Direct Multipath-Based SLAM: Supporting Documents

Mingchao Liang, Erik Leitinger, and Florian Meyer

This manuscript provides derivations and further experimental results for the publication, "Direct Multipath-Based SLAM" by the same authors [1].

1 The Joint Posterior PDF $f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k}|\mathbf{z}_{1:k})$

In this section, we present the derivation of the factorization of the joint posterior probability density function (PDF) [1], which lays the foundation of the factor graph representation in [1] and the starting point of the proposed belief propagation (BP) algorithm.

1.1 The Joint Likelihood PDF $f(\mathbf{z}_{1:k}|\mathbf{x}_{1:k}, \mathbf{y}_{1:k}, \boldsymbol{\eta}_{1:k})$

At each time step k, each physical anchor (PA) j receives a measurement $\mathbf{z}_k^{(j)} \in \mathbb{C}^M$. The measurements $\mathbf{z}_k^{(j)}$ are independent conditioned on the $\mathbf{x}_k$, $\mathbf{y}_k^{(j)}$ and $\eta_k^{(j)}$ as discussed in [1, Sec. II-B]. As a result, the joint likelihood function can be represented as

$$f(\mathbf{z}_{1:k}|\mathbf{x}_{1:k}, \mathbf{y}_{1:k}, \boldsymbol{\eta}_{1:k}) = \prod_{k'=1}^{k} \prod_{j=1}^{J} f(\mathbf{z}_{k'}^{(j)}|\mathbf{x}_{k'}, \mathbf{y}_{k'}^{(j)}, \eta_{k'}^{(j)}) \tag{1}$$

where $f(\mathbf{z}_{k'}|\mathbf{x}_{k'}, \mathbf{y}_{k'}, \eta_{k'}^{(j)})$ is introduced in [1, eq. (5)].

1.2 The Joint Prior PDF $f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k})$

We start from applying the product rule on the joint prior PDF $f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k})$, from which we have

$$f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k}) = f(\mathbf{x}_0, \mathbf{y}_0, \boldsymbol{\eta}_0) \prod_{k'=1}^{k} f(\mathbf{x}_{k'}, \mathbf{y}_{k'}, \boldsymbol{\eta}_{k'}|\mathbf{x}_{0:k'-1}, \mathbf{y}_{0:k'-1}, \boldsymbol{\eta}_{0:k'-1}) \tag{2}$$

We then apply the product rule to $f(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\eta}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{0:k-1}, \boldsymbol{\eta}_{0:k-1})$

$$\begin{aligned} f(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\eta}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{0:k-1}, \boldsymbol{\eta}_{0:k-1}) &= f(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{0:k-1}, \boldsymbol{\eta}_{0:k-1}) \\ &\quad \times f(\mathbf{y}_k|\mathbf{x}_{0:k}, \mathbf{y}_{0:k-1}, \boldsymbol{\eta}_{0:k-1}) \\ &\quad \times f(\boldsymbol{\eta}_k|\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k-1}). \end{aligned}$$

Following the conditional independence assumption in [1, Sec. II-C], we obtain

$$\begin{aligned} f(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{0:k-1}, \boldsymbol{\eta}_{0:k-1}) &= f(\mathbf{x}_k|\mathbf{x}_{k-1}) \\ f(\mathbf{y}_k|\mathbf{x}_{0:k}, \mathbf{y}_{0:k-1}, \boldsymbol{\eta}_{0:k-1}) &= \prod_{j=1}^{J} \prod_{n=1}^{N_{k-1}^{(j)}} f(\mathbf{y}_{k,n}^{(j)}|\mathbf{y}_{k-1,n}^{(j)}) \prod_{n=N_{k-1}^{(j)}+1}^{N_k^{(j)}} f(\mathbf{y}_{k,n}^{(j)}|\mathbf{x}_k) \\ f(\boldsymbol{\eta}_k|\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k-1}) &= \prod_{j=1}^{J} f(\eta_k^{(j)}|\eta_{k-1}^{(j)}) \end{aligned}$$

The the variables $\mathbf{x}_0$, $\eta_0^{(j)}$, $j \in \{1, \ldots, J\}$, and $\mathbf{y}_{0,n}^{(j)}$, $n \in \{1, \ldots, N_0^{(j)}\}$, $j \in \{1, \ldots, J\}$ are also assumed to be independent, i.e.,

$$f(\mathbf{x}_0, \mathbf{y}_0, \boldsymbol{\eta}_0) = f(\mathbf{x}_0) \prod_{j=1}^{J} f(\eta_0^{(j)}) \bigg(\prod_{n=1}^{N_0^{(j)}} f(\mathbf{y}_{0,n}^{(j)}) \bigg).$$

Finally, we insert the results to (2) and have

$$\begin{aligned} f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k}) = {} & f(\mathbf{x}_0) \prod_{j=1}^{J} f(\eta_0^{(j)}) \bigg(\prod_{n=1}^{N_0^{(j)}} f(\mathbf{y}_{0,n}^{(j)}) \bigg) \prod_{k'=1}^{k} f(\mathbf{x}_{k'} | \mathbf{x}_{k'-1}) \\ & \times \prod_{j=1}^{J} f(\eta_{k'}^{(j)} | \eta_{k'-1}^{(j)}) \bigg(\prod_{n=1}^{N_{k'-1}^{(j)}} f(\mathbf{y}_{k',n}^{(j)} | \mathbf{y}_{k'-1,n}^{(j)}) \bigg) \bigg(\prod_{n=N_{k'-1}^{(j)}+1}^{N_{k'}^{(j)}} f(\mathbf{y}_{k',n}^{(j)} | \mathbf{x}_{k'}) \bigg). \end{aligned} \tag{3}$$

1.3 Final Expression for the Joint Posterior PDF $f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k} | \mathbf{z}_{1:k})$

