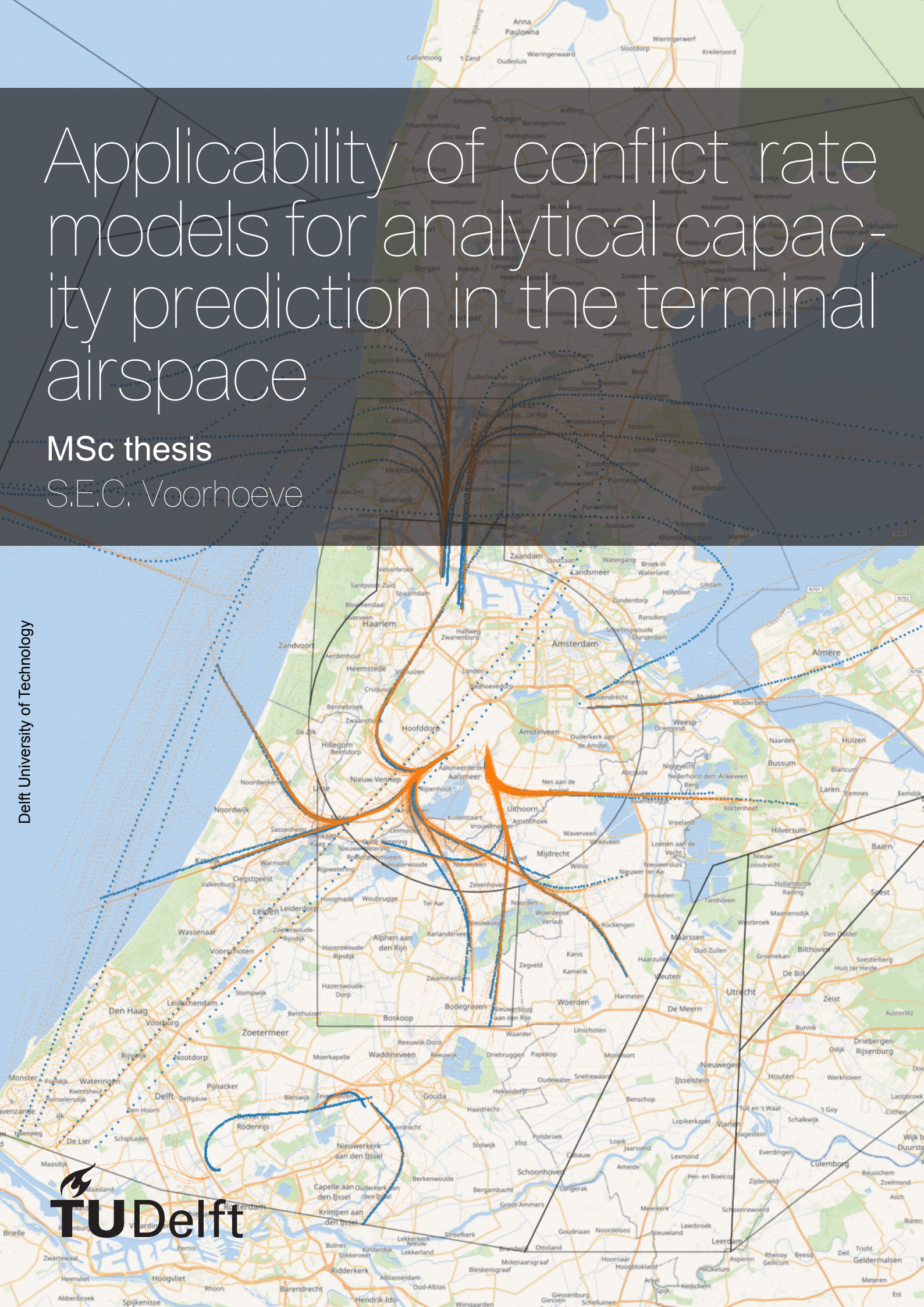
Applicability of conflict rate models for analytical capacity prediction in the terminal airspace

MSc thesis

S.E.C. Voorhoeve

Delft University of Technology



Applicability of conflict rate models for analytical capacity prediction in the terminal airspace

by

S.E.C. Voorhoeve

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday May 21, 2021 at 09:30.

Student number: 4228766

Thesis committee:

Prof. dr. ir. J. M. Hoekstra,	TU Delft, supervisor
Dr. ir. J. Ellerbroek,	TU Delft
Dr. O. A. Sharpanskykh,	TU Delft

TU Delft

Contents

Nomenclature

Symbols

$\mathbb{T}$	(mean) Track matrix
$\mathbb{W}$	Similarity matrix
$\mathbf{x}$	Vector of position data
$\underline{V}_{geo}$	geovector
a	Horizontal airway cross section
B	Along-track separation
b	Vertical airway cross section
C	Conflict rate
c	Number of principal components
g	Horizontal aircraft dimension
h	Altitude
i, j	Flow/cluster number
M	Number of flows
N	Number of aircraft
n	Vector length
N_c	Number of conflicts
p	(Conflict) probability
R	Traffic replications
R^2	Coefficient of determination
R_0	Distance from the origin
S	Separation requirement
s	Vertical aircraft dimension
S_h	Horizontal separation requirement, usually 3 nm or 5 nm
S_v	Vertical separation requirement, usually 1000 ft
T	Duration
t_l	Look-ahead time
T_{log}	Simulation logging time
T_{settle}	Simulation settling time
V	Velocity (groundspeed, unless otherwise specified)

$V_{r,h}$ Relative horizontal velocity

VS Vertical speed

w Critical standard deviation for clustering

x, y Position coordinates

Greek symbols

α Angle between (flows of) aircraft

χ Heading angle

λ Arrival rate of aircraft; Poisson parameter

μ Critical mean for clustering

ϕ Bearing angle

σ Scale parameter of the Gaussian similarity kernel

Abbreviations

ACAS Airborne Collision Avoidance System

AIP Aeronautical Information Package

ATC Air Traffic Control

ATCo Air Traffic Controller

CAMDA Capacity Assessment Method for Decentralized ATC

IAF Initial Approach Fix

IFR Instrument Flight Rules

IV Independent Variable

nm Nautical mile

PCA Principal Components Analysis

TCAS Traffic Collision Avoidance System

TMA Terminal Maneuvering Area

List of Figures

List of Tables

Part I

Paper

Applicability of conflict rate models for analytical capacity prediction in the terminal airspace

Author: S.E.C. Voorhoeve
Supervisors: J.M. Hoekstra, J. Ellerbroek

***Abstract*—The applicability of conflict rate models to controlled, structured, terminal airspace is investigated. A collision rate model is adapted to conflicts between flows of aircraft, and its validity is empirically confirmed. Conflicts are divided into between-flow conflicts and within-flow conflicts. Applicability to a terminal airspace without strictly adhered-to procedural routes is investigated by clustering logged mode S-surveillance data into flows using a self-tuning spectral clustering method. This clustering method can effectively partition flight tracks into clusters, but is sensitive to the stop conditions, and unclusterable flight tracks make up a small but significant proportion of conflicts. The predictions of the conflict rate model correlate with observed conflict rates in a replay simulation, but are initially biased towards underpredicting the conflict rates. Altering the timing, intensity and separation requirements for the replay greatly reduces this bias for converging flows, but leads to an overprediction of other conflict types, and altered replay results should be interpreted with care. Analytical predictions for between-flow conflicts can be made, but applicability to within-flow conflicts is limited due to the different types of interactions. The resulting conflict rate model explicitly defines the relation between separation requirements, relative velocities, and flow intensities.**

I. Introduction

The demand for air traffic generally increases steadily. For The Netherlands' largest airport, Schiphol, the previously agreed upon limit of 500.000 movements a year is reached in 2018 [1]. For safety reasons, regulators have set a minimum separation between aircraft, depending on flight type and airspace classification [2]. In the airspace around major airports, Air Traffic Control (ATC) is responsible for ensuring these separation minima, while at the same time promoting throughput of aircraft. Separation ensurance is in part achieved by limiting the number of aircraft that can enter the airspace, known as Air Traffic Flow Management (before aircraft depart, they can be held to ensure that once they enter a certain airspace, that airspace is not over its maximum capacity).

Aircraft that get too close to each other need to be separated, and the rate at which this happens depends on the traffic properties. Airspace capacity is not only affected by the number of aircraft at a given time (traffic density), but also by the way in which aircraft interact (traffic complexity). A measure for the capacity is the number of *conflicts* (predicted close encounters between aircraft that require mitigation). The number of conflicts can be experimentally determined in fast-time simulations, but this gives little to no information about the underlying mechanisms that cause conflicts to occur. Understanding analytically how traffic density and traffic complexity influence the number of conflicts is helpful in the airspace design process as well as in flow planning. An analytical model can not only identify high-conflict traffic, but also indicate what properties of that traffic are the cause of the high *conflict counts* (or conflict *rate*) for different airspace concepts and airspace designs.

A first distinction is made between unstructured and structured airspace. In *unstructured* airspace, traffic uses *direct routing*, i.e. traffic can have an origin and destination anywhere in the airspace. Traffic properties, particularly headings, are independent of horizontal position (though different *layers* of traffic may have different traffic property distributions). In *structured* airspace, especially in the terminal airspace (Terminal Maneuvering Area, TMA), structured flows exist. Arriving traffic enters the airspace at a limited number of fixed points, Initial Approach Fixes (or IAF), heading towards the airport. Departing traffic heads to a limited set of fixed points, where they enter an airway. This leads to a concentration of traffic and similar traffic properties along these arrival and departure routes, such that traffic properties are largely *dependent* on (horizontal) position.

A second distinction is made between decentralized and centralized ATC. In decentralized ATC, aircraft are responsible for ensuring their own separation to other traffic, whereas in centralized ATC, such as in the TMA, separation is the responsibility of a central Air Traffic Controller (ATCo).

For *unstructured* airspace concepts with *decentralized* ATC, Sunil developed a Capacity Assessment Method for Decentralized ATC (CAMDA) [3], using conflict count models. CAMDA cannot be applied to structured, controlled airspace cooncepts, as the assumptions on uniformity in heading differences [4] that are made for the conflict count model in unstructured concepts are not satisfied. The goal of the current research is to determine the applicability of conflict count models to centralized, structured airspace concepts by using a separate conflict count model, based on an airway collision model [5]. Next, an identification of traffic concentrations is required. Though departing traffic generally follows published procedural routes, this is not always the case for arriving traffic at busy airports such as Schiphol. In fact, the instrument approach charts state that navigation "is primarily based on radar vectors provided by ATC", and that "execution of the complete procedure" is (only) in case of a communication failure[1]. Therefore, in the current research, logged mode S

[1]See e.g. `AD 2.EHAM-IAC-18R.1` in the Dutch Aeronautical Information Package.

flight track data (surveillance messages containing position, altitude, and under some conditions also other data such as velocity and heading) of real traffic [6] is employed. The dimensionality of the data is reduced by applying a spectral clustering algorithm [7] to aggregate several hundreds of individual logged flight tracks to several tens of *flows* of similar flight tracks. Subsequently, the interactions between (and within) these flows can be predicted using analytical conflict count models. However, analytical conflict counts may not match actual conflict counts of real traffic, since air traffic controllers will have performed actions to avoid or resolve conflicts. Several options on how to remove these *deconflictive* effects are proposed and qualitatively compared.

The structure of this paper is as follows. Section II introduces the background and main concepts. Section III describes the acquisition of logged flight track data and the clustering methodology. Section IV discusses how to extend a collision rate model to conflict rates and the experiments used to validate this model for conflict counts. In section V, this model is applied to clustered flight track data for flow conflict prediction. The results of the clustering and flow conflict prediction experiments are presented in section VI and discussed in section VII. Finally, the paper is concluded in section VIII.

II. Background

A. Collisions, intrusions and conflicts

When two aircraft would make physical contact, a *collission* takes place, which is obviously undesirable. Collisions are meant to be avoided by ensuring a certain minimum separation distance. Generally, in the terminal airspace, this means that the difference in altitude is at least $S_v = 1000$ ft, or that the lateral distance is at least $S_h = 3$ nautical mile (nm). When an aircraft is laterally closer than 3 nm with a vertical separation of less than 1000 ft to another aircraft, an *intrusion*, or *loss of separation*, occurs (Fig. 1a). When such a loss of separation is not yet the case but is expected in the near future (between a time t and a look-ahead time t_l), a *conflict* is said to occur (Fig. 1b) and (in centralized ATC airspace) ATC needs to intervene to avoid a conflict from evolving into a loss of separation. Commercial passenger aircraft are also equipped with a collision avoidance system (ACAS/TCAS) for when air traffic control would fail to ensure separation, giving a Traffic Advisory and possibly a Resolution Advisory (alerting the pilots to an imminent close encounter and, depending on the circumstances, recommending an evasive maneuver).

It should be noted that conflicts are inevitable, especially for longer look-ahead times, and no cause for grave concern as long as ATC can avoid conflicts from evolving into intrusions by taking action to deconflict these aircraft. However, taking such action requires resources from the ATCo, and the number of conflicts is directly related their workload and as such a measure for airspace capacity. This is the reasoning behind the current research, focusing on conflicts in airspace with centralized ATC. Note that even in a system with decentralized ATC, an airspace with a lower number of conflicts is inherently safer, as there are less potential intrusions or collisions.

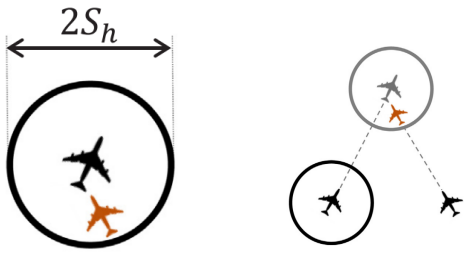


Fig. 1. Intrusions (left) and conflicts (right). Figures taken from from [3].

B. Modelling conflicts

The number of (instantaneous) conflicts $N_{\mathrm{c,inst}}$ in a volume of airspace can be modelled by the total number of possible combinations of aircraft $N_{\mathrm{ac,inst}}$ in that airspace, times the average conflict probability between any two aircraft:

$$N_{\mathrm{c,inst}} = \binom{N_{\mathrm{ac,inst}}}{2} p \tag{1}$$

In general, the conflict probability p should be considered in three dimensions: $p = p_{3d}$. However, consider a terminal airspace close to an airport under the following assumptions:

(i) the routes that arriving aircraft and departing aircraft take are sufficiently separated, so that no conflict between arriving and departing aircraft will occur,
(ii) all *arriving* aircraft have a similar altitude profile and speed, and
(iii) all *departing* aircraft have a similar altitude profile and speed.

The altitude profiles of arriving and departing traffic differ not only in sign, but also in magnitude (departing traffic climbs faster than arriving traffic descends); additionally, the fact that aircraft usually take-off and land against the wind means that close to the ground, departing traffic will be upwind, and arriving traffic downwind, with respect to the airport. Combined, this means that even when departing and arriving traffic approach each other laterally, their vertical separation will most likely already exceed the minimum.

Under the above assumptions, the traffic scenario can be considered as two layers of traffic: one layer for departing traffic, one layer for arriving traffic, and no interaction between these layers. In essence, the vertical dimension drops out, and the conflict probability within a layer can be regarded as a two-dimensional problem, i.e. $p = p_{2d}$. The number of conflicts in the terminal airspace can be calculated by summing the number of conflicts in each of these 2 'layers'. When assumption (i) is invalid, this can be compensated by adding an interaction term between the two layers. If the altitude and speed profiles of traffic within a layer are similar but not exactly equal, a separate term for conflicts *within* a layer may be needed.

C. Flows concept

Instead of regarding separate layers, it is more appropriate to consider the arriving and departing traffic as a special case of a situation with multiple *flows* of traffic, where the traffic in each flow is similar with respect to other traffic in that flow. A *flow* can be thought of as the 3D-trajectory traced by a group of aircraft, with similar lateral tracks and similar altitude profiles, such that the vertical dimension can be ignored and the flow can be thought of as 2D. When all aircraft in a flow have similar characteristics, the conflicts between aircraft in a pair of flows will also have similar characteristics and can be calculated relatively straightforward. The goal is to analytically predict the conflicts between, and if necessary within, flows, based on these flow characteristics: the mean path, altitude profile, velocity profile and intensity; and potentially the distributions of these characteristics.

Though procedural arrival and departure routes exist for air traffic in the terminal area, these are often deviated from (at the initiative of the air traffic controller) to achieve tactical separation with a higher throughput, especially in the case of arriving traffic. In order to obtain realistic flow profiles, logged mode S surveillance flight track data, captured using pyModeS [6], was used. Depending on the scenario, the logged flight track data was modified to remove the effect of controller deconflictive action (see subsection V-C); furthermore, only the segments within the terminal airspace of Schiphol Airport was considered. These segments were subsequently grouped into flows of traffic using clustering techniques (see section III). After determining the relevant characteristics predicting conflicts (see section IV), these characteristics were extracted from the flows and used to analytically predict conflict rates. Finally, these analytically predicted conflict rates are compared to *replay simulation conflict rates*, the observed conflicts in a replay simulation of the logged flight track data.

III. Flight-track clustering

A. Data acquisition

Flight tracks are obtained from mode S surveillance data broadcast from aircraft, which were received on a receiver in Delft, The Netherlands and processed using pyModeS [6]. Data is available for the entire month of Januari, 2018.

The airspace definition is obtained from data in the Dutch AIP [2]. For each track, the intersection of the track points with the selected airspace objects is used. The track points are converted from latitude/longitude to a 2D projection centered around the center of the airspace under consideration, to prevent the clustering algorithm from having a bias towards proximity in longitude at higher latitudes.

B. Clustering algorithms

Partitioning flight track data into clusters of similar flight tracks generally involves reducing the dimensionality of the input data by converting it to a set of features for each track. Once the dimensionality has been reduced, these lower-dimensional features are partitioned into groups. Several options exist for each step and different combinations have been proposed in literature. First, several partitioning options are described, after which several clustering approaches are listed, as well as the choice in this paper. Different approaches to clustering flight track data have been proposed in literature.

1) Feature set clustering: Low-dimensional features can be used as an input for a partitioning process that groups together entries with similar features, i.e. the actual clustering step. Possible clustering algorithms include DBSCAN, *k*-means, and graph partitioning:

a) DBSCAN (Density-based spatial clustering of applications with noise): DBSCAN identifies points in a high density region by finding collections of points having a minimum number of $minPts$ (the first hyperparameter) which are a maximum distance ϵ (the second hyperparameter) away (points that satisfy the ϵ criterion but themselves do not have $minPts$ other points are said to be *directly reachable* and are included) [8]. All these distinct collections of points are separate clusters; points not in a cluster are classified as noise. These points need not be location data, since in that case two distinct but crossing tracks could be classified together (depending on the hyperparameters); the clustering can be applied to inferred parameters of tracks. The number of clusters is an *output* of the clustering algorithm.

b) k-means clustering: the number of clusters is an input (hyperparameter k). k-means clustering partitions a dataset into k partitions centered around cluster *means* or centroids, minimizing within-cluster variances.

The necessity for *a priori* specification of k is a downside for the current application, as the number of clusters may not always be clear. This is sometimes overcome by trying multiple values of k and optimizing for some cost function, such as in [9].

c) Graph partitioning: A similarity matrix $\mathbf{W}$ is recursively partitioned in two submatrices in each step, based on the sign of the elements in the *Fiedler vector* [10] of $\mathbf{W}$. In this way, each partition has maximum similarity within a group, and minimum similarity between groups. A stopping condition can be e.g. comparing the partition variance $\omega_{partition} = \mathrm{var}\left(\mathbf{W}_{partition}\right)$ against a threshold variance ω_{min}.

2) Conversion to features:

a) Turning points: In reference [11], turning points are extracted from flight tracks, and similar turning points are clustered by applying either *k-means* or *DBSCAN* as a clustering algorithm. A combination of one or more turning points in a cluster is considered as one *way point*. Then, each flight track is considered as a sequence of these way points, with varying length. Finally, trajectories are clustered together by considering the longest common subsequence and generating clusters of sequences.

b) Principal Component Analysis (PCA): As a starting point, the (x, y, h) positions of flight tracks are considered and all tracks are resampled to a vector of length n [11]. These

[2]TMA/CTA definition is described in ENR 2.1; CTR definition is described per airport in AD 2.17

vectors are augmented with other features (distance from a center point providing rate of convergence; distance from top left corner representing symmetry; angular position providing information about the overall location of the trajectory; and heading of the aircraft providing information about straight or curved trajectories). Then, the dimensionality is reduced by performing a Principal Component analysis and projecting each trajectory on the first c Principal components, resulting in a vector of length c for each trajectory. Finally, these vectors are clustered using DBSCAN.
This approach was an improvement to [9], whre a similar algorithm was proposed without augmenting the trajectories and using k-means clustering instead of DBSCAN.

c) Adjacency: Reference [12] argues that PCA may not always capture local or nonlinear characteristics and that the choice of features for augmenting the data matrix can seem arbitrary. They proposed an algorithm based on *spectral clustering*. It starts by converting the (logged) flight track positions to a number of n datapoints per track (downsampling when more data is available, and only considering tracks where at least n datapoints are available), and considers *similarity* between flight tracks by calculating a square adjacency matrix $\mathbf{W}$. Element $W_{i,j}$ encodes the similarity between track $\mathbf{x}_i$ and $\mathbf{x}_j$ by computing a Gaussian kernel:

$$W_{i,j} = e^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma_i \sigma_j}} \tag{2}$$

In Eq. (2), $\mathbf{x}_i$ and $\mathbf{x}_j$ are $(n, 2)$ matrices of (x, y) records and σ is a scale parameter, controlling the width of the neighborhoods, which can be a local scale parameter specific to track i or a global scale parameter, in which case $\sigma_i = \sigma_j = \sigma$. The authors note that the effect of time parametrization and Euclidation distance is that flights of similar lengths will be clustered together; consider two similar tracks where one aircraft flew a holding pattern and the other aircraft did not - the similar (x, y) positions will be at different (relative) timesteps resulting in dissimilarity. The resulting adjacency matrix $\mathbf{W}$ has values ranging from zero for completely dissimilar tracks to one for completely similar tracks and is partitioned using a graph-partitioning approach. An example of the graph partitioning is shown in Fig. 2.

For the current research the spectral clustering method is selected (the adjacency matrix, partitioned using graph partitioning). It achieves good results given the vectoring nature of tracks in e.g. the terminal Schiphol airspace, where a lot of different tracks exist. Clustering them based not only on spatial proximity but also similar path lengths achieves a relatively large number of flows but with a small spread. The flexibility in stopping conditions is an additional advantage.

Note that this spectral clustering has also been extended to obtain *temporally persistent* flows, flows persisting across the time dimension (e.g. multiple days), by running the the algorithm on multiple periods (e.g. days) and computing the nominal tracks of each cluster in each period. These nominal tracks are then used as an input for the same clustering method (possibly with different hyperparameters, e.g. inputs to the stop condition) [7]. In this way, weeks or months worth of data can be analysed at a reduced computational cost (compared to aggregating all input data and using just one clustering step)[3]. An additional advantage is that 'outlier' days can be identified and dealt with - either to remove this as 'noise', or to study those outliers specifically.

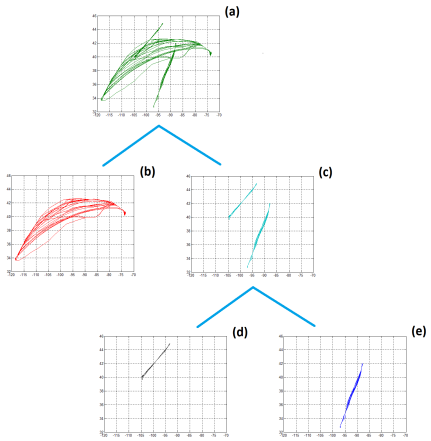


Fig. 2. Clustering based on similarity. The grids represent longitude-latitude position data of flight tracks. Every level shows a subdivision of flight tracks. Figure taken from [12]

C. Clustering hyperparameters

The spectral clustering algorithm has a number of hyperparameters:

a) Aircraft states to consider: the fundamental aircraft states to consider are the x and y coordinates of the aircraft tracks. Other parameters can be included as well in the Gaussian kernel (Eq. (2)) simply by including more states in the $\mathbf{x}_i$ matrices. However, the difference in magnitudes of the states (e.g. altitude and location) induces a bias unless corrected for with a scaling factor. Additionally, the circular nature of aircraft headings requires attention when calculating similarities in headings. Nevertheless, altitude or groundspeed could be included as logical extra states.

b) Vector dimension n: Including more points per track increases computation time and memory usage. $n = 200$ was found to give sufficient results at acceptable computation times.

c) Scale parameter σ: σ influences when flights are considered similar. Instead of using a global scale parameter, the scale parameter for flight i, σ_i, is determined using a self-tuning spectral clustering method, which "allows self-tuning of the point-to-point distances according to the local statistics of the neighborhoods surrounding points" [13] by setting the local scale parameter to the distance to the K-th neighbor of $\mathbf{x}_i$ for some constant value K; in this research, $K = 8$:

$$\sigma_i = \mathrm{d}(\mathbf{x}_i, \mathbf{x}_K) \tag{3}$$

[3]The similarity matrix has dimension (N, N) so the required computational resources grows with $\mathcal{O}(N^2)$

d) Stop condition: Reference [12] suggests stopping when the ratio of maximum distances (converging to zero during the recursive partitioning process) is lower to some critical value ω_{min}:

$$\frac{\max\left(W_{i.,i.}\right)}{\max\left(W\right)} < \omega_{min} \tag{4}$$

However, since this stop condition did not yield satisfactory results for the considered data (achieving very granular partitioning), the current research uses a different stopping condition which was found to give better results. We stop when the subset under consideration[4] has an average similarity $\overline{W_{i,i}}$ of at least $\mu_{stop} = 0.25$, *and* when the standard deviation is at most $w_{stop} = 0.25$.

$$\text{stop}\left(W_{i,i}, \mu_{stop}, w_{stop}\right) = \overline{W_{i,i}} > \mu_{stop} \cap \text{std}\left(W_{i,i}\right) < w_{stop} \tag{5}$$

Additionally, when a partition has a size less than some minimum (in this research, 7 tracks), the tracks in that partition are considered 'unclusterable' and are labeled as noise.

IV. Rate of conflicts between traffic flows

This section describes the *conflict rate* between two traffic flows based on the work of [5], who developed a *collision rate* model, which is introduced and adapted in this section. The adapted model, analytically predicting conflict rates between flows of traffic, is verified by (i) calculating conflict rates analytically using an adapted version of the collision rate model (Section IV-A, (ii) obtaining conflict rates empirically from a simulation (Section IV-B) in several experiments, and (iii) comparing the analytical conflict rates to the simulated conflict rates.

A. Theoretical model

Reference [5] considered the collision rate for a crossing between two airways (see Fig. 3).

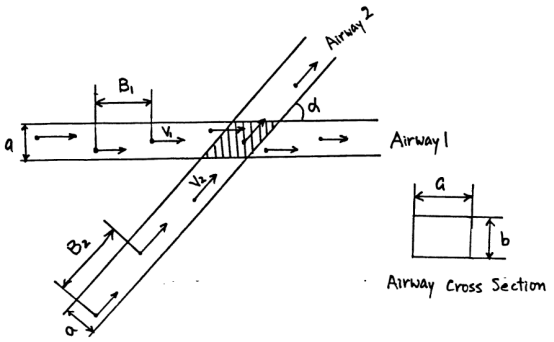


Fig. 3. Geometry of the intersection of airways or flows. Figure taken from [5]

1) Assumptions:

a) Speed and separation: Aircraft on airway 1 and airway 2 have constant airspeeds of V_1 and V_2. The along-track separations are B_1 and B_2, which can be constant values or stochastic variables as a result of a Poisson process. Therefore, more generally, expected values of the along-track separations are used: $\mathbf{E}(B_1)$ and $\mathbf{E}(B_2)$, respectively.

b) Airway geometry: The calculation is made for two airways, regarded as rectangular tubes with height b and width a, crossing at an angle α. It is assumed that the probability distribution of aircraft on the airway cross section (so along a and b) is uniform.

c) Aircraft geometry: Aircraft are modelled by a cylinder with a diameter of $2g$ and height $2s$ (g and s are assumed to be sufficiently small compared to a and b).

Under these assumptions, the *collision rate* CR can be calculated by considering the rate at which aircraft on airway 1 enter the intersection (the shaded area in Fig. 3) and multiplying this with the expected number of airway 2 aircraft in the intersection.

$$CR_{1,2} = (\text{airway 1 arrival rate})(\text{expected}N_{\text{ac,airway 2}}) \tag{6}$$

The first term can be split in the rate of horizontal overlap (which includes the ratio of the relative horizontal velocity $V_{r,h}$ to the along-track separation), and the probability of vertical overlap $P(\text{vo})$[5]:

$$\text{airway 1 arrival rate} = \frac{2gV_{r,h}}{a\mathbf{E}(B_1)}P(\text{vo}) \tag{7}$$

The expected number of airway 2 aircraft (i.e. along the skewed side of the parallelogram, consisting of an along-track distance of $a/\sin(\alpha)$) is the ratio of that length and the expected along-track separation:

$$\text{expected}N_{\text{ac,airway 2}} = \frac{1}{a\mathbf{E}(B_2)}\frac{a^2}{\sin(\alpha)} \tag{8}$$

The product of these gives the *collision rate* in the intersection:

$$CR_{1,2} = \frac{2gV_{r,h}}{\mathbf{E}(B_1)\mathbf{E}(B_2)\sin(\alpha)}P(\text{vo}) \tag{9}$$

Because this research focuses on *conflict* rates[6], the aircraft dimension g is replaced by the horizontal separation requirement S_h (the vertical dimension would be S_v but cancels out). Finally, substituting the full expression for $V_{r,h}$, assuming a Poisson process for aircraft arrivals on airway n, and substituting the vertical separation requirement for vertical overlap,

$$V_{r,h} = \sqrt{V_i^2 + V_j^2 - 2V_iV_j\cos(\alpha)} \tag{10}$$

$$\mathbf{E}(B_n) = \frac{V_n}{\lambda_n} \tag{11}$$

$$P(\text{vo}) = P(\Delta h_{i,j} < S_v) \tag{12}$$

[4]The diagonal entries are left out: only similarity values between different flows are considered

[5]For the airway case considered by [5], $P(\text{vertical overlap}) = 2s/b$

[6]Reference [5] treats the equivalence of conflicts and collisions in a special note in their work.

the *conflict* rate model between airways i and j $C_{i,j}$ is obtained, i.e. the conflicts between airways i and j per unit of time:

$$C_{i,j} = \frac{2S_h\sqrt{V_i^2 + V_j^2 - 2V_iV_j\cos(\alpha_{i,j})}}{\frac{V_i}{\lambda_i}\frac{V_j}{\lambda_j}\sin(\alpha_{i,j})} P(\Delta h_{i,j} < S_v) \tag{13}$$

B. Simulation environment

To simulate and count conflicts between flows of traffic a simulation environment was developed using a Test-Driven Development method, allowing straightforward generation of traffic flows and detecting the conflict counts within and between these flows. An example is shown in Fig. 4, showing aircraft (circles with velocity vector) in 3 flows with two aircraft in a conflict state (bigger circles).

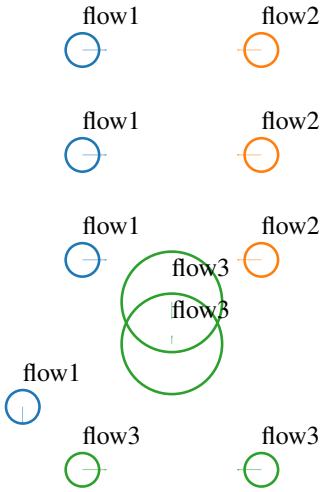


Fig. 4. Example visualisation of the simulation environment

C. Varying isolated quantities

In the simulation, M *flows* of each N aircraft are created. Each flow $i (i = 1 \dots M)$ has aircraft flying from a starting position on a circle with radius R_0 and on a bearing ϕ_k from the origin, heading towards the origin (i.e. the heading angle $\chi_k = 360 - \phi_k \pmod{360°}$). Aircraft 'spawn' according to a Poisson process with intensity λ_i, and fly with velocity V_i. Conflicts are logged during a time T_{log} following a settling time T_{settle}. These times should be chosen such that the simulation has time to settle and no flows are 'exhausted' before $T_{end} = T_{settle} + T_{log}$.

For every aircraft pair *within* a flow i, a boolean (N_i, N_i) matrix registers whether aircraft pair $l, m (l, m = 1 \dots N)$ has been in conflict with another within the conflict time window. Every flow pair i, j has a similar matrix (of size (N_i, N_j) registering Boolean conflict states between aircraft pairs l, m $(l = 1 \dots N_i, m = 1 \dots N_j)$ in that flow pair. The true values in these matrices are counted and divided by the simulation time T_{log}, resulting in the within-flow simulated conflict rates $C_{i,i}^{simulated}$ (when applicable) and between-flow simulated conflict rates $C_{i,j}^{simulated}$. Each experiment is repeated 10 times.

TABLE I
BASELINE CONTROL VARIABLES

Variable	Value	Unit
ϕ	$\in [0, 30, 60, 90, 120]$	°
V	200	kt
λ	15	aircraft/hr
S_h	3	nm
Δh	0	ft
VS	0	ft/min
T_{settle}	2.5	hr
T_{log}	2.5	hr
t_l	300	s
R_0	200	nm
N	100	[-]
Repetitions	10	[-]

a) Independent variables (IV): these are described per experiment in the respective experiment description.

b) Dependent variable: the simulated conflict rate between flows i and j, $C_{i,j}^{measured}$. This is compared to the analytical conflict rate calculated using Eq. (13).

c) Control variables: the baseline values for the control variables of the experiments are listed in Table I. The experiment descriptions below specify deviations from this baseline when applicable.

1) Flow intersection angle: The first experiment is to determine the simulated conflict rates when the flow intersection angle α is varied. This is achieved by varying the flow bearing angle ϕ between 0 and 360° with a step of 2.5° and calculating the flow intersection angle according to:

$$\alpha_{i,j} = \begin{cases} |\phi_i - \phi_j| & \text{if } |\phi_i - \phi_j| < 180° \\ 360° - |\phi_i - \phi_j| & \text{otherwise} \end{cases} \tag{14}$$

In the plots, $\alpha_{i,j}$ is rounded to the nearest 10 ° to avoid clutter; 0 and 180 ° are ignored because of the zero-division $\sin(\alpha_{i,j})$ would give in Eq. (13).

2) Arrival rate: The second experiment involves varying the arrival rate λ. Flows are created at several different bearing angles; only conflicts between flows with different bearing angles are incorporated[7]. Results are plotted for the mean arrival rate, i.e. the IV is $\lambda = (\lambda_i + \lambda_j)/2$. For the highest arrival rates, the flows would be 'exhausted' sooner, so the number of aircraft per flow is doubled: $N = 200$.

3) Groundspeed: The third experiment is varying the groundspeed of the flows. Several flows are created with different groundspeeds; just as in the previous experiment, several bearing angles are chosen, but only conflicts between flows with different bearing angles are incorporated.

Results are plotted based on the difference in groundspeed between flow i and j. Groundspeed ranges between 100

[7]This is done to cancel the possible effect that choosing just one angle could have. Note that Eq. (13) accounts for the angle.

and 400 knots, with a step size of 20 knots. Results are plotted based on the difference in groundspeeds, i.e. the IV is $|V_j - V_i|$.

4) Separation requirement: The final parameter of Eq. (13) is the separation requirement S_h. Seeing as the most common values for S_h in the terminal airspace are either 3 or 5 nm, the groundspeed experiment is repeated, but setting $S_h = 5$ nm. If the formula holds, both the calculated and measured conflict rates should increase, but their ratio should remain the same as in the second experiment.

D. Results

The results are discussed based on two figures per experiment. The top figure shows the analytical conflict rates on the x-axis and the simulated conflict rates on the y-axis. The bottom figure shows the ratio of simulated to analytical conflict counts as a function of the IV; a ratio of 1 means an accurate prediction, and the spread indicates the consistency of the prediction.

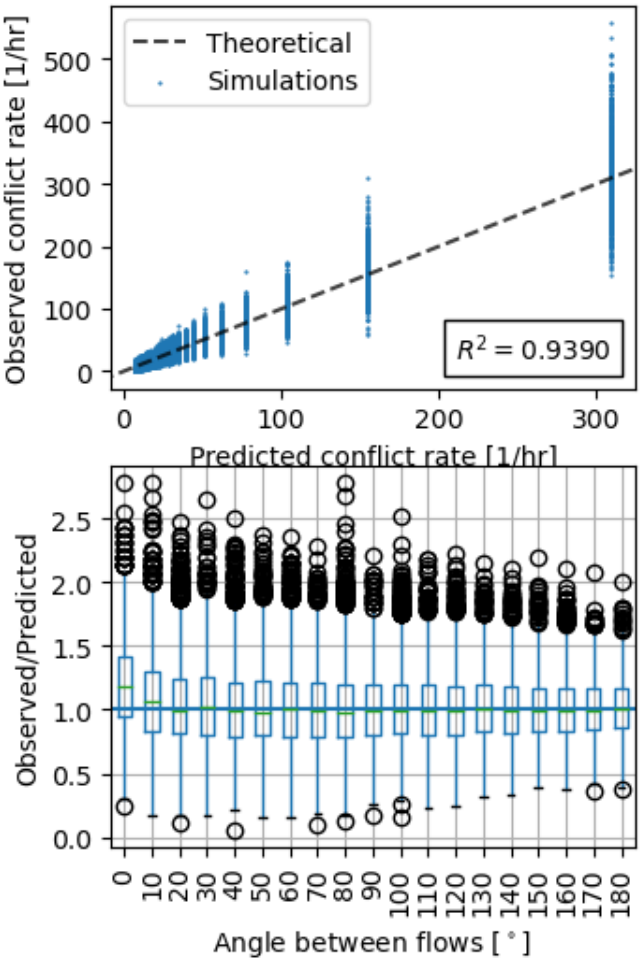


Fig. 5. Predicted and simulated conflicts for varying α

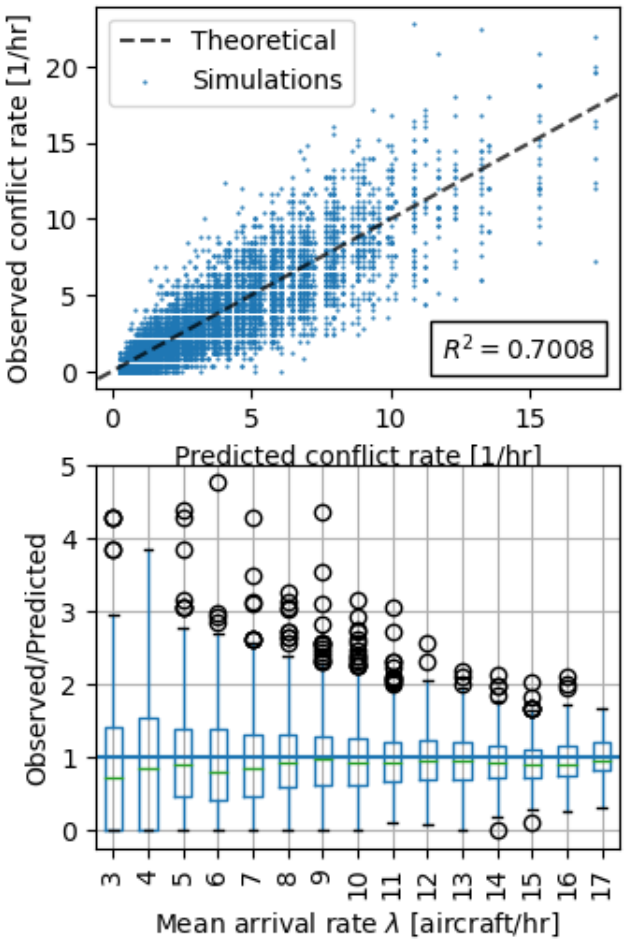


Fig. 6. Predicted and simulated conflicts for varying λ

a) Track intersection angle: The graph for the track intersection angle (Fig. 5) shows good consistency; conflict rates are very high for aircraft that are almost head-on, as can be expected.

b) Arrival rate λ: Fig. 6 shows the result for the arrival rate experiment. The simulated to analytical conflict rates converges to one for the higher arrival rates. The lower arrival rates show a higher spread and a mean (slightly) below 1, which is due to the discrete nature of observed conflicts and not unexpected: as the numbers are much lower, one or two extra conflicts have a larger effect on the ratio than for the higher numbers.

c) Groundspeed: Conflict rates show good consistency for a broad range of groundspeeds. Note that there is a significant spread, especially for the lower groundspeed differences.

d) Separation requirement: For a separation requirement of $S_h = 5$nm, the results (Fig. 8) show a similar trend as in Fig. 7.

