Thirteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2019)

Operational Concept of Traffic Pattern Classifier for Optimal Ground Holding

Adriana Andreeva-Mori, Naoki Matayoshi
Next Generation Aeronautical Innovation Hub Center, Aeronautical Technology Directorate
Japan Aerospace Exploration Agency (JAXA)
Tokyo, Japan
andreevamori.adriana, matayoshi.naoki@jaxa.jp

Abstract— **A dual-component ground holding (GH) algorithm based on real-time air traffic classification and offline ground holding program parameter optimization is proposed. Numerical simulations are developed to quantitatively evaluate this new concept. GH program performance is evaluated based on airborne delay, ground delay, and lost throughput costs. Preliminary results show that the developed machine-learning-based traffic pattern classifier can propose ground holding control parameters which would result in savings within mean absolute percentage error of 17.96% of the potential optimal ones.**

Keywords- ground holding, traffic pattern, machine learning, new traffic management initiatives, decision-support tool

I. INTRODUCTION

During nominal operations, flights are scheduled so that demand does not exceed capacity at neither the departure nor the arrival airport. Weather uncertainty and unexpected passenger or aircraft troubles, however, often lead to uncoordinated demand and cause congestions in the arrival flow. In order to manage such congestions at the arrival airport, some flights can be held on the ground at the departure airport. This traffic management initiative, called ground holding (GH) or ground delay, is meant to reduce airborne holding, thus leading to fuel savings, lower air traffic controllers' workload and higher safety. When GH is modeled as a completely deterministic problem, the optimal time for which a flight needs to be delayed at the origin airport can be calculated accurately. In reality, however, traffic flow includes many uncertainties, such as departure time errors and flight time errors, which makes the GH problem a probabilistic one. If the calculated GH is too short, the flight will still have to spend unnecessary time in the air, thus burning fuel and occupying airspace. On the other hand, if the calculated GH is too long, the flight will be able to land without any holding in the air, but landing capacity (throughput efficiency), will be lost. Finding the balance between those two is the key to the GH problem.

The most common approaches to the GH problem are 1) either set a constant buffer (maximum allowed value for the airborne delay) regardless of the specific traffic situation, which can then be somewhat adjusted for each flight manually by air traffic controllers, or 2) using complex dynamic models which include real-time optimization and, in most cases, require time-based management. Cox and Kochenderfer [1] showed that the dynamic approach (2) outperforms the static approach (1) in simulation, but it is often difficult to implement in real environment. Our preliminary discussions with involved parties also confirmed that complex dynamic solutions are hard to implement in a still human-centered air traffic management.

On the other hand, recent technological advances have widened the opportunities for data collection and analysis, thus triggering more intensive data-driven research in the air traffic management field. EUROCONTROL's researchers have shown that machine learning (in particular deep learning) combined with traditional prediction methods can tackle route uncertainty with success, thus increasing the trajectory prediction accuracy and overall ATM system performance [2]. Predicting delays is of key importance for air traffic flow management, so naturally data-driven research in this field can also be seen [3].

Data-driven research for ground holding program optimization is most often based on the so-called "similar days" concept [4], [5], [6], [7]. Essentially, such researches aim at identifying similar days in the past data available, so that they can provide hints on the potential performance of the ground delay program applied on the target day.

Estes, Ball and Lovell [5] proposed a black-box-like tool which can help the decision-makers with ground delay planning. The key strength of their tool lies in the fact that it does not require any explicit modeling of the airspace system, but relies on finding similar days in the data and makes predictions based on the data (ground delay program's performance metrics) available for those similar days. Their research provided strong foundation of machine learning applications to the GH problem. They also mentioned that similar approach can be used by decision-makers to help them "propose GDPs and receive estimates of the predicted performance".

Our research aims in the same direction as Estes, Ball and Lovell's work. We look for efficient ways to help air decision-makers to plan traffic management initiatives, in particular ground holding assignments. The lack of sufficient data, however, does not allow us to apply the same approach as the one proposed in [5]. Instead, we build a simulated database based on numerical simulation of ground holding program practiced in Japan, i.e. the constant buffer method. We evaluate each GH control based on airborne delay costs, ground delay costs, and lost throughput costs. The lost throughput cost corresponds to the capacity utilization metric introduced by Liu and Hansen [8]. We then develop a traffic pattern classifier which predicts the optimal ground holding control parameters based on traffic features. Our approach will allow us to simulate both past and future traffic initiatives, and thus be used in immediate as well as long-term, tactical level planning and performance analysis.

The rest of the paper is organized as follows. The operational concept and basic principles governing the development of the proposed classifier are presented in Section II. The numerical simulations developed to generate the ground holding control database are shown in Section III. The design and feasibility testing of the proposed traffic pattern classifier are presented in Section IV. We finish with some concluding remarks and discuss directions for future study in Section V.

II. Operational Concept

A. *Overall concept*

The long-term goal of this research is to develop a ground holding algorithm based on real-time air traffic pattern classification and off-line buffer optimization. The operational concept is shown in Figure 1. The input of the real-time component consists of traffic features, which might include the initial ETA queue or the corresponding traffic density, uncertainties related to departure and flight times, as well as capacity prediction. Weather information is also considered part of the traffic features, and can be handles as raw information (for example, current and predicted wind, visibility, etc.) or translated into some of the components mentioned above (for example, bad weather is likely to result in high departure and flight time uncertainties, and reduced capacity). The traffic pattern classifier feeds the traffic features into a pre-trained machine learning algorithm to determine the class to which the current traffic most likely belongs to. Each class is characterized by the potential results of the ground holding when performed for this class's traffic and the optimal ground holding decision parameters associated with it. For example, Class A might mean high effect of the ground holding program, i.e. traffic should be managed through ground holding to achieve fuel burn savings, reduction of air traffic controllers' workload and increased air traffic safety; Class B, on the other hand, might mean that the effect of ground holding cannot compensate for uncertainties in the environment and the air traffic managers are therefore not advised to enforce ground holding program. Once the traffic is classified, the optimal ground holding parameters will be extracted from a database created beforehand. This database is the output of the off-line component of the algorithm. Based on the ground holding optimal control parameters, departure times can be assigned to each flight part of the ground holding program.

