



M.Sc. Thesis

Direction of Arrival based Distributed Target Tracking using Primal-Dual Method of Multipliers

ing. O.R. Bolleurs

Abstract

Underwater surveillance of large areas, such as the North Sea, is an increasingly relevant problem, driven by growing concern over activity targeting undersea infrastructure. A sensor network can address this by measuring the Direction of Arrival (DoA) of a target. Centralised tracking with an Extended Kalman Filter (EKF) requires every measurement to reach a central node, a communication cost that is prohibitive for remote, battery-powered sensors. This thesis instead develops a distributed solution, formulated as a consensus problem and solved using the Primal-Dual Method of Multipliers (PDMM).

The EKF is reformulated as a distributed maximum a posteriori (MAP) problem, yielding EKF-PDMM. An equivalent, faster-converging formulation is then derived in information form, exploiting the additive structure of the Extended Information Filter (EIF), yielding EIF-PDMM. EIF-PDMM is extended with gating for multi-target association (EIF-PDMM-MT) and with sensor selection to reduce active nodes by an order of magnitude in large networks (EIF-PDMM-SS). Combined into EIF-PDMM-SS-MT, the algorithm achieves accurate, scalable target tracking without any central point of fusion.

Direction of Arrival based Distributed Target Tracking using Primal-Dual Method of Multipliers

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

by

ing. O.R. Bolleurs
born in Houten, The Netherlands

This work was performed in:

Signal Processing Systems Group
Department of Microelectronics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology



Delft University of Technology

Copyright © 2026 Signal Processing Systems Group
All rights reserved.

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
MICROELECTRONICS

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled “**Direction of Arrival based Distributed Target Tracking using Primal-Dual Method of Multipliers**” by **ing. O.R. Bolleurs** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 28-08-2026

Chairman:

prof.dr.ir. R. Heusdens

Committee Members:

dr.ir. J.N. Driessen

dr. R.T. Rajan

Abstract

Underwater surveillance of large areas, such as the North Sea, is an increasingly relevant problem, driven by growing concern over activity targeting undersea infrastructure. A sensor network can address this by measuring the Direction of Arrival (DoA) of a target. Centralised tracking with an Extended Kalman Filter (EKF) requires every measurement to reach a central node, a communication cost that is prohibitive for remote, battery-powered sensors. This thesis instead develops a distributed solution, formulated as a consensus problem and solved using the Primal-Dual Method of Multipliers (PDMM).

The EKF is reformulated as a distributed maximum a posteriori (MAP) problem, yielding EKF-PDMM. An equivalent, faster-converging formulation is then derived in information form, exploiting the additive structure of the Extended Information Filter (EIF), yielding EIF-PDMM. EIF-PDMM is extended with gating for multi-target association (EIF-PDMM-MT) and with sensor selection to reduce active nodes by an order of magnitude in large networks (EIF-PDMM-SS). Combined into EIF-PDMM-SS-MT, the algorithm achieves accurate, scalable target tracking without any central point of fusion.

Acknowledgments

It has finally come to the point to write the last part of my thesis, and it is reserved for thanking everyone who helped me accomplish this. First of all, I would like to thank my advisor prof.dr.ir. R. Heusdens for his assistance during the writing of this thesis. I look back at a great time and really like the discussions we had.

To all the people at the SPS group, thank you all for the discussions during a coffee or lunch break. I felt part of the group.

To my friends, thanks for the laughs, discussions, brainstorm sessions, and most importantly the beers. You all helped me get through this.

A special thanks to my family, for those who are still here and those who have sadly passed. Without you, I could not have done this, especially my mother and sister. Without you both I would not have been where I am today, thank you.

To Annewieke, my girlfriend, thank you for all your patience, motivation, celebrations, and most of all support. I am glad you were the one next to me during this time.

ing. O.R. Bolleurs
Delft, The Netherlands
28-08-2026

The assistant of AI has been used for correcting and applying structure to this thesis. All ideas and methods introduced here are mine, AI only has been used to make this thesis readable with correct grammar and spelling.

Contents

Abstract	v
Acknowledgments	vii
List of Acronyms	xv
Nomenclature	xvii
1 Introduction	1
1.1 Research Question and Outline	1
2 Centralised Model	3
2.1 State-Space Model	3
2.2 Kalman Filters	5
2.2.1 Kalman Filter (KF)	5
2.2.2 Extended Kalman Filter (EKF)	6
2.3 Extended Information Filter (EIF)	7
2.4 Multi-Target Tracking	8
2.4.1 Multi-Target Notation	9
2.4.2 Gating	9
3 Distributed Model	13
3.1 General Idea of Distributed Algorithms	13
3.2 Primal-Dual Method of Multipliers (PDMM)	14
3.3 Extended Kalman Filter – Primal-Dual Method of Multipliers	16
3.3.1 Maximum a Posteriori Formulation for the EKF	16
3.3.2 Update Equations for EKF-PDMM	17
3.4 Extended Information Filter – Primal-Dual Method of Multipliers	19
3.5 Convergence Comparison: EKF-PDMM vs. EIF-PDMM	20
3.5.1 Experimental Setup	20
3.5.2 Results	21
4 Extension to Multi-Target Tracking and Sensor Selection	23
4.1 Multi-Target in EIF-PDMM	23
4.1.1 Experimental Setup	24
4.1.2 Results	25
4.1.3 Improvements	26
4.2 Sensor Selection	27
4.2.1 Experimental Setup	29
4.2.2 Results	29
4.2.3 Improvements	31
4.3 Sensor Selection in a Multi-Target Environment	32

4.3.1	Experimental Setup	32
4.3.2	Results	32
5	Conclusion	37
5.1	Future Work	37
	References	39
A	EIF Derivation	41
B	EIF-PDMM Algorithm	43

List of Figures

2.1	Sensor geometry for a single sensor node i	4
3.1	Information exchange between neighbouring nodes i and j , node i receives data $y_{j i}$ from j and treats it as $z_{i j}$, while node j receives $y_{i j}$ from i and treats it as $z_{j i}$	15
3.2	Consensus error against PDMM step size c , used to select the optimal c for EKF-PDMM and EIF-PDMM.	21
3.3	Consensus error $e_{\text{cons}}(k)$ against PDMM iteration k at the final time step, for EKF-PDMM and EIF-PDMM. The horizontal axis is truncated at 400 of the 4000 iterations simulated.	21
3.4	State error $e_{\text{state}}(t)$ over time for EKF-PDMM vs. EIF-PDMM.	22
4.1	Mean state error per target over time for EIF-PDMM-MT and EIF-MT, with their absolute difference shown in the bottom subplot.	25
4.2	Estimated target trajectories in three dimensions, EIF-PDMM-MT.	26
4.3	Estimated target trajectories projected onto the X-Y and X-Z planes, EIF-PDMM-MT.	26
4.4	Mean state error over 100 Monte Carlo runs, EIF-PDMM-SS vs. EIF-PDMM.	30
4.5	Mean number of selected nodes per time step t , EIF-PDMM-SS.	30
4.6	Estimated target trajectory in three dimensions, EIF-PDMM-SS.	31
4.7	Estimated target trajectory projected onto the X-Y and X-Z planes, EIF-PDMM-SS.	31
4.8	Mean state error over 100 Monte Carlo runs, EIF-PDMM-SS-MT vs. EIF-PDMM-MT (first scenario).	33
4.9	Mean state error over 100 Monte Carlo runs, EIF-PDMM-SS-MT vs. EIF-PDMM-MT (second scenario).	33
4.10	Mean number of selected nodes per time step t , EIF-PDMM-SS-MT (first scenario).	34
4.11	Mean number of selected nodes per time step t , EIF-PDMM-SS-MT (second scenario).	34
4.12	Estimated target trajectories in three dimensions, EIF-PDMM-SS-MT (first scenario).	34
4.13	Estimated target trajectories in three dimensions, EIF-PDMM-SS-MT (second scenario).	34
4.14	Estimated target trajectories projected onto the X-Y and X-Z planes, EIF-PDMM-SS-MT (first scenario).	35
4.15	Estimated target trajectories projected onto the X-Y and X-Z planes, EIF-PDMM-SS-MT (second scenario).	35
4.16	Estimated target states and active nodes at $t = 100$, EIF-PDMM-SS-MT (first scenario).	35

List of Tables

0.1	Latin notation used throughout this thesis.	xvii
0.2	Greek notation used throughout this thesis.	xix
0.3	Notational conventions used throughout this thesis.	xx
3.1	Simulation parameters used in the convergence comparison.	20
4.1	Simulation parameters used in the multi-target scenario.	24
4.2	Simulation parameters used in the sensor selection scenario.	29
4.3	Simulation parameters used in the sensor selection multi-target scenario.	32

List of Acronyms

ADMM Alternating-Direction Method of Multipliers. 1, 14, 16, 19

CDF Cumulative Distribution Function. xvii, 10

DoA Direction of Arrival. i, v, 1, 37

EIF Extended Information Filter. i, v, xvii, xix, 1, 3, 7, 8, 13, 19, 21–23, 25, 27, 29, 30, 32–34, 37, 41

EKF Extended Kalman Filter. i, v, xvii, 1, 3, 5–8, 11, 13, 15–17, 19, 21, 22, 27, 37, 41

i.i.d Independent and Identically Distributed. 3

JPDA Joint Probabilistic Data Association. 26, 38

KF Kalman Filter. xvii, 1, 3, 5, 6, 11, 16

MAP maximum a posteriori. i, v, 13, 16, 17, 19, 22

MMSE Minimum Mean Squared Error. 5

PDMM Primal-Dual Method of Multipliers. i, v, xvii–xx, 1, 13–25, 27–30, 32–34, 37

Nomenclature

Table 0.1: Latin notation used throughout this thesis.

Symbol	Description
$a_i, \bar{a}$	Local measurement at node i and its network average (introductory distributed example)
$A_{i j}$	Coefficient matrix imposing a constraint between the local variables of neighbouring nodes i and j , indexed by the directed edge from i to j (PDMM)
$b_{i,j}$	Constant offset vector for the edge (i, j) (PDMM consensus constraint)
c	PDMM step size, controls convergence rate
C, C_i	Constant matrix used to rewrite the PDMM objective, and its per-node component
d_i	Degree of node i (PDMM consensus constraint)
d_i^{active}	Degree of node i within the currently active subgraph (sensor selection)
D^2	Squared Mahalanobis distance
$e_{\text{cons}}(k)$	Consensus error at PDMM iteration k
$e_{\text{state}}(t)$	State estimation error at time t ; per-track variant $e_{\text{state},\ell}(t)$ for track ℓ (multi-target notation)
$\mathcal{E}$	Set of edges of the network graph $G(\mathcal{V}, \mathcal{E})$
$f(\cdot), f_i(\cdot)$	Global and per-node objective function minimised by PDMM
F	State transition matrix
$F_{\chi_{n_z}^2}$	CDF of the $\chi_{n_z}^2$ distribution
$F_{\chi_{n_z}^2}^{-1}$	Inverse CDF (quantile function) of the $\chi_{n_z}^2$ distribution
$G(\mathcal{V}, \mathcal{E})$	Network graph with node set $\mathcal{V}$ and edge set $\mathcal{E}$
$h(\cdot)$	Non-linear measurement function
H	Measurement matrix (linear KF)
H_t	Measurement Jacobian at time t (EKF); indexed analogously as $H_{i,t}$ for sensor node i and $H_{\ell,t}$ for track ℓ
i	Sensor node index
I	Identity matrix
I_t	Measurement information matrix (EIF, additive term); distributed analogue is $\Xi_{i,t}$ below
i_t	Measurement information vector (EIF, additive term); distributed analogue is $\xi_{i,t}$ below
j	Index of a neighbouring sensor node
k	PDMM iteration index
K_t	Kalman gain

Table 0.1 – continued

Symbol	Description
ℓ	Selected track index
L	Number of tracks, equal to the number of targets in this work
m	Selected measurement index
$\hat{m}$	Measurement selected for track ℓ by the nearest-neighbour association rule (gating)
M	Number of measurements per sensor, equal to L under the assumptions of this work
n	Dimension of the state vector
N	Number of sensor nodes
N^{active}	Number of currently active nodes (sensor selection)
$\mathcal{N}_i$	Set of (currently active) neighbours of node i
n_z	Dimension of the measurement vector
P	State estimate covariance matrix
P_0	Initial state estimate covariance matrix
p_x, p_y, p_z	Target position components
q, q_x, q_y, q_z	Power spectral density of the random acceleration noise; q_x, q_y, q_z denote the per-axis values, assumed equal in this work
Q	Process noise covariance matrix
r_{max}	Maximum assumed measurement range of a node (sensor selection)
r_i^x, r_i^y, r_i^z	Position of node i
R, R_i	Measurement noise covariance matrix, and its per-node block
S_t	Innovation covariance matrix; per-track variant $S_{\ell,t}$ for track ℓ (multi-target notation)
t	Discrete time instant
T	Maximum simulation time, in seconds
v	Measurement noise
v_x, v_y, v_z	Target velocity components
$\mathcal{V}$	Set of nodes of the network graph $G(\mathcal{V}, \mathcal{E})$
w	Process noise
x_t	State vector at time t ; local, per-node ($x_{i,t}$) and per-track ($x_{\ell,t}$) variants follow the conventions in Table 0.3
x_0	True initial target state used to generate the initial estimate $\hat{x}_{1 0}$
$\hat{x}_{1 0}$	Initial state estimate, generated from x_0
$\hat{x}_{t t-1}$	Predicted state estimate
$\hat{x}_{t t}$	Posterior (corrected) state estimate
$\bar{x}_{t t}^{(k)}$	Average state estimate across all nodes at PDMM iteration k

Table 0.1 – continued

Symbol	Description
y_t	Linear measurement vector
$y^{(k)}, z^{(k)}$	PDMM auxiliary (dual) variables at iteration k ; node-specific versions $y_{i j}, z_{i j}$ for the information vector $\xi_{i,t}$ and $Y_{i j}, Z_{i j}$ for the information matrix $\Xi_{i,t}$
z_t	Non-linear measurement vector
$z_{m,t}$	Measurement m at time t
$z_{1:t}$	Sequence of measurements from time 1 to time t
$z_{1:N,t}$	Stacked measurement vector containing the measurements of all N nodes at time t

Table 0.2: Greek notation used throughout this thesis.

Symbol	Description
β, β_i	PDMM constant vector and its per-node component
δt	Sampling interval between discrete time steps, in seconds
$\Delta x, \Delta y, \Delta z$	Position differences between the target and a sensor node
θ_i	Elevation angle measured at sensor node i
ν_t	Innovation
$\nu_{m,\ell,t}$	Innovation between measurement m and track ℓ
$\tilde{\nu}_{m,\ell,t}$	Whitened innovation
$\xi_{t t}$	Information vector (EIF, full posterior)
$\xi_{i,t}$	Local, N -scaled measurement information vector at node i (EIF-PDMM); distinct from $\xi_{t t}$ above
$\Xi_{t t}$	Information matrix (EIF, full posterior)
$\Xi_{i,t}$	Local, N -scaled measurement information matrix at node i (EIF-PDMM); the measurement-only contribution, distinct from $\Xi_{t t}$ above
Π	Permutation matrix ensuring $z_{j i} = y_{i j}$ (PDMM data exchange)
$\sigma_\varphi^2, \sigma_\theta^2$	Measurement noise variances of the azimuth and elevation angle
φ_i	Azimuth angle measured at sensor node i
$\chi_{n_z}^2$	Chi-squared distribution with n_z degrees of freedom

Table 0.3: Notational conventions used throughout this thesis.

