Gaussian Belief Propagation for Distributed Swarm Sensing

Simon Jones and Sabine Hauert

***Abstract*— Gaussian Belief Propagation (GBP) shows great potential as a general distributed knowledge inference algorithm for use within swarms of robots. Individual robots build local factor graphs describing the environment, continually running GBP within their fragment to infer state. Robots coming within communication range of each other exchange information that connects these graph fragments and widens perception of each agent. The swarm in total holds an adaptive distributed situational awareness of the environment.**

I. INTRODUCTION

Swarm robotics, inspired by swarms in nature, has the potential for resilient, robust, and redundant solutions to a wide range of problems such as mapping, logistics, search and rescue, disaster recovery, and environmental monitoring. Many relatively simple and cheap robots, each following simple rules, with local interactions between themselves and the environment are capable of producing a desired emergent swarm-level behaviour [1].

Gaussian Belief Propagation [2], [3] is an algorithm for iteratively solving the probabilistic state estimation problem represented by a factor graph [4]. By passing messages between nodes of the graph, based purely on local computation and knowledge, state variables within the graph converge on estimates that closely match those produced by centralised solvers.

The distributed paradigm of GBP is a natural fit with that of swarm robotics, and we intend to use it to realise our vision of Distributed Situational Awareness for swarms of robots [5]. By giving swarms an enhanced awareness of the world, we can enable simple and robust swarm algorithms to approach the performance of centrally planned solutions in logistics applications.

Initial work focussed on applying GBP focussed on more typical uses of centralised factor graph solvers such as localisation [6] and bundle adjustment [7], albeit exploiting the distributed nature of the algorithm to run on the massively multicore Graphcore architecture. Other work, e.g [8] demonstrating distributed cooperative trajectory planning showed that it is possible to pose many different types of problem as optimisation on a factor graph.

[1]Simon Jones and Sabine Hauert are with the Department of Engineering Mathematics, University of Bristol, UK. simon2.jones@bristol.ac.uk, sabine.hauert@bristol.ac.uk

II. SWARM WAREHOUSE

As a use case, we focus on robot swarms used for intralogistics. Specifically, we consider the pop-up automated cloakroom scenario described in [5]. A small company running conferences wants to automate their cloakroom. A conventional approach would have sophisticated robots to map the allocated area, centralised control, infrastructure. An alternative vision is that the company would simply delimit an area of floor and add robots and small carriers capable of holding belongings out-of-the-box. Users at the event would download an app to their phone and use this to call for a carrier to deposit an item. Via Bluetooth, any robot within range would respond and provide the carrier and take it and the item to be stored. To retrieve the item, the user would use the app again, robots would talk with their neighbours until a robot with recent knowledge of the item heard, which would pick up the carrier and take it to the user. Even very simple random walk algorithms are capable of effective retrieval in logistics applications [9], [10]. Messages from such users propagate from robot to robot, robots store and retrieve items, all in parallel and without central resources.

Moving this vision into reality, we have built a swarm of 20 fast-moving robots and instrumented arena, called the DOTS (Distributed Organisation and Transport System) [5], shown in Figure 1. We have already demonstrated autonomous distributed logistics retrieval[1] and intend to use this platform to demonstrate a complete distributed storage and retrieval task, using GBP as an enabling technology.

In this scenario, we focus on the location and the carried state of a carrier. The location has obvious utility; if a user has requested a given unique carrier (containing their item), being able to move there quickly helps retrieval speed. The carried state of the carrier is more interesting. When a robot picks up a carrier, it is no longer available for another robot, in fact managing the contention between multiple resources in a distributed manner using GBP is, we feel, key to making the swarm warehouse work well. It would be possible to handle conflict in other ways, but the elegance of a unifying framework appeals to us.

III. Methods

The task takes place in a 3.7x3.7m arena, with up to 10 250mm diameter robots, and multiple 330mm square carriers. The robots are equipped with a lifting platform and four cameras capable of identifying the relative pose of both carriers and other robots, when within a certain range. Robots can communicate with each other, mediated though a synthetic communication model to limit the range. Robot movement is fully holonomic.

