Progressive Inter-Path Interference Cancellation Algorithm for Channel Estimation Using Orthogonal Time–Frequency Space

(article starts on next page)

Article

Progressive Inter-Path Interference Cancellation Algorithm for Channel Estimation Using Orthogonal Time–Frequency Space

Mauro Marchese [1], Henk Wymeersch [2], Paolo Spallaccini [3] and Pietro Savazzi [4],*

1 Department of Electrical, Computer and Biomedical Engineering, University of Pavia, 27100 Pavia, Italy; mauro.marchese01@universitadipavia.it
2 Department of Electrical Engineering, Chalmers University of Technology, 412 96 Gothenburg, Sweden; henkw@chalmers.se
3 HCL Software, Vimodrone, 20055 Milan, Italy; paolo.spallaccini@hcl.software
4 Department of Electrical, Computer and Biomedical Engineering, University of Pavia and CNIT Consorzio Nazionale Interuniversitario per le Telecomunicazioni—Unità di Pavia, 27100 Pavia, Italy
* Correspondence: pietro.savazzi@unipv.it

Abstract: Fractional delay-Doppler (DD) channel estimation in orthogonal time–frequency space (OTFS) systems poses a significant challenge considering the severe effects of inter-path interference (IPI). To this end, several algorithms have been extensively explored in the literature for accurate low-complexity channel estimation in both integer and fractional DD scenarios. In this work, we develop a variant of the state-of-the-art delay-Doppler inter-path interference cancellation (DDIPIC) algorithm that progressively cancels the IPI as estimates are obtained. The key advantage of the proposed approach is that it requires only a final refinement procedure reducing the complexity of the algorithm. Specifically, the time difference in latency between the proposed approach and the DDIPIC algorithm is almost proportional to the square of the number of estimated paths. Numerical results show that the proposed algorithm outperforms the other channel estimation schemes achieving lower normalized mean square error (NMSE) and bit error rate (BER).

Keywords: OTFS; channel estimation; (DD) delay-Doppler domain; fractional parameter estimation; inter-path interference

1. Introduction

Reliable communication in doubly dispersive channels, where high delay and Doppler spreads due to multipath and mobility are experienced, is becoming an increasingly prominent research topic toward the development of next-generation wireless networks [1,2]. The widely adopted OFDM (orthogonal frequency division multiplexing) is suitable for static frequency-selective multipath channels but experiences drastic performance degradation in high-mobility scenarios due to the loss of orthogonality between the subcarriers, as analyzed in [3], inducing intercarrier interference (ICI). To cater to high-mobility scenarios, several novel waveforms have been introduced in literature. An overview of some of the existing modulation schemes for high-mobility wireless channels is reported in [4]. Orthogonal time–frequency space (OTFS) modulation was proposed in [5,6] and is garnering a lot of attention due to its robustness against the Doppler effect. OTFS is a two-dimensional modulation scheme that multiplexes information symbols in the delay-Doppler domain, where the channel representation becomes sparse and the channel can be considered almost time-invariant [5–7]. In contrast, OFDM multiplexes information symbols in the time–frequency (TF) domain, where mobility generates a frequency response of the time-varying channel.

1.1. Problem Context

In this work, we consider a scenario comprising a single-antenna transmitter and receiver, focusing on the channel estimation problem with fractional delays and Doppler

shifts and no prior information about the number of propagation paths [8,9]. The transmitter sends a pilot symbol in the delay-Doppler (DD) domain, while the receiver processes the received pilot frame to extrapolate the knowledge of the channel parameters.

The channel estimation problem is often considered in the *integer DD scenario,* which is facilitated by the assumption of having a channel with delays and Doppler shifts that are multiples of delay and Doppler resolutions. In this case, there is no interference between adjacent DD bins, and a simple threshold method can be used to obtain good performances, as shown in [10]. On the other hand, in reality, this assumption may not hold, which motivates the study of *fractional DD scenario,* which is more challenging, since the received replica of the pilot symbol in the DD domain spreads into adjacent bins after 2D sampling of the DD domain [7,11]. This is the well-known inter-path interference phenomenon. Our focus is on the effects of IPI caused by fractional DDs and IPI cancellation to recover channel state information (CSI) with high accuracy.

1.2. State of the Art

Channel estimation in OTFS systems has been widely investigated in the literature since the detection performance of the OTFS receiver is based on the estimated channel matrix [12–15]. In [10], an embedded-pilot-aided method for channel estimation is proposed, achieving good performances in both integer DDs and fractional Doppler scenarios. Channel estimation in massive multiple-input multiple-output scenarios (MIMO) has been investigated in [16], where additional sparsity due to a large number of transmit and receive antennas allows the channel estimation problem to become a simpler sparse recovery problem taking advantage of the angular domain. In [17–19] a novel tensor-based approach for channel estimation in massive MIMO systems is proposed to further exploit channel sparsity in the delay-Doppler-angle domain and reduce signal processing burden. Low-complexity channel estimation schemes were considered in single-antenna systems in [8], where modified maximum likelihood estimator (M-MLE) and two-step estimator (TSE) algorithms were introduced. The first is based on the asymptotic orthogonality of a delay-Doppler parameter matrix for large OTFS frames, which allows for the simplification of the cost function. On the other hand, the second one has lower complexity and relies on disjoint delay and Doppler estimation. Under the assumption of having good separability of the received pilot replica due to fine delay and Doppler resolutions, in both algorithms, the IPI is cancelled, and at each iteration, a residual received vector is used for estimating channel parameters of the next path. Similarly, IPI cancellation is performed in the low-complexity channel estimation scheme for discrete Zak transform-based OTFS (DZT-OTFS) proposed in [20]. Sparse Bayesian learning (SBL) algorithms for channel estimation are also considered in [21,22], but M-MLE and TSE algorithms are shown to perform better then SBL in [8] when delay and Doppler resolutions are fine enough. In [23], different channel estimation schemes for the fractional DD case are proposed with the scope of improving the spectrum efficiency and reduce the peak-to-average-power-ratio (PAPR) by sending pilots in the time–frequency (TF) domain. The problem of high-precision channel estimation in IPI scenarios has been addressed in [11], where an iterative delay-Doppler IPI cancellation (DDIPIC) algorithm was proposed. In this case, the IPI cancellation is performed indirectly through refinement procedures carried out at each iteration. These refinements of the channel parameters take time, increasing the computational burdening of the channel estimation algorithm. In more detail, the computational complexity of the cost function, which is evaluated multiple times during each iteration, increases with the cube of the iteration index due to matrix inversion. However, in [11], the DDIPIC algorithm is shown to perform better than the M-MLE algorithm. Hence, in this work, we considered only the comparison between the proposed approach and the state-of-the-art DDIPIC algorithm.

1.3. Contributions

The goal of this work is to propose a variant of the DDIPIC algorithm from [11], named the progressive IPI cancellation (P-IPIC) algorithm, which cancels the IPI by performing

residue cost function (RCF) maximization at each step. The main difference of the proposed algorithm with respect to the DDIPIC is that it operates in two phases: A *search phase*, where a certain number of possible paths are detected and a first rough estimate of the channel parameters is provided, A *refinement phase*, where the found paths are refined, looking for ones declared as false alarms and discarded, until the stopping criterion is met. This procedure allows the correct number of paths to be found while the accuracy of the estimates is improved. The contributions of this work can be summarized as follows:

- *Progressive IPI cancellation*: Instead of discovering possible paths and performing refinements in each iteration maximizing the observation-based cost function as in [11], we perform maximization of a residue-based cost function (RCF) at each step during the discovery phase. This progressive cancellation of the IPI allows a first channel estimate to be provided, aiming to reduce the residue energy as iteration progress, achieving high accuracy in channel matrix reconstruction.

- *Latency reduction due to single global refinement of the estimates*: The main drawback of the state-of-the-art DDIPIC algorithm is that it performs refinement procedures at each iteration, as discussed in Section 3.5. Progressive IPI cancellation allows the latency to be reduced, since, as discussed above, the algorithm can operate in two distinct phases and a single global refinement is needed.

