Abstract—Recently, Noam et al. ([1]) proposed the BNSL algorithm, based on the blind Jacobi eigenvalue method, in order to learn fast and accurately the null-space of the channel matrix between a secondary transmitter and a primary receiver. In this paper, we propose a channel tracking variation of the algorithm that allows simultaneous transmission of information to the Secondary Receiver (SR) along with the learning of the null-space of the time-varying target channel. Specifically, the enhanced algorithm initially performs a BNSL sweep as described in [1] in order to acquire the null space. Then, it keeps performing modified Jacobi rotations in a way that it keeps the induced interference to the primary receiver lower than a given threshold $P_{Th}$ with probability $0.95$ and it transmits information to the SR simultaneously. We present simulation results that indicate that the proposed approach works for channels with independent Rayleigh fading with small doppler frequency.

I. INTRODUCTION

MIMO communication can reform the way we think about wireless communications. One of the areas that MIMO can actually turn out to be significantly revolutionary is in the paradigm of the cognitive radios. The motivation of the cognitive radios is well-known and has been addressed in several papers in recent years.([5], [6])

The starting point of this work is the recent publications from Noam et al. ([1], [2]) where the authors provide a null space learning algorithm with which a secondary transmitter (SU-Tx) can learn where it should not transmit in order not to interfere with the primary receiver (PU-Rx). The algorithm has very nice convergence properties and it works only with energy measurements and independently of the modulation or other transmission parameters of the primary system.

Typical channel estimation problems are divided to channel acquisition and channel tracking ([4]). The channel acquisition refers to the problem of learning without any prior knowledge. The channel tracking refers to the problem of continuing updating the channel, using the previous knowledge. In most of the work in the literature that are trying to tackle either one of those two problems the authors are making too strong assumptions. For example in [3] the authors assume reciprocity of the channels, only TDD transmission to both primary and secondary system and transmission frame structure of the secondary system synchronized with that of the primary system.

Obtaining the null space via the BNSL algorithm as proposed in [1] is not sufficient in practice since the channel is always time-varying. Due to time variations, after the SU-Tx has acquired the knowledge of the null space, it needs to update it regularly. We are able to enhance the system by introducing a tracking algorithm to learn the channel and send information at the same time to the SR.

In section II we present the system model of this work, in section III we indicate the performance metric that we are going to use in order to test the tracking algorithm. Then, in section IV we present simulation results that indicate that the null-space of a MIMO Rayleigh fading channel varies faster than the coherence time could explain and then in section V we indicate necessary modifications in the BNSL algorithm in order to work in a changing environment. Moreover, in section VI we present and analyze a general transmission scheme that can be used in order to perform the BNSL learning together with the transmission of information. Simulation results of the BER performance of this scheme are presented in section VIII and in section IX we indicate some future research directions. Finally, section X concludes the paper.

II. CHANNEL MODEL

Consider the case that the secondary transmitter (SU-Tx) would like to learn the channel matrix $H_{12}(t) \in \mathbb{C}^{N_r \times N_t}$ between the SU-Tx and the PU-Rx, using either the active or the passive BNSL algorithm. $h_{ij}(t)$ denotes the channel between the $i$-th antennas of the SU-Tx and the $j$-th antenna of the PU-Rx. Throughout this paper we consider only flat fading channels. Denote as $T_{FB}$ the amount of time needed from the time that SU-Tx starts transmitting a learning message, until it measures the effect of its learning signal on the PU-Tx’s transmit power. Also, denote as $T_s$ the symbol duration and as $F_c$ the carrier frequency. We make the assumption of Wide Sense Stationary Uncorrelated Scattering (WSSUS) and that the channels $h_{ij}(t)$ have independent Rayleigh fading with maximum doppler frequency $F_d$.