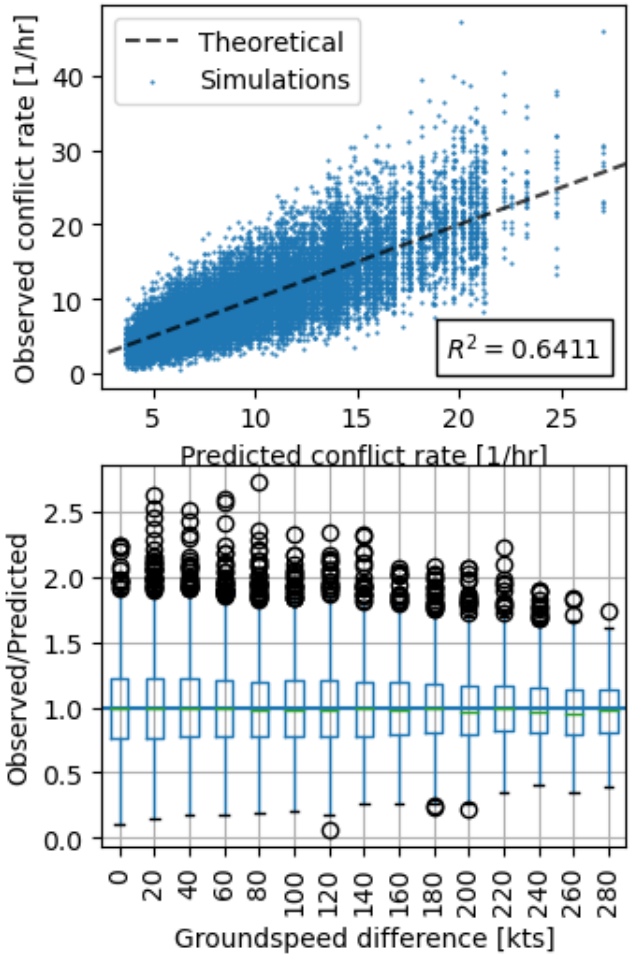


Fig. 7. Predicted and simulated conflicts for varying V, with $S_h = 3$ nm.

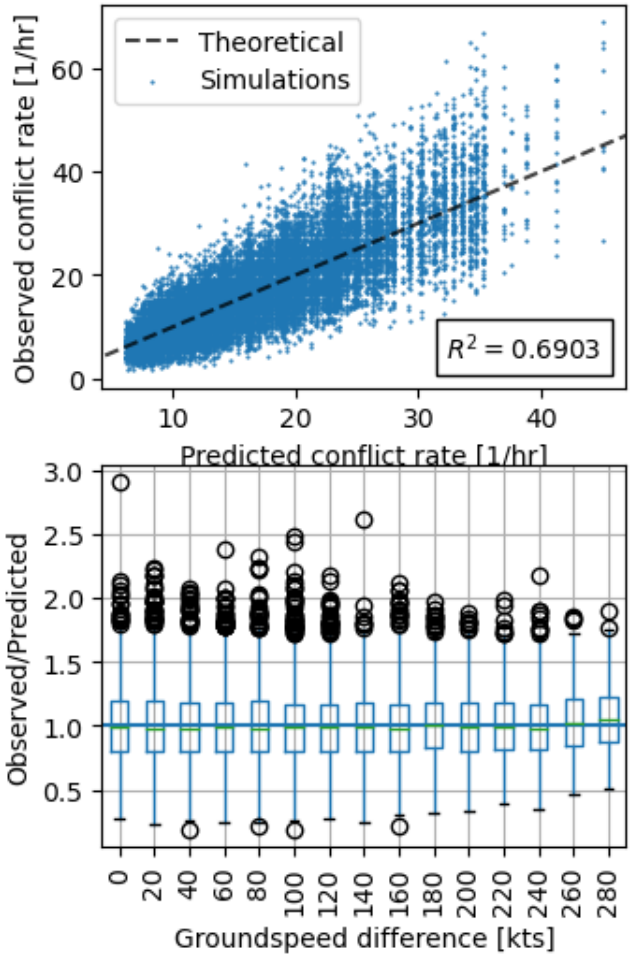


Fig. 8. Predicted and simulated conflicts for varying V, with $S_h = 5$ nm.

E. Discussion of verification

In the initial experiments, T_{settle} was just one hour and R_0 was 20nm. However, though the graph for the track intersection angle showed good consistency for the lower conflict rates, it showed worse behaviour at the higher conflict counts were a substantial underprediction took place. Initially this was thought to be real behaviour limiting the validity of the model; however, it turned out to be caused by the initial radius R_0 of 20 nm. Aircraft on opposing tracks ($\alpha_{i,j} = 180°$) had a relative velocity of 400 knots, but arrived only $2R_0 = 40$ nm away from each other. On an 'infinite' airway where aircraft would arrive at one of the ends, all aircraft would have the chance to interact witch each other, but in this case the aircraft only had approximately 0.1hr of flight time until they would have passed the spawn point at the opposing end. Increasing R_0 to 200nm solved this issue.

A similar effect occurred in the groundspeed experiment. For the initial settling time of 1 hr, the higher groundspeed differences showed an underprediction. However, this effect disappeared when changing the settling time to 2.5 hr.

F. Conclusion of verification

a) Parameters: The model accurately predicted conflict rates when varying groundspeeds, flow intersection angles, arrival rates and the separation requirement, at least for values which can be expected in the terminal airspace.

b) Length of flow segments: The model is, however, sensitive to the length of the flow segments. If the length of the flow segments is small compared to the relative velocity of enclosing aircraft, aircraft may have already crossed the intersection point before conflicts can arise, resulting in an overprediction of conflicts.

c) Transient effects: When aircraft start arriving at the start of a longer airway segment and still need some time to get to the intersection point (e.g. during an arrival peak), this too will lead to an overprediction of conflicts.

The above effects lead to either an accurate prediction or an overprediction. From a safety point of view this is desirable (since there is no underprediction that could lead to an under-allocation of resources).

V. Flow conflict prediction

To apply the conflict rate model to clustered flight track data, the data of the clusters need to be reduced such that Eq. (13) can be used. This is done in two steps:

1) Averaging the flight track data to an indicative 'mean' track with mean states (altitude, heading, velocity and vertical speed);
2) Taking representative states per flow interaction to calculate the predicted conflict rate.

A choice needs to be made for how to calculate the flow intensity rates λ_i, λ_j. What matters is the combined intensity $\lambda^2_{i,j}$, i.e. the product of the flow intensity rates *for the time period when both flow i and j are active*[8]. In an airspace synthesis scenario, flow intensity could be set as a function of time and the combined intensity can be calculated for every flow pair.

Additionally, the effect of deconflictive actions should be accounted for, including choosing an appropriate value for S_h. This is discussed in more detail in Section V-C.

Finally, unclustered flight tracks are not included in the calculations, as calculating an average would not be sensible since the unclustered tracks are by definition dissimilar. However, the interaction with the unclustered tracks is included in the replay, to give an indication of their effect.

A. Calculating predicted conflicts

The tracks in a cluster are downsampled to 200 datapoints per track (discarding about 5 percent of the tracks which had less than 200 datapoints[9]), before being combined and reduced into one mean track, having a mean position and altitude, as well as groundspeed, heading (using a circular mean), and vertical speed. Hence, each cluster i is represented by a $(200, 6)$ track matrix T:

$$\mathbf{T}_i = \begin{bmatrix} x_0 & y_0 & h_0 & V_0 & \chi_0 & \mathrm{VS}_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & h_n & V_n & \chi_n & \mathrm{VS}_n \end{bmatrix} \tag{15}$$

This is similar to the GeoVectoring concept [14], which places restrictions on allowed groundspeeds, headings, and vertical speeds, as a function of position:

$$\underline{V}_{geo} = \begin{pmatrix} [V_{min}, V_{max}] \\ [\chi_{min}, \chi_{max}] \\ [VS_{min}, VS_{max}] \end{pmatrix} = f(x, y, h) \tag{16}$$

In the current research, $\underline{V}_{geo} = f(x, y, h, i)$, i.e. the constraints are specific to that cluster i, and there is no guarantee that other clusters at similar spatial coordinates have the same constraints. Vice versa, though, the conflict rate prediction could be applied for a merge point between sufficiently narrow defined GeoVectoring airspaces.

[8]Note that the model calculates conflict *rates*; total conflict *counts* for an entire day would also need a correction for which portion of the day the flow(s) is/are active

[9]Parts of the Schiphol TMA are also used for traffic to or from Rotterdam Airport, using just a small portion of the Schiphol TMA and as such having few datapoints

Applying the conflict rate model (Equation 13) is only meaningful for converging or crossing flows. The question is which row to select for calculating the conflict rate, since contrary to the theoretical model, speed and heading cannot be assumed to be constant. Two options were considered: the first row where spatial overlap is detected, or the row where the horizontal separation has a minimum. The first of these options gave the best fit and was used, which determined how to calculate V_i, V_j, and $\alpha_{i,j}$.

For all combinations i, j of the mean tracks, it is determined for which rows (if any) there is proximity, i.e. for which rows the horizontal and vertical distance are below the separation minima S_h and S_v. Together with a distinction between between clustered tracks and unclustered tracks, the following classification is made:

1) Classification:

- `within-cluster`: If $i = j$, conflicts are within-flow. All other cases are between-clusters.

The first proximity calculation is done for *along-vector* points: for two clusters i and j, the horizontal and vertical separation between each row of $\mathbf{T}_i$ and $\mathbf{T}_j$ is checked against the separation requirements S_h and S_v, respectively, resulting in a Boolean row vector of length 200 indicating, for each row, whether the horizontal and vertical separation is below the requirements. If this is the case, along-vector proximity is registered, depending on which rows have proximity:

- `diverging_av`: Some rows, *including* the first rows. These flows diverge (e.g. departures from the same runway) and the number of conflicts (which is by nature a convergence phenomenon) is expected to be low.
- `converging_av`: Some rows, *excluding* the first row(s). These flows have convergence, and this category is expected to give the most conflicts. The index of the first row with loss of separation is used to calculate the analytical conflict rate.
- `all_av`: In this case all rows are within the separation minima, i.e. tracks are very close. A row for conflict calculation cannot meaningfully be selected.

If no along-vector proximity is registered, the proximity comparison is done pairwise for all combinations of rows, resulting in a Boolean matrix of size $(200, 200)$.

- `cross`: If this matrix has true values, flows cross for aircraft in dissimilar flight phases. The conflict rate calculation is done using the states are determined by taking the average argument of the True-values along each axis for $\mathbf{T}_i$ and $\mathbf{T}_j$, respectively.

If the pairwise comparison matrix has no true values, an extrapolation is done based on the velocities in the track matrices times the look-ahead time, resulting in matrices of *predicted* positions. The same pairwise comparison is done for these predicted positions, resulting in again a $(200, 200)$ Boolean matrix.

- `converging_lookahead`: If a predicted loss of separation occurs for rows that are close (no more than 4 rows apart), a predicted convergence is registered. Conflict rate

calculation is done based on the states in the track matrix corresponding to the first proximity row.
- `cross_lookahead`: If a predicted loss of separation occurs only for states that are more than 4 rows apart, proximity occurs for flows in different flight phases. Conflict rate calculation is done using the rows as described for `cross`.

Unclustered flights are treated separately:

- `unclustered-unclustered`: Interactions within the group of aircraft that could not be assigned to a cluster.
- `clustered-unclustered`: Interactions between the group of aircraft that could not be assigned to a cluster, and another flow.

Finally, if none of the above classifications apply, the overlap is:

- `none`: No overlap. This means no (predicted) proximity is detected. Conflicts in this category stem primarily from flows where the clustering algorithm has prematurely stopped partitioning, such that some tracks are significantly far away from the cluster mean track.

B. Comparison to replay

The calculated conflict rates are compared to a replay scenario: the original flight track data is replayed in the same simulation environment (see subsection IV-B), with some changes. In every simulation step, the latest states of aircraft are used for aircraft position and conflict determination (deactivating aircraft after about one minute after the last occurrence). The simulation and conflict calculation frequencies were increased to reflect the non-constant nature of the traffic; a trial with a simulation and conflict frequency of once per minute yielded up to a factor two difference in detected conflicts. Note that there was no extrapolation of position data: position data was updated discretely solely based on the logged positions.

Conflicts between aircraft are counted and categorised as within- or between-flow conflicts. Every unique aircraft pair has a binary conflict flag for the simulation duration, such that repeat conflicts are counted only once and a multi-conflict between three aircraft is *not* considered as a special case.

The conflict counts need to be divided by some time duration to get to conflict rates. Between-flow conflict determination is only sensible per flow pair, and options include the duration for which *both* flows are active, or the duration for which *either* flow is active. The first option is the most sensible as long as the conflict prediction is done with the combined intensity parameter $\lambda^2_{i,j}$.

C. Removing controller deconflictive action effects

The predicted conflict rates are compared to the observed conflict rates in a simulation replay. However, due to the nature of the captured mode S data, this replay data is inherently deconflicted: air traffic flow management and tactical air traffic control actions will have reduced the number of occurring conflicts substantially; either by influencing the arrival time of an aircraft or its speed (separation in time), or by changing the position or altitude (spatial separation). The interest of this research can be seen as the *conflict potential*, i.e. the ability of air traffic to cause conflicts under a certain route structure. The following options of removing the effect of the deconflictive actions are considered:

1) Increasing the look-ahead time and the separation requirements. In this way, a deconflicted aircraft pair, or an aircraft pair that has been spatially deconflicted to a separation distance of (just) over the original separation requirement, e.g. by maintaining an altitude difference of 1000ft, is counted as a conflict again.
2) Selecting a number N_{days} (e.g. 4) of comparable days (in terms of runway configuration), evenly subdividing the flights per day over N_{days} subsets, and combining the resulting subsets (in this example 16) to 4 new traffic sets, each having a quarter of the data of each original day. This preserves timing data but will lead to more conflict potential as most traffic pairs are not deconflicted. Simulations are run for each new traffic set and then averaged.
3) Considering every flight in a cluster as a separate event which can start at a time t based on the probability distribution of arrivals in that cluster (and shifting the first original occurrence to $t = 0$). This will lead to a high density at the start of the simulation which decreases as time progresses. Such a changing density would be unsuitable for an experiment measuring total conflict counts with conflict resolution and a domino effect, but conflicts are not resolved in this experiment. It is an artificial way to measure the conflict potential of a traffic scenario.
4) Increasing the density by replicating each track R times, in combination with seeing each flight as an event. This will increase the arrival rate (and reduce the time between arrivals) by a factor R.

A selection of the above options are qualitatively compared in this paper. Predicted and calculated conflict rates are compared by plotting these in a scatterplot for every conflict pair. Since it is not obvious which choice of rate and conflict time parameter is most indicative, multiple options are qualitatively considered. Of interest is not necessarily the ratio of observed conflicts to calculated conflicts, but also a measure of how much of the spread can be explained by the trendline, i.e. the *coefficient of determination*, R^2. A very low R^2 indicates little to no correlation between observed and calculated conflicts, meaning the prediction cannot meaningfully contribute to understanding the interaction between flows; a higher R^2 indicates it can.

VI. Results

This section discusses the results of the clustering, as well as the prediction and replay experiment.

A. Clustering

After inspection of the traffic patterns of the dataset, four days with comparable runway configurations (southern wind) in the beginning of Januari 2018 were selected: Jan 1-2 and 4-5. Tracks from these days were randomly and evenly divided over 4 scenario's. The input tracks for the first scenario is visualised in Fig. 9a; the resulting cluster means are shown in Fig. 9b. An example cluster is shown in Fig. 10. The clustering statistics are shown in Table II.

TABLE II
RESULTS OF CLUSTERING

Scenario	Tracks	Clusters	Tracks per cluster	Unclustered
1	982	46	20.50 ± 18.46	39 (3.97%)
2	995	40	23.75 ± 19.58	45 (4.52%)
3	991	37	24.97 ± 19.91	67 (6.76%)
4	991	40	23.62 ± 16.32	46 (4.64%)

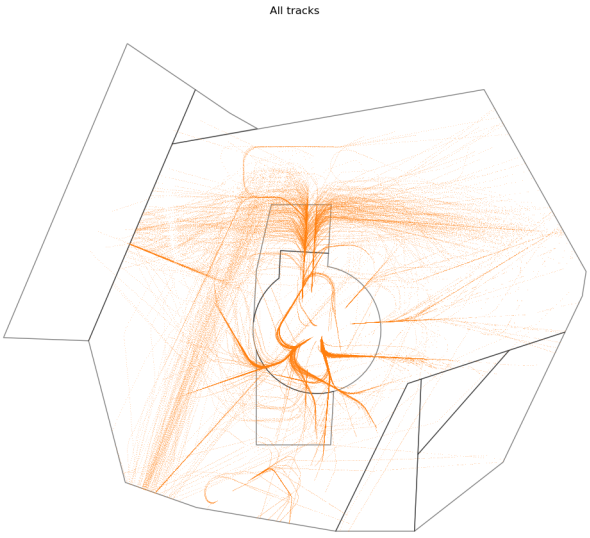
B. Conflict prediction

Considering the spread in the results of Section V-C, especially for the lower numbers, it is apparent that a lot of realisations are substantially far away from the trendline, which can only be distinguished with a lot of data. Even then, for conflict rates below 5 per hour, there is a lot of spread; the trend is most obvious for unrealistic high conflict rates (20 conflicts per hour per possible flow combination would lead to an unacceptable workload for the controller).

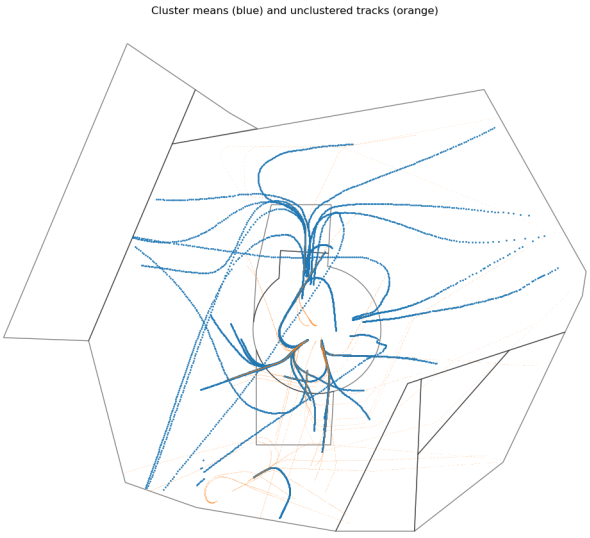
1) Unaltered replay: As a starting point, consider the conflict counts and rates for four comparable days, without any alteration to remove the effect of the controller deconflictive actions. 4 days were separately replayed using $S_h = 3$ nm, $S_v = 1000$ ft, and $t_l = 5$ minutes, after which the total conflict counts are summed, see Fig. 11. The top graph shows the absolute number of observed conflicts, summed for the four scenario days; the bottom half of the graph shows calculated conflict rates, which is the absolute number of observed conflicts per flow pair, divided by the number of hours where both flows had at least one arrival. Additionally, Table III contains the division of flow pairs with and without conflicts.

The main conflict source is converging flows. Though conflicts still exist, the rates are lower than predicted by the model, indicating the deconflicted nature of the data. Flows with no along-vector convergence also produce conflicts, but with a much lower intensity per flow[10]. Within-flow conflicts do occur; these are mostly between aircraft descending from the IAF with a high rate of descent, and aircraft on final approach, several thousand feet lower. Diverging-type interactions produce relatively few conflicts (as can be expected since these are most likely departure flows which are separated in time). Crossing flows mostly produce conflicts some time out, i.e. look-ahead conflicts.

[10]The highest conflict rate in the `none`-type is from two flows with significant outliers of some tracks w.r.t. the cluster mean, so clustering improvement would eliminate large parts



(a) all position data



(b) clustering results (mean tracks in blue) and noise (unclustered data, orange)

Fig. 9. Clustering results for the first scenario: a combination of Jan 1-2 and 4-5, 2018.

2) Removing controller effects: The first considered method of removing deconflictive action is to create four new scenarios from the four comparable days, and randomly distribute the traffic for each day evenly over the four new scenario's. The results for these replays are shown in Fig. 12a, where $S_h = 3$ nm, $S_v = 1000$ ft, and $t_l = 5$ minutes. The largest absolute number of conflicts is for `converging` flows. The `none`-type is the next largest group, but with a substantially higher number of flow pairs; these conflicts mostly arise from a cluster with dissimilar tracks, and from a small number of flights that took place during opposite runway usage. There are more `diverging` conflicts as can be expected for mixing multiple days. `within-cluster` conflicts has a higher average per

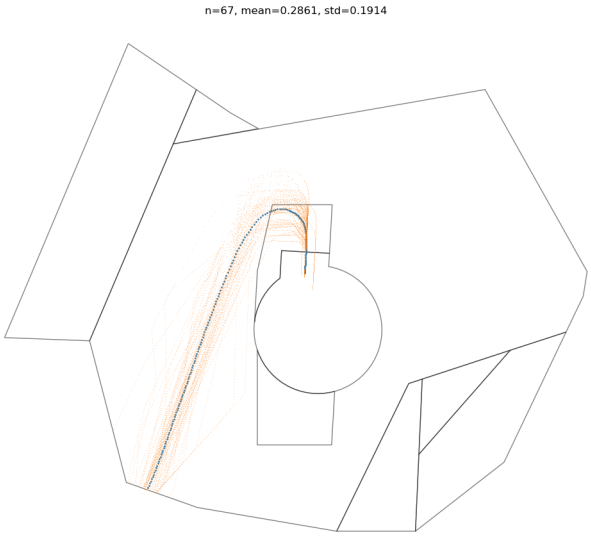


Fig. 10. An indication of one clustering result.

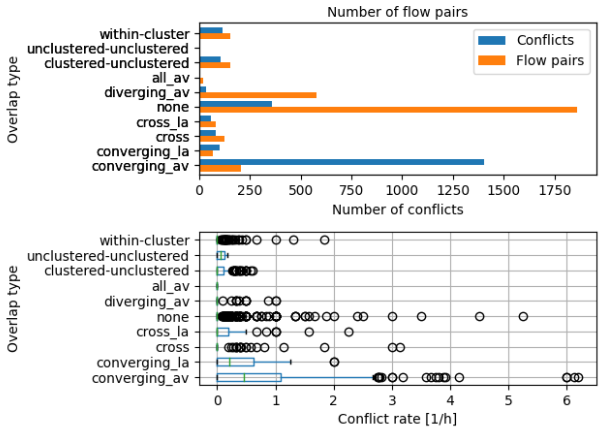
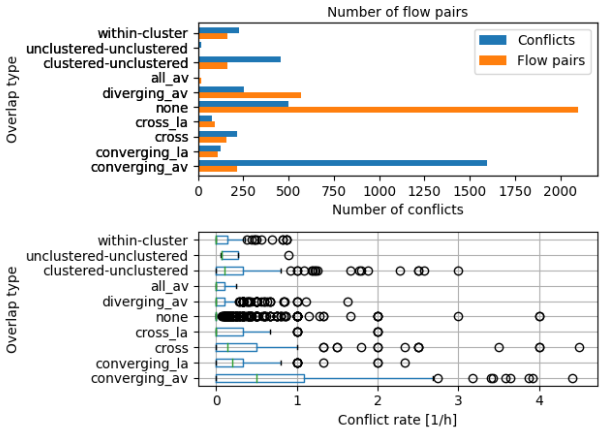


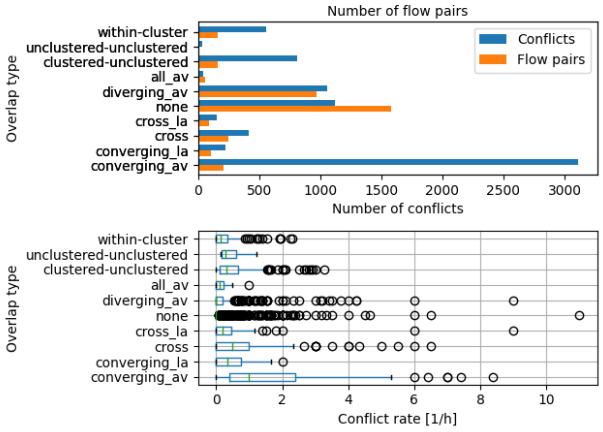
Fig. 11. Total conflict counts for unaltered replay of Jan 1-2,4-5, 2018, summed.

TABLE III
FLOW PAIRS AND SUBDIVISION
(UNALTERED REPLAY OF JAN 1-2,4-5, 2018, SUMMED).

Overlap Type	Number of flow pairs Total	With conflicts	Without conflicts
within-cluster	155	30	125
unclustered-unclustered	4	2	2
clustered-unclustered	155	47	108
all_av	22	0	22
diverging_av	580	13	567
none	1863	158	1705
cross_la	82	18	64
cross	126	19	107
converging_la	66	36	30
converging_av	207	133	74



(a) $S_h = 3$ nm, $S_v = 1000$ ft, and $t_l = 5$ minutes.



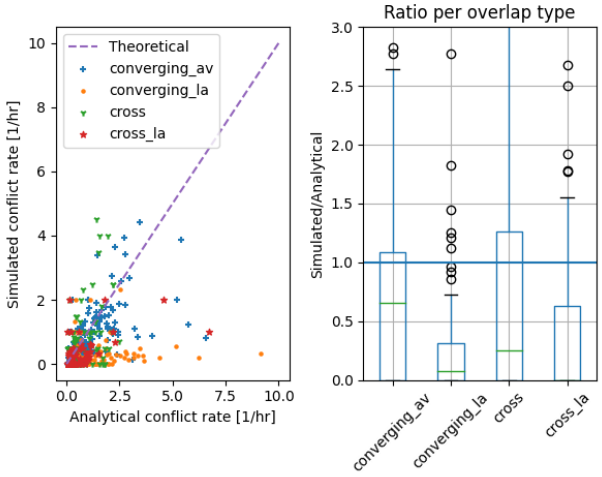
(b) $S_h = 6$ nm, $S_v = 1000$ ft, and $t_l = 10$ minutes.

Fig. 12. Conflict counts for mixed-days replay; different separation requirements.

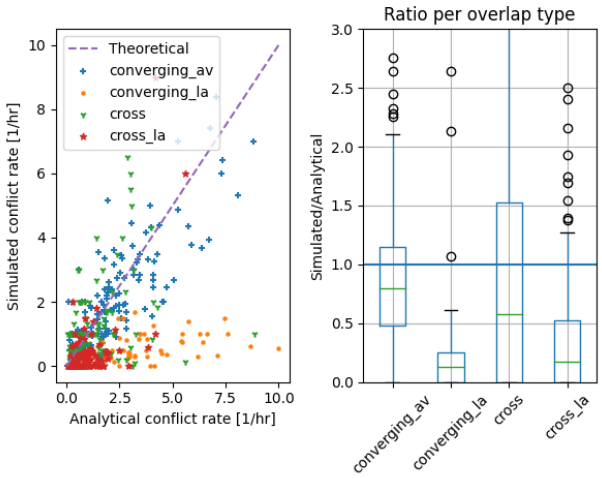
day, but still a relatively low conflict rate (with a mean close to zero). The largest conflict rates are for `converging` pairs and `clustered-unclustered` (this effect of this group may be substantially reduced by an improvement in clustering; note also the low absolute numbers). Fig. 13a shows the ratio between analytical and replay simulation conflict rates; there still is a substantial overprediction and the correlation is weak, see Fig. 14a.

Doubling both the horizontal separation requirement to $S_h = 6$ nm and the look-ahead time to $t_l = 10$ minutes increases the conflict numbers, with a pronounced effect on `diverging` and `within-cluster` type conflict counts (see Fig. 12b. Additionally, for the `converging` type conflict rates, a clear correlation between predicted and observed conflict rates can be distinguished (see Figs. 13b and 14b), with an R^2 similar to the experiments in Section IV-D. Note however that there is still a slight overprediction.

To generate more data with even less effect of controller deconflictive action, the experiment was repeated using a five-

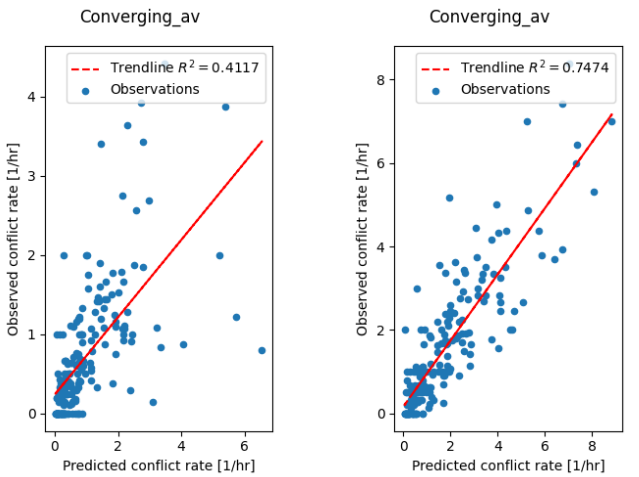


(a) $S_h = 3$ nm, $S_v = 1000$ ft, and $t_l = 5$ minutes.



(b) $S_h = 6$ nm, $S_v = 1000$ ft, and $t_l = 10$ minutes.

Fig. 13. Conflict rate prediction for mixed-days replay; different separation requirements.



(a) $S_h = 3$ nm, $S_v = 1000$ ft, and $t_l = 5$ minutes.

(b) $S_h = 6$ nm, $S_v = 1000$ ft, and $t_l = 10$ minutes.

Fig. 14. Conflict rate prediction for mixed-days replay (`converging_av`-type)

fold increase in traffic and concentrating arrivals at the start of the scenario. This results in a clear trendline at an almost 1-to-1 ratio of predicted to observed events, even though significant outliers are present; see Fig. 15. Note that some additional outliers exist in Figs. 13, 14 and 15 but that axes are truncated to increase readability.

VII. Discussion

A. Clustering

The chosen clustering algorithm can efficiently and effectively and group identify similar flows. Of attention is setting the parameters; the parameters for calculating the adjacency matrix (which states to choose and the value of K or σ) influence the choice of stop condition, which in turn influences the number of 'unclusterable' tracks, that have a considerable effect on the conflict counts. A further study of the optimal clustering parameters to maximize within-cluster similarity and minimize unclusterable tracks would likely lead to a larger proportion of conflicts that can be accurately predicted.

B. Conflict rate prediction of replay data

Using Eq. (13), conflict rates between flows can be predicted based on the mean tracks of clustered flight track data, establishing a clear trend in the correlation between predicted and observed conflict rates.

The derivation assumes that the probability distribution along the airway cross section is uniform. For modern aircraft this will usually only be satisfied for small values of airway cross sections. However, the horizontal cross section a cancels out in the final conflict count equation, as long as this dimension is equal for both airways. In contrast, in clustered flight track data, the width of a cluster may be significant and there may be differences in this width. The spread around the cluster mean may vary on track position, and the spread may be different for different clusters. The theoretical model may be improved by correcting for the ratio of cluster spread at the convergence locations.

The probability of vertical overlap between two clusters is essentially used as a binary value in this research: proximity is based on horizontal and vertical overlap, and if both are present, the calculations are made as if vertical overlap is detected. However, since the probability of vertical overlap essentially scales the conflict rate, conflict rate predictions are likely improved when the vertical overlap probability is more accurately included.

Replay conflict observations did include a spread around the mean and vertical speed differences, but though the conflict rate prediction did not take these into account, it still produced meaningful results. This shows that using cluster means based on a 2D clustering methodology can give an indication of

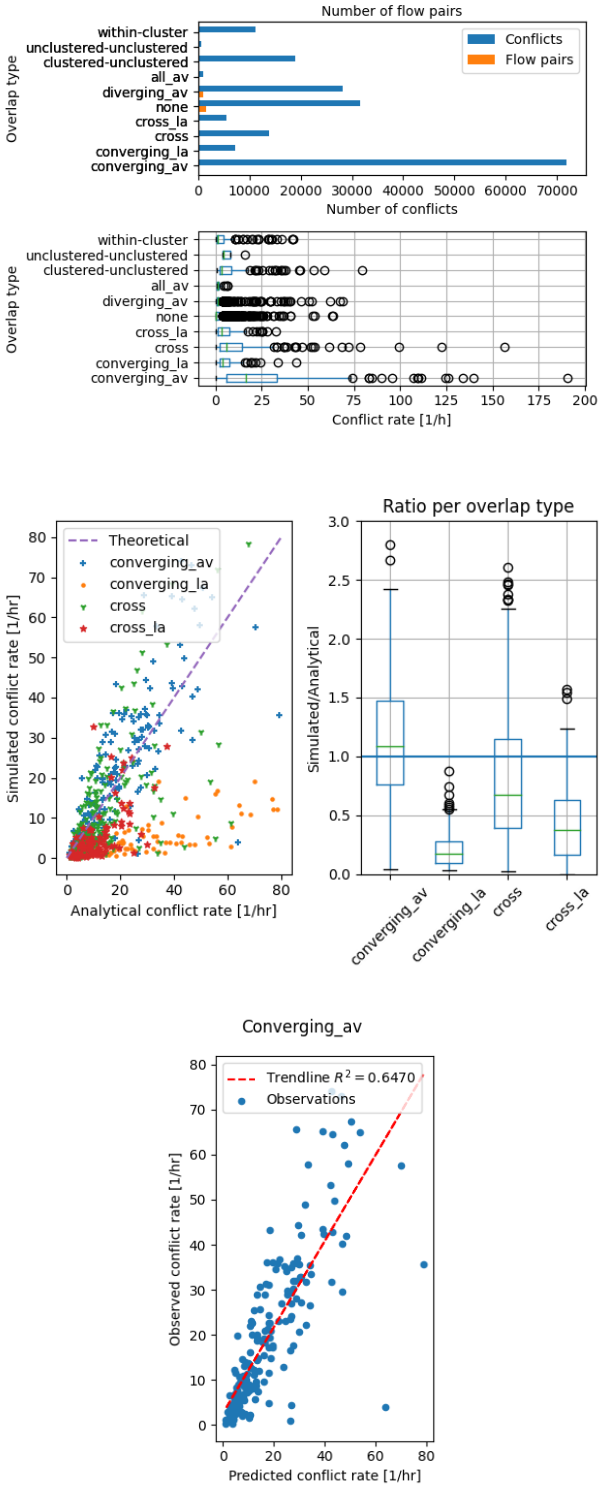


Fig. 15. Conflict counts and rate prediction for a 5-time increase in traffic (`converging_av`-type)

3D conflict rates in the terminal airspace. Additionally, it shows that logged flight track data can be used instead of standard instrument procedures when those procedures are not representative of actual traffic.

Improvements to the clustering parameters could probably reduce the number of 'unclusterable' flight tracks and thus the unpredicted conflicts that originate from these, reducing the `clustered-unclustered` and `unclusterd-unclustered` categories as well as the `all_av` category. In that case, `within-cluster` conflicts would make up a small minority of total conflicts, but they cannot be predicted by the current model. However, since the majority of `within-cluster` conflicts is between aircraft in dissimilar flight phases, it may be possible to predict within-cluster conflicts using the between-cluster methodology. Another possibility might be to use the CAMDA-framework [3] to quantify within-cluster conflicts.

1) Deconflictive actions: Using unaltered replay data leads to an overestimation of converging conflicts. Combining traffic from comparable days and increasing separation minima and lookahead time yields higher conflict rates and a slight improvement in prediction ratio; however, it does change the ratio of conflict types (e.g. much more begin-type conflicts). Both this underprediction and change in relative proportion of conflict types should be taken into account when selecting a deconflictive action method.

VIII. Conclusion

This research investigated the applicability of conflict models on controlled, structured airspace concepts using flight track clustering of logged mode S surveillance data in the terminal airspace. The collision rate model of [5], adapted for conflict rates, is valid for 2D flows across a broad range of the dependent variables when enough realisations are present; for low conflict rates the stochastic effect can be large, leading to substantial differences from the predicted values.

When IFR procedures are not followed, captured ADS-B data can be clustered into flows in the terminal airspace, even using only 2D position data, using a spatial clustering algorithm. These flow means can effectively predict conflict rates, even when taking into account 3D conflicts. Clustering results are, however, sensitive to the hyperparameters of the clustering algorithm, which can influence the conflict predictions.

The nature of retrospectively looking at logged flight tracks leads to an overprediction of conflict rates; by mixing and repeating traffic predictions have less bias, but these actions also influence the ratio of conflict types, leading to a higher incidence of e.g. conflicts in departure flows.

Improvements to the clustering algorithm could likely increase the proportion of conflicts that can be predicted, and may reduce the effect of conflicts between flows that have no proximity of mean tracks. Conflicts within a flow make up a small proportion of total conflicts; these may be predicted by the between-flow conflict rate model or the CAMDA framework, however, this requires further study.

REFERENCES

[1] Ministerie van Infrastructuur en Waterstaat, "Luchtvaart nota: Verantwoord vliegen naar 2050," Rijksoverheid, Tech. Rep., 2020.

[2] ICAO, "Procedures for Air Navigation Services. Air traffic management (Doc 4444). 16th edition." 2016.

[3] E. Sunil, J. Ellerbroek, and J. M. Hoekstra, "CAMDA: Capacity Assessment Method for Decentralized Air Traffic Control," *International Conference on Research in Air Transportation*, no. April, 2018.