Using Bayes' rule and inserting (1) and (3), the joint posterior PDF $f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k} | \mathbf{z}_{1:k})$ can be represented as in [1, eq. (13)]

$$\begin{aligned} f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k} | \mathbf{z}_{1:k}) & \propto f(\mathbf{z}_{1:k} | \mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k}) f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k}) \\ & = f(\mathbf{z}_{1:k} | \mathbf{x}_{1:k}, \mathbf{y}_{1:k}, \boldsymbol{\eta}_{1:k}) f(\mathbf{x}_{0:k}, \mathbf{y}_{0:k}, \boldsymbol{\eta}_{0:k}) \\ & = f(\mathbf{x}_0) \prod_{j=1}^{J} f(\eta_0^{(j)}) \bigg(\prod_{n=1}^{N_0^{(j)}} f(\mathbf{y}_{0,n}^{(j)}) \bigg) \prod_{k'=1}^{k} f(\mathbf{x}_{k'} | \mathbf{x}_{k'-1}) \\ & \quad \times \prod_{j=1}^{J} \bigg(\prod_{n=1}^{N_{k'-1}^{(j)}} f(\mathbf{y}_{k',n}^{(j)} | \mathbf{y}_{k'-1,n}^{(j)}) \bigg) \bigg(\prod_{n=N_{k'-1}^{(j)}+1}^{N_{k'}^{(j)}} f(\mathbf{y}_{k',n}^{(j)} | \mathbf{x}_{k'}) \bigg) \\ & \quad \times f(\eta_{k'}^{(j)} | \eta_{k'-1}^{(j)}) f(\mathbf{z}_{k'}^{(j)} | \mathbf{x}_{k'}, \mathbf{y}_{k'}^{(j)}, \eta_{k'}^{(j)}). \end{aligned} \tag{4}$$

2 Computation of Matrices $\mathbf{C}_{\iota,k}^{(j)}(\mathbf{x}_k)$, $\mathbf{C}_{\kappa,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j)})$, and $\mathbf{C}_{\nu,k}^{(j)}$

In this section, we present the derivation of the covariance matrices in [1, (21)-(23)]. By definition, $\mathbf{C}_{\kappa,k,n}^{(j)}$ is computed as

$$\begin{aligned} \mathbf{C}_{\kappa,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j)}) = {} & \int \mathbf{z}_k^{(j)} \mathbf{z}_k^{(j)\mathrm{H}} \kappa(\mathbf{y}_{k,n}^{(j)}; \mathbf{z}_k^{(j)}) \, \mathrm{d}\mathbf{z}_k^{(j)} \\ = {} & \sum_{r_{k,1}^{(j)} \in \{0,1\}} \cdots \sum_{r_{k,n-1}^{(j)} \in \{0,1\}} \sum_{r_{k,n+1}^{(j)} \in \{0,1\}} \cdots \sum_{r_{k,N_k^{(j)}}^{(j)} \in \{0,1\}} \int \cdots \int \bigg(\int \mathbf{z}_k^{(j)} \mathbf{z}_k^{(j)\mathrm{H}} f(\mathbf{z}_k^{(j)} | \mathbf{x}_k, \mathbf{y}_k^{(j)}, \eta_k^{(j)}) \, \mathrm{d}\mathbf{z}_k^{(j)} \bigg) \\ & \times \xi(\eta_k^{(j)}) \beta(\mathbf{x}_k) \prod_{\substack{n'=1 \\ n' \neq n}}^{N_k^{(j)}} \alpha(\mathbf{y}_{k,n'}^{(j)}) \mathrm{d}\boldsymbol{\phi}_{k,1}^{(j)} \cdots \mathrm{d}\boldsymbol{\phi}_{k,n-1}^{(j)} \, \mathrm{d}\boldsymbol{\phi}_{k,n+1}^{(j)} \cdots \mathrm{d}\boldsymbol{\phi}_{k,N_k^{(j)}}^{(j)} \, \mathrm{d}\eta_k^{(j)} \, \mathrm{d}\mathbf{x}_k \end{aligned} \tag{5}$$

where $\int \mathbf{z}_k^{(j)}\mathbf{z}_k^{(j)\mathrm{H}} f(\mathbf{z}_k^{(j)}|\mathbf{x}_k,\mathbf{y}_k^{(j)},\eta_k^{(j)})\,\mathrm{d}\mathbf{z}_k^{(j)} = \mathbf{C}_{k,n}^{(j)} = \eta_k^{(j)}\mathbf{I}_M + \sum_{n=1}^{N_k^{(j)}} r_{k,n}^{(j)}\gamma_{k,n}^{(j)}\mathbf{h}_{k,n}^{(j)}\mathbf{h}_{k,n}^{(j)\mathrm{H}}$ as discussed in [1, Sec. II-B]. Plugging in the result we obtain

