Thèse n° 10 043

# Analytical and Machine Learning Methods for the Complete Safe Coordination of Astrobot Swarms

Présentée le 26 février 2021

Faculté des sciences et techniques de l'ingénieur
Groupe SCI STI DG
Programme doctoral en robotique, contrôle et systèmes intelligents

pour l'obtention du grade de Docteur ès Sciences

par

**Matin MACKTOOBIAN**

Acceptée sur proposition du jury

Prof. F. Mondada, président du jury
Dr D. Gillet, Prof. J.-P. R. Kneib, directeurs de thèse
Dr E. Tempel, rapporteur
Dr L. Makarem, rapporteuse
Prof. A. Martinoli, rapporteur

École
polytechnique
fédérale
de Lausanne

2021

# Thesis Committee

President:

**Prof. Francesco Mondada**, EPFL, Switzerland

Advisor:

**Dr. Denis Gillet**, EPFL, Switzerland

Co-advisor:

**Prof. Jean-Paul Kneib**, EPFL, Switzerland

Internal examiner:

**Prof. Alcherio Martinoli**, EPFL, Switzerland

External examiners:

**Dr. Elmo Tempel**, Tartu Observatory, Estonia
**Dr. Laleh Makarem**, Logitech Europe S.A., Switzerland

# Abstract

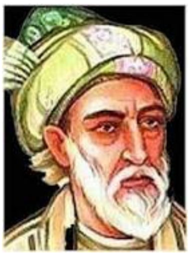
The recent generations of massive spectroscopic surveys aim at the ray collection from a multitude of cosmological targets in the course of observations. For this purpose, astrobots are used to change the configuration of optical fibers from one observation to another in relatively short periods of time instead of tedious manual replacements. The dense formations of astrobots on focal planes enhance the number of the potential targets to be observed. However, the safe coordination of astrobot swarms is challenging. The more astrobots are coordinated, the more data are sent to a spectrograph, thereby the higher the resolution of a resulted survey will be. However, traditional collision-avoidance coordination strategies often give rise to the partial convergence of astrobot swarms. Thus, this thesis focuses on the solutions to the complete safe coordination of astrobots, particularly in the case of Sloan Digital Sky Survey V. We increase coordination convergence rates not only by directly improving the state-of-the-art coordination solution but also by optimizing target-to-astrobot assignments. Namely, we propose an optimal assignment scheme which minimizes both the likelihood of collisions between astrobots and the effort demanded to preform coordination in terms of the required time to perform coordination. We also propose a cooperative coordination method in which, given particular settings of astrobots and/or targets, each astrobot stops at its goal point when its other neighboring peers have already reached theirs, as well. So, we derive a localized completeness condition that, if sufficed, generates the trajectories which completely coordinate an astrobot swarm in a guaranteed manner. We also propose a logic-based formally-verifiable supervisory coordination technique whose behavior is always safe and complete without any need to simulation-based validations. Finally, we employ machine learning tools to train models to predict the feasibility of complete coordination only according to initial and final configurations of astrobots and their targets' projected locations on their focal plane. These models contribute to the identification of those target-astrobot pairings which do not fulfill the completeness condition. So, one may re-plan such ill-posed assignments before getting involved in potentially-pointless completeness checking simulations. Our simulated and experimental tests manifest the efficiency of the developed methods quoted above.

*Keywords* — astrobots, coordination, safety, collision avoidance, dark energy, massive spectroscopic surveys, cosmology, machine learning, convergence prediction, artificial potential fields

# Résumé

Les récentes générations de relevés spectroscopiques massifs visent à la collecte de rayons à partir d'une multitude de cibles cosmologiques au cours des observations. À cette fin, les astrobots sont utilisés pour changer la configuration des fibres optiques d'une observation à une autre dans des périodes de temps relativement courtes au lieu de remplacements manuels fastidieux. Les formations denses d'astrobots sur les plans focaux augmentent le nombre de cibles potentielles à observer. Cependant, la coordination sûre des essaims d'astrobots est difficile. Plus les astrobots sont coordonnés, plus le nombre de données envoyées à un spectrographe est élevé, plus la résolution de l'enquête résultante sera élevée. Cependant, les stratégies traditionnelles de coordination d'évitement des collisions donnent souvent lieu à la convergence partielle des essaims d'astrobots. Ainsi, cette thèse se concentre sur les solutions à la coordination sécurisée complète des astrobots, en particulier dans le cas de Sloan Digital Sky Survey V. Nous augmentons les taux de convergence de coordination non seulement en améliorant directement la solution de coordination de pointe, mais aussi en optimisation des affectations cible-astrobot. À savoir, nous proposons un schéma d'assignation optimal qui minimise à la fois la probabilité de collisions entre astrobots et l'effort requis pour préformer la coordination en termes de temps requis pour effectuer la coordination. Nous proposons également une méthode de coordination coopérative dans laquelle, étant donné les paramètres particuliers des astrobots et / ou des cibles, chaque astrobot s'arrête à son point de destination lorsque ses autres pairs voisins ont déjà atteint le leur. Ainsi, nous dérivons une condition d'exhaustivité localisée qui, si elle est suffisante, génère les trajectoires qui coordonnent complètement un essaim d'astrobots de manière garantie. Nous proposons également une technique de coordination de supervision formellement vérifiable basée sur la logique, dont le comportement est toujours sûr et complet sans avoir besoin de validations basées sur la simulation. Enfin, nous utilisons des outils d'apprentissage automatique pour former des modèles afin de prédire la faisabilité d'une coordination complète uniquement en fonction des configurations initiales et finales des astrobots et des emplacements projetés de leurs cibles sur leur plan focal. Ces modèles contribuent à l'identification des couples cible-astrobot qui ne remplissent pas la condition d'exhaustivité. Ainsi, on peut planifier de telles affectations mal posées avant de s'impliquer dans des simulations de vérification d'exhaustivité potentiellement inutiles. Nos tests simulés et expérimentaux démontrent l'efficacité des méthodes développées citées ci-dessus.

*Mots clés* — astrobots, coordination, sécurité, évitement des collisions, énergie sombre, levés spectroscopiques massifs, cosmologie, apprentissage automatique, prédiction de convergence, champs de potentiel artificiel



خمش ز دعوی دانش، که جهل را صائب
هزار حجت ناطق چو خودستایی نیست

ـ صائب (۱۰۵۵-۹۷۱ خورشیدی)

“Be silent, Saeb, not to claim owning wisdom,
as ignorance, over thousands of proofs, is the reminiscent of swagger”

– Saeb (1592-1676)

To my Hadis,

and the rest of the JooJoo-ShooShoo gang:

Doolati, Pashmak, Sahra, the wheatland, and the diary,

who witnessed our “vow at the dawn”

# Acknowledgements

To me, the research whose essence are rooted in the intersection of engineering and basic sciences have always been the most fascinating. In this regard, I am grateful to my advisor *Dr. Denis Gillet* and my co-advisor *Prof. Jean-Paul Kneib* for providing me the opportunity to contributing to the complete coordination control of astrobot swarms in Sloan Digital Sky Survey V (SDSS-V). I indeed enjoyed doing cutting-edge research in the project which contributes to the eventual discovery of the nature of dark energy.

During this journey, my collaborations with the SDSS-V consortium and the astrobots team of EPFL impeccably culminated the depth of my engagement in the control strategy development for the complete safe coordination of astrobot swarms. In particular, I appreciate the following colleagues of the astrobots team because of their influential impacts on my smooth yet rigorous involvement in the project: *Ricardo Araújo* and *Dr. Luzius Kronig*. I also took delight in collaborations with *Francesco Basciani*, *Juan-Carlos Farah* and *Dr. Ezequiel González Debada*, in the course of my association with the robotics, control, and intelligent systems (EDRS) Ph.D. program.

The valuable comments and the critical perspective of my thesis committee improved the coherency of this thesis because of which I thank the following individuals: *Prof. Francesco Mondada*, *Prof. Alcherio Martinoli*, *Dr. Elmo Tempel*, and *Dr. Laleh Makarem*.

My accomplishments during my Ph.D. years might not be yielded if I have not been blessed by the unconditional love, the never-ending energy, and the encouragements of my family: *Marzieh* and *Moeen*. One cannot imagine a better mother-brother pair whose energy make any dream into a goal for me.

*Dr. Ali Paknahad*, to me, is the symbol of a friend who is such impeccably meshed with me that his friendship may be deservedly called a brotherhood. His company, shining by the warmth of the lady *Madi*'s love, during our long ride, has been extremely appreciated. Special thanks also *Farhad Sadeghlo* and *Ahmad Khataminejad Tehrani* for their friendship even from thousands of kilometers away. The companionship of *Kevin Copes* during my time in Switzerland may also not be forgotten.

Reaching the apex of any success may be way more fun, enjoyable, and robust when you have a company who is as passionate and motivated as yourself to achieve your goals. So, I am at a loss for words to express my gratitude to my lovely *Hadis* because of her inspiring ever-lasting presence by my side in all ups and downs of these years. This milestone belongs to both of us, as I hope it will pave our way for facing the future adventures we will together go through to realize our common goals by a mixture of love and wisdom.

The financial supports of Swiss National Science Foundation and Sloan Foundation have also been appreciated throughout my Ph.D. studies.

*Lausanne, Switzerland* *Matin Macktoobian*
*February 2021*

# Contents

**Abstract (English/français)** iii

**Acknowledgments** viii

**Contents** x

**List of Figures** xiii

**List of Tables** xvi

**List of Symbols** xvi

**1 Preface** 1
- 1.1 An Introduction to Astrobotics . . . 1
  - 1.1.1 Massive Spectroscopic Surveys and SDSS-V . . . 1
  - 1.1.2 Astrobots Characterization . . . 4
    - 1.1.2.1 Mechanical Specification . . . 4
    - 1.1.2.2 Communications and System Interface . . . 6
- 1.2 Motivation . . . 6
- 1.3 Research Goals and Contributions . . . 8
  - 1.3.1 Optimal Target-to-Astrobot Assignment . . . 8
  - 1.3.2 Complete Coordination . . . 8
  - 1.3.3 Completeness Prediction . . . 9
  - 1.3.4 List of Publications . . . 10
- 1.4 Organization . . . 11

**Part A Target-to-Astrobot Assignment** 13

**2 Optimal target-to-astrobot assignment** 15
- 2.1 Introduction . . . 15
- 2.2 Algorithm . . . 19
- 2.3 Simulations . . . 22
- 2.4 Conclusion . . . 24

**3 Heterogeneous Target-to-Astrobot Assignment** 27
- 3.1 Introduction . . . 27
- 3.2 Algorithm . . . 28
  - 3.2.1 Bundle Formation . . . 30
  - 3.2.2 Algorithm . . . 31
- 3.3 Simulations . . . 33
- 3.4 Conclusion . . . 34

Part B Analytical Completeness Seeking 35

4 Supervisory Complete Coordination 37
4.1 Introduction . . . . . 37
4.2 Background . . . . . 39
4.3 Completeness-Seeking Supervision . . . . . 40
4.4 Example . . . . . 45
4.5 Conclusion . . . . . 46

5 Cooperative Complete Coordination 47
5.1 Introduction . . . . . 47
5.2 From Local to Global Completeness . . . . . 49
5.3 Cooperative Artificial Potential Fields . . . . . 51
5.3.1 Motivation . . . . . 52
5.3.2 Formulation . . . . . 52
5.4 Completeness Analysis . . . . . 54
5.4.1 Completeness Condition . . . . . 54
5.4.2 Completeness Seeking by Parameter Modification . . . . . 58
5.5 Discussion . . . . . 59
5.6 Results . . . . . 60
5.6.1 Simulations . . . . . 60
5.6.2 Experimental Validation . . . . . 61
5.6.2.1 Setup . . . . . 61
5.6.2.2 Parameters Impacts on Complete Coordination . . . . . 64
5.6.2.3 Convergence Time . . . . . 66
5.6.2.4 Target Distribution Influence on Completeness . . . . . 67
5.7 Conclusion . . . . . 69

Part C Completeness Prediction 71

6 Constrained-Parity Convergence Prediction 73
6.1 Introduction . . . . . 73
6.2 Problem Statement . . . . . 76
6.3 Convergence Prediction Strategy . . . . . 76
6.3.1 Imbalanced Data Compensation . . . . . 78
6.3.2 Prediction Probability Computation . . . . . 79
6.3.3 Prediction Probability Localization . . . . . 80
6.4 Simulations and Results . . . . . 81
6.4.1 Performance Measures . . . . . 81
6.4.2 A swarm including 116 astrobots . . . . . 83
6.4.3 A swarm including 487 astrobots . . . . . 85
6.5 Conclusion . . . . . 87

7 Variable-Parity Convergence Prediction 91
7.1 Introduction . . . . . 91
7.2 Constrained Convergence Prediction . . . . . 92
7.2.1 Data Definition . . . . . 92
7.2.2 Feature Scaling . . . . . 93
7.2.3 Predictor Synthesis . . . . . 93
7.2.4 Validation . . . . . 94
7.3 Generalized Convergence Prediction . . . . . 95

7.3.1 Data Definition Generalization . . . 95
7.3.2 Parity Normalization . . . 95
7.4 Results . . . 98
7.4.1 Constrained Scenario . . . 98
7.4.2 Generalized Scenario . . . 99
7.5 Conclusion . . . 102

**Final Words** **102**

**8 Conclusions** **105**
8.1 Highlights . . . 105
8.2 Future Directions . . . 106
8.2.1 Distributed Target-to-Astrobot Assignment . . . 106
8.2.2 Finite State Machine Reduction . . . 107
8.2.3 Optimal Coordination . . . 108
8.2.4 Completeness Prediction . . . 108

**9 Appendix** **111**

**References** **116**

**Curriculum Vitae** **124**

# List of Figures

1.1 A typical SDSS-V astrobot and overall focal plane . . . . . . . . . . . . . . . . . . . . . . . . 2

1.2 The multi-fiber spectroscopy operation process (partially reprinted from [Kronig 2020] with permission) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.3 The overlapping areas in a typical neighborhood . . . . . . . . . . . . . . . . . . . . . . . . 5

1.4 The domains of the safe complete coordination problem . . . . . . . . . . . . . . . . . . . . . . 7

1.5 The chapter dependencies in the thesis . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

2.1 Optimization requirements evaluations in a typical assignment scenario . . . . . . . . . . . . 19

2.2 The convergence rate comparison between optimal and target-based assignments during hybrid coordinations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

2.3 The deadlock occurrences using optimal assignment during hybrid coordination . . . . . . . 24

2.4 The decrement of coordination supervisors' state space size during supervisory coordination . 24

3.1 A typical bundle whose corresponding surface covers a specific area around a particular dynamic object . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

3.2 The assignment results after the first iteration of the algorithm's loop . . . . . . . . . . . . . . 33

3.3 The assignment results after the second iteration of the algorithm's loop . . . . . . . . . . . . 34

3.4 The final results of the assignment process . . . . . . . . . . . . . . . . . . . . . . . . . . . 34

4.1 The discrete-event model of a typical astrobot $\pi$ with $n^{\pi} = 3$ . . . . . . . . . . . . . . . . . . 40

4.2 The initial and the target points corresponding to a typical astrobot $\pi$ with $n^{\pi} = 3$ . . . . . . 42

4.3 The specification corresponding to the Figure 4.2 . . . . . . . . . . . . . . . . . . . . . . . . 43

4.4 Completeness seeking via supervisory coordination . . . . . . . . . . . . . . . . . . . . . . . 43

4.5 A coordination problem . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

4.6 The generator of $\mathbf{S}^{\pi_1}$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

4.7 The generator of $\mathbf{S}^{\pi_2}$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

4.8 The generator of $\mathbf{S}^{\pi_3}$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

5.1 The competitive control architecture of the priority-based coordination . . . . . . . . . . . . 52

5.2 The cooperative control architecture of the complete coordination . . . . . . . . . . . . . . . 53

5.3 The arrangement of astrobots in a typical neighborhood subject to the maximum contention . 55

5.4 The parameter modification process . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59

5.5 Visual illustrations of the convergence rate and the convergence time corresponding to test batch 1 . . . 61

5.6 Visual illustrations of the convergence rate and the convergence time corresponding to test batch 2 . . . 62

5.7 The 19-astrobot setup of the applied experimental tests . . . 62

5.8 The setup of the hardware interconnections in our experimental tests . . . 63

5.9 The neighborhood types in the experimental focal plane . . . 64

5.10 The number of the required modification rounds to reach completeness in 1000 experimental tests . . . 65

5.11 The number of the required modification steps to reach completeness in 1000 experimental tests 67

5.12 Convergence times of 1000 tests on the experimental setup with respect to both coordination strategies . . . 68

5.13 Convergence times of 1000 tests on the experimental setup with respect to both distribution selections . . . 69

5.14 Completeness seeking using CAPF strategy. (a) Control architecture. (b-e) A typical coordination process started from a random initial configuration to a synchronized one. . . . 70

6.1 The schematic of the convergence prediction problem . . . 77

6.2 The folded formation of astrobots representing their initial configuration . . . 77

6.3 A typical probability localization scenario . . . 79

6.4 The flow of the convergence prediction algorithm . . . 81

6.5 The layout of the information represented by a typical confusion matrix . . . 82

6.6 The confusion matrices of the 116-astrobot swarm . . . 83

6.7 The neighborhood analyses of the 116-astrobot swarm corresponding to two neighborhood selections . . . 84

6.8 The evolution of the confusion matrices of the 116-astrobot swarm given different corrector factors . . . 85

6.9 Accuracy measures for the 116-astrobot swarm . . . 86

6.10 The ROC curve associated with the 116-astrobot swarm . . . 87

6.11 Two confusion matrices and a neighborhood analysis corresponding to the 487-astrobot swarm 88

6.12 Accuracy measures for the 487-astrobot swarm . . . 88

6.13 Further accuracy results associated with both swarms . . . 89

7.1 Prediction accuracy per neighborhood type in the constrained scenario . . . 96

7.2 Parity normalization impact on the magnitudes of data features . . . 96

7.3 TNR, TPR, and balanced accuracy dynamics with respect to average kernel size in the constrained scenario . . . 97

7.4 TNR, TPR, and balanced accuracy with respect to average class weight in the constrained scenario . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 97

7.5 Precision, recall and F1 dynamics with respect to average class weight in the constrained case . 99

7.6 Comparative ROC curves corresponding to the constrained scenario . . . . . . . . . . . . . . 99

7.7 Prediction accuracy per neighborhood type in the generalized scenario . . . . . . . . . . . . 100

7.8 TNR, TPR, and balanced accuracy dynamics with respect to average kernel size in the generalized scenario . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 101

7.9 TNR, TPR, and balanced accuracy with respect to average class weight in the generalized scenario 101

7.10 Precision, recall and F1 dynamics with respect to average class weight in the generalized scenario 102

7.11 Overall comparative ROC curves of both the constrained and the generalized scenarios . . . . 102

9.1 The software architecture of the coordinator . . . . . . . . . . . . . . . . . . . . . . . . . 112

9.2 A typical YAML file in which velocities are in radian per second and time stamp is in second. 114

9.3 The trajectory evolution of the sample YAML file . . . . . . . . . . . . . . . . . . . . . . . . 115

9.4 The ferrule motion of the sample trajectory (Here motion starts from the fully-folded formation at (22,0), then it gradually converges to its target in the left-hand-side of the plot.) . . . . 115

# List of Tables

1.1 The plan of the past, current, and future spectroscopic surveys . . . . . . . . . . . . . . . . . . 4

2.1 The parameters applied to the simulations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

4.1 Event specifications of the example astrobots . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

5.1 The convergence rate and the convergence time corresponding to test batch 1 . . . . . . . . . 60

5.2 The convergence rate and the convergence time corresponding to test batch 2 . . . . . . . . . 61

5.3 Parameters of astrobots in the performed tests . . . . . . . . . . . . . . . . . . . . . . . . . . . 65

5.4 The number of the required modification rounds to reach completeness in 1000 simulated tests 66

6.1 The best prediction results corresponding to the 116- and 487-astrobot swarms . . . . . . . . 86

7.1 The best prediction results corresponding to the constrained case . . . . . . . . . . . . . . . . . 100

7.2 The best prediction results corresponding to the generalized case . . . . . . . . . . . . . . . . . 101

# List of Symbols

| | | |
|---|---|---|
| $z$ | Redshift value | 1 |
| $\boldsymbol{q}_b^i$ | Base coordinate vector of astrobot $i$ | 4 |
| $\boldsymbol{l}$ | Length vector of an astrobot | 4 |
| $l_1$ | Length of an astrobot's first arm | 4 |
| $l_2$ | Length of an astrobot's second arm | 4 |
| $\boldsymbol{q}^i$ | Ferrule coordinate vector of astrobot $i$ | 4 |
| $\theta^i$ | First arm's angular devition of astrobot $i$ | 4 |
| $\phi^i$ | Second arm's angular devition of astrobot $i$ | 4 |
| $\pi$ | Astrobot | 20 |
| $t$ | Target | 20 |
| $\mathcal{R}(\pi, t)$ | Reachability relation given astrobot $\pi$ and target $t$ | 20 |
| $\mathcal{P}$ | Astrobots set | 20 |
| $\mathfrak{R}_t^{\mathcal{P}}$ | Reachable subset of astrobot set $\mathcal{P}$ with respect to target $t$ | 20 |
| $\mathcal{I}$ | Index set | 20 |
| $\boldsymbol{\mathcal{Q}}$ | Coordinate set corresponding to an astrobots set | 20 |
| $\overline{\boldsymbol{\mathcal{Q}}}$ | Average coordinate of an astrobots set | 20 |
| $\boldsymbol{q}_t$ | Projected location of target $t$ on focal plane | 20 |
| $\mathcal{J}$ | Index set | 20 |
| $\mathcal{T}$ | Targets set | 21 |
| $t_S$ | Static target | 28 |
| $\mathcal{T}_S$ | Static target set | 28 |
| $\mathcal{T}_S^H$ | High-priority static target set | 28 |
| $\mathcal{T}_S^L$ | Low-priority static target set | 28 |
| $t_D$ | Dynamic target | 29 |
| $\mathcal{T}_D$ | Dynamic target set | 29 |
| $\mathcal{B}$ | Bundle | 29 |
| $\mathcal{A}_{\mathcal{B}}$ | Base area of bundle $\mathcal{B}$ | 30 |
| $d$ | Distance of bundle to a particular dynamic target | 30 |

| | | |
|---|---|---|
| $\mathcal{A}$ | Field-of-view surface area | 30 |
| $\mathcal{F}(\cdot,\cdot)$ | Bundle geometry function | 30 |
| $\mathcal{C}(\mathcal{P},\mathcal{T})$ | Complete recheability relation with respect to astrobots set $\mathcal{P}$ and targets set $\mathcal{T}$ | 31 |
| $\mathcal{K}(\pi,t',t'')$ | Critical recheability relation with respect to astrobot $\pi$, dynamic target $t'$, and high-priority static target $t''$ | 31 |
| $\mathcal{P}_{\mathcal{K}}$ | Critical positioner set | 31 |
| $\mathbf{G}$ | Discrete-event system | 39 |
| $Q$ | Finite state set | 39 |
| $\Sigma$ | Finite alphabet set | 39 |
| $\mathcal{D}$ | Extended Partial transition function | 39 |
| $q_0$ | Initial state | 39 |
| $Q_m$ | Marked state set | 39 |
| $\Sigma_c$ | Controllable events | 39 |
| $\Sigma_u$ | Uncontrollable events | 39 |
| $L(\mathbf{G})$ | Closed behavior of language **G** | 39 |
| $L_m(\mathbf{G})$ | Marked behavior of language **G** | 39 |
| $\mathcal{G}$ | Control pattern | 39 |
| $\mathrm{Pwr}(\Sigma)$ | Power set of set $\Sigma$ | 39 |
| $\mathbf{V}/\mathbf{G}$ | **G** under supervision of **V** | 39 |
| $\overline{L_m(\mathbf{V}/\mathbf{G})}$ | Prefix closure of language $L_m(\mathbf{V}/\mathbf{G})$ | 39 |
| $Z$ | Backtracking forcibility analysis solutions set | 39 |
| $n^{\pi}$ | Motion step size of astrobot $\pi$ | 40 |
| $v^{\pi}$ | Forward event of astrobot $\pi$ | 40 |
| $w^{\pi}$ | Backward event of astrobot $\pi$ | 40 |
| $n_v^{\pi}$ | Forward counter of astrobot $\pi$ | 40 |
| $n_w^{\pi}$ | Backward counter of astrobot $\pi$ | 40 |
| $\mathcal{N}$ | Astrobots neighborhood | 40 |
| $r(\pi)$ | Relative priority of astrobot $\pi$'s target compared to those of its peers in its neighborhood | 40 |
| $l^{\pi}$ | Low-priority event of astrobot $\pi$ | 40 |
| $h^{\pi}$ | High-priority event of astrobot $\pi$ | 40 |
| $m^{\pi}$ | Medium-priority event of astrobot $\pi$ | 40 |
| $k^{\pi}$ | Collision event of astrobot $\pi$ | 41 |
| $e^{\pi}$ | Free event of astrobot $\pi$ | 41 |
| $\mathbf{G}^{\pi}$ | Plant model of astrobot $\pi$ | 41 |

| | | |
|---|---|---|
| $[\mathbf{0}]$ | State 0 of a generator | 41 |
| $\mathbf{S}^{\pi}$ | Specification model of astrobot $\pi$ | 42 |
| $\Vert$ | $n$-ary synchronous product operator | 43 |
| **supcon** | Binary operator which computes a nonblocking supervisor corresponding to its first-argument generator with respect to its second-argument generator | 43 |
| **allevents** | Unary operator which returns a one-state generator with a selfloop including all of the states of its argument generator | 43 |
| $\Sigma_{\mathbf{G}}^{*}$ | All possible strings over alphabet $\Sigma_{\mathbf{G}}$ | 44 |
| $\mathcal{F}(s,q,q')$ | Forcibility relation given string $s$ reacing state $q'$ from state $q$ | 44 |
| $\langle 73 \rangle$ | Event #73 | 45 |
| $\boldsymbol{q}^{\boldsymbol{i}}_{\star}$ | Equilibrium position of astrobot $\pi^{i}$ | 49 |
| $\mathcal{I}_{\mathcal{N}}$ | Index set including indices of set $\mathcal{N}$ | 49 |
| $\mathcal{V}^{i}$ | Neighbor set of astrobot $\pi^{i}$ | 50 |
| $\mathcal{N}^{i}$ | Neighborhood of astrobot $\pi^{i}$ | 50 |
| $\mathcal{C}(\mathcal{N}_{i})$ | Local completeness relation with respect to the neighborhood of astrobot $\pi^{i}$ | 50 |
| $\mathcal{C}(\mathcal{N})$ | Global completeness relation with respect to neighborhoods of an astrobots swarm | 50 |
| $\lambda_1$ | Attractive weighting factor | 52 |
| $\lambda_2$ | Repulsive weighting factor | 52 |
| $D$ | Radius of collision avoidance envelope in which repulsive term is activated | 52 |
| $d_0$ | Radius of safety region around each astrobot | 52 |
| $\boldsymbol{\lambda_1}$ | Attractive weighting vector | 52 |
| $\otimes$ | Kronecker product operator | 52 |
| $\mathbb{I}_2$ | 2-all-1 vector | 52 |
| $\boldsymbol{\lambda_2}$ | Repulsive weighting vector | 52 |
| $\psi(\boldsymbol{q^{i}})$ | Reference artificial potential field corresponding to astrobot $i$ | 52 |
| $\rho(\boldsymbol{q^{i}})$ | Cooperative artificial potential field corresponding to astrobot $i$ | 53 |
| $\boldsymbol{u^{i}}$ | Control law associated with astrobot $i$ | 53 |
| $P(m,n)$ | Permutation of $n$ selections out of $m$ options | 59 |
| $\delta\lambda$ | Variation step | 65 |
| $\overline{\delta\lambda}$ | Step array | 66 |
| $(r_f,\theta_f)$ | Polar coodinate system of focal plane | 67 |
| $\mathcal{U}$ | Uniform normal generator | 67 |
| $r_{\max}$ | Radius of focal plane | 67 |
| $N(x,y)$ | Probability density function of bi-variate normal distribution in Cartesian coordinate | 68 |

|  |  |  |
|---|---|---|
| $c$ | Correlation coefficient | 68 |
| $k$ | Number of neighbors | 76 |
| $\boldsymbol{P}$ | Configuration matrix | 77 |
| $\boldsymbol{m}$ | Frequency vector | 78 |
| $\boldsymbol{v}$ | Pseudo vector | 78 |
| $\boldsymbol{\mathfrak{w}}$ | Weight vector | 78 |
| $\alpha$ | Corrector co-efficient | 78 |
| $\beta$ | Corrector co-efficient | 78 |
| $\boldsymbol{T}$ | Test configuration | 79 |
| $\Delta(\cdot,\cdot)$ | Metric distance | 79 |
| $\boldsymbol{P}^{\boldsymbol{T},k}$ | $k$-closest configurations set | 79 |
| $\text{fetch}(\text{set},k)$ | Operator returning the first $k$ elements of sorted set | 79 |
| ${}^{\mathfrak{w}}\boldsymbol{g}^{\boldsymbol{P}^{\boldsymbol{T},k}}$ | Weighted ground truth vector | 79 |
| $\hat{\boldsymbol{\Gamma}}^{\boldsymbol{P},\boldsymbol{T}}$ | Primary prediction probability vector | 80 |
| $\oslash$ | Hadamard division operator | 80 |
| $\boldsymbol{\nu}_{\pi}$ | Neighborhood matrix of astrobot $\pi$ | 80 |
| $\boldsymbol{\eta}$ | Counter vector | 80 |
| $\widetilde{\boldsymbol{\Gamma}}^{\boldsymbol{\nu}_{\pi},\boldsymbol{T}}$ | Neighborhood probability vector | 80 |
| $\boldsymbol{\Gamma}^{\boldsymbol{P},\boldsymbol{T}}$ | Final probability vector | 80 |
| $q$ | Decision filter | 80 |
| $\boldsymbol{Y}$ | Output vector | 80 |
| $\boldsymbol{\pi}^{\mathrm{C}}$ | Constrained model of astrobot $\pi$ | 92 |
| $x$ | Feature | 93 |
| $x'$ | Normalized feature $x$ | 93 |
| $\kappa(\cdot,\cdot)$ | Kernel | 93 |
| $\Phi$ | Kernel function | 93 |
| $\sigma$ | Kernel size | 93 |
| $\boldsymbol{w}$ | Normal vector | 94 |
| $\mathcal{C}_0$ | Miss-classification penalty of class 0 | 94 |
| $N_0$ | Numbers of samples of class 0 | 94 |
| $\xi_i^0$ | Slack variable corresponding to sample $i$ of class 0 | 94 |
| $\mathcal{C}_1$ | Miss-classification penalty of class 1 | 94 |
| $N_1$ | Numbers of samples of class 1 | 94 |
| $\xi_i^1$ | Slack variable corresponding to sample $i$ of class 1 | 94 |

| | | |
|---|---|---|
| $b$ | Hyperplane intercept | 94 |
| $\mathcal{C}$ | Miss-classification measure | 94 |
| $\omega_1$ | Class weight of class 1 | 94 |
| $\mathtt{D}_\mathtt{T}$ | Train partition of dataset $\mathtt{D}$ | 94 |
| $\mathtt{D}$ | Dataset | 94 |
| $\mathtt{D}_\mathtt{S}$ | Train partition of dataset $\mathtt{T}$ | 94 |
| $\boldsymbol{\pi}^\mathrm{G}$ | Generalized model of astrobot $\pi$ | 95 |
| $P^i$ | Parity of astrobot $i$ | 95 |

# Chapter 1

# Preface

Publication Note

The material presented in this chapter is adapted from:

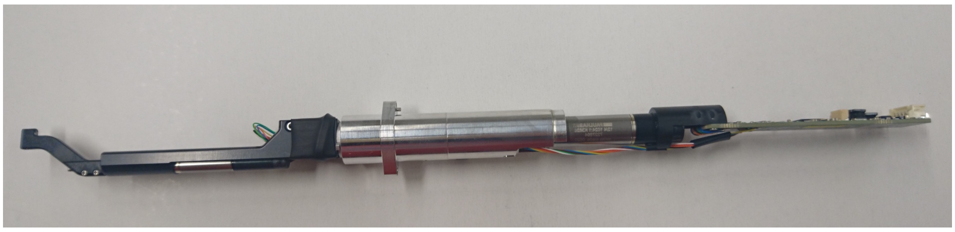
- **Matin Macktoobian**, Ricardo Araújo, Loïc Grossen, Luzuis Kronig, Mohamed Bouri, Denis Gillet, and Jean-Paul Kneib, Experimental Evaluation of Complete Safe Coordination of Astrobots for the Sloan Digital Sky Survey V, *Experimental Astronomy* 51(1), pp. 77-94, 2021.

## 1.1 An Introduction to Astrobotics

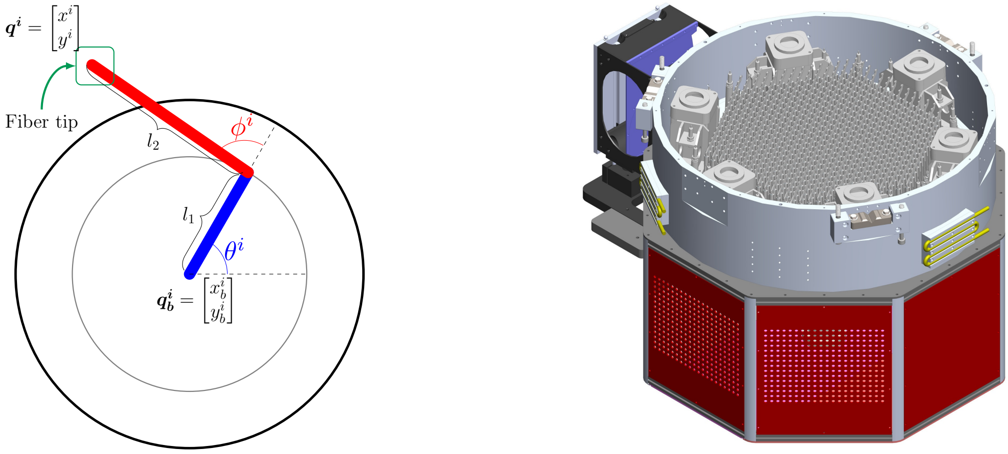
### 1.1.1 Massive Spectroscopic Surveys and SDSS-V

Dark energy studies [Joyce et al. 2016] have been revolutionized once the accelerated expansion of the universe was observed [Schrabback et al. 2010]. In particular, the evolution of the universe, which has been under intense scrutiny over the recent decades, is found to be correlated with the distribution of dark matter all over the cosmos. A data-driven strategy to obtain the desired distribution requires abundant mass-energy recording of the universe. Accordingly, the map of the observable universe is expected to convey valuable information about the geometry and the evolution of the cosmos. Universe dynamics vary to get expanded over time. Thus, redshift-based observation strategies effectively provide various volumes of the space in the course of different cosmological era associated with the age of the universe. In particular, the measurements of baryonic acoustic oscillations (BAO) [Seo et al. 2003] have already shed light on the filament-void interactions of the cosmos [Forero–Romero et al. 2009]. So, BAO analysis with respect to various redshift ranges is known to generate spectroscopic surveys which eventually yield a significant repertoire of the cosmological data to investigate the universe's evolution.

The dependence of spectroscopic surveys to redshift constitutes interactions between mass and energy throughout the history of the universe during which one of them mostly dominates. Namely, low-redshift surveys [Hamuy et al. 2006] extensively contributes to the study of dark energy, while high-redshift ones [Takada et al. 2006] are often the target to investigate the universe when its radiation is dominated by its mass. To be specific, the Anglo-Australian Telescope (AAT) [Blake et al. 2011] was used to generate surveys associated with star-forming galaxies at lower redshifts, say, $z < 0.8$. Middle-range redshifts, such as $0.7 < z < 2.2$, have been widely taken into account by various instruments, e.g., SDSS telescope.

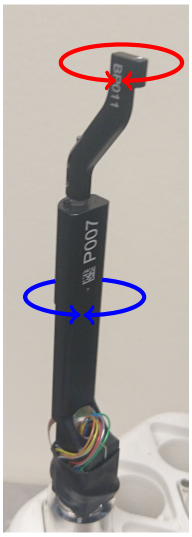


(a) An SDSS-V astrobot


$q^i = \begin{bmatrix} x^i \\ y^i \end{bmatrix}$
Fiber tip
$l_2$
$\phi^i$
$l_1$
$\theta^i$
$q_b^i = \begin{bmatrix} x_b^i \\ y_b^i \end{bmatrix}$

(b) The astrobot top view

(c) The schematic of SDSS-V focal plane



(d) The rotational motion representation of a typical astrobot

Figure 1.1: A typical SDSS-V astrobot and overall focal plane (Figure 1.1c is reprinted from the SDSS wiki [S. Consortium n.d.].)

The resulting baryonic oscillation spectroscopic survey [Dawson et al. 2012] is supposed to cover almost 1.4 million galaxies and quasars. High-redshift spectroscopic surveys (at $2 < z < 5$) are of utmost importance in the space-time inflation studies and dark energy observations, as studied by [Ferraro et al. 2019]. The combination of BAO analysis and redshift-space distortions [Scoccimarro 2004] seems to be promising enough to evaluate General Relativity with respect to various cosmological scales [D. J. Schlegel et al. 2019]. To this aim, massive spectroscopic surveys have to be planned which essentially include huge numbers of optical fibers, multi-segment focal planes, and vast telescope apertures.

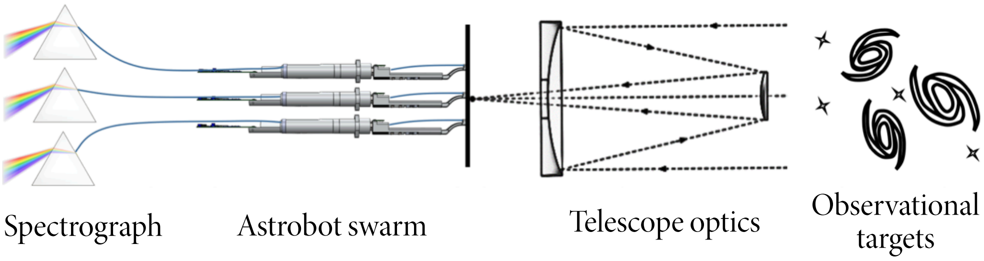


Figure 1.2: The multi-fiber spectroscopy operation process (partially reprinted from [Kronig 2020] with permission)

Sloan Digital Sky Survey (SDSS) represents a family of spectroscopic projects aiming to the generation of surveys using various observational technologies in different ground telescopes. Early 2000s witnessed the first generation of these projects, i.e., SDSS-I [Zehavi et al. 2002]. This project, based at APO, covered various spectral bands using camera-based photometry. APO later hosted SDSS-II [Ivezić et al. 2008] which yielded the first class of the multi-object optical spectroscopic surveys of the SDSS family. This project exclusively used optical fibers to collect visible lights of its desired targets. Then, the researchers' attention was shifted to near-infrared rays of targets which can also be captured by ground telescopes. Consequently, SDSS-III [Alam et al. 2015] ended up with the APOGEE survey supported by a near-infrared multi-object spectrograph at APO. The first three classes of SDSS projects were executed at APO in the northern hemisphere. To cover data acquisition from the southern hemisphere, SDSS-IV [Bundy et al. 2014] collected spectroscopic data, based on a regime similar to its predecessors, at Las Campanas Observatory (LCO). This project extensively covered many surveys such as APOGEE-2 [Zasowski et al. 2017], MaNGA [Wake et al. 2017], and the extended Baryon oscillation spectroscopic surveys [Dawson et al. 2016]. SDSS-V [Kollmeier et al. 2017] is about to conduct multi-object spectroscopic surveys in both hemispheres, namely, using Sloan Foundation telescope at APO in the northern hemisphere [J. E. Gunn et al. 2006] and the du Pont telescope at LCO in the southern hemisphere [Way et al. 2005]. Both types of optical and near-infrared fibers are used in this project which will be fed into a pair of optical BOSS and near-infrared APOGEE spectrographs. The resulting surveys are expected to extensively contribute to the characterization of Milky Way galaxy.

Massive spectroscopic survey projects, see, Figure 1.2, such as DESI [Dey et al. 2019], MOONS [Michele Cirasuolo et al. 2014], PFS [Takada et al. 2014], LAMOST [Cui et al. 2012], and in particular SDSS-V, include hundreds to thousands of fibers to maximize the information throughout each observation in a mission-wise manner. For this purpose, one may increase the aperture size of a host telescope to have a larger focal plane, thereby covering larger number of fibers. Second, fibers have to be placed in more dense formations to tile the area of the focal plane with higher observational resolution. However, fiber multiplexing raises operational challenges. In particular, fibers have to point to different locations of their fields from one observation to another. So, their reconfiguration needs to be performed in the available spare time between consecutive observations. To hit this mark, fibers were manually replaced in early versions of the surveys using SDSS spectrograph [Smee et al. 2013]. However, given the gradual increment of the employed fibers, robot fiber placement was taken into account in the case of AAT spectrograph. The cited process was inefficient because of the lack of any parallelism in the coordination of fibers. Then, the first generation of robotic coordination showed up in the case of fiber multi-object spectrograph (FMOS) [Kimura et al. 2010], attached to LAMOST [G. Zhao et al. 2012], and MOONS [Cirasuolo et al. 2016]. The

Table 1.1: The plan of the past, current, and future spectroscopic surveys (partially adapted from [Ferraro et al. 2019])

| Spectroscopic instrument | Deployment year | Number of fibers | Coordination |
|---|---|---|---|
| SDSS-I | 1991 | 640 | Manual |
| BOSS | 2009 | 1000 | |
| MOONS | 2013 | 1000 | Automatic (robotized) |
| DESI | 2019 | **5000** | |
| PFS | 2020 | 2400 | |
| 4MOST | 2022 | 1624 | |
| MegaMapper | TBA | **20,000** | |
| FOBOS | | 1800 | |
| MSE | | 3249 | |
| LSSTspec | | **8640** | |
| Spectel | | **15,000** | |

presence of astrobots in the various massive spectroscopic surveys is expected to be noticeable, as stated in Table 1.1.

### 1.1.2 Astrobots Characterization

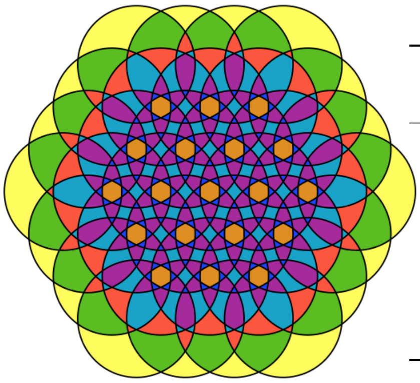
#### 1.1.2.1 Mechanical Specification

Each astrobot is a SCARA-like two-degree-of-freedom rotational-rotational robotic manipulator [Araujo et al. 2020] which is sketched in Figure 1.1a. An optical fiber is mounted onto the astrobot such that its tip is located at the end-effector of the astrobot, called ferrule. Rotational combinations of the astrobot's arms move the ferrule in the working space of the astrobot such that it can reach some targets corresponding to a planned observation. Astrobots are located in a hexagonal formation to maximize the rigidity of the overall focal plane. Various kinds of fibers exist in terms of their manufacturing technology and their connections to spectrographs, such as monolithic fibers (SDSS-I and BOSS projects [Dawson et al. 2012]), connectorized fibers (SDSS-V project), and plasma-fused fibers (as done for DESI project). A target may technically be projected at any point on a focal plane [Pogge et al. 2020]. In the worst case, if a target is projected exactly at the base location of an astrobot, a neighboring peer has to be able to access it. Thus, the overall chain of the arms of each astrobot needs to be at least as long as the distance between its base spot to that of its neighbors. Given[1] an astrobot $i$ associated with a swarm, let $\boldsymbol{q_b^i} = \begin{bmatrix} x_b^i & y_b^i \end{bmatrix}^\intercal$ be the coordinate of the astrobot's base. This coordinate can be defined as a part of a universal frame attached to the focal plane of its host telescope. Or, it can be simply a relative coordinate with respect to those of the astrobot's neighbors. The lengths of the links are denoted by $\boldsymbol{l} = \begin{bmatrix} l_1 & l_2 \end{bmatrix}^\intercal$. Then, one is interested in the location of the ferrule, say, $\boldsymbol{q^i} = \begin{bmatrix} x^i & y^i \end{bmatrix}^\intercal$, as stated below, given the angular deviations of the astrobot's arms represented by $\theta^i$ and $\phi^i$.

$$\boldsymbol{q^i} = \boldsymbol{q_b^i} + \begin{bmatrix} \cos(\theta^i) & \cos(\theta^i + \phi^i) \\ \sin(\theta^i) & \sin(\theta^i + \phi^i) \end{bmatrix} \boldsymbol{l} \tag{1.1}$$

---

[1] Throughout this thesis, scalars and matrices are represented by regular and bold symbols, respectively.



| Number of fibers reaching area | Color |
|---|---|
| 1 | Yellow |
| 2 | Green |
| 3 | Red |
| 4 | Light blue |
| 5 | Purple |
| 6 | Dark blue |
| 7 | Orange |

Figure 1.3: The overlapping areas in a typical neighborhood

The quoted parameters are depicted in Figure 1.1b. A typical spectrograph, which generates a survey corresponding to the optical signals of an observation, is illustrated in Figure 1.1c. Figure 1.1d represents the rotational motion of the arms of a typical astrobot.

The torque of each astrobot is a nonlinear function of various factors governed by the following law [Kronig 2020].

$$J = \frac{\int_{t1}^{t2} \left(\tau - F - F_f - T_s\right) dt}{\dot{\theta}_{t2} - \dot{\theta}_{t1}} \tag{1.2}$$

In the equation above, $J$ is the total moment of inertial of an astrobot. The temporal range in the course of which the formula represents the dynamics of torque is $[t_1, t_2]$. $\tau$ is the torque. $F$ and $F_f$ denote the force of the motor and friction, respectively. $T_s$ is the spring-reduction torque term. The rotational velocity of end-effector at time $t$ is $\dot{\theta}_t$.

The larger the workspace of an astrobot is, the more targets may be potentially reachable by its ferrule during each observation. So, the axes of each astrobot's arms are slightly tilted rather than completely straight. In this case, the radius of the stated workspace is larger than the area reachable by a straight arrangement of arms. In the scale of an astrobot swarm, the accumulation of all such trivial tilts gives rise to the significant expansion of reachable areas on focal planes.

Each astrobot is actuated by the technology of brushless DC motors according to which high sensorless repeatability is realized. The smallest available diameter of these motors is 2 mm, which is favorable to manufacture miniaturized astrobots. These motors require complex control, and they don't hold torque. The precision positioning is the second important requirement of of astrobot coordination, after their safety. In this regard, the maximum precision positioning of the SDSS-V astrobot model is $5\mu$m.

The pitch of each astrobot, i.e., the overall length of its arms, has to at least reach the center of each of its neighboring peers, so the whole area of their focal plane is reachable. Due to the dense formation of astrobots, that relatively large pitch gives rise to the existence of some areas on a focal plane which are reachable by more than one astrobot. The more astrobots may can reach an area, the more challenging the coordination of those astrobots are because they may collide in those areas[2]. Figure 1.3 renders the critical areas in a typical astrobot neighborhood which are prone to collisions.

---

[2] One may argue about the usage of proximity sensors to manage collisions. However, these artifacts occupy extra spaces in ferrule areas. So, they by themselves pose more deadlock situations. On the other hand, miniaturized sensors are often expensive to be used for large packs of astrobots.

#### 1.1.2.2 Communications and System Interface

Each astrobot uses a microcontroller on which a firmware controls the astrobot. This firmware is equipped with a bootloader to change the different operational modes of an astrobot. It also allows error corrections and motion command transfer from a path generator to astrobots. This firmware can calibrate motors and detect both motor malfunctionalities and collisions in short ranges. Thanks to a syncline signal embedded into this framework, one can broadcast control signals to all astrobots of a focal plane via a CAN bus. Thus, astrobots simultaneously starts to move in a synchronized manner.

Control distribution is the most reliable and efficient strategy which has been used to govern multi-agent systems [Hsu 1993; Tan et al. 2013]. In the case of astrobots, one first needs to plan the generation of trajectories. This task is often very labor-intensive in view of desired computational resources because collision avoidance and completeness seeking have to be taken into account for all of the astrobots of a swarm. Thus, trajectories shall be computed on a central workstation according to a particular observation as well as the initial configuration of the swarm. A workstation is connected to a spectrograph and the instrumentation subsystems of the focal plane of a host telescope. Each trajectory encompasses two arrays of data corresponding to each astrobot. Each array includes position-time tuples to determine the location of each arm of the astrobot at any particular moment. These arrays are sent to an embedded supervisor mounted on the astrobot via a set of CAN buses. A microcontoller buffers the trajectory of its astrobot prior to its execution. The supervisor mainly governs two responsibilities. First, it buffers the trajectory arrays and interpolates them to continuous signals to be sent to the astrobot's motors. Second, the supervisor microcontroller monitors the status of its astrobot to keep track of any illegal trajectory values which are hazardous in view of safety.

A spectrograph only calls for the optical signals collected by astrobots at the end of each observation. The spectrograph may lack any embedded register to first cash all signals sent by astrobots before their processing. Then, a synchronized data transfer mechanism may be required to orchestrate the flow of the data from the focal plane to the spectrograph. Another interface-related feature may exhibit in the case of multi-spectrograph telescopes. In particular, each spectrograph specially operates in a specific range of redshift, either low [Meiring et al. 2011] or high redshift [Anderson et al. 2001]. However, not all observations assigned to a telescope necessarily correspond to a single redshift range. In this case, the telescope has to be equipped with more than one spectrograph. Switching from one spectrograph to another requires memory reset of all astrobots and potentially their re-calibration. Thus, the communication protocol between astrobots and their workstation needs to manage such high-level reconfigurations, as well. A more detailed description about the specification of the communication and the interface subsystems of astrobots are presented in [Hörler 2018].

## 1.2 Motivation

The more data one collects in the course of an observation, the higher the resolution of the resulted survey will be. Observational astronomers are specially interested in such detailed surveys since these surveys may reveal more information regarding the distribution of dark energy associated with its particular observation. The maximum throughput of an observation is directly associated with the complete coordination of all astrobots in the course of that observation. In this case, the optical rays of all of the targets that observation are collected.

