

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/180199/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Ning, Zekun, Pullin, Rhys and Ji, Ze 2025. GBP-LOAM: Lidar odometry and mapping using Gaussian Belief Propagation. Presented at: The 30th International Conference on Automation and Computing (ICAC 2025), Loughborough, UK, 27-29 August 2025. Proceedings of the 30th International Conference on Automation and Computing. IEEE, 10.1109/ICAC65379.2025.11196354

Publishers page: <https://doi.org/10.1109/ICAC65379.2025.11196354>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See <http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



GBP-LOAM: Lidar Odometry and Mapping using Gaussian Belief Propagation

Zekun Ning, Rhys Pullin, and Ze Ji

School of Engineering, Cardiff University, Cardiff, CF24 3AA, UK

Abstract—In Simultaneous Localisation and Mapping (SLAM), factor graphs represent one of the most mature back-end optimisation frameworks. However, most methods are centralised and constrained by computational resources and efficiency. Gaussian Belief Propagation (GBP) is a distributed probabilistic inference method for approximating the probability distribution of a system that can be represented by a Factor Graph. We hypothesise that GBP-based solutions could provide a distributed solution that can efficiently tackle the distributed SLAM challenges. However, there is no existing SLAM framework available based on the GBP solution. As a pilot exploration, this work introduces a new GBP-based 3D SLAM framework, demonstrating the feasibility of GBP-based SLAM in 3D. We construct a factor graph on the back-end of a Lidar Odometry and Mapping (LOAM) pipeline and execute GBP for system optimisation, validating the potential and stability of GBP in SLAM. Our algorithm is evaluated on public datasets and benchmarked against state-of-the-art centralised solvers. The results demonstrate that the GBP-based distributed solver attains accuracy comparable to centralised approaches. However, achieving robust convergence on the SE(3) Lie-group state space remains challenging. Finally, we analyse the convergence of GBP and show that incorporating damping factors and regularisation offers a reliable means to enhance convergence. This work provides valuable insights and foundations for future exploration into multi-robot collaborative SLAM implementations.

Index Terms—SLAM, Lidar Odometry, Gaussian Belief Propagation, State Estimation, Factor Graph

I. INTRODUCTION

Simultaneous Localisation and Mapping (SLAM) remains a fundamental challenge in robotics, essential for autonomous navigation in complex and unstructured environments. Lidar-based odometry and mapping algorithms, such as Lidar Odometry and Mapping (LOAM) [1], Cartographer [2] and Lego-LOAM [3], have demonstrated remarkable performance by leveraging geometric features for pose estimation and map construction. To enhance the efficiency and broaden the applicability of SLAM, particularly in scenarios requiring rapid area coverage, such as rescue operations, multi-robot collaborative SLAM has garnered significant research focus in recent years. Current mainstream multi-robot SLAM frameworks predominantly rely on centralised optimisation strategies. Despite their stability and robustness, these centralised approaches often suffer from communication latency, and susceptibility to single-point failures in centralised nodes, limiting their scalability for deployments.

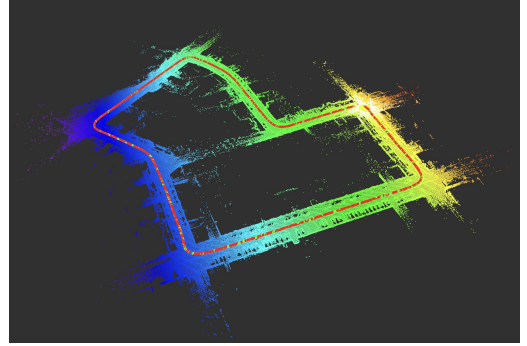


Fig. 1. The performance of GBP-based LOAM in KITTI dataset

Gaussian Belief Propagation (GBP), as an efficient distributed probabilistic inference method, has demonstrated considerable potential across various domains, such as wireless sensor networks, distributed state estimation [4], and multi-robot relative localisation [5]. GBP operates through local message passing between variables and factor nodes to perform global state estimation, inherently supporting distributed computation and eliminating the necessity of central nodes, thereby mitigating computational and communication constraints. Despite the promising potential of GBP in multi-robot SLAM, existing studies predominantly focus on simplified 2D scenarios or simulation environments [6]. The application of GBP to more complex real-world 3D SLAM scenarios, particularly involving non-linear optimisation on the SE(3) Lie Group, remains largely unexplored and inadequately validated.

