Safety Monitoring and Prognostics for Automatic Aircraft Take-off

Johann Schumann[1]*, NASA Ames Research Center, Alexander W. Zollitsch[2], Nils Mumm[3], and Florian Holzapfel[4]

[1,2,3,4] *Institute of Flight System Dynamics, Technical University of Munich, Germany*

johann.schumann@tum.de, alexander.zollitsch@tum.de, nils.christian.mumm@tum.de, florian.holzapfel@tum.de

Abstract

The take-off of an aircraft is one of the most dangerous flight phases, as failures and adverse environmental conditions can lead to a catastrophe. Should abnormal events occur during the roll phase, the crew or the flight computer has to make the decision if the take-off can be safely rejected and the aircraft can brake and come to a standstill on the runway, or if the take-off has to be attempted in any case.

This decision has to be made instantaneously upon the estimate of the current state of the aircraft, using available sensor data under noise and potential failure conditions. In order to do so, at any time during the roll phase, a prediction has to be made, if the aircraft can come to a safe stop within the boundaries of the runway.

In this paper, we formulate this decision making task as an online prognostics problem and develop a model-based architecture that allows us to perform a probabilistic prediction of the aircraft's braking distance given the current aircraft state. We are using particle filter and Monte-Carlo based prediction algorithms. Because this task has to be performed in real-time on the on-board flight computer, computational resources are very restricted. We therefore propose several models of increasing fidelity, which have substantially different computational footprints and exhibit different levels of accuracy that can impose severe restrictions on the handling of uncertainties and on the failures that can be modeled.

1. Introduction

For any aircraft, take-off is one of the flight-phases with highest risk. According to (Boston Globe, 2014), NTSB records reaching back to 1982 reveal that the take-off is the second riskiest flight phase (14,201 incidents) after landing (17,914). Catastrophic accidents during take-off have been caused by wake turbulence of a preceding aircraft (National Transportation Safety Board, 2004), obstacles on the runway (for example, Concorde AF4590 (Bureau Enquêtes-Accidents, 2000) or (Netherlands Aviation Safety Board, 1978)), sudden moves of center of gravity (National Transportation Safety Board, 2015), or incorrect take-off configurations (Comisión de Investigación de Accidentes e Incidentes de Aviación Civil, 2008; National Transportation Safety Board, 1988).

Other dangerous situations can include high air temperatures (Fortune, 2017), icy or wet runways, blown tires, failing brakes, covered/blocked Pitot tubes (Australian Transportation Safety Board, 2016; Hinson, 1996; Directorate General of Air Transport, 1996), as well as errors during pre-take-off calculation of the take-off parameters (Hughes & Godley, 2011). Finally, tail strikes during take-off can damage the aircraft.

The take-off phase is particularly risky, because a decision about performing a take-off or aborting (rejecting) it must be done within a very short time-frame, taking current aircraft status (e.g., speed), aircraft configuration (e.g., weight), as well as environmental issues (e.g., wet or icy runway) and failures (e.g., failed engine) into account.

The entire take-off procedure consists of a ground roll phase, where the aircraft accelerates, a rotation phase, and an initial climb phase, which ends once the aircraft has reached an altitude of $35\,\mathrm{ft}$ above ground level (AGL) and the aircraft reaches the take-off safety speed V_2 (U.S. National Archives and Records Administration, 1965, 14CFR25.111).

Only during the initial ground roll, a safe take-off rejection is possible. Usually, flight procedures (Federal Avation Administration, 2016) are guided by a pre-calculated and fixed decision speed V_1. If the aircraft during take-off run has not yet reached V_1, a safe abort is possible and the aircraft can come to a standstill before the runway ends. This abort criterion implicitly assumes that the aircraft accelerates reasonably on the runway, which is intuitively monitored by the pilot throughout take-off run. After passing V_1 the pilot will have to attempt a take-off, even with one engine off. For an unmanned aircraft (UAV), the three phases of take-off are the

*on leave from Stinger Ghaffarian Technologies, Inc., NASA Ames Research Center, johann.m.schumann@nasa.gov same. However, all decisions have to be taken by the flight-control software as no pilot is on board. Even a ground operator might not be a means for adequate mitigation as he or she does not have any haptic feedback of the behavior of the system and therefore has only limited situational awareness. Therefore, a better solution, in addition to a ground operator, is to automatically monitor the current situation with respect to the take-off process and evaluate the risks and consequences of alternative decisions. On today's UAVs a large amount of sensor measurements and information is already available, which can be used for this task.

In essence, the determination if a take-off can be executed or rejected safely corresponds to a prediction problem: given the current state of the aircraft (position, speed, failures, etc.), what is the probability P_{to} that a take-off can be done safely (according to the regulations), and what is the probability P_{rej} that a reject and brake maneuver can slow down the aircraft sufficiently such that it comes to a standstill on the runway? As soon as one of those probabilities, P_{to} and P_{rej}, becomes smaller than a certain threshold, a decision must be made.

Since failures and uncertainties can occur on both possible actions, a single and fixed speed V_1 is not universally suited to make that decision. For example, engine problems, locked brakes, or heavy head-wind can cause the aircraft to accelerate slower than expected. So, when finally V_1 has been reached, the aircraft is at a location, where the remaining runway distance does not suffice to stop safely. Therefore, (U.S. National Archives and Records Administration, 1965) takes the failure of one engine into account for the calculation of V_1, but no other effects.

In our approach, we do not calculate V_1. Rather, at each moment during the take-off run we calculate the necessary remaining safety distances and their probabilities for accepted and rejected take-off. For this problem, noise as well as numerous sources of uncertainties (e.g., wet or icy runway, wind, turbulence, or failures) have to be taken into account. We therefore estimate the probability density of the required remaining runway distance for both possible actions.

In this paper, we address this problem using model-based prognostics techniques, which allow us to use physical dynamics models for the aircraft acceleration and brake behavior as well as probabilistic models for failures and environmental influences. We develop an architecture for a take-off monitoring system that uses particle filters for state estimation and prediction, and which can provide the necessary information to support the on-board decision-making logic that ultimately will accept or actively reject the take-off. This software algorithm will be running on the on-board flight computer to be able to produce its results in real time. Since particle filters and Monte Carlo-based prediction algorithms require substantial computational resources, we propose to use models of different complexity to optimize the computational burden without sacrificing too much accuracy and confidence.

