Estimation d'état distribuée sous contraintes pour une mission de surveillance multi-capteurs multi-robots

Antonello Venturino

université
PARIS-SACLAY

Constrained distributed state estimation for surveillance missions using multi-sensor multi-robot systems

Estimation d'état distribuée sous contraintes pour une mission de surveillance multi-capteurs multi-robots

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n°580 : Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Automatique
Graduate School : Sciences de l'Ingénierie et des Systèmes, Référent : CentraleSupélec

Thèse préparée dans les unités de recherche Laboratoire des Signaux et Systèmes (Université Paris-Saclay, CNRS, CentraleSupélec), Traitement de l'Information et Systèmes (Université Paris-Saclay, ONERA) et Département de *Ingeniería de Sistemas y Automática* (Université de Séville) sous la direction de Cristina STOICA MANIU, professeur, le co-encadrement de Sylvain BERTRAND, maître de recherche, le co-encadrement de Teodoro ALAMO, professeur et le co-encadrement d'Eduardo F. CAMACHO, professeur

Thèse soutenue à Paris-Saclay, le 22 septembre 2022, par

Antonello VENTURINO

Composition du jury

Mohammed CHADLI Professeur, Université d'Evry-Val-d'Essone - Université Paris-Saclay	Président
John Jairo MARTINEZ MOLINA Professeur, INP Grenoble	Rapporteur, Examinateur
Vicenç PUIG Professeur, Universitat Politècnica de Catalunya, Espagne	Rapporteur, Examinateur
Dan SELIŞTEANU Professeur, Universitatea Craiova, Roumanie	Rapporteur, Examinateur
Estelle COURTIAL Maître de Conférence, Université d'Orléans	Examinatrice
Cristina STOICA MANIU Professeur, CentraleSupélec/L2S - Université Paris-Saclay	Directrice de thèse

THESE DE DOCTORAT

NNT : 2022UPAST118

"...e quindi uscimmo a riveder le stelle."
(Dante Alighieri - Divina Commedia 'Inferno' XXXIV, 139)

"...et dès lors, nous sortîmes revoir les étoiles."
(Dante Alighieri - Divine Comédie 'Enfer' XXXIV, 139)

"...and thence we came forth to see again the stars."
(Dante Alighieri - Divine Comedy 'Inferno' XXXIV, 139)

"...y así salimos a ver de nuevo las estrellas."
(Dante Alighieri - Divina Comedia 'Infierno' XXXIV, 139)

Acknowledgements

These three years as PhD student would not have been the same without sharing them with most of the people I met during this doctoral journey. So, let me thank them all.

My particular gratitude goes to the members of my thesis committee. I would like to thank John Martinez, Vicenç Puig and Dan Selisteanu for accepting to review my manuscript. I also thank Estelle Courtial for accepting to be part of the committee and Mohammed Chadli for presiding it. I think the questions and remarks the entire PhD committee had are a valuable contribution for this thesis and maybe for my future works. Thank you.

I would like to acknowledge my supervisors Cristina and Sylvain for being always supportive and for their trust. I think they are the best supervisors one can have, so I feel very lucky to have worked with you. I remember every time I had to do something which I tough could not make it for short of time or difficulties, I always went to Cristina saying "I can't do this" and she was "yes, you can", "no, I can't", "yes, you can". "OK" she always trusted me, I think this is very encouraging. Thank you.

I am also grateful to Teo and Eduardo, my two supervisors from University of Seville in Spain, for their innumerable ideas on how to improve my work and the scientific insight which we discussed together on our meetings. And of course for welcoming me for an exchange period in Seville, which also gave me the opportunity to learn Spanish. So, *muchas gracias*!

This doctoral journey was a great experience also because I met very good colleagues and friends. There are really a lot of people to thank here, starting from my office mates. Julius, the first person I met here in France and that comes to be my office mate at ONERA. Then, the office mates from L2S. So, I would like to thank Joy for trying always to spread out positive vibes and unsuccessfully to keeping me straight and narrow, Nihed for always trying to find something to argue about and Song Li for taking my side even I was wrong, that is what good friends do. They are the best office mates in the world. Thank you all.

From the lab, I would like to thank Thomas for having always the answer to all the questions I had, Vincent for making me realize that my French was never good enough, he was always speaking too fast and I could not catch it, Dario for proving me that North Italy is not actually Italy, Dora for telling me all the time that I am not that young, Mahdi for all the stimulating discussion during lunch time, Aarsh for sharing with me all the troubles on setting up the experiments in the flight arena, Miguel for the football match that we never did, and also to Martin, Matthieu, Benjamin, Baptiste, Nouha, Hung, Hanane, Juan Pablo, Elio, Diego, Guillaume and Andrei with whom I spent less time but were always there to participate in any discussions or activities. Thanks you all, it was a very pleasant adventure.

Well, since the next acknowledgments will be more international, outside the research community, I would like to switch languages.

Quindi, i prossimi ringraziamenti vanno alla mia famiglia, in particolare ai miei genitori e ai miei fratelli per essere venuti fin qui ad assistere alla discussione della tesi. Ai miei genitori voglio dire che questo traguardo è anche merito vostro, perché mi avete sempre lasciato che esprimessi al meglio le mie capacità senza nessun vincolo e questa libertà alla fine mi ha portato fin qui, quindi grazie!

También tengo que agradecer a mi novia Diana por venir hasta aqui a apoyarme y ayudarme en este último período. Gracias a ti, los momentos más estresantes fueron mucho más fáciles. Gracias, te quiero.

Enfin, je tiens à remercier toutes les entités impliquées l'Université Paris Saclay, l'ONERA, CentraleSupeléc de m'avoir donné l'opportunité de faire mon doctorat ici. C'était vraiment une super expérience, merci !

Contents

List of Figures ix

List of Tables xi

List of Definitions xiii

List of Symbols xvii

Résumé en français xix

1 Introduction and research objectives 3
- 1.1 Distributed State Estimation . . . 4
- 1.2 State of the art . . . 4
  - 1.2.1 Objectives . . . 5
  - 1.2.2 Moving Horizon Estimation . . . 5
  - 1.2.3 Main motivations . . . 8
- 1.3 Contributions of the thesis . . . 9
  - 1.3.1 Distributed Moving Horizon Estimation with pre-estimation over Sensor Network . . . 9
  - 1.3.2 ℓ-step Neighborhood Distributed Moving Horizon Estimation . . . 10
  - 1.3.3 Constrained DMHE with sporadic measurements for Multi-Vehicle Systems . . . 11
  - 1.3.4 Publications . . . 12
- 1.4 Thesis outline . . . 13

2 Distributed Moving Horizon Estimation with pre-estimation over Sensor Network 15
- 2.1 Introduction . . . 15
- 2.2 Notations, definitions and assumptions . . . 17
  - 2.2.1 System modeling . . . 17
  - 2.2.2 Graph theory tools . . . 18
  - 2.2.3 Sensor network . . . 20
  - 2.2.4 Local, regional and collective data . . . 22
- 2.3 Centralized MHE with pre-estimation . . . 23
- 2.4 Distributed MHE with pre-estimation . . . 24
  - 2.4.1 Local minimization problem . . . 24
  - 2.4.2 Network information exchange . . . 27
  - 2.4.3 DMHE procedure . . . 27
- 2.5 Example . . . 28
- 2.6 Extension with observability rank-based consensus weights . . . 34
- 2.7 Simulation results . . . 35
- 2.8 Conclusion . . . 39

3 ℓ-step Neighborhood Distributed Moving Horizon Estimation **41**
3.1 Introduction . . . 41
3.2 Problem statement . . . 42
3.2.1 Communication protocol . . . 43
3.2.2 Problem formulation . . . 46
3.3 Proposed DMHE technique . . . 47
3.3.1 Local optimization problem . . . 47
3.3.2 Fused arrival cost . . . 49
3.3.3 DMHE algorithm . . . 51
3.4 Simulations . . . 52
3.5 Conclusion . . . 57

4 Constrained DMHE with sporadic measurements for Multi-Vehicle Systems **59**
4.1 Introduction . . . 60
4.2 Distributed State Estimation over a static Sensor Network with Sporadic Measurements . . . 61
4.2.1 Problem description . . . 61
4.2.2 Considered dynamical model . . . 62
4.2.3 Constraints . . . 64
4.2.4 Problem statement . . . 64
4.3 Proposed DMHE technique . . . 65
4.3.1 Pre-estimation observer . . . 65
4.3.2 Field of View constraints and sporadic measurements . . . 66
4.3.3 Objective function with sporadic measurements . . . 66
4.3.4 Consensus terms . . . 67
4.3.5 Observability rank-based weights technique with sporadic measurements . . . 67
4.3.6 Local optimization problem with sporadic measurements . . . 68
4.3.7 Distributed MHE (DMHE) *modus operandi* . . . 68
4.4 Realistic simulations . . . 69
4.4.1 Scenario and simulation setup . . . 69
4.4.2 Results' analysis . . . 72
4.5 Experiments . . . 74
4.5.1 Experiments setup . . . 75
4.5.2 Experimental results . . . 78
4.5.3 Performance evaluation . . . 81
4.6 Conclusion . . . 88

5 Conclusion and perspectives **91**
5.1 Conclusion . . . 91
5.2 Perspectives . . . 92

Bibliography **95**

List of Figures

1.1 Possible scenarios for teams of mobile robots in surveillance missions. . . . 4
1.2 Moving Horizon Estimation sliding window. . . . 6

2.1 Convex and non-convex set examples. . . . 18
2.2 Undirected and directed graphs examples. . . . 19
2.3 Weighted digraph example. . . . 19
2.4 Strongly connected and complete graphs examples. . . . 20
2.5 Topology of the sensor network. . . . 29
2.6 Components of the states $x_t = [x_{1,t}\ x_{2,t}\ x_{3,t}\ x_{4,t}]^\top$ and the estimates $\hat{x}_t = [\hat{x}_{1,t}\ \hat{x}_{2,t}\ \hat{x}_{3,t}\ \hat{x}_{4,t}]^\top$ computed by the MHE$_{pre}$ and DMHE$_{pre}$ algorithms. . . . 31
2.7 Components of the estimation error $e_t = [e_{1,t}\ e_{2,t}\ e_{3,t}\ e_{4,t}]^\top = x_t - \hat{x}_{t|t}$ of the MHE$_{pre}$ and DMHE$_{pre}$ algorithms. . . . 32
2.8 Components of the states $x_t = [x_{1,t}\ x_{2,t}\ x_{3,t}\ x_{4,t}]^\top$ and the estimates $\hat{x}_t = [\hat{x}_{1,t}\ \hat{x}_{2,t}\ \hat{x}_{3,t}\ \hat{x}_{4,t}]^\top$ computed by the MHE and DMHE algorithms. . . . 33
2.9 Comparison among the computation times of all algorithms run in the simulation: MHE, MHE$_{pre}$, DMHE, DMHE$_{pre}$. . . . 34
2.10 Sensor network. . . . 36
2.11 Time behavior of the trials-averaged RMSE for 100 Monte Carlo trials. . . . 37
2.12 Sum of RMSEs in the transient period (for $t \in \{1, \ldots, 9\}s$) and in the steady state (for $t \in \{10, \ldots, 50\}s$). . . . 37
2.13 Sum of the trials-averaged computation times τ for 100 Monte Carlo trials. . . . 38
2.14 Sum of x_0-averaged RMSEs of the proposed DMHE$_{w-pre}$ with K varying as in (2.38). . . . 38
2.15 x_0-ε-averaged computation time τ. . . . 39

3.1 Three ℓ-step neighborhood sets. . . . 43
3.2 4-node digraph sensor network. . . . 45
3.3 Topology of the sensor network composed by 100 nodes [Battistelli, 2018]. . . . 52
3.4 PRMSE time behavior comparison. . . . 54
3.5 Computation time comparison. . . . 54
3.6 Averaged PRMSE time behavior comparison. . . . 55
3.7 Comparison of the sum of PRMSE for a different horizon length N. . . . 56
3.8 Comparison of the sum of computation time for a different horizon length N. . . . 56

4.1 Experiment scenario setup: Multi-Vehicle System with 5 vehicles in the starting place and Sensor Network composed of 12 cameras and Raspberry PI computers. . . . 62
4.2 Road $\mathcal{R}$ (blue line), fields of view $\mathcal{F}^1$ and $\mathcal{F}^2$ (yellow), and convexified constraints $\mathcal{S}^1$ (red polygone) and $\mathcal{S}^2$ (blue polygone). . . . 64
4.3 Example of active and inactive sensors. . . . 66
4.4 Scenario illustrated in Gazebo: Multi-Vehicle System with 5 vehicles in the starting place. . . . 70
4.5 TurtleBot3 within the indoor arena. . . . 70

4.6 Simulation scenario with road (blue line), cameras (numbered nodes), projection of the field of view of each camera (yellow rectangles), and communication link between the sensors (red arrows). . . . 71
4.7 Averaged Root Mean Square Error (RMSE) among all the sensors and all the trials. . . . 72
4.8 Averaged computation time τ among all the sensors and all the trials. . . . 73
4.9 Estimation of the position along the x-axis of the fourth vehicle ${}^4\hat{p}_{x,t}$ by all the active sensors. 73
4.10 Three out of five vehicles detected by one camera. . . . 75
4.11 Scenario 1, maximum coverage area with 12 sensors. . . . 76
4.12 Scenario 2, maximum coverage with area 6 sensors. . . . 76
4.13 Scenario 3, less coverage area with 12 sensors. . . . 77
4.14 Computation time τ of all algorithms with FoV constraints during the real experiments. . . 79
4.15 RMSE of all algorithms with FoV constraints during the real experiments. . . . 79
4.16 Scenario 1: Planar trajectory of the vehicles and estimations by DMHE$^{\mathcal{S}}$ using 12 cameras (real experiment). . . . 80
4.17 Examples of communication topologies of a Sensor Network composed of 4 nodes. . . . 82
4.18 Scenario 1: RMSE averaged among the observers and time (all sensors). . . . 82
4.19 Scenario 1: RMSE averaged among the observers and time (active sensors only). . . . 83
4.20 Scenario 1: RMSE of DMHE$^{\mathcal{S}}$ over time for all communication links radius (all sensors). . 83
4.21 Scenario 1: Planar trajectory of the vehicles and estimations with FoV constraints (left) and without (right). . . . 84
4.22 Scenario 1: Part of the planar trajectory of the vehicles and estimations with FoV constraints (left) and without (right). (Zoom of Figure 4.21) . . . 85
4.23 Scenario 2: RMSE averaged among the observers and time (all sensors). . . . 85
4.24 Scenario 2: RMSE of DMHE$^{\mathcal{S}}$ over time with a different communication link radius. . . . 86
4.25 Scenario 2: Planar trajectory of the vehicles and estimates with FoV constraints (left) and zoom (right). . . . 87
4.26 Comparison between the RMSE of the DMHE$^{\mathcal{S}}$ for the 3 scenarios for $\rho = 1$, with an average among 10 trials for Scenario 3. . . . 88

List of Tables

2.1 Eigenvalues of $\Phi = A - LC$ and $\Phi^i = A - L^i \bar{C}^i$. 30
2.2 Minimum, maximum and sum of the computation time τ_t (in seconds) and RMSE with $t_c = 5$ s of all algorithms collected in the simulation. 33

3.1 Collected data over time by the considered nodes for N=3. 46

List of Definitions

2.1 Convex set . . . 18
2.2 Undirected graph . . . 19
2.3 Directed graph . . . 19
2.4 Weighted graph . . . 19
2.5 Strongly connected graph . . . 20
2.6 Complete graph . . . 20
2.7 Peer-to-peer sensor network . . . 20
2.8 Static topology sensor network . . . 20
2.9 Dynamic topology sensor network . . . 21
2.10 Stochastic matrix . . . 21
2.11 Local, regional and collective information . . . 22
2.12 Local, regional and collective observability . . . 22

3.1 ℓ-step neighborhood set . . . 43
3.2 Inactive sensor . . . 44
3.3 Poorly-Observing Sensor Network . . . 46

4.1 Active and inactive sensor . . . 63
4.2 Radius communication link . . . 81
4.3 Neighborhood based on radius ρ . . . 81

List of Acronyms

DMHE Distributed MHE

DSE Distributed State Estimation

FoV Field of View

MHE Moving Horizon Estimation

RMSE Root Mean Square Error

ROS Robot Operating System

SN Sensor Network

List of Symbols

Sets

$\mathbb{N}$	Set of the natural numbers
$\mathbb{R}$	Set of the real numbers
$\mathbb{R}^n$	Set of the n-dimensional real vectors
$\mathbb{R}^{n\times m}$	Set of all real n-by-m matrices
$\mathcal{A} \subseteq \mathcal{B}$ (resp. $\mathcal{A} \subset \mathcal{B}$)	$\mathcal{A}$ is a subset (resp. strict subset) of $\mathcal{B}$
$\mathcal{A} \cap \mathcal{B}$	Intersection of sets $\mathcal{A}$ and $\mathcal{B}$
$\mathcal{A} \cup \mathcal{B}$	Union of sets $\mathcal{A}$ and $\mathcal{B}$
$\mathcal{A} \times \mathcal{B}$	Cartesian product of sets $\mathcal{A}$ and $\mathcal{B}$
$\text{Co}(\mathcal{A})$	Convex hull of set $\mathcal{A}$
$\text{card}(\mathcal{A})$	Cardinality of set $\mathcal{A}$, i.e. the number of elements in $\mathcal{A}$

Algebra

$a \in \mathbb{R}$	Real scalar number
$x \in \mathbb{R}^n$	Real vector of n elements
$A \in \mathbb{R}^{n\times m}$	Real matrix of n rows and m columns
a_{ij}	Element (i, j) of matrix A
$A^\top$	Transpose of the matrix A
A^{-1}	Inverse of the matrix A
$\text{diag}(a_1, \ldots, a_n)$	Diagonal matrix having the vector $[a_1 \ \ldots \ a_n]^\top$ as main diagonal
$I_n \in \mathbb{R}^{n\times n}$	n-by-n identity matrix
$O_{n,m} \in \mathbb{R}^{n\times m}$	n-by-m matrix filled of zeros
$\|x\|_2$	Euclidean norm of vector x such that $\|x\|_2 = \sqrt{x^\top x}$
$\|x\|_Q$	Weighted norm of vector x such that $\|x\|_Q = \sqrt{x^\top Q x}$

Estimation

z_t	Generic variable at time t
$z_{t_1\|t_2}$	Generic variable at time t_1 and estimated at time t_2
$z_{(\cdot)\|t}$	Generic variable calculated or given at time t
$(\hat{\cdot})$	Generic parameters of the optimization problem to be estimated
$[t - N, t]$	Time interval that begins at time $t - N$, where $t, N \in \mathbb{N}$, and finishes at time t

Sensor Network

$(\cdot^i)$	Right superscript i referring to the i-th sensor
$\mathcal{N}^i$	Set of the neighbors nodes of sensor i
$\mathcal{N}_\ell^i$	ℓ-step neighborhood set of sensor i (with size ℓ)
x^i	Generic *local* information, i.e. referring only to sensor i
$\bar{x}^i$	Generic *regional* information, i.e. referring to sensor i and its neighborhood
$\mathbf{x}$	Generic *collective* information, i.e. referring to the entire sensor network

Multi-Vehicle System

$({}^\nu\cdot)$	Left superscript ν referring to the ν-th vehicle. ***Nota bene***: When the left superscript ν is omitted it refers to the whole Multi-Vehicle System

Résumé en français

Les algorithmes distribués sont dorénavant présents dans de nombreux aspects de l'Automatique avec des applications pour des systèmes multi-robots, des réseaux de capteurs, couvrant des sujets tels que la commande, l'estimation d'état, la détection de défauts, la détection et l'atténuation des cyberattaques sur les systèmes cyber-physiques, etc. En effet, les systèmes distribués sont confrontés à des problèmes tels que l'extensibilité à un grand nombre d'agents et la communication entre eux. Dans les applications de systèmes multi-agents (par exemple, flotte de robots mobiles, réseaux de capteurs), il est désormais courant de concevoir des algorithmes d'estimation d'état de manière distribuée afin que les agents puissent accomplir leurs tâches sur la base de certaines informations partagées au sein de leur voisinage.

Dans le cas de missions de surveillance, un réseau de capteurs statique et à faible coût (par exemple, caméras) pourrait ainsi être déployé pour localiser de manière distribuée des intrus dans une zone donnée. Dans ce contexte, l'objectif principal de cette thèse est de concevoir des observateurs distribués pour estimer l'état d'un système dynamique (par exemple, flotte de robots intrus) avec une charge de calcul réduite tout en gérant efficacement les contraintes et les incertitudes.

Ce manuscrit propose plusieurs variantes d'algorithmes d'estimation d'état distribués pour les réseaux de capteurs. Ces algorithmes sont fondés sur le paradigme de l'estimation par horizon glissant et la convergence par consensus. Dans ce qui suit sont listées les contributions de cette thèse pour chacune des techniques d'estimation distribuées à horizon glissant proposées.

Dans [Sui et al., 2010] il a été proposé une stratégie d'estimation à horizon glissant avec une pré-estimation fondée sur un observateur de Luenberger qui a conduit à de bonnes performances, en termes de précision d'estimation, en particulier pour de grands horizons d'estimation. Ce travail a également proposé un problème d'optimisation pour régler les paramètres en minimisant les effets du bruit de mesure et des erreurs de modèle. L'article [Sui and Johansen, 2014] a généralisé la formulation en utilisant une matrice pour pondérer la fonction de pénalité et a également ajouté des contraintes d'état. Une estimation à horizon glissant avec pré-estimation a également été proposée pour les systèmes non linéaires [Suwantong et al., 2014]. Les auteurs de [Farina et al., 2010a] ont proposé un algorithme distribué pour les systèmes linéaires avec la preuve de stabilité dans des conditions d'observabilité faible (en exploitant un consensus sur l'estimation et un terme de pondération du consensus dans la formulation de l'estimation distribuée à horizon glissant). Cependant, le problème du temps de calcul devient crucial, car le réseau est généralement composé de capteurs à faible coût. De plus, l'algorithme de [Farina et al., 2010a] pour calculer les termes de consensus nécessite des informations

sur l'ensemble du réseau de capteurs et n'est donc pas adapté à un schéma entièrement distribué, en particulier lorsque la topologie de communication change dans le temps. Fondée sur les résultats de [Sui and Johansen, 2014, Farina et al., 2010a], la présente thèse propose un algorithme d'estimation distribuée à horizon glissant avec pré-estimation présenté dans [Venturino et al., 2020] et son extension publiée dans [Venturino et al., 2021b]. Un observateur de Luenberger à pré-estimation est considéré dans la formulation du problème local à résoudre par chaque capteur, ce qui entraîne une réduction significative du temps de calcul. La contribution principale de cette technique couvre :

- réduire le temps de calcul nécessaire à la résolution du problème d'optimisation local ;
- préserver la précision des erreurs afin de permettre l'utilisation de ce type d'algorithmes pour les applications sensibles au facteur temps ;
- mieux ajuster les poids de consensus associés à la topologie du réseau via un nouvel algorithme de pondération fondé sur le rang d'observabilité qui tire parti des informations locales disponibles.

[Battistelli, 2018] a introduit un autre mécanisme fondé sur le consensus dans une approche d'estimation distribuée à horizon glissant afin de fusionner les coûts d'arrivée locaux et de garantir la stabilité des erreurs d'estimation de manière totalement distribuée, c'est-à-dire que chaque capteur est capable de garantir la convergence de l'estimation vers l'état du système en utilisant uniquement les informations disponibles localement. Dans [Venturino et al., 2020, Venturino et al., 2021b], nous avons proposé deux algorithmes fondés sur [Farina et al., 2010a] qui, grâce à l'introduction d'un observateur de pré-estimation et aux pondérations fondées sur le rang de la matrice d'observabilité, ont permis de réduire le temps de calcul et d'améliorer la précision de l'estimation. Cette thèse étend l'approche proposée dans [Venturino et al., 2020] à la formulation d'estimation distribuée à horizon glissant de [Battistelli, 2018] qui s'est avérée obtenir des résultats de stabilité plus généraux ainsi que des performances améliorées par rapport à [Farina et al., 2010a]. Par rapport à [Battistelli, 2018], l'algorithme d'estimation distribuée à horizon glissant avec voisinages à ℓ-pas proposé dans ce manuscrit conduit également à un temps de calcul réduit grâce à observateur de pré-estimation. Une autre contribution concerne l'amélioration de la convergence de l'erreur d'estimation en atténuant les problèmes d'inobservabilité. Cette situation peut survenir dans les réseaux de capteurs lorsque certains nœuds n'ont aucune capacité de détection (capteurs inactifs) ou ne peuvent mesurer que certaines parties de l'état du système, ce qui le rendrait non observable en utilisant uniquement ces capteurs. À cette fin, la nouvelle technique d'estimation distribuée à horizon glissant proposée exploite les échanges d'informations entre les nœuds locaux sur la base d'un mécanisme de diffusion de l'information de voisinage à ℓ-pas, qui est naturel dans

les données de fenêtre glissante présentes dans le paradigme de l'estimation de l'horizon glissant.

La dernière contribution porte sur la localisation de systèmes multi-véhicules à l'aide d'algorithmes distribués d'estimation à horizon glissant. Des travaux similaires ont été menés par les auteurs de [Simonetto et al., 2011] et [Yousefi and Menhaj, 2014]. Dans [Simonetto et al., 2011], le problème de l'estimation distribuée à horizon glissant a été abordé en se concentrant sur la non-linéarité du modèle et sur les éventuels problèmes d'observabilité locale au niveau des capteurs. Dans [Yousefi and Menhaj, 2014], les auteurs ont pris en compte les nœuds mobiles dans le réseau de capteurs, ce qui a conduit à traiter une topologie dynamique. Cette contribution se concentre sur l'aspect temps de calcul, qui est un facteur clé pour la mise en œuvre en temps réel. L'originalité de notre algorithme porte sur deux directions. Premièrement, en plus d'un temps de calcul réduit et d'une précision améliorée grâce à la pré-estimation, la technique d'estimation distribuée à horizon glissant proposée est conçue pour des scénarios réalistes de systèmes à grande échelle impliquant des mesures sporadiques (c'est-à-dire disponibles à des instants a priori inconnus). Dans ce but, les contraintes sur les mesures (provenant de la connaissance de l'environnement dans lequel le système multi-véhicules évolue) sont incorporées en utilisant des paramètres binaires dans cette nouvelle formulation d'estimation distribuée à horizon glissant. Ainsi, les informations sur l'environnement sont exploitées pour mieux estimer l'état du système. Deuxièmement, nous évaluons la performance de l'approche distribuée proposée pour l'estimation à l'horizon glissant (en termes de précision et de temps de calcul) sur une étude de cas réaliste, c'est-à-dire la localisation distribuée d'un système multi-véhicules par un réseau de capteurs statiques, développée sous ROS (*Robotic Operating System*) et avec le simulateur Gazebo. Cette implémentation logicielle permet un déploiement sur un réel. Pour confirmer son efficacité, la formulation contrainte de l'estimation distribuée à horizon glissant proposée est comparée à l'algorithme de référence de [Farina et al., 2010a]. En outre, l'approche proposée pour l'estimation distribuée à horizon glissant est évaluée, en termes de précision et de temps de calcul, sur trois expérimentations (impliquant des caméras réelles et des véhicules terrestres autonomes) utilisant différents nombres de capteurs, des topologies de réseau de communication distinctes et une couverture diverse des champs de vision des caméras. En effet, l'une des principales contributions porte sur la validation expérimentale de la technique proposée d'estimation distribuée à horizon glissant pour la localisation d'un système multi-véhicules. Dans la configuration expérimentale développée, le réseau de capteurs statiques est composé de caméras à bas coût qui fournissent des mesures sur les positions des véhicules. Chaque caméra est reliée à une carte Raspberry PI pour les capacités de calcul et de communication. L'algorithme d'estimation distribuée à horizon glissant proposé a été mis en œuvre sous ROS pour fonctionner de manière distribuée sur chaque Raspberry PI. Le système multi-véhicule est composé de cinq robots TurtleBot3

effectuant des déplacements en formation dans une zone de type route située dans l'arène de vol de CentraleSupélec équipée d'un système de capture de mouvement. Une vidéo explicative, permettant une meilleure compréhension des expérimentions est disponible sur https://youtu.be/1CkSba2wVuI.

Les algorithmes proposés sont également comparés à des résultats de la littérature en considérant diverses métriques telles que le temps de calcul et la précision des estimées.

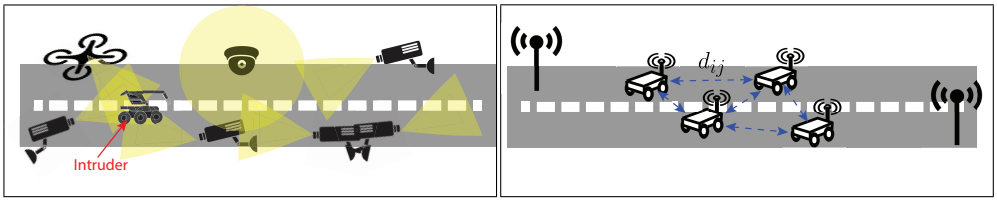
1 - Introduction and research objectives

In the last few years, distributed algorithms have pervaded many aspects of control engineering with applications for multi-robot systems, sensor networks, and others, covering topics such as control [Segovia et al., 2021b, Bertrand et al., 2020, Bono et al., 2019, Ding et al., 2019, Conte et al., 2016], state estimation [Simonetto et al., 2011, Sarras et al., 2017, J. Zeng and Liu, 2015], fault detection and mitigation [Marzat et al., 2018, Wu et al., 2019], cyber-attack detection and mitigation on cyber-physical systems [Gheitasi et al., 2019, Gallo et al., 2020, Jin et al., 2021], both control and estimation [Segovia et al., 2019], formation control [Nguyen et al., 2015, Chevet et al., 2018, Venturino and Lucia, 2019], etc. Despite their different purposes, these topics share common characteristics as a consequence of the development in distributed schemes. Indeed, distributed schemes face problems like scalability (which handles growing or decreasing number of resources by keeping nearly constant the complexity of the problem to solve) and communication between agents (in particular which and what amount of information to share and with whom). If, on one hand, sharing more data could lead to have better performance in terms of accuracy, on the other hand, the complexity could raise as well as the communication burden. In centralized frameworks, with the same purpose, usually a central unit manages all the resources involved in the network; nevertheless it can hardly deal with scalability issues due to physical and computational limitations. Moreover, it is not robust with respect to the loss of the central unit. However, as an ideal objective, distributed algorithms are usually compared with the centralized counterpart in terms of performance.

In multi-agents applications (fleet of mobile robots, sensor networks) it is now common to design control or state estimation algorithms in a distributed way so that the agents can accomplish their tasks based on some shared information within their neighborhoods [Hespanha et al., 2007]. Using centralized scheme leads to achieve global optimum, however centralized systems face strong constraints in the real world. Indeed, they often require full connectivity and most of them hardly scale up, in particular with the number of agents [Moors et al., 2005]. In addition, robustness with respect to the loss of the central entity is a major weakness of centralized approaches [Khauphung et al., 2008].

In surveillance missions, teams of robots could be deployed to track/follow intruders in a given area [Zhang and Meng, 2010, Pimenta et al., 2009]. In this case, the problem consists in estimating the trajectory of the intruders from measurements provided by the sensors (cameras, lidars, radars, etc.), see Figure 1.1a. In the case of mobile sensors/robots, the agents also have to control their own motion. Estimates of their relative positions, see Figure 1.1b, should be computed in this case in addition to the intruders' trajectory. For such problems, distributed state estimation algorithms proposed in the literature are suitable as theoretical

basis [Olfati-Saber, 2005, Carli et al., 2008, Robin and Lacroix, 2016].



(a) Estimation of the intruder's trajectory. (b) Estimation of the team's relative positions.

Figure 1.1: Possible scenarios for teams of mobile robots in surveillance missions.

1.1 Distributed State Estimation

The estimation algorithms used in such applications over sensor network are mainly formulated as centralized sensor fusion architectures, where all the sensors transmit their local measurements to a central unit which processes the provided data to update the estimation. Unlike centralized schemes, in distributed approaches [Zhang et al., 2019] each sensor computes a local estimation (at least partially) using the information acquired only from locally connected neighbors or local measurements. This can improve robustness to sensor failure exploiting redundancy [Tošić et al., 2013, Cărbunar et al., 2006] and also lower the communication burden since data is transmitted only among local nodes in the network [Hespanha et al., 2007]. In the context of large-scale systems, the algorithms have to be scalable to deal with large-scale networks, need low computation load (necessary when dealing with low-cost sensors with low computation capabilities), have to minimize the utilization of communication resources, etc.

1.2 State of the art

During the past years, several research works have been dedicated to model, estimate and control distributed Multi-Agent Systems (MAS), see [Negenborn and Maestre, 2014, Vadigepalli and Doyle III, 2003, Millán et al., 2013, Nguyen et al., 2017, Rego et al., 2019a], etc. Considerable studies help to better cope with Multi-Agent Systems by developing suitable distributed algorithms to deal with state estimation of large-scale systems [Segovia et al., 2021a, Wang et al., 2018], with applications on Multi-Vehicle System [Halsted et al., 2021, Vargas et al., 2022, Mechali et al., 2022], also used to localize such systems. Two different main classes of problems can be considered, depending whether the localization is performed by the vehicles themselves and/or accounting for information from them (cooperative localization) [Nogueira et al., 2010, Franzè et al., 2020], or by an exogenous system without any exchange of information with the vehicles (non-cooperative localization) [Mei et al., 2019, Viani et al., 2015]. For example,

in [D'Alfonso et al., 2015], the authors compared the well-known extended Kalman filter and unscented Kalman filter for nonlinear models to estimate the pose of a single mobile robot by fusing the measurements taken by ultrasonic sensors located onboard the robot. For non-cooperative localization, such experiments are rare in literature. Nevertheless, in recent decades, the interest in distributed state estimation increased tremendously. For instance, in [He et al., 2020], the authors reviewed several outcomes of distributed state estimation over a low-cost Sensor Network (SN), pointing out their characteristics, benefits, and challenging issues. In [Halsted et al., 2021], the authors surveyed and compared several classes of distributed optimization algorithms for multi-robot applications. One critical point for such Sensor Networks (SN) comes from the limited computation and communication resources of local sensors, which strongly motivates the current thesis.

Consensus techniques have often been used for designing distributed state estimation algorithms for Sensor Networks. A consensus problem in which the agreement value is a distributed estimation of some non-constant quantity of interest is referred to as a dynamic consensus, see [Manfredi, 2013]. Recently, such algorithms have pervaded the control engineering literature on numerous topics, e.g. state estimation [Olfati-Saber, 2007, Soatti et al., 2016], fault detection and isolation [Lauricella et al., 2017, Lauricella et al., 2020], detection and mitigation of cyber-attacks [Sargolzaei et al., 2019, Ghafouri et al., 2020], combined control and state estimation [Garin and Schenato, 2010, Zhao and Zelazo, 2015], etc.

1.2.1 Objectives

In this context, the main objective of this work is to design distributed observers with reduced computation load to estimate the state of multi-robot systems that efficiently handle constraints and uncertainties. It is, indeed, necessary to be able to take into account realistic measurement models of the sensors and of the intruder's or agent's dynamics that can both exhibit uncertainties. Moreover, different types of constraints can be explicitly considered in the estimation problem to account for some specificity of the application and prior information that could be available: limitations of sensor field of view, path constraints, bounds on parameters or inputs of the intruder's dynamic model and/or on its state (e.g. *a priori* information on the maximum speed of the intruder or on the possible trajectories).

The Moving Horizon Estimation (MHE) paradigm is prone to be a good candidate for this problem. Indeed, it has the particularity to cope with constraints and dealing with a sequence of past measurements and predicted output, see Figure 1.2.

1.2.2 Moving Horizon Estimation

Among all state estimation algorithms, MHE techniques have been studied over the past decades due to their capacity to consider constraints within their formulation. The first idea of MHE proposed in [Muske et al., 1993] consists in

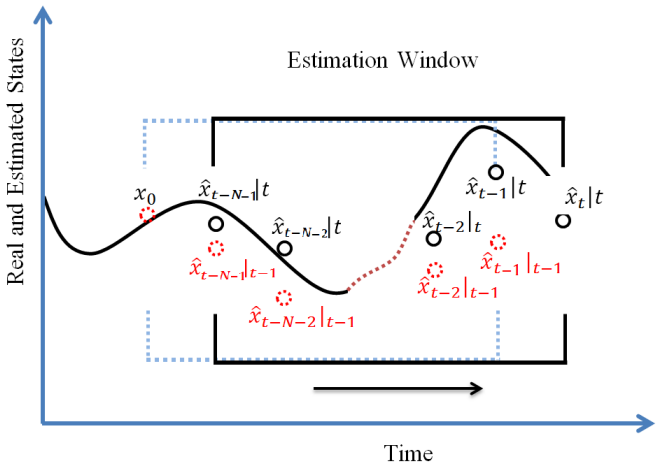


Figure 1.2: Moving Horizon Estimation sliding window.

estimating the current state of a system by solving a least-square optimization problem penalizing on one hand the deviation between the measurements and the predicted outputs, and on the other hand a distance between the estimated state and some *a priori* information about the state, see Figure 1.2. Often considered as a dual approach to Model Predictive Control for state estimation, MHE is implemented in a receding horizon way by considering a finite number of the most recent past measurements. Thus, these techniques use an estimation window of specified size, which moves forward at each instant. Therefore, the problem remains computationally tractable since only the latest measurements are processed, while previous information is condensed in the so-called *arrival cost* [Rao and Rawlings, 2000, Alessandri et al., 2008, Tebbani et al., 2013]. Figure 1.2 shows the state sequence estimated at time $t-1$ (red points), the one estimated at time t (black points) and real trajectory of the state (black line). In the past decades, the MHE technique has gained particular attention within the research community. Indeed, there is a multitude of works dedicated to this approach that address a variety of aspects within the state estimation domain. For instance [Isaza-Hurtado et al., 2019] proposed a technique to cope with estimation for Linear Time-Invariant Systems in the presence of non-uniform sampled measurements. [Alessandri et al., 2011] proposed an MHE technique for non-linear systems using neural networks which led to a reduced online computational effort. Although the MHE approach is functional for control engineers offering the freedom to tune the parameters of the cost function, a strength and a weakness of this approach is the use of an optimization problem to be solved within the sampling period. However, for large-scale systems and centralized approaches, this issue becomes critical due to the complexity of the optimization problem [Farina et al., 2010b, Wang et al., 2017]. Indeed, the constrained optimization problem brings with it a certain computational load, which may become problematic for large-scale systems [Haber and Verhae-

gen, 2013, Vukov et al., 2015], or when using MHEs techniques for computationally demanding purposes (see e.g. [Famularo et al., 2022] where the authors proposed a MHE scheme using Linear Matrix Inequalities for fault detection and isolation). In addition to specific optimization methods, several techniques have been developed to reduce the computational load in Moving Horizon Estimation. Recent approaches on learning-based MHE rely on existing machine learning frameworks to computationally improve the estimator [Muntwiler et al., 2022, Karg and Lucia, 2021]. Another method, *inter alia*, is to reduce the number of optimization parameters by replacing the sequence of unknown inputs (or state noise) to be estimated in the dynamical model of the system by a Luenberger *pre-estimation* observer, which leads to less computation time, while preserving the accuracy of the estimates, as introduced in [Sui and Johansen, 2014] for linear systems, in [Suwantong et al., 2014] for nonlinear systems.