B. *Basic principles*

The proposed ground holding algorithm is governed by the following basic design principles:

1) The traffic pattern classifier acts as a decision support tool to aid the selection of the most adequate ground holding parameters.

2) The ground holding database can contain both simulated and real data, and thus model past as well as new operations.

3) Introduction of the traffic pattern classifier decreases optimality, but eases real-life implementation of the algorithm.

These basic design principles are discussed in this subsection.

Decision support tools are no novelty in the air traffic management field. The proposed traffic pattern classifier's output will not be automatically reflected into the ground holding program chosen for the particular day and time. Our discussions with air traffic controllers have suggested that controller's experience and expertise play a considerable role in ground holding assignments. In our view, the traffic pattern classifier can, to a certain extent, automate this process and deliver predictable performance independent on individual experiences. However, there is a possibility that, eventually, the proposed algorithm will not outperform the top air traffic manager when it comes to mimicking past operations, in particular. The purpose of the traffic pattern classifier is to raise the average performance of the ground holding program.

One of the main strengths of the proposed algorithms is that it can be applied to new traffic management initiatives, as well. The introduction of new traffic management initiatives is often preceded by extensive simulations which estimate the benefits and compare the results with past operations. These numerical simulations include complex modeling of operations and traffic characteristics, and account for various uncertainties, for example departure time delays and enroute flight time errors. A common approach to statistically analyzing such uncertainty effects is performing Monte Carlo simulations, which are in computationally expensive. The computational time increases with the complexity and fidelity of the model. High-fidelity models take long computational time, but provide more reliable traffic simulations and ground holding program performance evaluation. By building a data base of simulated traffic and ground holding control results, the computational time necessary for each particular day/time analysis becomes irrelevant. Numerical simulations can also fill the gap of lack of experience with new traffic management initiatives, and thus help air traffic managers with their efficient implementation.

The machine learning used in the proposed ground holding algorithm does not generate any new knowledge on ground holding flight assignments itself. The ground holding parameters are chosen real time, but among the predefined set available in the database. Therefore, the introduction of traffic pattern classifier can lead to sub-optimal solutions. However, it solves the computational time issue caused by high-fidelity traffic model simulations and is an important step to real-world implementations. Such an approach might not be common in the air traffic management field, but has been widely discussed among weather, water, and climate researchers (for example, see the number of presentations related to model learning at the 99th AMS Annual Meeting [9]). Here, we test the feasibility of the concept by analyzing quantitatively results obtained through numerical simulations.

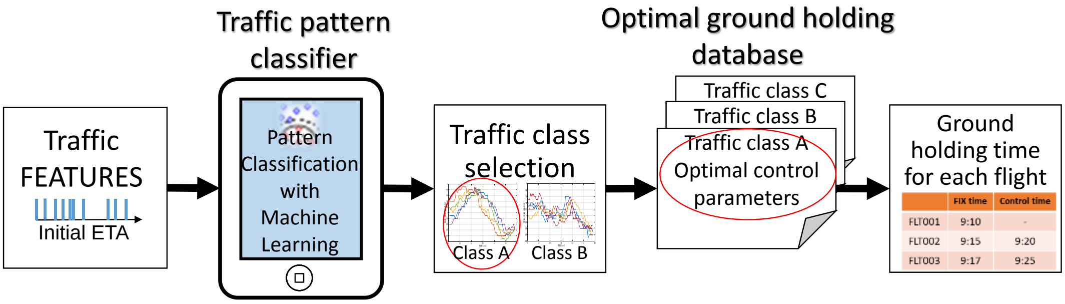


Figure 1. Operational concept of the proposed ground holding algorithm

III. Optimal Ground Holding Database Generation

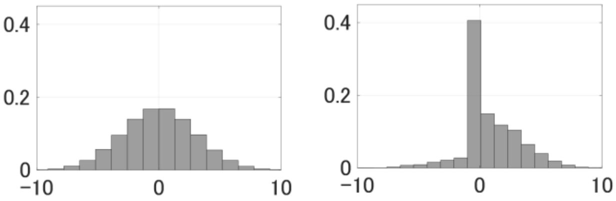
A. *Ground holding simulation for a single ETA queue: assumptions*

First, the ground holding database needs to be built. According to the operational concept, this database may consist of both real past data and simulated data. In this subsection, the assumptions governing the simulated data are described, and an example of the generated data is shown.

At this stage of the research, we consider the static case, i.e. information on all flights in the traffic is known at the start of the ground holding time calculations. Each traffic scenario is described by an Estimated Time of Arrival queue (ETA queue) for 30 aircraft. Assuming required separation of 2 min at the arrival control fix, 30 aircraft account for 1 hour traffic (2 min separation after the last aircraft assures the following ETA queue arrival time calculation can start at time zero again).

