Adaptive Task Scheduling Approaches for Weather Radar Networks

By

Jothiga Srinivasan

in partial fulfilment of the requirements for the degree of

Master of Science

in Electrical Engineering

at the Delft University of Technology

Student Number:	6133185
Track:	Wireless Communication and Sensing
Research Group:	Microwave Sensing, Signals and Systems (MS3)
Supervisor:	Dr. Francesco Fioranelli
Daily Supervisor:	Apostolos Pappas
Defense Date:	21 August 2026

Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science

August 14, 2026

Abstract

Weather-radar networks must allocate limited sensing time between repeated observations of evolving storms and background surveillance. Fixed scanning strategies provide systematic coverage, but cannot adapt the use of radar resources when several storm observations compete for service. This thesis investigates adaptive task scheduling for a multi-radar weather-sensing network through two proposed approaches: Radar-Level Task Selection (RLTS) and Load-Aware Task Scheduling (LATS).

RLTS combines Time-Balance revisit urgency with dynamic storm priority, task feasibility, and radar-dependent information to select the next observation. LATS extends this approach with explicit storm-to-radar assignment based on scan time, traverse time, urgency, and radar workload. The methods are evaluated against cyclic storm-by-storm tracking and continuous 360° scanning using 50 Monte Carlo trials of 20 min across five storm scenarios, simulated in a developed framework.

The results show that adaptive scheduling becomes increasingly beneficial as radar resources become constrained. LATS achieves the shortest median revisit time in four of the five scenarios, while the cyclic baseline remains strongest under low load. Under high load, LATS reduces the median revisit time by 12.9% relative to RLTS while increasing mean surveillance coverage from 3.79% to 13.13%. Tracking accuracy remains broadly comparable between the two adaptive schedulers.

The results demonstrate that adaptive task selection improves storm servicing when observations compete for radar time, while load-aware assignment provides an additional benefit by distributing the workload more effectively across the radar network and preserving greater surveillance capacity.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Dr Francesco Fioranelli and Apostolos Pappas, for their guidance, feedback, and support throughout this thesis. Their discussions and suggestions helped me refine both the technical direction of the work and the way in which the research was presented. I am especially grateful for the time they dedicated to reviewing my work and for encouraging me to think critically about the choices made throughout the project.

I would also like to thank everyone at TU Delft who contributed, directly or indirectly, to my Master's studies and to the completion of this thesis. The experience has given me the opportunity to develop not only my technical knowledge, but also my ability to approach open-ended research problems independently.

Finally, and most importantly, I would like to thank my parents, Srinivasan and Usharani, for their unconditional support throughout my studies. Their encouragement, patience, and belief in me have carried me through both the rewarding and difficult parts of this journey.

I am also very grateful to my friends for being a constant source of support, encouragement, and much-needed laughter throughout this Master's journey. Their presence made the difficult days easier, the good moments more memorable, and this entire experience far more enjoyable.

This thesis marks the end of an important chapter for me, and I am grateful to everyone who has been part of it.

Contents

List of Figures

List of Tables

List of Acronyms

AoI Area of Interest

B1 Baseline 1

B2 Baseline 2

CASA Center for Collaborative Adaptive Sensing of the Atmosphere

DCAS Distributed Collaborative Adaptive Sensing

IoU Intersection over Union

KNMI Royal Netherlands Meteorological Institute

LATS Load-Aware Task Scheduling

MESAR Multi-function Electronically Scanned Adaptive Radar

MS3 Microwave Sensing, Signals and Systems

PPI Plan Position Indicator

PRF Pulse Repetition Frequency

RLTS Radar-Level Task Selection

RRM Radar Resource Management

TB Time Balance

VCP Volume Coverage Pattern

WSR–88D Weather Surveillance Radar–1988 Doppler

Chapter 1

Introduction

1.1 Motivation and background

Weather radar plays an important role in observing precipitation systems and supporting applications such as severe-weather monitoring, rainfall estimation, aviation, and short-term weather forecasting [1, 2, 3, 4]. The usefulness of these observations depends not only on spatial coverage, but also on how frequently rapidly evolving weather features are revisited. Rapidly evolving convective weather can change substantially between successive radar observations. When the interval between updates is long, important changes in storm position, size, and motion may not be observed sufficiently frequently [5]. Reducing the revisit interval can therefore improve the temporal description of storm development [6]. However, continuously increasing the update frequency of every weather region is generally not possible because radar time is limited and each scan requires a finite number of dwells.

Conventional radar scanning strategies generally follow predefined scan patterns that provide systematic coverage of the atmosphere. For example, conventional full-azimuth scanning provides broad coverage, but its update time is mainly determined by the duration of the complete scan cycle. This approach does not distinguish between regions containing important weather features and regions requiring only general surveillance. Moreover, because the available radar time is finite, increasing spatial coverage or scan detail in a certain sector can increase the time required to complete a full scan and therefore reduce the temporal resolution of the observations.

Advances in flexible scanning and phased-array radar technology provide greater freedom in the way radar measurements are collected. Instead of repeatedly following a fixed scan sequence, radar time can be directed towards selected regions according to the current weather situation. Flexible volume-coverage strategies have shown that scan patterns can be adapted to improve the balance between spatial and temporal sampling [7]. Phased-array systems extend this capability through rapid electronic beam steering, allowing observations of meteorologically important regions to be updated more frequently while maintaining broader weather surveillance [8]. Adaptive scanning therefore allows radar resources to be concentrated on selected storm regions while the remaining radar time is used for broader

environmental coverage. The resulting scheduling problem is a balance between two competing objectives: providing sufficiently frequent updates of known storms and maintaining surveillance of the surrounding domain.

The introduction of adaptive scanning also creates a *radar resource-management problem*. When multiple storm cells require repeated observations, the radar must determine which region should be scanned next, how frequently each storm should be revisited, and how the available radar time should be divided between focused tracking and background surveillance. These decisions become more difficult when storms differ in size, motion, location, and requested update time. During periods of high weather activity, the total demand for radar time may exceed the available sensing capacity, making prioritisation and overload handling necessary.

The scheduling difficulty increases when several storms are present simultaneously. Different storms may require different levels of attention depending on their current state. For example, a large or rapidly moving storm may require more frequent monitoring than a smaller and slower storm. Similarly, a storm approaching a predefined area of interest may be considered more important because changes in its behaviour may have greater operational relevance [9, 10]. Since these properties change as the storm evolves, the priority assigned to a storm should also be allowed to change during the simulation.

Time-Balance (TB) scheduling was used in multifunction phased-array radar resource management to coordinate competing radar jobs, including surveillance, plot confirmation, track initiation, and target tracking [11]. In the modified MESAR formulation described by Butler, the time-balance value indicates how early or late a task is relative to its requested execution time. Reinoso-Rondinel et al. later adopted this general scheduling principle for adaptive weather sensing, where storm-tracking tasks with different update-time requirements were interleaved with background surveillance [12]. Subsequent work extended the weather-radar framework by incorporating task prioritisation and the adaptation of requested update times under overload conditions [13]. These developments provide an important basis for adaptive weather-radar scheduling, but they primarily address the management of multiple tasks within a single-radar setting.

Weather-radar networks introduce additional sensing opportunities. Multiple radars can provide overlapping observations, improved low-altitude coverage, and increased flexibility in observing different parts of a weather system [14]. Previous research has also investigated agile radar operation, knowledge-based radar control, meteorological command-and-control systems, and collaborative adaptive sensing in distributed radar networks [15, 16, 17, 18]. However, the availability of several radars also introduces an additional decision problem: storm-tracking tasks must be distributed across the network, while maintaining sufficient surveillance capability. The scheduler must consider not only which storm is most urgent, but also which radar can perform the task efficiently while accounting for scan duration, radar pointing direction, and the workload already assigned to each radar.

The use of a radar network provides additional capacity because tracking tasks can be shared between multiple radar nodes. Nevertheless, this capacity can only be used effec-

tively when the tasks are coordinated. Assigning a storm to a radar solely according to storm urgency may lead to inefficient radar motion or an uneven distribution of workload. Conversely, assigning tasks only according to geometric proximity may neglect storms that have waited too long for an update. An effective scheduling method must therefore consider both storm-related and radar-related information.

This thesis investigates adaptive task scheduling for a weather-radar network in which multiple storm-tracking tasks compete with background-surveillance requirements. The proposed approach combines revisit urgency with dynamically updated storm priority. The priority is derived from evolving storm-state information, including storm distance to a predefined area requiring enhanced monitoring, storm size, and storm speed. A second scheduling approach extends the radar-level task-selection method by introducing load-aware storm-to-radar assignment. The objective is to investigate whether coordinated assignment can improve the use of the available radar network while maintaining tracking quality and retaining background surveillance. The scheduling methods are evaluated not only through revisit performance, but also through tracking accuracy, surveillance coverage, and radar time allocation.

1.2 Problem Statement

Previous research has demonstrated the benefits of flexible weather-radar scanning and adaptive phased-array observations [7, 8], Time-Balance scheduling, task prioritisation, and overload management [12, 13], and networked weather-radar sensing [14, 19]. However, these aspects have generally been investigated in separate settings. Time-Balance scheduling studies primarily address the management of competing tasks for an individual radar [12, 13], whereas weather-radar-network studies mainly focus on sensing coverage, network geometry, or application-specific coordination [14, 19].

Consequently, the coordinated scheduling of multiple storm-tracking tasks across a radar network remains a relevant research problem. The scheduler must determine which storms should be observed, how their urgency and priority should influence task selection, and how tracking demand should be distributed among the available radars. These decisions must account for finite scan duration, radar traverse time, changing storm conditions, and variations in radar workload. At the same time, allocating radar time to storm tracking should not completely eliminate background surveillance.

The problem addressed in this thesis is therefore the development and evaluation of adaptive scheduling approaches that coordinate storm-tracking and surveillance tasks within a multi-radar weather-sensing network. The scheduling framework should respond to changing storm conditions, manage increased tracking demand, and use the available radar resources efficiently while preserving the ability to observe unoccupied regions of the surveillance domain.

1.3 Research Aim and Objectives

The aim of this thesis is to develop and evaluate adaptive task-scheduling approaches for coordinating storm-tracking and background-surveillance tasks in a weather-radar network while accounting for revisit urgency, dynamically updated storm priority, radar-dependent task duration, and changing radar workload.

To achieve this overarching aim, the following research objectives are defined:

1. Development of a simulation framework for evaluating radar task-scheduling methods under controlled and repeatable storm conditions.
2. Development of an adaptive radar-level scheduling method that combines Time-Balance urgency with dynamically updated storm priority, radar-dependent information, and task-feasibility constraints.
3. Development of a load-aware storm-to-radar assignment approach that distributes storm-tracking demand across multiple radars while accounting for scan duration, traverse time, storm urgency, and accumulated radar workload.
4. Comparison of the proposed adaptive scheduling methods with sequential storm-by-storm scanning and continuous 360° sit-and-spin scanning.
5. Evaluation of the scheduling methods under different storm conditions using revisit performance, tracking accuracy, surveillance coverage, and radar time-allocation metrics.
6. Assessment of the sensitivity of the load-aware assignment method to the frequency at which storm-to-radar assignments are recomputed.

1.4 Research Questions

The main research question of this thesis can be summarized to:

> *How can adaptive task scheduling be used to coordinate storm-tracking and background-surveillance tasks across a weather-radar network under varying storm and radar-load conditions?*

The main research question is supported by the following sub-questions:

RQ1: How can revisit urgency and dynamically changing storm characteristics be incorporated into radar task prioritisation?

RQ2: How can storm-tracking tasks be distributed across multiple radars while accounting for task duration, radar pointing requirements, and radar workload?

RQ3: How do the proposed adaptive scheduling approaches affect storm revisit performance, tracking accuracy, and retained surveillance compared with non-adaptive scanning strategies?

1.5 Contributions

This thesis contributes a simulation-based framework for investigating adaptive task scheduling in a weather-radar network. The main contributions are as follows:

1. **Dynamic storm-priority formulation:** A state-dependent storm-priority formulation is introduced in which priority is updated using the estimated distance to a predefined area of interest, storm size, and storm speed. This allows storm priority to change as the weather situation evolves.

2. **Adaptive radar-level task selection:** An adaptive scheduling method is developed that combines Time-Balance urgency with dynamic storm priority, radar-dependent task information, and feasibility constraints. The method selects storm-tracking tasks according to the current storm and radar state while retaining surveillance as a fallback activity.

3. **Load-aware storm-to-radar assignment:** A second adaptive scheduling method is developed by extending radar-level task selection with a planning-window-based assignment stage. Storm-tracking tasks are distributed among the radars using a cost that accounts for scan time, traverse time, storm urgency, and accumulated radar workload.

4. **Integrated tracking, surveillance, and multi-scenario evaluation:** Storm tracking and background surveillance are represented within the same scheduling framework, enabling the effect of increasing tracking demand on retained surveillance capability to be evaluated. The scheduling methods are assessed through Monte Carlo simulations across five characteristic storm scenarios that vary storm number, speed, and size.

1.6 Thesis Structure

The remainder of this thesis is organised as follows. Chapter 2 reviews the technical background and literature on weather-radar scanning, radar networks, adaptive sensing, and Time-Balance scheduling. Chapter 3 describes the simulation and system model, including the radar network, storm representation, scanning geometry, task timing, and tracking framework. Chapter 4 presents the baseline and adaptive scheduling methods, including dynamic storm priority and load-aware storm-to-radar assignment. Chapter 5 defines the experimental setup, storm scenarios, Monte Carlo design, and evaluation metrics. Chapter 6 presents the simulation results and compares the scheduling methods in terms of revisit performance, tracking accuracy, surveillance coverage, and radar time allocation. Chapter 7 discusses the main findings and limitations, while Chapter 8 summarises the conclusions and recommendations for future work.

Chapter 2

Background and Literature Review

This chapter presents the technical background and previous research relevant to adaptive task scheduling for weather-radar networks. The review focuses on the aspects of weather-radar operation that directly influence resource allocation: scan duration, temporal resolution, adaptive sector selection, radar-network coordination, task urgency, overload management, and surveillance retention.

The chapter first introduces the scanning constraints that create the need for adaptive scheduling. It then reviews fast and weather-adaptive scanning methods, distributed weather-radar networks, and broader radar-resource-management approaches. Particular attention is given to the Time-Balance (TB) scheduling framework and its extension for overload management and task prioritization, because these methods provide the main scheduling foundation for this thesis. The chapter concludes with a comparative synthesis and identifies the research gap addressed in the subsequent chapters.

2.1 Weather-Radar Scanning and Resource Constraints

Weather radar observes precipitation by transmitting electromagnetic pulses and processing the signals scattered by hydrometeors. The resulting measurements commonly include reflectivity, mean radial velocity, and spectrum width. These quantities describe precipitation intensity, motion along the radar line of sight, and variability in the measured radial velocities. Their quality depends on factors such as waveform design, pulse-repetition frequency, number of transmitted pulses, dwell time, signal-to-noise ratio, and the applied signal-processing method [1].

Spatial information is obtained by steering the radar beam over azimuth and elevation. In a Plan Position Indicator (PPI) scan, the radar scans in azimuth at a fixed elevation angle. Several PPI scans at different elevation angles can be combined to form a three-dimensional volume scan. The sequence of elevation angles, waveform settings, scan rates, and pulse-repetition settings is commonly defined through a volume-coverage pattern (VCP).

Radar scanning involves an inherent trade-off between spatial coverage, measurement quality, and temporal resolution. Increasing the number of elevation angles or angular samples can improve the spatial description of a weather system, but also increases the time

required to complete the scan. Increasing the scanning speed can reduce the update time, but may reduce the number of samples available for estimating radar moments. Therefore, improving temporal resolution generally requires either reducing the amount of spatial sampling, shortening the time required for individual measurements, or concentrating observations on selected regions.

This trade-off is particularly important for rapidly evolving convective weather. A complete volume scan may take longer than the time scale over which important changes in storm position, size, or internal structure occur. Consequently, conventional fixed scanning can provide systematic spatial coverage while still producing update intervals that are too long for selected weather features. Adaptive weather sensing attempts to address this limitation by allocating radar time according to the current weather situation rather than repeatedly observing the entire domain in the same manner.

From a scheduling perspective, every additional azimuth bin, elevation angle, dwell, or beam movement consumes radar time. The available radar time must therefore be divided among repeated observations of known storms and broader surveillance of regions in which new weather may appear. This resource constraint provides the motivation for adaptive scanning and task-scheduling approaches.

2.2 Fast and Adaptive Weather-Radar Scanning

Research on improved weather-radar temporal resolution has followed several complementary directions. These include adapting predefined scan patterns, reducing the measurement time required for individual observations, and redirecting radar measurements towards meteorologically significant regions. Although these approaches operate at different levels, each increases the flexibility available to the radar scheduler.

Brown et al. investigated the elevation-angle distributions used in convective-storm VCPs for the WSR–88D [7]. Fixed elevation-angle distributions can create range-dependent vertical-sampling errors because the vertical spacing between radar beams changes with distance. Their optimised VCPs redistributed the elevation angles to obtain more consistent estimates of storm-top height over different ranges. Flexible variants also allowed high-elevation scans to be omitted when significant echoes were absent at lower elevations. This reduced unnecessary scanning and shortened the volume-update time.

This work demonstrates that scan time can be reduced by adapting the scan geometry to the observed weather structure. However, the main adaptive decision concerns the configuration and termination of the volume scan. Individual storm regions are not formulated as separate tasks with independent revisit requirements.

A second research direction reduces the time required to obtain a radar measurement. Curtis and Torres introduced adaptive range oversampling with pseudowhitening for the National Weather Radar Testbed phased-array radar in the US [20]. By using oversampled range data and adapting the transformation according to the observed signal statistics, the method reduces the variance of meteorological-variable estimates. This allows the required

number of pulses per dwell to be reduced while maintaining a desired level of estimation accuracy.

Reducing dwell time increases the radar time available for scanning and can therefore improve temporal resolution. Nevertheless, this improvement does not determine how the newly available radar time should be divided among multiple storm-tracking and surveillance requests. Faster measurement acquisition increases radar capacity, while a scheduling method is still required to allocate that capacity.

Phased-array radar technology provides further flexibility through rapid electronic beam steering. A phased-array antenna can change beam direction without following the continuous mechanical rotation of a conventional reflector. As a result, observations from spatially separated regions can be interleaved, and selected weather features can be revisited more frequently than the complete radar domain.

Torres et al. presented adaptive weather-surveillance and multifunction capabilities implemented on the National Weather Radar Testbed phased-array radar in Norman, Oklahoma, USA [8]. The system included automatic pedestal orientation, beam-significance-based scanning, storm-cluster subvolume scans, adaptive vertical sampling, and adaptive pulse-repetition-time selection. The reported demonstrations showed that general weather observations could be updated on approximately one-minute time scales, while selected convective regions could receive updates on the order of ten seconds. Weather and aircraft-surveillance functions were also interleaved within a multifunction schedule.

These results show that weather-adaptive scanning can substantially improve the temporal resolution of selected regions without requiring the full radar domain to be scanned at the same rate. Significant-beam and storm-cluster methods determine where focused observations should be performed, while periodic detection scans preserve broader weather awareness.

However, the identification of significant weather regions does not by itself determine how several competing observations should be ordered. When multiple storms require updates at similar times, the radar must still decide which task should be executed first. The decision becomes more difficult when the tasks differ in duration, requested revisit interval, storm importance, and beam-movement requirement. Adaptive scan construction therefore provides the observations that may be performed, while task scheduling determines when those observations are executed and how finite radar time is shared.

2.3 Weather-Radar Networks and Collaborative Sensing

A single long-range weather radar can provide broad spatial coverage, but its observation capability varies across the domain. As range increases, the radar beam broadens and generally rises above the surface because of the antenna elevation angle, Earth curvature, and atmospheric refraction. This can reduce the ability to observe low-altitude weather phenomena at long distances. Terrain blockage and local propagation conditions may create further coverage limitations.

Weather-radar networks address these limitations by distributing several radar nodes over the observation region. A weather feature located far from one radar may be observed at a shorter range by another. This can reduce beam height and beam size at the location of interest. Overlapping coverage also provides observations from different viewing directions and can introduce redundancy when one radar has an unfavourable geometry.

Junyent and Chandrasekar developed a general framework for characterising weather-radar networks using both individual-radar properties and network topology [14]. Their formulation considered elemental network cells defined by the number of radars and the overlap among their coverage regions. Spatial density functions were used to characterise properties including minimum beam height, effective beam size, detection sensitivity, radar density, and the number of overlapping observations.

Their analysis showed that networks of smaller short-range radars can provide lower minimum beam heights, smaller effective beam sizes, and more uniform coverage than a single long-range radar over a comparable region. The study establishes the sensing benefits that can result from distributed radar placement. However, it does not prescribe how observation tasks should be allocated once several radar nodes are capable of observing the same weather feature.

Regional radar deployments have demonstrated both the potential and practical challenges of distributed weather sensing. Antonini et al. described a regional X-band weather-radar network along the Tuscan coast [21]. The network provided high-resolution observations of small-scale precipitation structures and complemented existing national radar systems. The study also identified practical considerations related to attenuation, clutter, calibration, scan configuration, and measurement compositing. These results show that network performance depends on both sensor placement and the operational strategy used by the individual radar nodes.

The Center for Collaborative Adaptive Sensing of the Atmosphere (CASA) developed a networked weather-radar system based on Distributed Collaborative Adaptive Sensing (DCAS) [18]. Wang et al. proposed a scan strategy for dual-Doppler wind retrieval in which storm polygons and precomputed geometric-error fields were used to determine the azimuth sectors assigned to individual radars during a common network heartbeat [19]. This work demonstrates that weather location and network geometry can be used jointly to coordinate sector scans across multiple radar nodes.

The objective of the method, however, is primarily to obtain suitable dual-Doppler geometry for wind retrieval. The task-allocation decision is therefore driven by a specific measurement objective. It does not formulate a general scheduler for several independent storm-tracking tasks with different revisit urgency, priority, execution time, and radar-workload conditions.

The reviewed network literature therefore distinguishes two related but separate questions. Network-coverage analysis determines which observations are possible and which radar geometries are favourable. Network scheduling must additionally determine which observations should be performed, which radar should perform them, and when they should be

executed. This distinction becomes important when several radars can observe the same storm and several storms compete simultaneously for radar time.

2.4 Radar Resource Management and Cognitive Approaches

Radar-resource management concerns the allocation of finite sensing resources among competing functions. Depending on the radar system, these resources may include time, energy, waveform choices, beam directions, processing capacity, and communication bandwidth. A multifunction radar may need to divide these resources among surveillance, detection, tracking, classification, and other sensing tasks. Radar-resource management is a well-established problem in surveillance and target-tracking applications, where finite radar resources must be allocated among multiple competing sensing tasks [22, 23, 24].

Much of the broader radar-resource-management literature has been developed for surveillance and tracking of discrete targets rather than for meteorological sensing. In these applications, resource-allocation decisions are commonly related to target-tracking performance, target priority, and the management of multiple competing radar tasks.

The broader radar literature includes several adaptive and cognitive architectures for managing these decisions. The Cognitive Fully Adaptive Radar concept proposed a Sense–Learn–Adapt architecture in which environmental knowledge, prior information, and real-time measurements are used to adapt radar operation [25]. Resource allocation is embedded within a feedback process that changes radar behaviour according to the observed environment and the information obtained from previous measurements.

Smits et al. extended cognitive resource-management concepts to a multi-platform radar network [26]. Their architecture allowed information to be shared at different levels and used reinforcement learning to develop mode-selection strategies for a surveillance problem. The work demonstrates that learning-based methods can discover coordinated network policies that may not be obvious from manually designed rules.

Charlish et al. reviewed the development from adaptive to cognitive radar-resource management and discussed knowledge-based methods, quality-of-service formulations, utility functions, stochastic control, partially observable Markov decision processes, and learning-based approaches [27]. A central principle in these approaches is that radar resources should be allocated according to the expected quality or operational value of the resulting information rather than according to a fixed sequence of actions.

Weather-radar scheduling can be viewed as a specialised radar-resource-management problem in which radar time is the principal constrained resource. Storm-tracking tasks consume time for beam movement and sector scanning, while background surveillance consumes time for observing regions without currently identified storms. The total demand changes as storms enter, leave, grow, or move through the observation domain. Although weather-radar resource management shares the same underlying constraint of limited sensing resources, it differs from the target-oriented applications discussed above because the objects of interest are spatially extended and evolving weather regions, while background

surveillance must also be maintained.

These methods provide powerful frameworks for multi-objective and multi-platform decision making. However, their application to weather-radar scheduling requires suitable state representations, utility or reward functions, uncertainty models, and, in some cases, training environments. In weather-radar scheduling, the operational value of a measurement may depend on storm severity, forecast impact, warning requirements, and downstream processing. Representing all of these effects in one validated objective function is not straightforward.

For the weather-radar scheduling problem considered in this thesis, interpretable rule- and model-based methods provide a useful basis for examining the direct effects of revisit urgency, task duration, radar geometry, priority, and workload. Although several of the cognitive and resource-management approaches discussed above originate from target-surveillance applications, their general principles remain relevant to weather-radar scheduling. Cognitive and learning-based approaches therefore remain potential future extensions, but they are not the principal methodological basis of the present work.

2.5 Scheduling Algorithms for Adaptive Weather Sensing

Once weather observations are represented as separate tasks, a scheduling algorithm is required to determine their execution order. A weather-radar task may be described by its execution time, requested update interval, current waiting time, priority, and radar-dependent geometric cost. Because the weather environment and radar state evolve continuously, the scheduler must operate online rather than relying only on a predefined, fixed sequence prepared in advance.

Let tracking task i require an execution time T_i and request an update interval U_i. The fraction of radar time requested by the task can be represented by its occupancy:

$$O_i = \frac{T_i}{U_i}. \tag{2.1}$$

A task with a long execution time or a short requested update interval therefore consumes a larger fraction of the available radar time. When the combined occupancy of the requested tasks exceeds the available capacity, the radar is overloaded and not all requested update intervals can be maintained simultaneously, meaning that some tasks cannot be executed leading to some performance loss.

2.5.1 Time-Balance Scheduling

Reinoso-Rondinel et al. introduced a TB scheduler for adaptive weather sensing using a multifunction phased-array radar [12]. The study considered two weather-radar functions: repeated observations of identified storm cells and surveillance of regions without known storms. Each tracking task was characterised by a task execution time and a requested update time.

Tracking tasks were treated as non-interruptible, whereas the surveillance task could be

divided into smaller fragments and interleaved between tracking observations. This formulation allowed the radar to concentrate observations on identified storms while using the remaining radar time for broader weather detection.

In their work, the central scheduling quantity is the time balance associated with each tracking task. Time balance represents the timing state of a task relative to its requested update interval. A positive time balance indicates that a task is due or overdue, while a negative value indicates that the task has been serviced ahead of its requested timing. The value increases as the task waits and is reduced when the task is executed.

The scheduler selects tasks according to their time-balance values rather than following a fixed cyclic order. As a result, tasks that have waited longer relative to their requested update interval become more urgent. This allows different storm cells to request different update intervals and provides a direct and interpretable representation of revisit urgency.

The TB formulation is suitable for online radar scheduling because it does not require complete prediction of the future schedule. The scheduling state has a clear physical interpretation, and tracking observations can be interleaved with fragmented surveillance. The original study [12] evaluated the adaptive-sensing benefit using revisit-improvement and acquisition-time measures and showed that storm regions could be revisited more frequently than under conventional volume scanning when the requested demand remained feasible.

A limitation of the original formulation is its behaviour during overload. When the total requested tracking occupancy approaches or exceeds the available radar capacity, the amount of radar time assigned to surveillance can decrease substantially. Some tracking tasks may also remain delayed because changing the task order alone cannot remove overload when the requested demand remains greater than the available time.

The formulation was also developed for one stationary phased-array radar. Therefore, the scheduler determines which task should be executed next, but it does not need to determine which radar should perform the task. Radar-dependent task costs and the distribution of work across several nodes are essentially outside the scope of the original formulation.

2.5.2 Overload Management and Task Prioritisation

Reinoso-Rondinel et al. subsequently extended their original TB framework to improve scheduling behaviour under overload [13]. The extended method introduced two main mechanisms: adaptation of requested task-update intervals and the use of task-priority levels in addition to time balance.

In the update-time-adjustment approach, requested update intervals are increased when the radar is overloaded. Since task occupancy is proportional to T_i/U_i, increasing U_i reduces the requested radar-time fraction. When sufficient radar capacity becomes available, the requested update intervals can be reduced again. This mechanism adapts the requested service rate to the available radar resources instead of allowing task delays to increase indefinitely.

The second mechanism introduces task-priority levels. The scheduler first considers the priority assigned to each task and then uses time balance to distinguish between tasks at the same priority level. Higher-priority tasks can therefore remain closer to their requested

update intervals under overload, while lower-priority tasks absorb a larger share of the delay.

Priority and urgency represent different scheduling concepts. Time balance describes whether a task is due relative to its requested service interval. Priority represents the relative importance assigned to the task. A high-priority task may have low urgency immediately after execution, while a lower-priority task may become highly overdue after waiting for a long period. Combining the two quantities allows the scheduler to account for both timing requirements and operational importance.

The overload extension also includes a surveillance-protection mechanism. When surveillance reaches a specified maximum update interval, its priority can be increased so that it is forced into the schedule. This prevents repeated storm tracking from indefinitely suppressing the broader weather-detection function.

In the adaptive TB formulation, task-priority levels are externally assigned parameters that can be changed according to the operational situation. As a limitation, this method does not prescribe how weather-object priority should be calculated automatically from changing estimated storm characteristics. Furthermore, the extended scheduler remains a single-radar formulation and does not address how tasks should be distributed when several radar nodes can observe the same storm.

2.6 Literature Synthesis and Research Gaps

The reviewed studies address different layers of adaptive weather sensing. Flexible VCP design changes the geometry or termination of a scan. Faster measurement methods reduce the acquisition time of individual observations. Weather-adaptive phased-array methods identify significant beams and storm subvolumes. Radar-network studies improve coverage and, in some cases, coordinate observations for specific measurement objectives. TB scheduling determines the execution order of competing weather tasks, while its adaptive extension introduces overload management and priority.

These contributions are complementary, but they do not address the same decision problem. Table 2.1 summarises the main approaches according to their scheduling level, principal contribution, and limitation relevant to this thesis.

Within the scope of the reviewed literature, three main research gaps are identified.

1. **Multi-radar allocation of competing storm-tracking tasks.**

   Weather-radar-network studies mainly characterise the sensing benefits of distributed coverage or coordinate observations for specific retrieval geometries. In contrast, the reviewed TB methods schedule competing weather tasks for a single radar. A general online mechanism for allocating several storm-tracking tasks across multiple feasible radar nodes while considering urgency, radar-dependent execution time, pointing requirements, and radar workload is not provided by the reviewed approaches.

2. **State-dependent storm priority.**

Table 2.1: Comparison of approaches relevant to adaptive weather-radar scheduling.

Study or approach	**Decision level**	**Main contribution**	**Limitation relevant to this thesis**
Brown et al. [7]	Volume-scan configuration	Improves elevation-angle placement and removes unnecessary high-elevation scans.	Adapts the volume pattern rather than scheduling independent storm tasks.
Curtis and Torres [20]	Measurement acquisition	Reduces the number of pulses and dwell time required for radar-variable estimation.	Increases available radar capacity but does not allocate that capacity among competing tasks.
Torres et al. [8]	Adaptive scan construction	Selects significant beams and focused storm subvolumes while retaining weather detection.	Does not formulate general multi-radar allocation of independent storm tasks.
Junyent and Chandrasekar [14]	Network coverage	Quantifies distributed-radar benefits in beam height, beam size, sensitivity, and overlap.	Characterises sensing capability but does not provide online task assignment.
Wang et al. [19]	Collaborative sector selection	Coordinates radar sectors according to storm location and dual-Doppler geometry.	Uses a specific wind-retrieval objective rather than general urgency- and workload-aware scheduling.
Base TB scheduler [12]	Single-radar task scheduling	Uses time balance to schedule tracking and fragmented surveillance tasks with different update intervals.	Has limited overload treatment and does not require storm-to-radar assignment.
Adaptive TB scheduler [13]	Single-radar task scheduling	Adds update-time adaptation, task-priority levels, and surveillance protection.	Uses externally assigned priority and does not distribute tasks across multiple radar nodes.
Cognitive and utility-based RRM [25, 26, 27]	General radar-resource management	Supports utility-based, stochastic, and learned allocation of radar resources.	Primarily developed for surveillance and tracking of man-made targets rather than weather phenomena, and requires suitable state, reward, utility, transition, or training models for the selected application.

The adaptive TB framework supports priority levels, but these levels are externally assigned. The reviewed method does not prescribe a transparent procedure for automatically updating task priority from changing estimated storm characteristics. Such a mechanism is relevant when storm importance changes as a function of motion, size, or location.

3. **Joint evaluation of storm tracking and retained surveillance.**

   Focused adaptive scanning can improve storm revisit frequency, but allocating additional radar time to tracking can reduce the time available for broader surveillance. The reviewed studies do not jointly address multi-radar storm allocation and evaluate its effect on both storm-revisit performance and retained surveillance within one task-scheduling framework.