Notation	Meaning
$\hat{(\cdot)}$	Estimate of the quantity $(\cdot)$
$(\cdot)_{t t-1}$	Predicted value at time t , given measurements up to time $t - 1$
$(\cdot)_{t t}$	Posterior (corrected) value at time t , given measurements up to time t
$(\cdot)_i, (\cdot)_{i,t}$	Local, node-specific copy of a quantity at node i (PDMM)
$(\cdot)_\ell, (\cdot)_{\ell,t}$	Per-track quantity for track ℓ (multi-target notation)
$(\cdot)_{i j}$	Quantity transmitted from node i to neighbour j (PDMM exchange)
$(\cdot)^{(k)}$	Value at PDMM iteration k
$(\cdot)^{\text{active}}$	Quantity restricted to the currently active subgraph (sensor selection)
$(\cdot)^\top$	Matrix or vector transpose
$(\cdot)^{-1}$	Matrix inverse
$:=$	Assignment / iterative update, as opposed to mathematical equality
$\mathbb{R}^{a \times b}$	Set of real $a \times b$ matrices
$\{\cdot \mid \cdot\}$	Set-builder notation, read as “such that”
$\emptyset$	Empty set
$\ \cdot\ $	Euclidean (ℓ_2) norm
$\text{diag}(\cdot)$	Diagonal matrix formed from the given entries
$\det(\cdot)$	Determinant of a matrix
$\text{tr}(\cdot)$	Trace of a matrix
$\succeq, \preceq, \succ$	Loewner (positive semi-/definite) matrix ordering
$\arg \min, \arg \max$	Argument that minimises/maximises the given objective
$\mathcal{N}(\mu, \Sigma)$	Gaussian distribution with mean μ and covariance Σ
$\mathcal{N}(x; \mu, \Sigma)$	Gaussian density with mean μ and covariance Σ , evaluated at x
$E[\cdot]$	Expectation operator

Underwater surveillance of large areas such as the North Sea is a challenging and highly relevant problem. With an increase in suspicious activity targeting critical undersea infrastructure, there is a growing demand for monitoring of these large areas [1].

To tackle this problem, a large sensor network could be used to monitor this vast area. These sensors would then measure the Direction of Arrival (DoA) of, for example, the sound produced by a target. A traditional solution would be to combine all measurements of the sensor network at a central location such that a target can be tracked using an Extended Kalman Filter (EKF). This method introduces a problem: combining all the measurements requires them to travel to the central location. In these remote places, transmitting over such distances requires significant power, drastically increasing cost and reducing the battery life of the sensors. This centralised solution also has a single point of failure: the central location where all measurement are combined. It would thus be beneficial to solve this problem in a distributed fashion, combining the measurements locally using only sensor-to-sensor communication. This is also known as distributed optimisation. Multiple distributed solvers exist that can be used to solve such problems, such as Alternating-Direction Method of Multipliers (ADMM) [2] and the more recently introduced Primal-Dual Method of Multipliers (PDMM) [3].

1.1 Research Question and Outline

This thesis addresses the following research question: *How can a distributed sensor network, providing Direction of Arrival (DoA) measurements, be used for target tracking in large, remote areas?* To this end, this thesis develops a distributed algorithm for multi-target tracking.

The remainder of this thesis is structured as follows: Chapter 2 provides background knowledge on the centralised state-space model and the Kalman Filter framework, including the Kalman Filter (KF), the Extended Kalman Filter (EKF), and the Extended Information Filter (EIF). Chapter 3 extends the centralised model to a distributed model and introduces and compares two new distributed algorithms, EKF-PDMM and EIF-PDMM. Chapter 4 shows how the EIF-PDMM algorithm can be extended to a multi-target scenario resulting in EIF-PDMM-MT. Furthermore, it extends the algorithm to do a simple sensor selection method to reduce the number of sensors needed in the network resulting in EIF-PDMM-SS. Lastly, it combines the two resulting in EIF-PDMM-SS-MT. All algorithms are accompanied by simulations. Chapter 5 concludes the thesis with a brief summary and steps to take for future work.

Centralised Model

This chapter introduces the background and notation required to understand the main contribution of this thesis: a distributed target tracking algorithm. To this end, target tracking is first considered for a centralised system, in which all measurements are evaluated at a single, central location. Understanding this centralised setting is a prerequisite for the distributed methods discussed in later chapters. The chapter is structured as follows: first, the state-space model is introduced. Second, the Kalman Filter and Extended Kalman Filter are discussed. Third, the Extended Information Filter is presented, including why it is particularly well-suited for a distributed implementation. Finally, multi-target notation and gating are introduced.

2.1 State-Space Model

In this work, the motion of a target is described using a state-space model. At discrete time instant t , the target state is represented by the state vector $x_t \in \mathbb{R}^{n \times 1}$, where n denotes the dimension of the state vector. For three-dimensional motion, the state vector contains the target position p and velocity v in the x -, y -, and z -directions (giving $n = 6$),

$$x_t = [p_x \ p_y \ p_z \ v_x \ v_y \ v_z]^\top. \quad (2.1)$$

The evolution of the target state is described by the state transition model, while the observations are related to the state through the measurement model. Assuming linear dynamics and linear measurements y_t , the state-space model is given by

$$\begin{aligned} x_{t+1} &= Fx_t + w_t, & w_t &\sim \mathcal{N}(0, Q), \\ y_t &= Hx_t + v_t, & v_t &\sim \mathcal{N}(0, R), \end{aligned} \quad (2.2)$$

where $F \in \mathbb{R}^{n \times n}$ is the state transition matrix and $y_t \in \mathbb{R}^{n_z \times 1}$ is the measurement vector, with n_z denoting the dimension of the measurement vector. The measurement matrix is denoted $H \in \mathbb{R}^{n_z \times n}$, where H is the linear transformation of x_t to y_t . The process noise w models uncertainties in the target dynamics, whereas the measurement noise v accounts for sensor inaccuracies. Both noise terms are assumed to be Independent and Identically Distributed (i.i.d), zero-mean Gaussian random vectors with covariance matrices $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{n_z \times n_z}$, respectively. The specific structure of these covariance matrices is discussed below.

In target tracking, the measurement model is typically non-linear, since sensors commonly measure quantities that are non-linear functions of the Cartesian state. The state-space model is then given by

$$\begin{aligned} x_{t+1} &= Fx_t + w_t, & w_t &\sim \mathcal{N}(0, Q), \\ z_t &= h(x_t) + v_t, & v_t &\sim \mathcal{N}(0, R), \end{aligned} \quad (2.3)$$

where $h(x_t)$ is a non-linear function that maps the state vector x_t to the non-linear measurements $z_t \in \mathbb{R}^{n_z \times 1}$. The specific state transition model and measurement model used in this work are introduced below.

The state transition matrix F is modelled using a constant-velocity model, which assumes zero acceleration between time steps and therefore captures only simple, linear dynamics. However, in combination with the process covariance Q , more complex dynamics such as accelerations or turning motions, can still be captured through the added process noise. The state transition matrix is given by

$$F = \begin{bmatrix} I & \delta t I \\ 0 & I \end{bmatrix}, \quad (2.4)$$

where I denotes the 3×3 identity matrix and δt denotes the sampling interval between discrete time steps t and $t + 1$. Using this constant-velocity model, the state is updated as,

$$x_{t+1} = [p_x + v_x \delta t \quad p_y + v_y \delta t \quad p_z + v_z \delta t \quad v_x \quad v_y \quad v_z]^\top + w_t. \quad (2.5)$$

Different models can be used for the process covariance Q , capturing effects such as acceleration or turning motion. In this work, the process covariance is given by

$$Q = \begin{bmatrix} q(\delta t^4/3)I & q(\delta t^3/2)I \\ q(\delta t^3/2)I & q(\delta t^2)I \end{bmatrix}, \quad (2.6)$$

where q denotes the power spectral densities of the random acceleration noise, assumed equal along the x -, y -, and z -axes. This model adds random accelerations to the state transition, with uncoupled noise between the axes. Further details are given in [4].

Having introduced the state transition model, the measurement model used in this work is discussed next and illustrated in Figure 2.1.

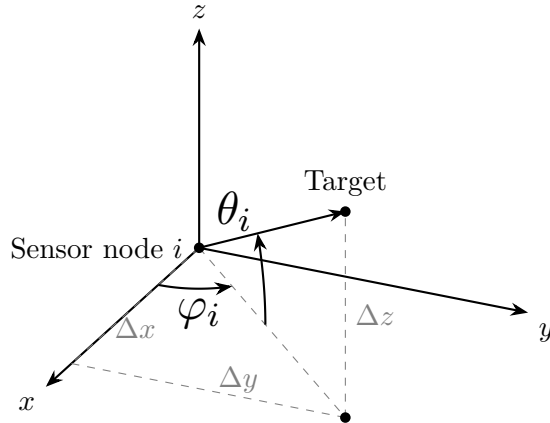


Figure 2.1: Sensor geometry for a single sensor node i .

The measurement model considers angle measurements, namely azimuth and elevation. These measurements are obtained from a sensor network with known sensor positions. For a sensor node $i \in \{1, \dots, N\}$, where N denotes the number of sensor

nodes, the non-linear function $h(x_t)$ that maps the Cartesian position states to the azimuth and elevation angle measurements, is given by

$$h(x_t) = \begin{bmatrix} \varphi_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{\Delta y}{\Delta x}\right) \\ \arctan\left(\frac{\Delta z}{\sqrt{\Delta x^2 + \Delta y^2}}\right) \end{bmatrix}, \quad (2.7)$$

where φ_i and θ_i denote the azimuth angle and elevation angle, respectively, measured at sensor node i , and Δx , Δy , and Δz denote the position differences in the x -, y -, and z -directions between the target and sensor node i . The measurement noise covariance R is given by

$$R = \begin{bmatrix} \sigma_\varphi^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}, \quad (2.8)$$

where σ_φ^2 and σ_θ^2 denote the measurement noise variances of the azimuth and elevation angle, respectively, giving $n_z = 2$ for the angle measurements considered in this work.

2.2 Kalman Filters

In this section, the notation and operation of the Kalman Filter and Extended Kalman Filter are discussed.

2.2.1 Kalman Filter (KF)

The Kalman Filter (KF) is a recursive algorithm that computes the optimal estimate of the state x_t given a sequence of measurements, under the assumption of a linear system as shown in Equation 2.2. Provided the system matrices F and H and the noise covariances Q and R are known, the KF is the optimal linear estimator in the Minimum Mean Squared Error (MMSE) sense [5]. If the process and measurement noise are additionally Gaussian, the KF is optimal among all estimators, not only the linear ones [6].

The KF consists of two steps, the prediction step and the correction step. The notation $\hat{x}_{t|t-1}$ denotes the predicted mean at time t conditioned on the observations $y_0, \dots, y_{t-1}$, with associated covariance $P_{t|t-1} \in \mathbb{R}^{n \times n}$ denoting the uncertainty in this estimate. The prediction step is given by

$$\begin{aligned} \hat{x}_{t|t-1} &= F\hat{x}_{t-1|t-1}, \\ P_{t|t-1} &= FP_{t-1|t-1}F^\top + Q, \end{aligned} \quad (2.9)$$

which propagates the previous posterior estimate forward using the state transition model F . Correspondingly, the predicted covariance reflects how the uncertainty in the previous estimate evolves under this same linear transformation. The term $FP_{t-1|t-1}F^\top$ propagates the existing uncertainty through the dynamics, while the addition of Q accounts for the new uncertainty introduced by the process noise at each time step. As a result, $P_{t|t-1}$ is always at least as large as $P_{t-1|t-1}$, reflecting the fact that, in the

absence of new measurements, uncertainty in the state estimate can only grow over time. Before the correction step can be given, two intermediate quantities are defined,

$$\begin{aligned} S_t &= HP_{t|t-1}H^\top + R, \\ K_t &= P_{t|t-1}H^\top S_t^{-1}, \end{aligned} \quad (2.10)$$

where S_t is the innovation covariance, with the innovation $\nu_t = y_t - H\hat{x}_{t|t-1}$ denoting the discrepancy between the actual and predicted measurement, and K_t is the so-called Kalman gain. Using these definitions, the correction step is given by

$$\begin{aligned} \hat{x}_{t|t} &= \hat{x}_{t|t-1} + K_t(y_t - H\hat{x}_{t|t-1}), \\ P_{t|t} &= P_{t|t-1} - K_tHP_{t|t-1}, \end{aligned} \quad (2.11)$$

which updates the predicted state and covariance using the new measurement y_t . Since the correction step incorporates new information, the resulting covariance $P_{t|t}$ is reduced relative to $P_{t|t-1}$, with the magnitude of this reduction governed by the Kalman gain K_t . A larger gain, corresponding to a more reliable measurement relative to the current prediction uncertainty, results in a larger reduction in uncertainty.

Due to the non-linear mapping of the azimuth and elevation angle measurements introduced in Section 2.1 to Cartesian coordinates, the standard KF cannot be used, since it assumes a linear measurement model. The Extended Kalman Filter (EKF) is introduced next to address this non-linearity in the measurement update.

2.2.2 Extended Kalman Filter (EKF)

This work considers the non-linear system given in Equation 2.3, where the measurement model is non-linear while the state transition remains linear. Because only the measurement model is non-linear, the prediction step of the EKF is identical to that of the linear KF,

$$\begin{aligned} \hat{x}_{t|t-1} &= F\hat{x}_{t-1|t-1}, \\ P_{t|t-1} &= FP_{t-1|t-1}F^\top + Q. \end{aligned} \quad (2.12)$$

What changes is the measurement update. The measurements considered in this work are the azimuth (φ_i) and elevation (θ_i) angles. The same $t|t-1$ conditioning convention introduced in subsection 2.2.1 applies here, with the non-linear measurement z_t taking the place of y_t . The measurement model is given by

$$z_t = [\varphi_i \ \theta_i]^\top + v_t = h(x_t) + v_t, \quad (2.13)$$

where the non-linear function $h(x_t)$ was already introduced in Equation 2.7, with x_t the state vector as described in Equation 2.1.

The EKF approximates the non-linear function $h(x_t)$ using a first-order Taylor expansion around the current predicted state $\hat{x}_{t|t-1}$. This linearisation replaces the constant measurement matrix H used in the linear KF (Equation 2.2) with the Jacobian matrix $H_t \in \mathbb{R}^{n_z \times n}$, which is evaluated locally at $\hat{x}_{t|t-1}$ and therefore has to be recomputed at every time step,

$$H_t = \left. \frac{\partial h}{\partial x_t} \right|_{\hat{x}_{t|t-1}}. \quad (2.14)$$

Because the linearisation in Equation 2.14 is only locally accurate, the EKF can diverge when $\hat{x}_{t|t-1}$ is a poor estimate of the true state, a risk that is especially pronounced during initialisation, i.e. for $\hat{x}_{1|0}$.