The convergence rates of astrobots in the first generations of robotized surveys were way behind completeness. Since any coordination has to guarantee collision-less motions of astrobots, the coordination control problem of astrobots in indeed challenging in view of not only fulfilling the required safety but also realizing the convergence of astrobots. In particular, a variation of navigation functions, say, priority-based ones, were used for MOONS astrobots[Mathews et al. 2006; Makarem et al. 2014]. This method could generally not completely coordinate massive swarms of astrobots. In this method, a noticeable pairs or triplets of confronting astrobots were trapped in deadlocks when their targets were located behind one of their confronting peers. So, they perpetually oscillated in such situations without any eventual convergence to their target spots. Later, a revised version of the cited strategy was taken into account in which a new automaton-based decision layer was added to the navigator of the algorithm [Tao et al. 2018]. This layer directly resolves the deadlocks, which can not be handled by the main navigation function, based on a priority-based set of criteria. Thus, convergence rate reached $\sim$85%.

In this thesis, we seek the complete safe coordination of astrobots. To be specific, we take various operational stages of a general observation-to-survey process into account in an algorithmic perspective. Then, we propose new approaches in each stage which contributes to either the improvement of convergence rates or reaching completeness. Figure 1.4 depicts the domains of the complete coordination problem of astrobot swarms which are considered in this thesis. Section 1.3 briefly outlines the goals, the underlying ideas, and the accomplishments associated with each domain. We also cite the publications whose contributions have constructed the core of this thesis. The detailed description of the contributions corresponding to each domain will be scrutinized in a separate part of this thesis. The organization of the thesis, in terms of the dependencies between its various chapters, is expressed in Section 1.4.

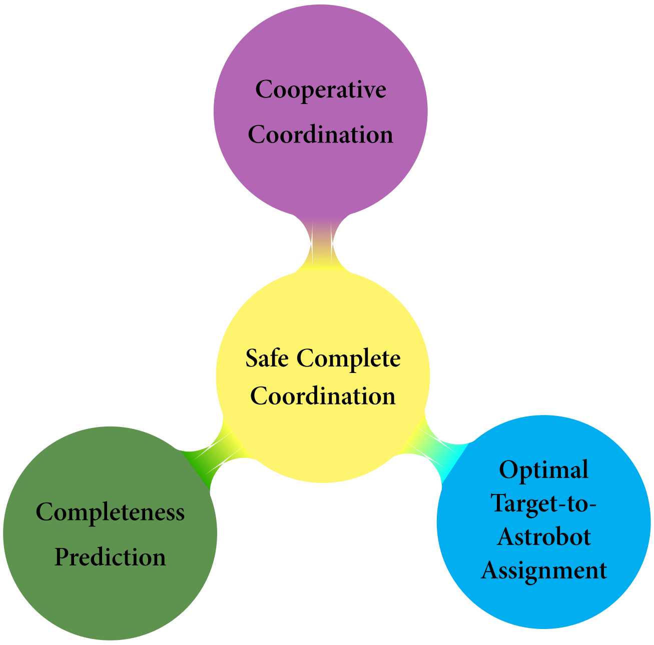

Cooperative Coordination
Safe Complete Coordination
Completeness Prediction
Optimal Target-to-Astrobot Assignment

Figure 1.4: The domains of the safe complete coordination problem

## 1.3 Research Goals and Contributions

### 1.3.1 Optimal Target-to-Astrobot Assignment

Target-to-astrobot assignment is an important operational stage in survey generations which determines the initial and the final configurations of each astrobot in the course of its coordination. A class of strategies have already been used to allocate astrobots to observational targets such as random assignment, drainage algorithm [Morales et al. 2012], flow-based assignment [Blanton et al. 2003], target-based assignment [Schaefer et al. 2016]. However, none of them is purposefully and efficiently designed to facilitate the coordination planning of potential navigators toward reaching high convergence rates and even potentially completeness.

To this aim, Part A of this thesis is dedicated to the presentation of novel target-to-astrobot assignment methods. In particular, **Chapter 2** illustrates an optimal technique which assigns targets to astrobots according to the minimization of a particular cost function. The algorithm using this cost function seeks the maximum spatial distribution of assigned astrobots in a swarm as well as the minimum coordination of them to reach their targets. Thus, we observe that, should one use this strategy to handle target-to-astrobot assignments, navigators' coordination results in terms of the convergence rates and times of astrobots will be much better compared to the results obtained by the methods cited above [Macktoobian et al. 2020a].

A subset of the astrobots of a particular focal plane may be even used in other domains such as space debris tracking [Mehrholz et al. 2002] as a potential venue of astrobots usage. In such a scenario, the types of targets are not longer all the same. In other words, we face a level of heterogeneity among those classes of astrobots which are supposed to be used for different applications, e.g., a group for spectroscopic survey generations and another for the space debris tracking. Thus, **Chapter 3** proposes a target-to-astrobot assignment algorithm which takes various characteristics of such heterogeneous scenarios into account to optimally allocate astrobots to desired applications [Macktoobian et al. 2019b].

### 1.3.2 Complete Coordination

We already noted that the former instances of navigation functions can, generally speaking, only coordinate a subset of the astrobots of a ground telescope. Thus, Part B of this thesis focuses on new navigation methods which provide completeness in the course of general coordination.

A disadvantage of navigation-function perspective in coordination control development for astrobots is that the functional and safety verification of the coordination of an astrobot swarm are often checked using intensive simulations. Namely, these swarms are so complicated that formally-verifiable methods have not been found for them. To simultaneously fill this gap and achieve complete safe coordination solutions, we model the coordination problem of astrobots as a reconfiguration problem [Macktoobian et al. 2017] in the domain of discrete-event systems [Cassandras et al. 2009]. Thanks to the verification-friendly nature of supervisory control theory [Wonham 2017b], **Chapter 4** illustrates a procedure to synthesize coordination supervisors for astrobot swarms whose logic-based behaviors are automatically safe and completeness-preserving.

The supervisory control above quickly faces what most of the large-scale applications using supervisory control theory eventually encounter: the curse of dimensionality. Once the number of astrobots increases, the state/event size of a synthesized coordination supervisor intractably grows. So, finding coordination paths of the closed behavior of the supervisor would resembles searching for a needle in a haystack. Instead, we bring the notion of navigation functions into focus again. Various formulations of these mathematical artifacts have already been used in the navigation of different robotic systems including aerial vehicles [Roelofsen et al. 2016; Roelofsen et al. 2017]. One observes that an astrobot is controlled by a specific navigation function instance. In such a distributed framework, each priority-based navigation function instance [Makarem et al. 2014] only cares about the convergence of its own astrobot in a selfish manner. It implies that an astrobot which has already reached its target actually refuses to move anymore; even if it has blocked the paths of some of its neighboring peers to reach their targets. In these scenarios, if the converged astrobot slightly deviates from its equilibrium temporarily, its peers may cross over the deadlock intersection of theirs toward their targets. Then, the astrobot may get again back to its desired spot. This level of cooperativity and its insertion into the formulation of navigation functions is the subject of **Chapter 5**. In particular, the resulting notion of cooperative artificial potential field [Macktoobian et al. 2019a; Macktoobian et al. 2019d] is flexible enough to completely coordinate astrobot swarms in a majority of observation scenarios. This formalism provides a completeness condition, based on the settings of targets' positions and astrobots' initial positions, to check the feasibility of complete coordination. If completeness is not feasible, a simple procedure can be used to trivially modify the swarm settings to achieve completeness. The applied simulations and experimental evaluations illustrate that incomplete coordination rarely occur when cooperative artificial potential fields are used to coordinate astrobot swarms. However, even in the presence of such partially-coordinated cases, incompleteness may be straightforwardly resolved in just one or two rounds of parameter modification [Macktoobian et al. 2021b].

### 1.3.3 Completeness Prediction

Analytical methods noted in the previous section are computationally intensive, specially when one moves from small packs of astrobots to huge ones. In such cases, one has to employ a lot of time and computational resources to check whether a particular setting of targets and astrobots may end up with their complete convergence. If the answer is negative, then the same amount of time and resources have to be used after each round of parameter modification. Thus, it is beneficial if one can first estimate a rough completeness likelihood with respect to particular planned astrobot-target pairs. If the estimation is lower than a threshold, one can simply ignore the setting without further verification simulations and immediately modify the parameters of the swarm to reach a larger likelihood. Once the predicted convergence rate passes the minimum chance to reach completeness, one can verify it using simulations. Thus, Part C of this thesis exhibits how machine learning algorithms may be used to obtain desired completeness predictions. The remarkable feature of these algorithms is that they only need the target-astrobot pairing to evaluate completeness likelihoods, whereas analytical methods indeed have to iteratively solve a multitude of coupled differential equations in every step of a coordination. Thus, the computational benefits of convergence predictions are notable.

Parity of an astrobot is the rotation direction of its second arm. This parameter may be either constrained, say, the second arm of an astrobot can only move in one direction, or variable, thereby, moving in both directions. A constrained-parity astrobot is less flexible in terms of control, but easier to be predicted in terms of its convergence. In this regard, **Chapter 6** introduces an algorithm based on the idea of weighted $k$-NN classifiers [Macktoobian et al. 2020b], in which parities are assumed to be constrained. The performance metrics of our classifier display its effectiveness in the correct prediction of converging astrobots in ∼75% of applied tests. In the next step, we relax the constrained assumption by letting parities vary. This assumption drastically complicates convergence predictions. In particular, in the case of variable-parity coordination, even a trivial change in the positional values of targets and/or astrobots may flip a convergence result from successful to unsuccessful or vice versa. For this purpose, **Chapter** 7 reports an SVM-based predictor which yields ∼78% accuracy in its convergence predictions [Macktoobian et al. 2020c].

### 1.3.4 List of Publications

**The publications included in this thesis**

The publications listed below constitute the majority of the content of this thesis. The first author of them, i.e., the author of this thesis, defined their problem statements, developed their methods, performed experiments to assess their underlying strategies, and wrote the manuscripts.

9. **M. Macktoobian**, F. Basciani, D. Gillet, and J-P. Kneib, Data-Driven Convergence Prediction of Astrobot Swarms, *IEEE Transactions on Automation Science and Engineering*. (Under review)

8. **M. Macktoobian**, F. Basciani, D. Gillet, and J-P. Kneib, Learning Convergence Prediction of Astrobots in Multi-Object Spectrographs, *Journal of Astronomical Telescopes, Instruments, and Systems*. (In Press)

7. **M. Macktoobian**, D. Gillet, and J-P. Kneib, Astrobotics: Swarm Robotics for Astrophysical Studies, *IEEE Robotics and Automation Magazine*, 2021.

6. **M. Macktoobian**, R. Araújo, L. Grossen, L. Kronig, M. Bouri, D. Gillet, and J-P. Kneib, Experimental Evaluation of Complete Safe Coordination of Astrobots for the Sloan Digital Sky Survey V, *Experimental Astronomy* 51(1), pp. 77-94, 2021.

5. **M. Macktoobian**, D. Gillet, and J-P. Kneib, Optimal Target Assignment for Massive Spectroscopic Surveys, *Astronomy and Computing* 30, Article 100364, 2020.

4. **M. Macktoobian**, D. Gillet, and J-P. Kneib, Complete Coordination of Robotic Fiber Positioners for Massive Spectroscopic Surveys, *Journal of Astronomical Telescopes, Instruments, and Systems* 5(4), Article 045002, 2019.

3. **M. Macktoobian**, D. Gillet, and J-P. Kneib, Heterogeneous Target Assignment to Robotic Fiber Positioner Systems, *Australian and New Zealand Control Conference (ANZCC)*, IFAC/IEEE, pp. 48-53, 2019.

2. **M. Macktoobian**, D. Gillet, and J-P. Kneib, The Navigation of Robotic Fiber Positioners in SDSS-V Project: Design and Implementation, in *15th Conference on PhD Research in Microelectronics and Electronics (PRIME)*, pp. 85-88, IEEE, 2019.

1. **M. Macktoobian**, D. Gillet, and J-P. Kneib, Supervisory Coordination of Robotic Fiber Positioners in Multi-Object Spectrographs, *15th IFAC Symposium on Large Scale Systems (LSS): Theory and Applications*, pp. 61-66, 2019.

**Related publications**

The author of this thesis has also contributed to the following related publications.

2. R. W. Pogge, M. A. Derwent, T. P. O'Brien, C. A. Jurgenson, D. Pappalardo, M. Engelman, C. Brandon, J. Brady, N. Clawson, J. Shover, J. Mason, J-P. Kneib, R. Araújo, M. Bouri, L. Kronig, L. Grossen, D. Gillet, **M. Macktoobian**, S. E. Tuttle, E. Farr, J. Sánchez-Gallego, and C. Sayres, A Robotic Focal Plane System (FPS) for the Sloan Digital Sky Survey V, *Ground-based and Airborne Instrumentation for Astronomy VIII, Proc. of SPIE 11447*, Article 1144781, 2020.

1. R. Araújo, M. Bouri, C. Brandon, S. Caseiro, D. Gillet, L. Grossen, L. Kronig, **M. Macktoobian**, T. P. O'Brien, R. W. Pogge, S. E. Tuttle, and J-P. Kneib, Design of a Theta/Phi Fiber Positioner Robot for the Sloan Digital Sky Survey V, *Ground-based and Airborne Instrumentation for Astronomy VIII, Proc. of SPIE 11447*, Article 1144790, 2020.

## 1.4 Organization

This thesis has started with the current introductory chapter. Each of the next three parts includes the contributions of one of the domains depicted in Figure 1.4. Our conclusions and remarks for future research themes in this field are the subject of the last chapter, i.e., **Chapter 8**. The overall dependencies between the chapters of the thesis are explicitly stated in Figure 1.5. In particular, **Chapter 2** initiates the thesis contribution in the scope of target-to-astrobot assignment. The terminology of this chapter is extensively used in **Chapter 3**. **Chapter 3** illustrates a potential application of astrobots to space debris tracking. So, no other chapter depend on it. **Chapter 4** and **Chapter 5** both enjoy the contributions of **Chapter 2**, since the optimal target-to-astrobot assignment is utilized in both supervisory and cooperative coordination methods. The complete coordination problem is carefully explained in **Chapter 4**. **Chapter 5** also requires those information, thereby depending on **Chapter 4**. **Chapter 6** requires results of **Chapter 5**, say, cooperative coordination, as a means of data acquisition for its learning processes. **Chapter 7** generalizes the constrained parity idea of **Chapter 6**. **Chapter 8** includes the final words corresponding to all of the quoted chapters.

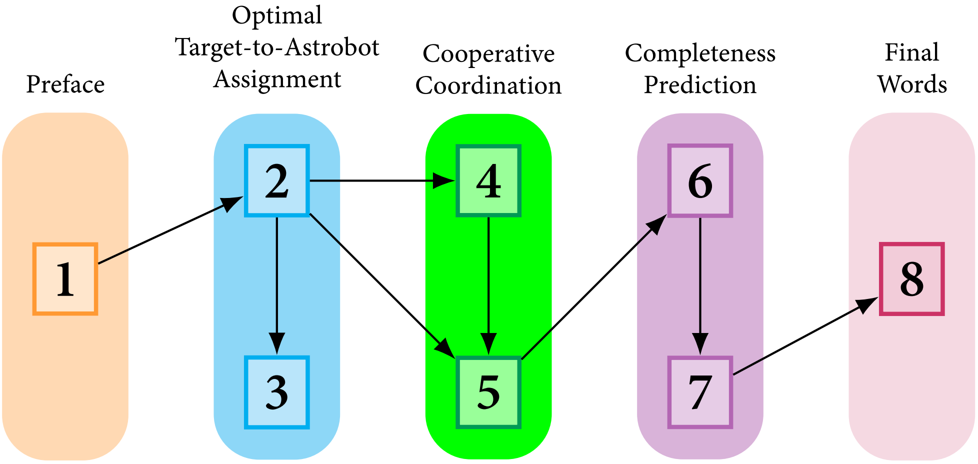


Figure 1.5: The chapter dependencies in the thesis

# Part A

# Target-to-Astrobot Assignment

## Chapter 2

# Optimal target-to-astrobot assignment

### Publication Note

The material presented in this chapter is adapted from:

- **M. Macktoobian**, D. Gillet, and J-P. Kneib, Optimal Target Assignment for Massive Spectroscopic Surveys, *Astronomy and Computing* 30, Article 100364, 2020.

## 2.1 Introduction

THE target-to-astrobot assignment process may be extremely influential on the quality of the final coordination process in terms of convergence rate and collision freeness. In this chapter, we posit that the target-to-fiber assignment problem is a constrained and specialized derivation of the set-to-set assignment problem [Cattrysse et al. 1992]. Thus, we review the classical solutions proposed to the latter. We also describe how those potential solutions may not be properly applicable to our target-to-fiber assignment problem. In other words, we aim to justify the effectiveness of our constrained optimality idea compared to former solutions to the set-to-set assignment problem. Since our strategy is established based on optimization, we basically focus on the comparative study of the optimization-driven methods.

Classical optimization modeling based on constrained formulas have been studied for a long time, see, e.g., [Fisher et al. 1986; Ross et al. 1977]. Most of the available schemes in this area often define an optimization problem including a set of linear constrained to take the requirements of an assignment problem into account in the course of optimization. Many applications have used this strategy including optimal routing problem [Fisher et al. 1981; Cantor et al. 1974], resource scheduling in large-scale project management [Balachandran 1976; Menesi et al. 2013], etc. The solutions to these problems have to be recursively achieved because of the dynamic nature of their constraints. Thus, constraints are mostly defined as inequalities. To do so, a procedure called LP-relaxation [Gavish et al. 1991; Sanghavi et al. 2011] first replaces are linear equality constraints of a system to their inequality equivalents. The resulting set of inequalities takes all dynamic complicated requirements into account at the cost of very time-consuming recursive computations. Thus, a class of heuristics [Van Den Briel et al. 2007; Klose 1999] were designed to improve the performance of traditional LP-relaxation-based methods. Despite the achieved performance improvements, a drawback of the heuristic LP-formulation is its degeneracy. Put differently, increasing the system and the constraints set size (at the level of hyper-complex systems) yields inefficient computation times tending to intractably grow. In terms of the requirements corresponding to our assignment problem, these methods do not seem to be promising for a variety of reasons. First, the number of our constraints are very smaller than those of existing in usual problems seeking LP-relaxation. Thus, transformations of our requirements to inequalities do not provide any advantage compared to pure equality-based constraints. Numerous recursions of LP-relaxation also seem to be unnecessary in the course of our problem since the convergence to the optimal point of our two spatial constraints is in fact enough not to seek extra recursive computations. Finally, LP-relaxation efficiently works if effective heuristics have already supported its formulation. Since we plan to solve our optimization assignment in a localized perspective, local geometrical interactions among astrobots are not so complicated that one has not to design specific heuristics for this purpose.

The next step to improve LP-relaxation is constraint deleting [Ross et al. 1975; Ceselli et al. 2006]. Namely, one observes that not all constraints contribute to decision making process in all recursions of optimization. So given a hypothetical constraint, the idea is its deletion as soon as it has no impact on the function subject to minimization. Constraints deletion requires to identify the constraints to be deleted in each optimization recursion. Thus, greedy methods emerged to modify the heuristics of LP-relaxations to fulfill such requirement [Martello et al. 1992]. Constraint deletion is a trade-off in terms of computational labor. In particular, one eliminates useless constraints in the course of the optimization algorithm, at the cost of permanent execution of the greedy elimination detector algorithm. The constraint deletion notion is implicitly taken into account in our optimal target-to-astrobot assignment idea. Namely, once an astrobot is assigned to a target, it will be removed from the set of available astrobots to be allocated associated with the next target. However, our implicit constraint deletion is fairly cheaper in a computational perspective compared to the one applied to LP-relaxation. The LP-relaxation version adds a constraint (set) to the whole process which is computationally more labor-intensive compared to what we do, say, removing one astrobot from a list of available astrobots before starting the next target-to-fiber assignment round.

Next improvement applied to the optimized set-to-set problem formulation is dualizing constraints [Fisher 2004]. This idea, known as Lagrangian relaxation [Geoffrion 1974], reduces the space which algorithm has to search in each iteration. However, one can show that it does not improve the performance of LP-relaxation when one faces an integer set-to-set optimization problem. One may note that our target-to-fiber assignment problem is not inherently an integer optimization problem in view of its constraints because our optimality criterion in general returns non-integer evaluations. Consequently, our strategy does not require the duality idea proposed by Lagrangian relaxation.

Lagrangian decomposition [Jörnsten et al. 1986; Escudero et al. 2016; Baumann et al. 2014] is proposed to transform a multi-variable set-to-set optimization problem to a set of single-variable ones aiming to reduce the computational complexity of the solution seeking process. The effectiveness of this method is only limited to those problems whose variables are independent [Karuppiah et al. 2008]. In contrast, our problem statement includes positional variables composed of Cartesian coordinates. The definition of two dimensional points does not simply authorize the detachment of a point's coordinate components. Moreover, the variable set of our assignment problem is restricted to only one type of spatial coordinates, thereby encountering no computational bottleneck in their manipulations in the course of optimization.

Set-to-set assignment problem may be approximated to multiple knapsack optimization problems [Dudziński et al. 1987]. If so, a noticeable simplification in optimization process shall be observed. However, the conditions required for such approximation are not met in our problem statement. In particular, this strategy assigns mutual weights to various elements of the domain set. Then given a pair of domain items, an optimization procedure assigns a potential range set item to that pair element for which the weight is the smallest compared to its other weights with respect to the remainder of domain set elements. But our target-to-fiber assignment problem does not hold this condition. Namely, our domain set is a set of independent observational targets without any mutual impact on each other. Even in the presence of hypothetically defined mutualities, it is not clear how this idea would pave the way for simplifying and/or improving our optimization procedure.

Pure heuristic solutions to the set-to-set assignment problem also exist. For example, evolutionary computational algorithms are proposed to solve the problem in dynamic scenarios where both domain and range sets may evolve in terms of their elements throughout assignment processes [Sethanan et al. 2016]. In a similar vein, parallel genetic algorithms [Y. Y. Liu et al. 2015] seem to be promising in considering some factors in dynamic scenarios, e.g., wider coverage of search spaces in the course of optimized assignments. However, our specific problem statement is static in this viewpoint. Thus, the advantages of differentially evolving methods are not noticeable in our problem's case. The lack of inherent safety control mechanisms in some other interesting assignment methods like Hungarian algorithm makes them inappropriate options to tackle our problem.

The set-to-set assignment methods specially designed for the purpose of assigning observational targets to fibers (or astrobots) were already discussed in the proposal in detail. To restate the main points, we note that random assignment [Doeleman et al. 2009], drainage method [Morales et al. 2011], and target-based strategy [Schaefer et al. 2016] do not take our optimality requirements into account. Flow-based assignment [Blanton et al. 2003] is another scheme which solves the target-to-astrobot assignment problem as a network flow one. The method aims to maximize the number of targets which have fibers assigned to them. This strategy uses network flow graphs to solve the problem. Namely, it first identifies a set of fibers which do not collide any other peers. Then, the set of unallocated fibers are taken into account to resolve collisions using a network flow graph. This method is basically designed to address target-to-astrobot assignment in non-robotic focal planes. In other words, however collisions among fibers are addressed by the method, it may not be able to properly handle collisions among astrobots. In automatized focal planes, collisions depend on real-time states of astrobots during coordinations. So in practice, it is unlikely to be able to consider all potential collisions corresponding to numerous coordination solutions of a system of astrobots in the assignment phase. Thus, modular perspective in decoupling assignment phase from coordination phase may be more effective to avoid collisions and to find fast and safe coordinations during coordination phases.

The spectroscopic survey generation is typically done in 3 phases. Namely taking a specific observation into account, one first has to assign each target of an observation to a specific astrobot of a planned telescope (*assignment phase*). Then, the astrobots have to be coordinated to reach their assigned targets such that no collision occurs during the *coordination phase* [Macktoobian et al. 2019d; Macktoobian et al. 2019a]. Finally, the collected spectroscopic signals are sent to a spectrograph to generate the map of the observation (*processing phase*). Because of the complexity of the system and the dense placement of astrobots, the coordination phase is often challenging to be handled. In particular, one is interested in fast collision-free coordinations whose convergence rates[1] are maximized. Targets are heterogeneous in view

---

[1] Given a set of astrobots, convergence rate is defined as the number of those which reach their desired position at the end of their coordination divided by their overall number.

of size and the distance from fibers. One may note that the assignment phase dramatically influences the achievement of the quoted favorite coordination. Thus, any improvement of the assignment process may directly contribute to the realization of safer collision-free and faster coordinations.

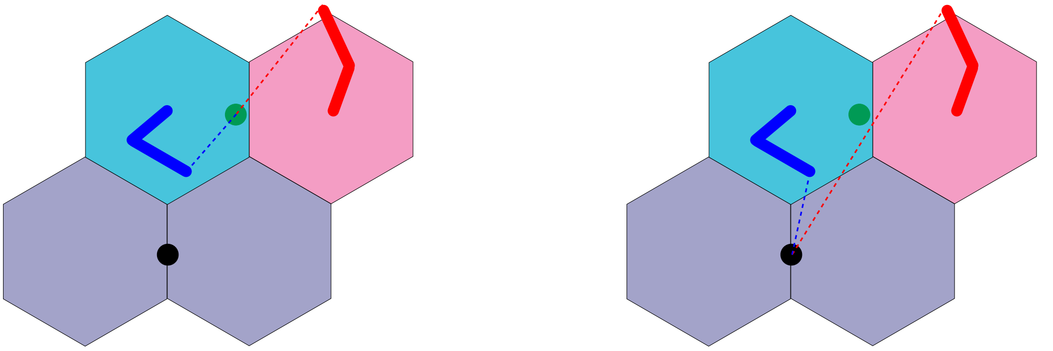
As already stated, each observation often requires a large number of objects to be assigned to a set of astrobots [Macktoobian et al. 2019b]. So, a set of methodologies were proposed to handle assignments. For example, random assignment simply assigns each target to a random unassigned astrobot. This method takes no specific criteria into account, thereby being computationally plausible. However, the coordination phase may become very challenging because random assignment may assign some targets to a dense neighborhood of astrobots. In this case, the coordination phase may struggle to simultaneously reach proper convergence rates and avoid potential collisions between the astrobots. Flow-based assignment [Blanton et al. 2003] was another scheme which solves the target-to-astrobot assignment problem as a network flow one. The method aims to maximize the number of targets to which fibers are assigned. This strategy uses network flow graphs to solve the problem. Namely, it first identifies a set of fibers which do not collide any other peers. Then, the set of unallocated fibers are taken into account to resolve collisions using a network flow graph. This method was basically designed to address target-to-astrobot assignment in non-robotic focal planes. In other words, however collisions among fibers are addressed by the method, it may not be able to properly handle collisions among astrobots. In automatized focal planes, collisions depend on real-time states of astrobots during coordinations. So in practice, it is unlikely to be able to consider all potential collisions corresponding to numerous coordination solutions of a system of astrobots in the assignment phase. Thus, modular perspective in decoupling assignment phase from coordination phase may be more effective to avoid collisions and to find fast and safe coordinations during coordination phases. Later, the drainage algorithm [Morales et al. 2012] was proposed based on a tiling approach. This method ensures that the maximum number of targets are observed in an observation. In particular, targets are moved among various lists of unassigned astrobots each of which can observe a specific set of targets. Finally, each target is assigned to the shortest list including observable targets. This method assimilates the target-to-astrobot assignment ratio corresponding to the overall desired targets with respect to astrobots. As a disadvantage, this strategy assumes no physical size for astrobots, thereby neglecting any potential collisions between them.

Alternatively, target-based assignment method [Schaefer et al. 2016] assigns astrobots to targets instead of targets to astrobots. The assignment ratio of this algorithm is improved compared to that of the drainage algorithm. However, this method is computationally more intensive than the drainage algorithm. The target-based assignment method seeks the ease of path finding, say, decreasing the occurrences of collisions[2] and deadlocks[3] using a parity-based mechanism. In particular, each astrobot can rotate in two different directions one of which may be less susceptible to collisions and deadlocks. Thus, one may set and vary parities associated with an astrobot set to find a parity set according to which the astrobots can converge to a specific configuration with less colliding and deadlock issues. This method checks the existence of conflicts at the assignment time. Thus, achieving the least collision-prone solution is not guaranteed. Nonetheless, there is no guideline based on which one could set parities in a systematic way to achieve the reported ∼2%-3% improvement rate in assignment gain. Thus, none of the methods above simultaneously

---

[2] A collision occurs when two neighboring astrobots violently strike against another.

[3] A deadlock is a situation in which two neighboring astrobots, which blocks each other's paths through their target spots, stop moving at a location which is not their target spots. Each back-and-forth movement of such a pair of astrobots is counted as one deadlock.



(a) The blue astrobot fulfills the minimum coordination criterion.

(b) The red astrobot satisfies the maximum distribution criterion.

Figure 2.1: Optimization requirements evaluations in a typical assignment scenario. (As a patch of a typical focal plane, the gray hexagons represent astrobots which have been already allocated, and the black circle depicts their average coordinate. Two unallocated astrobots, i.e., the dark blue and red manipulators, are confined to their corresponding hexagons. Each of the astrobots can reach the green circle representing an unassigned target.)

minimizes both the required coordination and the occurrence of collisions and/or deadlocks.

In this chapter, we find an optimal solution to the target-to-astrobot assignment problem which minimizes the required coordination and maximizes the distance between assigned astrobots to minimize collisions and/or deadlocks. In particular, we embed the quoted criteria into a cost function. Then, we propose an algorithm which finds the optimal target-to-astrobot assignment assignments constrained to the cost function minimization. We show that our quadratic algorithm is computationally efficient enough to solve the target-to-astrobot assignment problem associated with crowded astrobot sets.

Our method improves the performances of both hybrid [Tao et al. 2018] and supervisory [Macktoobian et al. 2019c] coordination approaches. In particular, hybrid coordination reaches higher convergence rates using our optimal assignment strategy. The supervisory coordination, which seeks complete coordination of an astrobot set, often suffers from computational complexity issues. To be specific, supervisory coordination requires intensive computations to solve the completeness problem if the random, drainage, or target-based algorithms are used in the assignment process. In contrast, we observe that optimal target-to-astrobot assignment efficiently reduces the state space size of coordination supervisors. Put differently, optimal target-to-astrobot assignment surpasses the methods quoted above to solve the coordination problem in a shorter time.

The remainder of the paper is organized as follows. Optimal target-to-astrobot assignment process is presented in Section 2.2. In particular, a cost function is defined to address the requirements of the problem solution. Then, an algorithm is established to solve the problem. We also demonstrate the quadratic complexity of our algorithm. Section 2.3 illustrates how our optimal assignment algorithm improves the coordination results of both hybrid and supervisory methods. Section 2.4 summaries our accomplishments.

## 2.2 Algorithm

In this section, we start by defining a cost function which takes the trade-off between the following requirements into account.

1. the minimum coordination of the astrobots which reach a particular set of targets: each target should be assigned to the astrobot whose initial ferrule's coordinate is in the closest distance to the target's projected location on the host focal plane compared to the projected locations of the reminder of the reaching astrobots. (see, Figure 2.1a)
2. the maximum distribution of the astrobots which reach a particular set of targets: each target should be assigned to the astrobot which is located at the farthest distance from the already-allocated astrobots. (see, Figure 2.1b)

We first define the "reachability relation" which specifies an astrobot reaching a particular target as below.

**Definition 1** [Reachability Relation]**.** Let $\pi$ and $t$ be an astrobot and a target, respectively. Then, the *reachability relation* $\mathcal{R}(\pi, t)$ holds if $t$ is reachable by $\pi$.

Using the definition above, We also specify the set of all astrobots reaching a specific target as follows.

**Definition 2** [Reachable Set]**.** Let $\mathcal{P}$ and $t$ be a set of astrobots and a target, respectively. Then, $\mathfrak{R}_t^{\mathcal{P}}$ is the largest subset of $\mathcal{P}$ whose elements reach $t$ as follows.

$$\begin{aligned} \mathfrak{R}_t^{\mathcal{P}} :=& \{\pi \in \mathcal{P} | \mathcal{R}(\pi, t)\} \\ :=& \{{}^{i}\mathfrak{R}_t^{\mathcal{P}} | 1 \leq i \leq |\mathcal{P}|\} \end{aligned} \tag{2.1}$$

We are interested in a measure to represent the distance of a target from a set of astrobots. Thus, we define the "average coordinate" associated with a batch of astrobots as below.

**Definition 3** [Average Coordinate]**.** Given an index set $\mathcal{I}$ and a set of astrobots $\mathcal{P} := \{\pi_i | i \in \mathcal{I}\}$, let $\mathcal{Q} := \{\boldsymbol{q}_i | i \in \mathcal{I}\}$ be the coordinate set corresponding to the end-effectors of $\mathcal{P}$. Then, the *average coordinate* $\overline{\mathcal{Q}}$ associated with $P$ is defined as below.

$$\overline{\mathcal{Q}} := \frac{\sum\limits_{i \in \mathcal{I}} \boldsymbol{q}_i}{|\mathcal{I}|} \tag{2.2}$$

**Definition 4** [Projected Coordinate]**.** Given a target $t$, $\boldsymbol{q}_t$ represents the *projected coordinate* of $t$ on the focal plane corresponding to the set of astrobots which can observe $t$.

Now we are set to mathematically formulate our requirements. In particular, we take the minimum coordination criterion into account corresponding to a target $t$ by assigning the astrobot $\pi_i \in {}^{i}\mathfrak{R}_t^{\mathcal{P}}$ to it which minimizes the following subfunction

$$\|\boldsymbol{q}_i - \boldsymbol{q}_t\|_2 \,. \tag{2.3}$$

Furthermore, we consider the maximum distribution criterion for a target $t$ by assigning the astrobot $\pi_i \in {}^{i}\Gamma_t^{\mathcal{P}}$ to it which maximizes the following subfunction

$$\sum_{j \in \mathcal{J}} \|\boldsymbol{q}_i - \boldsymbol{q}_j\|_2 \,. \tag{2.4}$$

Here $\mathcal{J}$ is an index set corresponding to the set of already allocated astrobots.

One may note that the maximum distribution criterion is computationally labor-intensive. So, we present a lower bound for this criterion which is computationally more plausible.

**Proposition 1.** *Let $\mathcal{P}$ and $t$ be a set of astrobots and a target to be observed, respectively. Given the average coordinate $\overline{\boldsymbol{\mathcal{Q}}}$ associated with $\mathcal{P}$, a lower bound for the maximum distribution criterion, i.e., (2.4), is*

$$\left\lVert \boldsymbol{q}_i - \overline{\boldsymbol{\mathcal{Q}}} \right\rVert_2 . \tag{2.5}$$

*Proof.* According to the extended triangle inequality, we have

$$\sum_{j \in \mathcal{J}} \left\lVert \boldsymbol{q}_i - \boldsymbol{q}_j \right\rVert_2 \geq \left\lVert \sum_{j \in \mathcal{J}} \left( \boldsymbol{q}_i - \boldsymbol{q}_j \right) \right\rVert_2 , \tag{2.6}$$

which gives

$$\begin{aligned} \left\lVert \sum_{j \in \mathcal{J}} \left( \boldsymbol{q}_i - \boldsymbol{q}_j \right) \right\rVert_2 &= \left\lVert |\mathcal{J}| \boldsymbol{q}_i - \sum_{j \in \mathcal{J}} \boldsymbol{q}_j \right\rVert_2 \\ &= |\mathcal{J}| \left\lVert \boldsymbol{q}_i - \frac{\sum\limits_{j \in \mathcal{J}} \boldsymbol{q}_j}{|\mathcal{J}|} \right\rVert_2 \\ &\geq \left\lVert \boldsymbol{q}_i - \overline{\boldsymbol{\mathcal{Q}}} \right\rVert_2 . \end{aligned} \tag{2.7}$$

Thereby, proving the claim. ■

Therefore, a lower bound for the maximum distribution criterion, i.e., (2.4), is

$$\left\lVert \boldsymbol{q}_i - \overline{\boldsymbol{\mathcal{Q}}} \right\rVert_2 . \tag{2.8}$$

Thus, the overall cost function, the minimization of which simultaneously satisfies the both optimal criteria, is as follows.

$$\frac{\left\lVert \boldsymbol{q}_i - \boldsymbol{q}_t \right\rVert_2}{\left\lVert \boldsymbol{q}_i - \overline{\boldsymbol{\mathcal{Q}}} \right\rVert_2} \quad \left( \forall i : 1 \leq i \leq \lVert \mathfrak{R}_t^{\mathcal{P}} \rVert \right) \tag{2.9}$$

The optimal assignment solver (OAS) algorithm takes (2.9) into account to solve the target-to-astrobot assignment optimization problem[4].

```
Algorithm  Optimal Assignment Solver (OAS)
Input:
    $\mathcal{P}$                                  ▷ Astrobot set
    $\mathcal{T}$                                  ▷ Targets set
Output:
    Assigned target-astrobot tuples

$\boldsymbol{\mathcal{Q}} \leftarrow \varnothing$
sort $\mathcal{T}$ based on the observation priorities in descending order
for each $t \in \mathcal{T}$ do
    assign $t$ to $\pi_i$ such that $i = \underset{i}{\operatorname{argmin}} \frac{\lVert \boldsymbol{q}_i - \boldsymbol{q}_t \rVert_2}{\lVert \boldsymbol{q}_i - \overline{\boldsymbol{\mathcal{Q}}} \rVert_2}$
    $(\forall i : 1 \leq i \leq \lVert \mathfrak{R}_t^{\mathcal{P}} \rVert)$
    $\mathcal{P} \leftarrow \mathcal{P} \setminus \{\pi_i\}$
    $\boldsymbol{\mathcal{Q}} \leftarrow \boldsymbol{\mathcal{Q}} \dot{\cup} \{\pi_i\}$          ▷ Symbol $\dot{\cup}$ denotes disjoint union operation.
end for
return The assigned target-astrobot tuples
```

---

[4] We sort targets based on their priorities so that those with higher priorities can be assigned to the astrobots which have higher chances, in view of encountering less deadlocks, to reach those targets.

Now, we demonstrate the quadratic computational complexity of the OAS algorithm.

**Proposition 2.** *The computational complexity of the OAS algorithm is quadratic.*

> *Proof.* Let $\mathcal{T}$ and $\mathcal{P}$ be a set of targets and astrobots, respectively. Given $n := |\mathcal{T}|$ and $k := |\mathcal{P}|$, the sorting process can be executed in $\mathcal{O}(n \log n)$, which is not the computational bottleneck of the algorithm. Then, the loop has to check a subset of astrobots for each target. In the worst case, the $i^{\text{th}}$ iteration of the loop has to check $k - i + 1$ astrobots with respect the $i^{\text{th}}$ target. Thus, the overall number of loop operations is
>
> $$nk - (1 + 2 + \cdots + n) + 2 = nk - \frac{n(n+1)}{2} + 2. \tag{2.10}$$
>
> In the worst case, we ideally suppose $k = n$ to maximize the information obtained from an observation. It turns out that the complexity is $\mathcal{O}(n^2)$. Thus, overall computational complexity of the OAS algorithm is quadratic. ■

One may argue that the optimal target-to-astrobot solution above may be written in the formalism of classical optimization problems. However, the constraints we have are nonlinear, so none of the efficient tools of linear programming may be used especially when we have thousands of astrobots and targets. Thus, we believe that in small-scale assignment tasks, classical optimization form of our solution may be taken into account. But in the case of massive surveys, the optimality-criterion-based formulation is the superior method.

## 2.3 Simulations

In this section, we observe how our optimal target-to-astrobot assignment method improves the performance of the overall coordination of astrobot set in view of different measures. Our simulation parameters are set based on the information of Table 2.1. The navigation tests are also based on the kinematic equations of astrobots explained in Chapter 1. As well, we consider no positional noise in the target-astrobot coordinates since this assumption does not bring any insights into our desired comparisons. In particular, we compare what our strategy achieves to those of the target-based assignment, known as one of the most promising available assignment methods up to now. The target distributions are assumed to be random. In particular, we use a circular uniform distribution specified as follows.

$$\begin{aligned} r^2 &\sim U[0; \sqrt{r_{\max}}] \\ \theta &\sim \pi U[0; 2] \end{aligned} \tag{2.11}$$

Here, $r_{\max}$ depends on the number of the astrobots in each test. So, the Cartesian coordinates of targets are generated according to the following rules.

$$\begin{aligned} x &:= r \sin\theta \\ y &:= r \cos\theta \end{aligned} \tag{2.12}$$

The number of the generated targets is the same as that of the astrobots, while all targets have to pass the reachability criterion, say, each target has to be at least reachable by one astrobot.

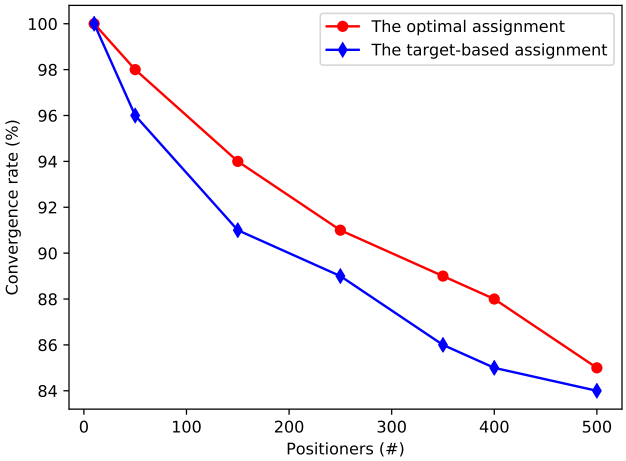


Figure 2.2: The convergence rate comparison between optimal and target-based assignments during hybrid coordinations

We performed 50 simulated scenarios per each population[5] of astrobots whose overall averages are presented below. Major parameters of the simulation are specified in the Table 2.1.

The hybrid coordination [Tao et al. 2018] generally does not guarantee the complete coordination of a typical astrobot set. Thus, the convergence rate is a measure to assess the effectiveness of a coordination process. In particular, Figure 2.2 illustrates the convergence rates corresponding to various numbers of astrobots. One observes that the hybrid coordination method yields higher convergence rates using our optimal target-to-astrobot assignment method.

The number of deadlocks during a coordination process is correlated to the required time to reach the coordinated configuration. Put differently, the less deadlocks occur in the course of coordination, the faster the final coordinated configuration is achieved. In particular, Figure 2.3 depicts the decrement of deadlocks during coordination when optimal target-to-astrobot assignment is taken into account.

The supervisory coordination [Macktoobian et al. 2019c] is also improved using the optimal target-

---

[5] As Figure 2.2 illustrates, we consider various packs of astrobots whose populations are 63, 157, 241, 350, 400, and 500. In the case of deadlock analysis and state space analysis, see, Figure 2.3 and 2.4, respectively, we have 500 populations each of which includes a unique number of astrobots in the range [0, 500].

Table 2.1: The parameters applied to the simulations

| **Parameter** | **Value** |
|---|---|
| The length of the first arm | 8 mm |
| The length of the second arm | 17 mm |
| The width of the first arm | 8 mm |
| The width of the second arm | 4 mm |
| The ferrule size | 1 mm |
| The maximum speed of the first arm's actuator | 30 rpm |
| The maximum speed of the second arm's actuator | 20 rpm |
| The time step of control command generation | 10 ms |

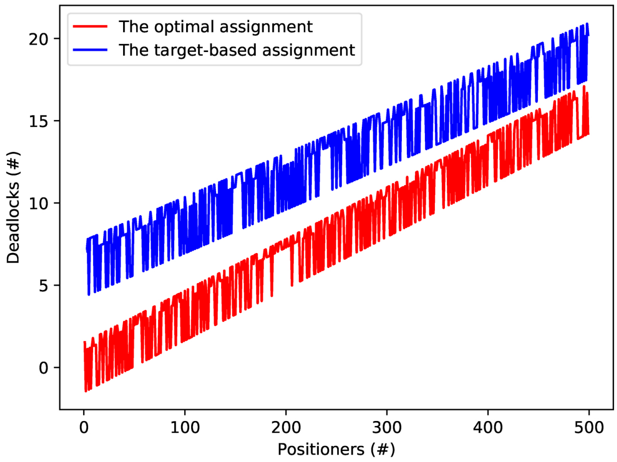


Figure 2.3: The deadlock occurrences using optimal assignment during hybrid coordination

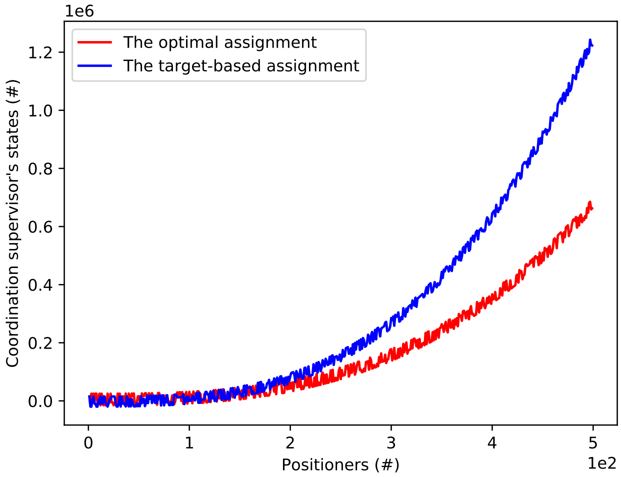


Figure 2.4: The decrement of coordination supervisors' state space size during supervisory coordination

to-astrobot assignment. Namely, a coordination supervisor is a finite state machine some of whose strings are solutions to the complete coordination problem associated with a specific set of astrobots. Since the solutions have to be found through the overall structure of a coordination supervisor, the state size of the coordination supervisor significantly impacts on the time required to solve a complete coordination problem. In particular according to the Figure 2.4, the optimal target-to-astrobot assignment method resembles the target-based one in view of the state space size of coordination supervisors if the overall astrobot sets are not very populated. However, we observe that the optimal target-to-astrobot assignment method surpasses the target-based counterpart regarding this factor when the number of astrobots is increased.

## 2.4 Conclusion

This chapter introduces a new method to improve the state-of-the-art target-to-astrobot assignment algorithm based on an optimization perspective. In particular, the optimal target-to-astrobot assignment takes a cost function into account according to which the distribution density of assigned astrobots is locally decreased, thereby less frequent occurrences of deadlocks during coordination. We also minimize the cost function seeking the target-fiber pairs whose Euclidean distance is as small as possible. Thus, the overall coordination time is minimized. We also show that our optimization algorithm is efficiently scalable in quadratic time. The applied simulations represent the efficiency of the optimal target-to-astrobot assignment scheme to improve the performances of both the hybrid and the supervisory coordination techniques.

# Chapter 3

# Heterogeneous Target-to-Astrobot Assignment

Publication Note

* * *

The material presented in this chapter is adapted from:

- **M. Macktoobian**, D. Gillet, and J-P. Kneib, Heterogeneous Target Assignment to Robotic Fiber Positioner Systems, in *Australian and New Zealand Control Conf. (ANZCC)*, IFAC/IEEE, pp. 48-53, 2019.

## 3.1 Introduction

The cited target-to-astrobot assignment approaches, quoted in the previous chapter, are only applicable to the set of homogeneous targets. In particular, when all of the targets (for example, astronomical objects) and their desired operations (for example, astronomical observations) are the same for all astrobots, then one can use the quoted algorithms to resolve assignment processes. However, if various subsets of astrobots are supposed to be used to handle different types of operations, then the algorithms above shall not be applicable. For example, active space debris mitigation [Pirat et al. 2017] seeks the strategies to detect space debris at long ranges. Since space debris radiate near-infrared waves, infrared fibers [Kimura et al. 2010] can be used to detect them at the range of low earth orbit (LEO), which is an interesting region in view of debris mitigation. As another motivation, the number of available fibers may exceed the number of the desired targets to be observed. Thus, one would like to assign some of the free fibers to another application such as space debris detection, as explained above. In this case, the assignment procedure is heterogeneous, in that there are two classes of targets which are substantially different and need to be treated differently in view of the fiber assignment procedure. Heterogeneous assignment maximizes the operational capacity of the fibers of a telescope such that the fibers which are not assigned to any observational target may be assigned to other targets corresponding to a secondary application.

This chapter proposes a multilinear-time algorithm to assign astrobots of a telescope to more than one operational application. In view of our heterogeneous algorithm, the targets are classified into two categories: the observational targets which are often very far from the earth and move very slow, so one can consider them static; the second group include the targets which are relatively closer to the earth and move faster, so one takes them as dynamic targets into account. Since astronomical observations are the main applications of astrobots, we divide observation targets to two groups of high-priority and low-priority targets. The high-priority targets are assigned to fibers ahead of the dynamic targets. Finally, we assign the low-priority targets to any remaining fiber which has not yet been assigned to either a high-priority static or dynamic target. As we explain in detail further, our heterogeneous algorithm assigns the minimum number of astrobots, called bundle, to a dynamic target which is necessary to detect it. Thus, our strategy is optimized in view of bundle formation. In other words, minimizing the bundle size provides more unallocated fibers to be assigned to static targets. Since our algorithm takes two partitions of low- and high-priority static targets into account, it is flexible enough to manage the assignment completeness. To be specific, if the astrobots assignment to dynamic targets is not satisfactorily complete, then one may decrease the number of high-priority static targets to provide more available fibers to dynamic targets. Overall, there is generally a trade-off between the complete assignment of fibers to static targets and those of to dynamic ones.

The remainder of the chapter is organized as follows. Heterogeneous target-to-astrobot assignment process is investigated in Section 3.2. Namely, Section 3.2.1 describes a bundle formation method to select a minimal subset of available astrobots to each dynamic target. We use the bundle formation method to propose our heterogeneous target-to-astrobot assignment algorithm in Section 3.2.2. We also show that the algorithm is multilinear in view of computational complexity. Section 3.3 solves a heterogeneous target-to-astrobot assignment scenario using our algorithm to illustrate its efficiency. We share our conclusion in Section 3.4.

## 3.2 Algorithm

We start from a set of definitions based on which the problem statement of the heterogeneous target-to-astrobot assignment is defined. We will also later use these definitions to develop our heterogeneous target-to-astrobot assignment algorithm. We already noted that static targets refer to those whose movements are negligible compared to the observation exposure times such as galaxies. Thus, the following definition addresses the notion of "static target" based on the concept of exposure time.