III. PERFORMANCE METRICS

In order to compare the different methods we need to introduce a natural performance metric. The final metric that we are concerned about is the maximum interference caused by the SU-Tx. Denote $H_{12} \in \mathbb{C}^{N_r \times N_t}$ the channel matrix between the SU-Tx and PU-Rx, $T \in \mathbb{C}^{N_t \times (N_t - N_r)}$ the precoding matrix that the SU-Tx uses and $x_2$ the message
that the latter wants to send. Then the maximum interference is defined as:

\[
\|H_{12}T\| = \max_{\|x_2\| \neq 0} \frac{\|H_{12}x_2\|_2}{\|x_2\|_2}
\]  (1)

We will say that the tracking of the null-space is successful if

\[
\Pr \left\{ \frac{10 \log_{10} \|H_{12}(t)T(t)\|_2^2}{\|H_{12}(t)\|_2^2} \leq -P_{Th} \right\} = 0.95
\]

that is if 95% of the time the interference achieved with the use of the precoding matrix \(T(t)\) is at least \(P_{Th}\) dB less than if we use as \(T(t) = I_{N_t \times N_t}\) (i.e. no precoding).

IV. NULL-SPACE VARIATIONS

In this section we motivate the need of using the proposed enhanced version of the BNSL algorithm based on the observation that even in very slowly Rayleigh fading MIMO channels, the null space changes much faster than the coherence time of each of the independent SISO channel can explain.

We know that the coherence time \(T_c\) is a widely accepted characteristic of a channel in order to quantify how fast it changes. We assume that the correlations between flat fades at different times depends only on the time difference \(\Delta t\) and we define the normalized autocorrelation of a SISO channel \(h(t)\) as

\[
\rho_{\Delta t} = \frac{E\{h(t)h(t + \Delta t)^*\}}{E\{|h(t)|^2\}}
\]

then the \(X\%\) coherence time is the value \(\Delta t\) such that \(\rho_{\Delta t} = \frac{X}{100}\). Assuming Clarke’s model, the 50% coherence time is defined as \(T_{c}^{[0.5]} = \frac{\mu}{10F_d} \approx \frac{0.18}{F_c}\). Simulations show that the null space of a MIMO channel with channel matrix \(H_{12}(t)\) changes much faster than \(T_c\). For example, consider a \((3,1)\)-system that for each \(h_{1j}(t)\) we generate independent Rayleigh fading channels with Doppler Frequency of \(F_d = 6.48\) Hz using the Clarke’s model with 40 multi path components. (relative speed of 10 Km/h and carrier frequency of \(F_c = 700\) MHz). In Figure 1 we plot the amplitude of the normalized autocorrelation \(|\rho_{\Delta t}|\) and the quantity \(d_{M1}(\Delta t)\) where

\[
d_{M1}(\Delta t) = 10 \log_{10} \left( E \left\{ \frac{\|H_{12}(t + \Delta t)N(H_{12}(t))\|}{\|H_{12}(t + \Delta t)\|} \right\} \right)
\]  (3)

The motivation of \(d_{M1}(\Delta t)\) is that it calculates the average minimum decrease of interference inflicted to the PU-Rx if the current channel is \(H_{12}(t + \Delta t)\) whereas the SU-Tx transmits using the null space of an outdated channel, i.e. \(N(H_{12}(t))\). Note that \(d_{M1}(\Delta t)\) increases surprisingly fast. Actually, \(d_{M1} \geq -15\) dB for \(\rho_{\Delta t} \leq 0.95!\) This means, that even if the SU-Tx generates instantly the null space of the channel at time \(t\), it will have to generate another estimate after \(T_{c}^{[0.95]}\) seconds, if it needs to cause at least \(-15\) dB interference to the primary system. An intuition behind this observation is that \(H_{12}(t)\) is a time-varying matrix whose elements change approximately every \(T_c\). The null space is a quantity that depends on all the elements of the matrix and therefore it is likely to change must faster. Note that we have verified our results using a different Rayleigh fading generator ((7)) and with several different number of multi path components (from 5 up to 100).

\[\text{Fig. 1: } (3,1)-\text{System. } d_{M1} \geq -15\text{dB for } \rho_{\Delta t} \leq 0.95.\]

V. CHANNEL TRACKING

In this section, we summarize the channel tracking variation of the BNSL algorithm and we present simulation results that indicate that in the case of slow-varying MIMO channels the algorithm successfully tracks the channel. Note that in this section we do not explain how we perform the simultaneous transmissions. This is explained in section VI.

