A Testbed for Performance Analysis of Algorithms for Engineless Taxiing with Autonomous Tow Trucks

Stefano Zaninotto
Institute of Aerospace Technologies
University of Malta
Msida Malta
stefano.zaninotto@um.edu.mt

Jason Gauci
Institute of Aerospace Technologies
University of Malta
Msida, Malta
jason.gauci@um.edu.mt

Brian Zammit
Department of Electronic Systems Engineering
University of Malta
Msida, Malta
brian.zammit@um.edu.mt

2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC) | 978-1-6654-3420-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/DASC52595.2021.9594411

***Abstract* — One of the solutions proposed by the aerospace industry to reduce fuel consumption, air pollution and noise at airports consists of using electric tow trucks to tow aircraft from the gate to the runway (or vice-versa). However, the introduction of tow trucks results in an increase in surface traffic, potentially increasing the workload of Air Traffic Controllers (ATC). Many solutions have been proposed in the literature in an attempt to mitigate this problem through the introduction of automated planning and execution. The majority of these solutions have been tested only under strict scenario conditions using a limited number of performance metrics. This paper proposes a simulation testbed that characterizes and compares the performance of such algorithms. By adopting common performance metrics, the testbed allows an objective comparison through the extraction of statistical data using a significant number of scenarios.**

Keywords — taxi systems, electric taxiing, testbed, shortest path, conflict resolution, tow trucks

I. Introduction

The significant growth in air traffic during the last decades has had a big impact on the environment in terms of fuel emissions, air pollution and noise pollution. These effects have been recognized by the European Commission and emission targets have been identified for the next couple of decades. Specifically, the “Flight Path 2050” strategy [1] which was signed in 2017 and the European Green Deal [2] of 2019. The first document aims to mitigate the impact of air traffic on the environment by setting ambitious targets, such as a 70% reduction of CO2 and a 90% reduction of NOx as compared to the year 2000 levels and minimization of noise, while the second document targets emission reductions of 80% by 2035. In addition to these requirements, all taxiing procedures will be required to be carbon neutral by 2050.

Historically, efforts related to reducing emissions have mainly focused on the airborne phase of flight as this constitutes the majority of the flight duration. However, since the aircraft’s engines are optimised for cruising speed, they are highly inefficient when used for taxiing purposes. In addition, high traffic levels at airports, coupled with an inefficient way to manage the taxi operations, will lead to congestion on the taxiways and queues at runway thresholds. Such situations necessitate repeated stop-and-go movements and introduce long engine idle times which increase the levels of emissions. In view of this, aircraft are considered to be the largest single emission source at airports [3].

Reducing emissions throughout the taxi phase of flight is one of the challenges that is being addressed by the Single European Sky ATM Research Joint Undertaking (SESAR JU) programme and two main technologies are being considered by the aerospace industry [4]. For instance, the technology adopted by systems such as WheelTug [5] and the Electric Green Taxiing System (EGTS) [6], relies on the use of electric motors that are installed in the main (or nose) landing gear of the aircraft. An alternative technology relies on the use of manned or unmanned electric tow trucks to tow aircraft from the gate to the runway (or vice-versa). For instance, the TaxiBot solution [7] uses a semi-robotic, pilot-controlled tow truck which has been successfully tested and currently in use at Frankfurt Airport. This solution does not require modifications to the aircraft and it does not add to the aircraft weight. However, the introduction of tow trucks to taxi operations increases surface traffic at the airport, potentially increasing the workload of Air Traffic Controllers (ATC) and creating congestion. This work focuses on the latter technology, that is taxiing using tow trucks and characterises the strengths and weaknesses of selected algorithms by exposing them to various test scenarios using a common testbed and performance metrics.

A number of concepts and systems for taxi operations (both for conventional taxiing and for taxiing with the use of tow trucks) have been proposed in the literature to bring autonomy to ground operations, with the aim of decreasing ATC workload and increasing their situation awareness. However, most of the reviewed solutions have only been tested and characterised under very strict controlled environments (e.g. only at one airport and with a normal level of traffic [8]) and using a single performance metric (e.g. considering only length of the routes [9] or number of conflicts [10]). This limits the validity of the assessment and does not allow a fair comparison of the effectiveness of these and possibly future similar solutions. In view of the above, this paper details the development of a simulation testbed that has been designed specifically to enable an objective performance evaluation of engine-less taxi solutions. A number of solutions, both existing in literature and newly designed, are loaded in the testbed and an objective comparison is carried out by adopting common performance metrics and extracting statistical data from a number of scenarios.

The rest of the paper is organised as follows. Section II gives an overview of an airport environment with autonomous towing operations. Section III discusses the proposed testbed. Section IV introduces the algorithms which were implemented in the testbed. Section V presents the performance metrics and the test results and, finally, Section VI outlines the key conclusions of this paper and highlights areas for future work.

II. The Airport Environment

This section discusses various elements of the aerodrome environment in the context of aircraft taxi operations.

A. Airport Roads

Airport roads can be divided into three main categories: runways, taxiways and service roads.

Runways are reserved for aircraft take-offs and landings and cannot be used by unloaded tow trucks (i.e. tow trucks which are not towing an aircraft), while loaded tow trucks should also avoid using runways (e.g. to cross from one taxiway to another via a runway) unless there is no alternative route. Along the runways, some thresholds are identified for take-off and landing, where departing aircraft complete their mission and arriving aircraft start their mission.