In this paper, we will address the following research question:

- Can our model-based approach be used to predict probability distributions for safe take-off and stopping distances in the nominal case, under wind/turbulence, and under analyzed failure modes (e.g., engine failure, brake failure)?

In order to address our research question, models at varying fidelity are presented in this paper. However, only one model, namely model M_3, is assessed in detail in the context of this paper, as the others are still subject of research.

The remainder of this paper is structured as follows: Section 2 gives an introduction into the phases of a typical aircraft take-off procedure and provides a short overview on state estimation and prognostics. In Section 3, we present the model-based architecture of our monitoring system and the underlying algorithms for prediction and resampling. Section 4 focuses on our models $M_1, \ldots, M_4$ of increasing accuracy and computational footprint. In Section 5 we show results of simulation experiments. Section 6 discusses related work and Section 7 presents future work and concludes.

2. BACKGROUND

2.1. Take-off Phases and Procedures

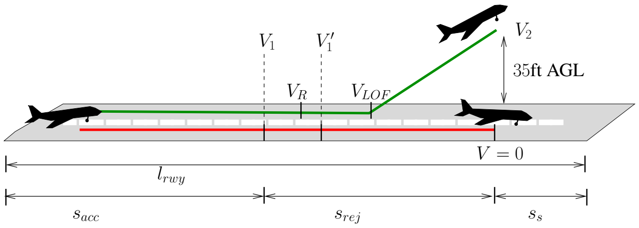

Figure 1. Phases of successful (green) and rejected take-off (red).

Figure 1 shows the typical phases of the take-off. The aircraft starts accelerating and reaches after some time the rotation speed V_R. Then, the aircraft rotates, reaching lift-off speed V_{LOF}, then climbs to a height of 35 ft AGL, where it reaches speed V_2. This successful (accepted) take-off is shown as a green line in Figure 1. If the take-off needs to be rejected, that decision must be made before or at the point in time when speed V_1 has been reached. To compensate pilot reaction time and engine spin-down, that speed is assumed to stay constant for 2 seconds (U.S. National Archives and Records Administration, 1965). Then the aircraft slows down and comes to a stop after a distance s_{rej} before the end of the runway has been reached. Figure 1 (red line) shows a rejected take-off performed at the latest safe point in time.

To ensure a safe take-off, regulations like (U.S. National

Archives and Records Administration, 1965) define necessary runway distances and the selection of different V-speeds. The decision speed has to be lower than the rotation speed, thus, a rejected take-off may only be performed during the ground acceleration phase. Lift-off speed V_{LOF} and V_2 must provide a sufficient margin to stall and minimum control speeds.

Figure 2 shows typical profiles of speed V and runway position s for accepted and rejected take-offs under different wind conditions. Other external influences like tire pressure or (brake) failures can also strongly change these profiles. Without wind, the aircraft reaches V_1 at a point in time when the aircraft is at position s_{acc} and can lift off or come to a standstill before the end of the runway l_{rwy} (Figure 2A). Safe accept (green) or reject (red) is possible. In Figure 2B, a strong headwind slows down the acceleration so much that when V_1 is reached, the aircraft can neither stop safely nor take off safely (blue arrows). In order to avoid such situations, our approach will not base the reject decision on V_1 but rather on the probability distributions of the required remaining runway distance for each of the two possible actions (take-off continue or abort).

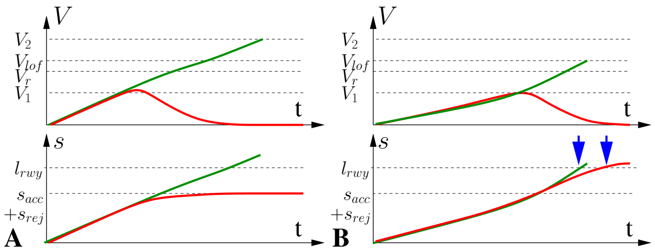

Figure 2. (A) Speed V and position s over time for accepted take-off (green) and rejected take-off (red). The aircraft takes off and comes to a standstill at $s_{acc} + s_{rej}$ well before the end of the runway. (B) Strong headwind slows down acceleration, so that a decision at V_1 comes too late. Runway overruns occur in attempts to accept and reject the take-off (blue arrows).

2.2. State Estimation and Prognostics

In this paper, we formulate the task of estimating the take-off and stopping distance of the aircraft as a prognostics problem, similar to the estimation of RUL (Rest of Useful Life) for system components like a battery (Daigle & Kulkarni, 2013).

In general, the prognostics task of estimating the most likely values of RUL and their probability distribution is performed by three distinct components (Jouin, Gouriveau, Hissel, Péra, & Zerhouni, 2016) as shown in Figure 3: at time k, based upon a sequence of measurements $z_{1,...,k}$ and commands $u_{1,...,k}$, the system is identified by parameters θ_k. A detailed (partially observable) state X can be reconstructed using the system model and the parameters θ_k. This information now can be used to perform the actual prognosis: the estimation of the desired value (e.g., RUL) based upon the current estimated state of the system (X, θ'), given expected command inputs $u_{k+1}, \ldots, u_{k+h}$ for the time until the prognosis horizon h.

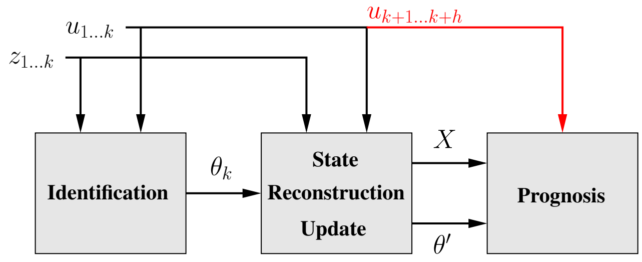

Figure 3. A generic prognostics architecture

In a model-based prognostics framework, typically filters (e.g., Kalman filters or particle filters) are used for identification and state reconstruction, and probabilistic algorithms are used for the proper prognosis (Jouin et al., 2016).

