Word count= 4098

Reachable Sets for Flight Trajectories: An Application of Differential Inclusions to Flight Maneuver Simulation

Stanislaw Raczynski
Universidad Panamericana
Augusto Rodin 498, Mixcoac, 03910 Mexico City
stanracz@prodigy.net.mx

Reachable Sets for Flight Trajectories: An Application of Differential Inclusions to Flight Maneuver Simulation

ABSTRACT

The main mathematical tool in flight simulation is the ordinary differential equation. However, in many situations this is not the sufficient way to solve problems. In robust flight control design, safety, missile and aircraft guidance, influence of perturbances or differential games of pursuit-evasion problem more versatile tools are needed. *Differential Inclusion* (DI) is a generalization of a differential equation that can be extremely useful. The solution of a DI is not just a model trajectory or a set of trajectories obtained by a randomization of the original problem. The solution is a reachable set, and it is a deterministic object.

It is pointed out that the reachable sets cannot be assessed properly while treating the uncertain variables as random ones. The application of the *differential inclusion solver* can give the proper view of the regions in the state space where all the possible model trajectories belong.

Reachable Sets for Flight Trajectories: An Application of Differential Inclusions to Flight Maneuver Simulation

Keywords: Flight simulation, differential inclusions

ABSTRACT

The main mathematical tool in flight simulation is the ordinary differential equation. However, in many situations this is not the sufficient way to solve problems. In robust flight control design, safety, missile and aircraft guidance, influence of perturbances or differential games of pursuit-evasion problem more versatile tools are needed. *Differential Inclusion* (DI) is a generalization of a differential equation that can be extremely useful. The solution of a DI is not just a model trajectory or a set of trajectories obtained by a randomization of the original problem. The solution is a reachable set, and it is a deterministic object.

It is pointed out that the reachable sets cannot be assessed properly while treating the uncertain variables as random ones. The application of the *differential inclusion solver* can give the proper view of the regions in the state space where all the possible model trajectories belong.

GENERAL PROPERTIES OF DIFFERENTIAL INCLUSIONS

There is a strange conviction among some of the modeling and simulation specialists that everything what is continuous can be described by ordinary (ODE) of partial differential equations. In general, this is not exactly truth. The ODE model for the system can have no unique solution, or it may not exist at all. Some simulation tasks which involve uncertainty or those which are carried out to determine all possible solutions need other tools, like differential inclusions.

Recall that a differential inclusion (DI) is a differential equation with multi-valued right-hand side. An ordinary differential equation (ODE) in a real vector space can be defined as follows

$$\frac{dx}{dt} = f(x,t)$$

where x is an n-dimensional vector and f is a vector-valued function. A DI can be treated as a generalization of the above equation, given in the following form:

$$(1) \qquad \frac{dx}{dt} \in F(x,t)$$

where $F(x,t)$ is a set. The solution to a ODE with a given initial condition, over a time-interval $J=(0,T)$ is a model trajectory $x(t)$, which graph belongs to the space R^n x J. The solution to a DI is neither a function nor a set of functions. It is a set in $R^n \, x \, J$, called Reachable set (RS) or Attainable Set.

The DIs has been known for about 70 years and there is wide literature available on the DIs theory and applications. The first works have been published in 1931-32 by Marchaud and Zaremba (Marchaud 1934, Zaremba 1936). They used the terms "contingent" or "paratingent" equations. In the decade 1930-40 such problems as the existence and properties of the solutions to the DIs have been resolved in the finite-dimensional space. Later, in 1960-70, T. Wazewski and his collaborators (Turowicz 1962, 1963, Wazewski 1961, 1962a, 1962b, Plis 1961) published a series of works, referring to the DIs as *orientor conditions* and *orientor fields.* After this, many works appear on DIs in more abstract, infinite-dimensional spaces. Within few years after the first publications, the DIs resulted to be the basic tool in the optimal control theory. An excellent text on DIs can be found in Aubin and Cellina (1984). See also Raczynski (1986,1996).

