CEAS EuroGNC 2019
"Conference on Guidance, Navigation and Control"
3-5 April 2019 @ Politecnico di Milano, Milan, Italy.

CEAS-GNC-2019-029

Finding all trimmable flight conditions of an over-actuated aircraft using interval analysis

Stephen C. Hungs and Erik-Jan van Kampen

Abstract In this work interval analysis is applied to find the trimpoints of the Innovative Control Effectors aircraft model. At low speed the method is capable of finding interval enclosures of single trim points with a high accuracy. At higher speeds the found accelerations are larger. When looking for a full trim set the method finds continuous bounds on the control effectors for the entire input range in one run. This is a good demonstration of the advantages that interval analysis has over conventional methods that generally can only find one trim point at a time. However, remaining accelerations can be too large to be acceptable as trim conditions. On the other hand the potential that interval analysis has as a trimming method is demonstrated, since continuous bounds on trim sets have been found in a single run.

1 Introduction

The most frequent cause of fatalities in aviation is loss of control in-flight (LOC-I) [10]. It has been shown that LOC-I "*is generally associated with flight outside of the normal flight envelope, with nonlinear influences, and with a significantly diminished capability of the pilot to control the aircraft*" [17]. In that research a bifurcation method is applied to a set of trim points to identify the normal flight envelope. A different approach to finding the safe flight envelope relies on the intersection of the forwards and backwards reachable sets, starting from a safe set [21]. Again a trim set is used as an initial safe set. This shows the importance of knowing the trim set of an aircraft to determine safe operating limits. The reason that trim

Stephen C. Hungs
Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands, e-mail: stephen.hungs@gmail.com

Erik-Jan van Kampen
Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands e-mail: e.vankampen@tudelft.nl

sets can be used as a safe set is that a trim point is a dynamic equilibrium state of the aircraft. This means that the aircraft will maintain that state indefinitely, which can be considered a safe situation.

This research focuses on the innovative control effectors (ICE) model developed by Lockheed Martin. The main purpose of the ICE model was "*to investigate the potential for new and innovative methods for stabilization and control for high performance, low all-aspect signature fighters*" [4]. The ICE model is an interesting model to try trim methods on because it has a total of thirteen control effectors. A consequence of this is that there is redundancy in control power, potentially leading to trim curves or even surfaces instead of points for given flight conditions.

Aircraft trim can have various meanings depending on the context. In this work trim is considered to be all the states of the aircraft model where the accelerations are equal to zero. The trim problem is the task of finding those states, and is generally solved with optimization methods. Interval analysis is a branch and bound optimization method that is capable of computing bounds on all possible solutions in one single run [8], [12]. Although it is a computationally heavy method it has been shown to have potential as an aircraft trim method [14], as it found both the normal side and the back side of the power curve in one run. This makes it a very strong method compared to other methods that have been tried in the past, such as gradient methods [3], sequential quadratic programming [7], bifurcation and continuity analysis [22] and [24] and evolutionary algorithms [25]. All these methods find only one of the possible solutions, based on what the initial conditions were. An analytical solution to the trim problem was proposed in [6], but in order for this to work the model had to be linear in the control inputs. This limits its usability. Another strong feature of interval analysis is that it is very easy to enforce constraints on a problem [8], [12]. Applying constraints is almost unavoidable when dealing with an aircraft model that has thirteen control effectors.

First a short introduction to interval analysis will be given in Section 2. The ICE model is discussed in Section 3. In Section 4 the approach to solving the trim problem will be explained. This is followed by the results in Section 5. This paper ends with conclusions and recommendations in Section 6.

2 Interval Analysis

Interval analysis is a field of mathematics that deals with intervals instead of crisp numbers. Each number is represented by a lower bound a and an upper bound b to make the interval $[x] = [a,b]$. Initially it was developed to put bounds on rounding errors that originate from the limited precision of numbers on digital computers [19], but it was soon realized that interval analysis could be used as an optimization tool as well [18]. If for example 64-bit floating point arithmetic is considered in accordance with IEEE Standard 754 [11], the smallest number greater than 1 is $1 + z$ where $z = 2.220446049250313 \cdot 10^{-16}$. With normal computation the result of $1 + \frac{z}{3}$ would be simply 1, but with interval arithmetic it would be $[1, 1 + z]$. Neither

the lower or the upper bound are the exact solution, but the exact solution is certain to be contained by the interval solution. Normal math operations can be done on intervals, but this requires that the rules are extended. If any operator is expressed as $\bullet$ then [8]:

$$[x] \bullet [y] = \{x \bullet y | x \in [x], y \in [y]\} \tag{1}$$

For example, refer to Equations 2 and 3 to see what this means for addition and multiplication.

$$[a,b] + [c,d] = [a+c, b+d] \tag{2}$$

$$[a,b] \cdot [c,d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)] \tag{3}$$

When intervals are used as a function argument the function value will also be an interval. Care must be taken that the function has as few occurrences as the interval variable as possible. As an example take Equations 4 to 6. In crisp arithmetic these expressions would be equivalent. If now the interval $x = [-1, 3]$ is used instead they evaluate to 3 different intervals. This problem is known as interval dependence. It is certain that all possible values that the function assumes on the argument interval are contained by the resulting interval. If the function is expressed without taking dependence into account there will be overestimation in the function evaluation. When every variable is in the function expression only once the interval will be tight, meaning that the lower bound equals the minimum value of the function on the domain and the upper bound the maximum value. This is the case in Equation 6.

$$[x]^2 - [x] = [-3, 10] \tag{4}$$

$$[x]([x] - 1) = [-6, 6] \tag{5}$$

$$\left([x] - \frac{1}{2}\right)^2 - \frac{1}{4} = [-0.25, 6] \tag{6}$$

In [8] approaches to optimization algorithms based on interval analysis are explained. Used in such a way interval analysis is a very powerful tool, because it finds all possible solutions in one single run. This feature is shown to be extremely useful in for example parameter estimation, robust control and robotics [12] or for nonlinear aircraft trim, optimization of human perception modeling[15], orbital dynamics[2] and spacecraft rendezvouz and docking [27]. It has also been used to evaluate the effect of uncertainties for re-entry flight clearance [13]. Interval analysis does have downsides, such as the computational load associated with it. By definition it will not find the exact solutions, but only bounds on the solutions. However, for practical purposes the mid point of the interval can be considered a crisp solution if the bounds are tight enough. When considering intervals as inputs to an optimization problem they are generally called boxes. In multidimensional problems a box

consists of a vector of intervals. All calculations in this work are done with Matlab in combination with the Intlab toolbox [23]. The implementation of the optimization algorithm will be discussed later in Section 4.

