VERTICAL ACTIVITY ESTIMATION USING 2D RADAR

Henri Hakl and Willem le Roux

Council for Scientific and Industrial Research, South Africa

Abstract

Understanding airspace activity is essential for airspace control. Being able to detect vertical activity in aircraft allows prediction of aircraft intent, thereby allowing more accurate situation awareness and correspondingly more appropriate airspace control response. The method for qualitative vertical activity estimation as presented is characterised by a very fast response time and requires minimal sensor input. The method relies on the interplay of two opposing motion prediction models. The efficacy of the method is demonstrated in both simulated and real-world data.

Introduction

The development of radar technology in the early decades of the 20th century proved an important factor in the Second World War, where it was used by both Allied and Axis forces to obtain a measure of control over the skies and seas. The principles of 2D radar technology that were developed in that era are still effective in radars today.

A 2D radar emits pulses of energy in a broad, vertical arc. This allows the radar to effectively scan and search for aircraft. The 2D radar however only yields two-dimensional data, indicating the direction and range to the detected target, but offering little indication of its flight behaviour.

The problem of estimating the behaviour of an aircraft is well known with applications in airspace control and threat evaluation.[1,2] In spite of the relevance of the problem, the set of literature available on the topic is relatively small, possibly due to its sensitive nature in military applications.

The existing literature focuses on the problem of accurate altitude determination. The accompanying solutions typically make use of two or more separate radars or similar sensor resources and compute an estimate of aircraft altitude through techniques such as trilateration and triangulation. For examples, refer to Manolakis[1], Rekkas et al.[3] as well as Springarn and Weidemann.[4]

The advantage of these solutions is immediately obvious: the airborne entity can be tracked in three-dimensions. Hence, the determined flight path of the entity can be readily compared with manoeuvre profiles[2] and a good measure can be obtained for its projected future trajectory as well as offering a strong basis for threat evaluation.

This article presents a novel technique to determine relative vertical behaviour of an aircraft using highly limited data. As such, the resulting data is not as comprehensive as that of other solutions, but it can offer valuable information to an air picture operator or automated threat evaluation system by providing immediate estimates on aircraft vertical behaviour from a single 2D radar track.

Problem definition

The problem we wish to address comprises determining whether an aircraft tracked on a single 2D radar is performing a vertical manoeuvre. The difficulty herein lies in the nature of the data provided by 2D radar: tracked entities are specified through the slant range (their distance to the sensor) and their azimuth (the horizontal angle from the observer relative to a reference direction). These two attributes are insufficient to uniquely determine the position of the tracked target and, over time intervals, the plot indicates the straight and turning flight paths of aircraft.

However, it is possible to infer changes in perceived velocity of a tracked target from a 2D radar track. These perceived changes may be due to changes in the target acceleration, changes in target altitude, or a combination of both.

Solution and limits

Determining whether a target is performing a vertical manoeuvre need not necessarily be accomplished using absolute vertical position data. For the purposes of vertical manoeuvre detection, it is sufficient to determine whether *relative* vertical motion is taking place.

Fortunately, the problem of detecting relative vertical motion using a single 2D sensor is considerably more tractable than the problem of determining absolute vertical motion. The solution suggested in this article makes use of two separate motion models that independently estimate the behaviour of the tracked entity, i.e. the relative errors between the predictions of these models and observations from the 2D radar are then used to make probabilistic statements on the current vertical behaviour of the aircraft.

As the intention of the research on which this article is based, is to act as a facilitator to situation awareness and threat evaluation, it was restricted to making predictions based on minimal data. As a 2D radar sensor typically possesses an update rate measured in the order of seconds, it is essential to deliver results within as few sensor updates as possible, otherwise a theoretical threat may become an actual threat before the warning system can issue an alarm.

The solution presented cannot afford considering extensive track history. Only the most recent track updates are considered, allowing the solution to indicate discrepancies within one or two sensor updates.

Solution — Model 1

The first model assumes that the aircraft is flying at a constant altitude. Therefore, any perceived changes reported by a sensor that deviate from the expected flight path must be due to changes in velocity of the target. Since a relatively short period passes between track updates, this model assumes that the estimated acceleration in the current update is equal to the acceleration in the next update, hence the prediction is the projection of the flight path based on the observed acceleration and velocity at the current point in time.