$$\begin{aligned}
\mathbf{C}_{\kappa,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j)}) &= \sum_{r_{k,1}^{(j)}\in\{0,1\}} \cdots \sum_{r_{k,n-1}^{(j)}\in\{0,1\}} \sum_{r_{k,n+1}^{(j)}\in\{0,1\}} \cdots \sum_{r_{k,N_k^{(j)}}^{(j)}\in\{0,1\}} \int\cdots\int \Bigg(\eta_k^{(j)}\mathbf{I}_M + \sum_{n=1}^{N_k^{(j)}} r_{k,n}^{(j)}\gamma_{k,n}^{(j)}\mathbf{h}_{k,n}^{(j)}\mathbf{h}_{k,n}^{(j)\mathrm{H}}\Bigg) \\
&\quad \times \xi(\eta_k^{(j)})\beta(\mathbf{x}_k) \prod_{\substack{n'=1\\ n'\neq n}}^{N_k^{(j)}} \alpha(\mathbf{y}_{k,n'}^{(j)})\mathrm{d}\boldsymbol{\phi}_{k,1}^{(j)}\cdots\mathrm{d}\boldsymbol{\phi}_{k,n-1}^{(j)}\,\mathrm{d}\boldsymbol{\phi}_{k,n+1}^{(j)}\cdots\mathrm{d}\boldsymbol{\phi}_{k,N_k^{(j)}}^{(j)}\,\mathrm{d}\eta_k^{(j)}\,\mathrm{d}\mathbf{x}_k \\
&= \sum_{r_{k,1}^{(j)}\in\{0,1\}} \cdots \sum_{r_{k,n-1}^{(j)}\in\{0,1\}} \sum_{r_{k,n+1}^{(j)}\in\{0,1\}} \cdots \sum_{r_{k,N_k^{(j)}}^{(j)}\in\{0,1\}} \int\cdots\int \Bigg(\sum_{n=1}^{N_k^{(j)}} r_{k,n}^{(j)}\gamma_{k,n}^{(j)}\mathbf{h}_{k,n}^{(j)}\mathbf{h}_{k,n}^{(j)\mathrm{H}}\Bigg) \\
&\quad \times \beta(\mathbf{x}_k) \prod_{\substack{n'=1\\ n'\neq n}}^{N_k^{(j)}} \alpha(\mathbf{y}_{k,n'}^{(j)})\mathrm{d}\boldsymbol{\phi}_{k,1}^{(j)}\cdots\mathrm{d}\boldsymbol{\phi}_{k,n-1}^{(j)}\,\mathrm{d}\boldsymbol{\phi}_{k,n+1}^{(j)}\cdots\mathrm{d}\boldsymbol{\phi}_{k,N_k^{(j)}}^{(j)} + \int \eta_k^{(j)}\mathbf{I}_M\xi(\eta_k^{(j)})\,\mathrm{d}\eta_k^{(j)} \\
&= r_{k,n}^{(j)}\gamma_{k,n}^{(j)}\int \mathbf{h}_{k,n}^{(j)}\mathbf{h}_{k,n}^{(j)\mathrm{H}}\beta(\mathbf{x}_k)\,\mathrm{d}\mathbf{x}_k + \sum_{\substack{n'=1\\ n'\neq n}}^{N_k^{(j)}} \sum_{r_{k,n'}^{(j)}\in\{0,1\}} \int r_{k,n'}^{(j)}\gamma_{k,n'}^{(j)}\mathbf{h}_{k,n'}^{(j)}\mathbf{h}_{k,n'}^{(j)\mathrm{H}}\alpha(\mathbf{y}_{k,n'}^{(j)})\beta(\mathbf{x}_k)\,\mathrm{d}\boldsymbol{\phi}_{k,n'}^{(j)}\,\mathrm{d}\mathbf{x}_k \\
&\quad + \int \eta_k^{(j)}\mathbf{I}_M\xi(\eta_k^{(j)})\,\mathrm{d}\eta_k^{(j)} \\
&= r_{k,n}^{(j)}\gamma_{k,n}^{(j)}\int \mathbf{h}_{k,n}^{(j)}\mathbf{h}_{k,n}^{(j)\mathrm{H}}\beta(\mathbf{x}_k)\,\mathrm{d}\mathbf{x}_k + \sum_{\substack{n'=1\\ n'\neq n}}^{N_k^{(j)}} \int \gamma_{k,n'}^{(j)}\mathbf{h}_{k,n'}^{(j)}\mathbf{h}_{k,n'}^{(j)\mathrm{H}}\alpha(\boldsymbol{\phi}_{k,n'}^{(j)},1)\beta(\mathbf{x}_k)\,\mathrm{d}\boldsymbol{\phi}_{k,n'}^{(j)}\,\mathrm{d}\mathbf{x}_k \\
&\quad + \int \eta_k^{(j)}\mathbf{I}_M\xi(\eta_k^{(j)})\,\mathrm{d}\eta_k^{(j)} \\
&= r_{k,n}^{(j)}\mathbf{C}_{2,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j)}) + \sum_{\substack{n'=1\\ n'\neq n}}^{N_k^{(j)}} \mathbf{C}_{3,k,n'}^{(j)} + \eta_{\xi,k}^{(j)}\mathbf{I}_M. \qquad (6)
\end{aligned}$$

which is the same as [1, (21)-(23)]. The matrices $\mathbf{C}_{\iota,k}^{(j)}(\mathbf{x}_k)$ and $\mathbf{C}_{\nu,k}^{(j)}$ can be derived in a similar way.

3 Monte Carlo Integration of $\mathbf{C}_{1,k,n}^{(j)}(\mathbf{x}_k)$, $\mathbf{C}_{2,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j)})$, and $\mathbf{C}_{3,k,n}^{(j)}$

In this section, we discuss how the “stacking technique” from [1, (32)-(34)] is used within our direct SLAM framework. We start from the“non-stacked” Monte Carlo integration of $\mathbf{C}_{3,k,n}^{(j)}$

$$\mathbf{C}_{3,k,n}^{(j)} \approx \sum_{p=1}^{P}\sum_{p'=1}^{P} w_{\beta,k}^{(p)} w_{\alpha,k,n}^{(j,p')}\mathbf{H}_{k,n}(\mathbf{x}_k^{(p)},\boldsymbol{\phi}_{k,n}^{(j,p')}). \qquad (7)$$

This "non-stacked" version has complexity $\mathcal{O}(P^2)$. To avoid the square complexity, we stack the particles and only evaluate pairs $(p, p'), p = p'$. As a result,

$$\begin{aligned}\mathbf{C}_{3,k,n}^{(j)} &\approx \sum_{p=1}^{P} w_{\beta,k}^{(p)} \Big(\sum_{p'=1}^{P} w_{\alpha,k,n}^{(j,p')} \Big) \mathbf{H}_{k,n}(\mathbf{x}_k^{(p)}, \boldsymbol{\phi}_{k,n}^{(j,p)}) \\ &= \sum_{p=1}^{P} w_{\beta,k}^{(p)} \Big(\sum_{p'=1}^{P} w_{\alpha,k,n}^{(j,p')} \Big) \tilde{\mathbf{H}}_{k,n}^{(j,p)} = \tilde{\mathbf{C}}_{3,k,n}^{(j)}. \end{aligned} \tag{8}$$

Similarly, $\mathbf{C}_{1,k,n}^{(j)}(\mathbf{x}_k)$ and $\mathbf{C}_{2,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j)})$ are approximated as

$$\begin{aligned}\mathbf{C}_{1,k,n}^{(j)}(\mathbf{x}_n^{(p)}) &\approx \sum_{p'=1}^{P} w_{\alpha,k,n}^{(j,p')} \mathbf{H}_{k,n}(\mathbf{x}_k^{(p)}, \boldsymbol{\phi}_{k,n}^{(j,p')}) \\ &\approx \Big(\sum_{p'=1}^{P} w_{\alpha,k,n}^{(j,p')} \Big) \mathbf{H}_{k,n}(\mathbf{x}_k^{(p)}, \boldsymbol{\phi}_{k,n}^{(j,p)}) \\ &= \Big(\sum_{p'=1}^{P} w_{\alpha,k,n}^{(j,p')} \Big) \tilde{\mathbf{H}}_{k,n}^{(j,p)} = \tilde{\mathbf{C}}_{1,k,n}^{(j)}(\mathbf{x}_n^{(p)}) \end{aligned} \tag{9}$$

$$\begin{aligned}\mathbf{C}_{2,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j,p)}) &\approx \sum_{p'=1}^{P} w_{\beta,k}^{(j,p')} \mathbf{H}_{k,n}(\mathbf{x}_k^{(p')}, \boldsymbol{\phi}_{k,n}^{(j,p)}) \\ &\approx \Big(\sum_{p'=1}^{P} w_{\beta,k}^{(j,p')} \Big) \mathbf{H}_{k,n}(\mathbf{x}_k^{(p)}, \boldsymbol{\phi}_{k,n}^{(j,p)}) \\ &= \mathbf{H}_{k,n}(\mathbf{x}_k^{(p)}, \boldsymbol{\phi}_{k,n}^{(j,p)}) = \tilde{\mathbf{H}}_{k,n}^{(j,p)} = \tilde{\mathbf{C}}_{2,k,n}^{(j)}(\boldsymbol{\phi}_{k,n}^{(j,p)}). \end{aligned} \tag{10}$$

