UiO : Department of Mathematics
University of Oslo

Modeling Aircraft Movements Using Stochastic Hybrid Systems

STK4000 - Risk and Reliability Analysis (Exam)

Jonas Semprini Næss

April 29, 2024

Table of contents I

Overview and Introduction

- What are we studying?

- Air Traffic Control (ATC)

- How are we approaching the problem?
  - Ordinary Differential Equations (ODE)
  - Stochastic Environments
  - Counting Processes

Table of contents I

Table of contents II

Conceptual Model

- $\mathcal{A}$ Airspace of interest
- $\{N(t)\}$ Aircraft arrival process
- $\left(T_{a,i}, T_{d,i}\right)$ Arrival and departure times of *i*th aircraft
- $T_{p,i} = T_{d,i} - T_{a,i}$ Processing time of ith aircraft
- $\left(\boldsymbol{X}_{a,i}, \boldsymbol{X}_{d,i}\right)$ Arrival and departure points of *i*th aircraft

Derived processes

- We denote the number of aircraft that arrived at time t by

$$N(t) = \sum_{i=0}^{\infty} I(T_{a,i} \leq t).$$

- Moreover, is it reasonable to address the number of processed aircraft at t. Which can be described as follows

$$M(t) = \sum_{i=0}^{\infty} I(T_{d,i} \leq t).$$

- We can view this model as a queueing system where aircraft are the clients arriving at times $T_{a,i}$ and the processing times are given by $T_{p,i} = T_{d,i} - T_{a,i}$. The queueing process, denoted $\{Q(t)\}$, represents the queue length over time, defined as

$$Q(t) = \sum_{i=1}^{\infty} I(T_{a,i} \leq t < T_{d,i})$$

Where by observation we see that $Q(t) = N(t) - M(t)$

Aircraft Position Model

- The position of the ith aircraft at time t is represented as $x_i(t)$, where $i = 1, 2, \ldots$. Therefore, the trajectories of the ith aircraft are described by:

$$\left\{ \boldsymbol{x}_i(t) : T_{a,i} \leq t \leq T_{d,i} \right\}$$

- With boundary conditions:

$$\boldsymbol{x}_i \left(T_{a,i} \right) = \boldsymbol{X}_{a,i}$$
$$\boldsymbol{x}_i \left(T_{d,i} \right) = \boldsymbol{X}_{d,i}$$

- Position expressed in terms of velocity:

$$- \boldsymbol{x}_i(t) = \boldsymbol{X}_{a,i} + \int_{T_{a,i}}^{t} \dot{\boldsymbol{x}}_i(u) du$$

- If we additionally assume constant velocity, and speed s_i, then we can express $\dot{\boldsymbol{x}}_i$ as:

$$\dot{\boldsymbol{x}}_i(t) = \frac{\boldsymbol{X}_{d,i} - \boldsymbol{X}_{a,i}}{\left\| \boldsymbol{X}_{d,i} - \boldsymbol{X}_{a,i} \right\|} s_i$$

Risk Measures

- We define a risk event as the proximity of two or more aircraft in the air, determined by a critical distance C. At each time t, $J(t)$ represents the index set of aircraft present in $\mathcal{A}$

- The minimum distance between aircrafts at time t :

$$D(t) = \min_{i,j \in J(t), i \neq j} \left\{ \left\| \boldsymbol{x}_i(t) - \boldsymbol{x}_j(t) \right\| \right\}$$

- We determine the Critical distance (C) (2.5 nautical miles [4,630 meters])

- The system is considered to be in a risky state (or at risk) at time t if: -

$$D(t) < C$$

Risk Measures (2)

- The limiting fraction of time where the system is in a state of risk:

$$\bar{R} = \lim_{t\to\infty} \frac{1}{t} \int_0^t \mathrm{I}(D(u) < C)du$$

- The asymptotic average minimum distance between two aircrafts in $\mathcal{A}$:

$$\bar{D} = \lim_{t\to\infty} \frac{1}{t} \int_0^t D(u)du$$

- The asymptotic average throughput:

$$\bar{M} = \lim_{t\to\infty} \frac{M(t)}{t}$$

- The asymptotic average processing time:

$$\bar{T}_p = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} T_{p,i}$$

Table of contents I

Software Implementation

1. Handles both discrete and continuous events:
   - Generates aircraft (discrete events)
   - Updates aircraft positions (continuous events)
2. Permits flexible update intervals for each aircraft:
   - Utilizes short intervals during periods prone to risk
   - Employs long intervals during safe periods

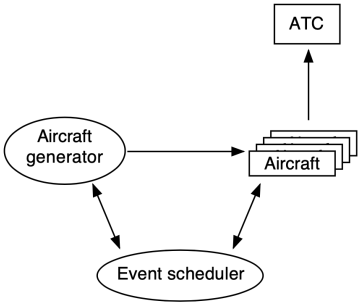


Table of contents I

Table of contents II

Setup and Assumptions

- We chose a counting process, $\{N(t)\}$, with i.i.d waiting times between arrivals sampled from a censored exponential distribution.