Therefore, as the first step, this work aims to implement and thoroughly evaluate GBP within a single-robot context operating Lidar-SLAM on the SE(3) state space, providing insights into its convergence properties, computational efficiency, robustness, and accuracy performance. Specifically, we integrate GBP into the back-end optimisation module of a Lidar Odometry and Mapping (LOAM) system, performing extensive experimental evaluations using publicly available datasets. This allows us to systematically compare GBP against traditional centralised optimisation methods, such as GTSAM [7], highlighting both its strengths and limitations. The outcomes of this preliminary research form a solid theoretical foundation and practical insights for extending GBP to genuine multi-robot collaborative SLAM systems in future studies. The primary contributions of this paper are:

- Proposal of a novel GBP-based Lidar Odometry and Mapping system (GBP-LOAM), introducing a distributed

probabilistic inference framework for 3D SLAM.

- Comprehensive evaluation of the convergence properties, accuracy, and computational efficiency of GBP within 3D SLAM scenarios using the KITTI dataset, demonstrating comparable accuracy to traditional centralised methods.
- In-depth comparison with traditional centralised solvers, providing a detailed analysis of performance characteristics and applicability, showing that achieving convergence in complex systems remains a significant challenge, thus informing future developments in multi-robot collaborative SLAM.

The remainder of this paper is organised as follows. Section II reviews related work on Lidar odometry and GBP. Section III details the proposed GBP-based back-end optimisation framework. Section IV presents experimental results and analysis. Finally, Section V concludes the paper and discusses future directions.

II. BACKGROUND AND RELATED WORK

Traditional centralised optimisation frameworks such as Gauss-Newton [8] and Levenberg-Marquardt [9] algorithms have been extensively used in single-robot Lidar-based SLAM systems due to their robust performance in pose estimation and environmental mapping. LOAM operates by extracting edge and plan features from Lidar scans to estimate pose and build point cloud maps. To enhance computational efficiency and adaptability, several variants of LOAM have been proposed. For instance, Light-LOAM [10] introduces a graph-based point cloud association strategy to achieve reliable SLAM performance on computation-limited platforms like quadrotor UAVs. Similarly, SC-F-LOAM [11] utilises GTSAM [7] for global optimisation, aiming to improve pose estimation accuracy in large-scale environments. Despite these advantages, centralised methods inherently suffer from scalability constraints, computational bottlenecks, and susceptibility to single-point failures, limiting their efficacy in multi-robot or large-scale scenarios.

Bickson et al. [12] introduced Gaussian Belief Propagation (GBP), which gained significant attention as a powerful distributed inference approach. In GBP, variables and factors within the graph exchange information exclusively with their immediate neighbours, propagating messages asynchronously to achieve globally consistent estimates without requiring comprehensive historical data storage or processing. This locality method provides a highly adaptable solution, capable of efficiently addressing various linear and non-linear Gaussian measurement problems. GBP can be applied in many different ways, such as linear detection [13] and the distributed Kalman filter [14]. Meanwhile, the decentralised nature of GBP inherently supports scalability and robustness, making it particularly suitable for distributed multi-robot SLAM.

Recent research has explored the application of GBP in SLAM and pose estimation. For example, Murai et al. [4] proposed a fully distributed and online method for simultaneous localisation and extrinsic calibration in multi-robot systems using GBP, demonstrating its scalability and robustness in challenging conditions. A GBP framework, named Robot Web,

was introduced to demonstrate the feasibility of using GBP for distributed localisation with relative positioning among over 50 robots in 2D environments [5], reinforcing its substantial potential for multi-robot distributed systems. Additionally, the Hyperion framework [15] implements a symbolic GBP approach for continuous-time SLAM, achieving significant speed-ups over traditional methods and facilitating decentralised probabilistic inference across agents.

However, despite these promising developments, the application of GBP in practical Lidar-based 3D SLAM systems, particularly involving complex non-linear optimisation on SE(3) state spaces, remains relatively unexplored. This study aims to bridge this gap by systematically evaluating GBP's capabilities within a single-robot Lidar SLAM context, comparing its performance against conventional centralised optimisation methods. Through comprehensive experiments conducted on the KITTI dataset [16], we investigate the convergence behaviour, accuracy, and computational efficiency, thereby laying the critical groundwork for its future application in multi-robot collaborative SLAM systems.