Before the update equations of the EKF are given, the innovation is reintroduced. The innovation represents the discrepancy between the actual measurement z_t and the measurement predicted from the current state estimate, $h(\hat{x}_{t|t-1})$, and is given by

$$\nu_t = z_t - h(\hat{x}_{t|t-1}). \quad (2.15)$$

As in subsection 2.2.1, the innovation covariance S_t and Kalman gain K_t are defined, now using the Jacobian H_t from Equation 2.14 in place of the constant H ,

$$\begin{aligned} S_t &= H_t P_{t|t-1} H_t^\top + R, \\ K_t &= P_{t|t-1} H_t^\top S_t^{-1}. \end{aligned} \quad (2.16)$$

Using the Jacobian H_t and the innovation (Equation 2.15) written explicitly, the update equations of the EKF are given by

$$\begin{aligned} \hat{x}_{t|t} &= \hat{x}_{t|t-1} + K_t (z_t - h(\hat{x}_{t|t-1})), \\ P_{t|t} &= P_{t|t-1} - K_t H_t P_{t|t-1}. \end{aligned} \quad (2.17)$$

2.3 Extended Information Filter (EIF)

The EKF can also be written in the so-called information form, referred to as the Extended Information Filter (EIF) [7]. Using the EIF, as will be shown in this section, multiple independent measurements will contribute additively, which is particularly advantageous for a distributed implementation. To derive the information form, the Woodbury matrix inversion lemma [8] is applied to the covariance update equation of the EKF,

$$\begin{aligned} P_{t|t} &= P_{t|t-1} - P_{t|t-1} H_t^\top (H_t P_{t|t-1} H_t^\top + R)^{-1} H_t P_{t|t-1}, \\ P_{t|t} &= \left(P_{t|t-1}^{-1} + H_t^\top R^{-1} H_t \right)^{-1}. \end{aligned} \quad (2.18)$$

This motivates defining the information matrix

$$\Xi_{t|t} = P_{t|t}^{-1}. \quad (2.19)$$

To obtain the corresponding information vector, the state update equation of the EKF is first restated,

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + P_{t|t-1} H_t^\top (H_t P_{t|t-1} H_t^\top + R)^{-1} (z_t - h(\hat{x}_{t|t-1})). \quad (2.20)$$

Multiplying this equation by $\Xi_{t|t}$ motivates defining the information vector

$$\xi_{t|t} = \Xi_{t|t} \hat{x}_{t|t} = P_{t|t}^{-1} \hat{x}_{t|t}, \quad (2.21)$$

which, with the full derivation given in Appendix A, yields the following update,

$$\xi_{t|t} = \xi_{t|t-1} + H_t^\top R^{-1} (z_t - h(\hat{x}_{t|t-1}) + H_t \hat{x}_{t|t-1}). \quad (2.22)$$

The quantities $\Xi_{t|t}$ and $\xi_{t|t}$ can also be written as an update relative to their previous values $\Xi_{t|t-1}$ and $\xi_{t|t-1}$, plus an additive factor,

$$\begin{aligned}\Xi_{t|t} &= \Xi_{t|t-1} + I_t, \\ \xi_{t|t} &= \xi_{t|t-1} + i_t,\end{aligned}\tag{2.23}$$

where the additive factor I_t , referred to as the measurement information matrix, depends only on the predictions of the EKF, while i_t , the measurement information vector, additionally depends on the measurement z_t . Using H_t as defined in Equation 2.14, these factors are given by

$$\begin{aligned}I_t &= H_t^\top R^{-1} H_t, \\ i_t &= H_t^\top R^{-1} (z_t - h(\hat{x}_{t|t-1}) + H_t \hat{x}_{t|t-1}).\end{aligned}\tag{2.24}$$

When multiple independent measurements are available, the update reduces to a summation over all measurements, following from the block-diagonal structure of R^{-1} . When new information is added to the model, this corresponds to a simple addition in the information form,

$$\Xi_{t|t} = \Xi_{t|t-1} + \begin{bmatrix} H_{1,t}^\top & H_{2,t}^\top & \dots & H_{i,t}^\top \end{bmatrix} \begin{bmatrix} R^{-1} & 0 & \dots & 0 \\ 0 & R^{-1} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & R^{-1} \end{bmatrix} \begin{bmatrix} H_{1,t} \\ H_{2,t} \\ \dots \\ H_{i,t} \end{bmatrix}.\tag{2.25}$$

Summing over the measurements from all N sensor nodes introduced in Section 2.1 yields the following update equations for both the information matrix and the information vector,

$$\begin{aligned}\Xi_{t|t} &= \Xi_{t|t-1} + \sum_{i=1}^N H_{i,t}^\top R^{-1} H_{i,t}, \\ \xi_{t|t} &= \xi_{t|t-1} + \sum_{i=1}^N H_{i,t}^\top R^{-1} (z_{i,t} - h(\hat{x}_{t|t-1}) + H_{i,t} \hat{x}_{t|t-1}).\end{aligned}\tag{2.26}$$

This additive structure allows the EIF to fuse several independent measurements without recomputing a matrix inverse for each one, making it particularly well-suited for a distributed implementation. The state and covariance can easily be retrieved by rewriting Equation 2.19 and Equation 2.21,

$$\begin{aligned}P_{t|t} &= \Xi_{t|t}^{-1}, \\ \hat{x}_{t|t} &= P_{t|t} \xi_{t|t}.\end{aligned}\tag{2.27}$$

This leads back to the normal state-space notation introduced in Section 2.1.

2.4 Multi-Target Tracking

The formulations discussed above all implicitly considered a single-target scenario, while most real-world scenarios involve multiple targets in the area of interest. This section discusses the changes to the notation, as well as methods to select the correct measurement for each target.

2.4.1 Multi-Target Notation

When considering multiple targets, measurements must be associated with the track they originate from, where a track denotes the state maintained by the tracking algorithm to represent an individual target over time. The selected measurement is denoted by $m \in \{1, \dots, M\}$, where M denotes the number of measurements per sensor. The selected track is denoted by $\ell \in \{1, \dots, L\}$, where L denotes the number of tracks.

If $m = \ell$, the selected measurement originates from the corresponding track. If $m \neq \ell$, the measurement originates from another track or is a false positive. False positives and missed detections are not considered in this work. Therefore, every track produces exactly one measurement per sensor, and every measurement originates from a track, giving $M = L$. The measurement model for a correct measurement-to-track association is thus given by,

$$z_{m,t} = h(x_{\ell,t}) + v_t. \quad (2.28)$$

It is not known, however, which measurement corresponds to which track. A method used for measurement-to-track association, called gating, is discussed in the next subsection.

2.4.2 Gating

A commonly used method to associate measurements with targets is called gating [9]. Gating establishes a statistical threshold, admitting only measurements that are sufficiently likely to originate from a given track. This is achieved by computing the Mahalanobis distance between a measurement $z_{m,t}$ and the predicted measurement $h(\hat{x}_{\ell,t|t-1})$, using the inverse of the innovation covariance matrix $S_{\ell,t}$, defined as in Equation 2.16 but now evaluated per track using the Jacobian $H_{\ell,t}$. The innovation is defined as

$$\nu_{m,\ell,t} = z_{m,t} - h(\hat{x}_{\ell,t|t-1}), \quad (2.29)$$

and the Mahalanobis distance is defined as

$$D^2 = \nu_{m,\ell,t}^\top S_{\ell,t}^{-1} \nu_{m,\ell,t}. \quad (2.30)$$

For each track ℓ , the measurement yielding the smallest Mahalanobis distance is selected,

$$\hat{m} = \arg \min_{m \in \{1, \dots, M\}} \nu_{m,\ell,t}^\top S_{\ell,t}^{-1} \nu_{m,\ell,t}, \quad (2.31)$$

which is a nearest-neighbour association in the Mahalanobis metric. An upper bound on this distance can additionally be imposed using a $\chi_{n_z}^2$ distribution, because $z_{m,t}$ is assumed to be a Gaussian random vector, i.e. $z_{m,t} \sim \mathcal{N}(h(\hat{x}_{\ell,t|t-1}), S_{\ell,t})$. This is shown below by first deriving the mean of the innovation $\nu_{m,\ell,t}$, followed by its covariance.

Linearising $h(x_{\ell,t})$ around $\hat{x}_{\ell,t|t-1}$ gives $h(x_{\ell,t}) \approx h(\hat{x}_{\ell,t|t-1}) + H_{\ell,t}(x_{\ell,t} - \hat{x}_{\ell,t|t-1})$. Substituting this into the measurement model and the definition of the innovation gives

$$\nu_{m,\ell,t} = z_{m,t} - h(\hat{x}_{\ell,t|t-1}) \approx h(\hat{x}_{\ell,t|t-1}) + H_{\ell,t}(x_{\ell,t} - \hat{x}_{\ell,t|t-1}) + v_t - h(\hat{x}_{\ell,t|t-1}), \quad (2.32)$$

where the two $h(\hat{x}_{\ell,t|t-1})$ terms cancel, leaving

$$\nu_{m,\ell,t} = H_{\ell,t}(x_{\ell,t} - \hat{x}_{\ell,t|t-1}) + v_t. \quad (2.33)$$

Taking the expectation gives

$$E[\nu_{m,\ell,t}|\hat{x}_{\ell,t|t-1}] = H_{\ell,t}E[x_{\ell,t} - \hat{x}_{\ell,t|t-1}] + E[v_t]. \quad (2.34)$$

Because $x_{\ell,t} \sim \mathcal{N}(\hat{x}_{\ell,t|t-1}, P_{\ell,t|t-1})$ and v_t is zero-mean Gaussian noise, both terms vanish, leaving

$$E[\nu_{m,\ell,t}|\hat{x}_{\ell,t|t-1}] = 0, \quad (2.35)$$

confirming that the innovation, and hence $z_{m,t}$, has mean $h(\hat{x}_{\ell,t|t-1})$, as required. Next, the covariance of the innovation is derived,

$$\begin{aligned} E[\nu_{m,\ell,t}\nu_{m,\ell,t}^\top|\hat{x}_{\ell,t|t-1}] &= E\left[(H_{\ell,t}(x_{\ell,t} - \hat{x}_{\ell,t|t-1}) + v_t)(H_{\ell,t}(x_{\ell,t} - \hat{x}_{\ell,t|t-1}) + v_t)^\top\right] \\ &= H_{\ell,t}E[(x_{\ell,t} - \hat{x}_{\ell,t|t-1})(x_{\ell,t} - \hat{x}_{\ell,t|t-1})^\top]H_{\ell,t}^\top + E[v_tv_t^\top] \\ &= H_{\ell,t}P_{\ell,t|t-1}H_{\ell,t}^\top + R = S_{\ell,t}, \end{aligned} \quad (2.36)$$

where the cross terms vanish since v_t is independent of the state prediction error and has zero mean.

Now that the innovation is shown to be distributed as $\nu_{m,\ell,t} \sim \mathcal{N}(0, S_{\ell,t})$, the relation between the Mahalanobis distance and the χ^2 distribution can be established using a whitening transform. Define $\tilde{\nu}_{m,\ell,t} = S_{\ell,t}^{-1/2}\nu_{m,\ell,t}$. Since $\nu_{m,\ell,t}$ is Gaussian and $S_{\ell,t}^{-1/2}$ is a linear transformation, $\tilde{\nu}_{m,\ell,t}$ is also Gaussian, with mean $S_{\ell,t}^{-1/2} \cdot 0 = 0$ and covariance $S_{\ell,t}^{-1/2}S_{\ell,t}(S_{\ell,t}^{-1/2})^\top = I$, so that $\tilde{\nu}_{m,\ell,t} \sim \mathcal{N}(0, I)$. Using $(S_{\ell,t}^{-1/2})^\top S_{\ell,t}^{-1/2} = S_{\ell,t}^{-1}$, the Mahalanobis distance from Equation 2.30 can be rewritten in terms of $\tilde{\nu}_{m,\ell,t}$,

$$D^2 = \nu_{m,\ell,t}^\top S_{\ell,t}^{-1} \nu_{m,\ell,t} = \tilde{\nu}_{m,\ell,t}^\top \tilde{\nu}_{m,\ell,t}. \quad (2.37)$$

Since $\tilde{\nu}_{m,\ell,t} \sim \mathcal{N}(0, I)$, its components are independent, standard normal random variables, so by definition of the χ^2 distribution, $D^2 = \sum_{i=1}^{n_z} [\tilde{\nu}_{m,\ell,t}]_i^2 \sim \chi_{n_z}^2$, showing the relation between the Mahalanobis distance and the χ^2 distribution.

As a consistency check, since a $\chi_{n_z}^2$ -distributed variable has mean n_z , the expectation of D^2 can be verified directly using its original definition,

$$\begin{aligned} E[D^2] &= E[\nu_{m,\ell,t}^\top S_{\ell,t}^{-1} \nu_{m,\ell,t}] \\ &= E[\text{tr}(\nu_{m,\ell,t}^\top S_{\ell,t}^{-1} \nu_{m,\ell,t})] \\ &= E[\text{tr}(S_{\ell,t}^{-1} \nu_{m,\ell,t} \nu_{m,\ell,t}^\top)] \\ &= \text{tr}(S_{\ell,t}^{-1} E[\nu_{m,\ell,t} \nu_{m,\ell,t}^\top]) \\ &= \text{tr}(S_{\ell,t}^{-1} S_{\ell,t}) = n_z, \end{aligned} \quad (2.38)$$

where n_z is the number of dimensions of $z_{m,t}$, confirming the result.

The χ^2 distribution can thus be used as an upper bound to filter out outliers from a Gaussian-distributed random variable. Let $F_{\chi_{n_z}^2}$ denote the Cumulative Distribution

Function (CDF) of the $\chi_{n_z}^2$ distribution. A threshold for a given percentile is then found using its inverse $F_{\chi_{n_z}^2}^{-1}$, the quantile function. This $\chi_{n_z}^2$ result is exact only under the linear-Gaussian assumptions of the KF. For the EKF, both $S_{\ell,t}$ and the assumption that the innovation is normally distributed rely on the first-order linearisation of $h(\cdot)$, so the $\chi_{n_z}^2$ distribution is an approximation that degrades when $h(\cdot)$ is strongly non-linear or the prior uncertainty $P_{\ell,t|t-1}$ is large.

Distributed Model

This chapter presents the conversion of the centralised model introduced in Chapter 2 into a distributed formulation, such that it can be solved without a central node/fusion centre. The chapter is structured as follows: Section 3.1 illustrates, through a simple example, how a centralised estimation problem can be reformulated as a distributed consensus problem using local copies of the variable of interest. Section 3.2 introduces the Primal-Dual Method of Multipliers (PDMM), the distributed solver used throughout this work, along with the notation required to apply it to a general distributed optimisation problem. Building on this, Section 3.3 derives a distributed formulation of the EKF by first casting it as a maximum a posteriori (MAP) estimation problem and then rewriting this as a distributed consensus problem solved using PDMM, referred to as EKF-PDMM. Section 3.4 presents an alternative formulation in information form, EIF-PDMM, which reuses the additive structure of the EIF introduced in Section 2.3. Finally, Section 3.5 compares the convergence behaviour of EKF-PDMM and EIF-PDMM, showing that EIF-PDMM reaches consensus substantially faster while converging to the same result. Based on this, EIF-PDMM is adopted for the remainder of this work.

3.1 General Idea of Distributed Algorithms

The goal of a distributed algorithm is to solve a centralised problem in a distributed fashion. For example, a sensor network is considered in which spatially distributed sensors measure temperature over an area, where the mean temperature of that area is of interest. In a centralised method, all measurements would be gathered and the mean calculated. In this example, a_i denotes the measurement of node i , and the mean value of all measurements, $\bar{a}$, is given by

$$\bar{a} = \frac{1}{N} \sum_{i=1}^N a_i. \quad (3.1)$$

To construct an equation that yields the same average value in a distributed fashion, an auxiliary variable x is introduced. An equivalent problem to Equation 3.1, formulated as an optimisation problem using the auxiliary variable x , is given by

$$\min_x \sum_{i=1}^N \frac{1}{2} (x - a_i)^2. \quad (3.2)$$

Local copies of this auxiliary variable x are introduced for each node, denoted x_i for node i . This ensures node separability. To ensure that every node converges to the

same answer, the problem is constrained such that $x_i = x_j$ for neighbouring nodes, yielding

$$\begin{aligned} \min_{x_1, \dots, x_N} \quad & \sum_{i=1}^N \frac{1}{2} (x_i - a_i)^2, \\ \text{subject to} \quad & x_i - x_j = 0. \end{aligned} \quad (3.3)$$

These are the basic steps needed to rewrite a centralised problem into a distributed optimisation problem. The distributed solver used in this work is discussed next. This solver takes a distributed optimisation problem and handles the communication and consensus process.