- These waiting times, denoted by $W_1, W_2, \ldots$, are determined as:

$$W_i = \max(\kappa, U_i), \quad i = 1, 2, \ldots$$

- Here, $U_1, U_2, \ldots$ are independent and identically exponentially distributed variables with mean $\mu = 90$ seconds.

- We also introduced the constant κ as a control parameter, controlling the flow of traffic into $\mathcal{A}$. In our simulations, κ varied from 10 to 50 seconds.

Data Handling and Processing

- All flights through $\mathcal{A}$ are assumed to be either northbound or eastbound.
- The probability of a flight being northbound is the same as the probability of it being eastbound, which is $\frac{1}{2}$.
- For northbound flights, arrival points are randomly selected within the interval I_S along the southern border of $\mathcal{A}$, and departure points within the interval I_N along the northern border.
- For eastbound flights, arrival points are randomly selected within the interval I_W along the western border of $\mathcal{A}$, and departure points within the interval I_E along the eastern border.

Simulation results (Round 1)

κ	10	20	30	40	50
$\bar{R}$	0.250	0.239	0.227	0.066	0.020
$\bar{D}$	12,108	12,288	12,548	12,840	13,196
$\bar{M}$	0.660	0.650	0.632	0.611	0.588
$\bar{T}_p$	144	144	144	144	144

Figure: Estimated criticality and performance measures as a function of the control parameter κ.

Avoiding Risk Events

Trajectory Algorithm

For each point of time t and for each $j \in J(t)$ do the following:
STEP 1. Calculate the "ideal" velocity for the j th aircraft as:

$$\dot{\boldsymbol{x}}_j(t) = \frac{\boldsymbol{X}_{d,j} - \boldsymbol{x}_j(t)}{\left\|\boldsymbol{X}_{d,j} - \boldsymbol{x}_j(t)\right\|} s_j.$$

Avoiding Risk Events

Trajectory Algorithm

For each point of time t and for each $j \in J(t)$ do the following:
STEP 1. Calculate the "ideal" velocity for the j th aircraft as:

$$\dot{\boldsymbol{x}}_j(t) = \frac{\boldsymbol{X}_{d,j} - \boldsymbol{x}_j(t)}{\left\|\boldsymbol{X}_{d,j} - \boldsymbol{x}_j(t)\right\|} s_j.$$

STEP 2. Find the nearest aircraft and adjust the trajectory if needed to avoid a risk event.

Avoiding Risk Events

Trajectory Algorithm

For each point of time t and for each $j \in J(t)$ do the following:
STEP 1. Calculate the "ideal" velocity for the j th aircraft as:

$$\dot{\boldsymbol{x}}_j(t) = \frac{\boldsymbol{X}_{d,j} - \boldsymbol{x}_j(t)}{\left\|\boldsymbol{X}_{d,j} - \boldsymbol{x}_j(t)\right\|} s_j.$$

STEP 2. Find the nearest aircraft and adjust the trajectory if needed to avoid a risk event.

STEP 3a. If no action is needed, use the ideal velocity in the interval $[t, t + dt)$, and update the position: $\boldsymbol{x}_j(t + dt) = \boldsymbol{x}_j(t) + \dot{\boldsymbol{x}}_j dt$

Avoiding Risk Events

Trajectory Algorithm

For each point of time t and for each $j \in J(t)$ do the following:
STEP 1. Calculate the "ideal" velocity for the j th aircraft as:

$$\dot{\boldsymbol{x}}_j(t) = \frac{\boldsymbol{X}_{d,j} - \boldsymbol{x}_j(t)}{\left\| \boldsymbol{X}_{d,j} - \boldsymbol{x}_j(t) \right\|} s_j.$$

STEP 2. Find the nearest aircraft and adjust the trajectory if needed to avoid a risk event.

STEP 3a. If no action is needed, use the ideal velocity in the interval $[t, t + dt)$, and update the position: $\boldsymbol{x}_j(t + dt) = \boldsymbol{x}_j(t) + \dot{\boldsymbol{x}}_j dt$

STEP 3b.If an action is needed, make a “turn”. That is, replace the ideal velocity by:

$$\dot{\boldsymbol{x}}_j'(t) = \Lambda \dot{\boldsymbol{x}}_j(t)$$

in the interval $[t, t + dt)$ where Λ is the 3×3 horizontal rotation matrix.