3 Model

The ICE model was developed by Lockheed Martin as an effort to reduce the signature of fighter aircraft, while still being highly agile. To this extend the vertical tail has been removed and instead three promising alternative concepts for yaw control have been applied to the model [4], [5]. The three concepts are all moving wing tips (AMT), spoiler-slot deflectors (SSD), and differential leading edge flaps (DLEF). Combined with elevons, a pitch flap (PF), pitch thrust vectoring (PTV) and yaw thrust vectoring (YTV) this aircraft has a total of thirteen control effectors. There are two versions of it, a delta wing shaped model for the air force (configuration 101) and a diamond wing with canard layout for carrier based use for the navy (configuration 201). An impression of the 101 model is given in Figure 1. Wind tunnel tests on the 101 model have shown nonlinear characteristics of the aerodynamics and in some cases even discontinuities [1].
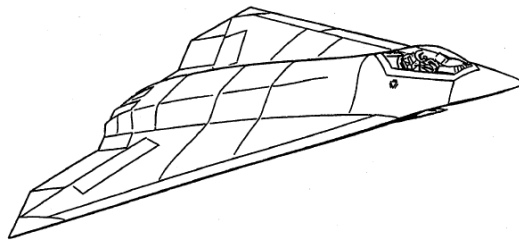
Fig. 1 Impression of what the 101-series ICE model looks like [4]

The model of the 101 configuration of ICE is written as a Simulink model. Unfortunately the interval toolbox Intlab is not compatible with Simulink, and because of that the model must be converted to a Matlab code equivalent. This has been taken as an opportunity to write the model specifically with trim in mind. All states in the original model that are used as feedback states are now linked directly to the input. Based on that expanded input set the aerodynamic coefficients for forces and moments can be calculated. These are then used to determine the aerodynamic body forces and moments. Because the converted model only will be used for trim the thrust setting can be coupled to the body forces. It simply needs to be equal to the body force in x-direction. The aerodynamic model is not symmetrical. This causes problems if straight and level flight conditions are assumed for trim. In those cases the pitch angle equals the angle of attack and the roll angle should be zero degrees. When zero degree roll angles are used the asymmetry will result in acceleration in-

tervals of the model not containing zeros. As a consequence of that no trim solutions will be found. To counter this the aerodynamic side force can be used as a measure for how large a roll induced gravitational side force must be. In this way roll angles can be found that do result in a trimmed state. Because the mass and the inertia of the model are constant the accelerations can easily be calculated from the forces and moments. All flight states required to do the calculations are used as an input or depend on the inputs directly. This removes the need to do integration in the model. This is a good thing because integration generally produces very wide intervals [20], [16].

The conversion has changed the list of inputs to the model. In the original model only the control deflections and the thrust setting could be controlled directly, while some flight parameters such as initial altitude and velocity could be set before a simulation. The states of the model were all calculated internally, based on integration from the initial values. The outputs were the states, with the speeds expressed in earth velocities, body velocities, and α, β, Mach representation. Instead of Mach number the airspeed can be used, as that is calculated too. Finally the dynamic pressure is given as an output as well.

In the converted model the thrust setting is internalized and can thus be removed from the list of inputs. On the other hand a lot more control on the flight state is desired, so that trim for specific scenarios can be found. Because of that the angle of attack, angle of sideslip and the Mach number are usable as inputs now. Table 1 gives an overview of the inputs and their allowable ranges, as well as the sign conventions used in this work. The choice of flight condition determines how the Euler angles need to be linked to the inputs. This means they can be internalized as well. The flight condition also dictates the values for the body rates. Since trim is the goal, it is known that the accelerations must be zero, so constant values can be used for the rates. The same holds for the altitude. The position on earth is arbitrary in this model. This means that only the accelerations are missing. These are the outputs. In order to reconstruct the power curve thrust is required as an output as well.

Table 1 Allowable ranges and sign conventions for the inputs to the converted model

Input	Lower limit	Upper limit	Sign convention positive
α	$-2.5°$	$42.5°$	Nose up
β	$-30°$	$30°$	Nose right
Mach number	0.3	2.16	Positive body x-axis
AMT	$0°$	$60°$	Trailing edge down
DLEF inboard	$0°$	$40°$	Leading edge down
DLEF outboard	$-40°$	$40°$	Leading edge down
Elevons	$-30°$	$30°$	Trailing edge down
PF	$-30°$	$30°$	Trailing edge down
SSD	$0°$	$60°$	Trailing edge up
PTV	$-15°$	$15°$	Nozzle down
YTV	$-15°$	$15°$	Nozzle left

The aerodynamic model of ICE is stored in 108 data tables. Of those 108, 96 use linear interpolation and twelve use cubic interpolation. Linear interpolation with intervals can be done by evaluating a finite number of points and taking the maximum and minimum value at those points as upper and lower bounds [14]. These points are every data point contained by the box, all points on the boundary of the box intersecting a line between data points and all corners of the box. Figure 2 shows this.
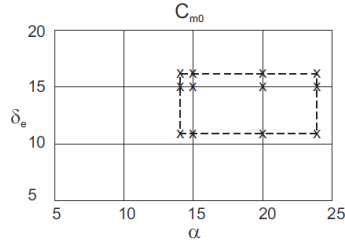
Fig. 2 All points that need to be considered during linear interpolation in an interval box [14]