In the context of MAS, Distributed Moving Horizon Estimation (DMHE) has received increased attention in recent years, starting from the estimation for linear systems by [Farina et al., 2010a], where the authors proved the convergence of the estimation error even under weak observability conditions. An extension of DMHE to nonlinear systems subject to constraints has been further proposed by the same authors in [Farina et al., 2012]. A DMHE scheme for a class of nonlinear systems with bounded output measurement noise and process disturbances is designed in [Zhang and Liu, 2013], while [J. Zeng and Liu, 2015] considered Distributed Moving Horizon Estimation of nonlinear systems subject to communication delays and data losses. The authors of [Yin and Liu, 2017] developed a DMHE for a class of two-time-scale nonlinear systems described in the framework of singularly perturbed systems. In addition, [Battistelli, 2018] developed a DMHE with fused arrival cost suitable for a fully distributed implementation. In the context of large-scale systems, [Segovia et al., 2021a] designed a two-step distributed state estimation scheme in the presence of unknown-but-bounded disturbances and noises, which involves a set-membership-based MHE. A Distributed Moving Horizon Estimation via operator splitting for automated robust power system state estimation has been proposed in [Kim et al., 2021]. More recently, in [Yin and Huang, 2022] an event-triggered DMHE is proposed for general linear systems that comprise several subsystems. A consensus variational Bayesian MHE for distributed Sensor Networks with unknown noise covariances has been proposed in [Dong et al., 2022], where three consensus tasks are performed in parallel at each time instant. More recently, [D'Amato et al., 2022] proposed a Decentralized Moving Horizon Estimation algorithm for a fleet of UAVs, the decentralization of the scheme is obtained by decomposing the overall estimation problem in several optimization sub-models whose convergence is guaranteed by consensus.

Various distributed state estimation algorithms have been developed to address the localization problem of Multi-Vehicle Systems. In distributed cooperative localization, the vehicles estimate usually by themselves their own state based on

the information exchanged with their neighbors, over a vehicle-to-vehicle network, see [Shorinwa et al., 2020, Nogueira and Pereira, 2019, Viegas et al., 2018]. In distributed non-cooperative localization, Sensor Networks are often considered to provide localization of the Multi-Vehicle System. In the context of Multi-Vehicle System localization over a Sensor Network, similar works have been conducted by the authors of [Petitti et al., 2011], [Simonetto et al., 2011] and [Yousefi and Menhaj, 2014]. In [Simonetto et al., 2011], the DMHE problem has been addressed by focusing on the non linearity of the model and on the possible local observability issues at the sensor level. In [Yousefi and Menhaj, 2014], the authors accounted for mobile nodes in the Sensor Network that led to a dynamic topology. Indeed, using a flocking algorithm for the motion control, the mobile sensors attempt to move in a specific way in order to get the best positions to observe the target and to avoid collisions between neighboring agents. In this context, distributed state estimation over not fully reliable Sensor Network could lead to *sporadic measurements*, i.e. available at time instants *a priori* unknown, as considered in [Ferrante et al., 2016], [Postoyan and Nešić, 2011]. In the case of sensors with limited field of views (e.g. cameras), sporadic measurements are prone to be even more frequent.

1.2.3 Main motivations

This thesis focuses on designing MHE-based distributed state estimation algorithms over Sensor Network (SN) with the final intent to apply them for localizing Multi-Vehicle System, as detailed in Section 1.2.1. From the literature overview of Section 1.2.2, it appears that this research topic is quite active.

This thesis aims to apply such algorithms for real application contexts using low-cost sensors, lifting the priority of dealing with the computation aspects of the optimization problem within the MHE paradigm. Section 1.2.2 lists some existing methods in the literature that reduce the computation time of centralized MHE methods but not for distributed ones. Our first focus is then on investigating existing methods and designing a DMHE algorithm able to estimate the state of the system without penalizing the accuracy of the estimation error while reducing the computational load.

Another essential aspect related to this research topic is the weak observability condition in which such algorithms have to operate. For example, when tracking a vehicle using cameras, or other similar sensors with a limited range of actions, the vehicle can be detected only when it is operating in the camera's field of view. Thus, this raises the issue of weak observability, or unobservability, conditions. In consensus-based distributed algorithms, it is possible to design the consensus strategy to relay information from (sub-)systems with better observability properties. To the best of our knowledge, this aspect has not been investigated for DMHE algorithms. Moreover, as pointed out in the perspectives of [Farina et al., 2010a], DMHE algorithms are prone to use a sequence of past measurements. Hence it could be possible to mitigate unobservability conditions by exploiting data from non-direct sensor neighbors.

Heading toward real hardware setup and experimental implementation, another aspect that needs more insights within the current manuscript's application context is that the measurements are available at time instants *a priori* unknown, as pointed out in the example of detecting vehicles with cameras. Thus, this issue is related to the unobservability conditions but with more difficulty since the distributed algorithms in the literature address this issue by considering the sensors with fixed properties. In other words, the problem is addressed by considering some cameras as always able to detect vehicles and the rest of the cameras always unable to do so. Indeed, all the algorithms in the literature are evaluated with numerical simulations, which is easy to fulfill such a restrictive assumption. In the real world, this is not always possible. Thus, this current thesis investigates how to face sporadic measurements using DMHE algorithms.

In addition, when designing algorithms for real-world applications, it is possible to consider some information when the environment is known. The same statement can be true when the types of the used sensors are known. Since the MHE involves an optimization problem, it is possible to include such information as constraints, enhancing the accuracy of the estimates. It has been done in [Brulin et al., 2009], where image occlusions are treated as visual constraints in the estimation process.

Finally, an actual experimental setup with ground mobile robots and a low-cost sensor camera network can validate the algorithm designed for this research work.

1.3 Contributions of the thesis

The current manuscript proposes several variants of Distributed State Estimation (DSE) algorithms for Sensor Network (SN). These algorithms are based on the MHE paradigm and consensus convergence. The following paragraphs detail the contribution of this thesis for each of the proposed DMHE techniques.

1.3.1 Distributed Moving Horizon Estimation with pre-estimation over Sensor Network

In [Sui et al., 2010] it has been proposed an MHE strategy with a Luenberger-based pre-estimation that led to good performance, in terms of estimation accuracy, especially for large estimation horizons. They also provided an optimization problem to tune the parameters minimizing the effects of measurement noise and model errors. The paper [Sui and Johansen, 2014] generalized the formulation using a matrix for weighting the penalty function and also added states constraints. An MHE with pre-estimation has been proposed also for non-linear systems [Suwantong et al., 2014]. The authors of [Farina et al., 2010a] proposed a distributed algorithm for linear systems with the stability proof under weak observability conditions (exploiting a *consensus-on-estimate* and a *consensus weight* term in the DMHE formulation). However, the computation time issue becomes crucial, since usually the network is composed of low-cost sensors. Moreover, the algorithm of [Farina et al., 2010a] to compute the consensus terms needs information about

the entire SN and thus is not suitable for a fully distributed scheme, especially when the communication topology changes in time.

Based on the results of [Sui and Johansen, 2014, Farina et al., 2010a], the present thesis proposes a DMHE algorithm with pre-estimation presented in [Venturino et al., 2020] and its extension published in [Venturino et al., 2021b]. A pre-estimating Luenberger observer is considered in the formulation of the local problem to be solved by each sensor, resulting in a significant reduction of the computation time. The main contribution of this technique covers:

- Reducing the computation time required for solving the optimization problem;
- Preserving the accuracy of the estimation errors, which allows the use of this type of algorithms for time-sensitive applications;
- Better tuning the consensus weights associated to the network topology via a new observability rank-based weight algorithm which takes advantage of the local available information.

1.3.2 ℓ-step Neighborhood Distributed Moving Horizon Estimation

[Battistelli, 2018] introduced another consensus-based mechanism in a DMHE approach to fuse local arrival costs and guarantee stability of the estimation errors in a fully distributed way, i.e. each sensor being capable of guaranteeing convergence of the estimate to the system state using only locally available information. In [Venturino et al., 2020, Venturino et al., 2021b], we presented two algorithms based on [Farina et al., 2010a] which, thanks to the introduction of pre-estimating observer and the observability rank-based weights technique, they were able to reduce the computation time and to enhance the estimation accuracy.

This PhD thesis extends the approach proposed in [Venturino et al., 2020] to the DMHE formulation of [Battistelli, 2018] which has proven to obtain more general stability results as well as enhanced performance compared to [Farina et al., 2010a]. With respect to [Battistelli, 2018], the ℓ-step Neighborhood DMHE algorithm proposed in this manuscript also leads to a reduced computation time due to a pre-estimating observer. Another contribution concerns the improvement of the convergence of the estimation error by mitigating unobservability issues. This situation could arise in sensor networks when some nodes may have no sensing capacities (inactive sensors), or may be able to only measure some parts of the state of the system that would make it non observable using only these sensors. For this purpose, the new proposed DMHE technique exploits the exchanges of information among local nodes based on an ℓ-step neighborhood information spreading mechanism, which comes natural in the sliding window data present in the MHE paradigm.

These contributions have led to the publication of the paper [Venturino et al., 2021a]. A journal paper is in preparation.

1.3.3 Constrained DMHE with sporadic measurements for Multi-Vehicle Systems

The current contribution focuses on Multi-Vehicle System localization using DMHE algorithms. Similar works have been conducted by the authors of [Simonetto et al., 2011] and [Yousefi and Menhaj, 2014]. In [Simonetto et al., 2011], the DMHE problem has been addressed by focusing on the non linearity of the model and on the possible local observability issues at the sensor level. In [Yousefi and Menhaj, 2014], the authors accounted for mobile nodes in the Sensor Network that led to deal with a dynamic topology. The current contribution focuses on the computation time aspect, which is a key factor for real-time implementation.

The contribution of the current algorithm is two-fold.

First, in addition to a reduced computation time and an improved accuracy due to the pre-estimation, the proposed DMHE technique is designed for realistic large-scale systems scenarios involving sporadic measurements (i.e. available at time instants *a priori* unknown). To this aim, constraints on measurements (coming from the knowledge of the environment where the Multi-Vehicle System is evolving) are embodied using binary parameters in this novel Distributed Moving Horizon Estimation formulation. Thus, the environment information is exploited to better estimate the system state.

Second, we evaluate the performance of the proposed DMHE approach (in terms of accuracy and computation time) on a realistic case study, i.e. the distributed localization of a Multi-Vehicle System by a static sensor network, developed within the Robot Operating System (ROS) framework and Gazebo environment. This realistic distributed implementation within ROS and Gazebo would enable the deployment on a hardware setup. To confirm its efficiency, the proposed DMHE constrained formulation is compared with the notable DMHE algorithm [Farina et al., 2010a]. Furthermore, the proposed DMHE approach is evaluated, in terms of accuracy and computation time, on three experiments (involving real cameras and autonomous ground vehicles) using different numbers of sensors, distinct communication network topologies and diverse coverage of the cameras' fields of view. Indeed, one of the main contributions consists in the experimental validation of the proposed distributed MHE localization technique of a Multi-Vehicle System. In the developed experiment setup, the static Sensor Network is composed of low-cost cameras which provide measurements on the positions of the vehicles. Each camera is attached to a Raspberry PI for computational and communication capabilities. The proposed DMHE algorithm has been implemented within the ROS framework to run in a distributed way on each Raspberry PI. The Multi-Vehicle System is composed of five TurtleBot3 robots performing formation motion within a road-like area located in an indoor arena equipped with a motion capture system.

These contributions have led to the publication of the papers [Venturino et al., 2022a] and [Venturino et al., 2022b].

1.3.4 Publications

The work during the preparation of this thesis has led to the submission and publication of several conference and journal papers.

Peer-reviewed journal papers

- **A. Venturino**, C. Stoica Maniu, S. Bertrand, T. Alamo, and E. F. Camacho. Distributed moving horizon state estimation for sensor networks with low computation capabilities. *System Theory, Control and Computing Journal*, 1(1):81–87, 2021.
- **A. Venturino**, C. Stoica Maniu, S. Bertrand, T. Alamo, and E. F. Camacho. Multi-vehicle localization by distributed MHE over a sensor network with sporadic measurements: further developments and experimental results. *Submission to the Control Engineering Practice Journal*, 2022.

Peer-reviewed conference papers

- **A. Venturino**, S. Bertrand, C. Stoica Maniu, T. Alamo, and E. F. Camacho. Distributed moving horizon estimation with pre-estimating observer. In *24th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 174–179, Sinaia, Romania, 8-10 October, 2020. **Best Paper Award**
- **A. Venturino**, S. Bertrand, C. Stoica Maniu, T. Alamo, and E. F. Camacho. A new ℓ-step neighbourhood distributed moving horizon estimator. In *60th IEEE Conference on Decision and Control*, pages 508–513, Austin, Texas, USA, 13-17 December, 2021.
- **A. Venturino**, S. Bertrand, C. Stoica Maniu, T. Alamo, and E. F. Camacho. Multi-vehicle system localization by distributed moving horizon estimation over a sensor network with sporadic measurements. In *6th IEEE Conference on Control Technology and Applications (CCTA)*, Trieste, Italy, 23-25 August, 2022.

Other publications

Peer-reviewed conference paper

- C. Stoica Maniu, C. Vlad, T. Chevet, S. Bertrand, **A. Venturino**, G. Rousseau, and S. Olaru. Control systems engineering made easy: Motivating Students through experimentation on UAVs. In *IFAC 2020-21st IFAC World Congress*, 2020.

Oral presentations

- **A. Venturino**, S. Bertrand, C. Stoica Maniu, T. Alamo, E.F. Camacho, "Distributed moving horizon localization for Multi-Vehicle System over a Sensor Network with sporadic measurements", *Journée du Comité Technique Commande Prédictive Non Linéaire*, Paris, June 3, 2022.

- **A. Venturino**, C. Stoica Maniu, S. Bertrand, T. Alamo, E.F. Camacho, "Distributed localization over sensor camera network for vehicles in urban-like environment", *Workshop L2S Robotique mobile – commande, estimation et applications*, Gif-sur-Yvette, June 8, 2022.

1.4 Thesis outline

The rest of this thesis is organized as follows.

Chapter 2: Distributed Moving Horizon Estimation with pre-estimation over Sensor Network. This chapter provides centralized and distributed state estimation algorithms based on the moving horizon estimation paradigm with a pre-estimation observer. It starts with some notation, definitions, and graph theory, also used in the rest of the thesis. Then, it describes the sensor network with its characteristics, properties, and assumptions. The centralized MHE with pre-estimation is next described, and then the distributed one. The distributed MHE, apart from the local optimization problem that each sensor needs to solve, also explains how to achieve a consensus convergence, the assumptions on the network information exchange, and the distributed algorithm. A numerical example shows the performance of these estimators in terms of computation time and accuracy of the estimates. An extension of the distributed algorithm is also provided. Such extension exploits local systems' observability properties to improve the distributed estimations' convergence. Finally, a Monte Carlo simulations analysis offers the results of this extended algorithm.

Chapter 3: ℓ-step Neighborhood Distributed Moving Horizon Estimation. This chapter follows the idea of the previous one on providing a distributed moving horizon estimation with pre-estimation. This new algorithm is based on a different consensus mechanism by fusing arrival costs, and each local sensor exploits measurements from non-neighbors nodes. It starts by describing the ℓ-step neighborhood set and how each sensor collects information from the sensor network. Then, it explains the proposed distributed moving horizon estimation technique by detailing the local optimization problem and the use of the information diffusion mechanism, how to fuse the arrival costs among neighbors, and the procedure of the distributed algorithm. Finally, two simulation cases provide the results to analyze the performance of the proposed algorithm.

Chapter 4: Constrained DMHE with sporadic measurements for Multi-Vehicle Systems. This chapter aims to apply one of the developed algorithms for the application context of surveillance missions. Indeed, it provides a new distributed moving horizon estimator algorithm that localizes a multi-vehicle system driving in a urban-like environment. It starts explaining the problem under investi-

gation. Then, it introduces the main elements for the algorithm (which has to deal, in this context, with sporadic measurements) and it also shows how to make use of some environment information as constraints. Then, it describes the proposed constrained DMHE with sporadic measurements for Multi-Vehicle Systems and details how to deal with the mentioned sporadic measurements and environment constraints as well as how to adapt the formulation of the optimization problem for this application context. Since this chapter aims at developing an algorithm for a realistic scenario, before going trough practice experiments, it investigates realistic simulations within the ROS and Gazebo environments. Finally, it shows the analysis and the comparison of results obtained by experiments on a real hardware setup.

Chapter 5: Conclusion and perspectives. This chapter finalizes the current thesis by means of concluding remarks and gathers several open directions of this work.

2 - Distributed Moving Horizon Estimation with pre-estimation over Sensor Network

This chapter proposes a new Distributed Moving Horizon Estimation (DMHE) algorithm for state estimation of a discrete-time linear system by a sensor network. The main contribution consists in using a pre-estimating Luenberger observer in the formulation of the local problem to be solved by each sensor. This results in a significant reduction of the computation time while preserving at the same time the accuracy of the estimates. In addition, the state estimate computed by each sensor is capable to converge even under weak observability conditions, which is one of the main advantages of the proposed technique. Moreover, the algorithm has been extended to exploit observability properties of the sensor network. Indeed, these properties of local sensors are used for tuning the weights related to consensus information fusion built on an observability rank-based condition, in order to improve the convergence of the estimation error. Results obtained by simulation examples are provided to compare the performance with existing approaches, in terms of accuracy of the estimates and computation time. The proposed algorithms have been published in one conference (24th International Conference on System Theory, Control and Computing) and one journal paper (System Theory, Control and Computing Journal).

This chapter is structured as follows. Section 2.1 introduces the considered research topic and summarizes the state of the art. Section 2.2 introduces the notations, definitions and assumptions, while Sections 2.3 and 2.4 describe the proposed Centralized MHE and Distributed MHE with pre-estimation, respectively. A simulation example shows the effectiveness of the proposed algorithms in Section 2.5. Furthermore, Section 2.6 describes the extension of the proposed DMHE by equipping it with an observability rank-based weights method. Before concluding remarks in Section 2.8, a Monte Carlo simulations analysis is provided in Section 2.7.

2.1 Introduction

In recent decades, there has been an increasing interest on research about distributed state estimation due to its variety of applications such as target tracking [Albert and Imsland, 2018, Petitti et al., 2011], exploration [Vincent et al., 2008], surveillance [Kong et al., 2009], microgrids [Ghaderyan et al., 2021], etc. The estimation algorithms used in these applications are mainly formulated as centralized fusion architectures, where all the sensors transmit their local measurements to a central unit that processes the provided measurements to update the estimate [Tedesco et al., 2016, Xie et al., 2012, Battistelli et al., 2012]. In general,

the centralized algorithms are not scalable, since with the increasing number of sensors the complexity of the problem to solve increases, too. Furthermore, the central unit cannot efficiently communicate with all sensors for large-scale sensor networks due to physical constraints, e.g., limited communication bandwidth, communication delay. Unlike centralized schemes [Grahn et al., 2017], in the distributed approaches [Hespanha et al., 2007, Rego et al., 2019b, Farina et al., 2010a, Farina et al., 2012] each sensor computes a local estimation using the information acquired only from locally connected neighbors. This can improve robustness to unexpected event (component fault, isolation of a specific area due to security concerns, etc.) exploiting redundancy [Trapiello et al., 2020, Tošić et al., 2013] and also lower the communication burden since data is transmitted among local nodes in the network.

The continuous decreasing costs of sensors is making these applications realizable, even tough there are still open problems to face with. In fact, the distributed algorithms need to have particular properties in order to make them attractive for the industrial community. Indeed, in the context of large-scale systems, the algorithms must be scalable to be able to deal with large networks, need low computation load since low-cost sensors are not powerful devices, have to minimize the utilization of communication resources, etc. In [He et al., 2020], the authors reviewed several works about distributed state estimation over low-cost sensors networks, pointing out their characteristics, advantages, and challenging issues.

In recent years, Moving Horizon Estimation (MHE) techniques and the Distributed MHE counterpart have been used to successfully deal with large sensor networks [Battistelli, 2018]. The first idea of the MHE approach in [Muske et al., 1993] consists in estimating the current state vector by solving a least-square optimization problem penalizing on one hand the deviation between the measurements and the predicted outputs, and on the other hand the distance from the estimated state to some *a priori* information about the state. MHE is a practical strategy for constrained state estimation and a lot of research has been devoted to develop stability guarantees on the estimation error dynamics, e.g., [Rao et al., 2001, Sui et al., 2010, Suwantong et al., 2014]. Although this approach is functional for control engineers offering the freedom to tune the parameters of the cost function, a strength and a weakness of this approach is the use of an optimization problem to be solved within the sampling time. However, this issue becomes critical for large-scale systems. There have been several attempts in trying to reduce the computation load of MHE. One idea is to add a pre-estimating observer in the formulation. The authors of [Sui et al., 2010] proposed a MHE strategy with a Luenberger observer that leads to good performance especially for large estimation horizons. They also provided an optimization problem to tune the parameters minimizing the effects of measurement noise and model errors. The paper [Sui and Johansen, 2014] generalized the formulation using a weight matrix for the penalty function and adding states constraints. A MHE with pre-estimation has been proposed also for non-linear systems [Suwantong et al., 2014]. The authors

of [Farina et al., 2010a] proposed a distributed algorithm for linear systems with the stability proof under weak observability conditions (exploiting a *consensus-on-estimate* and a *consensus weight* term in the DMHE formulation). However, the computation time issue becomes crucial, since usually the network is composed of low-cost sensors.

Based on the results of [Sui and Johansen, 2014, Farina et al., 2010a], this chapter proposes a new DMHE algorithm with pre-estimation presented in [Venturino et al., 2020] and its extension published in [Venturino et al., 2021b]. A pre-estimating Luenberger observer is considered in the formulation of the local problem to be solved by each sensor, resulting in a significant reduction of the computation time. The main contribution of this chapter covers:

- Reducing the computation time required for solving the optimization problem, which allows the use of this type of algorithms for time-sensitive applications;
- Preserving the accuracy of the estimation errors;
- Better tuning the consensus weights associated to the network topology via a new observability rank-based weighted distributed algorithm which takes advantage of the local available information.

Before presenting the developed algorithm, the necessary mathematical background is further proposed.

2.2 Notations, definitions and assumptions

2.2.1 System modeling

Consider the dynamics of the system under observation described by the following discrete-time linear time invariant (LTI) state-space model:

$$x_{t+1} = Ax_t + w_t, \tag{2.1}$$

where $x_t \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state and $w_t \in \mathcal{W} \subseteq \mathbb{R}^n$ is a zero mean white noise with covariance matrix Q.

Remark 1. *In the system* (2.1) *the control input* u_t *is assumed unknown and incorporated in* w_t*, but in the case that this control input* u_t *is known then the system can be considered as*

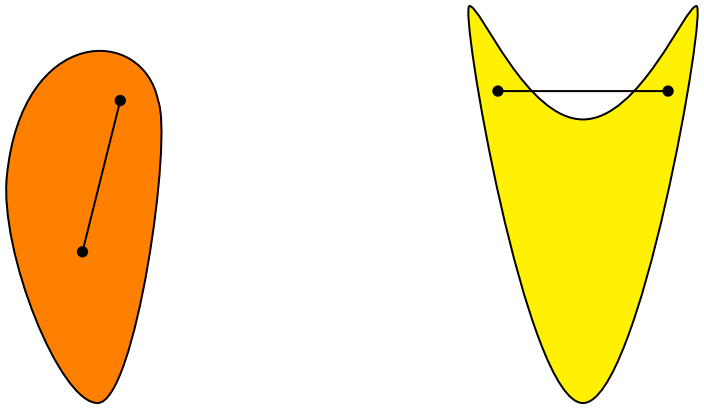
$$x_{t+1} = Ax_t + Bu_t + w_t. \tag{2.2}$$

Assumption 2.1 — **Convex constraints set**

The sets $\mathcal{X}$ *and* $\mathcal{W}$ *are assumed to be convex.*

Definition 2.1 **Convex set**

*A set $\mathcal{S} \in \mathbb{R}^n$ is said to be **convex** if, however we take two points $s_1, s_2 \in \mathcal{S}$, the line segment that joins them is entirely contained in $\mathcal{S}$, i.e.*
$\forall s_1, s_2 \in \mathcal{S}, \forall\ \lambda\ \in\ [0,1] \Rightarrow \lambda s_1 + (1-\lambda) s_2 \in \mathcal{S}$.



(a) A convex set. (b) A non-convex set.

Figure 2.1: Convex and non-convex set examples.

Figures 2.1a and 2.1b show, respectively, convex and non-convex set examples in a 2-dimensional space.

Assumption 2.2 **Unknown initial state**

The initial state x_0 is assumed to be unknown and modeled by a random variable of mean μ_0 and covariance matrix Π_0.

The measurements are performed by n_S heterogeneous sensors:

$$y_t^i = C^i x_t + v_t^i, \quad i = 1, \ldots, n_S, \tag{2.3}$$

where $y_t^i \in \mathbb{R}^{p^i}$ and $v_t^i \in \mathbb{R}^{p^i}$ is a zero mean white noise with covariance matrix R^i.

Remark 2 (Notation for sensor indexing)**.** *The superscript $(\cdot)^i$ indicates a variable that refers to the sensor i.*

To facilitate the understanding of the sensor network and the distributed algorithms, several notions from graph theory need to be introduced.

2.2.2 Graph theory tools

For the sake of clarity, some notions and definitions from graph theory are recalled in this subsection.

Definition 2.2 **Undirected graph**

*An **undirected graph** (usually called **graph**) $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is a relational structure composed of a finite number of nodes (or vertices) $\mathcal{N}$, and of a finite number of edges (or arcs) $\mathcal{E}$ that link the nodes among each other.*

Definition 2.3 **Directed graph**

*A graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is a **directed graph** (or **digraph**) if $\mathcal{E}$ is a set of ordered pairs of nodes.*

(a) An undirected graph. (b) A directed graph.

Figure 2.2: Undirected and directed graphs examples.

Figure 2.2 illustrates examples of undirected graph (Figure 2.2a) and digraph (Figure 2.2b).

Definition 2.4 **Weighted graph**

*A graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is a **weighted graph** if weights are assigned to its nodes $\mathcal{N}$ or edges $\mathcal{E}$.*

The graphs could be also weighted. An example is provided in Figure 2.3.

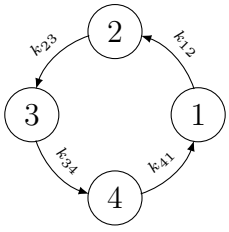


Figure 2.3: Weighted digraph example.

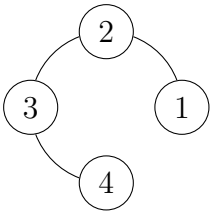
It is also important to distinguish *strongly connected graphs* from *complete graphs* in order to better describe some properties of the topology of the sensor network that will be used later on.

Definition 2.5 **Strongly connected graph**

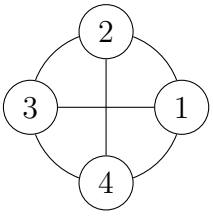
*A graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is a **strongly connected graph** if every node is reachable from every other node.*

Definition 2.6 **Complete graph**

*A graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is a **complete graph** if every pair of distinct nodes is connected by a unique edge.*



(a) A strongly connected graph.



(b) A complete graph.

Figure 2.4: Strongly connected and complete graphs examples.

After recalling these basic definitions, the sensor network is further described.

2.2.3 Sensor network

The sensor network is described by a digraph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where the nodes $\mathcal{N} = \{1, 2, \ldots, n_S\}$ represent the sensors and the edge $(j, i) \in \mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ represents the communication link from sensor j to sensor i. We assume that all the nodes have a self-loop $(i, i) \in \mathcal{E}$, $\forall i \in \mathcal{N}$. The neighborhood $\mathcal{N}^i$ of the sensor i is $\mathcal{N}^i = \{j \in \mathcal{N} : (j, i) \in \mathcal{E}\}$ which defines the set of nodes $j \in \mathcal{N}$ able to send information to sensor i. We denote by $n_S^i = \text{card}(\mathcal{N}^i)$ the number of nodes $j \in \mathcal{N}^i$.

Definition 2.7 **Peer-to-peer sensor network**

*A **peer-to-peer** sensor network is a group of sensors that are linked together with equal permissions and responsibilities for processing data.*

In this manuscript all proposed algorithms deal with peer-to-peer sensor network. Before going to the details of the algorithm it is necessary to introduce some properties about the topology of the sensor network.

Definition 2.8 **Static topology sensor network**

*In a **static topology** sensor network, the communication links among nodes are fixed and cannot be modified.*

Definition 2.9 **Dynamic topology sensor network**

*In a **dynamic topology** sensor network, the communication links among nodes can variate over time.*

Assumption 2.3 **Static topology**

The topology of the sensor network is static.

The following definition allows to describe the weights that will be assigned to the communication links of the static topology.

Definition 2.10 **Stochastic matrix**

*A **stochastic matrix** $A = (a_{ij})$ is a $n \times n$ matrix which has non-negative elements and in which the sum of the elements over each row (or each column) is equal to* 1*:*

$$a_{ij} \geqslant 0, \quad \forall i, j = 1, \ldots, n;$$
$$\sum_{j=1}^{n} a_{ij} = 1, \quad \forall i = 1, \ldots, n$$

or similarly by adding up on the columns.

$$\sum_{i=1}^{n} a_{ij} = 1, \quad \forall j = 1, \ldots, n.$$

Under Assumption 2.3 a constant stochastic matrix $K \in \mathbb{R}^{n_S \times n_S}$ can be associated to the digraph $\mathcal{G}$ such that the elements weight its edges:

$$k_{ij} > 0 \text{ if } (j, i) \in \mathcal{E}, \tag{2.4a}$$
$$k_{ij} = 0 \text{ otherwise}, \tag{2.4b}$$
$$\sum_{j=1}^{n_S} k_{ij} = 1, \quad \forall i = 1, \ldots, n_S. \tag{2.4c}$$

This matrix will be used to compute the consensus terms in the DMHE algorithm described in Section 2.4. Section 2.6 will present a method that determines the value of each element k_{ij}, compatible with the weighted digraph $\mathcal{G}$, exploiting observability properties. This method contributes to enhance the estimation accuracy.

Before introducing the centralized and distributed MHE, it is necessary to distinguish the different types of information involved in such algorithms.

2.2.4 Local, regional and collective data

In order to recognize *local*, *regional* and *collective* information (as in [Farina et al., 2010a]) a convenient notation is introduced.

Definition 2.11 **Local, regional and collective information**

*For a considered sensor i, an information is said **local** if it is related only to the node i. It is said **regional** concerning sensor i if it is related to the nodes in its neighborhood $\mathcal{N}^i$. Finally, an information is said **collective** when the entire network is involved.*

Therefore, for the sake of clarity, we distinguish these information using different notations for *local*, *regional* and *collective* variables.

Remark 3. *Given a variable z, then z^i, $\bar{z}^i$ and $\mathbf{z}$ represent local, regional and collective data, respectively.*

For example, consider the sensor i and its neighborhood $\mathcal{N}^i = \{j_1, \ldots, j_{n_S^i}\}$, composed of n_S^i sensors, then its regional measurements are:

$$\bar{y}_t^i = \bar{C}^i x_t + \bar{v}_t^i, \tag{2.5}$$

with the output vector $\bar{y}_t^i = [(y_t^{j_1})^\top \ \ldots \ (y_t^{j_{n_S^i}})^\top]^\top \in \mathbb{R}^{\bar{p}^i}$ of dimension $\bar{p}^i = \sum_{i \in \mathcal{N}^i} p^i$, the output matrix $\bar{C}^i = [(C^{j_1})^\top \ \ldots \ (C^{j_{n_S^i}})^\top]^\top \in \mathbb{R}^{\bar{p}^i \times n}$, and the measurement noise vector $\bar{v}_t^i = [(v_t^{j_1})^\top \ \ldots \ (v_t^{j_{n_S^i}})^\top]^\top \in \mathbb{R}^{\bar{p}^i}$. In addition, we denote by $\bar{R}^i$, the covariance matrix related to the regional noise vector $\bar{v}_t^i$ of sensor i, i.e., $\bar{R}^i = diag(R^{j_1}, \ldots, R^{j_{n_S^i}})$. For the sake of exhaustiveness, the following are examples of collective information. Thus, the collective output is defined as follows:

$$\mathbf{y}_t = \mathbf{C} x_t + \mathbf{v}_t, \tag{2.6}$$

with the output vector $\mathbf{y}_t = [(y_t^1)^\top \ \ldots \ (y_t^{n_S})^\top]^\top \in \mathbb{R}^p$ of dimension $p = \sum_{i \in \mathcal{N}} p^i$, the output matrix $\mathbf{C} = [(C^1)^\top \ \ldots \ (C^{n_S})^\top]^\top \in \mathbb{R}^{p \times n}$ and the measurement noise vector $\mathbf{v}_t = [(v_t^1)^\top \ \ldots \ (v_t^{n_S})^\top]^\top \in \mathbb{R}^p$.

According to this terminology, three different observability notions can be defined.

Definition 2.12 **Local, regional and collective observability**

*The system is **locally observable** by sensor i if the pair (A, C^i) is observable.*
*The system is **regionally observable** by sensor i if the pair $(A, \bar{C}^i)$ is observable.*
*The system is **collectively observable** if the pair $(A, \mathbf{C})$ is observable.*

In the remaining of this manuscript, by abuse of language, one will write that *sensor i is locally (respectively regionally) observable* when the system is locally (respectively regionally) observable by sensor i.

These definitions will be helpful for the algorithms proposed in the next sections.

2.3 Centralized MHE with pre-estimation

Moving Horizon Estimation with pre-estimation is based on the idea in [Sui et al., 2010] and [Sui and Johansen, 2014], where the authors proposed a centralized MHE formulation relying on a Luenberger observer. This centralized scheme makes use of the collective information solely.

For a given horizon length $N \geqslant 1$, the proposed strategy determines the *estimate $\hat{x}_{t|t}$ of the state at time t*, by solving the constrained minimization problem, hereafter denoted by MHE$_{pre}$:

$$\hat{x}_{t-N|t} = \arg\min_{\hat{x}_{t-N}} \quad J(t-N, t, \hat{x}_{t-N}, \hat{\mathbf{v}}, \Gamma_{t-N}) \tag{2.7}$$

$$\text{s.t.} \quad \hat{x}_{k+1} = A\hat{x}_k + L\hat{\mathbf{v}}_k, \qquad \forall k = t-N, \dots, t-1, \tag{2.8}$$

$$\hat{\mathbf{y}}_k = \mathbf{C}\hat{x}_k + \hat{\mathbf{v}}_k, \qquad \forall k = t-N, \dots, t, \tag{2.9}$$

$$\hat{x}_k \in \mathcal{X}, \qquad \forall k = t-N, \dots, t. \tag{2.10}$$

The gain matrix $L \in \mathbb{R}^{n \times p}$ in (2.8), with $p = \sum_{i=1}^{n_S} p^i$, is calculated such that $\Phi = A - L\mathbf{C}$ is Schur stable. The pre-estimation replaces the input sequence to be estimated, thus leads to have less optimization parameters involved in the minimization problem, hence less computation time. At the same time, it helps to preserve the accuracy of the estimates.