Ground delays are assigned so that the predicted airborne delay (*AirborneATCDelay*) does not exceed a certain maximum airborne delay threshold (*Buffer*). For example, assuming *Buffer* is set at 9 min, if the predicted *AirborneATCDelay* is 15 min, the ground delay *GroundDelay* will be 6 min. The role of *Buffer* is twofold. First, it is set to absorb departure time uncertainties and flight time uncertainties so that the throughput is maximum even in the case of late departures, for example. If *Buffer* is small, more flights will be subject to ground holding and there will be less airborne delays, but the risk of losing throughput increases. On the other hand, if *Buffer* is large, airborne delays will increase and the ground holding program will not be efficient. This tradeoff is a key consideration when setting the ground holding program parameters. The other aspect which needs to be taken into account when setting the *Buffer* is the maximum amount of airborne holding which can be managed safely within the airspace prior to the arrival fix. This constraint depends on the properties of the particular airspace and air traffic control practices. Here, we consider the arrival flow to Japan's busiest airport, Tokyo International Airport (Haneda Airport), and discussions with air traffic controllers have shown that a constraint of 9 min is feasible.

In our simulations, we model departure time errors (*DeptError*) of non-ground holding flights by a normal distribution with mean 0 min and standard deviation of 5 min. We assume that once a flight is assigned ground delay, it cannot depart prior to its assigned time, and therefore the *DeptError* distribution is changed as shown in Figure 2. Further sensitivity analysis of departure time errors can be found in previous work by the authors [10].



a) *DeptError* [min] when ground holding program is not enforced

b) *DeptError* [min] when ground holding program is enforced. *DeptError* of flights for which *GroundDelay*>0 is non negative.

Figure 2. *DeptError* distribution assumptions

The flight time error *EnrouteError* is modeled by a normal distribution (mean zero, standard deviation 1 min) for all flights. Under the above assumptions, the time of arrival (ATA) of each flight can be determined as follows:

$$\begin{aligned}\mathrm{ATA} = \mathrm{ETA} + \mathrm{GroundDelay} + \mathrm{DeptError} \\ + \mathrm{EnrouteError} \\ + \mathrm{AirborneATCDelay}\end{aligned} \qquad (1)$$

Note that the required separation is included in *AirborneATCDelay*. The calculation flow of ATA is shown in Figure 3. Further discussions regarding the sequencing timing are ongoing with air traffic controllers.

ETA queues are randomly generated so as to meet average demand/capacity ratios set in advance. We introduce capacity coefficient *Cap* defined as average capacity/demand, i.e. *Cap= 0.8* means that the average demand exceeds the capacity by 25%. Therefore, *ETA* of flight *i* is determined as:

$$ETA_i = (SepReq * NumFlights * Cap) * randETA \qquad (2)$$

where *randETA* is a random number (mean=0, std=1).

The evaluation of the ground holding for each ETA queue was done considering the following three costs: ground delay cost, airborne delay cost, and what we refer to as "lost capacity cost", or the decrease of the throughput at the control fix.

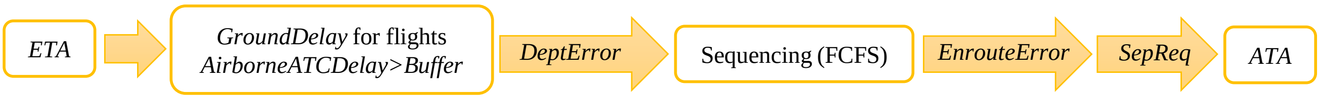


Figure 3. ATA calculation flow

The ground delay cost (c_g) and the airborne delay cost (c_a) are defined per flight per minute, whereas the lost capacity cost (c_c) is defined for the whole ETA queue per minute. Research with relative values of c_a /c_g varying between 1 and 10 can be found in [1]. Here, we determine the values for c_g and c_g based on the report prepared by Westminster University [11]. For B738, the cost of 5 min ground delay is 80 EUR, and the cost of 5 min airborne delay is 210 EUR. For B736 these values are 130 EUR and 370 EUR, and for B744- 190EUR and 540EUR, respectively. For JFY2016, the relative contribution of these aircraft classes to the entire traffic is 5:2.7:2.3 [12], so on average, the cost of 1 min ground delay is 24 EUR and the cost of 1 min airborne delay is 66 EUR. To evaluate lost capacity, we consider the average profit per passenger (approximately $8 [13]). The required separation is assumed to be 2 min, i.e. 2 min of capacity lost means another flight might have been accommodated at the airport (terminal area). Assuming 200 passengers per flight, a profit of $1600 per flight is unrealized. Conversion to EUR (2018/08/20, 1USD=0.8743EUR）gives the value of c_c=700 EUR/min. All cost assumption values are summarized in Table 1. These particular values are implemented in the simulation for quantitative analysis, and can be changed to suit new/more accurate data when such become available.

TABLE 1. COST ASSUMPTIONS

Notation	Explanation	Value [EUR]
c_g	Airborne delay cost per flight, per minute	24
c_a	Ground delay cost, per flight, per minute	66
c_c	Lost capacity cost, per minute	700

The numerical simulations developed in this research consider departure time errors and flight time errors. Therefore, even for the same ETA queue, the final ATA will vary. To exclude this variations from the results, the evaluation of each ground holding is done in respect to a nominal case, where no flights are subject to ground holding, i.e. the entire delay is absorbed in the air. Considering all of the assumptions above, the cost function is defined as follows:

$$\text{Minimize}\quad TotalDelayCost = \sum_{i=1}^{N=30}\left(GroundDelay_i^{GHP} - GroundDelay_i^{Nominal}\right) * c_g + \sum_{i=1}^{N=30}\left(AirborneATCDelay_i^{GHP} - AirborneATCDelay_i^{Nominal}\right) * c_a + \left(\max\left(ATA_i^{GHP}\right) - \max\left(ATA_i^{Nominal}\right)\right) * c_c \quad (3)$$

In the nominal case, $GroundDelay_i^{Nominal} = 0$.