The process for cubic interpolation is a bit more elaborate. The explanation of how and why it works is as follows. The second derivative of a cubic spline between two data point is at most a linear function. Therefore there can be at most one zero crossing for the second derivative, hence there can be at most one inflection point between two data points. If the tangent is taken at the data points and extended to the other side of the section an inclusion interval can be determined. When there are no crossings of the tangent and the curve the endpoints of the tangent are measures for the inclusion interval of the curve. If the curve intersects the tangent line before it reaches the other side of the section this means that there has to be an inflection point between the data point at which the tangent was taken and the intersection of the tangent and the curve. This means that the curve can only continue in the direction that it has crossed the tangent. If the tangent is taken from the left data point it runs from point the data point A to the mapped point B. The tangent from the right data point C runs to the mapped point D. It is certain that the curve is included in the interval $[min(A,B,C,D), max(A,B,C,D)]$. The left plot in Figure 3 shows this procedure. When this is done over all the sections defined by data points and domain boundaries an inclusion interval for the entire domain can be found by taking the maximum and minimum found values, as is illustrated in the right plot of Figure 3. The curves used to make the figures are based on a cubic spline created on a set of ten random numbers in the range $[0,1]$ randomly distributed on the domain $[0,5]$. Because it uses the gradient to find the inclusion interval this interpolation method is called the cubic gradient inclusion interval. The current implementation used in the model determines the gradient numerically by taking the symmetric derivative around the data points.
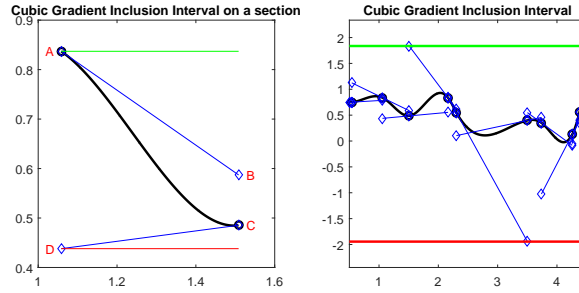
Fig. 3 Illustration of the workings of the cubic gradient inclusion interval

After the conversion the two versions of the model are compared. This is to validate the new model. To do this a set of 10,000 randomly generated input combinations was used. All inputs were within the ranges indicated in Table 1. Based on each combination of inputs both models are run in a block by block fashion, to identify mismatches. During this process the largest differences were six orders of magnitude smaller than the actual values, while most of them were in the order of 64-bit double machine precision. The larger differences generally occurred at large aerodynamic angles, when the control surface deflections are large and disordered.

4 Methodology

The process of finding a trim set starts with defining the flight condition and what control effectors can be used to steer the aircraft. Control effectors that are not allowed to move get assigned a fixed value, as do the flight states that do not change. The rest of the variables are independent variables. These get assigned an interval in which they are allowed to change. The altitude and the rates are in the current version not available for use as independent variables. This leaves α, β, and M as potential independent variables. Every control effector can be used as an independent variable as well. The sixteen element vector consisting of α, β, M and the control effector deflections is defined as a box. The input values set the search domain and make up the first box. The trim routine itself is a zero finding algorithm that is applied to the ICE model. An interval zero finding algorithm works as follows. To start the first box is evaluated. If there are zeros in this box there could be actual zeros in the function that is being optimized. If this is the case the initial box is split into subboxes. The function is evaluated again over these subboxes. If there are no zeros in a subbox it is certain that there are no zeros in the function for that part of the domain so the code can discard that box. The boxes that do have zeros can be split again, and the above process is repeated. This process is based on box consistency as explained in [8] and [26]. In box consistency the domain is split up in smaller pieces, for which it is ensured that the function values are consis-

tent with the solution of the optimization problem. To visualize this process refer to Figure 4. Here a one dimensional function is evaluated over 20 subboxes. Of those boxes there are only six that might contain zeros. For the other fourteen it is certain that they do not contain zeros, and are therefore removed for further evaluation. The six remaining boxes are each split into five smaller boxes and after evaluation eight boxes remain. If this process is repeated several times the bounds on the location of the zeros can become very tight. For the trim algorithm each split will happen in a different box direction, until all box directions have been split once. Then the code splits the remaining boxes again in the first direction and loops through all directions until all are split once more. This goes on until the stopping criteria are met or until there are no boxes remaining. The stopping criteria are based on practical limitations and limitations of the human vestibular system. The practical limitations come from state measuring accuracy and control deflection precision. There is no point in knowing how much the accelerations vary over a box that spans $0.00001°$ in angle of attack, because the angle of attack sensor at the actual aircraft is not able to measure that accurate. If boxes are split so often that they become smaller than $0.01°$ in angle of attack or angle of sideslip and smaller than 0.005 in Mach number the splitting is terminated in that direction. As will be explained below, it is possible to do trim runs focused on finding point enclosures and runs aiming to find the entire trim set on a domain. It can be expected that point enclosures do not require as much boxes as trim set enclosures. Therefore runs aimed at trim points get a sharper control deflector box diameter stopping criterion of $0.01°$. Box splitting in trim set runs terminate if the control deflector box diameter is smaller than $0.1°$. The limiting factor from the human vestibular system is the limit to what it can actually detect. From [9] it is found that the smallest accelerations that can be detected are $0.02 m/s^2$ for linear accelerations and $0.05°/s^2$ for rotational accelerations. These values indicate when the human sensory system can no longer detect accelerations in a controlled simulator environment. If either of the stopping criteria are met the code terminates. If the box precision is reached the found results should first be evaluated before going further, due to the computationally demanding nature of interval analysis. If the number of boxes starts to get excessive there it can become a very slow process that hardly gives any improvements. If on the other hand the sensory limits are met before the box precision is achieved this means that the entire box can be used as trim input. No further refinement is necessary in that case. An overview of the process is given in Figure 5.

With thirteen control effectors the dimensionality of the ICE model is much too large to solve in a single run with the current state of technology. The flight condition and limiting the number of active control effectors can help in the matter. Some of the more useful flight conditions are explained in [3]. From this it is known that choosing for steady straight and horizontal flight pins down quite some variables. The altitude is simply a constraint, while the rates are fixed at zero. It is also known that the pitch angle θ must be equal to the angle of attack α. As explained earlier there is an asymmetry in the ICE model that requires a nonzero roll angle ϕ in order to achieve a trimmed condition. Because of this it is not sufficient to simply use the elevons for pitch control and trim the aircraft like that. From [4] and [5] it is known
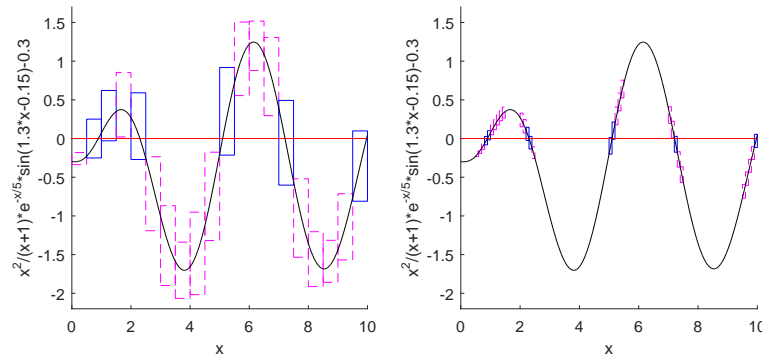
Fig. 4 Visualization of an interval analysis zero finding algorithm on an example function