III. METHOD

A. Gaussian Belief Propagation for state estimation

In this section, we will introduce the GBP in detail. The core idea of this approach is to compute the marginal posterior distribution for each variable, utilising only the set of factors that comprise the joint posterior. The key advantage is that each node requires only knowledge of itself and its immediate neighbours to perform probability inference [7]. GBP functions are updated through an iterative process, where messages are sent and received among neighbouring nodes to continuously update each node's locally stored belief [6].

Each iteration involves three key stages: belief updated, variable messages sent to factors, and factor messages sent to variables, as shown in Fig. 2.

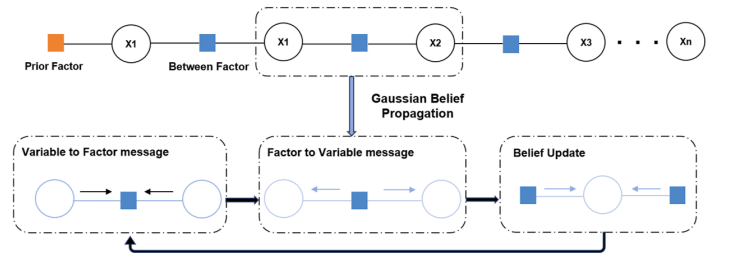


Fig. 2. The process of GBP within a factor graph optimisation framework. It depicts the interaction between factors (prior factor and between factor) and the corresponding variables (robot poses). Through iterative message passing, both variable-to-factor and factor-to-variable messages are exchanged to refine the beliefs regarding the state of each variable.

The core of GBP is to solve the classic least-squares problem, $Ax - b = 0$. In order to transfer the problem of solving linear systems from the algebraic domain to the probabilistic domain, we define the quadratic form given by $q(x) = x^T Ax/2 - b^T x$, so the joint Gaussian probability density function is:

$$p(x) \triangleq \mathcal{Z}^{-1} \exp(-q(x)) = \mathcal{Z}^{-1} \exp(-x^T Ax/2 + b^T x) \quad (1)$$

where the $\mathcal{Z}$ is a distribution normalization factor, and the mean vector of the standard Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ is $\mu = A^{-1}b$, $\Sigma = A^{-1}$. Here, when doing probabilistic inference with GBP, we use the information Gaussian distribution form like equation 1 because it is simple to take products, so it is easy to construct the posterior probability from the factors [6], which is a significant advantage in the cumbersome message passing in GBP. We define the precision matrix as $P = A = \Sigma^{-1}$, and the information vector as $\eta = b = \Sigma^{-1}\mu$.

A factor graph is a bipartite graph model that uses 1) factor nodes to represent local functions and 2) variable nodes to represent random variables. It can intuitively display the decomposition structure of global functions or joint probability distributions [7]. Therefore, in the factor graph, the information form of the factor negative log-likelihood (energy function) is given by

$$-\log p(x) \propto E(x) = \frac{1}{2} x^\top P_f x - \eta_f^\top x \quad (2)$$

where the P_f is the precision matrix of the Gaussian factors, and η_f is the information vector of factors.

1) *Factor Linearization*: The Lie theory is widely used as a fundamental component in modern state estimation research [17]. In our work, we define the state, the robot pose, using the SE(3) Lie group, where the state at time t can be represented by $x_t = T_t \in \{R_t, p_t\} \in SE(3)$, where R_t denotes the rotation and p_t is the translation.

Given a non-linear measurement function $h(x)$ and an observation z , we use $\boxminus$ and $\boxplus$ to denote the generalised subtraction and addition operators on the Lie group. Therefore, the residual is given by $r(x) = z \boxminus h(x)$. We now linearise the residual around the linearization point x_0 . Using a first-order Taylor expansion:

$$r(x) \approx r(x_0) + \left. \frac{\partial r(x)}{\partial x} \right|_{x_0} (x \boxminus x_0) = r(x_0) + J \delta x \quad (3)$$

Combine the equation 2 and we finally get the factor potential energy of the non-linear measurement function as:

$$\eta_f = -J^\top P r(x_0), \quad P_f = J^\top P J \quad (4)$$

where the J is the Jacobian matrix and the P is the measurement information matrix. We find that after linearising each factor, the precision vector η_f and matrix P_f assume the same form as the standard equations in Gauss-Newton. However, the classical Gauss-Newton method computes the increment for the entire system. In contrast, GBP operates on local factors in a distributed way, yielding a local approximate Hessian matrix P_f .