These gaps motivate the investigation of an interpretable multi-radar scheduling framework that combines revisit urgency with state-dependent storm priority and explicit storm-to-radar coordination. The purpose is not to replace more general cognitive or optimisation-based resource-management approaches, but to examine how observable scheduling quantities such as task duration, radar pointing, storm urgency, and radar workload influence network behaviour.

The simulation environment used for this investigation is presented in Chapter 3. The baseline and adaptive scheduling methods are then defined in Chapter 4.

Chapter 3

Simulation and System Model

The purpose of this chapter is to describe the simulation environment used to evaluate the scheduling methods in this thesis. Since the objective is to study radar task allocation, the simulator is designed to retain the geometric, timing, and tracking constraints that directly affect scheduling decisions. The chapter therefore describes the radar-network geometry, storm-cell representation, azimuth scanning model, timing assumptions, and state-estimation model used by the schedulers.

Importantly, the simulator is intended as a controlled scheduling testbed rather than a complete operational weather-radar simulator. It simplifies several aspects of real radar operation, including elevation scanning, beam-shape effects, attenuation, clutter filtering, Doppler moment estimation, and detailed storm microphysics. These simplifications allow the comparison to focus on how different scheduling policies allocate limited radar time between storm tracking and surveillance, while keeping the computational load and complexity tractable.

3.1 Overview of the Simulation Framework

The simulation framework is designed to evaluate how different scheduling strategies allocate radar time between storm tracking and background surveillance. The simulation is performed in a two-dimensional horizontal plane, where storm cells move through a bounded domain and are observed by a fixed network of radars. At each dwell interval, every radar either continues an active task, starts a new storm-tracking task, or performs a surveillance scan. After the scheduling decision is made, the simulation time is advanced by one dwell interval and the storm states are updated according to their motion model.

Figure 3.1 provides an overview of the simulation framework and shows the information flow between the storm model, radar network, tracker, and scheduler.

The radar-network model is described in Section 3.2, followed by the storm cell model in Section 3.3, the scanning and timing models in the subsequent sections, the tracking model in Section 3.6, and the task definitions and scheduler inputs in Section 3.7. The storm environment defines the initial position, radius, velocity, reflectivity, and growth rate of each storm cell. The radar network defines the radar locations, current pointing directions, maximum range, and azimuth grid. The scheduler receives the current radar state and the

estimated storm-state information provided by the tracking module and selects the next task for each radar based on the scheduling method being evaluated.

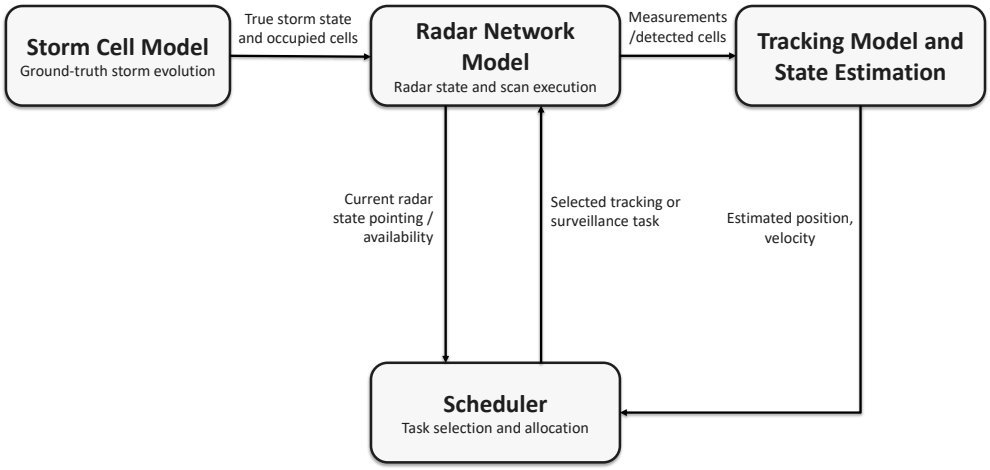


Figure 3.1: Overall simulation framework showing the interaction between the storm cell model, radar network model, tracking model, and scheduler, and the information exchanged between them.

The same simulation framework is used for both single-run examples and Monte Carlo experiments. In the Monte Carlo experiments, the initial storm positions, radii, velocities, reflectivities, and radial growth rates are randomly sampled for each trial according to the scenario definitions. The resulting underlying storm evolution is treated as the ground truth and is kept identical across all compared scheduling methods. This shared-truth design ensures that differences in performance can be attributed to the scheduling logic rather than to differences in the generated storm scenario.

Within the simulation, a radar task represents a sensing activity assigned to a radar. Two task types are considered: storm-tracking tasks and background-surveillance tasks. A storm-tracking task consists of traversing from the radar's current pointing direction towards a selected storm sector and subsequently scanning all azimuth bins belonging to that sector. Once started, the task remains active over successive dwell intervals until the complete storm sector has been scanned. A background-surveillance task consists of scanning one currently unoccupied azimuth bin during a dwell interval. A completed storm-tracking task can generate a storm measurement, whereas a surveillance task does not directly update an existing storm track.

3.2 Radar Network Model

The radar network is represented by a set of fixed radar nodes in a two-dimensional Cartesian coordinate system. The horizontal position of radar j in the network is denoted by

$$\mathbf{p}_j^r = \begin{bmatrix} x_j^r & y_j^r \end{bmatrix}^T. \tag{3.1}$$

Each radar node is assigned a maximum observation range, denoted by

$$R_{\max,j}, \tag{3.2}$$

where the subscript j refers to radar j. Storm observability is evaluated at the resolution-cell level rather than solely from the position of the storm centre. Let $\mathcal{C}_i$ denote the set of resolution cells occupied by storm i. The subset of storm cells lying within the observation range of radar j is defined as

$$\mathcal{C}_{ij}^{\text{vis}} = \left\{ c \in \mathcal{C}_i \;\middle|\; \left\| \mathbf{p}_c - \mathbf{p}_j^r \right\| \leq R_{\max,j} \right\}, \tag{3.3}$$

where $\mathbf{p}_c$ is the position of the centre of resolution cell c. A storm is considered at least partially observable by radar j when

$$\mathcal{C}_{ij}^{\text{vis}} \neq \varnothing. \tag{3.4}$$

Therefore, a valid observation can be obtained even when only part of the storm lies within the radar coverage. In such cases, the measured centroid and spatial extent, calculated from the detected resolution cells, may differ from the true centre and radius of the simulated circular storm.

For the final simulation configuration, two radars are considered and both are assigned the same maximum observation range:

$$R_{\max,1} = R_{\max,2} = R_{\max} = 30 \text{ km}. \tag{3.5}$$

The value of $R_{\max} = 30$ km was selected to provide a representative short-range radar-network configuration while keeping the simulation domain and storm–radar geometry sufficiently compact for the scheduling study. The radar locations were then chosen symmetrically at $R_{\max}/2 = 15$ km from the origin, giving a radar separation of 30 km and an overlapping central coverage region in which storms can be observed by either radar. These values are not intended to represent an optimal radar placement, but provide a controlled geometry for evaluating the effect of the scheduling methods.

The two radars are positioned symmetrically about the origin on a circle with radius $R_{\max}/2$. Their positions are therefore defined as

$$\mathbf{p}_1^r = (15, 0) \text{ km}, \qquad \mathbf{p}_2^r = (-15, 0) \text{ km}. \tag{3.6}$$

Each radar is modelled as an azimuth-scanning sensor restricted to the horizontal plane. The radar model does not explicitly include elevation scanning, vertical storm structure, beam-shape effects, attenuation, or clutter. This simplification is appropriate for the present study because the main objective is to evaluate how scheduling methods distribute horizontal sector-scanning tasks across the radar network.

Moreover, each radar in the network is subject to a finite angular scanning rate. Therefore, a radar cannot move instantaneously from one azimuthal sector to another. To scan a different sector, the radar must traverse from its current pointing direction towards the newly selected scan sector, and this will cost an amount of time. This constraint is incorporated into the simulation and is considered later in the scheduling formulation.

3.3 Storm Cell Model

Each storm is ideally represented as a circular region in the horizontal plane. This simplified representation allows the storm footprint to be described using a centre position and radius without requiring a more complex spatial model. Accordingly, storm i is represented by the following set of variables:

$$\mathcal{S}_i = \{\mathbf{p}_i^s, r_i, \mathbf{v}_i^s, Z_i, g_i\}, \tag{3.7}$$

where

$$\mathbf{p}_i^s = \begin{bmatrix} x_i^s & y_i^s \end{bmatrix}^T \tag{3.8}$$

is the storm-centre position, r_i is the storm radius,

$$\mathbf{v}_i^s = \begin{bmatrix} v_{x,i}^s & v_{y,i}^s \end{bmatrix}^T \tag{3.9}$$

is the horizontal storm velocity, Z_i is the storm reflectivity, and g_i is the radial growth rate.

The circular storm footprint is represented on the radar grid by the set of resolution cells whose centres lie inside the storm region. Let $\mathcal{C}_i(t)$ denote the set of resolution cells occupied by storm i at time t:

$$\mathcal{C}_i(t) = \{c \mid \|\mathbf{p}_c - \mathbf{p}_i^s(t)\| \leq r_i(t)\}, \tag{3.10}$$

where $\mathbf{p}_c$ is the Cartesian position of the centre of resolution cell c. This cell-based representation is used later to determine partial storm observability and to form centroid measurements from detected storm-occupied cells.

The storm-centre position evolves according to a constant-velocity model:

$$\mathbf{p}_i^s(t + \Delta t) = \mathbf{p}_i^s(t) + \mathbf{v}_i^s \Delta t. \tag{3.11}$$

The storm velocity remains constant throughout an individual simulation trial. Therefore, the model represents translational storm motion without explicitly modelling acceleration or

changes in propagation direction.

Changes in storm size are represented using the radial growth rate g_i. The storm radius evolves according to

$$r_i(t + \Delta t) = \max\left(0.1, r_i(t) + g_i \Delta t\right). \tag{3.12}$$

For the Monte Carlo experiments, the radial growth rate is independently sampled for each storm from a uniform distribution:

$$g_i \sim \mathcal{U}\left(0.001, 0.006\right) \text{ km/s}. \tag{3.13}$$

The selected range corresponds to radial expansion rates between 1 and 6 m/s. Since only positive growth-rate values are considered, the simulated storms expand gradually during a trial and do not contract. The lower bound of 0.1 km in the radius-update equation is retained as a numerical safeguard.

The reflectivity value is stored as part of the storm state but is not used directly in the scheduling decisions considered in this thesis. Retaining this variable allows future extensions in which storm intensity contributes to the priority formulation or measurement model.

Notably, several simplifying assumptions follow from this representation. Storms remain circular, do not merge or split, and retain a constant horizontal velocity and reflectivity during a simulation trial. These assumptions are not intended to reproduce the complete physical evolution of convective storms. Instead, they keep the scheduling problem interpretable and preserve a direct relationship between storm geometry, the angular sector requiring observation, and the corresponding radar scan duration.

3.4 Scanning Geometry and Azimuth Bins

The radars scan in azimuth using the meteorological convention in which 0° points North and angles increase clockwise. The azimuth grid is discretised with spacing:

$$\Delta\theta = 0.5^\circ. \tag{3.14}$$

This results in:

$$N_\theta = \frac{360^\circ}{0.5^\circ} = 720 \tag{3.15}$$

azimuth bins over a full scan.

The storm-centre measurement is derived from the resolution cells detected during the completed storm-sector scan. Let $\mathcal{D}_{ij}(t)$ denote the set of storm-occupied resolution cells detected by radar j for storm i at time t. This set represents the radar-observed subset of the true storm occupancy $\mathcal{C}_i(t)$ defined previously, such that $\mathcal{D}_{ij}(t) \subseteq \mathcal{C}_i(t)$. The two sets are equal when the complete storm footprint is observed, whereas a partial observation contains only a subset of the true storm-occupied cells.

Each detected polar resolution cell is represented by the Cartesian coordinates of its

centre,

$$\mathbf{p}_c = \begin{bmatrix} x_c & y_c \end{bmatrix}^T. \tag{3.16}$$

The measured storm centroid is computed as the arithmetic mean of the centres of all detected resolution cells:

$$\mathbf{z}_{ij}^p(t) = \frac{1}{|\mathcal{D}_{ij}(t)|} \sum_{c \in \mathcal{D}_{ij}(t)} \mathbf{p}_c, \qquad |\mathcal{D}_{ij}(t)| > 0, \tag{3.17}$$

where $|\mathcal{D}_{ij}(t)|$ is the number of detected storm-occupied resolution cells. Since the centroid is calculated only from the detected part of the storm, a partial observation may produce a measured centroid that differs from the true storm-centre position.

The measured centroid is provided to the Kalman-filter-based tracker described in Section 3.6. The current Kalman-filter estimate of the storm-centre position is denoted by

$$\hat{\mathbf{p}}_i^s = \begin{bmatrix} \hat{x}_i^s & \hat{y}_i^s \end{bmatrix}^T. \tag{3.18}$$

The corresponding radius estimate is denoted by $\hat{r}_i$. These estimates, rather than the true storm-centre position and radius, are used to define the radar–storm geometry required by the scheduler.

For a storm with Kalman-filter centre-position estimate $\hat{\mathbf{p}}_i^s$ and radius estimate $\hat{r}_i$, the relative position from radar j to storm i is defined as

$$\Delta\mathbf{p}_{ij} = \hat{\mathbf{p}}_i^s - \mathbf{p}_j^r = \begin{bmatrix} \Delta x_{ij} & \Delta y_{ij} \end{bmatrix}^T. \tag{3.19}$$

and thus the corresponding radar-to-storm range is:

$$d_{ij} = \|\Delta\mathbf{p}_{ij}\|_2 . \tag{3.20}$$

The centre azimuth of the storm sector is computed as

$$\theta_{ij} = \left(\operatorname{atan2}(\Delta x_{ij}, \Delta y_{ij}) \frac{180}{\pi} + 360 \right) \bmod 360. \tag{3.21}$$

The use of $\operatorname{atan2}(\Delta x, \Delta y)$ follows the adopted azimuth convention in which 0° corresponds to the positive y direction. In this form, the operator gives the directed angle between the positive y-axis and the radar-to-storm displacement vector, with the azimuth increasing clockwise and expressed over the interval $[0^\circ, 360^\circ)$.

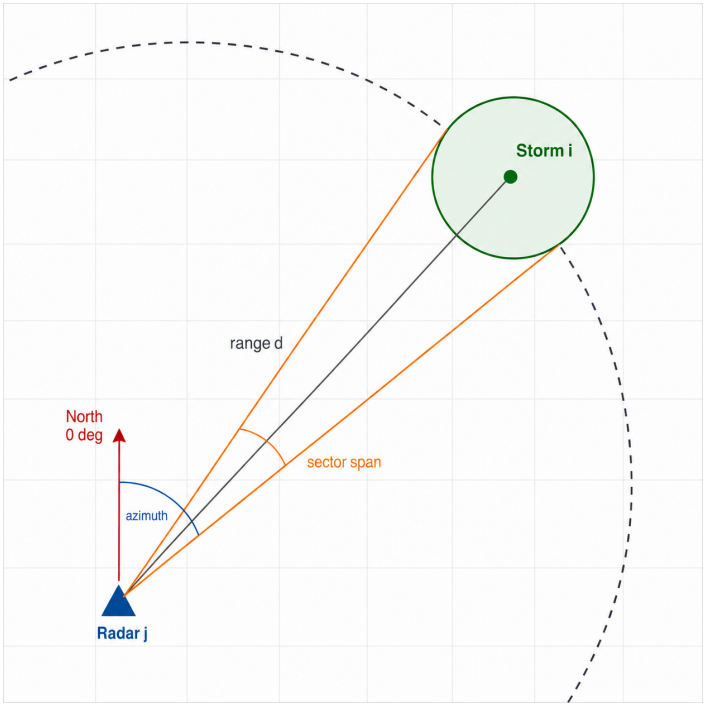


Figure 3.2: Radar–storm geometry used to determine the sector scan for storm i as observed by radar j. The angle Θ_{ij} denotes the azimuthal sector span required to cover the storm.

Since the storm has a finite radius, it occupies an angular sector rather than a single azimuth bin. The half angular span is approximated as

$$\alpha_{ij} = \tan^{-1}\left(\frac{\hat{r}_i}{d_{ij}}\right)\frac{180}{\pi}. \tag{3.22}$$

The occupied storm sector then spans:

$$\Theta_{ij} = [\theta_{ij} - \alpha_{ij}, \theta_{ij} + \alpha_{ij}], \tag{3.23}$$

with wrap-around at 0° and 360°. The storm-sector scan for storm i as observed by radar j consists of all azimuth bins that fall within this sector. The geometry used for this sector definition is illustrated in Figure 3.2.

3.5 Dwell Time and Traverse Time

The timing model determines how long a radar spends on each azimuth bin and how long it takes to move between different task directions. The dwell time for a single azimuth bin is defined by the pulse repetition frequency and the number of pulses transmitted per dwell:

$$T_{\text{dwell}} = \frac{N_p}{\text{PRF}}. \tag{3.24}$$

In the final simulations,

$N_p = 100$ and PRF $= 1111$ Hz which gives a dwell time per bin of $T_{\text{dwell}} \approx 0.09$ s. Then,

since the full azimuth grid contains 720 bins, the time required for a complete 360° scan is:

$$T_{360} = 720 \cdot T_{\text{dwell}} \approx 64.8 \text{ s}. \tag{3.25}$$

This value is later used as the window length for revisit-rate and surveillance-coverage evaluation, because it corresponds to one complete conventional azimuth scan under the chosen timing settings.

When a radar is required to scan a different azimuth sector, it may first need to move from its current azimuth θ_{cur} to a selected target azimuth θ_{tar}. The angular distance is:

$$\Delta\theta_{\text{trav}} = \min\left(|\theta_{\text{tar}} - \theta_{\text{cur}}|, 360^\circ - |\theta_{\text{tar}} - \theta_{\text{cur}}|\right). \tag{3.26}$$

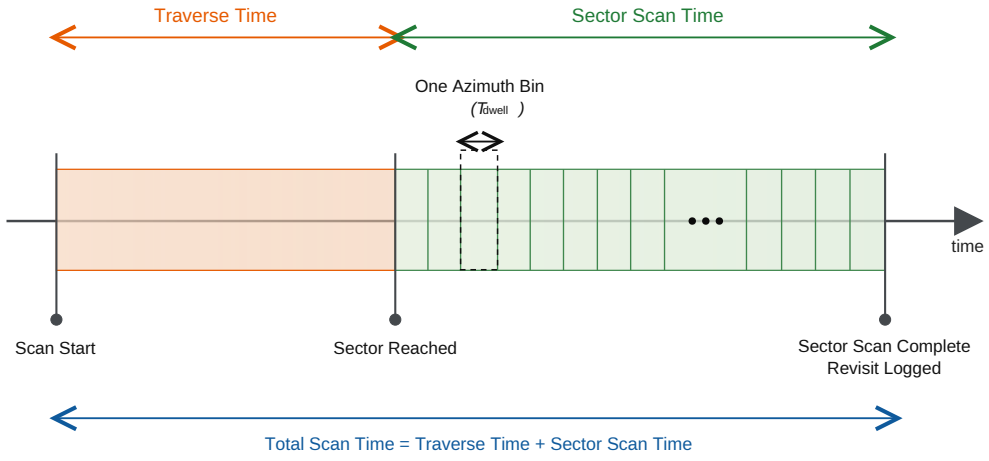


Figure 3.3: Traverse and dwell-time components of a storm-sector scan.

In the simulation, the radar is assigned an azimuthal traverse rate of $\omega = 20°/\text{s}$. This parameter represents the assumed rate at which the radar can reorient between azimuth sectors. This value was selected as an empirical modelling choice to provide a representative traverse time within the simulation. It is not intended to reproduce the mechanical characteristics of a specific radar system and can be readily adjusted in the simulator for different radar configurations. The corresponding continuous traverse time is

$$T_{\text{trav}} = \frac{\Delta\theta_{\text{trav}}}{\omega}. \tag{3.27}$$

In the implementation, this time is rounded up to an integer number of dwell intervals:

$$N_{\text{trav}} = \left\lceil \frac{T_{\text{trav}}}{T_{\text{dwell}}} \right\rceil. \tag{3.28}$$

The total time required for radar j to reorient towards and scan the azimuth sector associated with storm i combines the traverse time and the sector-scan time. It is therefore

written as

$$T_{ij}^{\text{scan}} = N_{\text{trav}} T_{\text{dwell}} + N_{ij}^{\text{az}} T_{\text{dwell}}, \tag{3.29}$$

where N_{trav} is the number of dwell intervals required for azimuthal reorientation and N_{ij}^{az} is the number of azimuth bins included in the sector associated with storm i as viewed from radar j. The corresponding timing structure is illustrated in Figure 3.3.

3.6 Tracking Model and State Estimation

Storm state estimation is performed using a Kalman filter. The scheduler obtains estimated storm position, velocity, and radius through the tracking module. These estimates are used to determine the storm sectors, task durations, storm priorities, and radar–storm geometry required by the schedulers.

The simulation assumes that all storms are already known at the beginning of each trial. Therefore, each Kalman track is initialised once at $t = 0$ using the initial simulated storm position, velocity, radius, and radial growth rate. After this initialisation, the simulated storm state is not used as a direct fallback for the tracker. Subsequent corrections are obtained only from measurements generated after completed storm-sector scans.

For each storm, the state vector is defined as

$$\mathbf{x}_{i,k} = \begin{bmatrix} x_{i,k}^s & v_{x,i,k}^s & y_{i,k}^s & v_{y,i,k}^s & r_{i,k} & \dot{r}_{i,k} \end{bmatrix}^T, \tag{3.30}$$

where $x_{i,k}^s$ and $y_{i,k}^s$ are the horizontal coordinates of the storm centre, $v_{x,i,k}^s$ and $v_{y,i,k}^s$ are the corresponding velocity components, $r_{i,k}$ is the storm radius, and $\dot{r}_{i,k}$ is its rate of change.

Under the known-initial-track assumption, the initial state estimate is

$$\hat{\mathbf{x}}_{i,0} = \begin{bmatrix} x_i^s(0) & v_{x,i}^s(0) & y_i^s(0) & v_{y,i}^s(0) & r_i(0) & g_i \end{bmatrix}^T, \tag{3.31}$$

where g_i is the true radial growth rate assigned to storm i in the simulation. Under the known-initial-track assumption, this rate is assumed to be known at $t = 0$ and is therefore used to initialise the corresponding radius-rate state. After initialisation, the simulated growth rate is not provided directly to the tracker; subsequent state estimates are obtained through the Kalman prediction and measurement-update process. The initial state covariance used in the implementation is

$$\mathbf{P}_{i,0} = \text{diag}\,(1.0,\, 0.5,\, 1.0,\, 0.5,\, 0.75,\, 0.25)\,. \tag{3.32}$$

The initial covariance values were selected empirically to provide moderate initial uncertainty in the different state components and were kept fixed for all simulated storm tracks. The prediction model assumes constant horizontal velocity and constant radius rate between measurement updates. The predicted state at time step k is

$$\hat{\mathbf{x}}_{i,k|k-1} = \mathbf{F}(\Delta t)\hat{\mathbf{x}}_{i,k-1|k-1}, \tag{3.33}$$

where Δt is the elapsed time since the previous prediction or measurement update. For the state ordering used in Eq. 3.30, the transition matrix is

$$\mathbf{F}(\Delta t) = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.34}$$

The corresponding covariance prediction is

$$\mathbf{P}_{i,k|k-1} = \mathbf{F}(\Delta t)\mathbf{P}_{i,k-1|k-1}\mathbf{F}^T(\Delta t) + \mathbf{Q}(\Delta t), \tag{3.35}$$

where $\mathbf{Q}(\Delta t)$ is the process-noise covariance matrix. For a position–velocity state pair, the process-noise block is defined as

$$\mathbf{Q}_a(\Delta t, \sigma) = \sigma^2 \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix}. \tag{3.36}$$

Independent process-noise blocks are used for the x motion, the y motion, and the radius evolution. Therefore,

$$\mathbf{Q}(\Delta t) = \text{blkdiag}\left(\mathbf{Q}_a(\Delta t, \sigma_a), \mathbf{Q}_a(\Delta t, \sigma_a), \mathbf{Q}_a(\Delta t, \sigma_{r,a})\right), \tag{3.37}$$

where σ_a is the horizontal acceleration process-noise parameter and $\sigma_{r,a}$ is the process-noise parameter used for the radius and radius-rate states.

A measurement is generated when a storm-sector scan is completed and at least one storm-occupied resolution cell has been detected. The measurement vector contains the measured storm-centroid coordinates and an estimate of the storm radius:

$$\mathbf{z}_{i,k} = \begin{bmatrix} z_{x,i,k} & z_{y,i,k} & z_{r,i,k} \end{bmatrix}^T. \tag{3.38}$$

The centroid is calculated from the Cartesian coordinates of the detected resolution-cell centres. The radius measurement is obtained from the estimated area of the detected occupied cells. Consequently, when only part of a storm is visible or scanned, the measured centroid and radius can differ from the true storm centre and radius.

The measurement model is

$$\mathbf{z}_{i,k} = \mathbf{H}\mathbf{x}_{i,k} + \mathbf{v}_{i,k}, \tag{3.39}$$

where $\mathbf{v}_{i,k}$ represents the measurement uncertainty. Since only the storm-centre position and radius are measured, the measurement matrix is

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{3.40}$$

The measurement-noise covariance is

$$\mathbf{R} = \begin{bmatrix} \sigma_p^2 & 0 & 0 \\ 0 & \sigma_p^2 & 0 \\ 0 & 0 & \sigma_r^2 \end{bmatrix}, \tag{3.41}$$

where σ_p is the assumed standard deviation of the centroid measurement and σ_r is the assumed standard deviation of the radius measurement.

When a new measurement is available, the innovation is calculated as

$$\mathbf{y}_{i,k} = \mathbf{z}_{i,k} - \mathbf{H}\hat{\mathbf{x}}_{i,k|k-1}. \tag{3.42}$$

The innovation covariance is

$$\mathbf{S}_{i,k} = \mathbf{H}\mathbf{P}_{i,k|k-1}\mathbf{H}^T + \mathbf{R}. \tag{3.43}$$

The Kalman gain is then given by

$$\mathbf{K}_{i,k} = \mathbf{P}_{i,k|k-1}\mathbf{H}^T\mathbf{S}_{i,k}^{-1}. \tag{3.44}$$

The state estimate is updated according to

$$\hat{\mathbf{x}}_{i,k|k} = \hat{\mathbf{x}}_{i,k|k-1} + \mathbf{K}_{i,k}\mathbf{y}_{i,k}, \tag{3.45}$$

and the covariance update is

$$\mathbf{P}_{i,k|k} = (\mathbf{I} - \mathbf{K}_{i,k}\mathbf{H})\,\mathbf{P}_{i,k|k-1}. \tag{3.46}$$

The prediction and measurement-update procedure is summarised in Figure 3.4.

The assumed centroid and radius measurement-noise standard deviations are

$$\sigma_p = \sigma_r = 0.30 \text{ km}. \tag{3.47}$$

The horizontal acceleration process-noise parameter is

$$\sigma_a = 0.003 \text{ km/s}^2, \tag{3.48}$$

and the radius process-noise parameter is

$$\sigma_{r,a} = 0.001 \text{ km/s}^2. \tag{3.49}$$

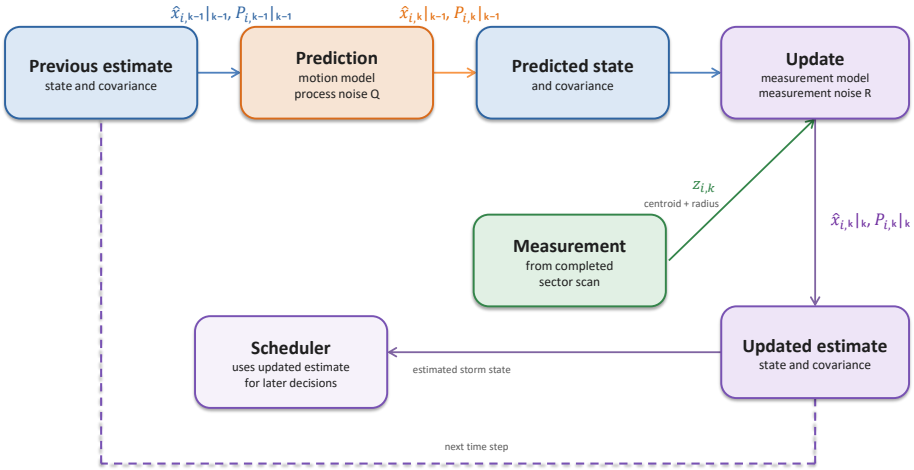


Figure 3.4: Kalman-filter prediction and measurement-update loop for storm-state estimation.

These values are selected empirically for the simplified storm-motion and measurement models. No additional random perturbation is added to the centroid measurements in the final simulations. Measurement errors instead arise from the finite radar-resolution grid and from partial observation of the storm footprint.

If a completed scan does not contain any occupied cells belonging to the selected storm, the Kalman filter is not updated and a radar-specific missed detection is recorded. After three consecutive missed updates, that radar–storm pair is marked unavailable. A storm track is considered globally lost only when the storm has become unavailable to every radar in the network.

The tracker provides the schedulers with estimated position, velocity, radius, and covariance information. These quantities are used for storm-sector calculation, task-duration estimation, priority computation, and tracking-performance evaluation.

3.7 Task Definition and Scheduler Inputs

Two task types are considered in this work: storm-tracking tasks and background-surveillance tasks. A storm-tracking task corresponds to the observation of a specific storm cell by a selected radar. It is defined by the storm index, the radar index, and the set of azimuth bins occupied by the storm as viewed from that radar. Since storms may subtend more than one azimuth bin, a tracking task is considered complete only after all azimuth bins belonging to the storm sector have been scanned. When the task is completed, the revisit event is recorded and the corresponding storm-state estimate is updated by the tracking filter.

Table 3.1: Main inputs used by the scheduler at each decision step.

Input source	**Scheduler input**	**Role in scheduling**
Simulation clock	Current simulation time, t	Determines task timing, revisit timing, and other time-dependent scheduling quantities.
Radar-network state	Radar positions $\mathbf{p}_j^r$ and current pointing directions θ_j	Used to determine radar–storm geometry, azimuth sectors, radar distance, and traverse time between tasks.
Tracking module	Estimated storm position $\hat{\mathbf{p}}_i^s$, velocity $\hat{\mathbf{v}}_i^s$, and radius $\hat{r}_i$	Used to determine storm-sector geometry, task duration, storm priority, and radar–storm distance.
Time-Balance module	Current time-balance value TB_i	Represents the revisit urgency of storm i and is used for task eligibility and task-selection scoring.
Revisit-interval adaptation	Current revisit interval U_i and bounds $[U_{\min}, U_{\max}]$	Define the requested storm-update interval and are adapted according to the estimated tracking load.
Storm-priority model	Storm priority P_i	Represents the relative importance of storm i based on AoI proximity, estimated size, and estimated speed.
Active-task state	Currently active and already assigned storm tasks	Prevents duplicate storm assignment and ensures that ongoing traverse and tracking tasks are completed before new tasks are selected.

Background surveillance is performed when a radar is not assigned to an eligible storm-tracking task. In this case, the simulator first determines which azimuth bins are occupied by currently known storm sectors. The remaining azimuth bins are treated as free surveillance bins. Each radar then scans one free bin per dwell interval and advances cyclically through the available surveillance bins. This ensures that radar time not used for storm tracking is still used to maintain background coverage of the domain.

At each scheduling decision, the scheduler receives information from the storm environment, the radar network, and the tracking module. These inputs are summarized in Table 3.1.

The time-balance value represents the urgency with which a storm should be revisited. It increases while the storm remains unobserved and is reduced after the corresponding tracking task is completed. The storm priority is computed from storm-related properties, including distance to the area of interest, storm size, and storm speed. These quantities allow the

scheduler to distinguish between storms that require more urgent attention and storms that can be revisited later.

The system model described in this chapter therefore provides the basis for the scheduling comparison in the following chapter. Although the simulator is deliberately simplified, it retains the main constraints that influence adaptive radar scheduling: limited dwell time, finite azimuth scan sectors, nonzero traverse time, uncertainty in storm-state estimates, and the need to allocate radar time between tracking and surveillance.

Chapter 4

Scheduling Methodology

The goal of this chapter is to describe the scheduling methodology used to allocate radar time between storm tracking and background surveillance within a network of fixed radars. All scheduling methods are evaluated using the same radar-network configuration, but they differ in how the available radar nodes are coordinated. The baseline methods use predefined scanning roles or independent scanning behaviour, whereas the proposed adaptive methods use a central scheduler with access to the radar states, storm-state estimates, and scheduling variables of the complete network.