that the AMTs perform reasonably constant up until an angle of attack of $40°$. At even higher angles the control power reduces, but it is still sufficient to keep full body control. AMTs can provide yaw control as well as roll control. Yaw control is linear with control deflection for a large range of angles of attack. Roll control is not linear and its effect can even change from adverse at low angles of attack to favorable at high angles of attack. If this is combined with elevons the model has full roll, pitch, and yaw control. To keep the trim problem small a simple test is performed to see if trim can be achieved by using only one AMT. It turns out that only using the left AMT is sufficient. The initial interval for each control effector will be the full allowable range, to ensure that all possible trim points will be found. The above has explained the choices for determining what control effectors to use. Furthermore the asymmetry in the model is resolved by just using roll. For steady straight and horizontal flight the sideslip angle can be set to zero. If both Mach number and angle of attack are used as free variables every combination of Mach number and α that gives a trimmed state will be found. This approach will also produce bounds on the free control effectors. Before attempting such a run it must be ensured that the algorithm is working properly. To do this Mach number is used as a constraint. By checking the solutions at multiple Mach numbers trends can be determined. For example, the shape of the power curve should comply with the shapes of the power curves of other aircraft. This approach also enables comparison between the interval trim method and the conventional trim method that was provided with the ICE model. That method uses the Matlab function `fmincon` with the active-set algorithm. Comparing the results found with both methods gives an indication if the interval method converges to the right solutions. After confirming that the interval methods find the right solutions a run that finds every trim combination of Mach number and angle of attack can be executed. For both variables their full allowable range is used as the input to the solver.
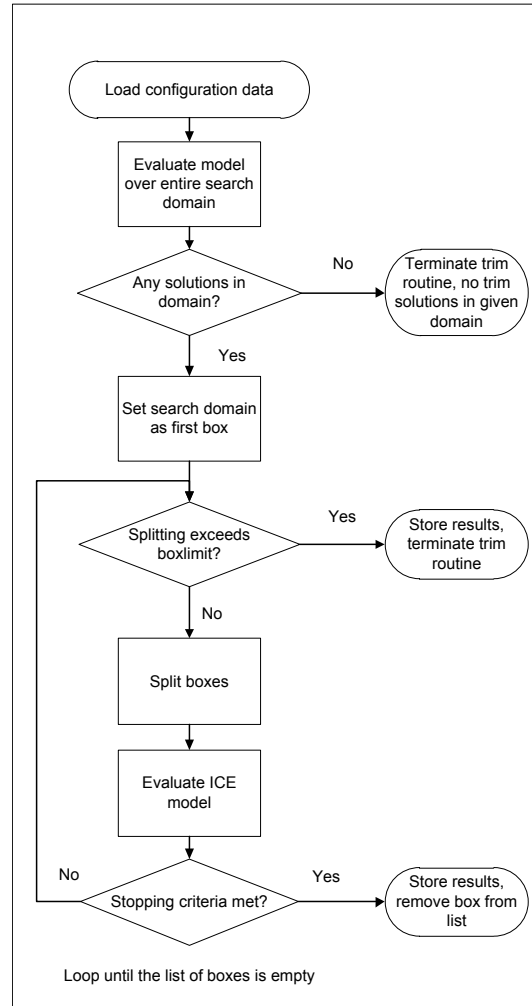
Fig. 5 Flowchart showing the process used for trimming the ICE model

5 Results

As explained above, it is important to ensure that the code actually finds the right solutions. To do that the first set of results is aimed at point solutions. Fifteen sample speeds are used to find these points. These are taken with equal spacing from the allowable Mach range, spanning the entire range from 0.3 to 2.16. Two different altitudes are used as well, one set at 10,000 ft and one at 20,000 ft. In total that means 30 runs are required to get this initial set of results. The code produces bounds on the true solutions, so the outcome is the list of boxes of input combinations that would result in interval accelerations that do contain zeros. To keep the plots of

these results clear the mean value of the midpoints of the list of remaining boxes is taken. The hull of these lists is at most $0.02°$ wide in angle of attack, $0.12°$ wide in the control deflections and a maximum of 120 lbf wide in terms of thrust setting. Relative to the absolute values the hulls are very small, so the difference between a crisp point or the hull of the boxes would not be visible. The plots can be found in Figure 6. Individual plots are made for all the free variables of the zero finding problem, as well as the associated thrusts.

The plot of the trimmed thrust against the Mach number is widely known as the power curve. This can be used as a indication if the code converges to the right solutions, since every aircraft has one and the trends in power curves are generally the same. For each aircraft there is a velocity that requires a minimum amount of thrust. This is because the combination of induced and parasitic drag is the lowest at that point. Fly faster and the parasitic drag will increase, requiring more thrust. Flying slower than that point increases the induced drag, again increasing the required thrust. This region where more thrust is needed to fly slower is known as the backside of the power curve. The trends between the points in the first plot in Figure 6 show this behavior as well. The impact of different altitudes is clear as well. Because the atmosphere at lower altitude is denser the power curve is steeper. The point of lowest thrust also occurs at a lower speed. This matches the trends seen in the plot. What also can be seen in the plot is that there are two points missing in the curve for 10,000 ft. A hint of what happens there can be seen in the plot for the angle of attack. The solution at Mach 1.3628 appears to be out of line with the rest of the solutions in that it finds a lower angle of attack compared to the ones at the higher Mach numbers. When allowing the code to exceed the allowable range in angle of attack as indicated in Table 1 to highly negative angles it does find a solutions. These solution require angles of attack of approximately $-15°$. When using the trim procedure that came with the model similar results are found. This indicates that there is nothing wrong with the method, but rather that there is some odd behavior in the model in that particular flight regime. The trend for the angle of attack curve at 20,000 ft shows, as expected, high angles at low airspeed with the curve flattening off to an almost constant angle at higher speeds.

