한 후, cnt=1022에서 file로 받기

⇒ Dump를 받는데 시간이 너무 오래 걸리기 때문에

RX

```
//decode Data
#ifdef MOD_STLC

vrx_decodeData(&cfgInfo, procCnt, procCnt);

//for each rxant MLdetection
vrx_MLdetector(&cfgInfo, (short*)(&(vrx_decDataBuf[0][procCnt][0])), &(vrx_MLdataBuf[0][0]));
vrx_MLdetector(&cfgInfo, (short*)(&(vrx_decDataBuf[1][procCnt][0])), &(vrx_MLdataBuf[1][0]));

#ifdef bitsend
    vrx_Dumppreamble(&cfgInfo, procCnt, time_offset);
#endif
```

안테나 별로 data를 decoding한 후,

Data가 1021이면 dump를 받기!!

2. Preamble 을 이용한 Channel estimation

[channel estimation 방법2 구현 설명 – 주의해서 볼 부분]

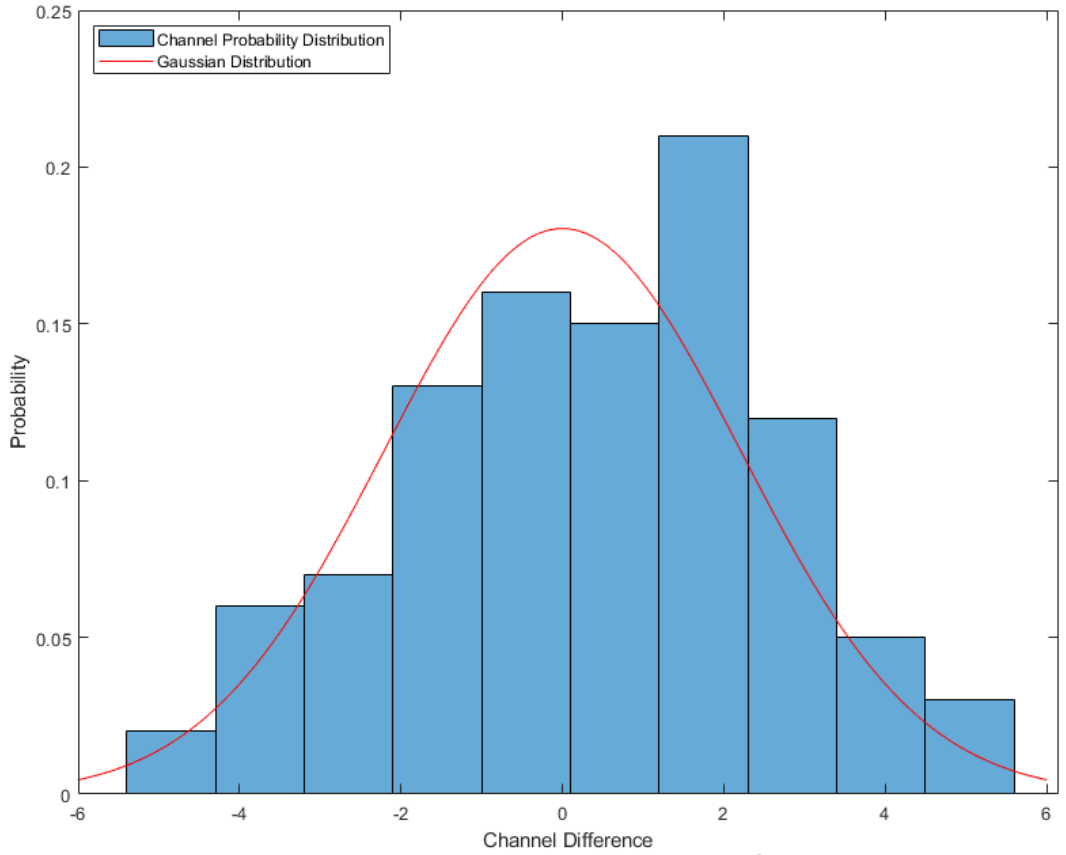
```
#if (defined MOD_STBC || defined MOD_STLC)
static int vrx_decodeSymbol(vrx_cfgInfo_t *cfgInfo, int linkID, int slot_nr, int decLen, int symOffset, int dmrs_nr, int dmrs_symb_nr, int decOffset)
#else

case hw_trxMode_1x2STLC:
    // shift scaling check
    // if wireless channel,
    // Ant 0 => chEst scaling = 7, vrx_decDataBuf scaling = 8 (with no attenuator)
    // Ant 1 => chEst scaling = 9, vrx_decDataBuf scaling = 10 (with no attenuator)
    // else if wire channel, check num according to attenuator scale

    //channel estimation
    if (rxAntID == 0)
