Self-Organised Multi-Agent Path Planning Based on STDMA

By

Runze Yuan

MSc Robotics Dissertation

Department of Engineering Mathematics
UNIVERSITY OF BRISTOL
&
Department of Engineering Design and Mathematics
UNIVERSITY OF THE WEST OF ENGLAND

A MSc dissertation submitted to the University of Bristol and the University of the West of England in accordance with the requirements of the degree of MASTER OF SCIENCE IN ROBOTICS in the Faculty of Engineering.

July 25, 2023

Declaration of own work

I declare that the work in this MSc dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

Name and Date

Acknowledgement

I would like to thank ...

Abstract

Abstract should give a short summary of the motivation, the approach and important insights and results.

Number of words in the dissertation: .... words.

Contents

List of Tables

1 Introduction

1.1 Aims

Based on the self-organised channel resource allocation principle of the STDMA (Self-organised Time Division Multiple Access) [1] communication protocol, develop a method for agents to achieve collision-free moving on a 2D plane, make it self-organised, decentralised and scalable.

1.2 Objectives

- Develop a multi-agent communication scenario with STDMA in ROS2. In this scenario, multiple identical agents will share the same channel and all agents have both receiving and transmitting capabilities. The goal is to enable agents to autonomously organise/join the communication process after start up.
- On the basis above, a grid world is implemented: a 2D map composed of grids, each grid represent a part of space. Agents could move from one grid to another in the map.
- ...

1.3 Motivation

The motivation of this project is to answer a question inspired by [2], which is:

What will happen if use STDMA for 2D resource sharing?

The detailed explanation of the motivation is as follows.

1.3.1 Why STDMA?

What is STDMA? The *Self-organised Time Divided Multi Access* (STDMA) is an channel access technique for communication. It is based on a set of policies dictating how agents ought to apply for slots in repeating frames. In short, STDMA is a protocol that allows agents to autonomously share 1D resources (time) among themselves.

The reason for using STDMA is its **characteristics**:

1. **Deterministic**: Agents arrange their data transmission based on a determined timetable.

2. **Decentralised**: Agents listen to the channel first, then find free slots in the timetable for themselves to use.

These characteristics could be useful for multiple agents to achieve collision-free (use free slots only), self-organised (find slots on its own) resource sharing.

There are also two **reasons that making this challenging**:

1. **Purpose difference**: In the original STDMA, an agent only needs to find one unused slot in the repeating frame (which is generally of a fixed length chosen manually) and maintain ownership of that slot, which means agents don't need to move from one slot to another slot, i.e., don't have destination in the channel allocation timetable. But that's different for 2D space moving, where agents have their destinations and need to move grid by grid to reach their goals.

2. **Complexity difference**: Channel sharing only requires randomly selecting one slot from the known free slots, while space sharing in a 2D plane requires the agent to find a path that doesn't collide with other agents, and the possibility space for the path could be enormous (map size $\times$ predicting horizon), but the processing time available to the agent is limited (must broadcast its presence at regular intervals so that other agents can be aware of its plans and thus avoid collisions).

These characteristics and challenges make this project interesting and worthy of research.

Figure 1.1: Kiva system operating in Amazon warehouse[2].

1.3.2 Why 2D plane path planning?

2D plane path planning is a name for better understanding and is not the accurate name for this problem. This problem is acturally a ***Multi-Agent Path Finding*** **(MAPF)** problem [3], [4].

Here, the term "finding" could be somewhat confusing. In fact, he MAPF problem combines both **collision-free movement** of multiple agents and **path efficiency optimising**.

Overall, the problem of *Multi-Agent Path Finding* (MAPF) concerns **the movement of multiple agents in a grid world**.

To easily get a better understanding of this problem, please refer to this website[1]. The website presents the MAPF problems and their solutions through a simple and engaging animation (estimated time required: 1∼2 min).

MAPF algorithms have broad prospects for use.

- **Warehouse Automation**: Pick-pack-and-ship system in warehouse (like Fig 1.1). Delivering items in sorting station [5], [6].
- **Intersection Management**: Coordinate autonomous vehicle movement through intersections [7].
- **Robot Fleet**: Automating fleets of autonomous robots like forklift fleets [8], [9].

[1] https://primalgrid.netlify.app/primal

[2] https://spectrum.ieee.org/interview-brad-porter-vp-of-robotics-at-amazon

- **Agents in video games and CGIs**: Flock simulating and animating[10], [11].
- **Swarm Robots**: Controlling self-organised robot swarms[12].

In general, the application of this algorithm is very broad. Furthermore, in the context of Industry 4.0 and flexible manufacturing, it has even better prospects for application[13]–[15], as centralized control is gradually becoming inadequate to meet new production needs.