The resulting estimated state $\hat{x}_{t|t}$ at the time instant t is obtained by the equation (2.8) and using the solution of the optimization problem $\hat{x}_{t-N|t}$ as initial condition. Indeed, the entire sequence of the estimated state within the horizon window $\{\hat{x}_{k|t}\}_{k=t-N}^{t}$ can be obtained. Explicitly, it leads to:

$$\left\{\hat{x}_{t-N|t} \quad \hat{x}_{t-N+1|t} \quad \hat{x}_{t-N+2|t} \quad \dots \quad \hat{x}_{t|t}\right\}. \tag{2.11}$$

Remark 4. *For the sake of clarity, it is important to distinguish a generic optimization parameter of the minimization problem, indicated with $\hat{x}_t$, from the optimal solution of this problem obtained at time instant t, indicated with $\hat{x}_{t|t}$.*

The cost function J is given by:

$$J(t-N, t, \hat{x}_{t-N}, \hat{\mathbf{v}}, \Gamma_{t-N}) = \sum_{k=t-N}^{t} \|\hat{\mathbf{v}}_k\|^2_{\mathbf{R}^{-1}} + \Gamma_{t-N}(\hat{x}_{t-N}, \hat{x}_{t-N|t-1}), \tag{2.12}$$

where $\mathbf{R} = diag(R^1, \dots, R^{n_S})$ with n_S the number of sensors. The so called *initial penalty* function $\Gamma_{t-N}(\cdot)$ in (2.12) is defined as follows:

$$\Gamma_{t-N}(\hat{x}_{t-N}, \hat{x}_{t-N|t-1}) = \left\|\hat{x}_{t-N} - \hat{x}_{t-N|t-1}\right\|^2_{\Pi^{-1}_{t-N|t-1}}, \tag{2.13}$$

where $\hat{x}_{t-N|t-1}$ is the second estimated state of the sequence computed at the previous time $t-1$. In order to clarify what is $\hat{x}_{t-N|t-1}$, consider the optimal

sequence obtained at time t as in (2.11), then the one obtained at time $t-1$ is the following:

$$\left\{\hat{x}_{t-N-1|t-1} \quad \hat{x}_{t-N|t-1} \quad \hat{x}_{t-N+1|t-1} \quad \dots \quad \hat{x}_{t-1|t-1}\right\} \tag{2.14}$$

where it is evident that $\hat{x}_{t-N|t-1}$ is the second element of the sequence (2.14).

The positive-definite symmetric weighting matrix $\Pi_{t-N|t-1}$ in (2.13) is the unique solution of the following discrete-time algebraic Riccati equation [Jazwinski, 2007]:

$$\begin{aligned}\Pi_{t-N|t-1} = Q + A\Pi_{t-N-1|t-2}A^\top - \\ A\Pi_{t-N-1|t-2}\mathbf{C}^\top(\mathbf{R}+\mathbf{C}\Pi_{t-N-1|t-2}\mathbf{C}^\top)^{-1}\mathbf{C}\Pi_{t-N-1|t-2}A^\top,\end{aligned} \tag{2.15}$$

subject to the initial condition Π_0. When $t \leqslant N$, it is possible to set $N = t$, leading to use the algorithm even if there are not even enough measurements to fill the sequence of past measurements within the horizon window.

The next section describes the distributed counterpart of this algorithm, which is one contribution of this chapter.

2.4 Distributed MHE with pre-estimation

In [Farina et al., 2010a] the authors proposed a Distributed Moving Horizon Estimation method to estimate the state and the input of the model (2.1). In their strategy, the optimization problem to be solved online at time t involves the computation of the state trajectory over the past horizon. As in classical MHE schemes, this computation is done by forward propagating the state, from its initial condition at $t-N$, using the dynamic model of the system. This can accumulate the estimation error, especially when N is large and the system is unstable. In this section a new DMHE strategy with pre-estimation by introducing a Luenberger observer in its formulation is proposed. This will mitigate the effect of model uncertainty in the *a priori* estimate and thus will contribute to enhance the estimation accuracy. This improvement will also reduce the computation time required to solve the optimization problem, because the number of optimization variables involved is lower. Indeed the algorithm of [Farina et al., 2010a] also estimates the unknown input sequence $\{w_k\}_{k=t-N}^{t-1}$. Further details are given in the next sections.

2.4.1 Local minimization problem

This subsection formulates the proposed DMHE scheme, from now on denoted by DMHE$_{pre}$, where each sensor $i \in \mathcal{N}$ solves its own local moving horizon estimation problem based on regional measurements $\bar{y}_t^i$ and some shared information[1] among the neighborhood $\mathcal{N}^i$. For a given estimation horizon $N \geqslant 1$, each node

[1]Details about the shared information are given later on.

$i \in \mathcal{N}$ at time t determines the estimate $\hat{x}^i_{t|t}$ of the state x_t by solving the following constrained minimization problem with pre-estimation, hereafter denoted by DMHE$_{i-pre}$:

$$\hat{x}^i_{t-N|t} = \arg \min_{\hat{x}^i_{t-N}} \quad J^i(t-N, t, \hat{x}^i_{t-N}, \hat{\bar{v}}^i, \Gamma^i_{t-N}) \tag{2.16}$$

$$\text{s.t.} \qquad \hat{x}^i_{k+1} = A\hat{x}^i_k + L^i\hat{\bar{v}}^i_k, \qquad \forall k = t-N, \dots, t-1, \tag{2.17}$$

$$\bar{y}^i_k = \bar{C}^i\hat{x}^i_k + \hat{\bar{v}}^i_k, \qquad \forall k = t-N, \dots, t, \tag{2.18}$$

$$\hat{x}^i_k \in \mathcal{X}, \qquad \forall k = t-N, \dots, t. \tag{2.19}$$

Notice that the equations remind the ones of the centralized MHE$_{pre}$ problem in (2.7) but using regional information instead. The Luenberger gain L^i is computed such that $\Phi^i = A - L^i\bar{C}^i$ is Schur stable when the sensor i is regionally observable. Otherwise, as *extrema ratio*, L^i can be computed in order to minimize the propagation of the error along the prediction horizon by keeping the spectrum radius of Φ^i the smallest possible.

A quadratic objective function J^i is considered:

$$J^i(\cdot) = \sum_{k=t-N}^{t} \left\|\hat{\bar{v}}^i_k\right\|^2_{(\bar{R}^i)^{-1}} + \Gamma^i_{t-N}(\hat{x}^i_{t-N}, \hat{\bar{x}}^i_{t-N|t-1}), \tag{2.20}$$

where the initial penalty function $\Gamma^i_{t-N}(\hat{x}^i_{t-N}, \hat{\bar{x}}^i_{t-N|t-1})$ in (2.20) defined as follows:

$$\Gamma^i_{t-N}(\cdot) = \left\|\hat{x}^i_{t-N} - \hat{\bar{x}}^i_{t-N|t-1}\right\|^2_{(\bar{\Pi}^i_{t-N|t-1})^{-1}}, \tag{2.21}$$

involves two consensus terms described below.

1st consensus term. Denote by $\hat{\bar{x}}^i_{t-N|t-1}$ the *weighted mean state estimation* computed by the neighborhood $\mathcal{N}^i$ as follows:

$$\hat{\bar{x}}^i_{t-N|t-1} = \sum_{j \in \mathcal{N}^i} k_{ij}\hat{x}^j_{t-N|t-1}, \tag{2.22}$$

where $\hat{x}^j_{t-N|t-1}$ is the second estimated state in the sequence computed at the previous time by sensor j as explained for the sequence in (2.14). Notice that the penalty function Γ^i_{t-N} includes a *consensus-on-estimates* term in the sense that it penalizes deviations of $\hat{x}^i_{t-N}$ from $\hat{\bar{x}}^i_{t-N|t-1}$. The penalty function helps to improve the accuracy of the local estimates and it is necessary to guarantee convergence of the state estimates to the state of the observed system even if it lacks of regional observability [Farina et al., 2010a].

2nd consensus term. The positive definite matrix $\bar{\Pi}^i_{t-N|t-1}$ is computed as in [Farina et al., 2010a]. For the sake of completeness, we recall here the procedure to compute it by:

$$\bar{\Pi}^i_{t-N|t-1} = \sum_{j \in \mathcal{N}^i} n^j_S k^2_{ij} \Pi^j_{t-N|t-1}, \tag{2.23}$$

where the update of $\Pi^i_{t-N|t-1}$ is performed by the sensor i on the basis of regionally available information. Notice that $\bar{\Pi}^i_{t-N|t-1}$ is the *weighted mean matrix* of all matrices $\Pi^j_{t-N|t-1}$ in the neighborhood $\mathcal{N}^i$. In particular, the matrix $\Pi^i_{t-N|t-1}$, with $i \in \mathcal{N}$, is given by one iteration of the difference Riccati equation associated to a Kalman filter for the system:

$$\begin{cases} x_{t-N} = Ax_{t-N-1} + w_{t-N-1} \\ \bar{z}^i_{t-N} = \bar{\mathcal{O}}^i_N x_{t-N} + \bar{V}^i_{t-N} \end{cases}$$

where $\bar{V}^i_{t-N}$ represents the measurements noise and $\bar{\mathcal{O}}^i_N$ defines the i-th sensor regional observability matrix:

$$\bar{\mathcal{O}}^i_N = \left[(\bar{C}^i)^\top \quad (\bar{C}^i A)^\top \quad \cdots \quad (\bar{C}^i A^{N-1})^\top\right]^\top. \tag{2.24}$$

Then, defining:

$$\mathcal{C}^i_N = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \bar{C}^i & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{C}^i A^{N-2} & \bar{C}^i A^{N-3} & \cdots & \bar{C}^i \end{bmatrix} \in \mathbb{R}^{\bar{p}^i N \times n(N-1)}, \tag{2.25}$$

$$\bar{R}^i_N = \mathsf{diag}(\bar{R}^i, \ldots, \bar{R}^i) \in \mathbb{R}^{\bar{p}^i N \times \bar{p}^i N}, \tag{2.26}$$

$$Q_{N-1} = \mathsf{diag}(Q, \ldots, Q) \in \mathbb{R}^{n(N-1) \times n(N-1)}, \tag{2.27}$$

$$\mathsf{Cov}[\bar{V}^i_t] = \bar{R}^{*i}_N = \bar{R}^i_N + \mathcal{C}^i_N Q_{N-1} (\mathcal{C}^i_N)^\top, \tag{2.28}$$

and setting the covariance of the estimate $\hat{x}^i_{t-N-1}$ like in the Kalman filter approach:

$$\Pi^{*i}_{t-N-1|t-2} = \left(\left(\bar{\Pi}^i_{t-N-1|t-2}\right)^{-1} + (\bar{C}^i)^\top (\bar{R}^i)^{-1} \bar{C}^i\right)^{-1}, \tag{2.29}$$

the resulting Riccati recursive equation is given by:

$$\begin{aligned} &\Pi^i_{t-N|t-1} = A\Pi^{*i}_{t-N-1|t-2} A^\top + Q - \\ &A\Pi^{*i}_{t-N-1|t-2} \left(\bar{\mathcal{O}}^i_N\right)^\top \left(\bar{\mathcal{O}}^i_N \Pi^{*i}_{t-N-1|t-2} \left(\bar{\mathcal{O}}^i_N\right)^\top + \bar{R}^{*i}_N\right)^{-1} \bar{\mathcal{O}}^i_N \Pi^{*i}_{t-N-1|t-2} A^\top. \end{aligned} \tag{2.30}$$

Since the communication network topology is assumed to be time-invariant, these equations could be computed off-line. However, once the matrices $\Pi^i_{t-N|t-1}$ have been computed, a *consensus weights update* is performed in order to compute the matrices $\bar{\Pi}^i_{t-N|t-1}$ according to (2.23).

Remark 5. *In the proposed DMHE formulation the sequence of the input noise $\{w_k\}_{k=t-N}^{t-1}$ are no longer considered as optimization parameters, contrary to the DMHE of [Farina et al., 2010a]. This allows to reduce the computation time required to solve the optimization problem.*

2.4.2 Network information exchange

It is worth highlighting the way each node exchanges information within its neighborhood. For this purpose, we recall here some assumptions that play a major role.

Assumption 2.4 **Network information exchange assumptions**

The following assumptions considered with respect to the sensor network:

a) The sensor network can be composed of heterogeneous sensors;

b) The sensors characteristics (noise covariance and type of measurements) are not time-varying;

c) The network topology is not time-varying;

d) Each sensor i knows its neighborhood $\mathcal{N}^i$;

e) There is no time delay nor packet loss in the communication network.

The Assumption 2.4 implies that the matrices C^i in (2.3) can be different for all $i \in \mathcal{N}$. Since the neighborhood $\mathcal{N}^i$ is known *a priori*, it is not necessary to exchange the information on the matrices C^i and R^i at each time. Moreover, this allows one to compute off-line the Luenberger gains L^i.

2.4.3 DMHE procedure

Finally, the procedure of the proposed distributed scheme is described in Algorithm 1.

It is evident that the steps 10, 15 and 19 involve synchronization among neighbors sensors, but these steps could be rearranged to have just one synchronization in the procedure of exchanging information. However, for clarity reasons with respect to calculation details, they have been described this way. The easiest way to reorganize the steps and have one synchronization is to put back the steps 15 and 19 at the time of step 10, this leads to **receive** in one step: from the neighbors $j \in \mathcal{N}^i$ the collected data in the step 9, the matrices $\Pi^j_{t-N|t-1}$ and their estimates $\hat{x}^j_{t-N+1|t}$.

The following section will evaluate the proposed centralized and distributed algorithms via a numerical example.

Algorithm 1 DMHE$_{pre}$ procedure

Off-line: $\forall i \in \mathcal{N}$
  receive from the nodes $j \in \mathcal{N}^i$: L^j, C^j, R^j
  compute the pre-observer gain L^i
  store the *a priori* initial estimation $\hat{x}^i_{0|0} = \hat{x}_0 = \mu_0$ of x_0, where μ_0 is given, and the covariance matrix Π_0 of x_0
Initialization: $\forall i \in \mathcal{N}$, at the first time step $t = 0$
  collect a first local measurement y^i_0
  receive from the neighbors $j \in \mathcal{N}^i$ their measurements y^j_0
Online: $\forall i \in \mathcal{N}$, $\forall t > 0$
  collect the local measurement y^i_t
  receive from the neighbors $j \in \mathcal{N}^i$ the collected data in the step 9
  if $1 \leqslant t \leqslant N$ **then**
    set the horizon length $N = t$, the covariance matrix $\bar{\Pi}^i_{t-N|t-1} = \bar{\Pi}^i_{0|t-1} = \Pi_0$ and the *a priori* initial estimation state $\hat{x}^i_{t-N|t-1} = \hat{x}^i_{0|t-1}$
  else
    compute $\Pi^i_{t-N|t-1}$ according to (2.28), (2.29) and (2.30)
    receive $\Pi^j_{t-N|t-1}$ from the nodes $j \in \mathcal{N}^i$
    compute $\bar{\Pi}^i_{t-N|t-1}$ according to (2.23)
  solve the local optimization problem of DMHE, minimizing J^i as in (2.20) and (2.21) subject to the constraints (2.17)-(2.19)
  store the solution $\hat{x}^i_{t-N|t}$ and the corresponding estimate $\hat{x}^i_{t|t}$
  receive from the neighbors $j \in \mathcal{N}^i$ their estimates $\hat{x}^j_{t-N+1|t}$

2.5 Example

In this section, the effectiveness of the proposed DMHE algorithm with pre-estimation is investigated. In order to evaluate its performance, it is compared to the centralized MHE of [Rao et al., 2001] as well as the DMHE algorithm of [Farina et al., 2010a]. To this end, we consider the system in [Farina et al., 2010a], recalled below:

$$x_{t+1} = \begin{bmatrix} 0.9962 & 0.1949 & 0 & 0 \\ -0.1949 & 0.3819 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1.21 & 1.98 \end{bmatrix} x_t + w_t, \tag{2.31}$$

where $x_t = \begin{bmatrix} x_{1,t} & x_{2,t} & x_{3,t} & x_{4,t} \end{bmatrix}^\top \in \mathcal{X} = \mathbb{R}^4$ is the state vector and $w_t \in \mathbb{R}^4$ is a white noise with covariance $Q = \text{diag}(0.0012,\ 0.038,\ 0.0012,\ 0.038)$. Notice that the system is unstable since the eigenvalues of A are 0.9264, 0.4517, $0.99 \pm 0.4795i$ and $|0.99 \pm 0.4795i| > 1$.

The initial values of the algorithms are set as $\mu_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top$, $\Pi_0 = I_4$ and $N = 5$.

To compare the results of the considered algorithms, $n_S = 4$ sensors are used both for the centralized and the distributed cases. For the centralized schemes,

the following measurement equation is considered:

$$y_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_t + v_t,$$

with $\text{Var}(v_t) = R = I_4$, while, for the distributed schemes, the following measurement equations are used:

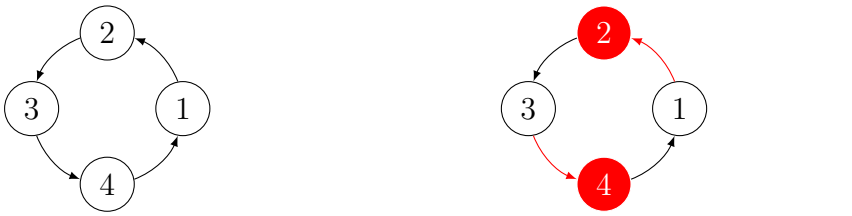
$$\begin{aligned} y_t^i &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x_t + v_t^i \quad \text{if } i \in \{1, 2\}, \\ y_t^i &= \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} x_t + v_t^i \quad \text{if } i \in \{3, 4\}, \end{aligned}$$

where $\text{Var}(v_t^i) = R^i = 1$, $i = 1, \ldots, 4$. Notice that the superscript i in R^i clearly indicates its association with sensor i, i.e. R^i does not mean R power i. The nodes are connected as reported by the graph in Figure 2.5a and the associated matrix is defined as follows:

$$K = \begin{bmatrix} 0.5 & 0 & 0 & 0.5 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0.5 \end{bmatrix}. \tag{2.32}$$

Notice that the sensors 2 and 4 are not observable. The weights k_{ij} are then chosen as in [Xiao et al., 2005], called *Metropolis weights* described below:

$$k_{ij} = \begin{cases} \frac{1}{1+\max\{n_S^i, n_S^j\}} & \text{if } \{i, j\} \in \mathcal{E} \\ 1 - \sum_{\{i,k\} \in \mathcal{E}} k_{ik} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{2.33}$$



(a) Communication network.

(b) Sensor network highlighting (in red) non observable sensors.

Figure 2.5: Topology of the sensor network.

As pointed out in [Farina et al., 2010a] and recalled next, this example is challenging due to the non regional observability of some of the sensors. In fact, it can be noticed that the information available to sensor 1 concerns the first and third states of the system, i.e. $x_{1,t}$, which is directly measured, and $x_{3,t}$, which is transmitted by sensor 4. Likewise, the information available to sensor 2 consists

of $x_{1,t}$, directly measured, and $x_{1,t}$, transmitted by sensor 1. The same applies to the sensors 3 and 4, which have information about $(x_{1,t}, x_{3,t})$ and $(x_{3,t}, x_{3,t})$, respectively. Consequently, the sensors 1 and 3 are regionally observable, while the sensors 2 and 4 are not, since the pairs $(A, \bar{C}^2)$ and $(A, \bar{C}^4)$ are not observable, see Fig 2.5b highlighting (in red) non observable sensors.

The Luenberger gains, both for MHE$_{pre}$ and DMHE$_{pre}$, respectively the proposed centralized and distributed MHE algorithms, have been chosen such that the eigenvalues of $\Phi = A - LC$, for the centralized scheme, and $\Phi^i = A - L^i\bar{C}^i$ for the distributed one, are equals to the values shown in the Table 2.1. These values ensure that matrices Φ, Φ^1 and Φ^3 are Schur stable, and minimize the spectrum radius of Φ^2 and Φ^4. The eigenvalues λ_k, $\forall k = 1, \ldots, 4$, of matrices Φ, Φ^1 and Φ^3, are chosen arbitrarily with the intention to insure their stability property. Most probably it is possible to have optimum values, according to some criteria, in order to have better estimation accuracy, but this is out of the scope of this chapter.

Table 2.1: Eigenvalues of $\Phi = A - LC$ and $\Phi^i = A - L^i\bar{C}^i$.

	λ_1	λ_2	λ_3	λ_4
MHE$_{pre}$	0.9	0.6	0.7	0.8
DMHE$_{1-pre}$	0.9	0.6	0.7	0.8
DMHE$_{2-pre}$	0.45	0.58	$0.99 + 0.48i$	$0.99 - 0.48i$
DMHE$_{3-pre}$	0.9	0.6	0.7	0.8
DMHE$_{4-pre}$	0.93	0.45	$1.17 + 0.98i$	$1.17 - 0.98i$

We consider two different performance metrics for the evaluation of the algorithms: the computation time τ_t (for which it has been examined the sum over the simulation duration, the minimum and the maximum value) and the Root Mean Square Error (RMSE):

$$\text{RMSE} = \left(\sum_{t=t_c}^{t_f} \frac{\|e_t\|^2}{t_f - t_c} \right)^{\frac{1}{2}}, \tag{2.34}$$

where t_c is the convergence instant of the algorithms, t_f is the final time instant, with $t_f \geqslant t_c$, and $e_t = x_t - \hat{x}_{t|t}$ is the estimation error, i.e. the error between the real state x_t and the estimated state $\hat{x}_{t|t}$. In this example, the sampling period T_s is chosen to be $T_s = 1$ s, while $t_f = 20$ s corresponds to the simulation duration.

Remark 6. *The algorithm is considered converged when the estimation error e_t remains below than a given threshold.*

The simulation has been carried out by using a setup implemented within the MATLAB R2019b environment and the solver *LMI Lab* in YALMIP toolbox

[Löfberg, 2004] over a Linux Ubuntu 20.04 PC equipped with an Intel Core i7-7700HQ, 2.80 GHz.

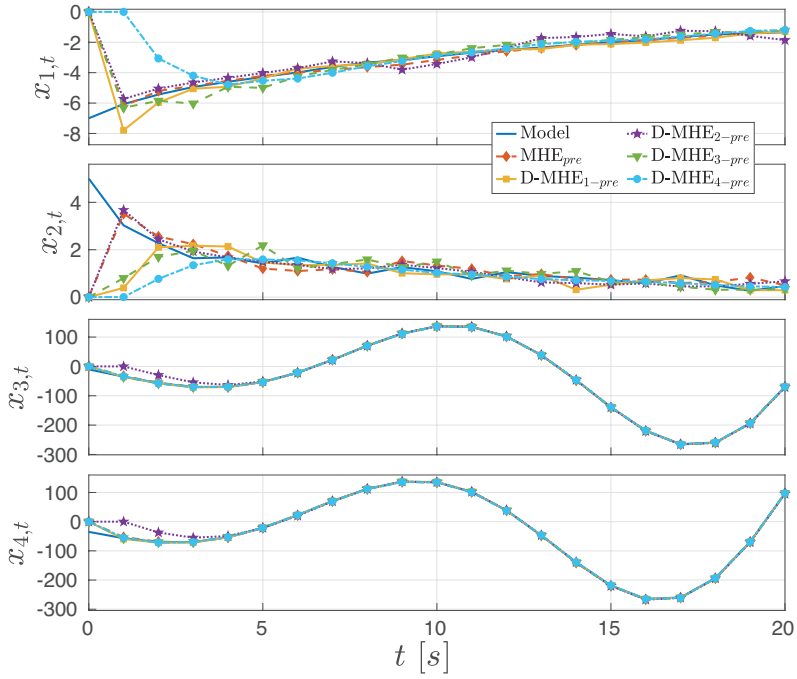


Figure 2.6: Components of the states $x_t = [x_{1,t}\ x_{2,t}\ x_{3,t}\ x_{4,t}]^\top$ and the estimates $\hat{x}_t = [\hat{x}_{1,t}\ \hat{x}_{2,t}\ \hat{x}_{3,t}\ \hat{x}_{4,t}]^\top$ computed by the MHE$_{pre}$ and DMHE$_{pre}$ algorithms.

Figure 2.6 shows the evolution of the system (2.31) and the estimates using the MHE$_{pre}$ and DMHE$_{pre}$ algorithms, i.e. the centralized MHE technique with pre-estimation and the distributed MHE technique with pre-estimation. It is worth noticing that the last two states have unstable dynamics and, thus, are more significant in terms of magnitude with respect to the first two. Nevertheless, one can consider that the convergence time is approximately $t_c = 5$ s. Notice that DMHE$_{pre}$ refers to the global distributed procedure, while DMHE$_{i-pre}$ refers to the local constrained minimization problem solved by sensor i.

Figure 2.7 shows the estimation errors produced by the algorithms MHE$_{pre}$ and DMHE$_{pre}$. It is noticeable that the MHE$_{pre}$ (in red) compensates the initial estimation error very fast, within a few iterations, since the system is collective observable. Regarding DMHE$_{pre}$, the estimates produced by sensor 2 depicted in purple (respectively 4 depicted in cyan), relative to the states $x_{3,t}$, $x_{4,t}$ (respectively $x_{1,t}$, $x_{2,t}$) exhibit large errors for $t < 5$. In fact, as pointed out before, these states cannot be observed by these sensors using regional information. Despite this, all the estimation errors tend to converge to the same values, due to the consensus terms in the DMHE$_{pre}$ scheme. Thus, the proposed distributed state estimation algorithm presents similar accuracy to its centralized counterpart in steady state.

For the sake of clarity, in Figure 2.6 and Figure 2.7 the estimates and the

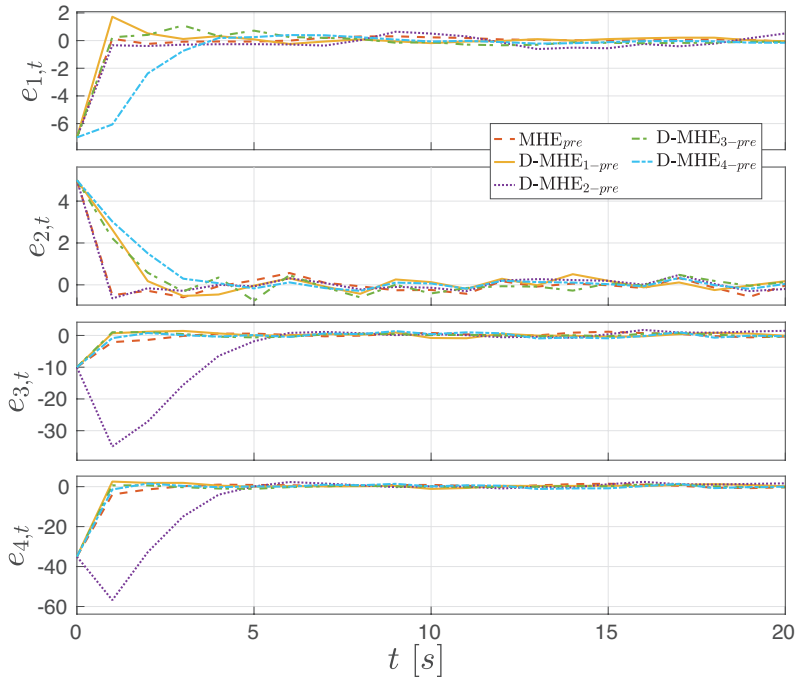


Figure 2.7: Components of the estimation error $e_t = [e_{1,t}\ e_{2,t}\ e_{3,t}\ e_{4,t}]^\top = x_t - \hat{x}_{t|t}$ of the MHE$_{pre}$ and DMHE$_{pre}$ algorithms.

estimation errors, respectively, from MHE (centralized MHE [Rao et al., 2001]) and DMHE (distributed MHE [Farina et al., 2010a]) are not shown. It is also because e_t and x_t from MHE and DMHE are very similar with the ones from MHE$_{pre}$ and DMHE$_{pre}$, respectively, as it can be checked by comparing Figure 2.6 and Figure 2.8.

Table 2.2 summarizes the performance metrics that have been taken into account for quantitative comparison. In particular, the computation time is always lower for our proposed algorithms, i.e. the centralized and distributed MHE with pre-estimation (MHE$_{pre}$, DMHE$_{1-pre}$, ..., DMHE$_{4-pre}$ – green rows), with respect to the ones without pre-estimation (MHE, DMHE$_1$, ..., DMHE$_4$ – white rows). This not comes at a cost in terms of accuracy, because the RMSEs produced are similar among them, see the column at the right. Moreover, the maximum computation time for the algorithms without pre-estimation is greater than the sampling period. This means that there are some time steps where real time feasibility is not obtained. This can be checked on Figure 2.9, indeed, from time $t \geqslant 10$ s, it is visible that the algorithms without pre-estimation take more than 1 s to estimate, which is more than the sampling period $T_s = 1$ s. The pre-estimation enables to reduce the computation time by a factor close to 5 and would enable real time implementation without requiring fast optimization or *ad hoc* implementation.

Figure 2.9 shows the computation time τ_t for each time instant of all algorithms. In this figure it is evident that the proposed schemes are less time demanding.

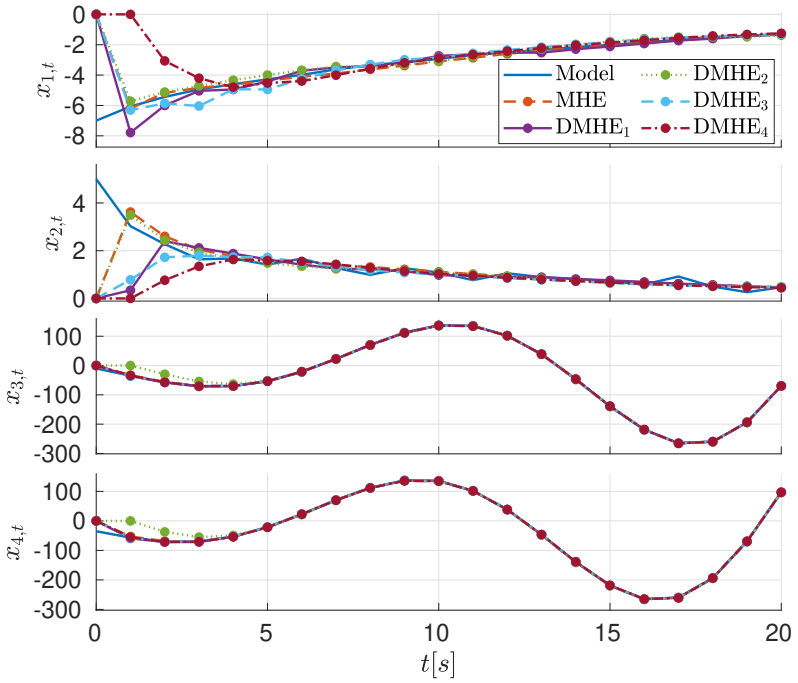


Figure 2.8: Components of the states $x_t = [x_{1,t}\ x_{2,t}\ x_{3,t}\ x_{4,t}]^\top$ and the estimates $\hat{x}_t = [\hat{x}_{1,t}\ \hat{x}_{2,t}\ \hat{x}_{3,t}\ \hat{x}_{4,t}]^\top$ computed by the MHE and DMHE algorithms.

	minτ_t	maxτ_t	$\sum \tau_t$	RMSE
MHE	0.27	1.13	17.25	1.4
MHE$_{pre}$	0.24	0.27	5.21	0.99
DMHE$_1$	0.26	1.10	17.30	0.74
DMHE$_{1-pre}$	0.24	0.29	5.20	0.84
DMHE$_2$	0.26	1.02	16.30	1.53
DMHE$_{2-pre}$	0.24	0.27	5.09	1.63
DMHE$_3$	0.27	1.08	16.49	0.77
DMHE$_{3-pre}$	0.25	0.29	5.18	0.76
DMHE$_4$	0.26	1.10	17.24	0.90
DMHE$_{4-pre}$	0.24	0.27	5.11	1.08

Table 2.2: Minimum, maximum and sum of the computation time τ_t (in seconds) and RMSE with $t_c = 5$ s of all algorithms collected in the simulation.

In fact, the computation time mainly depends on the number of optimization parameters n_{op}, the evaluation of the cost and constraints. As it can be observed from the evolution of the computation time, n_{op} is the most important, since the algorithms without pre-estimation have $n_{op} = 2 \cdot n \cdot N$, while the ones with pre-estimation have only $n_{op} = n \cdot N$. Thus, the pre-estimation divides by 2 the number of optimization parameters. Since $N = t$ for $t \leqslant 5$ and $N = 5$ for $t > 5$, as expected, the computation time rises more as N increases for MHE and DMHE (orange, green and cyan lines), i.e. the algorithms without pre-estimation.

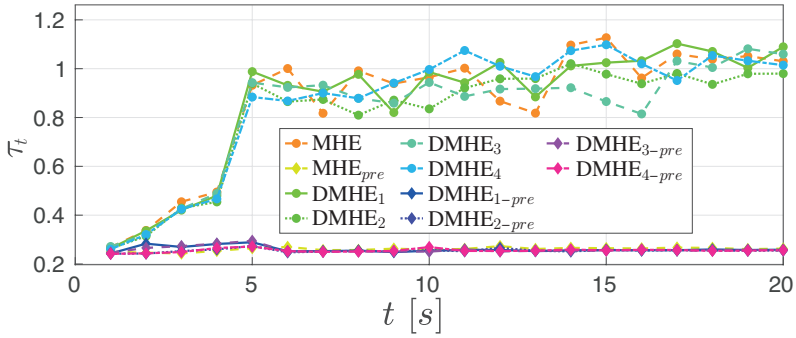


Figure 2.9: Comparison among the computation times of all algorithms run in the simulation: MHE, MHE_{pre}, DMHE, DMHE_{pre}.

Concluding this section, which presented two MHE algorithms capable to reduce the computation time compared with some algorithms in literature, the next section will extend the obtained results on improving also the estimation accuracy.

2.6 Extension with observability rank-based consensus weights

This section proposes an extension of the Distributed Moving Horizon Estimation with pre-estimation (DMHE_{pre}) by equipping it with an *observability rank-based weights* technique, hereafter denoted by DMHE_{w-pre}. This algorithm has been published in [Venturino et al., 2020].

The objective of this extension is to make the developed estimation approach fully suitable for a distributed scheme by exploiting only locally available information on computing the consensus weights terms (2.4). Moreover, relying on observability properties, this technique helps to enhance the accuracy of the state estimates.

In [Farina et al., 2010a], the authors proposed an algorithm to compute the K matrix, according to (2.4) and compatible with $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, to ensure the stability of their DMHE. This algorithm requires knowledge about the *global* network topology and, thus, the consensus weights have to be globally recomputed when the network topology changes. In order to overcome this issue, we further present a new observability rank-based weighted approach built on only local information available to each sensors $i \in \mathcal{N}$, thus representing an enhancement w.r.t. the algorithm in [Farina et al., 2010a]. The proposed technique relies on observability

properties associated to each sensor. With the intention to enhance the convergence of the consensus terms, the proposed method gives major importance to sensors that have better regional observability properties. A description of this approach is presented in the next subsection.

Observability rank-based weights technique

Consider a sensor i. Its regional observability matrix:

$$\bar{\mathcal{O}}_N^i = \begin{bmatrix} (\bar{C}^i)^\top & (\bar{C}^i A)^\top & \cdots & (\bar{C}^i A^{N-1})^\top \end{bmatrix}^\top \tag{2.35}$$

is of full rank if and only if the the pair $(A, \bar{C}^i)$ is completely observable, i.e. $\text{rank}(\bar{\mathcal{O}}_N^i) = n$. Otherwise, its rank is less than n, where n is the number of the states. For the sake of simplicity, we denote by $\rho_{\mathcal{O}}^i = \text{rank}(\bar{\mathcal{O}}_N^i)$. This information could be used as *reliability* of sensor i when choosing the weights, which according to (2.4) must be averaged among the neighbors, resulting in:

$$k_{ij} = \frac{\rho_{\mathcal{O}}^j}{\sum_{j \in \mathcal{N}^i} \rho_{\mathcal{O}}^j}, \quad \forall j \in \mathcal{N}^i. \tag{2.36}$$

It is worth noticing that each row of the matrix K concerns one sensor, e.g., the i-th row can be computed by sensor i using data coming from its neighbors $j \in \mathcal{N}^i$ solely. Hence, it can easily be recomputed online if the topology of the network changes.

2.7 Simulation results

This section illustrates the effectiveness of the proposed DMHE algorithm with pre-estimation and observability rank-based weights. In order to evaluate its performance, the proposed technique is compared to the centralized MHE of [Rao et al., 2001] as well as the DMHE algorithms of [Farina et al., 2010a] (without pre-estimation) and [Venturino et al., 2020] (with pre-estimation). To this end, we consider the same unstable system as in Section 2.5, recalled below:

$$x_{t+1} = \begin{bmatrix} 0.9962 & 0.1949 & 0 & 0 \\ -0.1949 & 0.3819 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1.21 & 1.98 \end{bmatrix} x_t + w_t, \tag{2.37}$$

where $x_t = \begin{bmatrix} x_{1,t} & x_{2,t} & x_{3,t} & x_{4,t} \end{bmatrix}^\top \in \mathcal{X} = \mathbb{R}^4$ is the state vector and $w_t \in \mathbb{R}^4$ is a zero-mean white noise with covariance matrix $Q = \text{diag}(0.0012,\ 0.038,\ 0.0012,\ 0.038)$.

The algorithms are initialized with $\mu_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top$ and $\Pi_0 = I_4$.

To compare the results of all algorithms, we use $n_S = 9$ sensors for the distributed algorithms. For the centralized scheme we consider the following measurement equation:

$$y_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_t + v_t,$$

with $\text{Var}(v_t) = R = I_2$. The considered distributed approaches are using the following measurement equations:

$$\begin{aligned} y_t^i &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x_t + v_t^i \quad \text{if } i \in \{1,2,6\} \\ y_t^i &= \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} x_t + v_t^i \quad \text{if } i \in \{3,4,9\} \\ y_t^i &= \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} x_t + v_t^i \quad \text{if } i \in \{5,7,8\} \end{aligned}$$

where $\text{Var}(v_t^i) = R^i = 1$, $i = 1, \ldots, 9$. The nodes are connected as reported by the graph in Figure 2.10 and the transition matrix K is computed with the observability rank-based weighted method (2.36). The different colors highlight the regional observability properties of each sensor i, thus of the pair $(A, \bar{C}^i)$. In particular, the green nodes mean that the pair $(A, \bar{C}^i)$ is completely observable, the yellow nodes are at least regional detectable, the orange nodes are not detectable (but having the capability to access to some components of the output, measured or received from neighbors) and the red node has no sensing information, i.e. $\rho_{\mathcal{O}}^i = 0$.