Taxiways connect runways with other sections of the airport (including aprons, hangars and other facilities) and can be used by aircraft and loaded tow trucks. Unloaded tow trucks may also use taxiways if no alternative route is available. The taxiways connect the runways with the parking positions (stands), where departing aircraft start their mission and arriving aircraft complete their mission. Each aircraft has an assigned parking position (see II.D).

Service roads are intended for small vehicles and connect various parts of the airport e.g. an aircraft hangar to a taxiway. They can be used by vehicles, such as cars and unloaded tow trucks, but are not suitable for aircraft movements. The service roads connect the tow truck depots to the taxiways (see II.B).

B. Autonomous Towing

In this work, it is assumed that tow trucks are available at a number of dedicated depots located at strategic positions around the aerodrome. The tow truck depots are used for parking purposes and to recharge the tow trucks and, as already mentioned, they can be reached from the taxiways through the service roads.

A tow truck is allocated to each departing or arriving aircraft. For a departing aircraft, the mission of a tow truck consists of three tasks: (a) move from a tow truck depot to the aircraft parking position; (b) attach to the aircraft, tow it to the runway holding point and detach from the aircraft and (c) return to one of the tow truck depots. Similarly, for an arriving aircraft, the mission of a tow truck consists of three tasks: (a) move from a tow truck depot to the runway holding position; (b) attach to the aircraft, tow it to its allocated parking position and detach from the aircraft and (c) return to one of the tow truck depots.

It is also important to note that a limited number of tow trucks is available in each simulation and, in the case of electric tow trucks, they can only be used when a minimum predefined threshold of State of Charge (SOC) is available. If no tow truck is available when required by an arriving or departing flight, the aircraft has to taxi under its own engine power.

C. Airport Model

The airport is modelled as a directed graph linking the airport's roads, parking positions and tow truck depots. The graph consists of two main components: nodes and edges. The nodes represent all of the relevant points of the airport, including the aircraft stands, tow truck depots, take-off/landing points and intersections between the airport roads. On the other hand, the edges connect pairs of nodes and represent the runways, taxiways and service roads. Therefore, an edge exists between nodes m and n if the nodes are physically connected by a road (e.g. a taxiway).

Using the directed graph representing the airport, the path P of a vehicle can be defined as a sequence of nodes (and corresponding edges) as shown in (1):

$$P = \{n_{st}, \ldots, n_i, \ldots, n_{end}\} \tag{1}$$

where n_{st}, n_i and n_{end} are the start node, the ith node and the end node of the path, respectively.

D. Flight Schedules

The airport's taxi operations are dependent on the airport's flight schedules, which determines the time frame of the arrivals and departures. Each departing aircraft has a planned Off-Block Time (OBT), which is the time that the aircraft is expected to leave its parking position and to start taxiing towards a runway. In reality, the aircraft may leave its parking position later than its planned OBT due to delays e.g. in the loading of cargos and passengers. Pilots may also intentionally delay the aircraft's departure from the stand e.g. to avoid queuing on the taxiways. In this sense, the OBT is somewhat flexible.

Each arriving aircraft has a planned Time of Arrival (TOA), which is the time that the aircraft is expected to land. When an aircraft lands, it is expected to vacate the runway without delay in order not to interfere with subsequent arrivals and departures. For instance, failure of an aircraft to vacate the runway efficiently may force the next landing aircraft to abort the landing and go around.

III. Testbed Design

The testbed is designed to test various algorithms existing in the literature and the results of the simulations allow a comparison between the selected algorithms in order to determine which algorithms perform best in a wide range of scenarios.It was implemented in Matlab and consists of four main components: the Datastore, the Taxiing Algorithm under test, the Graphic User Interface (GUI) and the Core Testbed, which is composed of the following sub-modules: Data Loader, Dispatcher, Vehicles Simulator, Conflict Detector and Performance Indicator. A functional block diagram of the algorithm is shown in Fig. 1 and each block is described in the following sections.

The scope of the testbed is to provide a test environment to compare various algorithms existing in literature, therefore the module 'algorithm' varies according to the solution which is being tested. These algorithm aim to provide strategic solutions to the taxiing operations, making decisions before the start of the simulation run.