- *Computational complexity reduction in search phase*: During the iterative search phase, the proposed algorithm looks for new paths by performing a RCF maximization. This allows us to reduce the computational cost since the computation of the RCF does not require matrix inversion. Moreover, the computational cost does not depend on the iteration index. As a consequence, coarse and fine estimation procedures described in Section 3.4 will take less time.

- *Resilience towards high-IPI scenarios*: In the literature, it is commonly assumed to have sufficiently fine delay and Doppler resolutions such that all the propagation paths are resolvable in the DD domain. In some cases, this may not hold, and robust low-complexity channel estimation algorithms must be designed to suit high-IPI scenarios. In this work, we consider both low- and high-IPI regimes, showing that the proposed approach is suitable for both regimes.

The remainder of this paper is organized as follows. In Section 2, the OTFS signal model is presented. In Section 3, OTFS channel estimation is covered, including the baseline DDIPIC method and our proposed P-IPIC method.

2. OTFS Modulation

In this section, the employed models for the OTFS signal, the propagation channel, and receiver-side processing are presented.

2.1. Transmitted Signal Model

In OTFS systems [5,7], the TX arranges MN symbols in the delay-Doppler grid, referred to as I_{DD}, where M is the number of subcarriers and N is the number of time slots of the OTFS frame. The DD-domain symbols are then converted to the time–frequency domain through inverse symplectic finite Fourier transform (ISFFT). Denoting with $\mathbf{X} \in \mathbb{C}^{M \times N}$ the delay-Doppler symbols multiplexed in the DD domain, the corresponding symbols in the time–frequency grid are obtained as

$$X_{tf}[l,k] = \frac{1}{\sqrt{MN}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} X[m,n] e^{j2\pi\left(\frac{nk}{N} - \frac{ml}{M}\right)}. \tag{1}$$

After that, the time-domain OTFS signal is obtained through the the Heisenberg transform as

$$s(t) = \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} X_{tf}[l,k] g_{tx}(t - kT) e^{j2\pi l \Delta f(t - kT)}, \tag{2}$$

where $g_{tx}(t)$ is the transmit-shaping pulse of duration T. The continuous-time signal in Equation (2) has a total duration of NT and a bandwidth of $M\Delta f$, where T corresponds to the time slot duration and $\Delta f = 1/T$ is the subcarrier spacing. Thus, the delay and Doppler resolutions are, respectively,

$$\tau_{\text{res}} = \frac{1}{M\Delta f}, \; \nu_{\text{res}} = \frac{1}{NT}. \tag{3}$$

2.2. Channel Model

The transmitted signal passes through the channel that in the DD domain is modeled as

$$h(\tau, \nu) = \sum_{i=1}^{P} g_i \delta(\tau - \tau_i) \delta(\nu - \nu_i). \tag{4}$$

where P is the number of propagation paths and g_i, τ_i, and ν_i are the complex channel gain, the propagation delay, and the Doppler shift of the i-th path, respectively.

2.3. Receiver Processing

The received signal is obtained by computing the Heisenberg transform and adding additive white Gaussian noise (AWGN) with monolateral power spectral density (PSD) N_0, referred to as $n(t)$. Thus, the received signal is

$$r(t) = \iint h(\tau, \nu) s(t - \tau) e^{j2\pi\nu(t-\tau)} d\tau d\nu + n(t), \tag{5}$$

At the receiver side, inverse operations are performed to recover the received symbols in the DD domain. Firstly, the receiver computes the Wigner transform by sampling the so-called cross-ambiguity function (CAF) $A_{g_{rx},r}(f,t)$ to convert the received time-domain signal into samples in the time–frequency grid as

$$Y_{tf}[l,k] = A_{g_{rx},r}(f,t)|_{f=l\Delta f, t=kT} = \int r(t') g_{rx}^*(t' - t) e^{-j2\pi f(t'-t)} dt' \Big|_{f=l\Delta f, t=kT}. \tag{6}$$

Secondly, a SFFT is applied to obtain the DD-domain received frame

$$Y[m,n] = \frac{1}{\sqrt{MN}} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} Y_{tf}[l,k] e^{-j2\pi\left(\frac{nk}{N} - \frac{ml}{M}\right)}. \tag{7}$$

From now on, we are going to consider OTFS modulation with rectangular shaping filters of duration T and amplitude $1/\sqrt{T}$ at both the transmitter and receiver side.

3. OTFS Channel Estimation

The purpose of channel estimation is to determine the parameters of (4), based on pilot signals. In this section, the pilot model and corresponding observation model are presented, followed by the baseline method and the proposed method.

3.1. Pilot Model

In our scenario [8,11], the single-antenna transmitter sends a pilot symbol in the DD domain. The corresponding pilot frame is given by

$$X[m,n] = \begin{cases} \sqrt{E_p} & m = m_p, \; n = n_p, \\ 0 & \text{otherwise}, \end{cases} \tag{8}$$

where $m = m_p, n = n_p$ is the delay-Doppler resource element (DDRE), in which the pilot is sent and E_p is the energy of the transmitted pilot signal.

At the receiver, after the operations described above and vectorization, the received pilot frame is given by

$$\mathbf{y} = \text{vec}(\mathbf{Y}) = \mathbf{H}\text{vec}(\mathbf{X}) + \mathbf{w} = \mathbf{H}\mathbf{x} + \mathbf{w}, \tag{9}$$

where $\mathbf{w} \sim \mathcal{CN}(0, N_0\mathbf{I}) \in \mathbb{C}^{MN \times 1}$ is the AWGN noise vector and $\mathbf{H} \in \mathbb{C}^{MN \times MN}$ is the delay-Doppler domain channel matrix, given by

$$\mathbf{H} = \sum_{i=1}^{P} g_i \mathbf{G}_i(\tau_i, \nu_i), \tag{10}$$

where $\mathbf{G}_i(\tau_i, \nu_i) \in \mathbb{C}^{MN \times MN}$ is a matrix that captures the effect of the propagation delay and the Doppler shift of the *i*-th path, and for $l', l'' = 0, 1, .., M - 1, k', k'' = 0, 1, ..., N - 1$ is computed as [8,11]

$$G_i[k'M + l', k''M + l''] = \frac{e^{-j2\pi\tau_i\nu_i}}{MN} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} r_{k'',l'}(m) e^{j2\pi \left(\frac{m}{M} \left(l' - l'' - \frac{M\tau_i}{T} \right) - \frac{n}{N} \left(k' - k'' - \frac{N\nu_i}{\Delta f} \right) \right)}, \tag{11}$$

where

$$r_{k'',l'}(m) = \sum_{s=-m}^{M-1-m} e^{j2\pi s \frac{l'}{M}} \left[\left(1 - \frac{\tau_i}{T} \right) e^{j\pi \left(1 + \frac{\tau_i}{T} \right) \left(\frac{\nu_i}{\Delta f} - s \right)} \text{sinc}\left(\left(1 - \frac{\tau_i}{T} \right) \left(\frac{\nu_i}{\Delta f} - s \right) \right) + \right.$$
$$\left. \frac{\tau_i}{T} e^{-j2\pi \frac{k''}{N}} e^{j\pi \left(\frac{\tau_i}{T} \right) \left(\frac{\nu_i}{\Delta f} - s \right)} \text{sinc}\left(\tau_i (\nu_i - s\Delta f) \right) \right]. \tag{12}$$

If the channel coefficients are normalized such that $\sum_{i=1}^{P} |g_i|^2 = 1$, the received pilot signal power equals the transmitted one. Under this hypothesis, the pilot signal-to-noise ratio (PSNR) can be computed as in [8]. Hence, the pilot signal power is the total energy E_p divided by the overall time duration NT, while the noise power is the product of the power spectral density N_0 and the bandwidth $M\Delta f$. Thus,

$$\text{PSNR} = \frac{E_p/NT}{M\Delta f N_0} = \frac{E_p}{MNN_0}. \tag{13}$$

During a channel coherence window, the receiver processes the received pilot frame running a channel estimation algorithm, and once the estimated channel matrix $\hat{\mathbf{H}}$ is obtained, uses the estimate to detect the incoming data frames.