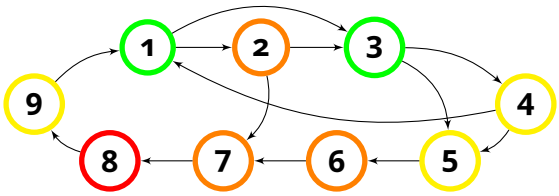


Figure 2.10: Sensor network.

To evaluate the performance of the proposed algorithm we take into account two performance metrics both averaged over the nodes in the network. The first metric is the Root Mean Square Error (RMSE) computed as follows:

$$\text{RMSE}_t = \frac{1}{n_S} \sum_{i=1}^{n_S} \left\| x_t - \hat{x}_{t|t}^i \right\|,$$

where x_t is the real state and $\hat{x}_{t|t}^i$ is the estimated state by sensor i. Moreover, $n_S = 9$ for distributed schemes and $n_S = 1$ for the centralized one. The second metric is the computation time τ_t needed by each algorithm to solve the local optimization problem at each time instant t.

We consider two simulations cases with time duration $t_f = 50s$.

Case 1. Setting the horizon length $N = 4$, one hundred Monte Carlo trials have been performed with each component of the initial state x_0 uniformly distributed in the interval $[-100, 100]$.

Figure 2.11 shows that the proposed algorithm DMHE$_{w-pre}$ with pre-estimation and observability rank-based weights (purple dotted line) converges and also shows better performance vs. the DMHE without pre-estimation [Farina et al., 2010a], denoted by DMHE (red line), and the DMHE only with pre-estimation (2.16) [Venturino et al., 2020], denoted by DMHE$_{pre}$ (orange dashed line), which are one

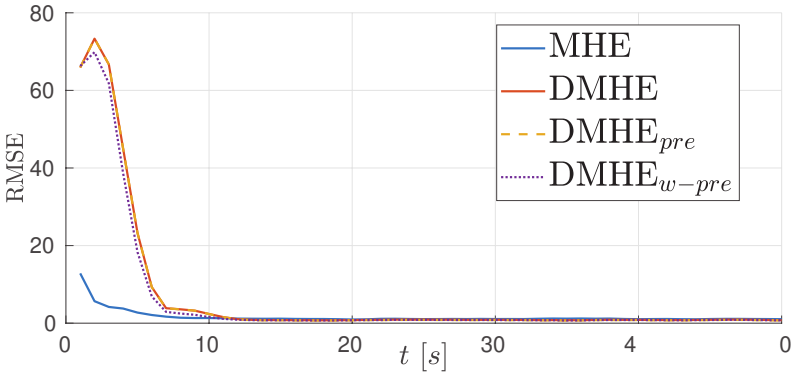
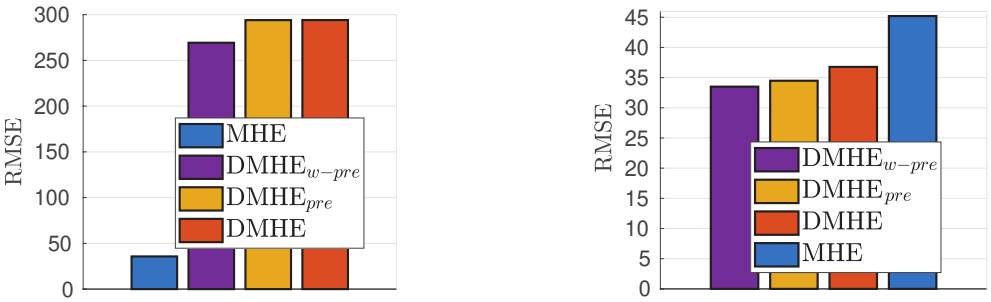


Figure 2.11: Time behavior of the trials-averaged RMSE for 100 Monte Carlo trials.

above the other because these last two have very similar RMSEs. Moreover, after the transient period all the distributed algorithms show similar RMSEs compared to the centralized one. This is more evident in Figure 2.12 which on one hand



(a) Transient period.

(b) steady state.

Figure 2.12: Sum of RMSEs in the transient period (for $t \in \{1, \ldots, 9\}s$) and in the steady state (for $t \in \{10, \ldots, 50\}s$).

shows the sum of RMSEs in the transient period and in the steady state. Indeed, in Figure 2.12a, as expected in the transient period, among the three distributed methods, the RMSE of DMHE_{w-pre} is the smallest among the DMHEs algorithms. On the other hand, as it can be seen in Figure 2.12b, all the distributed algorithms have comparable results in the steady state.

Regarding the second performance metric in Figure 2.13, it is clear that the algorithms with pre-estimation are less time demanding than the DMHE of [Farina et al., 2010a] and the centralized MHE. In fact, the computation time τ is reduced of *circa* 27% w.r.t. the DMHE of [Farina et al., 2010a] and 49% w.r.t. the centralized MHE of [Rao et al., 2001]. Notice that the computation time of DMHE in [Venturino et al., 2020] only with pre-estimation (2.16) and the extended version proposed in this section are the same. It is due to the fact that the difference of the extended version (DMHE_{w-pre}) concerns only the consensus weights k_{ij} computation, which can be computed off-line.

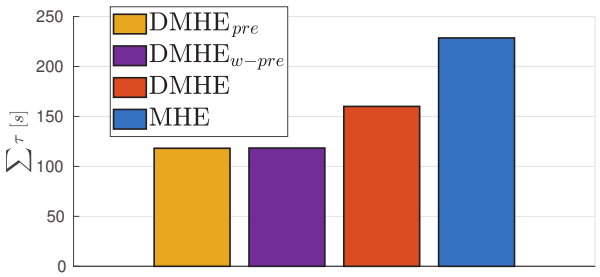


Figure 2.13: Sum of the trials-averaged computation times τ for 100 Monte Carlo trials.

Case 2. To better evaluate the proposed algorithm, a second simulation is performed with the same parameters as the first case except two crucial variables. First of all, a different horizon length is used, i.e. $N \in \{2, 3, \ldots, 10\}$. The matrix K used in this simulation results in a convex combination with the parameter $\varepsilon \in \{0, 0.1, \ldots, 1\}$ of the matrices $\tilde{K}$ and $\hat{K}$:

$$K = \varepsilon \tilde{K} + (1 - \varepsilon)\hat{K} \tag{2.38}$$

where $\tilde{K}$ is computed with the proposed observability rank-based weighted method, while $\hat{K}$ is computed as in [Farina et al., 2010a]. Moreover, a set of 100 different initial states $\{x_{0,z}\}_{z=0}^{100}$ is uniformly generated with components from $[-100, 100]$. Therefore, this simulation is composed by 9900 trials as result of the Cartesian product $N \times \varepsilon \times \{x_{0,z}\}$.

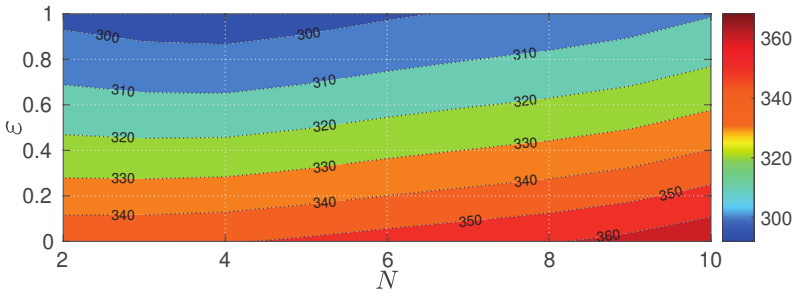


Figure 2.14: Sum of x_0-averaged RMSEs of the proposed DMHE$_{w-pre}$ with K varying as in (2.38).

Figure 2.14 shows the RMSEs of the DMHE$_{w-pre}$ with K as in (2.38). Notice that these curves should be seen as "discrete" not as continuous. Indeed, the RMSE is averaged among the $z = 1, \ldots, 100$ trials by changing $x_{0,z}$, computed as follows:

$$\text{RMSE}_{(N,\varepsilon)} = \frac{1}{100} \sum_{z=1}^{100} \sum_{t=1}^{t_f} \text{RMSE}_{t,x_{0,z}},$$

which results in a function of the horizon length N and the parameter ε. As it can be seen, along the N-axis the slope of the level curves of the RMSE changes

mildly, with a minimum for $N = \{3, 4\}$. Along the ε-axis it is evident that the minimum is at $\varepsilon = 1$, thus when $K = \tilde{K}$. This means that Figure 2.14 shows that the performance in terms of RMSE of the proposed observability rank-based weighted method is better than Algorithm 1 (i.e. with $\hat{K}$) in [Farina et al., 2010a].

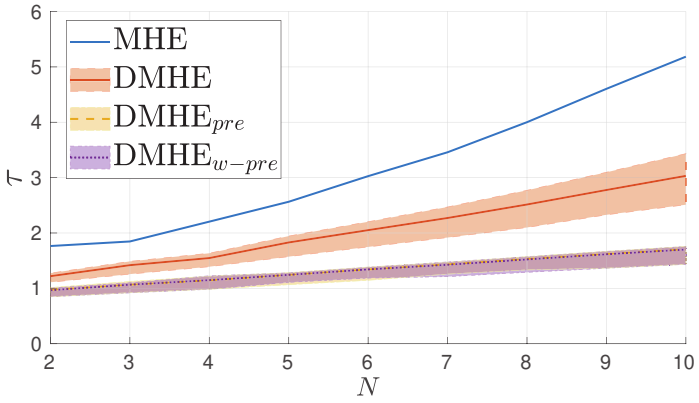


Figure 2.15: x_0-ε-averaged computation time τ.

Idem for this case simulation, the computation time is lower for the two DHMEs with pre-estimation, thus also for the proposed observability rank-based weighted method, as shown in Figure 2.15. This figure shows the computation time τ averaged over the trials in which ε and x_0 change, that results in τ as function of N. The slope of this function is the same for the two DMHEs with pre-estimation, and it is lower than the DMHE in [Farina et al., 2010a] and the centralized MHE [Rao et al., 2001]. Moreover, Figure 2.15 also shows the limits representing the minimum and maximum computation time of the DMHE algorithms. It can be noticed that these limits are narrower and less deviating w.r.t. the horizon length N for the algorithms with pre-estimation. This clearly shows the very promising performance of the proposed algorithm for implementation on networks of sensors with low computation capabilities.

2.8 Conclusion

This chapter presented a novel algorithm based on the Moving Horizon Estimation (MHE) concept for distributed state estimation of discrete-time linear time-invariant systems. The use of a pre-estimation observer results in a significant reduction of the computation time. The proposed strategy DMHE$_{pre}$ has been validated via illustrative numerical examples. Due to the consensus terms embodied in the optimization problem, the estimation errors produced by the DMHE$_{pre}$ are capable to converge even if some sensors in the network are not observable. Moreover, the accuracy of the estimation errors is preserved (together with a reduced computation time), in the sense that it is comparable with the one of the original formulation [Farina et al., 2010a].

Moreover, the extended algorithm with observability rank-based weights technique (DMHE$_{w-pre}$) showed that the accuracy of the estimation errors is improved both in the transient period and in the steady state w.r.t. the one of the original formulation [Farina et al., 2010a], as result of choosing the consensus weight matrix K with the presented rank-based weighted method. Furthermore, this method allows each sensor to determine its consensus weights on the basis of only local provided information contrary to Algorithm 1 in [Farina et al., 2010a] that, instead, needs knowledge about the global network topology. Thus the proposed DMHE$_{w-pre}$ is suitable for a fully distributed scheme and could be extended for time-varying topology.

Simulation results have shown the effectiveness of the proposed DMHE algorithms even in presence of weak regional observability conditions induced by some sensors of the considered network.

The proposed algorithms have been presented and published in:

- **A. Venturino**, S. Bertrand, C. Stoica Maniu, T. Alamo, and E. F. Camacho. Distributed moving horizon estimation with pre-estimating observer. In *24th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 174–179, Sinaia, Romania, 8-10 October, 2020. **Best Paper Award**

- **A. Venturino**, C. Stoica Maniu, S. Bertrand, T. Alamo, and E. F. Camacho. Distributed moving horizon state estimation for sensor networks with low computation capabilities. *System Theory, Control and Computing Journal*, 1(1):81–87, 2021.

3 - ℓ-step Neighborhood Distributed Moving Horizon Estimation

This chapter focuses on Distributed State Estimation over a peer-to-peer sensor network composed by possible low-computational sensors. We propose a new *ℓ-step Neighborhood Distributed Moving Horizon Estimation* technique with fused arrival cost and pre-estimation, improving the accuracy of the estimation, while reducing the computation time compared to other approaches from the literature. Simultaneously, convergence of the estimation error is improved by means of spreading the information among neighborhoods, which comes natural in the sliding window data present in the Moving Horizon Estimation paradigm. Illustrative numerical simulations are provided to analyze the performance of the proposed approach, with respect to existing algorithms, considering as metrics the accuracy of the estimates and the computation time. The proposed algorithm has been presented at the 60th IEEE Conference on Decision and Control.

The chapter is structured as follows. Section 3.1 briefly describes existing results on Distributed Moving Horizon Estimation. Section 3.2 introduces the problem formulation, explains the communication protocol, and details the exchanging information from neighborhood to neighborhood. The proposed ℓ-step Neighborhood DMHE algorithm is presented in Section 3.3. Before concluding remarks, simulations examples are presented and analyzed in Section 3.4.

3.1 Introduction

In [Farina et al., 2010a], the authors have proposed a DMHE algorithm for constrained linear systems proving that it is stable even under weak observability conditions (due to consensus on estimates and a consensus weight term in the DMHE formulation). The computation of the consensus weight matrix by each sensor involves a Kalman-like covariance update formula and a stochastic matrix. The authors also provided an algorithm to weight the components of this matrix. However, the main drawback of this algorithm is that it requires complete knowledge of the communication network topology, and thus, it is unsuitable for distributed schemes, where computations performed by each sensor should only rely on locally available data. More recently, [Battistelli, 2018] introduced another consensus-based mechanism in a DMHE approach to fuse local arrival costs and guarantee stability of the estimation errors in a fully distributed way, i.e. each sensor being capable of guaranteeing convergence of the estimate to the system state using only information locally available. In Chapter 2 were presented two algorithms based on [Farina et al., 2010a] which, thanks to the introduction of pre-estimating observer and the observability rank-based weights technique, they

were able to reduce the computation time to solve the optimization problem and to enhance the estimation accuracy.

This chapter extends the approach proposed in Chapter 2 [Venturino et al., 2020] to the DMHE formulation of [Battistelli, 2018] which has proven to obtain more general stability results as well as enhanced performance compared to [Farina et al., 2010a]. The current chapter also leads to a reduced computation time due to a pre-estimating observer. Another contribution concerns the improvement of the convergence of the estimation error by mitigating unobservability issues. This situation could arise in sensor networks when some nodes may have no sensing capacities (inactive sensors), or are able to only measure some parts of the state of the system that would make it non observable using only these sensors. For this purpose, the new proposed DMHE technique exploits the exchanges of information among local nodes based on an ℓ-step neighborhood information spreading mechanism, which comes natural in the sliding window data present in the MHE paradigm.

In the provided numerical simulations, it is shown the practical efficacy of the proposed ℓ-step neighborhood Distributed Moving Horizon Estimation technique, when compared with the DMHE in [Farina et al., 2010a, Battistelli, 2018], in which no ℓ-step neighborhood mechanism is incorporated. Moreover, thanks to the pre-estimation concept included in its formulation, the ℓ-step neighborhood DMHE is also able to decrease the computation time by a significant factor and also to enhance the convergence of the estimation errors.

The next section gives some definitions, details the communication protocol used in the sensor network and how the sensors could exploit the shared information within the MHE paradigm. In the end, it states the problem to be solved in this chapter.

3.2 Problem statement

This section describes the state estimation problem using a neighborhood diffusion mechanism over a sensor network. The dynamical system under investigation, the measurement equation and the sensor have already been described in Section 2.2, and they are still valid in this chapter. However it is necessary to add more notations and definitions.

Regarding the sensor network, in addition to the shared data coming from the neighbor sensors in $\mathcal{N}^i$, each sensor i could exploit past information from other sensors $j \notin \mathcal{N}^i$, if there exists a path from sensors j to sensor i according to the directions of the edges. For this reason, denoting by $d(i,j)$ the distance, in terms of number of edges, between nodes i and j, we define the *ℓ-step neighborhood set* as follows.

Definition 3.1 — ℓ-step neighborhood set

*The set $\mathcal{N}_\ell^i = \{j \neq i \in \mathcal{N} : d(i,j) \leqslant \ell\}$ is named ℓ-**step neighborhood set** of sensor i, which is the set of sensors $j \in \mathcal{N}$ at most distant ℓ edges from sensor i.*

Notice that $\mathcal{N}_1^i = \mathcal{N}^i \subseteq \mathcal{N}_\ell^i$ and $\mathcal{N}_0^i = \emptyset$. For example, Figure 3.1 shows three ℓ-step neighborhood sets for sensor 4 when considering $\ell = 1$, $\ell = 2$ and $\ell = 3$, respectively, leading to the sets $\mathcal{N}_1^4 = \{3\}$ (blue), $\mathcal{N}_2^4 = \{2, 3\}$ (red) and $\mathcal{N}_3^4 = \{1, 2, 3\}$ (green).

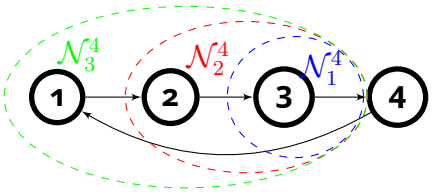


Figure 3.1: Three ℓ-step neighborhood sets.

Remark 7. *The DMHE approach is prone to use past information due to the fact that it aims at minimizing a distance from predicted and measured outputs over a sliding window of fixed size $N + 1$, i.e. $[t - N, \ldots, t]$, see [Muske et al., 1993, Rao et al., 2001], as it will be seen in Section 3.3.*

The next subsection describes how the MHE paradigm can exploit this sequence of past measurements.

3.2.1 Communication protocol

Assumption 3.1 — Heterogeneous sensor network

The network is composed of possibly different types of sensor, like cameras, LIDARs, etc.

According to Assumption 3.1 some sensors could be with no sensing capabilities, i.e. $C^i = 0$, or at least partially, meaning that a sensor may observe only some part of the state vector of the system, i.e. the pair (A, C^i) is possibly not detectable. Moreover, the network could be deployed such that some neighborhoods are composed only of nodes resulting in weak local or regional observability properties [Farina et al., 2010a], meaning that the pair $(A, \bar{C}^i)$ could be not detectable, where $\bar{C}^i$ is the regional output matrix, i.e. $\bar{C}^i = [(C^i)^\top, (C^{j_1})^\top, \ldots, (C^{j_{n_S^i}})^\top]^\top$, with $\{j_1, \ldots, j_{n_S^i}\} \in \mathcal{N}^i$ and n_S^i the number of sensors in the neighborhood set of sensor i.

Therefore, with the aim to enhance collective observability [Farina et al., 2010a] by the network, it is proposed that each node $i \in \mathcal{N}$ exploits measurements received

from its ℓ-*step neighborhood* $\mathcal{N}_\ell^i$. This section details how information coming from $\mathcal{N}_\ell^i$ will be considered in the formulation of the DMHE, by choosing $\ell = N$, where N is the length of the horizon of past information considered for state estimation by the algorithm. An example on a simple sensor network is also provided to simplify the understanding.

Definition 3.2 **Inactive sensor**

A sensor with no sensing capabilities is called ***inactive sensor****, therefore sensor i is inactive if its output matrix is $C^i = 0$.*

Assumption 3.2 **Single-hop routing protocol**

The communication network uses a single-hop routing protocol, i.e. the data are exchanged among sensors with one single hop (no intermediary communication devices involved).

Assumption 3.3 **No time delay nor packet losses**

In the communication network, it is assumed that there is no delay nor packet losses.

Assumption 3.4 **Time synchronization**

All sensors in the network are time synchronized, i.e. each sensor is able to exchange data with their neighbors at each time instant.

Assumption 3.5 **Time-sliding batch**

Each node $i \in \mathcal{N}$ keeps the information received from each of its in-neighbors nodes *in a time-sliding batch of size N and relays this information to* out-neighbors nodes *at the next time instant.*

Old information (i.e. received from time instant $t_r < t - N$) is removed from the batch. Therefore, at time t each node disposes of past measurements from nodes in its ℓ-step neighborhood over the time window $[t-N, t]$. Since dealing with a single-hop routing protocol, each sensor $i \in \mathcal{N}$ receives information only from its neighbors $j \in \mathcal{N}^i$ at each time t. Let $\bar{y}^i$ denote the measurements collected by sensor i from all the nodes $j \in \mathcal{N}^i$. Then, at the time instant t, the collected measurements from nodes j are

$$\bar{y}^i_{[t-N,\dots,t]} = \begin{bmatrix} y^{j_1}_{t-N} & \cdots & y^{j_1}_{t} \\ \vdots & \ddots & \vdots \\ y^{j_{n^i_S}}_{t-N} & \cdots & y^{j_{n^i_S}}_{t} \end{bmatrix}, \quad \text{with } j_1, \dots, j_{n^i_S} \in \mathcal{N}^i.$$

At the same time, each sensor $j \in \mathcal{N}^i$ has data collected from its own neighbors $z \in \mathcal{N}^j$, with $z \neq i$, from the previous time step $t-1$, i.e.

$$\bar{y}^j_{[t-N,\dots,t-1]} = \begin{bmatrix} y^{z_1}_{t-N} & \cdots & y^{z_1}_{t-1} \\ \vdots & \ddots & \vdots \\ y^{z_{n^j_S}}_{t-N} & \cdots & y^{z_{n^j_S}}_{t-1} \end{bmatrix}, \quad \text{with } z_1, \dots, z_{n^j_S} \in \mathcal{N}^j.$$

that they can share with sensor i. This philosophy can be reiterated back in time, and so along the communication links in $\mathcal{N}^i_{t-k}$, $\forall k = t-N, \dots, t-1$, with a maximum of $N = \ell$ back steps. It is worth to remind that the horizon length N is reduced when the inequality $t \leqslant N$ is satisfied, i.e. $N = t$, thus the just described mechanism of collecting information works for any $t > 0$.

To summarize, the node i has the collection of data:

$$\left\{ \bar{y}^i_{[t-N,\dots,t]}, \bar{y}^j_{[t-N,\dots,t-1]}, \dots, \bar{y}^z_{t-N} \right\}, \text{with } j, \dots, z \in \mathcal{N}^i_\ell,$$

which is useful in the local MHE optimization problem to improve the accuracy of the estimates. Indeed, the node i disposes of the information available within a "distance" of ℓ nodes within the sensor network.

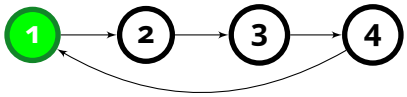


Figure 3.2: 4-node digraph sensor network.

In the following, an example clarifies how this information diffusion mechanism works.

Example. Consider a directed ring network composed of 4 nodes and connected as in Figure 3.2. Assume that the considered communication protocol is used with $N = 3$. Only node 1 (in green) is equipped with sensing capabilities, while the white nodes are *inactive* sensors. Hence, in this example, there is a single node that provides sensing information. Starting from $t_0 = 0$, Table 3.1 lists the measurements data collected from each sensor at each time instant t. Notice that the nodes get rid of obsolete data even if the window data is not full. For example, sensor 4 at time t_3 has collected data into a two-dimensional vector $y^1_{[t_0,t_1]}$, and, at the next step t_4, it gets rid of $y^1_{t_0}$ because this is outdated, in other words it is no longer useful because it is out of the window $[t-N, t] = [t_1, t_4]$.

It is worth noticing that nodes 3 and 4 will not have any measurement data if the information comes only from their (one-step) neighborhood $\mathcal{N}^i$ (i.e. $\ell = 1$). □

The next subsection formulates the problem addressed in this chapter.

Table 3.1: Collected data over time by the considered nodes for N=3.

t	1	2	3	4
$t_0 = 0$	$y^1_{t_0}$	$y^1_{t_0}$	-	-
$t_1 = 1$	$y^1_{[t_0,t_1]}$	$y^1_{[t_0,t_1]}$	$y^1_{t_0}$	-
$t_2 = 2$	$y^1_{[t_0,t_1,t_2]}$	$y^1_{[t_0,t_1,t_2]}$	$y^1_{[t_0,t_1]}$	$y^1_{t_0}$
$t_3 = 3$	$y^1_{[t_0,t_1,t_2,t_3]}$	$y^1_{[t_0,t_1,t_2,t_3]}$	$y^1_{[t_0,t_1,t_2]}$	$y^1_{[t_0,t_1]}$
$t_4 = 4$	$y^1_{[t_1,t_2,t_3,t_4]}$	$y^1_{[t_1,t_2,t_3,t_4]}$	$y^1_{[t_1,t_2,t_3]}$	$y^1_{[t_1,t_2]}$

3.2.2 Problem formulation

Definition 3.3 **Poorly-Observing Sensor Network**

*A **Poorly-Observing Sensor Network** is defined as a network containing at least one node $i \in \mathcal{N}$ having any of the following characteristics*

- *It has no sensing capabilities, i.e. $C^i = 0$;*
- *It has sensing capabilities and can provide a measurement on the state of the system, i.e. $C^i \neq 0$, but the pair (A, C^i) still remains non detectable;*
- *Nodes in its neighborhood are such that the pair $(A, \bar{C}^i)$ is non detectable.*

The problem addressed in this chapter, namely *Distributed State Estimation over a Poorly-Observing Sensor Network* can be stated as follows.

Problem 3.1 **Distributed State Estimation over a Poorly-Observing Sensor Network**

Given the discrete-time LTI system (2.1), the sensor network $\mathcal{G}$ with linear sensors as in (2.3), under the assumptions that:

- *The pair (A, C) is observable, where $C = col(C^i)$ with $i \in \mathcal{N}$ is the collective output matrix, i.e. $C = [(C^1)^\top, \dots, (C^{n_S})^\top]^\top$;*
- *The graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is strongly connected (see Definition 2.5).*

The role of each sensor $i \in \mathcal{N}$, at each time t, is to (possibly) get measurement on the system, to exchange information among neighbor nodes $\mathcal{N}^i$ and to process locally available information in order to determine a local estimate $\hat{x}^i_{t|t}$ of the real state of the system x_t.

The next section describes the distributed algorithm that solves the Problem 3.1.

3.3 Proposed DMHE technique

This section presents the proposed Distributed Moving Horizon Estimation approach. It extends the one of [Battistelli, 2018] based on consensus on the arrival costs, by accounting for information from ℓ-step neighborhoods and taking advantage of a pre-estimating observer as in [Venturino et al., 2020] to reduce computation time.

3.3.1 Local optimization problem

At time t, let $\hat{x}^i_{t-N|t}, \dots, \hat{x}^i_{t|t}$ be the sequence of estimates of the state of system (2.1) to be computed by each sensor $i \in \mathcal{N}$ over a given past horizon of length $N \geqslant 1$. The estimate of the state x_t to be provided by each sensor at time t corresponds to $\hat{x}^i_{t|t}$. To do so, a local minimization problem can be formulated for each sensor i as follows

$$\hat{x}^i_{t-N|t} = \arg\min_{\hat{x}^i_{t-N}} \quad J^i_t \tag{3.1}$$

$$\text{s.t.} \quad \hat{x}^i_{k+1} = A\hat{x}^i_k + L^i\hat{v}^i_k + \sum_{j\in\mathcal{N}^i_{t-k}} L^j\hat{v}^j_k, \qquad \forall k = t-N,\dots,t-1, \tag{3.2}$$

$$\hat{v}^j_k = y^j_k - C^j\hat{x}^i_k, \quad j \in \{i\}\cup\mathcal{N}^i_{t-k}, \qquad \forall k = t-N,\dots,t, \tag{3.3}$$

$$\hat{x}^i_k \in \mathcal{X}, \qquad \forall k = t-N,\dots,t, \tag{3.4}$$

$$\hat{v}^j_k \in \mathcal{V}^j, \quad j \in \{i\}\cup\mathcal{N}^i_{t-k}, \qquad \forall k = t-N,\dots,t. \tag{3.5}$$

where the set $\mathcal{X}$ is used to constrain the system state in its *a priori* knowledge and the sets $\mathcal{V}^i$ are the bounds for the unknown difference between the measured and predicted output of each sensor i. The sequence of state estimates $\hat{x}^i_{t-N+1|t}, \dots, \hat{x}^i_{t|t}$ is then computed from the optimal solution $\hat{x}^i_{t-N|t}$ and using (3.2).

The main difference in this formulation is that a Luenberger observer formulation is used in (3.2) instead of the state equation of the system, as classically used in MHE formulations and in [Battistelli, 2018], which requires to consider the disturbance sequence $\{w\}^{t-1}_{k=t-N}$ over the past horizon as additional optimization parameters. Moreover, on not taking into account the disturbance sequence leads on having different equations when calculating the consensus terms and weights, as it is described later on. As it is shown in Chapter 2, removing this sequence

reduces the computation cost, while simultaneously preserving the accuracy of the state estimate. Under the assumption that the gain L^i is computed such that:

$$\Phi^i = A - L^i C^i, \ \ \forall i \in \mathcal{N} \tag{3.6}$$

is Schur stable, then, in order to mitigate the effects on the estimation errors at certain frequencies or to increase robustness for each frequency, the gain L^i can be computed off-line according to some criteria, for example $\mathcal{H}_2$, $\mathcal{H}_\infty$ [Duan and Yu, 2013, p. 293]. Note that the assumption of (3.6) being Schur can be satisfied only when the pair (A, C^i) is observable, otherwise, as *extrema ratio*, it is sufficient to design L^i in order to keep the spectrum radius of Φ^i as low as possible.

Another difference w.r.t. [Battistelli, 2018] is that the optimization problem (3.1) uses the set $\mathcal{N}_\ell^i$ instead of $\mathcal{N}^i$, leading to improve the estimation accuracy. In fact, $\mathcal{N}_\ell^i$ appears in (3.2), (3.3) and (3.5) but also in the objective function J_t^i defined as:

$$J_t^i = \Gamma_t^i(\hat{x}_{t-N|t}^i) + \sum_{k=t-N}^{t} \left\| y_k^i - C^i \hat{x}_k^i \right\|_{R^i}^2 + \sum_{k=t-N}^{t} \sum_{j \in \mathcal{N}_{t-k}^i} \left\| y_k^j - C^j \hat{x}_k^i \right\|_{R^j}^2 \tag{3.7}$$

where the weight matrices R^i (resp. R^j) are positive definite matrices which define the reliability of sensor i on measuring y^i (resp. y^j). Thus, it comes natural to chose R^i (resp. R^j) as the inverse of the covariance matrix of the measurement noise. The first term is the so called *initial penalty* function $\Gamma_t^i(\cdot)$, known in the MHE environment as *arrival cost*. It is assumed to be non negative and it summarizes the effect of the past measurements, before time $t - N$. Further details on the arrival cost are provided in Section 3.3.2, because it plays a major role in the convergence and the performance of the algorithm. The second term is the weighted difference between the measured output, from the sensor i itself, and the predicted one. The third term is similar with the second one but the measurements come from the ℓ-step Neighborhood set of sensors.

Note that, when the current time instant t satisfies the inequality $t \leqslant N$ then the horizon length N is set to $N = t$. In this way it is possible to use the algorithm even if there are not enough measurements to fill in the sequence of past measurements within the horizon window.

Remark 8. *The local optimization problem can be formulated using information coming only from direct neighbor sensors, in other words $\ell = 1$. For later comparisons, we denote by:*

- *DMHE$_{pre}^{\ell}$ the minimization problem* (3.1) *having $\mathcal{N}_\ell^i$, with $\ell = N$;*
- *DMHE$_{pre}^{1}$ the one with $\mathcal{N}_\ell^i$ equal to $\mathcal{N}^i$, i.e. $\ell = 1$.*

Indeed, DMHE$_{pre}^{1}$ can be seen as a combination of the methods of [Battistelli, 2018] and [Venturino et al., 2020].

The next subsection describes how the arrival costs are locally fused.

3.3.2 Fused arrival cost

The objective function (3.7) contains the arrival cost term $\Gamma_t^i(\cdot)$. In the MHE approach, it usually penalizes deviations from some *a priori* information $\tilde{x}_{t-N}^i$ on the state at the beginning of the horizon as detailed in [Muske et al., 1993] and can be formulated as in [Battistelli, 2018] by:

$$\Gamma_t^i(x) = \left\| x - \tilde{x}_{t-N}^i \right\|_{P_{t-N}^i}^2, \tag{3.8}$$

with P_{t-N}^i a positive and symmetric definite weight matrix. The *a priori* state $\tilde{x}_{t-N}^i$ can be computed as a one step prediction from the solution of the optimization problem at the previous instant. In classical (D)MHE approaches, this prediction is done using the state equation. In the proposed algorithm with pre-estimation formulation, this prediction is computed as:

$$\tilde{x}_{t-N}^i = (A - L^i C^i)\hat{x}_{t-N-1|t-1}^i + L^i y_{t-N-1}^i. \tag{3.9}$$

The matrix P_{t-N}^i is defined as the inverse of the covariance matrix of the prediction $\tilde{x}_{t-N}^i$ and it can be computed recursively, as we explained later on, initialized as P_0^i in order to quantify the confidence on the initial a priori information $\tilde{x}_0^i$.

In a distributed setting, this cost has an essential role to propagate information among the sensors in the network in order to ensure convergence of the state estimations to the real state of the system, since the local observability for one node or observability among the neighborhood depends on the network topology and sensing capabilities. As in [Battistelli, 2018], the idea is then to fuse the arrival costs of the neighborhood $\mathcal{N}^i$ in a convex combination:

$$\Gamma_t^i(x) = \pi^{i,i} \left\| x - \tilde{x}_{t-N}^i \right\|_{P_{t-N}^i}^2 + \sum_{j \in \mathcal{N}^i} \pi^{i,j} \left\| x - \tilde{x}_{t-N}^j \right\|_{P_{t-N}^j}^2, \tag{3.10}$$

where all the weights $\pi^{i,i}$ and $\pi^{i,j}$ are strictly positive and fulfill the condition:

$$\pi^{i,i} + \sum_{j \in \mathcal{N}^i} \pi^{i,j} = 1, \quad \forall i \in \mathcal{N}. \tag{3.11}$$

Consequently, the initial penalty function is defined as a *consensus on the arrival costs* by means of the expression (3.10) ensuring that the local arrival cost is a weighted average of the local arrival costs from neighbors.

In the remaining part of this section, it is explained how to compute recursively the consensus weight matrix P_{t-N}^i and how to implement the function $\Gamma_t^i(x)$ in (3.10) by reporting and simplifying the *Proposition 1* in [Battistelli, 2018], considering only one consensus step. Indeed, in [Battistelli, 2018] the algorithm is developed to have multiple consensus steps within the same sampling period, i.e. each sensors exchange information among their neighbors to have a faster convergence. The number of consensus steps is denoted by L. Here, the proposed

algorithm does only one consensus step, then $L = 1$. Consider the following convex combination of the weights matrices among the neighborhood:

$$\Xi_t^i = \pi^{i,i} P_{t-N}^i + \sum_{j \in \mathcal{N}^i} \pi^{i,j} P_{t-N}^j \tag{3.12}$$

and the convex combination together with the predictions:

$$\xi_t^i = \pi^{i,i} P_{t-N}^i \tilde{x}_{t-N}^i + \sum_{j \in \mathcal{N}^i} \pi^{i,j} P_{t-N}^j \tilde{x}_{t-N}^j. \tag{3.13}$$

Then the fused arrival cost (3.10) coincides with the following:

$$\Gamma_t^i(x) = \left\| x - \left(\Xi_t^i\right)^{-1} \xi_t^i \right\|_{\Xi_t^i}^2. \tag{3.14}$$

Indeed, by making explicit the weighted norms in (3.10) and by substituting (3.12) and (3.13) in (3.14), the resulting equations will be equivalent. Notice that now the arrival cost is no longer formulated as an average of functions, hence its implementation is less complex than (3.10). Moreover, minimizing the objective function J_t^i with the arrival cost (3.10) or (3.14) leads to obtain the same estimates $\hat{x}_{t-N|t}^i, \ldots, \hat{x}_{t|t}^i$ (see [Battistelli, 2018]).

In the following, the covariance matrix P_{t-N}^i is recursively updated using only local information available at time t to sensor i. Consider the observability matrix F^i associated to the pair $(A - L^i C^i, C^i)$ along the horizon length N and its relative collective output weight matrix:

$$F^i = \begin{bmatrix} C^i \\ C^i(A - L^i C^i) \\ \vdots \\ C^i(A - L^i C^i)^N \end{bmatrix}, \; \Psi^i = \text{diag}(\underbrace{R^i, \ldots, R^i}_{N+1 \text{ times}}).$$

Then a preliminary consensus weight matrix Ω_{t-N-1}^i can be computed by using only data locally available at node i from the previous time instant:

$$\Omega_{t-N-1}^i = \pi^{i,i} P_{t-N-1}^i + \sum_{j \in \mathcal{N}^i} \pi^{i,j} P_{t-N-1}^j + \left(F^i\right)^\top \Psi^i F^i. \tag{3.15}$$

Following [Battistelli, 2018], we now introduce a scalar α such that $0 < \alpha < 1$ and a positive definite matrix S for any t, and adapt the equations to the proposed algorithm with pre-estimation. Then the updated consensus covariance matrix is defined by

$$P_{t-N}^i = \frac{\alpha}{8} \left[A_L^i \left(\Omega_{t-N-1}^i\right)^{-1} \left(A_L^i\right)^\top + S^{-1} \right]^{-1}, \tag{3.16}$$

where $A_L^i = A - L^i C^i$. The interested reader can refer to [Battistelli, 2018] for the complete proof of (3.16).