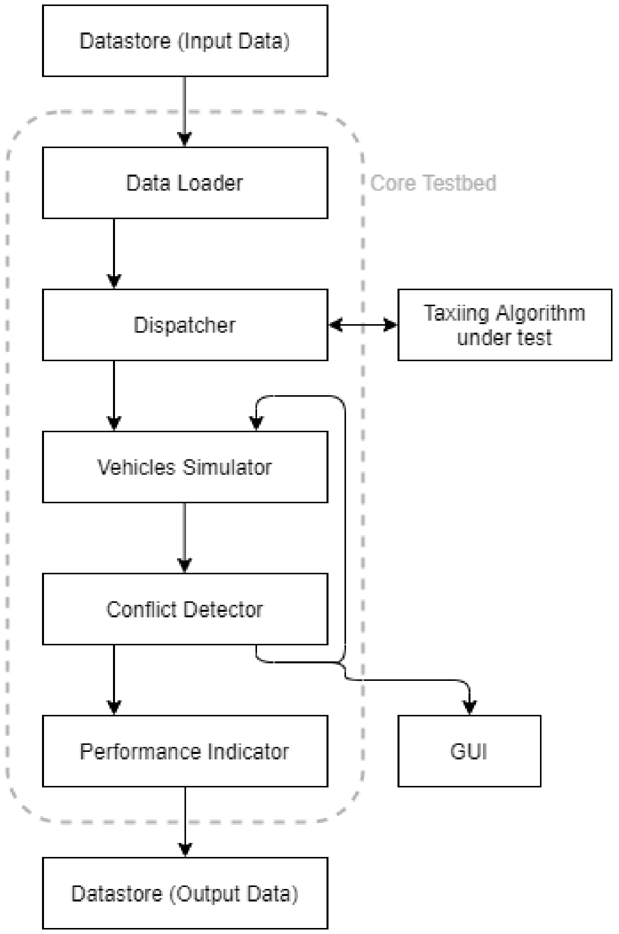

Fig. 1. Structure of the testbed.

A. *Datastore*

Table 1 shows how the input and output data of the testbed.

TABLE 1. LIST OF THE TYPES OF DATA INCLUDED IN THE DATASTORE.

Datastore	
Data type	***Input / Output***
General data	Input
Geometric data	Input
Simulations settings	Input
Performance data	Output

General data of the system includes time data (such as timestamp and length of simulations), relevant features of aircraft and tow trucks (e.g. dimensions and simulation model details of the vehicles), and graphic data (e.g. vehicle icons).

Geometric data contains tables with nodes and edges of the airport graphs, parking locations, take-off / landing points and tow truck depots positions. In particular, given the fact that the selected airports have no reserved tow truck depots, the position of the tow truck depots can be selected as necessary based on considerations such as available space and sufficient connections to the taxiways and to the service roads.

Simulations settings include all the data which define the different scenarios, for instance the minimum and maximum number of aircraft per hour for each airport. The chosen settings are described in detail in section V.B.

Finally, performance data are generated at the end of each simulation, and they represent the output of the test according to the chosen metrics as described in section V.A.

B. *Graphical User Interface (GUI)*

The GUI provides a real-time graphical representation of the position and heading of each vehicle and highlights conflicts between vehicles. The GUI module is an optional feature of the testbed and as such is not required during operation and can be turned off.

C. *Data Loader*

The Data Loader extracts layout information related to the selected airport, relevant simulation settings from the datastore, and creates the airport environment together with a random flight schedule for each type of simulation.

Fig.2 shows a graphical representation of Ben Gurion Airport in Tel Aviv superimposed with the identification of all roads, nodes and salient features.

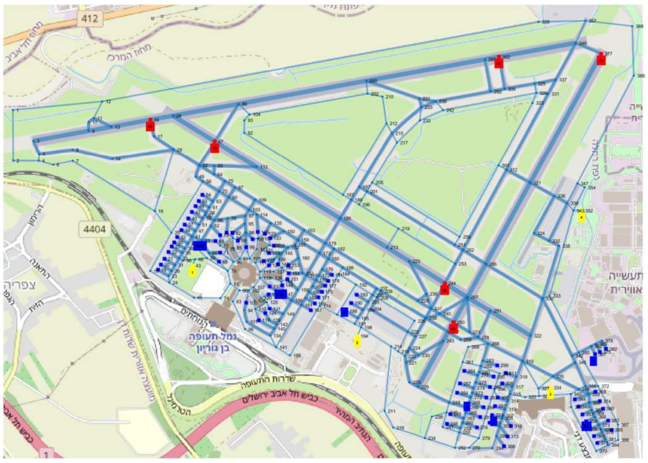

Fig. 2. Direct graph for Ben Gurion Airport. The thickest lines, the medium thickness lines and the thinnest lines represent respectively the runways, the taxiways and the service roads. The nodes in blue represent the parking locations, the nodes in red mark the take-off / landing points and the yellow nodes show the position of the tow truck depots.

Table 2 shows an extract of a flight schedule generated by the Data Loader module. Arrivals and departures are assigned randomly, however they both represent the 50% of the total flights. The parking stands (and the corresponding aprons) are also assigned randomly, but, once assigned, become unavailable to a new selection for a certain time, in order to avoid an overlaying of two or more aircraft on the same parking stand.

TABLE 2. EXAMPLE OF FLIGHT SCHEDULE GENERATED FOR BEN GURION AIRPORT.

Flight schedule					
Flight	***Aircraft***	***A/D***[a]	***TOA/OBT***	***Apron***	***Parking***
1	A330-202	D	10:00:00	APR J	257
2	A319	D	10:05:00	APR J	276
3	B767	A	10:05:00	APR WHS	71
4	A330-202	A	10:07:00	APR L	313
5	ARJ85	A	10:10:00	APR WHS	73
6	A340-500	D	10:10:00	APR EHS	138
7	A319	A	10:12:00	APR WHS	99
8	ARJ85	A	10:14:00	APR WHS	115
9	A330-202	D	10:15:00	APR WHS	19
10	A321	A	10:20:00	APR BE	374
11	A330-234	A	10:22:00	APR EHS	179
12	ARJ85	D	10:25:00	APR EHS	167
13	B767	D	10:25:00	APR EHS	181
14	A340-500	D	10:30:00	APR EHS	126
15	A319	A	10:30:00	APR EHS	127
16	B757	A	10:32:00	APR BE	385
17	A321	D	10:40:00	APR EHS	169
18	B767	D	10:40:00	APR L	300
19	ARJ85	A	10:45:00	APR L	289
20	ARJ85	A	10:47:00	APR L	316
21	A330-202	D	10:50:00	APR EHS	100
22	A319	D	10:55:00	APR WHS	106