3. ARCHITECTURE OF THE MONITOR

For our take-off monitor, we have designed an architecture (Figure 4) that is based upon a typical prognostics architecture shown in Figure 3 above. Sensors mounted on the aircraft are used to estimate the aircraft's position, speed, and attitude. Typically, available sensors include inertial navigation, air data systems, and GNSS (e.g., GPS) sensors. Often, filtered values are generated by a NAV filter (usually a Kalman filter) in the flight computer. In our case, these filtered values were directly available for processing. Wind speed, wind direction, and turbulence play an important role during take-off. We therefore estimate these values from measurements of ground speed and airspeed, using pre-take-off ground data as priors.

This part of our architecture can be extended easily, for example, by a more accurate method of dynamic estimation of wind (Langelaan, Alley, & Neidhoefer, 2010; Borup, Fossen, & Johansen, 2016; Sharifi & Nobahari, 2016). Additional sensors, like rotation sensors for the wheels, forward-facing cameras, Radar, or Lidar could be used to provide additional data on runway obstacles, state of the runway surface, microbursts, or wake turbulence. Finally, dynamic environmental data that are available from the ground station (e.g., from the Automatic Terminal Information Service (ATIS)) could be used to improve state and model-parameter estimation.

This state estimation and state construction component has the task to dynamically estimate an extended state of the aircraft, which includes current position s (along the runway), kinematic speed V_K, air speed V_A, wind speed and direction V_W, χ_W, thrust T, as well as aircraft failure information, most notably engine and brake failures. A model-based filter (Kalman or particle filter) is used to estimate the current state of the aircraft and obtain the most likely parameters and distributions. This model does not concern the entire aircraft dynamics, but only such parts that are important during the roll phase of take-off. It also can contain failure models with certain analyzed failure modes. At each point in time t the estimated state $\hat{X}^t$ and the parameters $\hat{\theta}^t$ are fed into the prediction/prognostics engine. Based upon expected future commands u (e.g., engage full brake), a probability distribution of the prognosed output is produced. In our case, the variables of interest are the stopping distance s_{rej} for rejected attempts and the distance s_2 to the regulatory screen height.

Runway geometry and safety requirements govern if a take-off or abort is considered safe or not. Here again, these calculations are performed over a distribution of values, yielding our desired probability distributions for required remaining runway distances for take-off continue and take-off abort. Then, a threshold is applied to yield a binary value. For example, an abort should only be attempted, if its probability for success under failures and uncertainties is larger than 0.999. This information is then used to drive the autonomous decision-making component of the aircraft, which also uses aircraft state and diagnosis results to make a decision (accept/reject) according to the given rules and regulations. That decision would then be executed by the on-board automatic take-off control system. In this paper, we will focus only on the prediction part and therefore will not discuss the thresholding, decision-making, and automatic control system component (dashed in Figure 4).

3.1. The Prediction Algorithm

We are using a Monte-Carlo prediction algorithm, which is based upon a particle filter. Since there are no measurements available for the forward prediction, no measurement update on the state occurs. We initialize the N particles (usually $N = 200$) with probabilistic values of the estimated aircraft state X^t by directly resampling its probability distribution. That way, uncertainties of the state estimator can be carried over to the forward prediction.

Then the module performs, for each of the particles, a forward prediction for up to a maximum of N_h steps, each forwarding the prediction time by, for example, 0.1 s. Each step in the forward projection corresponds to the integration of the model M, which is given as a set of differential equations and probabilistic state machines (see Section 4 below). The state machines govern the occurrence of failures, e.g., loss of brake power, which can occur with a certain probability.

The entire execution of state estimation and construction, prediction, as well as monitoring and thresholding is carried out with a fixed rate on the on-board computer, for example, with 2 Hz. We are using a relatively low rate in order not to use too many computational resources. Because the aircraft is moving, the predictions need to be adjusted to compensate for the delay in the calculation. For these short time periods, a simple linear compensation is sufficient.

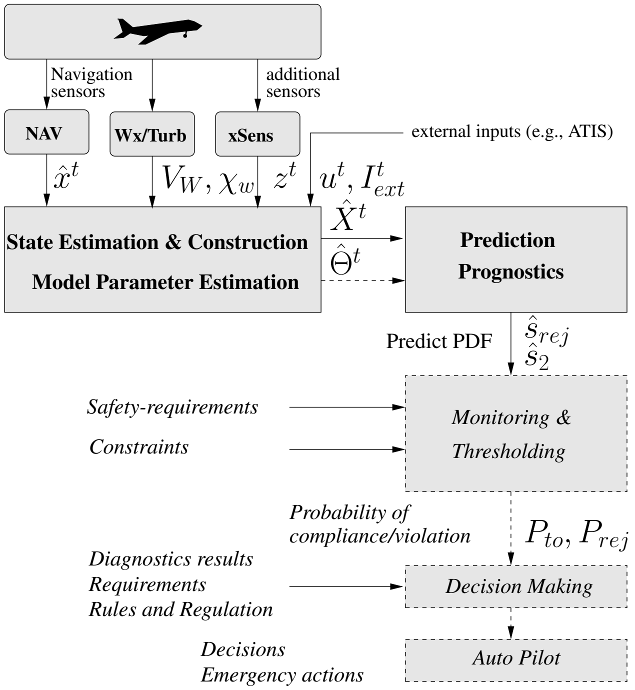

Figure 4. High-level architecture of the take-off monitoring component

In our models, we are using probabilistic state machines. This means that each particle carries a path probability P_p^t denoting the probability of the current system state at time t. All particles are initialized with $P_p^0 = 1$. At each update step during prediction, we use

$$P_p^{t+1} = P_p^t \cdot P_{x_1 \to x_2} \qquad (1)$$

where $P_{x_1 \to x_2}$ is the probability of taking the transition between states x_1 and x_2 in the state machine. For example, we might have $P_{\mathit{on} \to \mathit{off}} = 0.1$ indicating the probability of the brake becoming disengaged is 10%.