In order to solve the incremental update equations, P_f must be invertible. However, the computed P_f is only positive semidefinite in SLAM, and often becomes singular or ill-conditioned, leading to poor increment stability and potential non-convergence. Moreover, as our state space lies on a Lie group SE(3), the objective function is inherently non-linear and non-convex. Since GBP does not have access to the global cost function, the convergence cannot be guaranteed. To address these issues, we follow the approach of [4] and introduce a regularisation term λ_t into P_f . This transforms our

solver into a distributed Levenberg–Marquardt algorithm [9], as below:

$$P_f = J^\top P J + \lambda_t I \quad (5)$$

$$\lambda_t = \begin{cases} \lambda_{t-1} \cdot \lambda_i & \text{if } E^t - E^{t-1} > \epsilon \\ \lambda_{t-1} / \lambda_d, & \text{otherwise} \end{cases} \quad (6)$$

where E^t is the factor energy at time t , ϵ is a predefined threshold and λ_i , λ_d are the coefficients of increase and decrease respectively. This scheme adaptively adjusts λ_t in order to mitigate singularity and ill-conditioning in the coefficient matrix, and hence obtains a more stable and robust optimiser.

2) *Belief Update at the Variable Nodes*: The belief of each node is determined by computing the product of all incoming messages from neighbouring factors:

$$b(x_i) = \prod_{f \in n(x_i)} m_{f \rightarrow x_i} \quad (7)$$

where $b(x_i)$ is the belief of variable x_i , the neighbouring factor nodes are $f_s \in n(x_i)$. Each message $m_{f_s \rightarrow x_i}$ carries information from a connected factor.

We add up the incoming messages to solve the final total precision matrix and information vector. In GBP of the Lie group, the factor's information matrix is transported into the state's tangent space via right Jacobian J_r , and then this tangent space information matrix is applied to the local manifold error to form the natural parameter vector:

$$P_{\mathfrak{se}(3)_i} = J_r^\top(x_i \boxminus x_0) P_i J_r(x_i \boxminus x_0) \quad (8)$$

$$\eta_i = P_{\mathfrak{se}(3)_i}(x_i \boxminus x_0) \quad (9)$$

All incoming information vectors are summed and weighted to recover each node's update:

$$P_{\mathfrak{se}(3)} = \sum_N P_{\mathfrak{se}(3)_i} \quad (10)$$

$$\delta x_i = P_{\mathfrak{se}(3)}^{-1} \sum_N \eta_i \quad (11)$$

Then retracts back onto SE(3) using the exponential map, and the variable node updates its state:

$$x_i^* = x_0 \boxplus \delta x_i \quad (12)$$

$$P_i^* = J_r^{-T}(\delta x_i) P_{\mathfrak{se}(3)} J_r^{-1}(\delta x_i) \quad (13)$$

3) *Message Passing from Variable to Factor Nodes*: The message from the variable x_i to the factor f_j is defined as the product of all incoming messages from connected factor nodes of x_i except the message from f_j .

$$m_{x_i \rightarrow f_j} = \prod_{s \in N(i) \setminus j} m_{f_s \rightarrow x_i} \quad (14)$$

We can combine the messages from factors and minus the own message to obtain the tangent vector of the outgoing messages.

$$P_a = \sum_N P_{\mathfrak{se}(3)_i} - P_j, \eta_a = \sum_N \eta_i - \eta_j \quad (15)$$