[a.] Arrival or departure

D. Dispatcher

The Dispatcher is designed to collaborate with the Taxiing Algorithm under test. It requests to the algorithm the following tasks:

1. Assigns a taxi route to each aircraft;
2. Adjusts the OBT for each departing aircraft where necessary;
3. Allocates tow trucks (if available) to the selected aircraft;
4. Assigns a route to each tow trucks from the depot to the aircraft and vice-versa;
5. Creates a schedule for the tow trucks, assigning the time to leave the depot and estimating the time to return to the depot;

Once the algorithm elaborates solutions to the above mentioned tasks (e.g. route, adjusted schedule, tow truck's assignment), these are communicated to the Dispatcher, which store them and finally triggers the start of the simulation.

The Dispatcher can cater for hybrid taxi operations (i.e. operations where some aircraft taxi using their own engines, while others taxi by means of tugs) according to the number of tow trucks available and the behaviour of the algorithm under test.

E. Taxiing Algorithm under test

The decisions behind the behavior of the vehicles during the simulations are driven using the loaded taxiing algorithm under test, which makes decisions about the routes and the timetables of the vehicles in order to maximize some performance metrics.

The algorithm communicates directly with the Dispatcher module (see a), which provides inputs to the algorithm (such as whether an aircraft requires towing or otherwise) and receives back outputs (including for example assignment of tow truck or instructions to start taxiing using the aircraft engines when no free tow trucks are available for instance). Section IV describes the algorithms which have been implemented and tested in the testbed.

F. Vehicles Simulator

The Vehicles Simulator module periodically updates the position, heading, speed and acceleration of each vehicle with a frequency of 1 second. It is capable of executing the instructions given to each vehicle by the Dispatcher module and by the Conflict Detector module (see section III.G), such as following a certain route, halting before or at a certain node and stopping if necessary when a potential conflict is detected.

The equations that define the Vehicles Simulator were adopted from previous work [12], however in this work two different versions of the Vehicles Simulator are introduced. The first version is a deterministic model (as the one described in previous work) that assigns fixed values to the maximum acceleration a_{max}, to the maximum undisturbed velocity v_{maxu}, and to the maximum cornering velocity v_{maxc}, while the second version is a probabilistic model that samples the parameter values using a normal distribution. For instance, the maximum acceleration is determined through the mean and standard deviation values as shown in equation (2):

$$a_{max} = f\,(\mu_{a\_max}, \sigma_{a\_max}) \qquad (2)$$

where:

μ_{a_max} is the mean of the maximum acceleration;

σ_{a_max} is the standard deviation of the maximum acceleration.

Table 3 shows the mean and the standard deviation assigned to each variable in the Vehicles Simulator. The values have been chosen to mimic the expected variations during operation.

TABLE 3. MEANS AND STANDARD DEVIATIONS VALUES OF VEHICLES SIMULATOR VARIABLES.

Normal distribution values		
Variable	***Mean***	***Standard deviation***
a_{max}	5 m/s^2	1 m/s^2
v_{maxu}	15 m/s	1.5 m/s
v_{maxc}	5 m/s	1 m/s

The probabilistic model was implemented in order to introduce uncertainty to the simulations and to test the robustness of the algorithms under test. This is based on the fact that, in practice, each vehicle (tow truck or aircraft) may have different motion characteristics and therefore it is hard to predict how the speed and position of a vehicle will evolve over time. By introducing uncertainty to the Vehicles Simulator, a more realistic simulation can be generated and the sensitivity of the algorithms under test to this uncertainty can be determined.

G. *Conflict Detector*

The Conflict Detector runs in parallel to the Vehicles Simulator module. While the vehicles are in motion, it monitors potential conflicts between aircraft, as well as between aircraft and tow trucks, and it classifies these conflicts as solvable or non-solvable (e.g. deadlocks). A conflict is defined when two vehicles get closer to each other than a safety distance equal to 35m during a simulation run. When a conflict is detected, the Conflict Detector communicates with the Vehicles Simulator to slow down and, if necessary, to stop the vehicles involved in order to try to resolve the conflict.

However, in case the conflict is non-solvable, in order to interrupt the simulation, an arbitrary time penalty equal to 30s is assigned to the vehicles, and after that they can resume their mission again. This behaviour will be revised in the future works, as explained in section VII.B.

H. *Performance Indicator*

At the end of each simulation, the Performance Indicator module computes and processes the statistical results for each of the performance metrics.

IV. Selection of Algorithms Under Test

This section describes three algorithms which were selected for testing on the proposed testbed and discusses the motivation behind their selection.

A. *Algorithm Under Test 1: Baseline algorithm*

In order to have a baseline to be able to compare the other algorithms under test, a basic algorithm was introduced. This algorithm has the following behavior:

1. A route is assigned to each aircraft using the shortest-path Dijkstra algorithm;
2. OBT is fixed and no adjustments to this value are allowed;
3. The closest tow truck (if available) is assigned to the selected aircraft;
4. The route of the assigned tow truck is also determined using the Dijkstra's algorithm;
5. A tow truck schedule is generated by calculating the distance between the depot and the aircraft, and vice-versa.

