A Probabilistic Approach to Pose Synchronization for Multi-Reference Alignment with Applications to MIMO Wireless Communication Systems

Rob Romijnders
University of Amsterdam

Gabriele Cesa
Qualcomm AI Research

Christos Louizos
Qualcomm AI Research

Kumar Pratik
Qualcomm AI Research

Arash Behboodi
Qualcomm AI Research

Abstract

From molecular imaging to wireless communications, the ability to align and reconstruct signals from multiple misaligned observations is crucial for system performance. We study the problem of multi-reference alignment (MRA), which arises in many real world problems, such as cryo-EM, computer vision and, in particular, wireless communication systems. Using a probabilistic approach to model MRA, we find a new algorithm that uses relative poses as nuisance variables to marginalize out – thereby removing the global symmetries of the problem and allowing for more direct solutions and improved convergence. The decentralization of this approach enables significant computational savings by avoiding the cubic scaling of centralized methods through cycle consistency. Both proposed algorithms achieve lower reconstruction error across experimental settings.

1 Introduction

Multi-reference alignment (MRA) is a fundamental problem in signal processing and statistical inference that arises when reconstructing a signal from multiple noisy observations, and each observation has been transformed by an unknown group action such as rotation, translation, or other geometric transformation. This problem appears in diverse applications across scientific domains, from cryo-electron microscopy (Cryo-EM) [11, 9] and cryo-electron tomography (Cryo-ET) for molecular structure determination [17, 16, 14], to computer vision applications like Structure-from-Motion (SfM) and Simultaneous Localization and Mapping (SLAM), and even narrow-band 5G NR [13].

A core challenge in MRA lies in the fact that unknown transformations of the signal create an ambiguity. The signal could be recovered if the poses were known, and the unknown poses could be recovered when the signal was known. Moreover, the original signal can only be recovered up to a global transformation and, likewise, the observations' misalignments (i.e., their absolute poses) are only defined up to a joint transformation. In other words, any global transformation applied to all observations' poses yields an equally valid solution. Traditional approaches to MRA typically treat the absolute poses of the observations as nuisance variables to be estimated or marginalized over in an intermediate step before signal reconstruction. This includes various probabilistic approaches based on variants of the expectation-maximization algorithm [14] or other message-passing algorithms [12]. However, these methods can face challenges due to the inherent global symmetry of the problem.

Indeed, any disjoint estimate of the absolute poses necessarily breaks the global symmetry of the problem, which can lead to optimization difficulties in expectation-maximization or maximization-maximization algorithms. The fundamental issue is that global poses are not uniquely defined.

Any global transformation of the poses leads to another equally valid assignment. This can create convergence issues or trap optimization algorithms in local minima, e.g., due to subsets of observations initially converging to different global alignments.

In this work, we propose a novel probabilistic approach to MRA that addresses these challenges by reformulating the problem in terms of relative poses rather than absolute ones. Our key insight is that relative poses, in contrast to absolute ones, are uniquely defined and do not suffer from the global symmetries that can cause optimization difficulties. However, not all possible assignments of relative poses correspond to a set of absolute ones. As we prove in Appendix E, relative poses that are cycle-consistent are in one-to-one correspondence with equivalence classes of absolute pose assignments. Hence, to ensure that the relative poses represent valid configurations, we incorporate a cycle consistency constraint into our probabilistic model, which enforces that walks along any cycle in the relative-poses graph lead to identity transformations.

As a concrete example of the multi-reference alignment problem, we consider the problem of channel estimation (CE) in narrow-band 5G NR wireless communication systems [1, 13]. In the multi-input multi-output (MIMO) setting, the 5G NR specification allows channels to be precoded individually. Still, signals in neighboring PRG blocks can be correlated (Orthogonal Frequency Modulation and Demodulation). Current receivers do not utilize this correlation, resulting in suboptimal performance. We are the first to combine the group-theoretic approaches from multi-reference alignment with the application of 5G decoding using statistical methods, demonstrating how our relative pose framework can be applied to real-world communication systems.

Finally, we propose different algorithms to efficiently perform inference on this graph and evaluate them on multiple synthetic datasets, showing significant improvements over traditional approaches that rely on absolute pose estimation. In summary, this paper makes the following key contributions:

- We present the first MIMO decoding framework that explicitly incorporates relative rotations, addressing a fundamental limitation of existing methods.
- We develop both a direct method and an iterative refinement algorithm for joint denoising and synchronization, grounded in a principled probabilistic model, also known as a structured probabilistic graphical model.
- We provide theoretical insights and empirical evidence of the advantages of our approach in 5G-like MIMO scenarios.

The remainder of this paper is organized as follows: Section 2 formulates the problem, introduces our probabilistic graphical model, and reviews related work on MIMO synchronization and pose estimation. Section 3.2 details the proposed algorithms. Section 4 presents experimental results, and Section 5 concludes the paper with directions for future research.

2   Problem formulation

Background on Narrow-band 5G NR MIMO system   The 5G NR communication system uses an orthogonal-frequency division multiplexing (OFDM) scheme to communicate messages along orthogonal subcarrier signals at each time step. As a result, the communication channel is divided into a time-frequency grid, with each cell (or resource element, RE) carrying a single symbol. In a multiple-input-multiple-output (MIMO) scheme, both the transmitter and receiver have multiple antennas, which enables the transmission of multiple symbols on the same time-frequency location. Formally, assuming d transmitting and receiving antennas, the transmission at a frequency f and time step t of a symbol vector $x_{f,t} \in \mathbb{C}^d$ is

$$y_{f,t} = \boldsymbol{H}_{f,t}\boldsymbol{P}_{f,t}x_{f,t} + \varepsilon \tag{1}$$

where $H_{f,t} \in \mathbb{C}^{d \times d}$ is the MIMO channel matrix, $y \in \mathbb{C}^d$ the received signal, $\varepsilon \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^2 I)$ is Gaussian noise and $P \in U(n)$ is a unitary matrix which defines the *precoding*. The transmitter precodes the transmitted messages to help minimize interference and improve SNR, e.g., by aligning the message along the singular vectors of the channel matrix. For practical reasons, in narrow-band 5G NR, the precoding matrix is shared by multiple neighboring resource elements within the same physical resource block group (PRG). The challenge for the receiver is estimating the effective channel matrices (i.e., after precoding) from a few noisy observations obtained via some pilot messages. We consider a multi-reference alignment problem inspired by this narrow-band 5G

NR channel estimation problem. Our problem formulation allows us to abstract away from many technical details of the communication protocol, while preserving its relevant properties and the underlying synchronization challenges. For more details, we refer to Chen et al. [5], Pratik et al. [13] for a more precise description of the real channel estimation problem in narrow-band 5G NR.

Abstract formulation For ease of modelling, we model the channel matrix as real numbers and the precoding matrices as real orthogonal matrices $\boldsymbol{P} \in O(d) \subset \mathbb{R}^{d \times d}$. We denote the channel in the j-th PRG block by $\boldsymbol{H}_i \in \mathbb{R}^{D \times d}$, where the dimension D indexes all time-frequency positions within the block and the output antennas. In a realistic 5G MIMO scenario, the receiver observes the effective channels $\boldsymbol{H}_i^{'} = \boldsymbol{H}_i \boldsymbol{P}_i$ via some pilot messages only at certain time-frequency positions (demodulation reference signal, DMRS). This signal is then used to estimate the full channel via linear predictors. We assume the receiver observes the full channel[1] corrupted by real Gaussian noise, i.e. $\boldsymbol{B}_i = \boldsymbol{H}_i \boldsymbol{P}_i + \varepsilon_i \in \mathbb{R}^{D \times d}$ with $\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2 I)$.

Taken all random variables together, $\boldsymbol{H}$, $\boldsymbol{P}$, and $\boldsymbol{B}$, the joint probability distribution is:

$$p(\boldsymbol{B}, \boldsymbol{P}, \boldsymbol{H}) = p(\boldsymbol{B}|\boldsymbol{H}^{'}) \, p(\boldsymbol{H}^{'}|\boldsymbol{H}, \boldsymbol{P}) \, p(\boldsymbol{P}) \, p(\boldsymbol{H}) \tag{2}$$

where $p(\boldsymbol{B}|\boldsymbol{H}^{'}) = \mathcal{N}(\boldsymbol{B}|\boldsymbol{H}^{'}, \sigma^2 I)$, $p(\boldsymbol{H}^{'}|\boldsymbol{H}, \boldsymbol{P}) = \delta_{\boldsymbol{H}'-\boldsymbol{H}\boldsymbol{P}}$ is a deterministic factor applying the precoding to the channel. We assume the prior $p(\boldsymbol{P})$ on the rotation precoding matrices is uniform over the $O(d)$ group. Finally, $p(\boldsymbol{H})$ specifies a rotation invariant[2] prior via a matrix normal distribution over all underlying channel matrices $\mathcal{MN}(0, \boldsymbol{U}, \boldsymbol{I})$ with $\boldsymbol{I}$ the identity matrix and matrix $\boldsymbol{U}$ modelling the row-covariance. Note the rotation invariance: if $P \in O(d)$ and $\boldsymbol{X} \sim \mathcal{MN}(0, \boldsymbol{U}, \boldsymbol{I})$ then $\boldsymbol{X}\boldsymbol{P} \sim \mathcal{MN}(0, \boldsymbol{U}, \boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I})$. We also refer to this prior as 'spatial correlation,' which is shorthand for the correlation that occurs along the time and frequency axes among the PRG blocks due to Rayleigh scattering [18, 5]. This spatial correlation enables the estimation of the channel in a PRG block $\boldsymbol{H}_i^{'}$ from its noisy observations $\boldsymbol{B}_i$. However, the unknown precoding matrices prevent the receiver from leveraging the correlation between neighboring blocks. Our key insight is that by modeling the relative rotations $\boldsymbol{R}_{ij} = \boldsymbol{P}_i^{-1} \boldsymbol{P}_j$ between adjacent blocks in the time-frequency grid, we can leverage the spatial and temporal correlations to achieve more robust channel estimation.