B. Ground holding simulation for a single ETA queue: sample results

This subsection presents the simulation results for a sample ETA under the assumptions discussed above. Since uncertainties in departure time and flight time are considered, Monte Carlo simulation is used to evaluate the performance of the ground holding control. The value of *Buffer* is varied between 1 min and 15 min over 1 min interval. For each *Buffer* value, Monte Carlo simulations are run 1000 times. Figure 4 shows the median value convergence for a sample ETA queue.

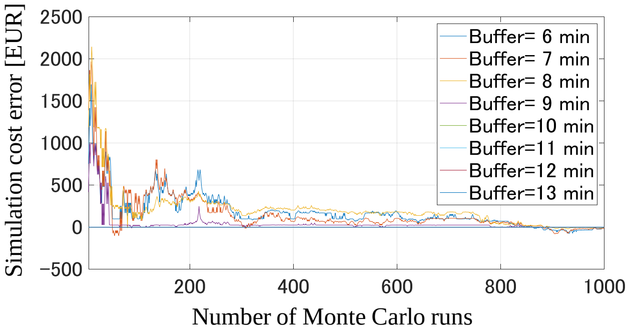


Figure 4. Convergence of the cost function

First, consider the ETA queue shown in Figure 7. Altogether, there are 30 flights with ETAs unevenly distributed between 0 and 60. The ground holding effect for ETA1 and Buffer varying between 1 min and 15 min is shown in Figure 6. The horizontal axis shows the Buffer value in minutes, while the vertical axis shows the cost compared to the nominal case, and lower values mean decreased cost, i.e. negative values, of mean savings. For each value, simulation results are shown by a box plot. The median value is shown in red, and the bottom and top edges of the box indicate the 25th and 75th percentiles. The whiskers show all points but the outliers. From these results it is obvious that the total delay cost determined according to Equation (3) varies with *Buffer* values. The median total cost is minimum for *Buffer = 6* min (median cost savings are 7800 EUR). With the increase of *Buffer* value, however, the effect of departure time and flight time uncertainties decreases, so from the operational perspective the optimal *Buffer* choice is not straightforward. Small *Buffer* leads to more ground holding, so the ground delay costs is maximum for *Buffer=1*. On the other hand, airborne delay costs increases with *Buffer* value.

Next, consider another sample ETA queue (Figure 7). The traffic is highly concentrated at the beginning, and sparse after that. The general trend for total costs, ground delay cost, airborne cost and lost capacity cost are similar to those observed for ETA1. However, most savings are achieved for *Buffer=11 min* (median savings 2794 EUR), which is considerably less than the savings for ETA1. For *Buffer* values less than 7, ground holding control will likely induce extra costs, not savings (the median exceeds zero).