4 Alternative Approximation of Measurement Update Messages

As discussed in [1, Sec. III-C], the measurement update messages $\iota(\mathbf{x}_k; \mathbf{z}_k^{(j)})$, $\kappa(\mathbf{y}_{k,n}^{(j)}; \mathbf{z}_k^{(j)})$, and $\nu(\eta_k^{(j)}; \mathbf{z}_k^{(j)})$ are approximated by Gaussian density functions with respect to $\mathbf{z}_k^{(j)}$, and the covariance matrices of the Gaussian PDFs are computed using moment matching. Since evaluating Gaussian PDFs requires the inverse of the covariance matrices, the computational complexity scales cubically with the number of measurements M. In this section, we provide an alternative approximation that reduces the complexity.

The main idea used for the derivation of alternative measurement update messages is to use the outer product of the mean vector to approximate the correlation (covariance) matrix. Specifically, we now compute an alternative approximation of [1, (26)] based on the outer product $\mathbf{C}_{3,k,n}^{(j)} = \boldsymbol{\mu}_{3,k,n}^{(j)} \boldsymbol{\mu}_{3,k,n}^{(j)\mathrm{H}}$, where

$$\boldsymbol{\mu}_{3,k,n}^{(j)} = \int \sqrt{\gamma_{k,n}^{(j)}} \mathbf{h}_{k,n}^{(j)} \alpha(\boldsymbol{\phi}_{k,n}^{(j)}, 1) \beta(\mathbf{x}_k) \mathrm{d}\boldsymbol{\phi}_{k,n}^{(j)} \mathrm{d}\mathbf{x}_k.$$

A particle-based computation of this outer product [1, (34)] is then computed as $\tilde{\mathbf{C}}_{3,k,n}^{(j)} = \tilde{\boldsymbol{\mu}}_{3,k,n}^{(j)} \tilde{\boldsymbol{\mu}}_{3,k,n}^{(j)\mathrm{H}}$ with

$$\tilde{\boldsymbol{\mu}}_{3,k,n}^{(j)} = \sum_{p=1}^{P} w_{\beta,k}^{(p)} \Big(\sum_{p'=1}^{P} w_{\alpha,k,n}^{(j,p')} \Big) \sqrt{\gamma_{k,n}^{(j,p)}} \mathbf{h}_{k,n}^{(j,p)}.$$

Based on this approximation, the matrix $\tilde{\mathbf{C}}^{(j)}_{\kappa,k,n}\big(\mathbf{y}^{(j,p)}_{k,n}\big)$ now reads

$$\begin{aligned}
\tilde{\mathbf{C}}^{(j)}_{\kappa,k,n}\big(\mathbf{y}^{(j,p)}_{k,n}\big) &= r^{(j)}_{k,n}\gamma^{(j,p)}_{k,n}\mathbf{h}^{(j,p)}_{k,n}\mathbf{h}^{(j,p)\mathrm{H}}_{k,n} + \sum_{\substack{n'=1\\ n'\neq n}}^{N_k^{(j)}} \tilde{\boldsymbol{\mu}}^{(j)}_{3,k,n'}\tilde{\boldsymbol{\mu}}^{(j)\mathrm{H}}_{3,k,n'} + \tilde{\eta}^{(j)}_{\xi,k}\mathbf{I}_M \\
&= r^{(j)}_{k,n}\gamma^{(j,p)}_{k,n}\mathbf{h}^{(j,p)}_{k,n}\mathbf{h}^{(j,p)\mathrm{H}}_{k,n} + \mathbf{B}^{(j)}_{k,n}\mathbf{B}^{(j)\mathrm{H}}_{k,n} + \tilde{\eta}^{(j)}_{\xi,k}\mathbf{I}_M \\
&= r^{(j)}_{k,n}\gamma^{(j,p)}_{k,n}\mathbf{h}^{(j,p)}_{k,n}\mathbf{h}^{(j,p)\mathrm{H}}_{k,n} + \mathbf{A}^{(j)}_{k,n} \qquad (11)
\end{aligned}$$

with $\mathbf{B}^{(j)}_{k,n} = [\tilde{\boldsymbol{\mu}}^{(j)}_{3,k,1}\cdots\tilde{\boldsymbol{\mu}}^{(j)}_{3,k,n-1}\;\tilde{\boldsymbol{\mu}}^{(j)}_{3,k,n+1}\cdots\tilde{\boldsymbol{\mu}}^{(j)}_{3,k,N_k^{(j)}}] \in \mathbb{C}^{M\times(N_k^{(j)}-1)}$ and $\mathbf{A}^{(j)}_{k,n} = \mathbf{B}^{(j)}_{k,n}\mathbf{B}^{(j)\mathrm{H}}_{k,n} + \tilde{\eta}^{(j)}_{\xi,k}\mathbf{I}_M$. To evaluate $\mathcal{CN}\Big(\mathbf{z}^{(j)}_k; \mathbf{0}, \tilde{\mathbf{C}}^{(j)}_{\kappa,k,n}\big(\mathbf{y}^{(j,p)}_{k,n}\big)\Big)$ for $\tilde{w}^{(j,p)}_{\mathbf{y},k,n}$ [1, (37)], one needs to compute the quantities $\det\Big(\tilde{\mathbf{C}}^{(j)}_{\kappa,k,n}\big(\mathbf{y}^{(j,p)}_{k,n}\big)\Big)$ and $\mathbf{z}^{(j)\mathrm{H}}_k\Big(\tilde{\mathbf{C}}^{(j)}_{\kappa,k,n}\big(\mathbf{y}^{(j,p)}_{k,n}\big)\Big)^{-1}\mathbf{z}^{(j)}_k$, which can be computed efficiently using the matrix determinant lemma and matrix inversion lemma, respectively, as