Absolute to relative poses Multi-reference alignment problems typically present a global symmetry, which implies that absolute poses and the underlying signal are not uniquely defined. Indeed, the set of channels $\boldsymbol{H} = \{\boldsymbol{H}_i\}_i$ and precoding matrices $\boldsymbol{P} = \{\boldsymbol{P}_i\}_i$ generate the same observations $\boldsymbol{B} = \{\boldsymbol{B}_i\}_i$ as the channels $\tilde{\boldsymbol{H}} = \{\boldsymbol{H}_i Q\}_i$ and the precoding $\tilde{\boldsymbol{P}} = \{Q^{-1}\boldsymbol{P}_i\}_i$ for any transformation $Q \in O(d)$. Any attempt to estimate these quantities requires some form of symmetry breaking. However, this problem vanishes when taking a local viewpoint and, therefore, considering relative poses; see Appendix E.1 For this reason, we develop our method to recover relative poses $\boldsymbol{R} = \{\boldsymbol{R}_{i,j} := \boldsymbol{P}_i^{-1}\boldsymbol{P}_j\}_{i,j}$, which are uniquely defined and suitable for an efficient distributed and local computational scheme.

Relation to prior work in synchronization: synchronization over compact groups has been approached with approximate message passing methods [12] or semi-definite programming [3]. Other works studied CryoEM [14] and stochastic block models [2]. The fundamental difference in our work, though, is that we aim to estimate the signal itself. [12] assume noisy observation of the relative poses and strive to find an optimal representation of the absolute poses. In our case, we have noisy observations of the underlying signal at each location. The goal is to efficiently and effectively marginalize over the relative poses to obtain a better estimate of the channel. Although our approach utilizes similar estimations of the relative poses, it does not synchronize the absolute poses. Instead, it can be said informally that we 'directly synchronize' the underlying signal. [13] first describes the MRA of narrow-band 5G NR channel estimation, but adopted complex deep learning architectures. **Relation to wider related work** Synchronization problems have been studied in a more abstract context for their group-theoretic properties [7, 6]. Some works have studied pairwise relative pose estimation by candidate elimination [19], but we are the first structured statistical model to marginalize over the unknown poses to reconstruct the underlying signal effectively and efficiently.

---

[1]Note that pilot DMRS signals provide linear measurements of the channel, so our simplification reduces to the use of a diagonal full rank covariance matrix between $\boldsymbol{B}_i$ and $\boldsymbol{H}^{'}$, rather than a more generic one.

[2]This is a reasonable assumption since it is equivalent to assuming zero mean and that the correlation depends only on the frequency and time position and is the same among all the antennas, since the precoding transformation only acts on the antennas' dimension of the channels matrices.

3 Method

3.1 Direct estimation

A direct approach is to estimate the relative rotations $\boldsymbol{R}_{ij}$ between adjacent nodes $\boldsymbol{H}$ in the time-frequency grid. This is done by maximizing the likelihood of the data $\log p(\boldsymbol{B})$. Observations $\boldsymbol{B} \in \mathbb{R}^{D \times d}$ is the matrix of D times a d dimensional observation. However, the direct approach has two disadvantages: estimating in pairs does not enable global synchronization, and direct estimates can be suboptimal as rotation and observation noise are confounded, where estimating one changes the other. We address both properties in the next section.

Estimating the channel $\boldsymbol{H}^{'}$ can be done by writing the posterior given the data $\boldsymbol{B}$ directly:

$$\arg \max_{\boldsymbol{H}^{'}} p(\boldsymbol{H}^{'}|\boldsymbol{B}) = \int_{\boldsymbol{R}} p(\boldsymbol{H}^{'}|\boldsymbol{B}, \boldsymbol{R}) p(\boldsymbol{R}|\boldsymbol{B}) d\boldsymbol{R} \tag{3}$$

The posterior $p(\boldsymbol{R}|\boldsymbol{B})$ can be heuristically approximated with a delta distribution – meaning that we can take a point estimate of the rotation. Assuming only two blocks $\{\boldsymbol{H}_1, \boldsymbol{H}_2\}$, if the covariance matrix of the prior on the correlated channel is $\boldsymbol{U}$, then one can find the relative rotation $\boldsymbol{R}_{12}$ between the two blocks from the following projection: $\arg \max_{\boldsymbol{R}_{12}} \langle \boldsymbol{R}_{21}, \boldsymbol{H}_2^{'T} \boldsymbol{U}_a^T \boldsymbol{H}_1^{'} \rangle_F$.

Likewise, for a triplet of three nodes $\boldsymbol{H}_1, \boldsymbol{H}_2, \boldsymbol{H}_3$ and relative poses $\boldsymbol{R}_{12}$ and $\boldsymbol{R}_{13}$ (note that $\boldsymbol{R}_{23} := \boldsymbol{R}_{12}^T \boldsymbol{R}_{13}$ and that $\boldsymbol{R}_{ij} = \boldsymbol{R}_{ji}^T$), one can iterate the two updates below in Equation 4 until convergence. Matrices $\boldsymbol{U}_a, \boldsymbol{U}_b, \cdots$ are submatrices of the covariance matrix $\boldsymbol{U}$, as described with Equation 30 in the Appendix.

$$\begin{cases} \boldsymbol{R}_{13} = \arg \max \ \langle \boldsymbol{R}_{21}, \boldsymbol{B}_2^T \boldsymbol{U}_a^T \boldsymbol{B}_1 \rangle_F + \langle \boldsymbol{R}_{31}, \boldsymbol{B}_3^T \boldsymbol{U}_b^T \boldsymbol{B}_1 \rangle_F + \langle \boldsymbol{R}_{31}, \boldsymbol{B}_3^T \boldsymbol{U}_c^T \boldsymbol{B}_2 \boldsymbol{R}_{21} \rangle_F \\ \boldsymbol{R}_{12} = \arg \max \ \langle \boldsymbol{R}_{21}, \boldsymbol{B}_2^T \boldsymbol{U}_a^T \boldsymbol{B}_1 \rangle_F + \langle \boldsymbol{R}_{31}, \boldsymbol{B}_3^T \boldsymbol{U}_b^T \boldsymbol{B}_1 \rangle_F + \langle \boldsymbol{R}_{21}, \boldsymbol{B}_2^T \boldsymbol{U}_c \boldsymbol{B}_3 \boldsymbol{R}_{31} \rangle_F. \end{cases} \tag{4}$$

The derivation is outlined in Appendix C. Crucially, the above solution arises from using properties of the Matrix Normal distribution. Ending up with the above Frobenius inner-products is a result of the Orthogonal Procrustes problem [8, 15], which is a mean-square solution when optimizing over the set of unitary matrices. The projection is a closed-form maximizer of the Procrustes problem. Therefore, each step of the iteration is non-decreasing in the objective value, and we find that this usually converges in three to five steps.

Secondly, obtaining the channel estimate from the observations and rotation matrices $\arg \max p(\boldsymbol{H}^{'}|\boldsymbol{B}, \boldsymbol{R})$ is a linear problem, which can be solved in $O(D^3)$ time:

$$\begin{bmatrix} \boldsymbol{H}_1^{'} \\ \boldsymbol{H}_2^{'} \hat{\boldsymbol{R}}_{21} \\ \boldsymbol{H}_3^{'} \hat{\boldsymbol{R}}_{31} \end{bmatrix} = \left(\sigma_\epsilon^2 \boldsymbol{U}^{-1} + \boldsymbol{I}_{3D} \right)^{-1} \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \hat{\boldsymbol{R}}_{21} \\ \boldsymbol{B}_3 \hat{\boldsymbol{R}}_{31} \end{bmatrix} \tag{5}$$

The above solutions are worked out for a 'triplet' of nodes, meaning that we consider three nodes and derive the estimation equations. This could be worked out for an arbitrary number of nodes. However, this has two downsides: (a) the computation scales cubically in the number of measurements and thus cubically in the number of nodes. Hence, scaling this approach on the wide time-frequency lattices of up to hundreds blocks per time slot is impractical, especially under the resource and power constraints of modern 5G telecom modems [5]. The second downside (b) to including more nodes is that synchronization becomes 'harder.' After all, only noisy nodes are available, and harmonizing among more noisy observations pertaining to random rotations has more numerical issues.

Therefore, we suggest decentralizing the estimation into triplets. A triplet of nodes is the smallest subgraph in which a local synchronization can imply a global synchronization over the entire grid, see Appendix E. Meanwhile, a triplet of nodes is most economical with the local 'spatial correlation' that exists among the channels. Due to Rayleigh scattering [18], only nearby PRG blocks will have correlation, and denoising among local subgraphs most economically uses the compute resources.

3.2 Synchronization and denoising via iterative refinement

From the direct solution, we suggest an iterative refinement method that is loosely inspired by the Expectation Maximization algorithm. The direct solution alternates between two maximizations,
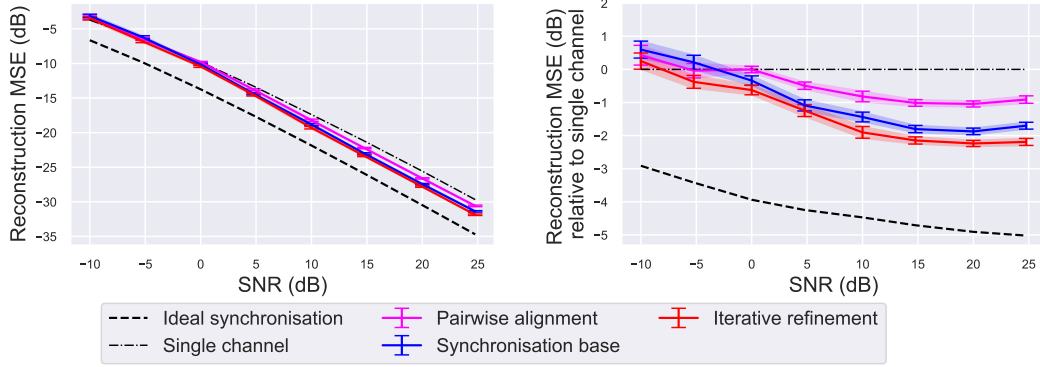
Figure 1: Iterative refinement achieves the lowest reconstruction error compared to signal denoising only. This is a synthetic 6-by-6 grid of 12 measurements each. Error bars indicate standard error among 25 random seeds. Additionally, the 'synchronization base' approach, by itself, achieves lower reconstruction error than prior art for reasonable signal-to-noise ratios.

over $p(\boldsymbol{H}'|\boldsymbol{B}, \boldsymbol{R})$ and over $p(\boldsymbol{R}|\boldsymbol{B})$. The latter maximization assumes that channel observations $\boldsymbol{B}$ are made and the relative poses can be estimated from them. By extension, $\boldsymbol{H}'$ will serve as a 'denoised version' of the channel. As such, we will report the Mean-Square error between $\boldsymbol{H}'$ and $\boldsymbol{B}$. Then, as $\boldsymbol{H}'$ is a denoised version of $\boldsymbol{B}$, we can refine the rotation estimate from $p(\boldsymbol{R}|\boldsymbol{H}')$ instead. Continuing, the relative poses can be found from $p(\boldsymbol{H}'|\boldsymbol{B}, \boldsymbol{R})$.