**Definition 5** [Static Target]. A target $t_S$ is a *static target* if its movements are negligible during the exposure time required to observe it. Then, the set of static targets is represented by

$$\mathcal{T}_S := \{t_S^i | i \in \mathcal{I}\}, \tag{3.1}$$

where $\mathcal{I}$ is an index set.

We partition the set of static targets to two disjoint sets based on their priorities. In other words, the high-priority subset $\mathcal{T}_S^H \subseteq \mathcal{T}_S$ includes those static targets whose observations are necessary during a particular observation period. Put differently, an astrobot has to be assigned to each of those targets. In contrast, the subset $\mathcal{T}_S^L \subseteq \mathcal{T}_S$ denotes the targets whose observations may be postponed to another observation period if there is currently no astrobot to be assigned to them. Thus, we have $\mathcal{T}_S := \mathcal{T}_S^H \dot{\cup} \mathcal{T}_S^L$.

We also exploit the concept of exposure time to define the notion of "dynamic target" as follows.

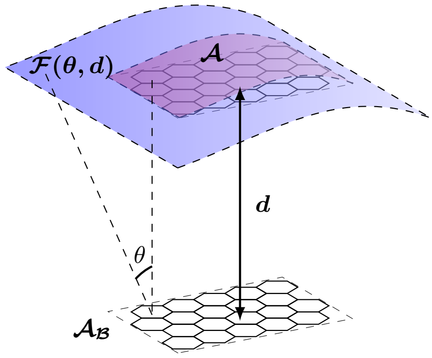


Figure 3.1: A typical bundle whose corresponding surface covers a specific area around a particular dynamic object

**Definition 6** [Dynamic Target]. A target $t_D$ is a *dynamic target* if its movements are not negligible during the exposure time required to observe it. Then, the set of static targets is represented by

$$\mathcal{T}_D := \{t_D^j | j \in \mathcal{J}\}, \tag{3.2}$$

where $\mathcal{J}$ is an index set.

Then, the overall set of targets is $\mathcal{T} = \mathcal{T}_S \dot{\cup} \mathcal{T}_D$.

**Definition 7** [Free Astrobot]. Let $\mathcal{P}$ be a set of astrobots. Then, an astrobot $\pi \in \mathcal{P}$ is *free* if it is not assigned to any specific target.

In general, a single astrobot cannot efficiently detect a dynamic target since the target often moves too fast, and the limited workspace of a single astrobot is not enough for this purpose. Thus to detect dynamic targets, we construct bundles of astrobots which can collectively cover a desired area in which a dynamic target moves.

**Definition 8** [Bundle]. Let $\mathcal{P}$ be a set of astrobots. Then, $\mathcal{B} \subseteq \mathcal{P}$ is a *bundle* if its astrobots shape a single astrobots island.

**Remark 1.** The definition of free astrobot can be extended to define a free bundle which is a bundle whose all astrobots are free.

Thus, we define the heterogeneous target-to-astrobot assignment problem as follows.

**Problem 1.** Denote by $\mathcal{P}$ a set of astrobots, and let $\mathcal{T}_S$ be a set of static targets subject to some priority specifications. Let also $\mathcal{T}_D$ be a set of dynamic targets with some region specifications. Assign a

(i) free $\pi \in \mathcal{P}$ to each $t \in \mathcal{T}_S$;

(ii) free bundle $\mathcal{B} \subseteq \mathcal{P}$ to each $t \in \mathcal{T}_D$.

In the following two sections, we solve the problem above.

### 3.2.1 Bundle Formation

Suppose a bundle $\mathcal{B}$ whose base area is $\mathcal{A}_{\mathcal{B}}$ as depicted in Figure 3.1. Let also $d$ denote the distance of the bundle to a particular dynamic target. We are interested in the area around the target at the distance of $d$ which is observable by $\mathcal{B}$. In other words, the field of view of $\mathcal{B}$ at distance $d$ is of interest. We already cited the tilted axes and the deviated ferrules of astrobots. Thus, the field of view of a bundle at distance $d$ encompasses a surface whose area $\mathcal{A}$ is greater than $\mathcal{A}_{\mathcal{B}}$. Denoting the overall deviation by $\theta$, the difference between $\mathcal{A}$ and $\mathcal{A}_{\mathcal{B}}$ depends on both $\theta$ and $d$ as follows

$$\mathcal{A} = \mathcal{A}_{\mathcal{B}} + \mathcal{F}(\theta, d), \tag{3.3}$$

where the definition of the binary function $\mathcal{F}(\cdot,\cdot)$ depends on the geometrical shape of the bundle. Each pair of the arguments associated with $\mathcal{F}(\cdot,\cdot)$ in fact corresponds to a particular dynamic target.

To form a bundle corresponding to a specific dynamic target, the area of the surface, i.e., $\mathcal{A}$, is known because the area in which we are interested in detecting the target is a part of the solutions requirements. Put differently, $\mathcal{A}$ is taken as a region specification into account associated with a specific dynamic target. Thus, one selects some unassigned astrobots to shape a bundle with the minimum number of elements which fulfills (3.3). As an example, we determine the function $\mathcal{F}(\cdot,\cdot)$ for a particular circular bundle geometry. We would like to find the radius of the bundle such that the surface area visible from the bundle's field of view will be $\mathcal{A}$. Note that $\mathcal{A} = \pi(r + x)^2$ where $x = d \tan\theta$. Thus, we have

$$\begin{aligned} \mathcal{A} &= \pi(r + d\tan\theta)^2, \\ &= \pi r^2 + \pi d \tan\theta(2r + d\tan\theta), \end{aligned} \tag{3.4}$$

which turns out that $\mathcal{F}(\theta, d) = \pi d \tan\theta(2r + d\tan\theta)$.

We establish the bundle former ($\mathcal{BF}$) algorithm as depicted by Algorithm 2. To analyze the computational complexity of the $\mathcal{BF}$ algorithm, we first introduce the notion of "reachability relation" as below.

**Definition 9** [Reachability Relation]. Let $t \in \mathcal{T}$ be a target. Given a set of astrobots $\mathcal{P}$, let $\pi \in \mathcal{P}$ be a positioner. Then, the binary *reachability relation* $\mathcal{R}(t, \pi)$ holds if $t$ is reachable by $\pi$.

We use the definition above to define the notion of "reachable target set" as follows.

**Definition 10** [Reachable Target Set]. Let $\mathcal{T}$ be an arbitrary set of targets. Let also $\mathcal{P}$ be a set of astrobots. Then, the *reachable target set* $\mathcal{P}_{\mathcal{T}}$ with respect to $\mathcal{T}$ is defined as follows.

$$\mathcal{P}_{\mathcal{T}} := \{\pi \in \mathcal{P} | (\forall t \in \mathcal{T}) \mathcal{R}(t, \pi)\} \tag{3.5}$$

Lemma 3 also proves that the computational complexity of the $\mathcal{BF}$ algorithm is multilinear.

**Lemma 3.** *Let $\mathcal{T}_D$ and $\mathcal{P}_{\mathcal{T}_D}$ be a set of dynamic targets and the set of astrobots which reach them, respectively. Given $|\mathcal{T}_D| := n$ and $|\mathcal{P}_{\mathcal{T}_D}| := m$, the computational complexity of the $\mathcal{BF}$ algorithm is $\mathcal{O}(mn)$.*

> *Proof.* Each astrobot can only be assigned to one dynamic target. Then in the worst case, each astrobot has to be checked to be assigned to every dynamic target. Then, the number of applied tests is $nm$. Therefore, the computational complexity is $\mathcal{O}(mn)$. ■

### 3.2.2 Algorithm

In this section, we use the $\mathcal{BF}$ algorithm to propose our heterogeneous assignment algorithm. For this purpose, we first complete the set of required definitions by introducing "complete reachability" and "critical reachability" relations.

**Definition 11** [Complete Reachability]. Let $\mathcal{P}$ and $\mathcal{T}$ be a set of astrobots and a set of targets, respectively. Then, the *complete reachability relation* $\mathcal{C}(\mathcal{P}, \mathcal{T})$ holds if the following formula is true.

$$(\forall t \in \mathcal{T})(\exists \pi \in \mathcal{P})\mathcal{R}(t, \pi) \tag{3.6}$$

**Definition 12** [Critical Reachability]. Let $\pi \in \mathcal{P}$ be an astrobot. Let also $t' \in \mathcal{T}_D$ and $t'' \in \mathcal{T}_S^H$ be a dynamic and a high-priority static target, respectively. Then, $\pi$ is *critically reachable*, say, the ternary relation $\mathcal{K}(\pi, t', t'')$ holds, if the following formula is true.

$$[\pi \in (\mathcal{P}_{t'} \dot{\cup} \mathcal{P}_{t''})] \cap [|\mathcal{P}_{t'}| = 1] \cap [|\mathcal{P}_{t''}| = 1] \tag{3.7}$$

**Definition 13** [Critical Astrobot]. Let $\mathcal{P}$ be a set of astrobots. Let also $\mathcal{T}_D$ and $\mathcal{T}_S^H$ be a set of dynamic and a set of high-priority static target, respectively. Then, *critical astrobot* set of $\mathcal{P}$ with respect to $\mathcal{T}_D$ and $\mathcal{T}_S^H$ is defined as below.

$$\mathcal{P}_{\mathcal{K}} := \{\pi \in \mathcal{P} | (\forall t' \in \mathcal{T}_D)(\forall t'' \in \mathcal{T}_S^H)\mathcal{K}(\pi, t', t'')\} \tag{3.8}$$

**Remark 2.** A critical astrobot is the one which is the exclusive option to reach a pair of dynamic and high-priority static targets. Since we are interested in the assignment of high-priority static targets ahead of dynamic targets, our heterogeneous assignment algorithm is authorized to assign an astrobot to a dynamic target only if that astrobot is not critical.

We present our heterogeneous target-to-astrobot assignment ($\mathcal{HTA}$) algorithm as depicted in Algorithm 3.

The algorithm in fact generates bundles of the largest possible set of astrobots which reach its desired dynamic targets (Line 3). If the remaining astrobots fulfill the complete reachability requirements of all high-priority static targets (Line 4), the algorithm assigns them to those targets properly (Line 5). Otherwise, if the complete reachability is not achieved (Line 6), one implies that some of the assigned astrobots to dynamic targets are critical. So, the algorithm excludes those critical astrobots by assessing the critical reachability relation (Line 7). Then, the bundles are recomputed with a new restricted set of reachable

**Algorithm 2** Bundle Former ($\mathcal{BF}$)

```
Input:
    $\mathcal{T}_D$                                  ▷ Dynamic targets set
    $\mathcal{P}_{\mathcal{T}_D}$                    ▷ Positioners set reaching dynamic targets
Output:
    $\mathcal{B}$                                    ▷ Bundle set
for each $t \in \mathcal{T}_D$ do $\mathcal{B} \leftarrow$ the smallest bundle whose astrobots fill the area $\mathcal{A}_{\mathcal{B}} = \mathcal{A} - \mathcal{F}(\theta, d)$
end for
return $\mathcal{B}$
```

astrobots which no longer includes any critical astrobot. One notes that the next attempt shall be successful to assign at least one astrobot to each of the high-priority static targets. In end end, the remainder of unassigned astrobots may be assigned to some low-priority static targets (Line **10**).

Now, we show that the computational complexity of the $\mathcal{HTA}$ algorithm is multilinear.

**Proposition 4.** *Let $\mathcal{P}$, $\mathcal{T}_D$, and $\mathcal{T}_S$ be a set of astrobots, a set of dynamic targets, and a set of static targets, respectively. Given $m := |\mathcal{P}|$, $n := |\mathcal{T}_D|$, and $k := |\mathcal{T}_S|$, the computational complexity of the $\mathcal{HTA}$ algorithm is $\mathcal{O}(mnk)$.*

*Proof.* To take the worst case into account, we assume the $\mathcal{T}_S^L = \varnothing$ which means $\mathcal{T}_S = T_S^H$ implying that all of the static targets are high-priority in view of the assignment process. An upper bound for the cardinality of auxiliary variable $\mathcal{M}$ is $m = |\mathcal{P}|$. Additionally, according to the result of Lemma 3, bundle formation requires $mn$ units of time. Checking the completeness condition requires the iteration over both $\mathcal{T}_D$ and $\mathcal{T}_S$ sets, thereby $mn$ units of time shall be spent. Thus, the assignment process associated with high-priority static targets shall be done during $mnk$ units of time. The loop of the algorithm is executed one and two times in the best and in the worst cases, respectively. Since we assumed that all static targets are high-priority, the loop is executed twice, and there is no low-priority static target to which an astrobot is assigned. Therefore, the overall worst case computational complexity of the $\mathcal{HTA}$ algorithm is

$$\mathcal{O}(2(mn + mnk)) \equiv \mathcal{O}(mnk). \tag{3.9}$$

■

**Algorithm 3** Heterogeneous Target Assigner ($\mathcal{HTA}$)

**Input:**

| | |
|---|---|
| $\mathcal{P}$ | ▷ Positioners set |
| $\mathcal{T}_D$ | ▷ Dynamic targets set |
| $\mathcal{T}_S$ | ▷ Static targets set |

**Output:**

Assigned astrobot-target pairings

```
 1: $\mathcal{M} \leftarrow \mathcal{P}_{\mathcal{T}_D}$
 2: do
 3:     $\mathcal{P}^\star \leftarrow \mathcal{BF}(\mathcal{M}, \mathcal{T}_D)$
 4:     if $\mathcal{C}(\mathcal{P}_{\mathcal{T}_S^H} \setminus \mathcal{P}^\star, \mathcal{T}_S^H)$ then                ▷ $\mathcal{P}_\mathcal{K} = \varnothing$
 5:         $\mathcal{P}^+ \leftarrow (\forall t \in \mathcal{T}_H^S)$ assign a $\pi \in \mathcal{P}_t \setminus \mathcal{P}^\star$ to $t$ randomly
 6:     else                                                              ▷ $\mathcal{P}_\mathcal{K} \neq \varnothing$
 7:         $\mathcal{M} \leftarrow \mathcal{M} \setminus \mathcal{P}_\mathcal{K}$        ▷ Computing $\mathcal{P}_\mathcal{K}$ using the critical reachability relation
 8:     end if
 9: while $\mathcal{P}_\mathcal{K} \neq \varnothing$
10: $(\forall t \in \mathcal{T}_L^S)$ assign a $\pi \in \mathcal{P}_t \setminus \{\mathcal{P}^\star \cup \mathcal{P}^+\}$ to $t$ randomly.
11: return The assigned target-astrobot tuples
```

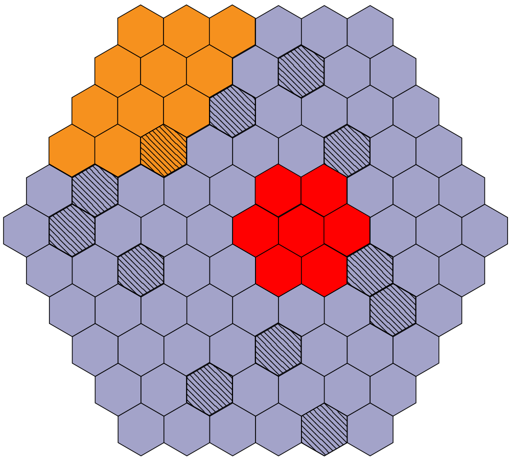


Figure 3.2: The assignment results after the first iteration of the algorithm's loop (The red and orange astrobots correspond to the bundles of the first and the second dynamic targets, respectively. The crossed astrobots are those which must be assigned to high-priority static targets.)

## 3.3 Simulations

In this section, we solve a heterogeneous target-to-astrobot assignment problem using the $\mathcal{HTA}$ algorithm. In particular, we assume a focal plane including 91 astrobots. We also assume two dynamic targets which are two hypothetical cubesat satellites [Pirat et al. 2017] operating for the purpose of active space debris removal at the range of the LEO orbit. We take overall 20 static targets, say, galaxies to be observed, 12 of which are high-priority, and the others are low-priority. A normal distribution, similar to the one used in the simulations of Chapter 2, is used to generate the relative locations of the static targets with respect to the focal plane. We assume the the SDSS-V project [Macktoobian et al. 2019d] the length of each of which is 350 mm. Furthermore, the vertical distance between the ferrule and the first axis of each astrobot is 15 mm. Then considering the tilt rate of $\sim$0.35°, the overall deviation of an astrobot's ferrule is $\sim$0.05 rad. We assume that all of the astrobots are identical, and the distance $d$ associated with the first and the second dynamic targets are 1200 km and 500 km, respectively. The radii of the surfaces around the first and the second dynamic target are required to be 48 km and 12 km, respectively.

The evolution of the solution to the problem is as follows. The first iteration of the algorithm's loop assigns a bundle to each dynamic target. The bundles are depicted in Figure 3.2. In particular, the red bundle and the orange bundle correspond to the first and the second dynamic targets, respectively. However, the crossed astrobots are those which have to be assigned to high-priority static targets. As the figure illustrates, one of the orange astrobots is a critical one. Thus, the second iteration of the loop removes that astrobots from the valid options to shape a bundle, thereby assigning it to the desired high-priority static target. Note that this removal decreases the area of the surface corresponding to the second dynamic target, but it is unavoidable since the critical astrobot has to be assigned to the high-priority static target. The second iteration of the algorithm is depicted in Figure 3.3. In this figure, the double crossed astrobots are those which may be assigned to low-priority static targets. However, one observes that this potential assignment required two astrobots which has been already assigned to dynamic targets. Thus, the algorithm neglects those two low-priority static targets, and the final result is illustrated in Figure 3.4.

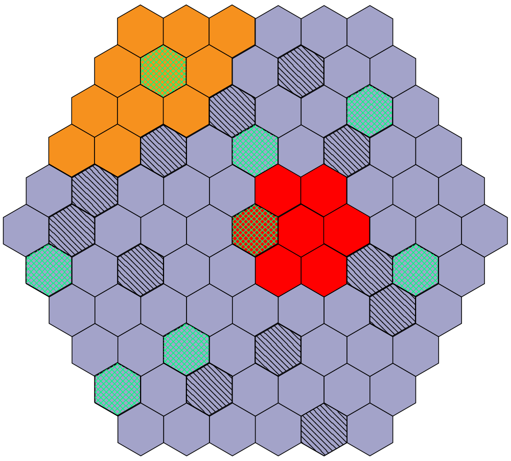


Figure 3.3: The assignment results after the second iteration of the algorithm's loop (The red and orange astrobots correspond to the bundles of the first and the second dynamic targets, respectively. The crossed and the double crossed astrobots are those which must be assigned to high-priority and low-priority static targets, respectively.)

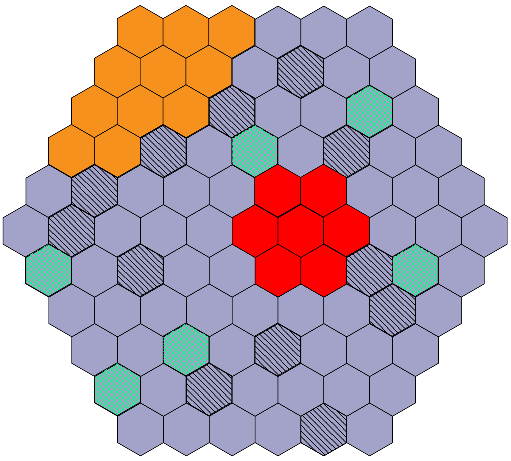


Figure 3.4: The final results of the assignment process (The red and orange astrobots correspond to the bundles of the first and the second dynamic targets, respectively. The crossed and the double crossed astrobots are those which must be assigned to high-priority and low-priority static targets, respectively.)

## 3.4 Conclusion

Astrobots have already been used to generate the map of the observable universe. Space applications, such as debris detection, can also benefit from these astrobots. This chapter presents an algorithm which assigns the astrobots of a telescope to various types of targets, from observational to space operational ones. Thus, not only a wider range of applications can be handled by astrobots but the overall information throughout of a typical astrobots set is maximized since more astrobots are used at the same time.

One may note that astrobots may need various exposure times to collect signals from different types of targets depending on their nature. So, the algorithm may be upgraded such that the distance between astrobots batches corresponding to those heterogeneous targets whose exposure times are close to each other will be maximized. Thus, a typical spectrograph can efficiently distinguish between the traces of the objects corresponding to each of the intended heterogeneous applications.

# Part B

# Analytical Completeness Seeking

# Chapter 4

# Supervisory Complete Coordination

### Publication Note

The material presented in this chapter is adapted from:

- **Matin Macktoobian**, Denis Gillet, and Jean-Paul. Kneib, Supervisory Coordination of Robotic Fiber Positioners in Multi-Object Spectrographs, in the 15th IFAC Symposium on *Large Scale Systems (LSS): Theory and Applications*, pp. 61-66, 2019.

## 4.1 Introduction

Modern astronomy aims to study the evolution of the universe using cosmological spectroscopy [Mazets et al. 1982] surveys. Each survey is a map corresponding to a piece of the observable universe. For this purpose, many projects, e.g., DESI [Aghamousa et al. 2016a; Aghamousa et al. 2016b], MOONS [Michele Cirasuolo et al. 2014], etc., are carried out to develop telescopes equipped with spectrographs. Each spectrograph is connected to a set of astrobots. Each astrobot is assigned to a specific galaxy in the sky corresponding to a particular observation. Then, the spectral information of that galaxy is transferred to its spectrograph through the observing fiber positioner. To increase the throughput of each observation in view of the collected information, one takes astrobot swarms into account. Optical fibers are mounted in a particular area of their hosting telescope called focal plane. The favorite hexagonal formation of astrobots shall be dense enough to mount as many as possible astrobots at the focal plane of the hosting telescope [Hörler et al. 2018; Hörler 2018]. Since the target assignment to an optical fiber is changed from one observation to another, an astrobot system is attached to each fiber to move it. Each astrobot is generally a rotational-rotational robot whose workspace overlaps with those of its neighbors. So, the collision avoidance, the trajectory planning, and the completeness associated with a set of astrobots are challenging. In particular, a set of astrobots is completely coordinated if all of its astrobots point to their targets at the end of the coordination process corresponding to a specific observation. We seek the complete coordination under the assumption that all of the target astrobots are assigned to some galaxies.

The current solutions to the coordination problem of astrobots lack general completeness. In particular, artificial potential fields [Macktoobian et al. 2013; Macktoobian et al. 2016] are used to plan collision-free trajectories for astrobots [Makarem et al. 2014]. The state-of-the-art solution [Tao et al. 2018] combines a low-level artificial-potential-based navigator with a high-level state-machine-based decision maker to handle deadlocks and oscillations. So, the astrobots with higher priorities are prioritized to be coordinated. However, the complete convergence of all astrobots is not guaranteed. Supervisory control theory and discrete-event systems are promising candidates by which we attack the complete coordination problem.

Discrete-event systems and supervisory control theory [Wonham 2017b] efficiently models and controls event-driven complex systems, respectively. In particular, automatic reconfiguration [Macktoobian et al. 2017] is addressed using supervisory control theory. Each configuration of a discrete-event system exhibits a specific set of functionalities. The discrete-event system is reconfigured from one configuration to another by activating a controllable reconfiguration event. Then, a supervisor is synthesized based on the behavioral and the reconfiguration requirements of the discrete-event system. Then to execute the reconfiguration, the supervisor shall find a string of events from its current state to a target state at which the reconfiguration event is eligible to occur. Backtracking forcibility technique [Macktoobian 2018] was developed to find the set of all forcible paths from the current state to the target state.

The automatic reconfiguration of discrete-event systems and the coordination of astrobots are intrinsically similar in a systematic point of view. So in this Chapter, we seek a solution to the complete coordination problem of astrobots using supervisory control theory and reconfiguration of discrete-event systems. Our supervisory control approach models each astrobot as a rotational arm with only one DoF. This assumption is based on a trade-off according to which the state size of the overall system remains reasonably tractable in view of the supervisor computation and backtracking forcibility. On the other hand, this assumption may limit the number of available solutions to the problem; yet our coordination method is efficient enough to achieve the desired completeness. In particular, we model each astrobot and its behavior as discrete-event systems. Then, we synthesize a supervisor, called *coordination supervisor*, to control the overall behavior of the complex set of astrobots. We define the plant model and the specifications such that the marked state of the synthesized supervisor represents the complete coordination of the system. Thus, the coordination problem is indeed reduced to finding forcible paths from the initial state of the supervisor to its target state. One notes that the cited process is similar to the solution checking of the reconfiguration problem briefly explained above. In other words, the complete coordination problem seeks the reachability of a particular state corresponding to a coordination supervisor. For this purpose, we propose an algorithm to realize the quoted completeness-checking process associated with a typical coordination problem.

The remainder of this chapter is organized as follows. A brief review of the supervisory control theory is represented in Section 4.2. Section 4.3 defines the complete coordination problem in the language of a backtracking forcibility problem and proposes an algorithm to solve it. Section 4.4 includes an example to illustrate how our algorithm practically solves a complete coordination problem. Our concluding remarks are finally drawn in Section 4.5.

## 4.2 Background

Supervisory control theory controls discrete-event systems modeled by the Ramadge-Wonham framework [Ramadge et al. 1987]. A discrete-event system is formally represented by a generator, say,

$$\mathbf{G} = (Q, \Sigma, \mathcal{D}, q_0, Q_m), \tag{4.1}$$

where $\Sigma = \Sigma_c \dot{\cup} \Sigma_u$ is a finite alphabet of event labels, partitioned into the *controllable* event labels and the *uncontrollable* ones; $Q$ is the finite *state set*; $\mathcal{D} : Q \times \Sigma^* \to Q$ is the *extended partial transition function*; $q_0$ is the *initial state*; and $Q_m \subseteq Q$ is the subset of *marked states*. The *closed behavior* and the *marked behavior* of $\mathbf{G}$ are the regular languages

$$\begin{aligned} L(\mathbf{G}) &:= \{s \in \Sigma^* | \delta(q_0, s)!\}, \\ L_m(\mathbf{G}) &:= \{s \in L(\mathbf{G}) | \delta(q_0, s) \in Q_m\}. \end{aligned} \tag{4.2}$$

Here $\delta(q_0, s)!$ means that $\delta(q_0, s)$ is defined.

A supervisory control function for $\mathbf{G}$ is a map

$$\mathbf{V} : L(\mathbf{G}) \to \mathcal{G} \tag{4.3}$$

in which

$$\mathcal{G} = \{\kappa \in \mathrm{Pwr}(\Sigma) | \kappa \supseteq \Sigma_u\} \tag{4.4}$$

is the set of *control patterns*. '$\mathbf{G}$ under supervision of $\mathbf{V}$' is written as $\mathbf{V}/\mathbf{G}$. Given a sublanguage $M \subseteq L_m(\mathbf{G})$ we define the *marked behavior* of $\mathbf{V}/\mathbf{G}$ as

$$L_m(\mathbf{V}/\mathbf{G}) := L(\mathbf{V}/\mathbf{G}) \cap M \tag{4.5}$$

$\mathbf{V}$ is a *marking nonblocking supervisory control* (MNSC) for the pair $(M, \mathbf{G})$ if

$$\overline{L_m(\mathbf{V}/\mathbf{G})} = L(\mathbf{V}/\mathbf{G}). \tag{4.6}$$

In practice, $\mathbf{V}$ is implemented by a *supervisor* representing the maximally permissive controlled behavior $L_m(\mathbf{V}/\mathbf{G})$ subject to a generator specification, say $\mathbf{S}$; we denote this computation by

$$\mathbf{V} = \mathbf{supcon}(\mathbf{G}, \mathbf{S}). \tag{4.7}$$

For details see, e.g., [Ramadge et al. 1987]. Given a generator $\mathbf{V}$, and states $q, q' \in Q_{\mathbf{V}}$, backtracking forcibility analysis is taken into account as

$$Z := \mathcal{BFA}(\mathbf{V}, q, q') \tag{4.8}$$

which yields set of all forcible paths which reach $q'$ from $q$ using backtracking forcibility [Macktoobian et al. 2017].

## 4.3 Completeness-Seeking Supervision

Each astrobot can rotate around its axis according to a specific number of discrete movements. In particular, given an astrobot $\pi$ and its motion step size $n^{\pi}$, $\pi$ rotates $360^{\circ}/n^{\pi}$ during each of its motion steps. From now on, $\mathcal{P}$ denotes the set of all astrobots corresponding to a specific telescope. We define the notions of "forward event" and "backward event" as follows.

**Definition 14** [Forward/Backward Events]. Let $\pi \in \mathcal{P}$ be an astrobot with motion step size $n^{\pi}$. Then, if the controllable *forward event* $v^{\pi}$ (resp., the *backward event* $w^{\pi}$) is enabled, $\pi$ rotates $360^{\circ}/n^{\pi}$ in a clockwise (resp., counterclockwise) direction around its axis.

We also define counters corresponding to the number of the required movements to reach a target in different directions as follows.

**Definition 15** [Forward/Backward Counters]. Let $\pi \in \mathcal{P}$ be an astrobot with motion step size $n^{\pi}$. Given a specific position at the motion space of $\pi$, the *forward counter* $n_v^{\pi}$ (resp., the *backward counter* $n_w^{\pi}$) represents the required number of clockwise (resp., counterclockwise) motion steps to reach the target position from the current position of $\pi$.

In a hexagonal neighborhood around a specific astrobot, the maximum number of the astrobots which may be involved in a colliding scenario to block each other is three. So, the relative priority of those typical astrobots shall be defined in the language of events as follows.

**Definition 16** [Priority Events]. Let $\mathcal{N} \subset \mathcal{P}$ be a set of two astrobots around astrobot $\pi \in \mathcal{P}$ with which construct a colliding scenario. Let also $r(\pi)$ denote the relative priority of $\pi$'s target compared to those of the astrobots of $\mathcal{N}$. Then, we define the following controllable events.

- if $(\forall \pi' \in \mathcal{N})\ r(\pi) < r(\pi')$, then the controllable event $l^{\pi}$ is exclusively enabled;
- if $(\forall \pi' \in \mathcal{N})\ r(\pi) > r(\pi')$, then the controllable event $h^{\pi}$ is exclusively enabled;
- otherwise, the controllable event $m^{\pi}$ is exclusively enabled.

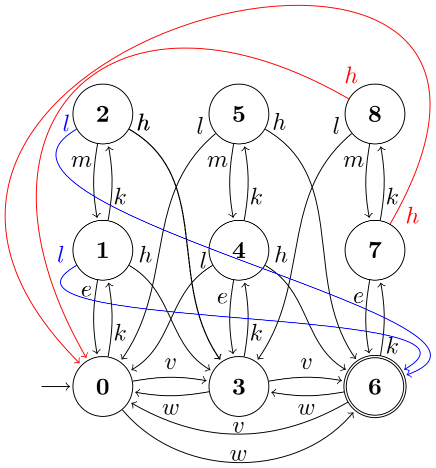


Figure 4.1: The discrete-event model of a typical astrobot $\pi$ with $n^{\pi} = 3$ (The astrobots indices are intentionally omitted for the better readability of the figure.)

We take some uncontrollable events into account to keep track of any potential colliding situation around a typical astrobot as follows.

**Definition 17** [Collision/Free Events]. Given astrobot $\pi \in \mathcal{P}$, if a neighboring astrobot enters the safety zone of $\pi$, uncontrollable *collision event* $k^\pi$ occurs. In contrast, if all of the neighboring astrobots which have resided at the safety zone of $\pi$ leave that area, then uncontrollable *free event* $e^\pi$ occurs.

We define the notion of an astrobot as a discrete-event system as follows.

**Definition 18** [Astrobot]. Let $\pi \in \mathcal{P}$ be an astrobot. Given motion step size $n^\pi$, forward event $v^\pi$, backward event $w^\pi$, forward counter $n^\pi_v$, backward counter $n^\pi_w$, priority events $l^\pi$, $m^\pi$, and $h^\pi$, collision event $k^\pi$, and free event $e^\pi$, the generator $\mathbf{G}^\pi$ associated with $\pi$ is defined as a discrete-event system

$$\mathbf{G}^\pi := (Q^\pi_\mathbf{G}, \Sigma^\pi_\mathbf{G}, \delta^\pi_\mathbf{G}, {q_0}^\pi_\mathbf{G}, {Q_m}^\pi_\mathbf{G}), \tag{4.9}$$

where

- $Q^\pi_\mathbf{G} := \bigcup_{j=0}^{n^\pi} \{q_j, q_j^1, q_j^2\}$,
- $\Sigma^\pi_\mathbf{G} := \{v^\pi, w^\pi, l^\pi, m^\pi, h^\pi, k^\pi, e^\pi\}$,
-

$$\begin{aligned}
\delta^\pi_\mathbf{G} := (\forall j | 0 \leq j \leq n^\pi - 1)\{&\delta(q_j, v^\pi) := q_{j+1}, && (4.10)\\
&\delta(q_{j+1}, w^\pi) := q_j, && (4.11)\\
&\delta(q^j, k^\pi) := q_j^1, && (4.12)\\
&\delta(q_j^1, k^\pi) := q_j^2, && (4.13)\\
&\delta(q_j^2, m^\pi) := q_j^1, && (4.14)\\
&\delta(q_j^1, e^\pi) := q_j, && (4.15)\\
&\delta(q_j^2, h^\pi) = \delta(q_j^1, h^\pi) := q_{j+1}, && (4.16)\\
&\delta(q_{j+1}^2, l^\pi) = \delta(q_{j+1}^1, l^\pi) := q_j\} && (4.17)\\
&\dot\cup\{\delta(q_0, w^\pi) := q_{n^\pi}, && (4.18)\\
&\delta(q_{n^\pi}, v^\pi) := q_0\}, && (4.19)
\end{aligned}$$

- ${q_0}^\pi_\mathbf{G} :=$ a $q \in Q^\pi_\mathbf{G}$ according to the initial position of $\pi$,
- ${Q_m}^\pi_\mathbf{G} :=$ a $q \in Q^\pi_\mathbf{G}$ based on the target position of $\pi$.

**Remark 3.** We note that the occurrence of collision event $k^\pi$ has to functionally preempt the other events eligible at a specific state of $\mathbf{G}^\pi$.

Figure 4.1 depicts the generator corresponding to the discrete-event model of a typical astrobot $\pi$ with $n^\pi = 3$. Each forward (resp., backward) event moves $\pi$ $360^\circ/3$ in a clockwise (resp., counterclockwise) direction. If no neighboring astrobot enters the safety zone of $\pi$, then it always remains in the scope of the lower-level states of its generator, i.e., $[\mathbf{0}]$, $[\mathbf{1}]$, and $[\mathbf{2}]$. Otherwise, it may encounter one or two neighboring astrobots at its safety zone, thereby occurring $k^\pi$ events at $[\mathbf{0}]$ and/or $[\mathbf{1}]$. In the presence of only one colliding astrobot, if it leaves the safety zone of $\pi$, then the free event $e^\pi$ occurs and $\pi$ returns to its normal state at the lower level of $\mathbf{G}^\pi$. If the colliding astrobot remains in the safety zone of $\pi$, then the astrobot with the higher priority moves clockwise to its target, and the other one with the lower priority moves in a counterclockwise direction. So, not only the collision is avoided, but the deadlock is also handled. In the case of three colliding astrobots, there are three relative priorities. In particular, the astrobots with the highest and the lowest priorities moves clockwise and counterclockwise, respectively. Furthermore, the astrobot with the medium priority remains at its position. However, its colliding state is changed since at least one of the other colliding peers goes far from it.

Now, we define the "specification" which determines the target position corresponding to a particular astrobot.

**Definition 19** [Specification]. Let $\pi \in \mathcal{P}$ be an astrobot. Then, considering forward counter $n^\pi_v$ and backward counter $n^\pi_w$ associated with $\pi$, *specification* $\mathbf{S}^\pi$ corresponding to $\pi$ is defined as a discrete-event system

$$\mathbf{S}^\pi := (Q^\pi_{\mathbf{S}}, \Sigma^\pi_{\mathbf{S}}, \delta^\pi_{\mathbf{S}}, {q_0}_,^{\pi} {Q_m}^\pi_{\mathbf{S}}), \tag{4.20}$$

where

- $Q^\pi_{\mathbf{S}} := \left\{ \bigcup_{j=1}^{n^\pi_v + n^\pi_w - 2} q_j \right\} \dot\cup \{q_0, q_m\}$,
- $\Sigma^\pi_{\mathbf{S}} := \{v^\pi, w^\pi\}$,
-

$$\begin{align} \delta^\pi_\pi :=& (\forall j | 0 \le j \le n^\pi_v - 2)\{\delta(q_j, v^\pi) := q_{j+1}, \delta(q_{j+1}, w^\pi) := q_j\} \dot\cup \tag{4.21}\\ & (\forall j | n^\pi_v + 1 \le j \le n^\pi_w - 1)\{\delta(q_j, w^\pi) := q_{j+1}, \delta(q_{j+1}, v^\pi) := q_j\} \dot\cup \tag{4.22}\\ & \{\delta(q_{n^\pi_v - 1}, v^\pi) = \delta(q_{n^\pi_w - 2}, w^i) := q_m\} \tag{4.23}\end{align}$$

- ${q_0}^\pi_{\mathbf{S}} := q_0$,
- ${Q_m}^\pi_{\mathbf{S}} := q_m$.

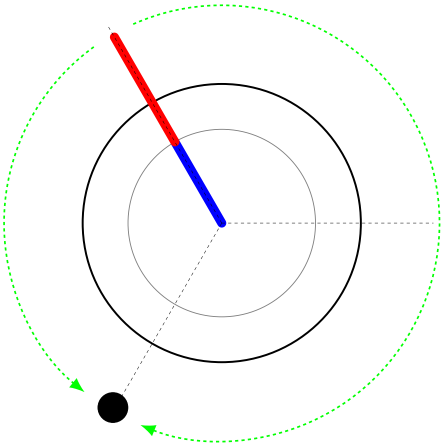


Figure 4.2: The initial and the target points corresponding to a typical astrobot $\pi$ with $n^\pi = 3$ (the blue and the red lines represent the arms of the depicted astrobot.)

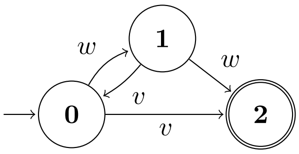


Figure 4.3: The specification corresponding to the Figure 4.2 (The astrobots labels are intentionally omitted to improve the figure clarity.)

The specification corresponding to a particular astrobot in fact determines the paths via which the astrobot reaches its target position from its current position in both clockwise and counterclockwise directions. For example, suppose an astrobot $\pi$ with $n^{\pi} = 3$. The initial and the target positions of $\pi$ are illustrated in Figure 4.2, and Figure 4.3 depicts its corresponding specification.

Now, we utilize supervisory control theory to generate a supervisor in which the solutions to the coordination problem are embedded as depicted in Figure 4.4. In particular, we first compute the overall model **G** of the astrobots system by synchronizing the generators corresponding to all of the astrobots.

$$\mathbf{G} := \mathop{\|}_{\pi \in \mathcal{P}} \mathbf{G}^{\pi} \tag{4.24}$$

We also obtain the overall model **S** of the specifications.

$$\mathbf{S} := \mathop{\|}_{\pi \in \mathcal{P}} \mathbf{S}^{\pi} \tag{4.25}$$

Then, we synthesize the coordination supervisor **V**.

$$\mathbf{V} = \mathbf{supcon}(\mathbf{G}, (\mathbf{allevents}(\mathbf{G}) \parallel \mathbf{S})) \tag{4.26}$$

The following theorem determines the role of a coordination supervisor in the definition and the solution to the complete coordination problem associated with it.

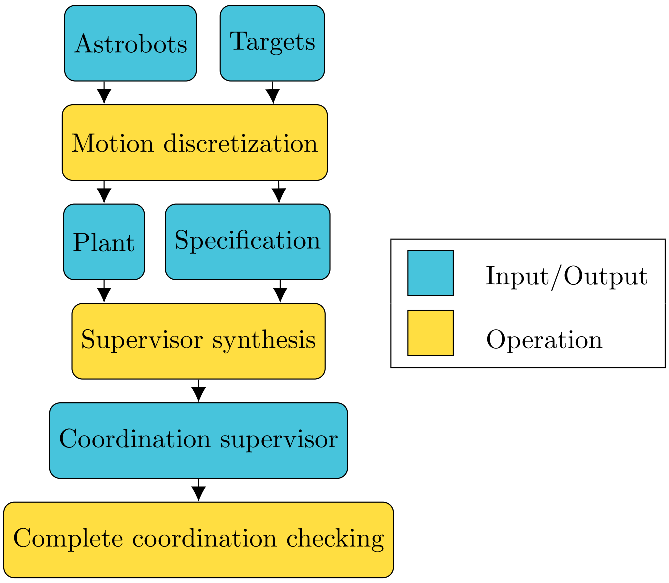


Figure 4.4: Completeness seeking via supervisory coordination

**Theorem 5.** *Let $\mathcal{P}$ be a set of astrobots modeled by $\{\mathbf{G}^{\pi}|\pi \in \mathcal{P}\}$ whose specifications are denoted by $\{\mathbf{S}^{\pi}|\pi \in \mathcal{P}\}$. Let* $\mathbf{V}$ *be a coordination supervisor synthesized to control* $\mathbf{G} := \underset{\pi\in\mathcal{P}}{\parallel} \mathbf{G}^{\pi}$ *with respect to* $\mathbf{S} := \underset{\pi\in\mathcal{P}}{\parallel} \mathbf{S}^{\pi}$. *Then,*

*(i)* $\mathbf{V}$ *has only one marked state;*

*(ii)* $\mathbf{G}$ *is completely coordinated at the marked state of* $\mathbf{V}$.

> *Proof.* $(i)$ By the definition, each element of $\{\mathbf{G}^{\pi}|\pi \in \mathcal{P}\}$ and $\{\mathbf{S}^{\pi}|\pi \in \mathcal{P}\}$ has only one marked state. Then, according to the definition of the synchronous product operator [Wonham 2017b], each of **G** and **S** has only one final state. Additionally, since **allevents(G)** includes only a single marked state, **allevents**(**G**) $\parallel$ **S** has also only one marked state. Since every generator argument of **supcon** has only one marked state, we conclude that **V** possesses only one marked state, as well.
> $(ii)$ Each astrobot is exclusively reached at its target position when its corresponding generator and the generator of its specification reside at their marked states. Then, the marked states of **G** and **S** represent the states whose simultaneous occupancies indicate the reachability of all of their corresponding astrobots. If **V** reaches its marked state, then the cited simultaneous occupancies are realized. Therefore, **G** is complete when **V** resides at its marked state.
>
> ■

A supervisor always starts its supervision process from its initial state. So, the strings of events which can forcibly reach the marked state from the initial state are in fact the solutions to the complete coordination problem. Backtracking forcibility is a promising approach to find the forcible paths which reach one state from another [Macktoobian et al. 2017; Macktoobian 2018]. Thus, the complete coordination problem can be written in the language of backtracking forcibility as follows.

**Definition 20** [Forcibility]**.** Let **G** be a discrete-event system. Given two states $q, q' \in Q_{\mathbf{G}}$ and a string $s \in \Sigma^{*}_{\mathbf{G}}$, if $s$ forcibly reaches $q'$ from $q$, then the following ternary relation holds.

$$\mathcal{F}(s, q, q') \tag{4.27}$$

**Problem 2** [Complete Coordination]**.** Let **G** be a discrete-event system representing a system of astrobots, **S** be the specification associated with **G**, and **V** be the coordination supervisor synthesized based them. Let

**Algorithm** Complete Coordination Checker ($\mathcal{CCC}$)

**Input:**
**V** ▷ Coordination supervisor
**Output:**
Complete coordination feasibility

```
if $\mathcal{BFA}(\mathbf{V}, q_{0_\mathbf{V}}, q_{m_\mathbf{V}}) \neq \varnothing$ then
    return True
else
    return False
end if
```

$q_{0\mathbf{V}}$ and $q_{m\mathbf{V}}$ be the initial and the marked states of **V**, respectively. Subject to an appropriate specification of forcibility, check whether the following *completeness condition* holds.

$$(\exists s \in \Sigma_{\mathbf{V}}^{*})\mathcal{F}(s, q_{0\mathbf{V}}, q_{m\mathbf{V}}) \tag{4.28}$$

We solve the problem above using the backtracking forcibility notion of supervisory control theory as illustrated by the $\mathcal{CCC}$ algorithm. In particular, the completeness condition (4.28) is checked. $\mathcal{BFA}$ function collects the forcible paths (if any) belonging to the marked behavior of **V** which forcibly reach the marked state of **V** from its initial state. If the result is nonempty, then the complete coordination of the desired system is feasible.

In the next section, we present an example whose completeness problem is assessed using the $\mathcal{CCC}$ algorithm.

## 4.4 Example

We solve a complete coordination problem using supervisory control theory implemented in TCT software [Wonham 2017a]. We consider three astrobots $\mathbf{G}^{\pi_1}$, $\mathbf{G}^{\pi_2}$, and $\mathbf{G}^{\pi_3}$ as depicted in Figure 4.5 whose initial states are $[\mathbf{0}]$, $[\mathbf{6}]$, and $[\mathbf{6}]$, respectively. The events of the astrobots are specified in Table 4.1 where events with odd labels are controllable, and those with even labels are uncontrollable. Suppose the relative priorities of the astrobots are the same as their indices. Specifications $\mathbf{S}^{\pi_1}$, $\mathbf{S}^{\pi_2}$, and $\mathbf{S}^{\pi_3}$ are illustrated in Figures 4.6, 4.7, and 4.8, respectively.

We compute the coordination supervisor **V** according to (4.26) whose marked state is $[\mathbf{14}]$. Using the $\mathcal{CCC}$ algorithm, we obtain the (shortest) forcible path $Z = \langle 73, 10, 30, 15, 31, 81, 20\rangle$ from $[\mathbf{0}]$ to $[\mathbf{14}]$. In particular, **V** commands $\pi_1$ to rotate in a counterclockwise direction by enabling $\langle 73\rangle$. Then, $\pi_1$

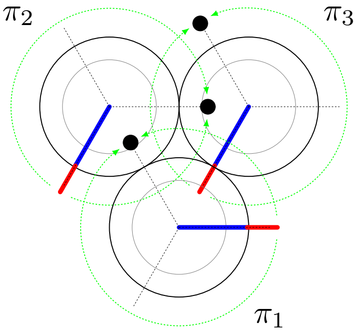


Figure 4.5: A coordination problem

Table 4.1: Event specifications of the example astrobots (the astrobots indices are intentionally omitted for better readibility.)

| Astrobot | $v$ | $w$ | $l$ | $m$ | $h$ | $k$ | $e$ |
|---|---|---|---|---|---|---|---|
| $\pi_1$ | 71 | 73 | 11 | 13 | 15 | 10 | 40 |
| $\pi_2$ | 81 | 83 | 21 | 23 | 25 | 20 | 50 |
| $\pi_3$ | 91 | 93 | 31 | 33 | 35 | 30 | 60 |

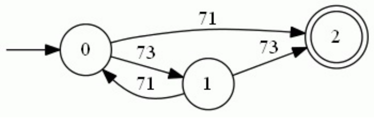


Figure 4.6: The generator of $S^{\pi_1}$

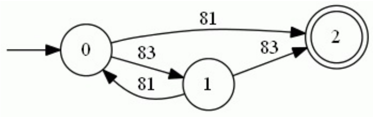


Figure 4.7: The generator of $S^{\pi_2}$

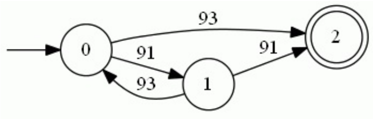


Figure 4.8: The generator of $S^{\pi_3}$

enters the safety zone of $\pi_3$. So, the collision events $\langle 10 \rangle$ and $\langle 30 \rangle$ occur. Since $\pi_1$ has a higher priority, $\langle 15 \rangle$ drives $\pi_1$ to its target point in a clockwise direction. In contrast, $\pi_3$ goes far from $\pi_1$ to avoid any collision by $\langle 31 \rangle$. So, $\pi_3$ eventually reaches its target point. $\langle 71 \rangle$ initially moves $\pi_2$ in a clockwise direction. However, it enters the safety zone of $\pi_1$, so $\langle 20 \rangle$ occurs, and $\pi_2$ changes it direction which gives rise to its convergence.

## 4.5 Conclusion

We report a supervisory-control-based solution to the complete safe coordination problem of astrobot swarms. We illustrate how to model astrobots systems and their specifications as discrete-event systems. We then generate a coordination supervisor in which the solution(s) to the problem were embedded. We found the solution(s) using the backtracking forcibility. The behavior of this supervisor is straightforwardly verifiable using SAT-solver tools, in contrary to the method based artificial potential fields which heavily rely on simulations.

# Chapter 5

# Cooperative Complete Coordination

### Publication Note

The material presented in this chapter is adapted from:

- **Matin Macktoobian**, D. Gillet, and J-P. Kneib, Complete Coordination of astrobots for Massive Spectroscopic Surveys, *Journal of Astronomical Telescopes, Instruments, and Systems* 5(4), Article 045002, 2019.
- **Matin Macktoobian**, Ricardo Araújo, Loïc Grossen, Luzuis Kronig, Mohamed Bouri, Denis Gillet, and Jean-Paul Kneib, Experimental Evaluation of Complete Safe Coordination of Astrobots for the Sloan Digital Sky Survey V, *Experimental Astronomy* 51(1), pp. 77-94, 2021.
- **Matin Macktoobian**, D. Gillet, and J-P. Kneib, The Navigation of astrobots in SDSS-V Project: Design and Implementation, in *the 15th Conference on PhD Research in Microelectronics and Electronics (PRIME)*, pp. 85-88, IEEE, 2019.

## 5.1 Introduction

Supervisory control theory [Ramadge et al. 1987] was used to seek complete coordination of astrobots [Macktoobian et al. 2019c]. The major hurdle to use this approach is the curse of dimensionality when the size of a astrobots system grows. Then, the required processing is not practically feasible to find a complete solution. To be specific, the coordination of astrobots is challenging because any solution to this problem has to fulfill some critical requirements in both spatial and temporal perspectives. In particular, astrobots are often arranged in hexagonal formations, so each astrobot neighbors 6 other peers. The on-time coordination of the astrobots set is desired after finishing a specific observation to point to the objects of the next observation. Since each observation is extremely time-dependent, the coordination of the system shall be executed in a limited amount of time between two successive observations. Thus, the solution to the coordination problem of astrobots has to be both reliable against collisions and efficient in view of performance.