Denote as \(T\) the time needed for a Jacobi sweep to finish. Assume at time \(t\) we have an approximation \(\hat{W}(t)\) of the eigenspace of the eigenspace \(W(t)\) of the channel matrix. Also, consider the strictly monotonous continuous increasing functions \(f(\cdot) : R_{+} \rightarrow R_{+}\) which are employed by the power control mechanism of the primary system. Then define as \(h(x(t)) = f(\|H_{12}(t)x(t)\|)^2\) the effective interference that the SU-Tx causes to the primary system when it transmits \(x(t)\).

Consider the “modified” Jacobi rotation that is described in Table 1. The input of the modified Jacobi rotation is the current estimated eigenspace \(\hat{W}(t)\) along with the parameters that define the search space of the line searches. These need to be optimally tuned in order to satisfy the constraints that the primary system demands.

The channel tracking works as follows. Firstly, the SU-Tx performs a complete sweep of the BNSL algorithm, i.e. it uses \(W(t) = I_{N_{tx} \times N_t}, \theta = \frac{\pi}{2}, \theta_{max} = \frac{\pi}{2}, \eta = 0.1\) ([1]). Note that the SU-Tx performs only one BNSL sweep and not multiple as it is proposed in [1]. This sweep creates the first estimate \(W(t + T)\) of the eigenspace. Note that the eigenspace is created at time \(t + T\) since the sweep need \(T\) seconds to finish. Then, the SU-Tx transmits information using as the precoding matrix \(T = [\tilde{w}_1, \tilde{w}_1, \ldots, \tilde{w}_{N_t - N_{re}}]\), where \(\tilde{w}_i\) are the columns of the matrix \(W(t + T)\). The modified BNSL sweep always returns the eigenspace in a decreasing order of the size of the eigenvector.
corresponding eigenvalues. The SU-Tx continuously senses the transmit power of the PU-Rx. When it senses that the transmit power has increased more than a predefined threshold then it re-estimates the null space of the channel. Since this threshold rule is mostly an implementation issue of the algorithm into real systems, in our current simulations we make the assumption that the SU-Tx knows the quantity \( |\mathbf{H}_{12}(t)|_dB \) and it performs an adaptation whenever \( |\mathbf{H}_{12}(t)|_dB < P_{Tr} \) where \( P_{Tr} = -20 \). The intuition behind this rule is that the SU-Tx should not wait until the current eigenspace has changed significantly since in that case the matrix \( \mathbf{W}(t + T) \) can not be of much help in the subsequent sweep.

In a modified rotation we assume that the SU-Tx already has a sufficiently good approximation of the eigenspace of \( \mathbf{H}_{12}(t) \) and it performs just a local search over the space of the \( \theta \) angles. This has two major advantages. Firstly, the search can be much faster, assuming that you keep the same parameter \( \eta \), and secondly and most significantly, the SU-Tx performs a rotation without inflicting prohibitive levels of interference since it is rotating only around the current null-space.

Specifically, there exists an explicit trade-off between how fast the channel changes and how large should the variable \( \theta_{max} \) be. \( \theta_{max} \) denotes that the SU-Tx should perform a line search in the interval \([-\theta_{max}, \theta_{max}]\). In theory it is known that if your estimate is very close to the real eigenspace then \( \hat{\theta} \approx 0 \) ((11)) and therefore you should not check for the maximizing \( \theta \) over a large interval around 0. On the other hand, if the estimate is severely outdated, either because the SU-Tx waited too long to perform another sweep, or because the doppler frequency is large, then the SU-Tx should search for the best \( \theta \) over a larger interval. For example, in our simulations \( \theta_{max} \) was \( \frac{\pi}{12} \) and \( \frac{\pi}{2} \) for small ((1 \(- 2 Hz\)) and large (> 2 Hz) doppler frequencies respectively. A complete sweep uses \( \theta_{max} = \frac{\pi}{2} \).