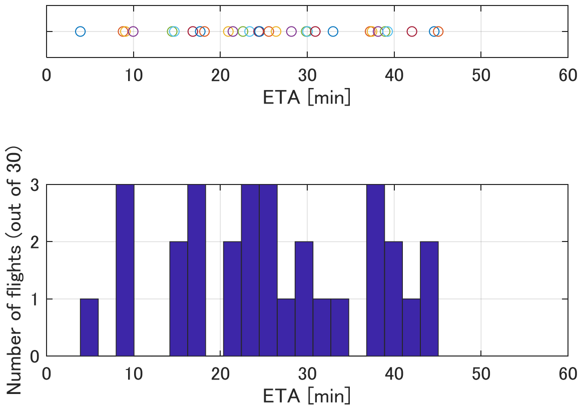


Figure 5. ETA1

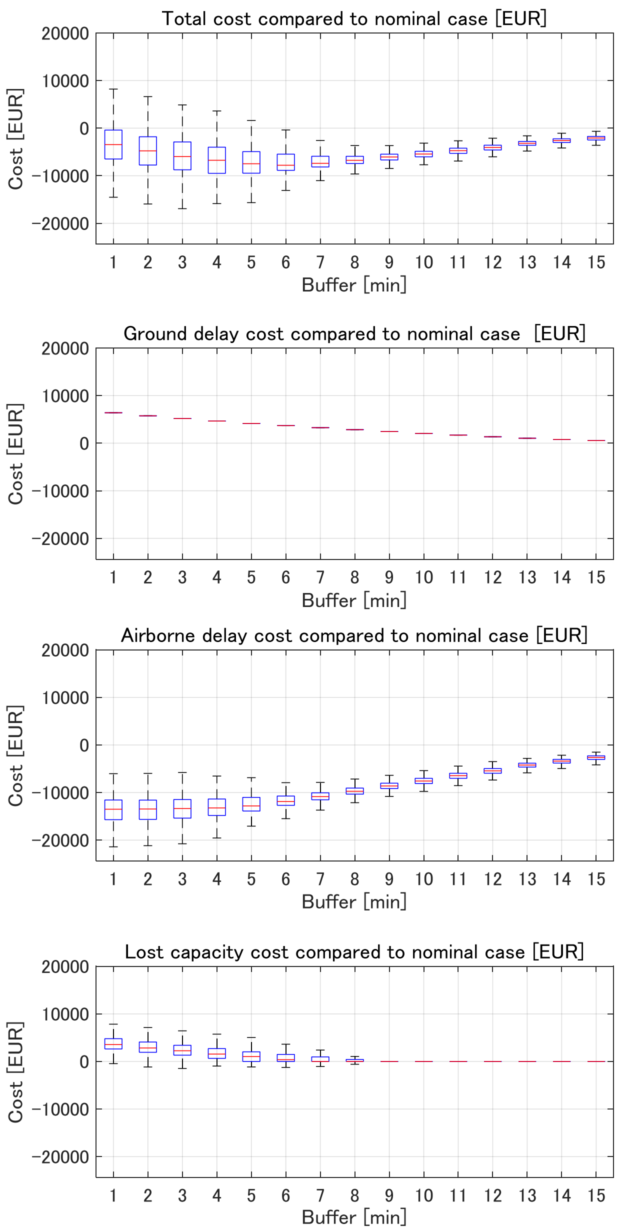


Figure 6. Simulation results for ETA1

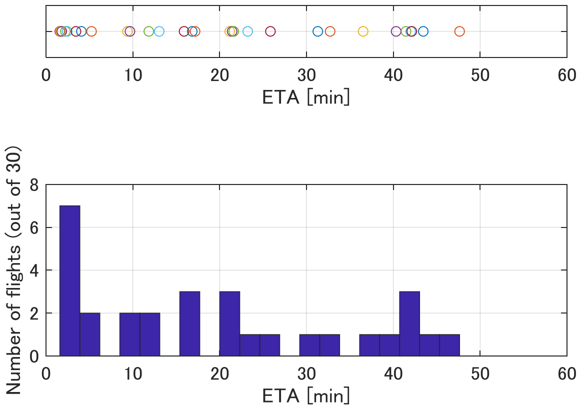


Figure 7. ETA2

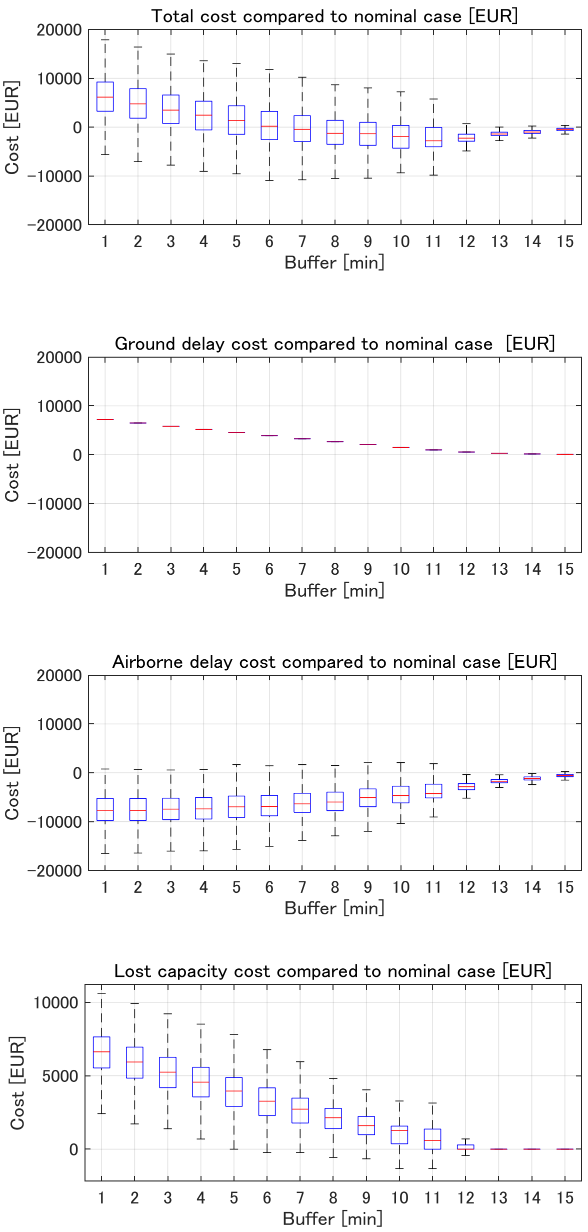


Figure 8. Simulation results for ETA2

These two sample ETA queues illustrate two important control results:

1) Savings due to ground holding are dependent on the ETA queue. Choosing the optimum *Buffer* value might not be sufficient to produce sufficient savings, i.e. the ground holding effect for some ETA queues is limited.

2) The optimum *Buffer* value which minimizes the cost function depends greatly on the individual ETA queue.

Therefore, if the air traffic manager can correctly classify the ETA queue pattern, i.e. the traffic pattern, they will be able to set the optimum ground holding program parameters (here, choose the *Buffer* value) and decide whether to actively pursue the ground holding program application to this particular traffic in view of the potential savings.

C. *ETA queue and ground holding results: database generation*

Following the methodology described above, a database for 1000 different ETA queues is generated. To account for departure time and flight time uncertainties in each ETA queue, 1000 run Monte Carlo simulations are done. As a results, the generated database has the following information for each ground holding control: ETA queue (ETA for all 30 flights in the queue), median value of the cost function for each *Buffer* between 1 and 15 min, the optimum *Buffer* which minimizes the median value of the cost function, traffic parameters such as separation required at the control fix, uncertainties distribution parameters of the departure and flight times.

IV. TRAFFIC PATTERN CLASSIFIER: DESIGN AND RESULTS

A. *Problem statement*

Here, we propose traffic pattern classifier which applies machine learning techniques to aid traffic controllers in their decision on ground holding program parameter settings. As with any machine learning problem, choosing appropriate features which describe the characteristics of the input and the phenomena involved is the key to correct classification. Apart from the ETA, all parameters defining the ground holding control are the same for all cases. Here, we define relative traffic density as the feature describing the ETA queue and the required separation at the control fix. The relative traffic density is calculated every minute in the (0, 60) interval and is here denoted as rtd_i(0<*i*<60). The calculation concept of rtd_i is illustrated in *Figure 9*. Assume one calculates rtd_{20}. We define the vicinity of *i=20* by two parameters, w_{minus} and w_{plus}. In Figure 9, both w_{minus} and w_{plus} take the value of 5 min. As stated earlier, the required separation is *ReqSep* is 2 min, so altogether 5 flights can be handled between $(i - w_{minus})$ and $(i + w_{plus})$. The number of flights with ETA between $(i - w_{minus})$ and $(i + w_{plus})$ is 7.