The solutions to the trajectory planning and the collision avoidance problems directly depend on the number and the mechanical specifications of the used astrobots in a particular subproject of the SDSS project. For example in the case of the "The Dark Spectroscopic Instrument" (DESI) [Aghamousa et al. 2016a; Aghamousa et al. 2016b] project, an artificial potential field (APF) approach is proposed to solve the collision-free trajectory planning of astrobots. This method uses a decentralized navigation function based on the notion of artificial potential fields. In particular, the arms of the astrobots used in this project are long enough to enter the workspace of any neighboring astrobots. However since the contentions are not considerable, all the astrobots can converge to their target positions. In other words, the astrobots of the DESI project compose a complete system. In contrast, the completeness is not realized in the case of "The Multi Object Optical and Near-infrared Spectrograph" (MOONS) [Michele Cirasuolo et al. 2014] project. In this case, the length of the second arm of each astrobot is two times longer than those of the astrobots of the DESI project. To solve the trajectory planning problem associated with the MOONS project, the planning algorithm was modified [Tao et al. 2018] to take two subjects into account. First, not every colliding situations is managed by the navigation function. So, a priority-based decision-making layer was added to the decentralized navigation function to handle deadlocks and oscillations which could not be handled by the navigation function. Based on this approach, the astrobots which are assigned to more important objects are prioritized in the coordination of the system. Thus, some astrobots may not reach their target at all. The algorithm cannot generally coordinate the system such that all astrobots reach their target positions. In other words, the coordination problem is not complete with respect to the solutions of this algorithm. Complete coordination leads to the collection of the full information which is planned to be collected during an observation. However, no analysis has been yet applied to explore the conditions based on which a solution to a coordination problem of astrobots is complete. Another coordination method for SDSS-V project does also exist [Sayres et al. 2020]. This method coordinate astrobots in a two-phase manner. In particular, to reach any general amorph configuration to another one, this strategy first move astrobots to their fully-folded formation. Then, the second phase of coordination seeks the final desired configuration. This method has some disadvantages. First, its two-phase coordination are taken into account to minimize the conflicting scenarios between astrobots. However, one may note that such two-phase coordination reach targets later than direct ones. So, we seek a coordination strategy which directly reaches any configuration from any arbitrary initial one. Second, the method seems to improve the convergence rates of astrobots compared to [Tao et al. 2018]. But, it does not claim to reach general completeness. Alternatively, we establish a coordination method whose completeness may be formally investigated. Thus, the cited gaps open an avenue for the potential modification of the current artificial-potential-based coordination algorithm to realize the complete safe coordination of astrobots.

In this chapter, we formally analyze and solve the complete coordination problem associated with astrobot swarms. We obtain a completeness condition whose fulfillment guarantees the complete coordination. The remainder of the report is organized and follows. Section 5.2 establishes the global completeness problem whose solution shall guarantee the convergence of all astrobots of a telescope. We then define the local completeness problem corresponding to the convergence of an astrobot and all of its neighboring astrobots. In particular, we take a distributed scheme into account to show that given a set of astrobots, if all local completeness problems corresponding to neighboring region of the system are complete, then the overall system is globally complete. Section 5.3 proposes a new class of artificial potential fields, i.e., cooperative artificial potential fields (CAPFs). The advantage of a CAPF compared to an APF is that the attractive term of the CAPF considers not only the convergence of its own astrobot agent but also the convergence of its neighboring astrobots. Thanks to the proved solvability of the global completeness problem based on the completeness of its local completeness problems, Section 5.4.1 obtains the required condition for the solvability of the local completeness problem. Section 5.4.2 establishes a strategy for completeness seeking when a system of astrobots is incomplete with respect to a particular set of parameter specifications of the system. In these situations, we indeed propose to modify the paramaters corresponding to the specification of the system's CAPFs and/or the definition of the desired observation to resolve the encountered incompleteness. Section 5.5 compares CAPF to APF in view of the properties of the navigation process such as computational complexity and convergence time. We evaluate our accomplishments by simulations in Section 5.6.1 and experimental validations in Section 5.6.2. Our concluding remarks are finally drawn in Section 5.7.

## 5.2 From Local to Global Completeness

In this section, we define the global and the local completeness problems. We then show that the solvability of a global completeness problem is equivalent to the solvability of all the local completeness problems associated with it.

We first define the notions of "position", "target position", and "equilibrium position" corresponding to an astrobot as follows.

**Definition 21** [Position]. Let $\mathcal{P}$ be the set of all the astrobots associated with a telescope. Given an astrobot $\pi^i \in \mathcal{P}$, $\boldsymbol{q^i} := \begin{bmatrix} x^i & y^i \end{bmatrix}^\intercal$ denotes the (planar) *position* of $\pi^i$.

**Definition 22** [Target Position]. Let $\mathcal{P}$ be the set of all the astrobots associated with a telescope. Given an astrobot $\pi^i \in \mathcal{P}$, $\boldsymbol{q^i_{\mathcal{T}}} := \begin{bmatrix} x^i_{\mathcal{T}} & y^i_{\mathcal{T}} \end{bmatrix}^\intercal$ represents the *target position* at which $\pi^i$ is planned to reach according to a specific observation.

**Definition 23** [Equilibrium Position]. Let $\mathcal{P}$ be the set of all the astrobots associated with a telescope. Given an astrobot $\pi^i \in \mathcal{P}$, $\boldsymbol{q^i_{\star}} := \begin{bmatrix} x^i_{\star} & y^i_{\star} \end{bmatrix}^\intercal$ represents the *equilibrium position* at which $\pi^i$ resides at the end of the coordination process.

**Definition 24** [Index Set]. Let $\mathcal{N} := \{\pi^1, \cdots, \pi^n\}$ be the set of $n$ astrobots. Then, $\mathcal{I}_{\mathcal{N}}$ is the *index set* of $\mathcal{N}$ denoting the set of all the indices of the elements of $\mathcal{N}$ as follows

$$\mathcal{I}_{\mathcal{N}} := \{\arg_k \pi^k | \forall \pi^k \in \mathcal{N}\}, \tag{5.1}$$

where $\arg(\cdot)$ operator returns the index of its arguments.

Now we define the "global completeness problem" as follows.

**Problem 3** [Global Completeness]. Subject to a set of astrobots $\mathcal{P}$ and its corresponding index set $\mathcal{I}_{\mathcal{P}}$, determine whether or not the following relation holds.

$$(\forall k \in \mathcal{I}_{\mathcal{P}}) \boldsymbol{q^k_{\star}} = \boldsymbol{q^k_{\mathcal{T}}} \tag{5.2}$$

Because of the dense hexagonal arrangements of astrobots in a focal plane, the direct solution to the problem above would be difficult. Instead, we define a local version of the completeness problem, and we show that how the solutions to a set of local completeness problems end up with the solution to the global completeness problem corresponding to them. For this purpose, we first define the notion of "neighborhood" with respect to a specific astrobot.

**Definition 25** [Neighborhood]. Let $\mathcal{P}$ be the set of all the astrobots associated with a telescope. Let $\pi^i \in \mathcal{P}$ be an astrobot. Given $\mathcal{V}^i$ denoting the neighboring astrobots of $\pi^i$, $\mathcal{N}^i \subseteq \mathcal{P}$ is the *neighborhood* with respect to $\pi^i$ defined as follows

$$\mathcal{N}^i := \mathcal{V}^i \dot{\bigcup} \{\pi^i\}. \tag{5.3}$$

The following definition establishes the "local completeness problem".

**Problem 4** [Local Completeness]. Let $\mathcal{P}$ be the set of all the astrobots associated with a telescope. Subject to the neighborhood $\mathcal{N}^i \subseteq \mathcal{P}$ with respect to an astrobot $\pi^i \in \mathcal{P}$, determine whether or not the following holds.

$$(\forall k \in \mathcal{I}_{\mathcal{N}^i}) \boldsymbol{q}^k_\star = \boldsymbol{q}^k_{\mathcal{T}} \tag{5.4}$$

Using the definition above, we establish the notion of "completeness relation"

**Definition 26** [Completeness Relation]. Let $\mathcal{P}$ be the set of all the astrobots associated with a telescope. Let also $\mathcal{N}^i \subseteq \mathcal{P}$ be a neighborhood with respect to the astrobot $\pi^i \in \mathcal{P}$. Then, if $\mathcal{N}^i$ is locally complete, then the following relation holds.

$$\mathcal{C}(\mathcal{N}_i) \tag{5.5}$$

As well, given the set of all neighborhoods $\mathcal{N}$ corresponding to astrobots set $\mathcal{P}$, if $\mathcal{N}$ is globally complete then $\mathcal{C}(\mathcal{N})$ holds.

We prove the following property of the completeness relation which is subsequently used to show the relationship between the notions of local and global completeness.

**Lemma 6.** *Completeness relation is closed under countable union operator.*

*Proof.* Let $\mathcal{N}^i$ and $\mathcal{N}^j$ be two locally complete neighborhoods with respect to astrobots $\pi^i$ and $\pi^j$, respectively, i.e., $\mathcal{C}(\mathcal{N}^i)$ and $\mathcal{C}(\mathcal{N}^j)$ hold. Then, the following two cases shall be mutual exclusively considered.

- $\mathcal{N}^i$ and $\mathcal{N}^j$ are not adjacent to each other, say,

  $$(\forall \pi \in \mathcal{N}^i) \pi \notin \mathcal{N}^j. \tag{5.6}$$

  Accordingly, there is no interaction between the quoted neighborhoods. So, the every astrobot also reaches its target position after the unification of the neighborhoods. Therefore, the resulting union in complete.

- $\mathcal{N}^i$ and $\mathcal{N}^j$ are adjacent to each other, say,

  $$(\exists \pi \in \mathcal{N}^i) \pi \in \mathcal{N}^j. \tag{5.7}$$

  In a hexagonal arrangement of astrobots, the minimum and the maximum numbers of the shared astrobots[a] between two adjacent neighborhoods are 1 and 3, respectively. Considering the minimum case, let $\pi$ be the shared astrobot, so it is the exclusive astrobot which can potentially disturb the overall completeness of $\mathcal{N}^i$ and $\mathcal{N}^j$. However according to the assumption of the completeness of both neighborhoods, $\pi$ shall reach its target in view of both neighborhoods. Thus, the unification of the neighborhoods is complete. The similar argument is valid to justify the completeness of the unified system of complete neighborhoods where the number of shared events is 2 or 3, as well.

■

---

[a] The maximum number of the shared astrobots varies with respect to the lengths of the astrobots' arms. The reader finds a thorough analysis of the cited relationship in [Hörler 2018].

Finally, the following theorem uses Lemma 6 to state how the local and the global completeness problems are related to each other.

**Theorem 7.** *Let $\mathcal{N}$ be the set of all neighborhoods to which the astrobots of a telescope are assigned. So, if all neighborhoods are locally complete, then the overall system of the astrobots is complete, i.e.,*

$$[(\forall \mathcal{N}^i \in \mathcal{N})\mathcal{C}(\mathcal{N}^i)] \Rightarrow \mathcal{C}(\mathcal{N}). \tag{5.8}$$

*Proof.* By induction, we show that the proof is a consequence of Lemma 6. In particular, let $k$ be the number of the neighborhoods. Then, we have

- base case: $k = 1$, say, the astrobots set includes only one (complete) neighborhood. So, the overall system is obviously complete.
- induction step: suppose the system with $k = n$ is complete, i.e., $\mathcal{C}(\bigcup_{i=1}^{n} \mathcal{N}^i)$ holds. We show that the system with $k = n + 1$ has to be complete. In particular given complete neighborhood $\mathcal{N}^{n+1}$, since the completeness relation is closed under countable union operator (see, Lemma 6), hence we have

$$\begin{aligned} \mathcal{C}(\bigcup_{i=1}^{n} \mathcal{N}^i) \cup \mathcal{C}(\mathcal{N}^{n+1}) &= \mathcal{C}(\bigcup_{i=1}^{n} \mathcal{N}_i \cup \mathcal{N}^{n+1}) \\ &= \mathcal{C}(\bigcup_{i=1}^{n+1} \mathcal{N}^i), \end{aligned} \tag{5.9}$$

  which concludes the global completeness of the astrobots set.

■

we later (see, Section 5.4) analyze the completeness condition for local systems. Thanks to the result of Theorem 7, if the conditions corresponding to the completeness of all of the neighborhoods are hold, then the global system is also complete. In the next section, we rewrite the definitions of the local and the global completeness problems in the language of artificial potential fields (APFs). Then, we revise the formulation of the decentralized navigation function, used in priority-based coordination method [Tao et al. 2018]. So, the equilibrium of the new APF could represent the complete result of a coordination process. We then uncover the condition for the existence of a solution to the local completeness problem.

## 5.3 Cooperative Artificial Potential Fields

In this section, we first explain our motivation to define a new type of APFs, called "cooperative artificial potential field" (CAPF). In particular, we elaborate on the effect of an APF on the completeness of the coordination process. In particular, Section 5.3.1 clarifies our angle of attack to tackle the completeness problem. Then, we formally introduce our proposed CAPF in Section 5.3.2. We also reformulate the local and the global completeness problems using the notion of CAPF.

### 5.3.1 Motivation

The priority-based algorithm [Tao et al. 2018] uses a two-layer competitive architecture to solve the coordination problem, as depicted in Figure 5.1.

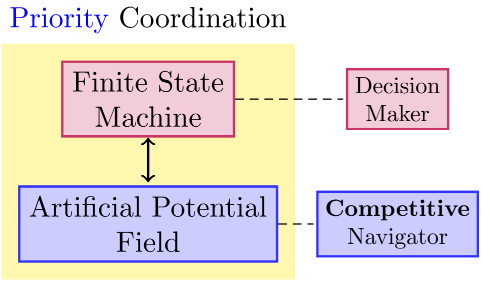


Figure 5.1: The competitive control architecture of the priority-based coordination

Let $\lambda_1$ and $\lambda_2$ be positive constant weighting factors. Let also $D$ be the radius of the collision avoidance envelope in which the repulsive term is activated. $d_0$ also represents the radius of the safety region around each astrobot. Then, we define

$$\begin{cases} \boldsymbol{\lambda}_1 := \lambda_1 \otimes \mathbb{I}_2, & (5.10) \\ \boldsymbol{\lambda}_2 := \lambda_2 \otimes \mathbb{I}_2. & (5.11) \end{cases}$$

Thus, the definition of the reference APF used in [Tao et al. 2018] is represented as follows

$$\psi(\boldsymbol{q}^i) := \underbrace{\boldsymbol{\lambda}_1 \|\boldsymbol{q}^i - \boldsymbol{q}^i_{\mathcal{T}}\|^2}_{\text{attractive term}} + \underbrace{\boldsymbol{\lambda}_2 \sum_{j \in \mathcal{I}_{\mathcal{N}^i} \setminus \{i\}} \min \left[ \mathbf{0}, \frac{\|\boldsymbol{q}^i - \boldsymbol{q}^j\|^2 - D^2}{\|\boldsymbol{q}^i - \boldsymbol{q}^j\|^2 - d_0^2} \right]}_{\text{repulsive term}}. \tag{5.12}$$

One notes that the attractive term above exclusively takes the convergence of the astrobot $\pi^i$ into account. So, the APFs corresponding to different astrobots in fact compete with each other because each artificial potential field only cares about the convergence of its own affiliated astrobot. Since an astrobot does not care about the convergence of its neighbors, this competitive manner of navigation potentially gives rise to the incomplete coordination of the overall system of astrobots.

Instead, we propose a cooperative scheme based on which each astrobot not only seeks its own convergence, but also cares about the convergence of its neighboring counterparts. Thus, the competitive architecture can be modified based on this cooperative perspective as depicted in Figure 5.2.

### 5.3.2 Formulation

We embed a particular attractive term in the definition of the reference APF (see, Eq. (5.12)) to realize the cooperation between neighboring astrobots to reach collective convergence to their target spots. Let $\lambda_3$ be a positive weighting factor corresponding to the cooperative attractive term. Let also $\boldsymbol{q}^i_{\mathcal{T}}$ (resp. $\boldsymbol{q}^j_{\mathcal{T}}$) be the target position of $\boldsymbol{q}^i$ (resp. $\boldsymbol{q}^j$). Then considering

$$\boldsymbol{\lambda}_3 := \lambda_3 \otimes \mathbb{I}_2, \tag{5.13}$$

we define a CAPF into which a cooperative attractive term is integrated as follows.

$$\rho(\boldsymbol{q}^i) := \underbrace{\boldsymbol{\lambda}_1\|\boldsymbol{q}^i - \boldsymbol{q}^i_{\mathcal{T}}\|^2 +}_{\text{attractive term}} \underbrace{\boldsymbol{\lambda}_3 \sum_{j\in\mathcal{I}_{\mathcal{N}^i}\setminus\{i\}} \|\boldsymbol{q}^j - \boldsymbol{q}^j_{\mathcal{T}}\|^2}_{\text{cooperative attractive term}} \underbrace{+\boldsymbol{\lambda}_2 \sum_{j\in\mathcal{I}_{\mathcal{N}^i}\setminus\{i\}} \min\left[\boldsymbol{0}, \frac{\|\boldsymbol{q}^i - \boldsymbol{q}^j\|^2 - D^2}{\|\boldsymbol{q}^i - \boldsymbol{q}^j\|^2 - d_0^2}\right]}_{\text{repulsive term}} \tag{5.14}$$

The cooperative attractive term inserts extra dynamics to the reference APF to involve all astrobots of a neighborhood in the convergence process. A rough guideline to set the value of $\lambda_3$ is $\lambda_3 < \lambda_1$ for two reasons. First, each CAPF instance should mainly focus on the convergence of its corresponding astrobot. So, one selects a larger weighting factor for the main astrobot to insure that the main portion of the attractive force of its corresponding CAPF comes from that astrobot. Second, $\lambda_3$ in fact injects the velocity profile of the neighboring astrobots to that of the main astrobot. Any large values corresponding to those velocity profiles may give rise to abrupt motions imposed to the main astrobot. Such unwanted and uncontrolled motions may not only damage the main astrobot's actuators but also leave it vulnerable to potential collisions.

We are interested in the conditions based on which a solution to a specific coordination problem is complete. Thus, we formulate the local and the global completeness problems in the language of CAPF. In particular, the equilibrium points for all astrobots in a neighborhood shall be their target points. Since the astrobots exclusively stop moving at their target points, one needs to obtain the equilibrium points corresponding to the derivative of CAPF as follows.

$$\nabla\rho(\boldsymbol{q}^i) = \begin{cases} 2\boldsymbol{\lambda}_1(\boldsymbol{q}^i - \boldsymbol{q}^i_{\mathcal{T}}) + 2\boldsymbol{\lambda}_3 \sum\limits_{j\in\mathcal{I}_{\mathcal{N}^i}\setminus\{i\}} (\boldsymbol{q}^j - \boldsymbol{q}^j_{\mathcal{T}}) \\ \qquad (\forall j \in \mathcal{I}_{\mathcal{N}^i} \setminus \{i\}) \|\boldsymbol{q}^i - \boldsymbol{q}^j\| \geq D \\ 2\boldsymbol{\lambda}_1(\boldsymbol{q}^i - \boldsymbol{q}^i_{\mathcal{T}}) + 2\boldsymbol{\lambda}_3 \sum\limits_{j\in\mathcal{I}_{\mathcal{N}^i}\setminus\{i\}} (\boldsymbol{q}^j - \boldsymbol{q}^j_{\mathcal{T}}) + 2\boldsymbol{\lambda}_2 \sum\limits_{j\in\mathcal{I}_{\mathcal{N}^i}\setminus\{i\}} \dfrac{(D^2 - d^2)(\boldsymbol{q}^i - \boldsymbol{q}^j)}{\left(\|\boldsymbol{q}^i - \boldsymbol{q}^j\|^2 - d_0^2\right)^2} \\ \qquad (\exists j \in \mathcal{I}_{\mathcal{N}^i} \setminus \{i\}) \|\boldsymbol{q}^i - \boldsymbol{q}^j\| < D \end{cases} \tag{5.15}$$

Then, the control law below is proposed to be applied to the joints of the astrobot $\pi^i$.

$$\boldsymbol{u}^i := -\nabla_{\theta_i, \phi_i}\rho(\boldsymbol{q}^i) \tag{5.16}$$

Now we can compose the CAPF-driven formalism of the local and the global completeness problems as follows.

**Problem 5** [Local Completeness (CAPF Derivation)]. Let $\mathcal{N}^i$ be a neighborhood with respect to the astrobot $\pi^i$ where $2 \leq |\mathcal{N}^i| \leq 7$. Then, the neighborhood is locally complete coordinated by a set of CAPFs

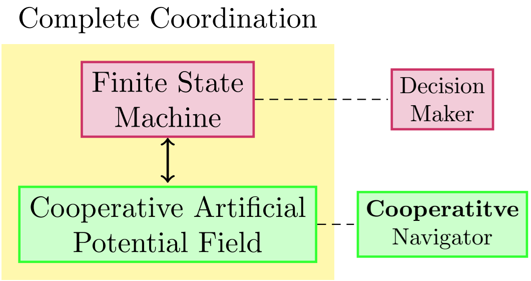


Figure 5.2: The cooperative control architecture of the complete coordination

if the following differential equations are simultaneously solvable.

$$\nabla\rho(\boldsymbol{q}^i) = 0 \quad \text{for} \quad 1 \leq i \leq |\mathcal{N}^i| \tag{5.17}$$

The global completeness problem is the generalization of the local completeness problem above as below.

**Problem 6** [Global Completeness (CAPF Derivation)]**.** Let $\mathcal{P}$ be the set of all astrobots of a telescope. Then, the overall system is globally complete coordinated by a set of CAPFs if the following differential equations are simultaneously solvable.

$$\nabla\rho(\boldsymbol{q}^i) = 0 \quad \text{for} \quad 1 \leq i \leq |\mathcal{P}| \tag{5.18}$$

**Remark 4.** Each CAPF has one exclusive minimum because it is a smooth Morse function [Milnor 1963], it is uniformly maximal on boundaries of a free space, and it has a unique minimum at a target point in its free space [Makarem 2015]. Then, if Eq. (5.17) (resp., Eq. (5.18)) is solvable, then its solution is essentially $\boldsymbol{q}_{\mathcal{T}} := [\boldsymbol{q}_{\mathcal{T}}^1 \; \dots \; \boldsymbol{q}_{\mathcal{T}}^{|\mathcal{N}^i|}]^\intercal$(resp., $\boldsymbol{q}_{\mathcal{T}} := [\boldsymbol{q}_{\mathcal{T}}^1 \; \dots \; \boldsymbol{q}_{\mathcal{T}}^{|\mathcal{P}|}]^\intercal$).

In the next section, we find the conditions for guaranteed solvability of the local and the global completeness problems.

## 5.4 Completeness Analysis

The preceding section revealed that the solutions to all local completeness problems are the keys to determine whether the global completeness problem corresponding to those problems is solvable. Here Section 5.4.1 focuses on the required condition for the completeness of a neighborhood. Then, Section 5.4.2 discusses a procedure based on which completeness is sought regarding a system of astrobots which is not complete according to a particular set of parameters.

### 5.4.1 Completeness Condition

We take a typical isolated neighborhood with the maximum number of astrobots, say, $\{\pi^i | 0 \leq i \leq 6\}$. We also consider the maximum contention between the astrobots of the neighborhood. In particular, we assume that two neighboring astrobots, e.g., $\pi^1$ and $\pi^2$, are at the collision zone of the central astrobot, i.e., $\pi^0$. The remaining four astrobots are assumed to be residing at each other's collision zones in a pair-wise manner, say, $\pi^3$ and $\pi^4$, and $\pi^5$ and $\pi^6$. Figure 5.3 represents the configuration of the neighborhood, in which the regions with the same color correspond to those astrobots which are suspected to collide and to block each other's movements. This scenario is the most collision-susceptible case to reach the full completeness for the explained neighborhood.

According to Problem 5, we need to find the solutions which simultaneously fulfill the following set of equations.

$$\nabla\rho(\boldsymbol{q}^i) = \boldsymbol{0} \quad \text{for} \quad 0 \leq i \leq 6 \tag{5.19}$$

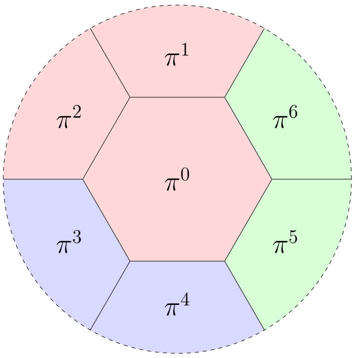


Figure 5.3: The arrangement of astrobots in a typical neighborhood subject to the maximum contention (The regions with the same color correspond to those astrobots which are suspected to collide and to block each other's movements.)

Using Eq. (5.15), we expand Eq. (5.19) as follows.

$$\begin{cases}
\nabla\rho(\boldsymbol{q^0}) = 2\boldsymbol{\lambda_1}(\boldsymbol{q^0} - \boldsymbol{q^0_{\mathcal{T}}}) + 2\boldsymbol{\lambda_3}\sum\limits_{j\in\{1,2\}}(\boldsymbol{q^j} - \boldsymbol{q^j_{\mathcal{T}}}) + 2\boldsymbol{\lambda_2}(D^2 - d_0^2)\sum\limits_{j\in\{1,2\}}\dfrac{\boldsymbol{q^0} - \boldsymbol{q^j}}{\left(\|\boldsymbol{q^0} - \boldsymbol{q^j}\|^2 - d_0^2\right)^2} = \boldsymbol{0} \\
\nabla\rho(\boldsymbol{q^1}) = 2\boldsymbol{\lambda_1}(\boldsymbol{q^1} - \boldsymbol{q^1_{\mathcal{T}}}) + 2\boldsymbol{\lambda_3}\sum\limits_{j\in\{0,2\}}(\boldsymbol{q^j} - \boldsymbol{q^j_{\mathcal{T}}}) + 2\boldsymbol{\lambda_2}(D^2 - d_0^2)\sum\limits_{j\in\{0,2\}}\dfrac{\boldsymbol{q^1} - \boldsymbol{q^j}}{\left(\|\boldsymbol{q^1} - \boldsymbol{q^j}\|^2 - d_0^2\right)^2} = \boldsymbol{0} \\
\nabla\rho(\boldsymbol{q^2}) = 2\boldsymbol{\lambda_1}(\boldsymbol{q^2} - \boldsymbol{q^2_{\mathcal{T}}}) + 2\boldsymbol{\lambda_3}\sum\limits_{j\in\{0,1\}}(\boldsymbol{q^j} - \boldsymbol{q^j_{\mathcal{T}}}) + 2\boldsymbol{\lambda_2}(D^2 - d_0^2)\sum\limits_{j\in\{0,1\}}\dfrac{\boldsymbol{q^2} - \boldsymbol{q^j}}{\left(\|\boldsymbol{q^2} - \boldsymbol{q^j}\|^2 - d_0^2\right)^2} = \boldsymbol{0} \\
\nabla\rho(\boldsymbol{q^3}) = 2\boldsymbol{\lambda_1}(\boldsymbol{q^3} - \boldsymbol{q^3_{\mathcal{T}}}) + 2\boldsymbol{\lambda_3}(\boldsymbol{q^4} - \boldsymbol{q^4_{\mathcal{T}}}) + 2\boldsymbol{\lambda_2}(D^2 - d_0^2)\dfrac{\boldsymbol{q^3} - \boldsymbol{q^4}}{\left(\|\boldsymbol{q^3} - \boldsymbol{q^4}\|^2 - d_0^2\right)^2} = \boldsymbol{0} \\
\nabla\rho(\boldsymbol{q^4}) = 2\boldsymbol{\lambda_1}(\boldsymbol{q^4} - \boldsymbol{q^4_{\mathcal{T}}}) + 2\boldsymbol{\lambda_3}(\boldsymbol{q^3} - \boldsymbol{q^3_{\mathcal{T}}}) + 2\boldsymbol{\lambda_2}(D^2 - d_0^2)\dfrac{\boldsymbol{q^4} - \boldsymbol{q^3}}{\left(\|\boldsymbol{q^4} - \boldsymbol{q^3}\|^2 - d_0^2\right)^2} = \boldsymbol{0} \\
\nabla\rho(\boldsymbol{q^5}) = 2\boldsymbol{\lambda_1}(\boldsymbol{q^5} - \boldsymbol{q^5_{\mathcal{T}}}) + 2\boldsymbol{\lambda_3}(\boldsymbol{q^6} - \boldsymbol{q^6_{\mathcal{T}}}) + 2\boldsymbol{\lambda_2}(D^2 - d_0^2)\dfrac{\boldsymbol{q^5} - \boldsymbol{q^6}}{\left(\|\boldsymbol{q^5} - \boldsymbol{q^6}\|^2 - d_0^2\right)^2} = \boldsymbol{0} \\
\nabla\rho(\boldsymbol{q^6}) = 2\boldsymbol{\lambda_1}(\boldsymbol{q^6} - \boldsymbol{q^6_{\mathcal{T}}}) + 2\boldsymbol{\lambda_3}(\boldsymbol{q^5} - \boldsymbol{q^5_{\mathcal{T}}}) + 2\boldsymbol{\lambda_2}(D^2 - d_0^2)\dfrac{\boldsymbol{q^6} - \boldsymbol{q^5}}{\left(\|\boldsymbol{q^6} - \boldsymbol{q^5}\|^2 - d_0^2\right)^2} = \boldsymbol{0}
\end{cases}$$

To compact the set of equations above, we define the following auxiliary function

$$\mathcal{Q}(\boldsymbol{q^\alpha}, \boldsymbol{q^\beta}) := \frac{\boldsymbol{q^\alpha} - \boldsymbol{q^\beta}}{\left(\|\boldsymbol{q^\alpha} - \boldsymbol{q^\beta}\|^2 - d_0^2\right)^2},$$

and the constant parameter below

$$\omega := D^2 - d_0^2,$$

which yield

$$
\begin{cases}
\nabla\rho(\boldsymbol{q}^0) = 2\begin{bmatrix}\boldsymbol{\lambda}_1 & \boldsymbol{\lambda}_3 & \boldsymbol{\lambda}_3\end{bmatrix}\begin{bmatrix}\boldsymbol{q}^0 & \boldsymbol{q}^1 & \boldsymbol{q}^2\end{bmatrix}^{\mathsf{T}} + 2\omega\boldsymbol{\lambda}_2\big(\mathcal{Q}(\boldsymbol{q}^0,\boldsymbol{q}^1) + \mathcal{Q}(\boldsymbol{q}^0,\boldsymbol{q}^2)\big) \\
\qquad - 2\big(\boldsymbol{\lambda}_1\boldsymbol{q}_T^0 + \boldsymbol{\lambda}_3(\boldsymbol{q}_T^1 + \boldsymbol{q}_T^2)\big) = \boldsymbol{0}, \\
\nabla\rho(\boldsymbol{q}^1) = 2\begin{bmatrix}\boldsymbol{\lambda}_3 & \boldsymbol{\lambda}_1 & \boldsymbol{\lambda}_3\end{bmatrix}\begin{bmatrix}\boldsymbol{q}^0 & \boldsymbol{q}^1 & \boldsymbol{q}^2\end{bmatrix}^{\mathsf{T}} + 2\omega\boldsymbol{\lambda}_2\big(\mathcal{Q}(\boldsymbol{q}^1,\boldsymbol{q}^0) + \mathcal{Q}(\boldsymbol{q}^1,\boldsymbol{q}^2)\big) \\
\qquad - 2\big(\boldsymbol{\lambda}_1\boldsymbol{q}_T^1 + \boldsymbol{\lambda}_3(\boldsymbol{q}_T^0 + \boldsymbol{q}_T^2)\big) = \boldsymbol{0}, \\
\nabla\rho(\boldsymbol{q}^2) = 2\begin{bmatrix}\boldsymbol{\lambda}_3 & \boldsymbol{\lambda}_3 & \boldsymbol{\lambda}_1\end{bmatrix}\begin{bmatrix}\boldsymbol{q}^0 & \boldsymbol{q}^1 & \boldsymbol{q}^2\end{bmatrix}^{\mathsf{T}} + 2\omega\boldsymbol{\lambda}_2\big(\mathcal{Q}(\boldsymbol{q}^2,\boldsymbol{q}^0) + \mathcal{Q}(\boldsymbol{q}^2,\boldsymbol{q}^1)\big) \\
\qquad - 2\big(\boldsymbol{\lambda}_1\boldsymbol{q}_T^2 + \boldsymbol{\lambda}_3(\boldsymbol{q}_T^0 + \boldsymbol{q}_T^1)\big) = \boldsymbol{0}, \\
\nabla\rho(\boldsymbol{q}^3) = 2\begin{bmatrix}\boldsymbol{\lambda}_1 & \boldsymbol{\lambda}_3\end{bmatrix}\begin{bmatrix}\boldsymbol{q}^3 & \boldsymbol{q}^4\end{bmatrix}^{\mathsf{T}} + 2\omega\boldsymbol{\lambda}_2\mathcal{Q}(\boldsymbol{q}^3,\boldsymbol{q}^4) - 2\big(\boldsymbol{\lambda}_1\boldsymbol{q}_T^3 + \boldsymbol{\lambda}_3\boldsymbol{q}_T^4\big) = \boldsymbol{0}, \\
\nabla\rho(\boldsymbol{q}^4) = 2\begin{bmatrix}\boldsymbol{\lambda}_3 & \boldsymbol{\lambda}_1\end{bmatrix}\begin{bmatrix}\boldsymbol{q}^3 & \boldsymbol{q}^4\end{bmatrix}^{\mathsf{T}} + 2\omega\boldsymbol{\lambda}_2\mathcal{Q}(\boldsymbol{q}^4,\boldsymbol{q}^3) - 2\big(\boldsymbol{\lambda}_1\boldsymbol{q}_T^4 + \boldsymbol{\lambda}_3\boldsymbol{q}_T^3\big) = \boldsymbol{0}, \\
\nabla\rho(\boldsymbol{q}^5) = 2\begin{bmatrix}\boldsymbol{\lambda}_1 & \boldsymbol{\lambda}_3\end{bmatrix}\begin{bmatrix}\boldsymbol{q}^5 & \boldsymbol{q}^6\end{bmatrix}^{\mathsf{T}} + 2\omega\boldsymbol{\lambda}_2\mathcal{Q}(\boldsymbol{q}^5,\boldsymbol{q}^6) - 2\big(\boldsymbol{\lambda}_1\boldsymbol{q}_T^5 + \boldsymbol{\lambda}_3\boldsymbol{q}_T^6\big) = \boldsymbol{0}, \\
\nabla\rho(\boldsymbol{q}^6) = 2\begin{bmatrix}\boldsymbol{\lambda}_3 & \boldsymbol{\lambda}_1\end{bmatrix}\begin{bmatrix}\boldsymbol{q}^5 & \boldsymbol{q}^6\end{bmatrix}^{\mathsf{T}} + 2\omega\boldsymbol{\lambda}_2\mathcal{Q}(\boldsymbol{q}^6,\boldsymbol{q}^5) - 2\big(\boldsymbol{\lambda}_1\boldsymbol{q}_T^6 + \boldsymbol{\lambda}_3\boldsymbol{q}_T^5\big) = \boldsymbol{0}.
\end{cases}
$$

The equations set above can be written as follows

$$
\underbrace{\begin{bmatrix}\nabla\rho(\boldsymbol{q}^0)\\ \nabla\rho(\boldsymbol{q}^1)\\ \nabla\rho(\boldsymbol{q}^2)\\ \nabla\rho(\boldsymbol{q}^3)\\ \nabla\rho(\boldsymbol{q}^4)\\ \nabla\rho(\boldsymbol{q}^5)\\ \nabla\rho(\boldsymbol{q}^6)\end{bmatrix}}_{\nabla\rho(\boldsymbol{q})}
=
\underbrace{\begin{bmatrix}
2\boldsymbol{\lambda}_1 & 2\boldsymbol{\lambda}_3 & 2\boldsymbol{\lambda}_3 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\
2\boldsymbol{\lambda}_3 & 2\boldsymbol{\lambda}_1 & 2\boldsymbol{\lambda}_3 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\
2\boldsymbol{\lambda}_3 & 2\boldsymbol{\lambda}_3 & 2\boldsymbol{\lambda}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 2\boldsymbol{\lambda}_1 & 2\boldsymbol{\lambda}_3 & \boldsymbol{0} & \boldsymbol{0}\\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 2\boldsymbol{\lambda}_3 & 2\boldsymbol{\lambda}_1 & \boldsymbol{0} & \boldsymbol{0}\\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 2\boldsymbol{\lambda}_1 & 2\boldsymbol{\lambda}_3\\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 2\boldsymbol{\lambda}_3 & 2\boldsymbol{\lambda}_1
\end{bmatrix}}_{\boldsymbol{\Lambda}}
\underbrace{\begin{bmatrix}\boldsymbol{q}^0\\ \boldsymbol{q}^1\\ \boldsymbol{q}^2\\ \boldsymbol{q}^3\\ \boldsymbol{q}^4\\ \boldsymbol{q}^5\\ \boldsymbol{q}^6\end{bmatrix}}_{\boldsymbol{q}}
+2\omega\boldsymbol{\lambda}_2
\underbrace{\begin{bmatrix}
\mathcal{Q}(\boldsymbol{q}^0,\boldsymbol{q}^1) + \mathcal{Q}(\boldsymbol{q}^0,\boldsymbol{q}^2)\\
\mathcal{Q}(\boldsymbol{q}^1,\boldsymbol{q}^0) + \mathcal{Q}(\boldsymbol{q}^1,\boldsymbol{q}^2)\\
\mathcal{Q}(\boldsymbol{q}^2,\boldsymbol{q}^0) + \mathcal{Q}(\boldsymbol{q}^2,\boldsymbol{q}^1)\\
\mathcal{Q}(\boldsymbol{q}^3,\boldsymbol{q}^4)\\
\mathcal{Q}(\boldsymbol{q}^4,\boldsymbol{q}^3)\\
\mathcal{Q}(\boldsymbol{q}^5,\boldsymbol{q}^6)\\
\mathcal{Q}(\boldsymbol{q}^6,\boldsymbol{q}^5)
\end{bmatrix}}_{\boldsymbol{\Omega}}
$$

$$
+\underbrace{\begin{bmatrix}
2\big(\boldsymbol{\lambda}_1\boldsymbol{q}_T^0 + \boldsymbol{\lambda}_3(\boldsymbol{q}_T^1 + \boldsymbol{q}_T^2)\big)\\
2\big(\boldsymbol{\lambda}_1\boldsymbol{q}_T^1 + \boldsymbol{\lambda}_3(\boldsymbol{q}_T^0 + \boldsymbol{q}_T^2)\big)\\
2\big(\boldsymbol{\lambda}_1\boldsymbol{q}_T^2 + \boldsymbol{\lambda}_3(\boldsymbol{q}_T^0 + \boldsymbol{q}_T^1)\big)\\
2\big(\boldsymbol{\lambda}_1\boldsymbol{q}_T^3 + \boldsymbol{\lambda}_3\boldsymbol{q}_T^4\big)\\
2\big(\boldsymbol{\lambda}_1\boldsymbol{q}_T^4 + \boldsymbol{\lambda}_3\boldsymbol{q}_T^3\big)\\
2\big(\boldsymbol{\lambda}_1\boldsymbol{q}_T^5 + \boldsymbol{\lambda}_3\boldsymbol{q}_T^6\big)\\
2\big(\boldsymbol{\lambda}_1\boldsymbol{q}_T^6 + \boldsymbol{\lambda}_3\boldsymbol{q}_T^5\big)
\end{bmatrix}}_{\boldsymbol{\Theta}'}
= \boldsymbol{0},
$$

whose compact form reads

$$
\nabla\rho(\boldsymbol{q}) = \boldsymbol{\Lambda}\boldsymbol{q} + 2\omega\boldsymbol{\lambda}_2\boldsymbol{\Omega} + \boldsymbol{\Theta}' = \boldsymbol{0}. \tag{5.20}
$$

The entries of $\boldsymbol{\Omega}$ above include function $\mathcal{Q}(\cdot,\cdot)$ which is nonlinear. We note that both astrobots monotonically head to their target points. So as an approximation, we linearize this function at the point whose coordinates are the average of the target positions' coordinates associated with the arguments of the function.

Put differently, we linearize $\mathcal{Q}(q^\alpha, q^\beta)$ at $\begin{bmatrix}\frac{q^\alpha_\mathcal{T}+q^\beta_\mathcal{T}}{2} & \frac{q^\alpha_\mathcal{T}+q^\beta_\mathcal{T}}{2}\end{bmatrix}^\mathsf{T}$ which is the closest point to both astrobots. Thus, the Newton method gives the following approximation for $\mathcal{Q}(\cdot,\cdot)$.

$$\begin{aligned}\mathcal{Q}(q^\alpha, q^\beta) \approx \mathcal{Q}(q^\alpha_\mathcal{T}, q^\beta_\mathcal{T}) &+ \left.\frac{\partial \mathcal{Q}(q^\alpha, q^\beta)}{\partial q^\alpha}\right|_{\left(\frac{q^\alpha_\mathcal{T}+q^\beta_\mathcal{T}}{2}, \frac{q^\alpha_\mathcal{T}+q^\beta_\mathcal{T}}{2}\right)} \left(q^\alpha - \frac{q^\alpha_\mathcal{T}+q^\beta_\mathcal{T}}{2}\right) \\ &+ \left.\frac{\partial \mathcal{Q}(q^\alpha, q^\beta)}{\partial q^\beta}\right|_{\left(\frac{q^\alpha_\mathcal{T}+q^\beta_\mathcal{T}}{2}, \frac{q^\alpha_\mathcal{T}+q^\beta_\mathcal{T}}{2}\right)} \left(q^\beta - \frac{q^\alpha_\mathcal{T}+q^\beta_\mathcal{T}}{2}\right)\end{aligned} \tag{5.21}$$

Taking the auxiliary constant parameters below into account

$$\begin{cases} \overline{\Delta}_{\alpha,\beta} = \overline{\Delta}_{\beta,\alpha} := \dfrac{q^\alpha_\mathcal{T}+q^\beta_\mathcal{T}}{2}, & (5.22) \\ \Delta^\alpha_{\alpha,\beta} = \Delta^\alpha_{\beta,\alpha} := \left.\dfrac{\partial \mathcal{Q}(q^\alpha, q^\beta)}{\partial q^\alpha}\right|_{\left(\overline{\Delta}_{\alpha,\beta}, \overline{\Delta}_{\alpha,\beta}\right)}, & (5.23) \\ \Delta^\beta_{\alpha,\beta} = \Delta^\beta_{\beta,\alpha} := \left.\dfrac{\partial \mathcal{Q}(q^\alpha, q^\beta)}{\partial q^\beta}\right|_{\left(\overline{\Delta}_{\alpha,\beta}, \overline{\Delta}_{\alpha,\beta}\right)}, & (5.24) \end{cases}$$

(5.21) is simplified as below

$$\mathcal{Q}(q^\alpha, q^\beta) \approx \mathcal{Q}(q^\alpha_\mathcal{T}, q^\beta_\mathcal{T}) - \overline{\Delta}_{\alpha,\beta}(\Delta^\alpha_{\alpha,\beta} + \Delta^\beta_{\alpha,\beta}) + q^\alpha \Delta^\alpha_{\alpha,\beta} + q^\beta \Delta^\beta_{\alpha,\beta}. \tag{5.25}$$

Therefore, the linearized version of $\boldsymbol{\Omega}$, i.e., $\boldsymbol{\Omega}^\star$, is obtained as the following:

$$\boldsymbol{\Omega} \approx \boldsymbol{\Omega}^\star = \underbrace{\begin{bmatrix} \Delta^0_{0,1}+\Delta^0_{0,2} & \Delta^1_{0,1} & \Delta^2_{0,2} & 0 & 0 & 0 & 0 \\ \Delta^0_{1,0} & \Delta^1_{1,0}+\Delta^1_{1,2} & \Delta^2_{1,2} & 0 & 0 & 0 & 0 \\ \Delta^0_{2,0} & \Delta^1_{2,1} & \Delta^2_{2,0}+\Delta^2_{2,1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta^3_{3,4} & \Delta^4_{3,4} & 0 & 0 \\ 0 & 0 & 0 & \Delta^3_{4,3} & \Delta^4_{4,3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Delta^5_{5,6} & \Delta^6_{5,6} \\ 0 & 0 & 0 & 0 & 0 & \Delta^5_{6,5} & \Delta^6_{6,5} \end{bmatrix}}_{\boldsymbol{\Delta}} \underbrace{\begin{bmatrix} q^0 \\ q^1 \\ q^2 \\ q^3 \\ q^4 \\ q^5 \\ q^6 \end{bmatrix}}_{\boldsymbol{q}} + \boldsymbol{\Theta}'',$$

where

$$\boldsymbol{\Theta}'' = \begin{bmatrix} \mathcal{Q}(q^0_\mathcal{T}, q^1_\mathcal{T}) + \mathcal{Q}(q^0_\mathcal{T}, q^2_\mathcal{T}) - \overline{\Delta}_{0,1}(\Delta^0_{0,1}+\Delta^1_{0,1}) - \overline{\Delta}_{0,2}(\Delta^0_{0,2}+\Delta^2_{0,2}) \\ \mathcal{Q}(q^1_\mathcal{T}, q^0_\mathcal{T}) + \mathcal{Q}(q^1_\mathcal{T}, q^2_\mathcal{T}) - \overline{\Delta}_{0,1}(\Delta^1_{1,0}+\Delta^0_{1,0}) - \overline{\Delta}_{1,2}(\Delta^1_{1,2}+\Delta^2_{1,2}) \\ \mathcal{Q}(q^2_\mathcal{T}, q^0_\mathcal{T}) + \mathcal{Q}(q^2_\mathcal{T}, q^1_\mathcal{T}) - \overline{\Delta}_{0,2}(\Delta^0_{2,0}+\Delta^2_{2,0}) - \overline{\Delta}_{1,2}(\Delta^1_{2,1}+\Delta^2_{2,1}) \\ \mathcal{Q}(q^3_\mathcal{T}, q^4_\mathcal{T}) - \overline{\Delta}_{3,4}(\Delta^3_{3,4}+\Delta^4_{3,4}) \\ \mathcal{Q}(q^4_\mathcal{T}, q^3_\mathcal{T}) - \overline{\Delta}_{3,4}(\Delta^3_{4,3}+\Delta^4_{4,3}) \\ \mathcal{Q}(q^5_\mathcal{T}, q^6_\mathcal{T}) - \overline{\Delta}_{5,6}(\Delta^5_{5,6}+\Delta^6_{5,6}) \\ \mathcal{Q}(q^6_\mathcal{T}, q^5_\mathcal{T}) - \overline{\Delta}_{6,5}(\Delta^5_{6,5}+\Delta^6_{6,5}) \end{bmatrix}.$$

We replace $\boldsymbol{\Omega}$ in Eq. (5.20) by its linear approximation $\boldsymbol{\Omega}^\star$.

$$\begin{aligned}
\nabla\xi(\boldsymbol{q}) =&\boldsymbol{\Lambda q} + 2\omega\boldsymbol{\lambda}_2\boldsymbol{\Omega} + \boldsymbol{\Theta}' = \boldsymbol{0} \\
\approx&\boldsymbol{\Lambda q} + 2\omega\boldsymbol{\lambda}_2\boldsymbol{\Omega}^\star + \boldsymbol{\Theta}' = \boldsymbol{0} \\
\approx&\boldsymbol{\Lambda q} + 2\omega\boldsymbol{\lambda}_2(\boldsymbol{\Delta q} + \boldsymbol{\Theta}'') + \boldsymbol{\Theta}' = \boldsymbol{0} \\
\approx&\boldsymbol{\Lambda q} + \underbrace{2\omega\boldsymbol{\lambda}_2\boldsymbol{\Delta}}_{\boldsymbol{\Gamma}}\,\boldsymbol{q} + \underbrace{2\omega\boldsymbol{\lambda}_2\boldsymbol{\Theta}'' + \boldsymbol{\Theta}'}_{\boldsymbol{\Theta}} = \boldsymbol{0}
\end{aligned} \tag{5.26}$$

Thus, we end up with

$$(\boldsymbol{\Lambda} + \boldsymbol{\Gamma})\boldsymbol{q} + \boldsymbol{\Theta} = \boldsymbol{0}. \tag{5.27}$$

Now, we can analyze the solvability of the local completeness problem based on Eq. (5.27), called *the completeness equation*. For a system of astrobots to be complete, this equation has be solvable, and its solution has to be the target points corresponding to the astrobots of the system. In particular, a system is complete if the following equation holds

$$\boldsymbol{q}_{\mathcal{T}} = -(\boldsymbol{\Lambda} + \boldsymbol{\Gamma})^{-1}\boldsymbol{\Theta}; \tag{5.28}$$

otherwise, it is incomplete.

The completeness equation asserts that the completeness of a system of astrobots depends on the parameters that are set by designers. Thus, modification of those parameters may resolve any potential incompleteness. For this purpose, in the next section we propose a parameter modification process to search for completeness encountering an incomplete system.

### 5.4.2 Completeness Seeking by Parameter Modification

As the completeness equation implies, the parameters which shape $\boldsymbol{\Lambda}$ and $\boldsymbol{\Gamma}$ directly influence on the completeness of system. Strictly speaking, parameter selections may give rise to incompleteness. So, one can take two approaches into account to search for the parameters based on which the system is complete. Considering an incomplete system with respect to a particular parameter specification, we modify entries of $\boldsymbol{\Lambda}$ and/or $\boldsymbol{\Gamma}$ to search for the other parameter specifications based on which the system is complete.

Matrix $\boldsymbol{\Lambda}$ is structured by the attractive and the cooperative attractive terms of CAPFs. So, if a system is incomplete, one can change the values corresponding the weighting factors of the cited terms. So, the overall summation of the $\boldsymbol{\Lambda}$ and $\boldsymbol{\Gamma}$ might be invertible. Theoretically, there are infinitely many numbers which can be attributed to the weighting factors. So, there is no upper bound for the number of the possible parameter modifications corresponding to $\boldsymbol{\Lambda}$. However, practical requirements constrain the scope of valid selections. For example, large weighting factors strictly increase the velocity of astrobots. The resulting high velocities may damage their motors and increase the possibility of collision when the astrobots are very close to each other. Thus, a reasonable range for each weighting factor can be determined from which new values are selected to modify the current values.