When looking the required deflections at the trimmed points there are three parts to each curve: the part on the backside of the power curve, the part on the normal side of the power curve up to Mach 1.5, and the part that covers the higher sonic regime. In the first domain it is very notable that the required LAMT deflection is larger than it needs to be for the other parts of the speed domains. The elevon deflections at 20,000 ft show the expected reversal of command effect at the slowest speeds, which cannot be seen at 10,000 ft. The section that covers the part up to Mach 1.5 has only small LAMT deflections, with maximum deflections of $1.5°$, while the higher sonic part of the domain sees even smaller LAMT deflections up to $0.5°$. This indicates that the faster the aircraft flies, the more the LAMT acts as a trim tab. This is as expected, because in this scenario the LAMT is used just to balance the asymmetry of the model. It is also observed that the difference between the left and right elevon is smaller if the LAMT deflection is smaller. It can be assumed that if the model was symmetric, trim would be possible with just the elevons. Both

elevons could use the same value in that case, leading to a very small optimization problem with only angle of attack and elevon deflection as free variables.
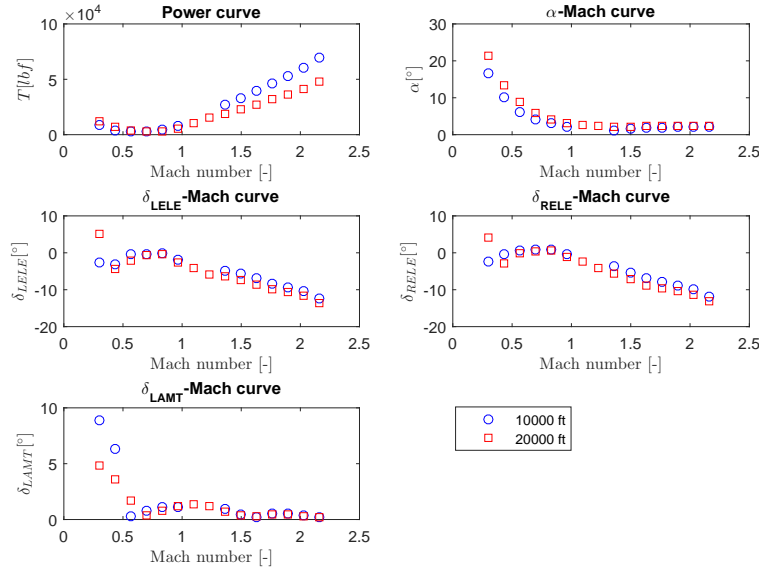
Fig. 6 Overview of the found trim points when using fixed Mach numbers

To compare the interval method with the more conventional trim method that came with the ICE model a similar set of runs is done with the standard trimming routine. At two different altitudes fifteen different velocities have been used to find a trim point. The found points are shown in Figure 7. For reference the points found with the interval method are given as well. As mentioned earlier the `fmincon` routine also finds odd results for Mach numbers 1.0971 and 1.2300 at 10,000 ft. Again, these points are not plotted, since this issue is rooted in the model itself. When comparing the two sets of solutions one must keep in mind that the interval method only works on the converted model which uses intervals. This in turn means that the converted model cannot be used with the conventional trim method. A pure comparison between the methods is therefore not possible. But as mentioned in Section 3 the differences between the two versions of the model have been shown to be very small. This is notable at subsonic speeds, where the two methods find almost identical solutions. Going to higher speeds there are only minor differences in terms of thrust at 20,000 ft in the area where 10,000 ft runs do not find any solutions at all and at the highest Mach number that has been evaluated. In angle of attack there are no significant differences. In the control deflections there are bigger differences that can be as large as 3°. The only logical explanation is that these differences are caused by the

combination of each model with each respective model. The differences between the models have been shown to be very small, so only part of the mismatch is caused by the cumulative mismatches of the individual parts of the models. The rest should therefore originate from the different features of each method. The interval method simply tries to bound zeros in the model accelerations, irrespective of the control surface deflections, while the `fmincon` code minimizes the sum of the squares of all control surface deflections. That the accelerations should be zero is only used as a constraint. This makes in logical that the absolute deflections with the `fmincon` method are smaller than the ones found with interval analysis. It must be noted that the `fmincon` code was not able to find a solution for Mach 1.4957 at 10,000 ft and for Mach 1.6286 at both altitudes with the used starting values of all control deflections equal to zero with an angle of attack of $5°$. However, by changing both initial elevon deflections to $-20°$ a solution is found for the 20,000 ft case. This point is indicated with a different marker in Figure 7. That the initial values can cause the method to not find a solution is a clear downside of the method. Interval analysis does not have this issue, as the only required input is the search domain.
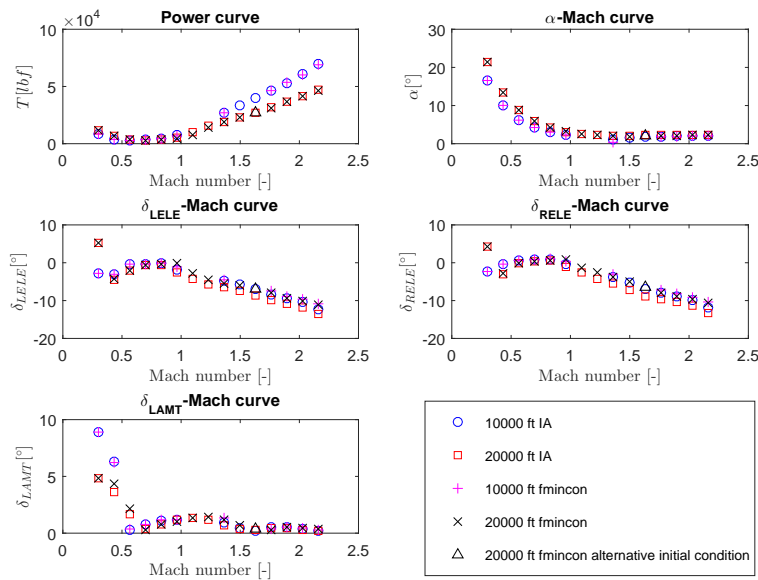

Fig. 7 Differences in the found results when using interval analysis and fmincon

The point solutions in Figure 6 have indicated that the trim routine is working as intended. It can now be expanded to also use Mach number as an independent variable. Two runs are done, one to cover each altitude that was used to get the previous results. The outcome of such a run is a set of continuous bounds on the free