Similarly, the value \( \hat{\theta} \) needs to be as small as possible given that we are able to track the channel. More precisely, if the SU-Tx has a good estimate of the eigenspace, a small \( \hat{\theta} \) means that the line search in line 4 of the modified rotation algorithm is performed around the null space of the channel. Then, the SU-Tx manages to adapt and still keep the inflicted interference in low levels.

The parameters \( \hat{\theta}, \theta_{max} \) should be chosen as a function of the doppler frequency \( F_d \). These can be predefined quantities that the SU-Tx chooses after it has estimated the doppler frequency of the channel.

We should mention that in the first line search in the algorithm 1 (line 4) the SU-Tx needs to search over all the space of \( \phi \) since no similar argument is valid, i.e. even the SU-Tx has a good estimation of the eigenspace, the optimal \( \phi \) can be anywhere in the \([-\pi, \pi]\) interval.

In Figure 2 we show an example of the minimum decrease in interference (in \( dB \)) as a function of time when \( \mathbf{H}_{12}(t) \) is a time varying channel with \( F_d = 1 \) Hz and \( F_d = 2 \) Hz respectively. The parameters of the simulations are: \( N_i = 2, N_r = 1, T_s = 66.7 \) mircosecond, \( T_{F0} = 1 \) msecond. In Figure 3 we summarize the final results when we employ the previous approaches averaging over \( 10^6 \) slots for different doppler frequencies \( F_d \).

**Algorithm 1: Modified Jacobi Sweep**

**Input:** \( \mathbf{W}(t), \theta, \theta_{max}, \eta \)

**Output:** \( \hat{\mathbf{W}}(t + T) \)

1. \( k = 1 \)
2. while \( k \leq n_e(n_e-1) \)
3. \( (l_k, m_k) = \text{NextElement}(k) \)
4. Define \( w_1(\phi) = h(\hat{\mathbf{W}}(t) \cdot \mathbf{r}_{l_k,m_k}(\hat{\theta}, \phi)) \)
5. Perform \( \phi = \text{LineSearch}(w_1(\phi), \pi) \)
6. Define \( w_2(\theta) = h(\hat{\mathbf{W}}(t) \cdot \mathbf{r}_{l_k,m_k}(\theta, \phi)) \)
7. Perform \( \theta = \text{LineSearch}(w_2(\theta), \theta_{max}) \)
8. Set \( \hat{\mathbf{W}}(t) = \hat{\mathbf{W}}(t) \cdot \mathbf{R}_{l_k,m_k}(\theta, \phi) \)
9. \( k = k + 1 \)
10. \( \hat{\mathbf{W}}(t + T) = \hat{\mathbf{W}}(t) \)

![Fig. 2: Channel Tracking with \( P_{Tr} = -20 \). (2, 1)-System. \( F_d = 1.3 \) Hz, (velocity 2 Km/h, \( F_c = 700MHz \)).](image)

![Fig. 3: (2, 1)-System. Simulation of the system over \( 10^6 \) slots for different doppler frequencies \( F_d \).](image)
VI. Transmitting and Tracking Simultaneously

As we have already observed, even in slow changing channels, the null space seems to change in a pace that allows only a “bursty” transmission behavior of the secondary system. That is, even if SU-Tx manages to track the channel, it will be able to transmit to SU-Rx for a small number of time-slots and then it will need to adapt again. Therefore, the secondary system uses a significant amount of its transmission time only to adapt to the changing environment. Motivated by this observation, we first make the assumption that the learning is performed in a way that always satisfies the interference constraints imposed by the primary network and then we are searching for a way to transmit simultaneously without introducing any error in the learning process.

More specifically, consider only one time period of $T_{FB}$ seconds in which the SU-Tx sends the same signal until it senses the change of the transmit power of the PU-Tx. Let $N = \frac{T_{FB}}{T_c}$ the number of consecutive slots that the SU-Tx transmits the same learning vector.