Loosely, this is inspired by the Expectation-Maximization algorithm [4, 10]. Analogous to the M-step, we estimate the parameters of the channel using an estimate for the rotation random variable. Analogous to the E-step, we find an estimate for the relative poses. In this case, however, we take only a point estimate, $q(\boldsymbol{R}) = \delta_{\boldsymbol{R}}$ in $\log p(\boldsymbol{B}) \geq \max_{q(\boldsymbol{R})} \mathbb{E}_{\boldsymbol{R} \sim q(\boldsymbol{R})}[\log p(\boldsymbol{B}, \boldsymbol{R}, \boldsymbol{H}'_t) - \log q(\boldsymbol{R})]$.

The estimation for the relative poses can be done using similar Frobenius projections, starting from $\arg\max_{\boldsymbol{R}} p(\boldsymbol{R}|\boldsymbol{H}')$. This time, however, the estimates for the channel are used:

$$\begin{cases} \boldsymbol{R}_{12} = \arg\max_{\boldsymbol{R}_{12}} \langle \boldsymbol{R}_{21}, \boldsymbol{H}'^T_{2,t}\boldsymbol{U}^T_a\boldsymbol{H}'_{1,t}\rangle_F + \langle \boldsymbol{R}_{31}, \boldsymbol{H}'^T_{3,t}\boldsymbol{U}^T_b\boldsymbol{H}'_{1,t}\rangle_F + \langle \boldsymbol{R}_{21}, \boldsymbol{H}'^T_{2,t}\boldsymbol{U}_c\boldsymbol{H}'_{3,t}\boldsymbol{R}_{31}\rangle_F \\ \boldsymbol{R}_{13} = \arg\max_{\boldsymbol{R}_{13}} \langle \boldsymbol{R}_{21}, \boldsymbol{H}'^T_{2,t}\boldsymbol{U}^T_a\boldsymbol{H}'_{1,t}\rangle_F + \langle \boldsymbol{R}_{31}, \boldsymbol{H}'^T_{3,t}\boldsymbol{U}^T_b\boldsymbol{H}'_{1,t}\rangle_F + \langle \boldsymbol{R}_{31}, \boldsymbol{H}'^T_{3,t}\boldsymbol{U}^T_c\boldsymbol{H}'_{2,t}\boldsymbol{R}_{21}\rangle_F \end{cases}$$
(6)

The channel estimation is derived in Appendix D and results in an estimator similar to Equation 5:

$$\begin{bmatrix} \boldsymbol{H}'_{1,t+1} \\ \boldsymbol{H}'_{2,t+1}\hat{\boldsymbol{R}}_{21} \\ \boldsymbol{H}'_{3,t+1}\hat{\boldsymbol{R}}_{31} \end{bmatrix} = \left(\sigma^2_\epsilon\boldsymbol{U}^{-1} + \boldsymbol{I}_{3D}\right)^{-1} \begin{bmatrix} \boldsymbol{H}'_{1,t} \\ \boldsymbol{H}'_{2,t}\hat{\boldsymbol{R}}_{21} \\ \boldsymbol{H}'_{3,t}\hat{\boldsymbol{R}}_{31} \end{bmatrix}$$
(7)

Together, Equations 6 and 7 can be repeated to refine the channel estimate and the rotation estimate. We typically find that this stabilizes after three or four iterations, which is illustrated in the experimental results section.

4 Experimental results

Our experimental evaluation demonstrates the advantages of our proposed approach. We compare three methods. Starting with the 'pairwise alignment' that operates on pairs of nodes in the time-frequency grid. It serves as the baseline, and, while computationally efficient, it suffers from the fundamental limitation that local synchronization in pairs does not guarantee global consistency across the entire network. Addressing this limitation, our 'synchronization base' solution operates on triplets of nodes, providing the crucial advantage that each local synchronization within a triplet inherently implies global synchronization (Appendix E. This triplet-based approach not only improves synchronization accuracy but also maintains computational tractability with $O(D^3)$ complexity.
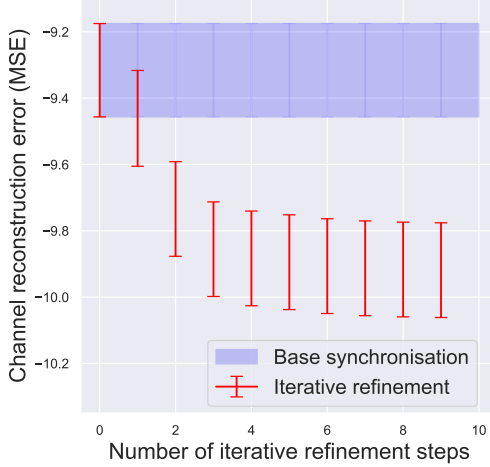
5

Figure 2: The reconstruction error for varying number of iterative refinement steps. Most improvement over the direct approach (blue shaded region) is achieved in two to five steps.
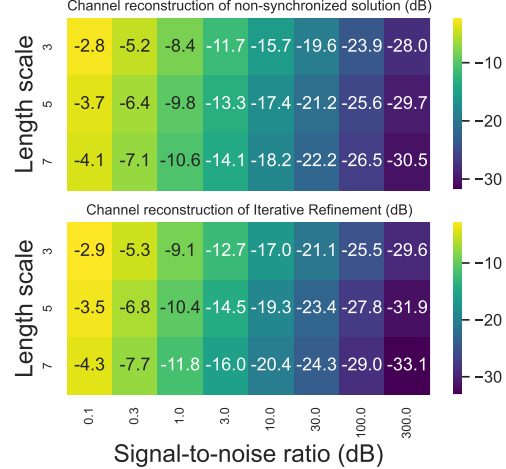

Figure 3: Numerical results along varying length scales, which determine the correlation properties. Our synchronization improves the channel reconstruction across these settings.

Building upon the triplet foundation, we then demonstrate the benefits of an 'Iterative Refinement' algorithm, which operates on the same triplet structure but leverages the denoised channel estimates H' rather than the noisy observations B. This final experiment demonstrates how our method enables more effective denoising strategies, resulting in superior overall performance in 5G-like scenarios. Each plot includes both a 'single channel' and an 'ideal line.' These are theoretical results for either not using synchronization at all or assuming perfect synchronization (explained in Appendix F).

For the experiments, we sample from the ground-truth graphical model and use each method to reconstruct the signal. The problem consists of $6 \times 6 = 36$ grids of PRG blocks, with $D = 3 \times 4 = 12$ measurements made in each block. This means that each block is 3 by 4 measurement cells. The spatial correlation has a length scale of five measurements and follows a squared-exponential decay, which is a simplified model for the Rayleigh-fading scatter [18]. Different values of the length scale indicate varying amounts of scatter in the environment and a comparison for multiple values of the length scale is shown in Figure 3.

Figure 1 shows the main experimental result. It compares all three methods on a 5G MIMO problem. In general, for larger Signal-to-Noise (SNR) ratios, the channel reconstruction achieves lower decibels, and this is reflected in the 'ideal synchronization' and 'single channel ideal' comparison lines. The right figure redraws each line compared to the 'single channel' MSE and indicates the relative improvement. Beyond 0 dB SNR, all three MRA methods exhibit improved performance, with synchronization achieving a lower MSE than pairwise alignment, and Iterative Refinement achieving the lowest MSE across experimental settings.

5 Conclusion

Our work presents a significant advancement in multi-reference alignment (MRA) by introducing a novel structured probabilistic framework that addresses the fundamental challenges of MIMO channel denoising. By reformulating the MRA problem in terms of relative poses within a principled statistical model, we eliminate the global redundancies that hinder optimization. We use cycle consistency and arrive at a computationally efficient method that can scale to realistic system sizes.

The iterative refinement algorithm consistently outperforms direct estimation methods and methods that use only pairwise alignment. Across experimental settings, our approach that is decentralized and refines relative pose and channel estimates achieves better reconstruction error consistently. The success of our approach demonstrates the promise of iterative refinement techniques for MIMO synchronization. It opens up new directions for developing more sophisticated denoising strategies that can leverage the global synchronization achieved through our triplet-based cycle consistency.

References

[1] 3GPP. Third generation partnership project (3gpp). *Technical Specification (TS)*, 38(10), 2020.

[2] Emmanuel Abbe. Community detection and stochastic block models: Recent developments. *Journal Machine Learning Research*, 2017.

[3] Afonso S Bandeira, Ben Blum-Smith, Joe Kileel, Amelia Perry, Jonathan Weed, and Alexander S Wein. Estimation under group actions: recovering orbits from invariants. *arXiv preprint arXiv:1712.10163*, 2017.

[4] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Springer, 2006.

[5] Wanshi Chen, Peter Gaal, Juan Montojo, and Haris Zisimopoulos. *Fundamentals of 5G communications: connectivity for enhanced mobile broadband and beyond*. McGraw-Hill Education, 2021.

[6] Yifeng Fan, Tingran Gao, and Zhizhen Jane Zhao. Unsupervised co-learning on g-manifolds across irreducible representations. *Advances in neural information processing systems*, 32, 2019.

[7] Tingran Gao, Jacek Brodzki, and Sayan Mukherjee. The geometry of synchronization problems and learning group actions. *Discrete & Computational Geometry*, 2021.

[8] John C Gower and Garmt B Dijksterhuis. *Procrustes problems*. Oxford university press, 2004.

[9] R. Henderson, J. M. Baldwin, T. A. Ceska, F. Zemlin, E. Beckmann, and K. H. Downing. Model for the structure of bacteriorhodopsin based on high-resolution electron cryo-microscopy. *Journal of Molecular Biology*, June 1990.

[10] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[11] Eva Nogales. The development of cryo-EM into a mainstream structural biology technique. *Nature methods*, 2016. Publisher: Nature Publishing Group.

[12] Amelia Perry, Alexander S Wein, Afonso S Bandeira, and Ankur Moitra. Message-passing algorithms for synchronization problems over compact groups. *Communications on Pure and Applied Mathematics*, 2018.

[13] Kumar Pratik, Pouriya Sadeghi, Gabriele Cesa, Sanaz Barghi, Joseph B Soriaga, Yuanning Yu, Supratik Bhattacharjee, and Arash Behboodi. Requestnet: A foundational learning model for channel estimation. *arXiv 2508.08790*, 2025.