A single update from the 2D radar sensor is sufficient to associate a 2D position with the target. An additional update can be used to estimate the perceived velocity of the target, and finally a third update can be used to estimate the change in velocity of the target. Thus, the following state variables may be associated with a target

$$\mathbf{X}_n \qquad \text{position}$$

$$\mathbf{V}_n = \mathbf{X}_n - \mathbf{X}_{n-1} \qquad \text{velocity}$$

$$\mathbf{A}_n = \mathbf{V}_n - \mathbf{V}_{n-1} \qquad \text{acceleration}$$

These may be used to predict the position of the target at the next update as follows

$$\mathbf{X}_{n+1} = \mathbf{X}_n + \mathbf{V}_n + \mathbf{A}_n$$

The time term has been eliminated in the above equation. It does not influence the result, provided updates occur at a constant rate.

Solution — Model 2

The second model assumes that the estimated speed of the aircraft remains constant between updates, thus all perceived changes reported by a sensor are due to changes in altitude. Due to the spherical nature of 2D radar detection, two distinct changes in altitude need to be considered, thus the second model must make two predictions: one prediction for the case that the aircraft gained altitude, another prediction for the case that the aircraft lost altitude.

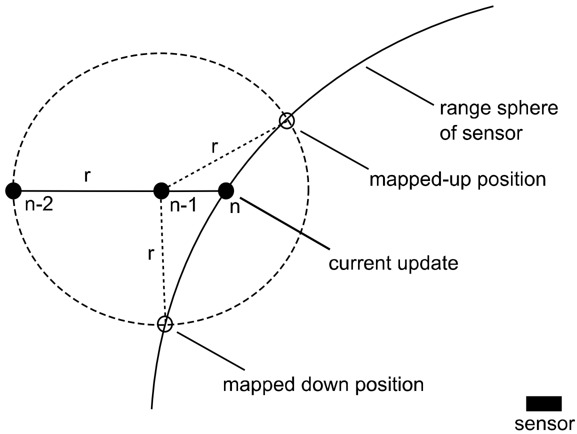

Figure 1: Mapping vertical motion

Model 2 assumes, at each update, that the aircraft has been flying level during the prior update. It offers estimates on the projected flight path based on mapping the velocity of the prior update onto the information of the current update by projecting the possible positions of the aircraft against the radar sphere. **Figure 1** illustrates this mapping in two dimensions for the sake of clarity and without loss of generality.

Note that the height along the sphere at *n* influences the positions for the mapped-up and -down positions. This sensitivity to height is relatively meaningless in practise, as it only significantly impacts the qualitative behaviour when the horizontal distance of the aircraft is closer to the radar than the vertical distance – typically, the horizontal distance well exceeds the vertical distance. Within the context of this text aircraft are (for lack of 3D sensing) assumed to be at a height of 1 000 m above the radar.

In the following, we will only discuss the case of a gain in altitude. The corresponding case of loss of altitude follows by applying the principles of upward mapping and reversing the vertical direction.

Model 2 makes use of the speed, s = |**V**|, of the target. Provided the observed speed at an update is equal to the prior speed, then the flight is assumed to have been level and no additional computation is required. If the current speed exceeds the prior speed, then the aircraft is either accelerating or levelling out from a steeper ascent or descent; again no additional computation is required.

If the speed at the current update is smaller than the speed at the prior update, then the case illustrated in Figure 1 takes effect. We now wish to determine the point that, in Figure 1 is dubbed "mapped-up position".

We solve the vertical mapping through consecutive approximation. A potential position is mapped onto the range sphere of the sensor and the resulting distance from the prior point, indicated as $n-1$ in Figure 1, is compared to the distance between points $n-1$ and $n-2$; when these distances match, an optimal approximation has been found. As appropriate, the approximated altitude is increased or decreased in successively smaller steps to converge to a good estimate.