3.3.3 DMHE algorithm

Finally, in this subsection the *modus operandi* of the proposed distributed algorithm is described. First of all, it is worth to mention that the steps of the algorithm could be run in a parallel scheme by each sensor $i \in \mathcal{N}$, after they have sent and received the information from the neighbors at each time t, with the assumptions on the network and communication protocol provided in Section 3.2.1. Furthermore, as discussed in Section 3.3.1, for the first N steps, i.e. when $1 \leqslant t \leqslant N$, the horizon length N is reduced to $N = t$.

The steps of the DMHE$^{\ell}_{pre}$ procedure are described in Algorithm 2. To get the one-step DMHE$^{1}_{pre}$ procedure, it is sufficient to remove the step 10 and to use $\mathcal{N}^i$ instead of $\mathcal{N}^i_\ell$.

Algorithm 2 DMHE$^{\ell}_{pre}$ procedure

1: **Off-line:** $\forall i \in \mathcal{N}$
2: **compute** the Luenberger gain L^i
3: **store** the *a priori* initial estimation $\hat{x}^i_{0|0} = \hat{x}_0$ of x_0 and the covariance matrix $P^i_0 = P_0$ of x_0
4: **receive** from the neighbors $j \in \mathcal{N}^i$: L^j, C^j, R^j, $\mathcal{V}^j$
5: **Initialization:** $\forall i \in \mathcal{N}$, at the first time step $t = 0$
6: **collect** a first local measurement y^i_0
7: **receive** from the neighbors $j \in \mathcal{N}^i$ their measurements y^j_0
8: **Online:** $\forall i \in \mathcal{N}$, $\forall t > 0$
9: **collect** the local measurement y^i_t
10: **gather** past information received at time $t-1$ from $j \in \mathcal{N}^i$, as in Section 3.2.1
11: **compute** the prediction $\tilde{x}^i_{t-N}$ and the consensus weight matrix P^i_{t-N} according to (3.9) and (3.16), resp.
12: **receive** from the neighbors $j \in \mathcal{N}^i$ the collected, gathered and computed data in the steps 9, 10 and 11
13: **compute** the fused arrival cost Γ^i_t according to (3.10)
14: **solve** the local MHE, minimizing J^i_t as in (3.7) subject to the constraints (3.2)-(3.5)
15: **store** the solution $\hat{x}^i_{t-N|t}$ and the corresponding estimate $\hat{x}^i_{t|t}$

Remark 9. *All nodes are synchronized at the step 12, since each sensor i needs the data from its neighbors $j \in \mathcal{N}^i$. This is the only communication step, in fact the rest of the online algorithm can be run in parallel by each node i, since only local provided data is used.*

Remark 10. *The path length ℓ of the ℓ-step neighborhood $\mathcal{N}^i_\ell$ can also be chosen lower than the horizon length N of the DMHE, i.e. $1 \leqslant \ell \leqslant N$. One can design then the DMHE$^{\ell}_{pre}$ in order to have a good trade-off between the accuracy of the estimation and the amount of data exchanged in the network. This trade-off depends on the observability conditions and the network topology.*

The proposed ℓ-step neighborhood DMHE is validated in simulations in the next section.

3.4 Simulations

This section provides an evaluation of the proposed DMHE algorithm via simulations examples. They have been performed by a setup implemented in MATLAB R2019b environment and the solver *quadprog* in Yalmip [Löfberg, 2004] on a Linux Ubuntu 20.04 PC equipped with an Intel Core i7-10875H, 2.30 GHz. To compare it with existing results in literature, the scenario in [Battistelli, 2018] is considered.

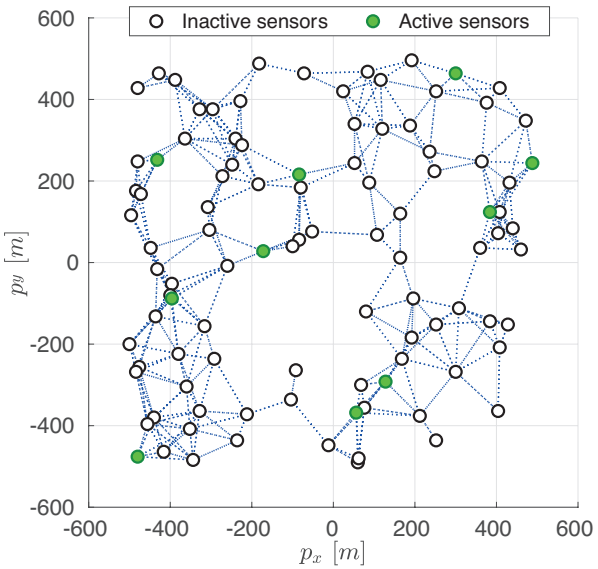


Figure 3.3: Topology of the sensor network composed by 100 nodes [Battistelli, 2018].

The goal is to track a 2D moving target using a sensor network, that could model, for example, a distributed camera network. As illustrated in Figure 3.3, the network is composed of 100 sensors randomly disposed with a uniform distribution on a plane of $[-500, 500] \times [-500, 500]$ m, in which only the 10 green nodes are active sensors (i.e. with sensing capabilities), while the 90 white nodes are inactive sensors (i.e. null output matrix). In the following it is considered that a communication link between two nodes exists if the distance between them is less than a given communication radius equal to 160 m.

The 2D moving target is modeled as a double integrator system. The state of the system is represented by $x = [p_x \ p_y \ v_x \ v_y]^\top$ which corresponds to the Cartesian coordinates of its position and velocity vectors. The dynamics of the target are described by model (2.1) with $A = \begin{bmatrix} I_2 & T_s I_2 \\ 0 & I_2 \end{bmatrix}$, where $T_s = 1$ s is the sampling time used for discretization of the continuous-time dynamics of the target. The input disturbance w_t in (2.1) is a four dimensional vector and is assumed to be modeled by a noise vector with uniform distribution in $\mathcal{W} = [-0.5, 0.5] \times [-0.5, 0.5] \times [-0.5, 0.5] \times [-0.5, 0.5]$. The 10 active sensors provide measurements of the target's position in conformity with the matrix $C^i = [I_2 \quad O_{2,2}]$, while the

remaining 90 inactive sensors have no ability to measure, i.e. their output matrix is $C^i = O_{2,4}$. The measurement noise v_t^i of each sensor i is a two dimensional vector with a uniform distribution in $\mathcal{V}^i = [-10, 10] \times [-10, 10]$.

Two simulation cases are further analyzed.

Case 1. A first simulation considers a horizon length $N = 4$; the *a priori* information about the state $\hat{x}_0^i$ is set equal to $[0\ 0\ 0\ 0]^\top$ for each node; the initial arrival cost weight matrix is $P_0^i = \text{diag}(10^{-5}, 10^{-5}, 1, 1)$, taking into account different magnitudes of the states, i.e. first two positions and last two velocities components; the matrices Q and R^i are set for each sensor i as the inverse of the covariance matrices of w_t and v_t^i, respectively. The weight Q is used in [Farina et al., 2010a] and [Battistelli, 2018] for (D)MHE algorithms without pre-estimation. This weight matrix penalizes the norm on the sequence of disturbance input terms in the cost function of these algorithms. All these parameters are identically set in all the considered algorithms. The consensus weights $\pi^{i,j}$ are chosen to be equal among the neighborhood, i.e. the *Metropolis weights* as in (2.33), satisfying (3.11).

In order to compare the proposed DMHE$^1_{pre}$ (namely DMHE with pre-estimation) and DMHE$^\ell_{pre}$ (namely DMHE with ℓ-neighborhood diffusion and pre-estimation) algorithms with existing techniques, the simulation has been run also for the centralized MHE of [Muske et al., 1993] (denoted as MHE) and the DMHE of [Farina et al., 2010a] (denoted as DMHE$_F$), [Venturino et al., 2020] (denoted as DMHE$_{pre}$, i.e. the DMHE with pre-estimation developed in Chapter 2) and [Battistelli, 2018] (denoted as DMHE$_B$).

The performance metrics that have been taken into account are the *Position Root Mean Square Error* (PRMSE) averaged over the $n_S = 100$ nodes of the network, denoted by:

$$\text{PRMSE}(t) = \frac{1}{n_S} \sum_{i \in \mathcal{N}} \left\| C^i \left(x_t - \hat{x}_{t|t}^i \right) \right\|,$$

and the *computation time* $\tau(t)$ averaged also over the entire network.

Figure 3.4 shows the time behavior of the PRMSE of all considered algorithms. The proposed DMHE$^1_{pre}$ (purple dotted-dashed line) technique offers similar results as DMHE$_B$ of [Battistelli, 2018] (green line), with a faster convergence (about 4 seconds) with respect to DMHE$_F$ of [Farina et al., 2010a] (yellow line) and DMHE$_{pre}$ [Venturino et al., 2020] (red dashed line) (about 18 seconds to converge). Further, we can notice that the proposed DMHE$^\ell_{pre}$ (blue dotted) ensures improved performance in terms of convergence time among the considered distributed algorithms.

Evaluating the computation times in Figure 3.5 shows that the algorithms with pre-estimation (red, purple and blue dots) are always less computationally demanding compared to their respective version without pre-estimation (yellow and green dots). Indeed, adding the pre-estimation reduces, in this case, the computation time of about 30%. In particular, the proposed DMHE$^1_{pre}$ technique

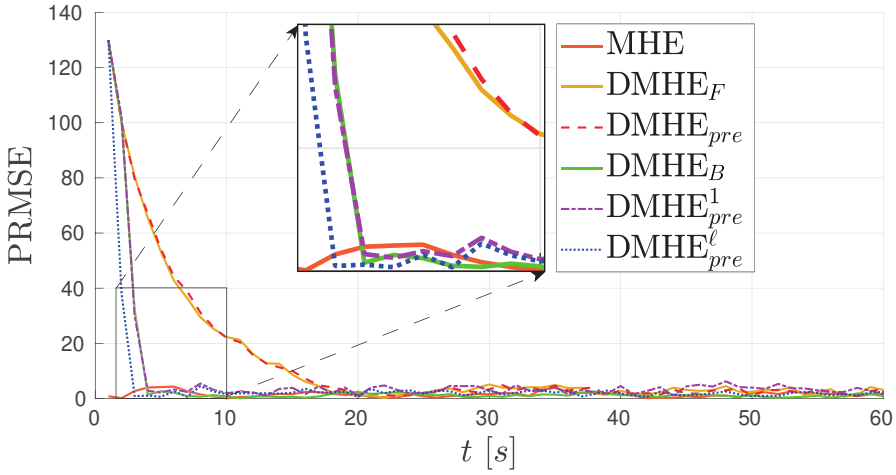


Figure 3.4: PRMSE time behavior comparison.

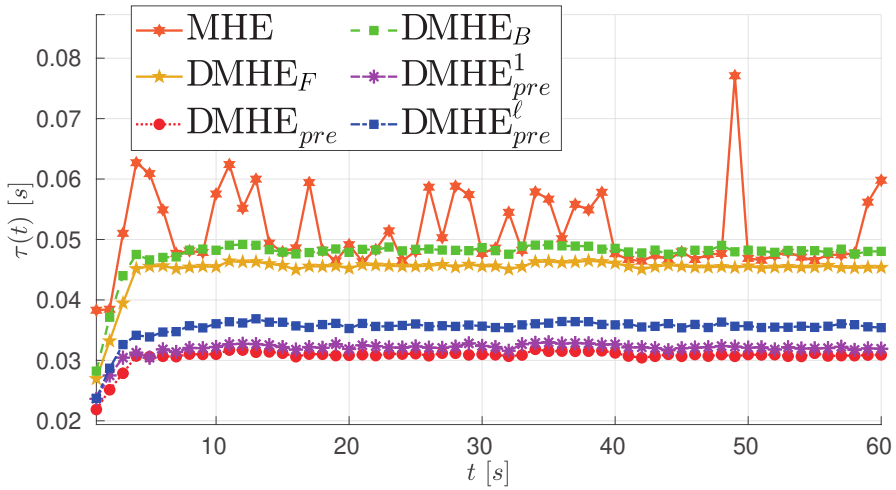


Figure 3.5: Computation time comparison.

(purple dots) converges faster and has comparable performances with the DMHE of [Venturino et al., 2020]. Moreover, it is worth to notice that DMHE^ℓ_{pre} has the best convergence time from all the considered approaches (see Figure 3.4) and needs almost the same computation time as DMHE^1_{pre} and DMHE_{pre} of [Venturino et al., 2020].

Case 2. A second simulation of nine trials has been performed using the same parameters but changing the fixed window size $N = \{2, 3, \ldots, 10\}$, to the end of evaluating how the horizon length affects the performance of the considered DMHE algorithms. In addition, the initial state of the system x_0 is randomly generated with uniform distribution over the plane $[-500, 500] \times [-500, 500]$ m and in velocity $[-1, 1] \times [-1, 1]$ m/s, while the initial estimates $\hat{x}^i_0$ are chosen as in Case 1. The simulation duration is chosen to be $t_f = 20s$.

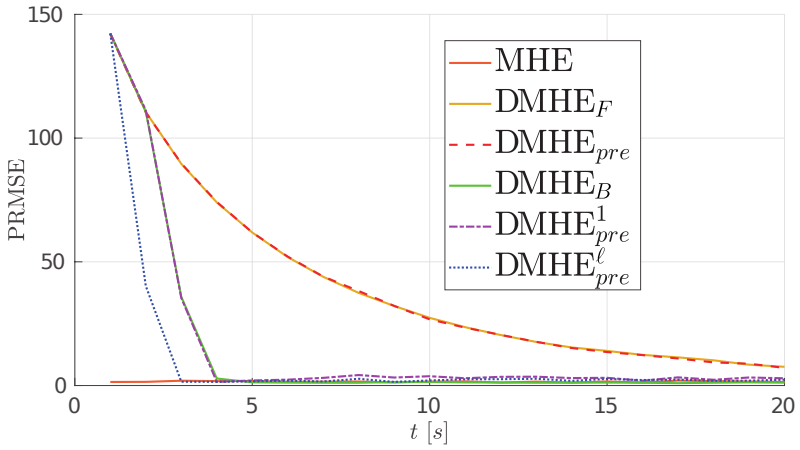


Figure 3.6: Averaged PRMSE time behavior comparison.

Figure 3.6 shows the time behavior of the PRMSE averaged over all horizons length N and sensors, i.e.:

$$\text{PRMSE}(t, N) = \frac{1}{n_S \cdot 9} \sum_{N=2}^{10} \sum_{i \in \mathcal{N}} \left\| C^i \left(x_t - \hat{x}^i_{t|t}(N) \right) \right\|.$$

We can notice that the trends are equivalent with the first simulation in Figure 3.4. To emphasize the influence of the horizon length N on the estimations, Figure 3.7 shows the evolution of the sum of the PRMSE over time of each algorithm with respect to N, i.e.:

$$\text{PRMSE}(N) = \frac{T_s}{n_S \cdot t_f} \sum_{t \in (0, t_f]} \sum_{i \in \mathcal{N}} \left\| C^i \left(x_t - \hat{x}^i_{t|t}(N) \right) \right\|.$$

As for $N = 4$ (see Case 1), the proposed technique DMHE^ℓ_{pre} (blue line) has always the best performance with respect to the distributed algorithms. Moreover, even considering information belonging only to direct neighbors, i.e. $\mathcal{N}^i_\ell = \mathcal{N}^i$, the DMHE^1_{pre} (purple dashed line) method has comparable results in terms of PRMSE with the DMHE_B of [Battistelli, 2018] (green line). In fact, this is noticeable in the zoom part on Figure 3.7 because the PRMSEs are one above the other.

Finally, Figure 3.8 points out the differences among the sum of the computation time τ of all algorithms when changing the horizon length N, i.e.:

$$\tau(N) = \sum_{t \in (0, t_f]} \tau(t, N).$$

As expected, the algorithms with pre-estimation (red, purple and blue lines) are less computation demanding for every N since their local optimization problems involves less optimization parameters. Another significant aspect to observe in Figure 3.8 is that the difference on τ among algorithms with pre-estimation (red,

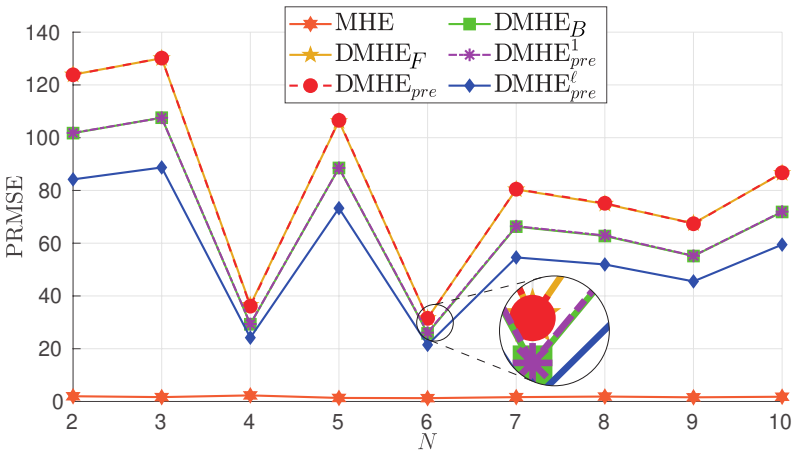


Figure 3.7: Comparison of the sum of PRMSE for a different horizon length N.

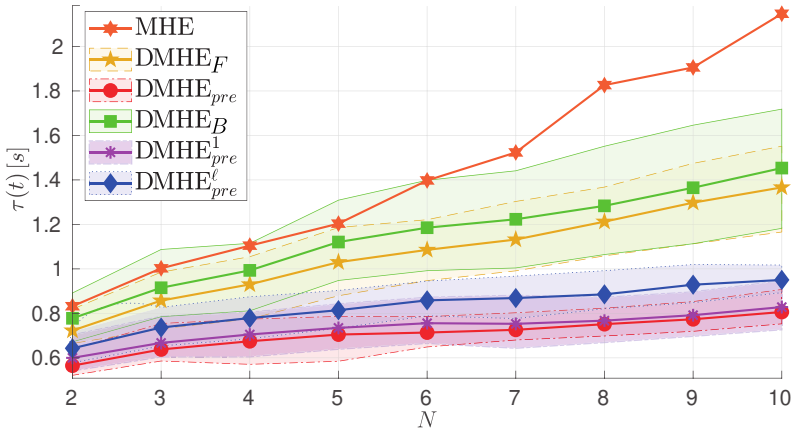


Figure 3.8: Comparison of the sum of computation time for a different horizon length N.

purple and blue lines) and without pre-estimation (yellow and green lines) increases with N. In addition, Figure 3.8 shows also the bounds representing the minimum and maximum computation time of the DMHE algorithms. It can be noticed that these bounds are tighter and less varying w.r.t. N for the algorithms with pre-estimation.

To summarize, the numerical simulations have shown that the proposed algorithm DMHE$^\ell_{pre}$ with pre-estimation and ℓ-step neighborhood information diffusion, is able to solve the considered distributed estimation problem while, at the same time, it turns out to be lower computation demanding and enhances the convergence speed of the estimates w.r.t. other existing methods [Farina et al., 2010a], [Battistelli, 2018].

3.5 Conclusion

This chapter proposed the ℓ-step neighborhood Distributed Moving Horizon Estimation (DMHE) algorithm which is able to solve the Distributed State Estimation problem for a linear system over a poorly-observing sensor network. In particular, the simulation results have shown that the proposed DMHE technique with pre-estimation DMHE^1_{pre} is able to converge with analogous performance with respect to the DMHE_B of [Battistelli, 2018] and, simultaneously, to reduce by a significant factor the computation time. The best result comes from the ℓ-step neighborhood DMHE algorithm DMHE^ℓ_{pre} that, spreading out information from neighborhood to neighborhood, both improves accuracy (in terms of the Position Root Mean Square Error), convergence speed of the estimates and reduces the computation time.

The proposed algorithm has been presented in:

- **A. Venturino**, S. Bertrand, C. Stoica Maniu, T. Alamo, and E. F. Camacho. A new ℓ-step neighbourhood distributed moving horizon estimator. In *60th IEEE Conference on Decision and Control*, pages 508–513, Austin, Texas, USA, 13-17 December, 2021.

A journal paper is in preparation.

4 - Constrained DMHE with sporadic measurements for Multi-Vehicle Systems

This chapter proposes a Distributed Moving Horizon Estimator (DMHE) for the Multi-Vehicle System localization problem using Sensor Networks with sporadic measurements, i.e. the problem of localizing vehicles using a sensor network which does not exchange information with the vehicles. In this context, measurements are available at time instants *a priori* unknown and the proposed DHME technique is designed to face this issue by resorting to time-dependent parameters in the problem formulation. Moreover, this technique is well-suited to better estimate the system state thanks to its capability to efficiently exploit environmental information via constraints. In fact, when dealing with sporadic measurements and biased noisy sensors data, the use of output constraints can contribute to locally enhance the estimation accuracy.

The effectiveness of the proposed algorithm is validated via two case studies:

- A realistic case study (video simulation available at https://youtu.be/bXV5gSmVjoc) is proposed in simulation within the Robot Operating System framework and Gazebo to localize a Multi-Vehicle System using an inexpensive Sensor Network with low-computation capabilities. A comparative campaign simulation is performed to confirm the effectiveness of the proposed DMHE algorithm in terms of accuracy, computation time, and constraints handling with respect to existing result. This result has been accepted for publication to IEEE Conference on Control Technology and Applications 2022.

- An experimental setup (video presentation of the experiment available at https://youtu.be/1CkSba2wVuI) is proposed within an indoor arena. Three scenarios are considered for the localization of a Multi-Vehicle System composed of five mobile ground robots, where the proposed DMHE technique is performed using sporadic position measurements provided by a Sensor Network with low-cost cameras and Raspberry PI computers. The estimated localization is comparable to the real position given by the motion capture system. In addition, the experimental data are further off-line analyzed. Online and off-line results are compared in terms of computation time and accuracy of the estimates with respect to existing results. These experimental results have been submitted to the Control Engineering Practice journal.

This chapter is structured as follows. Section 4.1 introduces the distributed state estimation problem using MHE schemes over Sensor Network with sporadic

measurements. Section 4.2 describes the problem under investigation and introduces the main elements for the algorithm, while Section 4.3 describe the proposed Constrained DMHE with sporadic measurements for Multi-Vehicle Systems. A realistic simulation scenario within the ROS and Gazebo environments is investigated in Section 4.4. Section 4.5 focuses on the hardware setup, the analysis and the comparison of the experimentation results obtained by using a Sensor Network with different communication topologies. Concluding remarks and further developments are drawn in Section 4.6.

4.1 Introduction

Numerous studies have been dedicated to Distributed State Estimation (DSE) over Sensor Network (SN) [Mo et al., 2011, Quevedo et al., 2012, Ding et al., 2012] during the last few years since these schemes are suitable for diverse applications and contexts. Some of these works have conducted only theoretical developments or have exclusively numerically shown the effectiveness of the considered techniques. Indeed, there is still a judicious need for deep insights such as applying distributed state estimation algorithms in real experiments and applications. For example, communication delays and packet losses [J. Zeng and Liu, 2015], computation time [Venturino et al., 2021a], the time-varying topology of the network [Yousefi and Menhaj, 2014], sporadic measurements [Postoyan and Nešić, 2011, Ferrante et al., 2016] are still open problems to cope with in theory and much more in practice.

The current chapter focuses on Multi-Vehicle System localization using DMHE algorithms. Similar works have been conducted by the authors of [Simonetto et al., 2011] and [Yousefi and Menhaj, 2014]. In [Simonetto et al., 2011], the DMHE problem has been addressed by focusing on the non linearity of the model and on the possible local observability issues at the sensor level. In [Yousefi and Menhaj, 2014], the authors accounted for mobile nodes in the Sensor Network that led to deal with a dynamic topology. Indeed, using a flocking algorithm for the motion control, the mobile sensors attempt to move in a specific way in order to get the best positions to observe the target and to avoid collisions between neighboring agents. In this chapter, we focus on the computation time aspect, which is a key factor for real-time implementation.

In Chapters 2 and 3, DMHE algorithms with pre-estimation have been proposed in order to reduce the computation time while preserving or improving the accuracy of the state estimation. To this aim, the input sequence of noise to be estimated has been replaced by a Luenberger observer leading to fewer optimization parameters to be accounted. Furthermore, an observability rank-based weights technique has been used to enhance the accuracy. The algorithm proposed in this chapter is developed using the DMHE in Chapter 2 as a theoretical basis. The main reason is that it is simpler to implement for experiments on a real harware setup with

respect to the ℓ-step DMHE algorithm developed in Chapter 3.

The contribution of the current chapter is two-fold. First, in addition to a reduced computation time and an improved accuracy due to the pre-estimation, the proposed DMHE technique is designed for realistic large-scale systems scenarios involving sporadic measurements (i.e. available at time instants *a priori* unknown). To this aim, constraints on measurements (coming from the knowledge of the environment where the Multi-Vehicle System is evolving) are embodied using binary parameters in this novel Distributed Moving Horizon Estimation formulation. Thus, the environment information is exploited to better estimate the system state. Second, this chapter aims at evaluating the performance of the proposed DMHE approach (in terms of accuracy and computation time) on a realistic case study, i.e. the distributed localization of a Multi-Vehicle System by a static sensor network, developed within the ROS framework and Gazebo environment (for the simulation part). This realistic distributed implementation within ROS and Gazebo has enabled the deployment on a hardware setup. Thus, the proposed DMHE approach is evaluated in terms of accuracy and computation time on three real experiments using different numbers of sensors, distinct communication network topologies and diverse coverage of the cameras' fields of view. Indeed, one of the main contributions of this chapter consists in the experimental validation of the proposed distributed MHE localization technique of a Multi-Vehicle System. In the developed experiment setup, the static Sensor Network is composed of low-cost cameras which provide measurements on the positions of the vehicles. Each camera is attached to a Raspberry PI for computational and communication capabilities. The proposed DMHE algorithm has been implemented within the ROS framework to run in a distributed way on each Raspberry PI. The Multi-Vehicle System is composed of five TurtleBot3 robots performing formation motion within a road-like area located in an indoor arena equipped with a motion capture system. This allows to compare the real position (provided by the Optitrack motion capture) with the position estimated by the low cost Logitech camera (webcam) network using the proposed DMHE algorithm.

The next section describes the problem under investigation and introduces the main elements for the algorithm.

4.2 Distributed State Estimation over a static Sensor Network with Sporadic Measurements

This section describes the problem of DSE of a Multi-Vehicle System over a Sensor Network (SN) with sporadic measurements, the considered models, and the characteristics of the Sensor Network.

4.2.1 Problem description

Consider the problem of Distributed State Estimation of the state (e.g. 2D position) of a Multi-Vehicle System by a SN. In this setting, we assume that the

Sensor Network is composed of n_S different sensors performing sporadic measurements, i.e. the measurements are not obtainable at all times by each sensor. For example, a moving vehicle can be detected by a camera only when it is within its field of view, or by a beacon when within its detection range, etc. A (formation[1] of) vehicle(s) moving in unknown directions can thus be detected by a given sensor belonging to a Sensor Network at time instants *a priori* unknown.

The Multi Vehicle System under observation consists of n_V ground vehicles which are restricted to move in specific locations, e.g. on roads in urban environments (delimited by the yellow borders in Figure 4.1). We additionally make use of this environment knowledge as position constraints in the Distributed State Estimation optimization problem.

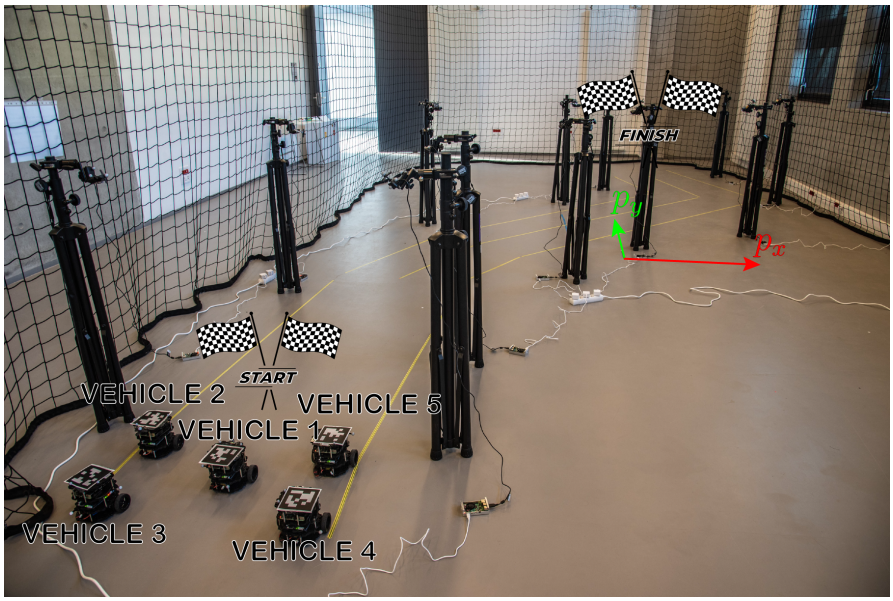


Figure 4.1: Experiment scenario setup: Multi-Vehicle System with 5 vehicles in the starting place and Sensor Network composed of 12 cameras and Raspberry PI computers.

4.2.2 Considered dynamical model

Consider n_V vehicles. The ν-th vehicle dynamical model is represented as a discrete-time linear time-invariant (LTI) system:

$$ {}^{\nu}x_{t+1} = {}^{\nu}A \ {}^{\nu}x_t + {}^{\nu}w_t, \qquad \nu = 1, \dots, n_V, \tag{4.1} $$

where ${}^{\nu}x_t \in {}^{\nu}\mathcal{X} \subseteq \mathbb{R}^{{}^{\nu}n_x}$ is the state vector and ${}^{\nu}w_t \in {}^{\nu}\mathcal{W} \subseteq \mathbb{R}^{{}^{\nu}n_x}$ is an exogenous input (e.g. an unknown control input, state perturbation, etc.), with ${}^{\nu}\mathcal{X}$ and ${}^{\nu}\mathcal{W}$ convex sets.

[1]This chapter also deals with the case when only a part of a formation of vehicles can be within the field of view of a camera.

Remark 11. *Notice that, according to the adopted notation, the state of the* global *Multi-Vehicle System is denoted by* $x_t = [{}^1x_t^\top, \ldots, {}^{n_V}x_t^\top]^\top$*, where its* global *dynamics is described by* $A = diag({}^1A, \ldots, {}^{n_V}A)$*.*

Given that each vehicle can be detected individually by each sensor i (right superscript), the following mathematical expression models the measurement provided by sensor i with respect to the ν-th vehicle:

$$ {}^\nu y_t^i = {}^\nu C^i \ {}^\nu x_t + {}^\nu v_t^i, \qquad i = 1, \ldots, n_S, \tag{4.2} $$

where ${}^\nu y_t^i \in \mathbb{R}^{n_y^i}$ is the measurement vector and ${}^\nu v_t^i \in \mathbb{R}^{n_y^i}$ is the measurement noise with covariance R^i.

Remark 12. *Notice that in* (4.2)*, the right superscript* i *refers to the* i*-th sensor and the left superscript* ν *to the* ν*-th vehicle. In this respect,* ${}^\nu C^i$ *is the output matrix specifying that the sensor* i *is producing a measurement on the state of the vehicle* ν*. The notation* C^i *(without the left superscript* ν*) refers to the output matrix of the* global *Multi-Vehicle System* $C^i = diag({}^1C^i, \ldots, {}^{n_V}C^i)$*.*

For simplicity of the formulation, we assume that each sensor can measure only the vehicles' position without losing generality. In this chapter, we consider that all the measurements are sporadic. This way, we avoid abstruse notation to discern sporadic and non-sporadic measurements.

The following notation is necessary to denote the global system's collective output matrix which aggregates both the measuring and non-measuring situations of each sensor i:

$$ C_{\alpha_t}^i = D_{\alpha_t}^i C^i, \tag{4.3} $$

where $D_{\alpha t}^i$ is a squared diagonal matrix of size $\sum_{\nu=1}^{n_V} {}^\nu n_y$ with ${}^\nu\alpha_t^i \in \{0, 1\}$ as components, for $\nu = 1, \ldots, n_V$, defined as:

$$ D_{\alpha_t}^i = \text{diag}({}^1\alpha_t^i \ I_{{}^1n_y}, \ldots, {}^{n_V}\alpha_t^i \ I_{{}^{n_V}n_y}), \tag{4.4} $$

with $I_{{}^\nu n_y}$ the identity matrix of dimension ${}^\nu n_y$.

Remark 13. *Notice that* ${}^\nu\alpha_t^i$ *is a time-dependent binary parameter marking if the sensor* i *can detect the* ν*-th vehicle at time* t *(i.e.* ${}^\nu\alpha_t^i = 1$*) or not (i.e.* ${}^\nu\alpha_t^i = 0$*).*

Definition 4.1 **Active and inactive sensor**

The sensor i *is called* active *sensor at time* t *if there* $\exists$ *at least one* $\nu \in \{1, ..., n_\nu\}$ *such that* ${}^\nu\alpha_t^i = 1$ *(sensor* i *is considered active iff it sees at least one of the vehicles),* inactive *otherwise.*

This definition plays an important role for the estimation of the trajectory of intruder vehicles by a Sensor Network.

4.2.3 Constraints

This subsection defines measurement constraints exploiting the *a priori* knowledge of the environment and the cameras composing the Sensor Network. First, denote by $\mathcal{R}$ the subset of planar coordinates corresponding to the road (assumed to be non-convex and marked by the blue lines in Figure 4.2) on which the vehicles can drive. Further on, denote by $\mathcal{F}^i$ the set of the points forming the sensor i field of view (in yellow in Figure 4.2). The convex hull of the intersection of these two sets denoted by:

$$\mathcal{S}^i = \text{Co}(\mathcal{R} \cap \mathcal{F}^i) \tag{4.5}$$

is further used to constrain the position of the vehicle in the state estimation process when the mobile vehicle is within the field of view of the sensor i, i.e. when this sensor detects the vehicle (see Figure 4.2 for a graphic illustration).

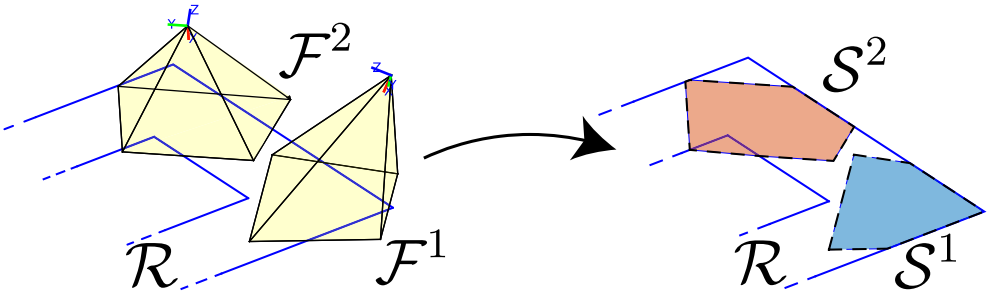


Figure 4.2: Road $\mathcal{R}$ (blue line), fields of view $\mathcal{F}^1$ and $\mathcal{F}^2$ (yellow), and convexified constraints $\mathcal{S}^1$ (red polygone) and $\mathcal{S}^2$ (blue polygone).

It is worth noticing that the convex hull operation leads to constraints with twofold aspects. The first one, is that the constraints are convex, hence easily tractable within quadratic programming solving algorithms. The second one is that the set $\mathcal{S}^i$ might contain points outside the road $\mathcal{R}$ since the convex hull is an outer approximation of a set (see the set $\mathcal{S}^2$ in Figure 4.2). Thus, this operation is a trade-off between simplicity (convex set) and accuracy of the estimation (estimates outside the road).

4.2.4 Problem statement

The Sensor Network description explained in Section 2.2.3 is used in this chapter.

Problem 4.1 Distributed State Estimation over Sensor Network with sporadic measurements for Multi-Vehicle Systems

Consider the discrete-time LTI system (4.1) *and the Sensor Network* $\mathcal{G}$ *with the linear measurement equation* (4.2)*, under the assumption that the graph* $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ *is strongly connected, i.e. every node is reachable from every other node (see Definition 2.5). The role of each sensor* $i \in \mathcal{N}$*, at each time* t*, is to (possibly, since measurements are sporadic) get measurement on (part of) the Multi-Vehicle System, to exchange information among neighbor nodes of* $\mathcal{N}^i$ *and to process locally available information in order to determine a local estimate* $\hat{x}_t^i$ *of the real state* x_t *of the Multi-Vehicle System.*

The next section gives the theoretical insights of the proposed DMHE solution taking into account sporadic measurements and describes the proposed DMHE algorithm that solves the Problem 4.1.

4.3 Proposed DMHE technique

This section recalls the Distributed Moving Horizon Estimation approach with pre-estimation and observability rank-based weights proposed in Chapter 2 and presents its novel formulation to handle the Multi-Vehicle localization application considered in the current chapter.

4.3.1 Pre-estimation observer

Given sensor i, let us consider the global Multi-Vehicle System dynamical evolution to estimate as follows:

$$\hat{x}_{t+1}^i = A\,\hat{x}_t^i + \hat{w}_t^i. \tag{4.6}$$

Equation (4.6) is usually used in classical MHE formulations. In this chapter, it is replaced by the following pre-estimation Luenberger observer:

$$\hat{x}_{t+1}^i = A\,\hat{x}_t^i + L_{\alpha_t}^i \left(y_t^i - C^i\,\hat{x}_t^i\right). \tag{4.7}$$

Thanks to the pre-estimation, the proposed DMHE technique reduces the computation time needed by the sensors (see Chapter 2) to estimate the state of the system compared to classical DMHE (see [Farina et al., 2010a]). This is due to a reduced number of optimization variables.