$$\begin{aligned}
\det\Big(\tilde{\mathbf{C}}^{(j)}_{\kappa,k,n}\big(\mathbf{y}^{(j,p)}_{k,n}\big)\Big) &= \det\Big(r^{(j)}_{k,n}\gamma^{(j,p)}_{k,n}\mathbf{h}^{(j,p)}_{k,n}\mathbf{h}^{(j,p)\mathrm{H}}_{k,n} + \mathbf{A}^{(j)}_{k,n}\Big) \\
&= \det\Big(1 + r^{(j)}_{k,n}\gamma^{(j,p)}_{k,n}\mathbf{h}^{(j,p)\mathrm{H}}_{k,n}\big(\mathbf{A}^{(j)}_{k,n}\big)^{-1}\mathbf{h}^{(j,p)}_{k,n}\Big)\det\big(\mathbf{A}^{(j)}_{k,n}\big) \qquad (12)
\end{aligned}$$

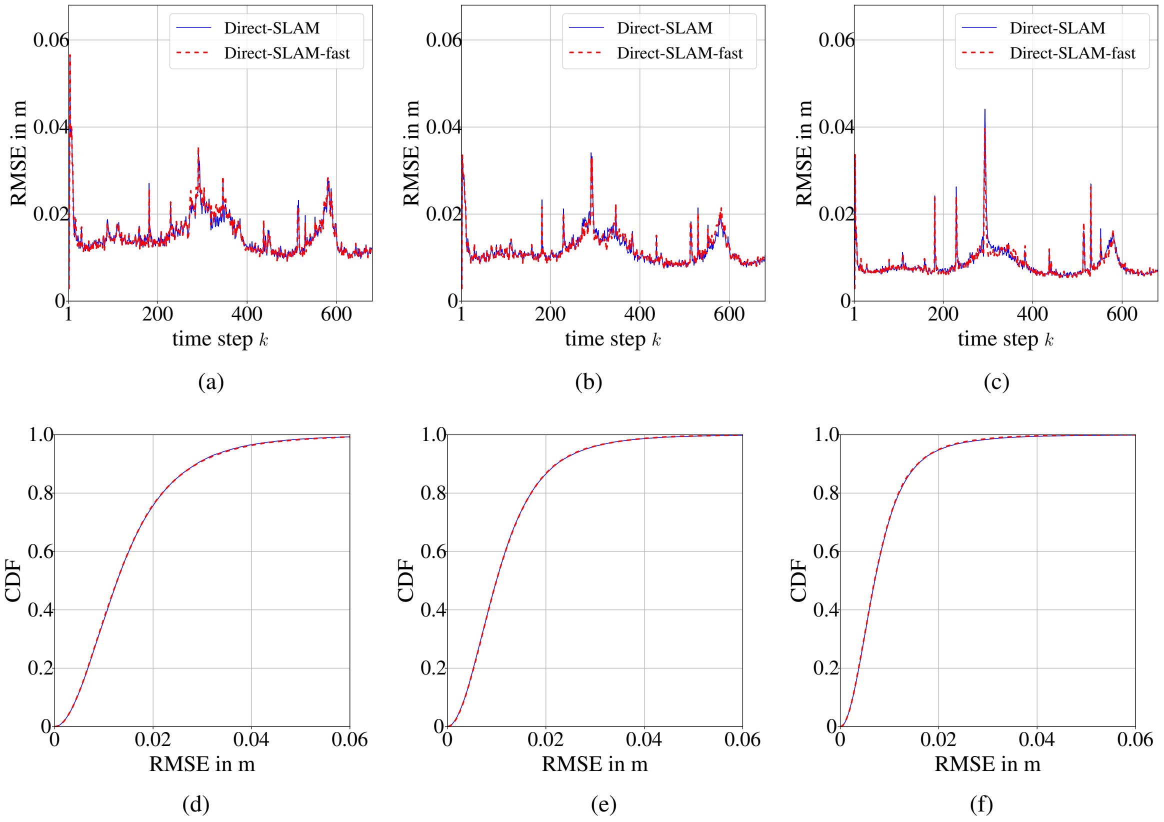
$$\begin{aligned}
\mathbf{z}^{(j)\mathrm{H}}_k\Big(\tilde{\mathbf{C}}^{(j)}_{\kappa,k,n}\big(\mathbf{y}^{(j,p)}_{k,n}\big)\Big)^{-1}\mathbf{z}^{(j)}_k &= \mathbf{z}^{(j)\mathrm{H}}_k\Big(r^{(j)}_{k,n}\gamma^{(j,p)}_{k,n}\mathbf{h}^{(j,p)}_{k,n}\mathbf{h}^{(j,p)\mathrm{H}}_{k,n} + \mathbf{A}^{(j)}_{k,n}\Big)^{-1}\mathbf{z}^{(j)}_k \\
&= \mathbf{z}^{(j)\mathrm{H}}_k\big(\mathbf{A}^{(j)}_{k,n}\big)^{-1}\mathbf{z}^{(j)}_k - \frac{r^{(j)}_{k,n}\gamma^{(j,p)}_{k,n}\big|\mathbf{h}^{(j,p)\mathrm{H}}_{k,n}\big(\mathbf{A}^{(j)}_{k,n}\big)^{-1}\mathbf{z}^{(j)}_k\big|^2}{1 + r^{(j)}_{k,n}\gamma^{(j,p)}_{k,n}\mathbf{h}^{(j,p)\mathrm{H}}_{k,n}\big(\mathbf{A}^{(j)}_{k,n}\big)^{-1}\mathbf{h}^{(j,p)}_{k,n}}. \qquad (13)
\end{aligned}$$