Therefore, rtd_{20} is calculated as:

$$rtd_i = \frac{number\ of\ ETA\ flights\ in\ [i - w_{minus}\ ,i + w_{plus})}{w_{minus+w_{plus}}/_{ReqSep}} \qquad (4)$$

$$rtd_{20} = \frac{7}{5} = 1.4$$

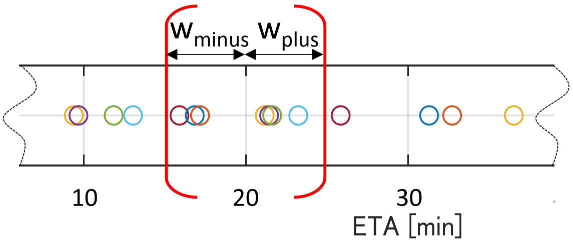


Figure 9. Relative traffic density calculation

For $i < w_{minus}$ and $i > 60 - w_{plus}$, evenly distributed ETA flights 2 min apart are added before and after the ETA queue for consistency in the calculations.

The simulations presented in this paper use all 58 *rtd* values and consider a window defined by $w_{minus} = 2$ min and $w_{plus} = 2$ min. Parameter trial and error tests showed that reducing the number of parameters, i.e. not using all 58 values, can be feasible if dimension reduction becomes necessary at a later stage.

We try to answer three questions by applying machine learning.

1) What are the potential cost savings for the particular ETA queue?

2) What *Buffer* should be set to achieve those savings?

3) How robust are the potential cost savings in respect to the *Buffer* value, i.e. if *Buffer* is selected with a certain error, how much will the achieved savings differ from the potential optimal ones?

We formulate the problem as a regression problem. We use support vector machine with a quadratic kernel. The classifier is developed in MATLAB® 2018b and uses Statistics and Machine Learning Toolbox [14]. Validation is done by cross-validation. The data consists of 1000 ETA queues (*rtd* values) and the median value of the cost function for $Buffer_i$ (*i*=1,2,…15) as determined for each ETA queue ($Cost_i$). We make use of the *Buffer* setting assumptions, which consider only integer values. As seen from the optimization results for each individual ETA queue shown in Section IIIB, The cost function value does not change greatly when the value of *Buffer* is in the vicinity of the optimal *Buffer* value. Therefore, the restriction on *Buffer* being an integer will not impact optimality significantly. In addition, the current system assigning ground holding delays uses integer values for *Buffer*, as well, so from practical perspective the assumption holds as well. $Cost_i$ are the target values for the regression problem. Therefore, for each ETA queue and each $Buffer_i$, a predicted value of $Cost_i$ is found. Next, for each ETA queue, we can determine *CostOptPredicted* and *BufferOptPredicted* as the minimum $Cost_i$ for $i \in [1, 15]$.

B. *Simulation results*

A plot of *CostTrue* versus *CostPredicted* is shown in Figure 10. The root mean square error RMSE of *CostPredicted* over all *Buffer* values is 632.4 EUR. However, the prediction accuracy is considerably higher for *Buffer* values greater than 4 min

(Figure 11). Here, it should be noted that all *BufferOpt* in the original data set lie in the [4,14] min interval. The difference between *CostTrue* and *CostPredicted* for each buffer is shown in Figure 12. As seen from the RMSE values, predictions of the potential cost savings for low buffer values are more inaccurate.