This algorithm does not take into account the position of other vehicles in the simulations. In other words, each vehicle is treated as if it is the only vehicle in the simulation. This means that each vehicle is assigned an optimal route.

B. *Algorithm Under Test 2: Path planning with dynamic obstacles*

Finding a suitable route for a vehicle in an airport environment, in the presence of other moving vehicles, can be seen as a problem of path planning with dynamic obstacles. This problem is considered by Vemula et al. [10], by using the idea of adaptive dimensionality to limit the number of solutions and speed up the calculation time. Specifically their approach considers the time dimension only in those regions of the environment where a potential collision may occur (see Fig. 3), and plans in a low-dimensional state-space elsewhere. The method is validated in the problem of 2D vehicle navigation (coordinates of position x and y, and heading) in a dynamic environment, which can be easily adapted to taxi operations in an airport environment. The calculation of the routes is based on the A* shortest path algorithm.

The algorithm is based on a graph which is divided into High Dimensional Regions (HDRs) and Low Dimensional Regions (LDRs). The HDRs represent the nodes and the edges of the graph where the path of a dynamic obstacle (or, in other words, the path of another vehicle) is planned to exist, while the LDRs represents the edges and the nodes where no paths of dynamic obstacles are planned to pass.

In order to implement this in the testbed, the HDRs and the LDRs are updated every time the path of a vehicle is calculated, extending the HDRs along its route (if not already part of it) and reducing therefore the LDRs.

The algorithm analyses one vehicle a time and is divided in three phases: Adaptive Dimensionality Planning phase, tracking phase and graph updating phase.

During the first phase, the graph is searched for a path from the start to the goal, using a suboptimal graph search algorithm like weighted A*, which assigns a heavier weight to HDRs, thus encouraging the passage of the vehicles on the LDRs.

In the second phase, a high-dimensional level (i.e. the time dimension) is assigned to the calculated path. In this way, all the nodes and the edges covered by the vehicle are time-tracked. Then, when the path meets a HDR, the time-tracked path of the vehicle is compared to the ones calculated for the other vehicles in that area. If a possible conflict (detected when two or more vehicles pass through the same edge at the same time) is detected (i.e. if the vehicle is in the same position of any other vehicle at the same time), a part of the route is recalculated until a new conflict-free path is found or until all the possible options are explored and no collision-free-paths are found. The algorithm then repeats the process for all the HDRs area met by the path.

Finally in the third phase, the graph is updated, extending the HDRs and reducing the LDRs, if needed.

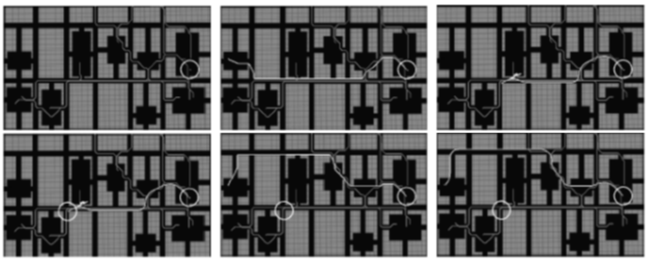

Fig. 3. Example run of the algorithm on a sample map. High Dimensional regions (i.e. regions which takes into account the time dimensions) are indicated by white circles, paths of dynamic obstacles by grey lines and final path by white lines [11].

To summarize, the algorithm has the following behavior:

1. A route is assigned to each aircraft with the process described above;
2. OBT is fixed and no adjustments to this value are allowed;
3. Tow trucks will be not assigned;
4. No tow trucks' routes is generated;
5. No tow trucks' schedule is generated.

The results in [10] show that the approach returns feasible, collision-free paths in dynamic environments. In addition, the approach achieves substantial speedups in planning time compared with a traditional 4D heuristic-based A*.

However, for the testbed of this work, this algorithm offers a limited number of solutions. For instance, it does not include the option of postponing the OBT of a departing aircraft.

Also, another limitation is represented by the fact that the algorithm was not designed for autonomous taxiing operations. Therefore the system can only be tested and verified for correct operation by excluding the tow trucks from the simulation runs.

Finally, the algorithm was designed assuming that the position over time of the dynamic obstacles is known a priori, therefore it was tested just with a deterministic version of the simulator.

C. Algorithm Under Test 3: Taxi Optimisation System Using Tow Trucks

The third algorithm was defined in previous work [12], and includes two functions that define the behavior of the vehicles prior to the start of the simulations.

The first function, called *Tow Truck Allocation*, allocates a tow truck to an aircraft by calculating the time it takes a tow truck to reach the aircraft attachment point from each of the tow truck depots available. The *Tow Truck Allocation* function also determines which depot a tow truck should return to once it detaches from an aircraft. This is done by calculating the time it takes for the tow truck to travel from the detachment point to each of the tow truck depots and selecting the depot which results in the lowest trip time.

The second function, called *Path and Time Allocation*, finds the least-cost path for each complete tow truck mission. The function also calculates the start time of each mission (i.e. the TDT) and may introduce minor adjustments to the OBT of a departing aircraft if necessary.