Next, $\big(\mathbf{A}^{(j)}_{k,n}\big)^{-1}$ can be further factorized again using the matrix inversion lemma

$$\begin{aligned}
\big(\mathbf{A}^{(j)}_{k,n}\big)^{-1} &= \Big(\mathbf{B}^{(j)}_{k,n}\mathbf{B}^{(j)\mathrm{H}}_{k,n} + \tilde{\eta}^{(j)}_{\xi,k}\mathbf{I}_M\Big)^{-1} \\
&= \big(\tilde{\eta}^{(j)}_{\xi,k}\big)^{-1}\mathbf{I}_M - \big(\tilde{\eta}^{(j)}_{\xi,k}\big)^{-1}\mathbf{B}^{(j)}_{k,n}\Big(\mathbf{B}^{(j)\mathrm{H}}_{k,n}\mathbf{B}^{(j)}_{k,n} + \tilde{\eta}^{(j)}_{\xi,k}\mathbf{I}_{N_k^{(j)}-1}\Big)^{-1}\mathbf{B}^{(j)\mathrm{H}}_{k,n} \qquad (14)
\end{aligned}$$

with which the quantities $\mathbf{z}^{(j)\mathrm{H}}_k\big(\mathbf{A}^{(j)}_{k,n}\big)^{-1}\mathbf{z}^{(j)}_k$, $\mathbf{h}^{(j,p)\mathrm{H}}_{k,n}\big(\mathbf{A}^{(j)}_{k,n}\big)^{-1}\mathbf{z}^{(j)}_k$, and $\mathbf{h}^{(j,p)\mathrm{H}}_{k,n}\big(\mathbf{A}^{(j)}_{k,n}\big)^{-1}\mathbf{h}^{(j,p)}_{k,n}$ can be computed in $\mathcal{O}\big(N_k^{(j)}M + (N_k^{(j)})^3\big)$ instead of in $\mathcal{O}(M^3)$. Note that since $\det\big(\mathbf{A}^{(j)}_{k,n}\big)$ appears in both the numerator and denominator of $\tilde{w}^{(j,p)}_{\mathbf{y},k,n}$ [1, (37)] and does thus not need to be computed. The same method can be applied to the computation of $\tilde{w}^{(p)}_{\mathbf{x},k}$ and $\tilde{w}^{(j,p')}_{\eta,k}$ [1, (36), (38)]. Including the operations of all particles and potential featuress (PFs), as well as the initialization of new PFs, the total complexity is $\mathcal{O}\bigg(M^2 + P\Big(\sum_{j=1}^{J}\big(N_k^{(j)}\big)^2 M + \sum_{j=1}^{J}\big(N_k^{(j)}\big)^4\Big)\bigg)$ which scales quadraticly with the number of measurements M. The number of PFs $N_k^{(j)}$ is typically smaller than ten and having five features is already sufficient to accurately localize the agent. Hence, this alternative approximation has time complexity advantages over the original one in high-bandwidth cases.

Next we repeat the experiments in [1, Sec. V-B] to show the efficacy of the alternative approximation that we refer to as "Direct-SLAM-fast". All the parameter settings remain the same. The results are shown in Figure 1, from which we can see that Direct-SLAM-fast has almost identical performance compared with the original Direct-SLAM. The averaged runtime shown in Table 1 verifies that Direct-SLAM-fast achieves lower computational complexity in the considered bandwidths.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 1: Performance of agent localization on synthetic data: The first row shows the agent position RMSEs averaged over 100 simulation runs with (a) 300 MHz, (b) 400 MHz, and (c) 600 MHz bandwidth, respectively. The second row shows the empirical CDFs of the RMSEs with (d) 300 MHz, (e) 400 MHz, and (f) 600 MHz bandwidth, respectively.

Table 1: Average runtime of different methods per time step

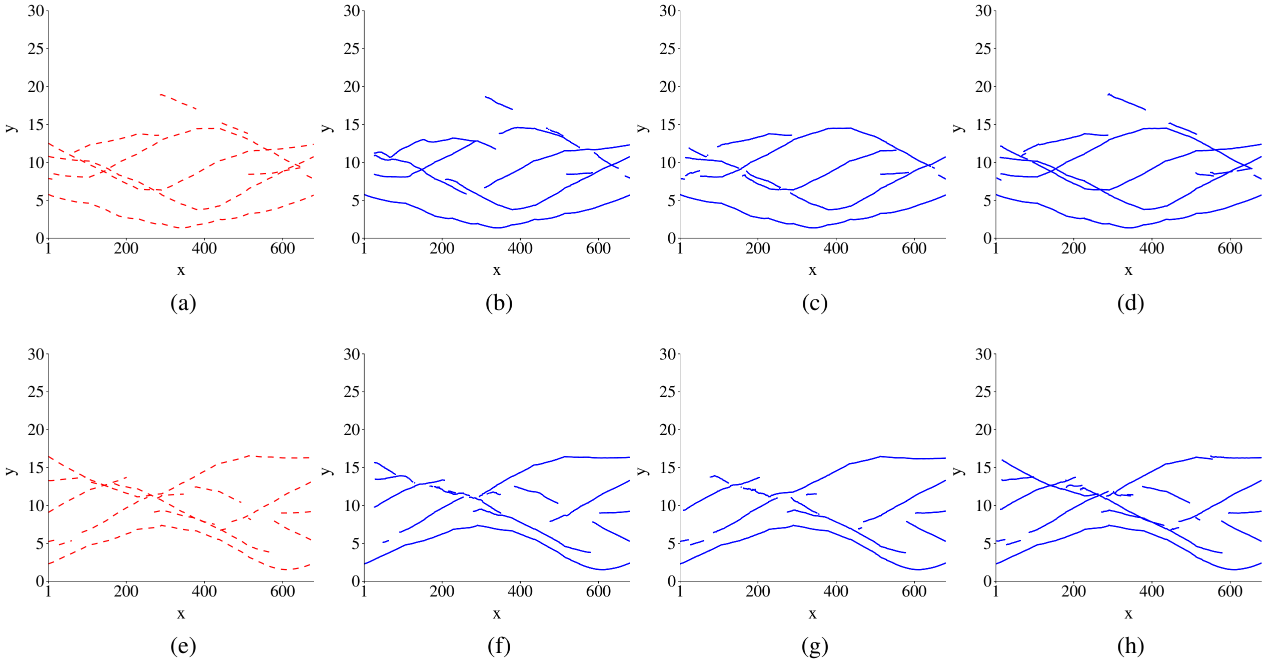
Method	Time (s)		
	300MHz	400MHz	600MHz
Direct-SLAM	1.19	1.53	2.58
Direct-SLAM-fast	0.74	0.88	1.25

5 Further Numerical Results

5.1 Range Measurements and PF Estimations

In this section, we plot the average distance between the particles of the PF positions and the true agent positions for one exemplary simulation run with 400 MHz in Figure 2b-2d and 2f-2h. Since BP-SLAM and BP-SLAM-AI rely on a channel estimator that cannot resolve multipath components (MPCs) having similar path lengths, both BP-SLAM and BP-SLAM-AI have problems when features “cross” or have similar distances with respect to the agent position. Conversely, Direct-SLAM directly operates on the radio signal and does consider the superposition of MPCs in the statistical model. For this reason, Direct-SLAM alleviates performance issues in situations with complicated propagation environments.