3.2. Observation Model for Channel Estimation

As in [11], channel estimation is performed considering the input–output relation in Equation (9), rewritten as

$$\mathbf{y} = \mathbf{R}(\boldsymbol{\tau}, \boldsymbol{\nu})\mathbf{g} + \mathbf{w}, \tag{14}$$

where $\mathbf{g} = [g_1 \, g_2 \, ... \, g_P]^T \in \mathbb{C}^{P \times 1}$ is the channel gains vector and $\mathbf{R}(\boldsymbol{\tau}, \boldsymbol{\nu}) = [\mathbf{r}_1 \, \mathbf{r}_2 \, ... \, \mathbf{r}_P] \in \mathbb{C}^{MN \times P}$ is a matrix called constituent delay-Doppler parameter matrix (CDDPM). Each column of $\mathbf{R}(\boldsymbol{\tau}, \boldsymbol{\nu})$ is related to a different path and is given by

$$\mathbf{r}_i(\tau_i, \nu_i) = \mathbf{G}_i(\tau_i, \nu_i)\mathbf{x}. \tag{15}$$

Given the input–output relation in Equation (14), the maximum likelihood (ML) estimate of the unknowns is given by

$$(\hat{\mathbf{g}}, \hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\nu}}) = \arg\min_{\mathbf{g}, \boldsymbol{\tau}, \boldsymbol{\nu}} \|\mathbf{y} - \mathbf{R}(\boldsymbol{\tau}, \boldsymbol{\nu})\mathbf{g}\|^2. \tag{16}$$

Although the minimization problem depends on three parameters, it can be simplified performing a two-step ML estimate of the unknowns as

$$(\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{v}}) = \arg\max_{\boldsymbol{\tau}, \boldsymbol{v}} \Phi(\mathbf{R}(\boldsymbol{\tau}, \boldsymbol{v}), \mathbf{y}), \tag{17}$$

and

$$\hat{\mathbf{g}} = \left(\mathbf{R}^H(\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{v}})\mathbf{R}(\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{v}})\right)^{-1}\mathbf{R}^H(\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{v}})\mathbf{y}. \tag{18}$$

Here, the cost function $\Phi(\mathbf{R}(\boldsymbol{\tau}, \boldsymbol{v}), \mathbf{y})$ is computed as

$$\Phi(\mathbf{R}(\boldsymbol{\tau}, \boldsymbol{v}), \mathbf{y}) = \mathbf{y}^H\mathbf{R}(\boldsymbol{\tau}, \boldsymbol{v})\left(\mathbf{R}^H(\boldsymbol{\tau}, \boldsymbol{v})\mathbf{R}(\boldsymbol{\tau}, \boldsymbol{v})\right)^{-1}\mathbf{R}^H(\boldsymbol{\tau}, \boldsymbol{v})\mathbf{y}. \tag{19}$$

The DDIPIC algorithm proposed in [11] performs an iterative maximization of the cost function in Equation (17) in order to jointly estimate delays and Dopplers of the paths. The operation of the algorithm is summarized below.

3.3. DDIPIC Algorithm

The algorithm proceeds iteratively for a maximum of $P_{\max}$ iterations and the CDDPM matrix is initialized as $\mathbf{R}(\boldsymbol{\tau}, \boldsymbol{v}) = \mathbf{0}$. The details of the DDIPIC algorithm are provided in Algorithm 1. At each iteration, the parameters of one path are estimated as follows:

1. *Coarse estimation (line 2)*: The cost function in (19) is maximized over the DD grid of integer multiples of delay and Doppler resolutions, providing a coarse estimate of delay and Doppler shift. The search area is limited to $[0, L\tau_{\text{res}}] \times [-Kv_{\text{res}}, Kv_{\text{res}}]$, where $L = \lceil \frac{\tau_{\max}}{\tau_{\text{res}}} \rceil$ and $K = \lceil \frac{v_{\max}}{v_{\text{res}}} \rceil$.

2. *Fine estimation (lines 5–11)*: An iterative procedure runs for a maximum of s_{max} iterations to provide a better estimate of the off-grid DDs. In each iteration, the cost function in (19) is maximized over a search area comprising fractional delays and Dopplers around, for $s = 1$, the coarse estimate obtained in step 1 or, for $s > 1$, the fine estimate obtained in the previous iteration. The search space is narrowed down as the iteration progresses, reducing the delay and Doppler search widths (lines 6–7). The fine estimation procedure ends when the stopping criteria for τ and v are met. The search space for the fine estimation procedure is called $F_{DD}^{(s)}$, and we define $\Gamma = \{-\lfloor \frac{m_\tau}{2} \rfloor, ..., 0, ..., \lfloor \frac{m_\tau}{2} \rfloor\}$ and $\Lambda = \{-\lfloor \frac{n_v}{2} \rfloor, ..., 0, ..., \lfloor \frac{n_v}{2} \rfloor\}$ where m_τ and n_v are design parameters. Notice that after the Cartesian product (denoted $\times$) of the two sets, only positive delays need to be considered. The parameters ϵ_τ and ϵ_v are chosen to achieve the desired accuracy compatibly with $s_{\max}$.

3. *Check stopping criterion (line 15–19)*: A residue vector is computed as $\mathcal{E}^{(i)} = \mathbf{y} - \mathbf{R}(\hat{\tau}, \hat{v})\hat{\mathbf{g}}$ and if $\|\mathcal{E}^{(i)}\| < \epsilon$ is satisfied, the algorithm stops and a number of $\hat{P}$ paths are detected; otherwise, before starting the search for a new path, all previous estimates are refined as described below. The convergence tolerance parameter ϵ for the stopping criterion must be chosen not to underestimate the number of propagation paths, not to miss those with smaller amplitudes, and not to overestimate the number of propagation paths due to residual IPI or noise. Since the noise is Gaussian, a reasonable value could be set taking into account the fact that the number of symbols per frame is MN, and the noise variance is N_0, so that we have set its value to $\epsilon = 3\sqrt{MNN_0}$.

4. *Refinement procedure (line 20–21)*: All coarse and fine estimates are performed again for all the previously detected paths until the current one. In particular, at the i-th iteration, lines 2–13 are run again iteratively for $i_{\text{ref}} = 1, ..., i$.

Algorithm 1: DDIPIC

　　Input: $P_{\max}$, ϵ, $s_{\max}$, ϵ_τ, ϵ_ν, m_τ, n_ν, **y**

1　**for** $i = 1$ **to** $P_{\max}$ **do**

2　　$\tau_{\text{coarse}}^{(i)}, \nu_{\text{coarse}}^{(i)} = \arg\max_{(\tau,\nu)\in I_{DD}} \Phi(\mathbf{R}(\tau,\nu), \mathbf{y})$;

3　　$\hat{\tau}_{\text{fine}}^{(0)} = \tau_{\text{coarse}}^{(i)}$;

4　　$\hat{\nu}_{\text{fine}}^{(0)} = \nu_{\text{coarse}}^{(i)}$;

5　　**for** $s = 1$ **to** $s_{\max}$ **do**

6　　　$W_\tau^{(s)} = \frac{\tau_{\text{res}}}{m_\tau^{s-1}}$: delay search width;

7　　　$W_\nu^{(s)} = \frac{\nu_{\text{res}}}{n_\nu^{s-1}}$: Doppler search width;

8　　　$F_{DD}^{(s)} = \left\{ W_\tau^{(s)}\Gamma + \hat{\tau}_{\text{fine}}^{(s-1)} \right\} \times \left\{ W_\nu^{(s)}\Lambda + \hat{\nu}_{\text{fine}}^{(s-1)} \right\}$;

9　　　$\hat{\tau}_{\text{fine}}^{(s)}, \hat{\nu}_{\text{fine}}^{(s)} = \arg\max_{(\tau,\nu)\in F_{DD}^{(s)}} \Phi(\mathbf{R}(\tau,\nu), \mathbf{y})$;

10　　　**if** $s = s_{\max}$ ***or*** $\left(\left|\hat{\tau}_{fine}^{(s)} - \hat{\tau}_{fine}^{(s-1)}\right| < \epsilon_\tau$ ***and*** $\left|\hat{\nu}_{fine}^{(s)} - \hat{\nu}_{fine}^{(s-1)}\right| < \epsilon_\nu \right)$ **then**

11　　　　**break**;

12　　$\hat{\tau}[i] = \hat{\tau}_{\text{fine}}^{(s)}$;