[4] W. Graham and R. H. Orr, "Comments on terminal air traffic flow and collision exposure," *Proceedings of the IEEE*, vol. 59, no. 1, pp. 97–98, 1971.

[5] S. Endoh, "Aircraft collision models," Ph.D. dissertation, Massachussets Institute of Technology, 1982.

[6] J. Sun, H. Vu, J. Ellerbroek, and J. M. Hoekstra, "PyModeS: Decoding Mode-S Surveillance Data for Open Air Transportation Research," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 2777–2786, 2020.

[7] M. Enriquez, "Identifying Temporally Persistent Flows in the Terminal Airspace via Spectral Clustering," *Proceedings of the 10th USA/Europe Air Traffic Management Research and Development Seminar, ATM 2013*, 2013.

[8] M. Daszykowski and B. Walczak, "Density-Based Clustering Methods," *Comprehensive Chemometrics*, vol. 2, pp. 635–654, 2009.

[9] A. Eckstein, "Automated flight track taxonomy for measuring benefits from performance based navigation," *Proceedings of the 2009 Integrated Communications, Navigation and Surveillance Conference, ICNS 2009*, pp. 1–12, 2009.

[10] M. Fiedler, "A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory," *Czechoslovak Mathematical Journal*, vol. 25, no. 4, pp. 619–633, 1975.

[11] M. Gariel, A. N. Srivastava, and E. Feron, "Trajectory clustering and an application to airspace monitoring," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1511–1524, 2011.

[12] M. Enriquez and C. Kurcz, "A Simple and Robust Flow Detection Algorithm Based on Spectral Clustering," in *International Conference on Research in Air Transportation*, no. May, 2012.

[13] L. Zelnik-Manor and P. Perona, "Self-Tuning Spectral Clustering," in *Adv. Neural Inf. Process. Syst*, vol. 17, 2004.

[14] J. . Hoekstra, J. . Ellerbroek, E. . Sunil, and J. Maas, "Geovectoring: Reducing Traffic Complexity to Increase the Capacity of UAV airspace," *2018 International Conference on Research in Air Transportation*, 2018.

Part II

Preliminary report [already graded]

1

Introduction

The demand for air traffic generally increases steadily, for The Netherlands' largest airport, Schiphol, even reaching the previously agreed upon limit of 500.000 movements a year in 2018 [20], meaning a lot of aircraft are using air space. For safety reasons, regulators have set a minimum separation between aircraft, depending on flight type and airspace classification [16]. In the current system, it is the job of air traffic controllers to ensure that this minimum separation is not violated. This is not an easy task, as can be deducted from the Dutch Civil air traffic controller organisation (Luchtverkeersleiding Nederland or LVNL) rejecting an overwhelming majority of applicants (only 22 out of a thousand applicants pass the selection and training [18]). Of course, even the ones that do graduate have limits to their ability to ensure separation between aircraft: there needs to be a sufficient spread in the number of aircraft in the airspace for the controllers to be able to properly guide them. In the medium term, this is achieved by limiting the number of aircraft that can enter the airspace: Air Traffic Flow Management [15]. Before aircraft take-off, they can be held to ensure that once they enter a certain airspace, that airspace is not over its maximum capacity.

The starting point for all this is an airspace design. Where can aircraft enter or leave the airspace, what routes are available to them, and what are the geographical limits of the airspace that the air traffic controller needs to supervise. The Dutch government has announced that the airspace design will be revised over the coming years: as the demand for air traffic grows, more airports are intended to open, and new generation military jets need more airspace, a major rework is underway, intended to be ready by 2023 [21]. However, future concepts to deal with air traffic are already being investigated, with part of LVNL looking at concepts to use after 2030 [19].

Eventually, the question that arises is: how many aircraft can safely and efficiently use the airspace? In other words: what is a safe capacity for the airspace? With the controller being responsible for separation, they need to resolve any conflicts. A conflict is a predicted future loss of separation; if nothing would be done, aircraft would get too close. Hence, the workload of the controller is directly related to the number of conflicts. The number of conflicts, in turn, is related to both the number of aircraft in the airspace, and their probability of having a conflict. Understanding this probability of conflicts is a key step in being able to predict the number of conflicts, and therefore the capacity of the airspace.

Knowledge gap

Previous work at this university has shed light on the conflict probability of *unstructured* airspace, where aircraft can depart and arrive at any point in a given airspace [28]. For instance, future personalised air vehicles in a big city could use the airspace around that city. However, passenger aircraft are still dependent on airports, so the number of origins and destinations is limited. This gives rise to structured traffic flows and currently air traffic controllers are still responsible for deconflicting these flows. The goal of this research is to contribute to the understanding of how these structured flows impact the number of (possible) conflicts of an airspace design. A deeper understanding of these conflicts can increase both the safety and capacity of airspace design, by choosing the concept with the lowest number of (possible) conflicts.

To assess the relationship between these structured traffic flows and capacity, this research uses previous flight data, comprised of collected aircraft transponder (ADS-B) data [27]. These flight tracks need to be grouped into clusters of flows to analyse them and distill properties that can predict conflicts. The aim is then to eventually allow air space designers to synthesise a design based on routes and flow properties of aircraft in these routes, and obtaining the predicted conflict counts as a measure of the airspace capacity. Ideally, the relationship between the flow and route properties on the one hand, and airspace conflict counts on the other hand, can be analytically expressed, so that it is immediately visible what the effect is of changing a certain parameter.

Research objective

Following the research gap in the previous section, the research objective can be posed:

> *To contribute to a safe and efficient air space design process by developing an analytic conflict count prediction framework for centralized air traffic control and obtaining a relationship between properties of historic traffic flows and air space capacity, resulting in a more thorough understanding of capacity in structured air space design and more informed air space design decisions.*

Report structure

The remainder of this reports consists of a a literature review in Chapter 2, followed by the methodology in Chapter 3. Finally, Chapter 4 outlines the experiment overview and planning for the remainder of the research.

2

Theoretical background

This chapter contains the theoretical background of the research. Section 2.1 gives a general introduction to the definitions, the motivation underlying conflict count models, and the types of airspace, before formulating the research questions. Section 2.2 discusses capacity prediction and conflict count models, the differences between unstructured and structured airspace, and discussed 2D versus 3D models, before introducing the concept of flows and arriving at a conflict rate formula. Finally, section 2.3 discusses ways to cluster air traffic into flows.

2.1. Introduction

Airport systems are resource constrained: resources like gates and taxiways have limited capacity. The take-off and landing of aircraft imposes interval constraints between these events to meet wake turbulence separation requirements. Outside these constraints, the driving limitation to capacity are the safety constraints: the introduction of radar surveillance lead to radar separation minimums: 3 or 5 nautical mile horizontally, 1000 or 2000 feet vertically[10], depending on the type of airspace.

2.1.1. Definitions

Visualise a cylindrical ”box” with a radius of 3 nautical mile and a height of 2000 ft, centered around the aircraft. This box should remain clear of other aircraft at all times and is called the Protected Zone [25]. This is because when another aircraft *intrudes* this box, it means that the horizontal distance is less than 3 nautical mile and the vertical distance is less than 1000 feet, thus violating the separation minimums. For cylindrical Protected Zones, this can equivalently be regarded by visualising a box with halved dimensions (a radius of 1.5 nautical mile and a height of 1000ft), where no *overlap* between these should exist. To avoid a collision, a key objective of Air Traffic Management is ensuring that no ’intruder aircraft’ enters these Protected Zones.

A **loss of separation** is *”the situation where the distance between two aircraft is less than the separation minima”* [11] (and, consequently, vice versa). A **conflict** is *”a predicted loss of separation”* [11]: at time t, a conflict is registered when in the time interval $[t, t + t_l]$ (where t_l is the look-ahead time, for instance 5 minutes), a loss of separation is predicted. An acceptable risk is the **Target level of Safety** (TLS), set by the International Civil Aviation Organisation (ICAO) as s 2.5×10^{-9} collisions per flight hour [14].

Conflict count models

When a conflict occurs, it should obviously be resolved to prevent it from developing into a loss of separation, or ultimately, a collision. This conflict resolution can be done centralized, by an Air Traffic Control entity, or decentralized, by each aircraft separately. For both approaches, a lower number of conflicts is beneficial: not only do fewer conflicts lead to less potential collisions, fewer conflicts to resolve also reduce the workload on the entity performing the deconfliction, whether that is a pilot or an air traffic controller. For this reason, a lower number of conflicts lead to a safer and more efficient airspace; or, conversely, if two airspace designs have a different number of conflicts for the same

throughput of traffic, the concept with the lowest number of conflicts is inherently more safe and efficient. At the same time, the focus on safety may lead to inefficient usage of airspace.

This is the reasoning for adopting *conflict count* models: models relating the number of conflicts (or the number of conflicts per unit of time, the *conflict rate*) to some parameters of the airspace.

2.1.2. Types of airspace

An important characteristic to consider is the nature of the traffic: can traffic originate and end at any point in the airspace considered, or are there certain paths relating entry and exit points within the airspace? These are, respectively, *unstructured* and *structured* airspace concepts. A lot of the current airspace can be considered structured due to the use of airways: 'highways' in the sky consisting of several waypoints that aircraft navigate consecutively, especially in the *en-route airspace*, where aircraft are not on a departure or arrival segment but somewhere in between, usually at their cruising altitude. An additional feature of structured airspace is the increased possibility of *overtaking conflicts*, a conflict occurring between two aircraft on the same airway and with a nonzero relative speed. These aircraft may be separated in altitude to prevent an overtaking conflict from occurring [11]. These overtaking conflicts (conflicts between aircrafts on the same route) should therefore also be included in a model.

A typical flight trajectory will have an aircraft take-off at a (controlled) airport, entering the Control zone (CTR), a volume of airspace around an airport, usually cylindrical, constrained in both altitude and distance from the airport[1]; air traffic control in the CTR is in principle done visually by the Tower controller. When leaving the CTR, aircraft continue climbing in the Terminal airspace (TMA), where an Approach/Departure controller controls both arriving and departing traffic in the vicinity of the airspace. After leaving this airspace, aircraft enter the en-route airspace and spend a large portion of their flight in this airspace; close to the destination, they will descend and enter the TMA and CTR again.

2.1.3. The terminal airspace

While some predetermined routes exist in the terminal airspace, at busy airports these are not always used, especially not during the arrival peaks. Instead, tactical deconfliction is used (so-called *vectoring*) to maximise throughput. This means that for these cases, only considering predetermined arrival routes does not lead to sensible results. As an alternative, this research looks at observed flight track data, in which patterns emerge. These patterns can be obtained by clustering similar flight track data.

Research at this university has produced a framework to relate (geometrical) properties of an *unstructured* airspace concept to its capacity by using conflict count models as intermediate step. In this way, a direct, analytical model of airspace capacity was obtained, allowing a direct inspection of the consequences of changing a certain parameter. The goal of the current research is to develop a similar technique for *structured* airspace, with a focus on the terminal airspace.

2.1.4. Research questions

Following this introduction and in line with the research objective formulated in Chapter 1, the following main- and sub- research questions can be formulated:

1. Can historic flight track data properties be used to predict structured air space conflict counts?
   (a) What properties of traffic flows can predict air space capacity?
   (b) How can historic flight track data be aggregated into *flows*?
   (c) What properties do these flows have?
2. What is the relation between the traffic flow properties and simulated airspace conflict counts?
   (a) How can conflict counts be predicted analytically by traffic flow properties?
   (b) How can conflict counts be obtained from simulations?
   (c) How do the the conflict counts obtained from simulations compare to the analytically predicted conflict counts?
3. How does the conflict count prediction based on traffic flow properties compare to the conflict counts by fast-time simulations or replays?

[1] Typical values for The Netherlands are a radius of 8 nautical miles around the airport, sometimes with an additional elongation along the center axis of a (dominant) runway, and vertically from the ground level to an altitude of 3000 feet

   (a) How can the effect of controller deconflictive action be mitigated?
   (b) What conflict counts are predicted by fast-time simulations?
   (c) How similar are these replay conflict counts compared to the analytically predicted conflict counts?

4. By how much can the conflict counts of an existing airspace design be reduced by analysing high-conflict counts flows?

   (a) What baseline scenario is representative?
   (b) What are the high-conflict count flows in this scenario?
   (c) What properties of these high-conflict count flows have the highest impact on these high conflict counts?
   (d) What change in these properties reduces the predicted conflict counts of the scenario?
   (e) Is the change in conflict count predicted accurately after altering these properties?

The research objective can be divided into several sub-goals:

G_1. Analyse historic tracks in the Dutch airspace to obtain clusters of tracks ('flows') and extract predictive properties of those flows.
G_2. Obtain the relationship between properties of flows and the conflicts between these flows.
G_3. Verify and validate the relationship between properties of flows and the conflict counts between these flows.
G_4. Evaluate the predicted capacity with respect to a baseline capacity of an (existing) airspace design.

The remainder of this chapter will consist of a description of the capacity prediction in unstructured and structured airspace in Section 2.2 and an overview of clustering methods in Section 2.3 , followed by Section 2.2.9, relating traffic flow properties to conflict count models.

2.2. Capacity prediction

This section will focus on capacity prediction using conflict count models. As a starting point, the capacity prediction framework for *unstructured* airspace will be explained. Next, the differences with regards to *structured* airspace will be reviewed.

2.2.1. Capacity prediction in unstructured airspace

In unstructured airspace, *"no constraints are imposed on aircraft motion"* and *"aircraft can … fly with preferred speeds and at optimum altitudes"*, resulting in a relatively uniform distribution of traffic [28]. While this reduces traffic concentrations, when two aircraft do near each other, there is no constraint on their relative heading or speed difference. [28] showed that, for aircraft at the same altitude (a 2D-case) in such an unstructured concept, the total number of instantaneous conflicts $N_{\text{conflicts,instantaneous}}$ is the product of the the average conflict probability p_{2d} and the number of combinations of two aircraft (with a total number of instantaneous aircraft of $N_{\text{aircraft,instantaneous}}$).

$$N_{\text{conflicts,instantaneous}} = \binom{N_{\text{aircraft,instantaneous}}}{2} p_{2d} \tag{2.1}$$

The upper bound on the number of conflicts is $\binom{N_{\text{aircraft,instantaneous}}}{2}$; p_{2d} is a probability (so between zero and one) and thus can be thought of as a *scaling* of this upper bound.

This average conflict probability p_{2d}, in turn, is the ratio of of the area searched for conflict A_c to the total airspace area A_{total} (see Fig. 2.1). For fixed values of S_h, t_l and A_{total}, it follows that p_{2d} is proportional to the expected value of the horizontal relative velocity between two aircraft:

$$p_{2d} = \frac{A_c}{A_{total}} = \frac{2S_h \mathbf{E}(V_{r,h}) t_l}{A_{total}} \propto \mathbf{E}(V_{r,h}) \tag{2.2}$$

The horizontal relative velocity $V_{r,h}$ depends on the magnitudes of the velocities of an aircraft pair as well as their respective heading difference. For aircraft i and aircraft j with velocities V_i and V_j and a heading difference $\Delta\psi$, their horizontal relative velocity $V_{r,h}$ can be calculated using the cosine rule:

$$V_{r,h} = \sqrt{V_i^2 + V_j^2 - 2V_i V_j \cos(\Delta\psi)} \tag{2.3}$$

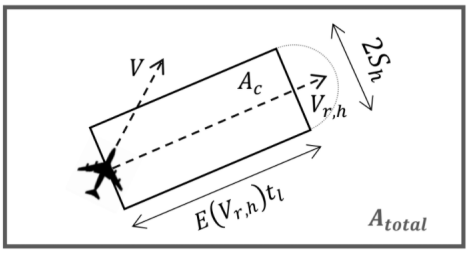


Figure 2.1: Area searched for conflicts A_c in 2D airspace. Illustration taken from [28]

A special case is the situation where the velocities can be assumed constant for all aircraft ($V_i = V_j = V_0$) and aircraft headings are uniformly distributed between 0 and α. In this case, due to symmetry $V_{r,h}$ reduces to $2V_0\sin(\Delta\psi)$, and the expected horizontal relative velocity becomes [11]:

$$\mathbf{E}(V_{r,h}) = \frac{8V_0}{\alpha}\left(1 - \frac{2}{\alpha}\sin\left(\frac{\alpha}{2}\right)\right) \tag{2.4}$$

From Equation 2.4 combined with Equation 2.2 it can be seen that reducing either the speed V_0 or the maximum heading difference reduces the number of conflicts. Conversely, with larger velocities or velocity differences and with larger heading differences, the number of conflicts increase.

2.2.2. Airspace stability

When a conflict occurs, evasive action needs to be taken to prevent the conflict from evolving into an actual loss of separation. This is called **conflict resolution**. For instance, when an aircraft threatens to 'overtake' a preceding aircraft, this can be prevented either by means of a velocity change (by increasing the groundspeed of the preceding aircraft or decreasing the groundspeed of the latter aircraft), or restoring the seaparation by increasing the lateral or horizontal distance between the aircraft (by changing altitude or by changing course).

Such an evasive maneuver may lead to a new conflict with a different aircraft, which in turn may lead to yet *another* conflict, and so on. Such a *domino* effect can be expressed by the *Domino Effect Parameter* or *DEP* [1], a measure of the ratio between the number of conflicts with conflict resolution $C_{total,wr}(\rho)$ and the number of conflicts with no conflict resolution $C_{total,nr}(\rho)$[2] and can be thought of as *"number of secondary conflicts per primary conflict"* [28, section 5.2.2]. The *DEP* depends on traffic density ρ: one can imagine that with low traffic densities, the odds of a new conflict happening are very low; the more congested the airspace is, the higher these odds are, and the subsequent deconfliction step may lead to an additional conflict. In this case, there is a *conflict chain reaction*.

$$DEP = \frac{C_{total,wr}(\rho)}{C_{total,nr}(\rho)} - 1 \tag{2.5}$$

2.2.3. Capacity Assessment Method for Decentralized Air Traffic Control (CAMDA)

The *DEP* relates to the notion of airspace **stability**: *the propagation of conflict chain reactions*. Sunil posed the *Capacity Assessment Method for Decentralized Air Traffic Control (CAMDA)*, which defines the maximum capacity limit ρ_{max} as *"the traffic density at which conflict chain reactions propagate uncontrollably throughout the entire airspace"*: [28, Chapter 5]

$$\lim_{\rho \to \rho_{max}} \frac{\mathrm{d}DEP(\rho)}{\mathrm{d}\rho} = \infty, \text{ where } \rho_{max} \equiv \text{capacity} \tag{2.6}$$

In short, the CAMDA framework consists of three steps relating to conflict counts *without* conflict resolution (CR), then applies two assumptions, and takes an additional three steps relating to conflict counts *with* resolution to arrive at an analytical formula for the Domino Effect Parameter (DEP). Then, using Equation 2.6, an analytical formula for the capacity can be found.

Though this framework cannot be directly used for structured airspace, the idea of an analytical formula for capacity is very appealing: it provides a clear insight into which factors contribute in which

[2] Note the dependence on the airspace density ρ

way to airspace capacity, and (even though small correction factors are required in certain cases) at least allows the comparison of different concepts with regards to their capacity.

It should be noted that ρ_{max} may not be a realistic or suitable measure for centralized air traffic control. In the decentralized case, it provides a saturation limit; but in the centralized case, the workload of the air traffic controller is a limit at a lower density. This may be deduced from the nature of Equation 2.6; even before conflict chain reactions propagate uncontrollably throughout the airspace, it is impossible for the controller to deconflict such a high number of aircraft pairs, let alone have the time to (verbally) give instructions for these deconflictive actions. For centralized air space concepts, the shapes of the curves relating density to conflicts for different designs should already allow a comparison between these designs.

2.2.4. Application of the 2D model to structured terminal airspace

As a starting point, reconsider the 2D-model of Section 2.2.1. This can be regarded as a portion of the airspace where only cruising flights occur. A realistic airspace will also have climbing and descending traffic, and possible conflicts between these 3 types of traffic need to be incorporated. Nevertheless, the 2D-model may have an application to the terminal airspace in a structured airspace.

Consider the case of an airport with arriving and departing traffic, where:

(i) the routes that arriving aircraft and departing aircraft take are sufficiently separated, so that no conflict between arriving and departing aircraft will occur,
(ii) all arriving aircraft have a similar altitude profile and speed, and
(iii) all departing aircraft have a similar altitude profile and speed.

The altitude profiles of arriving and departing traffic differ not only in sign, but also in magnitude (departing traffic climbs faster than arriving traffic descends); additionally, the fact that aircraft usually take-off and land against the wind means that close to the ground, departing traffic will be in the upwind sector, and arriving traffic will be in the downwind sector. Combined, this means that even when departing and arriving traffic approach each other laterally, their vertical separation will most likely already exceed the minimum.

Under the above assumptions, the traffic scenario is equivalent to two layers of traffic: one layer for departing traffic, one layer for arriving traffic, and no interaction between these layers. In this case, the number of conflicts in the terminal airspace can be calculated by summing the number of conflicts in each of these 2 'layers', predicted by Equation 2.4.

When assumption (i) is invalid, this can be compensated by adding an interaction term between the two layers. When the speed of aircraft within a layer is not constant, this may not be a problem when they accelerate or decelerate at similar locations within that layer. However, when two aircraft at similar locations within the layer have a difference in speed, the analytical equation 2.4 may not be used and a correction for the differences in speed and $\mathbf{E}(V_{r,h})$ may be determined numerically. Additionally, a separate term for conflicts *within* a layer may be needed.

Even though the above assumptions may not always hold, they provide a reasoning for considering conflict probability in the terminal airspace by means of conflict count models.

2.2.5. Conflict count models in structured airspace

Several conflict count models have been made for structured airspace. A distinction needs first to be made between *collision* and *conflict* count models: collision models predict the number of collisions that occur. Conflict count models focus on the number of *predicted* losses of separation. In other words, a collision is when aircraft physically come into contact with each other, a conflict is when aircraft are predicted to have a separation less than the minimum separation *at some time in the future*. The nature of collision risk research is different from the focus of the current work; focusing e.g. on path-keeping errors, possibly in areas with little radar coverage (such as oceanic paths). In contrast, in the terminal airspace radar coverage will usually be available.

[23] and [13] focus mainly on collision risk, making assumptions about aircraft dimensions and path-keeping errors. [5] calculates conflict intensity for a disk-shaped space around an airport, where aircraft can originate from all directions and head directly towards the center of the disk. However, this is incompatible with the practice of concentrating arriving aircraft on several Initial Approach Fixes (IAF's), where aircraft can only enter the terminal airspace at a limited number of locations (for Schiphol, e.g., there are 3: RIVER, ARTIP, SUGOL). [22] focuses on the en-route airspace, identifying crossing points

between (segments of) airways coupled with traffic data to identify collision hot-spot locations and using K-means clustering to get from collision risk points to Hot-spots. This assumes knowledge of intermediate waypoints of the airway, which can be valid for en-route airspace but is not always the case for traffic in the terminal airspace. Additionally, identifying the number of clusters (an input to the K-means algorithm) is achieved by a computationally expensive trial-and-error method.

This research proposes a different approach. Consider again the departing and arriving 'layers' of aircraft discussed in Section 2.2.4. Now, assume only that the traffic in each of the layers all have similar trajectory and velocity properties, so that the number of conflicts can easily be calculated using the 2D-model. The total number of conflicts is then the sum of the conflicts in each layer, plus an interaction between these layers. However, since the traffic in each layer is similar, the interactions between two layers will also have similar characteristics, so the conflicts between these layers can also be calculated relatively straightforward.

Flows

Instead of regarding separate layers, it is more appropriate to consider the arriving and departing traffic as a special case of a situation with multiple *flows* of traffic, where the traffic in each flow is similar with respect to other traffic in that flow. A *flow* can be thought of as the 3D-trajectory traced by a group of aircraft, with similar lateral tracks and corresponding altitude profiles.

Alternatively formulated, the special case of only departing and arriving traffic can be extended to a less restrictive model where multiple flows of traffic exist in an airspace. For instance, consider an airport with one departure runway on a given day, Standard Instrument Departures from that runway to three separate waypoints. This can be regarded as three separate traffic flows. It is relatively safe to assume that properly designed departure routes 'fan out' and will not cross each other, and that the aircraft are separated in time by spacing their departures. In this case, there will be no conflicts between the different departure routes, so there will not be conflicts *between* flows. However, different aircraft may have different velocities, so there can be conflicts *within* a flow. Conversely, aircraft arriving from multiple IAF's and landing at the same runway will merge, so in that case there may be conflicts *between* flows.

Flows may intersect both horizontally, vertically, or both. For instance, consider two sets of departing aircraft that trace the same lateral trajectory, but where the first set accelerates at a lower altitude before climbing with a higher flight path angle than the second set; in that case their corresponding flows will diverge, converge, overlap, and diverge again. A second example is the case of aircraft arriving from multiple IAF's and flying towards the same runway; these flows will first converge and then have complete overlap in the final approach phase, at least spatially.

As long as the properties of traffic in two flows are *similar enough* to each other, all possible interactions between these flows will have similar characteristics as well. By cleverly grouping aircraft in flows with similar profiles, the situation where the assumption of *similar enough* properties is valid, can be achieved[3]. Besides the horizontal and vertical trajectory of a flow, other flow properties need to be known as well; for instance, the spread around the mean[4] and a measure of how frequently the flow is used.

The conflict probability discussed so far is limited on 2D conflicts. Of course, real traffic in the terminal airspace inevitably has a 3D nature. [28] considers combinations of cruising, climbing and descending aircraft based on the proportion of cruising aircraft to the total number of aircraft and sums the result. However, assuming all climbing and descending aircraft have a similar climb and descend altitude profile (with respect to lateral track distance) respectively, this third dimension may be removed. By further assuming all aircraft in the terminal airspace are either cruising or descending, and that cruising and descending lateral tracks have sufficient separation, the total number of conflicts may be approximated by assuming 2D models and summing the results of climbing and descending conflict counts.

[3] In the limit, the situation where interactions between every possible combination of aircraft is treated separately, is achieved.

[4] For very 'narrow' flows, e.g. all aircraft are close to the mean, this spread may not directly show up in the equations, but it is more of an assumption

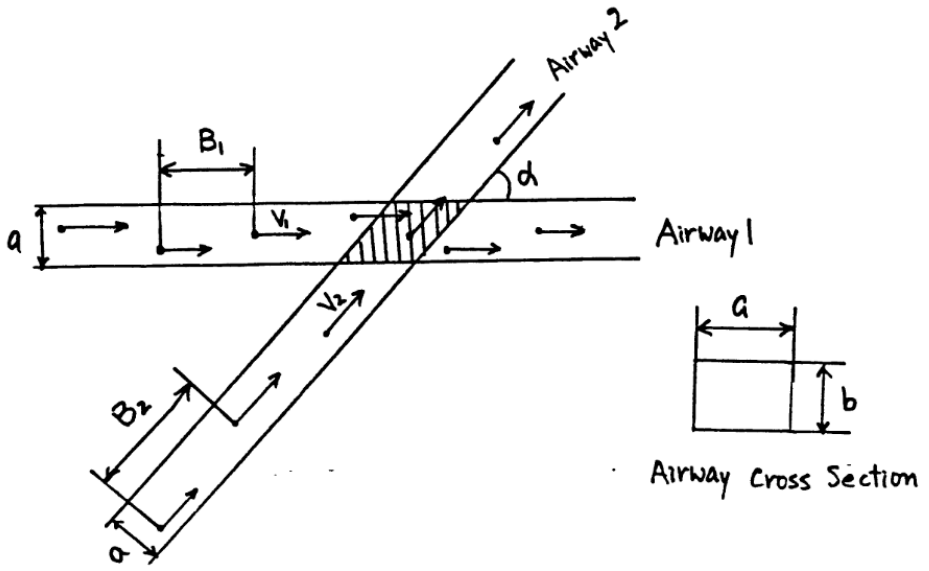


Figure 2.2: Geometry of the intersection of airways or flows. Figure taken from [5]

2.2.6. Conflicts *between* flows

For structured airspace, Equation 2.2 does not hold, as the geometry of the intersection needs to be taken into account. Consider Figure 2.2, showing two airways regarded as rectangular tubes with height b and width a, crossing at an angle α. Aircraft on airway 1 and airway 2 have an airspeed of V_1 and V_2 and an average along-track separation of $\mathsf{E}(B_1)$ and $\mathsf{E}(B_2)$, respectively. It is assumed that the probability distribution of aircraft on the airway cross section (so along a and b) is uniform.

First, consider the *collision* risk of an aircraft on airway 2 entering the intersection (the shaded area in Figure 2.2). Aircraft are modelled by a cylinder with a diameter of $2g$ and height $2h$ (g and h are assumed to be sufficiently small compared to a and b). The expected number of aircraft on airway 1 in the intersection in a unit of time is given by the product of the rate of horizontal overlap and the probability of vertical overlap [5]:

$$\frac{2g\mathsf{E}(V_{r,h})}{E(B_1)a}\frac{2h}{b} \tag{2.7}$$

To calculate the collision rate, the expected number of aircraft on airway 1 needs to be multiplied with the expected number of aircraft on airway 2, which is:

$$\frac{1}{a\mathsf{E}(B_2)}\frac{a^2}{\sin(\alpha)} \tag{2.8}$$

(In other words, the expected number of aircraft in the skewed side of the parallelogram, consisting of an along-track distance of $\frac{a}{\sin(\alpha)}$, is the ratio of that length and the expected along-track separation.)

Combining Equation 2.7 (replacing the second term with the more general $P(\text{vertical overlap})$ with Equation 2.8, the *collision rate* C in the intersection is obtained:

$$C = \frac{2gV_{r,h}}{\mathsf{E}(B_1)\mathsf{E}(B_2)\sin(\alpha)}P(\text{vertical overlap}) \tag{2.9}$$

Looking at aircraft on the same altitude (e.g. $P(\text{vertical overlap}) = 1$ and comparing this result to Equation 2.2 highlights some key differences. (Note the difference between *conflict* **probability** - explaining the t_l term - and *collision* **rate** - including total number of aircraft and factoring out time). First, $V_{r,h}$ is not stochastic but a deterministic value (since V_1, V_2 and α are assumed to be constant); the stochastic elements are the along-track separations. Second, α is not just an input to $V_{r,h}$ but also occurs in the denominator.

In essence, the stochastic element is different, and the area where a collision can occur is constrained to the overlap area between the airways, introducing the $\sin(\alpha)$ term in the denominator. [5]

includes a discussion on the extension of the model to *conflict* rates; simply replacing the aircraft dimension with the dimension of the protected zone violates the assumption $g \ll a$ and introduces a boundary problem, but the author argues that the same equation can be used nonetheless. A more thorough study of these effects are needed for the final thesis work. However, as a preliminary validation step, simulations on conflict counts for airways with different separations, speeds, and intersection angles show a linear relation between the number of conflicts predicted by Equation 2.9 and the observed conflicts, see Section 3.3.

Going from collisions to conflicts[5] also draws attention to the aircraft dimension g. [6] Note that the horizontal overlap for *collisions* is defined to take place 'when the horizontal coordinate of the center of one aircraft enters within a range of g from the horizontal coordinate of the center of another aircraft' [5]. A loss of separation occurs when the horizontal distance between the centers of two aircraft are less than S_h away. So, going from collisions to conflicts, g is replaced by S_h.

Revised conflict rate formula

For later purposes, Equation 2.9 is repeated in a revised way, replacing g with the horizontal separation requirement S_h, assuming a Poisson process for the arrivals so that $\mathsf{E}(B_{flow}) = \frac{V_{flow}}{\lambda_{flow}}$, and substituting the full Equation 2.3 for $V_{r,h}$. The resulting equation to obtain the conflict rate between flow i and flow j, intersecting at an angle α, is Equation 2.10.

$$C = \frac{2S_h\sqrt{V_i^2 + V_j^2 - 2V_iV_j\cos(\alpha)}}{\frac{V_i}{\lambda_i}\frac{V_j}{\lambda_j}\sin(\alpha)} P(\text{vertical overlap}) \tag{2.10}$$

Numerical example

As an example, consider two flows of aircraft at the same altitude (so the probability of vertical overlap is 1) in an airspace with a separation requirement of 3 nm. A conflict occurs when cylinders (or, because of the 2D-nature due to the same altitude, circles) with a radius of 3 nm, overlap (e.g. $S_h = 3[nm]$). Assuming a Poisson process[7], $\mathsf{E}(B_{flow}) = \frac{V_{flow}}{\lambda_{flow}}$. If $V_1 = V_2 = 250$ [knot], $\lambda_1 = \lambda_2 = 5$ [aircraft/hr] and flows intersect at an angle of 15 degrees, the resulting conflict rate is $C = \frac{2\cdot 3\cdot\sqrt{250^2+250^2-2\cdot 250\cdot 250 cos(15°)}}{\frac{250}{5}\frac{250}{5}sin(15°)} = 0.60$ [aircraft/hr]. This is the baseline, to which we can compare values obtained by modifying one parameter at a time:

- An increase in **intensity** in one airway (e.g. $\lambda_1 = 10$ [aircraft/hr] results in a proportional increase to $C = 1.21$ [aircraft/hr]; if both airways double in intensity, the effect is quadratic to $C = 2.42$ [aircraft/hr].
- Doubling or quadrupling of the intersection **angle** to 30 or 60 degrees has a much smaller effect: conflict rates of $C = 0.62$ and $C = 0.69$ [aircraft/hr], respectively;
- If the average **speed** of both flows is reduced by half to a speed of 125 knots, the resulting conflict rate is $C = 1.21$ [aircraft/hr] Conversely, if the average speed of *just one* flow is reduced by half, the relative velocity increases substantially and the resulting conflict rate is $C = 2.47$ [aircraft/hr].

2.2.7. Along-track separation

The two types of expected along-track separation $\mathsf{E}(B)$ considered by [5] are a deterministic constant value, and the expected value of a Poisson process. A justification for the Poisson process is given in [3][8]. Consider a hypothetical random variable H_j, taking the value 1 if aircraft pair j conflict with each other and zero otherwise, with $P(H_j = 1) = p_j \ll 1$. Then consider the sum of H_j for all h possible aircraft pairs that are at the same altitude and assume the H_j are mutually stochastically independent. In the limit, the total number of conflicts (the sum of all H_j) is a random variable approximating a Poisson distribution with intensity $\lambda = \sum_{j=1}^{j=h} p_j$.

[5] Technically, a loss of separation, but for flows with constant heading and speed, a conflict - which is a a *predicted* loss of separation - will inevitably turn into a loss of separation.

[6] [5] treats the equivalence of conflicts and collisions in a special note in their work.

[7] If flow management has lead to a uniform spacing of aircraft, the result is the same; then the expectation operator in $\mathsf{E}(B_{flow})$ collapses to this uniform spacing, which is then a deterministic value

[8] [5] refers to [3] when discussing collision rates, stating that the conflict probability equation used by [3] (arriving at a formula for the time between conflicts) is a special case of the collision rate equation (similar to Equation 2.9)

2.2.8. Conflicts *within* flows (overtake conflicts)

Apart from conflicts between flows, conflicts within flows need also be considered. These within-flow conflicts can occur when the speed of aircraft within a flow are not all equal. Under cruise conditions, the optimal airspeed may differ substantially for different aircraft. In the terminal airspace, ATC requests for specific speeds are not uncommon, which can reduce these type of conflicts. However, when aiming for Continuous Descent Operations (CDO), different aircraft fly different speeds (depending on type and weight) so overtake conflicts may need to be incorporated. [3] discusses these overtaking conflicts under the assumption that the airways are long, arriving at the conclusion that the average number of overtake conflicts *'represents the number of times aircraft would collide with each other if the controller did not take corrective action'*. However, for the shorter segments in the terminal airspace, this conclusion may not necessarily be justified. A further literature review and/or study into the occurrence of overtake conflicts may be needed at a later stage in the research or in further research.

2.2.9. Flow intensity

The relevant flow properties for conflict counts can be seen from an inspection of Equation 2.10, and include traffic intensity (λ) and speed (V), as well as the intersection angles between flows.

An important aspect in modelling structured traffic is intensity of the traffic. Various sources in literature assume a Poisson distribution, such as in [5] (without explicit justification, but referring to [3] which justifies this distribution - see subsection 2.2.7), or [24], showing that when departures are assumed to follow a Poisson distribution, arrivals also follow a Poisson distribution. [26] criticised this approach, as airports may have arrival and departure peaks, so that these traffic intensities may vary throughout the day, and suggested modelling the departure rates as a function of time throughout the day, and augmenting them with an uncertainty model.

Other properties influencing conflict counts may arise from the simulations described in Section 3.3.

2.3. Clustering flight tracks in terminal airspace

When a new airspace design is developed, flows may be defined by the instrument departure and arrival routes. However, in the analysis of an airspace design and for a realistic development of the proposed capacity estimation, especially for an airspace concept where vectoring is applied, an alternative means of grouping traffic is needed. The next section details the choice of a grouping (or 'clustering') technique.

Different approaches to clustering flight track data have been proposed. Most of them involve two separate steps:

1. convert the input data to a set of features for each track, and
2. subsequently cluster these sets of features.

This section outlines two frequently used clustering algorithms, followed by an overview of ways to convert input data into features.

2.3.1. Clustering algorithms

Two frequently used algorithms for this clustering task are DBSCAN and k-means. **DBSCAN** (*Density-based spatial clustering of applications with noise*) [2] identifies points in a high density region by finding collections of points having a minimum number of $minPts$ (the first hyperparameter) which are a maximum distance ϵ (the second hyperparameter) away. (Points that satisfy the ϵ criterion but themselves do not have $minPts$ other points are said to be *directly reachable* and are included). All these distinct collections of points are separate clusters; points not in a cluster are classified as noise. These points need not be location data, since in that case two distinct but crossing tracks could be classified together (depending on the hyperparameters); the clustering can be applied to inferred parameters of tracks.

In DBSCAN the number of clusters is a result of the clustering algorithm; for **k-means** clustering, it is an input (hyperparameter k). k-means clustering paritions a dataset into k partitions centered around cluster *means* or centroids. The necessity for *a priori* specification of k is a downside for the current application, as the number of clusters may not always be clear. This is sometimes overcome by trying multiple values of k and optimizing for some cost function, such as in [4].

2.3.2. Turning points as waypoints

[9] presented two methods. The first method uses turning points as feature; this is shown in Figure 2.3. The first step is extracting turning points from flight tracks, see Figure 2.3a. Next, similar turning points

are clustered (see Figure 2.3b) by applying either *k-means* or *DBSCAN* as a clustering algorithm. A combination of one or more turning points in a cluster is considered as one *way point*. Then, each flight track is considered as a sequence of these way points, possibly of different length. Finally, trajectories are clustered together by considering the longest common subsequence and generating clusters of sequences. The result is shown in Figure 2.3c, where each color represents a cluster.