A. Simulator

To simplify initial explorations, we assume a pure 2D representation of pose. This allows the use of a fully linear approach, as described in Section 4.2 of [2], avoiding the more complex non-linear rigid movement space of $SE(2)$. This can be translated from simulator to the real robot arena due to the robots holonomic drive.

In the style of [6], each robot maintains a factor graph of its own poses, with a limit on the number of variable nodes that are held of n_{hist}. Every timestep $t_{node} = 0.5s$, a new pose variable node x^t_{pose} is created, with a measurement factor $f^t_{meas}(x^t_{pose}, x^{t-1}_{pose})$ connecting it to the previous pose node, the measurement being the simulated odometry of the robot. The initial pose variable in the factor graph is anchored with a weak unitary factor as a prior. When the limit on the history n_{hist} is reached, the oldest pose variable in the graph is removed, and a weak prior containing the previously inferred belief of the now oldest variable is added.

[1] https://drive.google.com/file/d/1EuA8PS1qpqK6LIfPwCNXtQ3hHNWPDvtN/view?usp=sharing

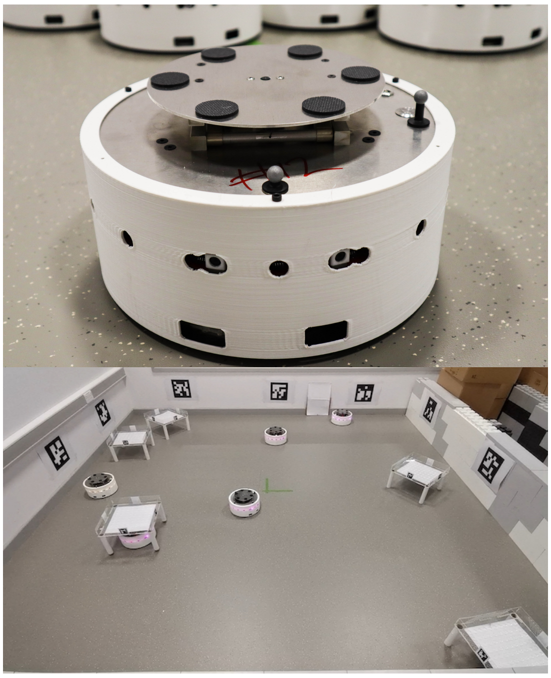

Fig. 1: Top: single DOTS robot, capable of carrying a 2kg load, moving at 2ms^{-}1, with a wide range of sensors and a long battery life. Bottom: Arena, showing multiple robots performing a carrier retrieval task.

Message passing on the local graph takes place once per simulation tick of 60Hz, according to various possible schedules.

Carriers C are regarded as landmarks, albeit ones that may move. If we assume the majority are stationary for most of the time and the local factor graphs have limited temporal window, we rely on the whole system having sufficient salience for navigation. Each robot maintains a set of variable nodes $x^i_{carrier}, i \in C$, one for each possible carrier. When a robot observes a carrier k, it adds a measurement factor $f^t_{meas}(x^t_{pose}, x^k_{carrier})$. As old pose variable nodes are removed, if there are no factors now connecting a carrier variable to