[14] Sjors HW Scheres. Relion: implementation of a bayesian approach to cryo-em structure determination. *Journal of structural biology*, 2012.

[15] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 1966.

[16] Amit Singer. Mathematics for cryo-electron microscopy. In *Proceedings of the International Congress of Mathematicians*. World Scientific, 2018.

[17] Amit Singer and Fred J Sigworth. Computational methods for single-particle electron cryomicroscopy. *Annual Review of Biomedical Data Science*, 2020.

[18] Bernard Sklar. Rayleigh fading channels in mobile digital communication systems. i. characterization. *IEEE Communications magazine*, 2002.

[19] Yifan Sun and Qixing Huang. Pose synchronization under multiple pair-wise relative poses. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Acknowledgements

A   Notation

An overview of the notation in this paper. Additionally, Figure 4 presents a factor graph to illustrate how the random variables relate to each other. That factor graph is for three PRG blocks.

Notation

D, d	Number of measurements and number of dimensions
H, W	Number of rows and columns in the MIMO grid
$N = H \cdot W$	The number of blocks in the MIMO grid
$\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3, \ldots, \boldsymbol{P}_J$	Rotation matrix of size $\mathbb{R}^{d \times d}$
$p(\boldsymbol{P}_i)$	Prior distribution of $\boldsymbol{P}_i$, uniform over the rotation angles, $\boldsymbol{P}_i \in SO(d)$
$\boldsymbol{R}_{ij} = \boldsymbol{P}_i^{-1} \boldsymbol{P}_j$	Relative rotation matrix of size $\mathbb{R}^{d \times d}$
$\boldsymbol{H}_1, \boldsymbol{H}_2, \boldsymbol{H}_3, \cdots, \boldsymbol{H}_J$	Hidden matrices of size $\mathbb{R}^{D \times d}$
$\boldsymbol{H} = [\boldsymbol{H}_1; \boldsymbol{H}_2; \boldsymbol{H}_3]$	Stacked hidden matrices of size $\mathbb{R}^{3D \times d}$
$\boldsymbol{H}_i^{'} = \boldsymbol{H}_i \boldsymbol{P}_i$	H-accent is the channel matrix that has undergone an unknown rotation, size $\mathbb{R}^{D \times d}$
$\boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3, \cdots, \boldsymbol{B}_J$	Observed matrices of size $\mathbb{R}^{D \times d}$
$\boldsymbol{B} = [\boldsymbol{B}_1; \boldsymbol{B}_2; \boldsymbol{B}_3]$	Stacked observed matrices of size $\mathbb{R}^{3D \times d}$
$\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$	Observation noise, shape clear from context
$\boldsymbol{B}_i = \boldsymbol{H}_i^{'} + \varepsilon_i$	Observation model with iid noise
$\mathcal{MN}(\boldsymbol{0}, \boldsymbol{U}, \boldsymbol{V})$	Matrix-normal distribution with mean 0, row covariance $\boldsymbol{U}$, column covariance $\boldsymbol{V}$
$p(\boldsymbol{H}) = \mathcal{MN}(\boldsymbol{0}, \boldsymbol{U}, \boldsymbol{I})$	Matrix normal distribution with mean 0, row covariance $\boldsymbol{U}$

A.1   Relative poses and channel prior

We model the channel with a Matrix-Normal distribution. The Matrix Normal has the following properties: if $\boldsymbol{X} \sim \mathcal{MN}(\boldsymbol{M}, \boldsymbol{U}, \boldsymbol{V})$ then $\text{vec}\,(\boldsymbol{X}) = \mathcal{N}(\text{vec}\,(\boldsymbol{M}), \boldsymbol{V} \otimes \boldsymbol{U})$. Here, $\text{vec}\,(\boldsymbol{X})$ is the Frobenius vectorization of a matrix $\boldsymbol{X}$ and $\otimes$ is the Kronecker product.

Let's denote the prior as $p(\cdot)$, then the following equivalence holds for the $\mathcal{MN}(0, \boldsymbol{U}, \boldsymbol{I})$ matrix normal:

$$p\left(\begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \\ \boldsymbol{H}_3 \\ \vdots \end{bmatrix}\right) = p\left(\begin{bmatrix} \boldsymbol{H}_1 \boldsymbol{P}_1^{-1} \\ \boldsymbol{H}_2 \boldsymbol{P}_1^{-1} \\ \boldsymbol{H}_3 \boldsymbol{P}_1^{-1} \\ \vdots \end{bmatrix}\right) \tag{8}$$
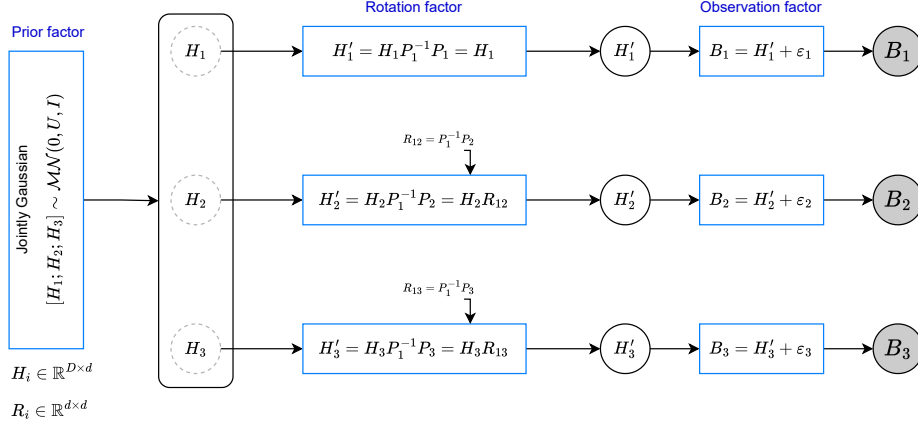
Figure 4: Structured Probabilistic Graphical Model to illustrate the joint distribution in Equation 2

This uses the rotated matrices:

$$\begin{bmatrix} \boldsymbol{H}_1^{'} \\ \boldsymbol{H}_2^{'} \\ \boldsymbol{H}_3^{'} \\ \vdots \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_1\boldsymbol{P}_1^{-1}\boldsymbol{P}_1 \\ \boldsymbol{H}_2\boldsymbol{P}_1^{-1}\boldsymbol{P}_2 \\ \boldsymbol{H}_3\boldsymbol{P}_1^{-1}\boldsymbol{P}_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2\boldsymbol{R}_{12} \\ \boldsymbol{H}_3\boldsymbol{R}_{13} \\ \vdots \end{bmatrix} \tag{9}$$

B   Estimate channel using a rotation estimate

We first derive the estimation of the channel if the rotations were known. We present the case for reconstructing four matrix-valued signals in the presence of relative rotation. The dots, $\cdots$, will indicate generality towards sets of more matrices. The variable J indicates the number of blocks considered in the denoising. For this example, $J = 4$.

The derivation starts from $\log p(\boldsymbol{B}, \hat{\boldsymbol{R}}, \boldsymbol{H})$. This assumes that one has an estimate of the Rotation matrices, $\boldsymbol{R}$, and the goal is to find the channel matrices $\boldsymbol{H}$ with the largest likelihood.

$$\arg\max_{\boldsymbol{H}} \log p(\boldsymbol{B}, \hat{\boldsymbol{R}}, \boldsymbol{H}) \tag{10}$$

$$= \arg\max_{\boldsymbol{H}} \log \int_{\boldsymbol{H}_1} \cdots \int_{\boldsymbol{H}_4} p(\boldsymbol{H}_1, \boldsymbol{H}_2, \boldsymbol{H}_3, \boldsymbol{H}_4) p(\boldsymbol{H}_1^{'}|\boldsymbol{H}_1) \prod_{j\in[2,3,4]} p(\boldsymbol{H}_j^{'}|\boldsymbol{H}_j, \hat{\boldsymbol{R}}_{1j})$$
$$\cdot p(\boldsymbol{B}_1|\boldsymbol{H}_1^{'})\Big(\prod_{j\in[2,3,4]} p(\boldsymbol{B}_j|\boldsymbol{H}_j^{'})\Big) d\boldsymbol{H}_1 d\boldsymbol{H}_2 d\boldsymbol{H}_3 d\boldsymbol{H}_4 \tag{11}$$

$$= \arg\max_{\boldsymbol{H}} = \log \int_{\boldsymbol{H}_1} \cdots \int_{\boldsymbol{H}_4} p(\boldsymbol{H}_1, \boldsymbol{H}_2, \boldsymbol{H}_3, \boldsymbol{H}_4)\delta_{\boldsymbol{H}_1=\boldsymbol{H}_1^{'}} \prod_{j\in[2,3,4]} \delta_{\boldsymbol{H}_j=\boldsymbol{H}_j^{'}\hat{\boldsymbol{R}}_{j1}}$$
$$\cdot \Big(\prod_{j\in[1,\cdots,4]} p(\boldsymbol{B}_j|\boldsymbol{H}_j^{'})\Big) d\boldsymbol{H}_1 d\boldsymbol{H}_2 d\boldsymbol{H}_3 d\boldsymbol{H}_4 \tag{12}$$

We integrate over the delta distributions and introduce a change of variables $\boldsymbol{H}_2^{'} = \boldsymbol{H}_2\hat{\boldsymbol{R}}_{12}$ and $\boldsymbol{H}_3^{'} = \boldsymbol{H}_3\hat{\boldsymbol{R}}_{13}$ and $\boldsymbol{H}_4^{'} = \boldsymbol{H}_4\hat{\boldsymbol{R}}_{14}$.

$$\arg\max_{\boldsymbol{H}} \log p(\boldsymbol{H}_1^{'}, \boldsymbol{H}_2^{'}\hat{\boldsymbol{R}}_{21}, \boldsymbol{H}_3^{'}\hat{\boldsymbol{R}}_{31}, \boldsymbol{H}_4^{'}\hat{\boldsymbol{R}}_{41}) \prod_{j\in[1,\cdots,4]} p(\boldsymbol{B}_j|\boldsymbol{H}_j^{'}) \tag{13}$$

$$= \arg\max_{\boldsymbol{H}} \log p(\boldsymbol{H}_1^{'}, \boldsymbol{H}_2, \boldsymbol{H}_3, \boldsymbol{H}_4) p(\boldsymbol{B}_1|\boldsymbol{H}_1^{'}) \prod_{j\in[2,3,4]} p(\boldsymbol{B}_j|\boldsymbol{H}_j\hat{\boldsymbol{R}}_{1j}) \tag{14}$$

9

Next, we write down the actual likelihoods that are Matrix Normal distribution by $\mathcal{MN}$ and align the writing of said distributions $\begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \\ \boldsymbol{H}_3 \\ \boldsymbol{H}_4 \end{bmatrix} \sim \mathcal{MN}(0, \boldsymbol{U}, \boldsymbol{I})$.