2.3.3. Principal Component Analysis

The second method considered (see also Figure 2.4) used the (x, y, z) positions of flight tracks and resampled all tracks to a vector of length n (in [9], $n = 50$), augmented with other features (distance from a center point providing rate of convergence; distance from top left corner representing symmetry; angular position providing information about the overall location of the trajectory; and heading of the aircraft providing information about straight or curved trajectories). Then, the dimensionality is reduced by performing a Principal Component analysis and projecting each trajectory on the first p Principal components, resulting in a vector of length p for each trajectory. Finally, these vectors are clustered using DBSCAN. This approach was an improvement to [4], which proposed a similar algorithm but did not augment the trajectories and used k-means clustering instead of DBSCAN.

It can be seen in Figure 2.4c that a lot of outliers are present, which is unwanted for the current research, since these still contribute to conflicts.

2.3.4. Spectral clustering

In [7], Enriquez and Kurcz argued that PCA may not always capture local or nonlinear characteristics and that the choice of features for augmenting the data matrix can seem arbitrary, which is one of their arguments for using spectral clustering. They proposed an algorithm based on *spectral clustering.* It starts by resampling the n flight track position data to length m, and considers similarity by calculating a square adjacency matrix **W**. Element $W_{i,j}$ encodes the similarity between track x_i and x_j by computing a Gaussian kernel, i.e. $W_{i,j} = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}$, where x_i is an $(m, 2)$ matrix of (x, y) records and σ is a scale parameter, controlling the width of the neighborhoods. The authors note that the effect of time parametrization and Euclidation distance is that flights of similar lengths will be clustered together; consider two similar tracks where one aircraft flew a holding pattern and the other aircraft did not - the similar (x, y) positions will be at different (relative) timesteps resulting in dissimilarity.

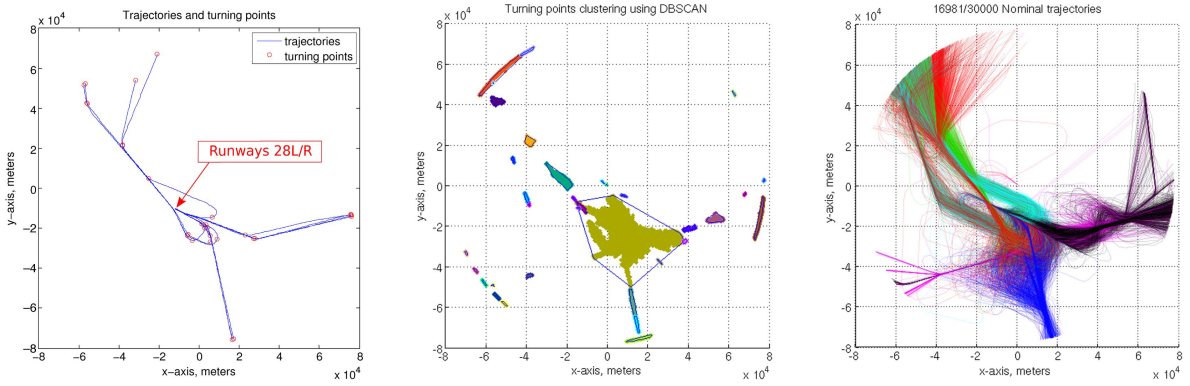
The resulting adjacency matrix **W** has values ranging from zero for completely dissimilar tracks to one for completely similar tracks. **W** is then recursively partitioned in two submatrices in each step based on the sign of the elements in the *Fiedler vector* of **W** [8]. In this way, each partition has maximum similarity within a group, and minimum similarity between groups. A stopping condition can be e.g. comparing the partition variance $\omega_{partition} = \text{var}(\mathbf{W}_{partition})$ against a threshold variance ω_{min}. An illustration of the resulting partitioning is shown in Figure 2.5 for a large scale (entire flights); applicability extends to the terminal area.

This spectral clustering has been extended to obtain *temporally persistent* flows, flows persisting across the time dimension (e.g. multiple days), by running the the algorithm on multiple periods (e.g. days) and computing the nominal tracks of each cluster in each period. These nominal tracks are then used as an input for the same clustering method (possibly with different hyperparameters, e.g. inputs to the stop condition) [6]. In this way, weeks or months worth of data can be analysed at a reduced computational cost (compared to aggregating all input data and using just one clustering step)[9]. An additional advantage is that 'outlier' days can be identified and dealt with - either to remove this as 'noise', or to study those outliers specifically.

2.3.5. Choice of clustering method

For the current research the spectral clustering method is selected: it achieves good results given the vectoring nature of tracks in e.g. the terminal Schiphol airspace, where a lot of different tracks exist, and clustering them based not only on spatial proximity but also similar path lengths achieves a relatively large number of flows but with a small spread. Additionally, the parallel approaches of runways that are less than the nominal separation minimum together should be clustered separately, which calls for more (and narrower) clusters. The flexibility in stopping conditions is an additional advantage.

[9] The similarity matrix has dimension (n, n) so the required computational resources grows with $O(n^2)$

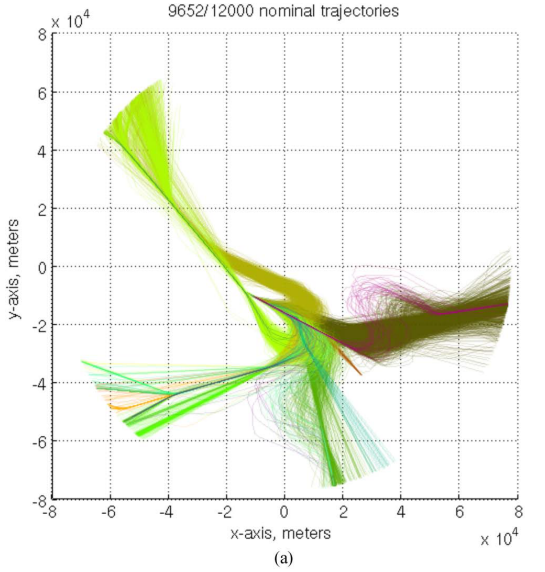


(a) Example of turning points

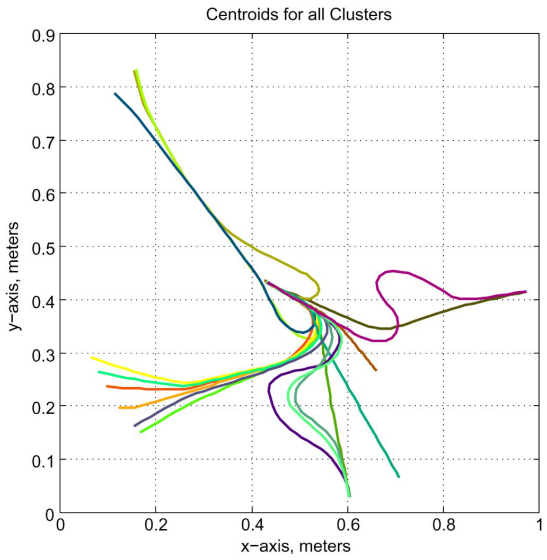
(b) Clustered turning points. The points in the pentagon in the center represent turns towards the runway and are not considered further.

(c) Resulting trajectory clustering

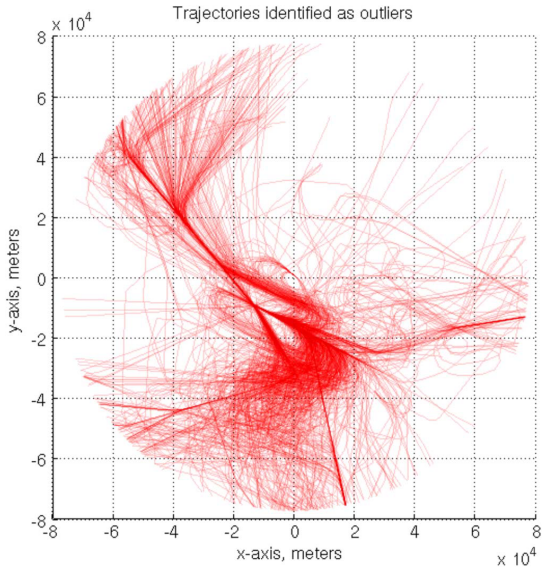
Figure 2.3: Trajectory clustering by analysis of sequences of turning points. Figures are taken from [9].



(a) clustering results, and the first three PCA vectorspaces



(b) Centroids of the clusters



(c) Oulier trajectories.

Figure 2.4: Trajectory clustering using Principal Component Analysis (PCA). Figures are taken from [9].

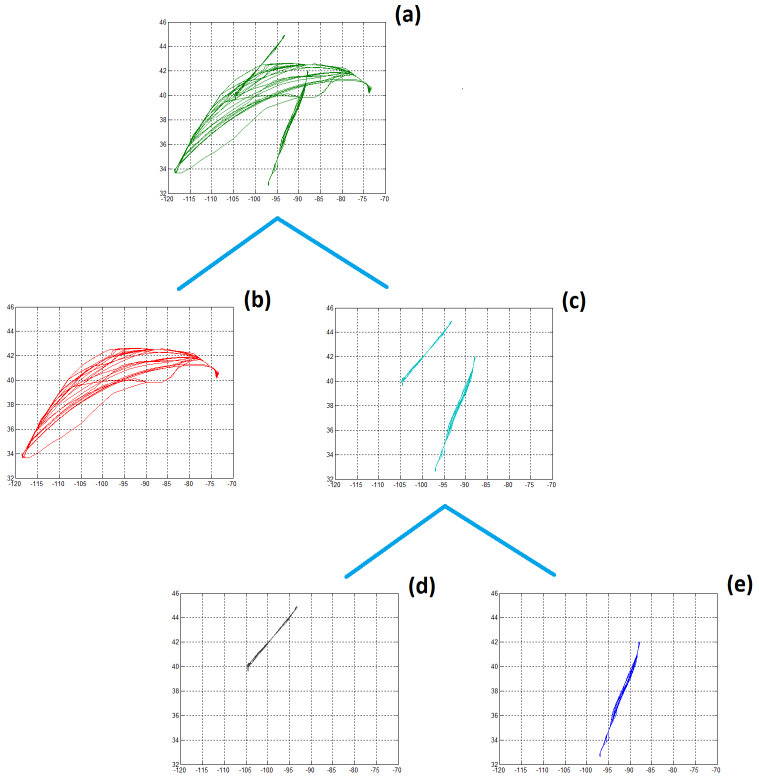


Figure 2.5: Clustering based on similarity. The grids represent longitude-latitude position data of flight tracks. Every level shows a subdivision of flight tracks. Figure taken from [7]

3

Methodology

In this chapter, the methodology for the research is presented. The experimental setup is explained in Section 3.1, containing and explaining the steps of the flowchart (Fig. 3.1). It outlines the need for a simulation framework, which is expanded on in Section 3.2. The relationship between flow properties, conflict counts, and capacity is discussed in Section 3.3. A final step in the methodology is putting the research to the test in a case study, which is explained in Section 3.4.

3.1. Experimental setup

An overview of the experiment setup is presented as a flowchart in Fig. 3.1. It consists of inputs (ovals on the top), *steps* (chevrons), *processes* (collections of steps), and (intermediate) **results** (rectangular boxes).

Raw ADS-B data is captured and *pre-processed*, producing **flight tracks** (Section 3.1.1). Additionally, this data is used for a replay to obtain the (total sum of) **real data conflict counts.**

The **flight tracks** are ***clustered***: first they are checked against the airspace definition (discarding the parts of the tracks that are not in the airspace under consideration) and the *minimum track length* (a clustering parameter). Tracks parts within the airspace and of sufficient length are *grouped on similarity* by the clustering algorithm described in Section 3.1.2, producing noise (which is **discarded**) and **flows**.

Three things are done with these **flows**:

i the **temporal flow properties** are obtained by *analysing the temporal behaviour in the flow*. These are needed as an input to the simulation;

ii the **flight tracks** are *simulated* to obtain **simulation conflict counts** (Section 3.1.3);

iii the flow properties knowledge is used to *extract flow properties influencing capacity*, which are in turn used as an input to *predict conflict counts*, producing an **analytical conflict count prediction** (Section 3.1.4).

The **simulation conflict counts** and the **analytical conflict count predictions** are *compared* for *verification* purposes (are the flow properties predicting the simulation conflict counts. Finally, the *validation* is performed by comparing the **real data conflict counts** against the **analytical conflict count prediction**: are the predicted conflict counts from extracted flow properties representative of the actual conflict counts? (Section 3.1.5)

The remainder of this section describes the *processes* in more detail.

3.1.1. Data acquisition

Flight tracks are obtained from processing ADS-B data; surveillance messages (containing position, altitude, and under some conditions also other data such as velocity and heading) broadcast by aircraft were received on a receiver in Delft, The Netherlands, and processed using a Python module. Both the data acquisition and the processing are described in [27]. Data is available for the entire month of Januari, 2018.

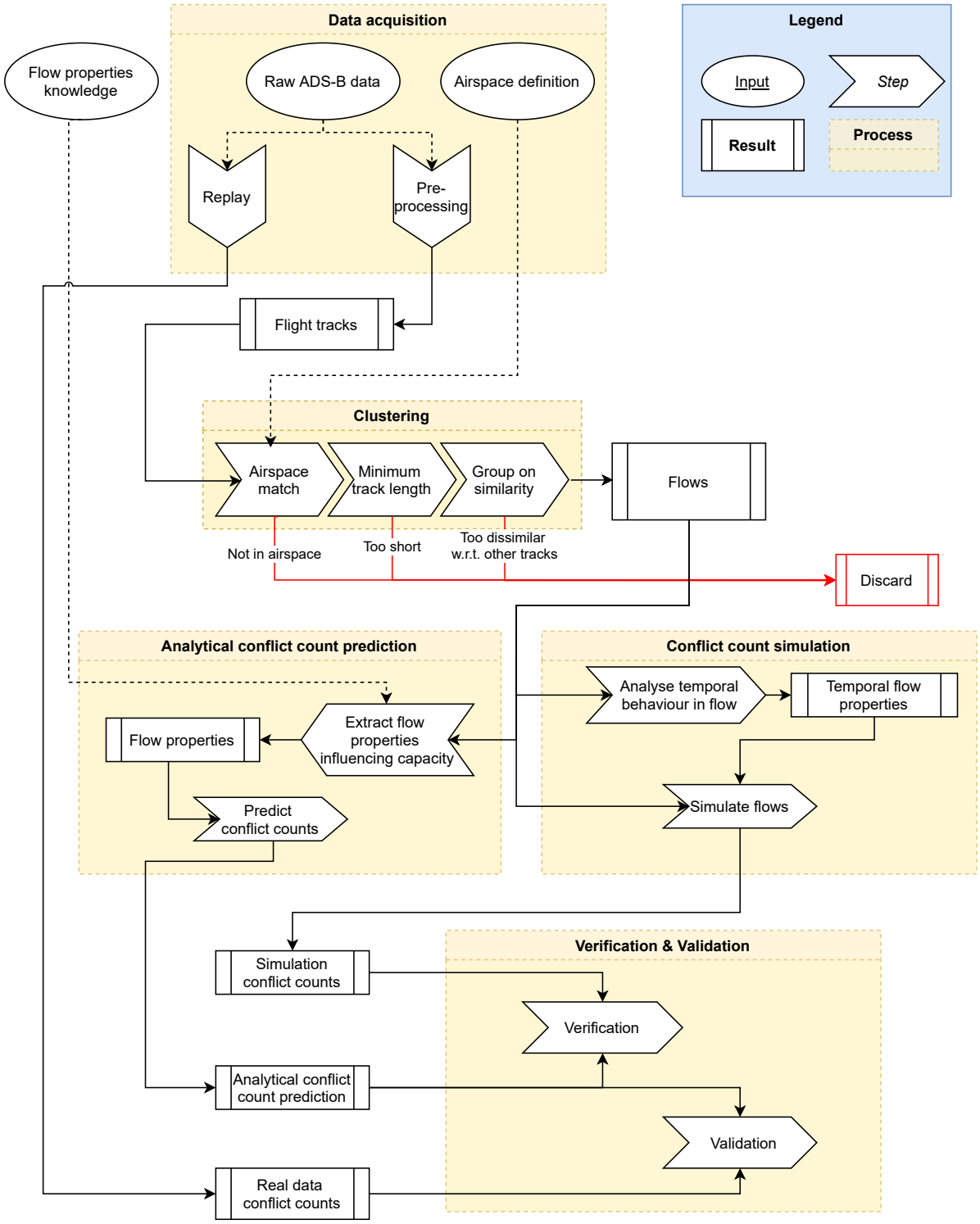


Figure 3.1: Flowchart of the experiment setup

The airspace definition is obtained from data in the Dutch AIP (Aeronautical Information Package[1]) and parsed into a Python *GeoPandas* object.

Flight tracks are converted into *GeoPandas* objects as well: for each track, the intersection of the track points with the selected airspace objects is used. The track points are converted from latitude/longitude to a 2D projection centered around center of the airspace under consideration, to prevent the clustering algorithm from having a bias towards proximity in longitude at higher latitudes.

3.1.2. Clustering

The spectral clustering algorithm described in Section 2.3.4 is applied to the pre-processed flight tracks. The parameters of this algorithm need to be determined: what are suitable stop conditions (under which conditions are flight tracks categorised as an existing cluster, a new cluster, or as noise), but also the inputs (apart from the minimum track length, also which states are to be included; it is possible to include not only 2D position information, but also altitude, velocity or heading data).

3.1.3. Conflict count simulation

The flows obtained from the clustering algorithm are simulated using the simulation framework described in Section 3.2, to obtain both the conflicts within aircraft in that flow, as well as the conflicts between different flows. Apart from the *spatial* information of the flow (mean track, some measure of the spread around this mean), the *temporal* characteristics need to be known as well: at what rate are aircraft arriving in the flow[2]? Additionally, the time of day when a flow is used may also need to be considered: if two flows are separated in time (think of e.g. night arrivals versus day arrivals), they may be conflicting spatially, but those conflicts should not be counted since they would not occur in practice.

3.1.4. Conflict count prediction

Parallel to simulating the conflict counts, they should also be predicted from flow properties; the main theme of this research. This involves finding the relevant flow properties that have an effect on capacity, as well as how that influence on capacity can be quantified for that property. This is based on the theory of Section 2.2.5, and is treated in more detail in Section 3.3.

3.1.5. Verification & Validation

The simulation conflict counts are compared to the predicted conflict counts to assess the consistency of the method: are the right properties used under the right mathematical relationships? Additionally, the analytical predicted conflict counts are compared to the replay conflict counts to verify that the model is accurately predicting the total conflict counts. To remove the effect of controller deconfliction in the datasets, a sampling trick may be required: either implementing a time offset, or using a random combination of tracks in certain days, or perhaps altering the altitudes of the aircraft to obtain a 2D-simulation. The comparison between this (modified) replay and the analytically predicted conflict counts can be used to perform a final validation: are the conflicts accurately predicted by the model?

3.2. Simulation framework

The requirements on the registration of conflict counts are different for the cases of structured and unstructured airspace. In the work of Sunil [28], the BlueSky ATC Simulator [12] was used and the conflict count of the entire experiment region for the entire experiment duration was used. However, for the current research the conflict counts between specific flows needed to be available.

At first, BlueSky was used with the addition of the built-in conflict logger; all flows had a flow identifier contained in their callsign, and a post-processing of the conflict log was used to determine the conflicts per flow. This post-processing was developed in Python using a Test-Driven Development approach.

During this development it became apparent that the conflict logger did not provide consistently accurate conflict counts. For example, when aircraft A and B are in conflict, both aircraft should register a conflict: aircraft A should register a conflict with aircraft B, and aircraft B with aircraft A. Either the number of aircraft pairs that have a conflict can then be extracted, or the number of total perceived conflicts by all aircraft; these should differ by a factor of exactly two.

[1] See `https://www.lvnl.nl/informatie-voor-luchtvarenden/publicaties-voor-luchtvarenden`; TMA/CTA definition is described in ENR 2.1; CTR definition is described per airport in AD 2.17 per airport

[2] If holding patterns exist the time spent in the flow will also be important, but this is outside of the scope of the current work.

However, not all conflict pairs were registered: sometimes aircraft A did show up as having a conflict with aircraft B, but not vice-versa. Additionally, when multiple aircraft were in conflict in the same simulation timestep, these entries could be scattered in the conflict log, and no reliable method was found to 're-pair' the aircraft conflict entries. And sometimes, two aircraft would have a conflict in a certain timestep, then no conflict in the next, and again a conflict in the timestep after that one. All this prevented a reliable measure of the conflict counts between certain flows.

3.2.1. Measuring conflicts between flows

These problems, in context with the hassle of generating traffic flows in python, converting them to BlueSky input files, running the simulations, extracting the conflict logs, the work-around with callsigns to assign flow identifiers, and analysing the conflict logs, prompted the development of a simulation framework that could reliably generate traffic in flows and measure the conflict counts between them.

This custom simulation framework was developed in Python, again using a test-driven development approach. It has an aircraft as a basic entity, with a position, altitude, track, groundspeed, altitude and vertical speed. These can be updated during the course of the simulation; additionally, aircraft can be activated and deactivated; only active aircraft have their position updated and their conflicts counted in a simulation timestep. This active/inactive property of aircraft allows an efficient generation of a group of aircraft with similar or identical initial properties, but with different activation times. In this way, the (stochastic) arrival of aircraft with similar properties in a certain airspace can be simulated.

Several aircraft can be combined into a *flow*. Conflicts are counted between flows and separately for within a flow. Within a flow, a triangular, binary matrix registers for all pairs of aircraft within that flow whether or not a conflict has occurred over the course of the simulation. Between flow A and flow B, each with m and n aircraft, respectively, a binary, triangular matrix of size (m, n) registers whether or not aircraft A_i and B_j have been in conflict during the simulation. These boolean conflict properties can be aggregated to integer conflict counts within a flow and between flows. In this way, an efficient and robust measure of the conflicts is available, ensuring a no-duplicate conflict count.

A graphical interface is available to visualize the simulation. Location, speed vector, flow membership and conflict state of aircraft can optionally be visualised, to enable visually inspecting the correctness of the simulation and the conflict states. An example is given in Section 3.3.1 (Fig. 3.2).

3.2.2. Stochastic element

A conflict count model needs a stochastic element, since it relates aircraft count to conflict count by means of a conflict *probability*. The experiments of Sunil involved simulating aircraft starting and ending at random locations within a certain region; to keep the traffic density constant, *"aircraft were introduced into the simulation at a constant spawn rate"* [28, Section 3.5.2] related to the average speed and trip distance of aircraft.

This approach cannot be used for structured airspace, since the goal of the current research is to find the conflict counts for groups of aircraft with paths between similar entry and exit points. If constant spawn rates are used together with similar paths, conflicts probability practically reduces to a function of the spawn rates and time difference between the first spawn occurrence for two flows. Additionally, a constant spawn rate is not a realistic assumption for structured airspace.

As a result, the deterministic and stochastic aspects are interchanged for a structured airspace experiment: the path aircraft in a certain flow take is predetermined, but their spawn times are randomly distributed according to an appropriate probability distribution function. As discussed in Section 2.2.9, a Poisson distribution is frequently used in literature. For modelling conflicts between two relatively homogeneous flows, a constant Poisson distribution per flow is a fair assumption. However, when actual traffic data is used, a check on the intensities will need to be performed to see if this assumption holds. Another possibility is subdividing traffic flows so that each subset has a Poisson parameter accurately describing the arrival intensity in that subset, and including them in simulations only during an appropriate time interval. When this is done throughout the simulation window based on the observed occurrences of these subsets, departure and arrival peaks can also be included in the model.

3.3. From flow properties to capacity

A major part of this research is developing the relationship between flow properties and capacity, so the "Analytical conflict count prediction" process in Fig. 3.1 needs to be expanded upon.

3.3.1. Influence of flow properties on conflict counts

The conflict rate of intersecting flows was discussed in Section 2.2.6, and the formula obtained there is repeated here for clarity:

$$C = \frac{2S_h\sqrt{V_i^2 + V_j^2 - 2V_iV_j\cos(\alpha)}}{\frac{V_i}{\lambda_i}\frac{V_j}{\lambda_j}\sin(\alpha)}P(\text{vertical overlap}) \qquad \text{(2.10 repeated)}$$

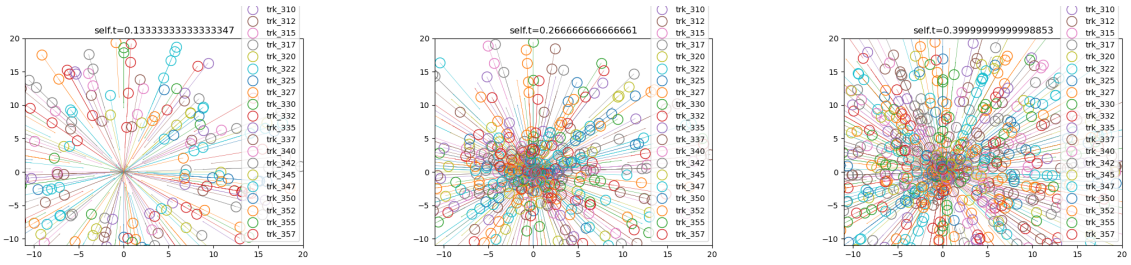
This suggests that the conflict rate depends on the separation requirement S_h (usually 3 nautical mile in the terminal airspace), the (average) flow speeds V_i and V_j, the flow intersection angle α and the flow spawn rates λ_i and λ_j. To verify this relationship, an experiment is performed where the conflicts between two intersecting flows are measured for certain flow speeds and intersection angles, which are varied. To minimise computational overhead, the simulation framework of Section 3.2 is used where several flows are created, originating on points of a circle every $\delta\theta$ and flying radially inwards; see Fig. 3.2[3]. Aircraft are spawned on the points of this circle based on a Poisson process with intensity λ. To remove transient effects, the resulting conflicts between pairwise flows are only counted after a set-in period, and measured for a preset measurement time during which new aircraft keep spawning. The resulting conflict counts are normalised for the measurement time window to obtain conflict rates, and then aggregated based on flow characteristics so that the results can be averaged. For instance, if flows exist at $\theta = 0, 10, 20, 30, \ldots$ ° and all have the same speeds, for $\alpha = 10°$ the collisions between flow 0 and 10 °, 10 and 20° and so on are collected.

More formally, consider the conflict rate C as a function of its variables:

$$C = C\left(\alpha, V_1, V_2, \lambda_1, \lambda_2, S_h\right) = \frac{2S_h\sqrt{V_i^2 + V_j^2 - 2V_iV_j\cos(\alpha)}}{\frac{V_i}{\lambda_i}\frac{V_j}{\lambda_j}\sin(\alpha)}P(\text{vertical overlap}) \qquad (3.1)$$

Then, N flows are simulated, with flow n having properties $(\theta_n, V_n, \lambda_n)$, and the measured conflict rates between flows i and j are measured to be $C_{i,j}^{measured}$ (for both i and j in $1, 2, 3, \ldots, N$). The resulting measured conflict rate function $C^{measured}(\alpha)$ is obtained by summing and averaging[4] for equal α.:

$$C^{measured}(\alpha) = \frac{1}{2}\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\begin{cases} C_{i,j}^{measured} & |\theta_i - \theta_j| = \alpha \\ C_{i,j}^{measured} & |\theta_i - \theta_j| > 180° \wedge 360° - |\theta_i - \theta_j| = \alpha \\ 0 & \text{otherwise} \end{cases} \qquad (3.2)$$



(a) After 0.13 hours **(b)** After 0.26 hours **(c)** After 0.4 hours

Figure 3.2: Verifying conflict rate equation by measuring conflicts between flows at varying angles. All situations are still in the set-in phase. Planes are represented as circles with speed vectors, legend names are based on their flight angle. The (x, y) plane represents spatial coordinates in nautical miles.

As an example, the ratio of observed conflicts to predicted conflicts is shown in Fig. 3.3. It shows that the ratio of observed to predicted conflict rates is rather invariant to α, except for effects at the low and high angles. These edge cases still need to be investigated.

Repeating this process for the other parameters verifies or rejects Equation 2.10; additionally, this can be used to find insight into the sensitivity of the collision rate to the input variables.

[3] In essence, this is a shortcut for repeatedly running the simulation for pairs of flows

[4] The division by two is to prevent (i, j) and (j, i) pairs from being counted double; alternatively, the top two lines of the summation expression could include the condition that $i < j$.

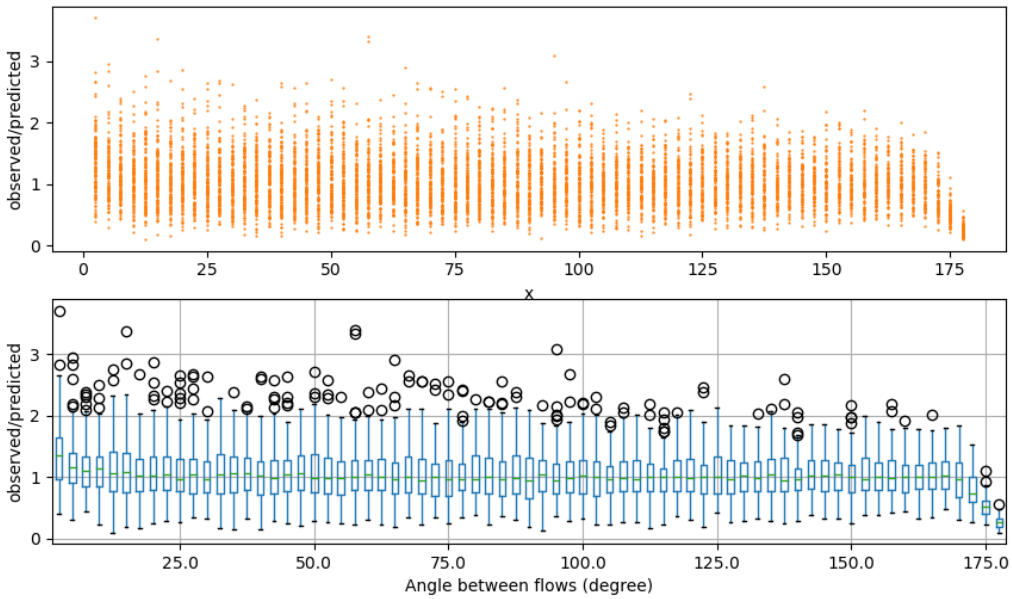


Figure 3.3: Ratio of observed/predicted conflicts for varying α. Top graph shows the scatter plot, bottom graph shows the boxplot of the same data.

3.3.2. Relation between conflict counts and capacity

Once a function relating flow properties to conflict counts has been found, the final step of the methodology would be to relate this to the capacity. More specifically of interest is when the conflict counts are such that the airspace is saturated and a higher volume of conflicts could not have been dealt with, leading to the use of air traffic flow management tools or holdings. Note that a *full* relationship between conflict counts and capacity may not be needed; if the conflict counts can reliably predict air space saturation, that may already be sufficient.

However, the identification of airspace saturation could require a substantial amount of extra work, and may therefore be left as a recommendation for future work.

3.4. Case study

Finally, to show the relevance of the research, a case study will be performed. This involves the selection of a baseline scenario, identifying high-conflict flows, develop a scenario where these flows are changed, predicting the change in conflict counts, and comparing the conflict counts of both scenarios in a fast-time simulation in BlueSky. Meaningful measures for high-conflict flows and the prediction for the change are dependant on the further insight developed by performing the previous experiments in this chapter, and as such are not specified at this stage. They will however be specified before the start of the case study.

4

Overview of experiments

Having discussed the theoretical background and methodology in previous chapters, this chapter gives an overview of the sub-experiments in Section 4.1, as well as a planning for the remaining work.

4.1. Sub-experiments

From the previous chapters, it follows that several sub-experiments need to be performed. These are described in this section.

4.1.1. Clustering

The output of the chosen clustering algorithm depends (apart from the input data) on:

CL_1 Scale parameter σ, influencing when flight tracks are considered *similar*;
CL_2 Stop condition
CL_3 Aircraft states to include in the clustering algorithm

The output is:

1. a set of clusters
2. noise

A qualitative analysis[1] of varying choices for CL_1, CL_2, and CL_3 will be carried out to find suitable combination. Early results suggest that the scale parameter is crucial for separating flights in the parallel approaches at Schiphol. There is also an interaction between CL_1 and CL_3, since the scale of different variables may be different; it may be needed to apply a weighing function to the clustering variables.

4.1.2. Analytical conflict count simulation and prediction

Formula verification

The heart of the research is the analytical conflict count prediction. The verification of the relationships between flow properties and conflict counts, as described in Section 3.3.1, needs to be carried out for each type of input variable:

AP_1 Flow intersection angle α
AP_2 Flow speed V
AP_3 Flow intensity λ
AP_4 Separation requirement S_h

This is done by a set of experiments, in each setting one of the variables as the independent variable, and the others as control variables at fixed values. These situations are simulated and their measured conflict counts (Equation 3.2 for variable AP_1, and the analogous conflicts for the other variables) are

[1] Note that a full, quantitative analysis of the choices are outside of the scope of this thesis work.

compared to the results of performing the calculation (Equation 3.1). The ratio of these functions should be close to one for a wide range of the varied variable. If this the case for all but the edges of certain intervals, these intervals can be used as an indication of the range of validity of the prediction.

More formally, several iterations of the *Kruskal–Wallis test* [17] are run, starting at the full range and, if necessary, reducing the range, such that the largest consecutive range of the input variable yields no dominance for one specific group. The resulting consecutive range can be interpreted as the range of values for which the model is valid and as such represents a restriction on the validity of the model.

Realistic flow scenarios

Parallel to establishing the type of relationship and the regions of validity, flow pairs from the clustered flight tracks are obtained. First, the flow properties need to be checked against the region of validity for the respective properties; if a substantial number of samples fall outside of the region of validity, an extension of the validity region may need to be made. Then, their conflict counts are predicted analytically and obtained from the same simulation framework. Comparing these establishes the validity of the prediction for realistic flow scenarios.

An item of attention is the spread in a flow. [5] assumes a uniform distribution along an airway of a certain width, but this width cancels out in the conflict count prediction formula. If the spread is non-uniform in the flow cross-section, the effect of violating this assumption needs to be studied; it may be needed to add a correction factor for flow width, or more generally, for the probability distribution of traffic along the flow cross-section.

4.1.3. Validation to simulation replay conflict counts

Once the previous experiments have finished, the validation of the method is performed, see Fig. 4.1. After a scenario selection, consisting of selecting a time period and artificially altering the the flight track data to remove the effect of deconflictive controller action. The real data conflict counts are obtained from a BlueSky replay and compared with the analytically predicted conflict counts obtained from the clustering and prediction routine. The result of this experiment is a conclusion on the validity of the method, optionally aided by an analysis of the sub-results from different clusters.

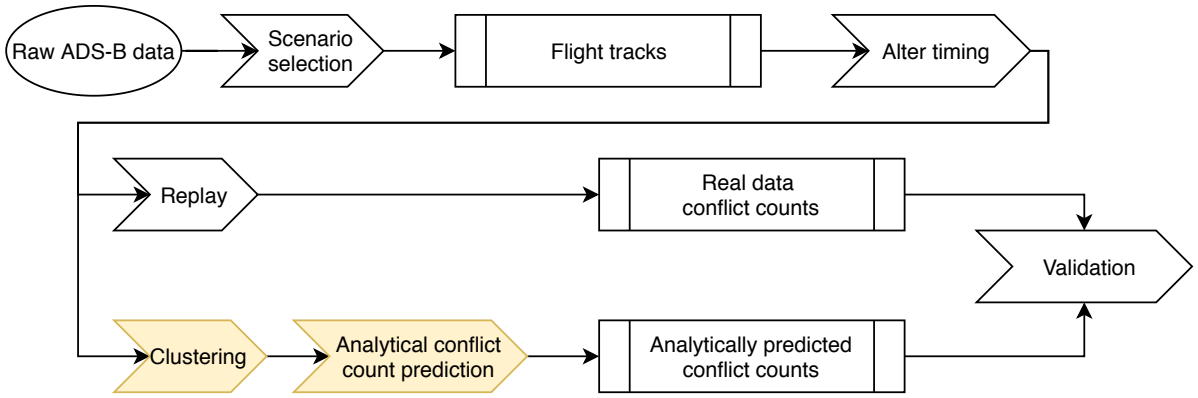


Figure 4.1: Flowchart of the validation procedure

Removing the effect of deconflictive controller action

Artificially altering the flight track data to remove the effect of deconflictive controller action is nontrivial, but a complete analysis of the *best* way to do this is outside the scope of this research. Instead, simulations of real data conflict counts will be performed for different realisations of the stochastic controller-action-removing procedure, and an average of the resulting conflict counts will be used. The most appropriate stochastic procedure is selected by repeating this for the procedures mentioned in Section 3.1.5, and selecting the procedure with the lowest spread.

4.1.4. Case study

The final experiment is a case study, as mentioned in Section 3.4. A flowchart can be seen in Fig. 4.2. Similar to the validation procedure, a scenario is selected, flight tracks are obtained, and their timing is

altered. They are clustered and their conflict counts are analytically predicted. But in the case study, high conflict counts flows (or flow pairs) are identified, and by analysing their properties, the reason for these high conflict counts is determined. By altering these flows (either changing the flight path, changing speed, altering the geometry of the intersection, or spreading out flow intensity) to parameters that predict lower conflicts, an improved scenario is obtained. The analytic results and a replay of this improved scenario are determined and compared to the baseline scenario in the final step of the case study.

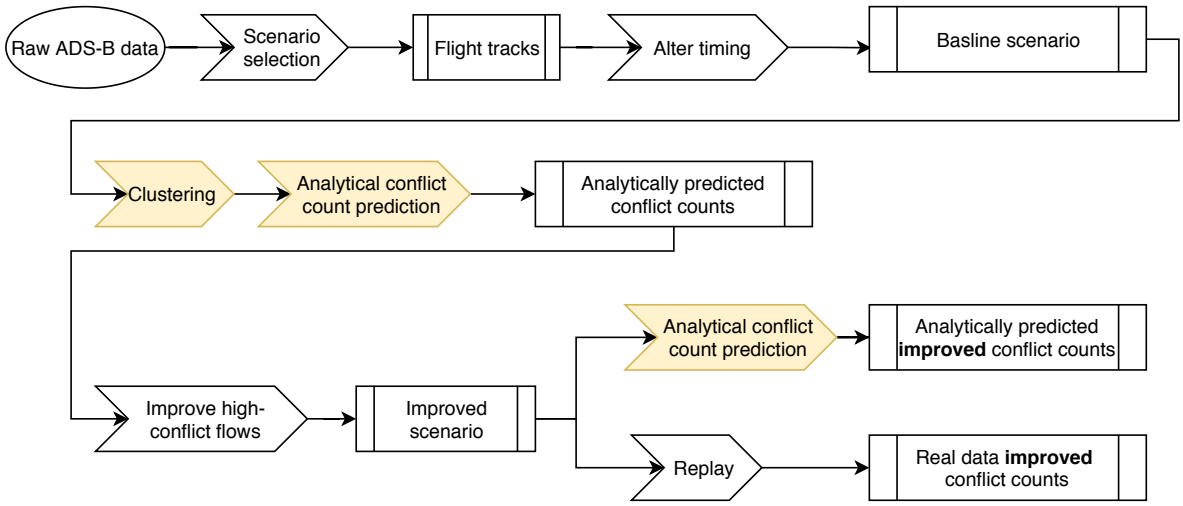


Figure 4.2: Flowchart of the case study

4.2. Planning

Finally, the planning for the remaining work is set out by means of a work breakdown structure, a description of the remaining work, and a Gantt chart.