Consider the case that the SU-Tx is supposed to send the learning signal $x(t) = r_1$, where $t$ denotes the index of the time-slot. Denote $y_2(t)$ the information signal that SU-Tx wants to send to SU-Rx. Denote as $\mathbf{H}_{12}(t)$ the channel between the SU-Tx and SU-Rx. Assume that $\mathbf{H}_{12}(t)$ has approximately the same coherence time with $\mathbf{H}_{22}(t)$. Let $T_c$ the coherence time of these channels. During the $T_{FB}$ period the channels $\mathbf{H}_{12}(t)$ and $\mathbf{H}_{22}(t)$ are approximately constant, i.e.

$$\mathbf{H}_{12}(t) \approx \mathbf{H}_{12} \text{ and } \mathbf{H}_{22}(t) \approx \mathbf{H}_{22}$$

because $T_{FB} << T_c$, otherwise the tracking would be impossible as we have already discussed in section IV.

We need to find a way to superimpose $y_2(t)$ so that the BNSL algorithm remains unaffected and the SU-Rx receives information. We start from the signal that the PU-Rx and the SU-Rx receive respectively:

$$y_1(t) = \mathbf{H}_{12} \mathbf{T} (r_1 + r_2(t)) + n_1(t)$$
$$y_2(t) = \mathbf{H}_{22} \mathbf{T} (r_1 + r_2(t)) + n_2(t)$$

Denote also as $y_1^0(t)$ and $y_2^0(t)$ the received signal at PU-Rx and SU-Rx when no information is superimposed to the learning algorithm, i.e.

$$y_1^0(t) = \mathbf{H}_{12} \mathbf{T} r_1 + n_1(t), \quad y_2^0(t) = \mathbf{H}_{22} \mathbf{T} r_1 + n_2(t)$$

Define as

$$\Delta y_1 = f(y_1) - f(y_1^0)$$

the difference in the measurement at the PU-Rx as a result of the introduction of the $r_2(t)$ signal, where $f(\cdot)$ is the measurement function $f : \mathbb{R}^N \to \mathbb{R}$.

A. First Model of Energy Measurement

Assume that the primary system uses an energy measurement such that it measures the quantity:

$$Q_1(y_1) = \frac{1}{N} \sum_{t=1}^{N} ||y_1(t)||^2$$

We make no assumption that the SU-Rx knows the current precoding matrix of SU-Tx nor the channel $\mathbf{H}_{22}$ between them.

From equations (4),(6),(7) we get that

$$\Delta y_1 = Q_1(y_1) - Q_1(y_1^0)$$
$$= \frac{1}{N} \sum_{t=1}^{N} \left( ||\mathbf{H}_{12} \mathbf{T} (r_1 + r_2(t)) + n_1(t)||^2 
- ||\mathbf{H}_{12} \mathbf{T} r_1 + n_1(t)||^2 \right)$$

In order the $r_2(t)$ not to affect the Jacobi method it is sufficient that $\Delta y_1 = 0$. Assume that

$$r_2(t) = c(t)r_1$$

where $c(t) \in \mathbb{C}$ are drawn according to a known discrete probability distribution on the support $\{c_1, c_2, \ldots, c_M\}$, and $M$ is the number of different messages that can be sent. A reasonable assumption would be $Pr[c(t) = c_i] = \frac{1}{M}$, $\forall i \in \{1, 2, \ldots, M\}$, but we do not restrict this scheme only to this case. Consider the following constraints that the $\{c_i\}$ should satisfy:

- $\mathbb{E}[|1 + c(t)|^2] = 1$. This is sufficient in order to ensure that the $\Delta y_1$ is close to 0. Note that this can always be ensured if $|1 + c_i|^2 = 1, \forall i \in \{1, 2, \ldots, M\}$, but still this is not the only way.
- $\mathbb{E}\{c(t)\} \neq -1$. The necessity of this assumption will become evident later on.