The operation $\mathrm{vec}(\cdot)$ indicates the "Frobenius vectorization" of matrix $\boldsymbol{X}$, which concatenates all columns from left to right in one long column vector. This means that we can conveniently write down the Matrix Normal distribution. If $\boldsymbol{X}$ has a matrix normal distribution $\boldsymbol{X} \sim \mathcal{MN}(0, \boldsymbol{U}, \boldsymbol{I})$, then $\mathrm{vec}(\boldsymbol{X})$ has Normal distribution $\mathrm{vec}(\boldsymbol{X}) \sim \mathcal{N}(0, \boldsymbol{I} \otimes \boldsymbol{U})$ where $\otimes$ indicates the Kronecker product.

$$
\underset{\boldsymbol{H}}{\arg\max} \log \mathcal{N}(\mathrm{vec}\left(\boldsymbol{H}_1^{'}, \boldsymbol{H}_2, \boldsymbol{H}_3, \boldsymbol{H}_4\right) | 0, \boldsymbol{I}_d \otimes \boldsymbol{U}) + \log \mathcal{N}(\boldsymbol{H}_1^{'} | \boldsymbol{B}_1, \sigma_\epsilon^2 \boldsymbol{I}_{dD})
$$

$$
+ \sum_{j \in [2,3,4]} \log \mathcal{N}(\boldsymbol{H}_j | \boldsymbol{B}_j \hat{\boldsymbol{R}}_{j1}, \sigma_\epsilon^2 \boldsymbol{I}_{dD}) \tag{15}
$$

$$
= \underset{\boldsymbol{H}}{\arg\max} \log \mathcal{N}\left(\mathrm{vec}\left(\boldsymbol{H}_1^{'}, \boldsymbol{H}_2, \boldsymbol{H}_3, \boldsymbol{H}_4\right) | 0, \boldsymbol{I}_d \otimes U \right)
$$

$$
+ \log \mathcal{N}\left(\mathrm{vec}\left(\boldsymbol{H}_1^{'}, \boldsymbol{H}_2, \boldsymbol{H}_3, \boldsymbol{H}_4\right) | \mathrm{vec}\left(\begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \hat{\boldsymbol{R}}_{21} \\ \boldsymbol{B}_3 \hat{\boldsymbol{R}}_{31} \\ \boldsymbol{B}_4 \hat{\boldsymbol{R}}_{41} \end{bmatrix} \right), \boldsymbol{I}_d \otimes \sigma_\epsilon^2 \boldsymbol{I}_{4D} \right) \tag{16}
$$

$$
= \underset{\boldsymbol{H}}{\arg\max} \log \mathcal{N}\left(\mathrm{vec}\left(\boldsymbol{H}_1^{'}, \boldsymbol{H}_2, \boldsymbol{H}_3, \boldsymbol{H}_4\right) | \boldsymbol{\Sigma}_{\boldsymbol{H}} \mathrm{vec}\left(\sigma_\epsilon^{-2} \boldsymbol{B}_1, \sigma_\epsilon^{-2} \boldsymbol{B}_2 \hat{\boldsymbol{R}}_{21}, \sigma_\epsilon^{-2} \boldsymbol{B}_3 \hat{\boldsymbol{R}}_{31}, \sigma_\epsilon^{-2} \boldsymbol{B}_4 \hat{\boldsymbol{R}}_{41} \right), \boldsymbol{\Sigma}_{\boldsymbol{H}} \right) \tag{17}
$$

We use shorthand for the denoising correlation matrix:

$$
\boldsymbol{\Sigma}_{\boldsymbol{H}} = \left((\boldsymbol{I}_d \otimes \boldsymbol{U})^{-1} + (\boldsymbol{I}_d \otimes \sigma_\epsilon^2 \boldsymbol{I}_{JD})^{-1} \right)^{-1}
$$

$$
= \left((\boldsymbol{I}_d \otimes \boldsymbol{U})^{-1} + \sigma_\epsilon^{-2} \boldsymbol{I}_{JDd} \right)^{-1}
$$

$$
= \left(\begin{bmatrix} (\boldsymbol{U}^{-1} + \sigma_\epsilon^{-2} \boldsymbol{I}_{JD})^{-1} & 0 & 0 & \cdots \\ 0 & (\boldsymbol{U}^{-1} + \sigma_\epsilon^{-2} \boldsymbol{I}_{JD})^{-1} & 0 & \cdots \\ 0 & 0 & (\boldsymbol{U}^{-1} + \sigma_\epsilon^{-2} \boldsymbol{I}_{JD})^{-1} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \right)^{-1}
$$

$$
= \boldsymbol{I}_d \otimes \left(\boldsymbol{U}^{-1} + \sigma_\epsilon^{-2} \boldsymbol{I}_{4D} \right)^{-1}. \tag{18}
$$

The argmax of a Gaussian likelihood is the mean parameter of the said Gaussian distribution. Therefore, we find the mean parameter of $\boldsymbol{H}^{'}$ from:

$$
\begin{bmatrix} \boldsymbol{H}_{1,t+1}^{'} \\ \boldsymbol{H}_{2,t+1}^{'} \hat{\boldsymbol{R}}_{21} \\ \boldsymbol{H}_{3,t+1}^{'} \hat{\boldsymbol{R}}_{31} \\ \boldsymbol{H}_{4,t+1}^{'} \hat{\boldsymbol{R}}_{41} \end{bmatrix} = (\boldsymbol{I}_d \otimes (\left(\boldsymbol{U}^{-1} + \sigma_\epsilon^{-2} \boldsymbol{I}_{4D} \right)^{-1})) \begin{bmatrix} \sigma_\epsilon^{-2} \boldsymbol{B}_1 \\ \sigma_\epsilon^{-2} \boldsymbol{B}_2 \hat{\boldsymbol{R}}_{21} \\ \sigma_\epsilon^{-2} \boldsymbol{B}_3 \hat{\boldsymbol{R}}_{31} \\ \sigma_\epsilon^{-2} \boldsymbol{B}_4 \hat{\boldsymbol{R}}_{41} \end{bmatrix}. \tag{19}
$$

The general equation for general J would be:

$$
\begin{bmatrix} \boldsymbol{H}_{1,t+1}^{'} \\ \boldsymbol{H}_{2,t+1}^{'} \hat{\boldsymbol{R}}_{21} \\ \boldsymbol{H}_{3,t+1}^{'} \hat{\boldsymbol{R}}_{31} \\ \boldsymbol{H}_{4,t+1}^{'} \hat{\boldsymbol{R}}_{41} \\ \vdots \end{bmatrix} = (\boldsymbol{I}_d \otimes (\left(\boldsymbol{U}^{-1} + \sigma_\epsilon^{-2} \boldsymbol{I}_{JD} \right)^{-1})) \begin{bmatrix} \sigma_\epsilon^{-2} \boldsymbol{B}_1 \\ \sigma_\epsilon^{-2} \boldsymbol{B}_2 \hat{\boldsymbol{R}}_{21} \\ \sigma_\epsilon^{-2} \boldsymbol{B}_3 \hat{\boldsymbol{R}}_{31} \\ \sigma_\epsilon^{-2} \boldsymbol{B}_4 \hat{\boldsymbol{R}}_{41} \\ \vdots \end{bmatrix}. \tag{20}
$$

C   Directly estimate rotations

For the direct approach, we estimate the rotation matrices within each subgraph. This step occurs even before iterative refinement and provides the first estimate of the rotation matrices based on the noisy observations.

C.1   MLE, three obs, directly from B

The comments prefixed with '...' in blue are comments to explain the derivation steps.

$$
\hat{\boldsymbol{R}}_{12}, \hat{\boldsymbol{R}}_{13}
$$

$$
= \underset{\boldsymbol{R}_{12},\boldsymbol{R}_{13}}{\arg\max} \ \log p(\boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3 | \boldsymbol{R}_{12}, \boldsymbol{R}_{13}) \tag{21}
$$

... Introduce and marginalize over H_1, H_2, and H_3

$$
= \underset{\boldsymbol{R}_{12},\boldsymbol{R}_{13}}{\arg\max} \ \log \int_{\boldsymbol{H}_1} \int_{\boldsymbol{H}_2} \int_{\boldsymbol{H}_3} p(\boldsymbol{H}_1, \boldsymbol{H}_2, \boldsymbol{H}_3) \int_{\boldsymbol{H}_1'} \int_{\boldsymbol{H}_2'} \int_{\boldsymbol{H}_3'} p(\boldsymbol{B}_1|\boldsymbol{H}_1')p(\boldsymbol{B}_2|\boldsymbol{H}_2')p(\boldsymbol{B}_3|\boldsymbol{H}_3')
$$
$$
p(\boldsymbol{H}_1'|\boldsymbol{H}_1) \, p(\boldsymbol{H}_2'|\boldsymbol{H}_2, \boldsymbol{R}_{12}) \, p(\boldsymbol{H}_3'|\boldsymbol{H}_3, \boldsymbol{R}_{13}) \, d\boldsymbol{H}_1' d\boldsymbol{H}_2' d\boldsymbol{H}_3' \, d\boldsymbol{H}_1 d\boldsymbol{H}_2 d\boldsymbol{H}_3 \tag{22}
$$

$$
= \underset{\boldsymbol{R}_{12},\boldsymbol{R}_{13}}{\arg\max} \ \log \int_{\boldsymbol{H}_1} \int_{\boldsymbol{H}_2} \int_{\boldsymbol{H}_3} p(\boldsymbol{H}_1, \boldsymbol{H}_2, \boldsymbol{H}_3) \int_{\boldsymbol{H}_1'} \int_{\boldsymbol{H}_2'} \int_{\boldsymbol{H}_3'} p(\boldsymbol{B}_1|\boldsymbol{H}_1')p(\boldsymbol{B}_2|\boldsymbol{H}_2')p(\boldsymbol{B}_3|\boldsymbol{H}_3')
$$
$$
\delta_{\boldsymbol{H}_1=\boldsymbol{H}_1'} \, \delta_{\boldsymbol{H}_2=\boldsymbol{H}_2'\boldsymbol{R}_{21}} \, \delta_{\boldsymbol{H}_3=\boldsymbol{H}_3'\boldsymbol{R}_{31}} \, d\boldsymbol{H}_1' d\boldsymbol{H}_2' d\boldsymbol{H}_3' \, d\boldsymbol{H}_1 d\boldsymbol{H}_2 d\boldsymbol{H}_3 \tag{23}
$$