The poor mapping performance of BP-SLAM and BP-SLAM-AI is closely related the inaccuracy of the channel estimator used as a preprocessing stage. The range measurements provided by the state-of-the-art sparse Bayesian learning channel parameter estimator are shown in Figure 3. The range measurements are of



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 2: Visualization of the distance between the agent position and the true/estimated feature position. The first column with (a)/(e) shows the true distances for PA 1 and PA 2, respectively. The following columns show the averaged distance between the true agent position and the particles of PF positions obtained using (b)/(f) BP-SLAM, (c)/(g) BP-SLAM-AI, and (d)/(h) Direct SLAM with 400 MHz bandwidth and for PA 1 and PA 2, respectively.

low quality during time steps $k \in [100, 400]$ and around $k = 600$ when different MPCs result in similar delays. In addition, as can also be seen in Figure 3, this inaccuracy exacerbates as the signal bandwidth decreases.

5.2 Qualitative Synthetic Data Results

Here we provide the qualitative results of Direct-SLAM on synthetic data as discussed in [1, Sec. V-B]. Figure 4 shows the result of a single simulation run at different time steps. It can be seen that the particles of PFs gradually converge to the true feature positions as the agent moves. We have also shown the magnitude spectrum $|\tilde{z}_{k,m}^{(j)}|$ of synthetic data in Figure 5. $\tilde{z}_{k,m}^{(j)} = \mathbf{h}^{\mathrm{H}}(\tau_m)\mathbf{z}_k^{(j)}$ with $\tau_m = (m-1)T_s$ is the match filtering output which represents the 1-D spectrum over the delay grid $[\tau_1 \cdots \tau_M]^{\mathrm{T}}$ as introduced in [1, Sec. V-A].

5.3 Qualitative Real Data Results

The real radio measurements have been collected in a room with the floorplan as shown in [1, Fig. 4], and the two PAs are in the same positions as well. The mobile agent moves following a track composed of 679 consecutive positions shown as yellow line in [1, Fig. 4], yet the accurate position of the agent at each time step is not available. The radio signals are measured with a vector network analyzer (VNA) over a bandwidth of $B = 2\,\mathrm{GHz}$ at a center frequency of $f_{\mathrm{c}} = 3\,\mathrm{GHz}$. The dipole-like ultra-wideband transmit antenna mounted at the mobile agent and the dipole-like ultra-wideband receiver antennas mounted at two static PAs have a uniform radiation pattern in the azimuth plane and nulls in the ground and ceiling directions. Using $S(f)$, the measured radio signals are filtered to a bandwidth of $B = 400\,\mathrm{MHz}$ and demodulated to received baseband signals $\mathbf{z}_k^{(j)}, j \in \{1, 2\}$ with $M = 41$. The magnitude spectrum $|\tilde{z}_{k,m}^{(j)}|, j \in \{1, 2\}$ of the real measurements used by the Direct-SLAM algorithm is shown in Figure 6a-6b. Compared to simulated radio signals shown

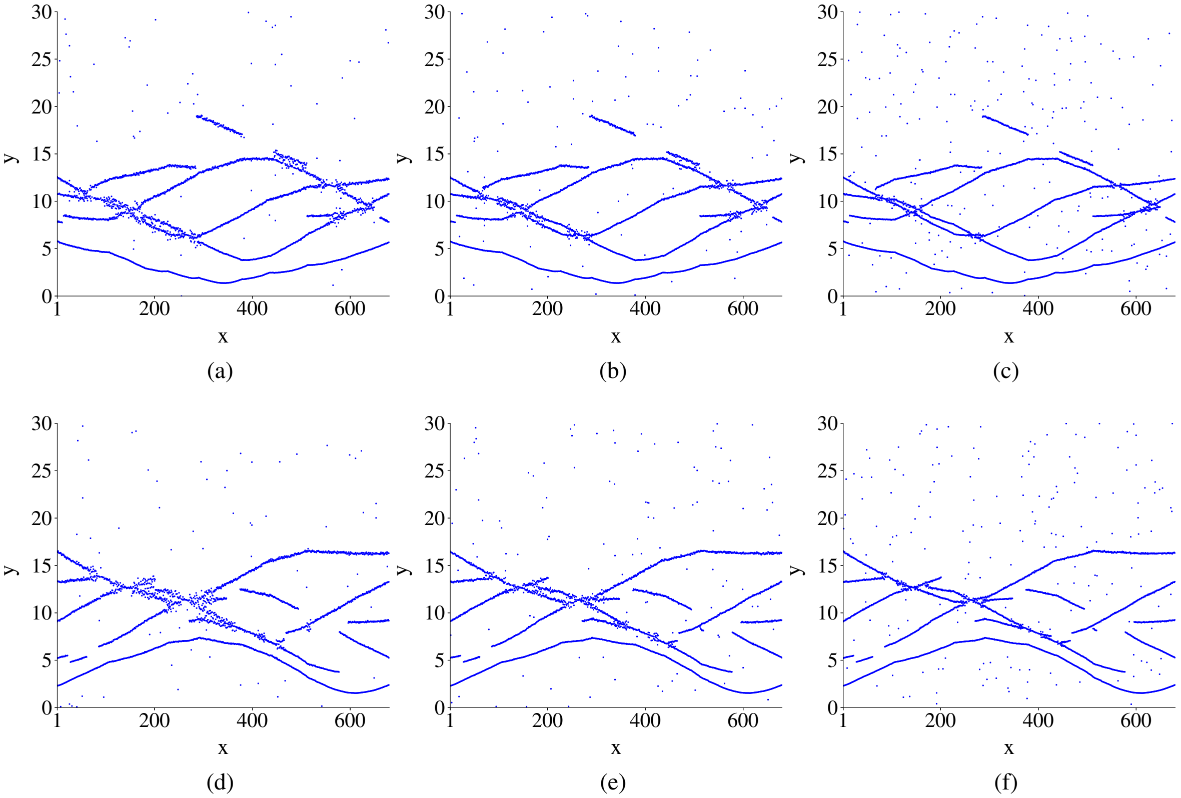


Figure 3: Visualization of the range measurements (blue dots) provided by the sparse Bayesian learning estimator for (a) PA 1 and (b) PA 2 with bandwidth 300 MHz. (c)\(d) and (e)\(f) are the counterparts of (a)\(b) for bandwidth 400 MHz and 600 MHz, respectively.