The chapter first formulates radar task scheduling as a discrete-time decision process and defines the main timing and scheduling quantities. Four scheduling methods are then presented. Baseline 1 uses a fixed division of responsibility between the radar nodes, while Baseline 2 allows each radar to perform continuous full-azimuth scanning without coordinated storm assignment. The Radar-Level Task Selection (RLTS) scheduler introduces centralised radar-level task selection across the network, and the Load-Aware Task Scheduling (LATS) scheduler extends this approach with periodic load-aware storm-to-radar assignment. The chapter concludes by summarising the differences between the four methods.

4.1 Radar Task Scheduling and Time-Balance Formulation

The scheduling problem considered in this thesis is the allocation of limited radar time between storm tracking and background surveillance across a radar network. The radar network consists of N_r fixed radars observing N_s moving storm cells. The radar-network topology and the spatial arrangement of the radar nodes are introduced in Chapter 3. Throughout this chapter, $i \in \{1, \ldots, N_s\}$ denotes a storm index, while $j \in \{1, \ldots, N_r\}$ denotes a radar index. All four scheduling methods operate on this same network, but they differ in the level of coordination between the radar nodes. At each dwell interval, every radar must either continue an active task, start a new storm-tracking task, or perform surveillance. Since one radar can scan only one azimuth bin during one dwell interval, assigning time to one storm delays the revisit of other storms. The scheduler must therefore balance storm-update frequency, radar availability, task feasibility, and surveillance coverage.

The simulation is formulated in discrete time. The smallest scheduling unit is therefore

the dwell interval, denoted by T_{dwell}. During one dwell interval, a radar scans one azimuth bin. A storm-tracking task usually requires multiple dwell intervals because a storm occupies a finite azimuth sector and therefore spans multiple azimuth bins. The storm-centre azimuth is used as the reference direction for estimating the radar traverse time, while the complete storm sector is subsequently scanned from its first selected azimuth bin to its last selected azimuth bin, as described previously in Chapter 3.

A storm revisit is counted only after the complete storm sector has been scanned. This definition is used because a partial sector scan does not represent a complete storm update in the simplified simulation model. The revisit log therefore records completed tracking tasks using the task start time, task end time, storm index, and radar index.

Following the Time-Balance formulation introduced for adaptive weather-radar scheduling by Reinoso-Rondinel et al. [12], each storm is assigned a time-balance value, denoted by TB_i, to represent its revisit urgency. The time-balance value increases while the storm waits to be revisited. If the tracking task for storm i is not completed during a dwell interval, its time-balance value is updated as

$$TB_i(t + T_{\text{dwell}}) = TB_i(t) + T_{\text{dwell}}. \tag{4.1}$$

When a storm-tracking task is completed, the time-balance value of the corresponding storm is reduced by its current revisit interval U_i:

$$TB_i \leftarrow TB_i - U_i. \tag{4.2}$$

A storm is considered eligible for tracking when its time-balance value is non-negative. A positive value therefore indicates that the storm is due, or overdue, for revisiting, while a negative value indicates that the storm has been updated recently enough relative to its revisit interval. This behaviour is illustrated conceptually in Figure 4.1.

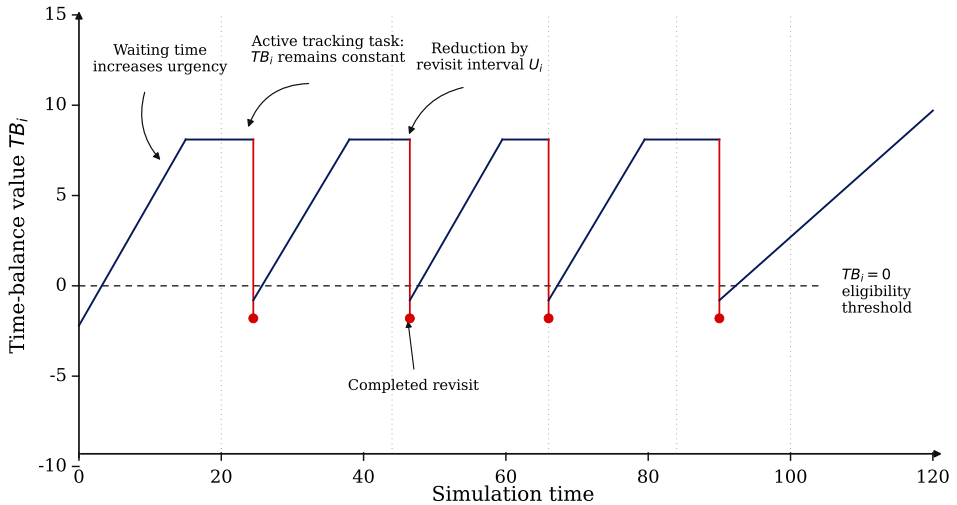


Figure 4.1: Example of a Time-balance evolution before and after a completed storm revisit.

Let $\mathcal{T}_j(t)$ denote the set of feasible storm-tracking tasks available to radar j at time t. The scheduling action for radar j is then selected from

$$a_j(t) \in \{\text{continue current task}, \tau_i \in \mathcal{T}_j(t), \text{surveillance}\}, \tag{4.3}$$

where τ_i denotes the tracking task associated with storm i.

A new storm-tracking task can be selected only when the radar is available and the storm is feasible for that radar. In the implemented schedulers, feasibility depends on storm visibility, current task activity, time-balance eligibility, and duplicate-assignment constraints. A radar is considered available only when it is not already traversing towards or scanning another storm sector. Storm visibility requires that the corresponding radar–storm pair remains available according to the tracking module. In addition, a storm that is already being scanned by another radar, or has already been assigned during the same scheduling decision, is excluded from the candidate set in order to prevent duplicate observations.

The overall tracking demand across the radar network is represented using a load factor. If T_i denotes the estimated average task duration required to update storm i, and U_i denotes its desired revisit interval, the load factor is

$$\rho = \sum_{i=1}^{N_s} \frac{T_i}{U_i}. \tag{4.4}$$

Here, T_i is the average total task duration for storm i across the feasible radar nodes. The load factor ρ therefore represents the overall storm-tracking demand considered by the radar network rather than the workload of an individual radar.

In the implemented adaptive schedulers, this quantity is used as an approximate indicator of tracking load. A value of $\rho > 1$ triggers an increase in the requested revisit intervals, while a value of $\rho < 1$ allows the revisit intervals to be reduced according to the adaptation logic.

This formulation provides the basis for the scheduling methods described in the following sections. The baseline methods use simpler scanning rules and do not explicitly exploit the full time-balance and load-aware formulation. The adaptive schedulers use time-balance values, task-duration estimates, storm priority, radar geometry, and workload information to select tracking tasks and allocate radar time.

4.2 Baseline 1: Storm-by-Storm Scanning

The first baseline represents a fixed division of responsibility within the radar network. One radar is assigned exclusively to storm tracking, while the remaining radar nodes are assigned to background surveillance. In the two-radar configuration used in this thesis, this corresponds to one tracking radar and one surveillance radar. The tracking radar visits the storms in a fixed cyclic order as:

$$1 \to 2 \to \cdots \to N_s \to 1. \tag{4.5}$$

For each storm, the tracking radar determines the azimuth sector occupied by the storm based on its estimated position and radius. The centre azimuth of the sector is used as the reference direction for estimating the radar traverse time. The radar then scans all azimuth bins belonging to the storm sector, from the first selected bin to the last selected bin. A revisit is recorded only after the complete storm sector has been scanned. Once the task is completed, the radar proceeds to the next storm in the cyclic sequence.

This baseline does not use time-balance values, storm priorities, adaptive revisit intervals, or multi-radar task allocation. It is therefore included as a reference case for sequential storm tracking. Since only one radar contributes to storm revisits, the method represents a conservative use of the radar network for tracking, while the other radar contributes only to surveillance by scanning free azimuth bins cyclically.

The main limitation of this baseline is that it does not adapt to storm urgency, storm size, storm motion, or radar geometry. All storms are treated according to the same fixed order, regardless of whether some storms require more frequent updates than others. In addition, the method cannot exploit multiple radars for coordinated storm tracking, even when several storms are simultaneously due for revisiting.

4.3 Baseline 2: 360-Degree Sit-and-Spin Scanning

The second baseline represents independent continuous 360° azimuth scanning by every radar in the network. No central storm-to-radar assignment or inter-radar coordination is applied. Instead, each radar repeatedly scans the full azimuth domain using the same predefined scanning rule. At each dwell interval, radar j advances by one azimuth bin:

$$\theta_j(t + T_{\text{dwell}}) = (\theta_j(t) + \Delta\theta) \bmod 360^\circ, \tag{4.6}$$

where $\theta_j(t)$ is the current pointing direction of radar j and $\Delta\theta$ is the azimuth-bin spacing.

In this baseline, storms are revisited when the continuous azimuth scan covers the complete sector occupied by the storm. A revisit is counted only after all azimuth bins belonging to the storm sector have been scanned. This definition is consistent with the tracking-task definition used in the other methods and avoids overestimating revisit performance by counting only the first contact with a storm sector.

The sit-and-spin baseline represents non-adaptive angular coverage. Its main advantage is that it provides regular surveillance of the full azimuth domain. However, it does not prioritise storms according to revisit urgency, distance to the area of interest, storm size, or storm speed. Consequently, the revisit timing is determined primarily by the full-scan period and by the angular extent of each storm sector.

This baseline is included as a reference case for uniform angular scanning. It represents a scanning strategy in which radar motion is independent of storm-state estimates and scheduling variables. Conceptually, this is similar to conventional operational weather-radar scanning, such as that used by the Royal Netherlands Meteorological Institute (KNMI), where the radar repeatedly performs predefined full-volume scans to provide systematic cov-

erage of the observation domain. The baseline used in this thesis is, however, a simplified two-dimensional representation of such operation, since only continuous full-azimuth scanning is modelled.

4.4 Radar-Level Task Selection (RLTS) scheduler

Scheduler Overview

The first adaptive scheduler is a centralised radar-level task-selection scheduler. A single scheduler has access to the current radar states, storm-state estimates, time-balance values, and task geometry. At each dwell interval, the scheduler determines the next action for each radar. If a radar is already traversing towards a storm sector or scanning an active storm sector, it continues the current task until completion. A new task is selected only when the corresponding radar becomes available.

The decision logic of the Radar-Level Task Selection (RLTS) scheduler is shown in Figure 4.2. When a radar is available, the scheduler first forms a set of feasible storm-tracking candidates. If no feasible candidate exists, the radar performs surveillance. If one or more feasible candidates are available, the scheduler assigns a score to each candidate and selects the storm with the highest score.

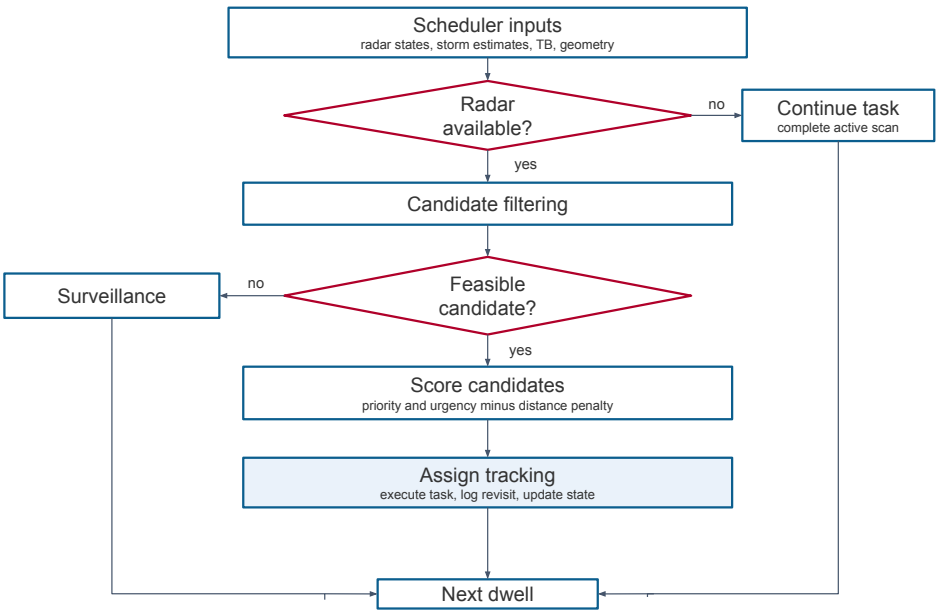


Figure 4.2: Workflow of the RLTS scheduler.

Storm Priority Computation

A predefined area of interest (AoI) represents a location where enhanced storm monitoring is required. In the simulation, the AoI is defined by the fixed position $\mathbf{p}_{\text{AoI}}$, it is fixed at $\mathbf{p}_{\text{AoI}} = (0, 0)$ km. The distance of storm i from the AoI is calculated from the estimated

storm-centre position $\hat{\mathbf{p}}_i$ as

$$d_i^{\mathrm{AoI}} = \|\hat{\mathbf{p}}_i - \mathbf{p}_{\mathrm{AoI}}\|_2 \,. \tag{4.7}$$

For each storm, a priority value is computed using three components: distance to the area of interest, storm size, and storm speed. The total storm priority is defined as

$$P_i = P_i^{\mathrm{AoI}} + P_i^{\mathrm{size}} + P_i^{\mathrm{speed}}. \tag{4.8}$$

The AoI-proximity component assigns higher priority to storms closer to the area of interest, the size-priority component gives higher weight to larger storms, while the speed-priority component gives higher weight to faster-moving storms.The scoring rules used to compute these components are summarized in Table 4.1.

The storm position, radius, and velocity used in the priority calculation are obtained from the current tracker estimates rather than from the true simulated storm state.

Table 4.1: Priority scoring rules used for storm-priority computation.

Priority component	**Condition**	**Score**
Proximity to AoI, P_i^{AoI}	$d_i^{\mathrm{AoI}} \leq 5$ km	3
	5 km $< d_i^{\mathrm{AoI}} \leq 10$ km	2
	$d_i^{\mathrm{AoI}} > 10$ km	1
Storm size, P_i^{size}	$r_i > 3$ km	2
	$r_i \leq 3$ km	1
Storm speed, P_i^{speed}	$v_i \geq 0.02$ km/s	3
	0.01 km/s $\leq v_i < 0.02$ km/s	2
	$v_i < 0.01$ km/s	1

Here, d_i^{AoI} is the distance between storm i and the area of interest, r_i is the estimated storm radius, and v_i is the magnitude of the estimated storm velocity. Notably, this priority formulation is heuristic. It is used to represent the assumption that storms closer to the area of interest, storms with larger spatial extent, and storms with faster motion should receive more attention from the scheduler. An analysis of the effect of changes of scores and priorities is left for future work.

Candidate Filtering and Feasibility Checks

Before a storm is scored, it must pass a set of feasibility checks. A storm is considered as a feasible candidate for radar j only if:

- its time-balance value is non-negative;
- it is visible to radar j;
- it is not already assigned to another radar during the same decision step;

- it is not currently being scanned by another radar;

The time-balance condition ensures that only storms that are due or overdue for revisiting are considered. The active-task and duplicate-assignment checks prevent two radars from simultaneously selecting or servicing the same storm.

Task-Selection Score

For each feasible storm–radar pair (i, j), the scheduler computes a task-selection score:

$$S_{ij} = P_i TB_i - 0.1 d_{ij}, \tag{4.9}$$

where P_i is the storm priority defined in Eq. 4.8, TB_i is the current time-balance value, and d_{ij} is the Euclidean distance between radar j and the estimated centre position of storm i.

The first term favours storms that are both important and urgent. A storm with a high priority but a negative or low time-balance value is not selected prematurely, while a storm with a high time-balance value becomes more likely to be selected as it becomes overdue. The second term introduces a mild distance penalty, so that less favourable radar–storm geometries are slightly discouraged. The composition of the score is visually illustrated in Figure 4.3.

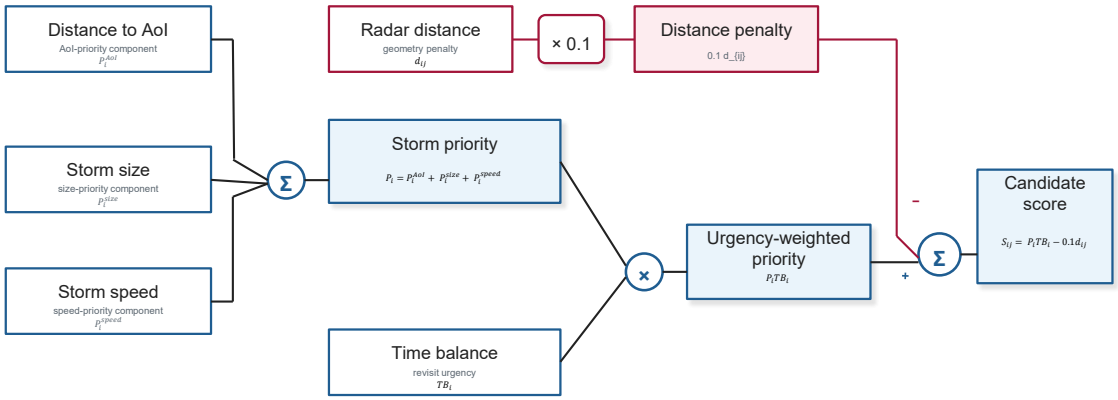


Figure 4.3: Task-selection score computation used by the RLTS scheduler.

Revisit-Interval Adaptation

RLTS Scheduler also adapts the revisit intervals using the load factor defined in Eq. 4.4. For storm i, the task duration is estimated for each radar and averaged across the radar network to obtain an average task duration T_i. The load factor ρ then provides an estimate of the total tracking demand relative to the requested revisit intervals.

When the estimated load is above the overload threshold, the requested revisit intervals are relaxed:

$$U_i \leftarrow \rho U_i. \tag{4.10}$$

This increases the revisit interval and reduces the requested tracking load. When the load is sufficiently below the overload threshold, the revisit intervals are reduced:

$$U_i \leftarrow \frac{U_i}{1+h}, \tag{4.11}$$

where $h = 0.05$ is the hysteresis parameter. The hysteresis term prevents frequent oscillations of the revisit interval when the load factor is close to unity. After each update, the revisit intervals are clipped to remain within predefined lower and upper bounds.

Task Assignment and Surveillance Fallback

After scoring, the highest-scoring feasible storm is assigned to the available radar. A tracking task consists of three stages. First, the radar traverses from its current pointing direction to the centre azimuth of the selected storm sector, if a traverse is required. Second, the radar scans each azimuth bin in the storm sector. Third, after the final bin is scanned, the task is marked as complete. At completion, the revisit event is logged, the storm-state estimate is updated, the time-balance value is reduced, and the radar becomes available for a new task.

For each radar, the scheduler estimates which azimuth bins are occupied by the current storm sectors. The free surveillance bins are defined as:

$$\Theta_j^{\text{free}} = \Theta^{\text{all}} \setminus \bigcup_{i=1}^{N_s} \Theta_{ij}, \tag{4.12}$$

where Θ^{all} is the full azimuth grid, Θ_{ij} is the sector occupied by storm i as seen from radar j, and $\setminus$ denotes set difference.

The radar then scans one free azimuth bin and advances cyclically through the free-bin set. Surveillance scans do not directly update storm time-balance values, but they are included in the Gantt logs and in the time-allocation metrics shown later in the thesis.

The RLTS scheduler therefore combines time-balance urgency, storm priority, radar-distance scoring, adaptive revisit-interval updates, and surveillance fallback. These mechanisms allow the scheduler to move beyond fixed scan ordering and select tasks based on the current storm and radar state.

4.5 Load-Aware Task Scheduling (LATS) scheduler

Scheduler Overview

The Load-Aware Task Scheduling (LATS) scheduler extends the Radar-Level Task Selection (RLTS) scheduler by adding a load-aware storm-to-radar assignment stage. The purpose

of this stage is to reduce inefficient competition between radars and distribute storm-tracking responsibility more evenly across the radar network. In the RLTS scheduler, each available radar evaluates feasible storm candidates directly. Simultaneous duplicate assignments are prevented by excluding storms that have already been selected during the current decision step or are currently being scanned by another radar. However, these checks do not distribute long-term storm-tracking responsibility across the radar network. Consequently, some storms may be repeatedly favoured over successive decisions because of their urgency or radar geometry, while other storms receive overall less balanced attention.

The LATS scheduler addresses this by periodically partitioning the storm set across the available radars. After the assignment step, each radar applies radar-level task selection only within its assigned storm subset. The task-selection score, time-balance update, priority computation, adaptive revisit-interval update, and surveillance fallback are otherwise the same as in RLTS scheduler. Therefore, LATS scheduler modifies the candidate set available to each radar, rather than changing the definition of a tracking task.

The overall workflow of LATS scheduler is shown in Figure 4.4.

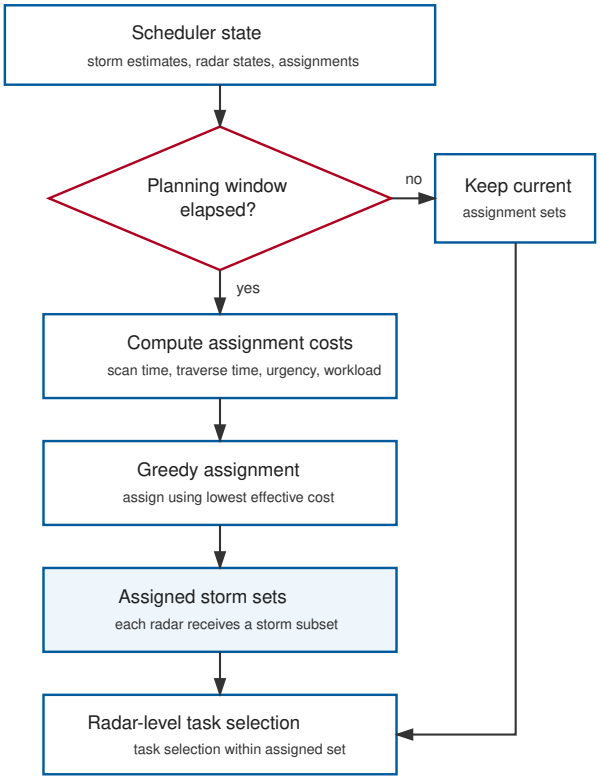


Figure 4.4: Workflow of the LATS scheduler.

Planning Window

The storm-to-radar assignment is recomputed after a fixed planning window of N_{plan} dwell intervals. The planning window defines how long an assignment remains active before it is updated. During this interval, each radar continues to select tasks only from its as-

signed storm subset, unless no feasible assigned storm is available and the radar falls back to surveillance.

The choice of N_{plan} determines the responsiveness of the assignment stage. A short planning window allows the scheduler to respond quickly to changes in storm position, radar pointing direction, and time-balance values. However, very frequent reassignment may reduce stability because storm-to-radar responsibility can change too often. A longer planning window provides more stable assignments, but may respond more slowly to changes in geometry and storm urgency in case of a dynamic scenario.

Assignment Cost

At each assignment update, the scheduler computes the cost of assigning storm i to radar j. The base assignment cost is defined as

$$C_{ij} = w_{\text{s}} T_{ij}^{\text{sector}} + w_{\text{t}} T_{ij}^{\text{trav}} - w_{\text{u}} TB_i. \tag{4.13}$$

Here, T_{ij}^{sector} is the estimated time required for radar j to scan the azimuth sector occupied by storm i, and T_{ij}^{trav} is the estimated traverse time from the current radar pointing direction to the storm sector. The first two terms therefore penalise radar–storm assignments that require long scan or traverse times. The third term decreases the assignment cost for storms with larger time-balance values, so overdue storms become more favourable assignment candidates.

The weights w_{s}, w_{t}, and w_{u} control the relative influence of sector scan time, traverse time, and time-balance urgency, respectively. This cost therefore combines task-duration information and revisit urgency into a single radar–storm pairing measure.

Workload Penalty

The base cost alone does not explicitly prevent several storms from being assigned to the same radar. To encourage a more balanced distribution, an additional workload penalty is included. The effective assignment cost is:

$$C_{ij}^{\text{eff}} = C_{ij} + w_{\text{l}} L_j, \tag{4.14}$$

where L_j is the number of storms already assigned to radar j, and w_{l} controls the strength of the workload penalty.

The effective assignment cost is therefore composed of two parts: a base radar–storm pairing cost and a workload penalty. The base cost increases with storm-sector scan time and radar traverse time, and decreases with increasing time-balance urgency. The workload penalty increases the effective cost when a radar already has several assigned storms. This discourages unbalanced storm allocation across the radar network. The cost structure is summarized in Figure 4.5.

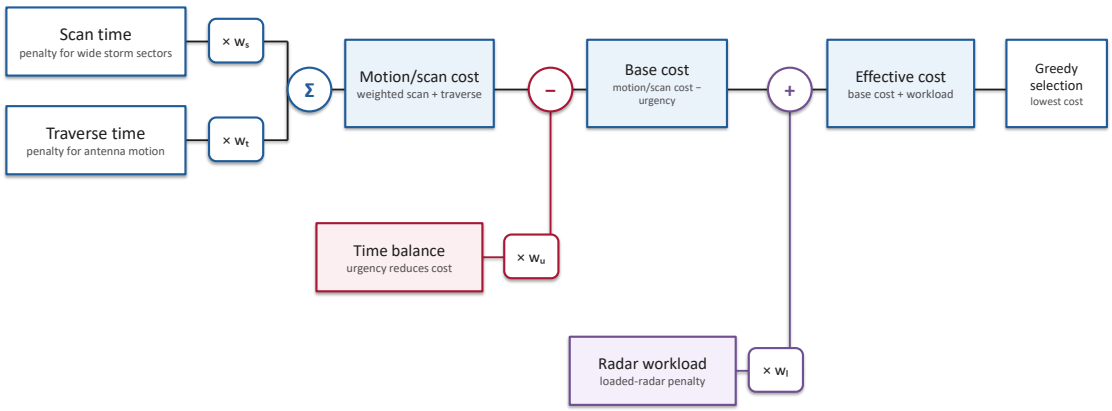


Figure 4.5: Effective assignment cost computation used by the LATS scheduler.

Greedy Storm-to-Radar Assignment

The storm-to-radar assignment is performed greedily. When possible, the scheduler first assigns at least one storm to each radar. This avoids leaving a radar without storm-tracking responsibility when the number of storms is sufficient. After this initial assignment, the remaining storms are assigned one at a time. For each unassigned storm, the scheduler evaluates the effective assignment cost C_{ij}^{eff} for all radars. The storm is then assigned to the radar with the lowest effective cost. After each assignment, the workload count L_j of the selected radar is updated. This update affects subsequent assignment decisions, because assigning additional storms to the same radar becomes less favourable as its workload increases. This greedy procedure does not solve a global optimisation problem over all possible storm-to-radar assignments. Instead, it provides a computationally simple load-aware assignment rule that can be applied repeatedly during the simulation.

Task Selection After Assignment

After the assignment step, each radar has an assigned subset of storms. During dwell-by-dwell scheduling, the radar applies the same candidate filtering and task-selection score used in RLTS, but only to storms in its assigned subset. If no feasible assigned storm is available, the radar performs surveillance. This means that LATS does not replace the radar-level task-selection logic. Instead, it adds a higher-level partitioning stage that restricts the candidate set considered by each radar. The final selected task still depends on time-balance urgency, storm priority, radar distance, task feasibility, and the current radar state.

Effect of Planning-Window Length

The planning-window length affects how frequently the load-aware assignment is updated.

A shorter planning window can react more quickly when storms move, grow, or become overdue. However, it may also cause frequent changes in storm-to-radar responsibility. A longer planning window produces more stable assignments, but may keep a radar assigned to storms that are no longer favourable from a geometric or urgency perspective. For this reason, the planning-window length is evaluated separately in the experimental analysis. This allows the influence of assignment-update frequency to be studied without changing the underlying radar-level task-selection logic.

To summarize, the Load-Aware Task Scheduling (LATS) scheduler extends the Radar-Level Task Selection (RLTS) scheduler by introducing periodic load-aware storm-to-radar assignment. The method uses scan time, traverse time, time-balance urgency, and radar workload to distribute storms across the radar network before radar-level task selection is applied.

4.6 Summary of Compared Methods

The decision logic of the two proposed schedulers is summarised in Algorithms 1 and 2. These algorithmic representations complement the preceding flowcharts by presenting the main scheduling steps in a compact form.

Algorithm 1: Radar-Level Task Selection (RLTS) scheduler

Require: Radar set $\mathcal{R}$, storm set $\mathcal{S}$, time-balance values TB_i, revisit intervals U_i

Predict the current storm states and update the storm priorities P_i
$\mathcal{A} \leftarrow$ storms currently being scanned or traversed to
for each radar $j \in \mathcal{R}$ **do**
    if radar j has an active tracking task **then**
        Continue the traverse or sector scan for one dwell interval
        if the complete storm sector has been scanned **then**
            Process the detection and update the storm track
            if storm i is detected **then**
                $TB_i \leftarrow TB_i - U_i$
            end if
            Update the revisit intervals and complete the task
        end if
    else
        Form the feasible candidate set $\mathcal{F}_j$ ▷ Available, eligible, and not active elsewhere
        if $\mathcal{F}_j \neq \emptyset$ **then**
            Compute S_{ij} for every $i \in \mathcal{F}_j$
            $i^\star \leftarrow \arg\max_{i \in \mathcal{F}_j} S_{ij}$
            Initialise the tracking task for storm $i^\star$
            Execute its first traverse or scan dwell
            $\mathcal{A} \leftarrow \mathcal{A} \cup \{i^\star\}$
        else
            Scan the next free surveillance azimuth bin
        end if

end if
end for
for each non-lost storm i not active during the current dwell **do**
    $TB_i \leftarrow TB_i + T_{\text{dwell}}$
end for

Algorithm 2: Load-Aware Task Scheduling (LATS) scheduler

Require: Radar set $\mathcal{R}$, storm set $\mathcal{S}$, planning-window length N_{plan}
Predict the current storm states and update the storm priorities P_i
if the planning window has elapsed **then**
    Initialise the assigned sets $\mathcal{S}_j \leftarrow \emptyset$ and workloads $L_j \leftarrow 0$
    for each feasible storm–radar pair (i, j) **do**
        Compute

$$C_{ij} = w_{\text{s}} T_{ij}^{\text{sector}} + w_{\text{t}} T_{ij}^{\text{trav}} - w_{\text{u}} TB_i$$

    end for
    Assign one feasible storm to each radar when possible
    while feasible unassigned storms remain **do**
        Compute $C_{ij}^{\text{eff}} = C_{ij} + w_{\text{l}} L_j$
        $(i^\star, j^\star) \leftarrow \arg\min_{(i,j)} C_{ij}^{\text{eff}}$
        $\mathcal{S}_{j^\star} \leftarrow \mathcal{S}_{j^\star} \cup \{i^\star\}$
        $L_{j^\star} \leftarrow L_{j^\star} + 1$
    end while
    Reset the planning-window counter
end if
for each radar $j \in \mathcal{R}$ **do**
    if radar j has an active tracking task **then**
        Continue the active traverse or sector scan
    else
        Form the feasible set $\mathcal{F}_j \subseteq \mathcal{S}_j$
        if $\mathcal{F}_j \neq \emptyset$ **then**
            Compute S_{ij} for every $i \in \mathcal{F}_j$
            Select $i^\star = \arg\max_{i \in \mathcal{F}_j} S_{ij}$
            Initialise and execute the tracking task
        else
            Scan the next free surveillance azimuth bin
        end if
    end if
end for
Update the time-balance values of storms not active during the dwell
Advance the planning-window counter

Table 4.2 provides a final comparison of the four scheduling methods considered in this thesis. The two baseline methods provide non-adaptive reference behaviours. The Radar-Level Task Selection (RLTS) scheduler evaluates the effect of centralised radar-level task

selection across the radar network. The Load-Aware Task Scheduling (LATS) scheduler evaluates whether introducing explicit storm-to-radar assignment and workload balancing improves the use of the radar network. The following chapters compare these methods across different storm-load scenarios using revisit rate, improvement factor, tracking accuracy, surveillance coverage, and radar time-allocation metrics.

Table 4.2: Summary of the scheduling methods compared in this thesis.

Method	Main idea	Key characteristics
Baseline 1	Storm-by-storm tracking	One radar tracks storms in fixed cyclic order; the other performs surveillance; no time balance, priority, or adaptive task selection.
Baseline 2	360-degree sit-and-spin scanning	Each radar continuously scans the full azimuth domain; storm revisits occur when the full storm sector is covered by the scan.
RLTS Scheduler	Radar-level task selection	Uses time balance, storm priority, radar distance, task feasibility, and load-dependent revisit-interval adaptation.
LATS Scheduler	Load-aware assignment-based scheduling	Extends RLTS Scheduler by periodically assigning storms to radars using scan time, traverse time, urgency, and workload.

Chapter 5

Experimental Setup and Evaluation Metrics

This chapter describes the experimental setup used to evaluate the scheduling methods introduced in Chapter 4. The purpose of the evaluation is to compare the behaviour of the baseline and adaptive schedulers under controlled storm conditions. Hence, the main aim of this chapter is to define the simulation configuration, the storm scenarios, the Monte Carlo trial design, and the metrics used to assess scheduler performance.

The evaluation is based on Monte Carlo simulations over multiple storm scenarios. Each scenario represents a different radar-load condition by varying the number of storms, storm size, or storm speed. This allows the schedulers to be tested under low-load, nominal, high-load, fast-storm, and large-storm conditions. These scenarios are designed to examine how the scheduling methods behave as the demand for radar time changes.