We get that

$$\Delta y_1 = \frac{1}{N} \sum_{t=1}^{N} \left( ||\mathbf{H}_{12} \mathbf{T} (r_1 + r_2(t)) + n_1(t)||^2 
- ||\mathbf{H}_{12} \mathbf{T} r_1 + n_1(t)||^2 \right)$$

$$(a) = \mathbb{E}[||\mathbf{H}_{12} \mathbf{T} (r_1 + c(t)r_1)||^2] - ||\mathbf{H}_{12} \mathbf{T} r_1 + n_1(t)||^2 + c_1(N)$$

$$(b) = ||\mathbf{H}_{12} \mathbf{T} r_1||^2 \mathbb{E}[|1 + c(t)|^2] - ||\mathbf{H}_{12} \mathbf{T} r_1||^2 + c_1(N)$$

$$= c_1(N)$$

where

- $(a)$ follows from the law of large numbers (LLN).
- $(b)$ follows from the fact that $\mathbb{E}[|1 + c(t)|^2] = 1$.
- $c_1(N) \to 0$ a.s as $N \to \infty$.

Thus, $\Delta y_1 \approx 0$ for sufficiently large $N$.

Now, we need the SU-Rx to be able to decode the received message. In the end of the $T_{FB}$ seconds, the SU-Rx can estimate the average of the received signal:

$$\bar{y}_2 = \frac{1}{N} \sum_{t=1}^{N} y_2(t)$$

$$= \frac{1}{N} \sum_{t=1}^{N} (\mathbf{H}_{22} \mathbf{T} (r_1 + r_2(t)) + n_2(t))$$

$$(a) = \mathbf{H}_{22} \mathbf{T} r_1 + \mathbf{H}_{22} \mathbf{T} (\mathbb{E}\{c(t)\}) + c_2(N)$$

$$= \mathbf{C} \mathbf{H}_{22} \mathbf{T} r_1 + c_2(N)$$

(13)
where

\( C = 1 + \mathbb{E}\{c(t)\} \) is a predefined parameter known to both the SU-Tx and SU-Rx.

\( \epsilon_2(N) \rightarrow 0 \) w.p.1 as \( N \rightarrow \infty \).

Therefore, in the end of the \( T_{FB} \) seconds the SU-Rx can learn the quantity:

\[
H_{22} T_1 = \frac{1}{C}(\bar{y}_2 - \epsilon_2(N))
\]

Then, the SU-Rx can subtract the \( H_{22} T_1 \) from \( y_2(t) \) and we get that:

\[
\Delta y_2(t) = y_2(t) - H_{22} T_1 = H_{22} T_2(t) + n_2(t) + \frac{\epsilon_2(N)}{C}
\]

\[
= c(t) H_{22} T_1 + n_2(t) + \frac{\epsilon_2(N)}{C}
\]

Note that the SU-Rx is aware of \( C \) and \( H_{22} T_1 \) and therefore it can decide among which \( c_i \) was sent in its timeslot. More specifically consider the set of assumptions \( \{H_i\} \) such that \( H_i = c_i H_{22} T_1 \). Then it follows that

\[
y_2(t) - H_i = \begin{cases} (c(t) - c_i) H_{22} T_1 + n_2(t) + \frac{\epsilon_2(N)}{C} & \text{if } c(t) = c_i \\ (c(t) - c_i) H_{22} T_1 + n_2(t) + \frac{\epsilon_2(N)}{C} & \text{other} \end{cases}
\]

The only serious assumption that we have made is that \( N \) is large enough in order to assume that the LLN holds. It is possible this not to be true but we can enforce the LLN to hold by introducing coding in the transmitted symbols. This will be described in section VII.

B. Example for a binary alphabet \((M = 2)\)

Assume that the SU-Tx is using a 2 constellation signal on top of \( r_1 \):

\[
r_2(t) = \begin{cases} c_1 r_1, & \text{SU-Tx transmits 1 with prob. 0.5} \\ c_2 r_1, & \text{SU-Tx transmits 2 with prob. 0.5} \end{cases}
\]

where \( c_1 \) and \( c_2 \) are complex numbers. The constraints that the \( c_1 \) and \( c_2 \) should satisfy are:

1. \( |1 + c_1|^2 + |1 + c_2|^2 = 2 \).
2. \( c_1 + c_2 \neq -2 \).