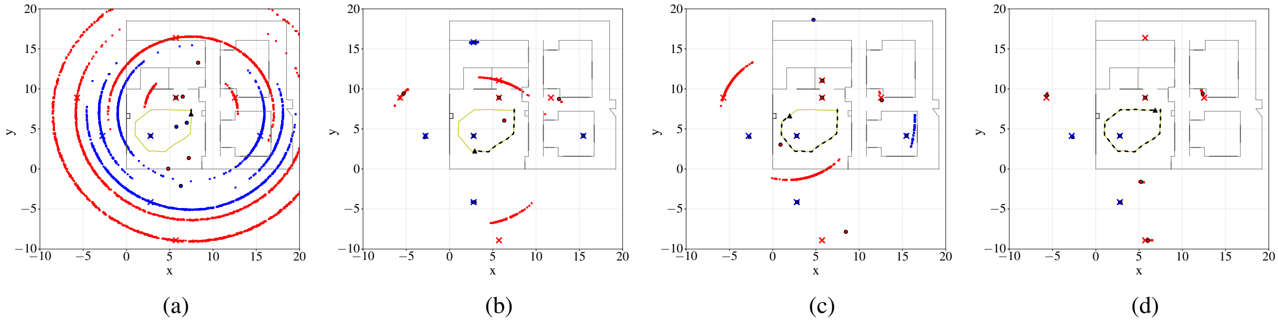
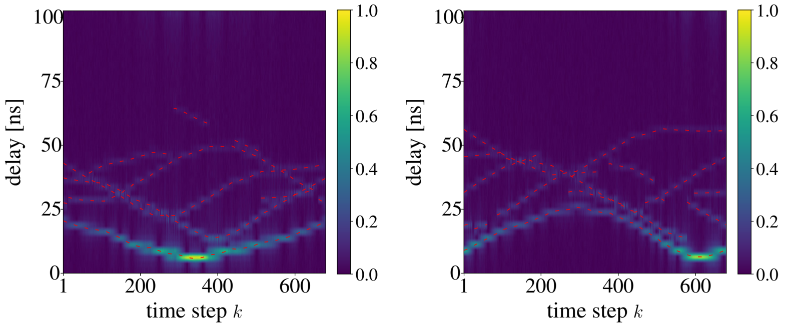


Figure 4: Example of a single simulation run using synthetic data with 400MHz bandwidth at times (a) $k = 18$, (b) $k = 282$, (c) $k = 476$, and (d) $k = 655$. Blue and red indicate features associated with PA 1 and PA 2, respectively. Crosses indicate the true positions of features. Bullets indicate the position particles of declared PFs, and circles with black edges are estimated positions of declared PFs. The yellow line is the true agent trajectory, the black dashed line is the estimated agent trajectory, and the black triangle is the agent position at the current time step.

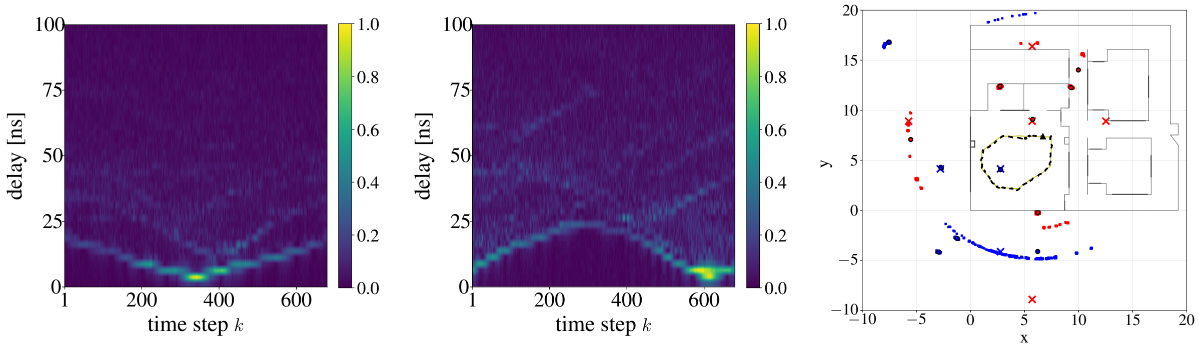
in [1, Fig. 5], it can be observed that the real measurements contain more measurement noise and non-specular multipath effects, including dense multipath component (DMC).

For the experiments running on real data, we set the driving noise variance for legacy PF positions $\sigma^2_{\mathbf{p},n}$ to $2.5 \times 10^{-5}\,\mathrm{m}^2$ for PAs, and to $6.4 \times 10^{-5}\,\mathrm{m}^2$ for other virtual anchors (VAs). Increasing driving noise



(a) (b)

Figure 5: The magnitude $|\tilde{z}_{k,m}^{(j)}|$ of synthetic data with 400MHz bandwidth normalized to $[0, 1]$ from PA (a) $j = 1$ and (b) $j = 2$. The red dashed lines are the delay between the agent and ground truth features (PAs plus VAs). VA positions are provided by ray tracing.



(a) (b) (c)

Figure 6: The magnitude specturm $|\tilde{z}_{k,m}^{(j)}|$ of real data with 400MHz bandwidth and normalized to $[0, 1]$ of PA (a) $j = 1$ and (b) $j = 2$.

variance improves robustness in this scenario. All other parameters remain the same as in [1, Sec. V-A]. Since there are no ground truth agent positions, we present a qualitative result in Figure 6c. It can be seen that some estimated PFs do not relate to any specular path. These result from the fact that real measurements include non-specular MPCs and there is a model mismatch. Nonetheless, the proposed Direct-SLAM algorithm is still able to provide reasonable estimates for the agent position and the features in the environment.

References

[1] M. Liang, E. Leitinger, and F. Meyer, “Direct multipath-based SLAM,” *IEEE Trans. Signal Process.*, 2024, to appear.