For a fair comparison, the same generated storm realisation is used for all scheduling methods within each Monte Carlo trial. This ensures that differences in performance are caused by the scheduling logic rather than by differences in the simulated weather scene. The chapter also defines the metrics used to evaluate the schedulers, including revisit rate, improvement factor, tracking accuracy, surveillance coverage, and radar time allocation.

5.1 Simulation Configuration

All experiments are performed using the two-dimensional simulation framework described in Chapter 3. The configuration parameters are grouped into three categories: radar-network geometry and spatial discretisation, scan timing and simulation duration, and tracking and scheduling parameters.

Radar-network geometry and spatial discretisation. The radar network consists of two fixed radars observing a set of moving storm cells. The maximum radar range is

$$R_{\max} = 30 \text{ km}. \tag{5.1}$$

The two radars are positioned on a circle of radius $R_{\max}/2$, resulting in the locations

$$\mathbf{p}_{r,1} = (15, 0) \text{ km}, \qquad \mathbf{p}_{r,2} = (-15, 0) \text{ km}. \tag{5.2}$$

The azimuth grid resolution is

$$\Delta\theta = 0.5^\circ, \tag{5.3}$$

which gives

$$N_\theta = \frac{360^\circ}{\Delta\theta} = 720 \tag{5.4}$$

azimuth bins over a complete scan. The polar range grid used for occupancy and tracking evaluation has a range resolution of

$$\Delta r = 0.5 \text{ km}. \tag{5.5}$$

Scan timing and simulation duration. The pulse repetition frequency and the number of pulses per dwell are set to

$$\text{PRF} = 1111 \text{ Hz}, \qquad N_p = 100. \tag{5.6}$$

The time required to scan one azimuth bin is therefore

$$T_{\text{dwell}} = \frac{N_p}{\text{PRF}} = \frac{100}{1111} \approx 0.09 \text{ s}. \tag{5.7}$$

A complete 360° scan requires

$$T_{\text{scan}} = N_\theta T_{\text{dwell}} = 720 \cdot T_{\text{dwell}} \approx 64.8 \text{ s}. \tag{5.8}$$

Tracking and scheduling parameters. The RLTS and LATS schedulers use storm revisit intervals bounded by

$$U_i \in [5, 15] \text{ s}. \tag{5.9}$$

The Kalman-filter measurement-noise standard deviation is

$$\sigma_{\text{meas}} = 0.30 \text{ km}, \tag{5.10}$$

while the acceleration process-noise parameter is

$$\sigma_a = 0.003 \text{ km/s}^2. \tag{5.11}$$

The main simulation parameters are summarised in Table 5.1.

Table 5.1: Main simulation parameters used in the experiments.

Parameter	Value
Number of radars	2
Maximum radar range	30 km
Radar positions	$(15, 0)$ km and $(-15, 0)$ km
Azimuth resolution	0.5°
Number of azimuth bins	720
Range resolution	0.5 km
PRF	1111 Hz
Pulses per dwell	100
Dwell time	≈ 0.09 s
Full-scan time	≈ 64.8 s
Simulation duration per trial	20 min
Revisit-time bounds	$[5, 15]$ s
Measurement-noise standard deviation	0.30 km
Acceleration process-noise parameter	0.003 km/s^2

5.2 Storm Scenarios

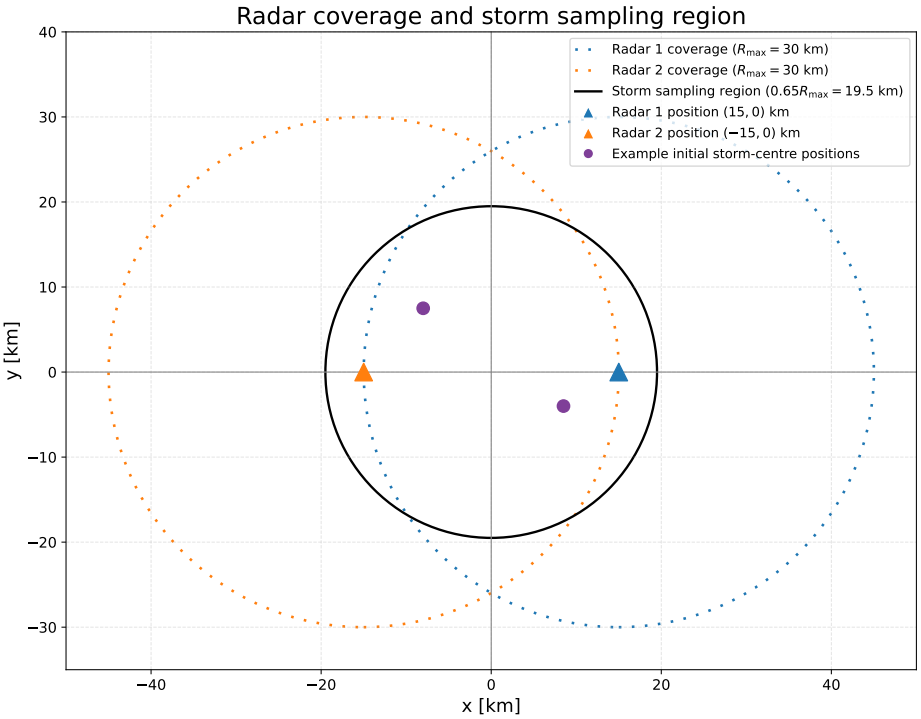


Figure 5.1: Example simulation topology showing the two radar locations and their coverage regions, together with the circular region from which the initial storm-centre positions are sampled. The purple markers indicate illustrative initial storm-centre positions within the sampling region.

Five storm scenarios are used to test the schedulers under different load conditions. The radar load is changed by varying the number of storms, their size, and their speed. In all scenarios, the reflectivity assigned to the theoretical centroid of each storm is sampled between 40 and 50 dBZ, while the storm radial growth rate is sampled between 0.001 and 0.006 km/s.

The initial storm positions are randomly sampled within a circular region of radius $0.65 \cdot R_{\max}$ around the origin, as illustrated in Figure 5.1. The figure also shows the positions and coverage regions of the two radar nodes used in the simulation. A minimum separation constraint is imposed to avoid unrealistic overlap between storm cells at the start of the trial. The same storm configuration is then used for Radar-Level Task Selection Scheduler (RLTS) , Load-Aware task Scheduling (LATS), Baseline 1, and Baseline 2.

5.2.1 Low-Load Scenario

The low-load scenario contains two storms. The storm radii are sampled from:

$$r_i \sim \mathcal{U}(2, 6) \text{ km}, \tag{5.12}$$

and the storm speeds are sampled from:

$$\|\mathbf{v}_i\| \sim \mathcal{U}(8, 20) \text{ m/s}. \tag{5.13}$$

The minimum initial gap between storms is set to 2 km.

This scenario represents a relatively light scheduling case. Since only two storms must be tracked by two radars, the adaptive schedulers are expected to have enough capacity to maintain storm revisits while preserving some surveillance. The low-load case is therefore used to evaluate scheduler behaviour when radar resources are not strongly constrained.

5.2.2 Nominal Scenario

The nominal scenario contains four storms. The storm radii and speeds are sampled from the same ranges as in the low-load scenario:

$$r_i \sim \mathcal{U}(2, 6) \text{ km}, \tag{5.14}$$

$$\|\mathbf{v}_i\| \sim \mathcal{U}(8, 20) \text{ m/s}. \tag{5.15}$$

The minimum initial storm gap is again set to 2 km.

This scenario represents the main reference operating condition. Compared with the low-load scenario, the number of storm-tracking tasks is doubled, increasing the demand on the radar network. The nominal scenario is used to assess whether the adaptive schedulers provide a consistent improvement over the baseline methods under moderate load.

5.2.3 High-Load Scenario

The high-load scenario contains six storms. The storm radius and speed ranges are:

$$r_i \sim \mathcal{U}(2, 6) \text{ km}, \tag{5.16}$$

$$\|\mathbf{v}_i\| \sim \mathcal{U}(8, 20) \text{ m/s}. \tag{5.17}$$

The minimum initial storm gap is reduced to 0.5 km to make it possible to place more storms within the same simulation domain.

This scenario imposes the highest task-count load on the radar network. The purpose is to test whether the adaptive schedulers can maintain revisit performance when many storms compete for radar time. It is also useful for observing the trade-off between storm tracking and surveillance, because high tracking demand may leave less time for background scanning.

5.2.4 Fast-Storm Scenario

The fast-storm scenario contains four storms, as in the nominal case, but the storm speed range is increased:

$$\|\mathbf{v}_i\| \sim \mathcal{U}(20, 35) \text{ m/s}. \tag{5.18}$$

The storm radii are sampled from:

$$r_i \sim \mathcal{U}(2, 6) \text{ km}, \tag{5.19}$$

and the minimum initial storm gap is set to 2 km.

This scenario is designed to evaluate the effect of faster storm motion. Faster storms change their position more rapidly between revisits, which can affect both scheduling urgency and tracking accuracy. The fast-storm case is therefore useful for testing whether the adaptive schedulers remain effective when the storm field evolves more quickly.

5.2.5 Large-Storm Scenario

The large-storm scenario contains four storms, but the storm radius range is increased:

$$r_i \sim \mathcal{U}(5, 8) \text{ km}. \tag{5.20}$$

The speed range remains

$$\|\mathbf{v}_i\| \sim \mathcal{U}(8, 20) \text{ m/s}, \tag{5.21}$$

and the minimum initial storm gap is set to 2 km.

Large storms occupy wider azimuth sectors and therefore require more dwell intervals to be completely scanned. This increases the duration of each tracking task, even though the number of storms is the same as in the nominal scenario. The large-storm scenario is therefore used to evaluate scheduler performance when the task duration increases due to storm geometry.

Table 5.2 summarises the five scenarios.

Table 5.2: Storm scenario definitions used in the Monte Carlo experiments.

Scenario	Number of storms	Radius range (km)	Speed range (m/s)	Minimum gap (km)
Low load	2	2–6	8–20	2.0
Nominal	4	2–6	8–20	2.0
High load	6	2–6	8–20	0.5
Fast storms	4	2–6	20–35	2.0
Large storms	4	5–8	8–20	2.0

5.3 Monte Carlo Trial Design

A Monte Carlo simulation design is used to evaluate the schedulers across independently generated storm configurations. An initial short-duration campaign was performed using 1000 trials per scenario, with each trial representing 2 min of simulated operation. The same analysis was also performed using 50 trials per scenario. The resulting scenario-level trends were consistent between the two campaign sizes, indicating that the main scheduler comparisons were already captured by the smaller trial set. The results of this comparison are provided in Appendix A.3.

Based on this consistency, the final evaluation uses 50 trials per scenario while extending the duration of each trial to 20 min. The longer simulation horizon allows the schedulers to be evaluated over repeated tracking, assignment, and surveillance cycles while keeping the overall computational cost tractable.

A different random seed is used for each trial to determine the initial storm positions, radii, velocities, reflectivities, and growth rates. Within each trial, the same generated storm configuration is used for all compared scheduling methods. This ensures that the methods are evaluated under identical underlying storm evolution, so that differences in performance can be attributed to the scheduling logic rather than to differences in the simulated storm conditions.

For each trial, the simulation stores the scheduler logs, revisit logs, tracking-evaluation logs, surveillance-coverage information, and summary metrics. Trial-level metrics are computed for each method, after which the results are aggregated across the 50 final trials for each scenario.

5.4 Evaluation Metrics

The schedulers are evaluated using metrics that describe storm revisit performance, improvement relative to the baselines, tracking accuracy, and surveillance behaviour. These metrics are computed from the revisit logs, tracker logs, surveillance logs, and Gantt logs produced during each simulation trial. The main metrics used in the evaluation are summarised in Table 5.3 and described after the table.

Table 5.3: Summary of evaluation metrics used in the scheduler comparison.

Metric	**Symbol**	**Unit**	**Interpretation**
Revisit time	$T_{i,n}^{\mathrm{rev}}$	s	Time between two successive valid completed updates of the same storm.
Revisit rate	$\lambda_m(t_k)$	s^{-1}	Normalised rate of completed storm revisits within a sliding window.
Effective revisit rate	$\lambda_{\mathrm{eff},m}(t_k)$	s^{-1}	Normalised rate of unique storm updates within a sliding window.
Improvement factor	$I_{a,b}(t_k)$	dimensionless	Ratio between adaptive and baseline revisit-rate performance.
Intersection over union	IoU_i	dimensionless	Overlap between true and estimated storm occupancy.
Surveillance coverage	$C_{\mathrm{surv}}(t_k)$	dimensionless	Fraction of azimuth bins covered by surveillance within a window.
Time-allocation fractions	$f_{\mathrm{track}}, f_{\mathrm{trav}}, f_{\mathrm{surv}}$	dimensionless	Fraction of radar time spent on tracking, traverse, and surveillance.

Revisit Time

For storm i, the revisit time is defined as the elapsed time between two successive valid completed storm updates:

$$T_{i,n}^{\mathrm{rev}} = t_{i,n}^{\mathrm{end}} - t_{i,n-1}^{\mathrm{end}}. \tag{5.22}$$

The per-trial revisit-time metric is the median of the valid revisit intervals recorded during the simulation. Lower values indicate more frequent storm updates.

A completed storm scan is counted as a valid revisit only when it produces a successful storm observation. Unsuccessful scans are therefore excluded from the revisit calculations. Storms that are no longer observable by any radar are excluded from subsequent revisit-time evaluation.

Revisit Rate

The main scheduling metric is the storm revisit rate. The sliding-window evaluation of revisit performance used in this work is inspired by the evaluation approach of Reinoso-Rondinel et al. [12]. A revisit is counted when a storm-tracking task is successfully completed, meaning that the full azimuth sector occupied by the storm has been scanned and a valid storm observation is obtained. The revisit rate is computed using a sliding evaluation window. For

an evaluation time t_k, the window is defined as

$$\mathcal{W}(t_k) = [t_k - T_w,\, t_k], \tag{5.23}$$

where T_w is the window length. In this work, the window length is $T_w = 64.8$ s, a value that corresponds to the time required for one full 360° azimuth scan under the selected dwell time and azimuth resolution.

A completed storm-tracking task is counted within $\mathcal{W}(t_k)$ only if both its start time and end time lie inside the window:

$$t_{\text{start}} \geq t_k - T_w, \qquad t_{\text{end}} \leq t_k. \tag{5.24}$$

This avoids counting tasks that only partially overlap the evaluation window.

Let

$$\mathcal{M} = \{\text{B1}, \text{B2}, \text{RLTS}, \text{LATS}\} \tag{5.25}$$

denote the set of scheduling methods considered in this thesis, where B1 and B2 denote Baseline 1 and Baseline 2, respectively. For a scheduling method $m \in \mathcal{M}$, the revisit rate at time t_k is defined as

$$\lambda_m(t_k) = \frac{N_m(t_k)}{N_s T_w}, \tag{5.26}$$

where $N_m(t_k)$ is the number of completed storm revisits by method m within the window, N_s is the number of storms, and T_w is the window duration. The normalisation by N_s makes the metric comparable across scenarios with different numbers of storms. The window is shifted forward in increments of $\Delta t = 1$ s, which produces a time sequence of revisit-rate values over the simulation.

To avoid overestimating revisit performance, overlapping scans of the same storm by different radars are grouped. In addition, repeated updates of the same storm are counted as separate revisits only if they are separated by at least $\Delta t_{\min} = 5$ s. This prevents duplicate or nearly simultaneous radar scans from artificially increasing the revisit rate.

An effective revisit rate is also computed to measure how many distinct storms are updated within the window:

$$\lambda_{\text{eff},m}(t_k) = \frac{N_{\text{unique},m}(t_k)}{N_s T_w}, \tag{5.27}$$

where $N_{\text{unique},m}(t_k)$ is the number of unique storms updated by method m within $\mathcal{W}(t_k)$. This metric is useful for identifying whether a scheduler distributes updates across many storms or repeatedly revisits only a subset of storms.

Improvement Factor

The improvement factor is based on the revisit-improvement measure introduced by Reinoso-Rondinel et al. [12]. In this work, it compares the revisit-rate performance of a proposed scheduler against a baseline scheduler. For adaptive method a and baseline method b, the

window-wise improvement factor is defined as:

$$I_{a,b}(t_k) = \frac{\lambda_a(t_k)}{\lambda_b(t_k)}. \tag{5.28}$$

Here, $\lambda_a(t_k)$ is the revisit rate of the adaptive scheduler and $\lambda_b(t_k)$ is the revisit rate of the baseline scheduler in the same sliding window. If the baseline revisit rate is zero in a window, the corresponding improvement factor is undefined. Such windows are excluded from the window-wise averaging of improvement factors to avoid reporting artificial infinite values.

Two baseline comparisons are used:

$$I_{a,\mathrm{B1}}(t_k) = \frac{\lambda_a(t_k)}{\lambda_{\mathrm{B1}}(t_k)}, \tag{5.29}$$

and

$$I_{a,\mathrm{B2}}(t_k) = \frac{\lambda_a(t_k)}{\lambda_{\mathrm{B2}}(t_k)}. \tag{5.30}$$

Baseline 1 denotes the storm-by-storm scanning method, while Baseline 2 denotes the 360° sit-and-spin scanning method. An improvement factor greater than one indicates that the adaptive scheduler achieves a higher revisit rate than the baseline within that window.

A value close to one indicates similar revisit-rate performance, while a value below one indicates that the adaptive scheduler achieves a lower revisit rate than the baseline for that window.

For one simulation trial, the mean improvement factor is obtained by averaging the window-wise improvement factors:

$$\bar{I}_{a,b} = \frac{1}{N_t} \sum_{k=1}^{N_t} I_{a,b}(t_k), \tag{5.31}$$

where N_t is the number of evaluation times associated with the same sliding windows of duration T_w used to compute the revisit rates. The trial-level mean improvement factor is computed for each simulation trial. The resulting trial-level values form the scenario-level distribution, which is summarised using its median and displayed using boxplots.

Tracking Accuracy

Tracking accuracy is evaluated by comparing the estimated storm footprint with the true simulated storm footprint using Intersection over Union (IoU). The intersection over union measures the overlap between the true storm occupancy and the estimated storm occupancy. For storm i, it is defined as

$$\mathrm{IoU}_i = \frac{|\Omega_i^{\mathrm{true}} \cap \Omega_i^{\mathrm{est}}|}{|\Omega_i^{\mathrm{true}} \cup \Omega_i^{\mathrm{est}}|}, \tag{5.32}$$

where Ω_i^{true} is the true storm occupancy region and Ω_i^{est} is the estimated storm occupancy region. A value of one indicates perfect overlap, while a value close to zero indicates poor overlap.

IoU is evaluated at completed storm updates and is used to assess how the measurement sequences produced by the different scheduling methods affect the accuracy of the estimated storm footprint.

Surveillance and Time Allocation

In addition to storm tracking, the schedulers are evaluated based on how much radar time remains available for surveillance. This is important because a scheduler could increase revisit rate by allocating most radar time to tracking, but this may reduce background coverage of the azimuth domain.

Radar time is separated into three categories: tracking time, traverse time, and surveillance time. For a simulation with R radars and total duration T_{sim}, the total available radar time is

$$T_{\text{avail}} = RT_{\text{sim}}. \tag{5.33}$$

The tracking-time fraction is

$$f_{\text{track}} = \frac{T_{\text{track}}}{T_{\text{avail}}}, \tag{5.34}$$

the traverse-time fraction is

$$f_{\text{trav}} = \frac{T_{\text{trav}}}{T_{\text{avail}}}, \tag{5.35}$$

and the surveillance-time fraction is

$$f_{\text{surv}} = \frac{T_{\text{surv}}}{T_{\text{avail}}}. \tag{5.36}$$

Surveillance coverage is computed using the same sliding-window approach. For each window, the number of unique azimuth bins scanned in surveillance mode is divided by the total number of azimuth bins:

$$C_{\text{surv}}(t_k) = \frac{N_{\text{surv,bins}}(t_k)}{N_\theta}. \tag{5.37}$$

Here, $N_{\text{surv,bins}}(t_k)$ is the number of unique azimuth bins scanned in surveillance mode by the radar network within the window, and

$$N_\theta = 720 \tag{5.38}$$

is the total number of azimuth bins. The mean surveillance coverage is obtained by averaging $C_{\text{surv}}(t_k)$ across all sliding windows. This metric indicates how much of the angular domain is covered by surveillance during the simulation.

5.5 Planning-Window Sensitivity

For LATS, an additional planning-window sensitivity analysis is performed. The planning window controls how often the load-aware storm-to-radar assignment is recomputed. A short

planning window allows the assignment to respond quickly to changes in storm position, urgency, and radar geometry, but it may also lead to frequent reassignment. A longer planning window produces more stable storm-to-radar assignments, but may respond more slowly when the storm field changes.

The tested planning-window lengths are

$$N_{\text{plan}} \in \{1, 2, 5, 10, 20, 40, 80\} \tag{5.39}$$

dwells. For each value, LATS is evaluated using the same scenario definitions and metric calculations described in the previous sections. The resulting average revisit rate is used to determine how sensitive the load-aware assignment scheduler is to the assignment-update frequency.

Chapter 6

Results

This chapter presents the performance of the Radar-Level Task Selection (RLTS) and Load-Aware Task Scheduling (LATS) schedulers and compares them with the two baseline strategies. The results are presented at two complementary levels. First, a representative 20 min trial from the nominal *scenario is examined to show how the scheduling behaviour develops over time and how it is reflected in storm urgency, revisit performance, and tracking quality.*

The analysis is then extended to the complete Monte Carlo campaign of 50 trials per scenario to determine whether the behaviour observed in the representative trial persists across the low-load*,* nominal*,* high-load*,* fast-storm*, and* large-storm *scenarios. The representative trial is therefore used to illustrate scheduler behaviour in detail, while the overall performance conclusions are drawn from the full 20 min simulations through the scenario-level distributions and statistics.*

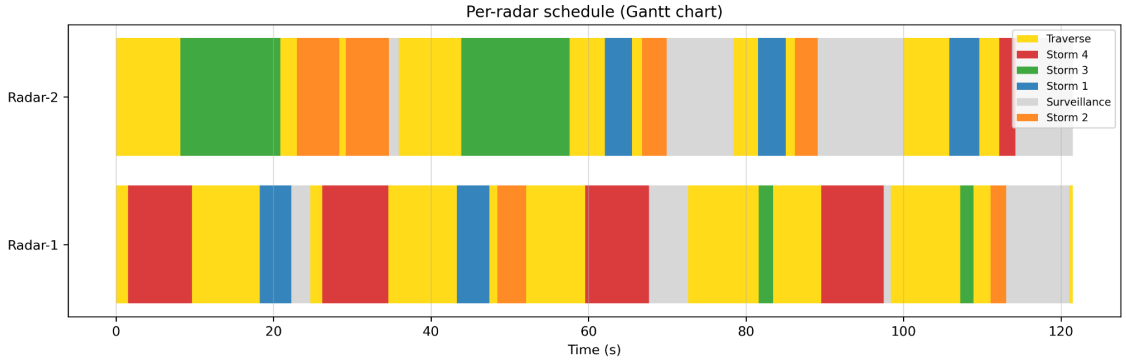
6.1 Representative Nominal-Scenario Results

The *nominal* scenario is selected as the representative case because it contains four simultaneously evolving storms and introduces competition for radar time without imposing the extreme conditions of the dedicated stress scenarios. The complete trial duration is 20 min, consistent with the experimental configuration defined in the previous chapter. For visual clarity, the detailed scheduling results in this section show a representative 2 min interval from the 20 min trial, allowing individual tracking, traverse, and surveillance activities to remain clearly distinguishable.

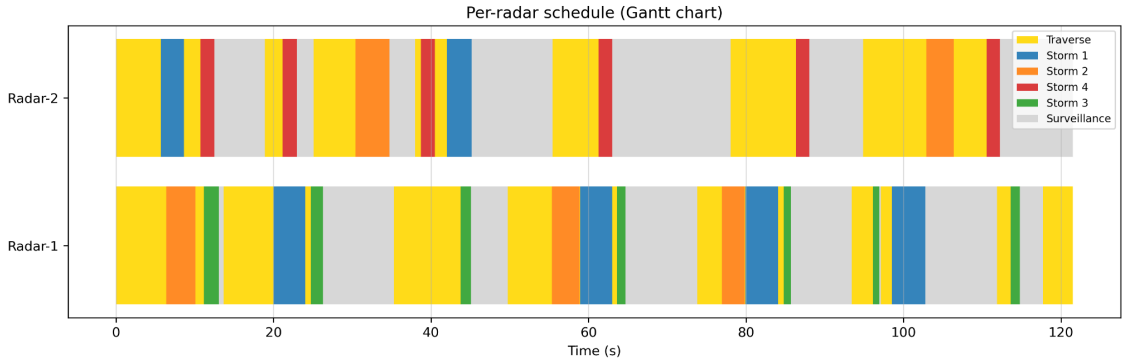
Representative trials for the remaining *low-load*, *high-load*, *fast-storm*, and *large-storm* scenarios are provided in Appendix A.1.

6.1.1 Radar Scheduling Behaviour

The first result examines how the two schedulers distribute the available radar time. Figure 6.1 shows the schedules generated by RLTS and LATS for the same *nominal* trial. Coloured intervals represent storm-sector scans, yellow intervals indicate radar traverse, and grey intervals indicate surveillance.



(a) Gantt chart of RLTS adaptive scheduler.



(b) Gantt chart of LATS adaptive scheduler.

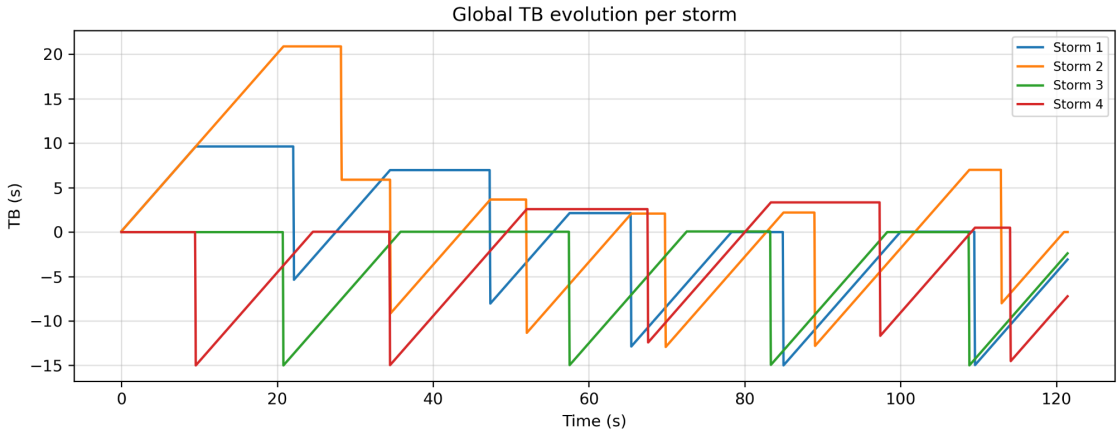
Figure 6.1: Per-radar schedules during the displayed 2 min interval of the representative 20 min *nominal* trial.

The Gantt charts of Figure 6.1 show how the two schedulers distribute radar time between storm tracking, traverse, and surveillance during the displayed interval. In the RLTS schedule, Figure 6.1a, both radars service several different storms, and storm responsibility can change as the feasible candidate set and task-selection scores evolve. In contrast, the LATS schedule in Figure 6.1b shows a more persistent division of storm responsibility between the two radars. This behaviour is consistent with the additional storm-to-radar assignment stage in LATS, which constrains the distribution of storm tasks before radar-level task selection.

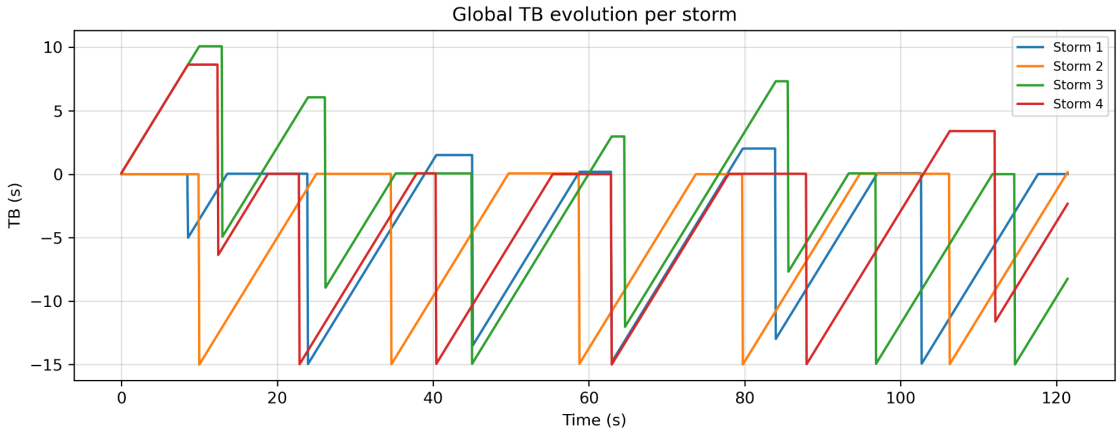
The LATS schedule also retains visibly more surveillance time during the displayed interval. This suggests that the more structured distribution of storm tasks can leave greater radar capacity for background surveillance, rather than using the additional coordination simply to increase tracking activity. Since the figure represents only one 2 min interval of a single trial, this observation is illustrative; the scenario-level surveillance and radar-time-allocation results later show whether the same behaviour persists across the Monte Carlo campaign.

6.1.2 Time-Balance Evolution

Figure 6.2 shows the time-balance evolution of the four storms for RLTS and LATS during the same *nominal* trial. As defined in Chapter 4, a rising TB_i indicates that storm i is waiting for an update. The time-balance value remains constant while the storm task is active and is reduced following a completed revisit.



(a) RLTS adaptive scheduler.



(b) LATS adaptive scheduler.

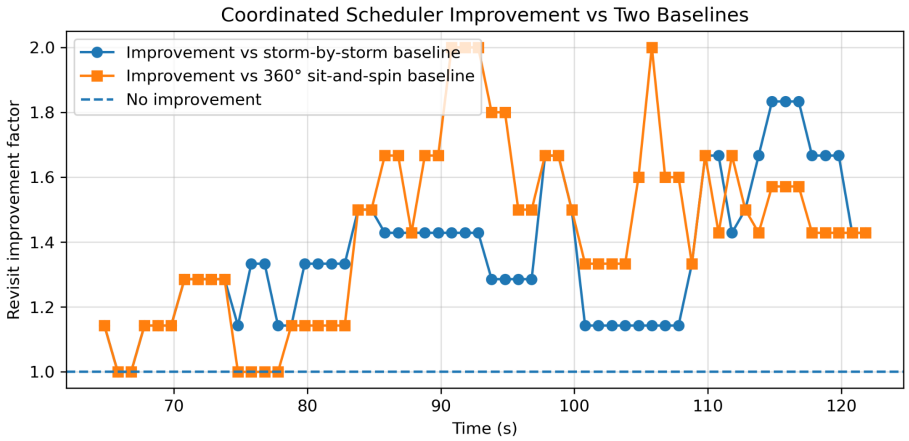
Figure 6.2: Time-balance evolution during the displayed 2 min interval of the representative 20 min *nominal* trial.

The RLTS result in Figure 6.2(a) shows different waiting histories for the four storms. In particular, Storm 2 develops the largest positive time-balance excursion during the first part of the displayed interval before its next completed revisit. The LATS result in Figure 6.2(b) shows smaller positive excursions and repeated reductions across all four storm histories. These reductions correspond to completed storm updates and can be related to the tracking intervals shown in Figure 6.1(b).

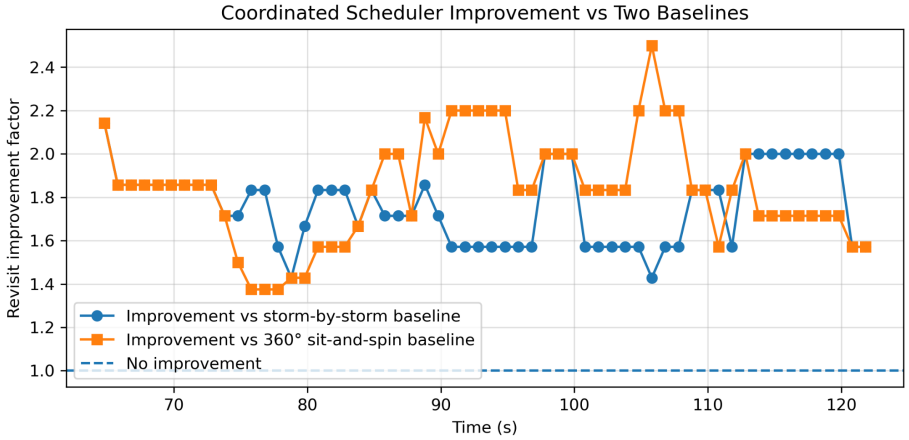
In this representative interval, the smaller positive time-balance excursions under LATS indicate that the storms accumulate less overdue time relative to their requested revisit intervals before being serviced. This is consistent with the more structured distribution of

storm responsibility introduced by the load-aware assignment stage. However, time balance is an internal scheduling state rather than a direct performance metric, so this figure alone does not establish that LATS performs better overall. The revisit-performance results in the following sections determine whether the same tendency persists across the Monte Carlo campaign.