3.2 Primal-Dual Method of Multipliers (PDMM)

The Primal-Dual Method of Multipliers (PDMM) algorithm [10] is not as well known as other distributed algorithms such as Alternating-Direction Method of Multipliers (ADMM) or randomized gossip [11]. It is shown in [3] that a 1/2-averaged PDMM is equivalent to ADMM, meaning ADMM is a special case of the more general PDMM framework. Because PDMM's averaging parameter is not restricted to this special case, it can be tuned to achieve a faster convergence rate than ADMM, provided it converges [12].

A minimisation problem is defined, involving a function $f_i(x_i)$ to be minimised in a distributed manner. To do this, a graph $G(\mathcal{V}, \mathcal{E})$ must be defined, where $\mathcal{V} = \{1, \dots, N\}$ denotes the set of all N nodes, and $\mathcal{E}$ denotes the set of undirected edges such that $(i, j) \in \mathcal{E}$ if node i and node j are connected. The set of neighbours of node i is given by $\mathcal{N}_i = \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\}$, i.e. all nodes j that are connected to node i by an edge. The distributed minimisation problem can then be written as

$$\begin{aligned} \text{minimise} \quad & \sum_{i \in \mathcal{V}} f_i(x_i), \\ \text{subject to} \quad & A_{i|j}x_i + A_{j|i}x_j = b_{i,j} \quad \forall (i, j) \in \mathcal{E}. \end{aligned} \quad (3.4)$$

The matrices $A_{i|j} \in \mathbb{R}^{n \times n}$, together with $b_{i,j} \in \mathbb{R}^{n \times 1}$, impose a constraint between the local variables of neighbouring nodes i and j . The notation $i|j$ denotes a directed edge, and i, j denotes an undirected edge. To transform the problem into a consensus problem, $A_{i|j} = -A_{j|i} = I$ and $b_{i,j} = 0$. This implies that every local value must be equal to the value of their neighbours, i.e. $x_i - x_j = 0$. The standard form of PDMM, as derived in [3], is given by

$$\begin{aligned} x^{(k)} &:= \arg \min_x \left(f(x) + (z^{(k)})^\top Cx + \frac{c}{2} \|Cx - \beta\|^2 \right), \\ y^{(k)} &:= z^{(k)} + 2c(Cx^{(k)} - \beta), \\ z^{(k+1)} &:= \Pi y^{(k)}, \end{aligned} \quad (3.5)$$

where Π is a permutation matrix that ensures that $(\forall (i, j) \in \mathcal{E}) \quad z_{j|i} = y_{i|j}$, i.e. it represents the actual data exchange in the network, illustrated in Figure 3.1. $y^{(k)}$

and $z^{(k)}$ denote auxiliary node variables, and c denotes the step size that controls the convergence rate. The index k denotes a PDMM iteration, in which every node shares its values with all its neighbours once. A broadcast method for communication is assumed in this work. PDMM is used as a synchronous algorithm. An asynchronous variant does exist [10] but it is not used in this work.

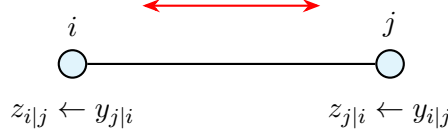


Figure 3.1: Information exchange between neighbouring nodes i and j , node i receives data $y_{j|i}$ from j and treats it as $z_{i|j}$, while node j receives $y_{i|j}$ from i and treats it as $z_{j|i}$.

The constant matrix C and vector β are used to rewrite the objective. The components of C , denoted by C_i , and the components of β , denoted by β_i , are given by,

$$C = \begin{bmatrix} C_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & C_N \end{bmatrix}, \beta = [\beta_1^\top \quad \dots \quad \beta_N^\top]^\top, \quad (3.6)$$

and the components themselves are given by

$$\begin{aligned} C_i &= [A_{i|1}^\top \quad \dots \quad A_{i|i-1}^\top \quad A_{i|i+1}^\top \quad \dots \quad A_{i|N}^\top]^\top \forall i \in \mathcal{V}, \\ \beta_i &= \frac{1}{2} [b_{i,1}^\top \quad \dots \quad b_{i,i-1}^\top \quad b_{i,i+1}^\top \quad \dots \quad b_{i,N}^\top]^\top \forall i \in \mathcal{V}. \end{aligned} \quad (3.7)$$

The terms $A_{i|j}$ and $b_{i,j}$ are only included in C_i and β_i if and only if $(i, j) \in \mathcal{E}$. To rewrite this standard notation of PDMM such that its distributed nature is easier to recognise, the structure of C can be exploited in the following two equations [3],

$$z^\top Cx = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} z_{i|j}^\top A_{i|j} x_i, \quad (3.8)$$

$$\|Cx - \beta\|^2 = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} \|A_{i|j} x_i - \frac{1}{2} b_{i,j}\|^2. \quad (3.9)$$

Due to the separability of these two equations over the nodes, this results in the node-specific PDMM update equations

$$\begin{aligned} x_i^{(k)} &:= \arg \min_{x_i} \left(f_i(x_i) + \sum_{j \in \mathcal{N}_i} \left(z_{i|j}^{(k)\top} A_{i|j} x_i + \frac{c}{2} \|A_{i|j} x_i - \frac{1}{2} b_{i,j}\|^2 \right) \right), \\ y_{i|j}^{(k)} &:= z_{i|j}^{(k)} + 2c(A_{i|j} x_i^{(k)} - \frac{1}{2} b_{i,j}), \\ z_{j|i}^{(k+1)} &:= y_{i|j}^{(k)}. \end{aligned} \quad (3.10)$$

These node-specific update equations form the basis for applying PDMM to the distributed EKF, discussed next.

3.3 Extended Kalman Filter – Primal-Dual Method of Multipliers

This section derives a distributed formulation of the EKF, building on [13], where the update equations for a KF solved by ADMM are obtained by first rewriting the problem as a maximum a posteriori (MAP) estimation problem. This formulation is extended to the EKF, using PDMM as the distributed solver.

3.3.1 Maximum a Posteriori Formulation for the EKF

This subsection shows how a centralised EKF can be solved as a MAP estimation problem. This formulation is then extended to a distributed notation, from which the PDMM update equations for EKF-PDMM are derived in subsection 3.3.2.

For the MAP problem, an x_t that maximises the posterior distribution $p(x_t|z_{1:t})$ is desired, where $z_{1:t}$ represents the measurements from time 1 to the current time t . By applying Bayes' rule, the MAP estimator can be defined as

$$\hat{x}_{t|t} = \arg \max_{x_t} p(x_t|z_{1:t}) = \arg \max_{x_t} p(z_t|x_t)p(x_t|z_{1:t-1}), \quad (3.11)$$

which can also be written as

$$\hat{x}_{t|t} = \arg \max_{x_t} \ln(p(z_t|x_t)p(x_t|z_{1:t-1})), \quad (3.12)$$

where $p(z_t|x_t)$ is the likelihood function, given by $\mathcal{N}(z_t; h(x_t), R)$, evaluated at z_t , with mean $h(x_t)$ and covariance R . Similarly, $p(x_t|z_{1:t-1})$ is the prior distribution, given by $\mathcal{N}(x_t; \hat{x}_{t|t-1}, P_{t|t-1})$, evaluated at x_t , with mean $\hat{x}_{t|t-1}$ and covariance $P_{t|t-1}$. Substituting these two distributions gives

$$\ln p(z_t|x_t) = -\frac{1}{2}(z_t - h(x_t))^\top R^{-1}(z_t - h(x_t)) - \frac{1}{2} \ln((2\pi)^{n_z} \det(R)), \quad (3.13)$$

$$\ln p(x_t|z_{1:t-1}) = -\frac{1}{2}(\hat{x}_{t|t-1} - x_t)^\top P_{t|t-1}^{-1}(\hat{x}_{t|t-1} - x_t) - \frac{1}{2} \ln((2\pi)^n \det(P_{t|t-1})), \quad (3.14)$$

where n_z is the number of measurements contained in z_t , and n is the number of states in x_t . Since a derivative will be taken with respect to x_t to find $\hat{x}_{t|t}$, the last term in each expression can be dropped, as it does not depend on x_t . In Equation 3.13, the non-linear function $h(x_t)$ appears. Since x_t is unknown at this point, this must be approximated. The EKF approximates this by linearising $h(x_t) \approx h(\hat{x}_{t|t-1}) + H_t(x_t - \hat{x}_{t|t-1})$, where H_t is again given by Equation 2.14. Using this approximation, and writing the maximisation as a minimisation to drop the minus sign, gives

$$\begin{aligned} \hat{x}_{t|t} = \arg \min_{x_t} & \left(\frac{1}{2}(\hat{x}_{t|t-1} - x_t)^\top P_{t|t-1}^{-1}(\hat{x}_{t|t-1} - x_t) \right. \\ & + \frac{1}{2} \left(z_t - (h(\hat{x}_{t|t-1}) + H_t(x_t - \hat{x}_{t|t-1})) \right)^\top R^{-1} \\ & \left. \left(z_t - (h(\hat{x}_{t|t-1}) + H_t(x_t - \hat{x}_{t|t-1})) \right) \right). \end{aligned} \quad (3.15)$$

This is not the exact MAP estimator of the true, non-linear estimation problem, since the linearisation of $h(x_t)$ has already been applied. It is, however, exactly the same approximation used by the EKF itself, so solving Equation 3.15 reproduces the EKF update exactly. This problem can be rewritten to be solved in a distributed fashion by PDMM. To use PDMM, Equation 3.15 has to be written in the form of Equation 3.4. First, multiple independent measurements from N nodes are considered. All measurements are stacked in the vector $z_{1:N,t} = [z_{1,t} \ \dots \ z_{N,t}]^\top$, with $h(\cdot)$ and H_t similarly stacked. R^{-1} is in block-diagonal form, which holds because the measurements, and their associated noise, are assumed independent. Using this form, the measurement term of Equation 3.15 can also be written as a summation over all nodes,

$$\begin{aligned} & \frac{1}{2} (z_{1:N,t} - (h(\hat{x}_{t|t-1}) + H_t(x_t - \hat{x}_{t|t-1})))^\top R^{-1} (z_{1:N,t} - (h(\hat{x}_{t|t-1}) + H_t(x_t - \hat{x}_{t|t-1}))) \\ &= \sum_{i=1}^N \frac{1}{2} (z_{i,t} - (h_i(\hat{x}_{t|t-1}) + H_{i,t}(x_t - \hat{x}_{t|t-1})))^\top R_i^{-1} (z_{i,t} - (h_i(\hat{x}_{t|t-1}) + H_{i,t}(x_t - \hat{x}_{t|t-1}))) . \end{aligned} \quad (3.16)$$

The measurement part of the problem is now almost written in a node-separable form. The state vector x_t and the predicted state vector $\hat{x}_{t|t-1}$ are not yet node-separable. By making local copies of x_t , the predicted state can similarly be represented by its local copy $\hat{x}_{i,t|t-1}$. Due to the local copies, a factor $1/N$ is needed to scale the summation, giving

$$\begin{aligned} & \frac{1}{2} (\hat{x}_{t|t-1} - x_t)^\top P_{t|t-1}^{-1} (\hat{x}_{t|t-1} - x_t) \\ &= \sum_{i=1}^N \frac{1}{2N} (\hat{x}_{i,t|t-1} - x_{i,t})^\top P_{i,t|t-1}^{-1} (\hat{x}_{i,t|t-1} - x_{i,t}) . \end{aligned} \quad (3.17)$$

3.3.2 Update Equations for EKF-PDMM

The local, node-specific version of the objective from Equation 3.15 is now written using the local copy $x_{i,t}$, following the local-copy construction introduced in Section 3.1,

$$\begin{aligned} f_i(x_{i,t}) &= \frac{1}{2N} (\hat{x}_{i,t|t-1} - x_{i,t})^\top P_{i,t|t-1}^{-1} (\hat{x}_{i,t|t-1} - x_{i,t}) \\ &+ \frac{1}{2} (z_{i,t} - (h_i(\hat{x}_{i,t|t-1}) + H_{i,t}(x_{i,t} - \hat{x}_{i,t|t-1})))^\top R_i^{-1} \\ &\quad (z_{i,t} - (h_i(\hat{x}_{i,t|t-1}) + H_{i,t}(x_{i,t} - \hat{x}_{i,t|t-1}))) . \end{aligned} \quad (3.18)$$

Summing the local objective $f_i(x_{i,t})$ over all nodes gives the node-separable objective,

$$\begin{aligned} f(x_{1,t}, \dots, x_{N,t}) &= \sum_{i=1}^N \left[\frac{1}{2N} (\hat{x}_{i,t|t-1} - x_{i,t})^\top P_{i,t|t-1}^{-1} (\hat{x}_{i,t|t-1} - x_{i,t}) \right. \\ &+ \frac{1}{2} (z_{i,t} - (h_i(\hat{x}_{i,t|t-1}) + H_{i,t}(x_{i,t} - \hat{x}_{i,t|t-1})))^\top R_i^{-1} \\ &\quad \left. (z_{i,t} - (h_i(\hat{x}_{i,t|t-1}) + H_{i,t}(x_{i,t} - \hat{x}_{i,t|t-1}))) \right] , \end{aligned} \quad (3.19)$$

subject to the consensus constraint,

$$\begin{aligned} \min_{x_{1,t} \dots x_{N,t}} \quad & \sum_{i=1}^N f_i(x_{i,t}), \\ \text{subject to} \quad & (\forall (i, j) \in \mathcal{E}) \quad x_{i,t} - x_{j,t} = 0. \end{aligned} \quad (3.20)$$

To find the PDMM update equations for the state, the derivative with respect to $x_{i,t}$ is taken and equated to zero,

$$\begin{aligned} 0 = & -H_{i,t}^\top R_i^{-1} (z_{i,t} - (h(\hat{x}_{i,t|t-1}) + H_{i,t}(x_{i,t} - \hat{x}_{i,t|t-1}))) \\ & - \frac{1}{N} P_{i,t|t-1}^{-1} (\hat{x}_{i,t|t-1} - x_{i,t}) + \sum_{j \in \mathcal{N}_i} (A_{i|j}^\top z_{i|j} + c A_{i|j}^\top A_{i|j} x_{i,t}). \end{aligned} \quad (3.21)$$

Since Equation 3.20 is a consensus problem, $A_{i|j} = -A_{j|i} = I$, so $A_{i|j}^\top A_{i|j} = I$, which implies $\sum_{j \in \mathcal{N}_i} c A_{i|j}^\top A_{i|j} x_{i,t} = c d_i x_{i,t}$, where d_i is the degree of node i . This equation can now be solved for $x_{i,t}$,

$$\begin{aligned} H_{i,t}^\top R_i^{-1} H_{i,t} x_{i,t} + \frac{1}{N} P_{i,t|t-1}^{-1} x_{i,t} + c d_i x_{i,t} &= H_{i,t}^\top R_i^{-1} (z_{i,t} - h(\hat{x}_{i,t|t-1}) + H_{i,t} \hat{x}_{i,t|t-1}) \\ &+ \frac{1}{N} P_{i,t|t-1}^{-1} \hat{x}_{i,t|t-1} - \sum_{j \in \mathcal{N}_i} A_{i|j}^\top z_{i|j}, \\ x_{i,t} &= \left(H_{i,t}^\top R_i^{-1} H_{i,t} + \frac{1}{N} P_{i,t|t-1}^{-1} + c d_i I \right)^{-1} \\ &\left(H_{i,t}^\top R_i^{-1} (z_{i,t} - h(\hat{x}_{i,t|t-1}) + H_{i,t} \hat{x}_{i,t|t-1}) + \frac{1}{N} P_{i,t|t-1}^{-1} \hat{x}_{i,t|t-1} - \sum_{j \in \mathcal{N}_i} A_{i|j}^\top z_{i|j} \right). \end{aligned} \quad (3.22)$$