        chEst = vhw_calcConjMult(ptrPil, ptrRx, 9);
    else if(rxAntID == 1)
        chEst = vhw_calcConjMult(ptrPil, ptrRx, 9);

    //chEst = vhw_calcConjMult(ptrPil, ptrRx, VRX_DMRS_AMP);
    shift = vhw_calcAmplitude((short*)&chEst);
    //chEstbuffer[i] = chEst;

    if (symIndex == 0)
    {
        short *chEstPtr = (short*)&chEst;
        LOG_D(VRX, "channel estimation (link:%i (tx%i/rx%i/t%i) (slot %i, sym:%i, dmrs_NR:%i, dmrs_symb_nr : %i, k %i) pilot : (%i %i), rx : (%i, %i) -
                                linkID, txAntID, rxAntID, symIndex, slot_nr, symOffset, dmrs_nr, dmrs_symb_nr, k, ptrPil[0], ptrPil[1], ptrRx[0], pt

        //channel compensation
        if (rxAntID == 0)
            vrx_decDataBuf[rxAntID][slot_nr][decOffset+i] = vhw_calcConjMult((short*)&chEst, ptrData, 10);
        else if(rxAntID == 1)
            vrx_decDataBuf[rxAntID][slot_nr][decOffset+i] = vhw_calcConjMult((short*)&chEst, ptrData, 10);


        #ifdef userxant1
            if (rxAntID == 1)
                vrx_decDataBuf[rxAntID][slot_nr][decOffset+i] = vhw_calcMult((short*)&vrx_decDataBuf[rxAntID][slot_nr][decOffset+i],exphalfpi,0);
        #endif

        decCnt++;
    } « end if symIndex==0 »

    break;
```