4.2.1. Work breakdown structure

The work breakdown structure from the Project Plan, revised and reflecting which work has already been completed, is shown in Fig. 4.3. The orientation phase is completed, the software has mostly been written and verified, and this report marks the end of the preliminary research phase.

4.2.2. Remaining work

The work that still needs to be done is subdivided into the Detailed Research Phase and the Final Phase:

Detailed Research Phase (DRP)

DRP-1 Choose the hyperparameters for the clustering algorithm
DRP-2 Extract predictive flow properties (temporal and spatial) from clustered flight track data
DRP-3 Verify & validate relationship between flow properties and conflict counts

DRP-3(a) …for a range of artificial flow properties
DRP-3(b) …for clustered flow pairs
DRP-3(c) …for a combination of clustered flow pairs

DRP-4 Perform the case study

After the research phase has been completed, the research paper and thesis report need to be written and the thesis presentation (contingent on the green-light review) needs to be prepared and given.

Final Phase (FP)

FP-1 Write research paper
FP-2 Achieve green light
FP-3 Write & hand-in thesis report
FP-4 Hold thesis presentation & defence

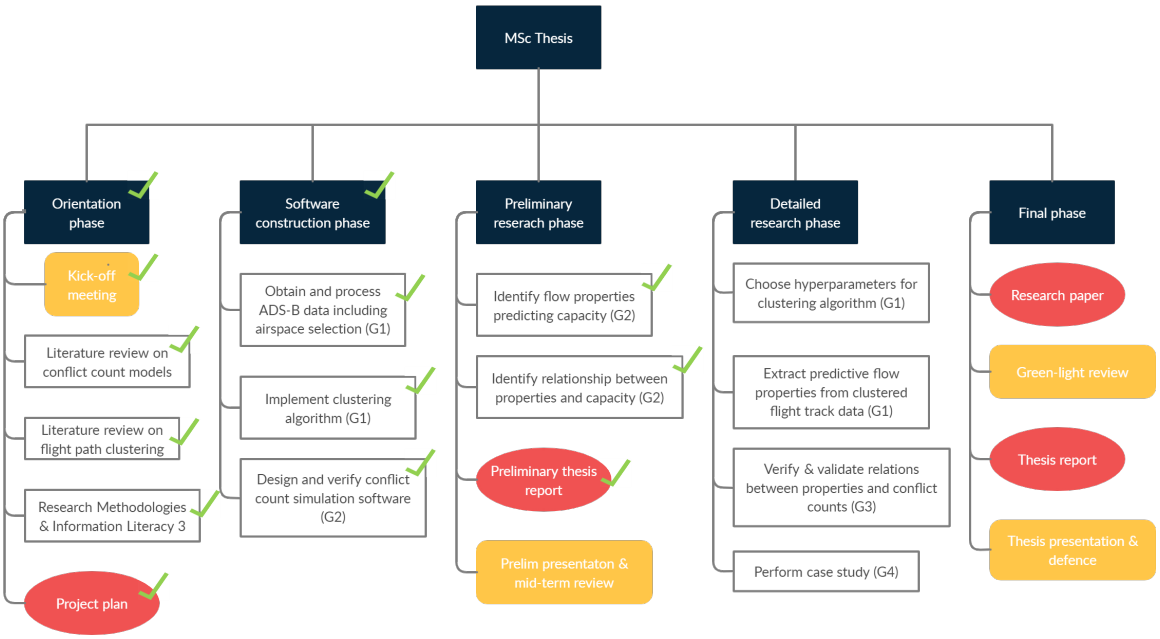


Figure 4.3: Work breakdown structure. G1 through G4 refer to the research goals. The green checkmarks indicate what work has already been completed

4.2.3. Gantt chart

A Gantt chart relating the remaining work of the previous subsection is included in Appendix E.

References

[1] Karl D. Bilimoria et al. “Performance evaluation of airborne separation assurance for free flight”. In: *18th Applied Aerodynamics Conference* August (2000). ISSN: 1064-3818. DOI: 10.2514/atcq.11.2.85.

[2] M. Daszykowski and B. Walczak. “Density-Based Clustering Methods”. In: *Comprehensive Chemometrics* 2 (2009), pp. 635–654. DOI: 10.1016/B978-044452701-1.00067-3.

[3] William J. Dunlay. “Analytical Models of Perceived Air Traffic Control Conflicts.” In: *Transportation Science* 9.2 (1975), pp. 149–164. ISSN: 00411655. DOI: 10.1287/trsc.9.2.149.

[4] Adric Eckstein. “Automated flight track taxonomy for measuring benefits from performance based navigation”. In: *Proceedings of the 2009 Integrated Communications, Navigation and Surveillance Conference, ICNS 2009* (2009), pp. 1–12. DOI: 10.1109/ICNSURV.2009.5172835.

[5] Shinsuke Endoh. “Aircraft collision models”. PhD thesis. Massachussets Institute of Technology, 1982.

[6] Marco Enriquez. “Identifying Temporally Persistent Flows in the Terminal Airspace via Spectral Clustering”. In: *Proceedings of the 10th USA/Europe Air Traffic Management Research and Development Seminar, ATM 2013* (2013).

[7] Marco Enriquez and Christopher Kurcz. “A Simple and Robust Flow Detection Algorithm Based on Spectral Clustering”. In: May (2014), pp. 1–6.

[8] Miroslav Fiedler. “A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory”. In: *Czechoslovak Mathematical Journal* 25.4 (1975), pp. 619–633. ISSN: 0011-4642. DOI: 10.21136/CMJ.1975.101357. URL: https://dml.cz/handle/10338.dmlcz/101357.

[9] Maxime Gariel, Ashok N. Srivastava, and Eric Feron. “Trajectory clustering and an application to airspace monitoring”. In: *IEEE Transactions on Intelligent Transportation Systems* 12.4 (2011), pp. 1511–1524. ISSN: 15249050. DOI: 10.1109/TITS.2011.2160628.

[10] Glen A. Gilbert. “Historical Development of the Air Traffic Control System”. In: *IEEE Transactions on Communications* 21.5 (1973), pp. 364–375. ISSN: 00906778. DOI: 10.1109/TCOM.1973.1091699.

[11] J. M. Hoekstra, R. N.H.W. Van Gent, and R. C.J. Ruigrok. *Designing for safety: The ’free flight’ air traffic management concept*. Vol. 75. 2. 2002, pp. 215–232. ISBN: 9080634328. DOI: 10.1016/S0951-8320(01)00096-5.

[12] Jacco Hoekstra, Joost Ellerbroek, and Jacco M Hoekstra. “BlueSky ATC Simulator Project: an Open Data and Open Source Approach”. In: *Proceedings of the seventh International Conference for Research on Air Transport (ICRAT)*. 2016. URL: https://www.researchgate.net/publication/304490055.

[13] D. A. Hsu. “The Evaluation of Aircraft Collision Probabilities at Intersecting Air Routes”. In: *Journal of Navigation* 34.1 (1981), pp. 78–102. ISSN: 14697785. DOI: 10.1017/S0373463300024279.

[14] ICAO. *A Unified Framework for Collision Risk Modelling in Support of the Manual on Airspace Planning Methodology With Further Applications, Circ 319-an/181*. Tech. rep. 2008.

[15] ICAO. *Air traffic management*. 2020. URL: https://www.icao.int/safety/airnavigation/Pages/atm.aspx.

[16] ICAO. *Procedures for Air Navigation Services. Air traffic management (Doc 4444). 16th edition.* 2016.

[17] William H. Kruskal and W. Allen Wallis. “Use of Ranks in One-Criterion Variance Analysis”. In: *Journal of the American Statistical Association* 47.260 (1952), pp. 583–621. ISSN: 1537274X. DOI: 10.1080/01621459.1952.10483441.

[18] Luchtverkeersleiding Nederland. *Veelgestelde vragen*. 2020. URL: https://www.lvnl.nl/werken-bij-lvnl/veel-gestelde-vragen.

[19] Ministerie van Infrastructuur en Waterstaat. *Herziening luchtruim*. 2020. URL: https://www.luchtvaartindetoekomst.nl/herziening-luchtruim/default.aspx.

[20] Ministerie van Infrastructuur en Waterstaat. *Verantwoord vliegen naar 2050: Ontwerp luchtvaartnota 2020-2050*. Tech. rep. 2020.

[21] Ministerie van Infrastructuur en Waterstaat. *Voortgangsbrief project Luchtruimherziening*. 2018. URL: https://www.rijksoverheid.nl/documenten/kamerstukken/2018/07/06/voortgangsbrief-project-luchtruimherziening.

[22] Minh Ha Nguyen and Sameer Alam. "Airspace Collision Risk Hot-Spot Identification using Clustering Models". In: *IEEE Transactions on Intelligent Transportation Systems* 19.1 (2018), pp. 48–57. ISSN: 15249050. DOI: 10.1109/TITS.2017.2691000.

[23] Peter G. Reich. "A theory of safe separation standards for air traffic control". In: 64041 (1964).

[24] S. Roy, B. Sridhar, and G. C. Verghese. "An aggregate dynamic stochastic model for air traffic control". In: *Proc. 5th USA/Eur. ATM 2003 R&D Seminar*. Budapest, Hungary, 2003.

[25] RTCA Task Force 3. *Final report of RTCA Task Force 3: Free flight implementation*. Tech. rep. Washington DC: RTCA Inc., 1995, chapter 3.

[26] Banavar Sridhar, Shon R. Grabbe, and Avijit Mukherjee. "Modeling and Optimization in Traffic Flow Management". In: *Proceedings of the IEEE* 96.12 (2008), pp. 2060–2080. ISSN: 15582256. DOI: 10.1109/JPROC.2008.2006141.

[27] J Sun et al. "pyModeS: Decoding Mode-S Surveillance Data for Open Air Transportation Research". In: *IEEE Transactions on Intelligent Transportation Systems* (2019). ISSN: 1524-9050. DOI: 10.1109/TITS.2019.2914770.

[28] Emmanuel Sunil. *Analyzing and Modeling Capacity for Decentralized Air Traffic Control*. 2019, p. 289. ISBN: 9789055841745. DOI: 10.4233/uuid. URL: https://repository.tudelft.nl/islandora/object/uuid%3A19aa4685-b75a-4fa3-bdfc-54401c6235d6?collection=research.

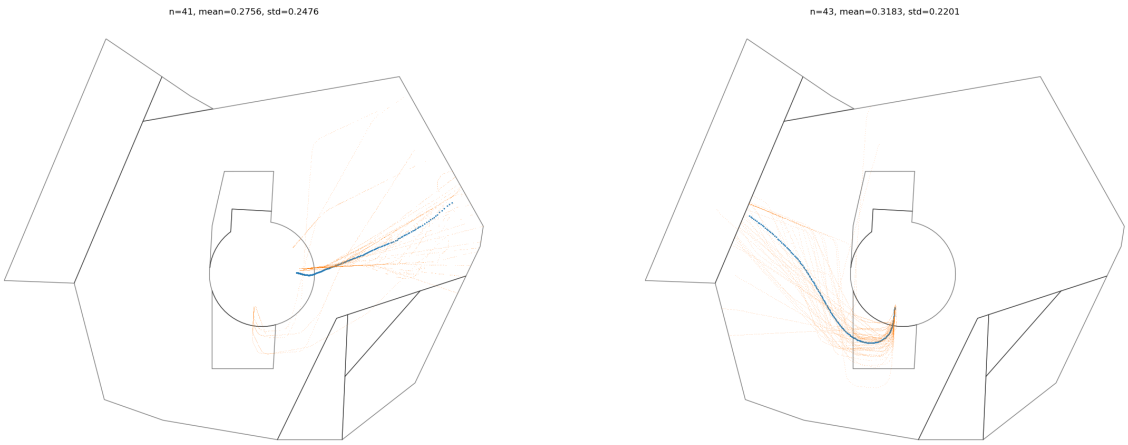
Part III

Appendices

A

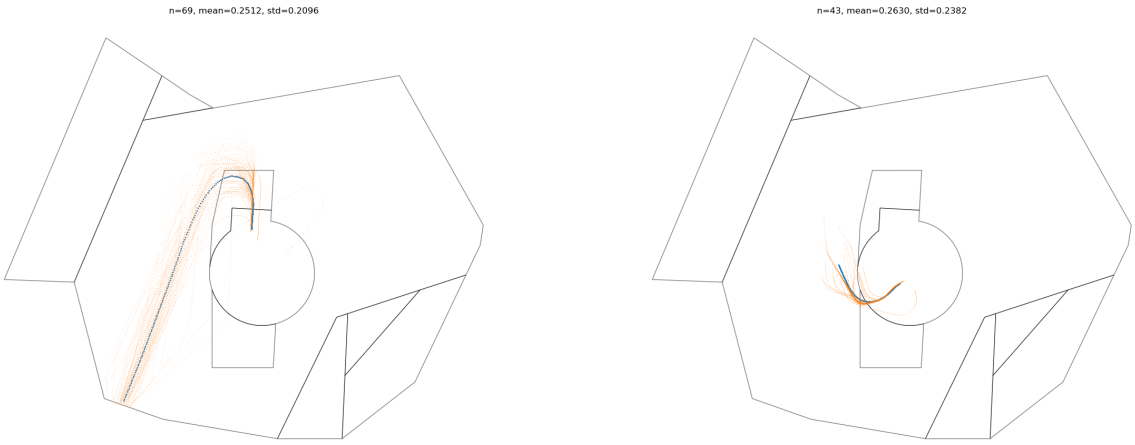
Unpredicted conflicts

A.1. Separate days



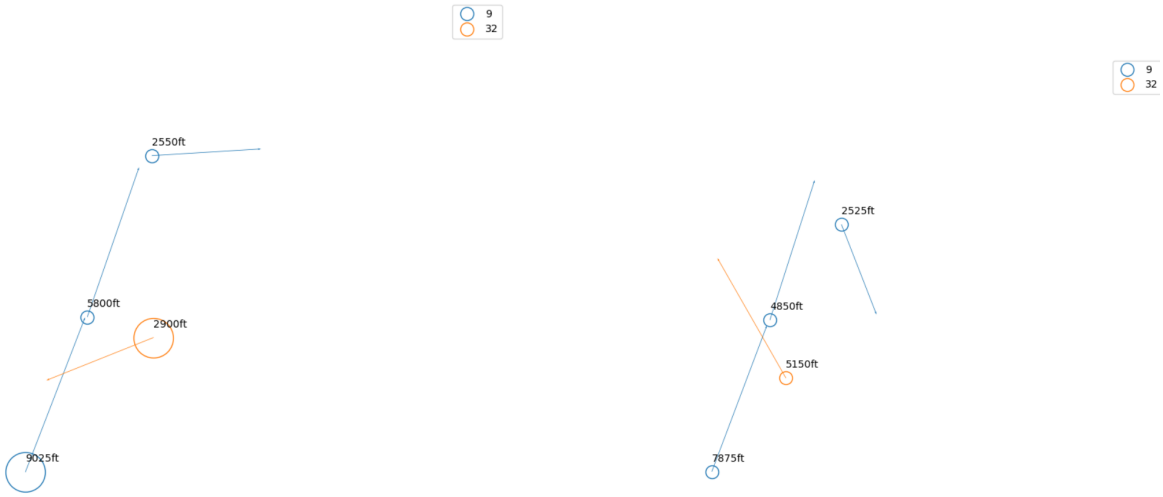
(a) Cluster 21 **(b)** Cluster 22

Figure A.1: Conflicts between Jan 2, clusters 21 and 22 (28 conflicts)



(a) Cluster 9 **(b)** Cluster 32

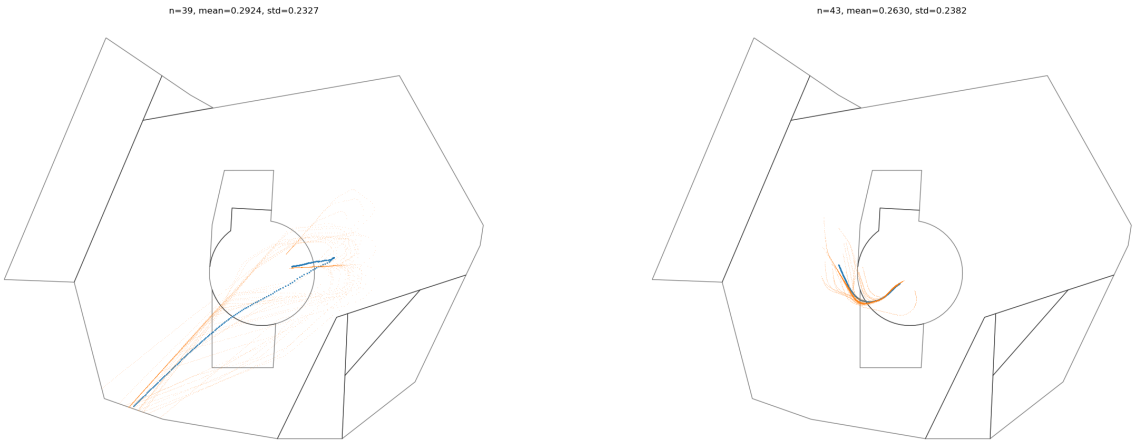
Figure A.2: Conflicts between Jan 4, clusters 9 and 32 (15 conflicts)



(a) Conflict. The bottom-left cluster 9 aircraft has a vertical speed of -576 feet/hour, the orange cluster 32 aircraft has a vertical speed of +1664 feet/hour.

(b) One minute later, no conflict. The bottom-left cluster 9 aircraft has a vertical speed of -1856 feet/hour, the orange cluster 32 aircraft has a vertical speed of +2944 feet/hour.

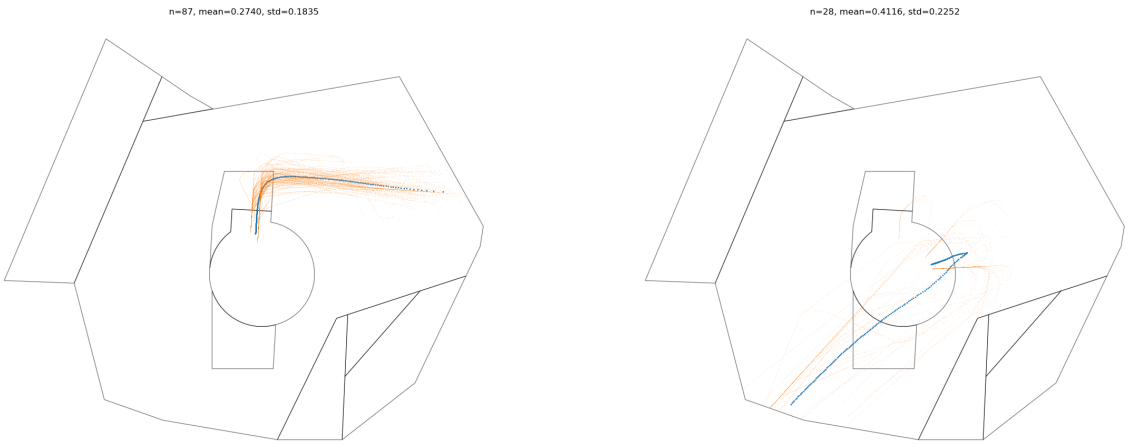
Figure A.3: Example of conflicts between Jan 4, clusters 9 and 32. Only 2 conflict pairs had conflicts that lasted for longer than one minute.



(a) Cluster 11 **(b)** Cluster 32

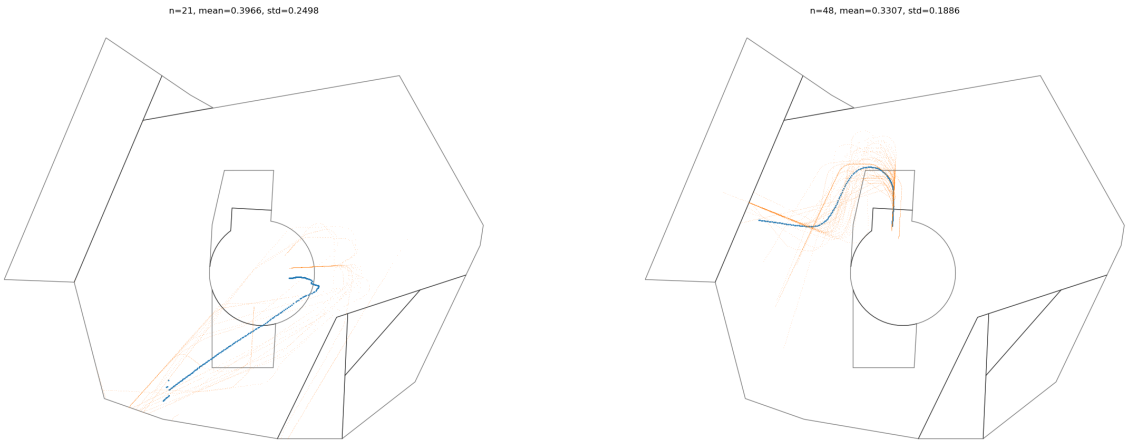
Figure A.4: Conflicts between Jan 4, clusters 11 and 32 (9 conflicts)

A.2. Combined days



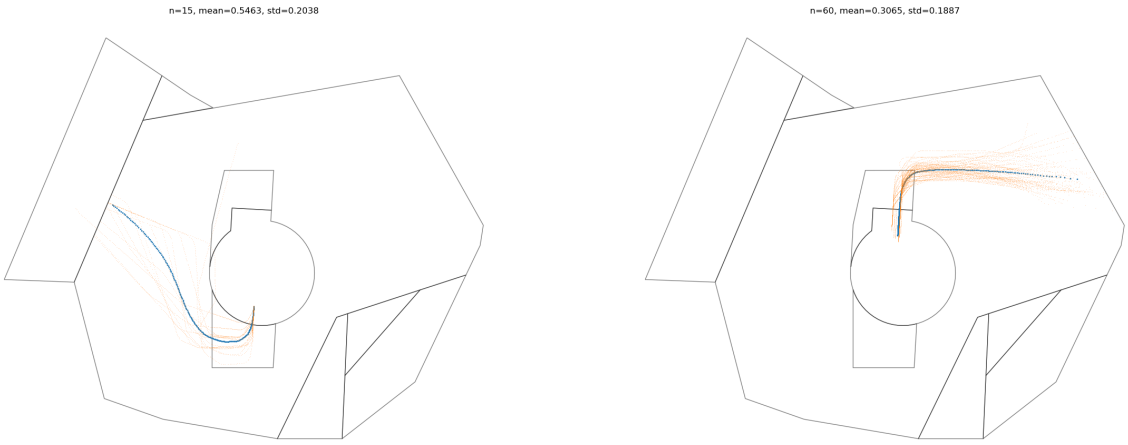
(a) Split 2, cluster 4 **(b)** Split 2, cluster 26

Figure A.5: Conflicts between clusters 4 and 26 (12 conflicts)



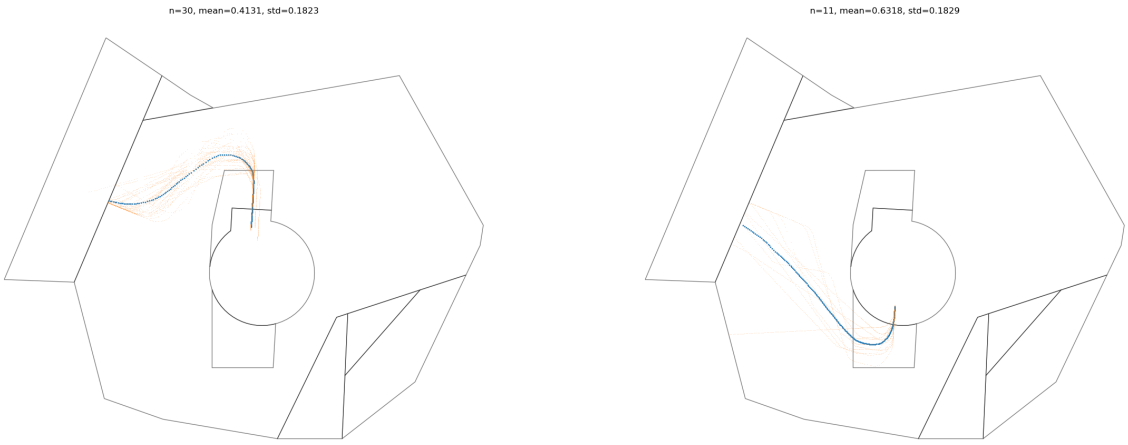
(a) Split 1, cluster 28 **(b)** Split 1, cluster 29

Figure A.6: Conflicts between clusters 28 and 29 (11 conflicts)



(a) Split 1, cluster 26 **(b)** Split 1, cluster 40

Figure A.7: Conflicts between clusters 26 and 40 (10 conflicts)

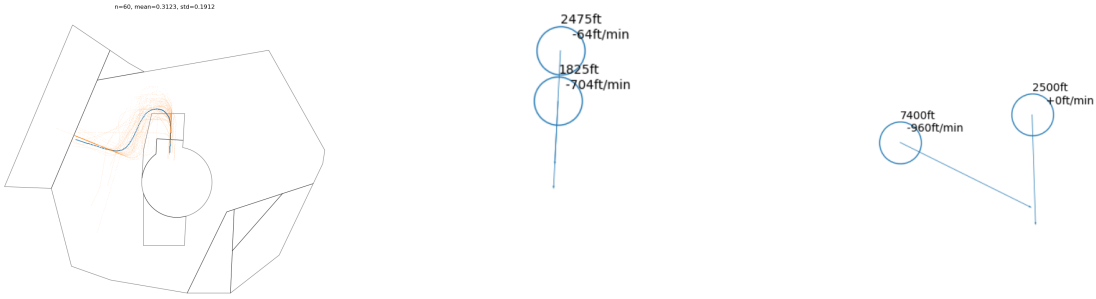


(a) Split 2, cluster 8 **(b)** Split 2, cluster 25

Figure A.8: Conflicts between clusters 2 and 25 (5 conflicts/hr)

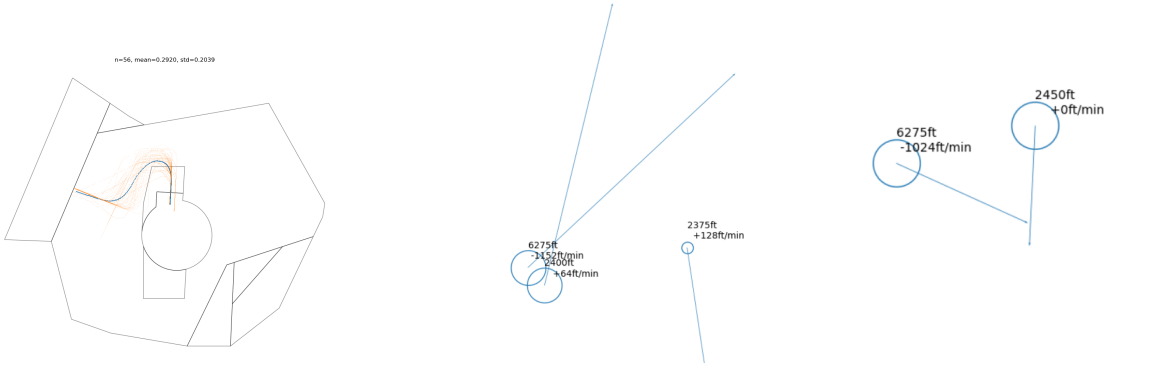
A.3. Conflicts within flows

Note that the between-flow-conflict-detection has been turned off so that these plots only show within-flow conflicts.



(a) Cluster tracks **(b)** Final approach conflict **(c)** Between final approach and other. Dominant within-flow conflict type

Figure A.9: Conflicts within clusters 7 (Jan 4); 24 within-flow conflicts total

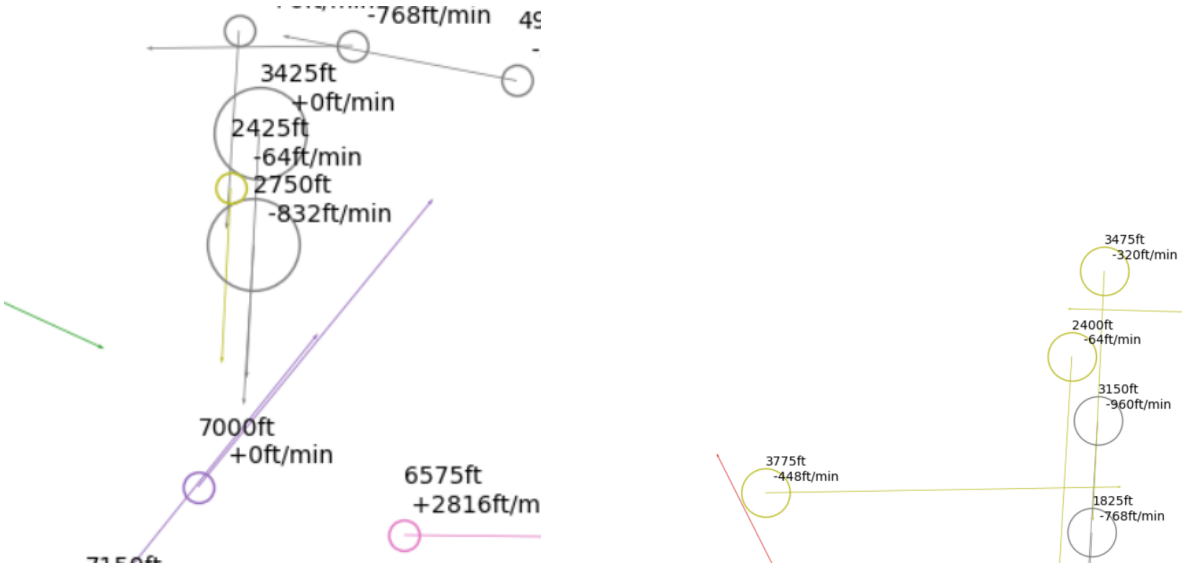


(a) Cluster tracks

(b) Conflict with neither aircraft on Final Approach. Aircraft almost on top of each other, but with a large vertical difference.

(c) Between final approach and other

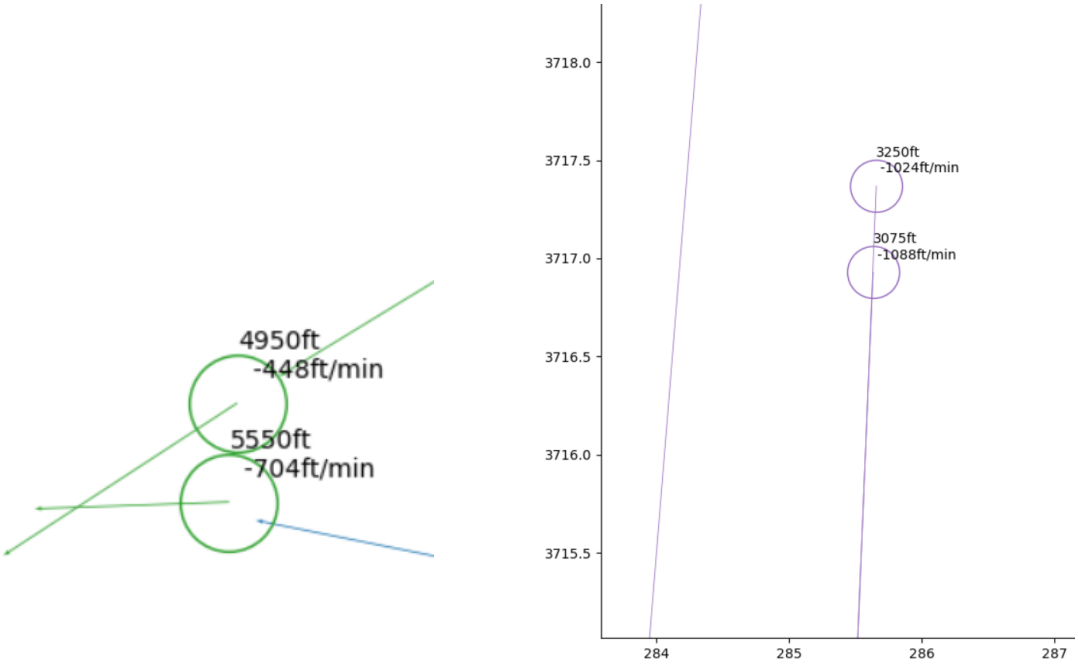
Figure A.10: Conflicts within clusters 38 (Jan 1); 17 within-flow conflicts total



(a) Parallel approach conflict

(b) Final approach conflict

Figure A.11: Two conflicts: one on final approach, another multi-conflict where the left aircraft has a conflict with two other aircraft of the same flow



(a) Merge conflict within flow

(b) Two aircraft of different days on final approach in a combined scenario, less than 500m apart (the axes are in km)

Figure A.12: Combined days: conflicts within flows

B

Comprehensive replay results

B.1. Unaltered

Days: Jan 1, 2, 4, 5; 2018 (replayed separately per day, then summed)

B.1.1. Day-by-day, S_h=3, S_v=1000, t_l=5

Table B.1: Day-by-day, S_h=3, S_v=1000, t_l=5

has_conflicts overlap_type	Total No. of flow pairs	With conflicts	Without conflicts
within-cluster	155	30	125
unclustered-unclustered	4	2	2
clustered-unclustered	155	47	108
all_av	22	0	22
diverging_av	580	13	567
none	1863	158	1705
cross_la	82	18	64
cross	126	19	107
converging_la	66	36	30
converging_av	207	133	74

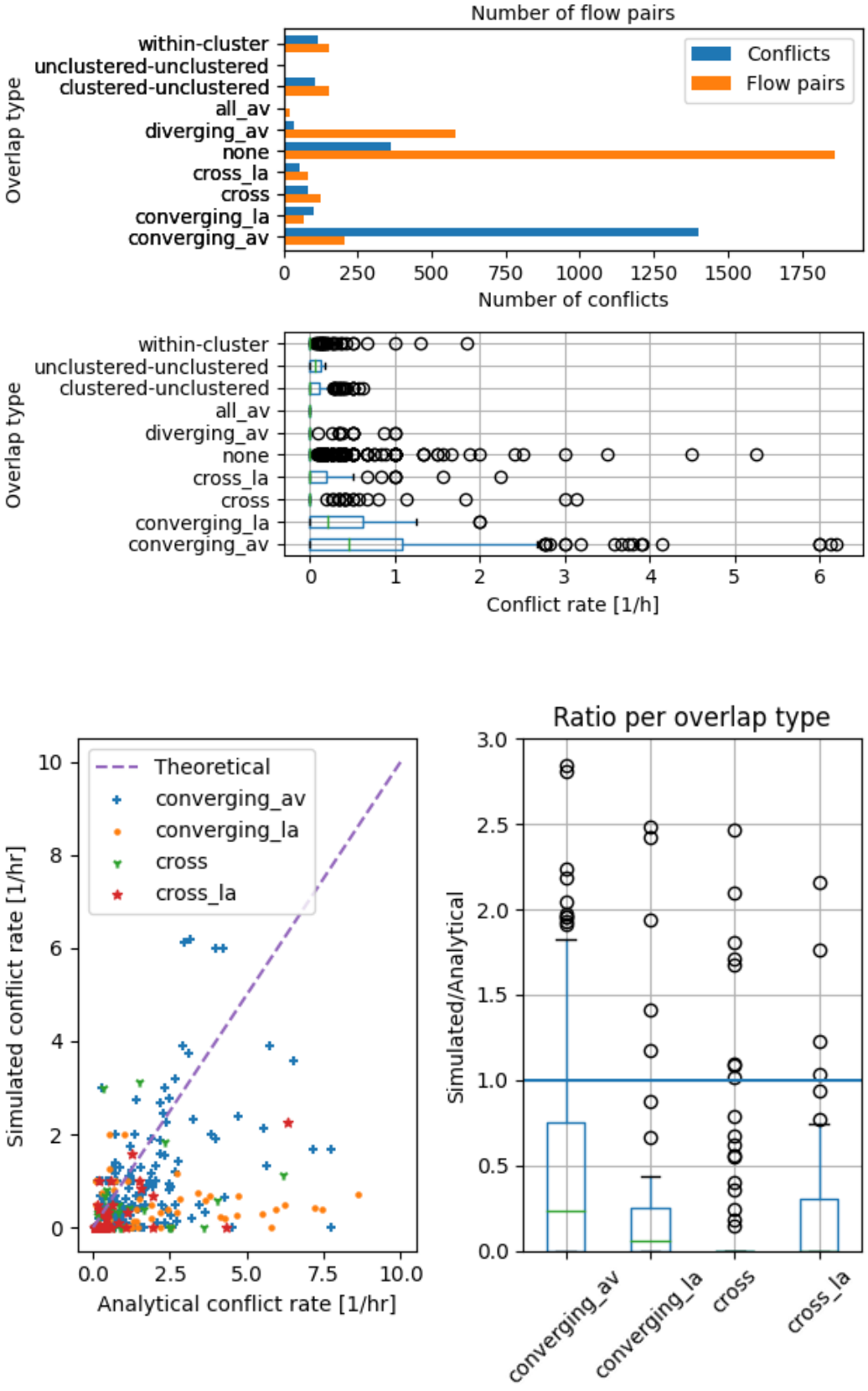


Figure B.1: Conflict counts and rate prediction, all types. Day-by-day, S_h=3, S_v=1000, t_l=5

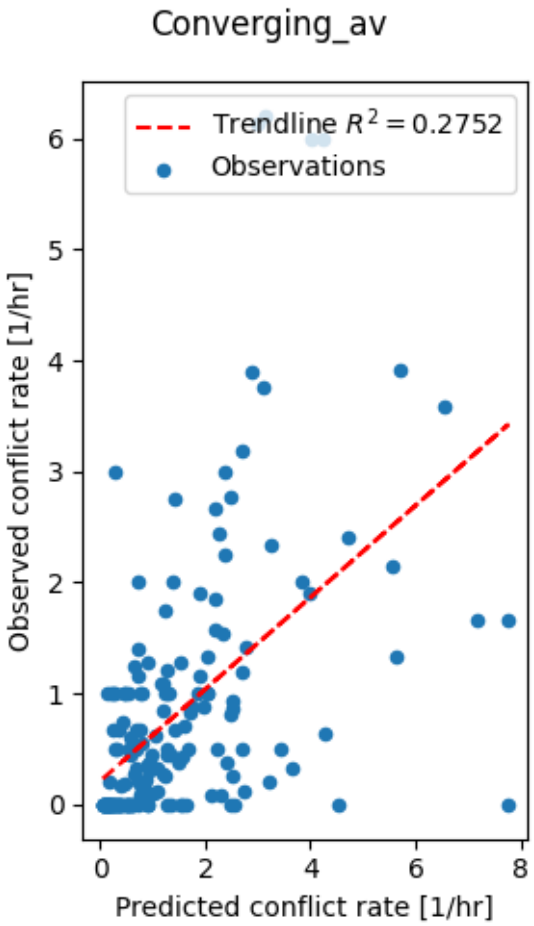


Figure B.2: Conflict rate prediction for `converging_av` type. Day-by-day, S_h=3, S_v=1000, t_l=5