Combining this result with the general PDMM update equations gives the following update equations for the state vector,

$$\begin{aligned} \hat{x}_{i,t|t}^{(k)} &= \left(H_{i,t}^\top R_i^{-1} H_{i,t} + \frac{1}{N} P_{i,t|t-1}^{-1} + c d_i I \right)^{-1} \\ &\left(H_{i,t}^\top R_i^{-1} (z_{i,t} - h(\hat{x}_{i,t|t-1}) + H_{i,t} \hat{x}_{i,t|t-1}) + \frac{1}{N} P_{i,t|t-1}^{-1} \hat{x}_{i,t|t-1} - \sum_{j \in \mathcal{N}_i} A_{i|j}^\top z_{i|j} \right), \\ y_{i|j}^{(k)} &= z_{i|j}^{(k)} + 2c A_{i|j} \hat{x}_{i,t|t}^{(k)}, \\ z_{j|i}^{(k+1)} &= y_{i|j}^{(k)}. \end{aligned} \quad (3.23)$$

For the update equation for the covariance, the minimisation problem as defined in [13] can be used. This definition already uses the information form of the covariance, which is why it reduces to the ℓ_2 -norm minimisation problem

$$\begin{aligned} \min_{\Xi_{i,t}} \quad & \frac{1}{2} \sum_{i=1}^N \|N H_{i,t}^\top R_i^{-1} H_{i,t} - \Xi_{i,t}\|^2, \\ \text{subject to} \quad & (\forall (i, j) \in \mathcal{E}) \quad \Xi_{i,t} - \Xi_{j,t} = 0. \end{aligned} \quad (3.24)$$

The difference between this formulation and that of [13] is that the matrix $H_{i,t}^\top R_i^{-1} H_{i,t}$ is not half-vectorised here. In [13], half-vectorisation is done to reduce the amount of data transmitted in the ADMM implementation. Since reducing this data is not a goal of this work, half-vectorisation is omitted here. The derivation of the primal update of the measurement information matrix is obtained by taking the derivative of the problem described in Equation 3.24, combined with Equation 3.10, with respect to $\Xi_{i,t}$, which gives

$$\begin{aligned} 0 &= \Xi_{i,t} - N H_{i,t}^\top R_i^{-1} H_{i,t} + \sum_{j \in \mathcal{N}_i} (A_{i|j}^\top Z_{i|j} + c A_{i|j}^\top A_{i|j} \Xi_{i,t}), \\ (I + c d_i I) \Xi_{i,t} &= N H_{i,t}^\top R_i^{-1} H_{i,t} - \sum_{j \in \mathcal{N}_i} A_{i|j}^\top Z_{i|j}, \\ \Xi_{i,t} &= (I + c d_i I)^{-1} \left(N H_{i,t}^\top R_i^{-1} H_{i,t} - \sum_{j \in \mathcal{N}_i} A_{i|j}^\top Z_{i|j} \right). \end{aligned} \quad (3.25)$$

This leads to the PDMM update equations for the measurement information matrix,

$$\begin{aligned} \Xi_{i,t}^{(k)} &= (I + c d_i I)^{-1} \left(N H_{i,t}^\top R_i^{-1} H_{i,t} - \sum_{j \in \mathcal{N}_i} A_{i|j}^\top Z_{i|j} \right), \\ Y_{i|j}^{(k)} &= Z_{i|j}^{(k)} + 2c A_{i|j} \Xi_{i,t}^{(k)}, \\ Z_{j|i}^{(k+1)} &= Y_{i|j}^{(k)}. \end{aligned} \quad (3.26)$$

3.4 Extended Information Filter – Primal-Dual Method of Multipliers

The EKF and its information form counterpart, the EIF, were shown in Chapter 2 to be equivalent representations of the same posterior, related through Equation 2.19 and Equation 2.21. Because of this equivalence, the MAP formulation of the distributed EKF problem derived in Section 3.3 can equally be expressed in information form, using the additive structure given in Equation 2.26 to formulate an alternative consensus problem. Only the local, N -scaled measurement information matrix $\Xi_{i,t}$ and measurement information vector $\xi_{i,t}$ need to be shared. These are the distributed analogues of I_t and i_t from Equation 2.24, excluding the prior. The prior is reintroduced when recovering the state and covariance, given by

$$\begin{aligned} P_{i,t|t} &= \left(P_{i,t|t-1}^{-1} + \Xi_{i,t} \right)^{-1}, \\ \hat{x}_{i,t|t} &= P_{i,t|t} \left(P_{i,t|t-1}^{-1} \hat{x}_{i,t|t-1} + \xi_{i,t} \right). \end{aligned} \quad (3.27)$$

The distributed minimisation for the measurement information vector is given by

$$\begin{aligned} \min_{\xi_{i,t}} \quad & \frac{1}{2} \sum_{i=1}^N \|N(H_{i,t}^\top R_i^{-1} (z_{i,t} - h(\hat{x}_{i,t|t-1}) + H_{i,t} \hat{x}_{i,t|t-1})) - \xi_{i,t}\|^2, \\ \text{subject to} \quad & (\forall (i, j) \in \mathcal{E}) \quad \xi_{i,t} - \xi_{j,t} = 0. \end{aligned} \quad (3.28)$$

The distributed optimisation problem for the measurement information vector $\xi_{i,t}$ is derived analogously to that of the measurement information matrix $\Xi_{i,t}$ in subsection 3.3.2, and only the resulting minimisation problem is stated here. The resulting PDMM update equations for the measurement information vector are given by

$$\begin{aligned}\xi_{i,t}^{(k)} &= (I + cd_i I)^{-1} \left(NH_{i,t}^\top R_i^{-1} (z_{i,t} - h(\hat{x}_{i,t|t-1}) + H_{i,t} \hat{x}_{i,t|t-1}) - \sum_{j \in \mathcal{N}_i} A_{i|j}^\top z_{i|j} \right), \\ y_{i|j}^{(k)} &= z_{i|j}^{(k)} + 2c A_{i|j} \xi_{i,t}^{(k)}, \\ z_{j|i}^{(k+1)} &= y_{i|j}^{(k)}.\end{aligned}\tag{3.29}$$

The measurement information matrix update equations are unchanged from Equation 3.26.

3.5 Convergence Comparison: EKF-PDMM vs. EIF-PDMM

The number of PDMM iterations run at each time step t is a design choice: fewer iterations reduce communication overhead but do not attain the same consensus precision, so the iteration count must be chosen according to the precision required by the application.

3.5.1 Experimental Setup

The simulation setup used to obtain the results presented in subsection 3.5.2 is described below. The initial state estimate is given by

$$\hat{x}_{1|0} = Fx_0,\tag{3.30}$$

where x_0 contains only the target's initial position, with the velocity components initialised to zero. The initial covariance is set to $P_0 = \text{diag}(500, 500, 500, 100, 100, 100)$. The process noise spectral density is assumed equal among the x -, y -, and z -axes, i.e. $q_x = q_y = q_z = q$ in Equation 2.6. The remaining simulation parameters are summarised in Table 3.1.

Table 3.1: Simulation parameters used in the convergence comparison.

Parameter	Value
Number of targets, L	1
Number of nodes, N	20
Maximum time, T	20
Sampling interval, δt	0.1
Process noise spectral density, q	0.2
Maximum PDMM iterations	4000

3.5.2 Results

To show that EIF-PDMM needs fewer iterations to achieve consensus in the network compared to EKF-PDMM, the optimal value of c must first be found for both algorithms. This can be done by trying all values of c between 0 and 1 with increments of 0.01.

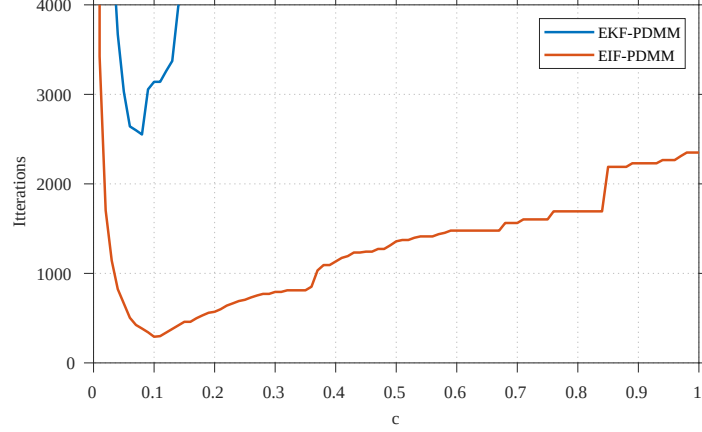


Figure 3.2: Consensus error against PDMM step size c , used to select the optimal c for EKF-PDMM and EIF-PDMM.

Figure 3.2 shows that the optimal c for EKF-PDMM is $c = 0.08$, and for EIF-PDMM is $c = 0.1$. With these c values, the consensus speed of both algorithms can be compared, defined as

$$e_{\text{cons}}(k) = \|\hat{x}_{i,t|t}^{(k)} - \bar{x}_{t|t}^{(k)}\|^2, \quad (3.31)$$

where $\bar{x}_{t|t}^{(k)}$ denotes the average state of all nodes for PDMM-iteration k .

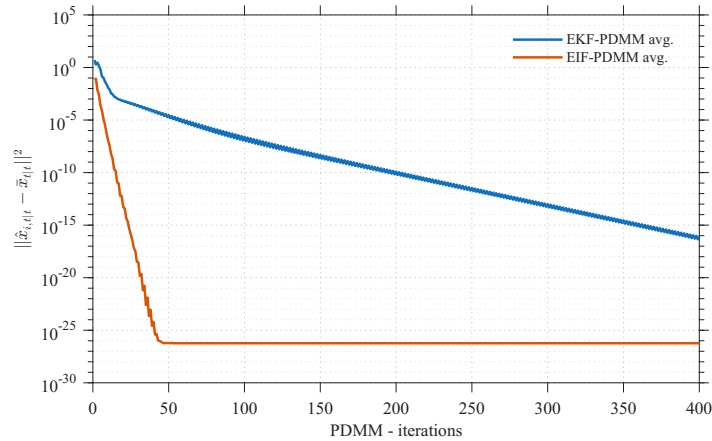


Figure 3.3: Consensus error $e_{\text{cons}}(k)$ against PDMM iteration k at the final time step, for EKF-PDMM and EIF-PDMM. The horizontal axis is truncated at 400 of the 4000 iterations simulated.

The consensus plot in Figure 3.3, truncated at 400 iterations for legibility, shows that the EIF-PDMM implementation needs about 45 PDMM iterations per time step to reach a consensus error of 10^{-26} . The EKF-PDMM implementation, while attaining a low consensus error, does not reach that level within the 400 iterations shown, so any practical iteration budget favours EIF-PDMM. Despite this difference in consensus speed, both algorithms produce the same state estimate. This is evaluated using the state error

$$e_{\text{state}}(t) = \|\hat{x}_{t|t} - x_t\|^2, \quad (3.32)$$

where x_t denotes the true target state at time t , and is shown in Figure 3.4.

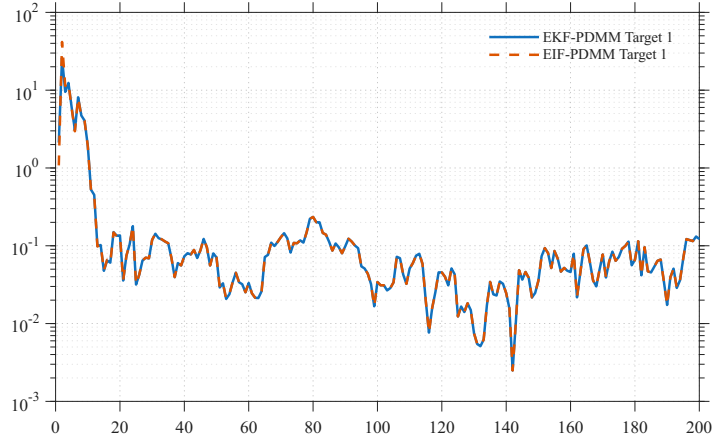


Figure 3.4: State error $e_{\text{state}}(t)$ over time for EKF-PDMM vs. EIF-PDMM.

The EIF-PDMM algorithm can be seen in Algorithm 2 in Appendix B.

This chapter has shown how the centralised model introduced in Chapter 2 can be reformulated to be solved in a distributed fashion. Section 3.1 and Section 3.2 established the general principle: introducing local copies of the variable of interest, subject to a consensus constraint, and solving the resulting problem using PDMM. This principle was then applied to the EKF, first via a direct MAP formulation (EKF-PDMM, Section 3.3), and subsequently via the information form of the EIF (EIF-PDMM, Section 3.4). The comparison in Section 3.5 showed that although both formulations converge to the same result, EIF-PDMM reaches consensus in substantially fewer PDMM iterations than EKF-PDMM. For this reason, EIF-PDMM is used as the distributed filtering algorithm for the remainder of this work. The formulation presented here considers only a single target and assumes every node participates in every update. The next chapter extends EIF-PDMM to multi-target tracking and introduces sensor selection, allowing the algorithm to scale to large sensor networks.

Extension to Multi-Target Tracking and Sensor Selection

4

This chapter extends EIF-PDMM, introduced in Chapter 3, beyond the single-target setting in which every node participates in every update. Two limitations of that formulation are addressed. First, EIF-PDMM provides no mechanism for associating a measurement with the track it originates from, and therefore cannot be applied directly once multiple targets are present. Second, requiring every node to communicate with every other node becomes infeasible as the sensor network grows, since nodes far from a target rarely observe it. The chapter is structured as follows: first, gating is incorporated into EIF-PDMM to resolve measurement-to-track association, yielding EIF-PDMM-MT. Second, sensor selection is introduced to address the scalability problem, yielding EIF-PDMM-SS. Finally, the two extensions are combined into EIF-PDMM-SS-MT, and its performance in a multi-target, large-scale setting is evaluated.

4.1 Multi-Target in EIF-PDMM

The method derived in Section 3.4 provides no mechanism for measurement-to-track association and therefore cannot be applied directly to multi-target scenarios. This section addresses this limitation using gating, the most straightforward method for measurement-to-track association, introduced in subsection 2.4.2 with the association rule given in Equation 2.31. This section shows how gating is incorporated into EIF-PDMM and presents the resulting performance. Applying gating to EIF-PDMM yields EIF-PDMM-MT (Multi-Target), given in Algorithm 1.

As shown in subsection 2.4.2, the χ^2 upper bound imposed on the Mahalanobis distance rejects measurements unlikely to originate from a track, reducing the impact of false positive measurements. This work, however, does not consider false positive measurements, so every measurement is known to originate from a target, and applying this upper bound as an outlier-rejection criterion would therefore only increase the overall estimation error. This follows directly from the additive structure of the EIF. As shown in Equation 2.26, each measurement contributes a positive semi-definite term $H_{i,t}^\top R_i^{-1} H_{i,t} \succeq 0$ (since $R_i \succ 0$) to the information matrix, so $\Xi_{t|t} \succeq \Xi_{t|t-1}$ and, since matrix inversion reverses the positive semi-definite ordering, $P_{t|t} \preceq P_{t|t-1}$. The same argument extends directly to the distributed formulation of Chapter 3, where $P_{i,t|t}^{-1} = P_{i,t|t-1}^{-1} + \Xi_{i,t}$ (Equation 3.27) gives $P_{i,t|t} \preceq P_{i,t|t-1}$ at every node. Discarding a valid measurement can therefore only remove information, never add it, so the χ^2 upper bound is not used here as an outlier-rejection criterion. Gating is instead used solely for measurement-to-track association.