The dependence with ${}^{\nu}\alpha_t^i$ is formulated via $L_{\alpha_t}^i = L^i D_{\alpha_t}^i$, with $D_{\alpha_t}^i$ defined by (4.4). Moreover, the global Luenberger gain L^i is computed such that $\Phi^i = A - L^i C^i$ is Schur stable when the Multi-Vehicle System is detectable by sensor i, i.e. the pair (A, C^i) is detectable. One may compute the gain related to the global Multi-Vehicle System or separately, since $L^i = [{}^1L^i, \ldots, {}^{n_V}L^i]$.

4.3.2 Field of View constraints and sporadic measurements

The binary parameter ${}^{\nu}\alpha_t^i$ allows to deal with the sporadic measurements. Indeed, it is effective to discern when the constraints $\mathcal{S}^i$ can be used by sensor i (active constraints) and when not (inactive constraints), i.e., respectively, when the sensor i can detect a specific vehicle and when not. In particular, considering the vehicle ν being detected or not by sensor i at time t, the following constraints ${}^{\nu}\mathcal{S}_{\alpha_t}^i$ are defined:

$${}^{\nu}\mathcal{S}_{\alpha_t}^i = \begin{cases} \mathcal{S}^i & \text{if } {}^{\nu}\alpha_t^i = 1 \\ \mathbb{R}^{{}^{\nu}n_x} & \text{if } {}^{\nu}\alpha_t^i = 0. \end{cases}$$

Figure 4.3 shows a graphical illustration example of a vehicle driving in the road and detected by the sensor 2. In this example then, at the current time t, ${}^{\nu}\mathcal{S}_{\alpha_t}^2 = \mathcal{S}^2$ (i.e. sensor 2 is active and uses constraints), while ${}^{\nu}\mathcal{S}_{\alpha_t}^1 = \mathbb{R}^{{}^{\nu}n_x}$ (i.e. sensor 1 is inactive and does not use constraints).

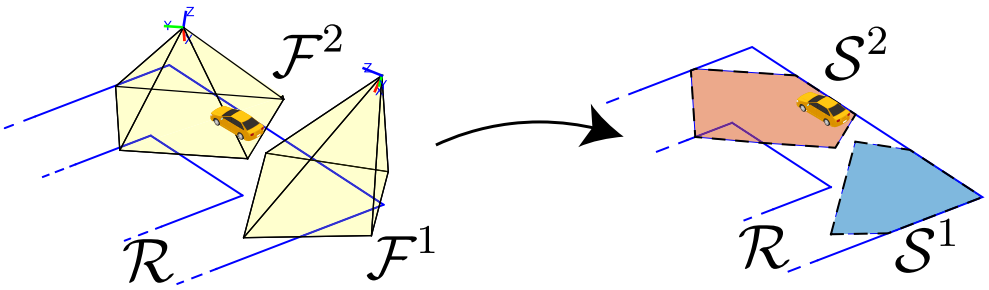


Figure 4.3: Example of active and inactive sensors.

4.3.3 Objective function with sporadic measurements

The objective function is similar to the one used in Chapter 2 with the difference of dealing with sporadic measurements. Indeed, the binary parameter ${}^{\nu}\alpha_t^i$ plays a crucial role also in the objective function $J_{\alpha_t}^i$, which is defined as:

$$J_{\alpha_t}^i(\cdot) = \frac{1}{2} \sum_{k=t-N}^{t} \left\| \bar{y}_k^i - \bar{C}_{\alpha_k}^i \hat{x}_k^i \right\|_{(\bar{R}^i)^{-1}}^2 + \Gamma_{t-N}^i(\cdot), \tag{4.8}$$

where $\bar{R}^i$ is the regional covariance matrix of the measurement noise. Here, we assume that $\bar{R}^i$ is a positive definite matrix. The term $\bar{R}^i$ weights the difference between the predicted outputs and the measurements within the fixed window of size N. Notice that when $\alpha_k^i = 0$, the following equality holds $\left\| \bar{y}_k^i - \bar{C}_{\alpha_k}^i \hat{x}_k^i \right\| = \left\| \bar{y}_k^i \right\|$, thus a term that does not depend on the optimization parameters and does not effect the solution. The same reasoning can apply when considering each vehicle individually, i.e. when ${}^{\nu}\alpha_k^i = 0$. The *arrival cost* $\Gamma_{t-N}^i(\cdot)$ is a non negative term summarizing the effect of the past measurements, before time $t-N$ and is usually approximated by some *initial penalty* function defined as follows:

$$\Gamma_{t-N}^i(\cdot) = \frac{1}{2} \left\| \hat{x}_{t-N}^i - \hat{\bar{x}}_{t-N|t-1}^i \right\|_{(\bar{\Pi}_{t-N|t-1}^i)^{-1}}^2, \tag{4.9}$$

which involves two consensus terms, $\hat{\bar{x}}^i_{t-N|t-1}$ and $\bar{\Pi}^i_{t-N|t-1}$, described below.

4.3.4 Consensus terms

The consensus terms are also similar to the ones of Chapter 2, but here adapted to the case of sporadicalness of the measurements. In fact, for this algorithm, the components k_{ij} of the stochastic matrix K are time-varying, denoted by $k_{ij|t}$ and computed as described in the next subsection.

The first term included in the penalty function Γ^i_{t-N} is the *consensus-on-estimates* term, denoted by $\hat{\bar{x}}^i_{t-N|t-1}$. It consists in a weighted average state estimate computed over the neighborhood $\mathcal{N}^i$ as follows:

$$\hat{\bar{x}}^i_{t-N|t-1} = \sum_{j\in\mathcal{N}^i} k_{ij|t}\hat{x}^j_{t-N|t-1}, \tag{4.10}$$

where $\hat{x}^j_{t-N|t-1}$ is the estimated state computed at time $t-1$ by sensor $j \in \mathcal{N}^i$. It is a consensus term in the sense that it penalizes deviations of $\hat{x}^i_{t-N}$ from $\hat{\bar{x}}^i_{t-N|t-1}$.

The second term is the positive definite matrix $\bar{\Pi}^i_{t-N|t-1}$ computed as:

$$\bar{\Pi}^i_{t-N|t-1} = \sum_{j\in\mathcal{N}^i} k_{ij|t}\Pi^j_{t-N|t-1}, \tag{4.11}$$

where the matrices $\Pi^j_{t-N|t-1}$ are obtained in the same way as described in Section 2.4.1 starting from Equation (2.23).

4.3.5 Observability rank-based weights technique with sporadic measurements

Here, we adjust the weights tuning technique in Chapter 2 for the stochastic matrix K associated with the graph $\mathcal{G}$ for the considered Multi-Vehicle localization problem by DMHE over a Sensor Network with sporadic measurements.

Thanks to this method, each sensor i computes its components of K based on only *locally* available data. Hence, it is appropriate for a distributed scheme, and furthermore, for the application considered in this chapter with sporadic measurements. Indeed, this technique enables to improve the accuracy of the estimates by relying more on the sensors that are currently sensing, in other words, by exploiting the observability properties of the neighborhoods. Since these properties are time-varying for this Sensor Network due to the sporadicalness aspect, the observability rank-based weights technique is suitable for enhancing the algorithm's accuracy and convergence time.

Consider a sensor i at time t. Its current N-step regional observability matrix:

$$\bar{\mathcal{O}}^i_{N|t} = \left[(\bar{C}^i_{\alpha_{t-N+1}})^\top \quad (\bar{C}^i_{\alpha_{t-N+2}}A)^\top \quad \cdots \quad (\bar{C}^i_{\alpha_t}A^{N-1})^\top\right]^\top \tag{4.12}$$

is of full rank if and only if the the pair $(A, \bar{C}^i_{\alpha_t})$ is completely observable at any instant k within the interval $[t-N+1, \ldots, t]$, i.e. $\text{rank}(\bar{\mathcal{O}}^i_{N|t}) = \sum_{\nu=1}^{n_V} {}^{\nu}n_x$. For

simplicity, we denote by $\rho^i_{\mathcal{O}|t} = \text{rank}(\bar{\mathcal{O}}^i_{N|t})$. This variable will be considered as an information on the *reliability* of node i, i.e. its sensing capability, and will be used to define the weighting coefficients k_{ij}, which must satisfy the constraint (2.4). Notice that, at some time instants *a priori* unknown, the entire neighborhood may not have sensing capabilities at all, i.e. $\rho^i_{\mathcal{O}|t} = 0$. To avoid division by zero a lower bound ε smaller than 1 is chosen for the rank, here $\varepsilon = 0.5$ is chosen, i.e. $\rho^i_{\mathcal{O}|t} = \max\left\{\text{rank}(\bar{\mathcal{O}}^i_{N|t}), \varepsilon\right\}$. Then the components $k_{ij|t}$ are computed as follows:

$$k_{ij|t} = \frac{\rho^j_{\mathcal{O}|t}}{\sum_{j\in\mathcal{N}^i} \rho^j_{\mathcal{O}|t}}. \tag{4.13}$$

4.3.6 Local optimization problem with sporadic measurements

Given an estimation horizon length $N \geqslant 1$, at each time t, each sensor $i \in \mathcal{N}$ determines the state estimate $\hat{x}^i_{t|t}$ by solving the following constrained minimization problem:

$$\hat{x}^i_{t-N|t} = \arg\min_{\hat{x}^i_{t-N}} \quad J^i_{\alpha_t}(\cdot) \tag{4.14}$$

$$\text{s.t.} \quad \hat{x}^i_{k+1} = A\,\hat{x}^i_k + L^i_{\alpha_k}\left(y^i_k - C^i\,\hat{x}^i_k\right), \quad \forall k = t-N,\dots,t-1, \tag{4.15}$$

$$\hat{x}^i_k \in \mathcal{X}, \qquad \forall k = t-N,\dots,t, \tag{4.16}$$

$$\bar{C}^i\,\hat{x}^i_k \in \mathcal{S}^i_{\alpha_k}, \qquad \forall k = t-N,\dots,t, \tag{4.17}$$

with $J^i_{\alpha_t}$ given by (4.8). The sequence of state estimates $\hat{x}^i_{t-N+1|t},\dots,\hat{x}^i_{t|t}$ is obtained from the optimal solution $\hat{x}^i_{t-N|t}$ and using the dynamics (4.15). The A matrix in (4.15) refers to the global Multi-Vehicle System. Moreover, the FoV constraints, as described in Section 4.2.3, are integrated in the optimization problem in (4.16).

Remark 14. *The local optimization problem could have been also divided in several problems, one for each vehicle, but solving one for the entire Multi-Vehicle System is less time demanding and less complex.*

4.3.7 DMHE *modus operandi*

Finally, the procedure of the proposed distributed scheme is described in Algorithm 3.

The specific steps related to sporadic measurements are integrated at step 19, with $D^i_{\alpha_t}$ computed at step 11. Notice that the steps 10, 18 and 21 in the procedure regarding the exchanging information could be rearranged to include only one synchronization. However, the current formulation has been chosen for clarity reasons w.r.t. calculation details.

Algorithm 3 DMHE procedure with sporadic measurements

1: **Off-line:** $\forall i \in \mathcal{N}$
2: **receive** from the neighbor nodes $j \in \mathcal{N}^i$: L^j, C^j, R^j
3: **compute** the pre-estimation Luenberger gain L^i
4: **store** the *a priori* initial estimation $\hat{x}^i_{0|0} = \hat{x}_0$ of x_0, where $\hat{x}_0$ is given, and the covariance matrix Π_0 of x_0
5: **Initialization:** $\forall i \in \mathcal{N}$, at the first time step $t = 0$
6: **collect** a first local measurement y^i_0
7: **receive** from neighbors $j \in \mathcal{N}^i$ their measurements y^j_0
8: **Online:** $\forall i \in \mathcal{N}$, $\forall t > 0$
9: **collect** the local measurement y^i_t
10: **receive** from the neighbors $j \in \mathcal{N}^i$ the collected data in step 9
11: **compute** the matrix $D^i_{\alpha_t}$ according to (4.4)
12: **compute** the components $k_{ij|t}$ according to (4.13)
13: **if** $1 \leqslant t \leqslant N$ **then**
14: **set** the horizon length $N = t$, the covariance matrix $\bar{\Pi}^i_{t-N|t-1} = \bar{\Pi}^i_{0|t-1} = \Pi_0$ and the *a priori* initial estimation state $\hat{x}^i_{t-N|t-1} = \hat{x}^i_{0|t-1}$
15: **else**
16: **compute** $\Pi^i_{t-N|t-1}$ according to (2.28), (2.29) and (2.30)
17: **receive** $\Pi^j_{t-N|t-1}$ from the neighbor nodes $j \in \mathcal{N}^i$
18: **compute** $\bar{\Pi}^i_{t-N|t-1}$ according to (4.11)
19: **solve** the local optimization problem of DMHE, minimizing J^i as in (4.8) and (4.9) subject to the constraints (4.15)-(4.16)
20: **store** the solution $\hat{x}^i_{t-N|t}$ and the corresponding estimate $\hat{x}^i_{t|t}$
21: **receive** from the neighbors $j \in \mathcal{N}^i$ their estimates $\hat{x}^j_{t-N+1|t}$

The next section provides the results obtained by implementing the proposed algorithm for a realistic simulation within the ROS framework and Gazebo simulator.

4.4 Realistic simulations

4.4.1 Scenario and simulation setup

In this section, the proposed DMHE is applied to estimate the positions of a team of $n_V = 5$ ground vehicles moving together. To evaluate its performance a realistic implementation in the ROS framework and in the Gazebo simulation is proposed, see Figure 4.4 showing the simulation scenario, i.e. the five vehicles at their starting positions, the road in which they have to drive and the final point to reach, and the cameras' field of view. A simulation video is available at https://youtu.be/KRvlQgvHGEo.

For the estimation models used in the DMHE optimization problem, each vehicle is modeled as single integrator, with a 2-dimensional state vector representing its Cartesian positions in the plane. The control input vector, i.e. Cartesian lin-

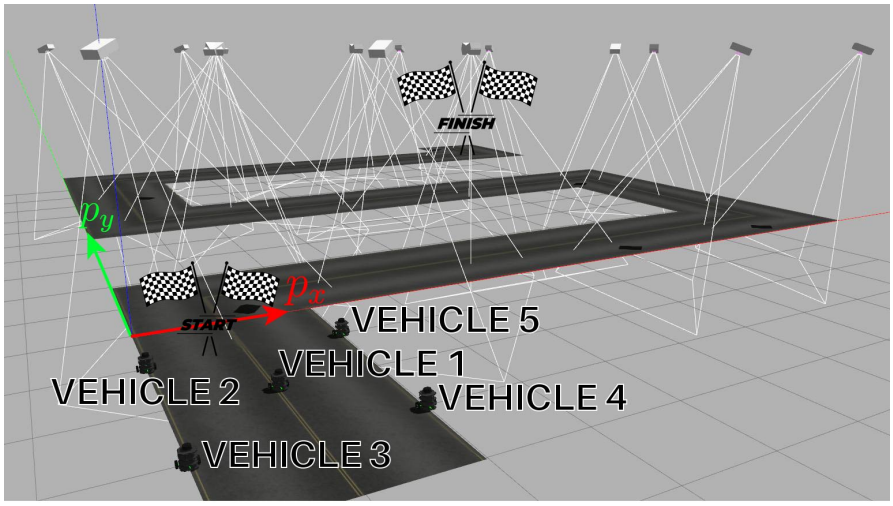


Figure 4.4: Scenario illustrated in Gazebo: Multi-Vehicle System with 5 vehicles in the starting place.

ear velocities, of each vehicle is assumed to be unknown by the Sensor Network (non cooperative), and it is further considered as an exogenous input, modeling ${}^{\nu}w_t \in \mathbb{R}^2$ as an uniformly distributed noise with covariance matrix $Q = I_2$.

Figure 4.5: TurtleBot3 within the indoor arena.

To simulate a realistic system, each vehicle is modeled in Gazebo as a differential drive robot (TurtleBot3, see Figure 4.5).

The Multi-Vehicle System goes from the starting point $(1, -2)$ m towards the final point $(11, 11)$ m driving within the road and controlled by a leader-follower formation control strategy.

The Sensor Network is composed of $n_S = 17$ cameras measuring the Cartesian positions of the vehicles, and connected as in the Figure 4.6 (see the red edges representing the communication links between the nodes depicting the cameras). Figure 4.6 also shows the projection on the ground of the field of view of each

camera (yellow rectangles), and the road (blue solid line). The start and finish position of the Multi-Vehicle System are clearly indicated in Figures 4.4 and 4.6.

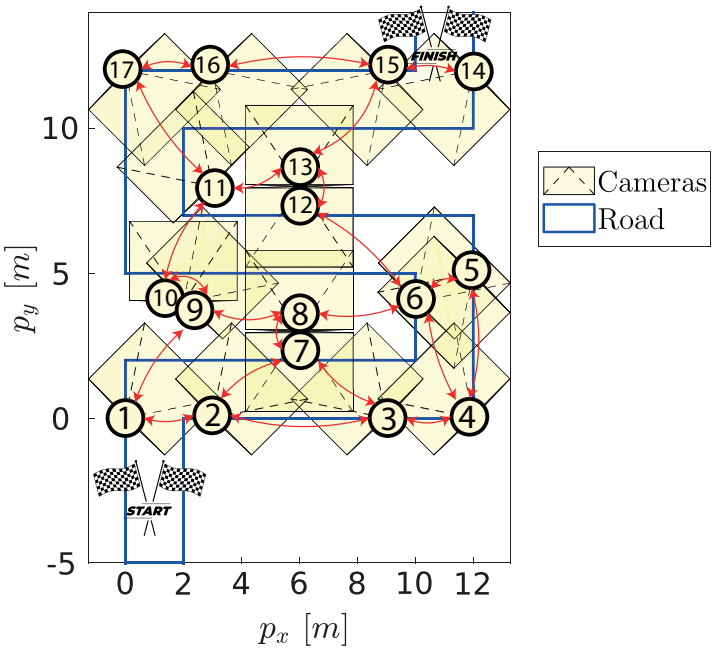


Figure 4.6: Simulation scenario with road (blue line), cameras (numbered nodes), projection of the field of view of each camera (yellow rectangles), and communication link between the sensors (red arrows).

Notice that the graph associated with the Sensor Network (see Figure 4.6) is not a complete graph, i.e. a graph in which every pair of distinct vertices is connected by a unique edge (see Definition 2.6). The measurements refer to the reference frame associated with each camera. Thus, to have them in the absolute reference frame it is necessary to translate and rotate them with a transformation matrix. To make the scenario more realistic, we added different biases for each camera (via the measurement equation (4.2) for each sensor) on these translations and rotations, allowing to model uncertainties related to the cameras' poses.

Assuming that these biases can not be easily estimated and compensated in the considered scenario, the purpose is to investigate the robustness of the proposed Distributed Moving Horizon Estimation to this additional source of uncertainty (i.e. sensor biases) and to validate the usefulness of *a priori* known Field of View (FoV) constraints considered in the DMHE optimization problem.

A Monte Carlo simulation with 100 runs with different measurements noises (per run) normally distributed (i.e. a white noise with zero mean and covariance matrix $R^i = 0.5I_2$, $\forall i$) was performed. The estimators runs with a sampling time $T_s = 0.5$ s and a horizon length $N = 4$. The initial values of the algorithms have been set as ${}^{\nu}\hat{x}_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top$, $\forall \nu$, $\Pi_0 = 10^5 I_2$. The optimization problem was implemented by using the quadratic programming solver from [Goldfarb and

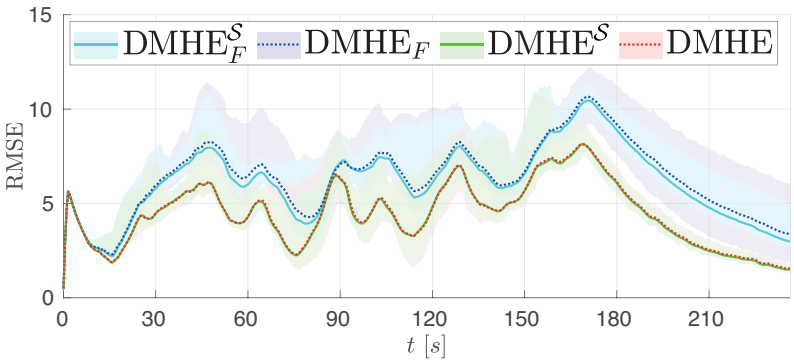


Figure 4.7: Averaged RMSE among all the sensors and all the trials.

Idnani, 1983] implemented in Python. The considered performance indexes are the computation time τ needed by the solver to estimate the positions of the Multi-Vehicle System, and the RMSE computed as follows:

$$\mathrm{RMSE}_t = \frac{1}{100 \cdot n_S} \sum_{\sigma=1}^{100} \sum_{i \in \mathcal{N}} \left\| x_t(\sigma) - \hat{x}^i_{t|t}(\sigma) \right\|,$$

both averaged among the trials and the sensors, where $x_t(\sigma)$ and $\hat{x}^i_{t|t}(\sigma)$ are, respectively, the realization of the real state of the system and the estimated one, by sensor i, for the trial σ. The simulation is carried out by a PC Linux Ubuntu 20.04 equipped with an Intel i9-11950H processor.

We compare the proposed Distributed Moving Horizon Estimation without FoV constraints (denoted by DMHE) and with FoV constraints[2] (denoted by $\mathrm{DMHE}^{\mathcal{S}}$). We compare the results with the algorithm in [Farina et al., 2010a], denoted hereafter by DMHE_F for the case without FoV constraints. We also added the FoV constraints (4.5) to this approach, denoted hereafter by $\mathrm{DMHE}_F^{\mathcal{S}}$.

4.4.2 Results' analysis

Figure 4.7 illustrates the averaged RMSE among all the sensors and all the 100 trials. It shows that the proposed DMHE (red dotted curve) and its constrained case $\mathrm{DMHE}^{\mathcal{S}}$ (solid green curve) have better accuracy w.r.t. to the approach in [Farina et al., 2010a], with FoV constraints (solid cyan curve) or without FoV constraints (dark blue dotted curve). Indeed, the RMSE obtained with the proposed estimation approaches (both DMHE and $\mathrm{DMHE}^{\mathcal{S}}$) are improved by a factor close to 30% w.r.t. the RMSE of DMHE_F and $\mathrm{DMHE}_F^{\mathcal{S}}$. This figure shows also the bounds (shaded colors) representing the minimum and the maximum RMSE of each trial and for each individual local observer. It is worth noticing that these bounds are narrower for the constrained algorithms w.r.t. their counterparts without FoV constraints.

[2]The constraints are added as in (4.5).

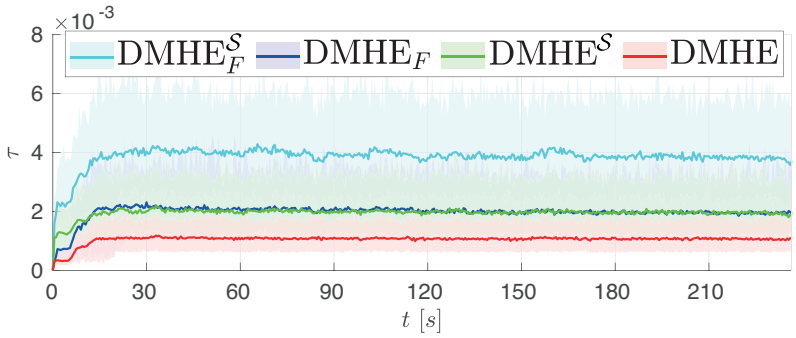


Figure 4.8: Averaged computation time τ among all the sensors and all the trials.

The same trend can be seen in Figure 4.8 showing the computation time τ averaged among all the sensors and trials. Accounting for FoV constraints is done at the cost of an increase of the computation time (close to a factor 2). The proposed pre-estimation mechanism enables to compensate that by drastically reducing the computation time. Here, the bounds, representing the minimum and the maximum τ of each trial and for each individual local observer of the proposed DMHEs are tighter than the bounds obtained with the DMHE$_F$ and DMHE$_F^{\mathcal{S}}$.

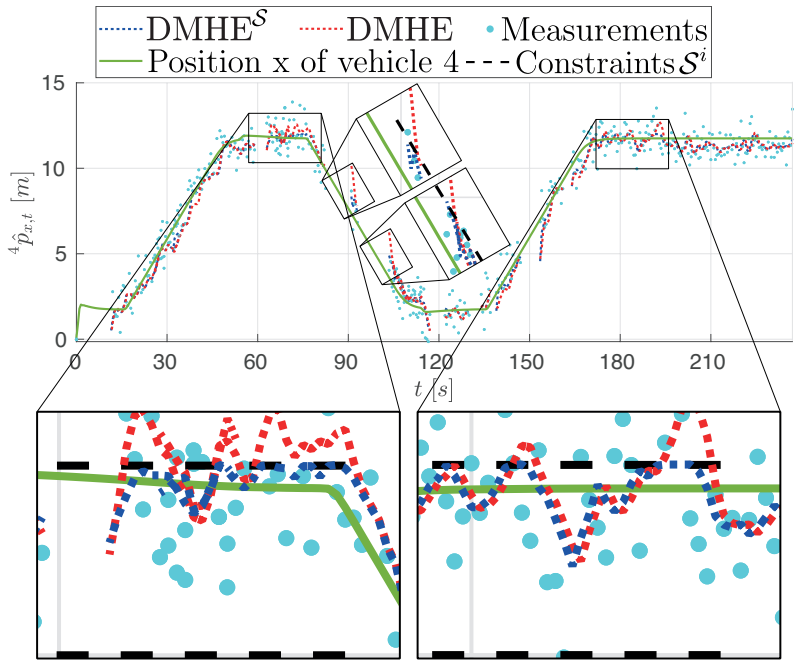


Figure 4.9: Estimation of the position along the x-axis of the fourth vehicle ${}^4\hat{p}_{x,t}$ by all the active sensors.

The FoV constraints $\mathcal{S}^i$ are used in the local optimization problem only when the camera is actually sensing a vehicle, i.e. when the sensor i is active as described in Section 4.3.2. In order to show the effects of these FoV constraints $\mathcal{S}^i$ (4.5),

consider, for one random trial, the estimations of the position along the x-axis for the fourth vehicle. In particular, it is more relevant to plot only the estimations during the time periods when their sensors are active, i.e. when the vehicle belongs to its field of view. Thus, Figure 4.9 illustrates the simulated position ${}^4p_{x,t}$ (in green solid line) provided by the Gazebo simulator, the measurements of the cameras (cyan dots), the estimations using DMHE (red dotted curves) and DMHE$^{\mathcal{S}}$ (dark blue dotted curves). The zoomed parts also show the local FoV constraints (black dashed lines). Due to the measurement noise and bias, some measurements (cyan dots) could not correspond to possible positions of the vehicle which are constrained to be within the road boundaries. Accounting explicitly for FoV constraints in the estimation, therefore, helps to improve the accuracy. Figure 4.9 illustrates that the estimations with the DMHE$^{\mathcal{S}}$ (dark blue dotted curves) method respects the FoV constraints represented by black dashed lines in the zoomed parts.

The next section provides the experimental results obtained by implementing the proposed algorithm for localizing an actual Multi-Vehicle System using the developed multi-camera sensor network.

4.5 Experiments

This section describes three conducted experiments within the indoor arena equipped with an OptiTrack motion capture system used to provide ground truth localization. This information is compared with the position estimates performed by the DMHE algorithms in order to evaluate the estimation accuracy. The estimation algorithms are using the measurements provided by low cost Logitech webcams. Figure 4.1 shows the scenario, i.e. five ground vehicles at their starting point, the finish point, the sensor camera network composed of 12 cameras and the road boundaries (yellow lines). As in the previous section, in the context of intruders' localization, the leading goal is to localize a formation of several mobile robots moving on a road, by performing the proposed Distributed Moving Horizon Estimation over a Sensor Network of low-cost cameras within a given communication topology, performing sporadic measurements. These vehicles are moving in a formation, along the road, controlled by a distributed algorithm using localization from the motion capture system. They are considered as non-cooperative vehicles for the localization problem performed by the Sensor Network.

The experiments are developed within the Robot Operating System (ROS) framework. Indeed, the distributed state estimation algorithm is deployed on several Raspberry PI, each one in charge of obtaining measurements from a single low-cost camera, of exchanging information among neighbors and of locally estimating the state of the Multi-Vehicle System. The cameras carry out vehicle detection thanks to the AprilTag algorithm (see [Wang and Olson, 2016]), which uses tags (known in size and pattern) placed on top of the vehicles to robustly and efficiently detect the positions of the vehicles. The Raspberry PI are not always able to detect the

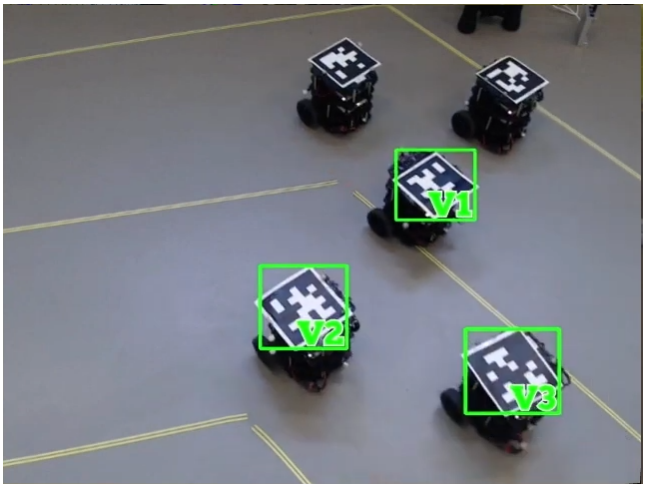


Figure 4.10: Three out of five vehicles detected by one camera.

vehicles even when these are within the field of view of the camera, this leads to have more sporadic measurements. For example, Figure 4.10 shows three out of five tags detected by one camera.

The experimentally collected data are further off-line re-processed and analyzed by adding artificial Gaussian noise to the measurements and changing the topology of the Sensor Network. We compare online and off-line results w.r.t. the DMHE algorithm of [Farina et al., 2010a].

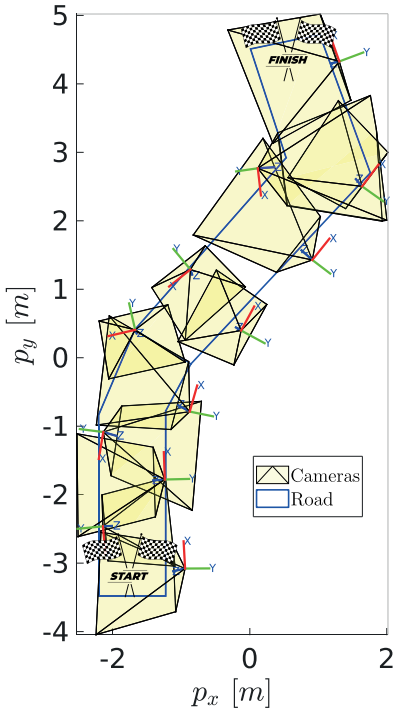
4.5.1 Experiments setup

The objective is to track a Multi-Vehicle System composed of $n_V = 5$ ground vehicles, one leader in the center of a square and four follower vehicles in the vertices of the square. The Multi-Vehicle System goes from the starting point $(-1.75, -3.25)$ m towards the final point $(0.75, 4)$ m driving within the road, clearly indicated in Figures 4.11a-4.13a, controlled by a leader-follower formation distributed control strategy. The control inputs of the intruders' vehicles are assumed to be unknown. The details of this control strategy are beyond the scope of this work and they are omitted here.

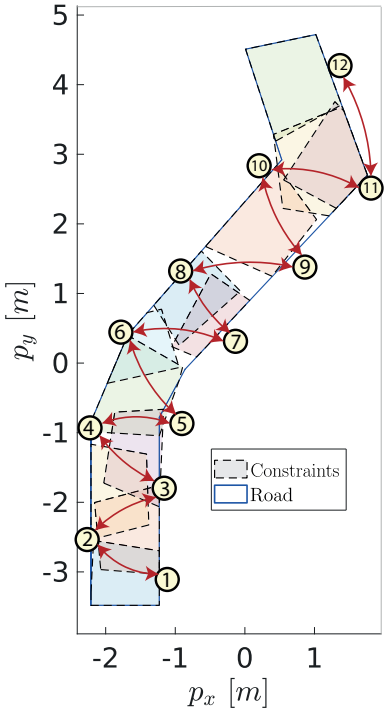
In order to analyze the performance of the proposed Distributed Moving Horizon Estimation, we designed three experimental scenarios, with a different number of sensors involved in the distributed state estimation and different poses for the cameras:

- **Scenario 1** (see Figure 4.11) uses 12 cameras for a maximum coverage area by their FoV. A video presentation is available at https://youtu.be/1CkSba2wVuI;

- **Scenario 2** (see Figure 4.12) uses 6 cameras for a maximum coverage area by their FoV;

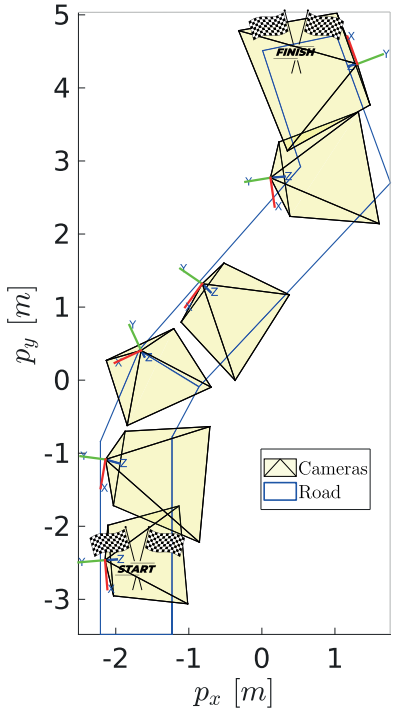


(a) FoVs real experiment.

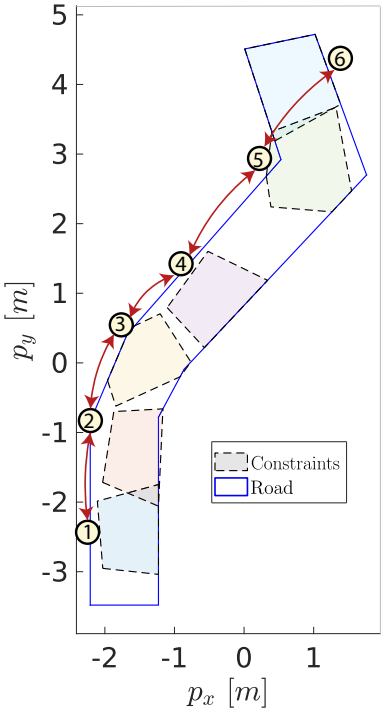


(b) FoV constraints real experiment.

Figure 4.11: Scenario 1, maximum coverage area with 12 sensors.

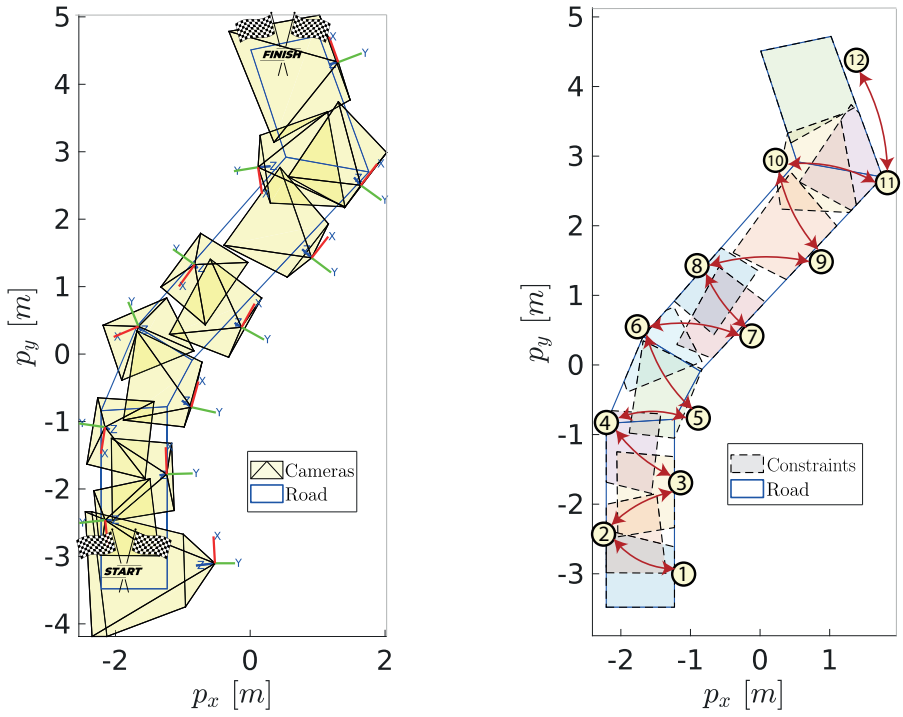


(a) FoVs real experiment.



(b) FoV constraints real experiment.

Figure 4.12: Scenario 2, maximum coverage with area 6 sensors.



(a) FoVs real experiment.

(b) FoV constraints real experiment.

Figure 4.13: Scenario 3, less coverage area with 12 sensors.

- **Scenario 3** (see Figure 4.13) uses 12 cameras but with a reduced coverage area by their FoV w.r.t. Scenario 1.

In Figure 4.11-Figure 4.13, the yellow polyhedra represent the FoV of the Logitech cameras (together with their reference frames), the solid blue lines define the road boundaries, the dashed colored polygons are the FoV constraints, and finally, the red arrows represent the communication links between the computing nodes (Raspberry PI) associated to the cameras. Notice that the poses of the 12 cameras used in Scenario 3 (except for the 12th sensor) are defined such that each camera can detect a maximum of 3 vehicles, i.e. their fields of view point only at half of the road (see Figure 4.13a).