The process of determining a position on the range sphere is illustrated in Figure 2:

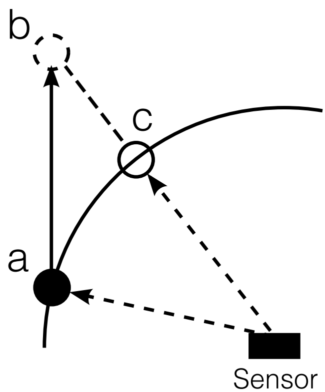

Figure 2: Mapping approximation onto range sphere

The altitude in the base position at *a* is increased to *b*. The size of this increase varies, depending on how well the resultant *c* matches the requirements. This position is then mapped to *c* by observing that the distance from *sensor* to *a* must be the same as the distance from *sensor* to *c*.

Algorithm 1 presents pseudo-code for the computation of the vertical mapping. The following conventions apply: variables and arrays of variables are integer values, floating point values or of type *vector*. The vector type is a structure consisting of three elements, each a floating-point value. The elements are dubbed *x*, *y* and *z*. Additionally, in this document, the *z* element is used to indicate altitude if the vector is used to encode a position.

The following variables are used:

- *n* = the time index, an integer value; *n* is the current discrete time step and $n - 1$ is the prior discrete time step
- *sensor* = of type vector; indicates the location of the sensor
- *state* = an array of type vector, represents the observed state of the aircraft, *state*[*n*] refers to the state of the aircraft at discrete time step *n*
- *step* = a floating point value; indicates the step size for adjustment in approximation, it is also used as the breaking condition in Algorithm 1
- *temp* = of type position; used to hold the current approximation of the aircraft position.

Furthermore, the following three functions are used in the algorithm:

- *normalise* (vector) returns vector — result is a vector in the same direction as the input vector, but of unit length
- *magnitude* (vector) returns float — result is the magnitude of the input vector
- *distance* (vector, vector) returns float — result is the Euclidean distance between the input vectors.
-

▪

Algorithm 1: Vertical mapping

```
Input: n, sensor, state
Output: vector
// initialise
delta_z := 10000
step := 10000
range := distance(state[n], sensor)
  // convergence loop
while step > 0.01 do begin
  step = step / 2
  temp = state[n]
  temp.z = temp.z + delta_z
  // normalise vector from sensor to temp
  temp = normalise(temp - sensor)
  // project vector onto range sphere
  test = sensor + range * test
  if magnitude(test - target) > speed then
    delta_z = delta_z - step
  else
    delta_z = delta_z + step
end

// resulting position estimate
output = temp
```

Since the algorithm loops while *step* ≥ 0.01, and *step* is initialised to a constant value and is halved on each repetition, the algorithm is guaranteed to converge in a finite number of repetitions.

As shown in Figure 3, the model then predicts a position estimate at the next update by assuming the velocity to achieve the projected position remains constant. The corresponding prediction is equal to the projected position to which the velocity from prior position to projected position to the projected position is added.

$$\mathbf{X}_{n+1} = \mathbf{X}_n + \mathbf{V}_n \ .$$

It is possible to define other predictions—such as that the aircraft will steepen its ascent, or level out—but these possibilities are not in general more likely than that the aircraft proceeds without changing climb rate.

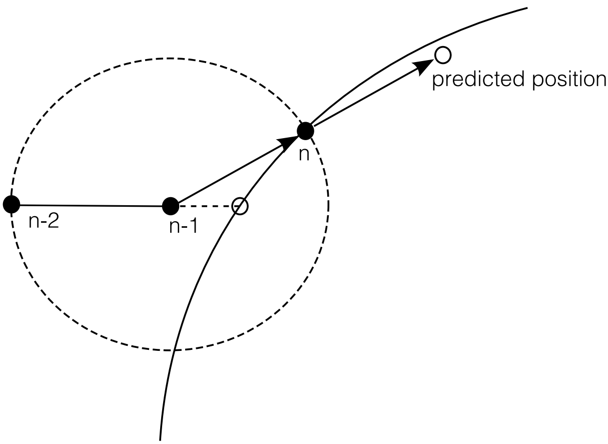

Figure 3: Predicted position

Vertical activity prediction

The predictions of Models 1 and 2 are compared with actual observations during the subsequent sensor update. The three-dimensional prediction is cast into two dimensions representing the slant range and azimuth of the prediction from the sensor. The measure of error for each prediction is then the Euclidean distance between the two-dimensional observed and predicted points. This text assumes that the smaller the observed error is, the more likely the corresponding manoeuvre. Thus, if the prediction error for the acceleration-based model is significantly smaller than the errors observed for the altitude-based model, then it is prudent to assume that an acceleration-based manoeuvre is being performed.