6.1.3 Revisit Improvement During the Trial



(a) Improvement factor of RLTS over time.



(b) Improvement factor of LATS over time.

Figure 6.3: Window-wise revisit improvement of RLTS and LATS relative to both baseline methods during the displayed 2 min interval of the representative 20 min *nominal* trial. In the figure legends, the storm-by-storm baseline corresponds to Baseline 1, while the 360° sit-and-spin baseline corresponds to Baseline 2. The improvement factor is evaluated using a 64.8 s sliding window shifted in 1 s increments.

Figure 6.3 shows the window-wise improvement factor of RLTS and LATS relative to Baseline 1 and Baseline 2. The horizontal line at $I = 1$ represents equal revisit-rate performance with the corresponding baseline. Values above this line indicate improvement over the baselines. For each evaluation time, the revisit rate is computed over a 64.8 s sliding window,

corresponding to one complete 360° scan, and only completed revisits whose start and end times both lie within the window are included. The window is advanced in 1 s increments. Unlike the absolute revisit time, the improvement factor expresses the scheduler performance relative to each baseline and therefore shows how the relative revisit advantage changes as the task configuration evolves.

For RLTS, shown in Figure 6.3(a), the improvement relative to Baseline 1 remains above unity throughout the displayed interval, while the comparison with Baseline 2 is more variable and approaches the equal-performance level during several earlier windows. For LATS, shown in Figure 6.3(b), both baseline comparisons remain above unity throughout the displayed interval, with the improvement relative to Baseline 2 reaching a peak of approximately 2.5.

The different improvement levels relative to the two baselines show that the measured benefit depends on the reference scanning strategy. The step-like changes in the curves are also a consequence of the sliding-window calculation: the metric changes when completed revisits enter or leave the 64.8 s evaluation window. Within this representative interval, LATS maintains a more consistent improvement over both baselines than RLTS, although the scenario-level Monte Carlo results are required to determine whether this behaviour persists across different storm realisations.

The final result from the representative trial considers whether these different update sequences are also reflected in tracking quality.

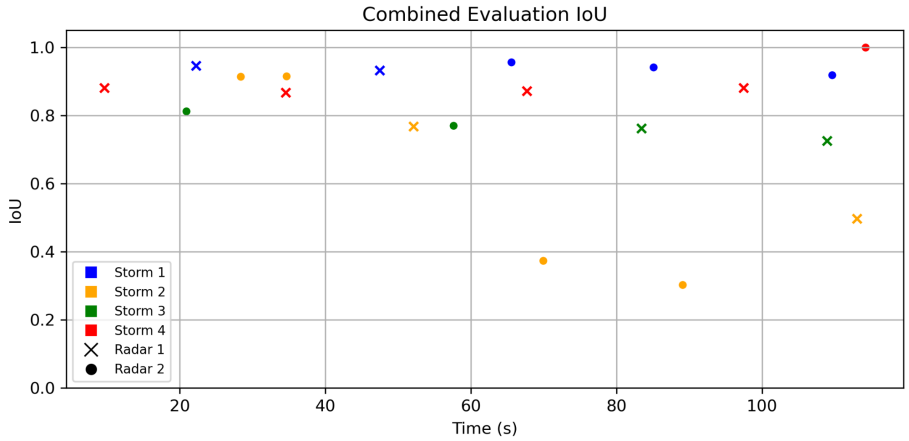
6.1.4 Tracking Quality During the Trial

Figure 6.4 shows the Intersection over Union (IoU) obtained at individual storm updates for RLTS and LATS. Marker colour identifies the different storm, while marker shape (cross or circle) identifies the radar responsible for the update.

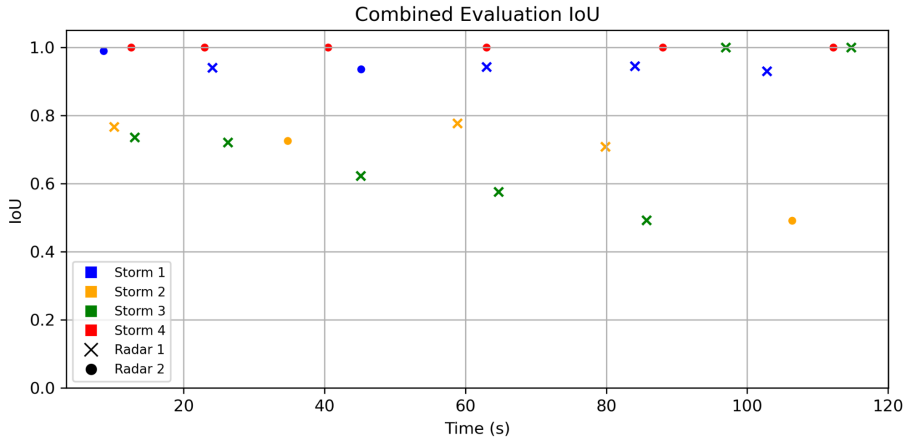
Both schedulers produce several updates with IoU values close to unity, while the tracking quality varies between storms and update times. In the RLTS result, Figure 6.4a, Storms 1 and 4 contain several high-IoU updates, whereas Storms 2 and 3 show larger variation. This variation can occur when only part of a storm is observed by the radar, since the estimated storm centre and radius are calculated from the detected storm cells. Partial observations can therefore reduce the overlap between the estimated and true storm regions.

The LATS result in Figure 6.4b shows the same storm-dependent behaviour. Several updates again lie close to unity, while lower-IoU observations remain present for Storms 2 and 3. Since the same storms show larger IoU variation under both schedulers, this suggests that the lower-IoU updates are more strongly related to the storm observation conditions than to the choice of scheduling method.

In this representative interval, the different scheduling strategies therefore do not appear to introduce a systematic difference in tracking quality, with the observed IoU variation instead appearing to depend more strongly on the storm being observed and the corresponding observation conditions.



(a) RLTS adaptive scheduler.



(b) LATS adaptive scheduler.

Figure 6.4: IoU at individual storm updates during the displayed 2 min interval of the representative 20 min *nominal* trial.

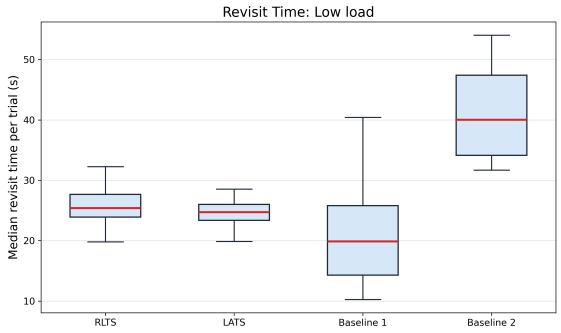
The individual-update plots provide a detailed view of the temporal and storm-to-storm variation in tracking quality during the displayed 2 min interval. Since this interval represents only a portion of the complete 20 min simulation and a single storm realisation, the following sections evaluate performance using the full-duration Monte Carlo campaign over different load and storm related scenarios.

6.2 Revisit Performance Across the Monte Carlo Campaign

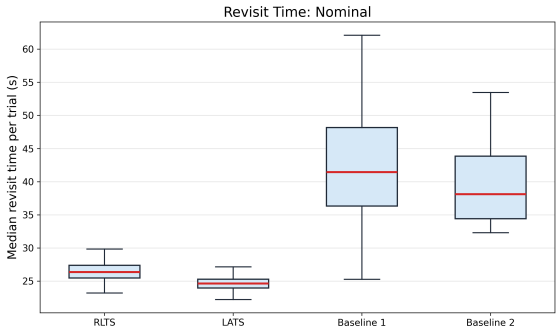
The representative *nominal* trial illustrates how RLTS and LATS operate for one storm realisation. The Monte Carlo campaign of 50 trials per scenario is used to determine whether the resulting revisit behaviour persists when the storm positions, sizes, velocities, reflectivities, and growth rates vary between trials.

In view of this, Figure 6.5 compares the distribution of the per-trial median revisit time for RLTS, LATS, Baseline 1, and Baseline 2 across the five scenarios. Since revisit time measures

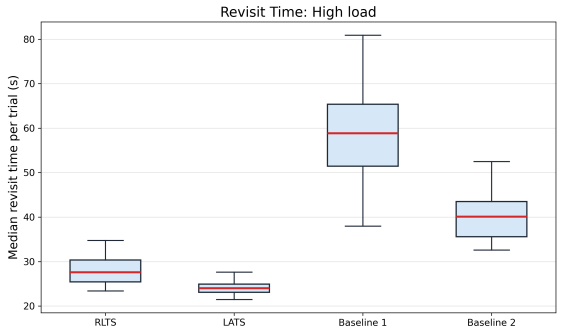
the interval between completed storm updates, lower values correspond to more frequent revisits, hence an overall better performance. The *low-load* scenario in Figure 6.5(a) differs from the remaining cases. Baseline 1 has the lowest median revisit time at 19.89 s, while the medians for RLTS, LATS, and Baseline 2 are 25.38 s, 24.73 s, and 40.03 s, respectively. Under this condition, the fixed storm-by-storm cycle is essentially sufficient to revisit the small storm set frequently.



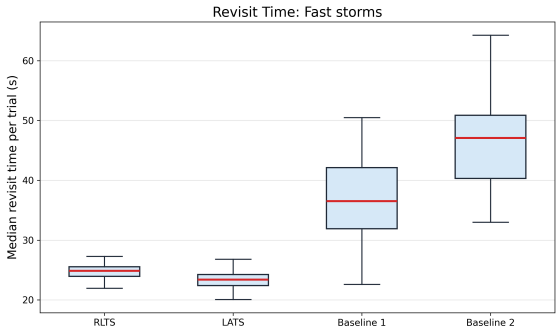
(a) Low load.



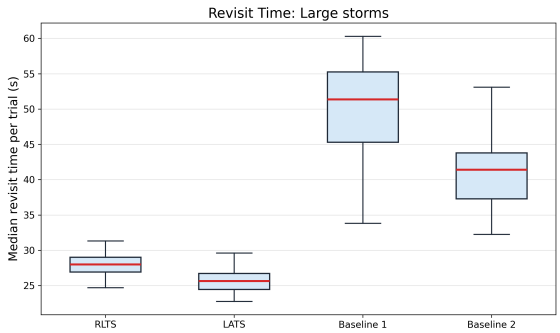
(b) Nominal.



(c) High load.



(d) Fast storms.



(e) Large storms.

Figure 6.5: Distribution of per-trial median revisit time across the five Monte Carlo scenarios (the lower the better), based on 50 trials per scenario. The red horizontal line in each box indicates the median. For each scenario, the four schedulers discussed in this work are analysed.

The *nominal* case in Figure 6.5(b) shows a different ordering. RLTS and LATS have median revisit times of 26.37 s and 24.64 s, respectively, while Baseline 1 and Baseline 2 require 41.45 s and 38.12 s between completed storm updates. The separation is the largest

under *high-load* in Figure 6.5(c). LATS achieves the lowest median revisit time at 24.01 s, followed by RLTS at 27.57 s. Baseline 1 and Baseline 2 have substantially longer medians of 58.82 s and 40.08 s, respectively. Baseline 1 shows the widest spread in this scenario.

The stronger separation under *high-load* is consistent with the increased competition for radar time. In particular, Baseline 1 relies on a single radar for storm tracking, which becomes increasingly restrictive as more storms require service, whereas RLTS and LATS can make use of both radar nodes for tracking.

The *fast-storm* result in Figure 6.5(d) follows the same general trend as the *nominal* case. The median revisit times are 24.86 s for RLTS, 23.40 s for LATS, 36.52 s for Baseline 1, and 47.07 s for Baseline 2. The *large-storm case* in Figure 6.5(e) represents a different type of increased radar demand. The number of storms remains unchanged, but the larger storm sectors require longer scans. RLTS and LATS have median revisit times of 27.99 s and 25.65 s, while Baseline 1 and Baseline 2 have medians of 51.37 s and 41.40 s, respectively.

The fast- and large-storm scenarios also show that increased scheduling difficulty can arise in different ways. Faster storm motion mainly changes how rapidly the storm state evolves, whereas larger storms directly increase the duration of the required sector scans. The latter therefore places a more direct demand on the available radar-time budget.

The adaptive schedulers also show comparatively tighter revisit-time distributions in the more demanding scenarios, indicating less trial-to-trial variation in revisit performance. In contrast, the wider spread of the baseline methods, particularly Baseline 1 under *high-load*, indicates a stronger dependence on the specific storm configuration generated in each trial.

Across the five scenarios, the revisit-time distributions show that the advantage of adaptive scheduling becomes clearer once the radar network must serve several competing or time-consuming storm tasks. Baseline 1 remains strongest only in the *low-load* case, where a simple cyclic strategy is already sufficient. In the remaining four scenarios, LATS achieves the shortest median revisit time, followed by RLTS, with the clearest separation from the baseline methods occurring under *high-load*. This indicates that adaptive task selection and load-aware assignment become most beneficial when the available radar resources are more strongly constrained.

6.2.1 Scenario-Level Revisit-Time Comparison

The scenario-level median revisit times are summarised in Table 6.1. Since lower revisit times correspond to more frequent storm updates, the relative differences between the schedulers are expressed in the subsequent analysis as percentage changes with respect to the method used for comparison.

The *low-load* scenario is the only case in which Baseline 1 provides the shortest median revisit time. Compared with Baseline 1, the revisit times of RLTS and LATS are 27.6% and 24.3% longer, respectively. However, both adaptive schedulers substantially outperform Baseline 2, with RLTS reducing the median revisit time by 36.6% and LATS by 38.2%. Compared directly with RLTS, LATS provides a small reduction of 2.6%.

Table 6.1: Median revisit time for each scheduler and scenario.

Scenario	RLTS (s)	LATS (s)	Baseline 1 (s)	Baseline 2 (s)
Low load	25.38	24.73	19.89	40.03
Nominal	26.37	24.64	41.45	38.12
High load	27.57	24.01	58.82	40.08
Fast storms	24.86	23.40	36.52	47.07
Large storms	27.99	25.65	51.37	41.40

In the *nominal* scenario, the advantage shifts to the adaptive schedulers. RLTS reduces the median revisit time by 36.4% relative to Baseline 1 and 30.8% relative to Baseline 2. LATS provides larger reductions of 40.6% and 35.4% relative to Baseline 1 and Baseline 2, respectively. Compared directly with RLTS, LATS further reduces the median revisit time by 6.6%.

The largest separation occurs in the *high-load* scenario. RLTS reduces the median revisit time by 53.1% relative to Baseline 1 and 31.2% relative to Baseline 2. LATS provides larger reductions of 59.2% and 40.1% relative to the two baselines, respectively. Compared directly with RLTS, LATS further reduces the median revisit time by 12.9%, representing the largest RLTS–LATS difference among the five scenarios.

The *fast-storm* scenario shows a similar advantage for the adaptive methods. RLTS reduces the median revisit time by 31.9% relative to Baseline 1 and 47.2% relative to Baseline 2. The corresponding reductions for LATS are 35.9% and 50.3%, respectively. Compared directly with RLTS, LATS provides a further reduction of 5.9%.

In the *large-storm* scenario, RLTS reduces the median revisit time by 45.5% relative to Baseline 1 and 32.4% relative to Baseline 2. LATS increases these reductions to 50.1% and 38.0%, respectively, while also reducing the median revisit time by 8.4% relative to RLTS.

Across the five scenarios, the benefit of adaptive scheduling becomes more apparent as competition for the available radar time increases. Baseline 1 remains strongest only in the *low-load* boundary case, where two storms can already be revisited efficiently using a simple cyclic strategy and there is therefore little opportunity for adaptive scheduling to provide an additional benefit. As more storms compete for service, or individual storm scans become more time-consuming, RLTS can respond to the current revisit urgency and task state rather than following a fixed scan sequence. LATS provides an additional benefit by explicitly distributing this workload between the two radar nodes, which becomes most relevant when the network is strongly constrained. This is consistent with the largest RLTS–LATS difference being observed under *high-load*. Since these trends are obtained from the scenario-level distributions across 50 Monte Carlo trials, they are not restricted to the single representative nominal realisation.

6.3 Improvement Relative to the Baseline Methods

The revisit-time results describe the absolute interval between completed storm updates. The improvement factor provides a normalised comparison between the proposed schedulers and each baseline. An improvement factor of $I = 1$ indicates equal revisit-rate performance. Values greater than one indicate a higher revisit rate for the proposed scheduler, while values below one indicate a lower revisit rate. For example, $I = 1.5$ corresponds to a revisit rate that is 50% higher than that of the baseline, while $I = 2$ corresponds to twice the baseline revisit rate.

Figure 6.6 summarises the median improvement factors of RLTS and LATS relative to Baseline 1 and Baseline 2 across the five scenarios.

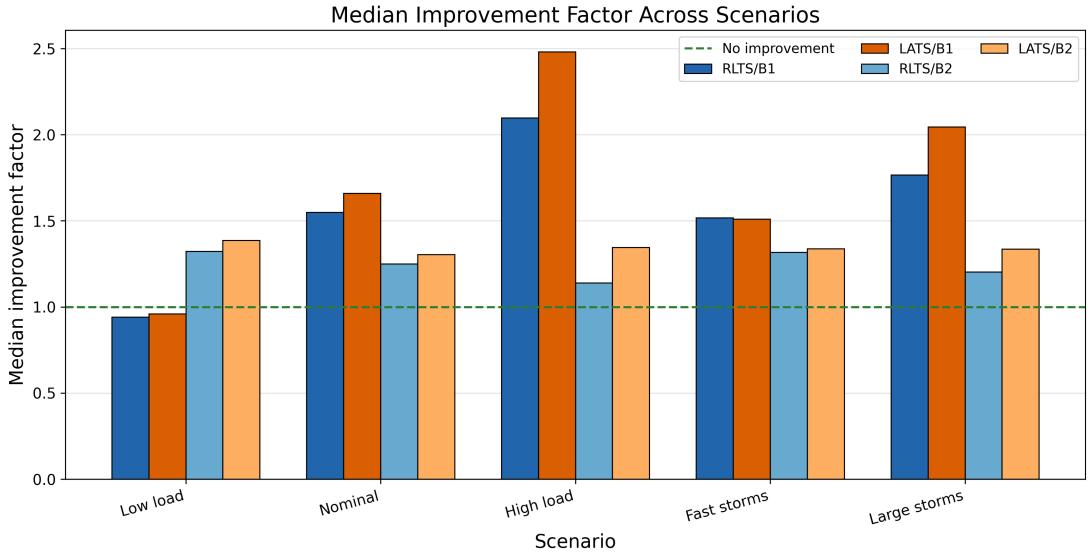


Figure 6.6: Median improvement factor of RLTS and LATS relative to Baseline 1 and Baseline 2 across the five Monte Carlo scenarios.

The comparison with Baseline 1 shows a strong dependence on radar load. In the *low-load* scenario, the median improvement factors are 0.94 for RLTS and 0.96 for LATS, placing both slightly below the equal-performance level. This agrees with the revisit-time results, where Baseline 1 provides the shortest revisit time when only two storms are present.

Once radar demand increases, the ordering changes. In the *nominal* scenario, the median improvement factors increase to 1.55 for RLTS and 1.66 for LATS. The strongest improvement relative to Baseline 1 occurs in the *high-load* scenario, where RLTS reaches 2.10 and LATS reaches 2.48. The LATS revisit rate is therefore 2.48 times that of Baseline 1, corresponding to a 148% higher revisit rate according to the defined improvement metric.

The *fast-storm* scenario gives median improvement factors of 1.52 for RLTS and 1.51 for LATS relative to Baseline 1, indicating almost identical performance between the two proposed schedulers for this comparison. In the *large-storm* scenario, the median factors increase to 1.77 for RLTS and 2.04 for LATS.

The contrast between these scenarios indicates that the additional benefit of load-aware

assignment depends on the source of difficulty. Under *high-load* and *large-storm* conditions, the radar-time demand is increased by a larger number of competing tasks or by longer sector scans, making the distribution of workload between radar nodes more important. In the *fast-storm* case, where RLTS and LATS perform almost identically relative to Baseline 1, faster storm evolution does not create the same workload-allocation problem.

The comparison with Baseline 2 shows a different and more stable pattern. Both RLTS and LATS remain above the equal-performance level in all five scenarios. The median improvement factors range from 1.14 to 1.32 for RLTS and from 1.30 to 1.39 for LATS, with LATS retaining the higher median in each scenario. Unlike the comparison with Baseline 1, the improvement does not increase systematically with radar load.

This difference is consistent with the operating principle of Baseline 2. Its revisit behaviour is primarily governed by the fixed 360° scan cycle rather than by the number of storm tasks competing for service. Furthermore, the revisit-rate calculation groups overlapping or near-simultaneous observations of the same storm and does not count them as independent revisits. Consequently, repeated multi-radar observations of the same storm do not artificially increase the measured revisit performance, which becomes particularly relevant when several storms compete for the available radar time.

The median values describe the typical improvement across the 50 trials but do not show the trial-to-trial variation. Figure 6.7 therefore presents the full distributions of the four improvement-factor comparisons in each scenario. The red line within each box indicates the median, while the dashed horizontal line at $I = 1$ marks equal revisit-rate performance with the corresponding baseline.

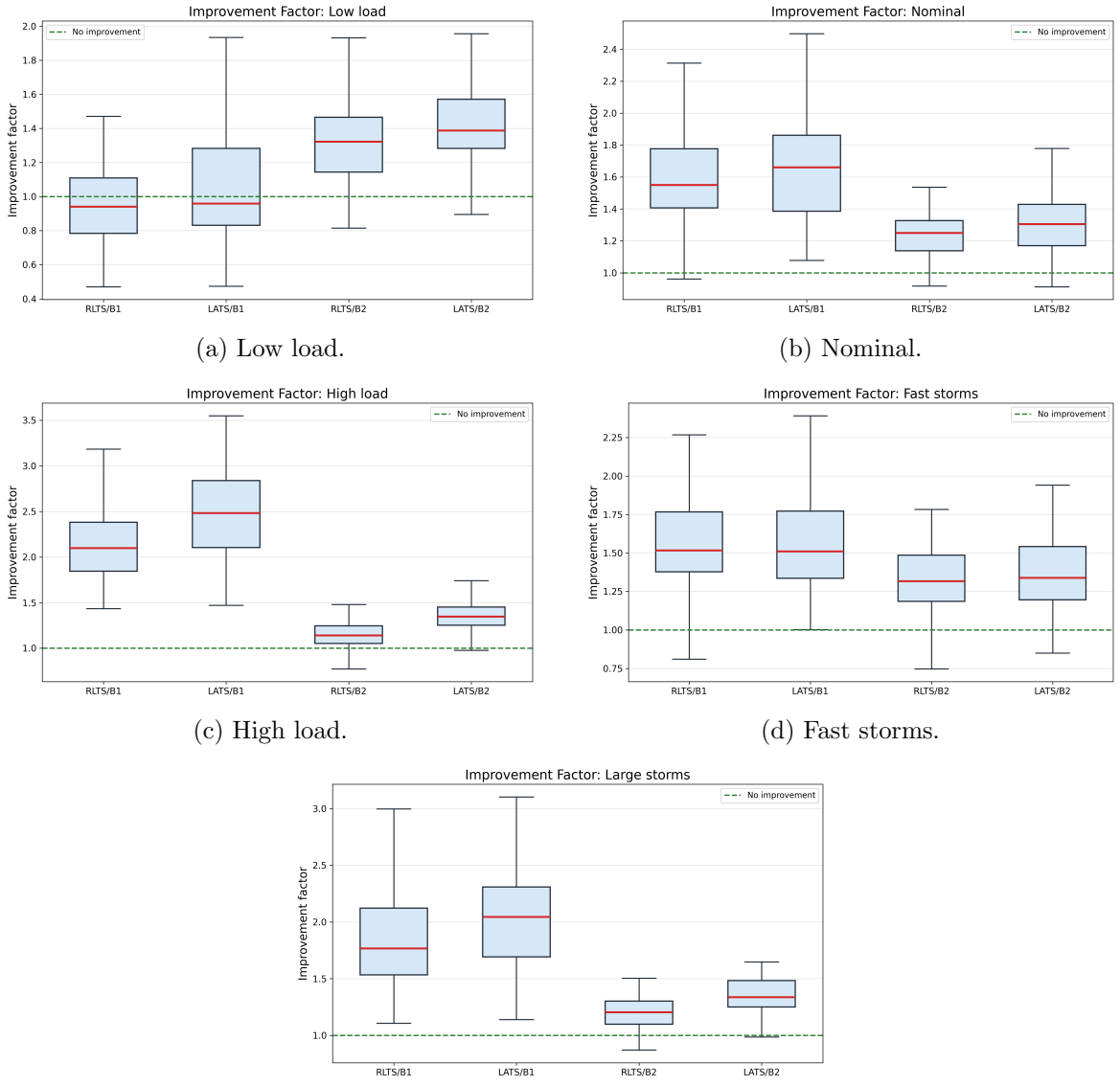
The *low-load* distributions in Figure 6.7a show the boundary condition already identified from the revisit-time results. The RLTS/B1 and LATS/B1 distributions are centred close to the equal-performance level, whereas the comparisons with Baseline 2 remain clearly above unity.

In the *nominal* scenario, Figure 6.7b, the distributions of all four comparisons are centred above $I = 1$, with the improvement relative to Baseline 1 generally stronger than that relative to Baseline 2.

The *high-load* distributions in Figure 6.7c show the strongest separation from Baseline 1 and greater trial-to-trial variation than the corresponding Baseline 2 comparisons. Despite this increased spread, both adaptive schedulers remain centred well above the equal-performance level.

In the *fast-storm* scenario, Figure 6.7d, RLTS and LATS show very similar distributions relative to Baseline 1, consistent with the nearly identical median values reported above. Relative to Baseline 2, the two distributions also remain close, with LATS retaining a small advantage.

The *large-storm* distributions in Figure 6.7e again show a stronger improvement relative to Baseline 1 than relative to Baseline 2, with LATS showing the larger improvement in both comparisons.



(a) Low load.

(b) Nominal.

(c) High load.

(d) Fast storms.

(e) Large storms.

Figure 6.7: Distribution of the improvement factor relative to both baseline methods across the five Monte Carlo scenarios, based on 50 trials per scenario.

Overall, the improvement-factor results show that the benefit of adaptive scheduling is conditional on how strongly the radar resources are constrained and on the baseline used for comparison. Baseline 1 remains competitive under *low-load*, where a simple cyclic strategy can already service the small number of storm tasks efficiently. As competition for radar time increases, RLTS benefits from selecting tasks according to their current scheduling state, while LATS gains an additional advantage when the workload must be distributed between the radar nodes. This explains why the largest LATS benefit occurs under *high-load* and remains pronounced for *large storms*, whereas little additional benefit over RLTS is observed for *fast storms*. In contrast, the improvement relative to Baseline 2 remains more stable because its revisit behaviour is dominated by the fixed full-scan cycle. The Monte Carlo distributions therefore indicate that these are systematic scenario-level trends, while

also showing that the magnitude of the improvement can vary between individual storm configurations.

6.4 Tracking Accuracy Across the Monte Carlo Campaign

The representative nominal results in Figure 6.4 illustrate how tracking quality varies between individual storm updates. The Monte Carlo campaign of 50 trials per scenario is used to examine how the overall tracking accuracy changes under different storm conditions over the complete 20 min simulations. Table 6.2 reports the mean IoU obtained by RLTS and LATS across the five scenarios.

Table 6.2: Mean IoU of RLTS and LATS across the five Monte Carlo scenarios.

Scenario	RLTS	LATS
Low load	0.69	0.68
Nominal	0.69	0.68
High load	0.67	0.67
Fast storms	0.58	0.58
Large storms	0.61	0.61

An IoU in the range of approximately 0.65–0.70 indicates substantial, but not complete, overlap between the estimated and true storm footprints. For example, an IoU of 0.69 indicates that the overlapping region represents approximately 69% of the combined area covered by the estimated and true storm regions. The remaining mismatch can arise from an offset in the estimated storm centre, an error in the estimated radius, or a combination of both. In the simulation, these differences can result from partial observation of the storm footprint and from prediction error accumulated between successive measurement updates.

The highest tracking accuracy is observed in the *low-load* and *nominal* scenarios, where the mean IoU remains close to 0.69 for both scheduling methods. Under *high-load*, the mean IoU decreases only slightly to approximately 0.67. The relatively small reduction despite the higher competition for radar time indicates that increasing the number of competing storm tasks primarily affects how frequently storms can be revisited, while the quality of the storm-footprint estimate at a completed update remains comparatively stable.

A considerably larger reduction is observed in the *fast-storm* scenario, where both methods obtain a mean IoU of approximately 0.58, compared with values of approximately 0.67–0.69 in the low-load, nominal, and high-load scenarios. This is the lowest tracking accuracy among the five scenarios. For faster-moving storms, a given interval between successive measurements corresponds to a larger spatial displacement. Consequently, prediction errors or delays between updates can produce a larger offset between the estimated and true storm footprints, reducing their spatial overlap and therefore the IoU.

Tracking accuracy also decreases in the *large-storm* scenario, where both methods obtain a mean IoU of approximately 0.61. The larger storm footprints require wider sector scans

and therefore longer scan durations. This increases the time required to complete an observation and can increase the mismatch between the predicted and true storm states between successive updates.

Overall, the differences in mean IoU between RLTS and LATS remain small across all five scenarios, whereas substantially larger differences are observed between the scenario conditions. This indicates that the choice between the two adaptive scheduling methods has only a limited influence on tracking accuracy in the evaluated campaign. Increased radar load alone has only a modest effect on IoU, while changes in the storm characteristics have a stronger influence. In particular, fast storm motion produces the clearest degradation in tracking quality, as the greater spatial displacement between updates makes accurate prediction of the storm footprint more difficult.

6.5 Surveillance and Radar-Time Allocation

The previous results show how the proposed schedulers affect storm revisit performance and tracking accuracy. However, storm tracking is only one part of the radar network's operation. The remaining radar capacity must also support background surveillance. The Monte Carlo results are therefore examined in terms of surveillance coverage and the fraction of radar time allocated to tracking, traverse, and surveillance.

The representative schedules in Figure 6.1 already showed that both RLTS and LATS retain periods of background surveillance while servicing storm tasks. Figure 6.8 extends this comparison across the five Monte Carlo scenarios.

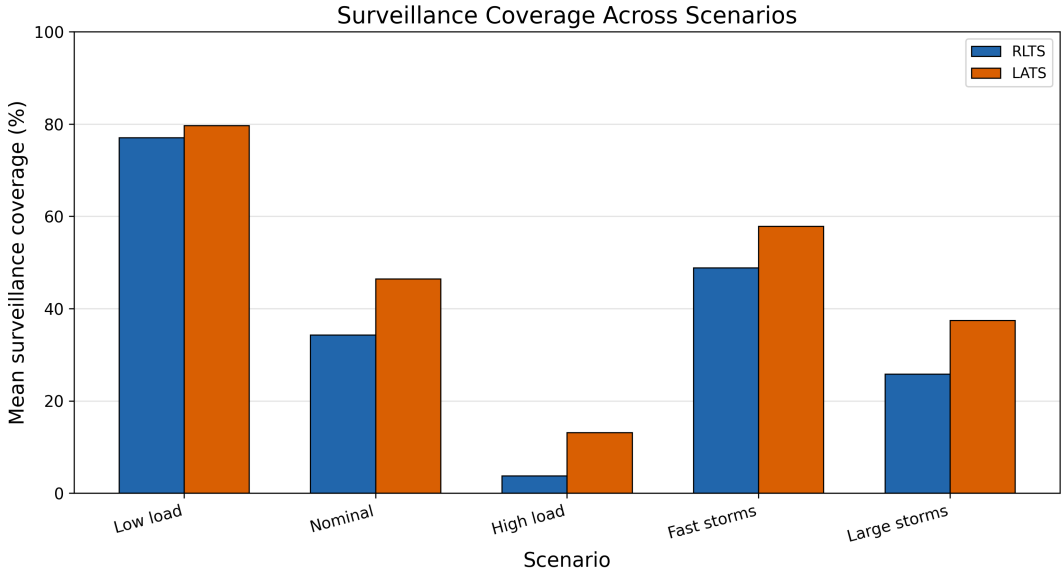


Figure 6.8: Mean surveillance coverage of RLTS and LATS across the five Monte Carlo scenarios.

Figure 6.8 shows that surveillance coverage varies considerably with scenario difficulty. The *low-load* scenario retains the highest coverage, with mean values of 77.05% for RLTS and

79.69% for LATS. In the *nominal* scenario, mean surveillance coverage decreases to 34.32% for RLTS and 46.48% for LATS. With four storms competing for radar service, a larger fraction of the available radar capacity is required for storm tracking and radar traverse.