13　　$\hat{\nu}[i] = \hat{\nu}_{\text{fine}}^{(s)}$;

14　　$\hat{\mathbf{g}} = \left(\mathbf{R}^H(\hat{\tau},\hat{\nu})\mathbf{R}(\hat{\tau},\hat{\nu}) \right)^{-1} \mathbf{R}^H(\hat{\tau},\hat{\nu})\mathbf{y}$;

15　　$\mathcal{E}^{(i)} = \mathbf{y} - \mathbf{R}(\hat{\tau},\hat{\nu})\hat{\mathbf{g}}$;

16　　**if** $i > 1$ **then**

17　　　**if** $\|\mathcal{E}^{(i)}\| < \epsilon$ **then**

18　　　　$\hat{P} = i$;

19　　　　**break**;

20　　　**else**

21　　　　Refinement procedure ($i_{\text{ref}} = 1, ..., i$);

22　**Output:** $\hat{\mathbf{H}} = \sum_{i=1}^{\hat{P}} \hat{g}_i \mathbf{G}_i(\hat{\tau}_i, \hat{\nu}_i)$;

3.4. Proposed P-IPIC Algorithm

　　The proposed variant of the DDIPIC algorithm aims to reduce the complexity of DDIPIC and is based on a RCF maximization and performs an iterative search as in the original algorithm until the maximum number of iterations is reached or the stopping criterion is met. The pseudocode of the proposed P-IPIC algorithm is provided in Algorithm 2. The main parameters are as for the DDIPIC algorithm.

Algorithm 2: Proposed P-IPIC

Input: $P_{\max}$, ϵ, $s_{\max}$, ϵ_τ, ϵ_ν, m_τ, n_ν, N_0, **y**

1   $\mathcal{E}^{(0)} = \mathbf{y}$;

2   **for** $i = 1$ **to** $P_{\max}$ **do**

3     $\tau_{\text{coarse}}^{(i)}, \nu_{\text{coarse}}^{(i)} = \arg\max_{(\tau,\nu) \in I_{DD}} \Phi\big(\mathbf{r}(\tau,\nu), \mathcal{E}^{(i-1)}\big)$;

4     $\hat{\tau}_{\text{fine}}^{(0)} = \tau_{\text{coarse}}^{(i)}$;

5     $\hat{\nu}_{\text{fine}}^{(0)} = \nu_{\text{coarse}}^{(i)}$;

6   **for** $s = 1$ **to** $s_{\max}$ **do**

7       $W_\tau^{(s)} = \frac{\tau_{res}}{m_\tau^{s-1}}$: delay search width;

8       $W_\nu^{(s)} = \frac{\nu_{res}}{n_\nu^{s-1}}$: Doppler search width;

9       $F_{DD}^{(s)} = \left\{ W_\tau^{(s)}\Gamma + \hat{\tau}_{\text{fine}}^{(s-1)} \right\} \times \left\{ W_\nu^{(s)}\Lambda + \hat{\nu}_{\text{fine}}^{(s-1)} \right\}$;

10       $\hat{\tau}_{\text{fine}}^{(s)}, \hat{\nu}_{\text{fine}}^{(s)} = \arg\max_{(\tau,\nu) \in F_{DD}^{(s)}} \Phi\big(\mathbf{r}(\tau,\nu), \mathcal{E}^{(i-1)}\big)$;

11   **if** $s = s_{\max}$ **or** $\left(\big|\hat{\tau}_{\text{fine}}^{(s)} - \hat{\tau}_{\text{fine}}^{(s-1)}\big| < \epsilon_\tau \text{ **and** } \big|\hat{\nu}_{\text{fine}}^{(s)} - \hat{\nu}_{\text{fine}}^{(s-1)}\big| < \epsilon_\nu \right)$ **then**

12   **break**;

13     $\hat{\tau}[i] = \hat{\tau}_{\text{fine}}^{(s)}$;

14     $\hat{\nu}[i] = \hat{\nu}_{\text{fine}}^{(s)}$;

15     $\hat{\mathbf{g}} = \left(\mathbf{R}^H(\hat{\tau},\hat{\nu})\mathbf{R}(\hat{\tau},\hat{\nu}) + N_0\mathbf{I} \right)^{-1} \mathbf{R}^H(\hat{\tau},\hat{\nu})\mathbf{y}$;

16     $\mathcal{E}^{(i)} = \mathbf{y} - \mathbf{R}(\hat{\tau},\hat{\nu})\hat{\mathbf{g}}$;

17   **if** $i > 1$ **then**

18   **if** $\|\mathcal{E}^{(i)}\| < \epsilon$ **then**

19         $\hat{P} = i$;

20   **break**;

21   Refinement procedure $(i_{\text{ref}} = 1, \ldots, \hat{P}) \rightarrow \hat{\hat{P}}$;

22   **Output:** $\hat{\mathbf{H}} = \sum_{i=1}^{\hat{\hat{P}}} \hat{g}_i \mathbf{G}_i(\hat{\tau}_i, \hat{\nu}_i)$;

The proposed algorithm features several key differences compared to Algorithm 1:

- *Coarse and fine estimation based on the residuals (lines 3 and 10):* The key idea is that once a path is detected and the residue vector is obtained, during the next search, a new path is discovered looking at the residual pilot signal of the previous iteration. Thus, at the *i*-th iteration, the algorithm looks for a new path maximizing an RCF given by

$$\Phi\big(\mathbf{r}(\tau,\nu), \mathcal{E}^{(i-1)}\big) = \big(\mathcal{E}^{(i-1)}\big)^H \mathbf{r}(\tau,\nu) \left(\mathbf{r}^H(\tau,\nu)\mathbf{r}(\tau,\nu) \right)^{-1} \mathbf{r}^H(\tau,\nu)\mathcal{E}^{(i-1)} \quad (20)$$

$$= \frac{\big(\mathbf{r}^H(\tau,\nu)\mathcal{E}^{(i-1)}\big)^* \mathbf{r}^H(\tau,\nu)\mathcal{E}^{(i-1)}}{\|\mathbf{r}(\tau,\nu)\|^2} = \frac{|\mathbf{r}^H(\tau,\nu)\mathcal{E}^{(i-1)}|^2}{\|\mathbf{r}(\tau,\nu)\|^2}. \quad (21)$$

In this way, at each iteration, the IPI between the paths is directly canceled, facilitating the new search.

- *Global refinement (line 21):* It is important to notice that this solution is prone to error propagation across all estimates. This may lead to residue vectors with residual peaks that are interpreted as small-amplitude paths by the algorithm. In order to avoid false alarms or multidetections due to residual uncancelled IPI, the P-IPIC algorithm needs a final refinement step in which all the estimates are refined while the residue in monitored. If the residue satisfies the stopping criterion, all remaining unrefined estimates are declared false alarms and, therefore, discarded. Specifically, during the *search phase*, a number of paths equal to $\hat{P}$ is detected. After that, during the *refinement*

phase the paths are refined as in the DDIPIC algorithm with the only difference in using the cost function in Equation (19) with a Tikhonov regularization, also known as ridge-regression, and in performing channel gain refinement as in line 15. As discussed above, during this phase some paths may be discarded. Therefore, the refinement procedure provides a number of estimated paths equal to $\hat{\hat{P}} \leq \hat{P}$. The number of paths detected during the *search phase* cannot be determined a priori but it is reasonable to assume that, if the IPI is not too large and the channel is made of a sufficiently high number of propagation paths, $\hat{\hat{P}} \approx \hat{P}$.

- *Regularization (line 15):* Another difference from the original version of the IPI cancellation algorithm is that the ordinary least squares (OLS) estimate of channel gains in Equation (18) is replaced by a regularized least squares (RLS) estimate to make the matrix inversion more stable. This is due to the fact that, due to residual IPI, some paths are detected multiple times, and as a result, the CDDPM matrix may have some almost collinear columns. Therefore, the RLS estimate of the channel gains vector is given by

$$\hat{\mathbf{g}} = \left(\mathbf{R}^H(\hat{\tau}, \hat{\nu}) \mathbf{R}(\hat{\tau}, \hat{\nu}) + N_0 \mathbf{I} \right)^{-1} \mathbf{R}^H(\hat{\tau}, \hat{\nu}) \mathbf{y} \tag{22}$$

where $\mathbf{I}$ is the identity matrix.