Based on the measured errors, a number of heuristics are used to place an estimate on the vertical activity of the tracked target:

- if the sum of all three prediction errors (representing upward, downward and acceleration-based errors) is less than a threshold value, then the aircraft is considered to be in straight flight. We make use of an error value of 3.0 (this value was obtained through trial and error); in real-world terms, factors such as sensor update rate and variance influence the optimal value of the threshold. As a rule of thumb, the threshold should be in the range of 5% to 10% of typically observed errors while aircraft are performing manoeuvres. Too small a threshold causes the system to be too sensitive to subtle changes in aircraft flight, such as course correction, whereas too large a threshold causes the system to be too unresponsive to changes in aircraft flight.

- if the acceleration error is at least 10% less than the average of upward and downward errors, then the aircraft is considered to be *accelerating*. By requiring the acceleration error to be at least 10% less of the mean vertical error, it is ensured that the accelerative manoeuvre can be unambiguously identified. If this step was not taken, then, in scenarios where all the errors are roughly the same, the system would jump between different manoeuvre estimations with each update. Such chaotic estimations prove more harmful than helpful. By clearly separating the thresholds for acceleration and vertical based motion, the system becomes inoculated to sensor variance.

- finally, if the above two cases do not apply, then, by elimination, the aircraft is considered to be flying a *vertical* manoeuvre. The odds of the manoeuvre being upward or downward is given by the ratio of the corresponding errors:

- $$prob_{up} = \frac{error_{down}}{error_{down} + error_{up}}$$
- $$prob_{down} = \frac{error_{up}}{error_{down} + error_{up}} .$$
- However, it should be noted that the distinction between upward and downward error is typically quite small. This causes the effects of sensor variance to become relatively prominent, thus not a lot of weight should be placed on probabilistic statements with respect to vertical manoeuvres being up or down. Additionally, note that if, for example, an aircraft is flying upward but is in the process of levelling out, then the system can report this as a vertical manoeuvre downward.

A final heuristic is applied over the observed errors: should the sum of errors exceed an upper threshold, then the aircraft is considered to be *energetic*, that is to say that it is flying a complex manoeuvre that exceeds the grasp of the underlying models to accurately predict aircraft behaviour. It is likely that a vertical component forms part of such a manoeuvre.

For the purposes of our experiments, it was determined that a suitable upper threshold for the *energetic* heuristic is a value of 240.0. Note that, similar to the lower threshold used to measure *straight* flight, this value depends on the underlying sensor update frequency and accuracy. A rule of thumb is to place the threshold for energetic flight at twice the typical observed errors during normal manoeuvres.

It should be noted that it is very unusual for the system to estimate an aircraft in an *energetic* state. Care should be taken to interpret the result correctly: if the energetic state is observed for only a single estimate then it is possible that an unusually large aberration occurred in sensing and the energetic estimate can probably be disregarded. On the other hand, if an energetic state is observed regularly and sense-based anomalies can be disregarded as the cause, it is possible that the observed aircraft is actually flying in formation with one or more aircraft and the observed energetic estimate is due to the sensor responding to different aircraft during updates.

Experimental results

Two environments were used to create the experimental results listed below. Initially, an interactive simulator was created that made use of a simple Newtonian physics system to model aircraft behaviour. This simulation was used to investigate the viability of vertical activity estimation using a two-dimensional sensor. Tables 1 and 2 below are derived from this simulator. Table 3 indicates results from real-world pitch and dive data.

The final two columns of each table indicate the desired (actual) manoeuvre and the estimated (output) manoeuvre. In the case of composite manoeuvres (relatively common in Table 3) featuring both acceleration as well as vertical based components, the more appropriate is chosen in the *Actual* column.