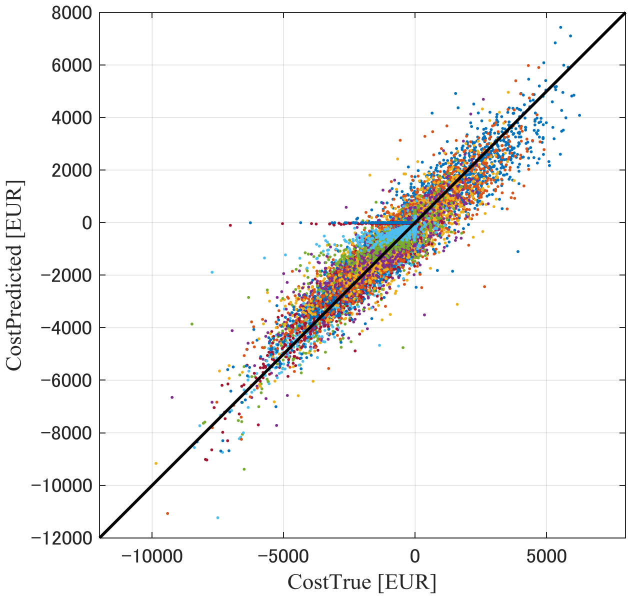


Figure 10. True versus predicted Cost

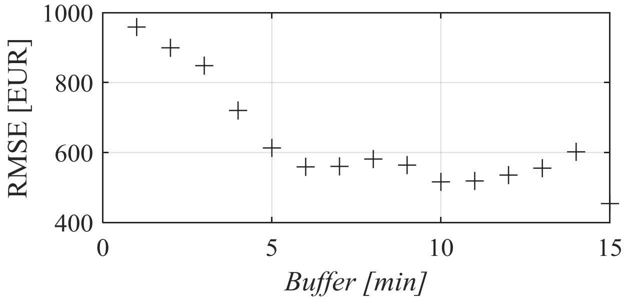


Figure 11. RMSE of *CostPredicted* over all Buffer values

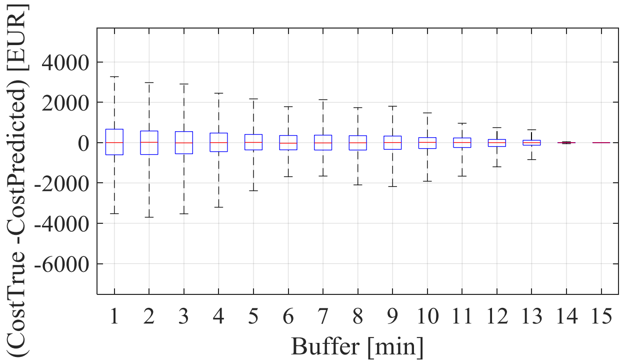


Figure 12. Prediction errors of *Cost* for each *Buffer* value

So far no analysis of the optimal Buffer (*BufferOpt*) and its associated cost (*CostOpt*) is done. Assuming no constraints on the *Buffer* choice exist, *BufferOptPredicted* can be found as the value which minimizes *CostPredicted* for each ETA queue. Comparison of *CostOptTrue* and *CostOptPredicted* is shown in Figure 13. RMSE is 451.5 EUR. The high accuracy of the prediction shows that a decision on whether ground delay should be introduced for a certain ETA queue can be made based on the cost savings predicted by the traffic pattern classifier. This answers Question 1 discussed in the subsection above.

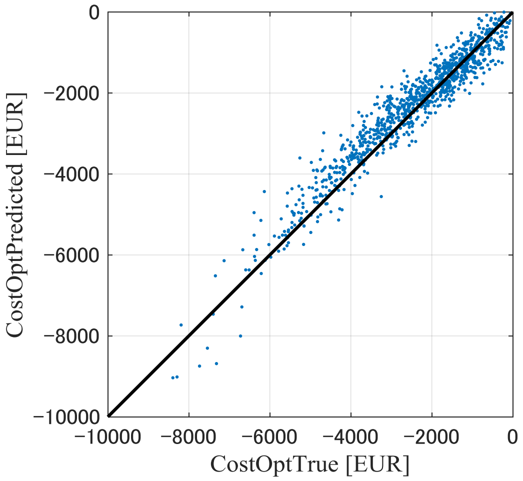


Figure 13. *CostOptTrue* versus *CostOptPredicted*

The answer to Question 2, i.e. what *Buffer* should be set to achieve optimal savings us given by the value of *BufferOptPredicted*. This root mean square error RMSE is 1.28 min.

Next, we investigate the robustness of the solutions, answering Question 3. Assume we choose the optimal *Buffer* value according to the prediction *BufferOptPredicted*. In such a case, the potential savings will be $Cost_{BufferOptPredicted}$, which, by definition are worse than $Cost_{BufferOptTrue}$=*CostOptTrue*. Let us denote these unrealised savings *by* $Cost_{Unrealised}$ and define the mean absolute percentage error *M* as follows:

$$Cost_{Unrealised} = Cost_{BufferOptPredicted} - CostOptTrue \qquad (5)$$

$$M = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{Cost_{BufferOptPredicted\,i} - CostOptTrue_i}{CostOptTrue_i} \right| \qquad (6)$$

Frequency of true and predicted values of *BufferOpt* are shown in Figure 14. Compared with the true values, the predicted values favor *BufferOpt*= 6 min more often. In the true values, the *BufferOpt* is more evenly distributed between 5, 6 and 7 min. *BufferOpt* is the control parameter defining the results of each GH control. Choosing *BufferOpt* according to the predicted value of *BufferOpt* will result is sub-optimal control (unrealised savings).

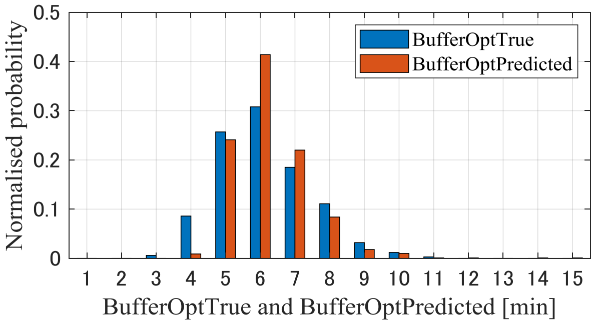


Figure 14. Frequency of each *BufferOpt* value in the true and predicted data

Such unrealised savings due to erroneous *Buffer* selection are shown in the histogram in Figure 15 (average value is 297.5 EUR). In the original data, however, the *CostOpt* sensitivity to *BufferOpt* is not particularly strong around *BufferOpt*, which explains the relative good performance and high accuracy with mean absolute percentage error *M* of 17.96% and RMSE of 472.4 EUR.

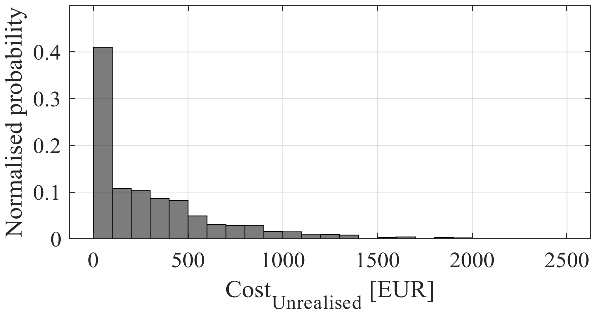


Figure 15. Unrealised savings due to suboptimal *Buffer* selection

From implementation perspective, each ETA queue can be characterized either by its potential *CostOpt*, or classified into a discrete operational class, such as the ones discussed in the operational concept description in Section IIA. Such a classification can provide a quick reference for decision-makers. Discussions with air traffic managers are ongoing regarding the introduction of such classes and the thresholds defining them.

Here, we conclude that our preliminary simulation results are sufficient to prove the feasibility of the traffic pattern classifier concept and its application to traffic management initiatives, in particular ground holding.