3.2. Resampling

During the execution of a particle filter, the probability density can decay, leading to an unfavorable density of particles. In order to avoid such decay, particle filters can use resampling, which is usually triggered when $1/\sum \bar{w}_i^2 < \gamma N$ where $\bar{w}_i = w_i/\sum w$ are the normalized weights and γ a threshold, e.g., $\gamma = 0.5$. A number of different resampling strategies have been proposed in the literature (e.g., systematic resampling, stratified resampling, multinomial resampling). Their aim is to focus the particles toward areas with high probabilities and weed out low-probability particles.

For our prediction task, however, we have opposite goals: instead of focusing on the likely areas of nominal operations, we want to explore areas of low probability that might lead into dangerous regions. For example, a brake failure has a low probability, but it can lead to a dangerous runway overrun. Therefore, that area should be explored with numerous particles.

For our architecture, we developed the following resampling mechanism: whenever the effective sample size with respect to their path probability is lower than a threshold, we trigger resampling. In this paper, we use $\gamma = 0.5$. During this activity, we retire all particles for which their velocity V (as part of the state) is lower than a threshold. In our experiments, we use $2\,\mathrm{kts}$. Then all remaining particles are resampled with their path probabilities as importance.

This rule is motivated by the fact that particles that are retired correspond to safe states of the aircraft. If the velocity is so low, the aircraft will stop in any case within a very short distance, which can be ignored. After resampling we still have the same total number N of active particles in the filter. Thus, after resampling, we obtain a substantially higher particle density in “dangerous” regions. Figure 5 shows this mechanism. Panel A shows the particle density over the distance s_{rej} during one prediction without resampling. Most particles are found around the nominal behavior, where the brake is operational and the aircraft stops after around $200\,\mathrm{m}$. Only very few particles are available to cover the areas of $s_{rej} > 1000\,\mathrm{m}$, where actual runway overruns may occur. In panel B, our resampling mechanism retires particles that are considered safe (i.e., those below the curve). We therefore obtain a suitable particle density in potentially dangerous regions.

For the final calculation of the probability density function, all retired particles must be taken into account. Overall, our algorithm must store $N + N_{ret}$ particles, but only a constant number of N particles are active and must be updated, saving considerable computational resources.

4. PREDICTION MODELS

The architecture, which we have described in the previous section, is model-based and requires a high number of model evaluations. Besides the proposed Monte-Carlo-based particle filter algorithm in general any other prediction algorithm can be applied that accounts for estimation of rare events such as the Particle Filter Prognostics Subset Simulation (PFP-SubSim) proposed by Chiachío, Chiachío, Sankararaman, Goebel, and Andrews (2017). Nevertheless, independent of the applied prognostics algorithm the overall computational cost is highly influenced by the computational cost of the applied model being evaluated at a high frequency. Since the monitoring has to be executed on-board in real-time and with few sensors, there is a trade space of model accuracy, model complexity, and the computational resources that are needed to produce the predictions. The adequate choice of the model for prediction in general is sensitive. Therefore, in this section we propose models for the prediction of automatic take-off at varying fidelity. In this paper, we look at 4 different models of different accuracy, complexity, and ability to model failures.

4.1. Model M_1

This simplistic model only uses the basic physics' relation that the braking distance is proportional to the square of the speed. This model is described by

$$s_{rej} = 0.5\frac{V_K^2}{a} \tag{2}$$

where a is the constant negative acceleration due to friction and braking. In its closed form, the desired s_{rej} can be estimated with minimal computational resources, but the model is not good at handling noise or failures. Any failure can only be expressed by different values of a.

Similarly, the acceleration is modeled with uniform acceleration. The horizontal distance from the current position to position at which the velocity V_2 is reached is given by s_2. To account for the required energy to climb to the screen height h_{ref}, an acceleration to an equivalent velocity V_2' is calculated based on equal energy:

$$\begin{aligned} s_2 &= 0.5at_2^2 + V_K t_2 \\ t_2 &= \tfrac{V_2' - V_K}{a} \\ V_2' &= \sqrt{V_2^2 + 2gh_{ref}} \end{aligned} \tag{3}$$

where g is the gravity constant. In case of available wind measurements, V_2 can be replaced by $V_2 - V_W$ to correct for constant wind. The difference between horizontal distance and slope distance are neglected.

4.2. Model M_2

This model as well as the subsequent ones are described using differential equations; integration is performed during the prediction. All models consist of two parts: a model to calculate the stopping distance s_{rej} in the case the take-off is rejected and a model for the successful take-off and the climb to the screen height. The model $\mathbf{M}_2$ is based on point mass dynamics in a kinematic frame, where the x-axis is aligned with the kinematic velocity.

4.2.1. M_2 for rejected take-off

This simplified model for the deceleration of the UAV in the case the take-off attempt is rejected looks at the basic forces and effects that influence its behavior. In this model, the temporal development of the speed (deceleration) is governed by several force components:

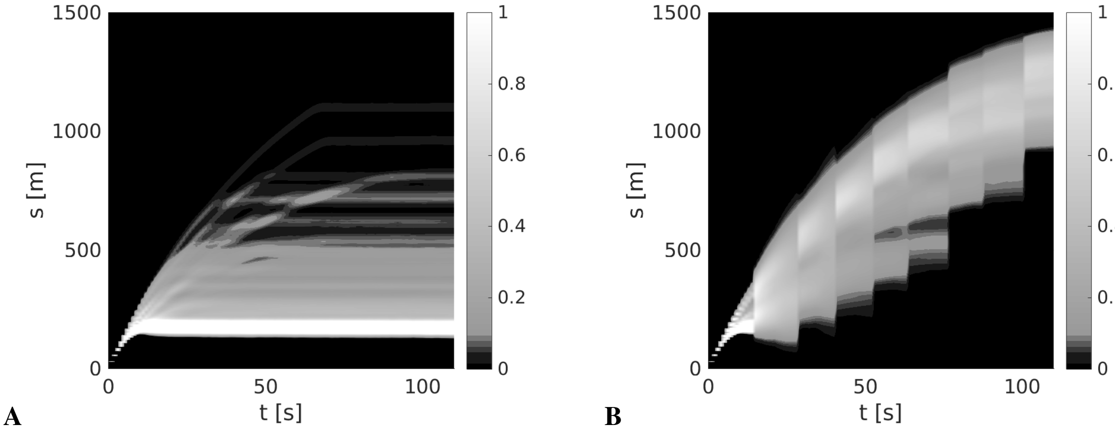

Figure 5. Relative density of active particles for distance s over time. (A) without resampling, (B) with resampling. Total number of active particles $N = 200$.