3.5. Latency/Complexity Comparison

The key advantage of the proposed approach is that it requires a single final refinement procedure compared to the $\hat{P} - 1$ refinements performed by the DDIPIC algorithm. A rough latency comparison of the two approaches is shown in Table 1, where we assumed that the P-IPIC algorithm does not overestimate the number of paths significantly during the first iterative procedure; thus, $\hat{\hat{P}} \approx \hat{P}$. Under this hypothesis, and denoting by T_{coarse} the average time needed to perform a coarse estimate and with T_{fine} the average time needed to perform a fine estimate, we can make the following observations.

1. DDIPIC: This performs a coarse and fine estimation for every path in a time equal to $\hat{P}(T_{\text{coarse}} + T_{\text{fine}})$. Moreover, from the second iteration to the end, it performs $\hat{P} - 1$ refinement procedures of the estimates. Therefore, at each refinement procedure, it performs i coarse and fine estimates.

2. P-IPIC: This performs a coarse and fine estimation for every path in a time equal to $\hat{P}(T_{\text{coarse}} + T_{\text{fine}})$ during the *search phase*. Moreover, the *refinement phase* performs a second coarse and fine estimate for each path.

Table 1. Latency comparison of the two IPIC algorithms.

Algorithm	Latency
DDIPIC [11]	$(T_{\text{coarse}} + T_{\text{fine}})\hat{P} + (T_{\text{coarse}} + T_{\text{fine}}) \sum_{i=2}^{\hat{P}} i =$ $2(T_{\text{coarse}} + T_{\text{fine}})\hat{P} + (T_{\text{coarse}} + T_{\text{fine}})\left(\frac{\hat{P}(\hat{P}-1)}{2} - 1 \right)$
P-IPIC (Prop.)	$2(T_{\text{coarse}} + T_{\text{fine}})\hat{P}$

During the *search phase*, the proposed P-IPIC algorithm runs coarse and fine estimation procedures of channel delays and Dopplers performing maximization of the cost function in Equation (21), while during the *refinement phase*, it maximizes the regularized version of Equation (19). The complexity comparison of the cost function in Equation (19) and the RCF is shown in Table 2. It can be noticed that the first advantage in computing the RCF is that it does not require matrix inversion but only multiplications and additions; instead, the computation of the cost function in Equation (19) does. In order to compare the two cost functions, we can make the following observations.

1. DDIPIC cost function in (19): During the *i*-th iteration, the number of required operations is

(a) $\mathbf{y}^H \mathbf{R}(\boldsymbol{\tau}, \boldsymbol{\nu})$, $\mathbf{y} \in \mathbb{C}^{MN \times 1}$, $\mathbf{R}(\boldsymbol{\tau}, \boldsymbol{\nu}) \in \mathbb{C}^{MN \times i}$ requires MNi multiplications and $(MN-1)i$ additions. Then, $\mathbf{R}^H(\boldsymbol{\tau}, \boldsymbol{\nu})\mathbf{y}$ is simply obtained applying the Hermitian operator.

(b) $\mathbf{R}^H(\boldsymbol{\tau}, \boldsymbol{\nu})\mathbf{R}(\boldsymbol{\tau}, \boldsymbol{\nu}) \in \mathbb{C}^{i \times i}$ requires MNi^2 multiplications and $(MN-1)i^2$ additions. After that, matrix inversion has a complexity in the order of $\mathcal{O}(i^3)$.

(c) $\mathbf{y}^H \mathbf{R}(\boldsymbol{\tau}, \boldsymbol{\nu}) \left(\mathbf{R}^H(\boldsymbol{\tau}, \boldsymbol{\nu})\mathbf{R}(\boldsymbol{\tau}, \boldsymbol{\nu}) \right)^{-1}$ requires i^2 multiplications and $i(i-1)$ additions.

(d) The matrix multiplication of the $1 \times i$ matrix obtained in the previous step by $\left(\mathbf{y}^H \mathbf{R}(\boldsymbol{\tau}, \boldsymbol{\nu}) \right)^H$ requires i multiplications and $i-1$ additions.

2. P-IPIC RCF: during the i-th iteration, the number of required operations is

(a) $\mathbf{r}^H(\tau, \nu)\mathcal{E}^{(i-1)}$, $\mathbf{r} \in \mathbb{C}^{MN \times 1}$, $\mathcal{E}^{(i-1)} \in \mathbb{C}^{MN \times 1}$ requires MN multiplications and $MN-1$ additions. After that, an additional multiplication is needed to compute the squared module.

(b) $\|\mathbf{r}(\tau, \nu)\|^2$ requires MN multiplications and $MN-1$ additions.

(c) One final division is required to compute the ratio between numerator and denominator.

The computation of the RCF is independent of the iteration index i, while the computational complexity of the cost function in Equation (19) increases as $\mathcal{O}(i^3)$ due to the inversion of a $i \times i$ matrix.

In the end, the search cost of the proposed P-IPIC is identical to that of the DDIPIC, since coarse and fine estimates are performed in the same way apart from the cost function. Hence, as in [11], the search cost depends on the number of points in the integer DD grid during the coarse estimate, or the number of points in the fractional search space $F_{DD}^{(s)}$ during the fine estimate. In particular, during the coarse estimation phase, both algorithms perform $(L+1)(2K+1)$ cost function maximizations, while during the fine estimation phase, they perform at most $\left(2\lfloor \frac{m_\tau}{2} \rfloor + 1\right)\left(2\lfloor \frac{n_\nu}{2} \rfloor + 1\right)$ maximizations.

Table 2. Comparison of computational complexity of cost functions.

Cost Function	Equation (19)	RCF
Multiplications	$(MN+1)(i^2+i)$	$2MN+1$
Additions	$(MN-1)(i^2+i)+i^2-1$	$2(MN-1)$
Total number of operations	$2MN(i^2+i)+i^2-1+\mathcal{O}(i^3)$	$4MN$

4. Simulation Results

In this section, the proposed method is evaluated and compared against the baseline DDIPIC method.

4.1. Simulation Parameters

To investigate the performance of the proposed approach, we consider OTFS modulation with $M = 16$ and $N = 16$ with subcarrier spacing of $\Delta f = 25$ kHz and a carrier frequency $f_c = 5.1$ GHz. Thus, delay and Doppler resolutions are $\tau_{\text{res}} = 2.5$ μs, and $\nu_{\text{res}} = 1.5625$ kHz. As in [11], the parameters for running channel estimation algorithms are $P_{\max} = 15$, $s_{\max} = 10$, $\epsilon_\tau = 10^{-10}$, $\epsilon_\nu = 10^{-2}$, $m_\tau = 10$, $n_\nu = 10$. Finally, as discussed above, the convergence tolerance parameter for the stopping criterion is set according to the noise variance and OTFS frame parameters.

4.2. Simulation Scenarios

The high-mobility multipath channel is assumed to have $P = 4$ Rayleigh faded paths with exponential power delay profile (PDP) with a decaying time constant of 10 μs and a maximum delay of 10 μs. We consider two different case studies corresponding to different scenarios:

1. *Low-IPI Regime*: Propagation delays and Doppler shifts assume fractional values but the paths are far enough to guarantee low IPI. Thus, all paths are separable in the DD domain.

   The delays and Doppler shifts of the paths are $\boldsymbol{\tau} = \begin{bmatrix} 0 & 0.86\tau_{\text{res}} & 2.11\tau_{\text{res}} & 3.2\tau_{\text{res}} \end{bmatrix}$ s, $\boldsymbol{\nu} = \begin{bmatrix} -1.87\nu_{\text{res}} & 2.24\nu_{\text{res}} & -3\nu_{\text{res}} & 1.3\nu_{\text{res}} \end{bmatrix}$ Hz, respectively.

2. *High-IPI Regime*: Propagation delays and Doppler shifts assume fractional values, and the paths are close enough to create high IPI.

   The propagation delays of the paths are $\boldsymbol{\tau} = \begin{bmatrix} 0 & 2.3 & 3.15 & 7.7 \end{bmatrix}$ μs. The maximum user equipment (UE) speed is $v_{UE} = 190 \, \frac{\text{m}}{\text{s}} (684 \frac{\text{km}}{\text{h}})$, resulting in a maximum Doppler spread of $\nu_{\max} = \frac{v_{UE}}{c} f_c = 3.23$ kHz, where $c = 3 \cdot 10^8 \, \frac{\text{m}}{\text{s}}$ is the speed of light.