Matrix $\boldsymbol{\Gamma}$ also contributes to the completeness (on incompleteness) of a system based on its parameters. Among all those parameters, the target positions extremely affect on the entries of the matrix. One may note that, the target positions are defined based on each observation. In particular, some algorithms are used to assign an object to each astrobot. For example, [Morales et al. 2011] handles the object-astrobot assignments such that the number of the observed objects is maximized. We note that changing the targets assigned to the astrobots ends up with a new matrix $\boldsymbol{\Gamma}$. So, such a parameter modification may resolve

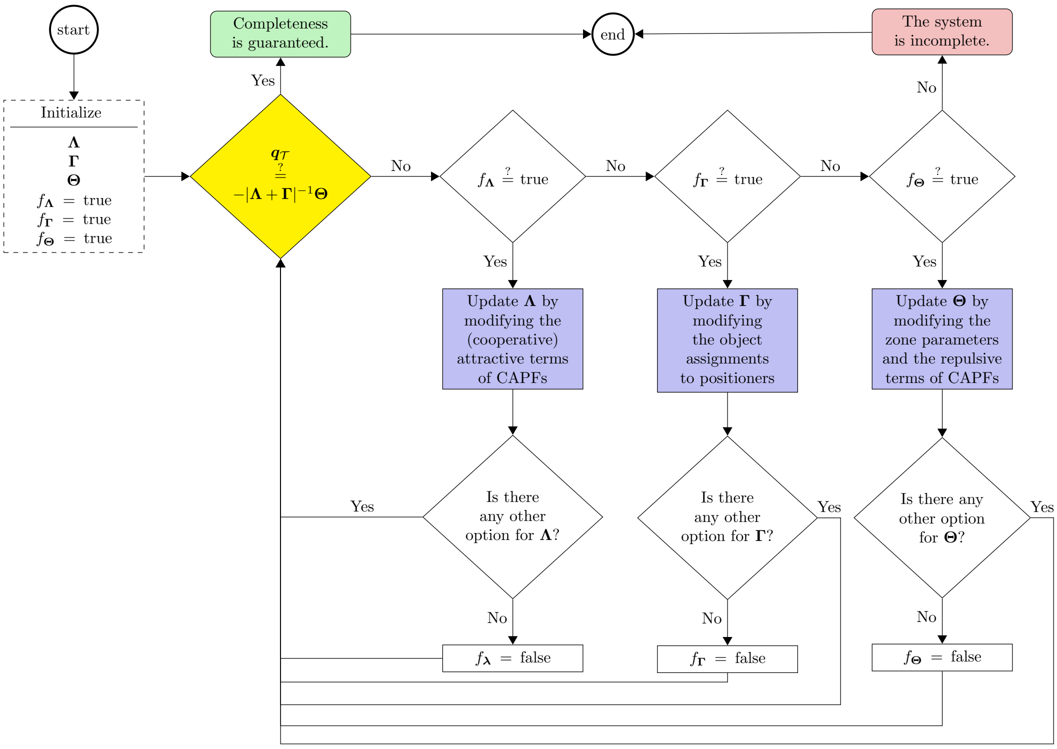

start
Completeness is guaranteed.
end
The system is incomplete.
Initialize
$\mathbf{\Lambda}$
$\mathbf{\Gamma}$
$\mathbf{\Theta}$
$f_{\Lambda} = $ true
$f_{\Gamma} = $ true
$f_{\Theta} = $ true
Yes
No
$q_{\mathcal{T}} \overset{?}{=} -|\mathbf{\Lambda}+\mathbf{\Gamma}|^{-1}\mathbf{\Theta}$
No
$f_{\Lambda} \overset{?}{=}$ true
No
$f_{\Gamma} \overset{?}{=}$ true
No
$f_{\Theta} \overset{?}{=}$ true
Yes
Yes
Yes
Update $\mathbf{\Lambda}$ by modifying the (cooperative) attractive terms of CAPFs
Update $\mathbf{\Gamma}$ by modifying the object assignments to positioners
Update $\mathbf{\Theta}$ by modifying the zone parameters and the repulsive terms of CAPFs
Yes
Is there any other option for $\mathbf{\Lambda}$?
Is there any other option for $\mathbf{\Gamma}$?
Yes
Is there any other option for $\mathbf{\Theta}$?
Yes
No
No
No
$f_{\lambda} = $ false
$f_{\Gamma} = $ false
$f_{\Theta} = $ false

Figure 5.4: The parameter modification process

the system incompleteness. In contrast to the $\mathbf{\Lambda}$ modification, the maximum number of the possible target position modifications is bounded. As already quoted, a specific procedure assigns a target to each astrobot according to a particular observation prior to the coordination. In particular, given $n$ objects corresponding to an observation and $m \geq n$ astrobots[1], the number of possible object-astrobot assignments is $P(m, n)$. However, every target cannot be observed by every astrobot because of the astrobots' motion limitations. Another option to modify $\mathbf{\Gamma}$ would be changing the value of the repulsive weighting factor, i.e., $\lambda_2$. However, manipulation of this factor is not recommended because of its critical role in the safety of the system and its performance. In particular, decreasing the factor may jeopardize the full control over movements of postioners when they are close to each other. In contrast, increasing the value of the factor can extremely increase the required time for completion of the coordination process. The explained parameter modification process is schematically illustrated in Figure 5.4.

## 5.5 Discussion

The computational complexity of the trajectory planning algorithm using the reference APF is $\mathcal{O}(n)$ where $n$ represents the number of the astrobots to be coordinated [Makarem et al. 2014]. The substitution of CAPF for APF does not adversely affect the computational complexity of the overall trajectory planning algorithm applied to astrobots sets. To be specific, the added cooperative attractive term is a polynomial similar to the attractive term of the algorithm. Thus, the linear-time computational complexity of the algorithm is preserved.

---

[1] We assume that an observation is planned such that all of its objects could be observed by the astrobots set. Thus, the number of the astrobots should essentially exceeds that of those objects.

Table 5.1: The convergence rate and the convergence time corresponding to test batch 1

| Swarm population (#) | | 52 | 106 | 234 | 449 | 730 | 980 |
|---|---|---|---|---|---|---|---|
| Converged astrobots (#) | APF | 50 | 90 | 196 | 382 | 621 | 844 |
| | CAPF | 52 | 106 | 234 | 449 | 730 | 980 |
| Convergence time (seconds) | APF | 14.8 | 36.3 | 89.7 | 173.2 | 317.6 | 386.9 |
| | CAPF | 16.7 | 49.3 | 96.1 | 199.8 | 359.4 | 503.5 |
| $\lambda_1$ | | 1 | 1 | 1 | 1 | 1 | 1 |
| $\lambda_2$ | | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| $\lambda_3$ (specific to CAPF) | | 0.03 | 0.03 | 0.03 | 0.03 | 0.04 | 0.04 |

The added cooperative attractive term increases the agility of the movements in the course of coordination. However, this agility has to be compensated and attenuated in practice because abrupt movements of astrobots may strengthen the collision possibility when they are close to each other. In other words, the added cooperative attractive term does not necessarily improve the convergence time of the coordination process. Furthermore, in some cases the convergence time might be even longer than that of corresponding to the reference APF. In the case of the reference APF, each astrobot stops moving upon reaching its target position. However, in the case of CAPF, an astrobot does not necessarily resides at its target spot immediately after reaching it because the cooperative term induces more dynamics to settle the maximum of the neighboring astrobots at their target points. Thus, an astrobot may temporarily pass its target to open the way for the remainder of its peers to get closer to their targets. This behavior does not give rise to endless oscillations since the high-level decision-making layer in fact handles these kind of scenarios. Thus, using CAPF rises a trade-off between the completeness seeking and potentially longer convergence time. The simulation results of the next section confirms this conclusion.

## 5.6 Results

### 5.6.1 Simulations

We modify the Python simulator developed in [Tao et al. 2018] according to our contributions. In particular, we substitute the reference APF [Tao et al. 2018] (see, Figure 5.1) with our CAPF (see, Figure 5.2).

We define two test batches. Each test batch includes six test scenarios each of which includes a specific number of astrobots as specified in the swarm population fields of Table 5.1 and 5.2. Furthermore, each test batch owns a specific set of initial and target points corresponding to its astrobots. The initial and target configurations are astrobots are either fully–folded or random[2]. The full specifications and the resulting number of the converged astrobots and the convergence times associated with test batch1 and 2 are illustrated in Table 5.1 and 5.2, respectively. The graphical representations of the convergence rates and the convergence times corresponding to test batch 1 and test batch 2 are illustrated in Figure 5.5 and 5.6, respectively.

We chose $\lambda_1 = 1$, $\lambda_2 = 0.05$, and $\lambda_3 = 0.03$ for our tests. However, these values do not fulfill the completeness condition corresponding to some of the test cases. So, we used the parameter modification procedure as explained in Section 5.4.2. In particular, the 5th and the 6th test cases of both the test batches cannot be completely coordinated by the quoted weighting factor parameters. These cases are highlighted

---

[2] This randomness is on the same footage as those of the circular uniform distributions in Chapter 2 and 3.

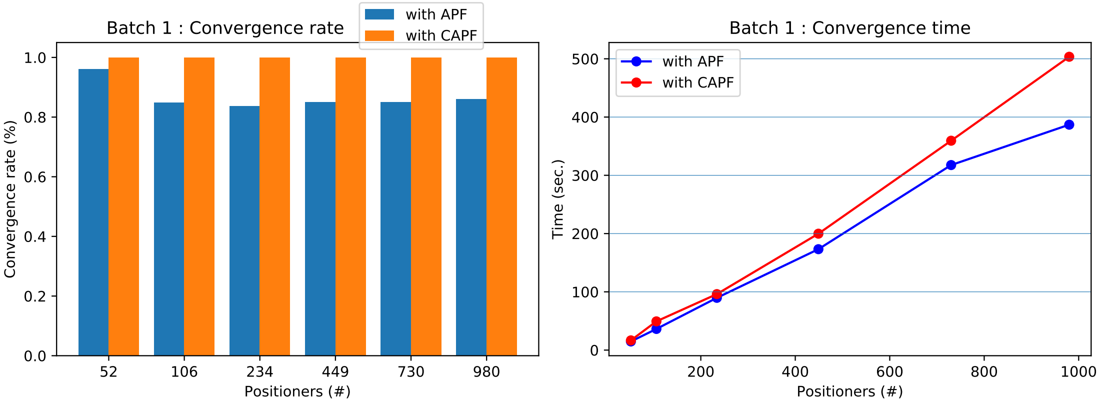


Figure 5.5: Visual illustrations of the convergence rate and the convergence time corresponding to test batch 1. (a) The convergence rate corresponding to test batch 1. (b) The convergence time corresponding to test batch 1.

in the last rows of Table 5.1 and 5.2. Thus, we modified $\lambda_3$ value which ended up with the complete coordination in those cases.

The results witness the completeness of the considered test cases using our cooperative navigator (see, Figure 5.2) which indicates the efficiency of our approach. As discussed in Section 5.5, the imposed necessity of completeness to the overall coordination process practically gives rise to longer movements and interactions between astrobots. So, the trade-off between the improved convergence rate and the longer convergence time leads to the following conclusion: the available time between two consecutive observations may be shorter than the required time for the complete coordination of astrobots. In this case, one has to use the competitive navigator (see, Figure 5.1).

## 5.6.2 Experimental Validation

### 5.6.2.1 Setup

Our hardware setup includes a miniature plate into which 19 astrobots are mounted as rendered in Figure 5.7. The relative distances between astrobots on this plate resemble those corresponding to the real focal planes of both SDSS-V telescopes. Parameters characterization of astrobots used in our experimental tests are described in Table 5.3. The sampling step can be manipulated as a degree of freedom in command generation. Smaller step sizes may provide more smooth and accurate motions specially in the maneuvers in which astrobots are so close to each other. On the other hand, very small step sizes increase the size of the

Table 5.2: The convergence rate and the convergence time corresponding to test batch 2

| Swarm population (#) | | 54 | 114 | 250 | 481 | 773 | 1006 |
|---|---|---|---|---|---|---|---|
| Converged astrobots (#) | APF | 52 | 105 | 228 | 434 | 675 | 889 |
| | CAPF | 54 | 114 | 250 | 481 | 473 | 1006 |
| Convergence time (seconds) | APF | 13.9 | 31.3 | 85.1 | 171.2 | 267 | 364.6 |
| | CAPF | 14.7 | 41.8 | 99.7 | 194 | 303.6 | 547.9 |
| $\lambda_1$ | | 1 | 1 | 1 | 1 | 1 | 1 |
| $\lambda_2$ | | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| $\lambda_3$ (specific to CAPF) | | 0.03 | 0.03 | 0.03 | 0.03 | 0.04 | 0.04 |

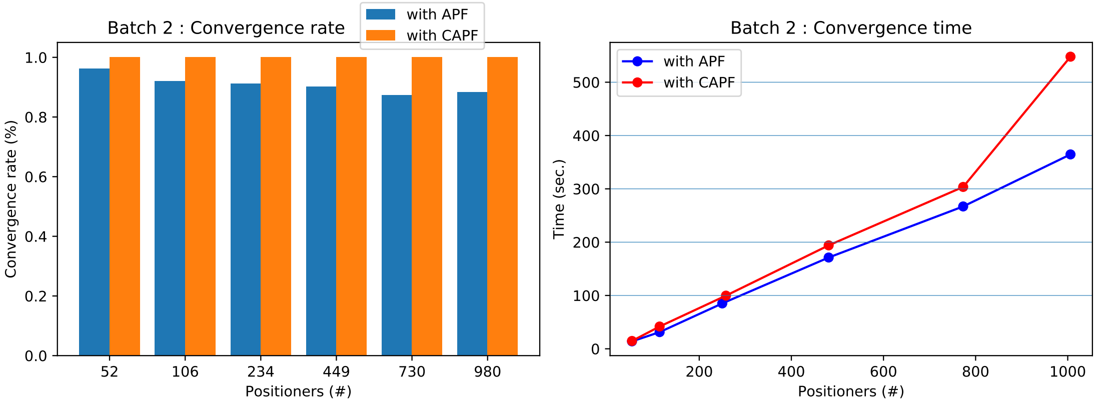


Figure 5.6: Visual illustrations of the convergence rate and the convergence time corresponding to test batch 2. (a) The convergence rate corresponding to test batch 2. (b) The convergence time corresponding to test batch 2.

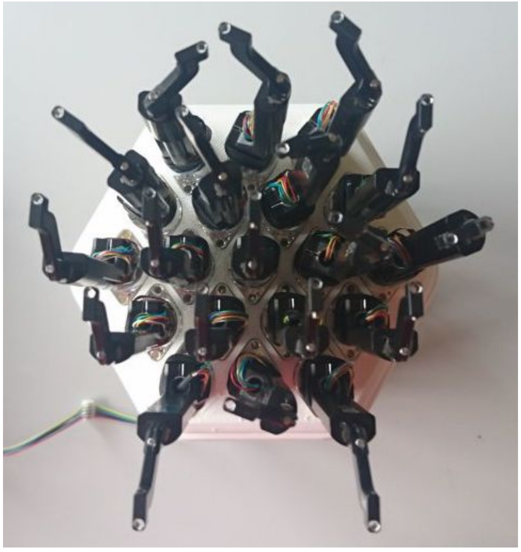


Figure 5.7: The 19-astrobot setup of the applied experimental tests

command file by many redundant entries. So, this trade-off has to be managed by trial and error according to the cardinality of swarms of astrobots and their pitch. The specification of CAPF parameters may differ from one coordination scenario to another because the completeness condition requires different setting of parameters to be fulfilled. Thus, we elaborate on various settings of parameters and their impacts on the convergence of the swarm in Section 7.4. The trajectories are generated by CAPF, running on an instrument control system (ICS). The result is a YAML[3] file representing two arrays of velocity data per astrobot each of which is associated with the joint of one of the arms. The file generated by the coordinator includes two arrays of positional data each of which corresponding to the extrapolated positions of one of the arms of each astrobot. The time step from one entry two other one is 0.25s. So, the dynamics of motions are based on variable speeds. Such variations do not jeopardize the safety of the system because of the collision-free formalism of CAPF. The format of the entries of the YAML associated with astrobot $i$ reads as follows.

```
i :
```

---

[3] YAML, recursively standing for “YAML Ain’t Markup Language”, is a data-serialization language to configure files data storage and transfer applications.

```
alpha: [[alpha_1, 0.25*0], [alpha_2, 0.25*1], ..., [alpha_n, 0.25*(n-1)]]
beta: [[beta_1, 0.25*0], [beta_2, 0.25*1], ..., [beta_n, 0.25*(n-1)]]
```

Here, $n$ denotes the number of the steps, set by the coordinator, which yields to a desired coordination. A real YAML file including the trajectories corresponding to an astrobots is as depicted in Figure 9.2. This entry illustrates that the convergence time for the whole coordination was 12.25s. Another interesting point is that the coordinator indeed generates speed signals. However, since the time steps are fixed, we overall control position of astrobots.

The software architecture of the simulator coordinator is described in the appendix. The configuration file associated with that specification is transferred from the ICS to a communication hub via an Ethernet cable. The communication hub sends the trajectories to a bridge from which a CAN cable feeds the trajectories to astrobots. Figure 5.8 depicts the described interconnections. In particular, the nominal voltage and current required by the whole pack of astrobots are 23V and 3A, respectively, as the power supply provides them to the astrobots. We generate trajectories of the astrobots on a PC running Linux Manjaru. Then, we use the async-based Jaeger framework [S. Consortium n.d.] to transfer the trajectories to astrobots via a CAN bus.

In Section 5.2, we noted that the completeness checking has be to done in a localized manner. The cardinality of astrobots in a neighborhood plays an important role in this analysis. In particular, full neighborhoods, which include 7 astrobots, are more prone to partial convergences compared to incomplete neighborhoods, i.e., those which are formed by less astrobots. So, we planned a radial placement of 19 astrobots as depicted in Figure 5.9. Each neighborhood is identified by its central astrobot. For example, the left-hand-side (full) neighborhood of Figure 6.3a refers to that of astrobot #10, while the (incomplete) right-hand-side one corresponds to astrobot #8.

We conduct our tests according to a real galaxy catalog. Namely, we take various partitions of the

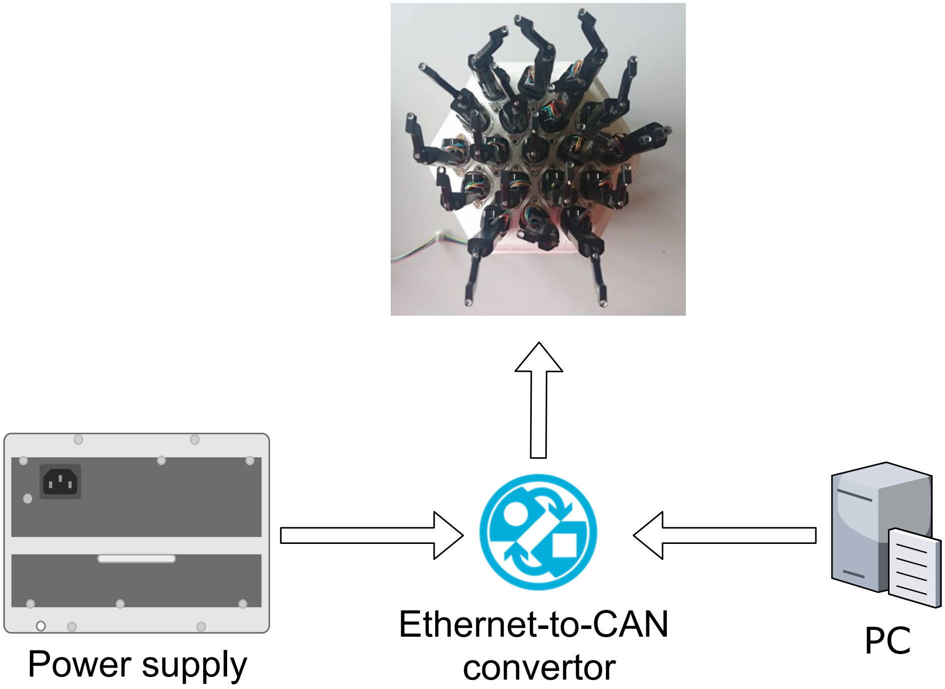
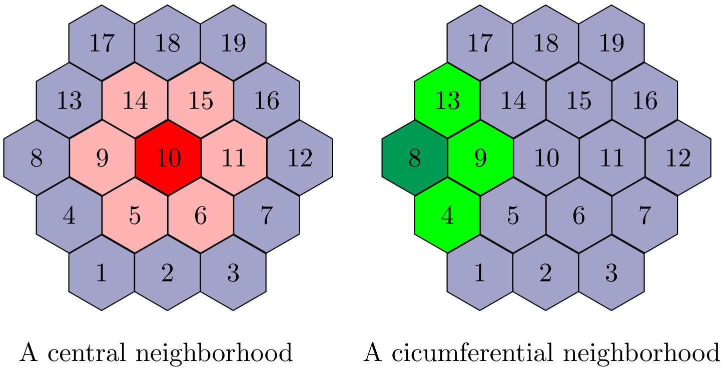


Figure 5.8: The setup of the hardware interconnections in our experimental tests



A central neighborhood A cicumferential neighborhood

Figure 5.9: The neighborhood types in the experimental focal plane. In a massive focal plane, the majority of neighborhoods are central ones. Thus, the completeness condition, i.e., Eq. (5.27), was obtained based on this critical case in [Macktoobian et al. 2019a].

eBOSS large-scale structure LRG catalog (data release 14) [Bautista et al. 2018][4]. In the experimental tests, we randomly select the targets, from the data entries of the catalog, whose projections are in the area of the test plane. The targets are assigned to astrobots using the optimal target assignment method [Macktoobian et al. 2020a]. So, we obtain the best possible performance in terms of the minimum effort for coordination and the maximum distribution of targets among astrobots to minimize the potential deadlock/collision situations. [Macktoobian et al. 2019a] reported extensive simulation results associated with astrobot swarms whose cardinality are comparable to those of the SDSS-V telescopes. In this paper, we expand those numerical simulations to even larger focal planes to illustrate the efficiency of CAPF in safe completeness seeking in massive swarms of astrobots.

#### 5.6.2.2 Parameters Impacts on Complete Coordination

In this section, we report the robustness of CAPF in achieving completeness. If a particular setting of parameters do not end up with completeness, the theory requires the modification of those parameters to yield completeness. The intrinsic robustness of CAPF in achieving completeness is investigated through 1000 tests applied to the 19-astrobot bench. Accordingly, Figure 5.10 indicates that CAPF reaches completeness in 97.4% of the applied tests without any parameter modification. In other words, the cooperative kernels of the CAPFs used in CAPF are efficiently sufficient to coordinate all of the astrobots given the initial configuration of the system and their targets. The results on the robustness notion may be extended to larger focal planes. Namely, Table 5.4 illustrates the simulated robustness results corresponding a class of massive focal planes. The first column of the modification rounds refers to the coordination which are complete, thereby needing no modifications. The large numbers of this column relative to the other columns per row exhibit the high performance of CAPF in achieving completeness without modifications in the majority of situations. In this regard, one observes that even in the case of these extremely complicated focal planes, ∼97% of coordination scenarios are inherently complete. The completeness condition, i.e., (5.27), is derived based on the local linearization of astrobot's motions in neighborhoods. Thus, the condition indeed approximates the completeness in a particular neighborhood. The quoted results echoes the efficiency of CAPF in view of robustness in this viewpoint, as well. In particular, one observes that the applied linear approximations do not severely undermine the coordination quality in terms of the required

---

[4] The data model can be found at https://www.sdss.org/dr14/data_access/value-added-catalogs/?vac_id=eboss-large-scale-structure-lrg-catalogs-dr14

Table 5.3: Parameters of astrobots in the performed tests

| Parameters | Values |
|---|---|
| pack cardinality | 19 |
| length of first arm | 7.4 mm |
| length of second arm | 15 mm |
| rotational step size | 0.1° |
| temporal step size | 0.25 s |
| pitch | 22.4 mm |

modification rounds. To specifically study the impact of parameter variation in completeness seeking, we first note that the parameter $\mathbf{\Lambda}$ is exclusively a function of attractive weight $\lambda_1$ and cooperative attractive term $\lambda_3$. On the other hand, $\mathbf{\Gamma}$ and $\mathbf{\Theta}$ are extremely non-linear parameters which also include target positions. So, varying $\mathbf{\Lambda}$ may be preferred to $\mathbf{\Gamma}$ and $\mathbf{\Theta}$ because of its more intuitive definition. $\mathbf{\Lambda}$ modification can influence both safety and performance measures of a swarm. First, both constituents of $\mathbf{\Lambda}$ are attractive weights. So, increasing both of them may escalate the collision hazards because of the potential faster motions of astrobots' arms. Additionally, convergence time may inefficiently increase if one reduce both of these weights because the whole attractive dynamics of the swarm diminish. Thus, given a variation step $\delta\lambda > 0$, we modify $\mathbf{\Lambda}$ based on the following rules.

$$\begin{aligned} \lambda_1 &\leftarrow \lambda_1 - \delta\lambda \\ \lambda_3 &\leftarrow \lambda_3 + \delta\lambda \end{aligned} \tag{5.29}$$

This update profile preserves safety by decreasing the dominant weight factor, i.e., $\lambda_1$. The loss in the performance is also relatively compensated by increasing the submissive factor, say, $\lambda_3$. The smaller the update step $\delta\lambda$ is, the less safety is put in jeopardy. So, we report how effective the small variations of $\delta\lambda$ are in resolving incomplete coordination scenarios. According to Figure 5.10, only 26 scenarios were not inherently coordinated with respect to initial system's configuration. To resolve these issues, we define an

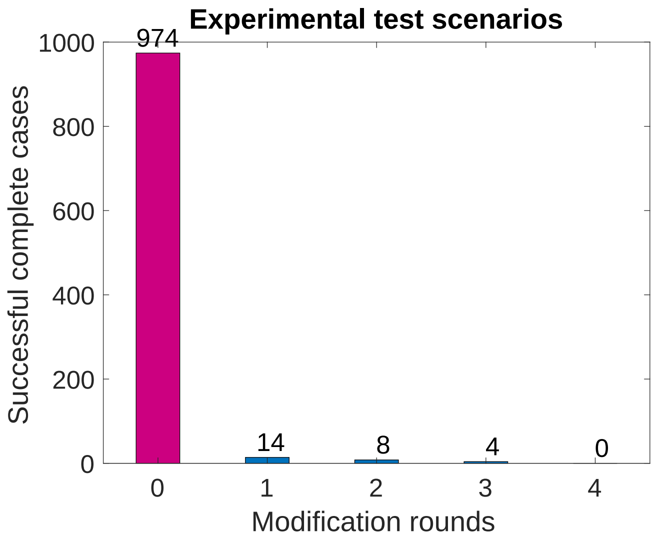


Figure 5.10: The number of the required modification rounds to reach completeness in 1000 experimental tests

step array $\overline{\delta\lambda}$ whose entries represent potential updates steps to be used.

$$\overline{\delta\lambda} := \{0.001, 0.005, 0.01, 0.05, 0.1\} \tag{5.30}$$

Since we are interested in the smallest possible variation which fixes an incompleteness, the array is sorted in ascending order. Then, we pick the entries and feed them into a problematic completeness condition. If that new parameter does not resolve the intended incompleteness, the next one is picked to be tested, and so on, to finally find a new parameter to meet the completeness condition. In Figure 5.11, we observe that 22 out of 25 incomplete cases of the 19-astrobot bench are resolved using the smallest entry of the step array, say, $\overline{\delta\lambda}[1] = 0.001$. The second bar of Figure 5.11 indicates 2 cases which were handled not by the first but by the second entry of the array. So, two modifications step have to be taken into account to first check $\overline{\delta\lambda}[1] = 0.001$ and then $\overline{\delta\lambda}[2] = 0.005$. Overall, there existed only one case in which the first two steps were not able to provide completeness but the third one. No incomplete scenario required larger steps, say, the two last entries of the array.

#### 5.6.2.3 Convergence Time

Coordination are in general conducted from one observation to another. For this purpose, there are two approaches to reconfigure astrobots. The first strategy directly coordinates them from the configuration of the latest observation to that of the upcoming one. This direct convergence is generally fast, but it may be fairly challenging in terms of collision avoidance. Another scheme is a two-phase coordination in which astrobots are first sent to their fully folded state in which $\theta = 0^\circ$ and $\phi = 180^\circ$. Then, they are coordinated to their target configuration. The advantage of this strategy is that astrobots may encounter less potential deadlock situations. However, this way of coordination is at the cost of longer times to reach final coordination. This idea also implies more fluctuations thereby requiring more energy. So, in long runs, astrobots may be more prone to amortization.

The convergence times of both methods associated with 1000 coordinated scenarios on the 19-astrobot bench are represented in Figure 5.12. One notes that the direct coordination under the control of CAPF are noticeably faster than those executed in the two-phase way. The completeness difference between the two is trivial, in that the two-phase strategy had only achieved six complete coordination more than the direct one. However, all these cases were compensated by only one modification round of parameters. The corresponding samples are signified using dark vertical lines in Figure 5.12. So, CAPF is efficient enough to simultaneously perform direct coordination and achieve high rates of completeness.

Table 5.4: The number of the required modification rounds to reach completeness in 1000 simulated tests

| Population | Modification rounds | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 |
| 500 | 490 | 8 | 2 | 0 | 0 | 0 |
| 1000 | 973 | 19 | 6 | 2 | 0 | 0 |
| 3000 | 2910 | 66 | 15 | 8 | 1 | 0 |
| 5000 | 4803 | 122 | 64 | 14 | 5 | 2 |

#### 5.6.2.4 Target Distribution Influence on Completeness

In previous sections, we used optimal target assignment [Macktoobian et al. 2020a] which supplies the maximum distribution of astrobots, i.e., their safety, and the minimum coordination, i.e., the minimum effort and time, required to arrange them in a desired coordination. In this section, we illustrate that quality of the coordination results of CAPF are even resilient to various target distributions. In the sections above, whenever we wanted to supply some targets to coordination computations, we picked a random subset of the targets available in the cluster galaxy catalog, presented by [Bautista et al. 2018], used in this study. The only selection condition was the reachability of each of those targets by at least one of the astrobots of our tests. In this section, however, we select targets in various unbalanced scenarios in some of which the targets may be distributed in non-uniform fashion over our focal plane. Intuitively, if targets are uniformly dispersed, it is more likely that each astrobot reaches more than one target. Thus, optimal target assignment may enjoy more flexibility in terms of matching astrobots to targets. In contrast, a more biased distribution of targets may degrade the quality of the optimal target assignment, thereby negatively impact the coordination phase. Such biased distributions increase the density of targets in various spots of a focal plane. So, the question is whether CAPF may have difficulties to deal with the coordination of astrobots in such dense localities. The following results indeed investigate this question showing that the sensitivity of the CAPF performance to the target distribution is not noticeable.

We first define a uniformly distributed set of targets in polar coordinate system $(r_f, \theta_f)$ whose center is assumed to be located at the base of the central astrobot of a swarm, which is astrobot #8 in our experimental test bench.

$$\begin{aligned} &\theta \sim \mathcal{U}[-180^\circ; 180^\circ] \\ &r^2 \sim \mathcal{U}(0; r_{\max}^2) \end{aligned} \tag{5.31}$$

In the equations above, $\mathcal{U}$ denotes a uniform distribution generator, and $r_{\max}$ represents the radius of the focal plane which reads 44.8 mm in the case of our test bench.

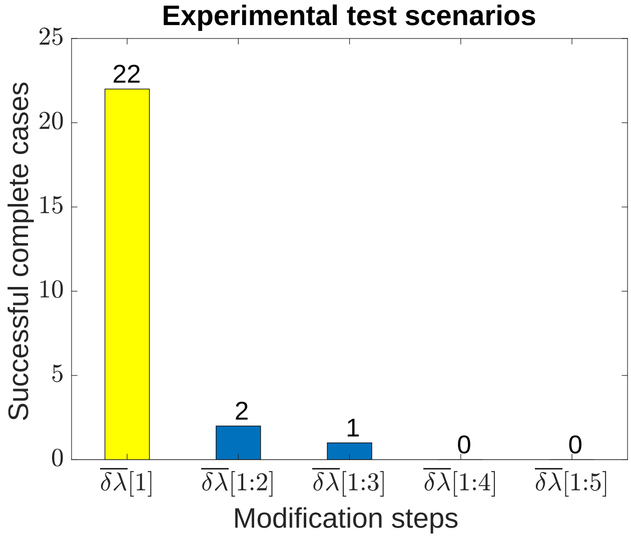


Figure 5.11: The number of the required modification steps to reach completeness in 1000 experimental tests. The symbol $\overline{\delta\lambda}[1:n]$ denotes that the entries 1 to $n-1$ of the step array $\overline{\delta\lambda}$ could not resolve an incompleteness but the $n$th entry does it. The first bar of this figure refers to $\overline{\delta\lambda}[1]$ stating that the majority of incomplete cases were simply resolved using the first entry of $\overline{\delta\lambda}$.

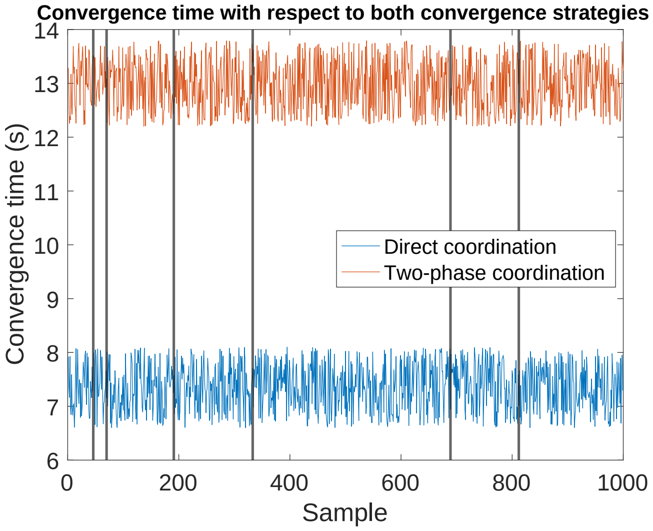


Figure 5.12: Convergence times of 1000 tests on the experimental setup with respect to both coordination strategies. The vertical black lines represent those scenarios which were complete in two-phase coordination. However, they were complicated enough to require one round of parameter modification to be also complete in direct coordination. Given the minority of these cases (6 occurrences) compared to the overall number of the tests, the direct coordination is the favorite approach.

We also take a bi-variate normal distribution into account such that the maximum concentration of targets are around the center of the swarm, and the distribution radially degrades while one moves toward the edge of the focal plane in any direction. The probability density function of this bi-variate normal distribution, in Cartesian coordinate system, is defined as

$$N(x,y) \sim \frac{1}{2\pi\sqrt{1-c^2}} \exp\left\{-\frac{1}{2(1-c^2)}\left[x^2 - 2cxy + y^2\right]\right\}, \tag{5.32}$$

in which variances are assumed to be 1, and the correlation coefficient $c$ equals 0.7.

These two distributions are applied to the galaxy catalog of targets as masks to filter those targets which are placed in the patterns similar to the intended distributions. After performing 1000 direct coordination scenarios for each of the uniform and bi-variate normal distributions, the convergence time results are obtained as depicted in Figure 5.13. In particular, the average coordination time corresponding to bi-variate normal selection is slightly longer than that of the coordination associated with the uniform selection. However, the performance of CAPF in the more crucial case (bi-variate normal selection) closely follows the coordination speed of the less-problematic one (uniform selection). Furthermore, there was only one coordination scenario which was not complete even after one round of parameter modification in the crucial case, whereas it was fully achieved in the uniform case. Thus, we conclude that CAPF is efficiently capable of yielding fast and safe coordination even in the case of biased selections of targets in view of their spatial positions.

Figures 5.14b-5.14e illustrate the steps of a coordination process, governed by CAPF scheme 5.14a, in which an initial amorph configuration is coordinated to a synchronized one[5].

---

[5] Recorded footage of the applied tests may be found at https://www.youtube.com/watch?v=MpXWvpz4h00.

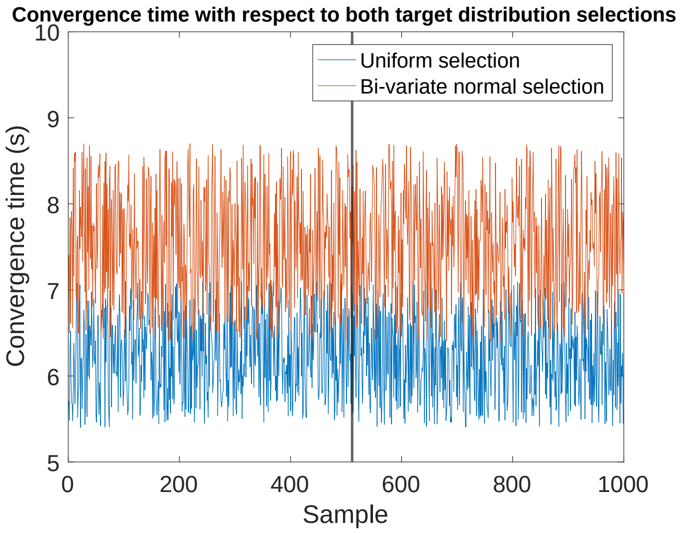
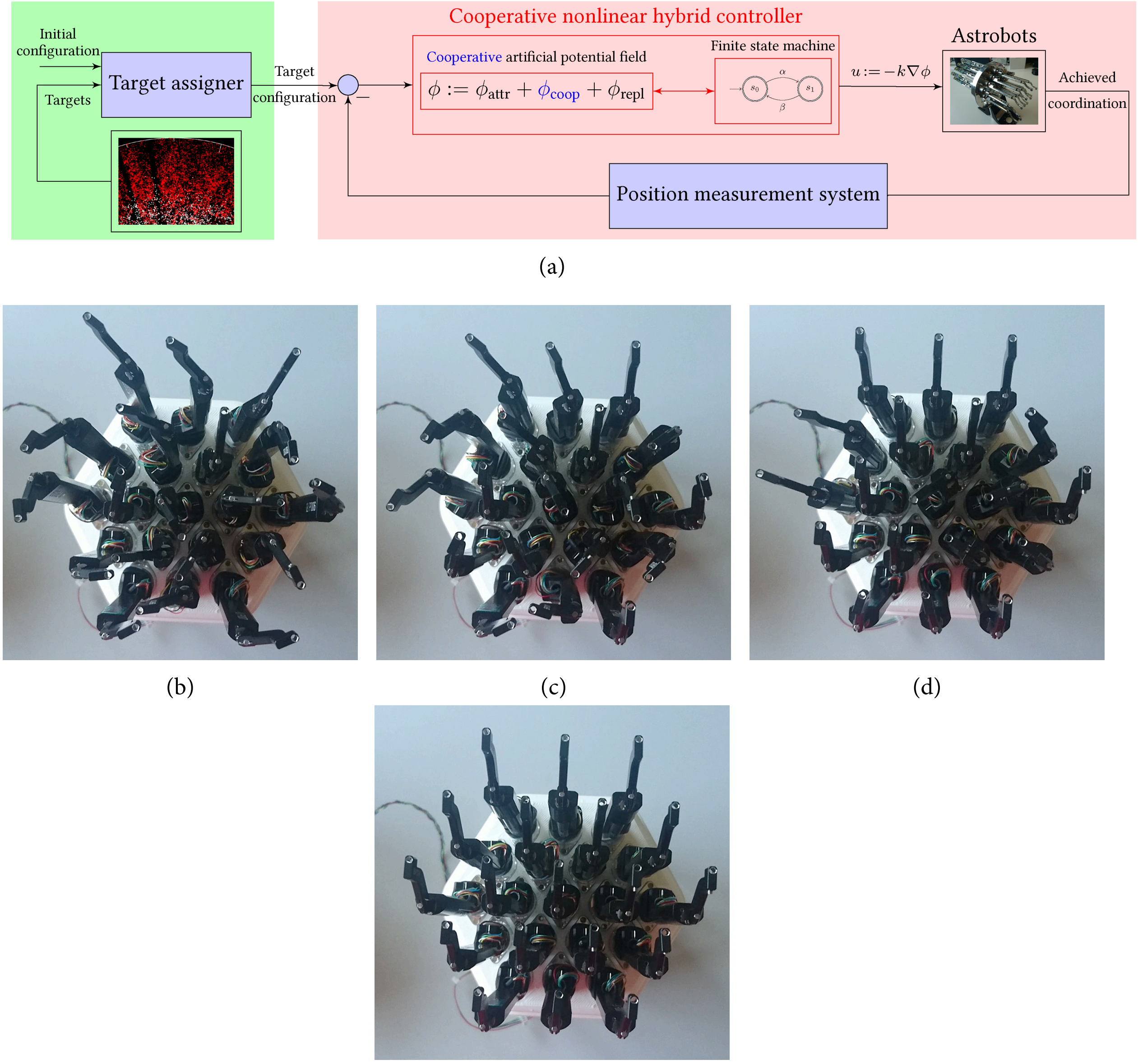


Figure 5.13: Convergence times of 1000 tests on the experimental setup with respect to both distribution selections. Uniform selections of targets are simpler to be handled by the optimal target assignment because of the availability of the more pairing options between astrobots and targets. The faster convergences of this selection class are shown in the figure. Nevertheless, the applied bi-variate normal selections' convergence times are noticeably comparable with those of the uniform selections. Given the fact that target assignments and coordination under bi-variate normal selections are more challenging, we conclude that CAPF is efficient under various potential distributions of targets.

## 5.7 Conclusion

This chapter studies the completeness problem corresponding to the coordination of robotic optical astrobots. In particular, we partition the complicated global completeness problem into a set of relatively simpler local completeness problems. We propose a new artificial potential field by which the completeness of an astrobot and its neighboring astrobots are cooperatively into account. Then, we find a completeness condition for the local completeness problem, and we show that the simultaneous fulfillment of all those conditions associated with a astrobots set in fact guarantees the global completeness of the overall system. We also present a completeness-seeking procedure to modify a system's parameters in case the system encounters an incomplete coordination. We obtain the complete coordination at the cost of longer coordination times compared to the required times using a common artificial potential field without cooperation mechanism.

The experimental results illustrated how the algorithm is efficient in achieving completeness in various settings of astrobots configurations and target assignments. We assessed the impact of parameter variations in resolving time-to-time incomplete scenarios by the minimum number of iterations and potential hazards to the safety of astrobot swarms. In view of convergence time, we observed that our algorithm can practically manage the safety through fast direct coordination. The robustness of our strategy considering various distributions of targets was also investigated. In particular, biased distributions of targets whose coordinates are not uniform often make a coordination critical for traditional planners. However, we validated the successful functionality of the proposed method in two uniform and bi-variate normal distributions.


Initial configuration
Targets
Target assigner
Target configuration
−
Cooperative nonlinear hybrid controller
Cooperative artificial potential field
$\phi := \phi_{\text{attr}} + \phi_{\text{coop}} + \phi_{\text{repl}}$
Finite state machine
$s_0$
$s_1$
$\alpha$
$\beta$
$u := -k\nabla\phi$
Astrobots
Achieved coordination
Position measurement system

(a)

(b) (c) (d)

(e)

Figure 5.14: Completeness seeking using CAPF strategy. (a) Control architecture. (b-e) A typical coordination process started from a random initial configuration to a synchronized one.

# Part C

# Completeness Prediction

## Chapter 6

# Constrained-Parity Convergence Prediction

Publication Note

The material presented in this chapter is adapted from:

- **Matin Macktoobian**, Francesco Basciani, Denis Gillet, and Jean-Paul Kneib, Data-Driven Convergence Prediction of Astrobot Swarms, (Preprint submitted to *IEEE Transactions on Automation Science and Engineering*). (Under Review)

## 6.1 Introduction

In the previous two chapters, we developed supervisory and cooperative coordination methods to achieve completeness in coordination of astrobots. We noted that the supervisory coordination may become inefficient to be used in the case of very crowded focal planes. The cooperative coordination also requires noticeable computational resources to check the completeness of each particular setting of astrobots and the projected positions of their assigned targets on their focal plane. In other words, this method, despite of its merit in completeness determination, is computationally so expensive that its real-time application may not be always feasible if the available times between successive coordination are too short. The convergence rate assessment of coordination may be done using numerical simulations of coordination with respect to various observation settings. This procedure is useful for small and medium surveys but not massive ones. Namely, convergence rate assessment requires the real-time solutions of hundreds to thousands of interdependent differential equations corresponding to distributed navigation functions of astrobots. Such analyses may not be feasible in the case of tight observation schedules in which the available times between observations (fairly less than ten minutes) are not long enough. In particular, the computation of trajectories for a massive astrobot swarm (e.g., including 6000 or beyond) takes at least one minute. However, the prediction of a machine learning model requires some hundredths of a second. Thus, convergence prediction is a promising approach to decrease the computational burden in simulating inefficient coordination scenarios in view of their convergence rates. If such assessment is possible, then inefficient coordination can be re-planned to those whose information throughput satisfy surveys expectations. In particular, a coordination output directly depends on the target-to-astrobot assignments corresponding to its observation. One may revisit an assignment to yield better coordination, thereby achieving higher convergence rates. Supervisory control was also employed to synthesize control commands whose safety and completeness can be formally verified [Macktoobian et al. 2019c]. However, this strategy also becomes inefficient because of the curse of dimensionality in the case of crowded astrobot swarm.

To alleviate the cited problem, we instead use machine learning to predict the convergence of astrobot swarm. Machine learning techniques have been partially contributed to the trajectory planning of multi-agent systems. For example in [M.-Y. Su 2011], an anomaly network traffic identification problem is studied for autonomous vehicles. This problem conceptually resembles the collision avoidance aspect of our prediction problem. In this method, the overall working space of the problem is so vast, yet the number of the number of vehicles are relatively small. So, collision avoidance is not a critical issue in the assumed sparse distribution of vehicles. In contrast, our convergence prediction problem indeed implies hazardous interactions in dense formations of astrobots, thereby entailing considerable risk of collisions between them. Additionally, our convergence prediction problem also features noticeable sensitivity to even trivial spatial deviations of configurations in terms of convergence results. On that account, any potential dataset representing our problem needs to encompass sufficient data to cover a wide range of similar configurations. A similar study takes the idea of moving ranges into account to assess neighbors more effectively for the vehicles in crowded urban areas [Lee et al. 2015]. The predictive model generated by this scheme relaxes the structured assumption by allowing movements of uncertain objects. The aforesaid relaxation, though, complicates the compliance with the safety requirements of this scenario. Instead, our problem enjoys the fully structured dynamics of astrobot swarm. Namely, the extremely constrained dynamics of each astrobot does not exert any uncertain feature to the prediction problem. As another example, learning-based strategies have been employed to predict trajectories of multi-agent systems in unconstrained or loosely constrained systems. For instance, route prediction for ships was investigated [Duca et al. 2017]. This study uses a variation of $k$-NN algorithm which exclusively models each ship as an isolated entity, say, in the absence of any collisions with other peers.

Limited applications of machine learning in trajectory prediction of more complex swarms are also reported. To give an instance, a class of aggregating behaviors in a self-organizing swarm were the subject of a prediction problem [Khaldi et al. 2018] using distance-weighted $k$-NN method [Jin 2019; Cataloluk et al. 2012]. The density metric of the swarm is modeled by hydrodynamical particle interpolation. This system seeks predictions through fairly complicated movements scenarios. However, the goal is the classification of collective behaviors while the involved non-interacting agents are subject to no collisions. Collision freeness was interestingly taken into account in a coordination scenario using artificial potential fields [J.-H. Chen et al. 2018]. This work is relatively comparable to what we seek in this paper, because the coordination control of astrobots is based on a class of artificial potential fields. However, the prediction application in this method is trivially concentrated on finding the closest point of an obstacle to a robot. Put differently, this strategy only guarantees collision freeness between a single robotic arm and a human's hand. Thus, in the absence of other agents, the complexity of this scenario is significantly less than what one encounters in the convergence prediction of astrobots.

The machine-learning-based behavioral predictions for multi-agent swarms have not been extensively studied. In particular, a learning system can efficiently train a model of a system if one feeds the data corresponding to all important features of that system. In the case of multi-agent swarms, these feature sets are often so large that final models may not be applicable for various reasons. First of all, training a predictive model requires enough data representing the behavioral patterns of system. The more complicated a system is, the more data of it one needs to effectively synthesize a predictor for it. The complexity of multi-agent swarms then requires huge datasets exhibiting their behaviors. But such amounts of data are often not available specially in the case of heterogeneous swarms. Moreover, a swarm system's functionalities are generally subject to many constraints whose presence may easily drive any learning model of that swarm toward common machine learning issues like underfitting and overfitting. Accordingly, the complete convergence of astrobots in the course of their coordination has not yet been efficiently resolved for the swarms including thousands of astrobots. On the other hand, partial coordination may lead to small convergence rates according to which the lack of enough data gives rise to the generation of the surveys whose wealth of information and details are not sufficient. Thus, instead of questing after analytical solutions to the completeness checking problem in more efficient ways, we shift our perspective to the prediction of complete coordination. In this framework, we seek to compute some models based on the data obtained from former coordination to predict the convergence rates of future ones in terms of some particular features. For this purpose, we propose a prediction algorithm based on the idea of weighted $k$-NN [Peterson 2009], given the relative simplicity and design intuitions which stems from the geometrical formulation of $k$-NN-driven strategies. Subject to a set of astrobots assigned to their targets, our method predicts whether or not each astrobot would successfully converge to its target spot. The applied evaluations to simulated results using our scheme exhibit high performances in those predictions.