If we restrict to the case that \( c_1 = e^{j\theta_0} \) and \( c_2 = e^{-j\theta_0} \) then both constraints are satisfied for \( \theta_0 = \frac{2\pi}{3} \). Notice that in this specific case we always get:

\[
||H_{12} T_1(1 + e^{j\theta_0})||^2 = ||H_{12} T_1(1 + e^{-j\theta_0})||^2 = ||H_{12} T_1||^2
\]

Then, \( C = 1 + \cos(\theta_0) \) and

\[
\Delta y_2(t) = y_2(t) - \frac{y_2(t)}{1 + \cos(\theta_0)}
\]

\[
= \begin{cases} \frac{e^{j\theta_0}}{1 + \cos(\theta_0)} \bar{y}_2(t) + n_2(t), & \text{if 1 was transmitted} \\ \frac{e^{-j\theta_0}}{1 + \cos(\theta_0)} \bar{y}_2(t) + n_2(t), & \text{if 2 was transmitted} \end{cases}
\]

The two assumptions \( H_1 \) and \( H_2 \) of the decoder are:

\[
H_1 = \frac{e^{j\theta_0}}{1 + \cos(\theta_0)} \bar{y}_2(t), \quad H_2 = \frac{e^{-j\theta_0}}{1 + \cos(\theta_0)} \bar{y}_2(t)
\]

The decision rule is

\[
data(t) = \begin{cases} 1, & \text{if } ||\Delta y_2(t) - H_1|| < ||\Delta y_2(t) - H_2|| \\ 2, & \text{other} \end{cases}
\]

C. Second Model of Energy Measurement

It is possible that the PU-Rx uses the more reliable energy measurements of the form:

\[
Q_2(y_1) = ||\frac{1}{N} \sum_{i=1}^{N} y_1(t)||^2
\]

In this case, the proposed scheme can be modified appropriately. Again, we do not assume that the SU-Rx knows the current precoding matrix \( T \) nor the channel \( H_{22} \). Therefore:

\[
\Delta y_1 = Q_2(y_1) - Q_2(y_0) = ||\frac{1}{N} \sum_{i=1}^{N} (H_{12} T (r_1 + r_2(t)) + n_1(t))||^2
\]

\[
- ||\frac{1}{N} \sum_{i=1}^{N} (H_{12} T_1 + n_1(t))||^2
\]

Assume again as before that \( r_2(t) = c(t) r_1 \) and consider the constraints on \( c_i \):

1. \( |1 + \mathbb{E}\{c(t)\}| = 1 \)
2. Note that the constraint \( \mathbb{E}\{c(t)\} \neq -1 \) is immediately satisfied.

Starting from (19) and using the LLN as before:

\[
\Delta y_1 = ||H_{12} T_1(1 + \mathbb{E}\{c(t)\})||^2 - ||H_{12} T_1||^2 + \epsilon_3(N) = \epsilon_3(N)
\]

where \( \epsilon_3(N) \rightarrow 0 \) as \( N \rightarrow \infty \).

The SU-Rx in this case works exactly the same as before so we do not repeat the procedure. Note, that the only difference is that the set \( \{c_i\} \) need to satisfy different constraints that might lead to different SNR at the receiver.

VII. Enumerative Coding

In the above scheme we made the assumption that \( N \) is large enough in order the LLN to hold. In order to be precise, what we actually need is

\[
\frac{1}{N} \sum_{i=1}^{N} c(t) = S
\]

for all possible transmitted sequences \( \{c(1), c(2), \ldots, c(N)\} \), where \( S \) is a predefined constant known to both the SU-Tx and SU-Rx. Assume the secondary system fixes \( S \), then it needs a coding strategy that maps a \( N \)-tuple of symbols to another \( N \)-tuple that satisfies the constraint.