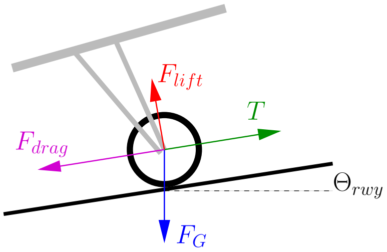

Figure 6. Wheel of aircraft and acting forces

- the (idle) thrust of the engine T. At the beginning of the abort maneuver, the engine is set to idle but may still generate some forward thrust. Also negative thrust might be used, if available. Thrust is assumed to only act in the direction of the kinematic velocity.
- the aerodynamic drag of the UAV F_{drag}. This component depends on the square of the airspeed, the drag coefficient C_D, the reference surface of the aircraft, and the air density ρ. Specific configurations, like settings of flaps or air-brakes can influence this component (not considered here).
- the negative acceleration caused by friction and braking. This component is most prominently influenced by the force on the wheels and the friction of tires and brakes (Figure 6): the force on the wheels is comprised of the difference between the gravitational force F_G (pointing toward the center of the earth) and the upwards lift F_{lift}, caused by the uplift of the wings. Since the upward lift again depends on the square of the speed, braking is less effective at higher speeds.
- an acceleration caused by the tilting angle of the runway. An ascending runway causes the UAV to stop earlier, a descending runway makes it harder for the UAV to stop.

Our model M_2 is described by

$$\begin{aligned}\dot{V}_K = \frac{g}{F_G} \quad & (\\ & T \\ & -\tfrac{1}{2}\rho S C_D V_A^2 \\ & -\mu u_t (F_G \cos\Theta_{rwy} - \tfrac{1}{2}\rho S C_L V_A^2) \\ & -F_G \sin\Theta_{rwy} \\ &) + \eta_a \end{aligned} \tag{4}$$

where $F_G = mg$ is the weight depending on the vehicle mass m, g the gravitational constant, V_A is the aerodynamic velocity, μ is the friction coefficient between tire and runway surface, u_t is the normalized brake setting (e.g., $u_t = 1$), C_L is the lift coefficient, and η_a adds Gaussian noise.

With that model, our desired estimate s_{rej} can be obtained by integration

$$s_{rej} = \int_{t_0}^{t_0+t_{max}} V_K \mathrm{d}t \tag{5}$$

Note that effects due to wind are inherently accounted for by using the aerodynamic velocity V_A to calculate the aerodynamic drag and lift forces.

4.2.2. M$_2$ for accepted Take-off

The accepted successful take-off consists of three distinct phases: the roll phase, the rotation phase, and the initial flight phase. For the roll phase, where the aircraft is accelerating and on the ground, we use the same model as Eq. 4, where the thrust is maximal and the brakes are not engaged in the nominal case. Rolling resistance is accounted for by setting $\mu_r = \mu u_t$, where μ_r is the rolling resistance coefficient. Failure cases, which are explored, concern low thrust (e.g., engine failure), high (head) wind, and blocked brakes.

During the rotation phase, the aircraft remains on the runway until lift-off speed, approximately adhering to the same differential equation as during acceleration. Simplifying, the aircraft is assumed to lift off at V_2 and uses all its excess power to climb (Torenbeek & Wittenberg, 2009) according to

$$\begin{aligned} \dot{V}_K &= \tfrac{T-F_{drag}}{m} - g\sin\gamma_K \overset{!}{=} 0 \\ \dot{h} &= V_K \tfrac{T-F_{drag}}{F_G} \\ \dot{s} &= V_K \cos\gamma_K, \end{aligned} \tag{6}$$

where γ_K is the kinematic flight path angle. The simplifying assumption means that we can set $\dot{V}_K = 0$. Integration can be stopped upon reaching $h = h_{ref}$.

4.3. Model M$_3$

Model M_3 is structurally similar to M_2 but has submodels for the spin-down of the engine and, more importantly, probabilistic models for the runway surface, failures of the brake, and engine failures.

Failures and thrust are represented using probabilistic models. The engine thrust T is governed by the throttle command. In our experiments, we assume that the decision to reject take-off at time t_0 immediately cuts the throttle.[1] After cutting power, the engines' RPM and thrust quickly decrease (Hynes, Hardy, & Sherry, 2007, Fig B-6), which is approximated by an exponential decay, which we model by $\dot{T} = \lambda T$ with a constant $\lambda < 0$.

The effectiveness of the brakes depends on two factors: wheel slippage and brake failures. In our model, we approximate wheel slippage, caused by wet or icy runways using a simple Wiener process that enables and disables the brakes with a certain probability. Figure 7 shows the probabilistic state machine and Figure 8 shows the development of the speed during the brake maneuver for different probabilities and conditions of the runway surface. This state machine is updated at each time step during the prediction.

Brake failures can occur during deceleration of the UAV and are modeled using a Poisson process with a given MTTF (Mean Time To Failure) rate. In contrast to the process shown above (Figure 7), transitions between effective and ineffective brakes usually occur only once during a single brake maneuver. Since the effect of such a brake failure is dramatic, in particular if it occurs early, the resulting distribution of the trajectories becomes bi-furcated and thus far from a Gaussian distribution.

4.4. Model M$_4$

The detailed dynamic model M_4 of the aircraft gear is based upon (Buhl, 2002). The dynamics model takes as input the current state of the aircraft, the gear commands (e.g., gear setting down and locked, brake commands, nose wheel deflection), the surface state of the runway, the pressure of each tire, as well as spring and damping characteristics of the struts. Our aircraft has a nose gear with a nose wheel and a left/right main gear with one wheel each. The wheels are connected via struts to the body of the aircraft. Based upon the flight dynamics model of the entire aircraft, the (vertical) forces on the wheel struts are obtained. A table-based model of the friction coefficients, which depend on the tire pressure, the state of the surface (dry, wet, icy), and the kinematic speed, is used to calculate the friction forces. Together with the information about how much the brakes are engaged, the model finally calculates the horizontal acceleration. Again, a numerical integration yields horizontal speed and stopping distance. This detailed model will also consider failure modes.