... Integrate against delta distributions

$$
= \underset{\boldsymbol{R}_{12},\boldsymbol{R}_{13}}{\arg\max} \ \log \int_{\boldsymbol{H}_1'} \int_{\boldsymbol{H}_2'} \int_{\boldsymbol{H}_3'} p(\boldsymbol{H}_1', \boldsymbol{H}_2'\boldsymbol{R}_{21}, \boldsymbol{H}_3'\boldsymbol{R}_{31}) \, p(\boldsymbol{B}_1|\boldsymbol{H}_1')p(\boldsymbol{B}_2|\boldsymbol{H}_2')p(\boldsymbol{B}_3|\boldsymbol{H}_3') \, d\boldsymbol{H}_1' d\boldsymbol{H}_2' d\boldsymbol{H}_3' \tag{24}
$$

... Make Matrix Normal explicit

$$
= \underset{\boldsymbol{R}_{12},\boldsymbol{R}_{13}}{\arg\max} \ \log \int_{\boldsymbol{H}_1'} \int_{\boldsymbol{H}_2'} \int_{\boldsymbol{H}_3'} \mathcal{MN}(\begin{bmatrix} \boldsymbol{H}_1' \\ \boldsymbol{H}_2'\boldsymbol{R}_{21} \\ \boldsymbol{H}_3'\boldsymbol{R}_{31} \end{bmatrix}; 0, \boldsymbol{U}, \boldsymbol{I}_d) \, \mathcal{MN}(\boldsymbol{H}_1'|\boldsymbol{B}_1, \sigma^2\boldsymbol{I}_D, \boldsymbol{I}_d)
$$
$$
\mathcal{MN}(\boldsymbol{H}_2'\boldsymbol{R}_{21}; B_2\boldsymbol{R}_{21}, \sigma^2\boldsymbol{I}_D, \boldsymbol{I}_d) \, \mathcal{MN}(\boldsymbol{H}_3'\boldsymbol{R}_{31}; B_3\boldsymbol{R}_{31}, \sigma^2\boldsymbol{I}_D, \boldsymbol{I}_d) \, d\boldsymbol{H}_1' d\boldsymbol{H}_2' d\boldsymbol{H}_3' \tag{25}
$$

... Fuse Matrix normals

$$
= \underset{\boldsymbol{R}_{12},\boldsymbol{R}_{13}}{\arg\max} \ \log \int_{\boldsymbol{H}_1'} \int_{\boldsymbol{H}_2'} \int_{\boldsymbol{H}_3'} \mathcal{MN}(\begin{bmatrix} \boldsymbol{H}_1' \\ \boldsymbol{H}_2'\boldsymbol{R}_{21} \\ \boldsymbol{H}_3'\boldsymbol{R}_{31} \end{bmatrix}; 0, \boldsymbol{U}, \boldsymbol{I}_d) \, \mathcal{MN}(\begin{bmatrix} \boldsymbol{H}_1' \\ \boldsymbol{H}_2'\boldsymbol{R}_{21} \\ \boldsymbol{H}_3'\boldsymbol{R}_{31} \end{bmatrix} | \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2\boldsymbol{R}_{21} \\ \boldsymbol{B}_3\boldsymbol{R}_{31} \end{bmatrix}, \sigma^2\boldsymbol{I}_{3D}, \boldsymbol{I}_d) \, d\boldsymbol{H}_1' d\boldsymbol{H}_2' d\boldsymbol{H}_3' \tag{26}
$$

$$
= \underset{\boldsymbol{R}_{12},\boldsymbol{R}_{13}}{\arg\max} \ \log \mathcal{MN}(\begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2\boldsymbol{R}_{21} \\ \boldsymbol{B}_3\boldsymbol{R}_{31} \end{bmatrix}; 0, \boldsymbol{U} + \sigma^2 I, \boldsymbol{I}_d) \tag{27}
$$

The simplification from two to one Matrix Normal uses some tricks: the multiplication of two normal distributions yields a normal distribution, and the integral will not change under rotation. Therefore, we can use standard properties of the Matrix Normal product. The argmax over the matrix normal can be obtained by writing out the Trace and removing scalar constants:

$$
\hat{\boldsymbol{R}}_{12}, \hat{\boldsymbol{R}}_{13}
$$

$$
= \underset{\boldsymbol{R}_{12},\boldsymbol{R}_{13}}{\arg\max} \ \frac{-1}{2} \operatorname{Tr} \left[\begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2\boldsymbol{R}_{21} \\ \boldsymbol{B}_3\boldsymbol{R}_{31} \end{bmatrix}^T (\boldsymbol{U} + \sigma^2\boldsymbol{I})^{-1} \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2\boldsymbol{R}_{21} \\ \boldsymbol{B}_3\boldsymbol{R}_{31} \end{bmatrix} \right]
$$

$$
= \underset{\boldsymbol{R}_{12},\boldsymbol{R}_{13}}{\arg\max} \ \frac{-1}{2} \operatorname{Tr} \left[\begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2\boldsymbol{R}_{21} \\ \boldsymbol{B}_3\boldsymbol{R}_{31} \end{bmatrix} \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2\boldsymbol{R}_{21} \\ \boldsymbol{B}_3\boldsymbol{R}_{31} \end{bmatrix}^T (\boldsymbol{U} + \sigma^2\boldsymbol{I})^{-1} \right] \tag{28}
$$

$$
= \underset{\boldsymbol{R}_{12},\boldsymbol{R}_{13}}{\arg\max} \ - \operatorname{Tr} \left[\begin{bmatrix} B_1\boldsymbol{B}_1^T & B_1\boldsymbol{R}_{12}\boldsymbol{B}_2^T & B_1\boldsymbol{R}_{13}\boldsymbol{B}_3^T \\ B_2\boldsymbol{R}_{21}\boldsymbol{B}_1^T & B_2\boldsymbol{B}_2^T & B_2\boldsymbol{R}_{21}\boldsymbol{R}_{13}\boldsymbol{B}_3^T \\ B_3\boldsymbol{R}_{31}\boldsymbol{B}_1^T & B_3\boldsymbol{R}_{31}\boldsymbol{R}_{12}\boldsymbol{B}_2^T & B_3\boldsymbol{B}_3^T \end{bmatrix} (\boldsymbol{U} + \sigma^2\boldsymbol{I})^{-1} \right] \tag{29}
$$

Equation 29 considers the matrix as composed of submatrices. Since the U matrix is always composed of $J \cdot D$ by $J \cdot D$ elements, we can always split it in J by J submatrices of size $D \times D$:

$$-(U + \sigma^2 I)^{-1} = \begin{bmatrix} U_1 & U_a & U_b \\ U_a^T & U_2 & U_c \\ U_b^T & U_c^T & U_3 \end{bmatrix} \tag{30}$$

$$\hat{R}_{12}, \hat{R}_{13} = \underset{R_{12}, R_{13}}{\arg\max} \operatorname{Tr} \left[\begin{bmatrix} B_1 B_1^T & B_1 R_{12} B_2^T & B_1 R_{13} B_3^T \\ B_2 R_{21} B_1^T & B_2 B_2^T & B_2 R_{21} R_{13} B_3^T \\ B_3 R_{31} B_1^T & B_3 R_{31} R_{12} B_2^T & B_3 B_3^T \end{bmatrix} \begin{bmatrix} U_1 & U_a & U_b \\ U_a^T & U_2 & U_c \\ U_b^T & U_c^T & U_3 \end{bmatrix} \right] \tag{31}$$

...Split to individual traces

$$= \underset{R_{12}, R_{13}}{\arg\max} 2 \operatorname{Tr} \left[B_1 R_{12} B_2^T U_a^T \right] + 2 \operatorname{Tr} \left[B_1 R_{13} B_3^T U_b^T \right] + 2 \operatorname{Tr} \left[B_2 R_{21} R_{13} B_3^T U_c^T \right] \tag{32}$$

The argument in Equation 32 is linear in both rotation matrices. In each argument, there is a closed-form expression for the optimum over the set of orthogonal matrices. Therefore, we suggest to iteratively determine this optimum. With each closed form solution, each step is guaranteed to non-decrease the objective value and we find that this usually converges to machine precision in less than eight steps.

$$\text{iterate} \begin{cases} R_{13} = \arg\max \langle R_{21}, B_2^T U_a^T B_1 \rangle_F + \langle R_{31}, B_3^T U_b^T B_1 \rangle_F + \langle R_{31}, B_3^T U_c^T B_2 R_{21} \rangle_F \\ R_{12} = \arg\max \langle R_{21}, B_2^T U_a^T B_1 \rangle_F + \langle R_{31}, B_3^T U_b^T B_1 \rangle_F + \langle R_{21}, B_2^T U_c B_3 R_{31} \rangle_F \end{cases} \tag{33}$$

The closed-form solution is known as the Orthogonal Procrustes problem [8, 15]. The projection is achieved by removing the singular values from the Singular-Value Decomposition:

$$\underset{R}{\arg\max} \langle R, X \rangle_F = A L V^T \tag{34}$$

from Singular Value Decomposition $X = A \Sigma V^T$ and with $L = I$ if $\det(X) > 0$ or $L = \operatorname{diag}(1, 1, 1, \ldots, -1)$ otherwise (assuming the smallest singular-value of X is at the bottom of Σ).

D Iterative refinement 3

Deriving the iterative refinement looks much like Equation 21. However, it crucially starts from optimizing the rotation matrices given the channel estimates H, instead of the noisy estimates B.