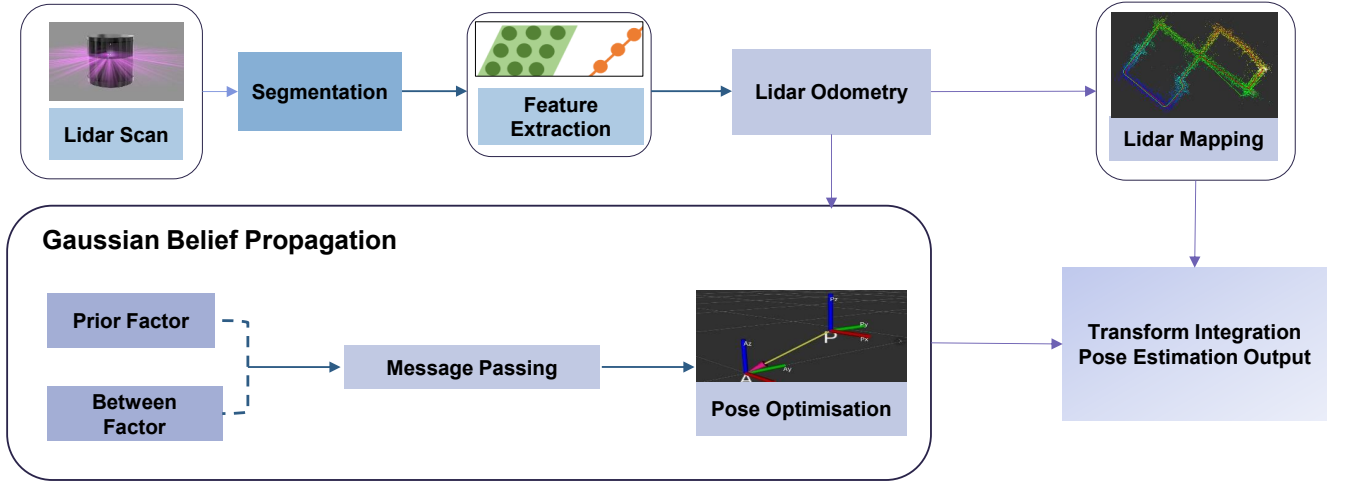


Fig. 3. The overview of GBP-LOAM System

Next, this tangent vector is retracted from the identity pose onto the Lie Group to get the message increment:

$$\delta x_a = P_a^{-1} \eta_a \quad (16)$$

Finally, the factor's total precision matrix is transformed into the Lie Group at this new state by warping it with the J_r , thereby the message (x_a^*, P_a^*) will be sent to factor f_j :

$$x_a^* = x_0 \boxplus \delta x_a \quad (17)$$

$$P_a^* = J_r^{-T}(\delta x_a) P_a J_r^{-1}(\delta x_a) \quad (18)$$

4) *Message Passing from Factor to Variable Nodes:* The message sent from a factor f_j to one of its neighbouring variable nodes x_i is just the “partial” marginal of the joint Gaussian defined by the factor and all other incoming messages into that factor, integrated out over all variables except x_i .

$$m_{f_j \rightarrow x_i} = \sum_{X_j \setminus x_i} f_j(X_j) \prod_{k \in N(j) \setminus i} m_{x_k \rightarrow f_j} \quad (19)$$

When a factor node receives messages from its neighbouring variable nodes, we first update its precision vector η_f and precision matrix P_f by adding the incoming messages in the corresponding dimensions to update the factor.

$$P_{s\epsilon(3)_j} = J_r^T(x_j \boxminus x_o) P_j J_r(x_j \boxminus x_o) \quad (20)$$

$$\eta_j = P_{s\epsilon(3)_j}(x_j \boxminus x_o) \quad (21)$$

$$[P^{(f)}]_{jj} = [P^f]_{jj} + P_{s\epsilon(3)_j}, [\eta^{(f)}]_{jj} = [\eta^f]_{jj} + \eta_j \quad (22)$$

Then, during the message-sending stage, we remove the message associated with the target variable and marginalise out all other variables via the Schur complement [18], thereby incorporating their influence into the outgoing message directed to the target variable. Accordingly, partition the factor's information as

$$P_f = \begin{pmatrix} P_{\alpha\alpha} & P_{\alpha\beta} \\ P_{\beta\alpha} & P_{\beta\beta} \end{pmatrix}, \quad \eta_f = \begin{pmatrix} \eta_\alpha \\ \eta_\beta \end{pmatrix}, \quad (23)$$

where the block α corresponds to the target variable x_i and β corresponds to the other variables.

$$P_M = P_{\alpha\alpha} - P_{\alpha\beta} P_{\beta\beta}^{-1} P_{\beta\alpha}, \quad (24)$$

$$\eta_M = \eta_\alpha - P_{\alpha\beta} P_{\beta\beta}^{-1} \eta_\beta. \quad (25)$$

Finally, we convert the information matrix in the tangent space back to the Lie group space to obtain the factor-to-variable messages (x_j^*, P_j^*) :

$$x_j^* = x_0 \boxplus P_M^{-1} \eta_M \quad (26)$$

$$P_j^* = J_r^{-T}(P_M^{-1} \eta_M) P_M J_r^{-1}(P_M^{-1} \eta_M) \quad (27)$$

This completes our detailed derivation of incremental GBP from the local linearisation of the factor to the final state update. After iteration, this belief will gradually converge to the marginal probability of the variable node itself.