Avoiding Risk Events

Trajectory Algorithm cont.

- To further specify the algorithm:
  1. We need a criterion for when an action is needed.
  2. We must determine matrix Λ based on relative positions and velocities of two aircraft.
- To do this:
  - Consider jth aircraft a_j at time t_0.
  - Assume a_k is the closest aircraft to a_j.
  - Focus on relative positions and velocities of a_j as seen from a_k, denoted $\boldsymbol{x}_{j,k}(t)$ and $\dot{\boldsymbol{x}}_{j,k}(t)$.

Where $\boldsymbol{x}_{j,k}(t)$ and $\dot{\boldsymbol{x}}_{j,k}(t)$ are expressed as follows:

$$\boldsymbol{x}_{j,k}(t) = \boldsymbol{x}_j(t) - \boldsymbol{x}_k(t)$$

$$\dot{\boldsymbol{x}}_{j,k}(t) = \dot{\boldsymbol{x}}_j(t) - \dot{\boldsymbol{x}}_k(t)$$

Avoiding Risk Events

Trajectory Algorithm cont.

- Distance between a_k and a_j at time t_0: $\left\|\boldsymbol{x}_{j,k}(t_0)\right\|$.
- If large, no trajectory modification for a_j is needed.
- Alert distance C_A introduced to manage this, set $> C$ to avoid risk events.
- In simulations, C_A set to twice C (5 nautical miles or $9,260$ meters).

Avoiding Risk Events

Trajectory Algorithm cont.

Case: $\left\|\boldsymbol{x}_{j,k}(t_0)\right\| \leq C_A$

- Determine if aircrafts get closer or further apart over time.
- Consider distance $\Delta(t) = \left\|\boldsymbol{x}_{j,k}(t)\right\|$.
- Assuming no action:

$$\boldsymbol{x}_{j,k}(t) = \boldsymbol{x}_{j,k}(t_0) + \dot{\boldsymbol{x}}_{j,k}(t_0)(t - t_0)$$

- Find time t_1 when $\Delta(t)$ is minimized:

$$t_1 = t_0 - \frac{\boldsymbol{x}_{j,k}(t_0)^T \dot{\boldsymbol{x}}_{j,k}(t_0)}{\dot{\boldsymbol{x}}_{j,k}(t_0)^T \dot{\boldsymbol{x}}_{j,k}(t_0)}$$

Avoiding Risk Events

Trajectory Algorithm cont.

- $t_1 > t_0$ if and only if: $\boldsymbol{x}_{j,k}(t_0)^T \dot{\boldsymbol{x}}_{j,k}(t_0) < 0$.
- By inserting the expression for t_1 into the equation for $\boldsymbol{x}_{j,k}(t)$, we find the position where a_j is closest to a_k:

$$\boldsymbol{x}_{j,k}(t_1) = \boldsymbol{x}_{j,k}(t_0) - \dot{\boldsymbol{x}}_{j,k}(t_0) \frac{\boldsymbol{x}_{j,k}(t_0)^T \dot{\boldsymbol{x}}_{j,k}(t_0)}{\dot{\boldsymbol{x}}_{j,k}(t_0)^T \dot{\boldsymbol{x}}_{j,k}(t_0)}$$

- If $t_1 > t_0$ and $\Delta(t_1) \leq C$, action is needed to avoid a risk event.
- Our approach involves making a "turn" by rotating the velocity vectors by λ radians.
- To determine λ, we introduce the normal vector to the relative position vector, $\boldsymbol{y}_{j,k}(t_0)$, obtained by rotating $\boldsymbol{x}_{j,k}(t_0)$ counterclockwise by $\pi/2$ radians.

Avoiding Risk Events

Trajectory Algorithm cont.

Case 1: $\dot{\boldsymbol{x}}_{j,k}(t_0)^T \dot{\boldsymbol{y}}_{j,k}(t_0) > 0$
Case 2: $\dot{\boldsymbol{x}}_{j,k}(t_0)^T \dot{\boldsymbol{y}}_{j,k}(t_0) < 0$
Case 3: $\dot{\boldsymbol{x}}_{j,k}(t_0)^T \dot{\boldsymbol{y}}_{j,k}(t_0) = 0$