Algorithm 1 EIF-PDMM-MT

```
for  $\ell = 1, \dots, L$  do
  Initialise:  $y, z \in \mathbb{R}^{n \times 1}, Y, Z \in \mathbb{R}^{n \times n}$ 
  for  $k = 1, \dots, \text{maxIter}$  do
    for all  $i \in \mathcal{V}$  do
       $\hat{x}_{\ell,i,t|t-1} = F\hat{x}_{\ell,i,t-1|t-1}$ 
       $P_{\ell,i,t|t-1} = FP_{\ell,i,t-1|t-1}F^\top + Q$ 
       $S_{\ell,i,t} = H_{\ell,i,t}P_{\ell,i,t|t-1}H_{\ell,i,t}^\top + R$ 
      for all  $m = 1, \dots, M$  do
         $\nu_{m,\ell,i,t} = (z_{m,i,t} - h(\hat{x}_{\ell,i,t|t-1}))$ 
         $\hat{m} = \arg \min_{m \in \{1, \dots, M\}} \nu_{m,\ell,i,t}^\top S_{\ell,i,t}^{-1} \nu_{m,\ell,i,t}$ 
         $\xi_{\ell,i,t}^{(k)} = (I + cd_i I)^{-1} \left( NH_{\ell,i,t}^\top R_i^{-1} (z_{\hat{m},i,t} - h(\hat{x}_{\ell,i,t|t-1}) + H_{\ell,i,t} \hat{x}_{\ell,i,t|t-1}) \right. \\ \left. - \sum_{j \in \mathcal{N}_i} A_{i|j}^\top z_{\ell,i|j} \right)$ 
         $\Xi_{\ell,i,t}^{(k)} = (I + cd_i I)^{-1} \left( NH_{\ell,i,t}^\top R_i^{-1} H_{\ell,i,t} - \sum_{j \in \mathcal{N}_i} A_{i|j}^\top Z_{\ell,i|j} \right)$ 
      end for
    end for
  end for
end for
```

4.1.1 Experimental Setup

The simulation setup used to obtain the results presented in subsection 4.1.2 is described below. The initial state estimate is given by

$$\hat{x}_{\ell,1|0} = Fx_{\ell,0}, \quad (4.1)$$

where $x_{\ell,0}$ contains only the target's initial position, with the velocity components initialised to zero. The initial covariance is set to $P_{\ell,0} = \text{diag}(500, 500, 500, 100, 100, 100)$. The process noise spectral density is assumed equal among the x -, y -, and z -axes, i.e. $q_x = q_y = q_z = q$ in Equation 2.6. As found in subsection 3.5.2, $c = 0.1$. The remaining simulation parameters are summarised in Table 4.1.

Table 4.1: Simulation parameters used in the multi-target scenario.

Parameter	Value
Number of targets, L	2
Number of nodes, N	20
Maximum time, T	100
Sampling interval, δt	0.5
Process noise spectral density, q	0.2
Maximum PDMM iterations	100
Angle noise standard deviation, $\sigma_\varphi, \sigma_\theta$	5°

The per-target state error, used to evaluate the results presented in subsection 4.1.2,

is given by

$$e_{\text{state},\ell}(t) = \|\hat{x}_{\ell,t|t} - x_{\ell,t}\|^2, \quad (4.2)$$

where $x_{\ell,t}$ denotes the true state of track ℓ at time t . This is the same error definition used for every error reported throughout this thesis, including the sensor selection results presented later in this chapter.

4.1.2 Results

The mean error per target over 100 Monte Carlo simulations is shown in Figure 4.1, where the node positions are kept fixed across runs and only the measurement noise realisation $v_{\ell,t}$ is varied. The presented algorithm, EIF-PDMM-MT, is compared against its centralised counterpart, EIF-MT. The bottom subplot shows the absolute difference between the two error curves, which remains on the order of 10^{-12} throughout, i.e. at the level of floating-point precision. This confirms that EIF-PDMM-MT converges to the same estimate as the centralised solution and can track multiple targets with no measurable loss of accuracy from distributing the computation.

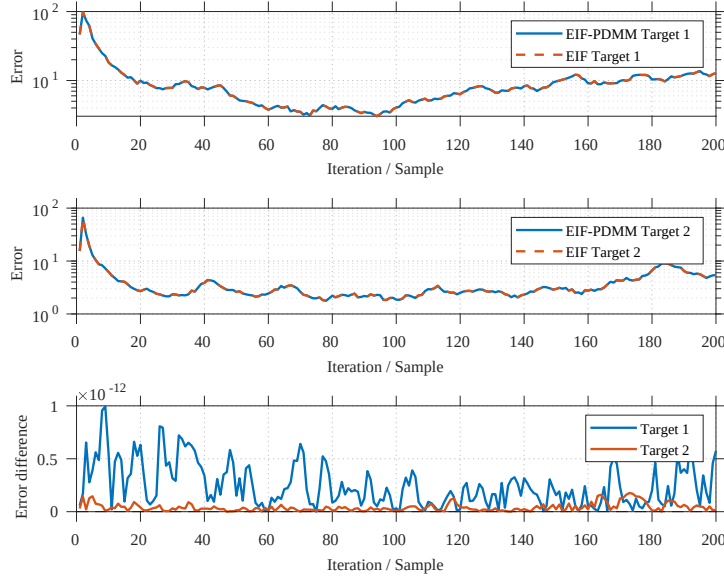


Figure 4.1: Mean state error per target over time for EIF-PDMM-MT and EIF-MT, with their absolute difference shown in the bottom subplot.

The estimated trajectories of EIF-PDMM-MT are shown in three dimensions in Figure 4.2, and projected onto the X-Y and X-Z planes in Figure 4.3. The trajectories of EIF-MT are omitted, since they are visually indistinguishable from those of EIF-PDMM-MT.

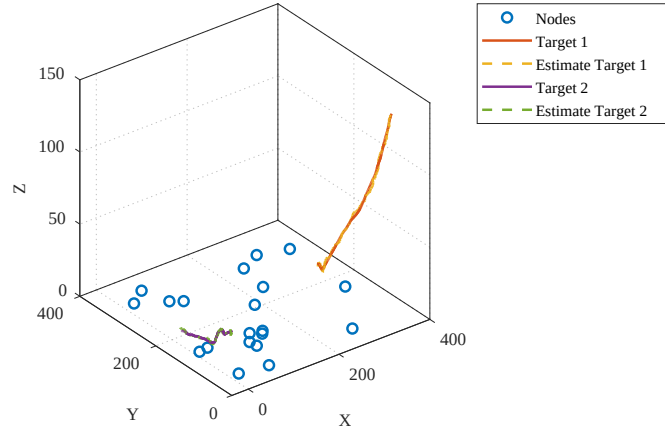


Figure 4.2: Estimated target trajectories in three dimensions, EIF-PDMM-MT.

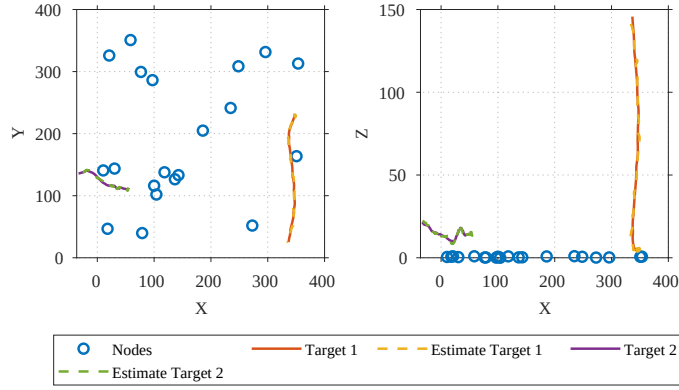


Figure 4.3: Estimated target trajectories projected onto the X-Y and X-Z planes, EIF-PDMM-MT.

The trajectories shown are not of closely spaced targets, since gating cannot correctly associate measurements once tracks are in close proximity.

4.1.3 Improvements

While the results confirm that gating provides adequate measurement-to-track association in this scenario, two extensions are worth considering for future work. First, Joint Probabilistic Data Association (JPDA) extends gating by weighting every candidate measurement within the gate by its likelihood, rather than selecting a single nearest neighbour, and would be expected to improve performance for closely spaced targets where the gates of different tracks overlap. Second, if false positive measurements were introduced, e.g. from clutter, the χ^2 upper bound on the Mahalanobis distance introduced in subsection 2.4.2 would become necessary to discard them. This work's argument against using it as an outlier-rejection criterion relies specifically on

the assumption that every measurement originates from a target, an assumption that no longer holds once false positives are present.

The EIF-PDMM formulation described so far introduces a scalability problem as the area of interest increases. All sensors are still required to communicate with each other, even though sensors that are far apart are unlikely to observe the same target. This problem can be addressed using a method called sensor selection. The method and its implementation are presented in the next section.

4.2 Sensor Selection

In order to make this algorithm feasible for large-scale sensor networks, it cannot be assumed that every node has a measurement of the target. A method is therefore needed to select the nodes that are likely to have a measurement of the target and can thus contribute to the state estimate. This introduces two problems. First, how to select the active nodes. Second, how to maintain consensus among the nodes that do not contribute a measurement. The second problem arises because it is not known in advance which node will be in range of the target at a future time step, so neighbouring nodes of an active node still need to track the target state to know when it is in range and to have a valid initialisation for the EKF or EIF once it becomes active.

The first problem is addressed by assuming a maximum measurement range and using the known node position together with the previous state estimate of the target. Using this information, the distance between the node and the target can be estimated and used to determine whether the node is active or inactive at the current time step. This is a simple method and could be improved in future work.

Sensor selection is possible because every node can compute a prediction of the target state $\hat{x}_{t|t-1}$ using

$$\hat{x}_{t|t-1} = F\hat{x}_{t-1|t-1}, \quad (4.3)$$

identical to the prediction step already introduced in Equation 2.12. The estimated x -position of the target is given by the first entry of the state vector, $\hat{x}_{t|t-1}(1)$, following the state vector convention of Equation 2.1. The x -distance between the target and node i is given by

$$\Delta x = \hat{x}_{t|t-1}(1) - r_i^x, \quad (4.4)$$

where r_i^x denotes the x -position of node i , with Δy and Δz defined analogously for the y - and z -axes. Node i is selected as active if the target falls within its assumed maximum measurement range $r_{\max}$,

$$\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \leq r_{\max}. \quad (4.5)$$

Applying this criterion to every node at each time step determines the set of currently active nodes. Because only active nodes transmit, both an active node and an inactive node only ever receive values from their currently active neighbours. For the remainder of this section, $\mathcal{N}_i$ therefore denotes the set of node i 's currently active neighbours, rather than its full set of neighbours in the network graph as used in Section 3.2,

with d_i^{active} denoting the resulting degree of node i . Using this, the information vector and information matrix are updated for each active node i as

$$\begin{aligned}\xi_{i,t}^{(k)} &= (I + cd_i^{\text{active}}I)^{-1} \left(N^{\text{active}} H_{i,t}^\top R_i^{-1} (z_{i,t} - h(\hat{x}_{i,t|t-1}) + H_{i,t} \hat{x}_{i,t|t-1}) - \sum_{j \in \mathcal{N}_i} A_{i|j}^\top z_{i|j} \right), \\ \Xi_{i,t}^{(k)} &= (I + cd_i^{\text{active}}I)^{-1} \left(N^{\text{active}} H_{i,t}^\top R_i^{-1} H_{i,t} - \sum_{j \in \mathcal{N}_i} A_{i|j}^\top Z_{i|j} \right),\end{aligned}\tag{4.6}$$

where N^{active} denotes the number of currently active nodes. The second problem, maintaining consensus without letting inactive nodes influence the state estimate, is addressed by allowing only active nodes to transmit their auxiliary variables $z_{j|i}$ and $Z_{j|i}$. Because inactive nodes do not transmit at all, they cannot contribute to any neighbour's update, whether that neighbour is itself active or inactive. Active nodes broadcast these variables using the general PDMM transmission rule introduced in Section 3.2, given here for node i transmitting to neighbour j as

$$\begin{aligned}y_{i|j} &= z_{i|j} + 2cA_{i|j}\xi_{i,t}^{(k)}, \\ Y_{i|j} &= Z_{i|j} + 2cA_{i|j}\Xi_{i,t}^{(k)},\end{aligned}\tag{4.7}$$

while inactive nodes only listen, without transmitting any values of their own. For an inactive node i , this means $y_{i|j} = 0$ and $Y_{i|j} = 0$ for every active neighbour $j \in \mathcal{N}_i$. Left-multiplying by $A_{i|j}^\top$ and summing over $\mathcal{N}_i$ gives

$$\begin{aligned}0 &= \sum_{j \in \mathcal{N}_i} A_{i|j}^\top z_{i|j} + 2c \sum_{j \in \mathcal{N}_i} A_{i|j}^\top A_{i|j} \xi_{i,t}^{(k)}, \\ 0 &= \sum_{j \in \mathcal{N}_i} A_{i|j}^\top Z_{i|j} + 2c \sum_{j \in \mathcal{N}_i} A_{i|j}^\top A_{i|j} \Xi_{i,t}^{(k)}.\end{aligned}\tag{4.8}$$

Since $A_{i|j} = I$ for the consensus constraint introduced in Section 3.2, $A_{i|j}^\top A_{i|j} = I$, so these reduce to $2cd_i^{\text{active}}I \xi_{i,t}^{(k)} = -\sum_{j \in \mathcal{N}_i} A_{i|j}^\top z_{i|j}$ and $2cd_i^{\text{active}}I \Xi_{i,t}^{(k)} = -\sum_{j \in \mathcal{N}_i} A_{i|j}^\top Z_{i|j}$, yielding the update equations for an inactive node i ,

$$\begin{aligned}\xi_{i,t}^{(k)} &= (2cd_i^{\text{active}}I)^{-1} \left(-\sum_{j \in \mathcal{N}_i} A_{i|j}^\top z_{i|j} \right), \\ \Xi_{i,t}^{(k)} &= (2cd_i^{\text{active}}I)^{-1} \left(-\sum_{j \in \mathcal{N}_i} A_{i|j}^\top Z_{i|j} \right),\end{aligned}\tag{4.9}$$

At $t = 0$, every node, active or inactive, is initialised with the same known initial state x_0 , before any activation decisions are made. This provides the seed estimate from which the first active nodes obtain their initialisation, since no node has a prior active neighbour to receive from at $t = 0$.

If node i has at least one active neighbour, $\mathcal{N}_i \neq \emptyset$, it computes an updated state estimate using Equation 4.9, since it is likely to come within measurement range and

become active in the near future. Once node i becomes active, this estimate, obtained without node i itself ever transmitting a measurement, serves as its initialisation for the EIF. This relies on the target never leaving the measurement range of every node simultaneously, discussed further in subsection 4.2.3. Applying this sensor selection method to EIF-PDMM yields EIF-PDMM-SS (Sensor Selection).

4.2.1 Experimental Setup

The simulation setup used to obtain the results presented in subsection 4.2.2 is described below. The initial state estimate is given by

$$\hat{x}_{1|0} = Fx_0, \quad (4.10)$$

where x_0 contains only the target's initial position, with the velocity components initialised to zero. The initial covariance is set to $P_0 = \text{diag}(500, 500, 500, 100, 100, 100)$. The process noise spectral density is assumed equal among the x -, y -, and z -axes, i.e. $q_x = q_y = q_z = q$ in Equation 2.6. As found in subsection 3.5.2, $c = 0.1$. The remaining simulation parameters are summarised in Table 4.2.