Recall that optimal trajectories of a dynamic system are those that lay on the boundary of the system reachable set. In the works of Pontiragin (1962), Lee and Markus (1961), Bellman and many others, one of the fundamental problems is the determination of the properties of reachable sets. Using the theory of Marchaud and Zaremba, T.Wazewski pointed out that in many optimal control problems the resulting control strategy is the so-called bang-bang control, generated by switching controllers.

We shall not give here a detailed review on the DIs. A more extended survey can be found in Raczynski, 1996. One of the best texts on DIs in finite-dimensional as well as abstract spaces is the book of Aubin and Cellina (1984). These references contain an exhaustive overview on the DIs. However, some basic properties should be recalled.

If the set F can be parametrized by certain variable u, then the DI can be defined in the form of an equivalent *control system*, as follows:

$$(2)\qquad \begin{cases} dx/dt = f(x,u,t) \\ x(0) \in I \\ u \in C(x,t) \\ f = (f_1, f_2, \dots, f_n) \\ x \in X, \quad C \subset U, \quad t \in J \end{cases}$$

Here U denotes the *control space*. The relation between the set F of (1) and the function f of (2) is as follows.

$$(3)\qquad F(x,t) = \{ z : z = f(x,u,t),\ u \in C(x,t) \}$$

For each particular function $u(t)$ which values belong to $C(x,t)$ we get a trajectory of the DI. The union of the graphs of all trajectories forms the reachable set. It was pointed out that each section of the RS with a plane $t=const$ is a connected set. Moreover, with certain regularity assumptions, the RS is a closed set. A trajectory, which reaches a boundary of the RS for the given final time T is an optimal one due to certain optimality criterion. Namely, it maximizes or minimizes a linear form $V = p_1x_1 + p_2x_2 + \dots + p_nx_n$ for a fixed vector p. This fact is well known in the optimal control theory. Moreover, the whole graph of such an optimal trajectory must belong to the boundary of the reachable set. This property can be used to scan the boundary of the RS, which is the solution to the DI. Such scanning algorithm has been implemented in the Differential Inclusion Solver, described further on in this paper.

APPLICATIONS OF DIFFERENTIAL INCLUSIONS

There is a strong relation between DIs and the Control Theory. For practical and computational reasons a DI is almost always represented in the form of an equivalent control system, as in equation (3). The main numerical problem is the representation of multi-dimensional sets in the computer memory. Frequently we are given a control system and then treat it as a DI. If the original problem is formulated in the form of a DI, we can find a corresponding control system, where the controls parameterize the right-hand side of the DI. Note that such representation is not unique and may change when we change the way we parameterize the set. One of the most typical problems which result in a DI in its original form (a set at the right-hand side), is the problem of dynamical uncertainty. If one or more model parameters are uncertain then the DI is the proper modeling tool. Note that the solution of a DI is the reachable set and it is deterministic. So, it is a fundamental difference between the uncertainty modeled through a DI, and a model where the uncertain parameters are treated as random variables (see Raczynski 1996). This difference will be commented further on.

Perhaps the most important problem in flight control is the resolution of conflict situations. What we need is the shape of the set of aircraft positions after a maneuver. In other words, the information that the probability of an accident *was* equal to 0.0001 is not very relevant to a victim of an accident.

Other possible applications belong to financial, marketing, stock market problems and differential games. A *differential game* is played in continuous time. The decisions of the players make the system state to evolve due to well-defined rules (in our case the state equations). There is an obvious relation between differential games and optimal control problems. The main difference consists in the fact that in the optimal control problem we are looking for a control that maximizes the optimality criterion, while in the differential game there are different (frequently opposite) optimality criteria for the two or more players. The strategy of each player can be treated as an uncertain control variable u which satisfies given restrictions. Also the state of the whole system can be subject to constraints. Consequently, we get a DI, which solution (the RS) tells us what is the possible outcome of the whole game, for all possible player strategies. This information may be important in many situations. For example, the classical missile-plane (pursuit-evasion) game cannot be won by the missile if the reachable sets for the plane and for the missile do not intersect.