V. Concluding remarks

In this paper, we proposed the concept of a traffic pattern classifier applied to optimal ground holding. We tested the concept which consists of real-time air traffic pattern classification and off-line buffer optimization on simulated data and proved its feasibility. The input of the real-time component consists of traffic features, which are fed into a pre-trained machine learning algorithm to determine whether ground holding should be applied to the current traffic and if so, what parameters should be selected for the ground holding program so that the potential savings are the greatest. Based on the ground holding optimal control parameters, departure times can be assigned to each flight part of the ground holding program. Even though the investigated machine learning algorithm needs tuning to improve the performance of the classifier, the classifier successfully predicted the potential savings from ground holding program and predicted an optimal buffer setting which would result in sub-optimal cost savings (the RMSE was 472.4 EUR).

Future work is planned in three major directions. First, the database will be improved to include more detailed models of the traffic and, whenever possible, real past data. We are currently developing departure time error models to describe the departures from major domestic airports in Japan. We are also working on the inclusion of predictability as traffic feature, as discussed by Liu and Hansen [15], to better describe the uncertainties of the environment. Second, the machine learning algorithms used in regression simulations will be revised to improve prediction accuracy. Third, opening the “black box” governing the classifier and transforming it into a grey one, i.e. visualizing some of the decision steps in the classification process and providing this information to controllers is being investigated. Such a “grey box” approach will be essential if the traffic pattern classifier is to be used in practice. Discussions with decision makers for prototype testing in real environment over the next 3 years are also ongoing.

Acknowledgment

This work was supported by JSPS KAKENHI “Synergistic ground holding algorithm based on real-time air traffic pattern classification and off-line buffer optimization”, Grant Number 18H05925 and 19K21092.

References

[1] J. Cox and M. J. Kochenderfer, "Optimization approaches to the single airport Ground-Holding Problem," *Journal of Guidance, Control and Dynamics,* vol. 38, no. 12, pp. 2399-2406, 2015.

[2] H. Naessens, T. Philip, M. Piatek, K. Schippers and P. Robert, ""Predicting flight routes with a Deep Neural Network in the operational Air Traffic Flow and Capacity Management system"," 22 4 2019. [Online]. Available: https://www.eurocontrol.int/sites/default/files/publication/files/predicting-flight-routes-with-neural-network-tpi-technical-document.pdf.

[3] S. Choi, Y. J. Kim, S. Briceno and D. Mavris, "Prediction of weather-induced airline delays based on machine learning algorithms," in *IEEE/AIAA 35th Digital Avionics Systems Conference* , 2016.

[4] S. Gorripaty, A. Pozdnukhov, M. Hansen and Y. Liu, "Identifying similar days for air traffic management," *World Conference on Transport research,* 2016.

[5] A. Estes, M. O. Ball and D. J. Lovell, "Predicting Performance of Ground Delay Programs," in *12th USA/Europe Air Traffic Management R&D Seminar*, Seattle, WA., 2017.

[6] S. Gorripaty, Y. Liu, M. Hansen and A. Pozdnukhov, "Identifying similar days for air traffic management," *Journal of Air Transport Management,* vol. 65, pp. 144-155, 2017.

[7] K. D. Kuhn, "A methodology for identifying similar days in air traffic flow management initiative planning," *Transportation Research Part C: Emerging Technologies,* vol. 69, pp. 1-15, 2019.

[8] Y. Liu and M. Hansen, "Evaluation of the performance of ground delay programs," *Transportation Research Record: Journal of the Transportation Research Board,* no. 2400, pp. 54-64, 2013.

[9] "99th AMS Annual Meeting," 2018. [Online]. Available: https://annual.ametsoc.org/index.cfm/2019/.

[10] A. Andreeva-Mori, Y. Matsuno and N. Matayoshi, "Considering Time Uncertainties in Ground Holding for Optimal Traffic Flow Management," *Aviation Technology, Integration, and Operations Conference,* p. 3841, 2018.

[11] EUROCONTROL, 2015. [Online]. Available: https://www.eurocontrol.int/publications/european-airline-delay-cost-reference-values.

[12] Japan Civil Aviation Bureau, 2017. [Online]. Available: http://www.mlit.go.jp/koku/haneda/news/i/20170201_1.pdf.

[13] IATA, 12 2017. [Online]. Available: https://www.iata.org/pressroom/pr/Pages/2017-12-05-01.aspx.

[14] MathWorks, "MATLAB Statistics and Machine Learning Toolbox," [Online]. Available: https://jp.mathworks.com/help/stats/index.html. [Accessed 31 1 2019].

[15] Y. Liu and M. Hansen, "Incorporating predictability into cost optimization for ground delay programs," *Transportation Science,* vol. 50, no. 1, pp. 132-149, 2016.

Author biographies

Adriana Andreeva-Mori holds B.Eng. (2007), M.Eng. (2009) and Ph.D. (2012) degrees in Aeronautics and Astronautics from the University of Tokyo. She has been with Japan Aerospace Exploration Agency since 2013. Her research interests include air traffic management, in particular trajectory and fleet optimization, new ATM operational initiatives and performance metrics development.

Naoki Matayoshi holds a M.S. (1997) in Aeronautical Engineering from the University of Tokyo. He joined Japan Aerospace Exploration Agency (formerly, National Aerospace Laboratory of Japan) in 1997, where he has conducted flight tests using a helicopter including airborne sensor (GPS, lidar) evaluations and atmospheric turbulence measurements. After a year as a research fellow at the University of Liverpool, since 2010, he is a senior researcher working on ATM-weather integration including dynamic wake vortex separation and advisory system for pilots on low-level windshear or clear air turbulence.