As the first solution to this problem, we propose a weighted $k$-NN-based algorithm which requires the initial status of a swarm as well as its observational targets to predict its convergence. In this chapter, we solve the constrained version of the prediction problem in which the motion direction of all astrobots of a swarm, so-called parity, are fixed. This assumption makes the prediction more complicated as the astrobots dynamics are taken more restricted. However, as we later see, the results show how the cooperative coordination is able to completely coordinate many scenarios even in the presence of constrained actuator. Our algorithm learns to predict based on the coordination data obtained from previous coordination of the desired swarm. This method first generates a convergence probability for each astrobot based on a distance metric. Then, these probabilities are transformed to either a complete or an incomplete categorical result. One may note that a coordination process is a finite set of movements corresponding to each astrobot of a swarm with respect to many functional and safety requirements. A formal convergence verification tool has to check every single coordination step according to the control signals generated for each astrobot in every step. However, the discussion presented in the previous sections clarified that such exact approach to convergence analysis may be practically infeasible. Thus, among all steps of a coordination process, our algorithm merely works based on the first (i.e., initial) and the last (i.e., final) astrobots-targets configurations of the process. Another challenge raises from the imbalanced nature of the data in our problem. Namely, the convergence rate of large astrobot swarms generally varies between 65% $\sim$ 85% depending on their populations. Thus, the number of the astrobots which converge is noticeably larger than those which don't converge. So, the data are inherently imbalanced. It is widely observed that imbalance data may adversely impact the output of any naive machine learning algorithm which does not counteract against this issue. The applied simulations of our algorithm with respect to various populations of large astrobots manifest its effectiveness in terms of various performance measures. In particular, the method is applied to two typical swarms including 116 and 487 astrobots. It turns out that the correct prediction of successful coordination may be up to 80% of overall predictions. Thus, these results witness the efficient accuracy of our predictive convergence analysis strategy.

The remainder of the chapter is structured as follows. We the specify the convergence prediction problem in Section 6.2. We particularly focus on the features according to which a data-driven solution to the convergence prediction problem is indeed challenging. Section 6.3 comprises a weighted $k$-NN-based solution to the cited problem. We then present detailed statistical analysis to express the credibility of our algorithm in Section 6.4. We indeed apply our algorithm to two complex instances of astrobot swarms which include 116 and 487 astrobots. In the end, Section 6.5 reflects our conclusions and discusses potential search ideas to improve our results in future.

## 6.2 Problem Statement

The more astrobots converge to their target spots, the more the throughout of the observation associated with the targets will be. The current trajectory planners are not always able to achieve desired high convergence rates [Makarem et al. 2016]. If a convergence rate is below a certain threshold, then its corresponding final survey will not represent the expected quality. Thus, one has to assess the performance of a potential coordination process in terms of its final convergence before its execution. The analytical [Macktoobian et al. 2019a] and logical [Macktoobian et al. 2019c] tools to verify the results before their execution are often computationally too expensive. In this regard, these methods may not be used in real-time scenarios when the time slots available between observations are too short. The cited tools analyze every coordination step to check the collision freeness of motions which eventually tend to final configurations of astrobots. However, in this research, we only take the initial configuration of astrobots and the locations of their targets into account. We intend to predict whether or not a particular number of astrobots completely converged in the course of an observation surpasses the minimum number of desired convergences. Then, if the predicted convergence rate is larger than the minimum expectation, we decide to let the trajectory planner coordinate our swarm. Otherwise, we re-plan the unsatisfactory astrobot-to-target assignments to yield better combinations.

The problem statement is graphically shown in Figure 6.1 in which we seek the synthesis of a predictor to solve the problem. In particular, we prepare a dataset including many coordination scenarios with respect to multitude of astrobot-to-target assignment pairings which had been already simulated and/or executed. In this dataset, each astrobot in each pairing is labeled by 1 (resp., 0) if it finally reaches (resp., doesn't reach) its target. The overall set of this results is called *ground truth vector*. We use these data to predict convergence rates using a weighted $k$-NN-based strategy. Since the number of converging astrobots is often larger than that of those which doesn't converge, our data are inherently biased. Such imbalance data have to become balanced to make predictions reliable. We also only consider safe coordination scenarios in our dataset.

## 6.3 Convergence Prediction Strategy

In this section, we elaborate on our convergence prediction algorithm, as shown in Figure 6.4. We first compensate the imbalanced data issue using a set of vector weights applied to our data. Then, a distance metric is defined to rank the astrobots neighborhoods with respect to a desired astrobot whose convergence is intended to be predicted. A prediction probability is computed associated with each astrobot. We then note that the prediction problem of each astrobot has to be essentially analyzed in its own neighborhood. Thus, we localize the analysis which is mathematically equivalent to a particular normalization

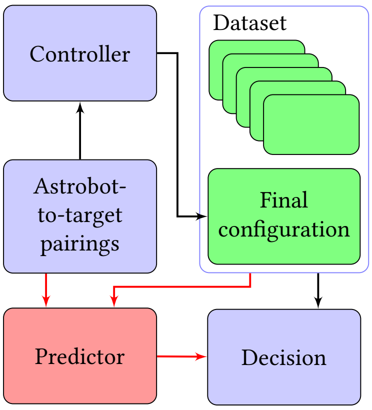


Figure 6.1: The schematic of the convergence prediction problem

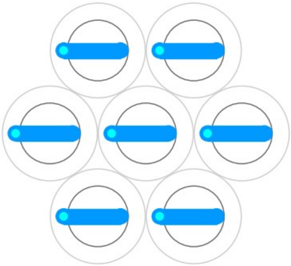


Figure 6.2: The folded formation of astrobots representing their initial configuration

of the quoted prediction probabilities. Next, given a particular *decision filter*, we transform the obtained probabilities to either of two categorical outcomes. Each of these outcomes represents the prediction of our algorithm regarding the successful or the unsuccessful convergence of their corresponding astrobots. We finally perform Monte Carlo cross-validation [Xu et al. 2001] to assess the reliability of the results of our algorithm.

One notes that the coordinate associated with each astrobot's initial configuration is fixed (see, Figure 6.2), and it does not impact the coordination phase. Thus, in the prediction process, we define the astrobot vector $\boldsymbol{\pi}$ according to the location of its projected target on the focal plane of the swarm as follows.

$$\boldsymbol{\pi} := \begin{bmatrix} x_t & y_t \end{bmatrix}^\mathsf{T} \tag{6.1}$$

Then, the *configuration matrix* $\boldsymbol{P}$ of a specific swarm $P$ including $n$ astrobots is indeed the accumulated configurations of its constituting astrobots which is

$$\boldsymbol{P} := \begin{bmatrix} x_{t^1} & x_{t^2} & \cdots & x_{t^n} \\ y_{t^1} & y_{t^2} & \cdots & y_{t^n} \end{bmatrix}^\mathsf{T}. \tag{6.2}$$

The ground truth vector corresponding to $\boldsymbol{P}$ is $\boldsymbol{g}^{\boldsymbol{P}}$. This vector represents the a posteriori information regarding the convergence of its corresponding configuration stored in a dataset. The more configurations exist in the dataset, the more representative the dataset is for its swarm. Since there are infinite number of configurations associated with a swarm, it is impossible to accumulate any possible coordination scenario in the dataset. However, the dataset has to be representative enough because changing the location of a target for just some tenths of millimeters just may change a successful convergence to a deadlock situation or vice versa. The dataset has to be divided into train and test partitions whose division proportion is discussed in Section 6.4.

### 6.3.1 Imbalanced Data Compensation

The family of $k$-NN algorithm is very sensitive to the local structure, i.e., the geometry, of data. We particularly enjoy this feature because the convergence prediction problem directly depends on geometrical characteristics of astrobot vectors. As already noted, configuration matrices often include many 1s compared to 0s because the majority of astrobots can be successfully coordinated using a swarm controller. So, their dataset is imbalanced according to which $k$-NN-based algorithms do not properly work [Dubey et al. 2013]. There are two typical approaches to resolving this issue neither of which is effectively applicable to our case. In particular, one may perform an oversampling (resp., undersampling) on the minority class (resp., majority class). This approach is infeasible in our case because an oversampling on the minority class requires the configurations whose ground truth vectors have more 0s than 1s. In the case of huge swarms, such configurations are extremely rare, if not nonexistent. Even if one could find such configurations, the next step would be the generation of a new group of targets which are very close to the targets of that configuration. But, it would be so likely that many 1s are also generated, thereby essentially canceling the purpose of oversampling. On the other hand, any undersampling needs to remove all the configurations whose ground truth vectors include more 1s than 0s. However, it gives rise to the loss of valuable information which are important for potential prediction cases.

Instead, we devise a vector of weights to enhance the impact of 0s in the ground truth vector of a specific configuration. This strategy is similar to the idea of class confidence weights [W. Liu et al. 2011]. The difference is that we apply the weights to single astrobots, not to data samples, i.e., configurations. Given a configuration $\boldsymbol{P}_i$ where $i \in \{1, 2, \cdots, N\}$, assume that ground truth vector $\boldsymbol{g}_i^{\boldsymbol{P}}$ is associated with it. We define *frequency vector* $\boldsymbol{m}$ and its complement, say, *pseudo vector* $\boldsymbol{v}$ as follows.

$$\begin{aligned} \boldsymbol{m} &:= \sum_i \boldsymbol{g}_i^{\boldsymbol{P}} \\ \boldsymbol{v} &:= N \cdot \mathbb{1}_{1\times N} - \boldsymbol{m} \end{aligned} \tag{6.3}$$

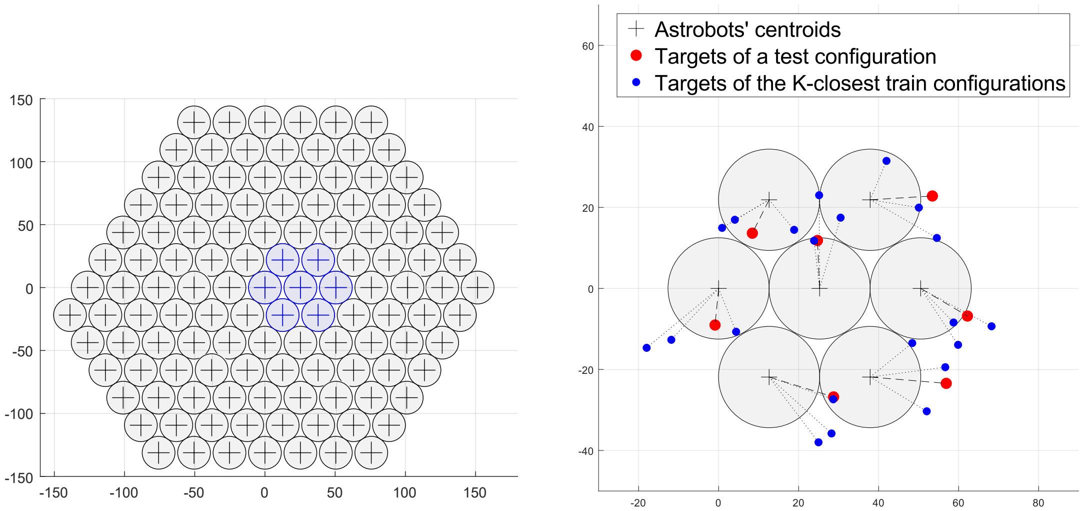
Then, the elements[1] of *weight vector* $\mathfrak{w} := \bigcup_i \mathfrak{w}_i$ read as below.

$$\mathfrak{w}_i := \begin{cases} m_i & \text{if } v_i = 0 \\ \dfrac{m_i}{v_i} & \text{otherwise} \end{cases} \tag{6.4}$$

Each element of $\mathfrak{w}$ has to be applied to the 0s of a particular astrobot of the configuration. We apply different weights to different astrobots because those which are in total neighborhoods, i.e., surrounded by 6 astrobots, generally don't reach their target positions as frequent as those which are in partial neighborhoods configuration. So, the 0s of the astrobots in total neighborhood configurations have smaller weights compared to those in partial neighborhoods. The notion of weight vector efficiently compensates the problem of imbalanced data. However in our problem, the two classes have not the same importance. In other words, we are more interested in the correct predictions of 1s rather than 0s in an operational point of view. So, we tune the elements of weight vectors according to our prediction requirements using two corrector coefficients $\alpha$ and $\beta$ on which we elaborate in Section 6.4.

---

[1] $n$-ary operator $\bigcup_i (\cdot)_i$ constructs a vector of the operator argument.



(a) A hypothetical neighborhood

(b) The localized probability computations within in a typical neighborhood

Figure 6.3: A typical probability localization scenario

### 6.3.2 Prediction Probability Computation

We define a distance metric to quantitatively compare various configurations with each other. Let $\boldsymbol{T}$ be a test configuration, say, the one we are interested in predicting its convergence. Let also $\boldsymbol{P}_i$ be a train configuration. We define *distance metric* $\Delta(\cdot,\cdot)$ which later is used to find the close train configurations to a particular test one as below[2]

$$\Delta(\boldsymbol{T},\boldsymbol{P}_i) := \sum_j \|\boldsymbol{T}_j - \boldsymbol{P}_{i,j}\|. \tag{6.5}$$

Here, $\boldsymbol{T}_j$ and $\boldsymbol{P}_{i,j}$ corresponds to the $j$th columns (i.e., astrobots) of $\boldsymbol{T}$ and $\boldsymbol{P}_i$, respectively.

Now, we select the *$k$-closest configurations set*, say, $\boldsymbol{P}^{\boldsymbol{T},k} \subset \boldsymbol{P}$, to $\boldsymbol{T}$. The specification of $k$ depends on the size of the train dataset and the complexity of the intended swarm. Namely, it must not be too small, otherwise there is some overfitting risk corresponding to the test configuration. On the other hand, if it is too large, one may take some train configurations into account which do not resemble the desired test one. So, it may lead to inaccurate predictions associated with some astrobots. Assume that function $\mathrm{sort}(\mathrm{set},\mathrm{metric})$ sorts its set argument with respect to its metric argument in ascending order. Moreover, fix function $\mathrm{fetch}(\mathrm{set},k)$ which returns the first $k$ elements of its sorted argument set. Then, given a particular $k$, $\boldsymbol{P}^{\boldsymbol{T},k}$ is defined as follows.

$$\boldsymbol{P}^{\boldsymbol{T},k} := \mathrm{fetch}\Big(\mathrm{sort}\big(\{\boldsymbol{P}_i \mid \boldsymbol{P}_i \in \boldsymbol{P}\},\Delta(\boldsymbol{T},\boldsymbol{P}_i)\big),k\Big) \tag{6.6}$$

Now, we use weight vector $\boldsymbol{w}$, $\boldsymbol{P}^{\boldsymbol{T},k}$, and its corresponding ground truth vector $\boldsymbol{g}^{\boldsymbol{P}^{\boldsymbol{T},k}} := \bigcup_i g_i^{\boldsymbol{P}^{\boldsymbol{T},k}}$ to compute the predictions corresponding to astrobots which converge to configuration $\boldsymbol{T}$. One notes that $\boldsymbol{w}$ has to be exclusively applied to the 0s in each element of ${}^{\mathfrak{w}}\boldsymbol{g}^{\boldsymbol{P}^{\boldsymbol{T},k}} := \bigcup_i {}^{\mathfrak{w}}g_i^{\boldsymbol{P}^{\boldsymbol{T},k}}$. Then, the result is *weighted ground truth vector* whose elements are defined as below.

$${}^{\mathfrak{w}}g_i^{\boldsymbol{P}^{\boldsymbol{T},k}} := \begin{cases} 1 & \text{if } g_i^{\boldsymbol{P}^{\boldsymbol{T},k}} = 1 \\ \mathfrak{w}_i & \text{otherwise} \end{cases} \tag{6.7}$$

---

[2] Unary operator $\|\cdot\|$ denotes the Euclidean norm of its vector argument.

Thus, *primary prediction probability vector* $\hat{\mathbf{\Gamma}}^{\boldsymbol{P},\boldsymbol{T}}$ with respect to test configuration $\boldsymbol{T}$ is given by

$$\hat{\mathbf{\Gamma}}^{\boldsymbol{P},\boldsymbol{T}} := \left(\sum_{i=1}^{k} \boldsymbol{g}_i^{\boldsymbol{P}^{\boldsymbol{T},k}}\right) \oslash \left(\sum_{i=1}^{k} {}^{\mathfrak{w}}\boldsymbol{g}_i^{\boldsymbol{P}^{\boldsymbol{T},k}}\right). \tag{6.8}$$

One may alternatively plan to apply different weights to each ground truth vector with respect to its distance metric from a particular test configuration. However, it increases the risk of overfitting.

If one deals with very large astrobot swarms, the distance metric $\Delta$ may not be reliable to assess the similarity between two configurations. In fact, once the number of astrobots extremely increases, there may be some astrobots among the closest train configuration whose targets are too far from their corresponding ones in the test configuration. This may be problematic even in the case of small swarms. In the next section, we mitigate this issue by localizing the derived prediction probability vector.

### 6.3.3 Prediction Probability Localization

The global neighborhood analysis of a large astrobot swarm is both inefficient and even problematic in view of the final results. In particular, large swarms geometrically encompass a massive number of neighborhoods. If one checks all available neighborhoods in the course of each lazy evaluation of the algorithm, then the solution would never be obtained after a reasonable amount of time. On the other hand, not all astrobots neighborhoods influence the coordination of a particular astrobot, but only those which are its immediate neighbors. Thus, we have to localize the probability computations of the algorithm. In particular, we perform a local analysis on the neighborhoods of each astrobot. Thereby, the algorithm is exclusively applied to a number of small configurations which includes a maximum number of 7 astrobots. By doing so, it would be less likely to have some astrobots the distances between whose targets and a test configuration are high. For example, Figure 6.3a depicts a neighborhood of astrobots the magnitudes of whose metric distances are reasonable as illustrated in Figure 6.3b.

Let $\boldsymbol{P}$ be a configuration of including $n$ astrobots. We define neighborhood matrix $\boldsymbol{\nu}_\pi \subset \boldsymbol{P}$ associated with a typical astrobot $\pi$ as the central entity of this neighborhood. The dimension of each instance of $\boldsymbol{\nu}_\pi$ is $2 \times r$, where $1 \leq r \leq 7$ denotes the number of the astrobots in the neighborhood. Thus, one has to overall perform $n$ local analyses. To do so, we introduce *counter vector* $\boldsymbol{\eta}$ whose dimension is $1 \times n$. Element $\eta_i$ of $\boldsymbol{\eta}$ corresponds to the number of the neighborhoods to which the $i$th astrobot of the swarm belongs. The elements of $\boldsymbol{\eta}$ are integers varying between 1 and 7. Thus, we yield *neighborhood probability vector* $\widetilde{\mathbf{\Gamma}}^{\boldsymbol{\nu}_\pi,\boldsymbol{T}}$ with respect to neighborhood $\boldsymbol{\nu}_\pi$ whose elements are defined as follows.

$$\widetilde{\Gamma}_i^{\boldsymbol{\nu}_\pi,\boldsymbol{T}} := \begin{cases} \hat{\Gamma}_i^{\boldsymbol{\nu}_\pi,\boldsymbol{T}} & \text{if } \pi_i \in \boldsymbol{\nu}_\pi \\ 0 & \text{otherwise} \end{cases} \tag{6.9}$$

Now, given, $\widetilde{\mathbf{\Gamma}}^{\boldsymbol{\nu}_\pi,\boldsymbol{T}} := \bigcup\limits_i \widetilde{\Gamma}_i^{\boldsymbol{\nu}_\pi,\boldsymbol{T}}$, *final probability vector* $\mathbf{\Gamma}^{\boldsymbol{P},\boldsymbol{T}}$ is computed as

$$\mathbf{\Gamma}^{\boldsymbol{P},\boldsymbol{T}} := \left[\sum_{\pi} \widetilde{\mathbf{\Gamma}}^{\boldsymbol{\nu}_\pi,\boldsymbol{T}}\right] \oslash \boldsymbol{\eta} \tag{6.10}$$

Finally, we need to transform these probabilistic entries to categorical 1s and 0s to represent successful or failed convergences, respectively. For this purpose, given a *decision filter* $q$, we define the elements of *output vector* $\boldsymbol{Y} := \bigcup\limits_i y_i$ as below.

$$y_i := \begin{cases} 1 & \text{if } \Gamma_i^{\boldsymbol{P},\boldsymbol{T}} > q \\ 0 & \text{otherwise} \end{cases} \tag{6.11}$$

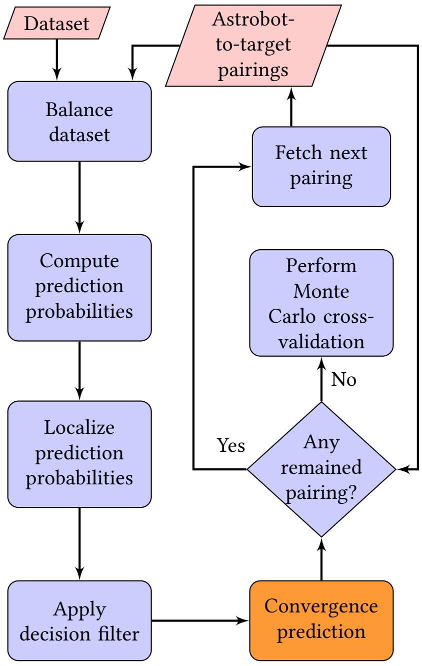

Dataset
Astrobot-to-target pairings
Balance dataset
Fetch next pairing
Compute prediction probabilities
Perform Monte Carlo cross-validation
No
Localize prediction probabilities
Yes
Any remained pairing?
Apply decision filter
Convergence prediction

Figure 6.4: The flow of the convergence prediction algorithm

The last phase of our convergence prediction algorithm performs Monte Carlo cross-validation to assess the credibility of the results. The rational behind preferring this method over $k$-fold cross-validation is the computational efficiency of the former. Namely, Monte Carlo cross validation enjoys a property that the proportional relation between train/test splits does not depend on the number of validation iterations. Thus, one can perform a series of iterations which are not linked to the dimensions of train and test datasets. The drawback of this method, though, is that some configurations may never be selected as test configurations, whereas others may be selected multiple times. For this reason, it is necessary to put a particular attention to the number of validation iterations. The choice of this number depends on how large a typical test dataset is compared to its corresponding train one. In other words, the smaller the test dataset is, the larger the number of iterations has to be.

## 6.4 Simulations and Results

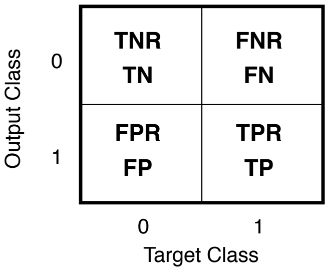
In this section[3], we illustrate how our algorithm efficiently predict the convergence of astrobots in massive swarms. We first define our evaluation measures and performance metrics. Then, we take two swarms into account each of which is constituted by 116 and 487 astrobots. The latter is particularly similar to the case of the astrobot swarm associated with the SDSS-V project [Kollmeier et al. 2017]. We also present some hints regarding the value selections for the algorithm's hyperparameters.

### 6.4.1 Performance Measures

Our performance measures are essentially defined based the following four notions.

---

[3] The datasets used in this section, associated with the initial positions of astrobots, their targets, and their coordination results, can be found in [Macktoobian n.d.]



|  |  |  |  |
|---|---|---|---|
| Output Class | 0 | **TNR**<br>**TN** | **FNR**<br>**FN** |
|  | 1 | **FPR**<br>**FP** | **TPR**<br>**TP** |
|  |  | 0 | 1 |
|  |  | Target Class |  |

Figure 6.5: The layout of the information represented by a typical confusion matrix

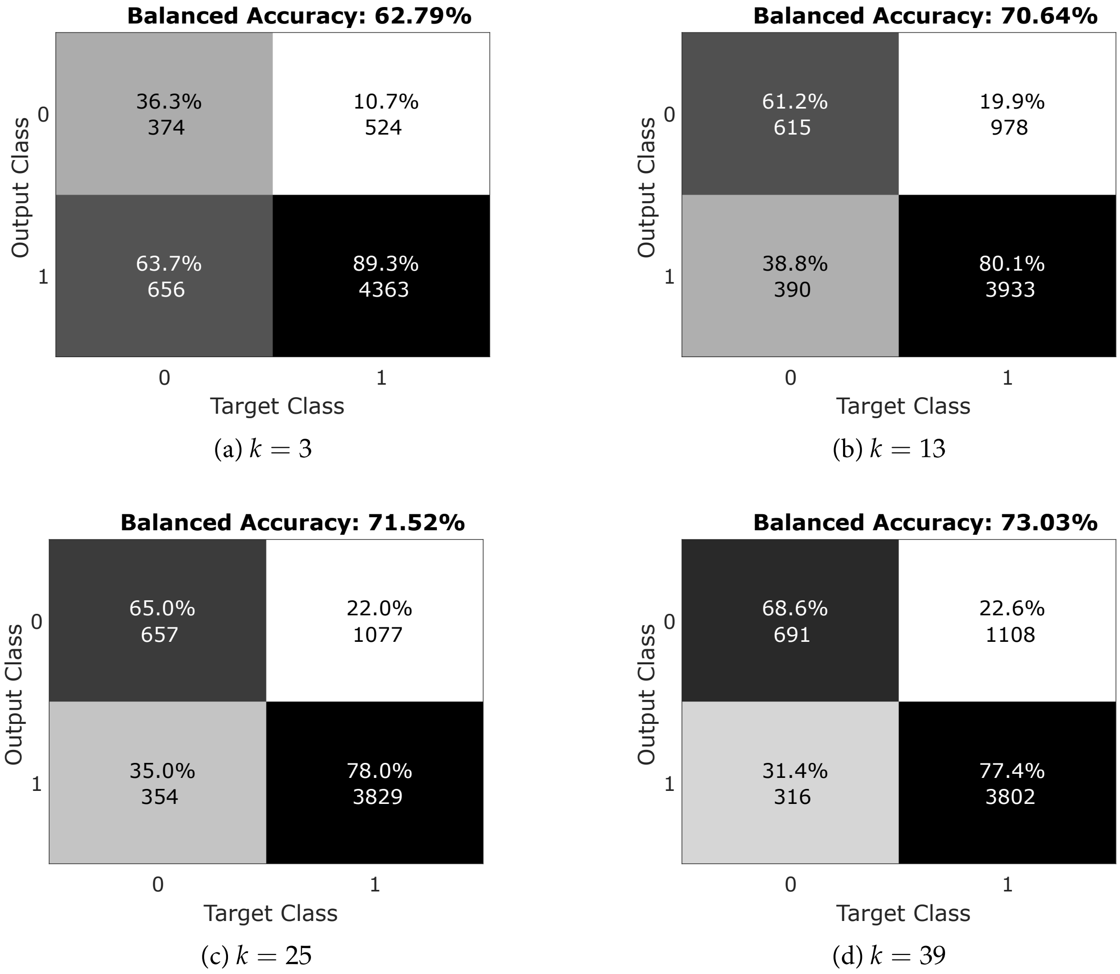
- A true positive (TP) is an astrobot which is predicted to converge (the predictor predicts 1), and it actually converges to its target position (its corresponding ground truth element is 1).
- A false positive (FP) is an astrobot which is predicted to converge (the predictor predicts 1), but it actually does not converge to its target position (its corresponding ground truth element is 0).
- A true negative (TN) is an astrobot which is not predicted to converge (the predictor predicts 0), and it actually does not converge to its target position (its corresponding ground truth element is 0)
- A false negative (FN) is an astrobot which is not predicted to converge (the predictor predicts 0), but it actually converges to its target position (its corresponding ground truth element is 1).

Accordingly, we take the standard rates of the above factors, i.e., TPR, FPR, TNR, and FNR, into account. These values are reported in based on Figure 6.5. If a predictor yields good performance, its corresponding confusion matrix has large values in its main-diagonal entries, indicating that the majority of samples have been correctly classified.

From an engineering point of view, we are more interested in the correct predictions of positives (the astrobots which converge to their target positions). It is because the information regarding the prediction of these astrobots would be crucial to decide whether or not a coordination process should be executed associated with a particular configuration of targets. On the other hand, the number of positives is much greater than that of negatives. If the predictor always predicts 1, the TPR would be perfect. But, the predictor does not indeed predict anything by completely neglecting 0s. So, we track the balanced accuracy measure established as the average of the TPR and the TNR to better assess the predictive essence of the algorithm. We also employ receiver operating characteristic (ROC) curves to illustrate the performance of our predictor in the course of varying one of its hyperparameters.

We also take precision and F1 score (harmonic mean) into the consideration. The precision measure is an index of how accurate the predictor is in predicting positives. Precision is an important measure to look at when FPs have significant impacts on our problem. We intend to maximize precision through minimization of FPs. F1 score indicates the trade-off between precision and TPR, say, recall. For example, if we increase the TPR, we indeed increase the number of predicted TPs. However, we also increase the number of FPs, thereby decreasing the precision. The bigger the F1 score is, the better the trade-off between precision and recall is.

We include corrector coefficients $\alpha$ and $\beta$, as well. These hyperparameters are used to manually tune the weight vector $\mathfrak{w}$ to obtain better accuracy rates with respect to positives and negatives. In particular, $\alpha$ tunes the $\mathfrak{w}_i$s of the astrobots in total neighborhoods, while $\beta$ does the same but for the astrobots residing in partial neighborhoods. In all simulations, we take the decision filter $q = 0.5$.



(a) $k = 3$

(b) $k = 13$

(c) $k = 25$

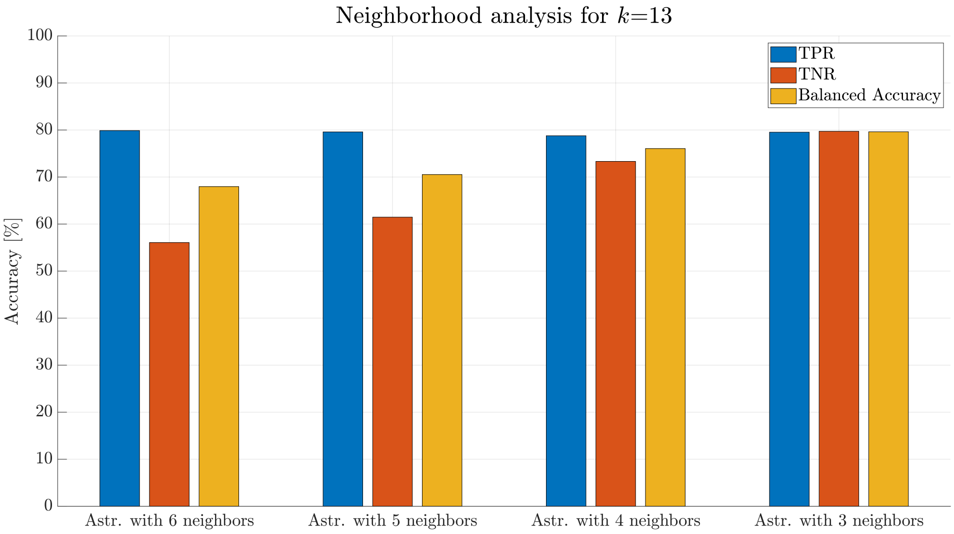
(d) $k = 39$

Figure 6.6: The confusion matrices of the 116-astrobot swarm

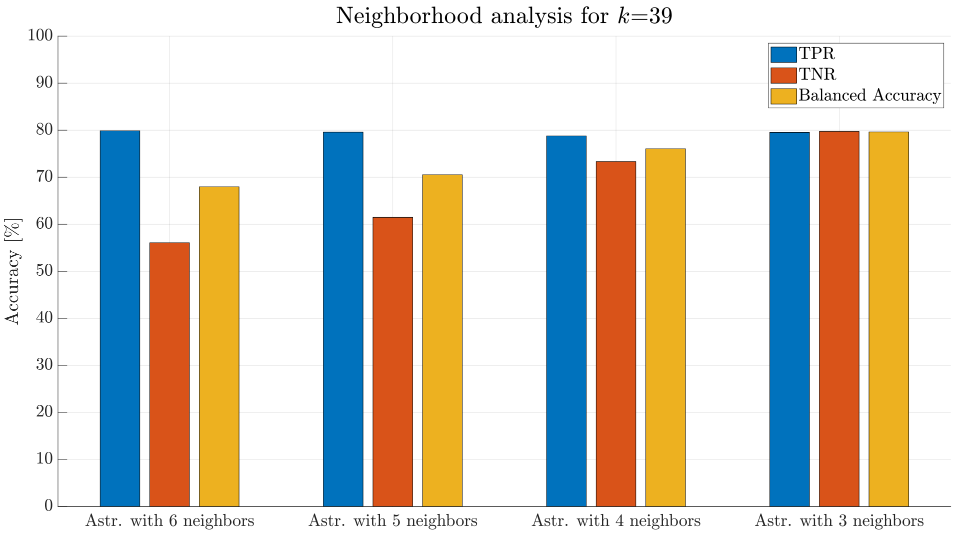
### 6.4.2 A swarm including 116 astrobots

Our complete dataset is composed of 10100 configurations, where the train and the test datasets include 10049 and 51 configurations, respectively. The algorithm iterations is set to 15. The confusion matrices corresponding to various values of $k$ are depicted in Figure 6.6. We observe that increasing $k$ increases and decreases the TNR and the TPR, respectively. It is reasonable since the more train configurations we take into account for the computation of the output, the higher the likelihood is to consider the train configurations whose astrobots don't converge. The selection of $k$ depends on how large the train dataset is. The larger the train dataset is, the larger $k$ may be. In this scenario, a proper $k$ may be chosen in the range of 10 to 50. If we increase $k$ too much, the information about the targets locations of the closest train configurations are no longer reliable.

It is also interesting to assess our performance indices for single astrobots. In particular, we obtain the TPR, the TNR, and the balanced accuracy, on the basis of the number of each astrobot's neighbors. To do so, we take the average of the performances of the astrobots with a specific number of neighbors, as rendered in Figure 6.7, where both corrector coefficients are 1. Namely, Figure 6.7a indicates that the prediction accuracy bottleneck refers to the astrobots in total neighborhoods. Figure 6.7b illustrates how the the balanced accuracy is improved in total neighbourhoods. On the other hand, the astrobots of partial neighborhoods experience the decrement and the increment of the TPR and the TNR, respectively.



(a) $k = 13$

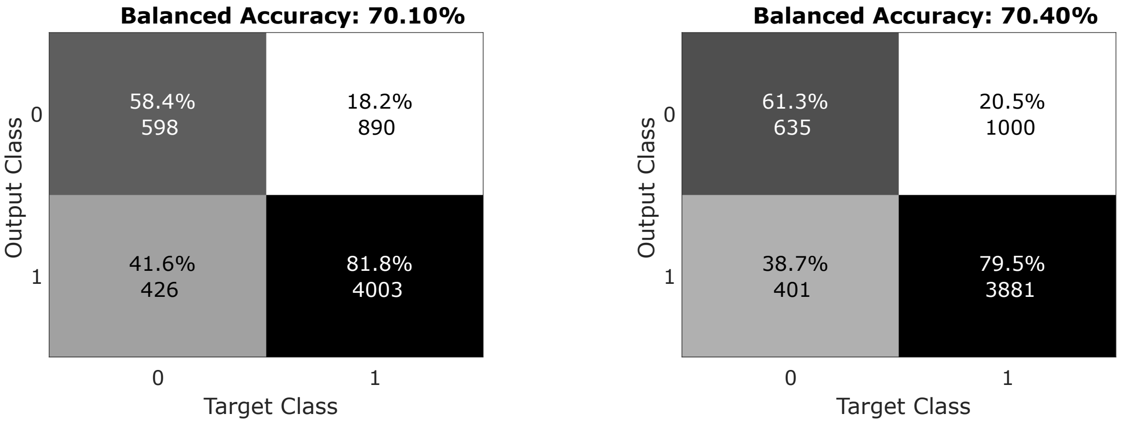


(b) $k = 39$

Figure 6.7: The neighborhood analyses of the 116-astrobot swarm corresponding to two neighborhood selections

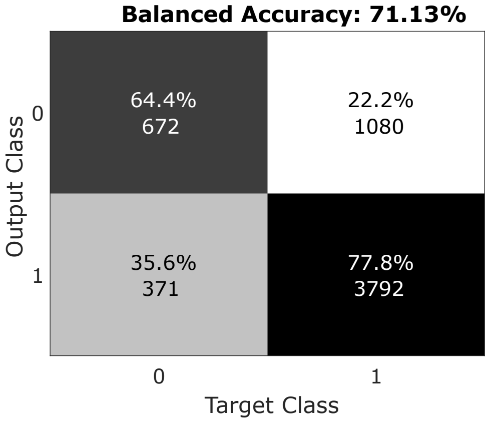
Corrector coefficients are expected to impact the qualities of the cases in which total neighborhoods are fairly abundant. In Figure 6.8, the confusion matrices of three different predictions are reported in which corrector coefficients are varied. In this case, we simply keep their values the same to show the overall effect of magnifying the weights of 0s. In all of the cases, we have $k = 13$. In particular, it is evident that increasing the corrector coefficients leads to the increment and the decrement of TNR and TPR, respectively, which is a direct effect of increasing the weight of the minority class.

To tune the hyperparameters, one may find Figure 6.9 very useful. Figure 6.9a illustrates that any $k > 21$ is stable. Specially, $k = 39$ realizes the best predictive performance for this swarm. One may note that the right choice of $k$ directly depends on what factor is the main goal of the prediction to be improved. For example, if we one would like to increase the balanced accuracy as much as possible, yet allow the TPR to drop under 80%, then $k = 39$ seems to be the best choice. But, if the TPR has to be over 80% with the maximum balanced accuracy, one may pick $k = 13$. The dynamical trends of the TPR, the TNR,



(a) $\alpha = \beta = 0.95$

(b) $\alpha = \beta = 1.00$



(c) $\alpha = \beta = 1.05$

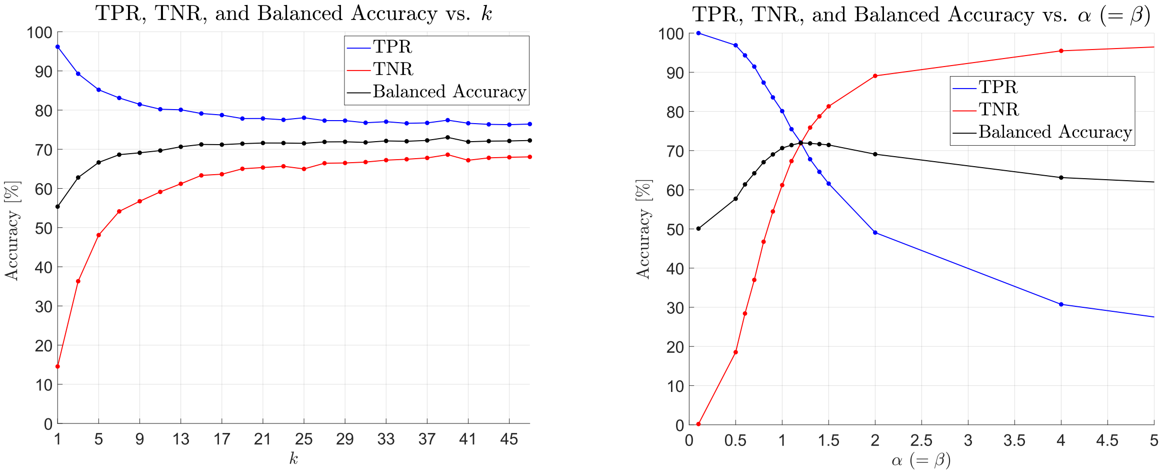
Figure 6.8: The evolution of the confusion matrices of the 116-astrobot swarm given different corrector factors

and balanced accuracy are also evident in Figure 6.9b in the course varying the two corrector coefficients while fixing $k = 13$. So, since we are more interested in the correct predictions of the positives, $k$ may be chosen large to increase the TPR as much as possible, while assuring that the balanced accuracy does not drop below a certain threshold. Moreover, Figure 6.9c shows the trends of the precision, the recall and the F1 score for different values of $k$. Table 6.1 reflects the best results in the convergence predictions of the 116-astrobot swarm. Finally, we can look at the ROC curve which visualizes the performance of our predictor. Every point on the ROC curve represents the result of a prediction experiment using a different value of $\alpha(= \beta)$ as shown in Figure 6.10. Here, we have $k = 13$.

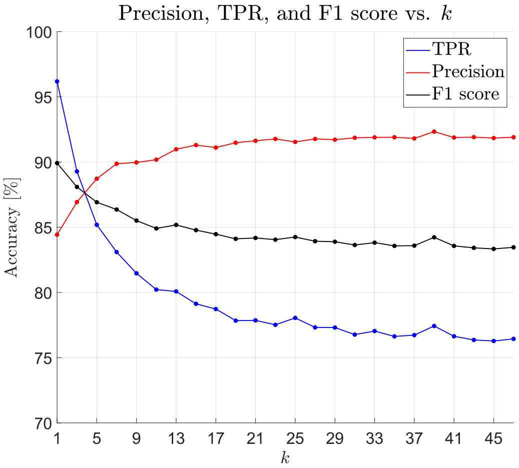
### 6.4.3 A swarm including 487 astrobots

The qualities of the results in this case fairly follows the qualities of the 116-astrobots case. So, we observe that our algorithm performance remains relatively acceptable even in the case of very complex swarms. To support this claim, we consider a dataset including 10100 configurations, 10049 of which are used to train the predictor and the remaining 51 ones are test configurations. The number of iterations are 15.

Confusion matrices of Figure 6.11 reiterate the point that larger $k$ values give rise to the better predictions of the negatives. Figure 6.11c witnesses the decrement of the balanced accuracy compared to



(a) $\alpha = \beta = 1$ (b) $k = 13$



(c) $\alpha = \beta = 1$

Figure 6.9: Accuracy measures for the 116-astrobot swarm

Table 6.1: The best prediction results corresponding to the 116- and 487-astrobot swarms

| Swarm population | K | $\alpha$ | $\beta$ | TPR(%) | TNR(%) | Balanced accuracy(%) | Precision(%) | F1(%) |
|---|---|---|---|---|---|---|---|---|
| 116 | 31 | 1 | 0.9 | 79.3 | 64.7 | 72 | 91.51 | 84.97 |
| | 39 | 1 | 0.85 | 88.44 | 63.23 | 71.83 | 91.4 | 85.57 |
| | 39 | 1 | 1 | 77.2 | 68.51 | 72.85 | 92.22 | 84.04 |
| 487 | 39 | 1 | 0.9 | 79.94 | 60.73 | 70.33 | 89.51 | 84.45 |
| | 51 | 1 | 0.88 | 80.20 | 60.97 | 70.59 | 89.52 | 84.61 |
| | 51 | 1 | 1 | 78.23 | 63 | 70.62 | 89.78 | 83.61 |

the 116-astrobot swarm. The reason is that the 487-astrobot swarm comprises more total neighborhoods than the 116-astrobot swarm. The stability analysis of this case, similar to the previous case, also indicates the variations of the accuracy rates with respect to the hyperparameters as shown in Figure 6.12. In particular, Figure 6.12a exhibits that the algorithm is stable for $k > 21$. Moreover, Figure 6.13a illustrates the upper bound of the precision which is around 90%. Finally, we observe that the algorithm works on this 487-astrobot swarm almost as good as the 116-astrobots one. Namely, Figure 6.13 exhibits the ROC curve of the 487-astrobots case which is trivially closer to the random guess line compared to that of the 116-astrobot swarm.

## 6.5 Conclusion

The first solution to the convergence prediction of populated packs of astrobots is studied. We observe that astrobot-to-target assignments provide a necessary feature subset of an astrobot swarm feature set to reach ∼80% of accuracy in predicting the completely-converging set of the pairings. The $k$-NN nature of the proposed algorithm makes the metric design process intuitive enough to exploit the geometrical characteristics of astrobots and their neighborhoods. The presented strategy also enjoys a fairly restricted number of hyperparameters. So, the design process is not only relatively straightforward but tuning processes also require less computational resources.

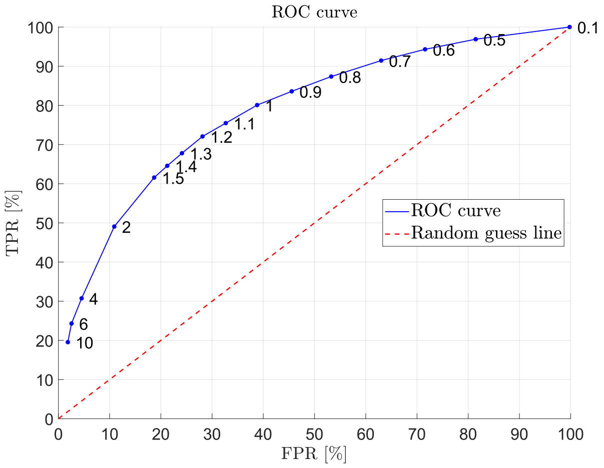
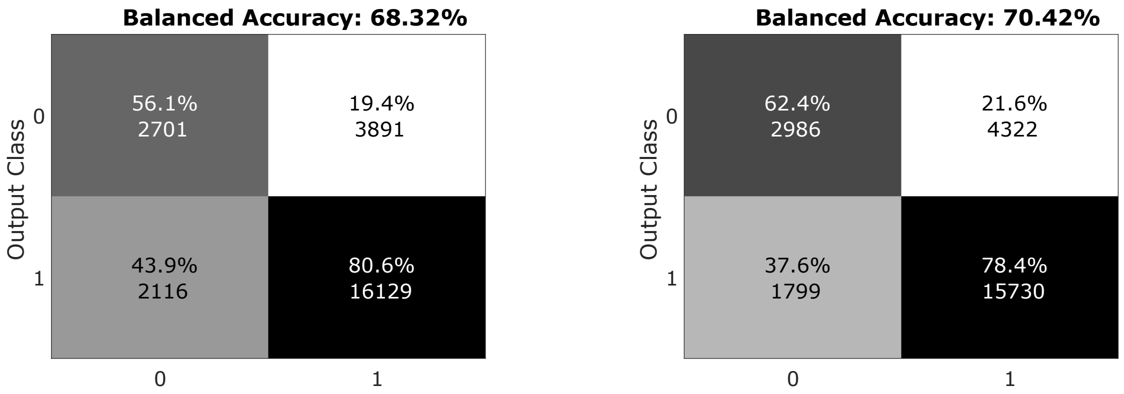
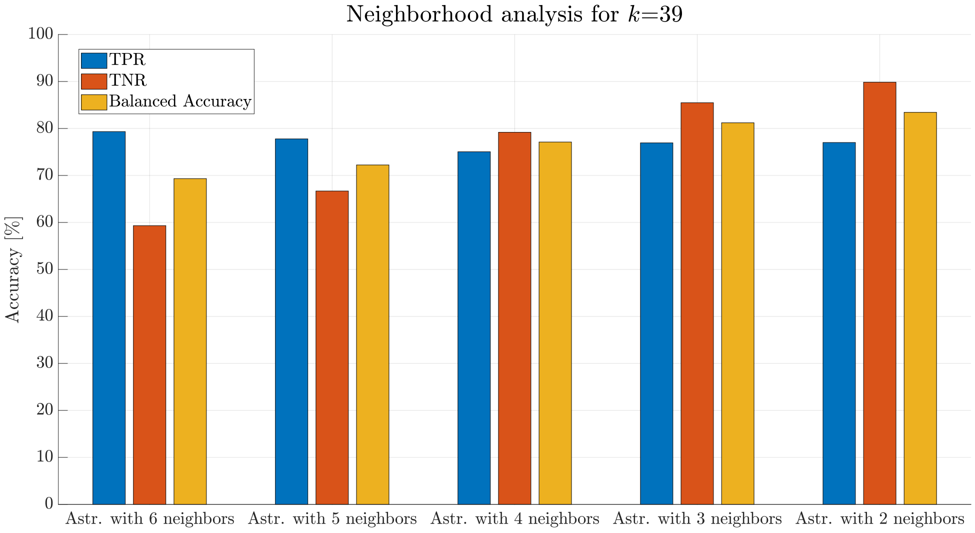


Figure 6.10: The ROC curve associated with the 116-astrobot swarm



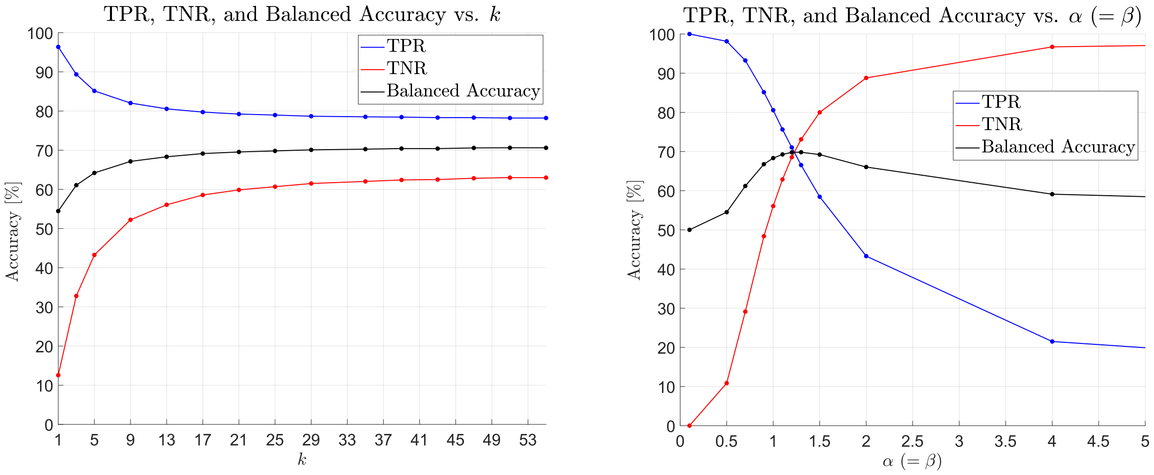
(a) $\alpha = \beta = 0.95$

(b) $\alpha = \beta = 1.00$



(c) $\alpha = \beta = 1.05$

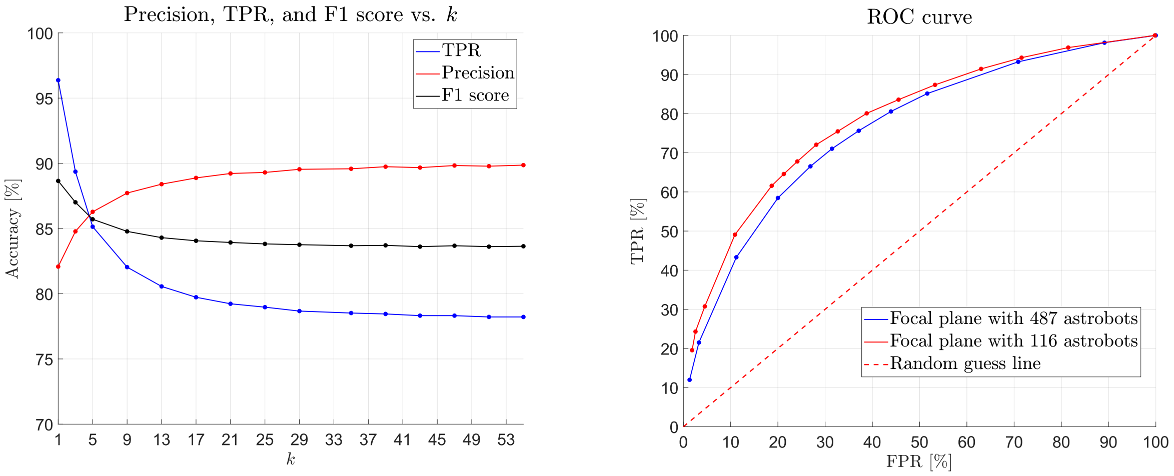
Figure 6.11: Two confusion matrices and a neighborhood analysis corresponding to the 487-astrobot swarm



(a) $\alpha = \beta = 1$

(b) $k = 13$

Figure 6.12: Accuracy measures for the 487-astrobot swarm



(a) The variation of performance criteria in the case of the 487-astrobots scenario

(b) The comparative ROC curve evolution of both scenarios

Figure 6.13: Further accuracy results associated with both swarms

# Chapter 7

# Variable-Parity Convergence Prediction

## Publication Note

The material presented in this chapter is adapted from:

- **Matin Macktoobian**, Francesco Basciani, Denis Gillet, and Jean-Paul Kneib, Learning Convergence Prediction in Astrobot Swarms, *Journal of Astronomical Telescopes, Instruments, and Systems*. (In Press)

## 7.1 Introduction

The convergence prediction using machine learning seeks an estimation of the post-coordination status of a swarm in view of those targets which can be reached. In this framework, we only take the initial configuration of astrobots without engaging with the intricacies of their interactions in the course of their coordination. Finally, if the estimated prediction is below a desired threshold, one simply switches to another plan of target-to-fiber assignments. The first solution to this problem [Macktoobian et al. 2020b] , as described in Chapter 6, proposed a weighted $k$-NN-based algorithm [Dubey et al. 2013] to predict the intended convergences. Despite the promising accuracies obtained by this algorithm, there are drawbacks which may be envisaged corresponding to this method. In particular, this algorithm is a lazy-evaluating method which indeed does not generate any model out of its learning process. Thus, all geometrical computations corresponding to every train data sample have to be done associated with every observation. This computational issue may significantly slow predictions.