In the DMHE optimization problem, each vehicle is modeled as single integrator, where the state vector ${}^{\nu}x = [{}^{\nu}p_x, {}^{\nu}p_y]^{\top} \in \mathbb{R}^2$ consists of planar position coordinates and the control input vector (i.e. the velocity components) is assumed unknown and considered as an exogenous input ${}^{\nu}w_t \in \mathbb{R}^2$, modeled as an uniformly distributed noise vector with covariance matrix $Q = I_2$.

Each camera provides position measurements of the vehicles in its own reference frame (indicated in Figure 4.11a-Figure 4.13a). Thus, in order to obtain position measurements in the common absolute reference frame used for the experiments and associated to the motion capture system, it is necessary to translate and rotate the measurements with a transformation matrix. To calculate such

a matrix, the poses of the cameras are necessary. However, these poses are not always available, or at least not precisely known, as it is the case in these experiments. Indeed, here we obtained the poses of the low-cost cameras using the OptiTrack motion capture system, which detects 3-4 markers glued on each cameras. The precision of these detections was falling on some areas of the arena, e.g. less observed areas such as corners, and there is a mismatch between the frame corresponding to the markers and the one associated to the optical axis of the camera (used by the AprilTag software). Thus, the resulting measurements are biased. Such an error in the transform matrix therefore results in some bias in the measurement translated in the global frame and provided to the estimators. Despite this, the robustness of the proposed DMHE to this additional source of uncertainty (i.e. sensor biases and noise) is further investigated by validating the usefulness of *a priori* known FoV constraints.

The estimators run with a sampling time $T_s = 0.5$ s and a horizon length $N = 3$. The initial values of the algorithms have been set as ${}^{\nu}\hat{x}_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top$, $\forall \nu$, $\Pi_0 = 10^5 I_2$. The measurements noises ${}^{\nu}v_t^i$ are assumed to be white normally distributed noises, with zero mean and covariance matrix $R^i = I_2$.

The optimization problem was solved by the quadratic programming solver from [Goldfarb and Idnani, 1983] implemented in Python on twelve Raspberry PI, each one associated to a single camera. The considered performance indexes are the computation time τ needed by the solver to estimate the positions of the Multi-Vehicle System, and the Root Mean Square Error (RMSE) are computed as follows:

$$\text{RMSE}_t = \frac{1}{n_S} \sum_{i \in \mathcal{N}} \left\| x_t - \hat{x}_{t|t}^i \right\|,$$

both averaged among all the sensors. The RMSE should remain small for good performance.

As for the simulated case, we compare the proposed Distributed Moving Horizon Estimation without FoV constraints (denoted by DMHE) and with FoV constraints[3] (denoted by DMHE$^{\mathcal{S}}$). We also compare the proposed DMHE approach with the algorithm in [Farina et al., 2010a], denoted hereafter by DMHE$_F$, for the case without FoV constraints, and by DMHE$_F^{\mathcal{S}}$ when considering the FoV constraints (4.5).

4.5.2 Experimental results

It is essential to highlight that between experiments some parameters are not exactly repeatable (e.g. unpredictable lack of measurements at time instants *a priori* unknown, different initial timing synchronization among sensor neighborhoods, initial positions of the vehicles, etc.) and that the qualitative evaluations may therefore suffer from some bias in the comparison between two experimental runs. This is why all the measurement data have also been recorded to addi-

[3]The constraints are added as in (4.5).

tionally perform offline evaluation from the same data. The video available at https://youtu.be/1CkSba2wVuI shows the online experiment of Scenario 1 on using DMHE$^{\mathcal{S}}$ and offers additional details.

Figure 4.14 illustrates the averaged computation time τ among all the sensors. The proposed DMHE$^{\mathcal{S}}$ (dotted lines) shows half of the time needed by DMHE$^{\mathcal{S}}_{F}$ (dashed lines) in all the scenarios. It is a consequence of replacing the system model with unknown input by a Luenberger observer (pre-estimation strategy) in (4.15). Indeed, the model with a Luenberger pre-estimation involves fewer optimization parameters. These results are coherent with the one obtained in the Gazebo simulations.

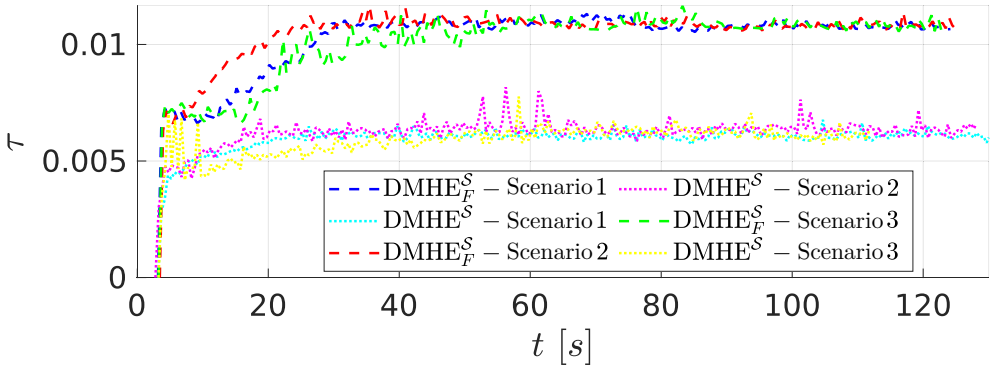


Figure 4.14: Computation time τ of all algorithms with FoV constraints during the real experiments.

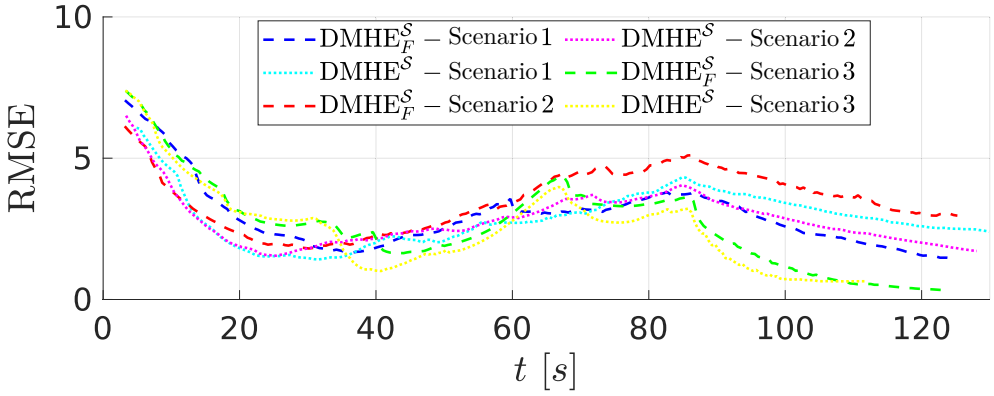
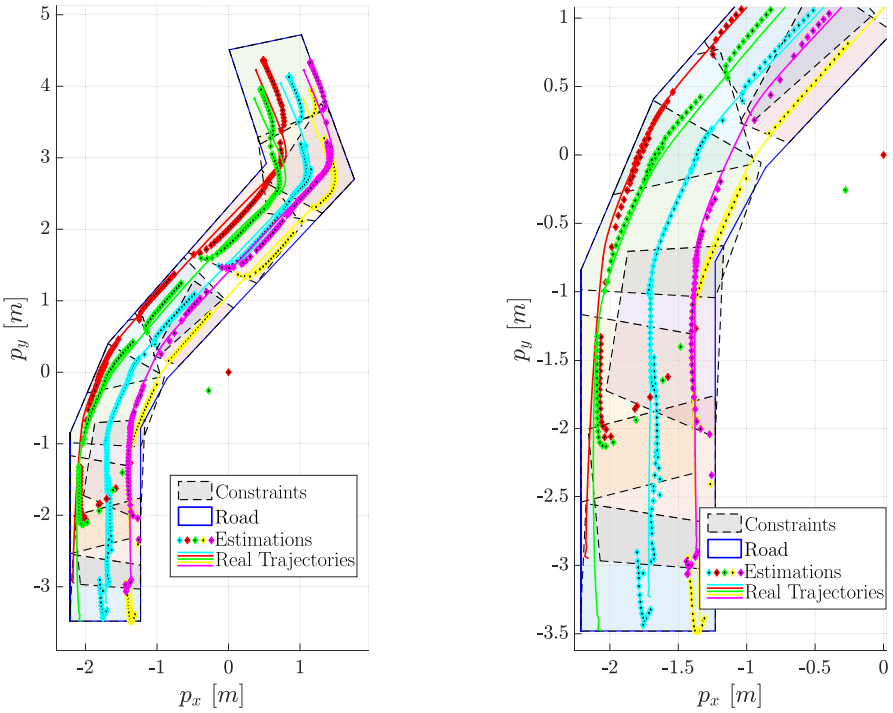


Figure 4.15: RMSE of all algorithms with FoV constraints during the real experiments.

The RMSEs averaged among all the sensors for all scenarios and algorithms with FoV constraints are shown in Figure 4.15. Regarding Scenario 1, DMHE$^{\mathcal{S}}$ (cyan dotted line) is better than DMHE$^{\mathcal{S}}_{F}$ (blue dashed line) only until *cerca* $t =$ 70 s. For Scenarios 2 and 3, DMHE$^{\mathcal{S}}$ has similar or better performance w.r.t. DMHE$^{\mathcal{S}}_{F}$. As explained before, a rigorous comparison is hard due to unrepeatable conditions, that is why we performed more rigorous comparisons in the next section.



(a) Entire scenario.

(b) Zoom.

Figure 4.16: Scenario 1: Planar trajectory of the vehicles and estimations by DMHE$^{\mathcal{S}}$ using 12 cameras (real experiment).

Figure 4.16 illustrates the results of the real experiment of Scenario 1. In particular, it shows the trajectories estimation (rhombus) by the active sensors operating with DMHE$^{\mathcal{S}}$ and the actual trajectories of the vehicles (lines) by the motion capture system of Scenario 1 on the left hand side and its zoom on the right hand side. Different colors highlight the vehicles: cyan, red, green, yellow, and magenta refer to vehicles 1, 2, 3, 4, and 5, respectively. Moreover, Figure 4.16 shows the road boundaries (blue lines) as well as the FoV constraints (dashed polygones) described as in (4.5). Notice that the starting position is about $p_y = -3.5$ m for the Multi-Vehicle System, which results in a large discrepancy w.r.t. ${}^{\nu}\hat{x}_0$ considered at the origin. As shown in Figure 4.16b, at the beginning only vehicles 1 (cyan rhombus), 4 (yellow rhombus), and 5 (magenta rhombus) are detected by the first camera (i.e. they are within the blue dashed polygone, at the bottom). Indeed, camera 1 does not detect vehicles 2 and 3 (i.e. no red or green rhombuses appearing inside this blue dashed polygone). Vehicles 2 and 3 start being detected later on. The rhombus outside the road are due to the initial state estimates considered by the observers, which are chosen to be at the origin of the plan. Notice that the biased sensors data can result in some bias in the estimations (see the difference between the real trajectories, which are represented in solid lines, and estimates, which are represented as rhombi, around the arrival position

in the light green polygone on the top in Figure 4.16a); however, the intensity of measurement noise is not perceptible (i.e. smooth estimations). This is because the AprilTag library (see [Wang and Olson, 2016]) provides accurate position of the tags mounted on the robots, by visual reconstruction using calibration information of the cameras and tags with known sizes and patterns. This has been one of the motivations for introducing additional measurement noise on the experimental data to validate and further analyze the algorithms' performance when they operate in a more realistic environment.

4.5.3 Performance evaluation

The ROS framework offered the opportunity to record data for the three scenarios explained above. This data includes time synchronization among measurements and other useful information. Thus, it allows us to replay these data in order to replicate the same experiments but with other algorithms and/or by changing parameters. We also added *a posteriori* artificial Gaussian noise in the measurements, with a variance of 0.15 m^2, and in the positions of the cameras (biased sensors), with a variance of 0.1 m^2 to make the scenarios more realistic, e.g. when using a low-cost Sensor Network with video cameras that would run computer vision algorithms for visual detection and position reconstruction of vehicles without tags (e.g. area monitoring scenario).

The first aspect we investigate for Scenario 1 is the topology of the Sensor Network, i.e. how the sensors are connected to each other. For this reason, we define the *radius communication link*.

Definition 4.2 **Radius communication link**

The radius communication link *is the number of nodes reachable in communication by each sensor.*

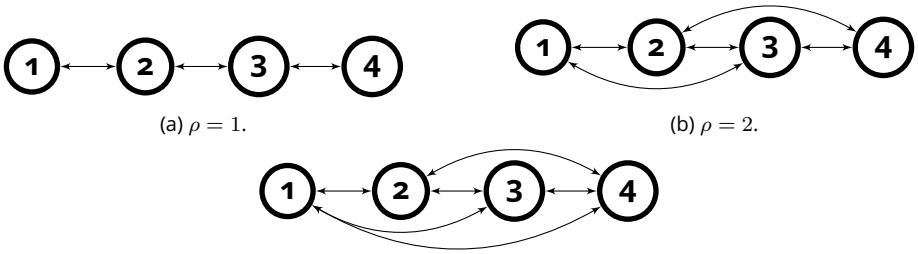
Thus, increasing or decreasing this radius can change the topology of the network, i.e. the edges of the graph. Then, we define $d(i,j) = |i-j|$ as the *distance*, in terms of node numbering, between nodes i and j.

Definition 4.3 **Neighborhood based on radius ρ**

Given a radius ρ, the neighborhood of sensor i is $\mathcal{N}^i = \{j \in \mathcal{N} : d(i,j) \leqslant \rho\}$, i.e. the set of nodes $j \in \mathcal{N}$ for which there exists a path at maximum distant ρ nodes from sensor i.

For example, Figure 4.17 shows a Sensor Network composed of four nodes and its possible different communication topologies by varying the *radius communication link* ρ from 1 to 3.

In the next sections, the algorithm performance on the three considered scenarios will be further analyzed by considering noisy measurements and the influence of the communication topology.



(a) $\rho = 1$.

(b) $\rho = 2$.

(c) $\rho = 3$ (complete graph).

Figure 4.17: Examples of communication topologies of a Sensor Network composed of 4 nodes.

Scenario 1

The aim of this scenario (see Figure 4.11) is to compare the accuracy of the algorithms, with and without FoV constraints, changing the topology of the Sensor Network by varying the radius communication link $\rho = \{1, 3, 6, 9, 12\}$. Notice that when the radius is 12, the graph is complete, i.e. each node is connected to anyone else (see Definition 2.6). Moreover, we show the effects on using or not the FoV constraints as in (4.5).

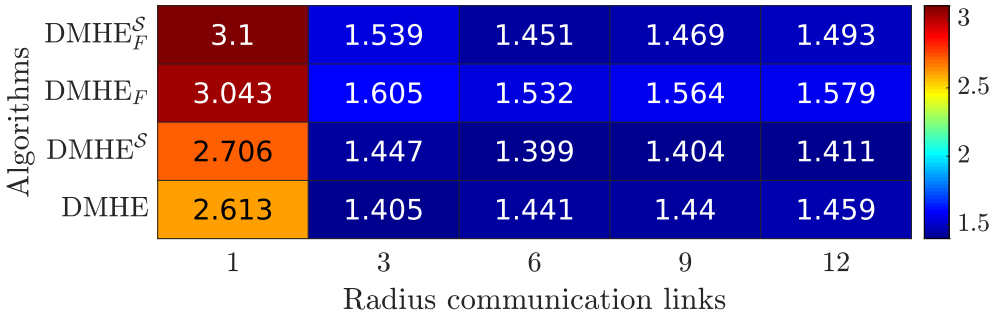


Figure 4.18: Scenario 1: RMSE averaged among the observers and time (all sensors).

The first column of Figure 4.18 shows the RMSE for the four implemented algorithms with $\rho = 1$. In this case, the smallest value 2.613 (and thus the best accuracy) is obtained with the proposed DHME algorithm without FoV constraints. Figure 4.18 also shows that starting from $\rho = 3$, the RMSEs are similar to each other. It means that, for this number of sensors, a graph with a radius communication link equal to 3 or higher performs as good as a complete graph. Moreover, the accuracy is always better for the proposed DMHE (lines 3 and 4, respectively) compared with DMHE$_F$ (lines 1 and 2, respectively), for both cases, i.e. with and without FoV constraints.

To check how the FoV constraints $\mathcal{S}^i$ influence the accuracy of the estimations, we have to look at the RMSE of the active sensors only since the FoV constraints are used in the local optimization problem only when the camera is detecting a

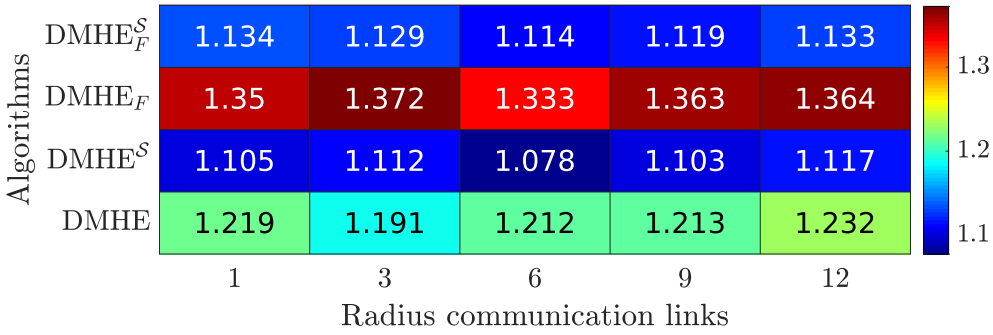


Figure 4.19: Scenario 1: RMSE averaged among the observers and time (active sensors only).

vehicle. In Figure 4.19, the RMSE of FoV constrained algorithms is always better than their respective version without FoV constraints. Indeed, it can be seen that the values on line 1 are always lower than the values on line 2, while the values on line 3 are always lower than the values on line 4.

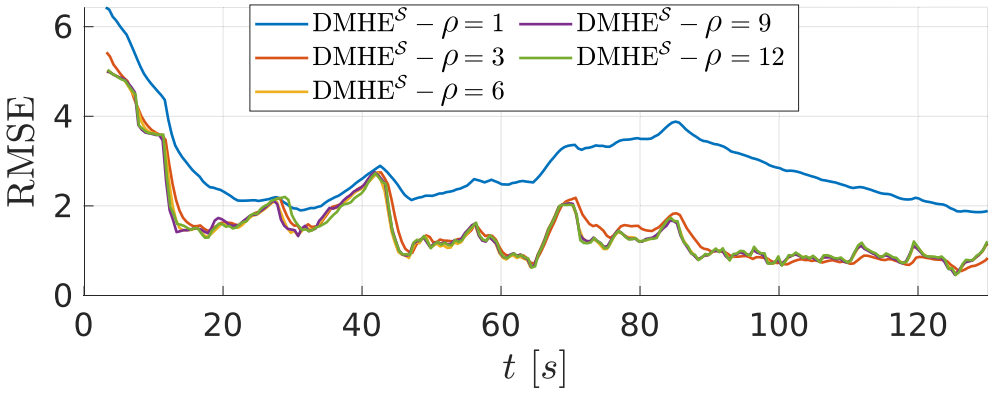
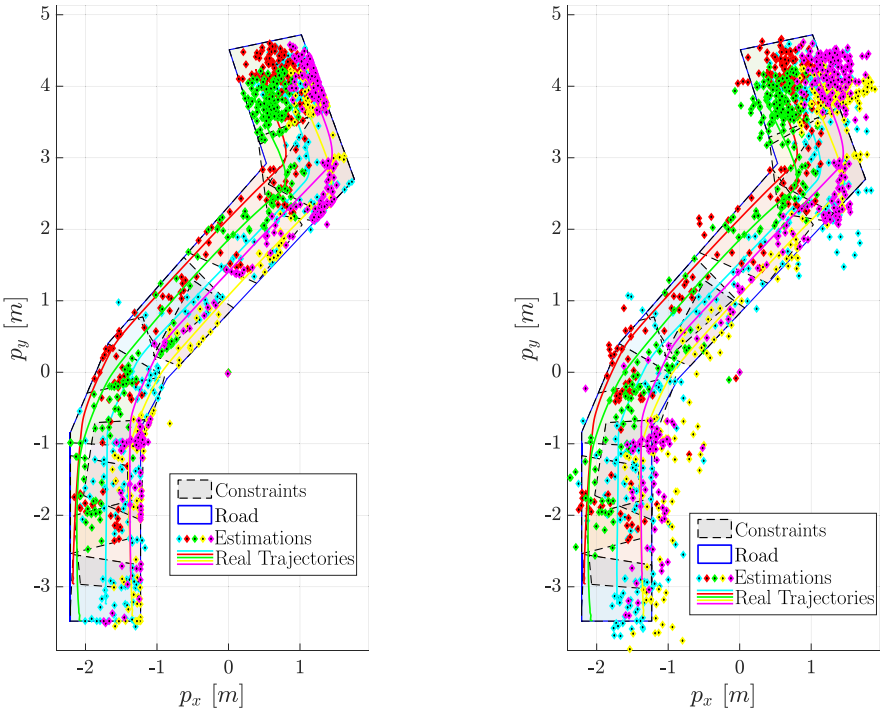


Figure 4.20: Scenario 1: RMSE of DMHE$^{\mathcal{S}}$ over time for all communication links radius (all sensors).

We have seen that the radius ρ has a specific effect on the estimation error. In Figure 4.20, we can see the RMSE overtime for the solely DMHE$^{\mathcal{S}}$, for different values of the radius ρ. It is worth noticing that $\rho > 3$ (yellow, purple and green curves) leads to having better RMSE than $\rho = 3$ (red curve) until the vehicles stop, around $t = 88$ s.

Figure 4.21 and its zoom (Figure 4.22) illustrate the planar trajectories of the vehicles (solid lines) by the motion capture system and their respective estimations (rhombus) by the active sensors. Different colors highlight the vehicles: cyan, red, green, yellow, and magenta refer to vehicles 1, 2, 3, 4, and 5, respectively. Moreover, this figure shows the road boundaries (blue line) as well as FoV constraints (dashed polygones), as in (4.5). In these figures, it is possible to see how the FoV constraints improve the estimation accuracy since the estimates are lying within



(a) DMHE with FoV constraints.

(b) DMHE without FoV constraints.

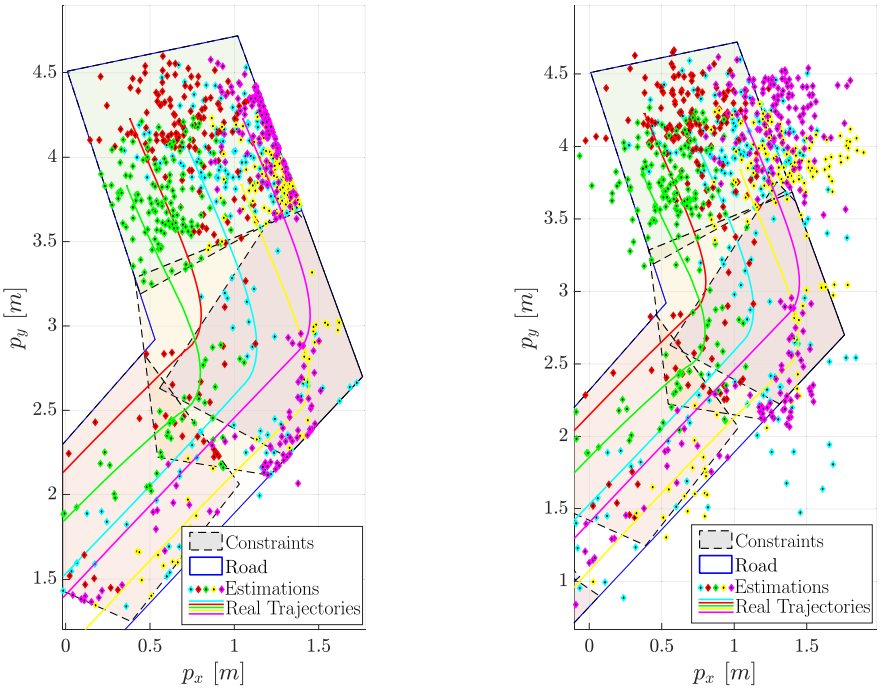
Figure 4.21: Scenario 1: Planar trajectory of the vehicles and estimations with FoV constraints (left) and without (right).

these constraints and are hence consistent with the locations of the vehicles within the road boundaries. This is more evident in the zoom proposed in Figure 4.22 which shows the final part of the Scenario 1. In particular, Figure 4.22a shows the results by using the proposed DMHE with FoV constraints (DMHE$^{\mathcal{S}}$) and Figure 4.22b shows the results by using the proposed DMHE without FoV constraints. Notice that in Figure 4.22a only two points are outside the FoV constraints (due to mismatch among the considered detection instants by low-cost cameras, Raspberry PI, ROS), while Figure 4.22b shows a lot of estimates outside the FoV constraints. Therefore, considering FoV constraints allows a better accuracy for the estimation.

Scenario 2

This scenario (see Figure 4.12) aims to evaluate the algorithms in terms of accuracy while using fewer sensors and, at the same time, to diversify the communication topology. Additionally, it highlights the effects on the convergence of the estimates when using a different number of sensors in such distributed algorithms compared to Scenario 1. In this case, we used half of the sensors, which led to more sporadic measurements since the total covered area by cameras is much smaller (see Figure 4.12) compared to Scenario 1.

Figure 4.23 shows that for different values of the radius communication links,



(a) DMHE with FoV constraints.

(b) DMHE without FoV constraints.

Figure 4.22: Scenario 1: Part of the planar trajectory of the vehicles and estimations with FoV constraints (left) and without (right). (Zoom of Figure 4.21)

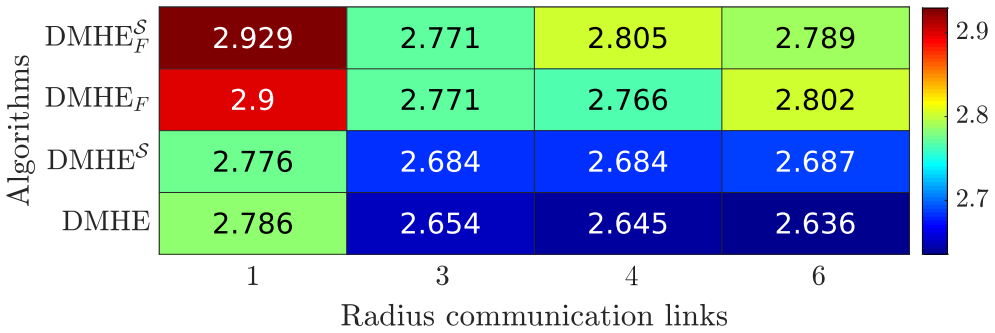


Figure 4.23: Scenario 2: RMSE averaged among the observers and time (all sensors).

i.e. $\rho = \{1, 3, 4, 6\}$, the RMSEs are not so different from each other, and probably a radius $\rho = 2$ would have been the optimum trade-off between the number of communication links and the accuracy of the estimates. Moreover, comparing the RMSE values in Figure 4.18 (Scenario 1) and in Figure 4.23 (Scenario 2) it is evident, as expected, that having less covered area by cameras leads to less accuracy, when $\rho > 1$. Although this is not always the case, as illustrated in the first column of these figures (i.e. for $\rho = 1$), the RMSE values are comparable among the same algorithms. Indeed, even though Scenario 1 has a larger covered area, it also has more sensors, which results in more consensus communication steps needed for the convergence of all the observers, accentuated by the fact that $\rho = 1$. For this reason, even with fewer communication links than in Scenario 1, the convergence is faster as we can clearly see by comparing the curves in Figures 4.20 and 4.24 around time $t = 88$ s.

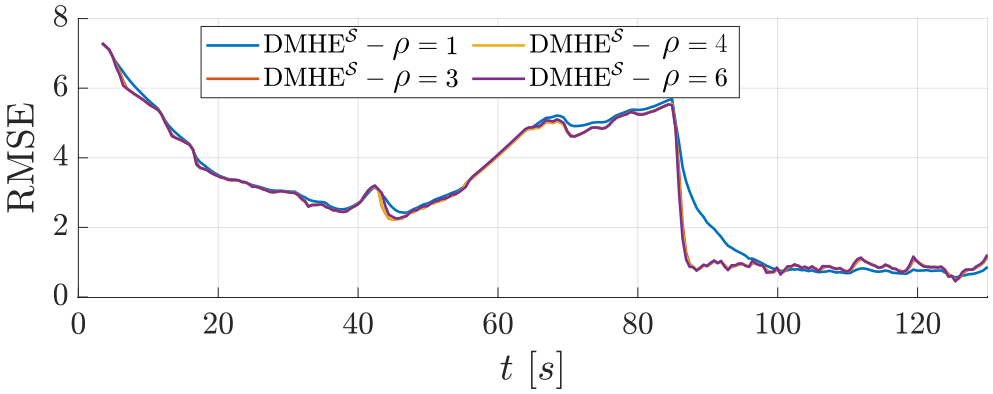
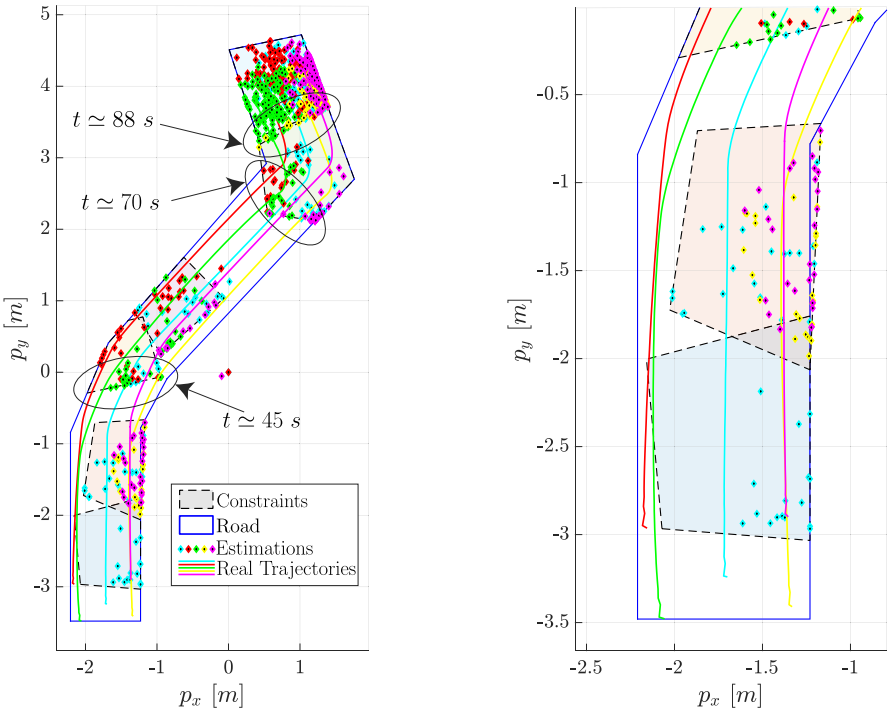


Figure 4.24: Scenario 2: RMSE of DMHE$^{\mathcal{S}}$ over time with a different communication link radius.

Furthermore, Figure 4.24 shows that the RMSE value obtained for $\rho > 1$ (red, yellow and purple curves) is better than the one with $\rho = 1$ (blue curve) only until the vehicles stop, around $t = 88$ s. Thus, in the end, the five vehicles are only detected by sensor 6, which is the only active sensor, thus the only one using FoV constraints as in (4.5). Moreover, it sends its measurements to its neighbors, which contributes to the global state estimation convergence without using the FoV constraints. Hence, the more neighbors sensor 6 has, the more they contribute to the global estimation without FoV constraints, i.e. emphasizing measurements noise and biased sensor data.

Figure 4.25a and its zoom (right hand side of Figure 4.25b) show the exact planar trajectories of the vehicles (solid lines) by the motion capture system and their respective estimates (rhombus) by the low cost active sensors. Different colors highlight the vehicles: cyan, red, green, yellow, and magenta refer to vehicles 1, 2, 3, 4, and 5, respectively. This figure also shows the road boundaries (blue line) as well as the FoV constraints (dashed polygones), as considered in (4.5). It is



(a) DMHE with FoV constraints.

(b) DMHE without FoV constraints.

Figure 4.25: Scenario 2: Planar trajectory of the vehicles and estimates with FoV constraints (left) and zoom (right).

worth noticing that in Scenario 2 most of the time the vehicles are not detected (see Figure 4.25a). Indeed, vehicles 2 (red rhombus) and 3 (green rhombus) began to be detected at around $t \simeq 45$ s (see the zoom Figure 4.25b where the green and red rhombuses appear only inside the light orange dashed polygone, on the top). Moreover, after some moments when no vehicle is detected, vehicles 1, 2, 3 and 5 are detected again around $t \simeq 70$ s (see the corresponding rhombuses in Figure 4.25a). Only nearby the arrival point all the vehicles are detected at once, *cerca* $t \simeq 88$ s (see Figure 4.25a). In addition, Figure 4.24 shows that around the time instants $t \simeq 45$ s, $t \simeq 70$ s, and $t \simeq 88$ s when the vehicles are detected again, the RMSE decreases and drops significantly especially around $t \simeq 88$ s since all the vehicles are detected (thus leading to a very small value for the RMSE).

Scenario 3

The goal of the last scenario (see Figure 4.13) is to evaluate the performance of the proposed DMHE when the poses of the cameras are such that a single camera cannot detect all the vehicles at once, except sensor 12. Moreover, 10 trials have been run to show the robustness against different realization of measurement noise and bias. The provided results are averaged among the 10 trials (only for Scenario 3) and compared with the previous scenarios (notice that only one trial is considered

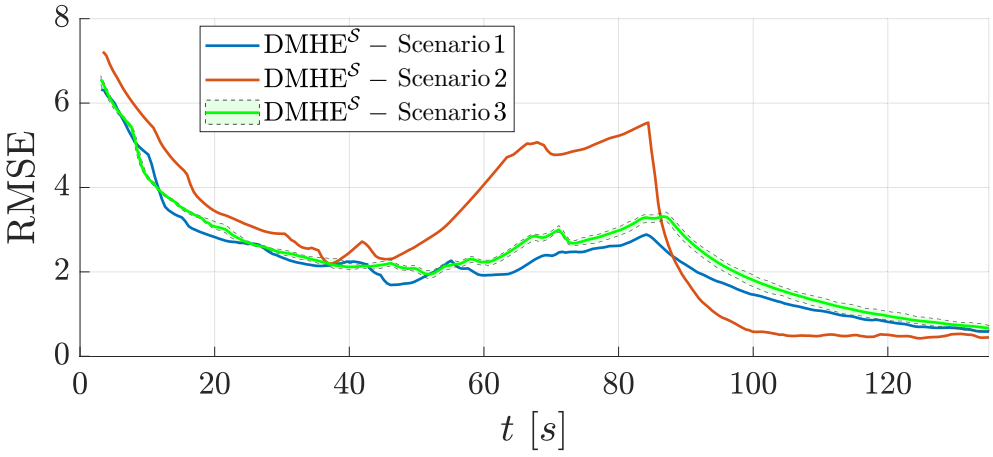


Figure 4.26: Comparison between the RMSE of the DMHE$^{\mathcal{S}}$ for the 3 scenarios for $\rho = 1$, with an average among 10 trials for Scenario 3.

for Scenarios 1 and 2). The radius communication link is $\rho = 1$.

Figure 4.26 illustrates the RMSEs of all scenarios considering the proposed DMHE algorithm with FoV constraints DMHE$^{\mathcal{S}}$. The green curve refers to the RMSE of the third scenario which also shows the bounds of the minimum and maximum RMSE of each trial (dashed lines). As expected, Scenario 3 (green curve) offers better accuracy than Scenario 2 (red curve) until the vehicles stop, since Scenario 3 has double of the sensors w.r.t. Scenario 2, thus leading to a larger covered area. Moreover, Scenario 3 (green curve) has always worse accuracy than Scenario 1 (blue curve), since Scenario 3 has the same number of sensors as Scenario 1 but less covered area. It can also be noticed that around $t \simeq 88$ s, the RMSE drops since all the vehicles are detected.

4.6 Conclusion

This chapter proposed a Distributed Moving Horizon Estimation (DMHE) algorithm for localizing a Multi-Vehicle System over a static sensor camera network with sporadic measurements, i.e. available at time instants *a priori* unknown.

The proposed approach, which takes into account measurement constraints, has been implemented first in a realistic distributed way in the Robot Operating System (ROS) middleware with a Gazebo simulation environment, then on a real Sensor Network composed of several low-cost cameras, each of them attached to a Raspberry PI for distributed implementation of the algorithms (using ROS middleware) and network communication. The objective was to track a fleet of autonomous vehicles (ground robots) moving in an urban-like environment with FoV constraints. Three different experimental scenarios evaluated distinct aspects of the proposed algorithm, which differ in the number of sensors involved, the topology of the communication network, and the covered area by the cameras'

fields of view.

The computation time of the proposed DMHE has been decreased by a factor two w.r.t. the time needed by the one of [Farina et al., 2010a]. This result is obtained thanks to the pre-estimation observer included in the optimization problem that replaces the need to estimate the sequence of unknown inputs of the model over the estimation window and therefore leads to fewer optimization parameters. Taking advantage of constraint handling in online optimization required by MHE, the proposed algorithm exploits *a priori* information as environment constraints (such as the road boundaries) to better estimate the state of the system. Moreover, the proposed DMHE formulation can deal with sporadic measurements thanks to the time-varying binary parameters embedded into the algorithm. Finally, it improves the accuracy of the estimation by utilizing an observability rank-based method to adjust the components of the consensus matrix associated with the graph of the Sensor Network. This particular aspect is especially well suited to cases with sporadic measurements, as the one considered in this chapter.

Several experiments have been realized, and collected data have been re-executed off-line in order to make rigorous performance analysis and comparison of the proposed DMHE with one of the reference DMHE algorithms in the open literature [Farina et al., 2010a] and analyzing the effect of changing some properties of the application. Indeed, results have been analyzed in terms of accuracy by changing the number of sensors composing the network and the communication topology among neighbor nodes.