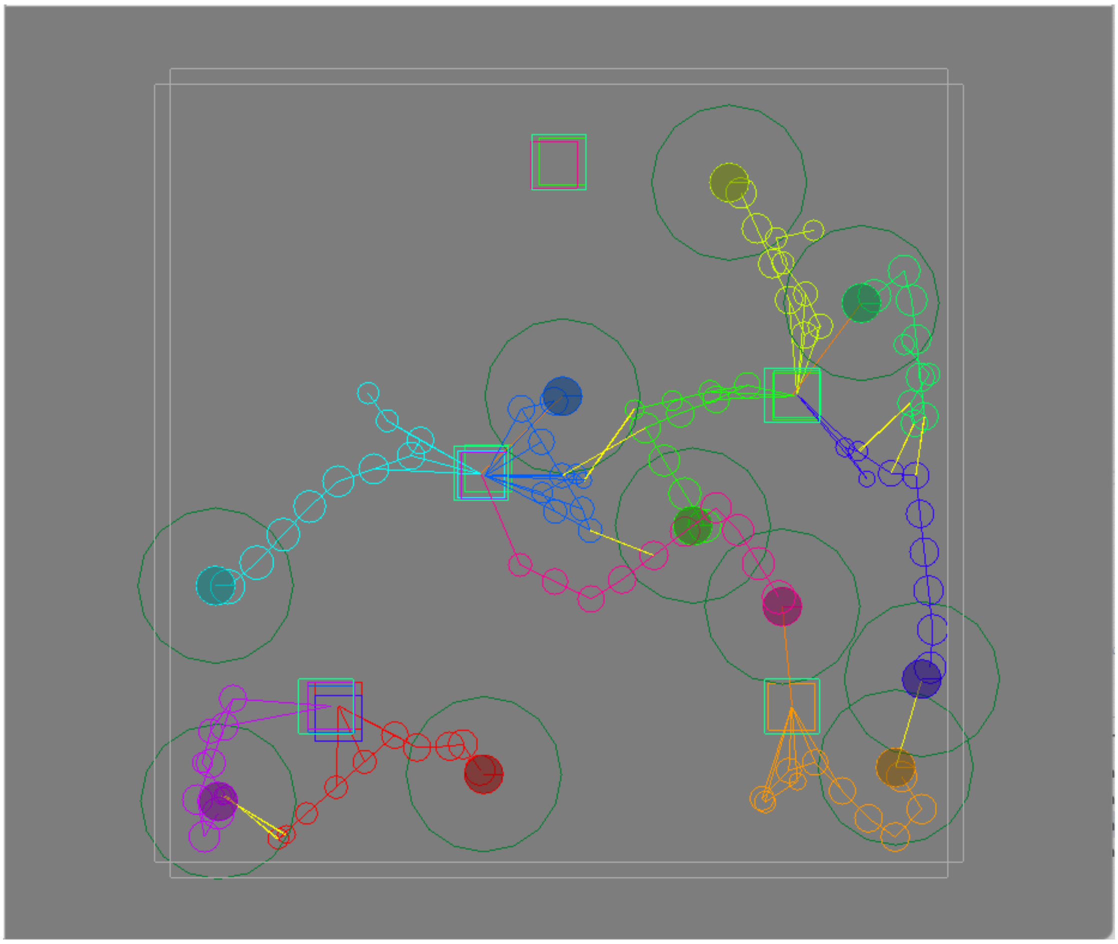

Fig. 2: Simulation of robots exploring an environment with a random walk, showing the robots as filled circles and the carriers as pale green squares within the environment. Also shown are the local factor graphs of each robot with the empty circles and lines representing variables and factors, with colours corresponding to the robot. Yellow lines denote factors joining the local graphs, and green circles the communication range of the robots. Within the carrier squares are overlayed the current estimates of each robot of the carrier position, already showing good correspondence after two minutes of simulated time.

a pose variable, a new factor is constructed by combining the factors:

$$f = f_{meas}(x_{pose}^{(t-n_{hist})}, x_{carrier}^{k}) \cdot f_{meas}(x_{pose}^{(t-n_{hist}+1)}, x_{pose}^{(t-n_{hist})}) \quad (1)$$

Robots may also make observations of other robots; this is the key to integrating the local factor graphs into a whole. Each time a robot i observes another j, it creates a measurement factor linking the two $f_{meas\_ext}^{t}(x_{ipose}^{t}, x_{jpose}^{t})$. This factor resides on the observing robot, but obviously cannot take part in the normal message passing of the GBP algorithm. To link robots together, we use the scheme outlined in [6]. Each robot maintains a page of data that is visible to any other robot that is within communication range. This contains three sets of data: the beliefs of the pose nodes $x^{t-n_{hist}+1}, .., x^{t}$, the outward messages of the external measurement factors f_{meas_ext} and the beliefs of the carrier state variable nodes $x^0_{carrier}, .., x^{C-1}_{carrier}$.

The standard messages passed described in [2] are as follows: all messages comprise an information vector and a precision matrix (η, Λ). For a variable node, all the incoming messages from factors *except* the destination factor are summed to create the outgoing message. For a factor node, all messages from connected variables are summed with the factor then the output variable is marginalised out to give the output message to that variable. This leads to the issue that a robot displaying just the beliefs of its variable nodes is not sufficient - the belief of a variable node is the sum of all incoming messages. We can, however, recover the information at the factor node side; $\mu^t_{x \to f} = bel(x) - \mu^{t-q}_{f \to x}$ by subtracting the last message sent from the factor. This allows the necessary one-to-many connection of variable nodes to factors. Note that this may not be from the last timestep but q timesteps ago, depending on when communication was last possible.