Table 1: Acceleration experimental data

Speed delta	error up	down	acc	Actual	Output
0.00	2.00	0.00	0.05	straight	straight
0.00	2.05	0.01	0.00	straight	straight
3.45	1.56	3.45	3.44	acceleration	vertical
19.95	19.93	19.94	16.49	acceleration	acceleration
20.44	20.42	20.42	0.48	acceleration	acceleration
6.94	6.93	6.93	13.48	acceleration	vertical
0.00	0.00	0.00	6.94	straight	vertical
0.00	2.19	2.19	0.00	straight	straight
0.00	2.14	2.14	0.00	straight	straight
0.00	2.09	2.09	0.00	straight	acceleration
-0.77	2.80	2.80	0.77	acceleration	acceleration
-14.64	17.64	17.64	13.64	acceleration	acceleration
-21.21	30.29	30.29	6.57	acceleration	acceleration
-23.14	33.47	33.47	1.93	acceleration	acceleration
-25.71	35.83	35.83	2.56	acceleration	acceleration
-18.25	28.18	28.18	7.46	acceleration	acceleration
-0.58	8.48	8.48	17.66	acceleration	vertical
0.01	1.87	1.87	0.59	straight	acceleration
0.01	1.11	1.11	0.01	straight	straight
0.00	1.12	1.12	0.01	straight	straight

Table 2: Pitch and dive experimental data

Height delta	Speed delta	error up	error down	error acc	Actual	Output
0.00	0.00	2.00	0.00	0.05	straight	straight
0.00	0.00	2.05	0.00	0.00	straight	straight
0.00	0.00	2.01	0.01	0.01	straight	straight
0.00	0.00	1.98	0.00	0.00	straight	straight
9.23	-1.02	2.93	1.02	1.02	vertical	accel
170.57	-71.34	74.39	70.10	70.26	vertical	vertical
193.02	-17.56	35.67	6.62	53.74	vertical	vertical
193.07	1.44	6.24	7.01	18.94	straight	vertical
193.07	1.48	1.48	1.48	0.05	straight	accel
193.07	1.46	1.46	1.46	0.02	straight	straight
193.07	1.43	1.42	1.42	0.03	straight	straight
193.07	1.40	1.40	1.40	0.03	straight	straight
168.36	24.95	24.93	24.93	23.57	vertical	vertical
80.20	49.37	49.33	49.33	24.42	vertical	accel
1.52	6.91	6.91	6.91	42.63	vertical	vertical
-84.07	-25.86	25.84	25.84	32.43	vertical	vertical
-116.63	0.96	0.96	0.96	0.25	straight	straight
-68.38	23.56	23.54	23.54	22.59	vertical	vertical
-7.46	14.27	14.25	14.26	9.30	accel	accel
12.00	2.04	2.03	2.03	12.23	straight	vertical
4.74	0.10	0.10	0.10	2.03	straight	straight