The proposed algorithm could be further extended by taking into account active and non active sensors when computing the consensus terms. Furthermore, in a context of fault detection, the possible sensor faults can be treated in the same manner as non active sensors, extending the application field of the current chapter.

The proposed algorithms have been presented and submitted in:

- **A. Venturino**, S. Bertrand, C. Stoica Maniu, T. Alamo, and E. F. Camacho. Multi-vehicle system localization by distributed moving horizon estimation over a sensor network with sporadic measurements. In *6th IEEE Conference on Control Technology and Applications (CCTA)*, Trieste, Italy, 23-25 August, 2022.

- **A. Venturino**, C. Stoica Maniu, S. Bertrand, T. Alamo, and E. F. Camacho. Multi-vehicle localization by distributed MHE over a sensor network with sporadic measurements: further developments and experimental results. Submitted to the *Control Engineering Practice Journal*, 2022.

5 - Conclusion and perspectives

5.1 Conclusion

The present PhD thesis proposed new distributed state estimation algorithms for discrete-time linear time-invariant systems over a low-cost Sensor Network. These estimators are based on the Moving Horizon Estimation (MHE) paradigm and consensus strategies, allowing them to take into account constraints and achieve the convergence of the distributed estimates. The contributions of this manuscript are threefold. Firstly, this PhD thesis proposed a Distributed MHE (DMHE) technique that reduces the computation time needed for the estimation w.r.t. existing results in the open literature. Then, an extended DMHE technique that exploits observability properties for the consensus strategy to enhance the estimation accuracy is proposed. Secondly, it studied how to take advantage of the Moving Horizon strategy in the context of distributed algorithms, and proposes a DMHE technique equipped with an information diffusion mechanism that improves the convergence time of the estimation. Finally, a DMHE algorithm is provided to deal with the realistic case study of localizing a Multi-Vehicle System over a Sensor Network. This algorithm is implemented within the Gazebo environment and on a real hardware setup, developed during the PhD, using the Robot Operating System (ROS) framework.

The first disadvantage of MHE is the computational aspect since it involves an optimization problem to be solved online. For this reason, Chapter 2 proposed centralized and distributed MHE algorithms with reduced computation load. Indeed, introducing a pre-estimation observer in the formulation problem led to replace the sequence of unknown inputs of the system to be estimated and thus to fewer optimization parameters than the original formulation of [Farina et al., 2010a]. Thus reducing significantly the computation time. The proposed distributed algorithm is able to converge even under weak observability conditions due to the embodied consensus terms within the formulation [Farina et al., 2010a]. In addition, an extension that exploits regional observability properties to compute the consensus terms is also proposed. This extension improves the algorithm for two reasons. First, the stochastic matrix used in the consensus strategy is now calculated based on only locally available information. Thus, it allows the use of the algorithm in a fully distributed manner. Second, relying on observability properties speeds up the convergence of the estimates among sensors since the sensors with more information are weighted more in the consensus strategy.

The research topic of this manuscript involved designing MHE-based distributed state estimation algorithms over Sensor Network. In addition, these algorithms have to deal with weak observability conditions when using sensors that make the local systems unobservable, as anticipated in Chapter 2. In order to benefit from

the advantages offered by the MHE paradigm and to mitigate these observability issues, Chapter 3 proposed a new DMHE technique which exploits the sliding time window of data within the MHE formulation. In particular, each sensor can use measurements from non-direct neighbors thanks to the developed information diffusion mechanism. This mechanism enhanced the convergence time and the accuracy of the state estimates. We named the proposed algorithm ℓ-step Neighborhood DMHE. It uses the algorithm of [Battistelli, 2018] as theoretical basis.

Further on, Chapter 4 proposed a novel DMHE algorithm which is able to localize a Multi-Vehicle System over a static sensor camera network with sporadic measurements, i.e. available at time instants *a priori* unknown. In order to take advantage of the simplicity of the implementation, this algorithm is based on the results proposed in Chapter 2, which have been adapted to deal with sporadic measurements by use of additional time-varying binary parameters embedded into the algorithm. Since the fleet of autonomous vehicles (ground robots) moves in a known urban-like environment and the Field of View (FoV) of each camera is known, the proposed algorithm exploits these information using them as constraints in the local optimization problems. This led to better estimation accuracy. The proposed approach has been implemented first in a realistic simulation using the Robot Operating System (ROS) framework and the Gazebo simulator, then on a real hardware setup, i.e. a Sensor Network composed of several low-cost cameras, each of them connected to a Raspberry PI, and a Multi-Vehicle System composed of five ground robots (Turtlebot3). Chapter 4 evaluated the proposed algorithm by showing the results obtained by three different experimental scenarios. They differ in the number of sensors involved, the topology of the communication network, and the covered area by the fields of view of the cameras. The collected data during the experiments have been re-executed off-line in order to make rigorous performance analysis and comparison of the proposed DMHE with [Farina et al., 2010a], in terms of accuracy and computation time, when changing the number of sensors composing the network and the communication topology among neighbor nodes.

5.2 Perspectives

Further improvements and directions are discussed below.

First of all, the proposed algorithms include a Luenberger pre-estimation observer which has been computed by considering stability guarantees. It might be possible to improve the accuracy of the estimates by using different criteria for computing the pre-estimation gain. There are many well-known options in literature such as $\mathcal{H}_2$, $\mathcal{H}_\infty$, combination of them, etc. (see [Duan and Yu, 2013, p. 293]). Moreover, it is possible to design a time-varying gain or Kalman-based one as pre-estimation observer.

Regarding the computation time needed by the sensors to obtain the state

estimation at each sampling time, it might be possible to further reduce it. Indeed, it could be possible to design formulation of the optimization problem in order to solve them with *ad hoc* solvers. For this purpose, the Alternating Direction Method of Multipliers is gaining particular attention in literature, see [Boyd et al., 2011].

Chapter 3 proposed the ℓ-step Neighborhood DMHE algorithm which involves an information diffusion mechanism based on a time-sliding batch of data of fixed size ℓ. An improvement in terms of communication burden could be to have time-varying ℓ depending on the necessity of the current situation, such as convergence of the estimates among neighbors, all sensors inactive within the neighborhood, etc.

Chapter 2 proposed an observability rank-based weight technique that enhances the convergence of the estimates. It has been adapted in Chapter 4 to deal with sporadic measurements. The results showed that this method could be improved. In particular, Section 4.5.3 (scenario 2) showed how the increasing number of inactive neighbors (i.e. not using FoV constraints) can degrade the estimates by emphasizing measurements noise and biased sensor data. To avoid this, a solution could have been to directly consider whether the sensor is active or not in the observability rank-based weight technique (Section 4.3.5). It might lead to better estimation accuracy. In addition, this technique can be further improved by considering other properties in its calculation than the solely observability. Indeed, the current calculation does not take into account how many times the sensor was active within the horizon length. Thus taking into account this aspect might lead to better performance.

Furthermore, in a context of fault detection, the possible sensor faults can be treated in the same manner as non active sensors, extending the application field of the proposed algorithm.

The assumption of static communication topology can be limited when the application, for example, involves cameras on-board of vehicles or drones. Indeed, using mobile sensors would lead to a dynamic (i.e. time-varying) communication topology as the agents communicate and exchange information while moving. Thus, deep insights are necessary to enable the use of the proposed algorithms when the topology of the network cannot stay *a priori* fixed.

Within the case of cooperative vehicles and Sensor Network, it might be possible to design the estimation algorithm combined with a control law. Since the MHE involves a receding horizon as Model Predictive Control, it could be possible to design a distributed algorithm for estimation and control. Alternatively, if the computation load is prohibitive for some applications, the DMHE formulation can be combined with a Command Governor to have a single optimization problem.

This present manuscript proposed distributed algorithms for linear systems. Further improvements could be designing similar algorithms for non-linear systems.

Stability and robustness analysis of the proposed algorithms are currently under investigation and will be submitted to a journal paper.

Finally, experimentation of the ℓ-step DMHE procedure on a real hardware setup is also one further development of this PhD thesis.

Bibliography

[Albert and Imsland, 2018] Albert, A. and Imsland, L. (2018). Survey: mobile sensor networks for target searching and tracking. *Cyber-Physical Systems*, 4(2):57–98.

[Alessandri et al., 2008] Alessandri, A., Baglietto, M., and Battistelli, G. (2008). Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes. *Automatica*, 44(7):1753–1765.

[Alessandri et al., 2011] Alessandri, A., Baglietto, M., Battistelli, G., and Gaggero, M. (2011). Moving-horizon state estimation for nonlinear systems using neural networks. *IEEE Transactions on Neural Networks*, 22(5):768–780.

[Battistelli, 2018] Battistelli, G. (2018). Distributed moving-horizon estimation with arrival-cost consensus. *IEEE Transactions on Automatic Control*, 64(8):3316–3323.

[Battistelli et al., 2012] Battistelli, G., Benavoli, A., and Chisci, L. (2012). Data-driven communication for state estimation with sensor networks. *Automatica*, 48(5):926–935.

[Bertrand et al., 2020] Bertrand, S., Sarras, I., Eudes, A., and Marzat, J. (2020). Voronoi-based geometric distributed fleet control of a multi-robot system. In *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 85–91.

[Bono et al., 2019] Bono, A., Fedele, G., and Franzè, G. (2019). A distributed model predictive control strategy for vehicle teams in uncertain narrowed environments. In *24th International Conference on Emerging Technologies and Factory Automation*, pages 927–932.

[Boyd et al., 2011] Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122.

[Brulin et al., 2009] Brulin, D., Courtial, E., and Allibert, G. (2009). Visual receding horizon estimation for human presence detection. In *IEEE International Conference on Robotics and Automation (ICRA 2009)*.

[Cărbunar et al., 2006] Cărbunar, B., Grama, A., Vitek, J., and Cărbunar, O. (2006). Redundancy and coverage detection in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 2(1):94–128.

[Carli et al., 2008] Carli, R., Chiuso, A., Schenato, L., and Zampieri, S. (2008). Distributed Kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in communications*, 26(4):622.

[Chevet et al., 2018] Chevet, T., Maniu, C. S., Vlad, C., and Zhang, Y. (2018). Voronoi-based UAVs formation deployment and reconfiguration using MPC techniques. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 9–14.

[Conte et al., 2016] Conte, C., Jones, C. N., Morari, M., and Zeilinger, M. N. (2016). Distributed synthesis and stability of cooperative distributed model predictive control for linear systems. *Automatica*, 69:117–125.

[D'Amato et al., 2022] D'Amato, E., Notaro, I., Iodice, B., Panico, G., and Blasi, L. (2022). Decentralized moving horizon estimation for a fleet of UAVs. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 989–996.

[Ding et al., 2019] Ding, D., Han, Q.-L., Wang, Z., and Ge, X. (2019). A survey on model-based distributed control and filtering for industrial cyber-physical systems. *IEEE Transactions on Industrial Informatics*, 15(5):2483–2499.

[Ding et al., 2012] Ding, D., Wang, Z., Dong, H., and Shu, H. (2012). Distributed H∞ state estimation with stochastic parameters and nonlinearities through sensor networks: the finite-horizon case. *Automatica*, 48(8):1575–1585.

[Dong et al., 2022] Dong, X., Battistelli, G., Chisci, L., and Cai, Y. (2022). Consensus variational bayesian moving horizon estimation for distributed sensor networks with unknown noise covariances. *Signal Processing*, 198:108571.

[Duan and Yu, 2013] Duan, G.-R. and Yu, H.-H. (2013). *LMIs in control systems: analysis, design and applications*. CRC press.

[D'Alfonso et al., 2015] D'Alfonso, L., Lucia, W., Muraca, P., and Pugliese, P. (2015). Mobile robot localization via EKF and UKF: A comparison based on real data. *Robotics and Autonomous Systems*, 74:122–127.

[Famularo et al., 2022] Famularo, D., Franzè, G., and Tedesco, F. (2022). Fault detection and isolation for uncertain linear systems: A robust moving horizon estimation scheme using LMIs. *Journal of the Franklin Institute*, 359(2):1692–1712.

[Farina et al., 2010a] Farina, M., Ferrari-Trecate, G., and Scattolini, R. (2010a). Distributed moving horizon estimation for linear constrained systems. *IEEE Transactions on Automatic Control*, 55(11):2462–2475.

[Farina et al., 2010b] Farina, M., Ferrari-Trecate, G., and Scattolini, R. (2010b). State estimation for large-scale partitioned systems: a moving horizon approach. In *Proceedings of the 2010 American Control Conference*, pages 3180–3185.

[Farina et al., 2012] Farina, M., Ferrari-Trecate, G., and Scattolini, R. (2012). Distributed moving horizon estimation for nonlinear constrained systems. *International Journal of Robust and Nonlinear Control*, 22(2):123–143.

[Ferrante et al., 2016] Ferrante, F., Gouaisbaut, F., Sanfelice, R. G., and Tarbouriech, S. (2016). State estimation of linear systems in the presence of sporadic measurements. *Automatica*, 73:101–109.

[Franzè et al., 2020] Franzè, G., Lucia, W., and Venturino, A. (2020). A distributed model predictive control strategy for constrained multi-vehicle systems moving in unknown environments. *IEEE Transactions on Intelligent Vehicles*, 6(2):343–352.

[Gallo et al., 2020] Gallo, A. J., Turan, M. S., Boem, F., Parisini, T., and Ferrari-Trecate, G. (2020). A distributed cyber-attack detection scheme with application to DC microgrids. *IEEE Transactions on Automatic Control*, 65(9):3800–3815.

[Garin and Schenato, 2010] Garin, F. and Schenato, L. (2010). A survey on distributed estimation and control applications using linear consensus algorithms. In *Networked control systems*, pages 75–107. Springer.

[Ghaderyan et al., 2021] Ghaderyan, D., Pereira, F. L., and Aguiar, A. P. (2021). A fully distributed method for distributed multiagent system in a microgrid. *Energy Reports*, 7:2294–2301.

[Ghafouri et al., 2020] Ghafouri, M., Au, M., Kassouf, M., Debbabi, M., Assi, C., and Yan, J. (2020). Detection and mitigation of cyber attacks on voltage stability monitoring of smart grids. *IEEE Transactions on Smart Grid*, 11(6):5227–5238.

[Gheitasi et al., 2019] Gheitasi, K., Ghaderi, M., and Lucia, W. (2019). A novel networked control scheme with safety guarantees for detection and mitigation of cyber-attacks. In *18th European Control Conference*, pages 1449–1454.

[Goldfarb and Idnani, 1983] Goldfarb, D. and Idnani, A. (1983). A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical programming*, 27(1):1–33.

[Grahn et al., 2017] Grahn, P., Briggner, V., Johansson, L., Babazadeh, D., and Nordstrom, L. (2017). Centralized versus distributed state estimation for hybrid AC/HVDC grid. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–6.

[Haber and Verhaegen, 2013] Haber, A. and Verhaegen, M. (2013). Moving horizon estimation for large-scale interconnected systems. *IEEE Transactions on Automatic Control*, 58(11):2834–2847.

[Halsted et al., 2021] Halsted, T., Shorinwa, O., Yu, J., and Schwager, M. (2021). A survey of distributed optimization methods for multi-robot systems. *arXiv preprint arXiv:2103.12840*, pages 1–17.

[He et al., 2020] He, S., Shin, H.-S., Xu, S., and Tsourdos, A. (2020). Distributed estimation over a low-cost sensor network: A review of state-of-the-art. *Information Fusion*, 54:21–43.

[Hespanha et al., 2007] Hespanha, J. P., Naghshtabrizi, P., and Xu, Y. (2007). A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162.

[Isaza-Hurtado et al., 2019] Isaza-Hurtado, J. A., Martinez, J. J., and Botero-Castro, H. A. (2019). A new approach to receding horizon state estimation for LTI systems in the presence of non-uniform sampled measurements. *International Journal of Control, Automation and Systems*, 17(3):679–690.

[J. Zeng and Liu, 2015] J. Zeng and Liu, J. (2015). Distributed moving horizon estimation subject to communication delays and losses. In *American Control Conference*, pages 5533–5538.

[Jazwinski, 2007] Jazwinski, A. H. (2007). *Stochastic processes and filtering theory*. Courier Corporation.

[Jin et al., 2021] Jin, X., Lü, S., Deng, C., and Chadli, M. (2021). Distributed adaptive security consensus control for a class of multi-agent systems under network decay and intermittent attacks. *Information Sciences*, 547:88–102.

[Karg and Lucia, 2021] Karg, B. and Lucia, S. (2021). Approximate moving horizon estimation and robust nonlinear model predictive control via deep learning. *Computers & Chemical Engineering*, 148:107266.

[Khauphung et al., 2008] Khauphung, C., Keeratiwintakorn, P., and Kaemarungsi, K. (2008). On robustness of centralized-based location determination using WSN. In *2008 14th Asia-Pacific Conference on Communications*, pages 1–5.

[Kim et al., 2021] Kim, J., Kang, J., Bae, J., Lee, W., and Kim, K. K. (2021). Distributed moving horizon estimation via operator splitting for automated robust power system state estimation. *IEEE Access*, 9:90428–90440.

[Kong et al., 2009] Kong, Q., Chen, Y., and Liu, Y. (2009). A fusion-based system for road-network traffic state surveillance: a case study of Shanghai. *IEEE Intelligent Transportation Systems Magazine*, 1(1):37–42.

[Lauricella et al., 2017] Lauricella, M., Farina, M., Schneider, R., and Scattolini, R. (2017). A distributed fault detection and isolation algorithm based on moving horizon estimation. *IFAC-PapersOnLine*, 50(1):15259–15264.

[Lauricella et al., 2020] Lauricella, M., Farina, M., Schneider, R., and Scattolini, R. (2020). Iterative distributed fault detection and isolation for linear systems based on moving horizon estimation. *International Journal of Adaptive Control and Signal Processing*, 34(6):743–756.

[Löfberg, 2004] Löfberg, J. (2004). Yalmip: A toolbox for modeling and optimization in MATLAB. In *2004 IEEE International Conference on Robotics and Automation*, pages 284–289.

[Manfredi, 2013] Manfredi, S. (2013). Design of a multi-hop dynamic consensus algorithm over wireless sensor networks. *Control Engineering Practice*, 21(4):381–394.

[Marzat et al., 2018] Marzat, J., Piet-Lahanier, H., and Bertrand, S. (2018). Cooperative fault detection and isolation in a surveillance sensor network: A case study. *IFAC-PapersOnLine*, 51(24):790–797.

[Mechali et al., 2022] Mechali, O., Xu, L., Xie, X., and Iqbal, J. (2022). Theory and practice for autonomous formation flight of quadrotors via distributed robust sliding mode control protocol with fixed-time stability guarantee. *Control Engineering Practice*, 123:105150, 1–26.

[Mei et al., 2019] Mei, X., Wu, H., Xian, J., Chen, B., Zhang, H., and Liu, X. (2019). A robust, non-cooperative localization algorithm in the presence of outlier measurements in ocean sensor networks. *Sensors*, 19(12):2708.

[Millán et al., 2013] Millán, P., Orihuela, L., Vivas, C., Rubio, F., Dimarogonas, D. V., and Johansson, K. H. (2013). Sensor-network-based robust distributed control and estimation. *Control Engineering Practice*, 21(9):1238–1249.

[Mo et al., 2011] Mo, Y., Ambrosino, R., and Sinopoli, B. (2011). Sensor selection strategies for state estimation in energy constrained wireless sensor networks. *Automatica*, 47(7):1330–1338.

[Moors et al., 2005] Moors, M., Rohling, T., and Schulz, D. (2005). A probabilistic approach to coordinated multi-robot indoor surveillance. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3447–3452.

[Muntwiler et al., 2022] Muntwiler, S., Wabersich, K. P., and Zeilinger, M. N. (2022). Learning-based moving horizon estimation through differentiable convex optimization layers. In *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, volume 168 of *Proceedings of Machine Learning Research*, pages 153–165.

[Muske et al., 1993] Muske, K. R., Rawlings, J. B., and Lee, J. H. (1993). Receding horizon recursive state estimation. In *American Control Conference*, pages 900–904.

[Negenborn and Maestre, 2014] Negenborn, R. and Maestre, J. (2014). Distributed Model Predictive Control: An overview and roadmap of future research opportunities. *IEEE Control Systems Magazine*, 34(4):87–97.

[Nguyen et al., 2017] Nguyen, M. T., Stoica Maniu, C., and Olaru, S. (2017). Optimization-based control for multi-agent deployment via dynamic Voronoi partition. *IFAC-PapersOnLine*, 50(1):1828–1833.

[Nguyen et al., 2015] Nguyen, M. T., Stoica Maniu, C., Olaru, S., and Grancharova, A. (2015). Formation reconfiguration using model predictive control techniques for multi-agent dynamical systems. In *Developments in model-based optimization and control*, pages 183–205.

[Nogueira and Pereira, 2019] Nogueira, M. B. and Pereira, F. L. (2019). Motion control of multiple auvs for simultaneous mapping and navigation. In *Cooperative Localization and Navigation*, pages 455–490.

[Nogueira et al., 2010] Nogueira, M. B., Sousa, J. B., and Pereira, F. L. (2010). Cooperative autonomous underwater vehicle localization. In *OCEANS'10 IEEE SYDNEY*, pages 1–9.

[Olfati-Saber, 2005] Olfati-Saber, R. (2005). Distributed kalman filter with embedded consensus filters. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 8179–8184.

[Olfati-Saber, 2007] Olfati-Saber, R. (2007). Distributed Kalman filtering for sensor networks. In *2007 46th IEEE Conference on Decision and Control*, pages 5492–5498.

[Petitti et al., 2011] Petitti, A., Di Paola, D., Rizzo, A., and Cicirelli, G. (2011). Consensus-based distributed estimation for target tracking in heterogeneous sensor networks. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 6648–6653.

[Pimenta et al., 2009] Pimenta, L. C., Schwager, M., Lindsey, Q., Kumar, V., Rus, D., Mesquita, R. C., and Pereira, G. A. (2009). Simultaneous coverage and tracking (SCAT) of moving targets with robot networks. In *Algorithmic foundation of robotics VIII*, pages 85–99.

[Postoyan and Nešić, 2011] Postoyan, R. and Nešić, D. (2011). A framework for the observer design for networked control systems. *IEEE Transactions on Automatic Control*, 57(5):1309–1314.

[Quevedo et al., 2012] Quevedo, D. E., Ahlen, A., and Johansson, K. H. (2012). State estimation over sensor networks with correlated wireless fading channels. *IEEE Transactions on Automatic Control*, 58(3):581–593.

[Rao and Rawlings, 2000] Rao, C. V. and Rawlings, J. B. (2000). Nonlinear moving horizon state estimation. In *Nonlinear model predictive control*, pages 45–69.

[Rao et al., 2001] Rao, C. V., Rawlings, J. B., and Lee, J. H. (2001). Constrained linear state estimation — a moving horizon approach. *Automatica*, 37(10):1619–1628.

[Rego et al., 2019a] Rego, F. F., Pascoal, A. M., Aguiar, A. P., and Jones, C. N. (2019a). Distributed state estimation for discrete-time linear time invariant systems: A survey. *Annual Reviews in Control*, 48:36–56.

[Rego et al., 2019b] Rego, F. F., Pascoal, A. M., Aguiar, A. P., and Jones, C. N. (2019b). Distributed state estimation for discrete-time linear time invariant systems: A survey. *Annual Reviews in Control*, 48:36–56.

[Robin and Lacroix, 2016] Robin, C. and Lacroix, S. (2016). Multi-robot target detection and tracking: taxonomy and survey. *Autonomous Robots*, 40(4):729–760.

[Sargolzaei et al., 2019] Sargolzaei, A., Yazdani, K., Abbaspour, A., Crane III, C. D., and Dixon, W. E. (2019). Detection and mitigation of false data injection attacks in networked control systems. *IEEE Transactions on Industrial Informatics*, 16(6):4281–4292.

[Sarras et al., 2017] Sarras, I., Marzat, J., Bertrand, S., and Piet-Lahanier, H. (2017). Collaborative multi-vehicle localization with respect to static/dynamic target from range and velocity measurements. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 850–859.

[Segovia et al., 2021a] Segovia, P., Puig, V., and Duviella, E. (2021a). Set-membership-based distributed moving horizon estimation of large-scale systems. *ISA Transactions*, pages 1–12.

[Segovia et al., 2021b] Segovia, P., Puig, V., Duviella, E., and Etienne, L. (2021b). Distributed model predictive control using optimality condition decomposition and community detection. *Journal of Process Control*, 99:54–68.

[Segovia et al., 2019] Segovia, P., Rajaoarisoa, L., Nejjari, F., Duviella, E., and Puig, V. (2019). Model predictive control and moving horizon estimation for water level regulation in inland waterways. *Journal of Process Control*, 76:1–14.

[Shorinwa et al., 2020] Shorinwa, O., Yu, J., Halsted, T., Koufos, A., and Schwager, M. (2020). Distributed multi-target tracking for autonomous vehicle fleets.

In *IEEE International Conference on Robotics and Automation*, pages 3495–3501.

[Simonetto et al., 2011] Simonetto, A., Balzaretti, D., and Keviczky, T. (2011). A distributed moving horizon estimator for mobile robot localization problems. *IFAC Proceedings Volumes*, 44(1):8902–8907.

[Soatti et al., 2016] Soatti, G., Nicoli, M., Savazzi, S., and Spagnolini, U. (2016). Consensus-based algorithms for distributed network-state estimation and localization. *IEEE Transactions on Signal and Information Processing over Networks*, 3(2):430–444.

[Sui and Johansen, 2014] Sui, D. and Johansen, T. A. (2014). Linear constrained moving horizon estimator with pre-estimating observer. *Systems & Control Letters*, 67:40–45.

[Sui et al., 2010] Sui, D., Johansen, T. A., and Feng, L. (2010). Linear moving horizon estimation with pre-estimating observer. *IEEE Transactions on Automatic Control*, 55(10):2363–2368.

[Suwantong et al., 2014] Suwantong, R., Bertrand, S., Dumur, D., and Beauvois, D. (2014). Stability of a nonlinear moving horizon estimator with pre-estimation. In *American Control Conference*, pages 5688–5693.

[Tebbani et al., 2013] Tebbani, S., Le Brusquet, L., Petre, E., and Selisteanu, D. (2013). Robust moving horizon state estimation: Application to bioprocesses. In *17th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 539–544.

[Tedesco et al., 2016] Tedesco, F., Ocampo-Martinez, C., Casavola, A., and Puig, V. (2016). Centralized and distributed command governor approaches for water supply systems management. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(4):586–595.

[Tošić et al., 2013] Tošić, T., Thomos, N., and Frossard, P. (2013). Distributed sensor failure detection in sensor networks. *Signal Processing*, 93(2):399–410.

[Trapiello et al., 2020] Trapiello, C., Puig, V., and Cembrano, G. (2020). System reconfiguration of large-scale control systems using back-up actuators. In *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, volume 1, pages 335–340.

[Vadigepalli and Doyle III, 2003] Vadigepalli, R. and Doyle III, F. J. (2003). Structural analysis of large-scale systems for distributed state estimation and control applications. *Control Engineering Practice*, 11(8):895–905.

[Vargas et al., 2022] Vargas, S., Becerra, H. M., and Hayet, J.-B. (2022). MPC-based distributed formation control of multiple quadcopters with obstacle avoidance and connectivity maintenance. *Control Engineering Practice*, 121:105054, 1–16.

[Venturino et al., 2020] Venturino, A., Bertrand, S., Stoica Maniu, C., Alamo, T., and Camacho, E. F. (2020). Distributed moving horizon estimation with pre-estimating observer. In *24th International Conference on System Theory, Control and Computing*, pages 174–179.

[Venturino et al., 2021a] Venturino, A., Bertrand, S., Stoica Maniu, C., Alamo, T., and Camacho, E. F. (2021a). A new ℓ-step neighbourhood distributed moving horizon estimator. In *60th IEEE Conference on Decision and Control*, pages 508–513.

[Venturino et al., 2022a] Venturino, A., Bertrand, S., Stoica Maniu, C., Alamo, T., and Camacho, E. F. (2022a). Multi-vehicle system localization by distributed moving horizon estimation over a sensor network with sporadic measurements. In *6th IEEE Conference on Control Technology and Applications*.

[Venturino and Lucia, 2019] Venturino, A. and Lucia, W. (2019). A flexible distributed control strategy for teams of vehicles moving within severe obstacle scenarios. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 941–946.

[Venturino et al., 2021b] Venturino, A., Stoica Maniu, C., Bertrand, S., Alamo, T., and Camacho, E. F. (2021b). Distributed moving horizon state estimation for sensor networks with low computation capabilities. *System Theory, Control and Computing Journal*, 1(1):81–87.

[Venturino et al., 2022b] Venturino, A., Stoica Maniu, C., Bertrand, S., Alamo, T., and Camacho, E. F. (2022b). Multi-vehicle localization by distributed MHE over a sensor network with sporadic measurements: further developments and experimental results. *Submitted to Control Engineering Practice*.

[Viani et al., 2015] Viani, F., Robol, F., Giarola, E., Rocca, P., Oliveri, G., and Massa, A. (2015). Passive imaging strategies for real-time wireless localization of non-cooperative targets in security applications. In *9th European Conference on Antennas and Propagation (EuCAP)*, pages 1–4.

[Viegas et al., 2018] Viegas, D., Batista, P., Oliveira, P., and Silvestre, C. (2018). Discrete-time distributed Kalman filter design for formations of autonomous vehicles. *Control Engineering Practice*, 75:55–68.

[Vincent et al., 2008] Vincent, R., Fox, D., Ko, J., Konolige, K., Limketkai, B., Morisset, B., Ortiz, C., Schulz, D., and Stewart, B. (2008). Distributed mul-

tirobot exploration, mapping, and task allocation. *Annals of Mathematics and Artificial Intelligence*, 52:229–255.

[Vukov et al., 2015] Vukov, M., Gros, S., Horn, G., Frison, G., Geebelen, K., Jørgensen, J. B., Swevers, J., and Diehl, M. (2015). Real-time nonlinear MPC and MHE for a large-scale mechatronic application. *Control Engineering Practice*, 45:64–78.

[Wang and Olson, 2016] Wang, J. and Olson, E. (2016). AprilTag 2: Efficient and robust fiducial detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198.

[Wang et al., 2017] Wang, Y., Alamo, T., Puig, V., and Cembrano, G. (2017). Distributed zonotopic set-membership state estimation based on optimization methods with partial projection. *IFAC-PapersOnLine*, 50(1):4039–4044.

[Wang et al., 2018] Wang, Y., Alamo, T., Puig, V., and Cembrano, G. (2018). A distributed set-membership approach based on zonotopes for interconnected systems. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 668–673.

[Wu et al., 2019] Wu, Y., Wang, Z., and Huang, Z. (2019). Distributed fault detection for nonlinear multi-agent systems under fixed-time observer. *Journal of the Franklin Institute*, 356(13):7515–7532.

[Xiao et al., 2005] Xiao, L., Boyd, S., and Lall, S. (2005). A scheme for robust distributed sensor fusion based on average consensus. In *4th International Symposium on Information Processing in Sensor Networks*, pages 63–70.

[Xie et al., 2012] Xie, L., Choi, D.-H., Kar, S., and Poor, H. V. (2012). Fully distributed state estimation for wide-area monitoring systems. *IEEE Transactions on Smart Grid*, 3(3):1154–1169.

[Yin and Huang, 2022] Yin, X. and Huang, B. (2022). Event-triggered distributed moving horizon state estimation of linear systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–13.

[Yin and Liu, 2017] Yin, X. and Liu, J. (2017). Distributed moving horizon state estimation of two-time-scale nonlinear systems. *Automatica*, 79:152–161.

[Yousefi and Menhaj, 2014] Yousefi, Z. R. and Menhaj, M. B. (2014). Constrained distributed estimation based on consensus algorithm for mobile robots tracking. In *IEEE International Conference on Control System, Computing and Engineering*, pages 124–129.

[Zhang and Liu, 2013] Zhang, J. and Liu, J. (2013). Distributed moving horizon state estimation for nonlinear systems with bounded uncertainties. *Journal of Process Control*, 23(9):1281–1295.

[Zhang et al., 2019] Zhang, X.-M., Han, Q., Ge, X., Ding, D., Ding, L., Yue, D., and Peng, C. (2019). Networked control systems: A survey of trends and techniques. *IEEE/CAA Journal of Automatica Sinica*, 7(1):1–17.

[Zhang and Meng, 2010] Zhang, Y. and Meng, Y. (2010). A decentralized multi-robot system for intruder detection in security defense. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5563–5568.

[Zhao and Zelazo, 2015] Zhao, S. and Zelazo, D. (2015). Bearing-based distributed control and estimation of multi-agent systems. In *2015 European Control Conference (ECC)*, pages 2202–2207.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Estimation d'état distribuée sous contraintes pour une mission de surveillance multi-capteurs multi-robots
Mots clés: Estimation d'état distribuée, Estimation sous contraintes, Estimation distribuée à horizon glissant, Réseaux de capteurs.

Résumé: Les algorithmes distribués sont dorénavant présents dans de nombreux aspects de l'Automatique avec des applications pour des systèmes multi-robots, des réseaux de capteurs, couvrant des sujets tels que la commande, l'estimation d'état, la détection de défauts, la détection et l'atténuation des cyber-attaques sur les systèmes cyber-physiques, etc. En effet, les systèmes distribués sont confrontés à des problèmes tels que l'extensibilité à un grand nombre d'agents et la communication entre eux. Dans les applications de systèmes multi-agents (par exemple, flotte de robots mobiles, réseaux de capteurs), il est désormais courant de concevoir des algorithmes d'estimation d'état de manière distribuée afin que les agents puissent accomplir leurs tâches sur la base de certaines informations partagées au sein de leur voisinage. Dans le cas de missions de surveillance, un réseau de capteurs statique et à faible coût (par exemple, caméras) pourrait ainsi être déployé pour localiser de manière distribuée des intrus dans une zone donnée. Dans ce contexte, l'objectif principal de cette thèse est de concevoir des observateurs distribués pour estimer l'état d'un système dynamique (par exemple, flotte de robots intrus) avec une charge de calcul réduite tout en gérant efficacement les contraintes et les incertitudes. Cette thèse propose de nouveaux algorithmes d'estimation distribuée à horizon glissant avec une pré-estimation de type Luenberger dans la formulation du problème local résolu par chaque capteur, entraînant une réduction significative du temps de calcul, tout en préservant la précision de l'estimation. En outre, ce manuscrit propose une stratégie de consensus pour améliorer le temps de convergence des estimations entre les capteurs sous des conditions de faible observabilité (par exemple, des véhicules intrus non visibles par certaines caméras). Une autre contribution concerne l'amélioration de la convergence de l'erreur d'estimation en atténuant les problèmes de non observabilité à l'aide d'un mécanisme de diffusion de l'information sur plusieurs pas (appelé ℓ-*step*) entre voisinages. L'estimation distribuée proposée est conçue pour des scénarios réalistes de systèmes à grande échelle impliquant des mesures sporadiques (c'est-à-dire disponibles à des instants a priori inconnus). À cette fin, les contraintes sur les mesures (par exemple, le champ de vision de caméras) sont incorporées dans le problème d'optimisation à l'aide de paramètres binaires variant dans le temps. L'algorithme développé est implémenté sous le middleware ROS (Robot Operating System) et des simulations réalistes sont faites à l'aide de l'environnement Gazebo. Une validation expérimentale de la technique de localisation proposée est également réalisée pour un système multi-véhicules (SMV) à l'aide d'un réseau de capteurs statiques composé de caméras à faible coût qui fournissent des mesures sur les positions d'une flotte de robots mobiles composant le SMV. Les algorithmes proposés sont également comparés à des résultats de la littérature en considérant diverses métriques telles que le temps de calcul et la précision des estimées.

Title: Constrained distributed state estimation for surveillance missions using multi-sensor multi-robot systems
Keywords: Distributed state estimation, Constrained state estimation, Distributed Moving Horizon Estimation, Sensor Networks.

Abstract: Distributed algorithms have pervaded many aspects of control engineering with applications for multi-robot systems, sensor networks, covering topics such as control, state estimation, fault detection, cyber-attack detection and mitigation on cyber-physical systems, etc. Indeed, distributed schemes face problems like scalability and communication between agents. In multi-agent systems applications (e.g. fleet of mobile robots, sensor networks) it is now common to design state estimation algorithms in a distributed way so that the agents can accomplish their tasks based on some shared information within their neighborhoods. In surveillance missions, a low-cost static Sensor Network (e.g. with cameras) could be deployed to localize in a distributed way intruders in a given area. In this context, the main objective of this work is to design distributed observers to estimate the state of a dynamic system (e.g. a multi-robot system) that efficiently handle constraints and uncertainties but with reduced computation load. This PhD thesis proposes new Distributed Moving Horizon Estimation (DMHE) algorithms with a Luenberger pre-estimation in the formulation of the local problem solved by each sensor, resulting in a significant reduction of the computation time, while preserving the estimation accuracy. Moreover, this manuscript proposes a consensus strategy to enhance the convergence time of the estimates among sensors while dealing with weak unobservability conditions (e.g. vehicles not visible by some cameras). Another contribution concerns the improvement of the convergence of the estimation error by mitigating unobservability issues by using a ℓ-step neighborhood information spreading mechanism. The proposed distributed estimation is designed for realistic large-scale systems scenarios involving sporadic measurements (i.e. available at time instants a priori unknown). To this aim, constraints on measurements (e.g. camera field of view) are embodied using time-varying binary parameters in the optimization problem. Both realistic simulations within the Robot Operating System (ROS) framework and Gazebo environment, as well as experimental validation of the proposed DMHE localization technique of a Multi-Vehicle System (MVS) with ground mobile robots are performed, using a static Sensor Network composed of low-cost cameras which provide measurements on the positions of the robots of the MVS. The proposed algorithms are compared to previous results from the literature, considering several metrics such as computation time and accuracy of the estimates.