The comments prefixed with '...' are comments to explain the derivation steps.

$$\hat{R}_{12}, \hat{R}_{13}$$

$$= \underset{R_{12}, R_{13}}{\arg\max} \, \log p(R_{12}, R_{13} | H') \tag{35}$$

$$= \underset{R_{12}, R_{13}}{\arg\max} \, \log p(H_1', H_2', H_3' | R_{12}, R_{13}) \tag{36}$$

... Introduce and marginalize over H_1, H_2, and H_3

$$= \underset{R_{12}, R_{13}}{\arg\max} \, \log \int_{H_1} \int_{H_2} \int_{H_3} p(H_1, H_2, H_3) \, p(H_1' | H_1) \, p(H_2' | H_2, R_{12}) \, p(H_3' | H_3, R_{13}) \, dH_1 \, dH_2 \, dH_3 \tag{37}$$

$$= \underset{R_{12}, R_{13}}{\arg\max} \, \log \int_{H_1} \int_{H_2} \int_{H_3} p(H_1, H_2, H_3) \, \delta_{H_1 = H_1'} \, \delta_{H_2 = H_2' R_{21}} \, \delta_{H_3 = H_3' R_{31}} \, dH_1 \, dH_2 \, dH_3 \tag{38}$$

$$= \underset{R_{12}, R_{13}}{\arg\max} \, \log p(H_1', H_2' R_{21}, H_3' R_{31}) \tag{39}$$

We'll write down the matrix-normal form of the channel prior: $\begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix} \sim \mathcal{MN}(0, U, I)$:

$$\hat{\boldsymbol{R}}_{12}, \hat{\boldsymbol{R}}_{13}$$

$$= \underset{\boldsymbol{R}_{12}, \boldsymbol{R}_{13}}{\arg\max} \frac{-1}{2} \operatorname{Tr}\left[\begin{bmatrix} \boldsymbol{H}_1^{'} \\ \boldsymbol{H}_2^{'}\boldsymbol{R}_{21} \\ \boldsymbol{H}_3^{'}\boldsymbol{R}_{31} \end{bmatrix}^T \boldsymbol{U}^{-1} \begin{bmatrix} \boldsymbol{H}_1^{'} \\ \boldsymbol{H}_2^{'}\boldsymbol{R}_{21} \\ \boldsymbol{H}_3^{'}\boldsymbol{R}_{31} \end{bmatrix} \right]$$

$$= \underset{\boldsymbol{R}_{12}, \boldsymbol{R}_{13}}{\arg\max} \frac{-1}{2} \operatorname{Tr}\left[\begin{bmatrix} \boldsymbol{H}_1^{'} \\ \boldsymbol{H}_2^{'}\boldsymbol{R}_{21} \\ \boldsymbol{H}_3^{'}\boldsymbol{R}_{31} \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_1^{'} \\ \boldsymbol{H}_2^{'}\boldsymbol{R}_{21} \\ \boldsymbol{H}_3^{'}\boldsymbol{R}_{31} \end{bmatrix}^T \boldsymbol{U}^{-1} \right] \tag{40}$$

$$= \underset{\boldsymbol{R}_{12}, \boldsymbol{R}_{13}}{\arg\max} -\operatorname{Tr}\left[\begin{bmatrix} \boldsymbol{H}_1^{'}\boldsymbol{H}_1^{'T} & \boldsymbol{H}_1^{'}\boldsymbol{R}_{12}\boldsymbol{H}_2^{'T} & \boldsymbol{H}_1^{'}\boldsymbol{R}_{13}\boldsymbol{H}_3^{'T} \\ \boldsymbol{H}_2^{'}\boldsymbol{R}_{21}\boldsymbol{H}_1^{'T} & \boldsymbol{H}_2^{'}\boldsymbol{H}_2^{'T} & \boldsymbol{H}_2^{'}\boldsymbol{R}_{21}\boldsymbol{R}_{13}\boldsymbol{H}_3^{'T} \\ \boldsymbol{H}_3^{'}\boldsymbol{R}_{31}\boldsymbol{H}_1^{'T} & \boldsymbol{H}_3^{'}\boldsymbol{R}_{31}\boldsymbol{R}_{12}\boldsymbol{H}_2^{'T} & \boldsymbol{H}_3^{'}\boldsymbol{H}_3^{'T} \end{bmatrix} \boldsymbol{U}^{-1} \right] \tag{41}$$

...Consider $-\boldsymbol{U}^{-1}$ a block matrix $\begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_a & \boldsymbol{U}_b \\ \boldsymbol{U}_a^T & \boldsymbol{U}_2 & \boldsymbol{U}_c \\ \boldsymbol{U}_b^T & \boldsymbol{U}_c^T & \boldsymbol{U}_3 \end{bmatrix}$ with $\boldsymbol{U} \in \mathbb{R}^{3D \times 3D}$ and $\boldsymbol{U}_a \in \mathbb{R}^{D \times D}$

$$= \underset{\boldsymbol{R}_{12}, \boldsymbol{R}_{13}}{\arg\max} \operatorname{Tr}\left[\begin{bmatrix} \boldsymbol{H}_1^{'}\boldsymbol{H}_1^{'T} & \boldsymbol{H}_1^{'}\boldsymbol{R}_{12}\boldsymbol{H}_2^{'T} & \boldsymbol{H}_1^{'}\boldsymbol{R}_{13}\boldsymbol{H}_3^{'T} \\ \boldsymbol{H}_2^{'}\boldsymbol{R}_{21}\boldsymbol{H}_1^{'T} & \boldsymbol{H}_2^{'}\boldsymbol{H}_2^{'T} & \boldsymbol{H}_2^{'}\boldsymbol{R}_{21}\boldsymbol{R}_{13}\boldsymbol{H}_3^{'T} \\ \boldsymbol{H}_3^{'}\boldsymbol{R}_{31}\boldsymbol{H}_1^{'T} & \boldsymbol{H}_3^{'}\boldsymbol{R}_{31}\boldsymbol{R}_{12}\boldsymbol{H}_2^{'T} & \boldsymbol{H}_3^{'}\boldsymbol{H}_3^{'T} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_a & \boldsymbol{U}_b \\ \boldsymbol{U}_a^T & \boldsymbol{U}_2 & \boldsymbol{U}_c \\ \boldsymbol{U}_b^T & \boldsymbol{U}_c^T & \boldsymbol{U}_3 \end{bmatrix} \right] \tag{42}$$

...Split to individual traces

$$= \underset{\boldsymbol{R}_{12}, \boldsymbol{R}_{13}}{\arg\max} 2\operatorname{Tr}\left[\boldsymbol{H}_1^{'}\boldsymbol{R}_{12}\boldsymbol{H}_2^{'T}\boldsymbol{U}_a^T \right] + 2\operatorname{Tr}\left[\boldsymbol{H}_1^{'}\boldsymbol{R}_{13}\boldsymbol{H}_3^{'T}\boldsymbol{U}_b^T \right] + 2\operatorname{Tr}\left[\boldsymbol{H}_2^{'}\boldsymbol{R}_{21}\boldsymbol{R}_{13}\boldsymbol{H}_3^{'T}\boldsymbol{U}_c^T \right] \tag{43}$$

The optimization argument in Equation 43 is linear in both rotation matrices. In each argument, there is a closed-form expression for the optimum over the set of orthogonal matrices. Therefore, we suggest to iteratively determine this optimum. With each closed form solution, each step is guaranteed to non-decrease the objective value and we find that this usually converges to machine precision in less than eight steps.

iterate until convergence

$$\begin{cases} \boldsymbol{R}_{12} = \arg\max_{\boldsymbol{R}_{12}} \langle \boldsymbol{R}_{21}, \boldsymbol{H}_2^{'T}\boldsymbol{U}_a^T\boldsymbol{H}_1^{'} \rangle_F + \langle \boldsymbol{R}_{31}, \boldsymbol{H}_3^{'T}\boldsymbol{U}_b^T\boldsymbol{H}_1^{'} \rangle_F + \langle \boldsymbol{R}_{21}, \boldsymbol{H}_2^{'T}\boldsymbol{U}_c\boldsymbol{H}_3^{'}\boldsymbol{R}_{31} \rangle_F \\ \boldsymbol{R}_{13} = \arg\max_{\boldsymbol{R}_{13}} \langle \boldsymbol{R}_{21}, \boldsymbol{H}_2^{'T}\boldsymbol{U}_a^T\boldsymbol{H}_1^{'} \rangle_F + \langle \boldsymbol{R}_{31}, \boldsymbol{H}_3^{'T}\boldsymbol{U}_b^T\boldsymbol{H}_1^{'} \rangle_F + \langle \boldsymbol{R}_{31}, \boldsymbol{H}_3^{'T}\boldsymbol{U}_c^T\boldsymbol{H}_2^{'}\boldsymbol{R}_{21} \rangle_F \end{cases}$$

The closed-form solution to the latter argmax is the same as the Procrustes problem in Section C. Each step is a closed-form maximizer, so the iteration is guaranteed to converge.

E   Local to Global synchronization

This section includes the theoretical proofs behind our main arguments for reducing the absolute pose estimation problem to a relative pose synchronization problem. In particular, we will show

1. that the set of absolute poses and the set of cycle-consistent relative poses are in one-to-one correspondence (up to a global transformation of the absolute poses) and that

2. if all 3-cycles (triplets) in a surface mesh (e.g., the physical resource blocks in the frequency-time lattice in the 5G NR channel) are cycle-consistent, then the whole mesh is cycle consistent.

First, we define the notion of cycle consistency precisely. Consider a **connected graph** $G = (V, E)$ with a set V of N nodes and a set E of edges. Then, an assignment of relative poses $R = \{R_{ij}\}_{(i,j) \in E}$ is cycle consistent along a k-cycle $\gamma = [\gamma_1, \gamma_2, \ldots, \gamma_k, \gamma_1]$ (with $\gamma_i \in V$ and $(\gamma_i, \gamma_{i+1}) \in E$) if

$$R_{\gamma_1, \gamma_2} R_{\gamma_2, \gamma_3} \cdots R_{\gamma_k, \gamma_1} = I \tag{44}$$

An assignment of relative poses $\{R_{ij}\}_{(i,j)\in E}$ is cycle consistent if it is cycle-consistent over all cycles in the graph G.