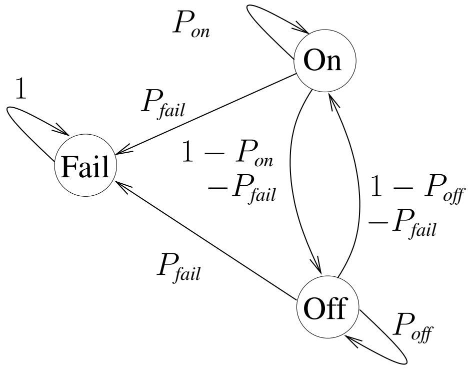

Figure 7. Probabilistic finite state machine to describe the brake status in M_3.

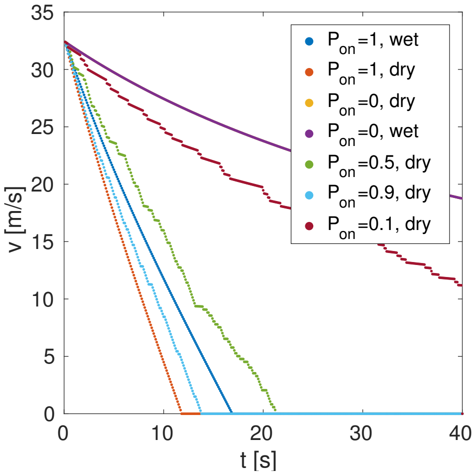

Figure 8. Decrease of velocity depending on the runway surface condition μ and P_{on}.

5. EXPERIMENTS

We illustrate the ability of our take-off monitor to predict probability density functions for the stopping distance s by

[1] Standards for piloted aircraft allow for a 2 seconds delay.

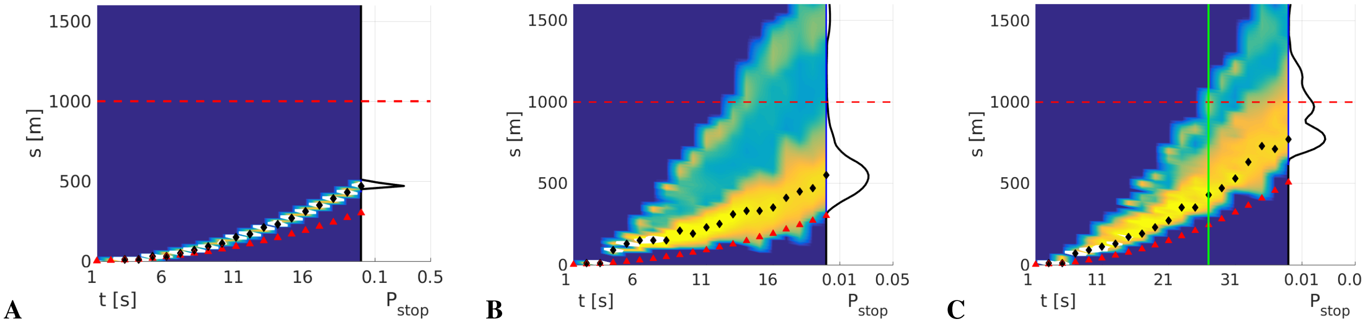

Figure 9. Probability density function for stopping distance $s = s_{acc} + s_{rej}$ over time during the acceleration phase. Red triangles show the position of the aircraft at each time, black diamonds indicate the most likely stopping distance. The red line marks the safe end of the runway $l_{rwy} - s_s$. (A) nominal under dry conditions with low probability of the brake failing, (B) under mixed conditions with less reliable brakes, (C) reduced thrust (engine problem) causes slower acceleration.

running the monitor and prediction algorithm using model $\mathbf{M}_4$ under various conditions of runway surface, brake failures, and reduction of thrust. Sensor noise is assumed to be low and $V_W = 0$. Figure 9 shows the detailed results. In each panel, the red triangles show the current position of the aircraft on the runway at time t of the roll phase. At $t = 0$ the aircraft starts to accelerate at the beginning of the runway. The y-axis indicates the stopping distance $s = s_{acc} + s_{rej}$ from the beginning of the runway. If an abort is to be made at this point, brakes will engage and after some time the aircraft will come to a complete stop. The distance s_{rej} depends on noise, the state of the runway as well as possible brake failures. Our algorithm predicts a probability density function, which is shown as colors; bright colors correspond to a high probability. For the last time step, the corresponding distribution is plotted on the side of the panel.

Figure 9A shows a nominal situation with low noise and uncertainties and a minimal probability for brake failure. The probability density curve stays narrow and the aircraft most likely comes to a standstill at around $500\,\mathrm{m}$ (black diamonds) after a roll phase of 20 seconds. Figure 9B shows a nominal situation with considerable uncertainties in the state of the runway as well as a potential of the brakes failing. The probability distribution is much wider and has some non-zero areas for $s > 1000\,\mathrm{m}$, which can be attributed to an early brake failure. If we assume that the safe usable length of the runway is $1000\,\mathrm{m}$ (red line), the monitoring and decision device must take into account that there is some risk of a runway overrun.

Finally, in Figure 9C, an unexpected reduction of thrust substantially lengthens the time until the speed of $40\,\mathrm{kts}$ would be reached after about 40 seconds. At that time, the density function indicates that with a probability of approximately 40%, a runway overrun would occur. Thus, it would be too late to reject the take-off then. Since the probability functions are predicted at each point in time, this decision can be made much earlier, namely as soon as the probability of an unsuccessful abort reaches a given threshold. In our case, that decision could be made at around $t = 30\,\mathrm{s}$ (green line).

6. RELATED WORK

Real-time take-off monitoring has been investigated at NASA Langley (Srivatsan & Downing, 1987). Using pre-calculated data and a model-based approach, the distance to reach rotation speed and the required stopping distance are calculated along with additional monitoring functions. The system does not predict past V_R and uses nominal performance for prediction.