안테나 별로 data를 디코딩을 할 때,
Data의 신호의 세기가 아직 적절히 정해지지 못함.
⇒ 통신 환경에 따라서 scaling을 바꿔줄 필요가 있음
(무선, 유선, attenuator의 세기에 따라 달라지므로
실험 환경을 세팅하기전에 신호의 세기를 보고 판단해야 함)
⇒ scaling이 잘못될 경우, Overflow가 날 수 있음
⇒그럼 디코딩이 안돼서 제대로 Dump를 받을 수 없음
⇒(Dump가 안받아질 때, scale 확인하기)

또한 안테나를 2개 쓰는 $1 \times 2\ STLC$ 실험에서 안테나1에서
디코딩시 phase가 $\frac{\pi}{4}$ 가 돌아가는 현상이 발생함!!
⇒ 현재는 안테나 1을 $\frac{\pi}{4}$를 보정해주는 상태임
⇒ USRP문제인지 살펴봐야함

2. Preamble 을 이용한 Channel estimation

[channel estimation 방법2의 결과]



확률분포표에 가우스 그래프를
겹쳐서 그려본 결과

평균 : 0.3257
표준편차 : 2.2109
표본 : 100개

결론
: UL, DL 채널의 각도 차이가 평균을 0으로 갖는 가우스 분포를 띔
: 높은 확률로 UL, DL의 각도가 일치함

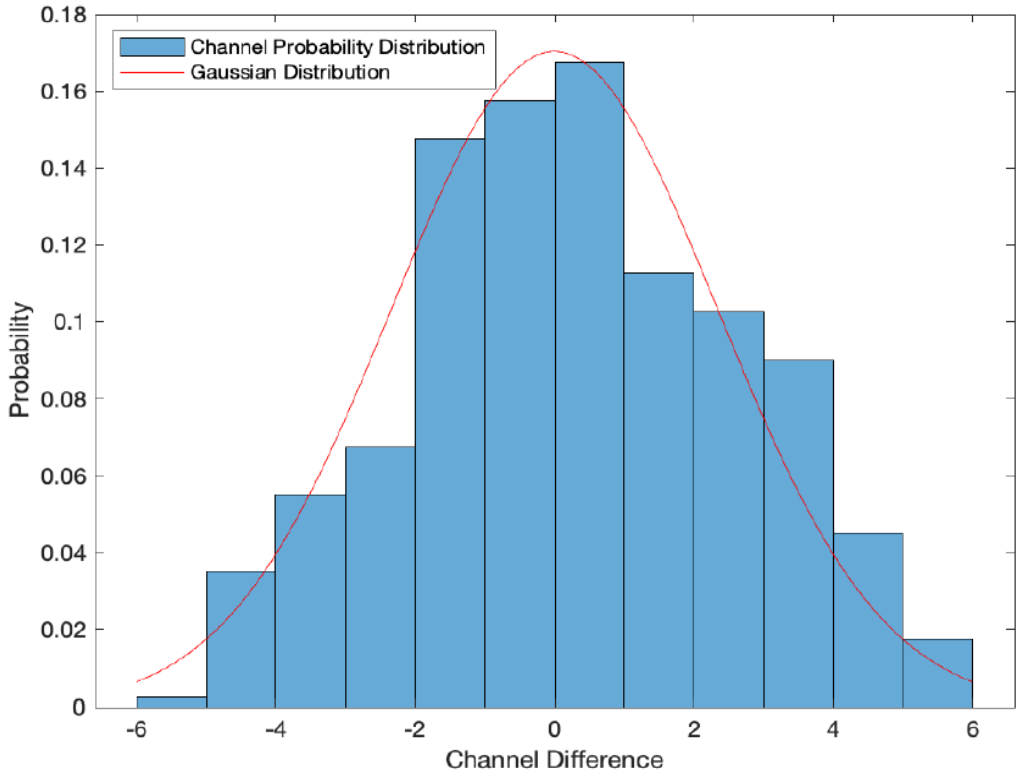
But, 여전히 분포가 완벽히 일치하지는 않는다는 문제점이 있음.

문제점1. 데이터가 충분하지 않은 것
⇒ 데이터가 100개밖에 되지 않기 때문에 정교한 확률분포표를 못 만듦
⇒ 이후 추가적인 실험을 통해 데이터를 더 뽑을 예정임

문제점2. 여전히 데이터의 각도의 차이가 0에 수렴하지 않음
⇒ 다른 원인에 의해서 channel의 phase차이가 발생하였다고 예측
⇒ UL, DL의 각도의 차이를 만든 원인을 추측해보기로 함