The strongest reduction occurs under *high-load*. Mean surveillance coverage falls to 3.79% for RLTS and 13.13% for LATS. This is the scenario in which six storms compete for the available radar resources, leaving only a small portion of the surveillance region to be revisited. The *fast-storm* scenario retains considerably more surveillance coverage than the *high-load* case. RLTS achieves a mean coverage of 48.84%, while LATS achieves 57.87%. This indicates that increased storm motion does not consume radar capacity in the same way as an increase in the number of simultaneously competing storm tasks. For large storms, mean surveillance coverage decreases to 25.78% for RLTS and 37.45% for LATS. In this case, the wider storm sectors increase the time required for individual storm scans and reduce the radar capacity available for background surveillance.

Across all five scenarios, LATS retains more surveillance coverage than RLTS. The difference is small under *low-load* but becomes more pronounced in the *nominal*, *high-load*, *fast-storm*, and *large-storm* scenarios. The radar-time allocation in Figure 6.9 provides further insight into how the available radar capacity is distributed between the different activities.

Surveillance coverage and surveillance-time allocation describe different aspects of this behaviour. Surveillance coverage measures how much of the surveillance region is revisited, whereas surveillance-time allocation measures how much of the available radar time is devoted to surveillance activity.

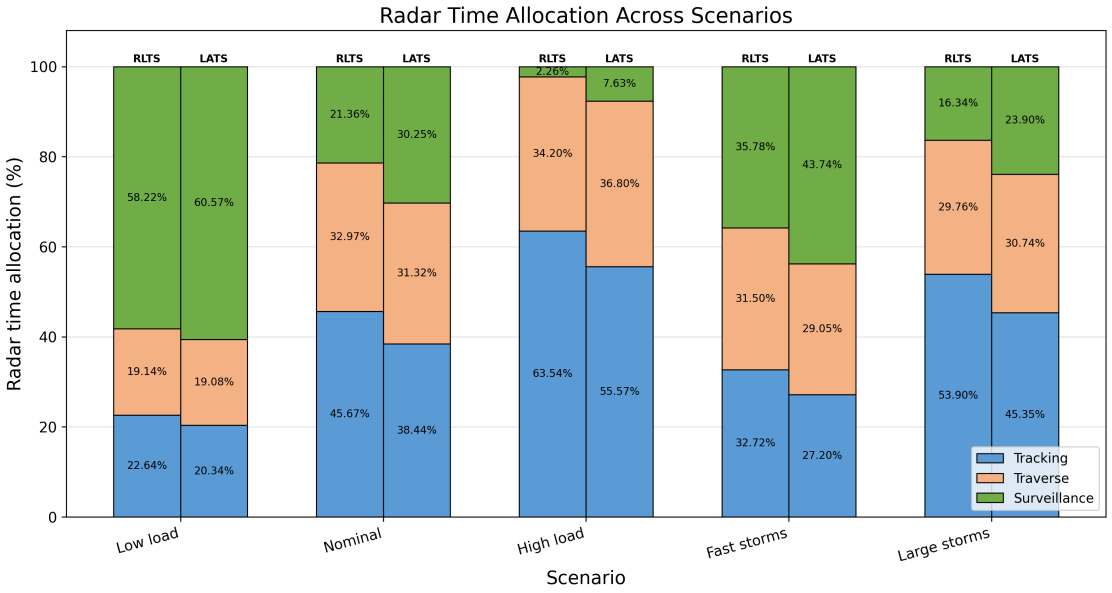


Figure 6.9: Mean fraction of available radar time allocated to tracking, azimuth traverse, and surveillance by RLTS and LATS across the five Monte Carlo scenarios.

Figure 6.9 separates the available radar time into tracking, traverse, and surveillance components. Under *low-load*, both schedulers devote the largest fraction of radar time to surveillance. RLTS allocates 22.64% of the available radar time to tracking, 19.14% to

traverse, and 58.22% to surveillance. LATS allocates 20.34% to tracking, 19.08% to traverse, and 60.57% to surveillance. The allocation patterns are therefore similar when radar demand is low. In the *nominal* scenario, the tracking demand increases considerably. RLTS allocates 45.67% of radar time to tracking and 32.97% to traverse, leaving 21.36% for surveillance. LATS allocates 38.44% to tracking and 31.32% to traverse, retaining 30.25% for surveillance.

The strongest resource constraint occurs under *high-load*. RLTS allocates 63.54% of radar time to tracking and 34.20% to traverse, leaving only 2.26% for surveillance. LATS allocates 55.57% to tracking and 36.80% to traverse, while retaining 7.63% for surveillance. The *high-load* scenario therefore represents the condition in which storm-related activities consume almost the complete radar-time budget. For fast storms, RLTS allocates 32.72% of radar time to tracking, 31.50% to traverse, and 35.78% to surveillance. The corresponding LATS values are 27.20%, 29.05%, and 43.74%. Compared with the *nominal* scenario, the *fast-storm* case therefore retains a larger surveillance fraction despite the increased storm motion. In the *large-storm* scenario, RLTS allocates 53.90% of radar time to tracking, 29.76% to traverse, and 16.34% to surveillance. LATS allocates 45.35% to tracking, 30.74% to traverse, and 23.90% to surveillance. The larger storm sectors therefore consume a substantial fraction of the available radar time through longer storm-sector scans.

The radar-time allocation shows that the main difference between RLTS and LATS lies in the balance between tracking and surveillance. Across all five scenarios, LATS uses a smaller fraction of the available radar time for tracking while retaining a larger fraction for surveillance. The traverse fractions remain comparatively similar between the two schedulers. This pattern is consistent with the load-aware assignment stage distributing storm tasks according to their scan cost, traverse cost, urgency, and the current workload of each radar.

This result is particularly important when considered together with the revisit-time results. In the *nominal*, *high-load*, *fast-storm*, and *large-storm* scenarios, LATS achieves shorter median revisit times than RLTS despite allocating a smaller fraction of radar time to storm tracking. The improvement is therefore not obtained by simply devoting more radar time to storm-sector scans. Instead, the load-aware assignment uses the available radar capacity more effectively, allowing storm revisits to be completed more frequently while preserving a larger fraction of the radar-time budget for background surveillance.

6.6 Planning-Window Sensitivity of LATS

The preceding results use a fixed planning-window length N_{plan} for the LATS scheduler. A sensitivity analysis was therefore performed using $N_{\text{plan}} = \{1, 2, 5, 10, 20, 40, 80\}$ dwells to determine whether the observed scheduling performance depends strongly on the assignment-update frequency.

Across the five scenarios, the revisit performance changes only modestly over the tested planning-window range. No single planning-window length consistently provides the best performance in every scenario. For example, the mean revisit rate remains within a narrow range across the tested values in the nominal, high-load, fast-storm, and large-storm cases.

The effective revisit-rate metric is particularly insensitive to the planning-window length, with the median remaining unchanged across the tested values in all five scenarios. This indicates that, within the range considered here, LATS is not strongly dependent on a specific planning-window setting.

The planning-window value of 10 dwells (approximately 0.90 s) used for the final Monte Carlo campaign was therefore retained as a practical operating choice. The sensitivity analysis shows only small differences across the tested planning-window values, with no single value providing the best performance in all scenarios. A 10-dwell window was retained to preserve frequent reassignment while providing strong performance under the high-load condition, where load-aware coordination is most important. The complete sensitivity results for all planning-window lengths and scenarios are provided in Appendix A.2.

6.7 Effect of Load-Aware Assignment Relative to RLTS

The preceding results allow the additional contribution of the load-aware assignment stage to be assessed directly. RLTS and LATS use the same time-balance formulation and radar-level task-selection principle, while LATS adds an explicit storm-to-radar assignment stage before individual task selection. Comparing the two methods therefore isolates the effect of distributing storm responsibility across the radar nodes on revisit performance, tracking accuracy, and retained surveillance capacity.

Table 6.3 provides a consolidated summary of this comparison across the five scenarios. The three performance dimensions are taken from the 50-trial Monte Carlo evaluations of revisit performance, tracking accuracy, and surveillance coverage presented in Sections 6.2, 6.4, and 6.5, respectively.

Table 6.3: Summary of LATS performance relative to RLTS using median revisit time, mean IoU, and mean surveillance coverage. The values are derived from the 50-trial Monte Carlo evaluations presented in Sections 6.2, 6.4, and 6.5.

Scenario	Revisit performance	Tracking accuracy	Surveillance coverage
Low load	2.6% shorter median revisit time	0.69 (RLTS) vs. 0.68 (LATS)	3.4% higher than RLTS
Nominal	6.6% shorter median revisit time	0.69 (RLTS) vs. 0.68 (LATS)	35.4% higher than RLTS
High load	12.9% shorter median revisit time	0.67 for both methods	+9.34 percentage points; 246.4% higher than RLTS
Fast storms	5.9% shorter median revisit time	0.58 for both methods	18.5% higher than RLTS
Large storms	8.4% shorter median revisit time	0.61 for both methods	45.3% higher than RLTS

The *low-load* scenario provides the clearest boundary case. With only two storms compet-

ing for two radars, the additional assignment stage provides little benefit: the revisit-time reduction is small, tracking accuracy remains similar, and surveillance coverage increases only slightly. This indicates that explicit workload distribution is of limited value when the available radar capacity is already sufficient for the small number of competing storm tasks.

The effect becomes more evident in the *nominal* scenario. LATS reduces the median revisit time while retaining substantially more surveillance coverage, without a meaningful change in tracking accuracy. The additional assignment stage therefore improves the balance between storm servicing and background surveillance once multiple storm tasks begin to compete for the same radar resources.

The strongest effect is observed under *high-load*, where six storms compete for two radars. LATS produces the largest reduction in median revisit time relative to RLTS while preserving considerably more surveillance coverage. The apparently large relative increase in surveillance coverage is partly caused by the very low RLTS reference value: coverage increases by 9.34 percentage points, corresponding to a 246.4% relative increase. Tracking accuracy remains essentially unchanged. Together with the radar-time allocation results, this shows that the improved revisit performance is not obtained by devoting more radar time to tracking, but by distributing the available workload more effectively between the two radar nodes.

The *fast-storm* scenario shows a more moderate benefit. LATS still reduces the median revisit time and retains more surveillance coverage, while tracking accuracy remains unchanged. The smaller difference between RLTS and LATS is consistent with the fact that faster storm motion primarily increases the difficulty of state prediction rather than the amount of radar-time demand that must be distributed between the nodes.

A stronger effect is again observed for *large storms*. The wider storm sectors increase the duration of individual tracking tasks and therefore make the allocation of radar workload more important. LATS reduces the median revisit time and retains substantially more surveillance coverage while maintaining essentially the same tracking accuracy as RLTS. This indicates that the load-aware assignment stage becomes increasingly useful when individual storm tasks consume a larger fraction of the available radar-time budget.

Overall, the principal contribution of load-aware assignment is not an improvement in tracking accuracy, but a more effective use of the available radar resources. LATS consistently produces shorter median revisit times and retains more surveillance coverage than RLTS, while mean IoU remains very similar between the two methods. The additional benefit is small when radar demand is low and becomes most evident when the radar-time budget is strongly constrained, either by a larger number of competing storms or by more time-consuming storm scans.

6.8 Summary of Main Findings

A 50-trial Monte Carlo campaign was used to compare RLTS, LATS, Baseline 1, and Baseline 2 across five scenarios using revisit performance, tracking accuracy, and surveillance coverage.

- **Revisit performance:** LATS achieves the shortest median revisit time in four of the five scenarios, with the largest advantage under *high-load* (24.01 s compared with 27.57 s for RLTS and 58.82 s for Baseline 1). Baseline 1 remains best only under *low-load*.
- **Tracking accuracy:** RLTS and LATS obtain very similar IoU values across all scenarios. The lowest accuracy occurs for *fast storms*, where both obtain a mean IoU of approximately 0.58.
- **Surveillance capacity:** LATS retains more surveillance coverage than RLTS in all five scenarios. Under *high-load*, coverage increases from 3.79% for RLTS to 13.13% for LATS while also achieving a shorter revisit time.

Overall, LATS provides the strongest balance between revisit performance and retained surveillance capacity without materially changing tracking accuracy. The planning-window sensitivity analysis shows that this conclusion is not strongly dependent on the selected planning-window value. These findings are interpreted further in Chapter 7.

Chapter 7

Discussion

The results show that the value of adaptive radar scheduling depends on the demand placed on the network. Under light loading, simple scheduling can already provide frequent storm updates, whereas increasing competition for radar time makes dynamic task selection and workload distribution more important.

The two proposed methods address this problem at different levels. RLTS introduces adaptive radar-level task selection based on task urgency, storm state, and radar-dependent geometry, while LATS additionally introduces an explicit storm-to-radar assignment stage. This chapter therefore discusses first when adaptive task selection becomes beneficial and then what additional value is provided by load-aware assignment. The interaction between scheduling, tracking quality, and surveillance is subsequently considered, followed by the broader implications and limitations of the study.

7.1 Adaptive Radar-Level Task Selection

This section considers when adaptive radar-level task selection provides an advantage over predefined scanning behaviour. Unlike the baseline methods, RLTS selects between currently feasible storm tasks according to their scheduling state. Time-Balance represents revisit urgency, while the priority and radar-dependent terms allow the scheduler to account for the current storm state and observation geometry. The relevant question is therefore not whether adaptive selection is always preferable, but under which operating conditions this additional flexibility becomes useful.

The *low-load* scenario provides the clearest boundary case. With only two storms requiring service, there is little competition between tasks and the fixed storm-by-storm cycle of Baseline 1 is already sufficient. Its median revisit time is 5.49 s shorter than that of RLTS. This shows that adaptive task selection does not provide an inherent advantage when the available radar capacity already exceeds the tracking demand.

The situation changes once several observations compete for the available radar time. Under the *nominal* and particularly the *high-load* conditions, RLTS can respond to differences in current revisit urgency instead of continuing through a predetermined task order. Under *high-load*, RLTS reduces the median revisit time by 31.25 s relative to Baseline 1,

showing that adaptive task ordering becomes increasingly valuable when several storm tasks are waiting simultaneously.

The *large-storm* and *fast-storm* scenarios further show that scenario difficulty can arise from different sources. Larger storms increase the duration of individual sector scans and therefore directly increase the demand on the available radar-time budget. Faster storm motion does not produce the same degree of radar-time saturation, but instead places greater demands on state prediction and tracking. Radar load should therefore not be interpreted only in terms of the number of storms; task duration and storm dynamics determine which part of the sensing and scheduling process becomes most constrained.

Key takeaway. The main value of RLTS is its ability to respond to a changing set of competing observation requests rather than committing the radar to a fixed scan sequence. This flexibility provides little additional benefit when radar capacity is abundant, but becomes increasingly useful when competition between tasks or longer scan durations make the ordering of observations an important scheduling decision.

7.2 Additional Role of Load-Aware Assignment

This section considers what additional benefit is obtained by introducing network-level storm-to-radar assignment beyond the adaptive task selection already provided by RLTS. RLTS determines which feasible task should be executed next at each radar, whereas LATS additionally determines how storm responsibility should be distributed between the radar nodes according to scan time, traverse time, radar workload, and Time-Balance urgency. The relevant question is therefore when this additional coordination improves the use of the available radar resources.

The *low-load* scenario again provides the boundary case. With only a small number of competing tasks, there is little workload imbalance for the assignment stage to correct, and LATS reduces the median revisit time by only 0.65 s relative to RLTS. This indicates that explicit network-level assignment provides limited additional value when both radars already have sufficient capacity to service the requested observations.

The benefit becomes more apparent as the radar-time budget becomes more constrained. Relative to RLTS, LATS reduces the median revisit time by 1.73 s under *nominal* conditions, with the largest reduction of 3.56 s occurring under *high-load*. A substantial reduction of 2.34 s is also obtained for *large storms*, where the longer sector scans increase the cost of individual tracking tasks. These results indicate that the assignment stage becomes most useful when either the number or the duration of competing tasks makes the distribution of workload between the radars an important scheduling decision.

The *fast-storm* scenario provides a useful contrast. The revisit-time reduction relative to RLTS is smaller than under *high-load* and *large-storm* conditions, because faster storm motion primarily increases the difficulty of state prediction rather than producing the same degree of radar-time competition. The benefit of load-aware assignment therefore depends

specifically on how strongly the available sensing resources are constrained, rather than on scenario difficulty in general.

The representative schedules provide a qualitative explanation for this behaviour. Under RLTS, storm responsibility can change as the instantaneous task-selection scores evolve, whereas LATS introduces a more persistent division of storm responsibility over the planning interval. This reduces the extent to which network-level workload distribution emerges only from successive local task-selection decisions and allows the two radar nodes to coordinate their available capacity more explicitly.

The radar-time-allocation results support this interpretation. Across the evaluated scenarios, LATS generally uses a smaller fraction of radar time for tracking while retaining more time for surveillance, despite achieving shorter median revisit times. The resource-allocation difference is most evident under *high-load* and *large-storm* conditions. Relative to RLTS, LATS reduces the tracking-time fraction by approximately 8.0 and 8.6 percentage points, respectively, while increasing the surveillance-time fraction by approximately 5.4 and 7.6 percentage points. The revisit improvement is therefore not obtained by allocating more radar time to storm tracking, but by organising the available tracking activity more effectively across the network.

Key takeaway. RLTS addresses the radar-level question of which observation should be performed next, while LATS additionally addresses the network-level question of which radar should be responsible for that observation. This additional coordination provides little benefit when radar capacity is abundant, but becomes increasingly valuable when competing tasks or longer scan durations make workload distribution a limiting factor. The principal contribution of LATS beyond RLTS is therefore a more effective distribution of radar resources, enabling shorter revisit intervals while preserving more capacity for background surveillance.

7.3 Scheduling, Tracking Quality, and Surveillance

This section considers how scheduling decisions influence not only storm revisit performance, but also tracking quality and the radar capacity remaining for background surveillance. Since the same tracking model is used for RLTS and LATS, the main question is whether the different scheduling decisions materially change tracking accuracy, and how the resulting resource allocation affects the balance between focused storm tracking and surveillance.

The tracking results show that RLTS and LATS achieve broadly similar mean IoU values across the evaluated scenarios. The differences between the two methods remain small, whereas larger changes are observed between the different storm conditions. The *low-load*, *nominal*, and *high-load* scenarios produce mean IoU values in a similar range, while tracking accuracy decreases more noticeably for *fast storms* and *large storms*.

The *fast-storm* scenario produces the lowest mean IoU, with both schedulers obtaining approximately 0.58. Faster storm motion increases the spatial displacement between succes-

sive measurements, making prediction of the storm state more difficult and increasing the possible mismatch between the estimated and true storm footprints.

The *large-storm* scenario also produces lower tracking accuracy, with both schedulers obtaining a mean IoU of approximately 0.61. In this case, the reduction arises from a different source: larger storm footprints require wider and longer sector scans, while partial observation of the larger storm region can also affect the estimated centre and radius. The fast- and large-storm cases therefore represent two distinct tracking challenges, associated with storm dynamics and storm extent, respectively.

The small IoU differences between RLTS and LATS indicate that the additional load-aware assignment stage does not materially change tracking accuracy in the evaluated campaign. The larger variations instead arise from the storm conditions and the resulting measurement sequence. The main benefit of LATS should therefore be interpreted primarily through revisit performance and resource use rather than through an improvement in the tracking model itself.

The surveillance results expose the other side of the resource-allocation problem. Radar time devoted to focused storm tracking cannot simultaneously be used for background surveillance, so surveillance capacity decreases as tracking demand becomes more severe. This is most evident under *high-load*, where several storms compete simultaneously for the available radar time. Large storms create a similar, although less severe, effect because individual storm-sector scans consume more of the radar-time budget. Under *low-load*, in contrast, substantial surveillance capacity remains because only a small number of storm tasks require service.

Key takeaway. The main scheduling trade-off is therefore between maintaining frequent storm updates and preserving radar capacity for background surveillance. LATS improves this balance by achieving shorter revisit intervals while retaining more surveillance capacity than RLTS, rather than improving revisit performance by sacrificing surveillance. However, the *high-load* results also show that scheduling cannot remove the underlying capacity constraint: once the total sensing demand approaches the available radar-time budget, surveillance is reduced for both methods. Load-aware coordination therefore improves how limited radar resources are used, but cannot eliminate the fundamental trade-off between tracking and surveillance.

7.4 Implications for Radar-Network Scheduling

This section considers the broader implications of the results for adaptive weather-radar-network scheduling. Time-Balance provides a natural mechanism for representing revisit urgency because the scheduling state of a waiting task evolves with the time elapsed since its previous update [12, 13]. The results of this work further show that effective network scheduling requires not only deciding which observation should be performed next, but also how those observations should be distributed between the available radar nodes.

The comparison between RLTS and LATS highlights these two levels of decision-making. RLTS addresses the temporal task-selection problem by responding to the current urgency, storm state, and radar geometry, whereas LATS additionally addresses the resource-assignment problem by coordinating which radar is responsible for each storm. The additional assignment stage provides little benefit when radar capacity is abundant, but becomes more important as the number or duration of competing tasks increases.

The evaluated scenarios also show that different forms of difficulty require different forms of adaptation. Increasing the number of storms or the size of the storm sectors directly increases competition for radar time, making network-level workload distribution more valuable. Increased storm speed, in contrast, has a stronger effect on tracking accuracy than on radar-time saturation. The required level of scheduling coordination should therefore depend not only on the number of requested observations, but also on the characteristics and time cost of those observations.

This suggests that a future radar network need not apply the same degree of coordination under every operating condition. A simpler scheduling strategy may be sufficient during periods of low sensing demand, whereas stronger network-level coordination may become beneficial as task competition or scan duration increases. Such dynamic switching between coordination levels was not investigated in this thesis, but follows naturally from the load-dependent behaviour observed across the evaluated scenarios.

Key takeaway. The broader implication is that adaptive radar-network scheduling should consider both *which task should be performed next* and *which radar should perform it*. The importance of the second decision increases as the network becomes resource constrained. The results therefore support a load-dependent view of coordination, in which the level of network-level scheduling effort is matched to the current sensing demand rather than being fixed for all operating conditions.

7.5 Limitations and Future Work

The findings should be interpreted within the controlled simulation framework used in this thesis. The framework was designed to isolate the scheduling problem and enable consistent comparison between the proposed methods, but several modelling and experimental assumptions limit how directly the results can be transferred to operational weather-radar networks. The main limitations and corresponding directions for future work are summarised below.

- **Radar-network configuration.** The evaluated network consists of two fixed radars with a prescribed geometry and coverage overlap. The importance of storm-to-radar assignment may change with the number of radar nodes, their separation, and the degree of overlapping coverage. Future work should therefore evaluate RLTS and LATS in larger and more heterogeneous radar networks.
- **Simulation duration and scenario scope.** Each scenario was evaluated using 50 Monte Carlo trials of 20 min operation. This captures repeated tracking, assignment,

and surveillance cycles, but does not represent longer-term weather evolution or transitions between different load conditions. Longer simulations with dynamically changing storm fields would provide a stronger assessment of scheduler behaviour under operationally varying demand.

- **Radar and storm modelling.** Storm evolution is represented by a simplified parametric model, while radar sensing is limited to two-dimensional azimuthal scanning. Operational systems additionally involve elevation scanning, three-dimensional storm structure, attenuation, clutter, and waveform constraints, all of which can change task duration and feasibility. Evaluation using higher-fidelity simulations or measured weather data is therefore an important next step.

- **Tracking model.** The same Kalman-filter-based storm-state model is used throughout the study. The reduced IoU observed for *fast storms* indicates that faster storm dynamics are more difficult to represent with the assumed motion model, while *large storms* introduce additional uncertainty in estimating the complete storm footprint. More flexible motion models, adaptive process-noise estimation, or alternative tracking approaches could therefore be investigated.

- **LATS parameterisation.** The assignment cost combines scan time, traverse time, radar workload, and Time-Balance urgency using fixed weights. Their individual contributions were not isolated, so an ablation or weight-sensitivity study would be required to determine whether different combinations are preferable under different operating conditions. The planning-window sensitivity analysis showed only modest variation over $N_{\text{plan}} = \{1, 2, 5, 10, 20, 40, 80\}$ dwells, with no single value performing best in every scenario. The 10-dwell value used in the final campaign should therefore be regarded as a practical operating choice rather than a uniquely optimal setting.

Key takeaway. The present results establish the scheduling behaviour within a controlled two-radar environment rather than providing a complete operational validation. The most important next step is therefore to test the proposed methods under larger and more realistic radar networks with dynamically changing weather conditions. A further extension would be to make the assignment weights and level of network coordination adaptive to the current sensing demand, so that the scheduler can use simpler coordination under light loading and stronger network-level coordination when radar resources become constrained.

Chapter 8

Conclusion

This thesis investigated adaptive task scheduling for a network of weather radars that must simultaneously track evolving storm cells and retain background surveillance. The central challenge is the limited radar time available when several storm observations compete for the same sensing resources. To address this problem, two scheduling approaches were developed and evaluated: Radar-Level Task Selection (RLTS) and Load-Aware Task Scheduling (LATS).

RLTS introduces adaptive radar-level task selection by combining Time-Balance revisit urgency with dynamic storm priority, task feasibility, and radar-dependent geometry. Rather than following a fixed scan sequence, the radar can therefore select the most appropriate feasible storm task according to the current network state. LATS extends this framework by adding an explicit storm-to-radar assignment stage that accounts for scan time, traverse time, radar workload, and Time-Balance urgency before radar-level task selection is performed.

The proposed methods were compared with a storm-by-storm cyclic tracking baseline and a continuous 360° *surveillance baseline across five simulation scenarios: low load, nominal, high load, fast storms, and large storms. The final Monte Carlo campaign consisted of 50 trials per scenario, with each trial representing 20 min of simulated operation. The evaluation considered revisit performance, storm-tracking accuracy, surveillance coverage, and the distribution of radar time between tracking, traverse, and surveillance.*

8.1 Main Findings

The results show that the value of adaptive scheduling depends on the demand placed on the radar network. In the *low-load* scenario, where only two storms compete for two radars, the storm-by-storm baseline remains highly effective and achieves the shortest median revisit time. This represents an important boundary condition: when sufficient radar capacity is available, a more complex scheduling strategy does not necessarily provide better revisit performance than a simple cyclic approach.

As the radar workload increases, the advantage of adaptive scheduling becomes clear. RLTS provides shorter median revisit times than both baseline methods in the nominal, high-load, fast-storm, and *large-storm* scenarios. The strongest separation from the base-

lines occurs under *high-load*, where several storm tasks compete simultaneously for the two available radar nodes. The *large-storm* scenario further shows that scheduling demand is affected not only by the number of storms but also by the duration of the required tracking tasks.

The main findings of the study can be summarised as follows:

- **Adaptive task selection becomes valuable when radar resources are constrained.** RLTS allows storm observations to be selected according to their current urgency and scheduling state rather than following a fixed scan sequence. Its benefit is limited under *low-load*, but becomes clear once several observations compete for the available radar time.
- **Load-aware assignment provides an additional resource-management benefit.** LATS achieves shorter median revisit times than RLTS in all five scenarios, with the largest reduction of 3.56 s under *high-load*. At the same time, it preserves more surveillance capacity while generally using a smaller fraction of radar time for tracking, showing that its advantage comes from a more effective distribution of tasks across the radar network.
- **Tracking accuracy is influenced more strongly by storm characteristics than by the scheduling method.** Mean IoU remains similar between RLTS and LATS, while the largest reduction occurs for *fast storms*, where both methods achieve approximately 0.58. The principal scheduling gains should therefore be interpreted in terms of revisit performance and resource allocation rather than improved tracking accuracy.
- **LATS does not depend strongly on a narrowly tuned planning window.** Revisit performance changes only modestly across the tested planning-window lengths, and no single value performs best in every scenario. The selected planning window should therefore be regarded as a practical operating choice rather than a uniquely optimal parameter.

Taken together, these findings show that adaptive weather-radar-network scheduling involves two complementary decisions: selecting which observation should be performed next and determining how the observation workload should be distributed across the available radar nodes. RLTS addresses the first through adaptive radar-level task selection, while LATS adds explicit network-level workload coordination. The main contribution of this thesis is therefore to demonstrate that combining these two levels of scheduling can reduce storm revisit times and preserve more surveillance capacity when radar resources become constrained, without materially changing tracking accuracy.

8.2 Answers to the Research Questions

The findings of the thesis can be summarised with respect to the research questions introduced in Chapter 1.

RQ1: How can revisit urgency and changing storm characteristics be incorporated into adaptive radar-level task selection?

RLTS addresses this question by combining Time-Balance revisit urgency with dynamic storm-priority information and radar-dependent task feasibility. Time-Balance represents the urgency created by the time elapsed since the previous completed storm update, while storm and radar information are used to distinguish between multiple feasible observations. This allows the radar schedule to respond dynamically to the evolving task state instead of following a predetermined sequence.

The simulation results show that this approach becomes particularly useful when several observations compete for limited radar time. While the storm-by-storm baseline remains effective under *low-load*, RLTS provides shorter revisit times than both baseline methods in the nominal, high-load, fast-storm, and *large-storm* scenarios.

RQ2: How can storm-tracking tasks be distributed across multiple radars while accounting for the workload of the individual radar nodes?

LATS addresses the network-level assignment problem by evaluating the cost of assigning each storm to each radar using expected scan time, traverse time, current radar workload, and storm urgency. This assignment is performed before radar-level task selection, allowing the distribution of storm responsibilities to be considered explicitly rather than emerging only from successive local scheduling decisions.

The comparison with RLTS shows that this additional coordination provides the greatest benefit when the available radar resources are strongly constrained. The largest reduction in median revisit time relative to RLTS occurs under *high-load*, while a clear benefit is also observed for large storms. LATS additionally retains more surveillance capacity while requiring a smaller fraction of the total radar time for storm tracking.

RQ3: How do the proposed scheduling approaches affect revisit performance, storm tracking, and retained surveillance?

The results show that both proposed methods provide clear revisit benefits over the baseline approaches once meaningful competition for radar resources is present. RLTS establishes the benefit of adaptive radar-level task selection, while LATS provides an additional improvement through load-aware assignment.

Tracking accuracy remains broadly comparable between RLTS and LATS, with the main variation occurring between the different storm scenarios rather than between the two proposed schedulers. Fast storms and large storms produce the lowest mean IoU values and therefore represent the most challenging tracking conditions in the evaluated campaign.

The clearest additional benefit of LATS is observed in resource utilisation. LATS retains more surveillance coverage than RLTS across all five scenarios while also achieving shorter median revisit times. The *low-load* scenario remains an important qualification, since the

simple storm-by-storm baseline can still provide the shortest revisit time when sufficient sensing capacity is available.

8.3 Final Remarks

The conclusions of this thesis are based on a controlled two-radar, two-dimensional simulation environment and should therefore be interpreted within that scope. Although the final evaluation uses 20 min simulations and 50 Monte Carlo trials per scenario, longer operational periods, larger radar networks, more complex storm evolution, three-dimensional scanning, and higher-fidelity radar measurements would provide important extensions of the present work.

Nevertheless, the results establish a clear progression. Fixed storm-by-storm scheduling is sufficient when radar demand is low. As competition for radar time increases, adaptive task selection through RLTS becomes beneficial. When the workload must also be coordinated between multiple radar nodes, the assignment mechanism introduced by LATS provides additional benefit, particularly when the available radar resources are strongly constrained.

The study therefore shows that adaptive weather-radar scheduling can make more effective use of limited network sensing resources by balancing frequent storm updates with continued surveillance. RLTS provides the adaptive task-selection mechanism, while LATS extends this capability through network-level workload coordination. These results provide a basis for further investigation of coordinated scheduling in larger and more realistic weather-radar networks.

Bibliography

[1] R. J. Doviak and D. S. Zrnic, *Doppler radar & weather observations*. Academic press, 2014.

[2] M. Steiner, R. A. Houze Jr, and S. E. Yuter, "Climatological characterization of three-dimensional storm structure from operational radar and rain gauge data," *Journal of Applied Meteorology and Climatology*, vol. 34, no. 9, pp. 1978–2007, 1995.

[3] R. J. Doviak, D. S. Zrnic, and D. S. Sirmans, "Doppler weather radar," *Proceedings of the IEEE*, vol. 67, no. 11, pp. 1522–1553, 1979.

[4] P. Heinselman, D. LaDue, D. M. Kingfield, and R. Hoffman, "Tornado warning decisions using phased-array radar data," *Weather and Forecasting*, vol. 30, no. 1, pp. 57–78, 2015.

[5] P. L. Heinselman, D. L. Priegnitz, K. L. Manross, T. M. Smith, and R. W. Adams, "Rapid sampling of severe storms by the national weather radar testbed phased array radar," *Weather and Forecasting*, vol. 23, no. 5, pp. 808–824, 2008.

[6] K. A. Wilson, P. L. Heinselman, C. M. Kuster, D. M. Kingfield, and Z. Kang, "Forecaster performance and workload: Does radar update time matter?" *Weather and Forecasting*, vol. 32, no. 1, pp. 253–274, 2017.

[7] R. A. Brown, V. T. Wood, and D. Sirmans, "Improved WSR-88D scanning strategies for convective storms," *Weather and Forecasting*, vol. 15, no. 2, pp. 208–220, 2000.