   The Doppler shifts of all paths are generated assuming Jakes' Doppler power spectrum (DPS) using $\nu_i = \nu_{\max} cos(\theta_i)$ where $\theta_i \sim \mathcal{U}[0, 2\pi]$.

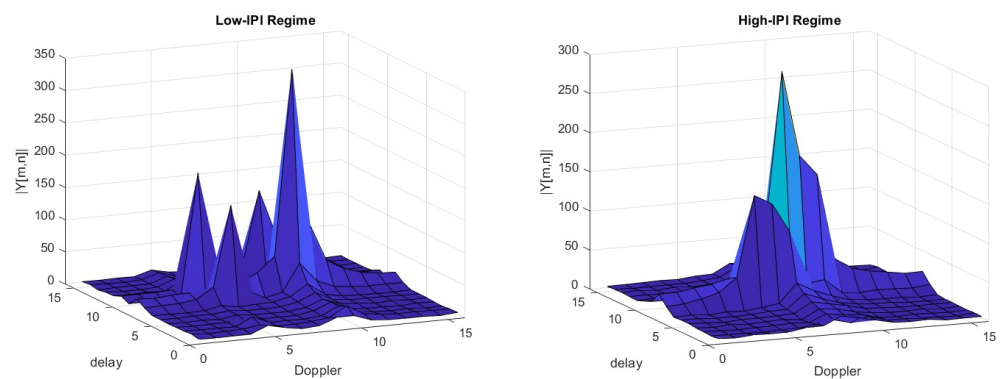   An example of a pilot frame received in both regimes is provided in Figure 1.

Figure 1. Received DD domain pilot signal.

4.3. Performance Metrics

We will consider the following performance metrics:

- *Channel estimation:* The accuracy of the channel estimate is measured through the normalized mean square error (NMSE) computed as

$$\text{NMSE} = \mathbb{E}\left[\frac{\|\mathbf{H} - \hat{\mathbf{H}}\|_F^2}{\|\mathbf{H}\|_F^2} \right]. \tag{23}$$

- *Channel parameter estimation:* The accuracy of the channel parameter estimates is also considered evaluating the normalized root mean square error (NRMSE) as

$$\text{NRMSE}(\vartheta) = \sqrt{\frac{\mathbb{E}\left[\sum_{i=1}^{P} |\vartheta_i - \hat{\vartheta}_i|^2 \right]}{\sum_{i=1}^{P} |\vartheta_i|^2}}, \tag{24}$$

where ϑ is a variable that represents delay, Doppler, or channel gain.

- *Model order performance:* The estimation of P will be evaluated by the average number of estimated paths and the probability mass function (PMF) of the number of detected paths.
- *Communication:* Communication performance of the proposed approach are evaluated by measuring the bit error rate (BER) as a function of the SNR/bit in order to investigate for the detection robustness when real channel estimation with the proposed variant is performed at the receiver. The SNR/bit is given by E_b/N_0 where E_b is the average energy per bit. We consider a quadrature amplitude modulation (QAM) with $Q = 16$ constellation points and $E_b = \frac{2(Q-1)}{3 log_2 Q}$. The input–output relation for data

frames is the one in Equation (9). At the receiver, a linear minimum mean square error (LMMSE) equalizer provides the estimated symbols as

$$\hat{\mathbf{x}} = \left(\hat{\mathbf{H}}^H\hat{\mathbf{H}} + \frac{1}{\text{SNR}}\mathbf{I}\right)^{-1}\hat{\mathbf{H}}^H\mathbf{y}, \tag{25}$$

where $\text{SNR} = \frac{MNE_s/NT}{M\Delta f N_0} = \frac{E_s}{N_0}$ is the communication signal-to-noise ratio and $E_s = E_b log_2 Q$ is the average energy per symbol. After that, a symbol-by-symbol maximum likelihood (ML) detector takes the decision.

4.4. Results and Discussion

4.4.1. Channel Estimation Performance

Figure 2 shows the NMSE as a function of the pilot SNR in both low- and high-IPI regimes. In the low-IPI case at low PSNR, the performance of the proposed variant and the DDIPIC algorithm are comparable, while at higher values of PSNR, the proposed variant outperforms the DDIPIC algorithm, achieving much lower NMSE. On the other hand, in the high-IPI regime, the P-IPIC algorithm achieves an NMSE 1–2 dB lower than the DDIPIC, except for the case of PSNR = 35 dB, where the NSME performances become comparable. We attribute this performance gain to the fact that during the *search phase,* possible paths are detected with the goal of reducing the residue energy, and thus to achieve high accuracy in channel matrix estimate. After that, the good channel estimate obtained in the *search phase* is used to refine the estimates and declare false alarms. The performance of the threshold method in [10] is also presented in both cases as a baseline for comparison, and it can be noticed they are drastically degraded due to the presence of IPI, since it assumes integer DDs.
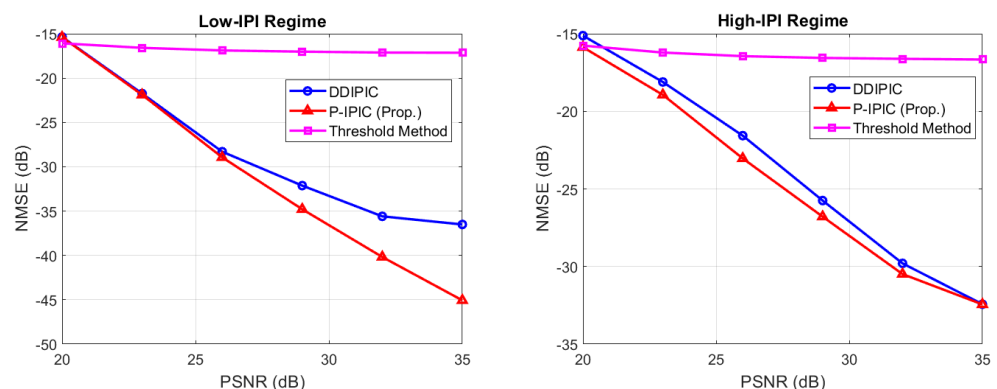

Figure 2. NMSE as a function of pilot SNR.