The applications of the DIs in system simulation are not very common. However, some relevant research on DIs in game theory has been done. Many of the works in the field use the Hamilton-Jacobi-Bellman's equations and the methods of the control theory, closely related to the DIs. See, for example Tsunumi and Mino (1990) , who work on the Markov Perfect Equilibrium problems in differential games. Grigorieva and Ushakov (2000) consider the differential game of pursuit-evasion over a fixed time segment. The attainability set is appointed with the help of the stable absorption operator. A more general, variational approach to differential games can be found in Berkovitz and Leonard (1964). The DIs are used by Solan and Wieille (2001) to study the equillibrium payoffs in quitting games. There is no room in this short paper to give a more extensive reference review. For a general problems of the Game Theory consult, for example, Petrosjan and Zenkevich (1996), Isaacs (1999) or Fudenberg and Tirole (1991).

DI SOLVER

One could expect that a solution algorithm for a DI might be obtained as some extension of known algorithms for the ODEs. Unfortunately, this is not the case. First of all, note that the solution to a DI is a set and not a function.

As stated before, the trajectories that scan the boundary of the RS are optimal ones. If we can calculate a sufficient number of such trajectories, then we can see the RS shape. The trajectories should be uniformly distributed over the RS boundary. This can be done by some kind of random shooting over the RS boundary. Such shooting has nothing to do with a *simple* or *primitive random shooting*, when the trajectories are generated randomly inside the RS.

My first attempts to develop a DI solver were presented on the IFAC Symposium on Optimization Methods, Varna, 1974. That was a random shooting method, but not a *simple* shooting. That algorithm generated trajectories inside the RS, but the control variable was being modified to obtain a nearly uniform distribution of points inside the RS at the end of the simulated time interval. The DI solver presented here is much more effective.

The DI solver in its recent version works as follows. The user provides the DI in the form of an equivalent control system. To do it he/she must parameterize the right-hand side (the set F) using an auxiliary variable u. The DI solver automatically generates the equations of so-called conjugated vector $p(t)$ and integrates a set of trajectories, each of them belonging to the boundary of the RS. To achieve this, over each trajectory the Hamiltonian $H(x,p,u,t)$ is maximized. To define the hamiltonian, we must define the conjugated vector $p \in R^n$ that satisfies (by definition) the following equations.

$$(4) \qquad dp_i/dt = -\sum_{j=1}^{n} \frac{\partial f_j}{\partial x_i} p_j + \frac{\partial f_0}{\partial x_i}$$

where $i = 1,..,n$ and f is the vector of the right-hand sides of (2). The component f_o is used in optimal control problems, where f_o corresponds to the object function. In our case it can be omitted, because we do not define any object function. The Hamiltonian is defined as follows.

$$H = \sum_{j=1}^{n} p_j \; f_j$$

In this case, the Maximum Principle states that the necessary condition for the trajectory to terminate at a boundary point of the reachable set is that the control $u(t)$ maximizes the Hamiltonian at each point $t \in J$. This can be used to generate trajectories of a differential inclusion that scan the reachable set boundary. If the inclusion is given in the form of a control system (1.2) we apply the Principle directly. If it is given in the general form (1.1) we must parameterise the set F and treat the parameter as the control.

This procedure is similar to that used in dynamic optimization. In the optimal control problem the main difficulty consists in the boundary conditions for the state and conjugated vectors. For the sate vector we have the initial conditions given, and for the conjugated vector only the final conditions (at the end of the trajectory) are known, given by the *transversality conditions*. This means that the optimal control algorithm must resolve the corresponding two-point-boundary value problem. In the case of a DI we are in better situation. There is no object function and no transversality conditions. As the consequence, for the vector p we can define the final as well as the initial conditions. Anyway we obtain a trajectory which graph belongs to the RS boundary. Defining initial conditions for p we integrate the trajectory only once, going forward. The only problem is how to generate these initial conditions in order to scan the RS boundary with nearly uniform density. The algorithm is quite simple: the initial conditions for *p* are generated randomly, due to a density function that is being automatically modified, covering with more density points that correspond to trajectories that fall into a low density regions at the RS boundary. Trajectories that are very close to each other are not stored (storing only one from each eventual cluster). As the result we obtain a set of trajectories covering the RS boundary that can be observed in graphical form and processed.