Table 4.2: Simulation parameters used in the sensor selection scenario.

Parameter	Value
Number of targets, L	1
Number of nodes, N	200
Maximum time, T	100
Sampling interval, δt	0.5
Process noise spectral density, q	0.2
Maximum PDMM iterations	100
Angle noise standard deviation, $\sigma_\varphi, \sigma_\theta$	5°
Monte Carlo runs	100
Measurement range, $r_{\max}$	700

4.2.2 Results

The mean error per target over 100 Monte Carlo simulations is shown in Figure 4.4, where the node positions are kept fixed across runs and only the measurement noise realisation $v_{\ell,t}$ is varied. The presented algorithm, EIF-PDMM-SS, is compared against EIF-PDMM. The error of EIF-PDMM-SS remains consistently above that of EIF-PDMM, which is expected because, as discussed in Section 4.1, more measurements will result in a lower error, and the EIF-PDMM algorithm assumes that all sensors have a measurement from the target and thus uses all sensors.

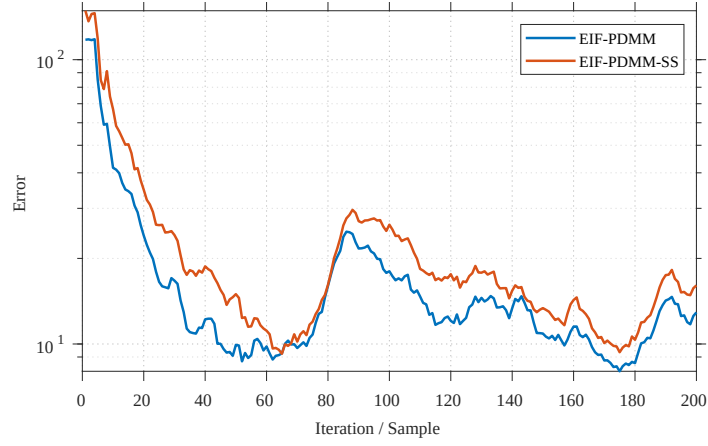


Figure 4.4: Mean state error over 100 Monte Carlo runs, EIF-PDMM-SS vs. EIF-PDMM.

Figure 4.5 shows the mean number of nodes used by the EIF-PDMM-SS algorithm. The number of selected nodes remains around 20, compared to the total of 200 nodes given in Table 4.2, a significant reduction in the number of nodes used.

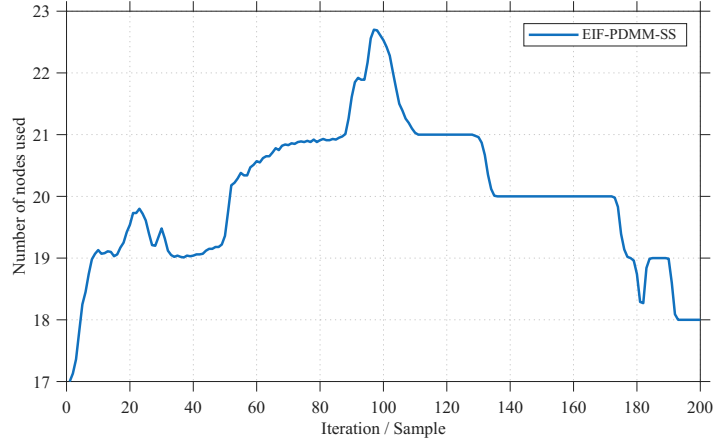


Figure 4.5: Mean number of selected nodes per time step t , EIF-PDMM-SS.

Figure 4.6 shows the estimated trajectory in three dimensions, and Figure 4.7 shows the same trajectory projected onto the XY and XZ planes, for the EIF-PDMM-SS algorithm.

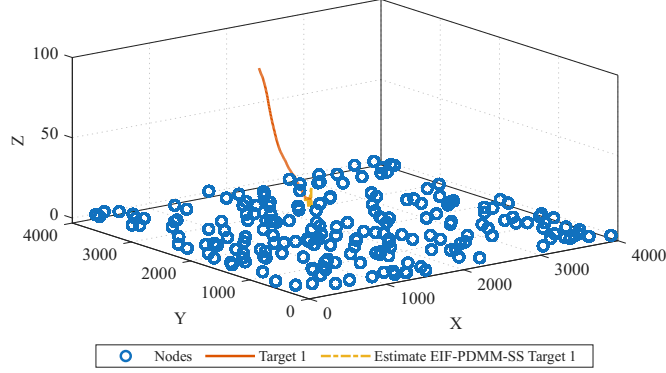


Figure 4.6: Estimated target trajectory in three dimensions, EIF-PDMM-SS.

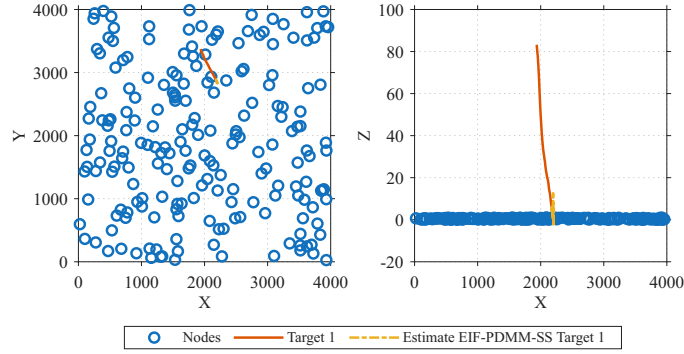


Figure 4.7: Estimated target trajectory projected onto the X-Y and X-Z planes, EIF-PDMM-SS.

4.2.3 Improvements

A node that becomes active without having had a prior active neighbour would have no valid state estimate to initialise from, since Equation 4.9 depends on receiving values from at least one active neighbour. The method therefore relies on the target never leaving the measurement range of every node simultaneously. The simulation setup of subsection 4.2.1 is designed to satisfy this condition, but it is not enforced by the sensor selection method itself. Handling scenarios in which the target may leave the range of all nodes, for example through explicit re-initialisation logic, is left as a direction for future work.

Because the current method for sensor selection is not optimal, different methods, as the ones described in [14], can be used to select sensors based on how informative their measurement is. This is likewise left as a direction for future work.

4.3 Sensor Selection in a Multi-Target Environment

The sensor selection method can also be combined with the multi-target method resulting in the EIF-PDMM-SS-MT. This is done by applying the gating principle discussed in Section 4.1 and the sensor selection method discussed in Section 4.2.

4.3.1 Experimental Setup

The simulation setup used to obtain the results presented in subsection 4.3.2 is described below. The initial state estimate is given by

$$\hat{x}_{\ell,1|0} = Fx_{\ell,0}, \quad (4.11)$$

where $x_{\ell,0}$ contains only the target's initial position, with the velocity components initialised to zero. The initial covariance is set to $P_{\ell,0} = \text{diag}(500, 500, 500, 100, 100, 100)$. The process noise spectral density is assumed equal among the x -, y -, and z -axes, i.e. $q_x = q_y = q_z = q$ in Equation 2.6. As found in subsection 3.5.2, $c = 0.1$. The remaining simulation parameters are summarised in Table 4.3.

Table 4.3: Simulation parameters used in the sensor selection multi-target scenario.

Parameter	Value
Number of targets, L	2
Number of nodes, N	200
Maximum time, T	100
Sampling interval, δt	0.5
Process noise spectral density, q	0.2
Maximum PDMM iterations	100
Angle noise standard deviation, $\sigma_\varphi, \sigma_\theta$	5°
Monte Carlo runs	100
Measurement range, $r_{\max}$	700

4.3.2 Results

Figure 4.8 shows the error of EIF-PDMM-MT and EIF-PDMM-SS-MT for both targets. The estimate of target 1 under EIF-PDMM-SS-MT deteriorates as time progresses. Inspecting Figure 4.12 confirms this: the later estimates for target 1 depart visibly from its true trajectory. A possible explanation is that, because target 1 moves almost exclusively in the z -direction and the sensors measure only angles, the resulting change in angle becomes difficult to detect once only a small subset of nodes is selected, degrading the estimate. This would also explain why the error remains lower when all sensors are considered, as in the EIF-PDMM-MT algorithm. To confirm this explanation, a second scenario is generated in which the two targets move less in the z -direction. The resulting mean error is shown in Figure 4.9, where the error of target 1 under EIF-PDMM-SS-MT remains close to that of EIF-PDMM-MT, rather than deteriorating as in the first scenario. This supports the explanation above, since reducing the motion of

the target along the least observable axis of the sensors removes the effect responsible for the earlier degradation.

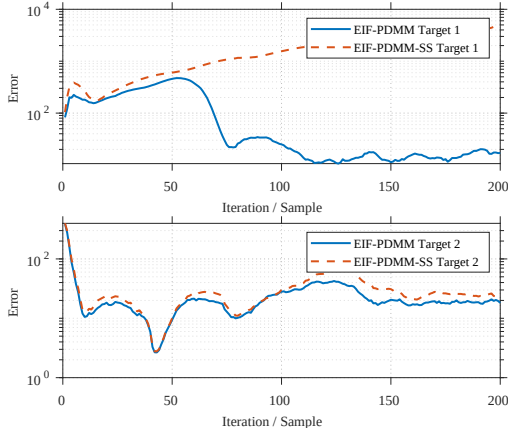


Figure 4.8: Mean state error over 100 Monte Carlo runs, EIF-PDMM-SS-MT vs. EIF-PDMM-MT (first scenario).

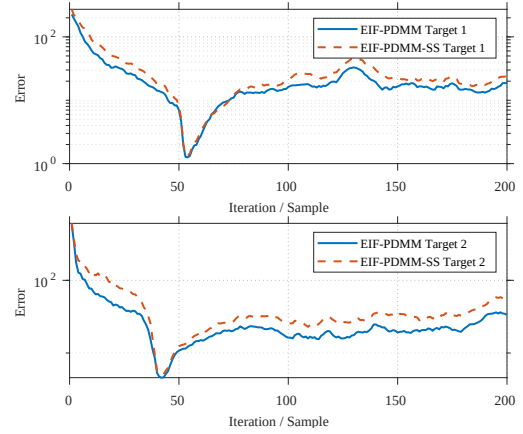


Figure 4.9: Mean state error over 100 Monte Carlo runs, EIF-PDMM-SS-MT vs. EIF-PDMM-MT (second scenario).

Figure 4.10 shows the mean number of nodes used by the EIF-PDMM-SS-MT algorithm for the first scenario, and Figure 4.11 shows the same for the second scenario. In the first scenario, the number of selected nodes remains around 20 for target 1 and around 16 for target 2, a reduction comparable to that observed in the single-target case of Section 4.2. The second scenario yields node counts of the same order but not identical, since the two targets follow different trajectories. Compared to the total of 200 nodes given in Table 4.3, both scenarios represent a significant reduction.

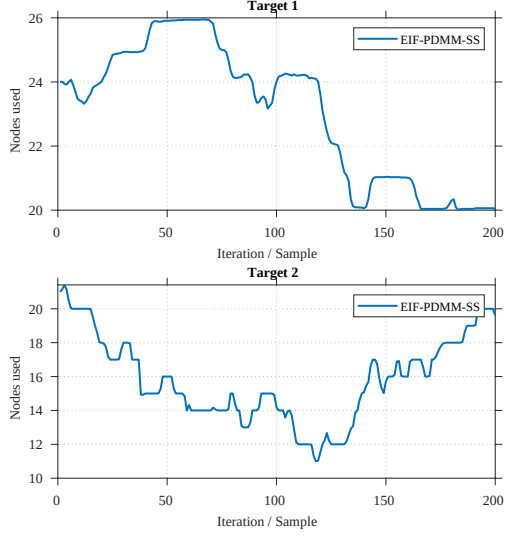


Figure 4.10: Mean number of selected nodes per time step t , EIF-PDMM-SS-MT (first scenario).

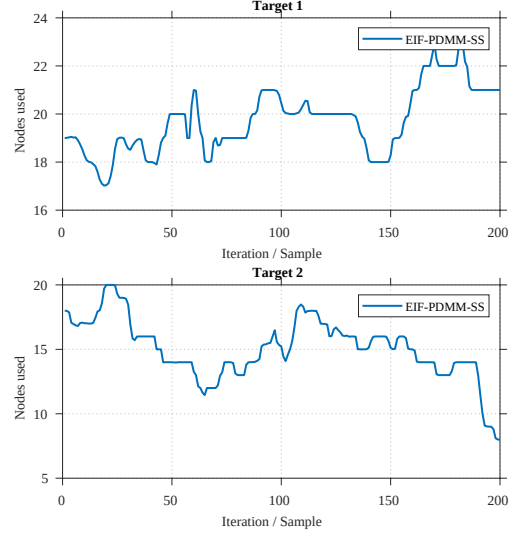


Figure 4.11: Mean number of selected nodes per time step t , EIF-PDMM-SS-MT (second scenario).

Figure 4.12 shows the estimated trajectories in three dimensions, and Figure 4.14 shows the same trajectories projected onto the XY and XZ planes, for the EIF-PDMM-SS-MT algorithm in the first scenario. Figure 4.13 and Figure 4.15 show the corresponding trajectories for the second scenario, where the estimate of target 1 no longer departs from its true trajectory.

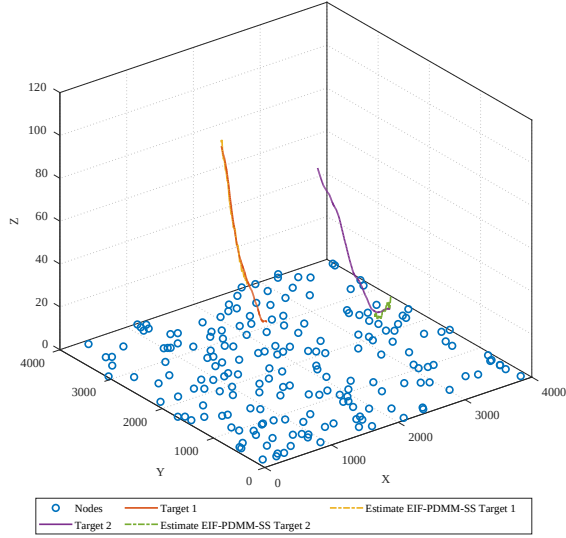


Figure 4.12: Estimated target trajectories in three dimensions, EIF-PDMM-SS-MT (first scenario).

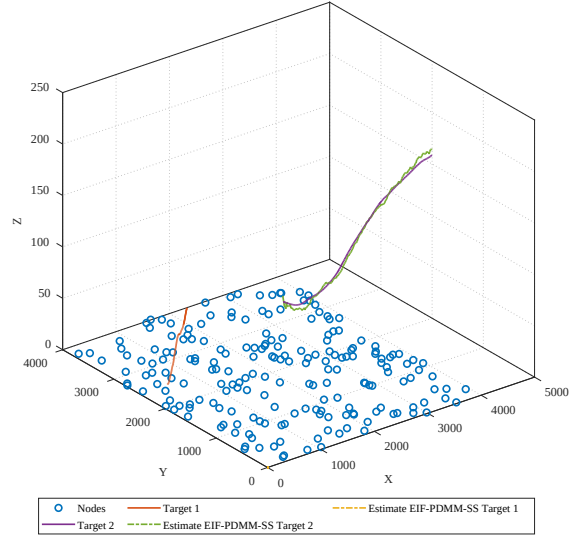


Figure 4.13: Estimated target trajectories in three dimensions, EIF-PDMM-SS-MT (second scenario).