The algorithm has the following behavior:

1. A route is assigned to each aircraft using the method described in previous work using *Path and Time Allocation* function;
2. OBT might be adjusted if required by the *Path and Time Allocation* function;
3. A tow truck (if available) is assigned to the selected aircraft according to the *Tow Truck Allocation* function;
4. The route of the assigned tow truck is also determined using *Path and Time Allocation* function;
5. A tow truck schedule is generated by calculating the distance between the depot and the aircraft, and vice-versa.

This system guarantees a wide number of solutions, including the possibility of postponing the OBT of the departing aircraft and a module to allocate tow trucks in the simulations. A limitation that is perceived for this algorithm is that the system has been calibrated using a deterministic version of the simulation model, and therefore it might not respond well to the use of the probabilistic model.

V. Performance Metrics and Test Scenarios

In this section, the metrics chosen for the simulations are defined and the scenarios implemented in the testbed are described.

A. Performance Metrics

The selection and definition of the performance metrics was carried out by considering the objective of the testbed. One of the main reasons to use tow trucks for taxiing purposes is to reduce fuel consumption and, as a result, fuel emissions. Hence, one of the metrics adopted is the **average fuel consumption**, which is estimated using a model provided by Khadilkar [13] and averaged between all the aircraft's fuel consumption of the simulation run. In this work, it is assumed that the tow trucks are electric and their fuel consumption is equal to zero.

Considering that the taxi operations must be carried out in restricted time ranges dictated by the flight schedule, the **average taxi time** and **average taxi delay** are also included as performance metrics. Average taxi delay is defined as the average difference between the actual taxi time of an aircraft and the taxi time obtained in an ideal situation (i.e. with a single aircraft) where an aircraft takes the shortest route (as in the case of the baseline algorithm).

One of the objectives of the work is to generate conflict-free routes as much as possible; thus, the performance metrics also include the **number of potential conflicts** and the **number of potential deadlocks**. These metrics provide an indication of how effective the strategic part of the algorithm under test is for various traffic conditions and how much the introduction of the tow trucks affects the system.

As discussed previously, one of the methods to create conflict-free routes consists of postponing the OBT by a few minutes. Therefore, two other metrics which are considered are the **average delay of the departing aircraft**, which measures the delay from the planned OBT, and the **ratio of delayed departure to the total number of departures**. A delayed departure is defined as a departure which occurs outside a certain time window (five minutes before the OBT to ten minutes after the OBT).

In order to ensure that the tow trucks are used efficiently during engineless taxiing operations, it is important to maximise their use. Hence, a last metric which is considered is the **tow truck utilisation**, defined as the ratio of the average time of tow truck activity to the total simulation time.

B. Test Scenarios

Each of the selected algorithms was tested by means of scenarios (a total of 3,234 when the deterministic model of the simulator was used, and a total of 323,400 when the probabilistic model of the simulator was used) defined for different combinations of the following parameters:

1. 1. Airport size and geometry: Four airports with different sizes and geometry were chosen, i.e. Malta International Airport (an airport of small dimensions with 2 runways), Toulouse–Blagnac Airport (an airport of medium dimensions with 4 parallel runways), Ben Gurion Airport in Tel Aviv (an airport of medium dimensions with 3 runways, chosen for the peculiar geometry of the runways and taxiways, as shown in Fig. 2), and Dallas/Fort Worth International Airport, (a large airport with 4 parallel runways and one of the busiest in the world [14]).
2. 2. Runways in use: Tests were carried out for each of the runways in use at each airport. Malta International Airport and Ben Gurion Airport can only have one active runway (for commercial aircraft) at a time; Toulouse–Blagnac Airport can have two active runways at a time; and Dallas/Fort Worth can have four active runways at a time;
3. Level of traffic: Defined as the number of aircraft per hour, the level of traffic was varied from a minimum equal to 20 aircraft per hour (which guarantees undisturbed routes to the vehicles) to a maximum of 60 aircraft per hour, which represents a saturation point for the chosen airport;
4. Percentage of tow trucks to aircraft: This parameter is defined as the ratio of the number of tow trucks available to the number of aircraft movements per hour. This varies from a minimum (zero) to a maximum (equal to the number of aircraft involved in the simulation).

Finally, when the probabilistic model of the Vehicles Simulator was used (testing the Taxi Optimisation System Using Tow Trucks algorithm), each scenarios was repeated 100 times resulting in a probability distribution of results.

VI. Results and Discussion

This section presents some of the results which highlight the performance of the selected algorithms.

A. Results and Discussion

The results obtained for different runways in use at the same airport were averaged in order to have more homogeneous results.

Fig. 4 compares the algorithms under test with respect to the average taxi delay for different levels of traffic at Ben Gurion Airport. The ratio of tow trucks to aircraft in the system was set to 0 in this case (i.e. all aircraft taxi using their own engine power) in order to allow a comparison between all algorithms.

As can be noticed from the graph, while the Baseline solution algorithm performs poorly over a certain level of traffic, the Path Planning with Dynamic Obstacles and the Taxi Optimisation System Using Tow Trucks algorithms, both operating with the deterministic model of the Vehicles Simulator function, are able to contain the delays even in heavy traffic situations. This is due to the fact that a planning of the route to minimize conflicts ensures a reduction of delays during the routes, as the vehicles are rarely force to stop to solve conflicts along their paths. Finally, the results of the Taxi Optimisation System Using Tow Trucks algorithm operating with the probabilistic model of the Vehicles Simulator function, show a reduction in performance, particularly at high levels of traffic.