The maximization of the Hamiltonian function is a classical problem of multivariable optimization and will not be discussed here. Note that in the above example of a tendor set with only four points, the maximization reduces to simple scanning over the set of four possible points. In general case an appropriate maximization algorithm might be inserted in the above DI solver. However, if we fix this algorithm, or provide some alternative ones, this may result in time-wasting for simple cases and might be insufficient in more complicated situations. This is why the actual DI solver does not provide parameterization and maximization algorithms, leaving this to the user choice. If the original DI is linear (linear equivalent control system), and if the control variable is restricted to an n-dimensional cube, the maximization of the Hamiltonian can be reduced to a simple scanning of the cube vertices. This can work also in the non-linear case, if we are sure that the maximum is reached on a cube vertice. However, experimenting with such kind of search, it can be seen that the resulting shape of the RS (its time-section for t=T) frequently reduces to the extremal points of the set, and does not provide a good image of the contour of the set. This occurs when the shape is non-convex, which is a common case. To avoid this, the search can be done for the cube vertices, but in the case when the Hamiltonian values in two or more vertices are equal to each other, we can define the maximizing control u being a random combination of the controls of the two vertices (lying on the line between the two vertices). This simple randomization trick makes the "holes" in the contour disappear. In the general non-linear case, the maximizing control can be inside the cube. This will require a good maximization procedure to be inserted in the algorithm (steepest descent, conjugated gradient, the Powell method or other). Not however that this will considerably slow down the whole algorithm. Observe also, that even if we do a simple scanning of the cube vertices, the trajectories we obtain must belong to the RS, so what we get is certain assessment of the RS which in many cases is a good one, though it may represent only a subset of the real RS.

A common error committed while solving similar problems (mainly while treating uncertainty) is to perform a *simple* or *primitive random shooting*. The results shown in this paper tell us how ineffective is that method. Generating about 10,000 random trajectories with uniform density in the set F we obtain a small cluster that has nothing to do with the true solution. The diameter of this cluster might be only 5-10% of the diameter of the real solution. Using the algorithm proposed here we could obtain the RS boundary with quite good accuracy generating only several hundred trajectories.

The following example shows the difference between a simple shooting (random control u at each integration step) and the contour of the RS found by the DI solver. The inclusion was as follows.

$$(5)\qquad \begin{cases} \dfrac{dx}{dt} = F(x) \\ F = \{z : z = u - x_1 - 1.2|x_2|\ ,\ u \in [-1,1]\} \\ where \quad x = (x_1, x_2), \quad T = 50 \end{cases}$$

The simple shooting is the result of integrating 10000 trajectories, while the contour found by the solver contains only 500 trajectories.

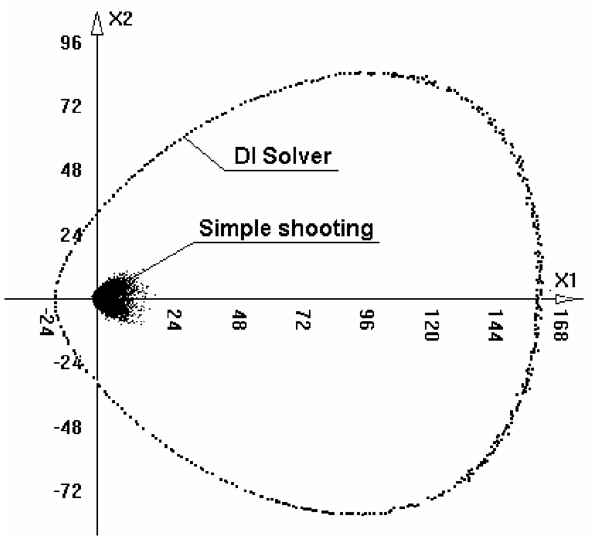

Figure 1. An example of the contour of the reachable set for a second order non-linear system.