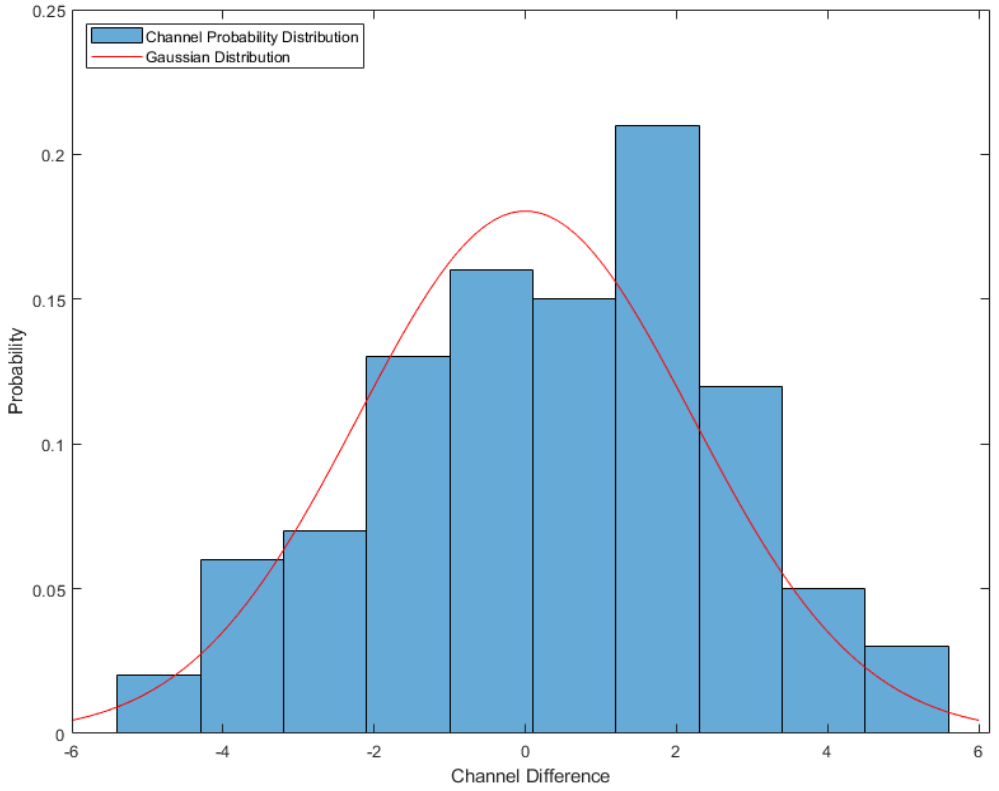
3. Channel estimation 결과 분석 및 추후 계획 설명

[channel estimation 방법2 결과 및 방법1과의 비교]



channel estimation 방법1

평균 : 0.2767
표준편차 : 2.3425
(표본 : 400개)



channel estimation 방법2

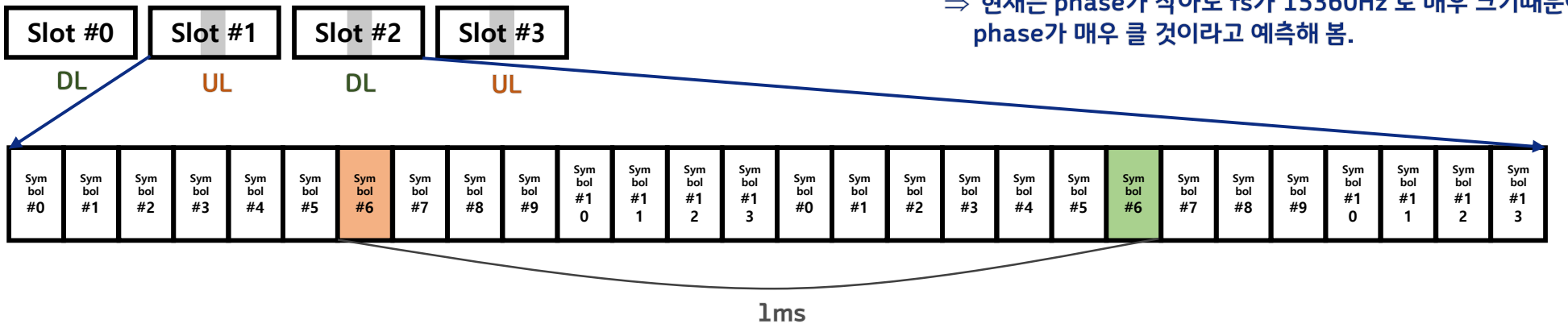
평균 : 0.3257
표준편차 : 2.2109
(표본 : 100개)

3. Channel estimation 결과 분석 및 추후 계획 설명

[channel estimation phase 차이가 나는 원인 추측]

원인 1) 1ms 동안 phase가 많이 돌아가기 때문에 frequency offset이 발생하여 각도의 차이가 발생함

⇒ 이후, 실제 주파수의 속도를 계산하여 slot마다 돌아가는 angle을 계산해볼 예정.
⇒ 현재는 phase가 작아도 fs가 15360Hz 로 매우 크기때문에 1ms마다 돌아가는 phase가 매우 클 것이라고 예측해 봄.



$\cos(2\pi f_c t + 2\pi f_e t)$

$\Delta\theta = 2\pi f_e t$ 이 때, $t = 1ms$, $f_e = -300 \sim 300Hz$ 면, $\Delta\theta = -1.89 \sim 1.89\ rad$

원인 2) RX에서는 frequency offset을 보정해주는 코드가 있는 반면, TX에서는 frequency offset을 보정해주는 코드가 없어서 이러한 오차 값이 발생함

THANK YOU!