Table 3: Pitch and dive real data

Height delta	Speed delta	error up	error down	error acc	Actual	Output
0.00	-0.02	0.25	0.02	0.02	straight	straight
0.23	-0.01	0.27	0.03	0.01	straight	straight
18.35	-1.55	1.78	1.53	1.54	vertical	vertical
47.44	-4.96	5.92	4.29	3.41	vertical	accel
76.00	-8.03	9.66	6.80	3.07	accel	accel
241.90	-33.57	36.29	31.42	2.93	accel	accel
245.17	-30.89	33.36	28.90	2.68	accel	accel
235.28	-7.96	9.91	6.40	27.04	vertical	vertical
209.33	40.28	40.59	40.02	34.31	vertical	accel
172.77	42.51	42.51	42.51	3.19	accel	accel
128.21	36.09	36.09	36.09	7.36	accel	accel
77.54	27.96	27.96	27.96	9.34	accel	accel
21.43	17.29	17.29	17.29	13.92	accel	accel
-34.30	13.20	13.21	13.21	15.46	vertical	vertical
-60.12	10.16	10.90	9.49	3.94	vertical	accel
-68.23	7.27	8.37	6.20	7.03	vertical	vertical
-87.67	1.91	1.91	1.91	5.76	vertical	vertical
-95.94	6.61	6.61	6.61	5.12	accel	accel
-100.64	8.02	8.02	8.02	1.43	accel	accel
-139.59	8.75	8.75	8.75	0.17	vertical	accel
-144.75	8.31	8.31	8.31	0.93	accel	accel
-151.36	20.96	20.96	20.96	19.54	vertical	vertical
-148.59	46.38	46.38	46.38	26.21	accel	accel
-136.47	44.02	44.01	44.01	5.22	accel	accel
-23.10	45.82	45.82	45.82	7.04	vertical	accel
-19.45	27.83	27.69	27.87	18.19	accel	accel
-19.86	8.36	8.36	8.36	35.95	accel	vertical
-20.40	42.64	42.65	42.65	34.67	accel	accel
-18.74	47.55	47.56	47.56	7.99	accel	accel
-9.54	47.51	47.51	47.51	7.01	accel	accel
-8.06	45.36	45.35	45.35	6.81	accel	accel
-7.92	14.20	14.20	14.20	31.16	vertical	vertical
-6.63	20.59	20.58	20.58	34.76	accel	vertical
-4.63	46.89	46.89	46.89	26.63	accel	accel
-3.43	47.70	47.24	48.04	6.89	accel	accel
-1.42	47.90	47.66	48.04	7.01	accel	accel
-1.24	38.30	38.12	38.40	10.80	accel	accel
-1.11	2.42	2.42	2.42	36.07	straight	vertical
-0.86	34.09	34.09	34.09	36.33	accel	vertical
-0.65	47.93	48.13	47.85	14.81	accel	accel
-0.51	47.95	48.31	47.74	7.00	accel	accel
-0.29	47.96	48.37	47.67	7.00	accel	accel
-0.25	42.87	43.26	42.59	7.70	accel	accel
-0.18	19.68	19.68	19.68	23.23	vertical	vertical
-0.05	6.18	6.18	6.18	13.52	vertical	vertical
0.12	0.87	0.87	0.87	6.08	straight	vertical
0.06	22.14	22.15	22.15	22.18	accel	vertical
-0.11	47.48	47.48	47.48	25.66	accel	accel

The first set of results, in Table 1, indicates the performance of the prediction models in a scenario of linear acceleration and deceleration. Initially, the aircraft flies at a constant velocity, indicated as *straight* flight, followed by a period of acceleration, *straight* flight, deceleration and finally, again *straight* flight.

During changes of phases, the models temporarily indicate the possibility of a vertical manoeuvre taking place but these estimates are promptly replaced with the accurate *acceleration* estimate. The reason for this momentary *faux pas* in estimates is that the acceleration-based model requires one more update than the altitude-based model before settling into a steady estimate.

The word "acceleration" is shortened to "accel" in Table 2 as well as Table 3.

Table 2 documents the behaviour of the models in the case of a pitch and dive manoeuvre. It should be noted that the vertical motion model is able to detect the vertical key points in the manoeuvre, accurately pointing out the initial start of the pitch, the apex of the manoeuvre, and finally the end of the dive.

Additionally, Table 2 illustrates the incapacity of the models to detect a constant gain or loss in altitude—the system reports *straight* flight in these cases. This is not an inherent failure of the models, but instead emphasises the intractability of the problem to infer altitude information from a single 2D track.

Table 3 shows predictions obtained from real-world pitch and dive profile data. The results mimic those of Table 2, with the notable difference that the ascent and descent of the manoeuvre are not flown at a constant velocity, but with a reasonably constant change in velocity. Correspondingly, the system indicates *acceleration* during those phases of the manoeuvre. The models still correctly detect start, apex and end-of-dive segments of the manoeuvre and delimits these with *vertical*.

Note that some portions of Table 3 have been omitted. These sections have been deleted for the sake of brevity. The corresponding entries are all marked as *accelerating* by the system and do not contain further pertinent data.

Limitations

The results presented in the previous section demonstrate the ability of the dual-model system to estimate aircraft vertical activity. Although the output of the system is limited to vertical and acceleration-based manoeuvres, as well as straight flight, the results are reasonably accurate.