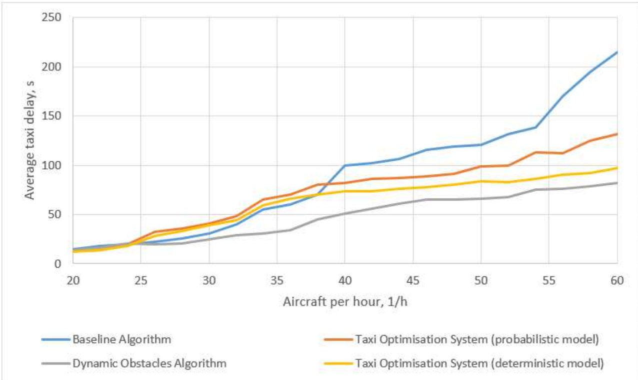

Fig. 4. Average taxi delay in Ben Gurion Airport.

A similar trend can be noticed in Fig. 5, where the same algorithms were compared with regards to the number of conflicts for different levels of traffic at Toulouse-Blagnac Airport. Also, in this case, the ratio of tow trucks to aircraft in the system was set to 0.

While the *Baseline Solution* algorithm again performs poorly, the *Path Planning with Dynamic Obstacles* solution shows a lower number of conflicts for the same level of traffic. The *Taxi Optimisation System Using Tow Trucks* algorithms, when operating with the deterministic model of *Vehicles Simulator*, performs very well, even in situations of high traffic, keeping the number of conflicts relatively low. This can be explained by the fact that the algorithm attempts to avoid conflicts not only by changing the route of the aircraft, but also by postponing its OBT (in case of departures), therefore offering a wider range of solutions. However, when the probabilistic model of the *Vehicles Simulator* is used, the performance of the same algorithm decreases decisively. In particular, for a number of aircraft per hour equal to 40, Fig. 6 shows the distribution of the recorded number of conflicts among the 100 simulations which were carried out. This is due to the fact that the algorithm uses a deterministic model to plan the paths, while the simulation varies the dynamic models, using the probabilistic model of the vehicle simulation. These changes in taxiing behavior are not accounted for during planning phase and therefore interfere with the plan, increasing the number of conflicts.

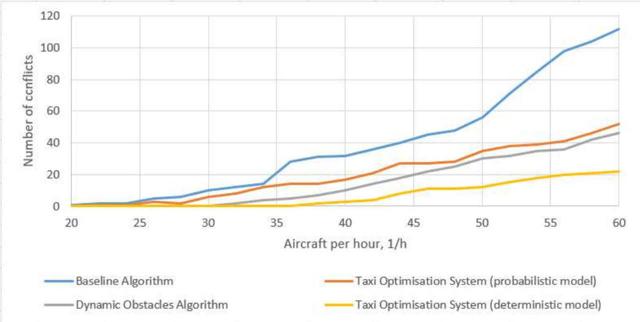

Fig. 5. Number of conflicts in Toulouse–Blagnac Airport.

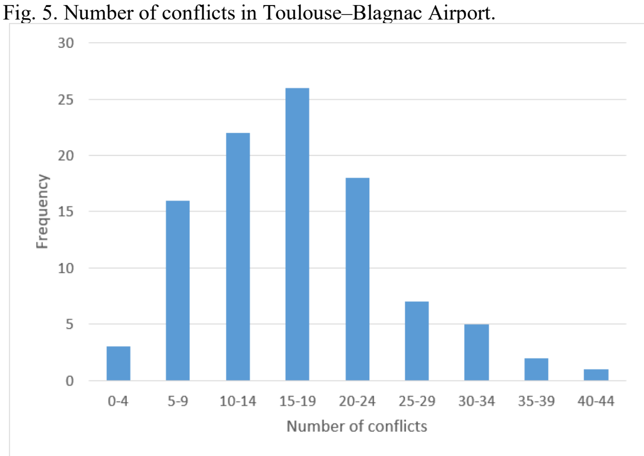

Fig. 6. Distribution of the recorded number of conflicts for 100 simulation runs using a single path plans.

The average fuel consumption for different percentages of tow trucks at Dallas-Fort Worth airport, is shown in Fig. 7. In this case, the number of aircraft per hour was set to 40. The results show that, even in the case of no tow trucks, the fuel consumption was lower when the tests were carried out with the Taxi Optimisation System Using Tow Trucks algorithm compared to when tested with the Baseline Solution algorithm, and decreased rapidly as the percentage of tow trucks increased. Also, when the deterministic model for the simulation is used, the algorithm performs better compared to when the probabilistic model is used. Furthermore, in this simulation, the Path Planning with Dynamic Obstacles algorithm was not tested, as it does not include a module to assign the tow trucks to the aircraft, as explained in IV.B. The good performance of the Taxi Optimisation System Using Tow Trucks algorithm can be explained with the fact that the creation of conflict-free routes allows the aircraft to avoid a stop-and-go situation (generated by repeated conflicts) and therefore to reduce the fuel consumption. For the same reason, as the algorithm tested with the probabilistic model does not perform as good as the algorithm tested with the deterministic model in avoiding conflicts, it performs worse also in terms of average fuel consumption.