B. System Overview

The GBP-LOAM system is shown in Fig. 3, the system consists of two principal stages: real-time Lidar odometry and mapping and global pose optimisation via GBP. First, each incoming Lidar scan is segmented to remove ground and dynamic objects and extracts the edge and plan type features [19]. Then, the system computes relative-pose increments between consecutive frames by matching these features, and then integrates those increments into a local map to drive real-time Lidar odometry. Next, it will assemble all inter-frame relative-pose constraints alongside the prior pose constraints into a factor graph. GBP is applied to this graph to iteratively pass and fuse Gaussian parameters across nodes. Finally, the optimal pose of each node is obtained from its posterior distribution, resulting in a globally consistent trajectory that is fused with the local map to produce a dense 3D environment reconstruction.

1) *Prior Factor:* The prior factor removes the inherent “gauge freedom” by anchoring a selected variable to a known pose via a Gaussian constraint:

$$r_i(x_i) = \text{Log}((x_0)^{-1} x_i) \quad (28)$$

2) *Between Factor*: For robot odometry, it unifies the Lidar relative measurement into a Gaussian noise constraint between two pose nodes on the graph, thereby effectively suppressing the cumulative drift. The binary factor with the measurement z represents a relative pose measurement:

$$r(x_i, x_j) = \text{Log}(z_{ij}^{-1}(x_i^{-1}x_j)) \quad (29)$$

IV. EVALUATION

In this section, we conduct both qualitative and quantitative analyses of two competitive methods—the centralised solver GTSAM [20] and our distributed solver GBP—on identical hardware (a laptop equipped with an Intel i5-12600KF). We implement a pure Lidar-based SLAM algorithm with each solver and test the algorithm in the KITTI dataset [16], both written in C++ and running under the Robot Operating System (ROS) on Ubuntu Linux.

A. Experimental Result

To quantitatively evaluate the accuracy of the SLAM system and facilitate comparison with other approaches, we employ the Absolute Trajectory Error (ATE) [21] and Relative Pose Error (RPE) metrics. The ATE measures the deviation between the trajectories estimated by our system and the corresponding ground truth trajectories, whereas the RPE quantifies the local consistency and drift of the estimated motion between keyframes. We conducted extensive experiments on the KITTI dataset. Fig. 4 shows the performance of GBP on Sequence 07 of the KITTI dataset. Since the primary objective of this work is to validate the application of GBP in SLAM, we employed a very simple front-end. Consequently, the overall accuracy is significantly lower than LeGo-LOAM [3] that of the mature LOAM framework (see Table I). Although the GBP-based LOAM exhibits drift in later stages, the overall trajectory remains consistent with the ground truth. In the linear system, GBP can converge to an accurate result with a single message round trip [22]. In non-linear optimisation, GBP needs to be iterated continuously to approach the optimal estimate. In this experiment, we found that in each round of iteration, convergence can be achieved after about 5 message transmissions. At the same time, each round of iteration takes about 5 ~ 10 ms.

To compare with a centralised optimiser, we use the same front-end and data set for multiple experiments with GBP and GTSAM, respectively. The results are summarised in Table I and Table II. It can be easily observed that our framework achieves accuracy comparable to that of the centralised solver. The principal discrepancy arises in Sequence 06, which is primarily attributable to the loop-closure detection incorporated into GTSAM via a between-factor to correct accumulated errors. However, the loop closure module introduces loops into the factor graph in GBP. During message passing, identical messages are reused, leading to “overconfidence” and adversely affecting convergence.

In summary, our GBP framework achieves accuracy in 3D SLAM comparable to that of centralised solvers. Moreover, its native support for distributed operation gives it advantages

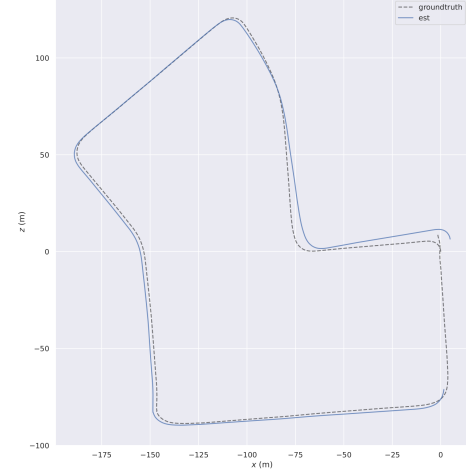


Fig. 4. Estimated trajectories and Ground truth for the KITTI 07 sequence.

in multi-robot SLAM scenarios that centralised solvers cannot offer, making it a highly promising algorithm.