The automatic detection of a potential runway overrun or runway excursion has been mainly studied for aircraft landing. For example, the Airbus ROPS (Runway Overrun Prevention System) (Villaume & Lagaillarde, 2015) analyzes the current state of the aircraft and warns the pilot if the runway may be too short if conditions are wet, or if the runway is definitely too short. This system, however, does not consider additional failure cases. The algorithms underlying ROPS have not been published. A similar system for predictive runway overrun awareness and alerting is under development at Gulfstream (Konrad, 2018).

In (Bukov & Bykov, 2017), the problem of predicting runway overruns during landing is addressed by minimizing a cost function over a predictive model of the aircraft dynamics. This approach makes it possible to dynamically calculate stopping distances with error bars. Failures or environmental influences are mentioned, but not handled explicitly.

Zhu, Wang, Chen, and Wu (2016) develop model and algorithm to efficiently calculate the speed V_1 for take-off and the stopping distance for landing under different runway conditions (wet and dry) and wind speed. As potential failure modes, only ”one-engine-out” is considered.

There is substantial work on the use of particle filters for prognostics; for an overview see, e.g., (Jouin et al., 2016). Most approaches discussed in the literature aim at the determination of EOL or RUL for a component (e.g., battery, ball bearing), rather than prognostics for an operational procedure. Our resampling schema that considers the safety of the state of a particle in order to determine if it can be retired is related to approaches described in (Orchard, Tang, Goebel, & Vachtsevanos, 2009; Thrun, Langford, & Verma, 2002).

7. CONCLUSIONS AND FUTURE WORK

We have presented an architecture and models for the on-board monitoring of the take-off procedure of an unmanned aircraft. Using a prognostics-based approach, our system uses a particle filter and Monte Carlo techniques to dynamically calculate probability density functions for reaching the screen height and the stopping distance for a rejected take-off under uncertainties, varying runway conditions, and failures. This information can be used by on-board software to decide, in real time, if the take-off must be attempted or if it can be safely rejected if necessary.

Future work will include the validation of this approach using a high-fidelity model and simulator of a realistic UAV. A detailed assessment of the prediction capabilities and computational cost of our models will be conducted to explore possible trade-offs with the aim to show application on real-time hardware. Furthermore, the evaluation of different prediction algorithms such as the aforementioned PFP-SubSim algorithm are planned. Finally, we are planning to study the integration of additional on-board sensors (e.g., tire pressure, wheel tachometer) to improve prediction accuracy. Our monitoring approach can also be extended by optical, Lidar, and Radar sensors that can help to dynamically detect additional failure modes of the UAV, as well as potentially dangerous situations like obstacles on the runway or wake turbulences from preceding aircraft, thus ultimately increasing the safety of automatic take-off.

ACKNOWLEDGMENTS

This research is funded by Bavarian Ministry of Economic Affairs, Energy and Technology.

NOMENCLATURE

C_D	drag coefficient
C_L	lift coefficient
F_G	aircraft weight
g	gravity acceleration
h	height
l_{rwy}	available length of runway
P	probability
S	wing reference area
s_{rej}	stopping distance for rejected t.o.
s_s	safety distance to end of runway
s_2	go distance for accepted take-off
T	engine thrust
t	time
V_A	airspeed
V_{LOF}	lift-off speed
V_K	kinematic speed
V_R	rotation speed
V_W	wind speed
V_1	critical engine failure recognition speed
V_2	take-off safety speed
γ_K	kinematic flight path angle
Θ_{rwy}	runway tilt angle
λ	spin down coefficient for engine
μ	friction coefficient
ρ	air density

REFERENCES

Australian Transportation Safety Board. (2016). *Wasp nest blocks A330 pitot tube: emergency landing at Brisbane* (Tech. Rep. No. ATSB AO-2013-212). Retrieved from https://news.aviation-safety.net/2016/05/06/wasp-nest-blocks-a330-pitot-tube-emergency-landing-at-brisbane/

Borup, K. T., Fossen, T. I., & Johansen, T. A. (2016). A nonlinear model-based wind velocity observer for unmanned aerial vehicles. *IFAC-PapersOnLine*, *49*(18), 276 - 283. (10th IFAC Symposium on Nonlinear Control Systems NOLCOS 2016)

Boston Globe. (2014). *Takeoff is the second most dangerous phase of flight.* Retrieved from https://www.bostonglobe.com/2014/06/03/takeoff-second-most-dangerous-phase-flight/rXNL2TUqnI79h6X359soML/story.html

Buhl, M. (2002). *Simulation eines Flugzeugfahrwerkes* (Thesis). FSD, Technische Universität München.

Bukov, V. N., & Bykov, V. N. (2017). A predictive algorithm for runway overrun protection. *Journal of Computer and Systems Sciences International*, *56*(5), 862–873.

Bureau Enquêtes-Accidents. (2000). *Accident on 25 July 2000 at La Patte d'Oie in Gonesse (95) to the Concorde registered F-BTSC operated by Air France.* Retrieved from http://lessonslearned.faa.gov/AirFrance4590/Concorde_Accident_Report.pdf

Chiachío, M., Chiachío, J., Sankararaman, S., Goebel, K., & Andrews, J. (2017). A new algorithm for prognostics using subset simulation. *Reliability Engineering and System Safety*, *168*, 189–199. doi: 10.1016/j.ress.2017.05.042

Comisión de Investigación de Accidentes e Incidentes de

Aviación Civil. (2008). *Accident involving a McDonnell Douglas DC-9-82 (MD-82) aircraft, registration EC-HFP, operated by Spanair, at Madrid-Barajas Airport, on 20 August 2008* (No. CIAIAC A-032/2008).

Daigle, M., & Kulkarni, C. S. (2013). Electrochemistry-based battery modeling for prognostics. In *PHM2013.*

Directorate General of Air Transport. (1996). *Accident of the Boeing 757-200 aircraft operated by empresa de transporte aereo del Peru s.a.* Retrieved from https://www.skybrary.aero/bookshelf/books/1719.pdf

Federal Avation Administration. (2016). *Pilot's Handbook of Aeronautical Knowledge* (No. FAA-H-8083-25B).