B.1.2. Day-by-day, S_h=3, S_v=1000, t_l=10

Table B.2: Day-by-day, S_h=3, S_v=1000, t_l=10

has_conflicts overlap_type	Total No. of flow pairs	With conflicts	Without conflicts
within-cluster	155	54	101
unclustered-unclustered	4	4	0
clustered-unclustered	155	72	83
all_av	22	0	22
diverging_av	580	25	555
none	1861	232	1629
cross_la	80	13	67
cross	126	20	106
converging_la	70	42	28
converging_av	207	154	53

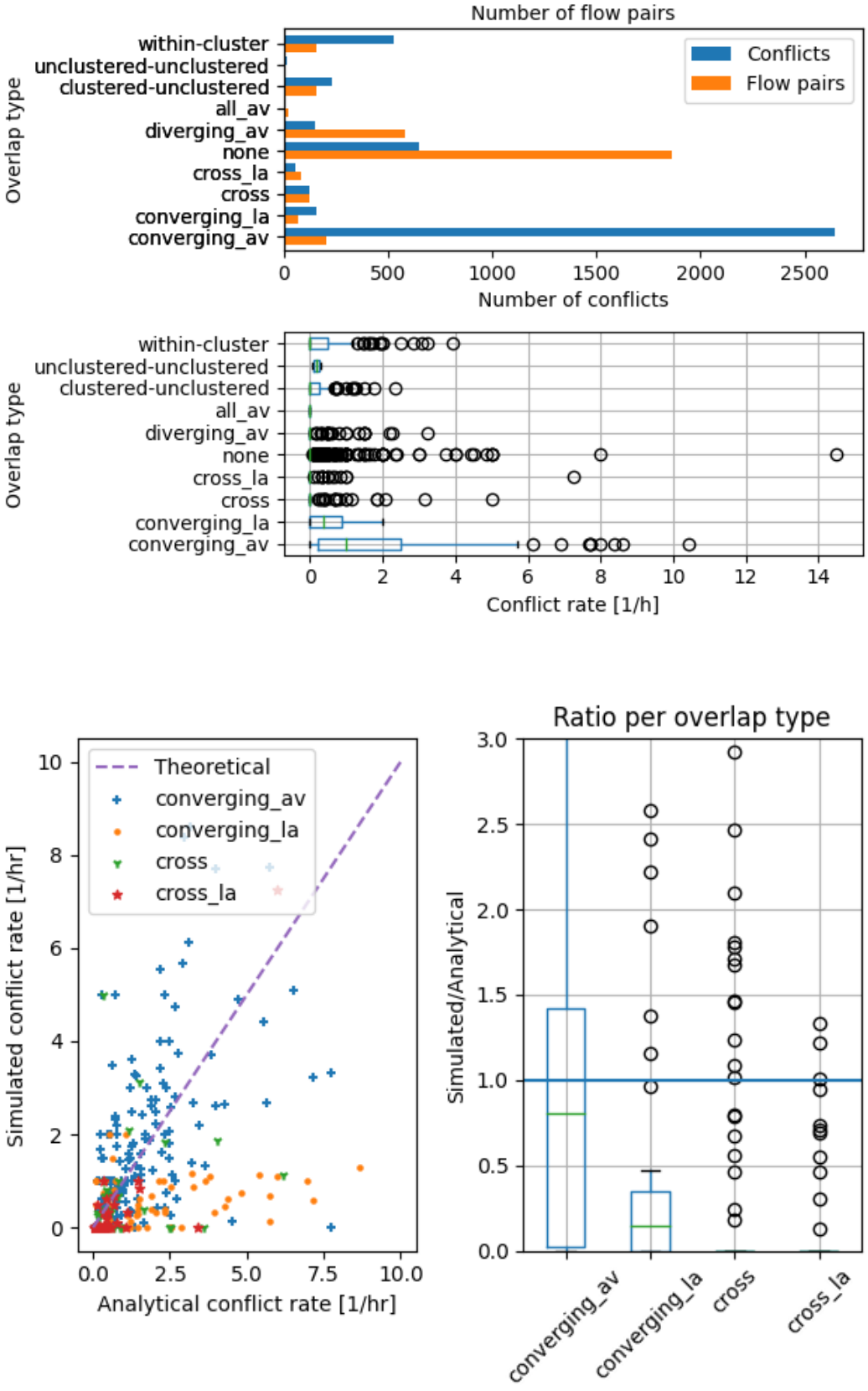


Figure B.3: Conflict counts and rate prediction, all types. Day-by-day, S_h=3, S_v=1000, t_l=10

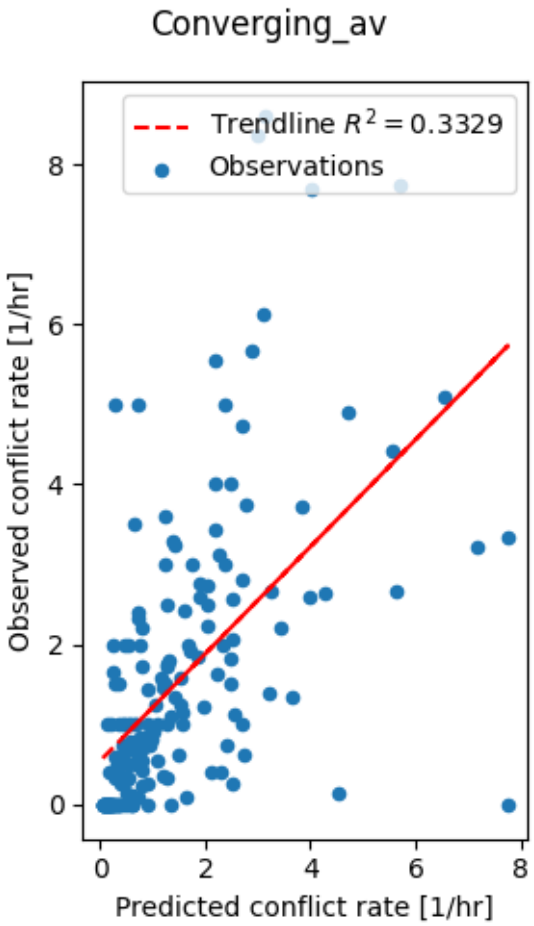


Figure B.4: Conflict rate prediction for `converging_av` type. Day-by-day, S_h=3, S_v=1000, t_l=10

B.1.3. Day-by-day, S_h=6, S_v=1000, t_l=10

Table B.3: Day-by-day, S_h=6, S_v=1000, t_l=10

has_conflicts overlap_type	Total No. of flow pairs	With conflicts	Without conflicts
within-cluster	155	51	104
unclustered-unclustered	4	4	0
clustered-unclustered	155	97	58
all_av	63	2	61
diverging_av	970	184	786
none	1416	238	1178
cross_la	68	22	46
cross	177	46	131
converging_la	68	43	25
converging_av	184	154	30

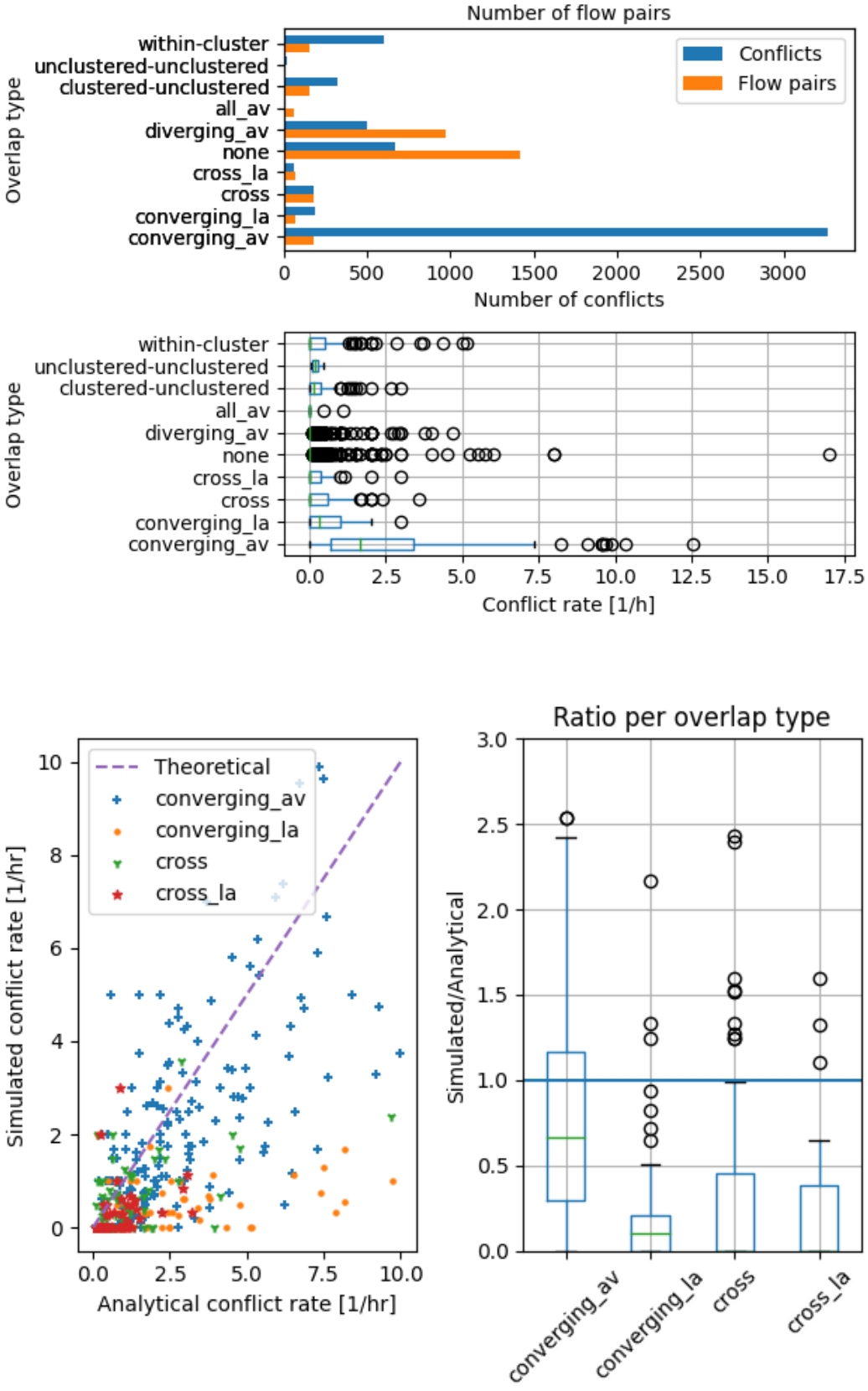


Figure B.5: Conflict counts and rate prediction, all types. Day-by-day, S_h=6, S_v=1000, t_l=10

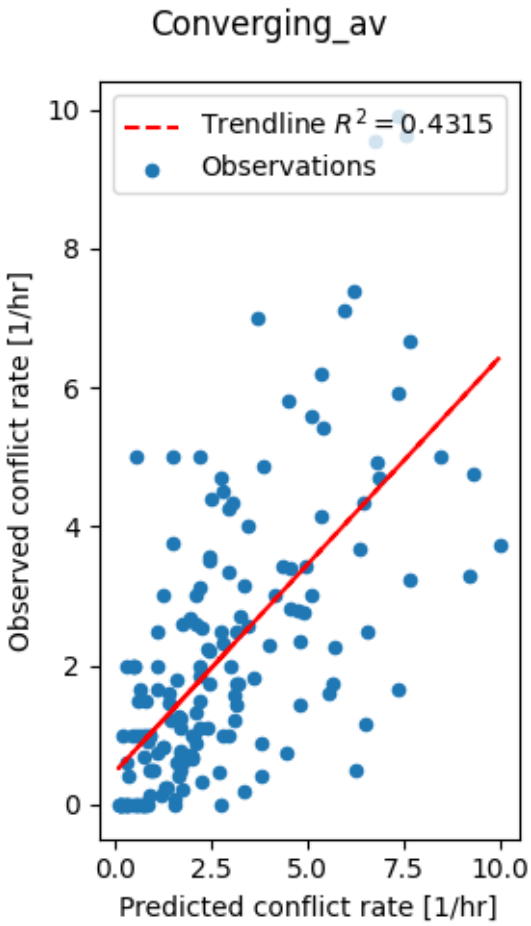


Figure B.6: Conflict rate prediction for `converging_av` type. Day-by-day, S_h=6, S_v=1000, t_l=10

B.1.4. Day-by-day, S_h=6, S_v=2000, t_l=10

Table B.4: Day-by-day, S_h=6, S_v=2000, t_l=10

has_conflicts overlap_type	Total No. of flow pairs	With conflicts	Without conflicts
within-cluster	155	51	104
unclustered-unclustered	4	4	0
clustered-unclustered	155	102	53
all_av	68	2	66
diverging_av	968	200	768
none	1398	254	1144
cross_la	56	19	37
cross	198	56	142
converging_la	53	43	10
converging_av	205	160	45

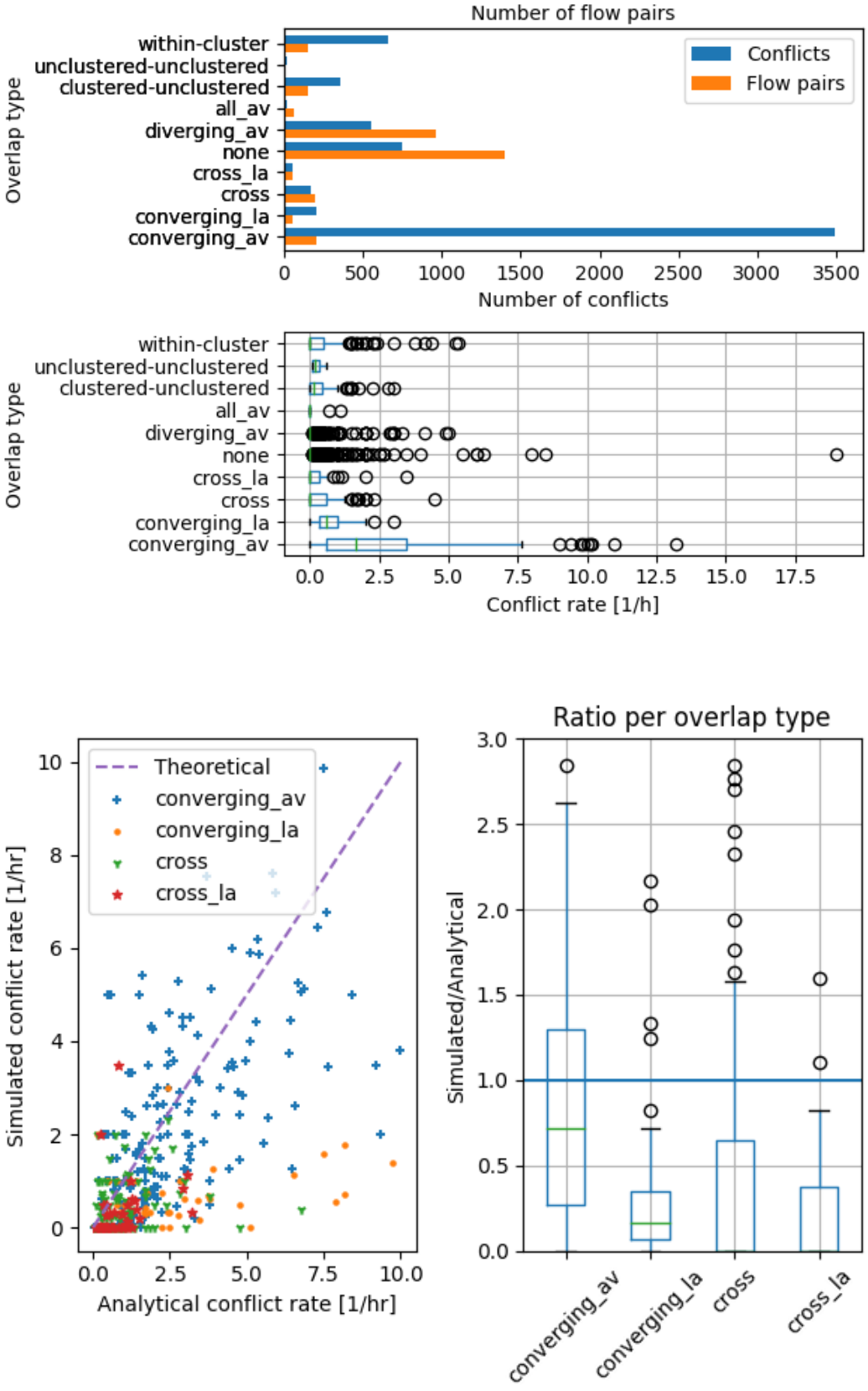


Figure B.7: Conflict counts and rate prediction, all types. Day-by-day, S_h=6, S_v=2000, t_l=10

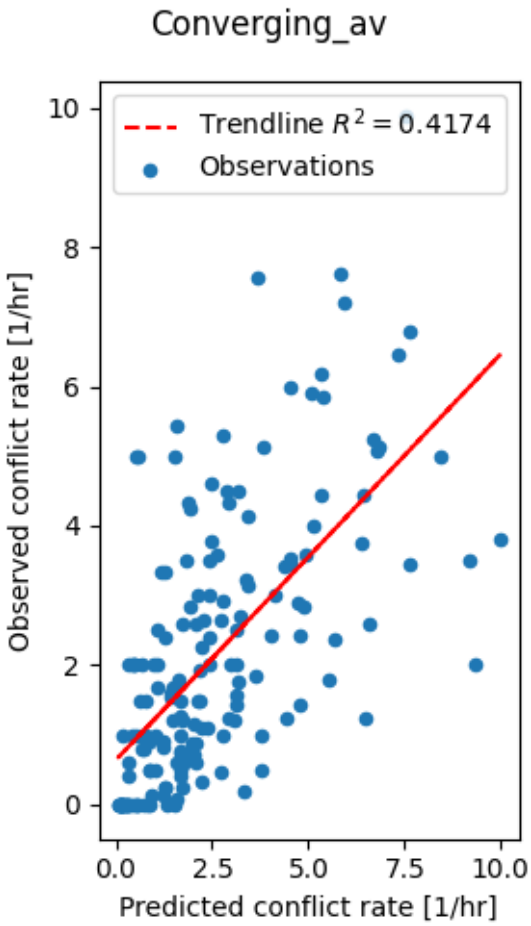


Figure B.8: Conflict rate prediction for `converging_av` type. Day-by-day, S_h=6, S_v=2000, t_l=10

B.2. Combined days

B.2.1. Combined days, S_h=3, S_v=1000, t_l=5

Table B.5: Combined days, S_h=3, S_v=1000, t_l=5

has_conflicts overlap_type	Total No. of flow pairs	With conflicts	Without conflicts
within-cluster	163	72	91
unclustered-unclustered	4	4	0
clustered-unclustered	163	88	75
all_av	18	5	13
diverging_av	567	145	422
none	2100	261	1839
cross_la	92	37	55
cross	155	78	77
converging_la	111	63	48
converging_av	218	151	67

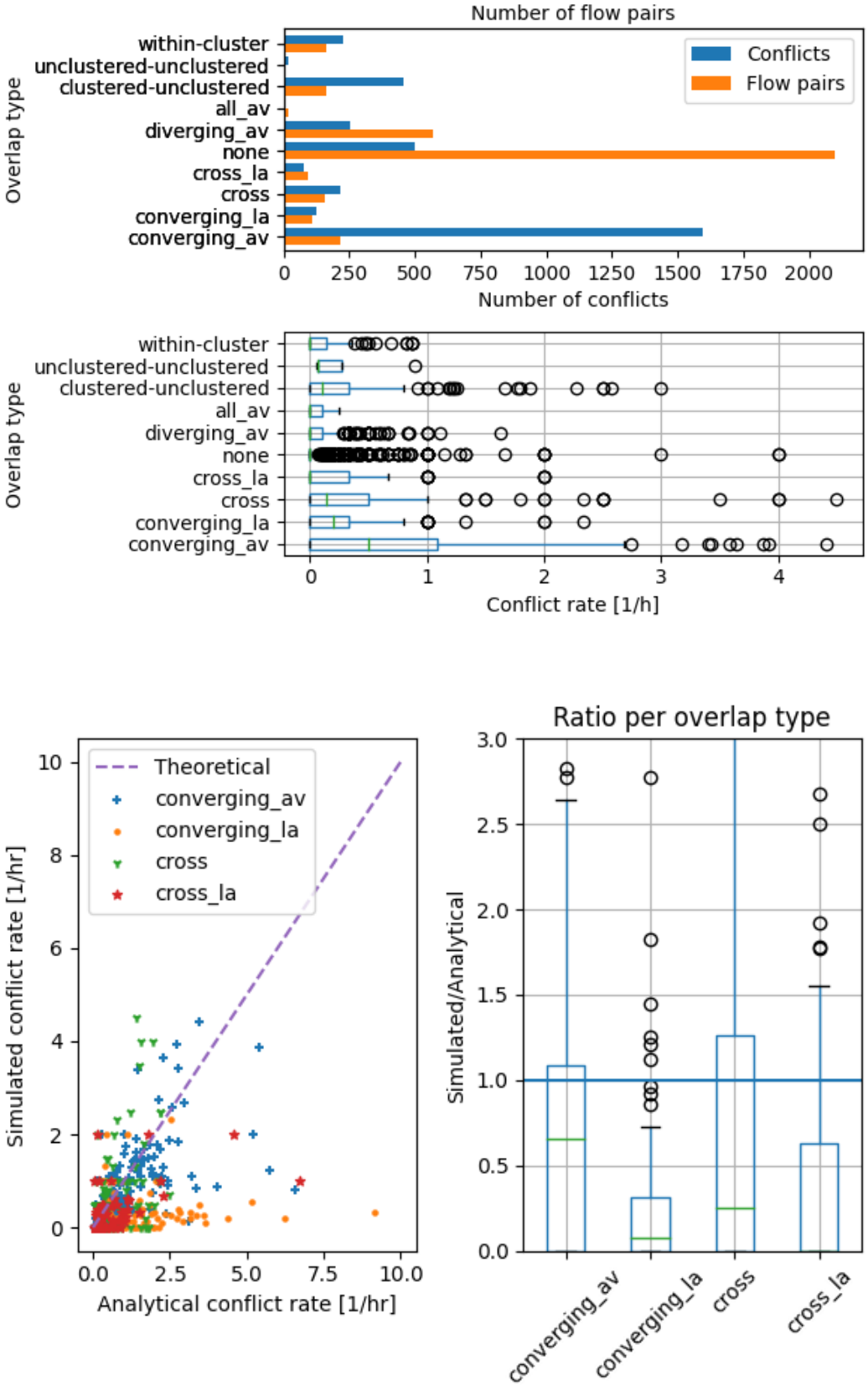


Figure B.9: Conflict counts and rate prediction, all types. Combined days, S_h=3, S_v=1000, t_l=5

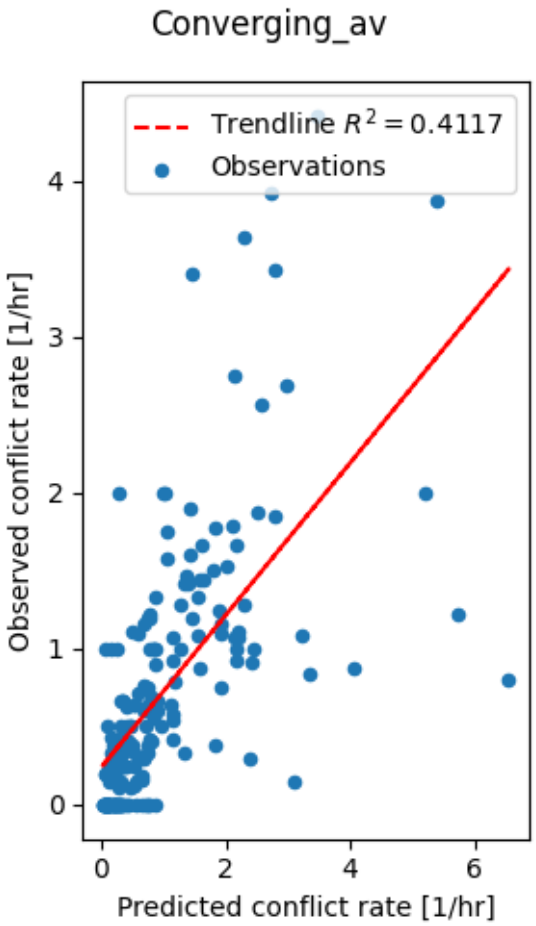


Figure B.10: Conflict rate prediction for `converging_av` type. Combined days, S_h=3, S_v=1000, t_l=5

B.2.2. Combined days, S_h=3, S_v=1000, t_l=10

Table B.6: Combined days, S_h=3, S_v=1000, t_l=10

has_conflicts overlap_type	Total No. of flow pairs	With conflicts	Without conflicts
within-cluster	163	87	76
unclustered-unclustered	4	4	0
clustered-unclustered	163	108	55
all_av	18	6	12
diverging_av	567	154	413
none	2083	340	1743
cross_la	104	47	57
cross	155	85	70
converging_la	116	70	46
converging_av	218	164	54

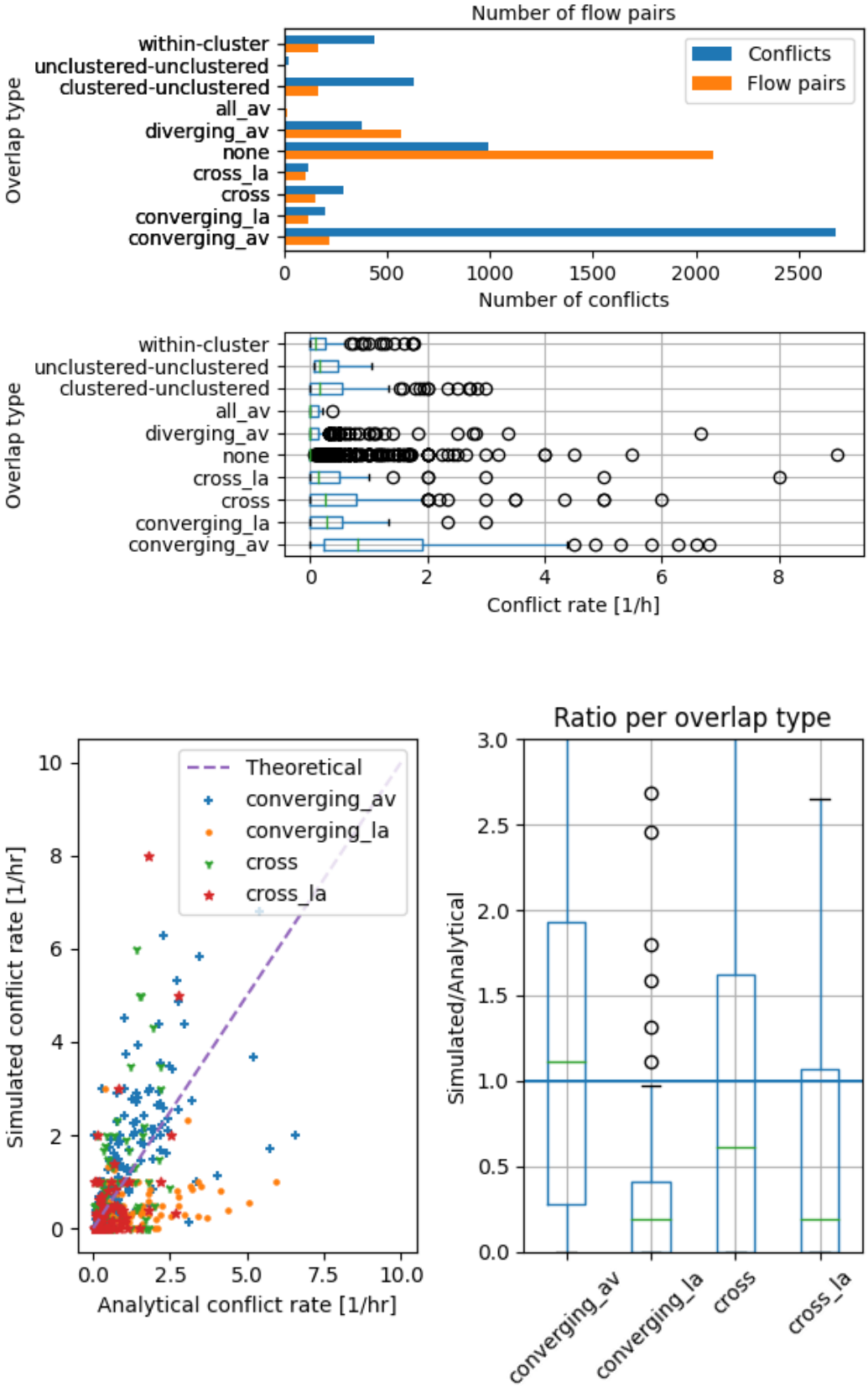


Figure B.11: Conflict counts and rate prediction, all types. Combined days, S_h=3, S_v=1000, t_l=10

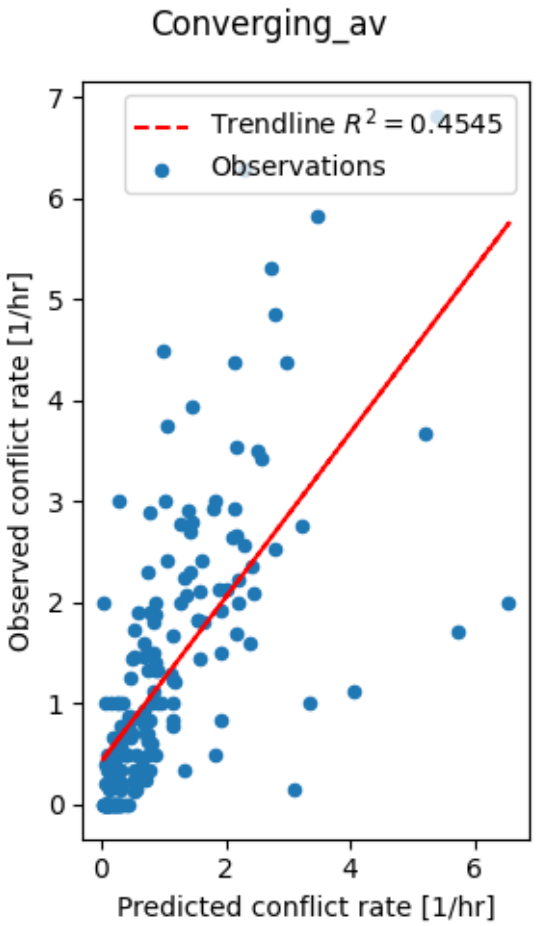


Figure B.12: Conflict rate prediction for `converging_av` type. Combined days, S_h=3, S_v=1000, t_l=10

B.2.3. Combined days, S_h=3, S_v=2000, t_l=5

Table B.7: Combined days, S_h=3, S_v=2000, t_l=5

has_conflicts overlap_type	Total No. of flow pairs	With conflicts	Without conflicts
within-cluster	163	72	91
unclustered-unclustered	4	4	0
clustered-unclustered	163	96	67
all_av	18	5	13
diverging_av	567	159	408
none	2081	268	1813
cross_la	66	31	35
cross	196	105	91
converging_la	96	57	39
converging_av	237	160	77

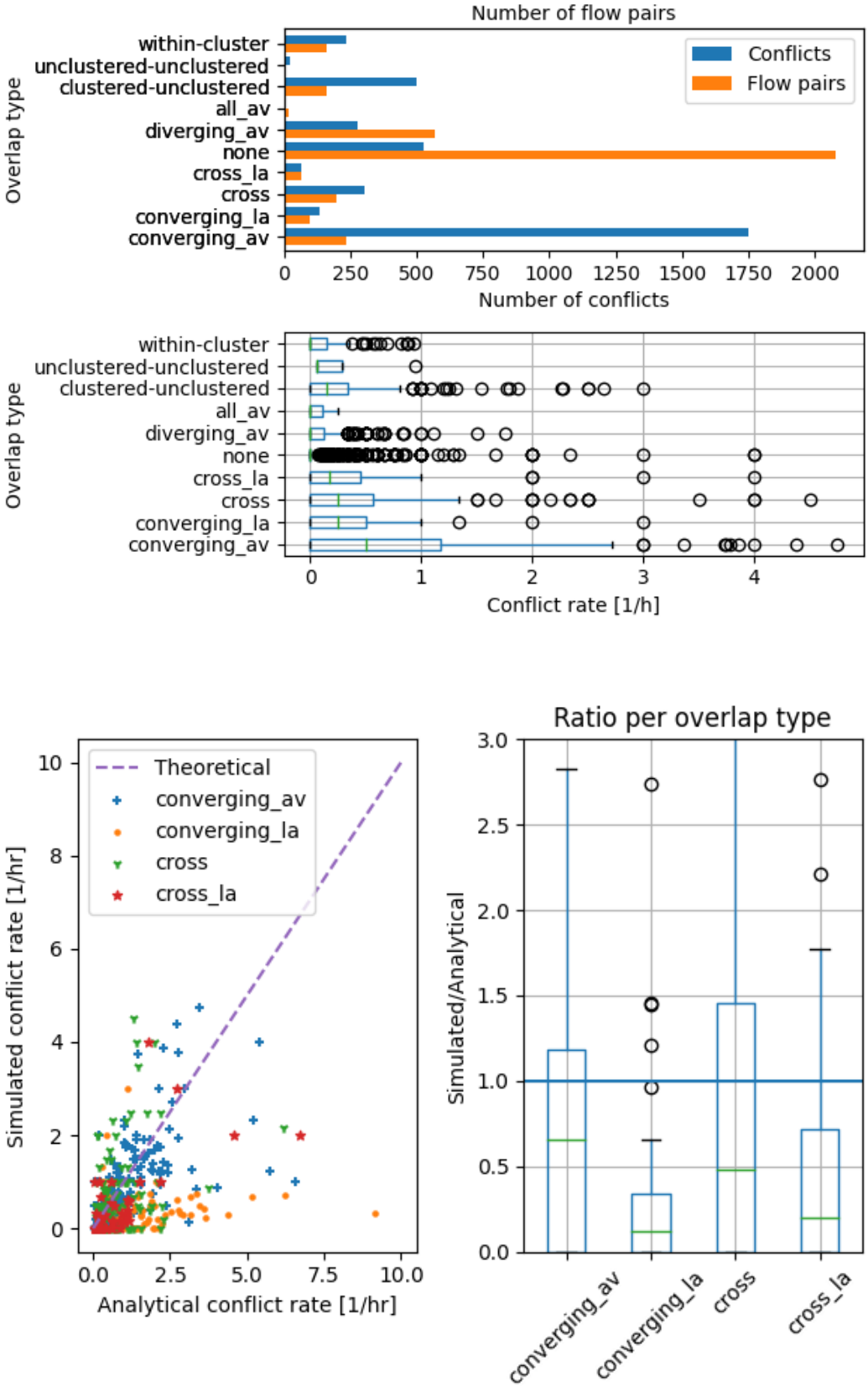


Figure B.13: Conflict counts and rate prediction, all types. Combined days, S_h=3, S_v=2000, t_l=5

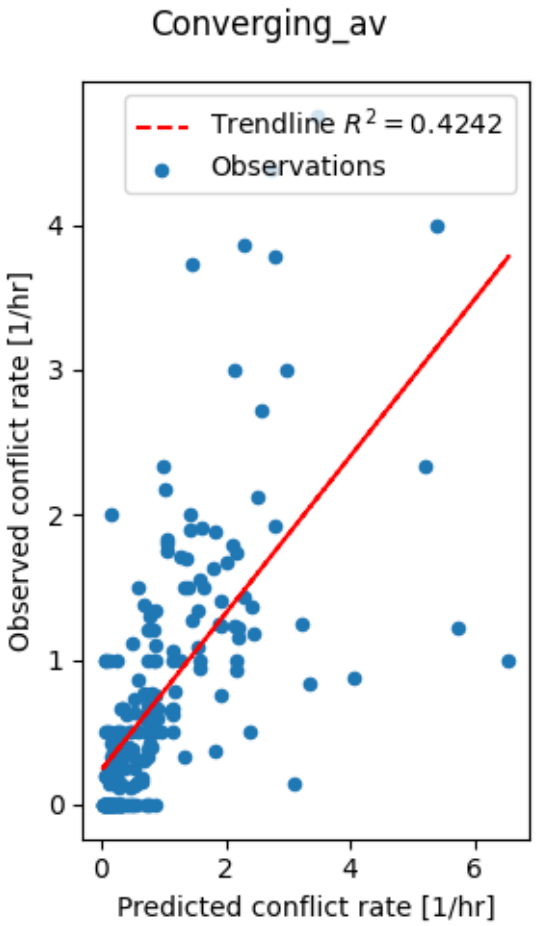


Figure B.14: Conflict rate prediction for `converging_av` type. Combined days, S_h=3, S_v=2000, t_l=5

B.2.4. Combined days, S_h=3, S_v=2000, t_l=10

Table B.8: Combined days, S_h=3, S_v=2000, t_l=10

has_conflicts overlap_type	Total No. of flow pairs	With conflicts	Without conflicts
within-cluster	163	79	84
unclustered-unclustered	4	4	0
clustered-unclustered	163	111	52
all_av	18	5	13
diverging_av	567	161	406
none	2067	345	1722
cross_la	76	32	44
cross	196	112	84
converging_la	100	63	37
converging_av	237	168	69