TABLE I
THE RMSE ATE ON KITTI ODOMETRY DATASET

Sequence	04	06	07	10	Average
GBP-	6.90/0.15	9.05/0.63	4.88/0.33	7.79/0.81	7.15/0.48
GTSAM-	6.67/0.13	6.52/0.82	4.35/0.21	7.80/0.73	6.33/0.47
LeGo-LOAM [3]	0.55/0.02	0.96/0.01	1.11/0.02	2.39/0.02	1.25/0.017

The metric is represented in the ‘A/B’ format, where ‘A’ denotes the translation error (meters), and ‘B’ represents the rotation error (radians).

TABLE II
THE RMSE RPE ON KITTI ODOMETRY DATASET

Sequence	04	06	07	10	Average
GBP-	2.00/0.32	1.79/0.93	1.58/0.86	1.46/0.37	1.71/0.62
GTSAM-	1.91/0.29	1.85/0.82	1.56/0.21	1.81/0.41	1.78/0.43

The metric is represented in the ‘A/B’ format, where ‘A’ denotes the translation error (meters), and ‘B’ represents the rotation error (radians).

B. Convergence Analysis

During algorithm experiments, maintaining convergence in GBP proved challenging; we observed unexpected divergence at specific frames, with the error growing exponentially and causing severe trajectory drift. The main reason is that GBP requires the information matrix $P = D - R$ to satisfy the walk-summability condition

$$\rho(|D^{-1}R|) < 1 \quad (30)$$

where D is the block-diagonal component of the information matrix, retaining only each variable’s own information matrix, and R consists of the off-diagonal components, containing the coupling information between different variables. The equivalent to P being strictly diagonally dominant under some diagonal scaling [23].

In SLAM back-ends, loop closure constraints and marginalisation of landmarks or past poses introduce strong, long-range

couplings and densely populated off-diagonal blocks, while the inherent rotation–translation dimension mismatch in SE(3) and the arbitrary weighting of measurements further amplifies these non-diagonal terms, breaking diagonal dominance and causing GBP updates to oscillate or diverge. GBP needs to relinearise the non-linear SE(3) error function at each iteration, and the combination of tight rotation–translation coupling leaves the normalised off-diagonal spectral radius $\rho \geq 1$, preventing convergence even on a simple odometry chain [20].

Therefore, we use the following damping factor method, to appropriately smooth the message update process in each round of message passing to avoid updating messages too quickly in some cases, which may cause system divergence or oscillation.

$$\hat{m}_{i \rightarrow j} = \beta \cdot m_{i \rightarrow j}^{(new)} + (1 - \beta) \cdot m_{i \rightarrow j}^{(old)} \quad (31)$$

At the same time, combined with the regularisation method mentioned in III.A part, we can get relatively robust convergence. After comprehensive experiments on the KITTI dataset, when $\beta = 0.3 \sim 0.5$, $\lambda_t = 10$, $\lambda_i = 15$, $\lambda_d = 5$ in Equation (6), we obtained better convergence results. However, this does not imply that the convergence issue is resolved, because, in this system, the factor graph uses an extremely simple structure. In more complex scenarios, convergence still poses one of the biggest bottlenecks in GBP and demands further research.

V. DISCUSSION AND CONCLUSION

In this paper, we introduced a new SLAM framework based on the Gaussian Belief Propagation (GBP). A Lidar odometry method is realised using this framework for performance evaluation and validation. Unlike centralised optimisation techniques, our approach performs probabilistic inference directly on the factor graph using GBP. We validate the feasibility of GBP for 3D SLAM experiments using public datasets and the results show that our algorithm achieves accuracy on par with centralised solvers. Furthermore, owing to the inherently distributed nature of GBP, our method holds significant promise for distributed systems, like multi-robot cooperative SLAM. Nonetheless, GBP still faces convergence challenges, particularly in loop graphs and on the Lie group state spaces such as SE(3). Our experiments indicate that the use of damping factors and regularisation can improve stability, but achieving reliable convergence while maintaining high accuracy remains a key challenge. Therefore, future work will focus on the development of distributed multi-robot SLAM algorithms, the integration of additional sensors such as IMU, camera, and UWB into the GBP framework for robust multi-robot state estimation, and the enhancement of GBP’s robustness to ensure both accurate results and efficient convergence under increased system complexity.