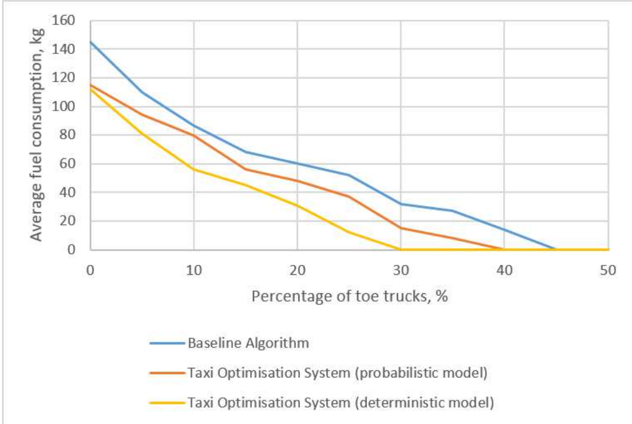

Fig. 7. Fuel consumption in Dallas Fort-Worth Airport.

VII. Conclusions and Future Work

This section outlines the key conclusions of this paper and highlights areas for future work.

A. Conclusions

This paper introduced a testbed to analyze the performances of various algorithms in the context of airport taxi operations with the use of tow trucks. An environment was created to perform the simulations and the testbed was structured to implement a number of algorithms, which were designed in previous work or chosen from the literature. A number of performance metrics were defined and a large number of simulations with a wide number of settings were carried out. In particular, two different models were implemented to simulate the movement of the vehicle: a deterministic model (based on previous works) and a probabilistic model, which attempts to reproduce the uncertainty of taxi operations.

An algorithm called *Baseline Solution* was introduced in order to show the performance of the system without a proper regulation of the taxiing. A second algorithm taken from the literature, *Path planning with dynamic obstacles*, was tested with the deterministic model of the simulator and without the use of tow trucks, and proved to perform well. A third algorithm, taken from previous work by the author, *Taxi Optimisation System Using Tow Trucks*, was tested both with the deterministic and with the probabilistic model, and both with and without the use of tow trucks. The performances of this algorithm proved to be very high when the deterministic model was used, but definitely worse when the probabilistic model was in use.

B. Future Work

All the algorithms implemented in the testbed offer strategic solutions. This approach might be not sufficient, as the taxi operations are subjected to a certain degree of uncertainty, which can disrupt the decision made by the strategic algorithms. Therefore future work should go in two directions: first, the strategic solutions should be adapted to cater for a probabilistic approach of the simulations, such as the one proposed in this paper; second, a number of tactical solutions, which should be able to operate in real-time and respond to unforeseen events, should be implemented in the testbed environment.

Other areas of future work might concern the research of solutions to prevent deadlocks (both strategically and tactically) and the inclusion of additional uncertainty factors in the probabilistic model of the *Vehicles Simulator*.

In addition, the testbed should be expanded (e.g. by implementing new airports) and new solutions to assign tow trucks to aircraft should be introduced to maximize tow truck utilization.

REFERENCES

[1] European Commission, "Flightpath 2050, Europe's Vision for Aviation." [Online]. Available: https://ec.europa.eu/transport/sites/transport/files/modes/air/doc/flightpath2050.pdf [Accessed: 13 July 2021].

[2] [Online]. Available: https://ec.europa.eu/clima/policies/transport/aviation_en [Accessed: 13 July 2021].

[3] E. Fleuti, and S. Maraini, Taxi-Emissions at Zurich Airport, 2017.

[4] R. Guo, Y. Zhang, and Q. Wang, "Comparison of emerging ground propulsion systems for electrified aircraft taxi operations", Department of Civil and Environmental Engineering, University of South Florida, Florida, USA.

[5] [Online]. Available: http://www.wheeltug.com/. [Accessed: 13 July 2021].

[6] G. Norris, "Honeywell/Safran Joint Venture Tests Electric Taxiing," 24 June 2013. [Online]. Available: http://aviationweek.com/awin/honeywellsafran-joint-venture-tests-electric-taxiing. [Accessed 13 July 2021].

[7] "Taxibot," [online]. Available: http://www.taxibot-international.com [Accessed 13 July 2021].

[8] J. Smeltink, P. de Waal, and R. van der Mei, "An Optimisation Model for Airport Taxi Scheduling," in Proceedings of the INFORMS Annual Meeting, Denver, Colorado, USA, 2004.

[9] G. Sirigu., M. Cassaro, M. Battipede., and P. Gili., "A Route Selection Problem Applied to Auto-Piloted Aircraft Tugs", in WSEAS Transactions on Electronics, International Conference on Applied and Theoretical Mechanics, Venice, Italy, 2017.

[10] A. Vemula, K. Muelling, and J. Oh, "Path Planning in Dynamic Environments with Adaptive Dimensionality", Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.

[11] A. Vemula, K. Muelling, and J. Oh, "Path Planning in Dynamic Environments with Adaptive Dimensionality", Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, p. 4, Fig.2.

[12] S. Zaninotto, J. Gauci, G. Farrugia, and J. Debattista, "Design of a Human-in-the-Loop Aircraft Taxi Optimisation System Using Autonomous Tow Trucks", University of Malta, Malta, 2017.

[13] H. Khadilkar, "Estimation of Aircraft Taxi-out Fuel Burn using Flight Data Recorder Archives", Massachusetts Institute of Technology, Cambridge, USA, 2011.

[14] [Online]. Available: https://www.world-airport-codes.com/world-top-30-airports.html [Accessed: 13 July 2021].