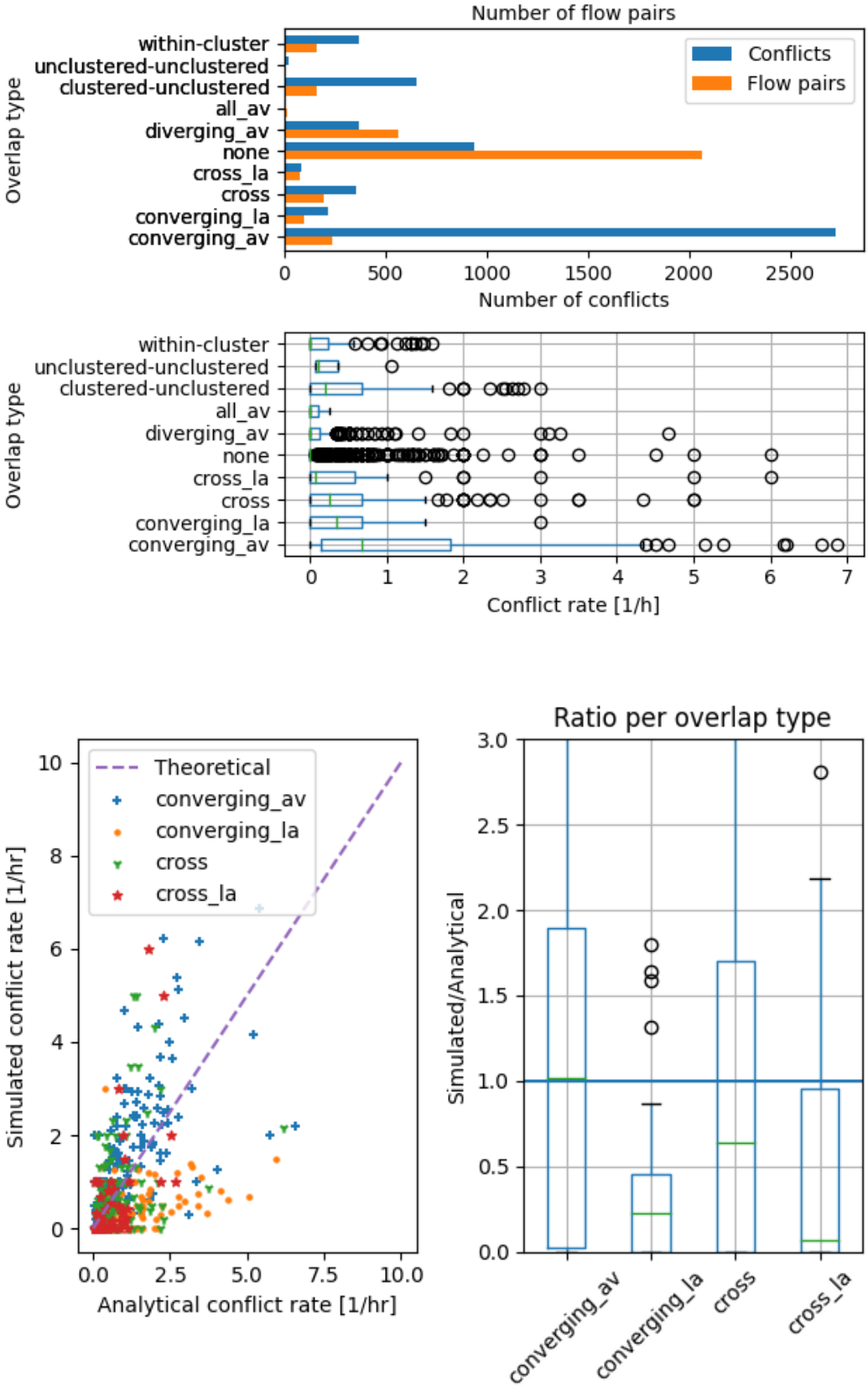


Figure B.15: Conflict counts and rate prediction, all types. Combined days, S_h=3, S_v=2000, t_l=10

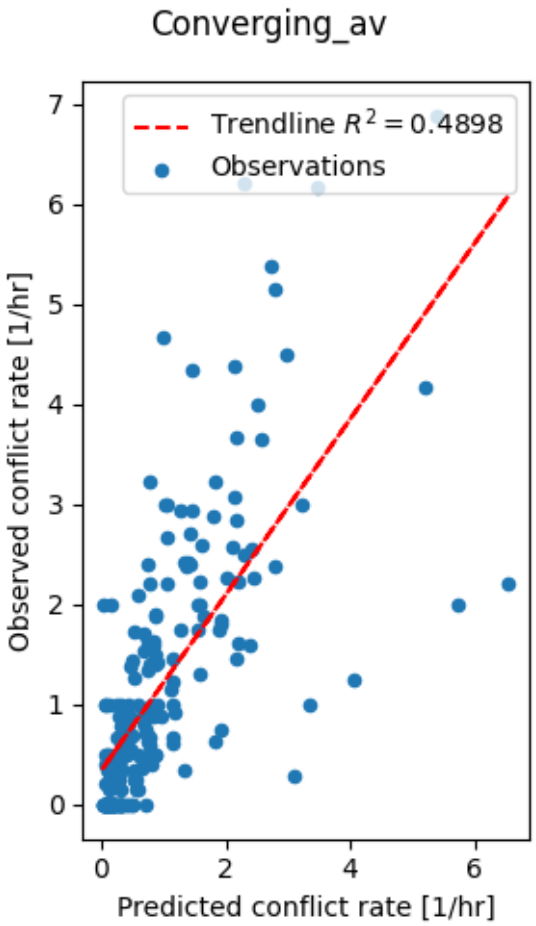


Figure B.16: Conflict rate prediction for `converging_av` type. Combined days, S_h=3, S_v=2000, t_l=10

B.2.5. Combined days, S_h=6, S_v=1000, t_l=10

Table B.9: Combined days, S_h=6, S_v=1000, t_l=10

has_conflicts overlap_type	Total No. of flow pairs	With conflicts	Without conflicts
within-cluster	163	98	65
unclustered-unclustered	4	4	0
clustered-unclustered	163	131	32
all_av	56	29	27
diverging_av	973	349	624
none	1581	341	1240
cross_la	91	50	41
cross	247	145	102
converging_la	105	67	38
converging_av	208	163	45

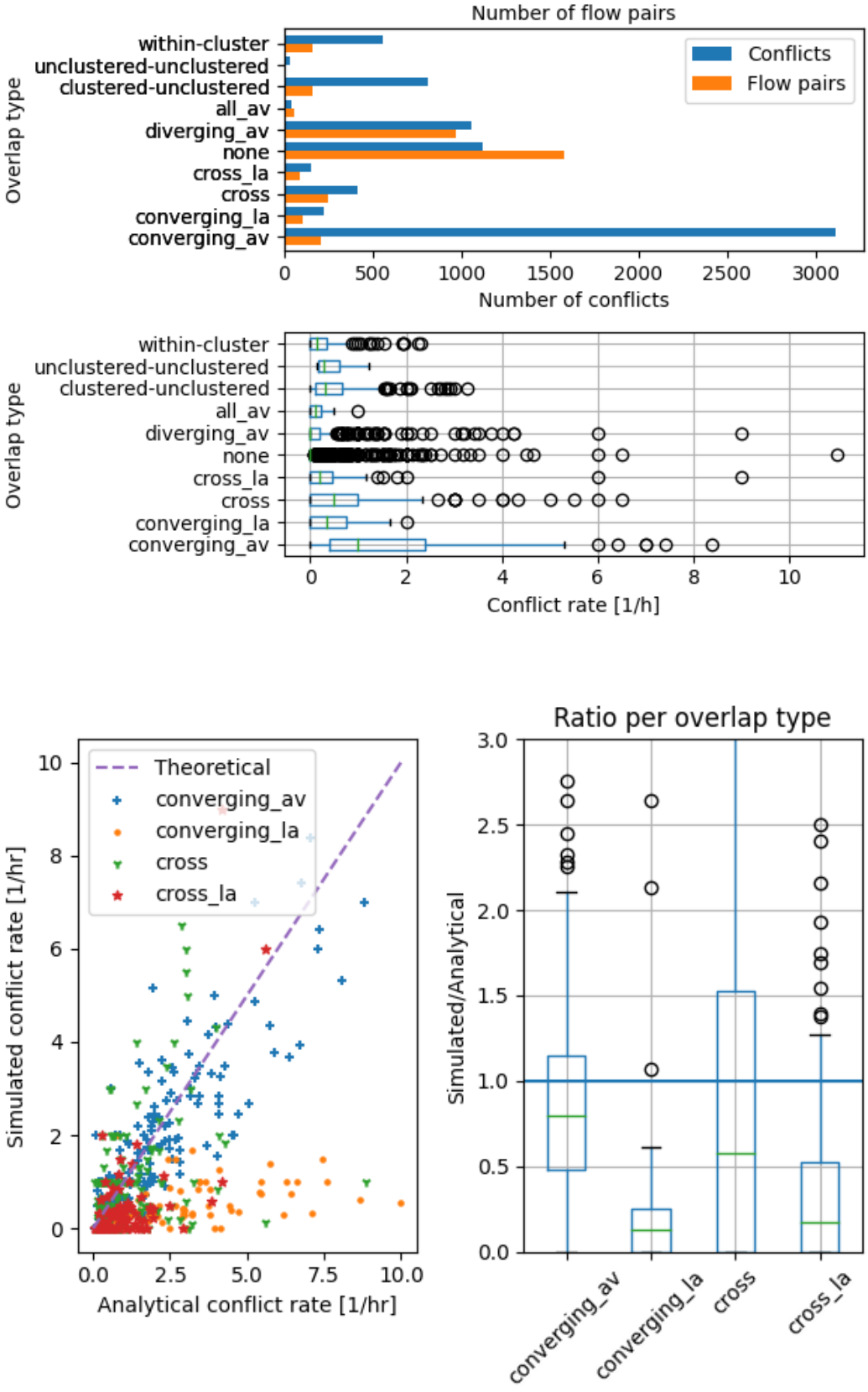


Figure B.17: Conflict counts and rate prediction, all types. Combined days, S_h=6, S_v=1000, t_l=10

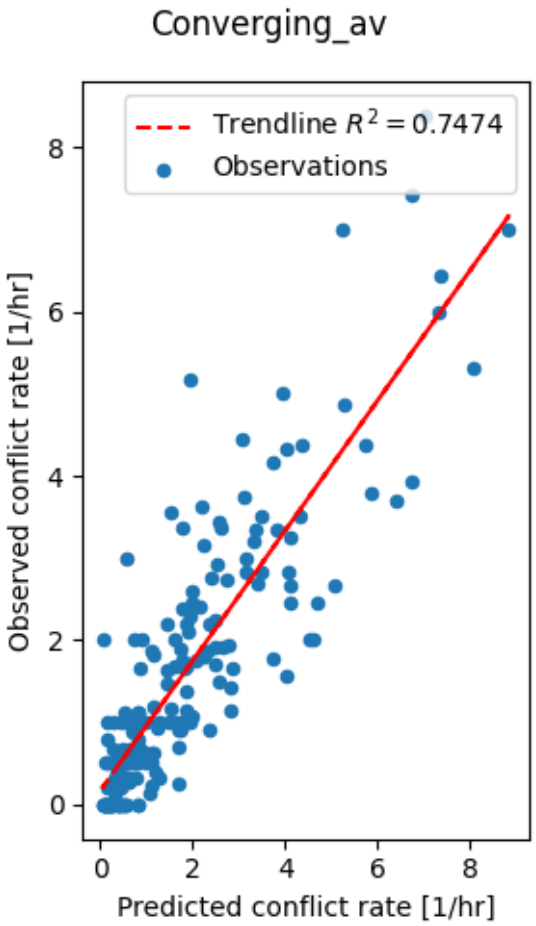


Figure B.18: Conflict rate prediction for `converging_av` type. Combined days, S_h=6, S_v=1000, t_l=10

B.2.6. Combined days, S_h=6, S_v=1000, t_l=10, as events, 5 replications

Table B.10: Combined days, S_h=6, S_v=1000, t_l=10, as events, 5 replications

has_conflicts overlap_type	Total No. of flow pairs	With conflicts	Without conflicts
within-cluster	163	163	0
unclustered-unclustered	4	4	0
clustered-unclustered	163	163	0
all_av	56	56	0
diverging_av	973	967	6
none	1581	924	657
cross_la	91	90	1
cross	247	247	0
converging_la	105	105	0
converging_av	208	208	0

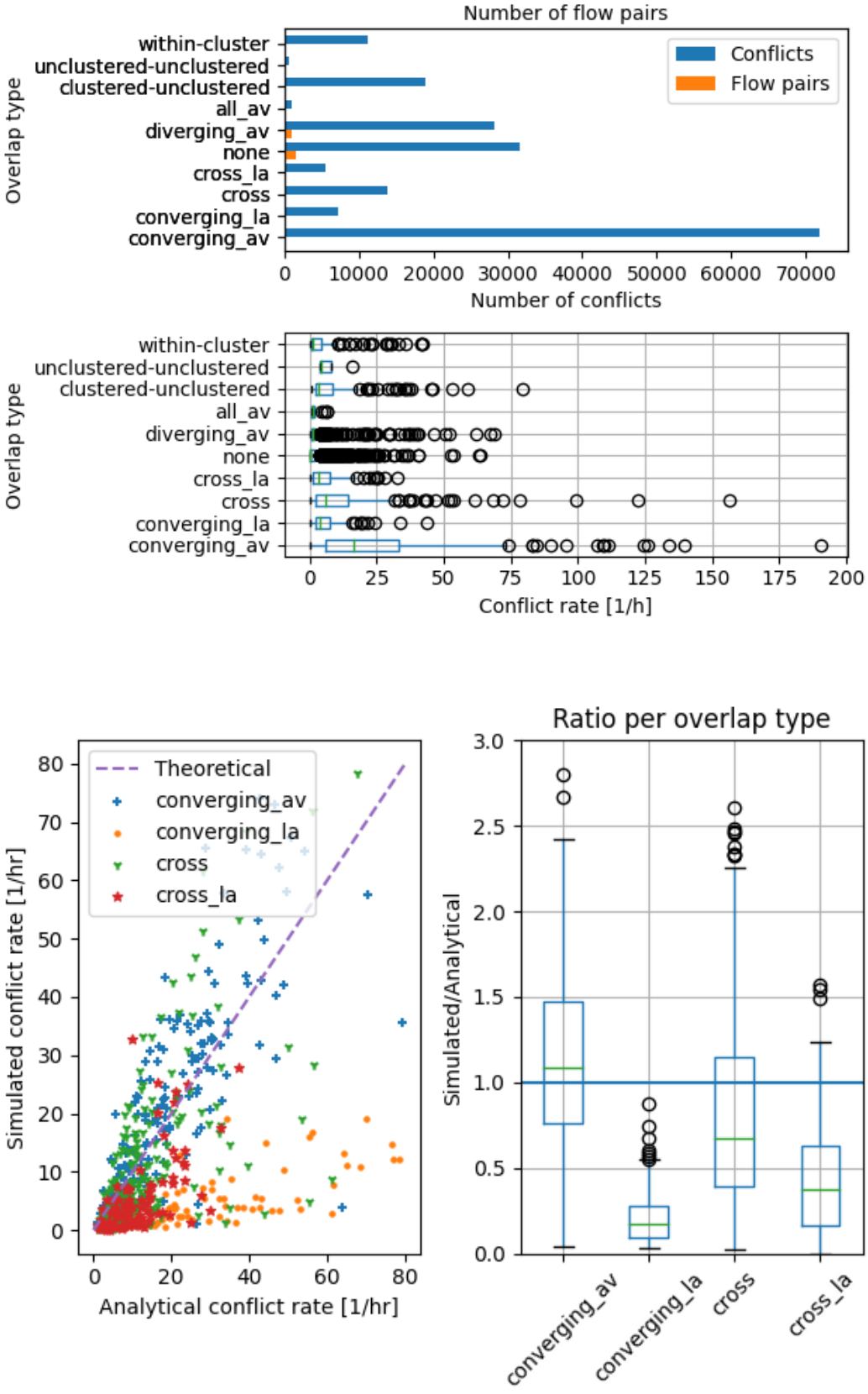


Figure B.19: Conflict counts and rate prediction, all types. Combined days, S_h=6, S_v=1000, t_l=10, as events, 5 replications

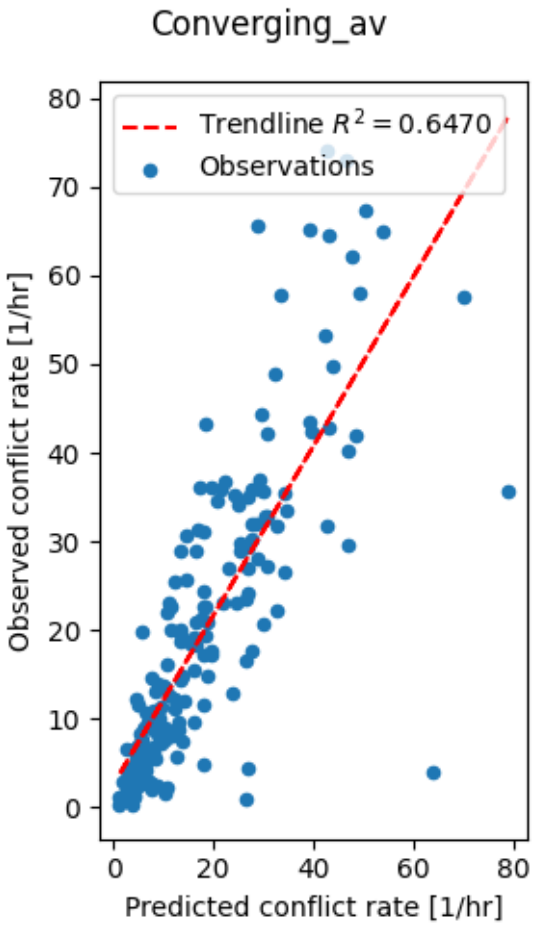


Figure B.20: Conflict rate prediction for `converging_av` type. Combined days, S_h=6, S_v=1000, t_l=10, as events, 5 replications

B.2.7. Combined days, S_h=6, S_v=2000, t_l=5

Table B.11: Combined days, S_h=6, S_v=2000, t_l=5

has_conflicts overlap_type	Total No. of flow pairs	With conflicts	Without conflicts
within-cluster	163	85	78
unclustered-unclustered	4	4	0
clustered-unclustered	163	123	40
all_av	58	26	32
diverging_av	971	363	608
none	1580	288	1292
cross_la	54	28	26
cross	266	147	119
converging_la	93	59	34
converging_av	239	176	63

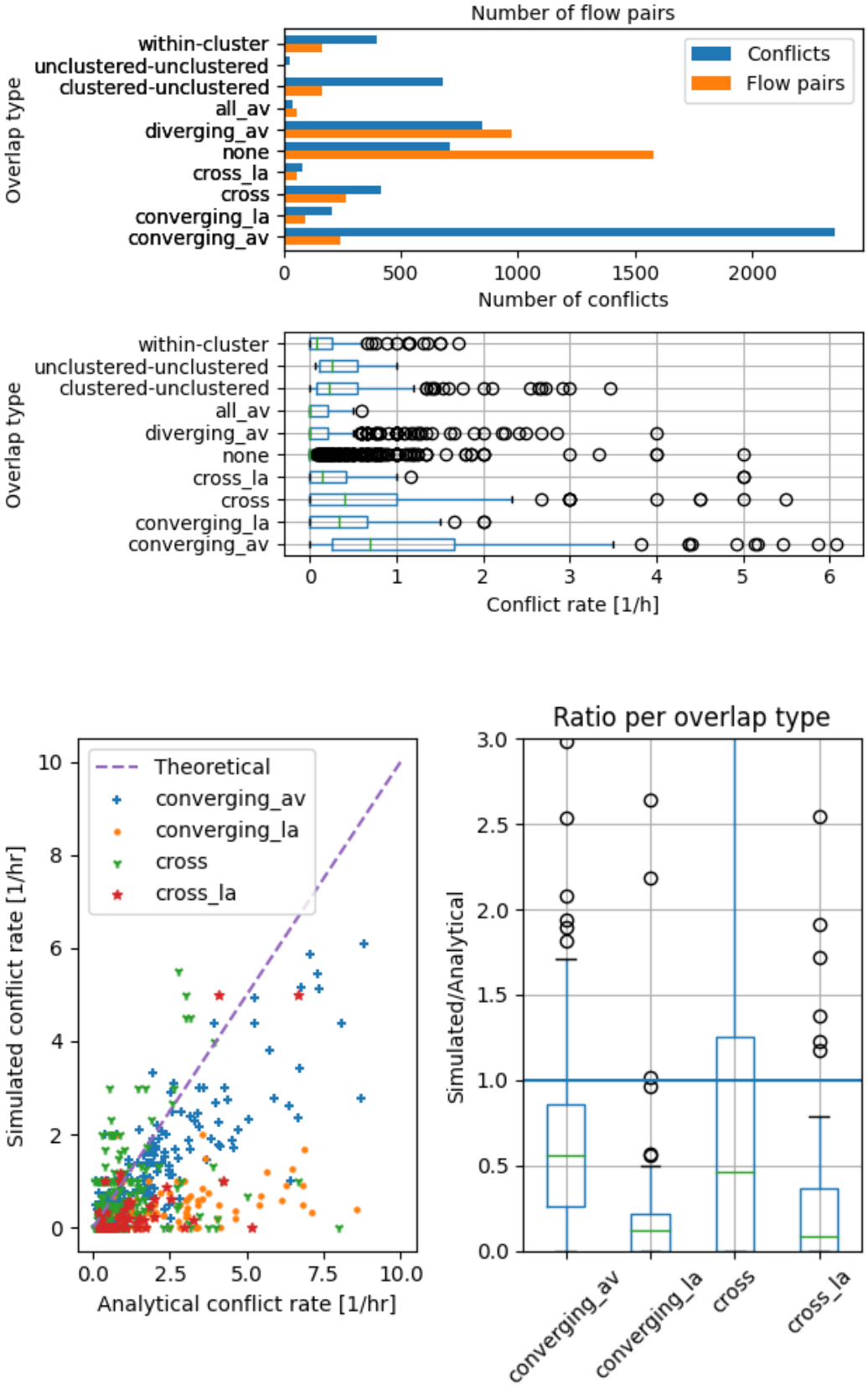


Figure B.21: Conflict counts and rate prediction, all types. Combined days, S_h=6, S_v=2000, t_l=5

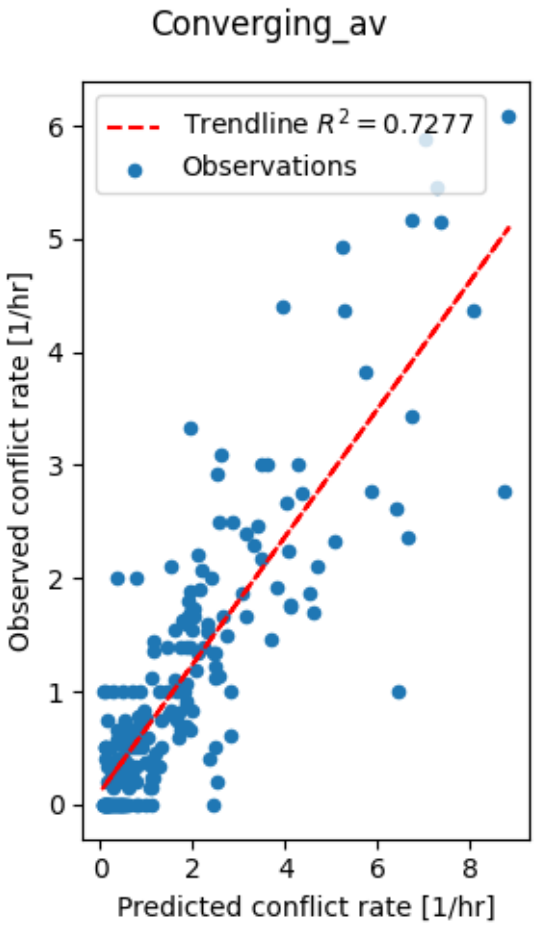


Figure B.22: Conflict rate prediction for `converging_av` type. Combined days, S_h=6, S_v=2000, t_l=5

B.2.8. Combined days, S_h=6, S_v=2000, t_l=10

Table B.12: Combined days, S_h=6, S_v=2000, t_l=10

has_conflicts overlap_type	Total No. of flow pairs	With conflicts	Without conflicts
within-cluster	163	88	75
unclustered-unclustered	4	4	0
clustered-unclustered	163	132	31
all_av	58	26	32
diverging_av	971	367	604
none	1572	353	1219
cross_la	63	32	31
cross	266	159	107
converging_la	92	63	29
converging_av	239	180	59

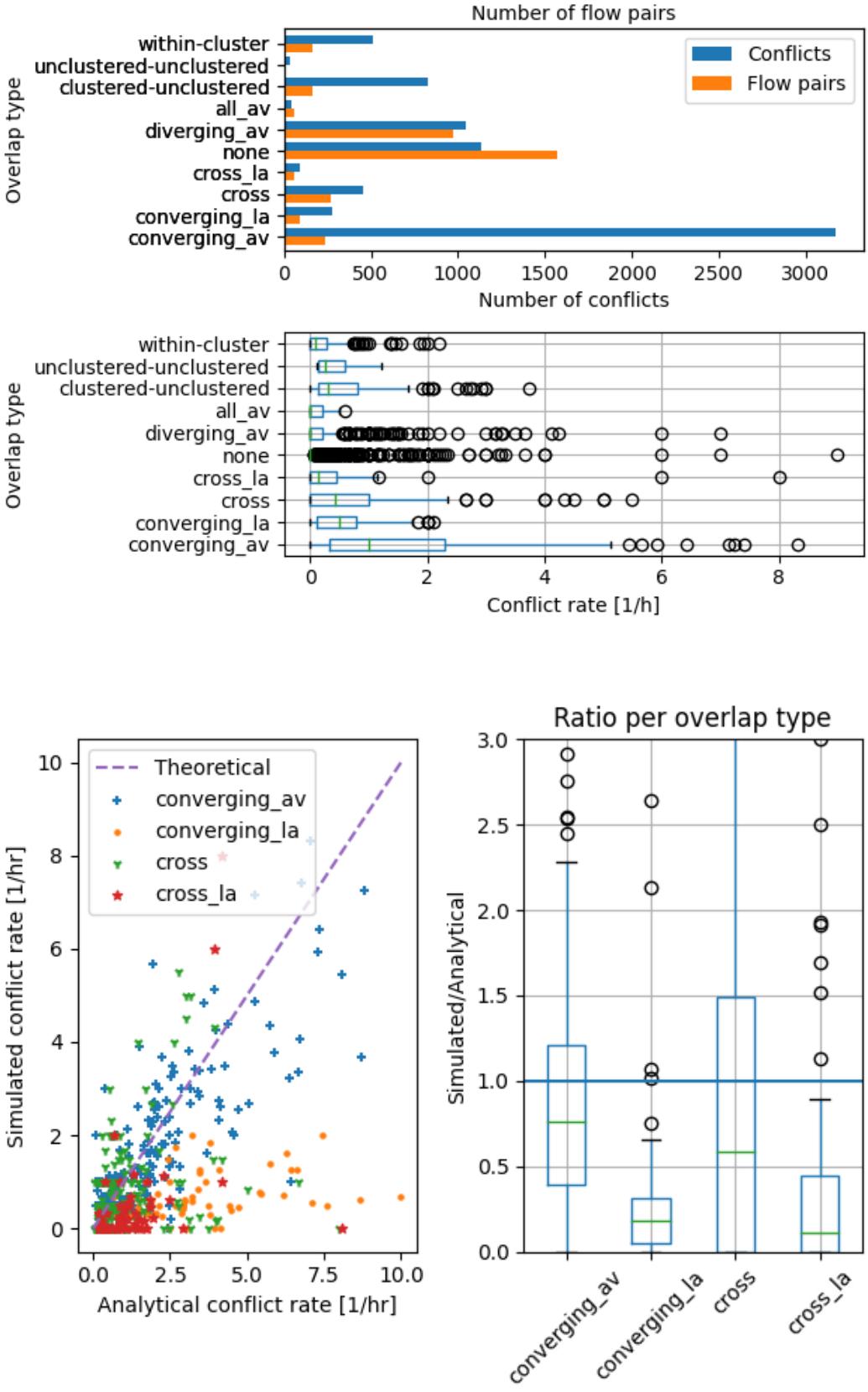


Figure B.23: Conflict counts and rate prediction, all types. Combined days, S_h=6, S_v=2000, t_l=10

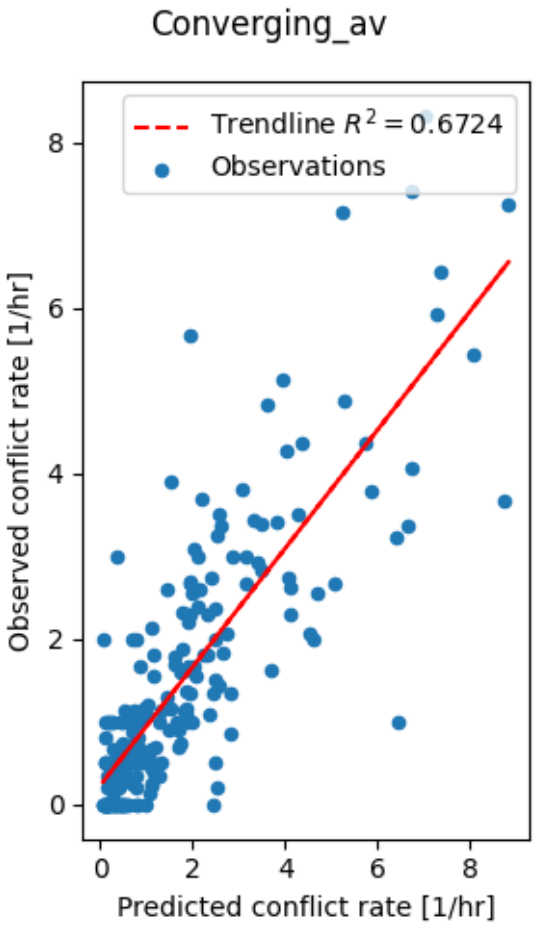


Figure B.24: Conflict rate prediction for `converging_av` type. Combined days, S_h=6, S_v=2000, t_l=10

B.2.9. Combined days, S_h=6, S_v=2000, t_l=10, as events, 5 replications

Table B.13: Combined days, S_h=6, S_v=2000, t_l=10, as events, 5 replications

has_conflicts overlap_type	Total No. of flow pairs	With conflicts	Without conflicts
within-cluster	163	162	1
unclustered-unclustered	4	4	0
clustered-unclustered	163	163	0
all_av	58	58	0
diverging_av	971	961	10
none	1572	925	647
cross_la	63	63	0
cross	266	266	0
converging_la	92	92	0
converging_av	239	239	0

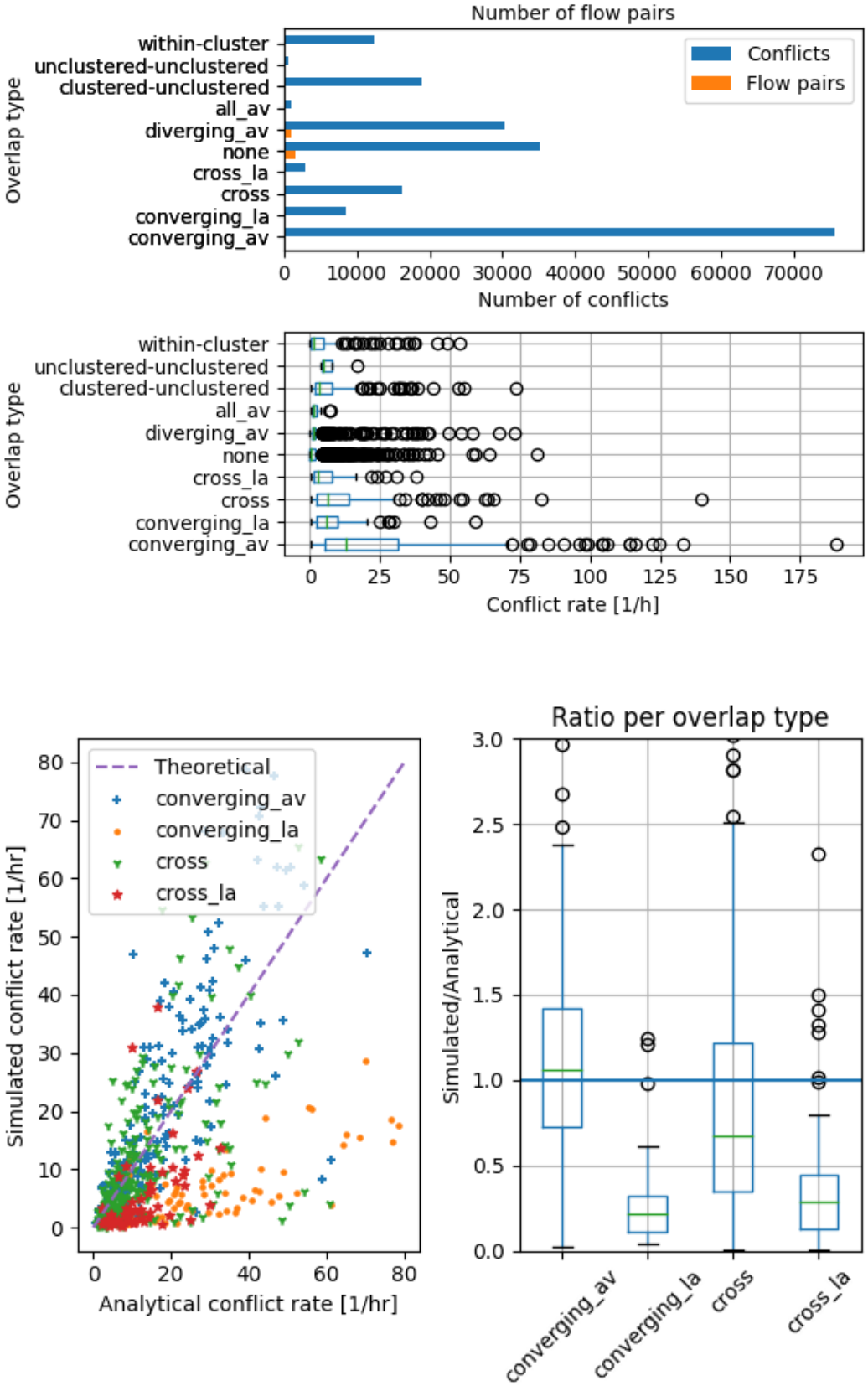


Figure B.25: Conflict counts and rate prediction, all types. Combined days, S_h=6, S_v=2000, t_l=10, as events, 5 replications

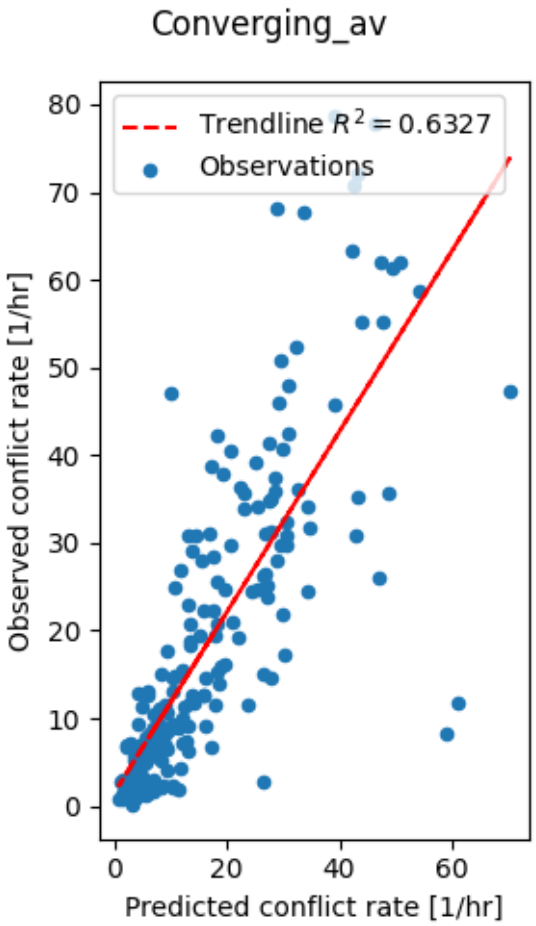


Figure B.26: Conflict rate prediction for `converging_av` type. Combined days, S_h=6, S_v=2000, t_l=10, as events, 5 replications

C

Additional conflict simulation results

C.1. Low radius or settling time

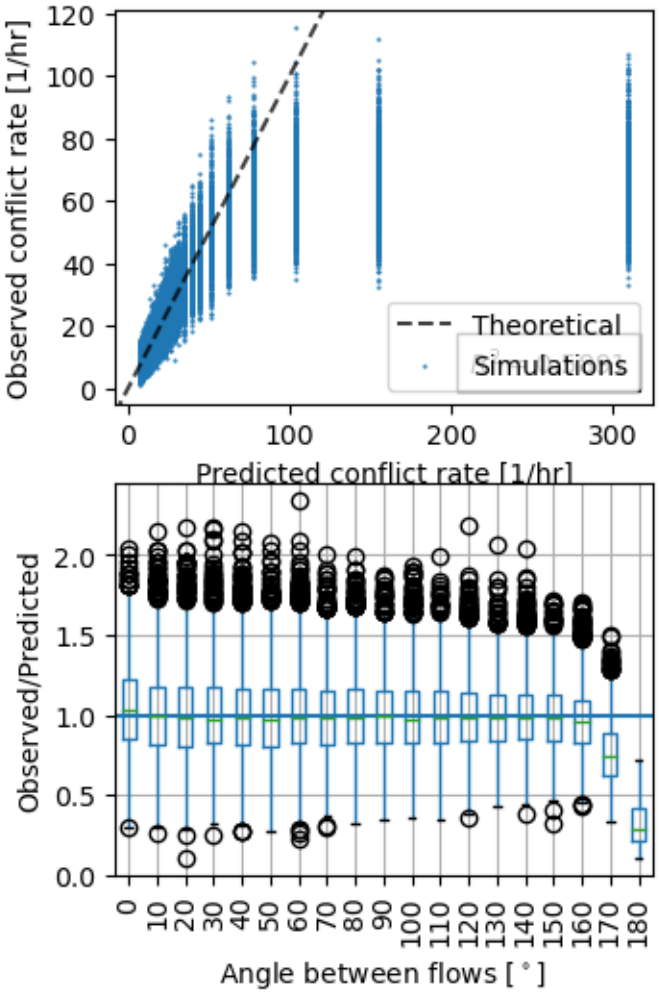


Figure C.1: Predicted and simulated conflicts for varying α with $R_0 = 20$ nm and $T_{settle} = 1$ and $T_{log} = 4$.

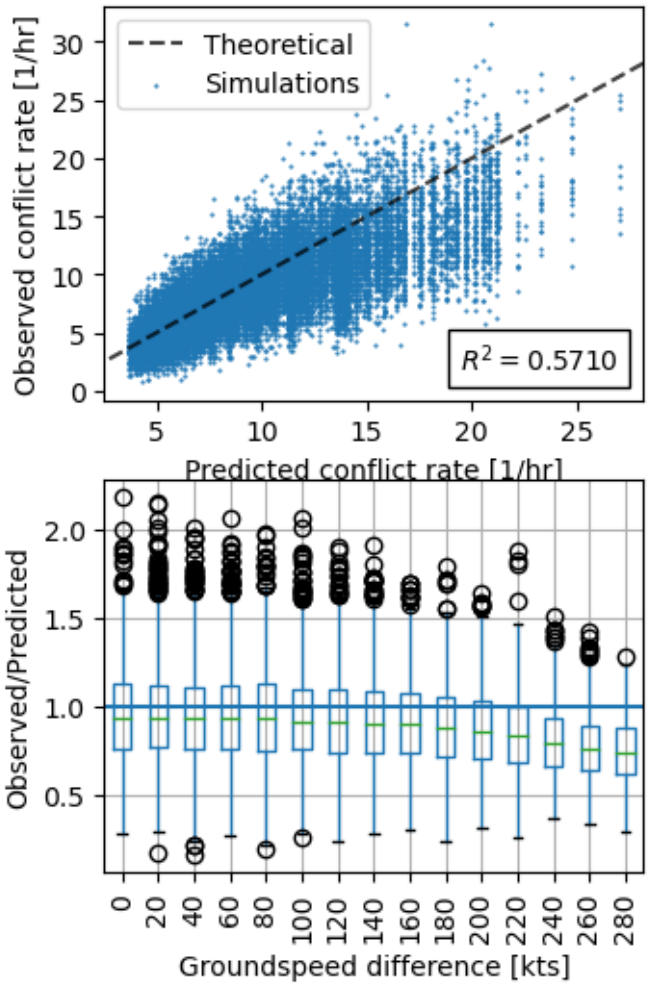


Figure C.2: Predicted and simulated conflicts for varying V, with $R_0 = 200$ nm and $T_{settle} = 1$ and $T_{log} = 4$.