2 Literature Review

2.1 STDMA Explained

As previously mentioned in Section 1.3.1, STDMA [1] allows multiple agents to share the same channel for communication without centralised control. The main assumption of this protocol is that all devices have synchronised clocks. In practice, this is achieved through GPS [16].

The core idea of STDMA can be summarized as follows: Represent continuous time with repeating frames, each consisting of an euqal number of discrete slots. Agents constantly monitor channel information, attempting to occupy a free slot in the frame for themselves. Once an agent secures a slot, it can broadcast its information in that slot of every frame.

Agents using STDMA have **four phases**, which are arranged in chronological order as follows:

1. **Initialisation**: Agents in this phase have not yet joined the network. The agent listens to an entire frame and determines the current slot allocation.

2. **Network Entry**: Randomly choose an unallocated slot to broadcast their existence and reserve one slot for the next phase.

3. **First Frame**: Use the slot reserved in the previous phase to reserve more slots for themselves. The number of reserved slots depends on the size of the data packet that the device needs to send.

4. **Continuous Operation**: Use the previously reserved slot to work normally. If some slots are released or more slots are needed, reapply for slots.

Although the above description omits some details (such as slot choosing methods, calculation of required number of slots, etc.), it is clear that **the core of STDMA is the strategy of finding and reserving unallocated slots**.

This protocol also has some limitations, such as: **(1)** Collision: In Network Entry phase, multiple devices may accidentally choose the same unallocated slot to broadcast their existence. **(2)** Capacity: When the slots are not enough, conflicts will inevitably occur. There are some studies [17], [18] that proposed improvements to the above limitations, but this is not the focus of this project.

2.2 MAPF Algorithm Review

2.3 MPC Review(?)

3 Research Methodology

This section provides detailed information on algorithm, its implementation, and experiment design.

3.1 Algorithm and Implmentation

Environment

- **Hardware:** ROG Zephyrus M16 Laptop
  - CPU: 11th Gen Intel(R) Core(TM) I7-11800H @ 2.30GHz
  - GPU: NVIDIA GeForce RTX 3060 Laptop GPU (unrelated to the experiment, information provided just for content completeness)
- **Software:**
  - OS: WSL2 (Ubuntu 22.04 LTS) in Windows 11 23H2
  - Implementation Platform: ROS2 Humble, all codes written in python

The algorithm's idea could be summarised in the following two parts:

1. **Communicate with STDMA**: Agents establish communication with STDMA protocol.
2. **Plan**: Agents plan their move based on the information received.

And they are explained as below.

3.1.1 Communicate with STDMA

As previously mentioned in Section 2.1, the STDMA protocol mainly consists of the following three parts:

1. Synchronised Clock: Agents have synchronised clock.
2. State Machine: Four states for an agent to control their behaviour and describe network usage.
3. Message Publishing and Sending: Agents could monitor the channel and send information through it.

And these features are implemented within the ROS2 structure.

Synchronised Clock

In this paper, shared clock pulses are used to achieve the purpose of synchronizing clocks between agents. A ROS2 publisher sends a square wave with a fixed period and a duty cycle of 50% to the designated topic, serving as the shared clock signal. Agents subscribe to this clock signal topic to achieve clock synchronisation.

3.2 Experiment Design

4 Results

The Results section should provide

- An overview of all obtained results
- An in detail discussion/explanation of all results
- A scientific interpretation of the results

4.1 Common attributes to pay attention to are:

- When comparing plots, keep the scale of axes consistent. To do otherwise is misleading for the reader.
- If you are going to compare separate plots, consider if they can be better evaluated when combined into a single plot.
- When plotting data, particularly the *mean*, ensure that you also plot error bars (or other method) of indicating the distribution.
- If a figure or plot is included, ensure it is referenced explicitly in the body text discussion.
- When a large table of data is included, consider whether it would be better communicated as a box-plot or something similar.
- All axes should be labelled and include units of measurement where applicable.
- All captions and figures should have captions with enough information to be understood at a glance. Do not use captions to provide information that is better placed in the body text.
- Remember to identify result outliers and anomalous data and to attempt an explanation or justification.

5 Discussion and Conclusion

The conclusion needs to provide

- A short summary (What has been done and what are the main results)
- Limitations of your work, where applicable.
- Discussion of your work in the bigger picture (How does this contribute to the research field?)
- Future work (What could be next steps in this work?). Remember to keep future work realistic. A good approach is to discuss what the next progression of this project would be, and to justify why this would be interesting.