However, the system relies on 2D data to be effective. 2D radars have excellent slant range sensitivity due to the nature of electro-magnetic propagation, but some possess poor azimuth sensitivity. Azimuth sensitivity is required to be sub-millirad to be able to provide useful 2D data if the target is 100 km away. As the range decreases, the required sensitivity lessens but remains relatively stringent.

A few words on the effect of turning manoeuvres: given the propensity of the models to act on immediate data, it is possible for aircraft that fly a turning manoeuvre to reduce the accuracy of estimates. This is not a prominent feature in the first model, as the model makes use of three plots to compute predictions, which are sufficient to include turning data in the result. The second model, however, can be influenced by turning manoeuvres. In the case of a single turning manoeuvre during a single-sensor interval, the accuracy degrades, depending on the severity of the turn. In the case of a continuous turn, the perceived speed is relatively constant, thus allowing the model to effectively compute estimates.

If a complex turning manoeuvre is performed, estimates become poor; in this case, the heuristic for *energetic* flight applies. Thus, in such a scenario, the system admits that its accuracy is compromised and that all estimates should be considered best effort attempts.

Conclusion and further work

We have presented a system for the probabilistic estimation of aircraft vertical activity from 2D radar tracks. It relies on the interplay of two competitive motion models – one assuming level flight, the other assuming constant speed. Based on the discrepancies of predictions and actual observations, it is possible to infer vertical activity of aircraft.

The experimental results indicate that the models are capable of recognising changes in aerial activity on aircraft. A vertical activity estimator, as demonstrated in this text, should therefore prove a useful aid in complementing situation awareness tools in scenarios where aerial activity sensing is typically limited to 2D radar tracking. Apart from classic situation awareness scenarios, the system may be valuable in early-warning threat evaluation as well as evaluation of targets beyond visual range.

It is, however, important to emphasise that only a limited degree of information can be inferred from two-dimensional data. In the system described the principle limitation of Model 2 is that it typically only recognises changes in vertical velocity and not vertical velocity itself. In other words, an aircraft flying level is indistinguishable to the model from an aircraft flying at a constantly increasing or decreasing altitude.

Four possible improvements suggest themselves:

- Complex models that capture the motion of aircraft more accurately in different scenarios. This option is viable, but it is not necessarily the most promising: more complex models typically rely on more data to make predictions. This invalidates the assumptions and limitations of vertical activity estimation for early warning and threat evaluation.

- Analysing the predictions of existing models and their corresponding errors more rigorously. In this text, we presented a simple set of heuristic rules that controls the inference of vertical activity from the generated predictions. Stronger statistical evaluation and state-based evaluation may be expected to offer better estimates of aircraft activity.

- By making use of specific sensor equipment, the corresponding vertical activity estimation can be geared to cope with inherent performance and error characteristics of the sensor in question.

- Noting that different flight profiles typically possess distinct signatures through their use of the straight, vertical and acceleration-based activity they exhibit, it should be possible to match potential manoeuvres to the observed estimates. This can help to predict future aircraft manoeuvres and offers an inexpensive approach to perform early threat evaluation.

- Furthermore, it may prove useful to perform additional research on the robustness of the technique described in this article. In real-world scenarios, it may be too strong an assumption that the radar provides regular updates, thus investigating solutions to cope with sporadic failures in updates should provide a layer of security for application of the algorithms in real-world applications.

References

[1] D. E. Manolakis. Efficient solution and performance analysis of 3-d position estimation by trilateration. *IEEE Transactions on Aerospace and Electronic Systems*, volume 32(4), pages 1239–1248, October 1996.

[2] D. E. Manolakis. Aircraft vertical profile prediction based on surveillance data only. *IEE Proceedings on Radar, Sonar and Navigation*, volume 144(5), pages 301–307, October 1997.

[3] C. M. Rekkas, C. C. Lefas and N. J. Krikelis. Improving the accuracy of aircraft absolute altitude estimation using dme measurements. *I. J. of Systems Science*, volume 21(7), pages 1381–1392, 1990.
[4] K. Springarn and H. L. Weidemann. Linear regression filtering and prediction for tracking maneuvering aircraft targets. *IEEE Transactions on Aerospace and Electronic Systems*, volume AES-8(6), pages 800–810, November 1972.