[8] S. M. Torres, R. Adams, C. D. Curtis, E. Forren, D. E. Forsyth, I. R. Ivić, D. Priegnitz, J. Thompson, and D. A. Warde, "Adaptive-weather-surveillance and multifunction capabilities of the national weather radar testbed phased array radar," *Proceedings of the IEEE*, vol. 104, no. 3, pp. 660–672, 2016.

[9] M. Dixon and G. Wiener, "Titan: Thunderstorm identification, tracking, analysis, and nowcasting—a radar-based methodology," *Journal of atmospheric and oceanic technology*, vol. 10, no. 6, pp. 785–797, 1993.

[10] J. Johnson, P. L. MacKeen, A. Witt, E. D. W. Mitchell, G. J. Stumpf, M. D. Eilts, and K. W. Thomas, "The storm cell identification and tracking algorithm: An enhanced wsr-88d algorithm," *Weather and forecasting*, vol. 13, no. 2, pp. 263–276, 1998.

[11] J. M. Butler, "Tracking and control in multi-function radar," Ph.D. dissertation, University College London, University of London, London, United Kingdom, August 1998.

[12] R. Reinoso-Rondinel, T.-Y. Yu, and S. M. Torres, "Multifunction phased-array radar: Time balance scheduler for adaptive weather sensing," *Journal of Atmospheric and Oceanic Technology*, vol. 27, no. 11, pp. 1854–1867, 2010.

[13] R. Reinoso-Rondinel, S. M. Torres, and T.-Y. Yu, "Task prioritization on phased-array radar scheduler for adaptive weather sensing," in *26th Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology.* American Meteorological Society, 2010, paper 14B.6.

[14] F. Junyent and V. Chandrasekar, "Theory and characterization of weather radar networks," *Journal of Atmospheric and Oceanic Technology*, vol. 26, no. 3, pp. 474–491, 2009.

[15] D. Zrnic, J. Kimpel, D. Forsyth, A. Shapiro, G. Crain, R. Ferek, J. Heimmer, W. Benner, F. T. McNellis, and R. Vogt, "Agile-beam phased array radar for weather observations," *Bulletin of the American Meteorological Society*, vol. 88, no. 11, pp. 1753–1766, 2007.

[16] S. Miranda, C. Baker, K. Woodbridge, and H. Griffiths, "Knowledge-based resource management for multifunction radar: a look at scheduling and task prioritization," *IEEE Signal Processing Magazine*, vol. 23, no. 1, pp. 66–76, 2006.

[17] M. Zink, D. Westbrook, S. Abdallah, B. Horling, V. Lakamraju, E. Lyons, V. Manfredi, J. Kurose, and K. Hondl, "Meteorological command and control: An end-to-end architecture for a hazardous weather detection sensor network," in *Proceedings of the 2005 workshop on End-to-end, sense-and-respond systems, applications and services*, 2005, pp. 37–42.

[18] F. Junyent, V. Chandrasekar, D. McLaughlin, E. Insanic, and N. Bharadwaj, "The casa integrated project 1 networked radar system," *Journal of atmospheric and oceanic technology*, vol. 27, no. 1, pp. 61–78, 2010.

[19] Y. Wang, V. Chandrasekar, and B. Dolan, "Development of scan strategy for dual-doppler retrieval in a networked radar system," in *2008 IEEE International Geoscience and Remote Sensing Symposium*, 2008, pp. V–322–V–325.

[20] C. D. Curtis and S. M. Torres, "Adaptive range oversampling to achieve faster scanning on the national weather radar testbed phased-array radar," *Journal of Atmospheric and Oceanic Technology*, vol. 28, no. 12, pp. 1581–1597, 2011.

[21] A. Antonini, S. Melani, M. Corongiu, S. Romanelli, A. Mazza, A. Ortolani, and B. Gozzini, "On the implementation of a regional X-band weather radar network," *Atmosphere*, vol. 8, no. 2, p. 25, 2017.

[22] A. Charlish, K. Woodbridge, and H. Griffiths, "Phased array radar resource management using continuous double auction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 2212–2224, 2015.

[23] P. W. Moo and Z. Ding, "Coordinated radar resource management for networked phased array radars," *IET Radar, Sonar & Navigation*, vol. 9, no. 8, pp. 1009–1020, 2015.

[24] M. I. Schöpe, H. Driessen, and A. Yarovoy, "Optimal balancing of multi-function radar budget for multi-target tracking using lagrangian relaxation," in *2019 22nd International Conference on Information Fusion (FUSION)*, Ottawa, Canada, July 2019.

[25] J. R. Guerci, R. Guerci, M. Rangaswamy, J. Bergin, and M. Wicks, "CoFAR: Cognitive fully adaptive radar," in *2014 IEEE Radar Conference*, 2014, pp. 984–989.

[26] F. Smits, A. Huizing, W. van Rossum, and P. Hiemstra, "A cognitive radar network: Architecture and application to multiplatform radar management," in *2008 European Radar Conference*, 2008, pp. 312–315.

[27] A. Charlish, F. Hoffmann, C. Degen, and I. Schlangen, "The development from adaptive to cognitive radar resource management," *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, no. 6, pp. 8–19, 2020.

Appendix A

Additional Results

A.1 Representative Trials for the Remaining Scenarios

The representative *nominal* trial presented in Chapter 6 illustrates the detailed operation of RLTS and LATS under the main reference scenario. For completeness, this section provides additional representative examples from the remaining scenarios: *low-load*, *high-load*, *fast-storm*, and *large-storm* conditions.

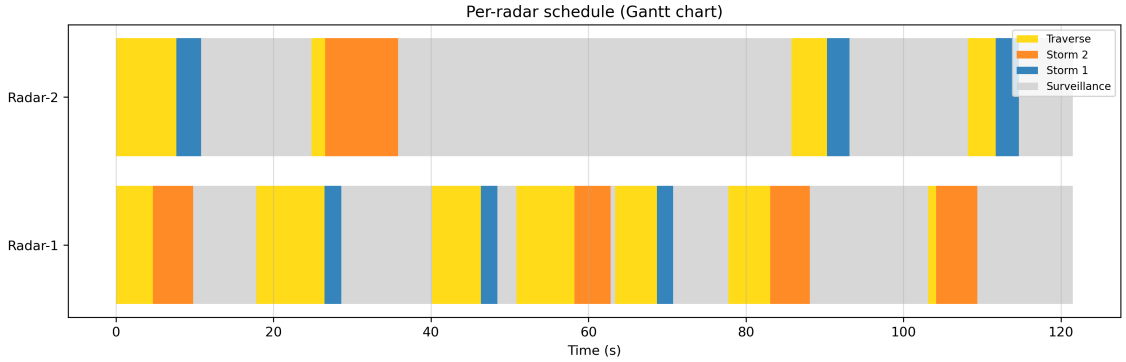
The examples presented here are selected from the short-duration simulation campaign and show 2 min of scheduler operation. They are intended to provide qualitative illustrations of how the scheduling behaviour changes between scenario types. The quantitative conclusions of the thesis remain based on the complete 50-trial, 20 min Monte Carlo campaign presented in Chapter 6.

A.1.1 Low-Load Representative Trial

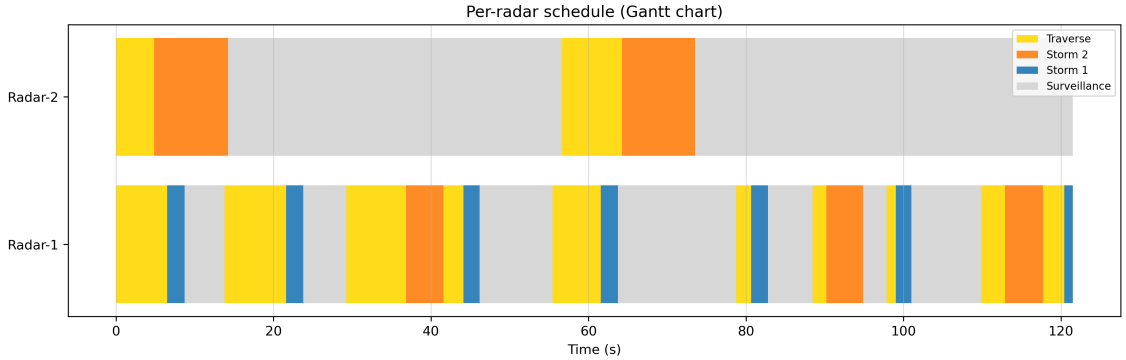
The *low-load* scenario contains only two storms and therefore represents a condition in which the available radar capacity is relatively abundant. Figure A.1 compares the schedules produced by RLTS and LATS for the same representative storm configuration.

Both schedules contain substantial periods of background surveillance, illustrating that the two-storm configuration does not strongly constrain the available radar-time budget. Under RLTS, responsibility for the two storms can change between the radar nodes as the task-selection state evolves. LATS produces a more persistent division of storm responsibility, consistent with its additional storm-to-radar assignment stage. However, because the tracking demand is already low, this additional coordination does not provide the same benefit observed under more strongly constrained scenarios.

Figure A.2 shows the corresponding window-wise improvement factors relative to Baseline 1 and Baseline 2.



(a) RLTS.



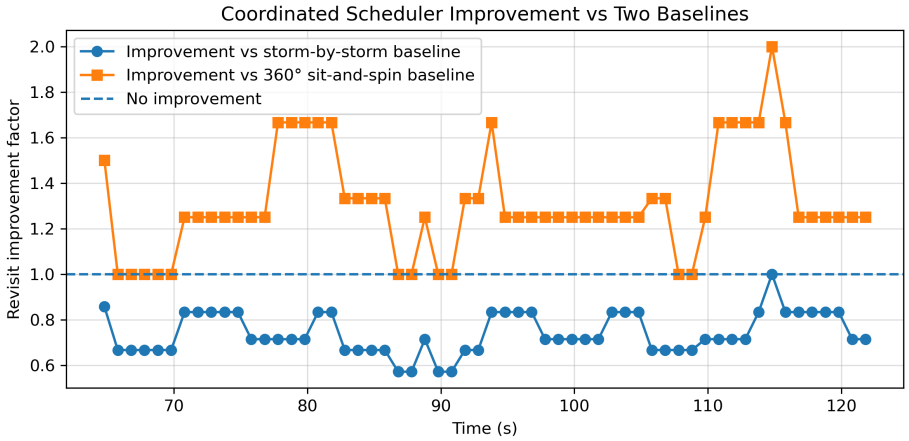
(b) LATS.

Figure A.1: Per-radar schedules for RLTS and LATS during the representative 2 min *low-load* trial.

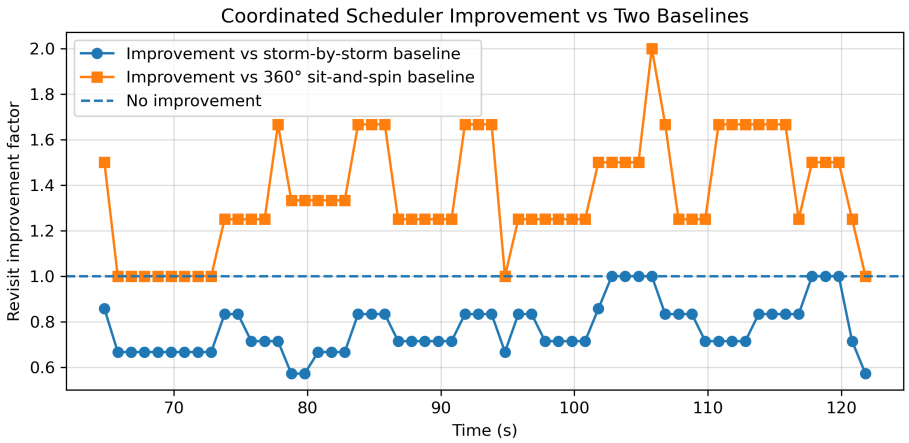
For both adaptive methods, the comparison with Baseline 1 remains largely at or below the equal-performance level during the displayed interval. In contrast, the comparison with Baseline 2 is generally at or above unity. This behaviour is consistent with the scenario-level results: when only two storms require service, the fixed storm-by-storm strategy already provides frequent updates and leaves limited opportunity for adaptive scheduling to improve revisit performance.

Tracking behaviour for the same representative trial is shown in Figure A.3.Both schedulers maintain high IoU values during the representative trial, with no clear difference in tracking quality between RLTS and LATS. This is consistent with the broader Monte Carlo results, where tracking accuracy under *low-load* remains similar for the two adaptive methods.

Representative-trial observation. The low-load example illustrates the boundary condition identified in the main results. Both adaptive schedulers have sufficient radar capacity to service the small number of storm tasks while retaining substantial surveillance activity. LATS introduces a more structured distribution of storm responsibility, but the additional coordination provides only limited revisit benefit when competition for radar resources is low.

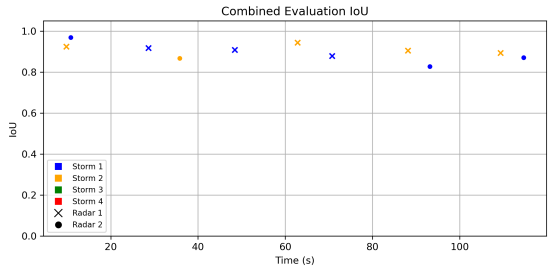


(a) RLTS.

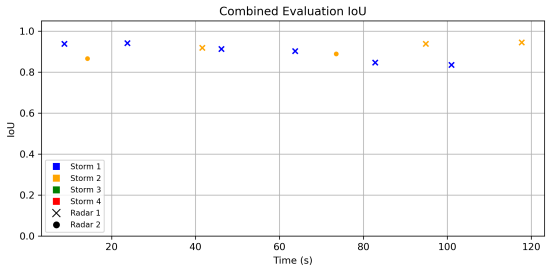


(b) LATS.

Figure A.2: Window-wise revisit improvement of RLTS and LATS relative to Baseline 1 and Baseline 2 during the representative *low-load* trial. The horizontal line at $I = 1$ indicates equal revisit-rate performance with the corresponding baseline.



(a) RLTS.

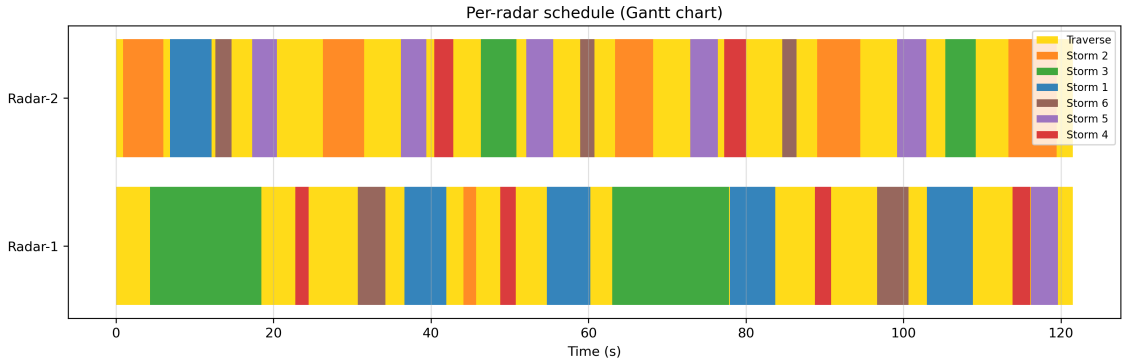


(b) LATS.

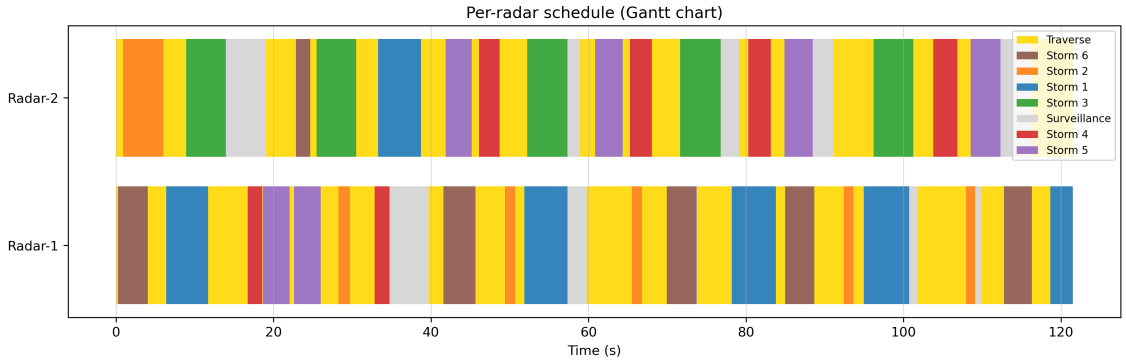
Figure A.3: IoU at completed storm updates for RLTS and LATS during the representative *low-load* trial.

A.1.2 High-Load Representative Trial

The *high-load* scenario contains six storms competing for service from two radars and therefore represents the most strongly resource-constrained condition considered in this thesis. Figure A.4 compares the schedules produced by RLTS and LATS for the same representative storm configuration.



(a) RLTS.

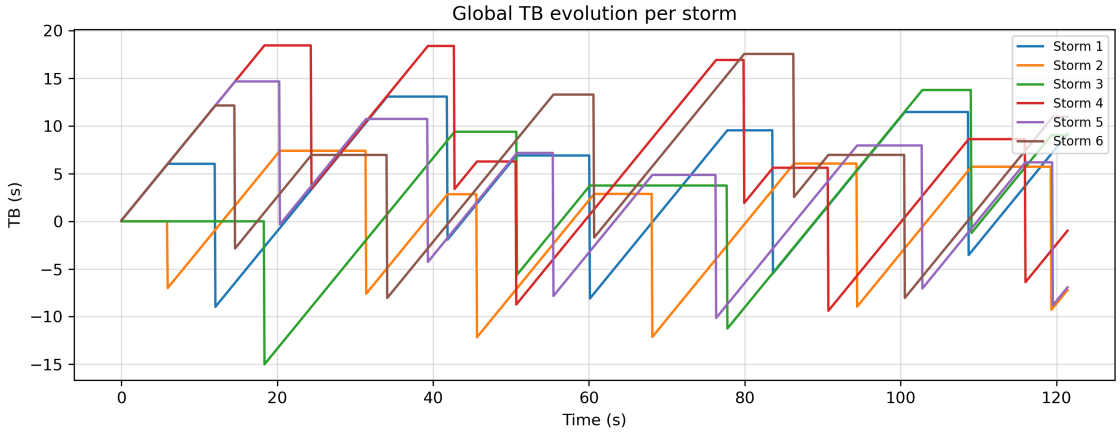


(b) LATS.

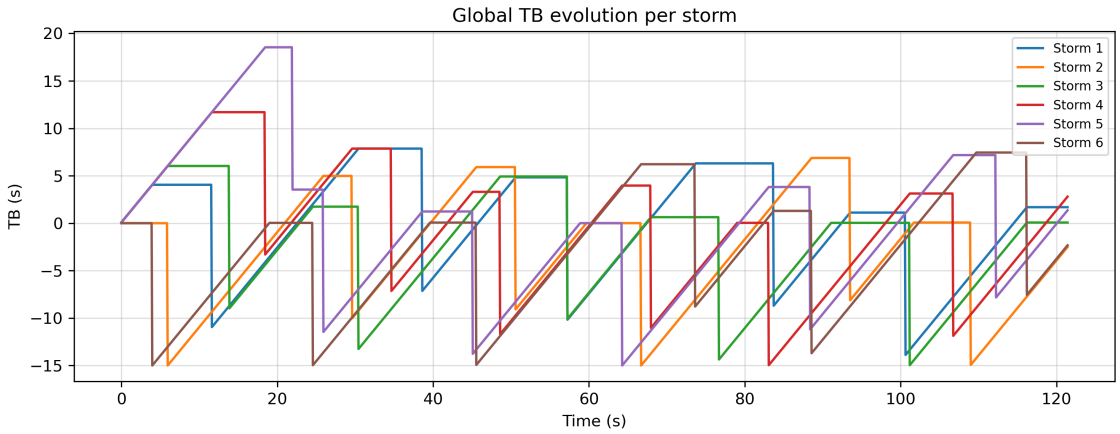
Figure A.4: Per-radar schedules for RLTS and LATS during the representative 2 min *high-load* trial.

The schedules show substantially greater competition for radar time than in the *low-load* case. Both radars repeatedly alternate between tracking different storms and traversing between storm sectors, leaving comparatively little time for background surveillance. The LATS schedule retains some visible surveillance intervals while coordinating the distribution of the six storms between the two radar nodes. This is consistent with the scenario-level result that load-aware assignment becomes more useful when a large number of storm tasks compete for the available radar-time budget.

Figure A.5 shows the corresponding global Time-Balance evolution of the six storms. Positive Time-Balance values indicate storms that are due or overdue for an update, while the reductions occur following completed storm observations.



(a) RLTS.



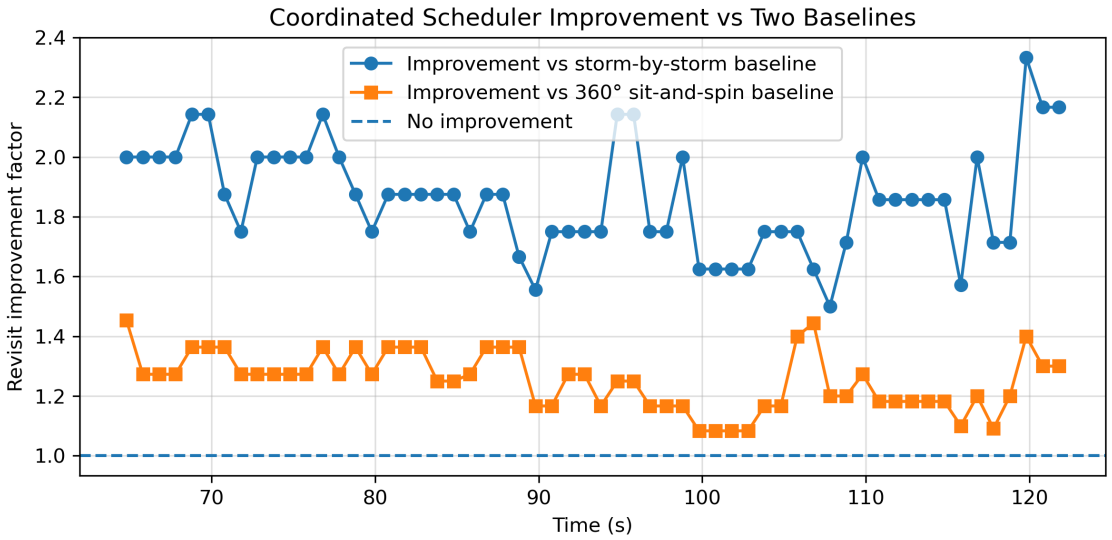
(b) LATS.

Figure A.5: Global Time-Balance evolution for RLTS and LATS during the representative *high-load* trial.

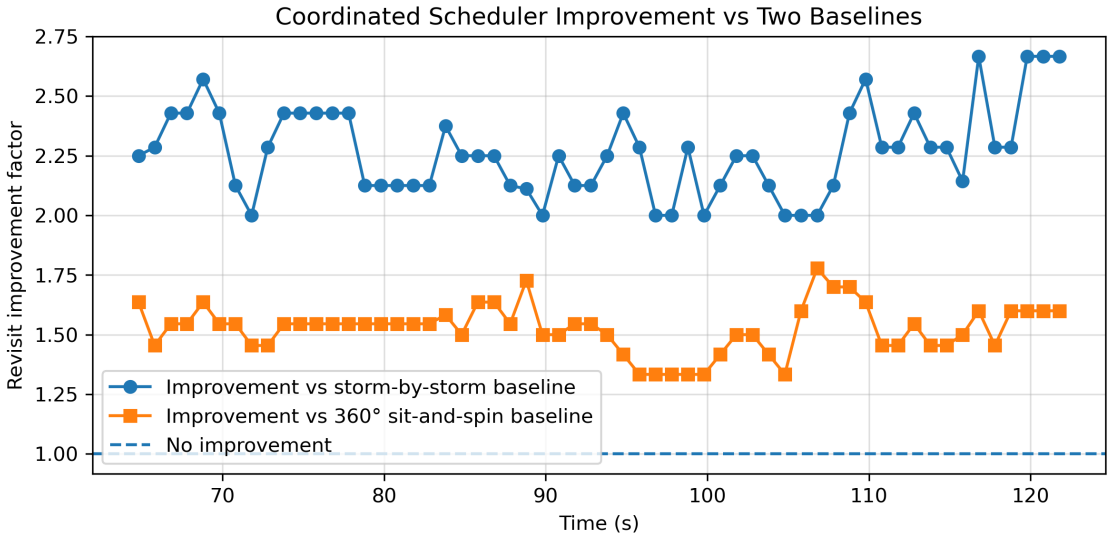
The Time-Balance histories illustrate the persistent competition between storm observations in this scenario. Under RLTS, several storms develop relatively large positive Time-Balance excursions before being serviced. LATS also experiences overdue tasks, particularly during the beginning of the trial, but the later Time-Balance excursions are generally more contained. This behaviour is consistent with the assignment stage distributing storm responsibility before radar-level task selection is performed.

The resulting revisit performance is shown in Figure A.6. The improvement factor is calculated relative to both Baseline 1 and Baseline 2, with $I = 1$ representing equal revisit-rate performance.

Both adaptive schedulers remain above the equal-performance level relative to the two baselines throughout the displayed interval. The improvement relative to Baseline 1 is particularly pronounced, reflecting the limitation of using only one radar for cyclic storm tracking when six storms require service. LATS maintains a larger improvement than RLTS relative to both baselines in this representative trial, illustrating the additional value of explicitly coordinating the storm workload under heavy radar loading.



(a) RLTS.



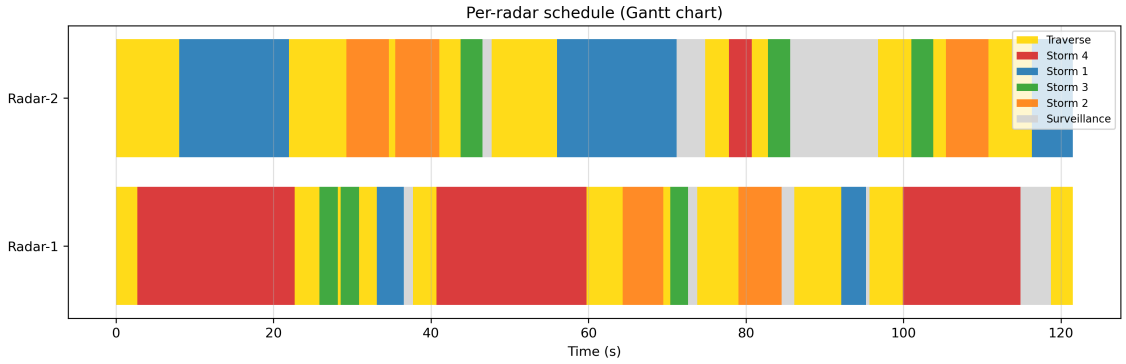
(b) LATS.

Figure A.6: Window-wise revisit improvement of RLTS and LATS relative to Baseline 1 and Baseline 2 during the representative *high-load* trial. The horizontal line at $I = 1$ indicates equal revisit-rate performance with the corresponding baseline.

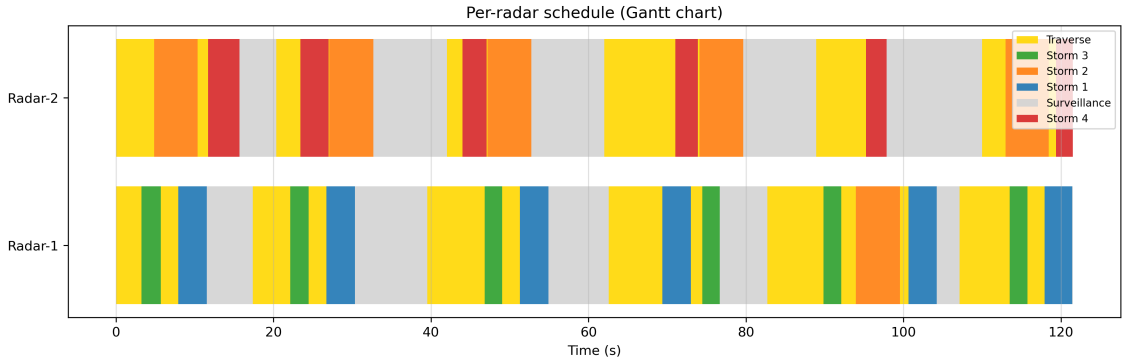
Representative-trial observation. The high-load example illustrates the operating condition in which network-level coordination becomes most valuable. The large number of competing observations produces substantial Time-Balance excursions and consumes most of the available radar capacity. RLTS benefits from adaptive task selection, while LATS additionally coordinates how this workload is distributed between the radar nodes, resulting in a clearer revisit advantage and retaining some capacity for surveillance.

A.1.3 Fast-Storm Representative Trial

The *fast-storm* scenario differs from the previous cases because the main additional challenge arises from faster storm motion rather than from a large increase in radar-time demand. Figure A.7 compares the schedules produced by RLTS and LATS for the same representative storm configuration.



(a) RLTS.

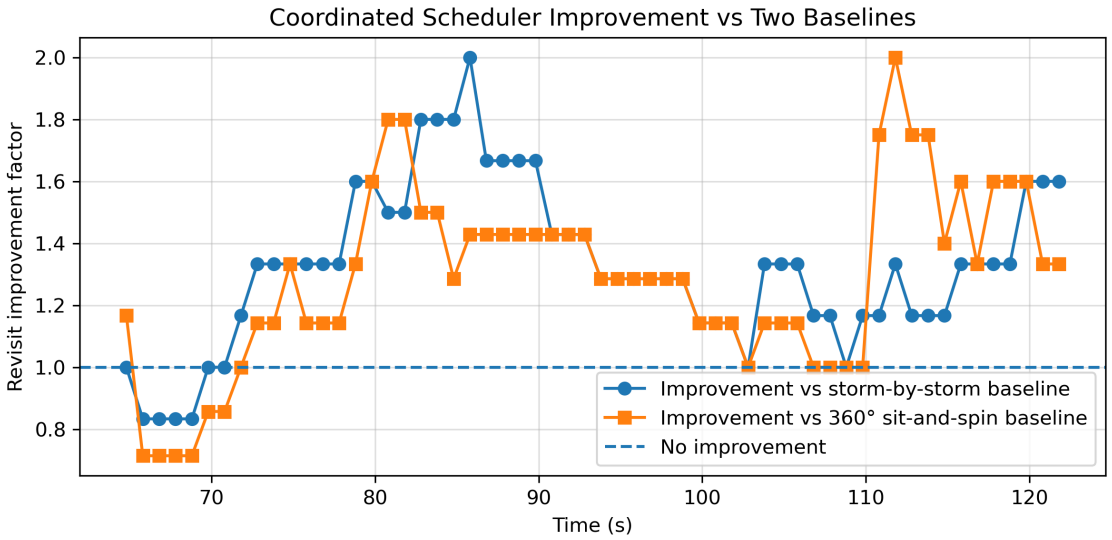


(b) LATS.

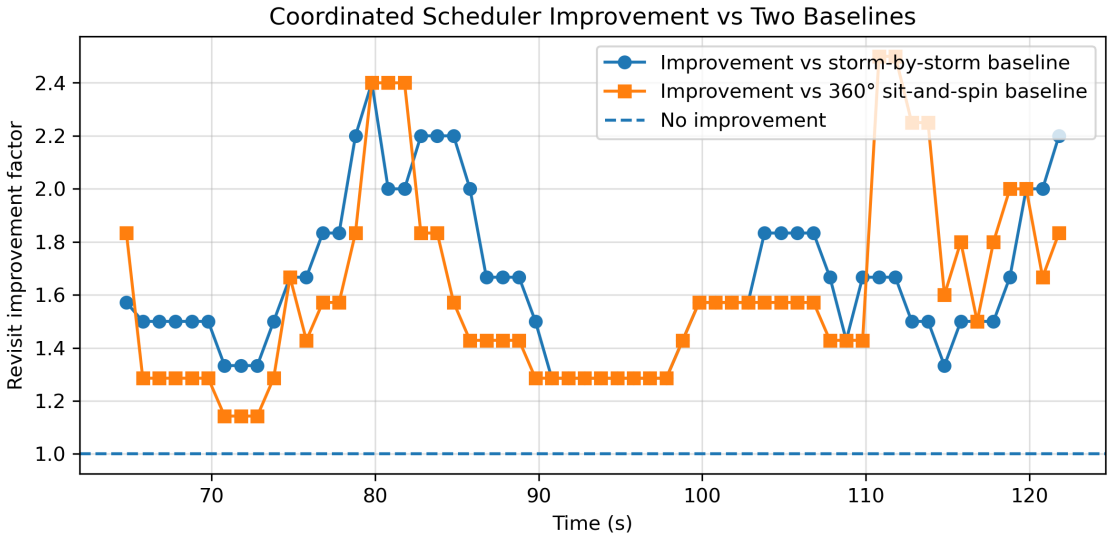
Figure A.7: Per-radar schedules for RLTS and LATS during the representative 2 min *fast-storm* trial.