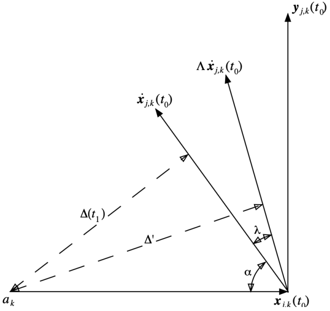
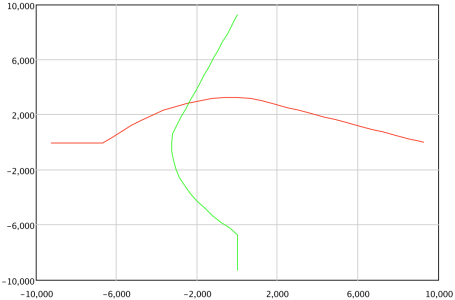
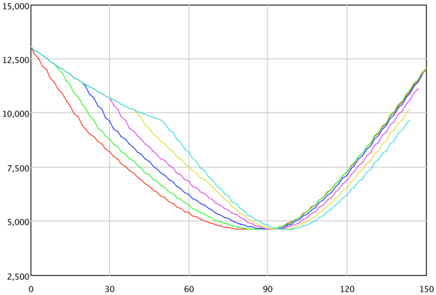


Figure: Rotating the relative velocity

Avoiding Risk Events

Simulation results (Round 2)





- **Plot left**: Trajectories of two flights
- **Plot right**: Distance (meters) between flights as a function of time

Avoiding Risk Events

Simulation results (Round 2)

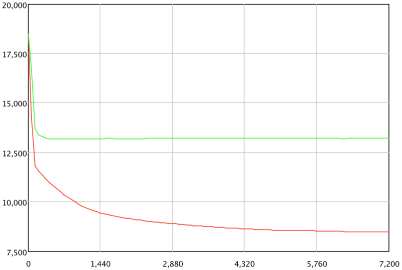
κ	10	20	30	40	50
$\bar{R}$	0.480	0.477	0.472	0.006	0.000
$\bar{D}$	8,558	8,672	8,813	12,881	13,223
$\bar{M}$	0.652	0.641	0.624	0.610	0.586
$\bar{T}_p$	219	221	224	145	144

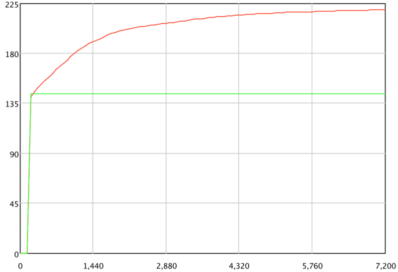
Avoiding Risk Events

Simulation results (Round 2)

κ	10	20	30	40	50
$\bar{R}$	0.480	0.477	0.472	0.006	0.000
$\bar{D}$	8,558	8,672	8,813	12,881	13,223
$\bar{M}$	0.652	0.641	0.624	0.610	0.586
$\bar{T}_p$	219	221	224	145	144

κ	10	20	30	40	50
$\bar{R}$	0.250	0.239	0.227	0.066	0.020
$\bar{D}$	12,108	12,288	12,548	12,840	13,196
$\bar{M}$	0.660	0.650	0.632	0.611	0.588
$\bar{T}_p$	144	144	144	144	144





- **Plot left**: Average minimum flight distances (meters) as functions of time for $\kappa = 10$ (lower curve) and $\kappa = 50$ (upper curve)
- **Plot right**: Average processing times as functions of time for $\kappa = 10$ (lower curve) and $\kappa = 50$ (upper curve)

Table of contents I

Summary

1. We have demonstrated how to create hybrid simulation models for aircraft paths in a random environment.
2. Examined two related concerns with the model:
   - Organizing the arrival process
   - Preventing situations/events of risk by adjusting paths dynamically
3. It's essential to study both issues together within a unified model.
4. Future work:
   - Enhancing and refining the models and algorithms
   - Modeling smoother aircraft movements (in three dimensions)
   - Incorporating other random factors such as weather

References I

- A.M. Bayen, R.L. Raffard, and C.J. Tomlin.
  Adjoint-based control of a new eulerian network model of air traffic flow.
  IEEE Transactions on Control Systems Technology, 14(5):804–818, 2006.

- Mikhail Bernadsky, Raman Sharykin, and Rajeev Alur.
  Structured modeling of concurrent stochastic hybrid systems.
  In Yassine Lakhnech and Sergio Yovine, editors, *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pages 309–324, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

- William Glover and John Lygeros.
  A stochastic hybrid model for air traffic control simulation.
  volume 2993, pages 372–386, 03 2004.

References II

Tak-Kuen Koo, Yi Ma, George Pappas, and Claire Tomlin.
Smartatms: A simulator for air traffic management systems, 01 1997.

UiO : Department of Mathematics
University of Oslo

Jonas Semprini Næss

Modeling Aircraft Movements Using Stochastic Hybrid Systems

STK4000 - Risk and Reliability Analysis (Exam)