4.4.2. Channel Parameter Estimation Performance

Figure 3 shows the NRMSE of channel gains, delays, and Doppler shift as a function of pilot SNR in a channel randomly generated according to channel models 1 and 2. In both low- and high-IPI regimes, the P-IPIC is capable of estimating channel parameters with higher accuracy. It can be noticed that in the high-IPI case, at low PSRN, the NRMSE is high due to the fact that small amplitude paths are missed. On the other side, at high PSNR, the NRMSE for the P-IPIC increases because small errors in the parameter estimates during the *search phase* lead to higher residual IPI that is interpreted as small-amplitude paths, reducing the overall accuracy of the estimates. However, the P-IPIC still outperforms the DDIPIC algorithm for high PSNR.
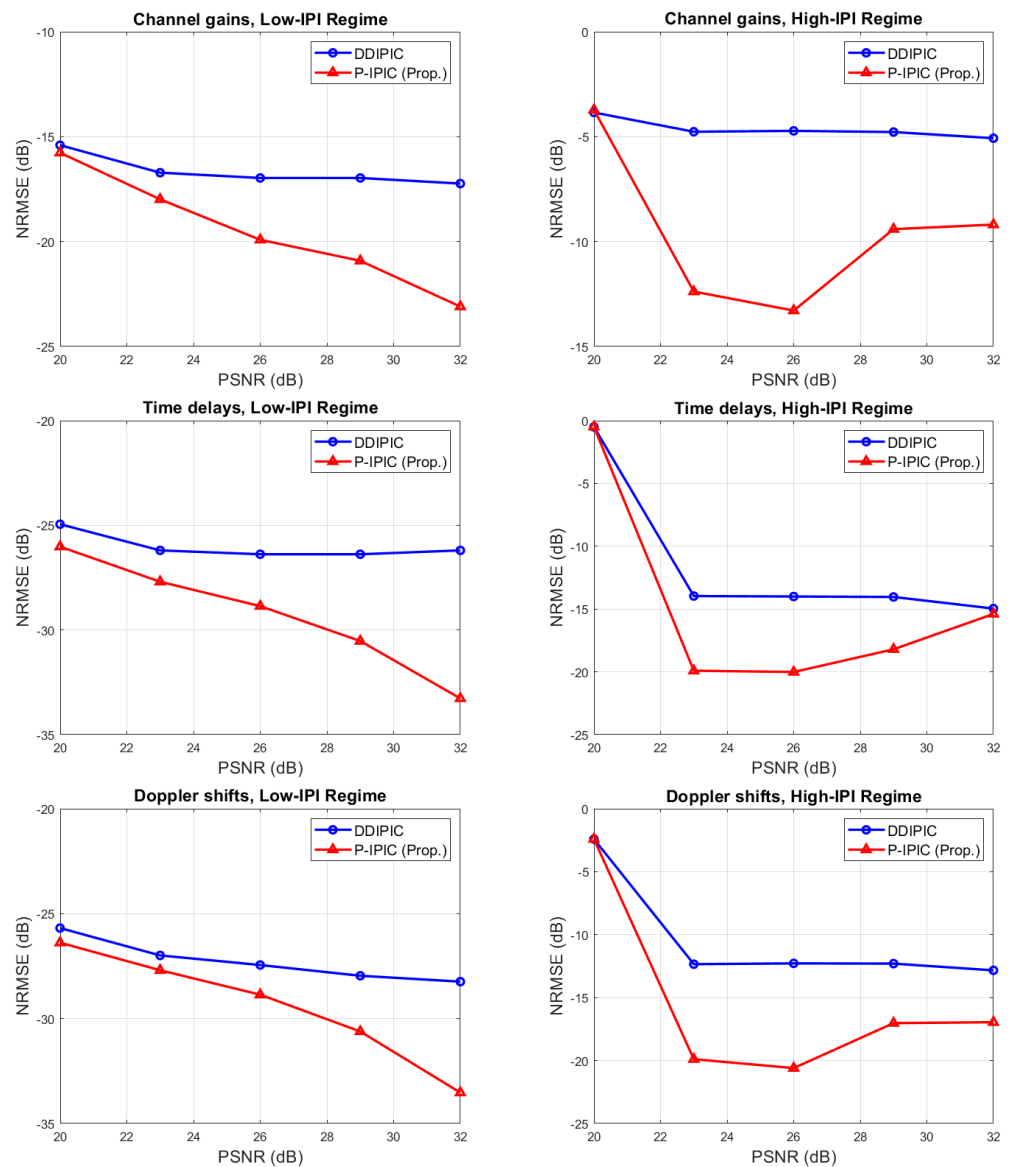
Figure 3. NRMSE of channel parameters as a function of pilot SNR.

4.4.3. Number of Detected Paths

Figure 4 shows the average number of detected paths as a function of PSNR. In the low-IPI case, performances of both algorithms are comparable. At low PSNR, the average number of detected paths is below the true value ($P = 4$) because small-amplitude paths are missed. At higher PSNR values, both algorithms are able to detect the right number of paths. On the other hand, in the high-IPI case, detection capabilities are still comparable, but at high PSNR values, both algorithms overestimate a bit the number of propagation paths due to high residual interference effects. Figure 5 shows the probability distribution among all the possible values for P in the case of PSNR = 35 dB. In the low-IPI regime, both algorithms detect the right number of paths with a probability close to ~0.98. Thus, the probability of miss detection and the probability of false alarm are close to zero (probability of miss dtection < 0.02). In the high-IPI regime, the probability that the algorithm finds the correct number of paths is reduced to ~0.73 for the DDIPIC and ~0.65 for the P-IPIC. The probability of miss detection is close to zero in both cases (~0.015 for the P-IPIC). In the end, the probability of false alarm is ~0.27 for DDIPIC and ~0.34 for P-IPIC.
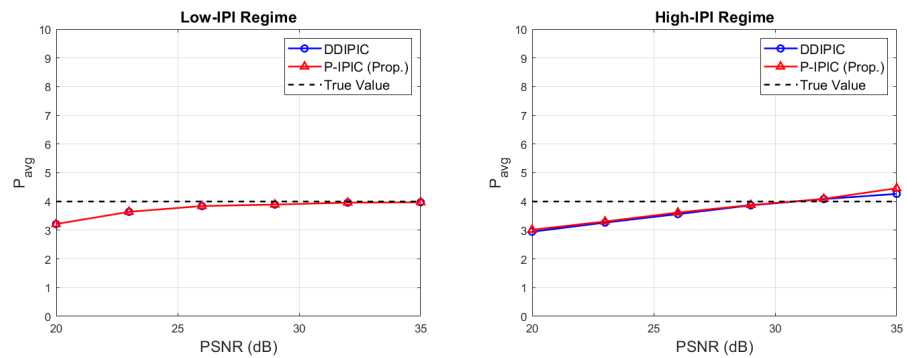
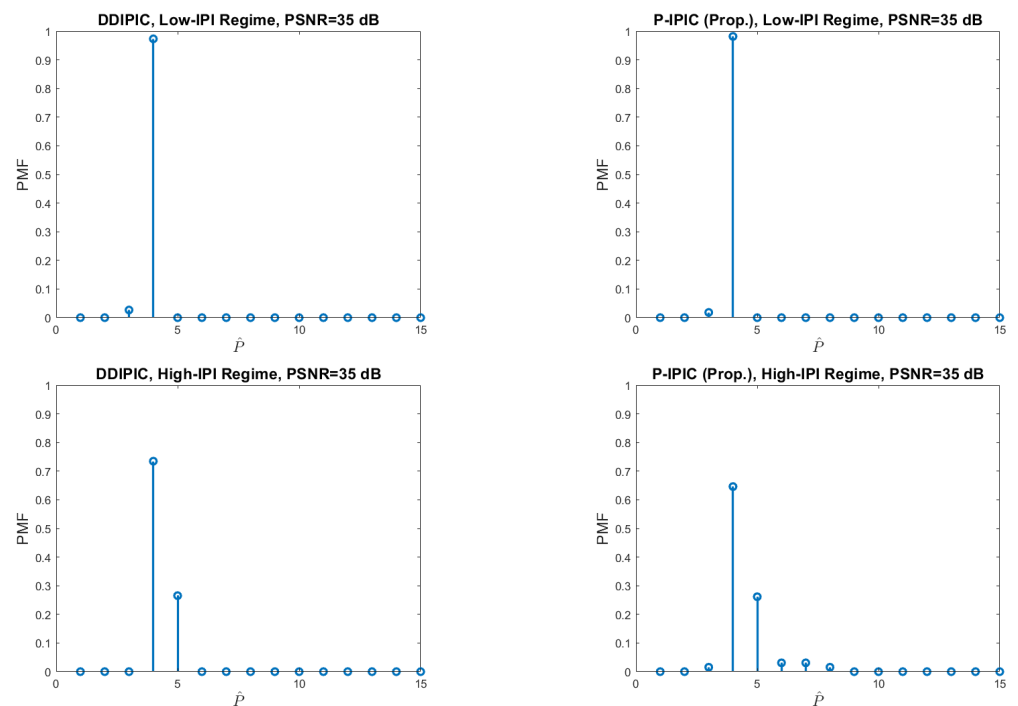**Figure 4.** Average number of estimated paths as a function of pilot SNR.

Figure 5. Probability mass function of the number of detected paths.