You will find it easier to write your conclusion if you copy-and-paste your *Aims, Objectives*, and any research questions or hypotheses you stated. You can then discuss each of these explicitly in turn, and how you were able to answer them or complete them successfully. When things have not gone as well as you would have hoped, demonstrate your critical thinking and reasoning to analyse the short-comings of your project - to demonstrate that you understand the underlying causes and that you could conduct good futurework from this learning experience.

A Appendix

This is optional. Not every report needs an appendix If you have additional information like code pieces, long tables, etc. that would break the flow of the text in the report, you can put it here.

References

[1] T. Gaugel, J. Mittag, H. Hartenstein, S. Papanastasiou, and E. G. Ström, *In-depth analysis and evaluation of self-organizing tdma*, 2013. DOI: 10.1109/VNC.2013.6737593.

[2] E. F. Asadi and A. Richards, Scalable distributed model predictive control for constrained systems, *Automatica*, vol. 93 2018, pp. 407–414, 2018.

[3] R. Stern, Multi-agent path finding–an overview, *Artificial Intelligence: 5th RAAI Summer School, Dolgoprudny, Russia, July 4–7, 2019, Tutorial Lectures* 2019, pp. 96–115, 2019.

[4] R. Stern, N. Sturtevant, A. Felner, *et al.*, "Multi-agent pathfinding: Definitions, variants, and benchmarks," in *Proceedings of the International Symposium on Combinatorial Search*, vol. 10, 2019, pp. 151–158.

[5] N. M. Kou, C. Peng, H. Ma, T. S. Kumar, and S. Koenig, "Idle time optimization for target assignment and path finding in sortation centers," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 9925–9932.

[6] W. Hönig, S. Kiesel, A. Tinka, J. W. Durham, and N. Ayanian, Persistent and robust execution of mapf schedules in warehouses, *IEEE Robotics and Automation Letters*, vol. 4, no. 2 2019, pp. 1125–1131, 2019.

[7] K. Dresner and P. Stone, A multiagent approach to autonomous intersection management, *Journal of artificial intelligence research*, vol. 31 2008, pp. 591–656, 2008.

[8] F. Pecora, H. Andreasson, M. Mansouri, and V. Petkov, "A loosely-coupled approach for multi-robot coordination, motion planning and control," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 28, 2018, pp. 485–493.

[9] J. Salvado, R. Krug, M. Mansouri, and F. Pecora, "Motion planning and goal assignment for robot fleets using trajectory optimization," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7939–7946. DOI: 10.1109/IROS.2018.8594118.

[10] R. Olfati-Saber, Flocking for multi-agent dynamic systems: Algorithms and theory, *IEEE Transactions on automatic control*, vol. 51, no. 3 2006, pp. 401–420, 2006.

[11] H. Ma, J. Yang, L. Cohen, T. Kumar, and S. Koenig, "Feasibility study: Moving non-homogeneous teams in congested video game environments," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 13, 2017, pp. 270–272.

[12] W. Hönig, J. A. Preiss, T. S. Kumar, G. S. Sukhatme, and N. Ayanian, Trajectory planning for quadrotor swarms, *IEEE Transactions on Robotics*, vol. 34, no. 4 2018, pp. 856–869, 2018.

[13] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, Industry 4.0, *Business & information systems engineering*, vol. 6 2014, pp. 239–242, 2014.

[14] D. A. Rossit, F. Tohmé, and M. Frutos, Industry 4.0: Smart scheduling, *International Journal of Production Research*, vol. 57, no. 12 2019, pp. 3802–3813, 2019.

[15] M. De Ryck, M. Versteyhe, and F. Debrouwere, Automated guided vehicle systems, state-of-the-art control algorithms and techniques, *Journal of Manufacturing Systems*, vol. 54 2020, pp. 152–173, 2020.

[16] Z. Fernández, I. Val, M. Mendicute, and E. Uhlemann, "Analysis and evaluation of self-organizing tdma for industrial applications," in *2019 15th IEEE International Workshop on Factory Communication Systems (WFCS)*, 2019, pp. 1–8. DOI: 10.1109/WFCS.2019.8758039.

[17] H. Mosavat-Jahromi, Y. Li, Y. Ni, and L. Cai, Distributed and adaptive reservation mac protocol for beaconing in vehicular networks, *IEEE Transactions on Mobile Computing*, vol. 20, no. 10 2020, pp. 2936–2948, 2020.

[18] T. Terauchi, K. Suto, and M. Wakaiki, "Harvest-then-transmit-based tdma protocol with statistical channel state information for wireless powered sensor networks," in *2021 IEEE 93rd vehicular technology conference (VTC2021-Spring)*, IEEE, 2021, pp. 1–5.