Fortune. (2017). *Here's why planes can't take off when it's too hot.* Retrieved from http://fortune.com/2017/06/20/phoenix-american-airlines-flights-cancelled-heat-wave/

Hinson, D. (1996). *NTSB Safety Recommendation* (No. A-96-15-20). Retrieved from https://www.ntsb.gov/safety/safety-recs/recletters/A96_15_20.pdf

Hughes, K., & Godley, S. (2011). *Take-off performance calculation and entry errors: A global perspective* (No. OCT10/ATSB128). Retrieved from https://www.atsb.gov.au/publications/2009/ar2009052

Hynes, C., Hardy, G., & Sherry, L. (2007). *Synthesis from design requirements of a hybrid system for transport aircraft longitudinal control* (Tech. Rep. No. NASA/TP-2007-213475). NASA.

Jouin, M., Gouriveau, R., Hissel, D., Péra, M.-C., & Zerhouni, N. (2016). Particle filter-based prognostics: Review, discussion and perspectives. *Mechanical Systems and Signal Processing*, *72-73*(Complete), 2-31. doi: 10.1016/j.ymssp.2015.11.008

Konrad, G. (2018). Development of a predictive runway overrun awareness and alerting system. In *AEE18 Avionics & Flight Ops.*

Langelaan, J. W., Alley, N., & Neidhoefer, J. C. (2010). Wind field estimation for small unmanned aerial vehicles. In *AIAA Guidance, Navigation and Control.*

National Transportation Safety Board. (1988). *Airline Accident report: Northwest airlines Detroit Metropolitan Wayne Country Airport Romulus Michigan, August 16, 1987* (No. NTSB/AAR-88/05). Retrieved from https://www.ntsb.gov/investigations/AccidentReports/Reports/AAR8805.pdf

National Transportation Safety Board. (2004). *In-Flight Separation of Vertical Stabilizer American Airlines Flight 587 Airbus Industrie A300-605R, N14053 Belle Harbor, New York November 12, 2001* (No. AAR-04/04). Washington, D.C.. Retrieved from https://www.ntsb.gov/investigations/AccidentReports/Reports/AAR0404.pdf

National Transportation Safety Board. (2015). *Steep Climb and Uncontrolled Descent During Takeoff National Air Cargo, Inc., dba National Airlines Boeing 747 400 BCF, N949CA Bagram, Afghanistan April 29, 2013* (No. AAR-15/01). Washington, D.C.. Retrieved from https://www.ntsb.gov/investigations/AccidentReports/Reports/AAR1501.pdf

Netherlands Aviation Safety Board. (1978). *Final report of the Investigation into the Accident with the Collision of KLM Flight 4805 and Pan American Flight 1736 at Tenerife Airport, Spain on 27 March 1977* (Tech. Rep. No. ICAO Circular 153-AN/56). Retrieved from http://www.project-tenerife.com/nederlands/PDF/finaldutchreport.pdf

Orchard, M. E., Tang, L., Goebel, K., & Vachtsevanos, G. J. (2009). A Novel RSPF Approach to Prediction of High-Risk, Low-Probability Failure Events. In *PHM 2009.*

Sharifi, A., & Nobahari, H. (2016). Multiple model filters applied to wind model estimation for a fixed wing UAV. In *7th International Conference on Mechanical and Aerospace Engineering (ICMAE)* (p. 109-115).

Srivatsan, R., & Downing, D. (1987). *Development of a Takeoff Performance Monitoring System* (Contractor Report No. NASA CR 178255). NASA.

Thrun, S., Langford, J., & Verma, V. (2002). Risk sensitive particle filters. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems 14* (pp. 961–968). MIT Press.

Torenbeek, E., & Wittenberg, H. (2009). *Flight physics: Essentials of aeronautical disciplines and technology, with historical notes.* Springer.

U.S. National Archives and Records Administration. (1965). *Code of federal regulations. title 14 PART 25—airworthiness standards: Transport category airplanes subpart B.*

Villaume, F., & Lagaillarde, T. (2015). Runway overrun prevention system (rops). *Airbus FAST Magazine*(55).

Zhu, Y., Wang, J., Chen, Y., & Wu, Y. (2016). Calculation of takeoff and landing performance under different environments. *International Journal of Modern Physics: Conf. Series*, *42*, 166–174.

BIOGRAPHIES

Dr. Johann Schumann received his PhD (1991) and German habilitation degree (2000) in Computer Science from the Technische Universität München (TUM) in Germany. He is engaged in research on system and software health management, autonomy for UAV, V&V of advanced air traffic control systems, and the automatic generation of reliable code. Dr. Schumann is author of a book on theorem proving in software engineering and has published numerous articles on automated deduction, automatic program generation, V&V of safety-critical systems, and neural network oriented topics. The work reported in this paper was performed while he was at the Institute of Flight System Dynamics (FSD) at the TUM on leave from SGT, Inc., NASA Ames, California.

Alexander W. Zollitsch is a Research Assistant at the Institute of Flight System Dynamics at Technical University of Munich. He received his B.Sc. and M.Sc. degrees in aerospace engineering from TUM in Germany, currently pursuing his PhD. Previous work include research and development of flight control software for a research aircraft, including work on the automatic take-off system. His current work focuses on the development and simulation of flight control software for further aircraft platforms.

Nils Mumm is Head of Flight Dynamics at Amazilia Aerospace GmbH in Munich. He received his diploma in aerospace engineering (Dipl.-Ing.) at Technical University of Munich in 2013 and was employed as Research Assistant at the Institute of Flight System Dynamics at TUM. His research focuses on the development, implementation and flight testing of an automatic landing system including online and offline failure probability estimation. The work reported in this paper was part of his research activities at the Institute of Flight System Dynamics.

Prof. Dr.-Ing. Florian Holzapfel is Institute Chair of the Institute of Flight System Dynamics at Technical University of Munich. After studying aerospace engineering at TUM and working in flight control, he returned to TUM to complete his doctorate. After that, he spent several years working in the industry (IABG, Ottobrunn) before accepting an invitation from TUM in 2007. His four key interests are flight control of manned and unmanned systems, trajectory optimization, sensor technology, data fusion and navigation and modeling, simulation and parameter estimation. He is an Associate Fellow of the AIAA.