Categorical data cannot be properly embedded into the $k$-NN-based scheme because this technique uses distance metric which is not applicable to categorical data in a straightforward manner. So due to this pitfall, the $k$-NN-based strategy only considers a constrained version of the prediction problem in which only spatial features of the locations of targets are taken into account. However, a more realistic and more accurate model may be trained if one takes the critical categorical features of astrobots, such as parity, as well. Namely, parity denotes the motion direction of the outer arm of an astrobot. The constrained case solved by the $k$-NN-based approach assumed that the parities of all astrobots are the same. This assumption is so restrictive in terms of decreasing the maneuverability of astrobots. In the sequel, swarm controllers may end up with very low convergence rates because of such extremely restrictive constraint. On the other hand, relaxation of the fixed-parity assumption makes a prediction substantially more difficult because even if one only toggles the parity of one astrobot of a swarm, the convergence of many astrobots may be affected. In other words, the consideration of parity has to be efficiently managed to reach high prediction accuracies. Accordingly, we seek a solution, based on the idea of support vector machines (SVM) [Suykens et al. 1999], to a generalized version of the convergence prediction problem in which categorical data, particularly parity, can also be incorporated into the process.

The contributions of this paper are three-fold as follows. First of all, we obtain a prediction model which can be simply evaluated for arbitrary coordination cases. Put differently, once our model is computed according to a particular swarm, any test scenario associated with that swarm can be instantly evaluated. It is a notable advancement compared to the lazy evaluations of constrained scenarios handled by the $k$-NN-based method. We observe that our SVM-based algorithm provides improved prediction performance with respect to those of the $k$-NN-based strategy. Second, the $k$-NN-based scheme requires a neighborhood analysis step to localize distance measurements in the course of evaluations. However, our SVM-based predictor models each astrobot such that computations are inherently localized. So, one needs no extra pre-processing to localize data before any learning phase. Third, we incorporate parity in our algorithm using a normalization phase. In particular, we transform the categorical parity pair to a numerical one whose variation resembles those of the spatial features of astrobots. Thanks to this formulation, no feature exerts excessive impact on the prediction results by overriding the influences of other ones.

The chapter is structured as follows. Section 7.2 describes the synthesis of our SVM-based predictor in the constrained case, say, when all astrobots have the same parity. The black processing path of Figure 6.4 represents the underlying steps of the constrained case. To be specific, we first define the computational model of an astrobot, including its selected features, which is later used in the learning phase. We then describe the necessity of scaling the features. A detailed treatment of the predictor synthesis process and the applied cross-validation procedure are also discussed. Section 7.3 illustrates how the constrained algorithm can be extended to a generalized one to cover parities, as well. Following the dotted red processing path of the algorithm in Figure 6.4, once parity is normalized, a predictor can be synthesized according to the formalism used to solve the constrained case. We apply our method to a 487-astrobot swarm to reflect the efficiency of the synthesized predictor, in Section 7.4. Finally, Section 7.5 remarks our conclusion.

## 7.2 Constrained Convergence Prediction

### 7.2.1 Data Definition

In the constrained scenario, we assume that the parities of all astrobots are fixed and similar to each other. Since each coordination starts from the folded formation of astrobots, similarly to the $k$-NN-based algorithm, the data construction associated with the coordinate of each astrobot corresponds to the coordinate of the projected location of the particular target assigned to it. Thus, given a fixed parity for all astrobots of a swarm, the constrained data model of astrobot $\pi$ that has $n$ neighbors is defined as a collection of spatial features as follows

$$\boldsymbol{\pi}^{\mathsf{C}} := \begin{bmatrix} x_t & y_t & x_t^1 & y_t^1 & \cdots & x_t^n & y_t^n \end{bmatrix}^{\mathsf{T}}. \tag{7.1}$$

Here $\begin{bmatrix} x_t & y_t \end{bmatrix}^{\mathsf{T}}$ denotes the target specification of $\pi$, and each vector $\begin{bmatrix} x_t^i & y_t^i \end{bmatrix}^{\mathsf{T}}$ is associated with the target coordinate of its $i$th neighbor. One observes the absence of any parity information in the data vector because it is fixed for all astrobots. Note that the model above is basically including only one neighborhood. Thus, one needs not to localize the data before being fed into any learning process, as the $k$-NN based method requires. In our SVM-based strategy, given a data vector $\boldsymbol{\pi}^{\mathbf{C}}$, the predictor exclusively returns its evaluation regarding the convergence of the central astrobot $\pi$. So, to predict the convergence of $n$ astrobots of a swarm, one has to take $n$ neighborhoods into account the central astrobot of each of which is one of those $n$ astrobots.

### 7.2.2 Feature Scaling

Before using data vectors to train any predictor, we have to scale the features of each data vector. Namely, the target's coordinates of different neighbors do not vary in similar ranges among all data vectors. So, there is an intrinsic spatial offset in the features with respect to a particular astrobot. Such an offset has to be removed not to synthesize biased predictors. Lack of feature scaling in the case of SVM-based predictors makes the setting of their hyperparameters very complicated. Comparatively, the $k$-NN-based algorithm does not require any feature scaling since the Euclidean distance metric applied to that method is relative and localized. In contrast, we will later see that the quoted metric is used in the Gaussian kernel of our SVM-based predictor. So, we have to take feature scaling into account. In particular, we use min-max normalization [Han et al. 2011] to linearly transform the range of all data features to the interval $[-1, 1]$. This range not only removes the mean value of each feature one but also yields satisfying results in view of the performance of synthesized predictors. Mathematically, the following formalism maps feature $x$ to its normalized counterpart $x'$ which varies in the cited range.

$$x' := \frac{2(x - \min(x))}{\max(x) - \min(x)} - 1 \tag{7.2}$$

Here, $\min(x)$ and $\max(x)$ operators return the minimum and the maximum values of the feature $x$, respectively, associated with a data vector $\boldsymbol{\pi}^{\mathbf{C}}$ of a particular dataset.

One may note that the linear nature of the transformation above indeed preserves relative distances between the targets of a particular neighbohrood. We generally assume a uniform distribution of targets all over a focal plane. Thus, min-max normalization is a better option than Z-score normalization [Jain et al. 2005] which is often particularly applicable to the data following Gaussian distributions.

### 7.2.3 Predictor Synthesis

It is unlikely that a linear boundary can generally solve the convergence prediction problem of an astrobot swarm. Thus, we apply the kernel trick [Hofmann 2006] to our linearly inseparable data. In particular, we map data vector $\boldsymbol{\pi}_i^{\mathbf{C}}$ to $\boldsymbol{\pi}_i^{\mathbf{C}'}$ by the following kernel $\kappa(\cdot, \cdot)$, known as radial basis function [Murphy 2012],

$$\kappa(\boldsymbol{\pi}_i^{\mathbf{C}}, \boldsymbol{\pi}_i^{\mathbf{C}'}) := \Phi(\boldsymbol{\pi}_i^{\mathbf{C}})^{\intercal} \Phi(\boldsymbol{\pi}_i^{\mathbf{C}'}) = \exp\left(-\frac{\|\boldsymbol{\pi}_i^{\mathbf{C}} - \boldsymbol{\pi}_i^{\mathbf{C}'}\|^2}{2\sigma^2}\right). \tag{7.3}$$

Here, the kernel size $\sigma$ determines the width of the Gaussian kernel.

The problem of predictor synthesis is equivalent to the solution of the optimization problem below. We seek an optimal hyperplane which classifies the convergence of a particular set of astrobots into the class of 1s (resp., 0s) if they are predicted to reach their targets (resp., otherwise).

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\boldsymbol{w}^{\mathsf{T}}\boldsymbol{w} + \mathcal{C}_0 \sum_{i=1}^{N_0} \xi_i^0 + \mathcal{C}_1 \sum_{i=1}^{N_1} \xi_i^1 & \\ \text{s.t.} \quad & g_i^j(\boldsymbol{w}\Phi(\boldsymbol{\pi}_i^{\mathbf{C}}) + b) \geq 1 - \xi_i^j, & j \in \{0,1\} \\ & \xi_i^j \geq 0, & j \in \{0,1\} \end{aligned} \tag{7.4}$$

The boundary between the two classes is denoted by normal vector $\boldsymbol{w}$. $N_0$ and $N_1$ are the numbers of the samples in the classes of 0s and 1s, respectively. Weights $\mathcal{C}_0$ and $\mathcal{C}_1$ represent the miss-classification penalties associated with the classes of 0s and 1s, respectively. Given miss-classification measure $\mathcal{C}$, we have

$$C_j := C\omega_j, \quad j \in \{0,1\}, \tag{7.5}$$

where $\omega_j$ is the class weight of the class $j$. The notion of class weight is also used to resolve the imbalanced data problem. To balance the bias with respect to the abundance of majority class, i.e., 1s, compared to the minority class, i.e., 0s, we apply the class weights to the class of 1s. For this purpose, one has to regulate the hyperparameter $\omega_1$. In general, one may safely apply either a smaller weight to the majority class or a larger one to the minority class. The quantities $\xi_i^0$ and $\xi_i^1$ are the slack variables corresponding to the incorrect classifications of the $i$th samples regarding the classes of 0s and 1s, respectively. Due to the complexity of our prediction problem, we use these variables to relax classification constraints and allow miss-classifications of some data samples. These values are larger than 0 only if their corresponding samples are miss-classified. Moreover, the ground truth of the $i$th sample is encoded by complement-pair $(g_i^0, g_i^1)$ with respect to the classes of 0s and 1s, respectively. The $i$th astrobot is represented by $\boldsymbol{\pi}_i^{\mathbf{C}}$, and kernel function $\Phi(\boldsymbol{\pi}_i)$ maps every feature of $\boldsymbol{\pi}_i$ into a higher dimensional space. $b$ indicates the hyperplane intercept. One observes that the term $\frac{1}{2}\boldsymbol{w}^{\mathsf{T}}\boldsymbol{w}$ is the inverse of the margin between the two classes. The minimization of this term indeed gives rise to the maximization of the desired margin corresponding to the predictor boundary. The setting of the cited hyperparameters are described in Section 7.4.

The optimization problem (7.4) is solved using the sequential minimal optimization algorithm [Platt 1998]. The obtained boundary hyperplane is built according to a subset of the data samples, i.e., support vectors, which are the closest data points to the hyperplane. This linear hyperplane in the expanded space is projected back to the original space as a nonlinear one. Once the hyperplane is found, we assign a new test vector to one of the two classes of the problem. This assignment depends on the relative position of the data vector with respect to the hyperplane model.

### 7.2.4 Validation

We employ $k$-fold cross validation method to check the performance of our algorithm. Namely, we synthesize a desire SVM-based convergence predictor using a train partition $\mathtt{D}_{\mathtt{T}}$ of a particular dataset $\mathtt{D}$. Then, we apply the algorithm to the second partition, say, a test partition $\mathtt{D}_{\mathtt{S}}$, to assess the algorithm's performance. In this regard, we perform $\lfloor |\mathtt{D}| / |\mathtt{D}_{\mathtt{S}}| \rfloor$ validation iterations. Thanks to this method, all elements of a dataset are used both as a part of the train and test partitions in the end of a cross validation process. Moreover, one may not use one data sample more than once, as it may happen in the Monte-Carlo cross validation [Xu et al. 2001] applied to the $k$-NN-based algorithm. In each iteration, we take a different partition of $\mathtt{D}$ as $\mathtt{D}_{\mathtt{S}}$. We compute the average of the performance results obtained at the end of every iteration over the total number of $k$ iterations, thereby obtaining the final results of the cross validation process. The value of $k$ depends on the ratio $\lfloor |\mathtt{D}| / |\mathtt{D_S}| \rfloor$. Decreasing the cardinality of $\mathtt{D_S}$ increase the value of $k$ and the required time of cross validation completion. However, increasing the size of $\mathtt{D_S}$ implies the usage of less samples in the training phase. Thus, less training data may escalate the risk of underfitting in the course of the SVM-based predictor synthesis. Section 7.4 illustrates our setting corresponding to this hyperparameter.

## 7.3 Generalized Convergence Prediction

In this section, we generalize our convergence prediction by adding parity to the features of each astrobot. Put differently, according to this generalization, each astrobot's parity may be different from those of other astrobots. This parity relaxation makes the convergence prediction process even more complicated because various parities increase the nonlinear interactions of astrobots. In particular, such interactions give rise to the grow of the potential deadlock and/or collision-prone scenarios. So, predicting the safety and eventual completeness of any swarm initial configuration is even more challenging. We describe how the generalized version of our SVM-based algorithm efficiently manages to predict the desired safe complete convergences.

### 7.3.1 Data Definition Generalization

The notion of parity denotes the rotation direction of rotation of the outer arm of a typical astrobot. Thus, it is inherently classified as a categorical information, contrary to the continuous numerical values corresponding to the remainder of an astrobot's spatial features. In this section, we add parity information to the constrained data vector of an astrobot (7.1) to obtain the generalized data vector as follows

$$\boldsymbol{\pi}^{\mathbf{G}} := \begin{bmatrix} x_t & y_t & P & x_t^1 & y_t^1 & P^1 & \cdots & x_t^n & y_t^n & P^n \end{bmatrix}^{\mathsf{T}}, \tag{7.6}$$

in which $P$ refers to the parity flag of the modeled astrobot $\pi$, and $\{P^i \mid 1 \geq i \geq n\}$ denotes the parity set corresponding to $n$ neighbors of $\pi$.

The embodiment of categorical data into the $k$-NN predictor is challenging because the distance metric defined in that framework may not compare parities of astrobots in different configurations similarly to the applied distance measurements to the spatial features. However, the metric of our SVM-driven predictor is applied to the data in another computational layer in the kernel. Moreover, SVM algorithm more efficiently deals with nonlinearities of astrobots' data.

### 7.3.2 Parity Normalization

In addition to the min-max normalization applied to the spatial features (7.3), this section describes the importance of parity normalization, as well. In particular, a categorical parity value is either 1 (resp., -1) to represent clockwise (resp., counterclockwise) motions of outer arm of an astrobot. In view of the optimization problem (7.4), parity data are processed as integer numbers. Thus, the range of their variation has to be normalized such that they vary in a more-or-less similar variation range as those of the normalized spatial features.

To efficiently normalize parity, we analyze the standard deviation corresponding to the spatial features. Targets distribution is generated by a standard uniform distribution. However, one cannot simply yield the desired standard deviation using ideal formula associated with standard uniform distribution, which

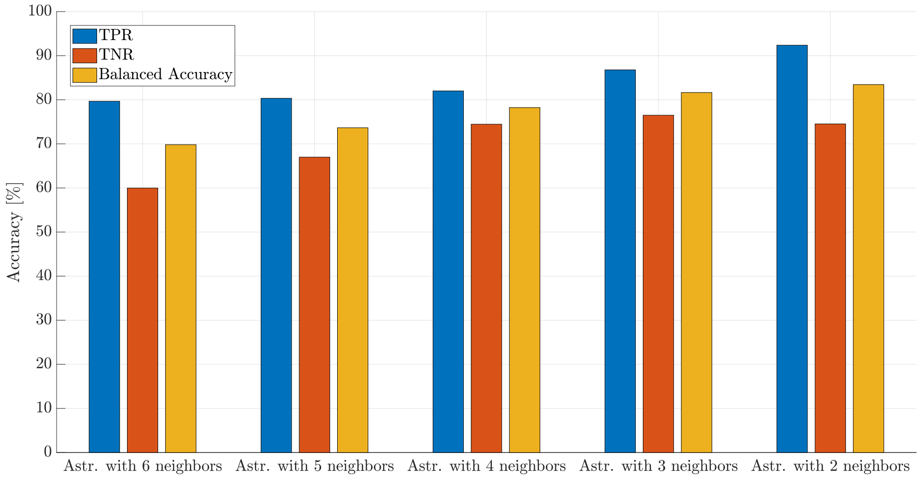
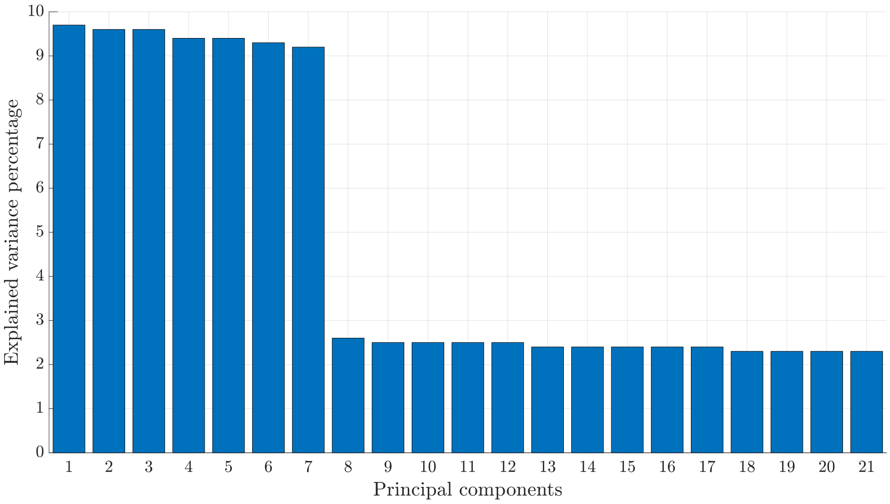
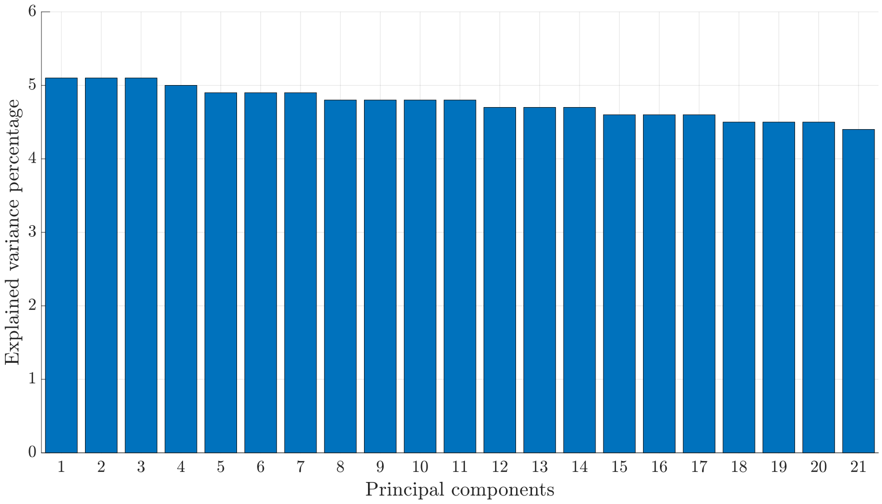


Figure 7.1: Prediction accuracy per neighborhood type in the constrained scenario



(a) Unbalanced explained variance before parity normalization



(b) Balanced explained variance after parity normalization

Figure 7.2: Parity normalization impact on the magnitudes of data features

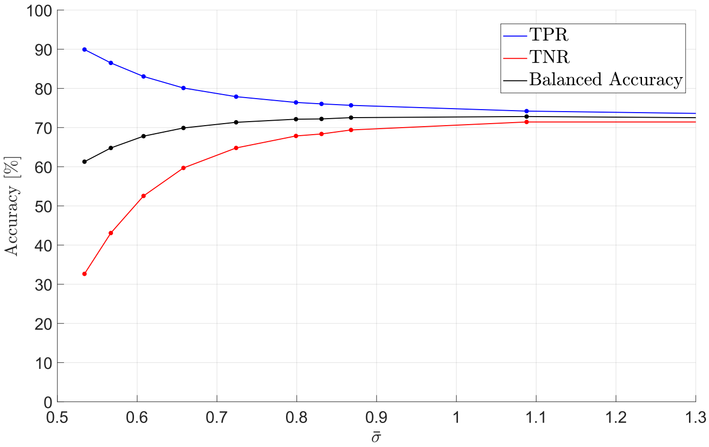


Figure 7.3: TNR, TPR, and balanced accuracy dynamics with respect to average kernel size in the constrained scenario

is ∼0.577. Since the outer arm of an astrobot is longer than its first arm, the reachability requirement of target-to-astrobot assignment requires that any generated target may be located outside of the focal plane as long as the target is still reachable by at least one astrobot. The realization of the aforesaid condition empirically requires that one computes the desired spatial standard deviation based on not the general formula but the available data. If we take values -1 and 1 to represent various parities, then the parity standard deviation is ∼1. It turns out that this encoding leads to a noticeable imbalance in view of the data explained variance[1]. In other words, the parity pair $(-1, 1)$ makes our SVM-based predictor biased in relying more on the information given by parities compared to those of the spatial features, thereby reducing the prediction quality of the final learning model.

We obtain the explained variance associated with the features of an astrobot with six neighbors using principal component analysis, as depicted in Figure 7.2a. In this figure, the feature imbalance is obvious. So, we need to scale parity values such that their resulting explained variance is fairly similar to that of the spatial features. In particular, the analysis of the target's distributions indicates that the standard deviation of spatial features is ∼0.5. On the other hand, imbalance feature issue mandates that the parity standard

---

[1] Explained variance is the ratio of the variance of a specific feature to the summation of the variances of all features of data.

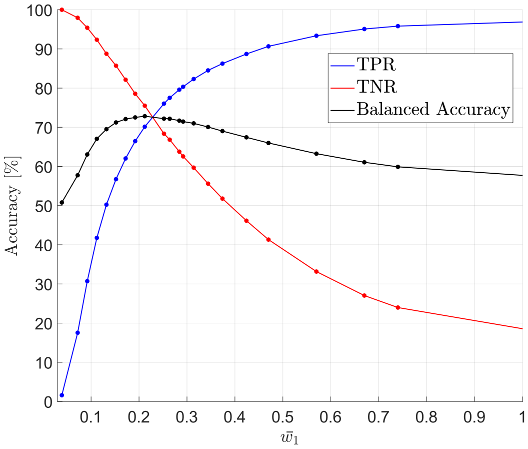


Figure 7.4: TNR, TPR, and balanced accuracy with respect to average class weight in the constrained scenario

deviation follows that of the spatial features. One notes that the parity pair $(-0.5, 0.5)$ fulfills the quoted conditions. Taking the planned parity pair into account efficiently scales the explained variance of all features, as Figure 7.2b illustrates. We note that choosing parity pairs with smaller variation range, e.g., $(-0.3, 0.3)$ leads to another issue. In particular, such pairs drive the prediction model to be mostly inclined to the information of spatial features by taking less impact of parities into account on the prediction model synthesis.

Once parities are normalized, one simply feeds all $\boldsymbol{\pi}^{\mathbf{G}}$ vectors to the optimization problem (7.4). The resulting boundary is the prediction model that is the solution to the generalized convergence prediction problem.

## 7.4 Results

In this section, we demonstrate the performance of our algorithm applied to a 487-astrobot swarm in both constrained and generalized cases. This swarm resembles the one corresponding to the SDSS-V project [Kollmeier et al. 2017]. We partition our dataset such that every time the test partition is 10% of the overall dataset, thereby $k = 10$. We describe how our algorithm not only solves the constrained case with higher performance compared to the $k$-NN-based algorithm but also efficiently solves the generalized case.

We synthesize an SVM model for each astrobot of the swarm. After the completion of each iteration, we compute each of the average performance metrics corresponding to each astrobot's atomic performance metric over the number of all iterations. Then, we obtain the performance metrics of the swarm by averaging over those of all astrobots.

We fix $C = 1$, so two hyperparameters $\omega_1$ and $\sigma$ have to be set for each astrobot. Such setting associated with a particular astrobot critically depends on the cardinality of its neighborhood. Each astrobot empirically possesses two to six astrobots in its neighborhood. Thus, the hyperparameter pair above has to be determined for five various scenarios. For this purpose, we tune class weight $\omega_1$ and kernel size $\sigma$ for each neighborhood type by performing a grid search.

### 7.4.1 Constrained Scenario

In this scenario, we assume that the parities of all astrobots are the same. Each astrobot dataset comprises 10100 samples. We set the hyperparameters regarding two interesting cases. As Table 7.1 illustrates, case I intends to simultaneously maximize balanced accuracy and keep TPR above 75%. Case II seeks a minimum TPR of 80% at the expense of the balanced accuracy decrement. One may note that, given either of the cases, each hyperparameter varies in a relatively narrow margin with respect to the neighborhood type. In other words, the prediction performance are fairly invariant to the variation of the hyperparameters around some particular values. This feature is computationally very important in that one may simply consider a single setting of each hyperparameter for all astrobots of a swarm regardless of differences among their neighborhood types. In this regard, the prediction performance would be sufficiently high, yet no extensive grid search is done to tune hyperparameters based on their specific neighborhood types. The described neighborhood-dependence of prediction accuracies is depicted in Figure 7.1.

The impact of the kernel size on the trade-off between TPR and TNR is illustrated in Figure 7.3. Accordingly, $\sigma \geq 0.8$ provides TPR values over 80%. The variation trends of balanced accuracy and F1 score with respect to the class weight $\omega_1$ are also rendered in Figure 7.4 and 7.5, respectively. The optimal trade-

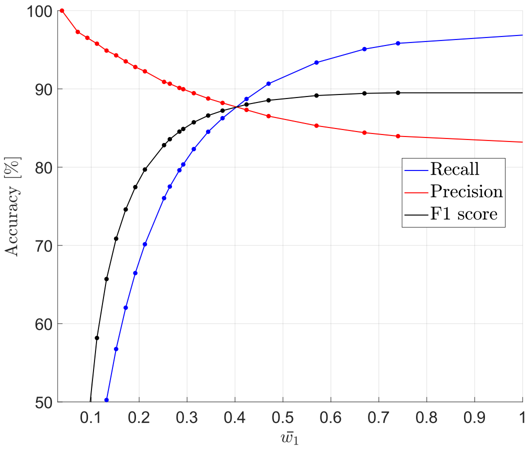


Figure 7.5: Precision, recall and F1 dynamics with respect to average class weight in the constrained case

off selections are those points at which the graphs intersect. The ROC curve represented in Figure 7.6 clearly depicts how our SVM-based algorithm is more efficient than the *k*-NN-based one. In particular, the ROC curve of our algorithm is located farther from the random guess line compared to that of the *k*-NN-based one. The overall report of the best prediction performance is reported in Table 7.1.

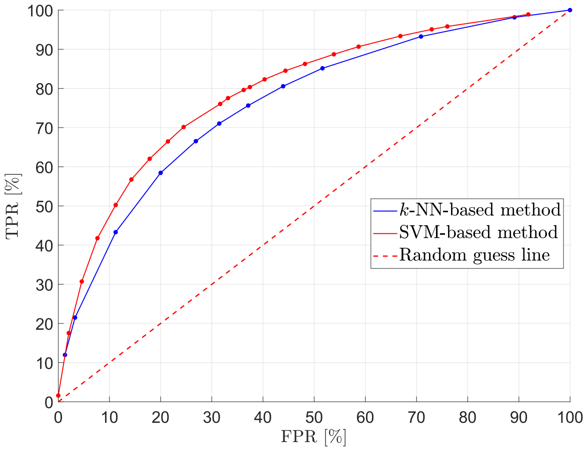


Figure 7.6: Comparative ROC curves corresponding to the constrained scenario

### 7.4.2 Generalized Scenario

We generalize the convergence prediction of the swarm studied in the previous section by randomly determining the parities of its astrobots. Because of the more complexity of the generalized case compared to the constrained one, we take 5000 extra samples per astrobots compared to the previous scenario, i.e., taking 15100 data samples into account. Table 7.2 includes the hyperparameter setting to achieve the best predictions in two cases similar to the constrained case. Namely, the case I and II seeks the maximized balanced accuracy and the maximized TPR, respectively.

The neighborhood analysis and the performance variation with respect to kernel size $\sigma$ are depicted in Figure 7.7 and 7.8. Performance dynamics regarding the variation of class weight $\omega_1$ are represented in Figure 7.9 and 7.10. These trends interestingly resemble those of the constrained case. So, the sensitivity of the prediction accuracy in terms of switching between the two scenarios is fairly robust. The complete

Table 7.1: The best prediction results corresponding to the constrained case[2]

| Case | NT | $\omega_1$ | $\sigma$ | TPR(%) | TNR(%) | BA(%) | Precision(%) | F1(%) |
|---|---|---|---|---|---|---|---|---|
| I | 6 | 0.277 | 0.86 | 75.7 | 69.4 | 72.5 | 91.12 | 82.68 |
| | 5 | 0.216 | 0.90 | | | | | |
| | 4 | 0.167 | 0.86 | | | | | |
| | 3 | 0.167 | 1.47 | | | | | |
| | 2 | 0.179 | 0.84 | | | | | |
| II | 6 | 0.317 | 0.82 | 80.3 | 62.6 | 71.45 | 89.96 | 84.88 |
| | 5 | 0.256 | 0.86 | | | | | |
| | 4 | 0.207 | 0.82 | | | | | |
| | 3 | 0.207 | 0.98 | | | | | |
| | 2 | 0.219 | 0.80 | | | | | |

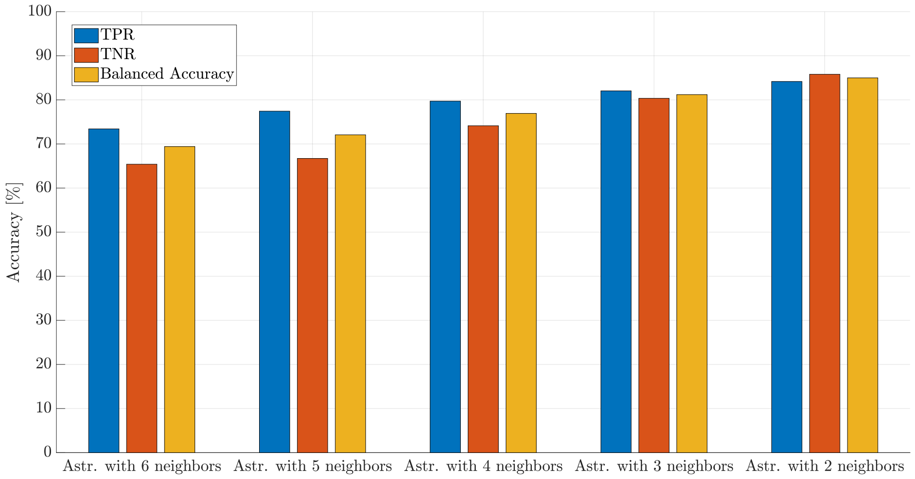


Figure 7.7: Prediction accuracy per neighborhood type in the generalized scenario

trace of the performance metrics of the generalized case is reflected in Table 7.2. This table exhibits the effective application of our SVM-based algorithm to incorporate the notion of parity in the convergence prediction of the swarm.

The comparison of the predictive functionalities of the all the available algorithms, i.e., the constrained $k$-NN-based algorithm, the constrained SVM-based algorithm, and the generalized SVM-based algorithm, is illustrated in the ROC curve set of Figure 7.11. In particular, one observes that the constrained SVM-based approach is more expressive than the constrained $k$-NN-based method. In other words, with the assumption of fixed parities, the former has to be preferred to the latter. Moreover, the ROC curve of the generalized SVM-based strategy is above that of the constrained $k$-NN-based one, but trivially below that of the constrained SVM-based method. Comparatively, the generalized SVM-based algorithm deals with the complexity of parity which is not taken into account by the constrained version. Nevertheless, it is not an excessive cost in the performance reduction of the prediction by adding parity to predictions.

---

[2] NT refers to neighborhood type which represents the number of the neighbors of an astrobot. BA designates balanced accuracy.

Table 7.2: The best prediction results corresponding to the generalized case[2]

| Case | NT | $\omega_1$ | $\sigma$ | TPR(%) | TNR(%) | BA(%) | Precision(%) | F1(%) |
|---|---|---|---|---|---|---|---|---|
| I | 6 | 0.290 | 1.55 | 75.1 | 66.6 | 70.8 | 90.3 | 82.0 |
| | 5 | 0.250 | 1.55 | | | | | |
| | 4 | 0.200 | 1.17 | | | | | |
| | 3 | 0.140 | 0.99 | | | | | |
| | 2 | 0.110 | 0.99 | | | | | |
| II | 6 | 0.330 | 1.42 | 79.7 | 60.5 | 70.1 | 89.4 | 84.3 |
| | 5 | 0.290 | 1.42 | | | | | |
| | 4 | 0.240 | 1.05 | | | | | |
| | 3 | 0.180 | 1.09 | | | | | |
| | 2 | 0.150 | 1.09 | | | | | |

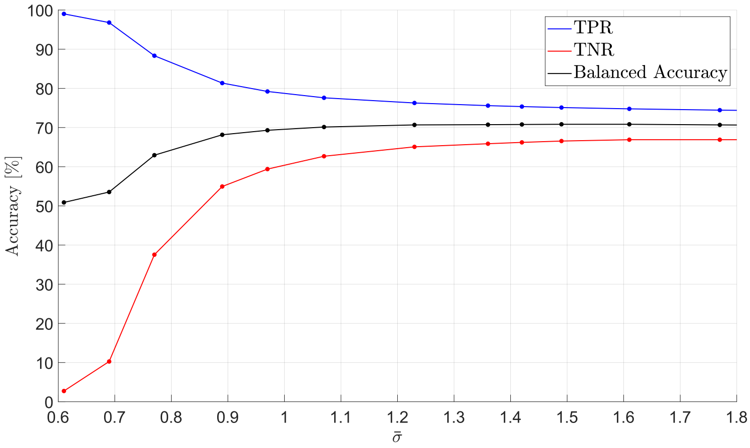


Figure 7.8: TNR, TPR, and balanced accuracy dynamics with respect to average kernel size in the generalized scenario

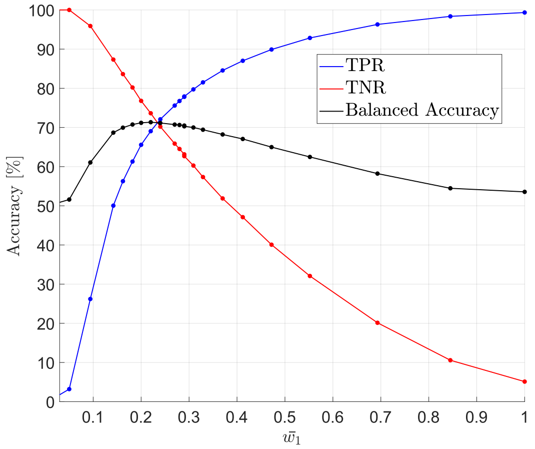


Figure 7.9: TNR, TPR, and balanced accuracy with respect to average class weight in the generalized scenario

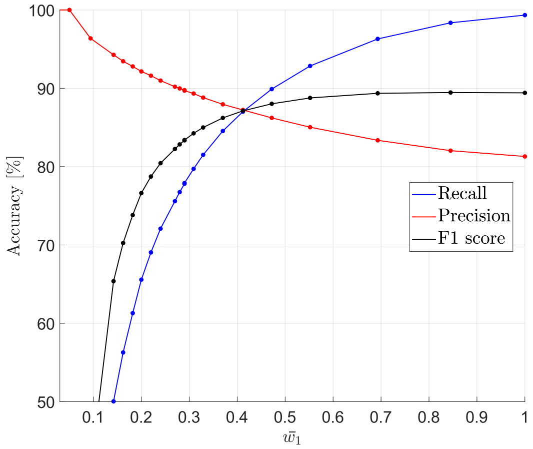


Figure 7.10: Precision, recall and F1 dynamics with respect to average class weight in the generalized scenario

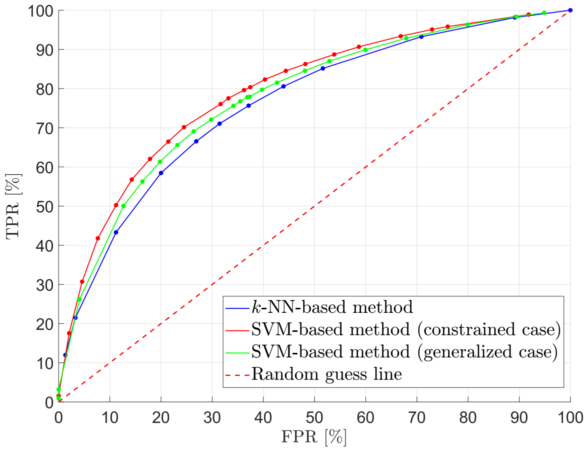


Figure 7.11: Overall comparative ROC curves of both the constrained and the generalized scenarios

## 7.5 Conclusion

Astrobot swarms are populated extremely dense formations of specific manipulators whose collision-free coordination are of utmost importance for astronomical operations. Cosmological operations often require that the number of fully coordinated astrobots is above a particular threshold. However, the convergence rates achieved by the distributed controllers of astrobots can only be studied using intensive simulations. The prediction of astrobots' convergences has been already done in a constrained case using a $k$-NN-driven strategy in which only spatial features of astrobots' targets are taken into account. In this chapter, we illustrate that the accuracy performance of our SVM-based algorithm is higher than that of the $k$-NN-based one. Moreover, our algorithm also brings parity, say, the rotation direction of the outer arm of an astrobot, into play, thereby generalizing the solution to the convergence prediction problem. The comparative study of the performance results indicates that the parity addition to the formulation only trivially reduces the quality of the prediction accuracy. So, one observes that the generalization of the convergence prediction is efficiently realized.

# Final Words

## Chapter 8

# Conclusions

### Publication Note

The material presented in this chapter is adapted from:

- **Matin Macktoobian**, Denis Gillet, and Jean-Paul Kneib, Astrobotics: Swarm Robotics for Astrophysical Studies, *IEEE Robotics and Automation Magazine*, 2021.

## 8.1 Highlights

We solve the complete safe coordination problem of astrobot swarms. In this regards, the target-to-astrobot assignment problem was studied noting the importance of this problem and its influence on the quality of its subsequent coordination phase. We explain how the former solutions to this problem do not particularly consider any criteria to ease coordination. We then established an optimal target-to-astrobot assignment strategy to minimize both the coordination time and the collision likelihood between astrobots. We yielded the latter by maximizing the distribution of allocated astrobots. Our quadratic algorithm indeed effectively scales to crowded swarms of astrobots and observations. We observed the efficiency of this method in terms of its functionality compared to the state-of-the-art scheme, say, target-based target-to-astrobots assignment technique. In particular, using our method, the number of deadlocks during coordination by navigation-function-based coordinators decreases. The sizes of coordination supervisors, synthesized by supervisory control theory, also shrinks.

We review the two leading candidate methods to safely and completely coordinate astrobot swarms, say, the usage of cooperative artificial potential fields and supervisory control. One observed that cooperative artificial potential fields elegantly seek global complete coordination of a whole swarm via the accumulation of locally complete partitions of astrobots. This strategy leads to a local completeness condition whose fulfillment has to be checked for every neighborhood of astrobots. The results exhibited how this scheme is able to yield completely in the majority of scenarios. However, if the condition is violated in a neighborhood, then the parameters of the target-astrobots pairing will be trivially modified to resolve the violation. On the other hand, the supervisory coordination strategy not only provides completeness but also automatically synthesize the solution which require no safety and functional validations. To hit this mark, we enjoy the embodiment of functional and safety requirements of astrobots into the discrete-event models of the supervisory control. Once a coordination supervisor is synthesized, the coordination problem indeed resembles the reconfiguration problem of discrete-event systems, whose solution is known.

The coordination problem of astrobot swarms is a complicated one in which many nonlinear interactions exist in the dynamics of the whole system. However, we use machine learning techniques to predict the result of the coordination associated with every astrobot of a swarm. Our strategies are reliably capable of predicting convergence rates given various motion specifications of astrobots, saying both constrained and variable parities. Thanks to these predictions, the likelihood of the complete convergence corresponding to a swarm can be predicted before performing any intensive time- and resource-consuming simulations. In particular, if the predicted convergence rate is lower than a desired threshold, then one may modify the assignment pairings of the system and/or functional setting of astrobots to reach an estimation over the convergence threshold. As the result, the coordination will end up with a configuration in which the resolution of the generated survey fulfills its intended level.

Overall, we improved the convergence rate by two means. First, we planned target-to-fiber assignments to ease coordination in view of both decreased convergence time and less deadlock-prone situations. Second, we orchestrated coordination in localities where astrobots consider the overall convergence of their localities, rather than merely their own convergence. We also used machine learning methods to synthesize some predictors which can roughly predict the convergence rate of a coordination scenario without in the absence og its solution. If the aforesaid prediction is less than a minimum, then one plans another pairings of assignments without wasting time on computing trajectories whose convergence rates are likely to be unfavorable.

## 8.2 Future Directions

### 8.2.1 Distributed Target-to-Astrobot Assignment

The spectroscopic projects which have been so far defined belong to the class of massive surveys. The number of targets in an observation corresponding to a massive survey roughly varies in the range of 500-5000, e.g., [Flaugher et al. 2014; R. Ellis et al. 2012; Kollmeier et al. 2017]. Optimal target assignment effectively manages the assignment process in the case of massive surveys. However, the next-generation giant surveys, whose white papers [D. J. Schlegel et al. 2019; Mandelbaum et al. 2019; Newman et al. 2019] have already been attracted much attention, include many thousands of targets and astrobots, say, around 20,000. In particular, optimal target assignment method may not be effective enough to be applied to giant surveys because of a variety of the reasons. First, optimal target assignment takes minimum coordination and maximum distribution requirements into account to perform the assignments. But in the case of extremely complex astrobot swarm of giant surveys, the cost function of the method may need to be revised to consider additional requirements. For example, if one or more astrobots are malfunctioned for any reason, then the coordination process may be adversely impacted in terms of the noticeable convergence rate loss of functional astrobots. This issue may be (at least partially) resolved by assessing other assignment options to minimize the negative influence of the cited malfunctions. To this aim, a key question is whether target assignment can be revised to be performed in a real-time manner. One may even imagine the development of adjustable focal planes which can be moved according to some degrees of freedom. So, in the case of such undesirable occurrences, trivial adjustment of focal planes may be helpful. Yet, the challenge would be the dynamics consideration of the focal plane's motions to update the set of the reachable astrobots associated with an observation before and after each adjustment.

Second, the very large number of targets and astrobots involved in giant surveys may nullify the quadratic-computational-complexity performance of optimal target assignment method. Namely given a particular target, optimal target assignment algorithm iterates over all of the unallocated astrobots reaching that target. However in the case of giant surveys, these reachable sets may be too large to be iterated in one piece. As another related concern, some batches of targets may be concentrated on a relatively small projection area of a focal plane. So, the reachable set associated with those targets may highly overlap each other. Traversing very large reachable sets also decelerates the functionality of optimal target assignment scheme. Hence, it would be beneficial to study the possibility of localized optimizations using heuristic partitioning of both targets and astrobots sets.

In the sections below, we focus on the related research venues which eventually give rise to more computationally-efficient coordination and completeness prediction methods.

### 8.2.2 Finite State Machine Reduction

Section 5.3 of Chapter 5 introduced a cooperative controller [Macktoobian et al. 2019a; Macktoobian et al. 2019d] which successfully governs complete coordination. The hybrid aspect of the controller clearly states the participation of a decision-making FSM in the controller formulation. As we already asserted, this FSM is taken into account as a utility to cover the deadlock situations which cannot be resolved by the planned CAPF. Despite the usefulness of the FSM, its presence adds another nonlinear layer to the overall controller architecture. This layer indeed hides the fairly intuitive mapping which corresponds a CAPF to its generated control law. We already noted that there is no formal verification method to assess the credibility of the generated control signals. Thereby, the more layers a controller has, the more simulation scenarios have to be run to assure the controller's fine-tuned behavior. A key question is how the CAPF formulation (5.14) can be revised so that it can also cover the functionality of the FSM.

The improvement of the verifiable supervisory solution [Macktoobian et al. 2019c] may also be the subject of active research. For instance, the current scheme formulates the discrete-event model of an astrobot as a one-degree-of-freedom artifact. This assumption implies that each astrobot has only one arm whose length is the overall length of the arms of an actual astrobot. The size of the discrete-event model of such a reduced astrobot is efficiently small to make synthesis computations fairly tractable. However, the elimination of one degree of freedom in fact neglects many potential coordination solutions which require the independent motions of both arms. On the other hand, naive formulation of both two degrees of freedom gives rise to the curse of dimensionality issue in view of the state size of the overall system. So, one may provide a guideline as to efficiently encode the kinematics of both two degrees of freedom to discrete-event models. Astrobots coordination may also encompass temporal requirements. For example, the deadlock scenarios can be more efficiently handled by a timer. In particular, once a deadlock occurs, a timer starts to tick. Then as long as the timer has not yet run out, the CAPF continues manipulating the velocity profiles of the parties involved in the deadlock case. The overflow of the timer would imply that the CAPF cannot resolve the issue, so the FSM will instead be in charge of the deadlock handling. Owing to the elegant formulation of timed discrete-event systems, one important area of further study is modeling astrobots using these temporal structures [Macktoobian et al. 2021a].

### 8.2.3 Optimal Coordination

The supervisory coordination strategy clearly exhibited how an astrobots coordination problem can be discretized with respect to the step-wise motions its astrobot's arms. We also studied the critical role of the forcible backtracking to find complete solutions. In view of the supervisory coordination, given a particular astrobot, all of the event paths which fulfill both the safety and the completeness requirements are the same. But one is logically more interested in the shorter solutions whose execution times are faster. Thus, optimized trajectory planning through the discrete motion space of astrobots is of utmost interest. The temporal efficiency of a path or its likelihood not to end up with a deadlock scenario can be reflected by a weight. Then, the whole graph of trajectories with respect to an initial configuration may be traversed to reach a target configuration using dynamic programming. The challenge would be encoding the whole motion space of an astrobot swarm to an efficient graph structure. A potential candidate of such structures may be multi-dimensional graphs [C. Chen et al. 2009]. Namely at each backtracking step, the sub-graph of each astrobot is backtracked one node while the neighboring astrobots can observe each other's sub-graph. Finally, once all agents ideally reach their targets spots, the optimal complete solution coordination of the swarm is the set of parallel and/or sequential steps which had been backtracked in the problem's sub-graphs. Effective pruning strategies may also be useful for the early detection and deletion of undesired paths before their involvement in the backtracking process. As another point, one observes that our proposed method is straightforward to check the complete coordination feasibility for 1 DoF astrobots. However, the extensive discretization of the space motions of astrobots may give rise to intractably large discrete-event systems. A future stream of research would be the usage of state tree structures to efficiently coordinate extremely complex systems of astrobots. Furthermore, untimed discrete-event systems do not consider any temporal requirements or constraints in the course of modeling and control process. Thus, one may take timed discrete-event systems into account to model astrobots not to skip potential important temporal characteristics of the system.

Astrobots and focal plane comprise many numerical parameters which specify their mechanical structures. These parameters are kept in particular configuration files which are used in the artificial potential computations in cooperative controllers (see, Section 5.3). The impact of each parameter on the cited potential calculations is not obvious. So, if one changes one parameter, there is not a direct way, except intensive simulations, to observe how the change influences a coordination. Accordingly, each parameter may be taken as a constraint, or objective, into account in view of coordination. Then, the constraints can be directly applied to a model predictive controller. Constraints have to be defined in terms of the state variables of an intended system. In the case of astrobot swarms, the state variables are position, velocity, and deviation. Yet, not all of potential constraints can be directly written in terms of those state variables. On this account, the derivation of a general formalism to automatic constraint generation for an astrobot swarm is a pivotal step to develop model-predictive-based solutions to the coordination problem. Additionally, collision avoidance and completeness have to be cast into cost functions [Macktoobian et al. 2021a].

### 8.2.4 Completeness Prediction

This research indeed takes only necessary positional features of swarms to predicate convergences. However, it is imperative to look for extra features which obtain better accuracies such as parity, i.e., the motion direction of an astrobot. The current problem statement only considers the results of a coordination as the label of that coordination. It implies that our algorithm only predicts the feasibility of complete coordination, regardless of the real-time and the control challenges one may face to achieve them. So, a challenge would be adding extra features to the label class. In that case, our algorithm may be extended to also predict the efficiency of trajectories which may lead to complete coordination. Needless to say that such feature expansion jeopardizes the computational efficiency of the prediction process as a trade-off. One may also utilize neural networks to train predictors which may provide more accurate results. However, neural networks include many hyperparameters whose proper setting may be challenging specially if one would like to avoid computationally intensive grid searches.