REFERENCES

[1] J. Zhang, S. Singh *et al.*, “Loam: Lidar odometry and mapping in real-time,” in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[2] A. Dwijotomo, M. A. Abdul Rahman, M. H. Mohammed Ariff, H. Zamzuri, and W. M. H. Wan Azree, “Cartographer slam method for optimization with an adaptive multi-distance scan scheduler,” *Applied Sciences*, vol. 10, no. 1, p. 347, 2020.

[3] T. Shan and B. Englot, “Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.

[4] R. Murai, I. Alzugaray, P. H. Kelly, and A. J. Davison, “Distributed simultaneous localisation and auto-calibration using gaussian belief propagation,” *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2136–2143, 2024.

[5] R. Murai, J. Ortiz, S. Saeedi, P. H. Kelly, and A. J. Davison, “A robot web for distributed many-device localisation,” *IEEE Transactions on Robotics*, 2023.

[6] J. Ortiz, T. Evans, and A. J. Davison, “A visual introduction to gaussian belief propagation,” *arXiv preprint arXiv:2107.02308*, 2021.

[7] F. Dellaert, “Factor graphs and gtsam: A hands-on introduction,” *Georgia Institute of Technology, Tech. Rep.*, vol. 2, p. 4, 2012.

[8] Y. Wang, “Gauss–newton method,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, no. 4, pp. 415–420, 2012.

[9] A. Ranganathan, “The levenberg-marquardt algorithm,” *Tutorial on LM algorithm*, vol. 11, no. 1, pp. 101–110, 2004.

[10] S. Yi, Y. Lyu, L. Hua, Q. Pan, and C. Zhao, “Light-loam: A lightweight lidar odometry and mapping based on graph-matching,” *IEEE Robotics and Automation Letters*, 2024.

[11] L. Liao, C. Fu, B. Feng, and T. Su, “Optimized sc-f-loam: Optimized fast lidar odometry and mapping using scan context,” in *2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI)*. IEEE, 2022, pp. 1–6.

[12] D. Bickson, “Gaussian belief propagation: Theory and application,” *arXiv preprint arXiv:0811.2518*, 2008.

[13] D. Bickson, O. Shental, P. Siegel, J. Wolf, and D. Dolev, “Linear detection via belief propagation,” in *Proc. 45th Allerton Conf. on Communications, Control and Computing*, 2007.

[14] D. Bickson, O. Shental, and D. Dolev, “Distributed kalman filter via gaussian belief propagation,” in *2008 46th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2008, pp. 628–635.

[15] D. Hug, I. Alzugaray, and M. Chli, “Hyperion—a fast, versatile symbolic gaussian belief propagation framework for continuous-time slam,” in *European Conference on Computer Vision*. Springer, 2024, pp. 215–231.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The international journal of robotics research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[17] J. Sola, J. Deray, and D. Atchuthan, “A micro lie theory for state estimation in robotics,” *arXiv preprint arXiv:1812.01537*, 2018.

[18] R. M. Eustice, H. Singh, and J. J. Leonard, “Exactly sparse delayed-state filters,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 2417–2424.

[19] H. Guo, J. Zhu, and Y. Chen, “E-loam: Lidar odometry and mapping with expanded local structural information,” *IEEE transactions on intelligent vehicles*, vol. 8, no. 2, pp. 1911–1921, 2022.

[20] F. Dellaert, M. Kaess *et al.*, “Factor graphs for robot perception,” *Foundations and Trends® in Robotics*, vol. 6, no. 1–2, pp. 1–139, 2017.

[21] Z. Zhang and D. Scaramuzza, “A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7244–7251.

[22] A. J. Davison and J. Ortiz, “Futuremapping 2: Gaussian belief propagation for spatial ai,” *arXiv preprint arXiv:1910.14139*, 2019.

[23] M. J. Wainwright, M. I. Jordan *et al.*, “Graphical models, exponential families, and variational inference,” *Foundations and Trends® in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.