Both schedulers alternate between storm tracking, traverse, and surveillance throughout the trial. RLTS shows more frequent changes in storm responsibility, whereas LATS produces a more structured distribution of observations between the two radar nodes. Unlike the *high-load* case, however, substantial surveillance intervals remain visible, illustrating that faster storm motion does not produce the same degree of radar-time saturation as an increase in the number of competing storms.

The corresponding revisit improvement factors are shown in Figure A.8. The horizontal line at $I = 1$ represents equal revisit-rate performance with the corresponding baseline.



(a) RLTS.



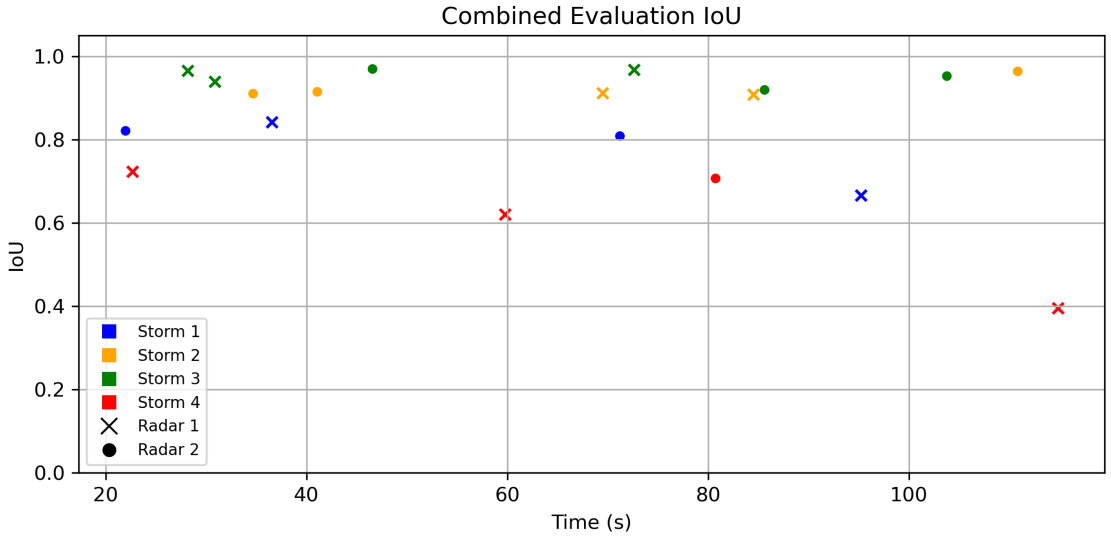
(b) LATS.

Figure A.8: Window-wise revisit improvement of RLTS and LATS relative to Baseline 1 and Baseline 2 during the representative *fast-storm* trial. The horizontal line at $I = 1$ indicates equal revisit-rate performance with the corresponding baseline.

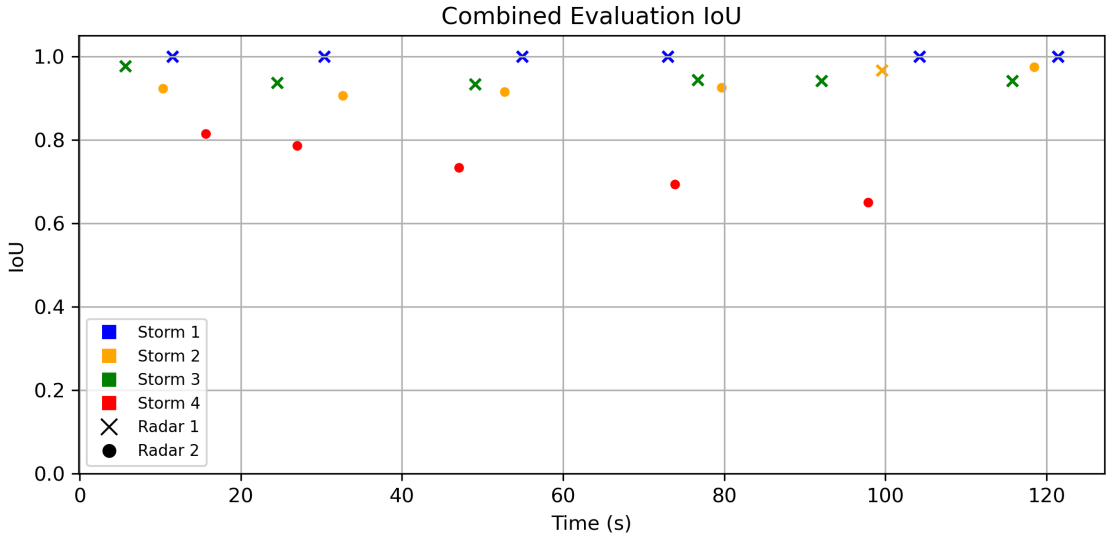
After the initial part of the displayed interval, both adaptive schedulers generally remain above the equal-performance level relative to the two baselines. LATS shows a somewhat stronger improvement in this representative trial, although the difference between RLTS and LATS is less pronounced than in the *high-load* case. This is consistent with the scenario-level results, where the additional benefit of load-aware assignment is smaller when the dominant challenge is faster storm motion rather than strong competition for radar time.

Tracking behaviour for the same representative trial is shown in Figure A.9. The IoU values vary between completed updates and include several noticeably lower values, particularly for some storm-radar combinations. This illustrates the greater tracking difficulty

introduced by faster storm motion.



(a) RLTS.



(b) LATS.

Figure A.9: IoU at completed storm updates for RLTS and LATS during the representative *fast-storm* trial.

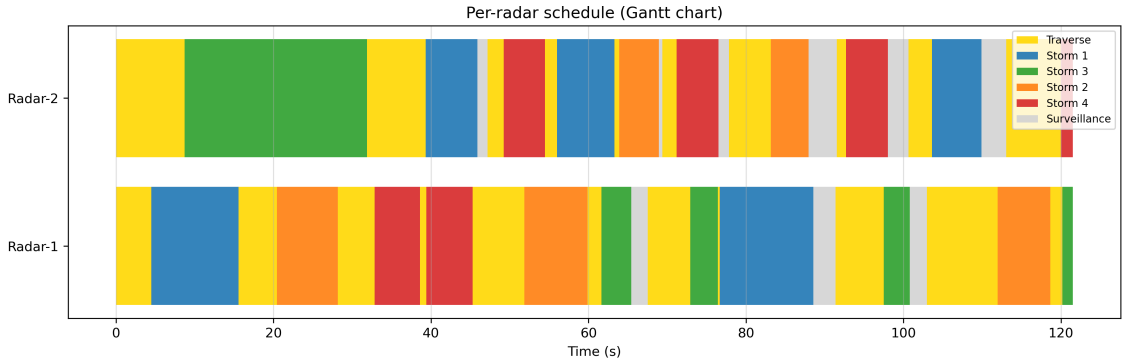
The representative IoU results show that the tracking quality can vary considerably between individual updates even though the schedulers continue to provide frequent storm observations. This supports the scenario-level finding that the reduced tracking accuracy in the *fast-storm* case is associated primarily with the more rapid storm dynamics rather than with a large difference between RLTS and LATS.

Representative-trial observation. The fast-storm example illustrates that scenario difficulty does not arise only from competition for radar resources. Both adaptive schedulers retain revisit advantages over the baseline methods and substantial opportunities for surveil-

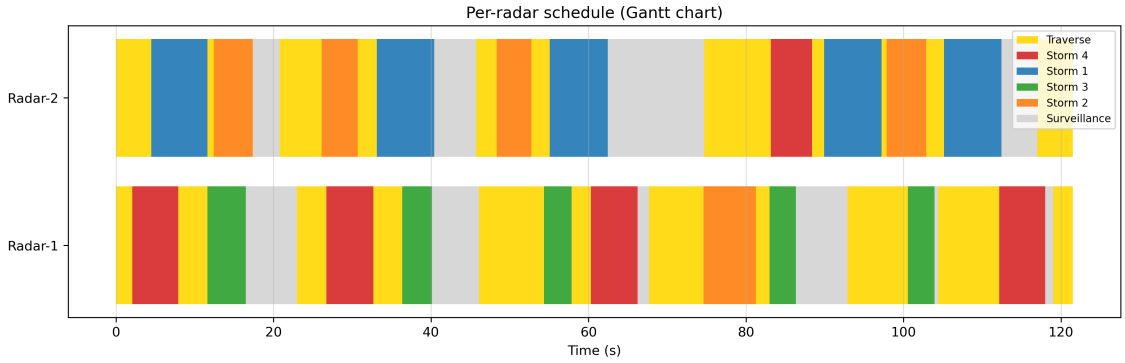
lance, but the faster storm motion introduces greater variation in the accuracy of the estimated storm footprint. The distinctive challenge in this scenario is therefore tracking rapidly evolving storm states rather than severe saturation of the radar-time budget.

A.1.4 Large-Storm Representative Trial

The *large-storm* scenario contains the same number of storms as the nominal case, but the larger storm footprints require wider sector scans and therefore longer tracking tasks. This increases the radar-time demand without increasing the number of competing storms. Figure A.10 compares the schedules produced by RLTS and LATS for the same representative storm configuration.



(a) RLTS.



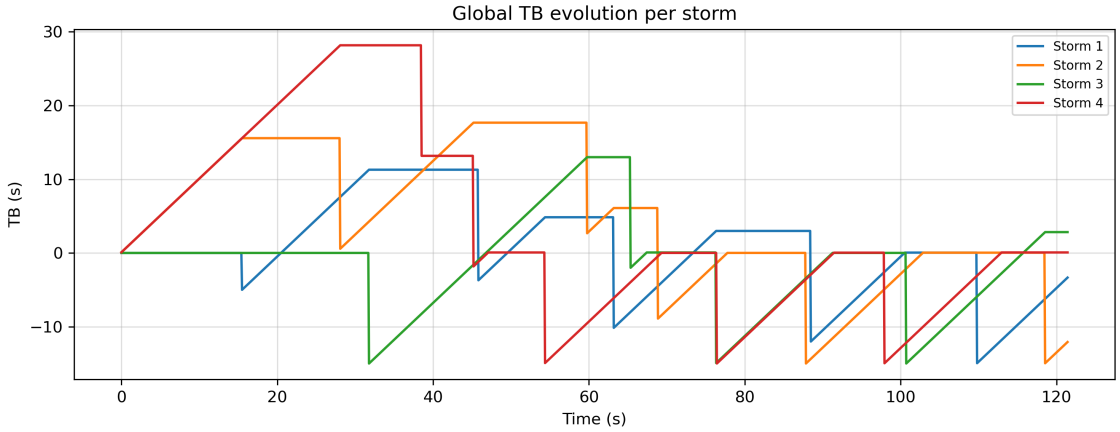
(b) LATS.

Figure A.10: Per-radar schedules for RLTS and LATS during the representative 2 min *large-storm* trial.

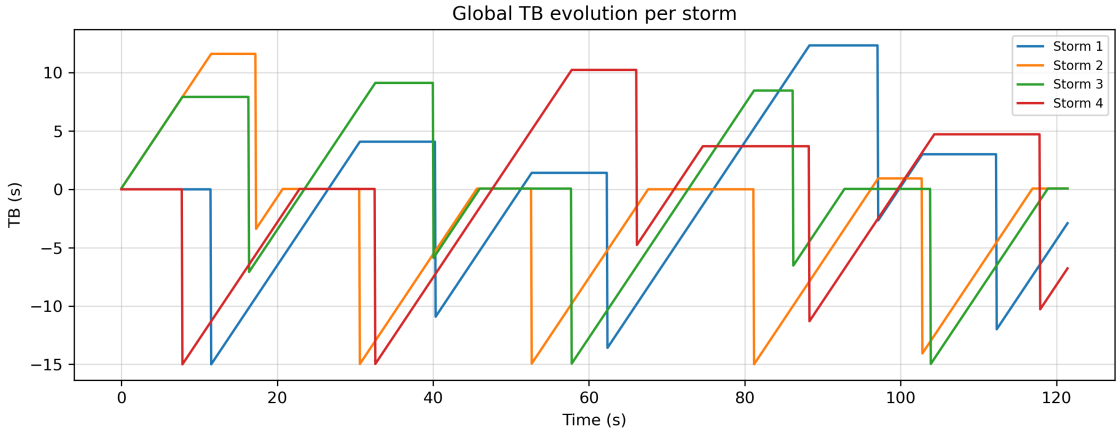
The schedules show that individual storm observations can occupy substantial continuous intervals of radar time because of the larger storm sectors. Consequently, fewer opportunities remain for other storm observations and background surveillance while a long sector scan is being completed. The LATS schedule shows a more structured distribution of the tracking workload between the two radars and retains several visible surveillance intervals, illustrating the role of explicit assignment when individual tracking tasks become more expensive.

Figure A.11 shows the corresponding global Time-Balance evolution. The larger posi-

tive excursions indicate periods in which storm observations remain waiting beyond their requested revisit interval before a new update is completed.



(a) RLTS.

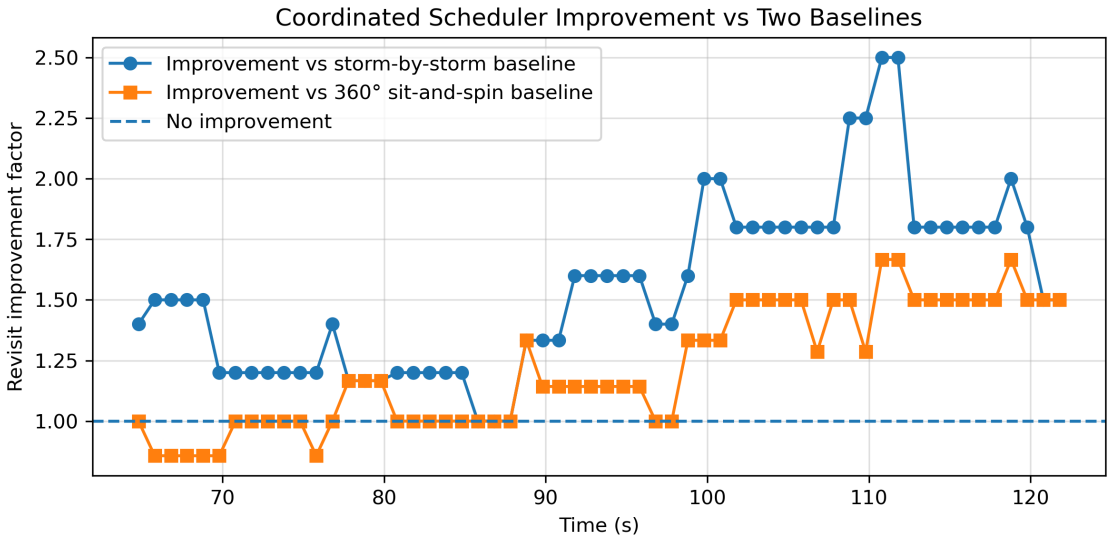


(b) LATS.

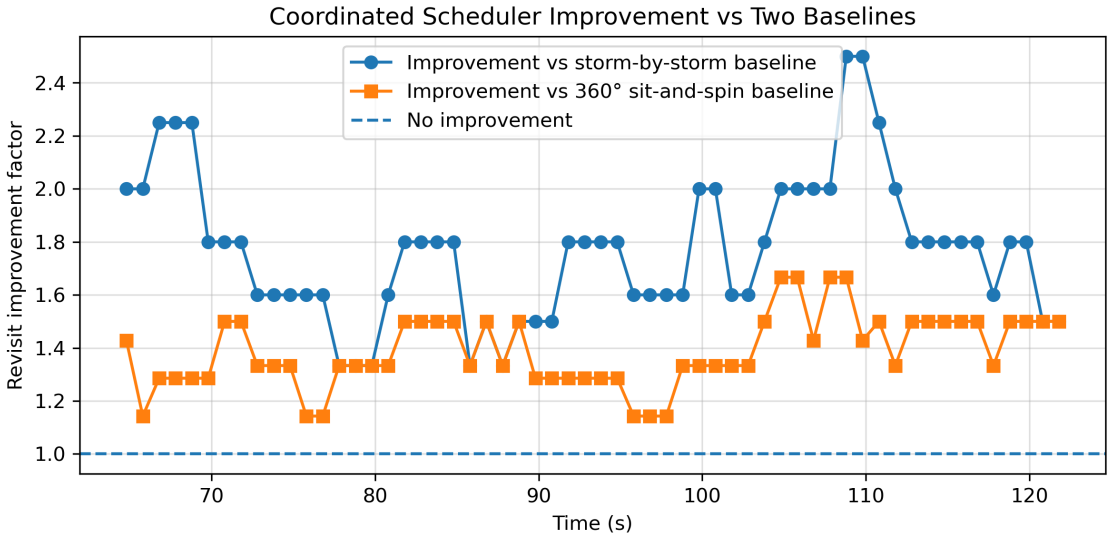
Figure A.11: Global Time-Balance evolution for RLTS and LATS during the representative *large-storm* trial.

The RLTS trial shows several large positive Time-Balance excursions during the first part of the simulation, reflecting the longer waiting times that can occur while other large storm sectors are being scanned. The corresponding LATS trial also contains overdue observations, but the positive excursions are generally more contained. This is consistent with the assignment stage accounting for scan duration and distributing the resulting workload between the radar nodes before radar-level task selection.

The resulting revisit performance is shown in Figure A.12. The horizontal line at $I = 1$ represents equal revisit-rate performance with the corresponding baseline.



(a) RLTS.



(b) LATS.

Figure A.12: Window-wise revisit improvement of RLTS and LATS relative to Baseline 1 and Baseline 2 during the representative *large-storm* trial. The horizontal line at $I = 1$ indicates equal revisit-rate performance with the corresponding baseline.

Both adaptive schedulers achieve an improvement relative to Baseline 1 over most of the displayed interval. The advantage of LATS is particularly clear in this representative trial, where the improvement remains above unity relative to both baselines throughout the displayed comparison interval. This supports the scenario-level finding that explicit workload assignment becomes useful when longer storm-sector scans place greater pressure on the available radar-time budget.

Representative-trial observation. The large-storm example shows that radar-network load is determined not only by the number of competing storms, but also by the duration of the requested observations. Wider storm sectors occupy the radar for longer periods and can

therefore increase the waiting time of other tasks. RLTS responds to these changes through adaptive task selection, while LATS additionally accounts for task cost when distributing the workload between the radars, providing a clearer revisit advantage under this form of resource constraint.

A.2 Planning-Window Sensitivity Results

This appendix provides the complete results of the planning-window sensitivity analysis introduced in Section 6.6. The planning window N_{plan} determines how frequently the storm-to-radar assignment in LATS is recomputed. The tested values were

$$N_{\text{plan}} \in \{1, 2, 5, 10, 20, 40, 80\}. \tag{A.1}$$

Since one dwell corresponds to approximately 0.09 s, these values represent assignment-update intervals ranging from approximately 0.09 s to 7.20 s.

The sensitivity study was performed separately for the five Monte Carlo scenarios. For each planning-window value, 50 trials were evaluated using the same scenario definitions. Within each scenario, the same trial seeds were used for the different planning-window values so that the comparison was performed under equivalent storm configurations. Each sensitivity trial contained 1350 dwell intervals, corresponding to approximately 121.5 s of simulation time. The average revisit rate was calculated using the same 64.8 s sliding-window formulation employed for the main revisit-performance evaluation.

Table A.1 reports the mean average revisit rate obtained across the 50 trials for every combination of scenario and planning-window length.

Table A.1: Mean average revisit rate of LATS for the tested planning-window lengths. Values are rounded to three decimal places; bold entries indicate the highest value within each scenario based on the unrounded results.

N_{plan}	Interval (s)	Low load	Nominal	High load	Fast storms	Large storms
1	0.09	**0.038**	0.036	0.041	0.036	0.036
2	0.18	0.038	0.036	0.041	0.036	0.036
5	0.45	0.037	0.036	0.041	0.036	0.036
10	0.90	0.037	0.036	**0.041**	0.036	0.036
20	1.80	0.037	0.036	0.041	0.036	0.036
40	3.60	0.037	**0.037**	0.041	0.037	0.036
80	7.20	0.037	0.037	0.040	**0.037**	**0.036**

The results show only modest changes in revisit performance over the tested planning-window range. No single value provides the highest mean revisit rate in every scenario. The shortest planning window performs best under *low-load*, while a value of 40 dwells gives the highest mean rate in the *nominal* scenario. The *high-load* scenario reaches its highest mean

revisit rate at 10 dwells, whereas the longest tested planning window gives the highest value for the *fast-storm* and *large-storm* scenarios.

The differences between neighbouring planning-window settings are small relative to the overall scenario-to-scenario differences. This indicates that LATS does not depend strongly on a narrowly tuned assignment-update frequency within the investigated range. The effective revisit-rate metric shows the same behaviour, with its median remaining unchanged across the tested planning-window values in all five scenarios.

For the final Monte Carlo campaign, a planning window of 10 dwells, corresponding to approximately 0.90 s, was retained. This value does not represent a unique numerical optimum across all scenarios. Instead, it provides a relatively short assignment-update interval while also giving the highest mean revisit rate in the *high-load* scenario, where explicit load-aware coordination is most relevant. The sensitivity results therefore support treating $N_{\text{plan}} = 10$ as a practical operating choice rather than as a finely tuned parameter.

A.3 Short-Duration Monte Carlo Consistency Analysis

Before performing the final 20 min Monte Carlo campaign, a short-duration analysis was carried out to examine whether the main scheduler trends remained consistent when a substantially larger number of independently generated storm configurations was considered. Simulations of 2 min duration were initially evaluated using 50 trials per scenario and were subsequently extended to 1000 trials per scenario.

The purpose of the 1000-trial campaign was not to provide the final performance results of the thesis, but to examine whether the relative behaviour of the scheduling methods observed in the smaller short-duration campaign persisted across a much larger set of randomly generated storm configurations. The final quantitative conclusions of this thesis are based on the 50-trial, 20 min campaign presented in Chapter 6.

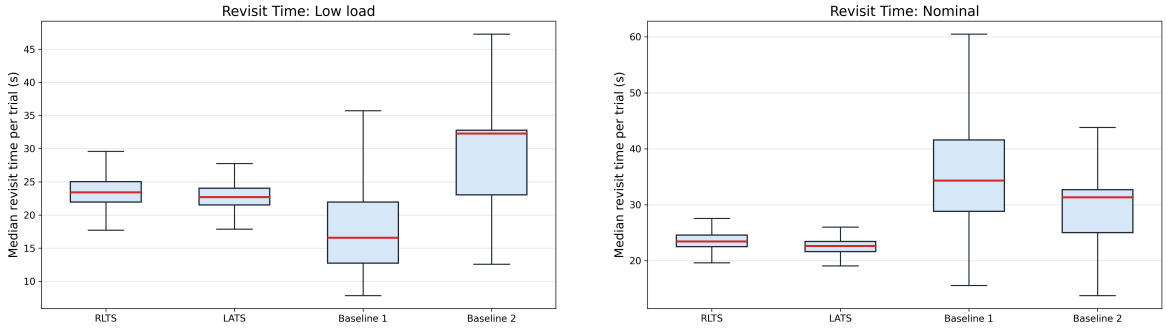
A.3.1 Revisit-Time Distributions

Figure A.13 presents the distribution of per-trial median revisit time for RLTS, LATS, Baseline 1, and Baseline 2 across the 1000-trial, 2 min campaign. Lower revisit times indicate more frequent completed storm updates.

The revisit-time distributions retain the same main scenario-dependent behaviour observed in the smaller short-duration campaign. Under *low-load*, Baseline 1 achieves the shortest typical revisit time, showing that a fixed cyclic strategy remains effective when only a small number of storm tasks require service.

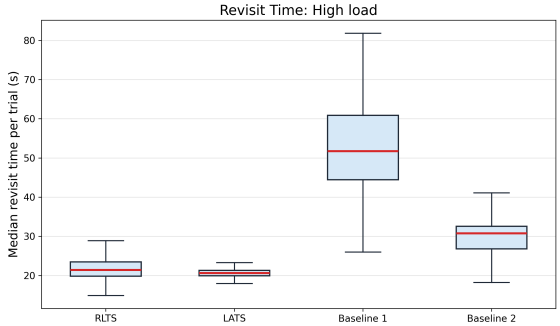
The ordering changes as radar demand increases. In the *nominal*, *high-load*, *fast-storm*, and *large-storm* scenarios, the adaptive schedulers provide shorter typical revisit times than the baseline methods. The clearest separation occurs under *high-load*, where Baseline 1 exhibits both substantially longer revisit times and greater trial-to-trial variation. A similar effect is visible for *large storms*, where the wider storm sectors increase the duration of

individual tracking tasks.

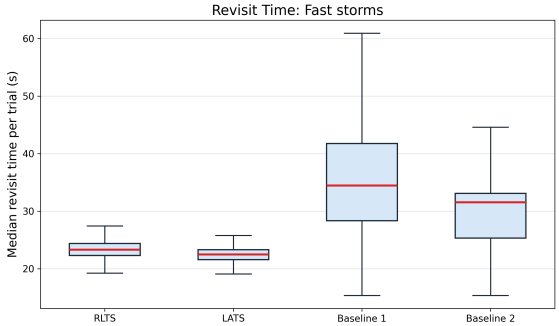


(a) Low load.

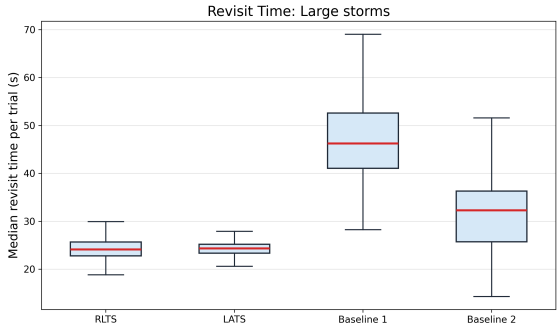
(b) Nominal.



(c) High load.



(d) Fast storms.



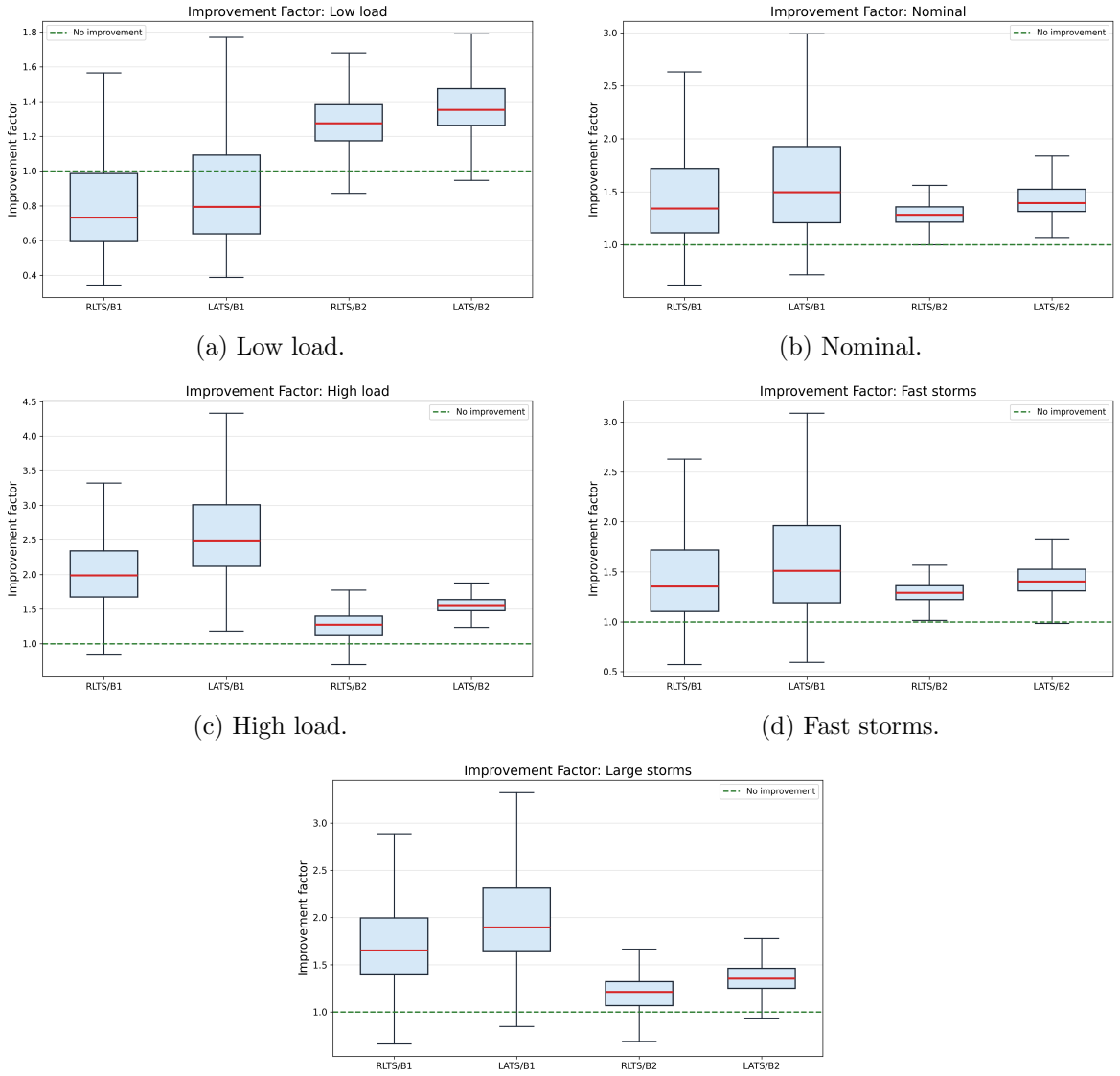
(e) Large storms.

Figure A.13: Distribution of per-trial median revisit time for RLTS, LATS, Baseline 1, and Baseline 2 across the five scenarios in the 1000-trial, 2 min Monte Carlo campaign. Lower values indicate more frequent completed storm revisits.

The 1000-trial distributions therefore show that the main dependence of scheduler performance on radar demand is retained across a substantially larger set of independently generated storm configurations.

A.3.2 Improvement-Factor Distributions

The same behaviour can be examined through the revisit improvement factor. Figure A.14 shows the distributions of the RLTS and LATS improvement factors relative to Baseline 1 and Baseline 2. The dashed horizontal line at $I = 1$ represents equal revisit-rate performance with the corresponding baseline.



(a) Low load.

(b) Nominal.

(c) High load.

(d) Fast storms.

(e) Large storms.

Figure A.14: Distribution of the revisit improvement factor of RLTS and LATS relative to Baseline 1 and Baseline 2 across the five scenarios in the 1000-trial, 2 min Monte Carlo campaign. The dashed line at $I = 1$ indicates equal revisit-rate performance with the corresponding baseline.

The improvement-factor distributions reinforce the behaviour observed in the revisit-time results. The *low-load* scenario remains the main boundary case. Relative to Baseline 1, both adaptive schedulers are centred below the equal-performance level, confirming that additional scheduling flexibility provides limited benefit when the available radar capacity is sufficient for the small number of tracking tasks.

For the remaining scenarios, the adaptive schedulers are centred above $I = 1$ relative to Baseline 1. The strongest improvement occurs under *high-load*, where several storm tasks compete simultaneously for radar time. A pronounced improvement is also observed for *large storms*, where longer sector scans increase the radar-time demand. These results are consistent with the interpretation that adaptive task selection and load-aware assignment

become more valuable as the available sensing resources become constrained.

The comparisons with Baseline 2 show a more stable improvement across the five scenarios. Both RLTS and LATS are generally centred above the equal-performance level, reflecting the fact that the revisit behaviour of Baseline 2 is primarily determined by its fixed full-azimuth scanning cycle.

A.3.3 Relation to the Final Monte Carlo Campaign

The short-duration results should not be interpreted as direct numerical replicates of the final 20 min results. Extending the simulation duration changes the number of tracking, assignment, and surveillance cycles observed within each trial and can therefore change the absolute values of the reported metrics. The purpose of the short-duration campaign is instead to examine whether the overall scheduler ordering and scenario-dependent behaviour remain consistent as the number of independently generated storm configurations is increased.

The 50-trial and 1000-trial short-duration evaluations showed consistent qualitative trends. In particular, Baseline 1 remained strongest under *low-load*, while the adaptive schedulers became more advantageous as competition for radar time increased. The strongest benefits of adaptive scheduling were again associated with resource-constrained conditions, particularly the *high-load* and *large-storm* scenarios.

Based on this consistency, the final campaign retained 50 trials per scenario while increasing the simulation duration from 2 min to 20 min. This allowed the evaluation to focus on scheduler behaviour over a much longer sequence of decisions while keeping the computational cost tractable. The final numerical results and conclusions of the thesis therefore remain those obtained from the 50-trial, 20 min campaign presented in Chapter 6.

Appendix B

Declaration of Generative AI Use

Generative AI, specifically ChatGPT, was used as a supporting tool during the preparation of this thesis. Its use was limited to improving the clarity and organisation of written sections and providing assistance during code review and debugging. Suggestions produced by the tool were checked, modified where necessary, and incorporated only after review by the author.

All decisions concerning the research approach, simulator development, experimental setup, evaluation of the scheduling methods, interpretation of the results, and conclusions were made by the author.