An unexplored venue to potentially further improve our results would be the usage of convolutional neural networks (CNN). The existence of hidden layers may provide novel ideas to perceive more information about the intermediate coordination steps from an initial configuration to a final one by exclusively owning these two configurations. However, CNN-driven designs are often less intuitive than the designs based on more geometrical approaches such as $k$-NN and SVM algorithms. Specially, planning the number of layers and the design of convolution computation and pooling operations are the challenges which have to be overcome [Macktoobian et al. 2021a].

# Chapter 9

# Appendix

## The simulator specification[1]

The software architecture of the simulator coordinator is illustrated in Figure 9.1.

- **util.py** This file contains a collection of shared functions and utilities.
- **plotting.py** This file contains a simple plotting interface, which uses a browser with SVG to present a plot of points represented as either complex numbers or 2-vectors.
- **path_generator.py** This file generates trajectories based on specifications of robotic positioners and targets.
- **pa_interface.py** This file has to be run if one needs to with the graphical user interface of the software.
- **pa_dnf** This file includes the decision-making logic based on artificial potential fields and finite state machines.
- **pa_animate** This file includes utility functionalities for animation outputs of the software.
- **fps_shared** This file is prototype module containing functions shared between the observation preparation software and the navigation software.
- **call_path_analysis** This file, as a wrapper, loads configuration and target files.

The I/O files of the software are introduced below.

- **configuration.cfg** This file includes all the information corresponding to the mechanical specifications of robotic positioners and their focal plane.
- **target.txt** This file includes all the information corresponding to the targets to be observed.
- **trajectories.yaml** This file stores generated trajectories.

Here is the specification of the parameters which have to be set as a part of the configuration file of the coordinator.

---

[1] The software tools developed and used in this research can be found here: https://github.com/Matin-Macktoobian.

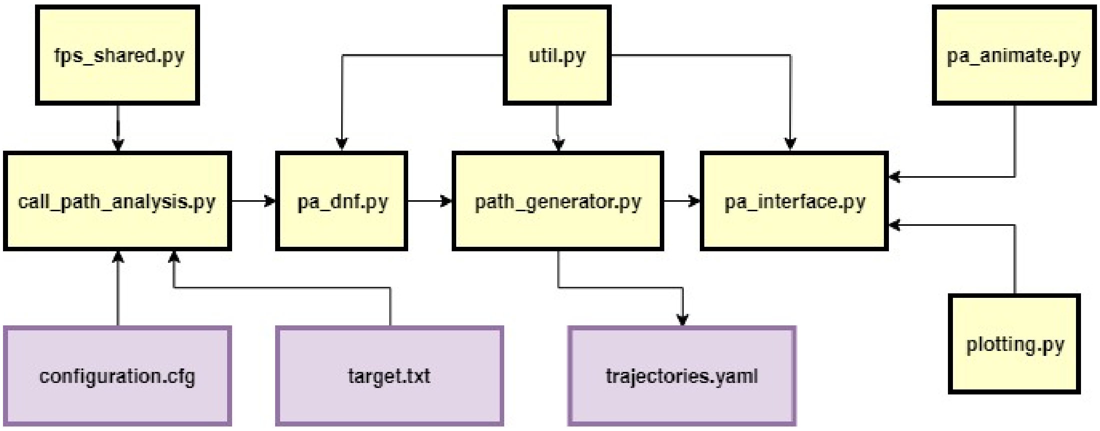


Figure 9.1: The software architecture of the coordinator

•**INS.POS.LENGTH1** This parameter specifies the length of positioner $\alpha$ arms in millimeters.

•**INS.POS.LENGTH2** This parameter specifies the length of positioner $\beta$ arms in millimeters.

•**INS.POS.WIDTH1** This parameter specifies the width of positioner $\alpha$ arms in millimeters.

•**INS.POS.WIDTH2** This parameter specifies the width of positioner $\beta$ arms in millimeters.

•**INS.POS.LOW1** This parameter specifies the lower travel limit of positioner $\alpha$ arm in degrees.

•**INS.POS.HIGH1** This parameter specifies the upper travel limit of positioner $\alpha$ arm in degrees.

•**INS.POS.LOW2** This parameter specifies the lower travel limit of positioner $\beta$ arm in degrees.

•**INS.POS.HIGH2** This parameter specifies the upper travel limit of positioner $\beta$ arm in degrees.

•**INS.POS.MINDIST** This parameter specifies the closest fiber approach distance in millimeters.

•**INS.POS.TOLER** This parameter specifies the fiber positioning tolerance in micrometers.

•**INS.POS.CURVRAD** This parameter specifies the radius of curvature of focal plane in millimeters.

Astrobot avoidance zones consist of:

- a circle of diameter **INS.POS.MINDIST** centred at the fiber of the $n$th robotic positioner;
- a rectangle of length INS.POS.B1 and width **INS.POS.WIDTH2**;
- a small triangle of length **INS.POS.TB2** and width **INS.POS.TW1**;
- a large triangle of length **INS.POS.TB3** and width **INS.POS.TW2**;
- a datum actuator switch of length **INS.POS.DL** and width **INS.POS.DW**;

specifically

•**INS.POS.B1** This parameter specifies the length $b1$ of collision avoidance rectangle in millimeters.

•**INS.POS.TB2** This parameter specifies the length $tb2$ of collision avoidance rectangle in millimeters.

- **INS.POS.TB3** This parameter specifies the length $tb3$ of collision avoidance rectangle in millimeters.
- **INS.POS.TW1** This parameter specifies the width $tw1$ of collision avoidance rectangle in millimeters.
- **INS.POS.TW2** This parameter specifies the width $tw2$ of collision avoidance rectangle in millimeters.
- **INS.POS.DL** This parameter specifies the length $d1$ of datum actuator in millimeters.
- **INS.POS.DW** This parameter specifies the width $2 \times d2$ of datum actuator in millimeters.
- **INS.POS.SAFETY** safety tolerance applied to avoidance zones in millimeters.

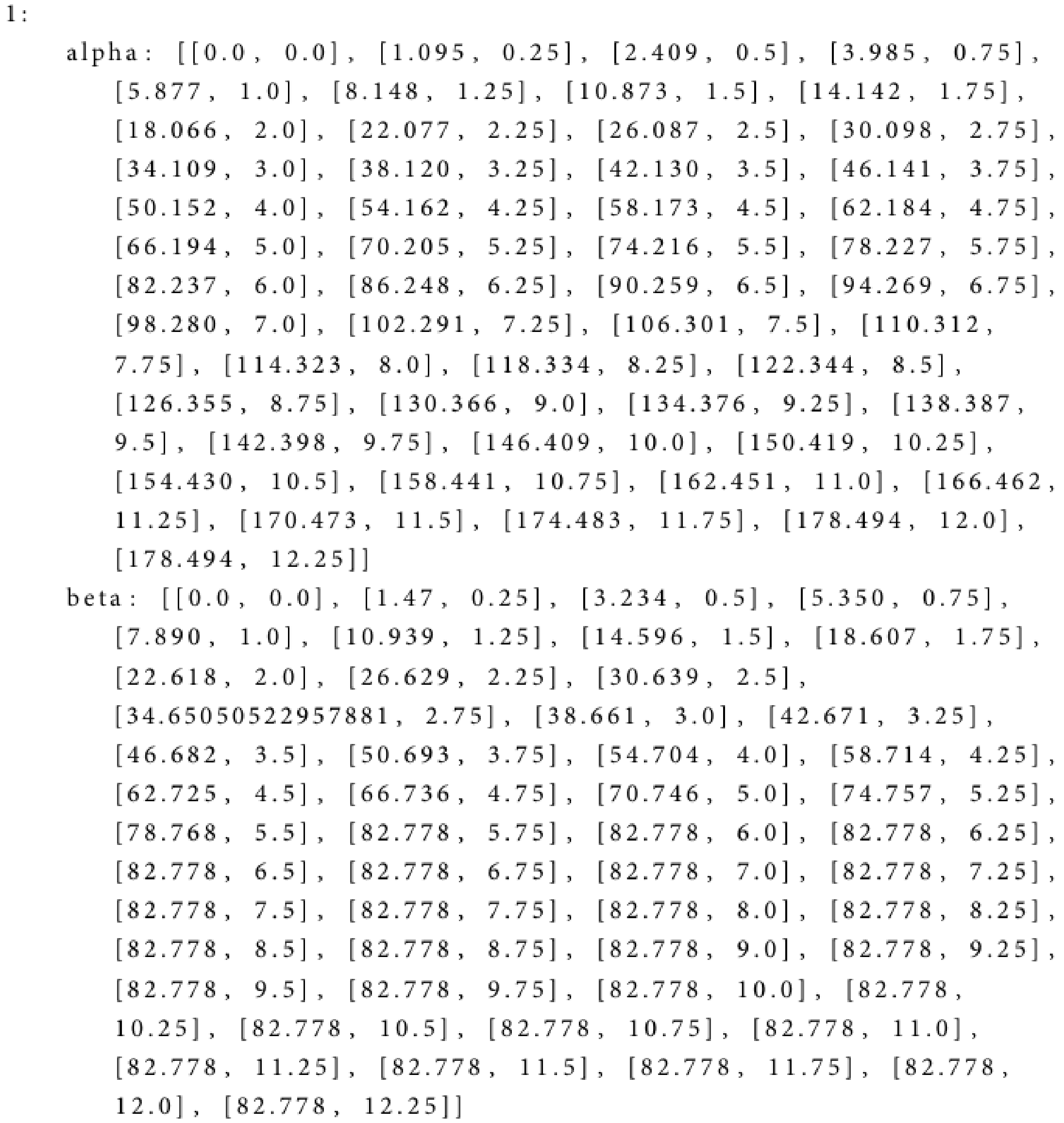
The following parameters have to be set for the $n$th acquisition camera corresponding to the grid of robotic positioners.

- **INS.ACQ$n$.RFOCAL** This parameter specifies the coordinate of the center of the $n$th acquisition camera in millimeters.
- **INS.ACQ$n$.THFOCAL** This parameter specifies the $\theta$ coordinate of the center of the $n$th acquisition camera in millimeters.
- **INS.ACQ$n$.FOV** This parameter specifies the field of view of the $n$th acquisition camera in millimeters.
- **INS.ACQ$n$.COLUMN** This parameter specifies the column number of the $n$th acquisition camera.
- **INS.ACQ$n$.ROW** This parameter specifies the raw number of the $n$th acquisition camera.

This coordinator uses a parity mechanism to indicate how a robotic arm may reach one particular target position with different configurations depending on its degrees of freedom. Since a robotic positioner has only two degrees of freedom , it has the maximum of two configurations by which the target position: either by clockwise (Arg1 = 0, Arg2 = 1) or counter-clockwise (Arg1 = 1, Arg2 = 0) motions. Thus, parity is given by a pair (Arg1, Arg2). The parity shall be indicated in the target file as explained below.

The target file, i.e., **target.txt**, corresponding to a particular observation has to be loaded using the window of the graphical user interface, as well. The $n$th row of this file corresponds to the $n$th robotic positioner which has already been assigned to a particular target. The coordinates are based on $(r, \theta)$ polar coordination system on a focal plane. Given a hypotheical row $i$, the columns of the file are specified as follows.

- **The 1st column** The $r$ parameter corresponding to the $i$th positioner.
- **The 2nd column** The $\theta$ parameter corresponding to the $i$th positioner.
- **The 3rd column** The $r$ parameter corresponding to the target.
- **The 4th column** The $\theta$ parameter corresponding to the target.
- **The parity argument 1** The Arg1 value
- **The parity argument 2** The Arg2 value
- **The priority positioner** This parameter is 0 (resp. 1) if the observation priority of the target is low (resp. high).



```yaml
1:
    alpha: [[0.0, 0.0], [1.095, 0.25], [2.409, 0.5], [3.985, 0.75],
       [5.877, 1.0], [8.148, 1.25], [10.873, 1.5], [14.142, 1.75],
       [18.066, 2.0], [22.077, 2.25], [26.087, 2.5], [30.098, 2.75],
       [34.109, 3.0], [38.120, 3.25], [42.130, 3.5], [46.141, 3.75],
       [50.152, 4.0], [54.162, 4.25], [58.173, 4.5], [62.184, 4.75],
       [66.194, 5.0], [70.205, 5.25], [74.216, 5.5], [78.227, 5.75],
       [82.237, 6.0], [86.248, 6.25], [90.259, 6.5], [94.269, 6.75],
       [98.280, 7.0], [102.291, 7.25], [106.301, 7.5], [110.312,
       7.75], [114.323, 8.0], [118.334, 8.25], [122.344, 8.5],
       [126.355, 8.75], [130.366, 9.0], [134.376, 9.25], [138.387,
       9.5], [142.398, 9.75], [146.409, 10.0], [150.419, 10.25],
       [154.430, 10.5], [158.441, 10.75], [162.451, 11.0], [166.462,
       11.25], [170.473, 11.5], [174.483, 11.75], [178.494, 12.0],
       [178.494, 12.25]]
    beta: [[0.0, 0.0], [1.47, 0.25], [3.234, 0.5], [5.350, 0.75],
       [7.890, 1.0], [10.939, 1.25], [14.596, 1.5], [18.607, 1.75],
       [22.618, 2.0], [26.629, 2.25], [30.639, 2.5],
       [34.65050522957881, 2.75], [38.661, 3.0], [42.671, 3.25],
       [46.682, 3.5], [50.693, 3.75], [54.704, 4.0], [58.714, 4.25],
       [62.725, 4.5], [66.736, 4.75], [70.746, 5.0], [74.757, 5.25],
       [78.768, 5.5], [82.778, 5.75], [82.778, 6.0], [82.778, 6.25],
       [82.778, 6.5], [82.778, 6.75], [82.778, 7.0], [82.778, 7.25],
       [82.778, 7.5], [82.778, 7.75], [82.778, 8.0], [82.778, 8.25],
       [82.778, 8.5], [82.778, 8.75], [82.778, 9.0], [82.778, 9.25],
       [82.778, 9.5], [82.778, 9.75], [82.778, 10.0], [82.778,
       10.25], [82.778, 10.5], [82.778, 10.75], [82.778, 11.0],
       [82.778, 11.25], [82.778, 11.5], [82.778, 11.75], [82.778,
       12.0], [82.778, 12.25]]
```

Figure 9.2: A typical YAML file in which velocities are in radian per second and time stamp is in second.

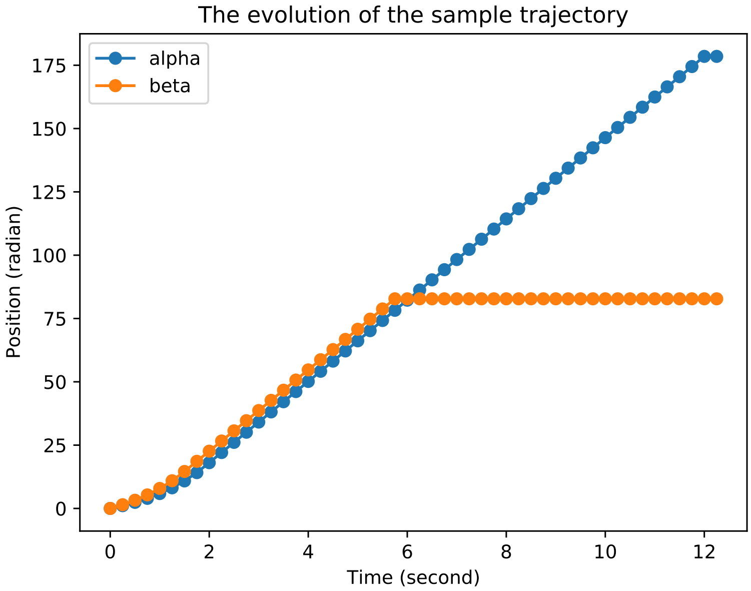


Figure 9.3: The trajectory evolution of the sample YAML file

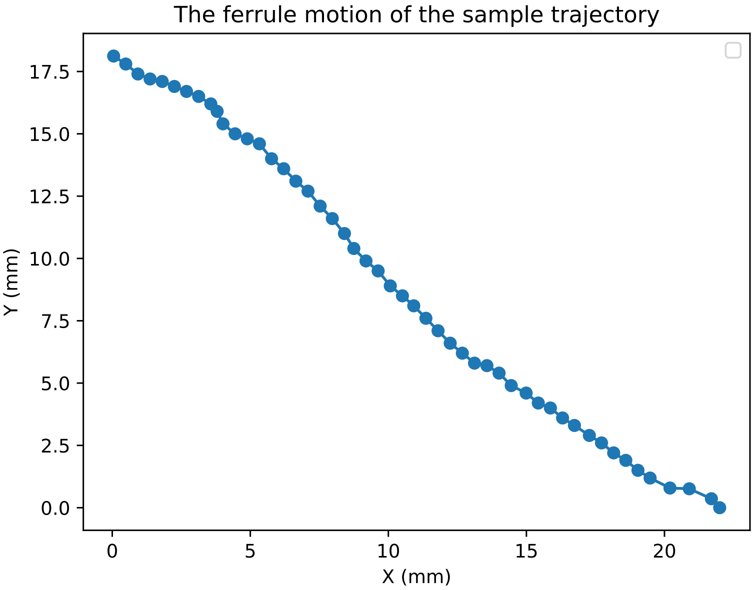


Figure 9.4: The ferrule motion of the sample trajectory (Here motion starts from the fully-folded formation at (22,0), then it gradually converges to its target in the left-hand-side of the plot.)

# References

Aghamousa, Amir et al. (2016a). “The DESI Experiment Part I: Science, Targeting, and Survey Design”. In: *arXiv preprint arXiv:1611.00036*.

Aghamousa, Amir et al. (2016b). “The DESI Experiment Part II: Instrument Design”. In: *arXiv preprint arXiv:1611.00037*.

Alam, Shadab et al. (2015). “The eleventh and twelfth data releases of the Sloan Digital Sky Survey: final data from SDSS-III”. In: *The Astrophysical Journal Supplement Series* 219.1, p. 12.

Anderson, Scott F et al. (2001). “High-redshift quasars found in sloan digital sky survey commissioning data. VI. Sloan digital sky survey spectrograph observations”. In: *The Astronomical Journal* 122.2, p. 503.

Araujo, Ricardo et al. (2020). “Design of a Theta/Phi fiber positioner robot for the Sloan Digital Sky Survey V”. In: *Proc. of SPIE Vol.* Vol. 11447, pp. 1144790–1.

Balachandran, V (1976). “An integer generalized transportation model for optimal job assignment in computer networks”. In: *Operations Research* 24.4, pp. 742–759.

Baumann, Frank, Christoph Buchheim, and Anna Ilyina (2014). “Lagrangean decomposition for mean-variance combinatorial optimization”. In: *International Symposium on Combinatorial Optimization*. Springer, pp. 62–74.

Bautista, Julian E et al. (2018). “The SDSS-IV extended baryon oscillation spectroscopic survey: baryon acoustic oscillations at redshift of 0.72 with the DR14 luminous red galaxy sample”. In: *The Astrophysical Journal* 863.1, p. 110.

Blake, Chris et al. (2011). “The WiggleZ Dark Energy Survey: the growth rate of cosmic structure since redshift z= 0.9”. In: *Monthly Notices of the Royal Astronomical Society* 415.3, pp. 2876–2891.

Blanton, Michael R et al. (2003). “An efficient targeting strategy for multiobject spectrograph surveys: The Sloan Digital Sky Survey “tiling” algorithm”. In: *The Astronomical Journal* 125.4, p. 2276.

Bundy, Kevin et al. (2014). “Overview of the SDSS-IV MaNGA survey: mapping nearby galaxies at Apache Point observatory”. In: *The Astrophysical Journal* 798.1, p. 7.

Cantor, David G and Mario Gerla (1974). “Optimal routing in a packet-switched computer network”. In: *IEEE Transactions on computers* 100.10, pp. 1062–1069.

Cassandras, Christos G and Stephane Lafortune (2009). *Introduction to discrete event systems*. Springer Science & Business Media.

Cataloluk, Hatice and Metin Kesler (2012). “A diagnostic software tool for skin diseases with basic and weighted K-NN”. In: *2012 international symposium on innovations in intelligent systems and applications*. IEEE, pp. 1–4.

Cattrysse, Dirk G and Luk N Van Wassenhove (1992). "A survey of algorithms for the generalized assignment problem". In: *European journal of operational research* 60.3, pp. 260–272.

Ceselli, Alberto and Giovanni Righini (2006). "A branch-and-price algorithm for the multilevel generalized assignment problem". In: *Operations Research* 54.6, pp. 1172–1184.

Chen, Chen et al. (2009). "Graph OLAP: a multi-dimensional framework for graph data analysis". In: *Knowledge and information systems* 21.1, pp. 41–63.

Chen, Jen-Hao and Kai-Tai Song (2018). "Collision-free motion planning for human-robot collaborative safety under cartesian constraint". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1–7.

Cirasuolo, M, MOONS Consortium, et al. (2016). "MOONS: A New Powerful Multi-Object Spectrograph for the VLT". In: *ASPC* 507, p. 109.

Cirasuolo, Michele et al. (2014). "MOONS: the Multi-Object Optical and Near-infrared Spectrograph for the VLT". In: *Ground-based and airborne instrumentation for astronomy V*. Vol. 9147. International Society for Optics and Photonics, 91470N.

Consortium, SDSS (n.d.). *SDSS Wiki*. URL: http://www.wiki.sdss.org.

Cui, Xiang-Qun et al. (2012). "The large sky area multi-object fiber spectroscopic telescope (LAMOST)". In: *Research in Astronomy and Astrophysics* 12.9, p. 1197.

Dawson, Kyle S et al. (2012). "The Baryon oscillation spectroscopic survey of SDSS-III". In: *The Astronomical Journal* 145.1, p. 10.

Dawson, Kyle S et al. (2016). "The SDSS-IV extended Baryon oscillation spectroscopic survey: Overview and early data". In: *The Astronomical Journal* 151.2, p. 44.

Dey, Arjun et al. (2019). "Overview of the DESI legacy imaging surveys". In: *The Astronomical Journal* 157.5, p. 168.

Doeleman, S, E Agol, D Backer, et al. (2009). "Astro2010: The Astronomy and Astrophysics Decadal Survey". In: *Science White Papers* 68.

Dubey, Harshit and Vikram Pudi (2013). "Class based weighted k-nearest neighbor over imbalance dataset". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, pp. 305–316.

Duca, Angelica Lo, Clara Bacciu, and Andrea Marchetti (2017). "A K-nearest neighbor classifier for ship route prediction". In: *OCEANS 2017-Aberdeen*. IEEE, pp. 1–6.

Dudziński, Krzysztof and Stanisław Walukiewicz (1987). "Exact methods for the knapsack problem and its generalizations". In: *European Journal of Operational Research* 28.1, pp. 3–21.

Ellis, Richard et al. (2012). *Extragalactic science and cosmology with the SUBARU Prime Focus Spectrograph (PFS)*. Tech. rep.

Escudero, Laureano F, María Araceli Garín, and Aitziber Unzueta (2016). "Cluster Lagrangean decomposition in multistage stochastic optimization". In: *Computers & Operations Research* 67, pp. 48–62.

Ferraro, Simone et al. (2019). "Inflation and Dark Energy from spectroscopy at $z > 2$". In: *arXiv preprint arXiv:1903.09208*.

Fisher, Marshall L (2004). "The Lagrangian relaxation method for solving integer programming problems". In: *Management science* 50.12_supplement, pp. 1861–1871.

Fisher, Marshall L and Ramchandran Jaikumar (1981). "A generalized assignment heuristic for vehicle routing". In: *Networks* 11.2, pp. 109–124.

Fisher, Marshall L, Ramchandran Jaikumar, and Luk N Van Wassenhove (1986). "A multiplier adjustment method for the generalized assignment problem". In: *Management science* 32.9, pp. 1095–1103.

Flaugher, Brenna and Chris Bebek (2014). "The dark energy spectroscopic instrument (desi)". In: *Ground-based and Airborne Instrumentation for Astronomy V*. Vol. 9147. International Society for Optics and Photonics, 91470S.

Forero–Romero, JE et al. (2009). "A dynamical classification of the cosmic web". In: *Monthly Notices of the Royal Astronomical Society* 396.3, pp. 1815–1824.

Gavish, Bezalel and Hasan Pirkul (1991). "Algorithms for the multi-resource generalized assignment problem". In: *Management science* 37.6, pp. 695–713.

Geoffrion, Arthur M (1974). "Lagrangean relaxation for integer programming". In: *Approaches to integer programming*. Springer, pp. 82–114.

Gunn, James E et al. (2006). "The 2.5 m telescope of the sloan digital sky survey". In: *The Astronomical Journal* 131.4, p. 2332.

Hamuy, Mario et al. (2006). "The Carnegie Supernova Project: The Low-Redshift Survey". In: *Publications of the Astronomical Society of the Pacific* 118.839, p. 2.

Han, Jiawei, Jian Pei, and Micheline Kamber (2011). *Data mining: concepts and techniques*. Elsevier.

Hofmann, Martin (2006). "Support vector machines-kernels and the kernel trick". In: *Notes* 26.3.

Hörler, Philipp (2018). *Robotic Fiber Positioning Systems for Massive Spectroscopic Surveys: Mechanical Design Guidelines and Technological Opportunities*. Tech. rep. EPFL.

Hörler, Philipp et al. (2018). "High density fiber postitioner system for massive spectroscopic surveys". In: *Monthly Notices of the Royal Astronomical Society* 481.3, pp. 3070–3082.

Hsu, Ping (1993). "Coordinated control of multiple manipulator systems". In: *IEEE Transactions on Robotics and Automation* 9.4, pp. 400–410.

Ivezić, Željko et al. (2008). "The milky way tomography with SDSS. II. Stellar metallicity". In: *The Astrophysical Journal* 684.1, p. 287.

Jain, Anil, Karthik Nandakumar, and Arun Ross (2005). “Score normalization in multimodal biometric systems”. In: *Pattern recognition* 38.12, pp. 2270–2285.

Jin, Wei (2019). “The Improvement of K-NN Classifier with GA-Based Weight-Tunning Method”. In: *Journal of Physics: Conference Series*. Vol. 1187. 4. IOP Publishing, p. 042032.

Jörnsten, Kurt and Mikael Näsberg (1986). “A new Lagrangian relaxation approach to the generalized assignment problem”. In: *European Journal of Operational Research* 27.3, pp. 313–323.

Joyce, Austin, Lucas Lombriser, and Fabian Schmidt (2016). “Dark energy versus modified gravity”. In: *Annual Review of Nuclear and Particle Science* 66, pp. 95–122.

Karuppiah, Ramkumar and Ignacio E Grossmann (2008). “A Lagrangean based branch-and-cut algorithm for global optimization of nonconvex mixed-integer nonlinear programs with decomposable structures”. In: *Journal of global optimization* 41.2, pp. 163–186.

Khaldi, Belkacem et al. (2018). “Self-organization in aggregating robot swarms: A DW-KNN topological approach”. In: *Biosystems* 165, pp. 106–121.

Kimura, Masahiko et al. (2010). “Fibre multi-object spectrograph (FMOS) for the Subaru telescope”. In: *Publications of the Astronomical Society of Japan* 62.5, pp. 1135–1147.

Klose, A (1999). “An LP-based heuristic for two-stage capacitated facility location problems”. In: *Journal of the Operational Research Society* 50.2, pp. 157–166.

Kollmeier, Juna A et al. (2017). “SDSS-V: Pioneering Panoptic Spectroscopy”. In: *arXiv preprint arXiv:1711.03234*.

Kronig, Luzius Gregor (2020). *Precision positioning of microrobots for multi-object spectrographs*. Tech. rep. EPFL.

Lee, Eun-Young et al. (2015). “Moving range k nearest neighbor queries with quality guarantee over uncertain moving objects”. In: *Information Sciences* 325, pp. 324–341.

Liu, Wei and Sanjay Chawla (2011). “Class confidence weighted knn algorithms for imbalanced data sets”. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, pp. 345–356.

Liu, Yan Y and Shaowen Wang (2015). “A scalable parallel genetic algorithm for the generalized assignment problem”. In: *Parallel computing* 46, pp. 98–119.

Macktoobian, Matin (2018). *Automatic Reconfiguration of Untimed and Timed Discrete-Event Systems*. M.A.Sc. thesis, The University of Toronto.

Macktoobian, Matin (n.d.). *k-NN astrobots convergence prediction dataset*. URL: https://git.io/JU4Y0.

Macktoobian, Matin, Denis Gillet, and J-P Kneib (2020a). “Optimal target assignment for massive spectroscopic surveys”. In: *Astronomy and Computing* 30, p. 100364.

Macktoobian, Matin, Denis Gillet, and Jean-Paul Kneib (2019a). “Complete coordination of robotic fiber positioners for massive spectroscopic surveys”. In: *Journal of Astronomical Telescopes, Instruments, and Systems* 5.4, p. 045002.

Macktoobian, Matin, Denis Gillet, and Jean-Paul Kneib (2019b). “Heterogeneous Target Assignment to Robotic Fiber Positioner Systems”. In: *2019 Australian & New Zealand Control Conference (ANZCC)*. IEEE, pp. 48–53.

Macktoobian, Matin, Denis Gillet, and Jean-Paul Kneib (2019c). “Supervisory Coordination of Robotic Fiber Positioners in Multi-Object Spectrographs”. In: *The Proceedings of the the 15th IFAC Symposium on Large Scale Systems (LSS): Theory and Applications*. CONF. Elsevier.

Macktoobian, Matin, Denis Gillet, and Jean-Paul Kneib (2019d). “The Navigation of Robotic Fiber Positioners in SDSS-V Project: Design and Implementation”. In: *2019 15th Conference on Ph. D Research in Microelectronics and Electronics (PRIME)*. IEEE, pp. 85–88.

Macktoobian, Matin, Denis Gillet, and Jean-Paul Kneib (2021a). “Astrobotics: swarm robotics for astrophysical studies”. In: *IEEE Robotics and Automation Magazine*. doi: 10.1109/MRA.2020.3044911.

Macktoobian, Matin and S Ali A Moosavian (2013). “Time-variant artificial potential fields: A new power-saving strategy for navigation of autonomous mobile robots”. In: *Proceedings of the 1st RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, pp. 121–127.

Macktoobian, Matin and Mahdi Aliyari Shoorehdeli (2016). “Time-variant artificial potential field (TAPF): a breakthrough in power-optimized motion planning of autonomous space mobile robots”. In: *Robotica* 34.5, pp. 1128–1150.

Macktoobian, Matin and W Murray Wonham (2017). “Automatic reconfiguration of untimed discrete-event systems”. In: *Electrical Engineering, Computing Science and Automatic Control (CCE), 2017 14th International Conference on*. IEEE, pp. 1–6.

Macktoobian, Matin et al. (2020b). “Data-driven convergence prediction of astrobots swarms”. In: preprint.

Macktoobian, Matin et al. (2020c). “Learning convergence prediction of astrobots in multi-object spectrographs”. In: preprint.

Macktoobian, Matin et al. (2021b). “Experimental evaluation of complete safe coordination of astrobots for the Sloan Digital Sky Survey V”. In: *Experimantal Astronomy* 51.1, pp. 77–94.

Makarem, Laleh (2015). *Decentralized multi-robot coordination in crowded workspaces*. Tech. rep. EPFL.

Makarem, Laleh, Jean-Paul Kneib, and Denis Gillet (2016). “Collision-free coordination of fiber positioners in multi-object spectrographs”. In: *Software and Cyberinfrastructure for Astronomy IV*. Vol. 9913. International Society for Optics and Photonics, p. 99130V.

Makarem, Laleh et al. (2014). “Collision avoidance in next-generation fiber positioner robotic systems for large survey spectrographs”. In: *Astronomy & Astrophysics* 566, A84.

Mandelbaum, Rachel et al. (2019). “Wide-field Multi-object Spectroscopy to Enhance Dark Energy Science from LSST”. In: *arXiv preprint arXiv:1903.09323*.

Martello, Silvano and Paolo Toth (1992). “Generalized assignment problems”. In: *International Symposium on Algorithms and Computation*. Springer, pp. 351–369.

Mathews, Michael B, Poul T Lomholt, and William A Littlewood (Dec. 2006). *Method and system for distributed navigation and automated guidance*. US Patent 7,149,625.

Mazets, EP et al. (1982). "Cosmic gamma-ray burst spectroscopy". In: *Astrophysics and Space Science* 82.2, pp. 261–282.

Mehrholz, Dieter et al. (2002). "Detecting, tracking and imaging space debris". In: *ESA Bulletin(0376-4265)* 109, pp. 128–134.

Meiring, Joseph D et al. (2011). "The first observations of low-redshift damped Ly$\alpha$ systems with the Cosmic Origins Spectrograph". In: *The Astrophysical Journal* 732.1, p. 35.

Menesi, Wail, Behrooz Golzarpoor, and Tarek Hegazy (2013). "Fast and near-optimum schedule optimization for large-scale projects". In: *Journal of Construction Engineering and Management* 139.9, pp. 1117–1124.

Milnor, J (1963). "Morse theory. Number 51 in Annals of Math". In: *Studies. Princeton University Press.*

Morales, Isaac et al. (2011). "Fibre assignment in next-generation wide-field spectrographs". In: *Monthly Notices of the Royal Astronomical Society* 419.2, pp. 1187–1196.

Morales, Isaac et al. (2012). "Fibre assignment in next-generation wide-field spectrographs". In: *Monthly Notices of the Royal Astronomical Society* 419.2, pp. 1187–1196.

Murphy, Kevin P (2012). *Machine learning: a probabilistic perspective*. MIT press.

Newman, Jeffrey A et al. (2019). "Deep Multi-object Spectroscopy to Enhance Dark Energy Science from LSST". In: *arXiv preprint arXiv:1903.09325*.

Peterson, Leif E (2009). "K-nearest neighbor". In: *Scholarpedia* 4.2, p. 1883.

Pirat, Camille et al. (2017). "Mission design and GNC for in-orbit demonstration of active debris removal technologies with CubeSats". In: *Acta Astronautica* 130, pp. 114–127.

Platt, John (1998). "Sequential minimal optimization: A fast algorithm for training support vector machines". In:

Pogge, Richard W et al. (2020). "A robotic Focal Plane System (FPS) for the Sloan Digital Sky Survey V". In: *Ground-based and Airborne Instrumentation for Astronomy VIII*. Vol. 11447. International Society for Optics and Photonics, p. 1144781.

Ramadge, Peter J and W Murray Wonham (1987). "Supervisory control of a class of discrete event processes". In: *SIAM journal on control and optimization* 25.1, pp. 206–230.

Roelofsen, Steven, Denis Gillet, and Alcherio Martinoli (2017). "Collision avoidance with limited field of view sensing: A velocity obstacle approach". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1922–1927.

Roelofsen, Steven, Alcherio Martinoli, and Denis Gillet (2016). "3D collision avoidance algorithm for Unmanned Aerial Vehicles with limited field of view constraints". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, pp. 2555–2560.

Ross, G Terry and Richard M Soland (1975). "A branch and bound algorithm for the generalized assignment problem". In: *Mathematical programming* 8.1, pp. 91–103.

Ross, G Terry and Richard M Soland (1977). "Modeling facility location problems as generalized assignment problems". In: *Management Science* 24.3, pp. 345–357.

Sanghavi, Sujay, Dmitry Malioutov, and Alan Willsky (2011). "Belief propagation and LP relaxation for weighted matching in general graphs". In: *IEEE Transactions on Information Theory* 57.4, pp. 2203–2212.

Sayres, Conor et al. (2020). "SDSS-V algorithms: fast, collision-free trajectory planning for heavily overlapping robotic fiber positioners". In: preprint.

Schaefer, Christoph ER, Laleh Makarem, and Jean-Paul Kneib (2016). "Target-based fiber assignment for large survey spectrographs". In: *Software and Cyberinfrastructure for Astronomy IV*. Vol. 9913. International Society for Optics and Photonics, p. 991335.

Schlegel, David J et al. (2019). "Astro2020 APC White Paper: The MegaMapper: az> 2 spectroscopic instrument for the study of Inflation and Dark Energy". In: *arXiv preprint arXiv:1907.11171*.

Schrabback, Tim et al. (2010). "Evidence of the accelerated expansion of the Universe from weak lensing tomography with COSMOS". In: *Astronomy & Astrophysics* 516, A63.

Scoccimarro, Roman (2004). "Redshift-space distortions, pairwise velocities, and nonlinearities". In: *Physical Review D* 70.8, p. 083007.

Seo, Hee-Jong and Daniel J Eisenstein (2003). "Probing dark energy with baryonic acoustic oscillations from future large galaxy redshift surveys". In: *The Astrophysical Journal* 598.2, p. 720.

Sethanan, Kanchana and Rapeepan Pitakaso (2016). "Improved differential evolution algorithms for solving generalized assignment problem". In: *Expert Systems with Applications* 45, pp. 450–459.

Smee, Stephen A et al. (2013). "The multi-object, fiber-fed spectrographs for the Sloan Digital Sky Survey and the Baryon Oscillation Spectroscopic Survey". In: *The Astronomical Journal* 146.2, p. 32.

Su, Ming-Yang (2011). "Using clustering to improve the KNN-based classifiers for online anomaly network traffic identification". In: *Journal of Network and Computer Applications* 34.2, pp. 722–730.

Suykens, Johan AK and Joos Vandewalle (1999). "Least squares support vector machine classifiers". In: *Neural processing letters* 9.3, pp. 293–300.

Takada, Masahiro et al. (2014). "Extragalactic science, cosmology, and Galactic archaeology with the Subaru Prime Focus Spectrograph". In: *Publications of the Astronomical Society of Japan* 66.1.

Takada, Masahiro, Eiichiro Komatsu, and Toshifumi Futamase (2006). “Cosmology with high-redshift galaxy survey: neutrino mass and inflation”. In: *Physical Review D* 73.8, p. 083520.

Tan, KT et al. (2013). “Coordinated control and energy management of distributed generation inverters in a microgrid”. In: *IEEE transactions on power delivery* 28.2, pp. 704–713.

Tao, Dominique et al. (2018). “Priority coordination of fiber positioners in multi-objects spectrographs”. In: *Ground-based and Airborne Instrumentation for Astronomy VII*. Vol. 10702. International Society for Optics and Photonics, 107028K.

Van Den Briel, Menkes et al. (2007). “An LP-based heuristic for optimal planning”. In: *International Conference on Principles and Practice of Constraint Programming*. Springer, pp. 651–665.

Wake, David A et al. (2017). “The SDSS-IV MaNGA sample: design, optimization, and usage considerations”. In: *The Astronomical Journal* 154.3, p. 86.

Way, MJ et al. (2005). “Redshifts in the Southern Abell Redshift Survey Clusters. I. The Data”. In: *The Astronomical Journal* 130.5, p. 2012.

Wonham, W Murray (2017a). *Design Software: TCT*. Univ. of Toronto.

Wonham, W Murray (2017b). *Supervisory Control of Discrete-Event Systems*. ECE Dept., Univ. of Toronto.

Xu, Qing-Song and Yi-Zeng Liang (2001). “Monte Carlo cross validation”. In: *Chemometrics and Intelligent Laboratory Systems* 56.1, pp. 1–11.

Zasowski, G et al. (2017). “Target selection for the SDSS-IV APOGEE-2 survey”. In: *The Astronomical Journal* 154.5, p. 198.

Zehavi, Idit et al. (2002). “Galaxy clustering in early Sloan Digital Sky Survey redshift data”. In: *The Astrophysical Journal* 571.1, p. 172.

Zhao, Gang et al. (2012). “LAMOST spectral survey—An overview”. In: *Research in Astronomy and Astrophysics* 12.7, p. 723.

# Matin Macktoobian

(Version date: February 15, 2021)

## Contact Information

🏠 Chemin des Triaudes 4, Ecublens VD, Switzerland
☎ +41-766345994
✉ matin.macktoobian@epfl.ch

## Research Interests

Social Robotics
Robot Learning
Cognitive Robotics
Human-Robot Interaction
Control Theory
Roboethics

## Education

- **Ph.D.** in Robotics, Control, and Intelligent Systems (11.2018 - 02.2021)
  **Swiss Federal Institute of Technology in Lausanne (EPFL)**, Lausanne, Switzerland
  Thesis: “Analytical and Machine Learning Methods for the Complete Safe Coordination of Astrobot Swarms”
  Advisor: Dr. Denis Gillet
  Co-advisor: Prof. Jean-Paul Kneib

- **M.A.Sc.** in Electrical and Computer Engineering - Systems Control (09.2016 - 08.2018)
  **University of Toronto**, Toronto, Canada
  Thesis: “Automatic Reconfiguration of Untimed and Timed Discrete-Event Systems”
  Advisor: Prof. W. Murray Wonham

- **B.Sc.** in Computer Engineering - Hardware Design (09.2009 - 08.2013)
  **K. N. Toosi University of Technology (KNTU)**, Tehran, Iran
  Thesis: “Smart Control of Time-Variant Artificial Potential Fields in Navigation of Space Rovers”
  Advisor: Prof. Mahdi Aliyari Sh.

## (Peer-Reviewed) Publications

### Journal Papers

8. **M. Macktoobian**, F. Basciani, D. Gillet, and J-P. Kneib, Data-Driven Convergence Prediction of Astrobot Swarms, *IEEE Transactions on Automation Science and Engineering*. (Under Review)

7. **M. Macktoobian**, F. Basciani, D. Gillet, and J-P. Kneib, Learning Convergence Prediction of Astrobots in Multi-Object Spectrographs, *Journal of Astronomical Telescopes, Instruments, and Systems*, 2021. (In Press)

6. **M. Macktoobian**, D. Gillet, and J-P. Kneib, Astrobotics: Swarm Robotics for Astrophysical Studies, *IEEE Robotics and Automation Magazine*, 2021.

5. **M. Macktoobian**, R. Araújo, L. Grossen, L. Kronig. M. Bouri, D. Gillet, and J-P. Kneib, Experimental Evaluation of Complete Safe Coordination of Astrobots for Sloan Digital Sky Survey V, *Experimental Astronomy* 51(1), pp. 77-94, 2021.

4. **M. Macktoobian**, D. Gillet, and J-P. Kneib, Optimal Target Assignment for Massive Spectroscopic Surveys, *Astronomy and Computing* 30, Article 100364, 2020.

3. **M. Macktoobian**, D. Gillet, and J-P. Kneib, Complete Coordination of Robotic Fiber Positioners for Massive Spectroscopic Surveys, *Journal of Astronomical Telescopes, Instruments, and Systems* 5(4), Article 045002, 2019.

2. **M. Macktoobian** and M. Aliyari Sh., Optimal Distributed Interconnectivity of Multi-Robot Systems by Spatially-Constrained Clustering, *Adaptive Behavior* 25(2), pp. 96-113, 2017.

1. **M. Macktoobian** and M. A. Shoorehdeli, Time-Variant Artificial Potential Field (TAPF): A Breakthrough in Power-Optimized Motion Planning of Autonomous Space Mobile Robots, *Robotica* 34(5), pp. 1128-1150, 2016.

## Proceeding Papers

15. R. W. Pogge, M. A. Derwent, T. P. O'Brien, C. A. Jurgenson, D. Pappalardo, M. Engelman, C. Brandon, J. Brady, N. Clawson, J. Shover, J. Mason, J-P. Kneib, R. Araújo, M. Bouri, L. Kronig, L. Grossen, D. Gillet, **M. Macktoobian**, S. E. Tuttle, E. Farr, J. Sánchez-Gallego, and C. Sayres, A Robotic Focal Plane System (FPS) for the Sloan Digital Sky Survey V, *Proc. of SPIE 11447, Ground-based and Airborne Instrumentation for Astronomy VIII*, Article 1144781, 2020.

14. R. Araújo, M. Bouri, C. Brandon, S. Caseiro, D. Gillet, L. Grossen, L. Kronig, **M. Macktoobian**, T. P. O'Brien, R. W. Pogge, S. E. Tuttle, and J-P. Kneib, Design of a Theta/Phi Fiber Positioner Robot for the Sloan Digital Sky Survey V, *Proc. of SPIE 11447, Ground-based and Airborne Instrumentation for Astronomy VIII*, Article 1144790, 2020.

13. **M. Macktoobian**, Reconfigurable Timed Discrete-Event Systems, *System Theory, Control and Computing (ICSTCC), 24th Intl. Conf. on*, IEEE, pp. 316-321, 2020. (Presented by **M. Macktoobian** in an oral communication)

12. **M. Macktoobian**, D. Gillet, and J-P. Kneib, Heterogeneous Target Assignment to Robotic Fiber Positioner Systems, *Australian and New Zealand Control Conf. (ANZCC)*, IFAC/IEEE, pp. 48-53, 2019. (Presented by **M. Macktoobian** in an oral communication)

11. **M. Macktoobian**, D. Gillet, and J-P. Kneib, The Navigation of Robotic Fiber Positioners in SDSS-V Project: Design and Implementation, *PhD Research in Microelectronics and Electronics (PRIME), 15th Conf. on*, pp. 85-88, IEEE, 2019. (Presented by **M. Macktoobian** in an oral communication)

10. **M. Macktoobian**, D. Gillet, and J-P. Kneib, Supervisory Coordination of Robotic Fiber Positioners in Multi-Object Spectrographs, *Large Scale Systems (LSS): Theory and Applications, 15th IFAC Symp. on*, pp. 61-66, 2019. (Presented by **M. Macktoobian** in an oral communication)

9. A. K. N. Tehrani, **M. Macktoobian**, and Sh. Kasaei, Color Image Segmentation Using a Fuzzy Inference System, *Digital Information Processing and Communications (ICDIPC), 7th Intl. Conf. on*, IEEE, pp. 78-83, 2019.

8. **M. Macktoobian** and W. M. Wonham, Automatic Reconfiguration of Untimed Discrete-Event Systems, *Electrical Engineering, Computing Science, and Automatic Control (CCE), 14th Intl. Conf. on*, IEEE, 2017. (Presented by **M. Macktoobian** in an oral communication)

7. **M. Macktoobian**, A. K. N. Tehrani, and M. Khodayar, Morphological Reconfiguration Monitoring for Homogeneous Self-Reconfigurable Robots, *Electrical Engineering, Computing Science, and Automatic Control (CCE), 14th Intl. Conf. on*, IEEE, 2017. (Presented by **M. Macktoobian** in an oral communication)

6. **M. Macktoobian** and A. KhatamiNejad T., On the Formal Development of Behaviorally-Evolved Braitenberg Vehicles: A Methodological Approach, *5th Basic and Clinical Neuroscience Congress (BCNC)*, pp. 493-498, 2016. (Presented by **M. Macktoobian** in an oral communication)

5. **M. Macktoobian**, M. Jafari, and E. Attarzadeh Gh., Applied Neural Cross Correlation into the Curved Trajectory Detection Process for Braitenberg Vehicles, *3rd Basic and Clinical Neuroscience Congress (BCNC)*, pp. 412-418, 2014. (Presented by **M. Macktoobian** in an oral communication)

4. **M. Macktoobian**, Curved Trajectory Detection: A Novel Neurocognitive Perception Approach for Autonomous Smart Robots, *2nd Basic and Clinical Neuroscience Congress (BCNC)*, pp. 467-473, 2013. (Presented by **M. Macktoobian** in an oral communication)

3. **M. Macktoobian**, Bidirectional Motion Detection: A Neural Intelligent Model for Perception of Cognitive Robots, *2nd Basic and Clinical Neuroscience Congress (BCNC)*, pp. 476-482, 2013. (Presented by **M. Macktoobian** in an oral communication)

2. **M. Macktoobian**, Smart Navigation of Autonomous Mobile Robots by Time-Variant Artificial Potential Fields: A Fuzzy Approach, *Fuzzy Systems (IFSC),13th Iranian Conf. on*, IEEE, pp. 1-6, 2013. (Presented by **M. Macktoobian** in an oral communication)

1. **M. Macktoobian** and S. Ali. A. Moosavian, Time-Variant Artificial Potential Fields: A New Power-Saving Strategy for Navigation of Space Rovers, *Robotics and Mechatronics (ICRoM),1st RSI/ISM Intl. Conf. on*, IEEE, pp.,121-127, 2013. (Presented by **M. Macktoobian** in an oral communication)

## Honors and Awards

- **Inaugural Swiss Robotics PhD Student Essay Award** winner, NCCR Robotics, 2020
- **The Gold Leaf Award** winner, The 15th Conf. on PhD Research in Microelectronics and Electronics (PRIME), 2019
- **The IFAC Young Author Award** finalist, The 15th IFAC Symposium on Large Scale Complex Systems: Theory and Applications (LSS), 2019
- **Graduate Fellowship** recipient, University of Toronto, 2016 - 2018
- **School of Graduate Studies Conference Grant** recipient, University of Toronto, 2017
- **Connaught International Scholarship** recipient, University of Toronto, 2016-2017
- **The Best B.Sc. Thesis Award** finalist, IEEE Iran Section, 2013

## Academic Experience

| | |
|---|---|
| Master Thesis Supervision | "Machine-Learning-Based Techniques for the Complete Coordination Prediction of Astrobots Swarms", Francesco Basciani, EPFL, 2020 |
| Teacher Assistantship | Multivariable Control and Coordination Systems, EPFL, 2018-2021<br>Robotics Practicals, EPFL, 2019<br>Linear Algebra, University of Toronto, 2017<br>Robot Modeling and Control, University of Toronto, 2017<br>Design of VLSI Systems, K. N. Toosi University of Technology, 2013 |
| Paper Review | IEEE Intl. Conf. on Systems, Man, and Cybernetics, 2019 - Present<br>IEEE Control Systems Letters, 2019 - Present<br>IFAC Symposium on Large Scale Complex Systems: Theory and Applications, 2019 - Present<br>IEEE Transactions on Automatic Control, 2018 - Present<br>IEEE Transactions on Industrial Electronics, 2016 - Present<br>IMA Journal of Mathematical Control and Information, 2015 - Present<br>IEEE Intl. Conf. on Robotics and Mechatronics, 2015 - Present<br>Robotica, 2012 - Present |

## Languages

Persian: Native
English: Fluent
German: Reading knowledge
French: Reading knowledge

## Computer Skills

Programming: Python, C#, LaTeX
Data Science: Tensorflow, sci-kit learn
Modeling: Adobe Illustrator, Adobe Photoshop, AutoCad

## Extracurricular Activities

- Gradlife advisory committee member, University of Toronto, 08.2017 - 06.2018
- The honorable winner of the "People and Places" category in the 8th annual photography contest, Center for International Experiences, University of Toronto, 05.2017