FLIGHT DYNAMICS

The model we use is the following set of 5 differential equations that describe a simplified flight dynamics (see figures 2 and 3).

$$(6)\qquad \begin{cases} \dfrac{dv}{dt} = \dfrac{(T-D)}{m} - g\gamma \\ \dfrac{dh}{dt} = v\gamma \\ \dfrac{d\psi}{dt} = \dfrac{L\sin(\phi)}{mv} \\ \dfrac{dx}{dt} = v\sin(\psi) \\ \dfrac{dy}{dt} = v\cos(\psi) \\ L\cos(\phi) = mg \end{cases}$$

where

$T - thrust \quad D - drag \quad L - lift$

$\phi - aerodynamic\ bank\ angle$

$\gamma - flight\ path\ angle$

$\psi - inertial\ heading$

$x, y - position\ (x - forward\ direction)$

This is a simplified flight dynamics. It is supposed that the aircraft is in en-route flight. The influence of the wind is neglected. The angle of attack and the flight path angles are assumed to be small. The thrust and drag are supposed to be aligned. The initial value of the inertial heading is set equal to 90^{o}. The fuel consumption is neglected during the maneuver and the air density is supposed to be constant.

The variables T , γ and Φ are not exactly defined, only the restrictions for these variables are given. So, if T , γ and Φ scan their limits, the right-hand side of (6) scans a set in the 5-dimensional state space. In this way we obtain a differential inclusion, which solution (reachable set) shows the possible range of the flight state variables. The drag D is calculated as follows. First, the coefficient C_L is calculated from the equation

(7) $$L = \frac{1}{2}\rho v^2 S C_L$$

where ρ *– air density*

S *– gross wing area*

L *– lift, calculated due to* (6)

Then, the drag coefficient C_D is calculated from the formula $C_D = C_0 + C_{DT} C_L^2$ where C_0 and C_{DT} are known from the aircraft data. Finally, the drag is calculated as follows:

(8) $$D = \frac{1}{2}\rho v^2 S C_D$$

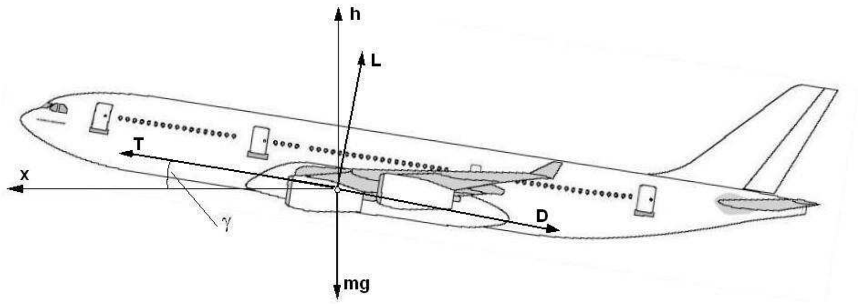

Figure 2. Aircraft side view.

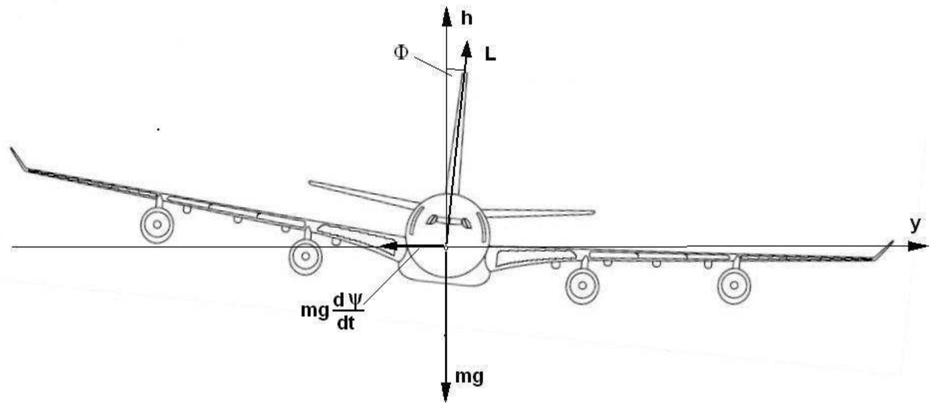

Figure 3. Aircraft front view.