Each robot observing a carrier state described on another robot constructs or updates an implicit measurement factor connecting the two states with a zero distance. As robots observe each other and see the other's observed carrier states, these factors 'pull' the individual factor graphs into alignment, providing a global, carrier-based reference.

IV. Results and discussion

Figure 2 shows a simulation of ten robots exploring a 3.7m square arena with five carriers. Robots can communicate within a 0.5m radius. The factor graph history $n_{hist} = 10$, with new nodes being constructed every 0.5s, giving a time window of 5s. Robots perform a random walk within the arena. Each robot is a solid circle, and leaves a trail of unfilled circles in the same colour representing the pose nodes of the local graph connected by odometry meaurements. The square boxes representing the carriers have smaller coloured boxes within representing the estimates of each robot. Yellow lines show where an external factor connects pose variables in two local factor graphs. It can be seen from this early trial over approximately 2 minutes of simulated time that the swarm has achieved good estimates of the carrier positions.

We are still experimenting with message passing schedules. Our current approach chooses a single factor on each robot at random, once per simulation tick of 60Hz, so 30 times between new nodes being added to the graph. The factor sends a message to its neighbours, and they update their beliefs. We sometimes observe some instability in convergence, particularly when there are multiple connections between local graphs. To handle this, we are using the damping approach described in [7].

V. Conclusion

We are excited by the possibilities offered by combining swarm algorithms with GBP. As noted earlier, we wish to bring other items of state and knowledge within the overarching framework; it will be interesting to compare some more traditional swarm consensus algorithms with GBP, many best-of-n swarm algorithms rely on local message passing [11], perhaps suggesting deeper similarities. The current simulation system relies on graph construction happening in synchronisation at regular timesteps, but we don't see this as necessary and plan to remove this restriction, e.g. by estimating between states at time of observation. Finally, we are looking forward to implementing the algorithm on a real swarm as part of our logistics demonstration task.

References

[1] M. Schranz, M. Umlauft, M. Sende, and W. Elmenreich, "Swarm Robotic Behaviors and Current Applications," *Frontiers in Robotics and AI*, vol. 7, p. 36, 2020.

[2] A. J. Davison and J. Ortiz, "Futuremapping 2: Gaussian Belief Propagation for Spatial ai," *arXiv preprint arXiv:1910.14139*, 2019.

[3] J. Ortiz, T. Evans, and A. J. Davison, "A visual introduction to Gaussian Belief Propagation," *arXiv preprint arXiv:2107.02308*, 2021.

[4] 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.

[5] S. Jones, E. Milner, M. Sooriyabandara, and S. Hauert, "Distributed Situational Awareness in Robot Swarms," *Advanced Intelligent Systems*, vol. 2, no. 11, p. 2000110, 2020.

[6] R. Murai, J. Ortiz, S. Saeedi, P. H. Kelly, and A. J. Davison, "A Robot Web for Distributed Many-Device Localisation," *arXiv preprint arXiv:2202.03314*, 2022.

[7] J. Ortiz, M. Pupilli, S. Leutenegger, and A. J. Davison, "Bundle Adjustment on a Graph Processor," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2416–2425.

[8] A. Patwardhan, R. Murai, and A. J. Davison, "Distributing Collaborative Multi-Robot Planning with Gaussian Belief Propagation," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 552–559, 2022.

[9] E. Milner, M. Sooriyabandara, and S. Hauert, "Stochastic behaviours for retrieval of storage items using simulated robot swarms," *Artificial Life and Robotics*, vol. 27, no. 2, pp. 264–271, 2022.

[10] ——, "Swarm diffusion-taxis: Transport of spatial information for cooperative gradient-based navigation," in *Proc. Swarm*, 2022.

[11] G. Valentini, E. Ferrante, and M. Dorigo, "The Best-of-n Problem in Robot Swarms: Formalization, State of the Art, and Novel Perspectives," *Frontiers in Robotics and AI*, vol. 4, p. 9, 2017.