4.4.4. Communication Performance

BER performance with the estimated channel (PSNR = 30 dB) and with perfect channel state information at the receiver (CSIR) is provided in Figure 6, showing the BER performance as a function of the SNR/bit in the low- and high-IPI regimes. As expected, the proposed P-IPIC achieves lower bit error rates in both regimes. In addition, BER curves are closer to the ideal curves obtained with perfect channel state information at the receiver.
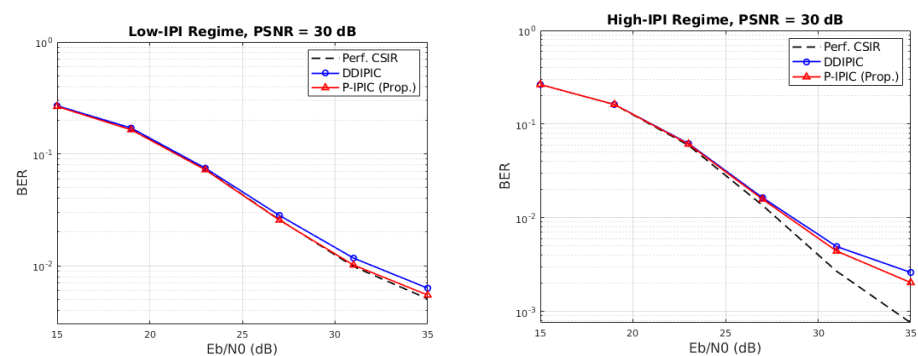

Figure 6. Bit error rate as a function of SNR/bit.

5. Conclusions and Future Works

In this work, we addressed the challenge of low-complexity fractional delay-Doppler (DD) channel estimation in orthogonal time–frequency space (OTFS) systems due to inter-path interference (IPI). A variant of the delay-Doppler inter-path interference cancellation (DDIPIC) algorithm is proposed, which progressively cancels IPI as estimates are obtained, requiring only a final refinement procedure. This approach reduces the latency of the algorithm, with the time difference in latency being almost proportional to the square of the number of estimated paths. Numerical results demonstrate that the proposed algorithm achieves both lower normalized mean square error (NMSE) and bit error rate (BER) compared to other schemes with comparable detection capabilities. The high-IPI scenario is also taken into account, in which the delay and Doppler resolutions are not sufficiently fine, and IPI cancellation becomes tricky.

In future work, the use of deep learning (DL) approaches, as in [24], can be considered to further reduce computational complexity while maintaining good performance and the analysis in presence of interference will be included.

Author Contributions: Conceptualization, M.M., P.S. (Paolo Spallaccini) and P.S. (Pietro Savazzi); Methodology, M.M., H.W. and P.S. (Pietro Savazzi); Validation, H.W. and P.S. (Paolo Spallaccini); Investigation, M.M.; Supervision, H.W. and P.S. (Pietro Savazzi). All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhang, Z.; Xiao, Y.; Ma, Z.; Xiao, M.; Ding, Z.; Lei, X.; Karagiannidis, G.K.; Fan, P. 6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies. *IEEE Veh. Technol. Mag.* **2019**, *14*, 28–41. [CrossRef]
2. Giordani, M.; Polese, M.; Mezzavilla, M.; Rangan, S.; Zorzi, M. Toward 6G Networks: Use Cases and Technologies. *IEEE Commun. Mag.* **2020**, *58*, 55–61. [CrossRef]
3. Wang, T.; Proakis, J.; Masry, E.; Zeidler, J. Performance degradation of OFDM systems due to Doppler spreading. *IEEE Trans. Wirel. Commun.* **2006**, *5*, 1422–1432. [CrossRef]
4. Zhou, Y.; Yin, H.; Xiong, J.; Song, S.; Zhu, J.; Du, J.; Chen, H.; Tang, Y. Overview and Performance Analysis of Various Waveforms in High Mobility Scenarios. *arXiv* **2023**, arXiv:2302.14224. https://doi.org/10.48550/arXiv.2302.14224.
5. Hadani, R.; Rakib, S.; Tsatsanis, M.; Monk, A.; Goldsmith, A.J.; Molisch, A.F.; Calderbank, R. Orthogonal Time Frequency Space Modulation. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6. [CrossRef]
6. Hadani, R.; Monk, A. OTFS: A New Generation of Modulation Addressing the Challenges of 5G. *arXiv* **2018**, arXiv:1802.02623. https://doi.org/10.48550/arXiv.1802.02623.
7. Hong, Y.; Thaj, T.; Viterbo, E. *Delay-Doppler Communications: Principles and Applications*; Elsevier: Amsterdam, The Netherlands, 2022. [CrossRef]
8. Khan, I.A.; Mohammed, S.K. Low Complexity Channel Estimation for OTFS Modulation with Fractional Delay and Doppler. *arXiv* **2021**, arXiv:2111.06009. https://doi.org/10.48550/arXiv.2111.06009
9. Zacharia, O.; Vani Devi, M. Fractional Delay and Doppler Estimation for OTFS Based ISAC Systems. In Proceedings of the 2023 IEEE Wireless Communications and Networking Conference (WCNC), Glasgow, UK, 26–29 March 2023; pp. 1–6. [CrossRef]
10. Raviteja, P.; Phan, K.T.; Hong, Y. Embedded Pilot-Aided Channel Estimation for OTFS in Delay–Doppler Channels. *IEEE Trans. Veh. Technol.* **2019**, *68*, 4906–4917. [CrossRef]
11. Muppaneni, S.P.; Mattu, S.R.; Chockalingam, A. Channel and Radar Parameter Estimation with Fractional Delay-Doppler Using OTFS. *IEEE Commun. Lett.* **2023**, *27*, 1392–1396. [CrossRef]

12. Raviteja, P.; Phan, K.T.; Hong, Y.; Viterbo, E. Interference Cancellation and Iterative Detection for Orthogonal Time Frequency Space Modulation. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 6501–6515. [CrossRef]
13. Raviteja, P.; Phan, K.T.; Jin, Q.; Hong, Y.; Viterbo, E. Low-complexity iterative detection for orthogonal time frequency space modulation. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, 15–18 April 2018; pp. 1–6. [CrossRef]
14. Tiwari, S.; Das, S.S.; Rangamgari, V. Low complexity LMMSE Receiver for OTFS. *IEEE Commun. Lett.* **2019**, *23*, 2205–2209. [CrossRef]
15. Surabhi, G.D.; Chockalingam, A. Low-Complexity Linear Equalization for OTFS Modulation. *IEEE Commun. Lett.* **2020**, *24*, 330–334. [CrossRef]
16. Shen, W.; Dai, L.; An, J.; Fan, P.; Heath, R.W. Channel Estimation for Orthogonal Time Frequency Space (OTFS) Massive MIMO. *IEEE Trans. Signal Process.* **2019**, *67*, 4204–4217. [CrossRef]
17. Zhang, R.; Cheng, L.; Wang, S.; Lou, Y.; Wu, W.; Ng, D.W.K. Tensor Decomposition-Based Channel Estimation for Hybrid mmWave Massive MIMO in High-Mobility Scenarios. *IEEE Trans. Commun.* **2022**, *70*, 6325–6340. [CrossRef]
18. Zhang, R.; Cheng, L.; Wang, S.; Lou, Y.; Gao, Y.; Wu, W.; Ng, D.W.K. Integrated Sensing and Communication with Massive MIMO: A Unified Tensor Approach for Channel and Target Parameter Estimation. *IEEE Trans. Wirel. Commun.* **2024**. [CrossRef]
19. Wu, X.; Ma, S.; Yang, X. Tensor-based low-complexity channel estimation for mmWave massive MIMO-OTFS systems. *J. Commun. Inf. Netw.* **2020**, *5*, 324–334. [CrossRef]
20. Yogesh, V.; Mattu, S.R.; Chockalingam, A. Low-Complexity Delay-Doppler Channel Estimation in Discrete Zak Transform Based OTFS. *IEEE Commun. Lett.* **2024**, *28*, 672–676. [CrossRef]
21. Zhao, L.; Gao, W.J.; Guo, W. Sparse Bayesian Learning of Delay-Doppler Channel for OTFS System. *IEEE Commun. Lett.* **2020**, *24*, 2766–2769. [CrossRef]
22. Wei, Z.; Yuan, W.; Li, S.; Yuan, J.; Ng, D.W.K. Off-Grid Channel Estimation with Sparse Bayesian Learning for OTFS Systems. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 7407–7426. [CrossRef]
23. Sheng, H.T.; Wu, W.R. Time-Frequency Domain Channel Estimation for OTFS Systems. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 937–948. [CrossRef]
24. Mattu, S.R.; Chockalingam, A. Learning in Time-Frequency Domain for Fractional Delay-Doppler Channel Estimation in OTFS. *IEEE Wirel. Commun. Lett.* **2024**, *13*, 1245–1249. [CrossRef]