In the experiments carried out with the above model the following parameter values have been assumed:

$m = 200{,}000kg$

$C_0 = 0.018, \quad C_{DT} = 0.0342$

$S = 363m^2$

$T = 1600000\, Newton$

$\rho = 0.5kg / m^2$

$\gamma \in [-18^o, +18^o\ ]$

$\phi \in [-21^o, +21^o\ ]$

$v(0) = 200m/\sec$

$h(0) = 0 \quad (relative\ height)$

$\psi(0) = 90^O$

$x(0) = 0$

$y(0) = 0$

On the following figures the state variables are as follows:
$X_1 = v \quad X_2 = h \quad X_3 = \psi \quad X_4 = x \quad X_5 = y$

Figure 4 shows the reachable set for the flight trajectories, coordinates y and h. Those are the trajectory end points reached after 15 seconds of flight. Each dot represents a position which belongs to the boundary of the reachable set. Some points seem to be inside the RS, but it is not true. Note that what we can see on a 2D image is only a projection from a 5-dimentional set of points. Observe the small cluster of points (small dots) in the centre of the figure. This set is the result of simple random shooting, where the control variables ($\gamma \quad and \quad \phi$) are generated as a random ones within the same limits. The cluster contains 10,000 trajectory end points, while the RS obtained with the DI Solver consists in only 5000 trajectories.

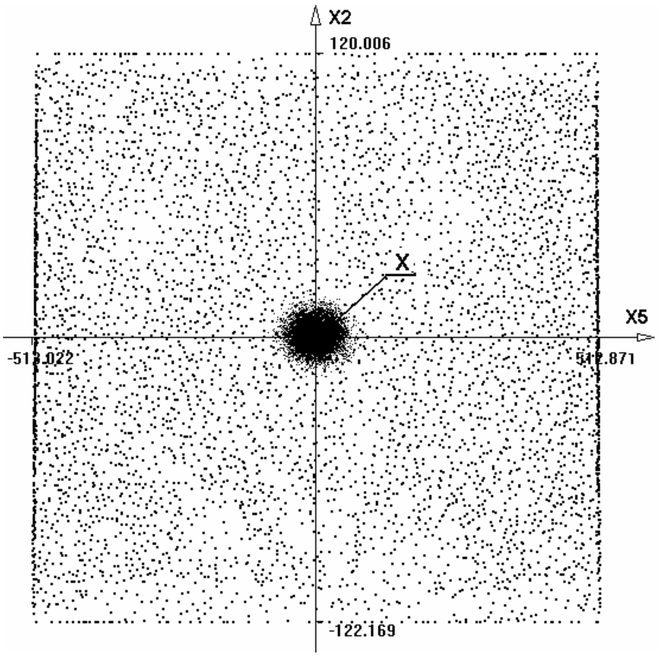

Figure 4. The reachable set for time=15, coordinates *y* and *h*.

On figure 5 you can see the 3D image of the same RS, shown as a cloud of points, in two different view angles. On the right image a bold lines have been manually added to make the shape of the cloud much visible.