variables. These are shown in Figure 8. In terms of thrust the bounds are less than 500 lbf apart for the most of the domain, only at high Mach numbers where high thrust settings are needed the difference goes up to a maximum of approximately 900 lbf. The upper and lower bounds on angle of attack are rarely more than $0.2°$ apart, while the control deflectors bounds are on average $0.37°$ wide, exceeding $1.0°$ only once. The points found previously are all contained by the bounds found in these runs. As with the point solutions, there is a domain where the model shows some odd behavior and requires highly negative angles of attack for trim. This can best be seen in the α-Mach curve, where the bounds on the angle of attack suddenly drop to the lower limit of the validated range, only to gradually recover at higher velocities. In these sections the bounds on thrust and angle of attack are further apart than the limits stated before. In the plots there are several kinks, such as the one at Mach 1.2 in the thrust plot and the ones at Mach 0.9 and Mach 0.95 in the elevon plots. These kinks happen at data points in the tables. Because most data tables interpolate linearly there are discontinuities in the derivatives of the data values. This carries through to the found solutions. Except for the gap in the 10,000 ft run the power curves look as they are supposed to look. The effect of altitude can again clearly be seen with the 20,000 ft power curve less steep and moved to the right. The higher density of the atmosphere at the lower altitude means that less angle of attack is needed to generate enough lift, which can also be clearly seen. The continuous bounds on the control effectors show that there is a lot more variation going on in the required deflections on the backside of the power curve than the point solutions initially suggested. The two slowest solutions on the LAMT plot in Figure 6 might have suggested that more LAMT deflection is needed at lower altitudes, but as it turns out the required deflections have a similar profile, with the one for greater altitude shifted to the right the same amount as the power curve is shifted. The fact that the point solutions of the 10,000 ft run were close to local maximums and the points for the 20,000 ft run close to local minimums gave an unclear picture of the true deflection profiles. Both elevon profiles show similar right shifting with increasing altitude. As with the point solutions, the difference between the left and right elevon deflection is smaller when the LAMT deflection is smaller. This enforces the hypothesis that the model could be trimmed with only elevons if it was symmetric.

In addition to the five plots that are also there in the point solution figure, there are two plots showing information about the branch and bound algorithm as it progressed to the solution. The one on the left gives the number of boxes that could potentially contain trim points after each step. The computer that was used during these calculations could evaluate an average of 160 boxes per minute. Because the splitting algorithm divides the remaining boxes into three subboxes the cumulative number of boxes for all iterations but the last must be multiplied by three to get the number of boxes that have been evaluated. What this translates to is that the 10,000 ft run took about 6 days and 10 hour to converge. The run at 20,000 ft used more boxes, which lead to a total calculation time of approximately 7 days and 8 hours. The runs enclosing trim points required significantly fewer boxes and because of that a lot less time too. For all 15 runs at 10,000 ft only 3 hours and 22 minutes were
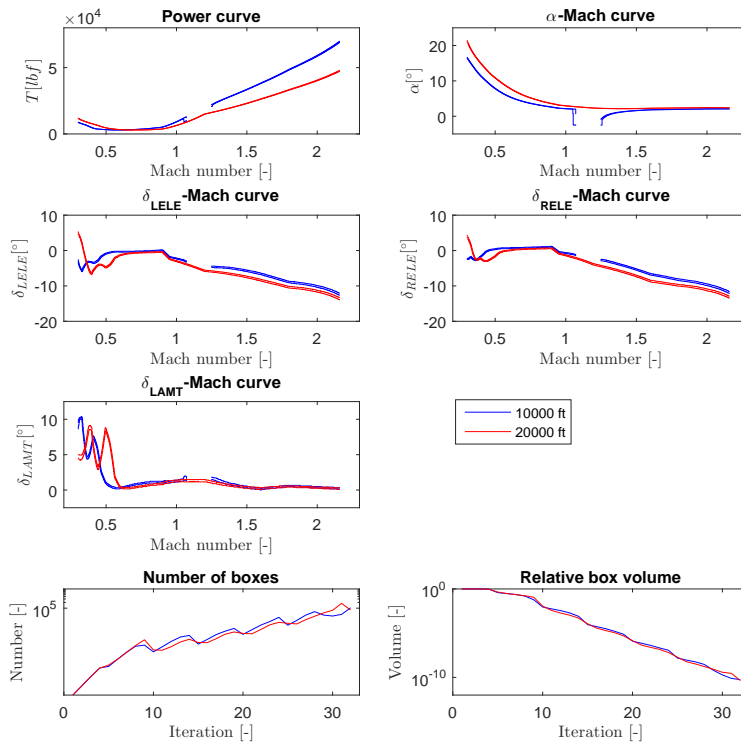
Fig. 8 Full interval analysis solution at 10000 ft and at 20000 ft

needed. In this case the 20,000 ft runs used more boxes too, leading to a calculation time of 3 hours and 37 minutes. Because the high number of boxes and the associated computation time was anticipated, the control deflection box diameter stopping criterion was stretched from $0.01°$ to $0.1°$ for the runs finding continuous bounds on the free variables. What effect this has on the results will be explained below. The right plot shows the relative volume that is remaining after each branch and bound step. This number should be very small, as that indicates that a lot of the initial search domain is removed from evaluation. With both runs ending up at a final ratio of less than 10^{-10} a lot of the initial search domain is lost. For the point solutions the remaining relative box volume was always less than 10^{-13}. Continuing splitting to the tighter $0.01°$ box width criterion certainly helped getting to this lower value, but the fact that the point solutions are a four dimensional problem compared with a five dimensional problem for the continuous solutions cannot be overlooked when comparing these values. The lower value of the point solutions suggest that these

results are more accurate than the continuous set. The elaboration below confirms this.

The target of trimming methods is to find the states where the accelerations are zero. The human vestibular system was found to be able to detect accelerations of $0.02 m/s^2$ for translational motions and $0.05°/s^2$ for rotational motions [9]. As these values are the limits of what humans can possibly detect they were not used as hard criteria. They can now be used as an indication for the quality of the results. For each solution the remaining accelerations are known. The hull of these accelerations is taken and its bounds are plotted in Figures 9 to 12. The most striking thing of these plots is that irrespective of the run the accelerations are much wider at high speeds compared to low speeds. The force that a control effector can exert on the aircraft body is a function of velocity. All the boxes are equally wide throughout the range of Mach numbers. That means that at low speeds a certain interval is multiplied with a small number, while at high speeds that same interval is multiplied with a much larger number. This results in wider intervals in terms of forces, and this caries through to the final accelerations. With this in mind it is not surprising that the only two trim points that meet the $0.02 m/s^2$ and $0.05°/s^2$ criteria are at 20,000 ft at the lowest two Mach numbers, 0.3000 and 0.4328. Because the available control forces also are a function of atmospheric density the widths of the accelerations are larger at lower altitudes. This lead to the best trim point at 10,000 ft being at Mach 0.3000. This solution only has two boxes out of 37 boxes violating the sensible acceleration limits. Comparing the point solutions to the set solution it becomes clear what effect the relaxed precision criteria has had: a factor ten less accurate in box accuracy leads to accelerations that are approximately a factor ten larger. The translational accelerations in x- and y-direction are acceptable, but the ones in z-direction are too large. In the full set solution rotational accelerations can be as large as $10°/s^2$. With this acceleration it only takes 8.49 seconds to do a full rotation. This is clearly not acceptable for a trim set. Further refinement is required, especially at high speeds. Keep in mind that every box does at least contain zero in the acceleration, otherwise it would have been removed from the list of boxes. So despite that the bounds on the control deflections less than 1° apart the bounds on the accelerations are too wide to accept these results as trim sets. The `fmincon` algorithm used in the trim routine that was supplied with the ICE model can have severe rotational accelerations at trim solutions. In one case these are as large as $34°/s^2$.