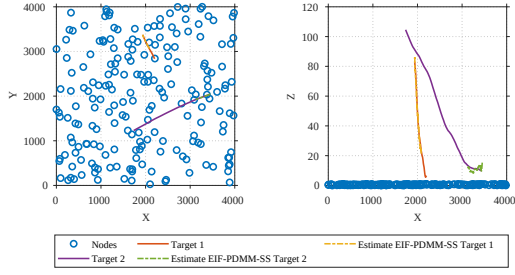


Figure 4.14: Estimated target trajectories projected onto the X-Y and X-Z planes, EIF-PDMM-SS-MT (first scenario).

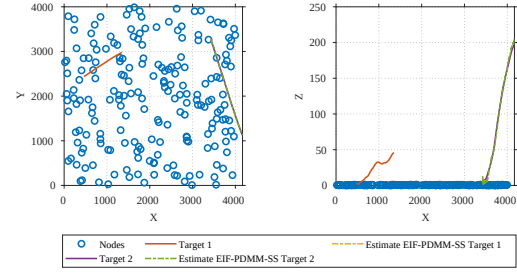


Figure 4.15: Estimated target trajectories projected onto the X-Y and X-Z planes, EIF-PDMM-SS-MT (second scenario).

Figure 4.16 shows the trajectories of scenario one of the two targets and the active nodes used to estimate them at $t = 100$.

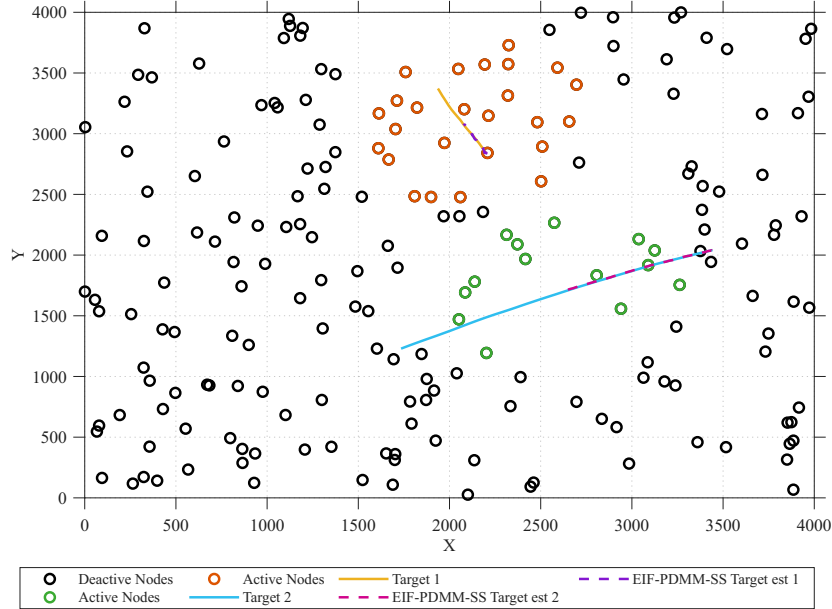


Figure 4.16: Estimated target states and active nodes at $t = 100$, EIF-PDMM-SS-MT (first scenario).

Conclusion

In this thesis, it has been shown that PDMM can be used to solve a target tracking problem using Direction of Arrival (DoA) measurements in a EKF and EIF framework. It also has been shown that the outcomes of these filters are equivalent to each other and to a centralised solution.

The algorithms introduced in this work, EKF-PDMM and EIF-PDMM, have advantages over their centralised counterparts: while not shown explicitly in this thesis, they are robust against node failures and transmission losses, and have no single point of failure. With EKF-PDMM and EIF-PDMM the problem is solved locally, which greatly reduces the communication range needed for the sensors in the sensor network in a vast area such as the North Sea.

Through simulation, it has been shown that EIF-PDMM converges faster than EKF-PDMM while attaining the same result, making EIF-PDMM the preferable choice in terms of convergence speed.

It has furthermore been shown that EIF-PDMM can be extended to a multi-target scenario by making use of gating, resulting in EIF-PDMM-MT. Additionally, a distance-based sensor selection method was shown to allow EIF-PDMM to operate using fewer sensors while maintaining consensus, resulting in EIF-PDMM-SS. As expected, this comes at the cost of a higher estimation error than when using the full set of sensors. These two extensions were also shown to be compatible, combining into EIF-PDMM-SS-MT.

5.1 Future Work

While this thesis demonstrates that EIF-PDMM can be extended to multi-target tracking and large-scale sensor networks, several directions remain open for future work. Some of these points were already raised in the relevant chapters and are restated here to give a single, consolidated overview of the limitations of the presented work.

Throughout this thesis, the number of targets L is assumed to be known and fixed, with every track already initialised and active for the full duration of the scenario. In practice, targets enter and leave the area of interest over time, and the algorithm presented in this work provides no mechanism for detecting the birth of a new track or terminating a track once its target has left the sensor network's coverage area. Extending EIF-PDMM-MT with a track initiation and termination scheme is therefore an important direction for future work.

A related simplification is that every measurement is assumed to originate from an active target, so false positive measurements are not considered. Should false positives be introduced, for example from clutter, the χ^2 upper bound on the Mahalanobis distance, introduced in subsection 2.4.2 but deliberately not used as an outlier-rejection

criterion in this work, would become necessary to discard them.

A further limitation concerns closely spaced targets, for which gating cannot correctly associate measurements once the gates of different tracks overlap. JPDA addresses this by weighting every candidate measurement within a gate by its likelihood, rather than selecting a single nearest neighbour, and would be expected to improve performance in this regime.

For sensor selection, two extensions are worth pursuing. First, the current method assumes the target never leaves the measurement range of every node simultaneously, since a node that becomes active without a prior active neighbour has no valid state estimate to initialise from. Handling the case in which the target does leave the range of all nodes, for example through explicit re-initialisation logic, would remove this restriction. Second, the current criterion selects nodes based only on distance to the target. Selecting nodes based on how informative their measurement is expected to be, using methods such as those described in [14], would likely allow the algorithm to extract the available information more efficiently.

References

- [1] Ministerie van Algemene Zaken. *Beschermen infrastructuur Noordzee, zoals gasleidingen en data- en elektriciteitskabels* — *Rijksoverheid.nl*. nl. webpagina. Last Modified: 2025-07-29T13:37:55.264Z. Apr. 2025.
- [2] Stephen Boyd. “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers”. en. In: *Foundations and Trends® in Machine Learning* 3.1 (2010), pp. 1–122. ISSN: 1935-8237, 1935-8245. DOI: [10.1561/22000000016](#).
- [3] Thomas William Sherson, Richard Heusdens, and W. Bastiaan Kleijn. “Derivation and Analysis of the Primal-Dual Method of Multipliers Based on Monotone Operator Theory”. In: *IEEE Transactions on Signal and Information Processing over Networks* 5.2 (June 2019), pp. 334–347. ISSN: 2373-776X. DOI: [10.1109/TSIPN.2018.2876754](#).
- [4] X. Rong Li and V.P. Jilkov. “Survey of maneuvering target tracking. Part I. Dynamic models”. In: *IEEE Transactions on Aerospace and Electronic Systems* 39.4 (Oct. 2003), pp. 1333–1364. ISSN: 1557-9603. DOI: [10.1109/TAES.2003.1261132](#).
- [5] R. E. Kalman. “A New Approach to Linear Filtering and Prediction Problems”. In: *Journal of Basic Engineering* 82.1 (Mar. 1960), pp. 35–45. ISSN: 0021-9223. DOI: [10.1115/1.3662552](#).
- [6] Brian D. O. Anderson and John B. Moore. *Optimal Filtering*. en. Google-Books-ID: VZ4oAwAAQBAJ. Courier Corporation, Jan. 2005. ISBN: 978-0-486-43938-9.
- [7] Sebastian Thrun et al. “Simultaneous Localization and Mapping with Sparse Extended Information Filters”. EN. In: *The International Journal of Robotics Research* 23.7-8 (Aug. 2004), pp. 693–716. ISSN: 0278-3649. DOI: [10.1177/0278364904045479](#).
- [8] William W. Hager. “Updating the Inverse of a Matrix”. In: *SIAM Review* 31.2 (June 1989), pp. 221–239. ISSN: 0036-1445. DOI: [10.1137/1031049](#).
- [9] Yaakov Bar-Shalom and Edison Tse. “Tracking in a cluttered environment with probabilistic data association”. In: *Automatica* 11.5 (1975). Publisher: Elsevier, pp. 451–460.
- [10] Guoqiang Zhang and Richard Heusdens. “Distributed Optimization Using the Primal-Dual Method of Multipliers”. In: *IEEE Transactions on Signal and Information Processing over Networks* 4.1 (Mar. 2018), pp. 173–187. ISSN: 2373-776X. DOI: [10.1109/TSIPN.2017.2672403](#).
- [11] S. Boyd et al. “Randomized gossip algorithms”. In: *IEEE Transactions on Information Theory* 52.6 (June 2006), pp. 2508–2530. ISSN: 1557-9654. DOI: [10.1109/TIT.2006.874516](#).
- [12] Sebastian O. Jordan, Thomas W. Sherson, and Richard Heusdens. “Convergence of Stochastic PDMM”. In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. June 2023, pp. 1–5. DOI: [10.1109/ICASSP49357.2023.10095808](#).
- [13] Muhammad Iqbal, Kundan Kumar, and Simo Särkkä. “Communication-Efficient Distributed Kalman Filtering Using ADMM”. In: *IEEE Transactions on Automatic Control* 71.3 (Mar. 2026), pp. 1916–1923. ISSN: 1558-2523. DOI: [10.1109/TAC.2025.3615237](#).
- [14] Siddharth Joshi and Stephen Boyd. “Sensor Selection via Convex Optimization”. In: *IEEE Transactions on Signal Processing* 57.2 (Feb. 2009), pp. 451–462. ISSN: 1941-0476. DOI: [10.1109/TSP.2008.2007095](#).

EIF Derivation



This appendix provides the full derivation of the EIF information vector update used in Section 2.3, obtained by multiplying the EKF state update equation by the information matrix $P_{t|t}^{-1}$.

$$\begin{aligned}
 \hat{x}_{t|t} &= \hat{x}_{t|t-1} + P_{t|t-1} H_t^\top (H_t P_{t|t-1} H_t^\top + R)^{-1} (z_t - h(\hat{x}_{t|t-1})), \\
 P_{t|t}^{-1} \hat{x}_{t|t} &= P_{t|t}^{-1} \left(\hat{x}_{t|t-1} + P_{t|t-1} H_t^\top (H_t P_{t|t-1} H_t^\top + R)^{-1} (z_t - h(\hat{x}_{t|t-1})) \right) \\
 &= \left(P_{t|t-1}^{-1} + H_t^\top R^{-1} H_t \right) \left(\hat{x}_{t|t-1} + P_{t|t-1} H_t^\top (H_t P_{t|t-1} H_t^\top + R)^{-1} (z_t - h(\hat{x}_{t|t-1})) \right) \\
 &= P_{t|t-1}^{-1} \hat{x}_{t|t-1} + H_t^\top R^{-1} H_t \hat{x}_{t|t-1} \\
 &\quad + H_t^\top (H_t P_{t|t-1} H_t^\top + R)^{-1} (z_t - h(\hat{x}_{t|t-1})) \\
 &\quad + H_t^\top R^{-1} H_t P_{t|t-1} H_t^\top (H_t P_{t|t-1} H_t^\top + R)^{-1} (z_t - h(\hat{x}_{t|t-1})) \\
 &= P_{t|t-1}^{-1} \hat{x}_{t|t-1} + H_t^\top R^{-1} H_t \hat{x}_{t|t-1} \\
 &\quad + H_t^\top \left((H_t P_{t|t-1} H_t^\top + R)^{-1} + R^{-1} H_t P_{t|t-1} H_t^\top (H_t P_{t|t-1} H_t^\top + R)^{-1} \right) (z_t - h(\hat{x}_{t|t-1})).
 \end{aligned} \tag{A.1}$$

With $A = H_t P_{t|t-1} H_t^\top$, only the last term of the equation above needs to be considered,

$$\begin{aligned}
 &H_t^\top ((A + R)^{-1} + R^{-1} A (A + R)^{-1}) (z_t - h(\hat{x}_{t|t-1})) \\
 &= H_t^\top ((I + R^{-1} A) (A + R)^{-1}) (z_t - h(\hat{x}_{t|t-1})) \\
 &= H_t^\top (R^{-1} (A + R) (A + R)^{-1}) (z_t - h(\hat{x}_{t|t-1})) \\
 &= H_t^\top R^{-1} (z_t - h(\hat{x}_{t|t-1})).
 \end{aligned} \tag{A.2}$$

Substituting this result back into the equation above yields

$$\begin{aligned}
 P_{t|t}^{-1} \hat{x}_{t|t} &= P_{t|t-1}^{-1} \hat{x}_{t|t-1} + H_t^\top R^{-1} H_t \hat{x}_{t|t-1} + H_t^\top R^{-1} (z_t - h(\hat{x}_{t|t-1})) \\
 &= P_{t|t-1}^{-1} \hat{x}_{t|t-1} + H_t^\top R^{-1} (z_t - h(\hat{x}_{t|t-1}) + H_t \hat{x}_{t|t-1}), \\
 \xi_{t|t} &= \xi_{t|t-1} + H_t^\top R^{-1} (z_t - h(\hat{x}_{t|t-1}) + H_t \hat{x}_{t|t-1}),
 \end{aligned} \tag{A.3}$$

which is the information vector update given in Equation 2.22 of Section 2.3.

EIF-PDMM Algorithm

Algorithm 2 EIF-PDMM

Initialize: F, H, Q, R, x_0, P_0
for $t = 1, \dots, T$ **do**
 Initialise: $y, z \in \mathbb{R}^{n \times 1}, Y, Z \in \mathbb{R}^{n \times n}$
 for $k = 1, \dots,$ **do**
 for all $i \in \mathcal{V}$ **do**
 $\hat{x}_{i,t|t-1} = F\hat{x}_{i,t-1|t-1}$
 $P_{i,t|t-1} = FP_{i,t-1|t-1}F^\top + Q$
 $\xi_{i,t}^{(k)} = (I + cd_i I)^{-1} \left(NH_{i,t}^\top R_i^{-1} (z_{i,t} - h(\hat{x}_{i,t|t-1}) + H_{i,t} \hat{x}_{i,t|t-1}) - \sum_{j \in \mathcal{N}_i} A_{i|j}^\top z_{i|j} \right)$
 $\Xi_{i,t}^{(k)} = (I + cd_i I)^{-1} \left(NH_{i,t}^\top R_i^{-1} H_{i,t} - \sum_{j \in \mathcal{N}_i} A_{i|j}^\top Z_{i|j} \right)$
 for all $j \in \mathcal{N}_i$ **do**
 $y_{i|j} = z_{i|j} + 2cA_{i|j}\xi_{i,t}^{(k)}$
 $Y_{i|j} = Z_{i|j} + 2cA_{i|j}\Xi_{i,t}^{(k)}$
 end for
 for all $i \in \mathcal{V}, j \in \mathcal{N}_i$ **do**
 $z_{i|j} = y_{j|i}$
 $Z_{i|j} = Y_{j|i}$
 end for
 end for
 end for
 $P_{t|t} = \left(P_{t|t-1}^{-1} + \Xi_{i,t}^{(k)} \right)^{-1}$
 $\hat{x}_{t|t} = P_{t|t} \left(P_{t|t-1}^{-1} \hat{x}_{i,t|t-1} + \xi_{i,t}^{(k)} \right)$
 end for