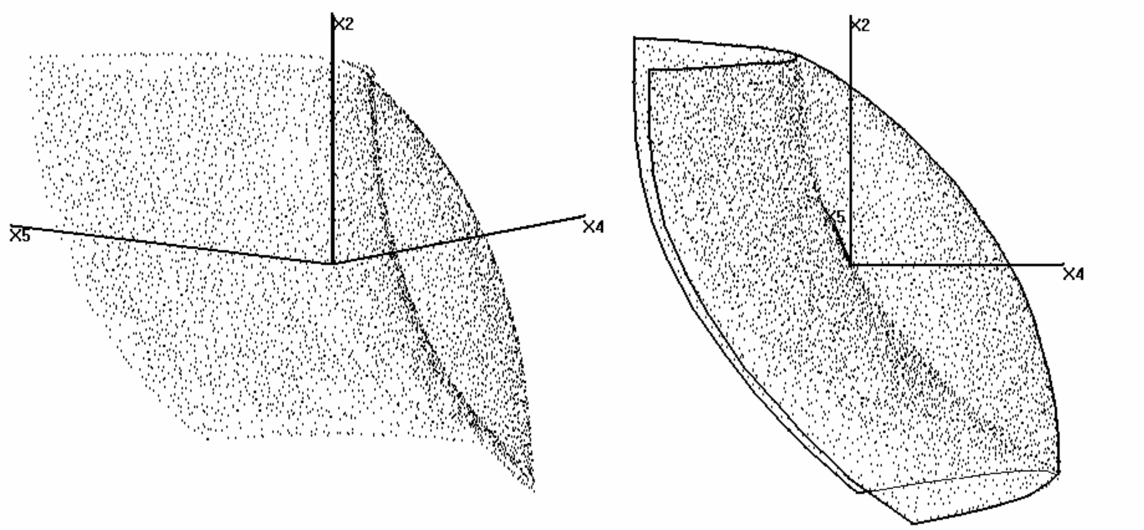

Figure 5. 3D images of the reachable set.

If we increase the limits for the control variables and the final flight time the RS becomes much deformed due to the non-linearities of the model. On figure 6 you can see the image similar to that of figure 4, with the final flight time equal to 30 sec and $\gamma \in [-36^o, +36^o]$.

The system trajectories are being stored in a file, together with the respective controls. So, selecting any point from the RS image we can see not only the model trajectory, but also the corresponding strategies (controls).

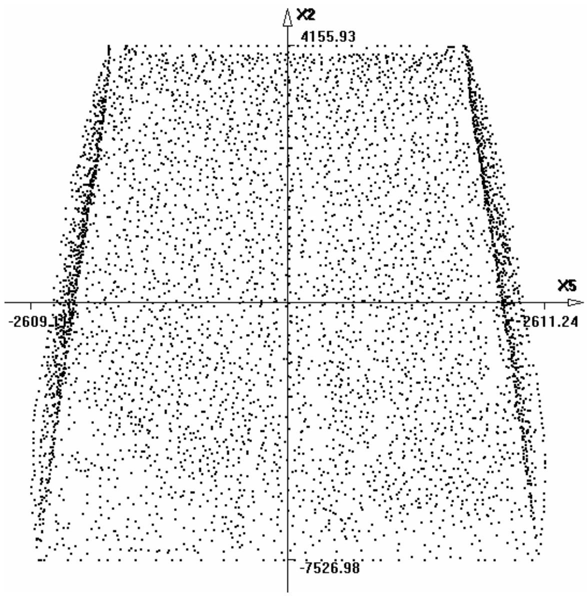

Figure 6. The reachable set after 30 seconds of flight, increased limits for the flight path angle.

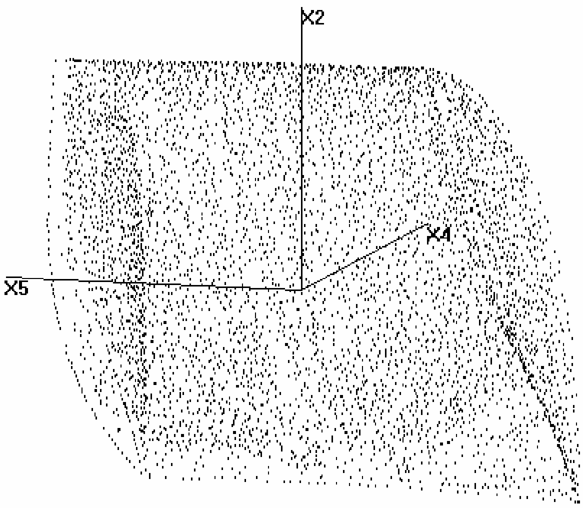

Figure 7. 3D view of the reachable set with model parameters as for the figure 6.

CONCLUSIONS

DI solver exists and works quite well, though further research is necessary. The main problems in developing DI solvers are the following.