A simple and efficient method that can ensure this constraint has been treated by Cover ([8]). Specifically, without loss of generality, assume that the SU-Tx wants to send symbols from the alphabet \( \mathcal{X} = \{1, 2, \ldots, M\} \) according to the uniform distribution. This assumption models well the distribution of the actual data that are sent in a wireless network. We will need some definitions from the information theoretic literature.
Define the type $P_x$ of a sequence $x = x_1^n$ as the relative proportion of occurrences of each symbol. Also, define as the type class of the probability distribution $P$, denoted as $T(P)$ the set of sequences of length $N$ that have type $P_x$, i.e. $T(P) = \{x \in \mathcal{X}^N : P_x = P\}$.

For simplicity we assume that $N/M$ is an integer. Equation (20) is satisfied if the transmitted sequence $c(1), c(2), \ldots, c(N)$ has $P_x = (\frac{1}{M}, \frac{1}{M}, \ldots, \frac{1}{M})$, or in other words, if the secondary system uses only those sequences that belong in the type class $T(P_c)$. We know that

$$|T(P_c)| = \binom{N}{N/M, N/M, \ldots, N/M}$$

Therefore, the SU-Tx can send only $|T(P_c)|$ different $N$-sequences from the $M^N$ available sequences. For example, for $M = 2$ and $N = 16$ we get that $|\log_2(|T(P_c)|)| = 13$. For $M=2$ and $N=32$, $|\log_2(|T(P_c)|)| = 29$ and for $M=4$ and $N=16$, $|\log_2(|T(P_c)|)| = 25$.

The encoding process is the following: Assume that the SU-Tx wants to transmit the sequence $x_1^n$. Then, it treats the sequence $x_1^n$ as the index of an $N$-sequence that belongs to the type class $T(P_c)$. Thus, it maps the index to a sequence and transmits the sequence. Then, the SU-Rx performs the reverse procedure. Both encoding and decoding thoroughly explained in [8] and can be easily performed with low complexity.

**VIII. Simulation Results**

In this section we test the effectiveness of the proposed scheme in its most generality. Specifically, we generate $10^3$ random matrices $H_{12}$, $H_{22}$ and a random unitary matrix $T$. Note that since $T$ is a random matrix will actually be independent of both $H_{12}$ and $H_{22}$ which can only degrade the performance of the transmission strategy. Then, for each simulation scenario we generate $10^3$ bits that are superimposed on a vector $r_1$ that was created by a random Jacobi rotation. We use $N = 16$, $M = 2$ (assuming $T_{FB} = 1$ msec and $T_s = 66.7\mu$sec) and $\theta_0 = \frac{2\pi}{3}$. $N = 16$ corresponds to $T_s = 66.7$ and $T_{FB} = 1$ msec. $P_s$ is the average probability of symbol error over all scenarios. In Figure 4 we plot the $P_s$ as a function of the SNR at the SU-Rx using the scheme presented in section VI-B.

The average value of $\Delta y_1$ is $0.13$ dB, which is an insignificant increase of the interference. From Figure 4 we observe that in the case of an $(3,1)$-system and $(2,1)$-system, for $SNR > 3.5$ dB or $SNR > 4.5$ dB respectively, the probability of error is less than $10^{-3}$. We observe that more antennas in the SU-Tx leads to a more robust system. Similar results are shown also for $M = 4$.

**IX. Future Work**

No matter which is the algorithm for null-space acquisition we observed through simulations that the null space under independent Rayleigh fading changes significantly faster than the coherence time. More research is needed in order to understand how fast the null space varies. Also, in this paper, we use the vector $r_1$, i.e. the current learning signal, as the carrier of the information we need to send. This gives us only one spatial degree of freedom. Is it possible to propose a similar algorithm with which we are transmitting using two or more vectors and still we do not affect the learning process?

**X. Conclusions**

The starting point of this work was the observation that the null space of a time-varying channel changes significantly faster than $T_c$. This motivated the need of an enhanced BNSL algorithm which learns the channel and transmits information simultaneously. We proposed a scheme that perform this task by superimposing the information on the learning signal. We demonstrated through simulations the validity of the proposed transmission scheme.

**REFERENCES**