E.1 Absolute to relative poses

We need to prove two directions. First, any set of absolute poses $P = \{P_i\}_{i=1}^N$ is associated with one unique set of cycle-consistent relative poses $R = \{R_{ij}\}_{1\le i,j\le N}$. This true by definition, since $R_{ij} := P_i^{-1}P_j$ and for any cycle $\gamma = [\gamma_1, \gamma_2, \ldots, \gamma_k, \gamma_1]$:

$$R_{\gamma_1,\gamma_2}R_{\gamma_2,\gamma_3}\cdots R_{\gamma_k,\gamma_1} = P_{\gamma_1}^{-1}P_{\gamma_2}P_{\gamma_2}^{-1}P_{\gamma_3}\cdots P_{\gamma_k}P_{\gamma_1} = P_{\gamma_1}^{-1}P_{\gamma_1} = I \tag{45}$$

Second, we show that any set of cycle-consistent relative poses corresponds to an *equivalence class* of absolute poses (defined up to a global transformation). First, we note that cycle-consistency implies that, for any pair of nodes i, j, walking along any path connecting them generates the same transformation, i.e., the relative pose R_{ij} is well defined (even if edge $(i, j) \notin E$). For any pair of paths α, β of length a and b connecting i to j, i.e. $\alpha_1 = \beta_1 = i$ and $\alpha_a = \beta_b = j$, concatenating α and β reversed form a cycle $\gamma = [\alpha_1 = i, \alpha_2, \alpha_3, \ldots, \alpha_a = j = \beta_b, \ldots, \beta_3, \beta_2, \beta_1 = i]$. Since the relative poses R are cycle-consistent, it follows that

$$R_{\alpha_1=i,\alpha_2}R_{\alpha_2,\alpha_3}\cdots R_{\alpha_{a-1},\alpha_a=j}R_{\beta_b=j,\beta_{b-1}}\cdots R_{\beta_2,\beta_1=i} = I \tag{46}$$

$$R_{\alpha_1=i,\alpha_2}R_{\alpha_2,\alpha_3}\cdots R_{\alpha_{a-1},\alpha_a=j} = R_{\beta_2,\beta_1=i}^{-1}\cdots R_{\beta_b=j,\beta_{b-1}}^{-1} \tag{47}$$

$$R_{\alpha_1=i,\alpha_2}R_{\alpha_2,\alpha_3}\cdots R_{\alpha_{a-1},\alpha_a=j} = R_{\beta_1=i,\beta_2}\cdots R_{\beta_{b-1},\beta_b=j} =: R_{i,j} \tag{48}$$

Hence, one can construct a set of absolute poses P by choosing an arbitrary pose for the first node P_0 and then defining $P_i := P_0 R_{0,i}$. Note that any choice of P_0 is valid, but leads to a different set of absolute poses. In particular, any two such equivalent sets of absolute poses P and P' differ precisely by a global transformation, i.e., $\exists T : \forall i, P'_i = TP_i$. Moreover, picking a different starting node j rather than 0 results in the equivalent solution one would have obtained by instead picking $P_0 = TR_{j,0}$.

Finally, any of these equivalent solutions generates the initial set of relative poses; hence, we have proved the bi-directional correspondence.

E.2 Sufficiency of 3-cycles

Consider a graph $G = (V, E)$ representing the edges of a triangulated **planar** mesh, i.e. each node in V can be embedded in $\mathbb{R}^2$ with no edges crossing. Given a certain assignment of relative poses R on G, if all 3-cycles in G (i.e. all triangles in the mesh) are cycle-consistent, then any cycle γ in G is also cycle-consistent, i.e.:

$$R_\gamma = R_{\gamma_1,\gamma_2}R_{\gamma_2,\gamma_3}\cdots R_{\gamma_k,\gamma_1} = I \tag{49}$$

To prove this, note that a cycle γ is the boundary of an enclosed area $\mathcal{U}_\gamma \subset \mathbb{R}^2$ which includes a subset $V_\gamma \subset V$ of nodes and a subset $E_\gamma \subset E$ of edges (including those in γ). We prove that $R_\gamma = I$ *by induction* on the size of E_γ.

Base case $|E_\gamma| = 3$: this implies the cycle γ contains a single triangle and, therefore, it's already cycle-consistent by assumption, i.e. $R_\gamma = I$.

Inductive case $|E_\gamma| > 3$: consider any edge $(i, j) \in \gamma$. There exists a node $k \in V_\gamma$ such that (i, k) and $(j, k) \in E_\gamma$. Then, (i, j, k) for a cycle-consistent triangle. Now we consider three cases.

- If $(j, k) \in \gamma$, we can define a new cycle γ' by replacing both edges (i, j) and (j, k) with the edge (i, k). Note that $R_\gamma = R_{\gamma'}$ since $R_{ij}R_{jk} = R_{ik}$, but $E_{\gamma'} = E_\gamma \cup \{(i, j), (j, k)\}$ and, therefore, $E_{\gamma'} = E_\gamma - 2$.
- If $(i, k) \in \gamma$, one can apply a similar construction.
- Otherwise, we can define a new cycle γ' by replacing the edge (i, j) with the two edge (i, k) and (k, j). Note that $R_\gamma = R_{\gamma'}$ since $R_{ij} = R_{ik}R_{jk}^{-1}$, but $E_{\gamma'} = E_\gamma \cup \{(i, j)\}$ and, therefore, $E_{\gamma'} = E_\gamma - 1$.

Thus, the relative poses R are cycle-consistent on γ.

More generally, the set of all sufficient cycles for a generic graph is the *cycle basis* for that graph. When considering meshes over more general surfaces, the result proved above is not necessarily true, i.e. even if all triangles are cycle consistent, there might exist cycles which are not. These cycles can be precisely identified in the first homology group of the surface. For instance, a 2-torus has two additional cycles, but a similar proof works for a spherical surface.

F Performance under ideal synchronization

For many of the results, we are seeking a comparison for the empirical results. Without the precoding, the MIMO task would amount to reconstructing a signal under structured Gaussian noise. As such, we can characterize the Mean-Squared Error under 'ideal synchronization' in closed form. This Appendix section aims to characterize the MSE value so that we can plot it in the diagrams.

The 'ideal synchronization' MSE is instructive to have, but it is not a strict lower bound, however. Due to finite sample sizes, some reconstructions could achieve lower MSE. Conversely, due to finite sample sizes, the performance might not be attainable. Therefore, it shouldn't be considered a 'target performance' either. The latter is due to the interaction between rotation and signal noise: two signals could be noised such that the optimal rotation on a finite sample under some probabilistic model would not match the true rotation.

In general, the case where rotations are known could be modeled as the reconstruction of a correlated Gaussian distribution. Consider the following system. Matrix $\boldsymbol{X}$ is drawn from a correlated Gaussian distribution. After additive i.i.d. Gaussian noise on each of the elements in the matrix, a noisy matrix $\boldsymbol{Y}$ is observed. Then, we aim to characterize the reconstruction error $\mathbb{E}[\|X - \hat{\boldsymbol{X}}(\boldsymbol{Y})\|^2]$.

$$\boldsymbol{X} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_X); \qquad \epsilon \sim \mathcal{N}(0, \boldsymbol{I}); \qquad \boldsymbol{Y} = \boldsymbol{X} + \sigma\epsilon \tag{50}$$

We are interested in the expected MSE of estimating $\boldsymbol{X}$ from $\boldsymbol{Y}$, as a function of $\boldsymbol{\Sigma}_X$ and σ.

$$\mathcal{L}(\boldsymbol{\Sigma}_X, \sigma) = \mathbb{E}[\|X - \hat{\boldsymbol{X}}(\boldsymbol{Y})\|^2] = \mathbb{E}[X^T X + \hat{\boldsymbol{X}}(\boldsymbol{Y})^T \hat{\boldsymbol{X}}(\boldsymbol{Y}) - 2 X^T \hat{\boldsymbol{X}}(\boldsymbol{Y})] \tag{51}$$

The covariance matrix $\boldsymbol{\Sigma}_X$ is of size ND by ND elements, which is the covariance between all measurements in all blocks. The optimal estimator is the MMSE estimator $hat\boldsymbol{X}(\boldsymbol{Y}) = \mathbb{E}[X|Y]$. This follows from the conditional distribution: $p(X|Y) \propto p(Y|X)p(X) \propto \mathcal{N}(\mu_c, \boldsymbol{\Sigma}_c)$.

This follows from estimating X with the MMSE estimator:

$$\boldsymbol{\Sigma}_c = \left(\boldsymbol{\Sigma}_X^{-1} + \frac{1}{\sigma^2}\boldsymbol{I}\right)^{-1} = \sigma^2 \left(\sigma^2 \boldsymbol{\Sigma}_X^{-1} + \boldsymbol{I}\right)^{-1} \tag{52}$$

$$\boldsymbol{\mu}_c = \boldsymbol{\Sigma}_c \left(\frac{1}{\sigma^2}\boldsymbol{Y}\right) \tag{53}$$

$$\hat{\boldsymbol{X}}(\boldsymbol{Y}) = \left(\boldsymbol{\Sigma}_X^{-1} + \frac{1}{\sigma^2}\boldsymbol{I}\right)^{-1} \left(\frac{1}{\sigma^2}\boldsymbol{Y}\right) = \left(\sigma^2 \boldsymbol{\Sigma}_X^{-1} + \boldsymbol{I}\right)^{-1} \boldsymbol{Y} \tag{54}$$

For the derivation, we will need the second moment matrices: $\mathbb{E}[\boldsymbol{X}\boldsymbol{X}^T] = \boldsymbol{\Sigma}_X$, $\mathbb{E}[\boldsymbol{Y}\boldsymbol{Y}^T] = \boldsymbol{\Sigma}_X + \sigma^2 \boldsymbol{I}$, $\mathbb{E}[\boldsymbol{Y}\boldsymbol{X}^T] = \boldsymbol{\Sigma}_X$.

Linear MMSE The covariance of the linear MMSE estimator is given by $\boldsymbol{\Sigma}_X - \boldsymbol{\Sigma}_{XY}\boldsymbol{\Sigma}_Y^{-1}\boldsymbol{\Sigma}_{YX}$. This follows from searching over the space of linear estimators: $\hat{\boldsymbol{X}}(\boldsymbol{Y}) = \boldsymbol{W}\boldsymbol{Y} + b$. The minimal error is achieved when $\boldsymbol{W} = \boldsymbol{\Sigma}_{XY}\boldsymbol{\Sigma}_Y^{-1}$. Therefore, the MMSE is

$$\hat{\mathcal{L}}(\boldsymbol{\Sigma}_X, \sigma) = \boldsymbol{\Sigma}_X - \boldsymbol{\Sigma}_X(\Sigma_X + \sigma^2 I)^{-1}\boldsymbol{\Sigma}_X \tag{55}$$

Equation 55 can also be used to characterize the performance of the per-block MSE. That is, the performance of considering any rotations at all and treating each PRG block in isolation. In that case, $\boldsymbol{\Sigma}_X \in \mathbb{R}^{D \times D}$ is the correlated matrix of all measurements within one cell, instead of $\boldsymbol{\Sigma}_X \in \mathbb{R}^{JD \times JD}$.