6 Conclusion

Looking at the results it can be seen that interval analysis is a method that has great potential as a trimming method. With the current implementation it was possible to get bounds of a continuous trim profile for the highly overactuated ICE model that are only 1° wide. Within these 1° bounds the accelerations are still up to maxima $0.5 m/s^2$ in translational motions and $10°/s^2$ in rotational motions. This is something that needs improvements before the trim sets can be accepted. The first potential for
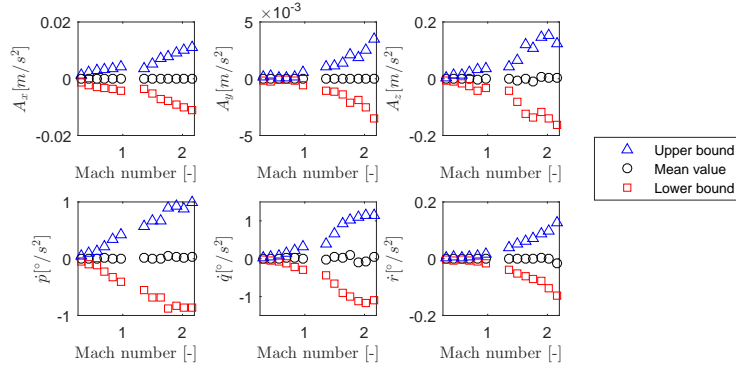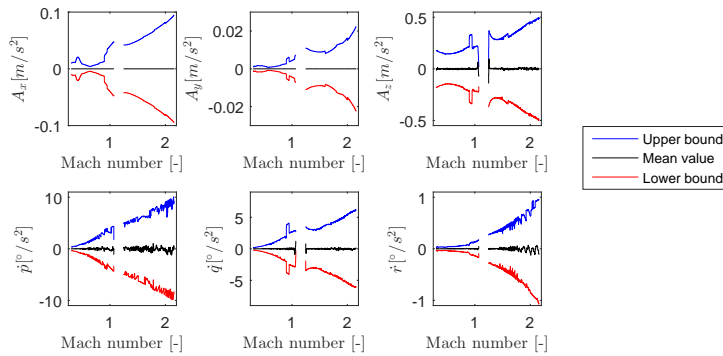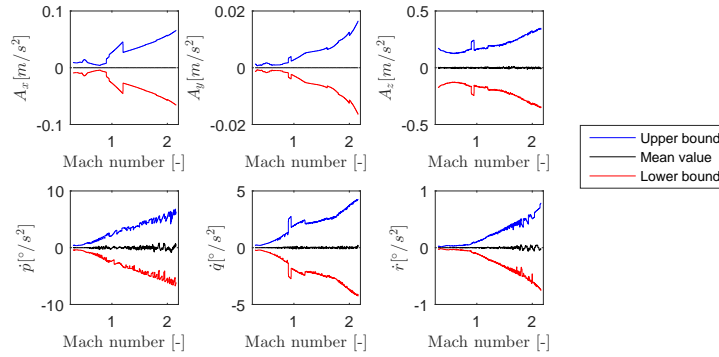
Fig. 9 Mean acceleration and bounds for trim points at 10000 ft
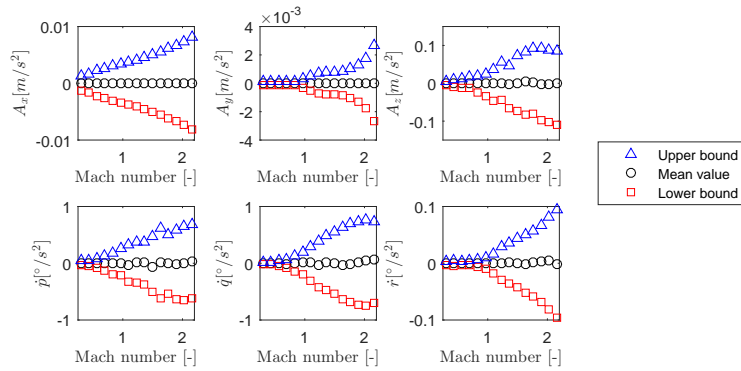
Fig. 10 Mean acceleration and bounds for trim points at 20000 ft

Fig. 11 Mean acceleration and bounds for trim set at 10000 ft

Fig. 12 Mean acceleration and bounds for trim set at 20000 ft

improved results is to make the model also calculate directional derivatives. These could then be used to implement the very efficient Newton interval algorithm. If there are no zeros in the derivative of a function the interval Newton algorithm has a huge box size reduction potential. The directional derivatives could also be used to choose the splitting direction for the box splitting procedure. If the gradient is large in one direction it can be expected that it will result in more boxes that do not contain zeros, which keeps the boxcount low. The next improvement is aimed at the current implementations efficiency. In its current form the model can only evaluate one box at a time. With an average throughput of 160 boxes per minute this becomes impractical if high precision is desired. Because the code is written in Matlab, it should be possible to exploit its power to work with vectorized lists of boxes, which allows for all boxes to be calculated at once, which can allow for huge gains in computation times. The challenge that is not overcome for this work is rooted in the interval interpolation functions. Finally a multivariate B-spline model has been developed at the institution where this work was performed. If the interval routine was altered so that it could work with that model the whole interpolation of data tables would be made redundant, because spline models are fully analytical. Another advantage is that the directional derivatives can be directly calculated from the spline model as well. This indicates that there is a lot to gain from applying interval analysis to spline models. There is also a big downside to spline models. Spline functions are summations of basis functions and each basis function has one entry for the barycentric coordinates of the free variables. This means that interval arguments occur many times in a spline function. This can give rise to a big dependency problem. With the above improvements it might be possible to calculate trim sets with more free control effectors. The increase in calculation time after going from a four variable problem to a five variable problem was immense, and with the current implementation this is expected to happen again if a six variable problem is attempted. Furthermore the required number of boxes is expected to cause problems with memory limitations of current computers. Compared to the more conventional trim algorithm based that uses the Matlab function `fmincon` the found results are slightly different. Because

the conventional method tries to minimize the squared sum of all control effectors it tends to find solutions that have smaller control deflections. The interval analysis trim procedure focuses purely at the accelerations of the model and tries to bound the zeros. This gives smaller leftover accelerations when very small box diameters are reached. It does require more time than the `fmincon` algorithm. The biggest drawback of the `fmincon` method is that the initial guess can influence whether or not a solution is found. Interval analysis does not have this issue. Instead the only input needed for it to start is the search domain. And despite the fact that it cannot find exact solutions like `fmincon` can it does have the ability to bound continuous trim sets over large domains. If the efficiency of the current approach can be increased the advantages of interval analysis can make it a trimming method that is very suitable for highly overactuated aircraft such as ICE with clear advantages over conventional trimming methods.