1. Computational complexity. To advance one integration time step, the solver must perform a multi-dimensional optimization. Moreover, to estimate the shape of the DI solution (its reachable set), several hundreds (or thousands, in a multi-dimensional case) of trajectories must be integrated. This make the whole process slow, compared to any algorithm for a single-trajectory integration using ordinary differential equations (ODE models).
2. DI right hand side and solution representation. While treating ODE models, the right-hand side of the equations is an N-dimensional point, and the resulting trajectory can be given as a sequence of points. In the case of a DI we must store in the computer memory sets instead of points. The final solution is a sequence of sets (one for each integration time interval) and the required representation of the reachable set can (and should) be given as its boundary surface. In multidimensional case this is not a trivial problem.
3. The extensions of the DIs to partial differential equations are possible. However, this means that the state space is no longer Euclidean multidimensional space, but rather an abstract one like Banach, Hilbert or Sobolev spaces. In this case serious computational difficulties may arise.

As for the flight simulation, in many situations the knowledge about the reachable set may be crucial in safety problems, parameter uncertainty and robust control design.

REFERENCES

Aubin J.P., Cellina A., 1984, *Differential Inclusions*. Springer-Verlag.
Tsunumi S., Mino K., 1990, Nonlinear Strategies in Dynamic Duopolistic Competition with Sticky Prices. *Journal of Economic Theory,* 52, pp.136-161.
Grigorieva S. V. and Ushakov V. N., 2000, Use Finite Family of Multivalued Maps for Constructing Stable Absorption Operator. *Topological Methods in Nonlinear Analysis,* Vol. 15, No. 1, March 2000, Juliusz Schauder Center, Torun, Poland
Berkovitz L.D., 1964, Variational approach to differential games. *Advances in game theory* pp. 127--174 Princeton Univ. Press, Princeton, N.J.
Solan E and Vieille N, 2001, Quitting Games. *Math. Oper. Res.,* 26, pp.265-285.
Petrosjan L., A. and Zenkevich N. A., 1996, *Game Theor.,* World Scientific Publishing Co., Inc.,Feb 1996.
Fudenberg D. and Tirole J. 1991, *Game Theory*. MIT Press.
Isaacs R., 1999, *Differential Games*, Dover Books on Mathematics, Dover Publications, Inc., NY.
Marchaud A., 1934, *Sur les champs de demi-cones et les équations differielles du premier ordre*. Bull. Soc. Math. France, 62.
Zaremba S.K. 1936, *Sur les équations au paratingen.,* Bull. Sci. Math., 60.
Turowicz A., 1962, *Sur les trajectoires et quasitrajectoires des systemes de commande nonlineaires.* Bull. Acad. Polon. Sci. Ser. Math. Astronom. Phys. 10 no. 10, Warszawa.
Turowicz A., 1963, *Sur les zones d'emision des trajectoires et des quasitrajectoires des systemes de commande nonlineaires*, Bull. Acad. Polon. Sci. Ser. Math. Astronom. Phys., 11 no. 2, Warszawa.
Wazewski T., 1961, *Sur une condition equivalente a l'equation au contingen.,* Bull. Acad. Polon. Sci. Ser. Math. Astronom. Phys., 9 no. 12, Warszawa.
Wazewski T., 1962a, *Sur une genralisation de la notion des solutions d'une equation au contingent.* Bull. Acad. Polon. Sci. Ser. Math. Astronom. Phys., 10 no.1, Warszawa.
Wazewski T., 1962b, *Sur les systemes de commande non lineaires dont le contredomaine de commande n'est pas forcement convex.,* Bull. Acad. Polon. Sci. Ser. Math. Astronom. Phys. 10 no.1, Warszawa.
Plis A., 1961, *Remark on measurable set-valued functions.* Bull. Acad. Polon. Sci. Ser. Math. Astronom. Phys. 9 no. 12, Warszawa.
Lee E.B., Markus L., 1967, *Foundations of Optimal Control Theory.* Wiley, New York.
Pontriagin L.S., 1962, *The mathematical theory of optimal processes.*(transl.) Wiley Interscience, New York.
Raczynski S., 1986, *Some Remarks on Nonconvex Optimal Control.* Journal of Mathematical Analysis and Applications, New York.
Raczynski S., 1996, *Differential Inclusions in System Simulation.* Transaction of the Society for Computer Simulation, vol. 13, no.1, pp 47-54.