C.2. Lower groundspeeds

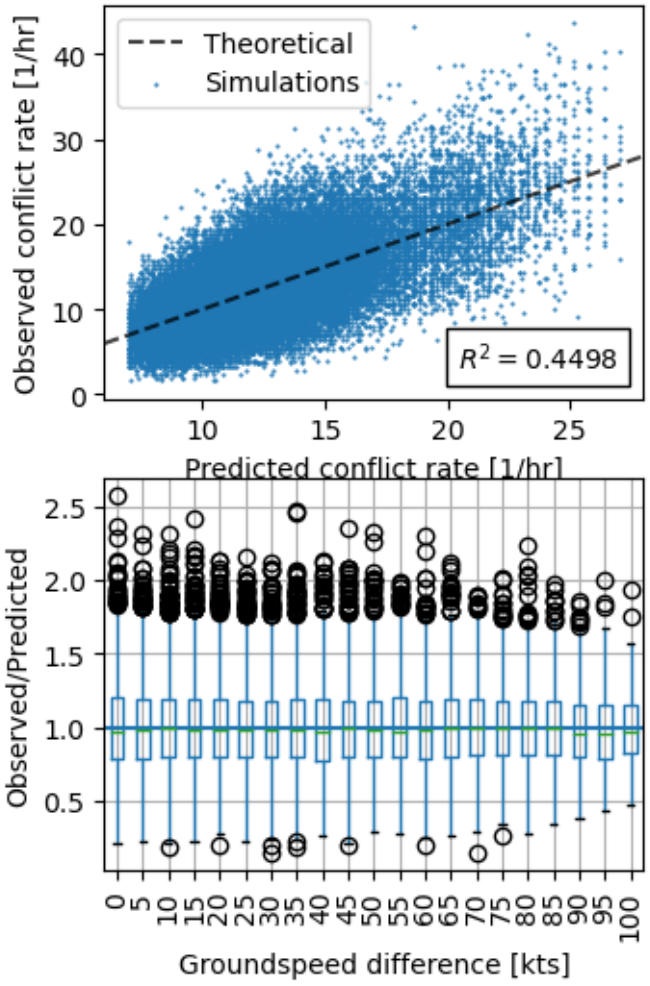


Figure C.3: Predicted and simulated conflicts for varying V between 100 and 200 knots, with $S_h = 3$ nm and $T_{settle} = T_{log} = 2.5$.

D

Clustering results

D.1. Days seperately

D.1.1. 01-01-2018.

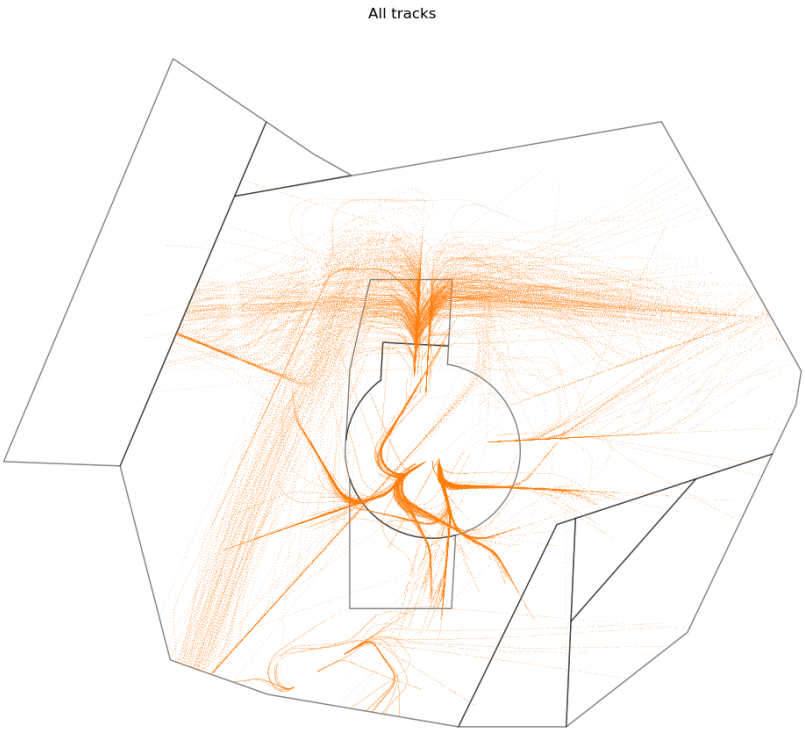


Figure D.1: All flight tracks, 01-01-2018, unclustered.

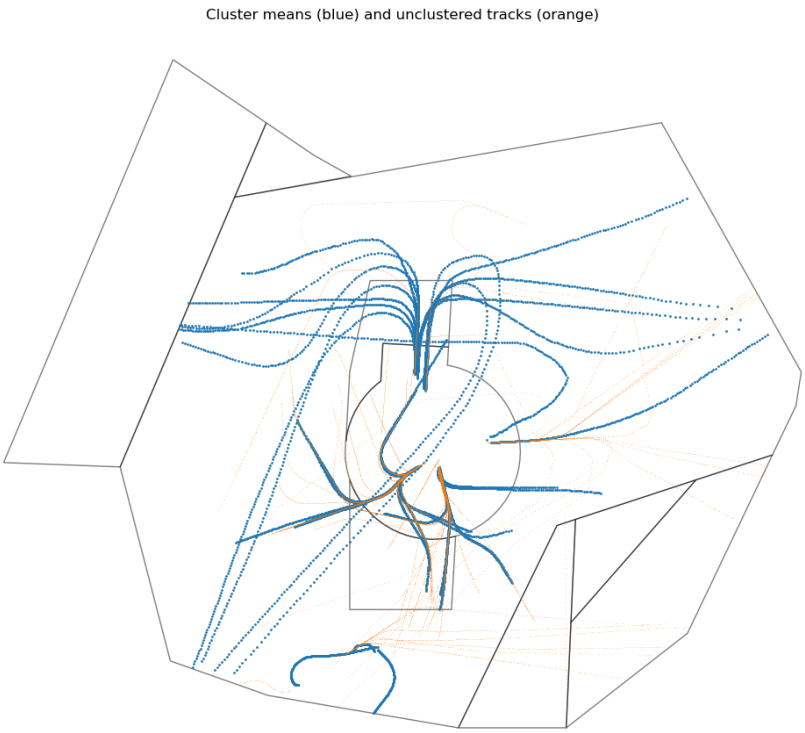


Figure D.2: Cluster means and unclustered tracks, 01-01-2018.

D.1.2. 02-01-2018.

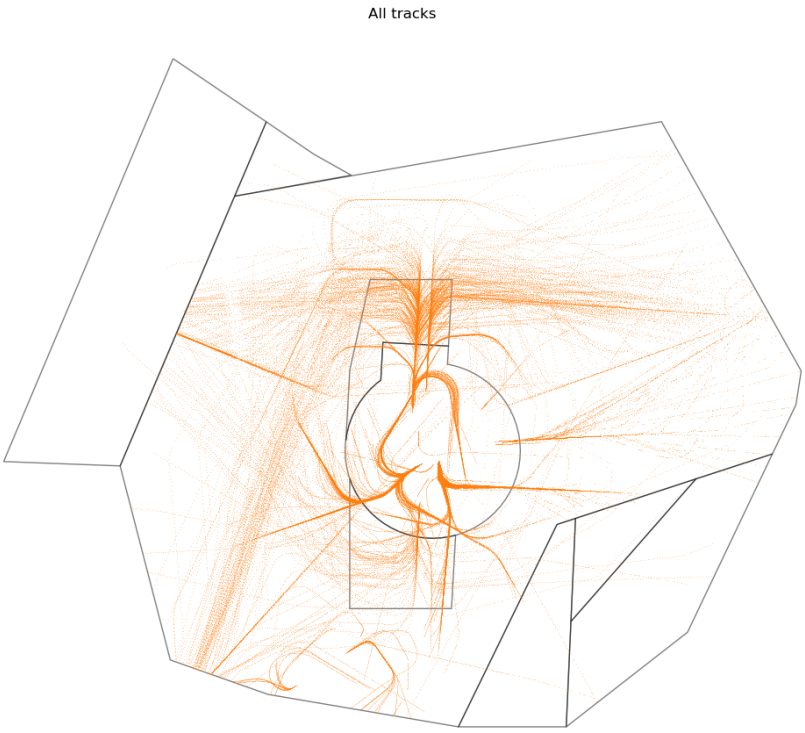


Figure D.3: All flight tracks, 02-01-2018, unclustered.

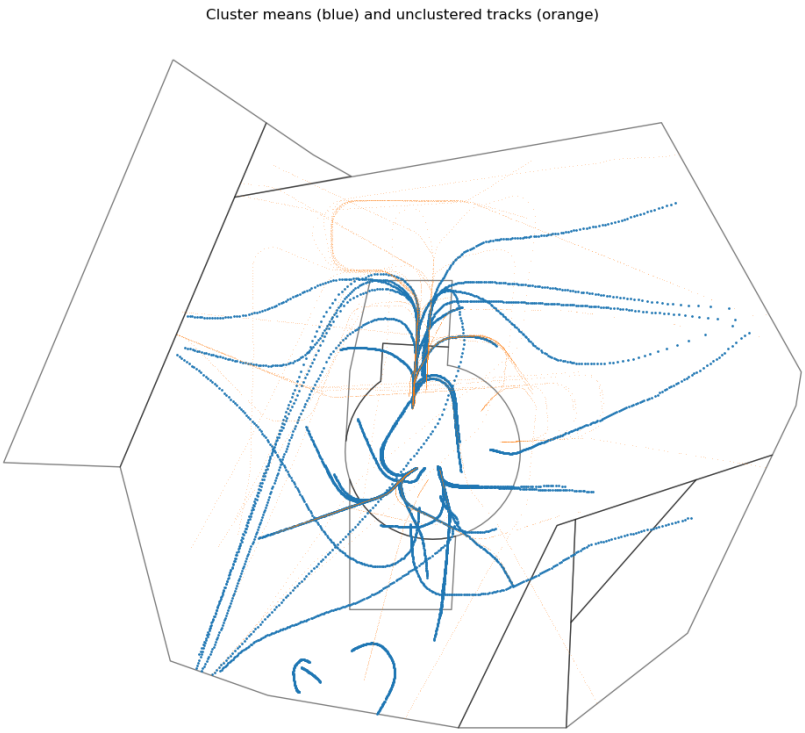


Figure D.4: Cluster means and unclustered tracks, 02-01-2018.

D.1.3. 04-01-2018.

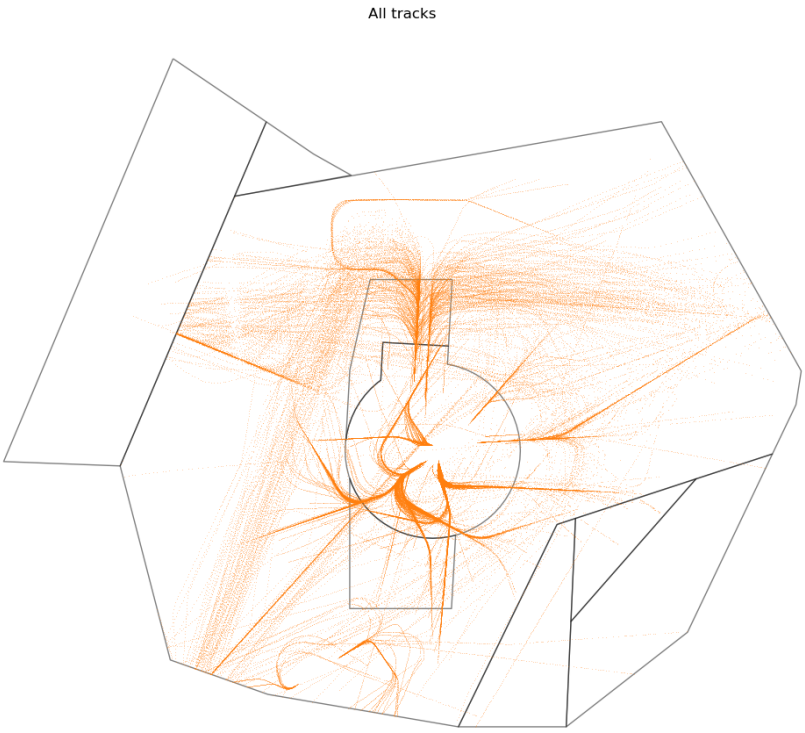


Figure D.5: All flight tracks, 04-01-2018, unclustered.

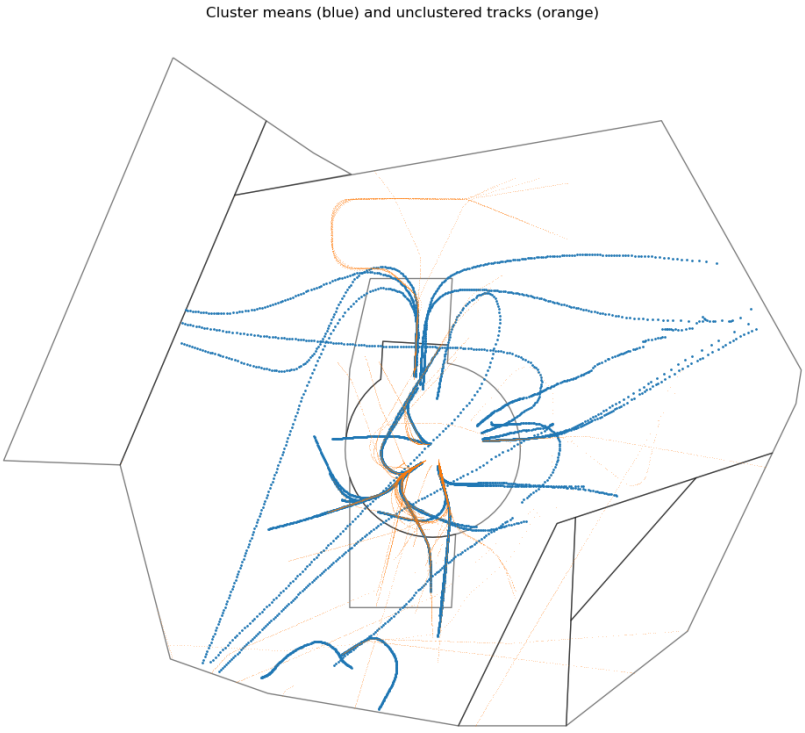


Figure D.6: Cluster means and unclustered tracks, 04-01-2018.

D.1.4. 05-01-2018.

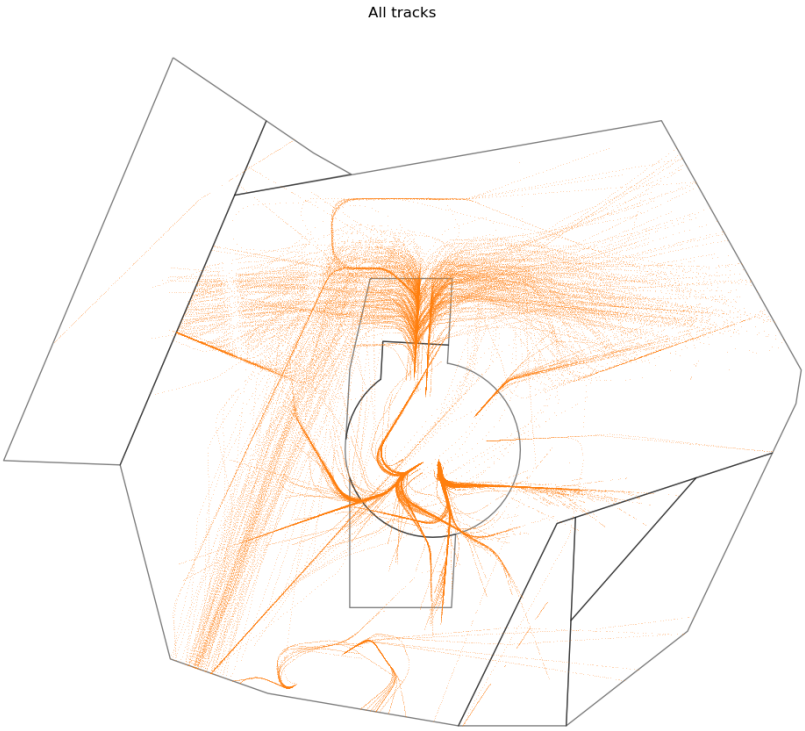


Figure D.7: All flight tracks, 05-01-2018, unclustered.

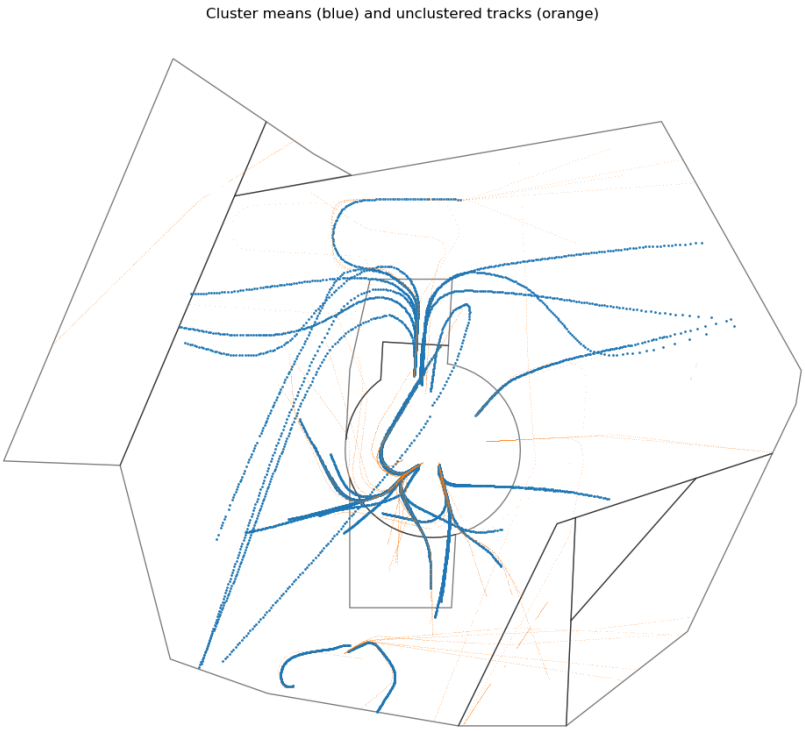


Figure D.8: Cluster means and unclustered tracks, 05-01-2018.

D.2. Days combined

Jan 1-2, 4-5, split evenly over 4 new scenario’s.

D.2.1. Scenario 1

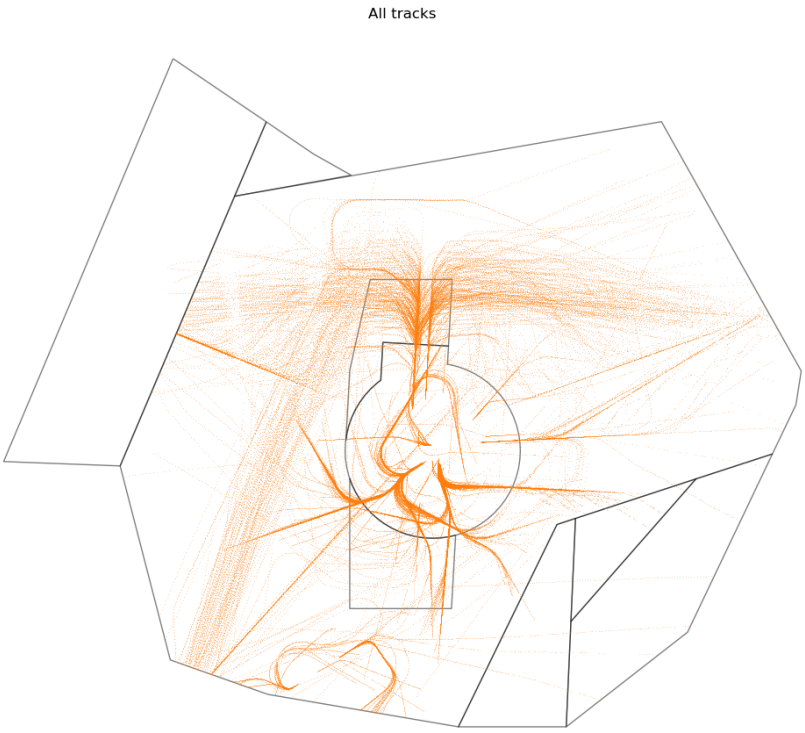


Figure D.9: All flight tracks, combined days, scenario 1, unclustered.

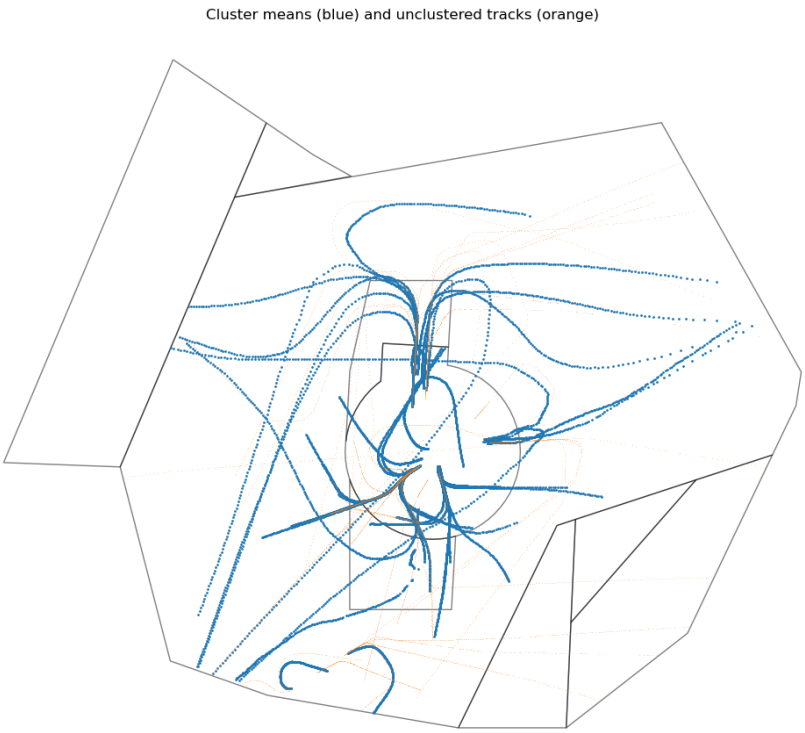


Figure D.10: Cluster means and unclustered tracks, combined days, scenario 1.

D.2.2. Scenario 2

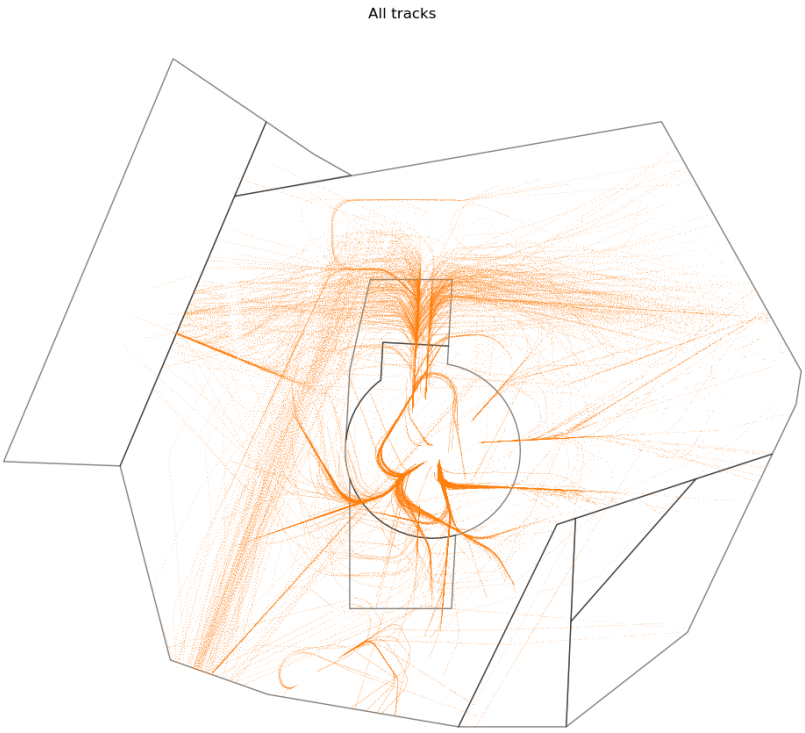


Figure D.11: All flight tracks, combined days, scenario 2, unclustered.

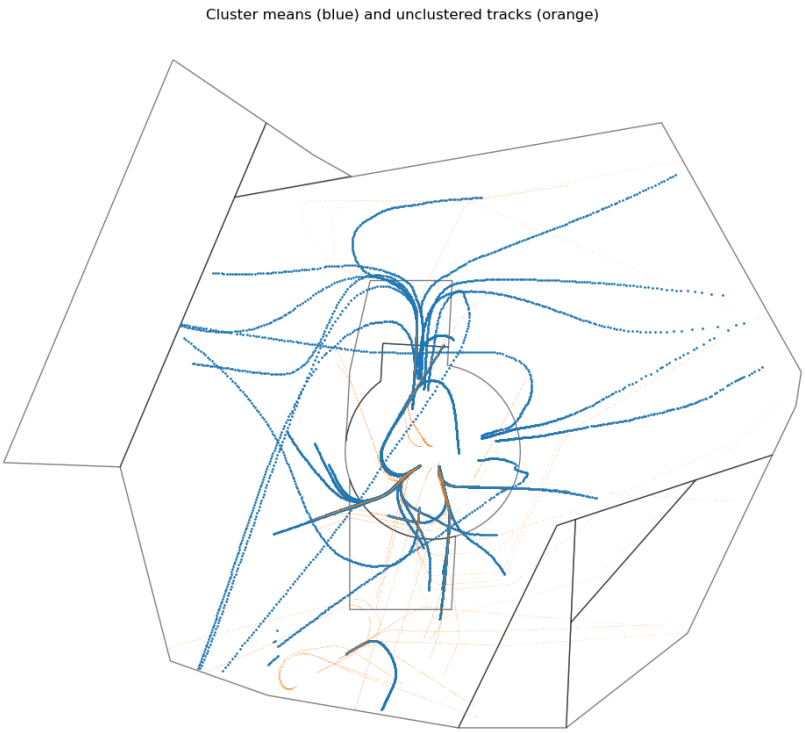


Figure D.12: Cluster means and unclustered tracks, combined days, scenario 2.

D.2.3. Scenario 3

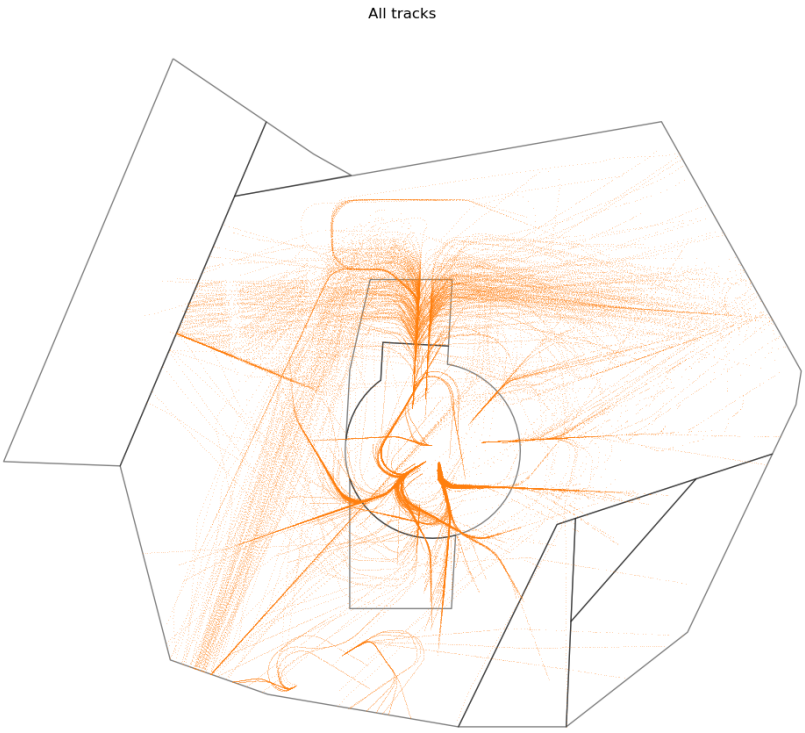


Figure D.13: All flight tracks, combined days, scenario 3, unclustered.

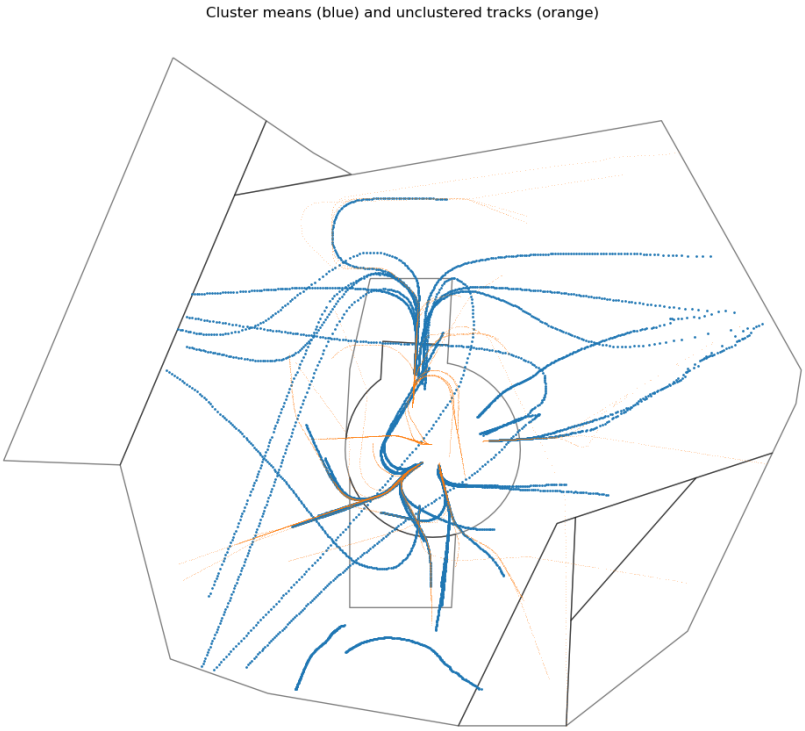


Figure D.14: Cluster means and unclustered tracks, combined days, scenario 3.

D.2.4. Scenario 4

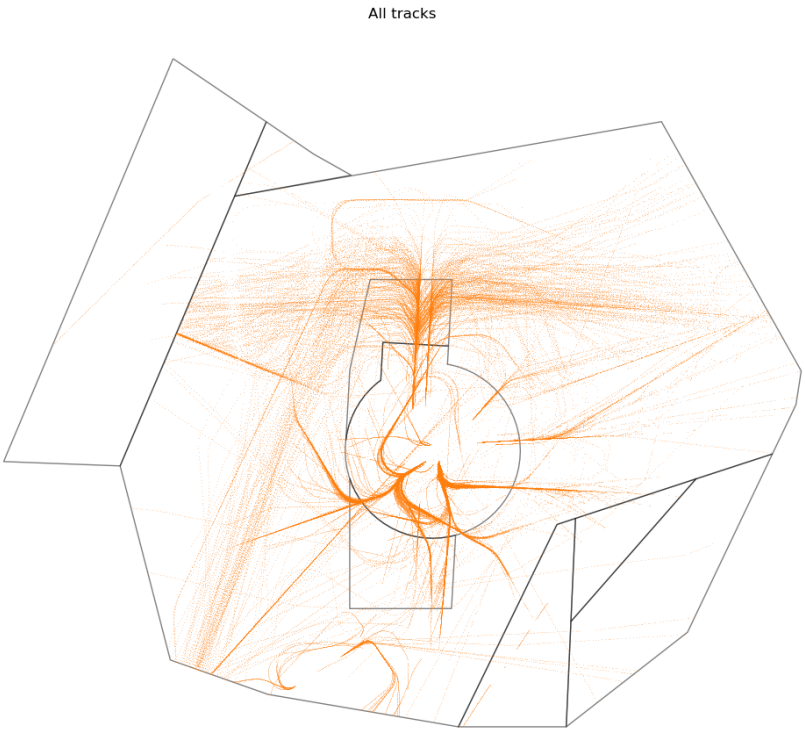


Figure D.15: All flight tracks, combined days, scenario 4, unclustered.

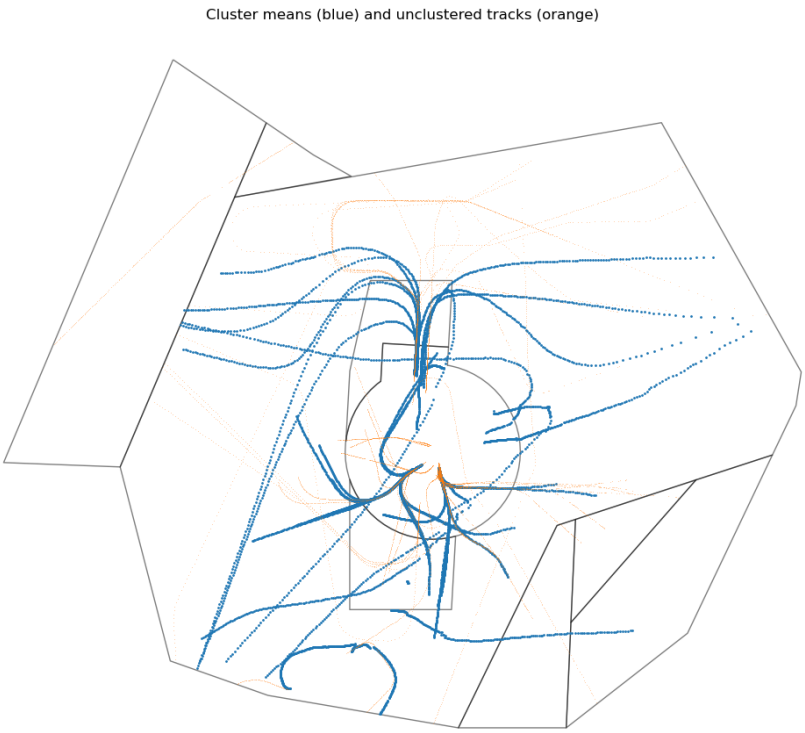
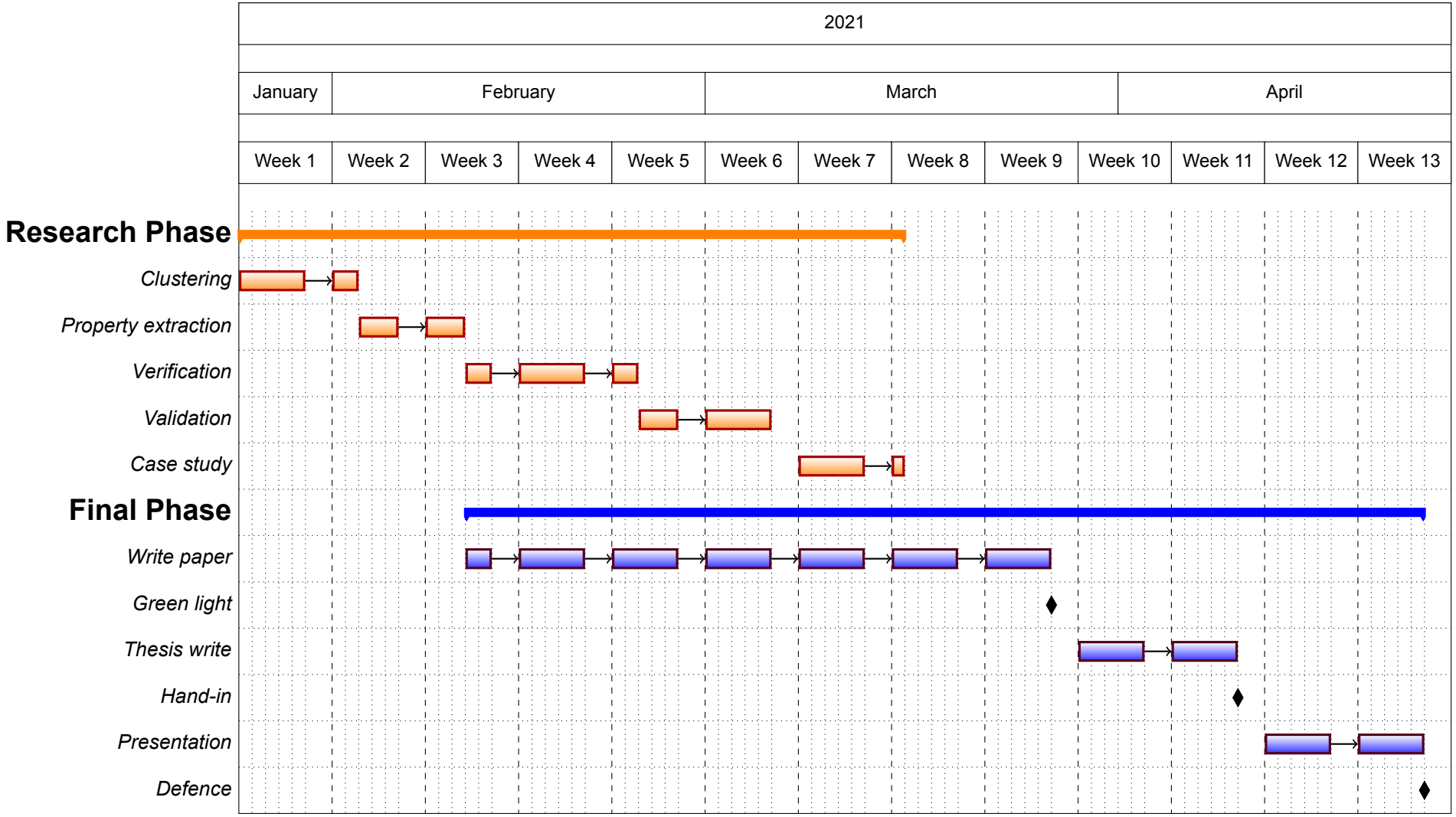


Figure D.16: Cluster means and unclustered tracks, combined days, scenario 4.

E

Planning Gantt chart

(See next page)


2021
January
February
March
April
Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
Week 8
Week 9
Week 10
Week 11
Week 12
Week 13
Research Phase
Clustering
Property extraction
Verification
Validation
Case study
Final Phase
Write paper
Green light
Thesis write
Hand-in
Presentation
Defence