References

1. Addington, G.A., Myatt, J.H.: Control-surface deflection effects on the innovative control effectors (ice 101) design. Tech. Rep. Ice 101, Wright-Patterson Air Force Base (2000)
2. Chen, T., van Kampen, E., Yu, H., Chu, Q.: Optimization of Time-Open Constrained Lambert Rendezvous Using Interval Analysis. Journal of Guidance, Control, and Dynamics **36**, 175–184 (2013). DOI 10.2514/1.56773
3. De Marco, a., Duke, E.L., Berndt, J.S.: A General Solution to the Aircraft Trim Problem. Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit (AUGUST 2007), 1–40 (2007). DOI doi:10.2514/6.2007-6703
4. Dorsett, K.M., Mehl, D.R.: Innovative Control Effectors (ICE). Tech. rep., Wright-Patterson Air Force Base, Fort Worth, Texas (1996)
5. Dorsett, K.M., Mehl, D.R.: Innovative control effectors (ice) phase ii. Tech. rep., Wright-Patterson Air Force Base, Fort Worth, Texas (1997). DOI WL-TR-96-3043
6. Elgersma, M.R., Morton, B.G.: Nonlinear Six-Degree-of-Freedom Aircraft Trim. Journal of Guidance, Control, and Dynamics **23**(2), 305–311 (2000). DOI 10.2514/2.4523
7. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. SIAM Journal on Optimization **12**(4), 979–1006 (2002). DOI 10.1137/S1052623499350013
8. Hansen, E., Walster, G.W.: Global Optimization Second Edition , Revised and Expanded. Marcel Dekker, Inc, New York (2004)
9. Heerspink, H.M., Berkouwer, W.R., Stroosma, O., Mulder, M.: Evaluation of vestibular thresholds for motion detection in the simona research simulator. AIAA Modeling and Simulation Technologies Conference and Exhibit (August), 1–21 (2005). DOI 10.2514/6.2005-6502
10. ICAO: A Coordinated, Risk-based Approach to Improving Global Aviation Safety. Tech. rep., International Civil Aviation Organization, Montreal (2016)
11. IEEE Standards Board: Ieee standard for binary floating-point arithmetic (1985)
12. Jaulin, L., Kieffer, M., Didrit, O., Walter, E.: Applied Interval Analysis. Springer-Verlag (2001)
13. Juliana, S.: Re-entry Flight Clearance. Delft University of Technology (2006)
14. van Kampen, E., Chu, Q., Mulder, J., van Emden, M.: Nonlinear aircraft trim using interval analysis. In: AIAA Guidance, Navigation and Control Conference and Exhibit, August (2007). DOI 10.2514/6.2007-6766
15. van Kampen, E., Zaal, P., de Weerdt, E., Chu, Q., Mulder, J.: Optimization of Human Perception Modeling Using Interval Analysis. Journal of Guidance, Control, and Dynamics **33**, 42–52 (2010). DOI 10.2514/1.46318

16. Kraemer, W.: Generalized Intervals and the Dependency Problem. Proceedings in Applied Mathematics and Mechanics **684**, 683–684 (2006). DOI 10.1002/pamm.200610

17. Kwatny, H.G., Dongmo, J.E.T., Chang, B.C., Bajpai, G., Yasar, M., Belcastro, C.: Nonlinear Analysis of Aircraft Loss of Control. Journal of Guidance, Control, and Dynamics **36**(1), 149, 162 (2013). DOI 10.2514/1.56948

18. Moore, E.R., Kearfott, R.B., Cloud, M.J.: Global Optimization using Interval Analysis: Interval Optimization for Aerospace Applications. Society for Industrial and Applied Mathematics (2009)

19. Moore, R.E., Yang, C.T.: Interval analysis 1. Tech. rep., Lockheed Aircraft Corporation Missiles and Space Division, Sunnyvale, California (1959)

20. Neumaier, A.: Taylor Forms - Use and Limits. Reliable Computing **9**(February), 43–79 (2003)

21. van Oort, E.R., Chu, Q.P., Mulder, J.A.: Maneuver Envelope Determination through Reachability Analysis. In: F. Holzapfel, S. Theil (eds.) Advances in Aerospace Guidance, Navigation and Control, pp. 91–102. Springer-Verlag, Berlin Heidelberg (2011)

22. Paranjape, A., Sinha, N.K., Ananthkrishnan, N.: Use of Bifurcation and Continuation Methods for Aircraft Trim and Stability Analysis A State-of the-Art. Journal of Aerospace Sciences and Technologies **60**(2), 85–100 (2008)

23. Rump, S.: INTLAB - INTerval LABoratory. In: T. Csendes (ed.) Developments in Reliable Computing, pp. 77–104. Kluwer Academic Publishers, Dordrecht (1999). URL http://www.ti3.tuhh.de/rump/

24. Shankar, P.: Characterization of Aircraft Trim Points Using Continuation Methods and Bifurcation Analysis. AIAA Guidance, Navigation, and Control (GNC) Conference pp. 1–12 (2013). DOI 10.2514/6.2013-4924

25. Tupy, J., Zelinka, I.: Evolutionary algorithms in aircraft trim optimization. In: Proceedings - International Workshop on Database and Expert Systems Applications, DEXA, pp. 524–530 (2008). DOI 10.1109/DEXA.2008.98

26. Van Hentenryck, P., McAllester, D., Kapur, D.: Solving Polynomial Systems Using a Branch and Prune Approach. SIAM Journal on Numerical Analysis **34**(2), 797–827 (1997)

27. Van Kampen, E.: Global Optimization using Interval Analysis: Interval Optimization for Aerospace Applications. Delft University of Technology (2010)