Homework 3 : Aircraft Collision Avoidance Analyses using Reachability

David McPherson

May 12, 2016

1 Linear Velocity Control

Consider a system of two airplanes (Dubin's vehicles) careening through the sky on a collision course. Our two planes can only control their linear velocity and cannot escape into a veering mode. This no-turning system defined in terms of relative coordinates is:

$$\begin{aligned}\dot{x}_r &= -u + dcos(\psi_r) \\ \dot{y}_r &= dsin(\psi_r) \\ \dot{\psi}_r &= 0\end{aligned}$$

We can use Ian Mitchell's "Toolbox of Level Set Methods" to perform reachability analysis on this system. We will modify the default example Matlab script "air3D" that comes with his ToolboxLS.We will need to change the functions defining the Hamiltonian ("air3DHamFunc") and the Hamiltonian's maximum partials ("air3DPartialFunc").

1.1 MATLAB Function: HamFunc

We need to find the optimal Hamiltonian so we can program it into the "air3DHamFunc" function. Let us recast the bounds on the controls u and d as:

$$\begin{aligned}\underline{v_1} &= c_u - o_u \\ \bar{v_1} &= c_u + o_u \\ \underline{v_1} &= c_d - o_d \\ \bar{v_1} &= c_d + o_d\end{aligned}$$

So that c_u is the center of the region U and o_u is the offset from the center to either end of the interval U. Accordingly, c_d is the center of D and o_d is the offset to either end of D.

Now, note that the system dynamics can be re-cast into a control-affine form:

$$\dot{z} = \begin{bmatrix}\dot{x}_r \\ \dot{y}_r \\ \dot{\psi}_r\end{bmatrix} = \begin{bmatrix}-u + dcos(\psi_r) \\ dsin(\psi_r) \\ 0\end{bmatrix} = \begin{bmatrix}0 \\ 0 \\ 0\end{bmatrix} + \begin{bmatrix}-1 \\ 0 \\ 0\end{bmatrix} u + \begin{bmatrix}cos(\psi_r) \\ sin(\psi_r) \\ 0\end{bmatrix} d$$

This allows us to decouple the effects of the input and output and optimize each individually. The corresponding (non-optimum) Hamiltonian is:

$$\max_u \min_d H(z,p,u,d) = p^T f(z,u,d) = -p_1 u + (p_1 cos(\psi_r) + p_2 sin(\psi_r))d$$

The optimum controls will always push the boundary of their set, depending on the sign of their coefficient. For example, for u to maximize the Hamiltonian it should assume it's minimum value when it's coefficient is negative ($-p_1 < 0$), and u should assume it's maximum value when it's coefficient $-p_1 > 0$. The converse is true for d as it seeks to minimize the Hamiltonian. Thus we can obtain expressions for the ideal controls u and d as:

$$\begin{aligned} u^* &= c_u + o_u sign(-p_1) \\ d^* &= c_d + o_d sign(p_1 \cos(\psi_r) + p_2 \sin(\psi_r)) \end{aligned}$$

Using these optimal controls in the Hamiltonian retursn the Optimal Hamiltonian as:

$$H^*(z,p) = \max_u \min_d H(z,p,u,d) = H(z,p,u^*,d^*) \tag{1}$$

$$= -c_u p_1 + o_u |p_1| + (p_1 cos(\psi_r) + p_2 sin(\psi_r))c_d - o_d |p_1 \cos(\psi_r) + p_2 \sin(\psi_r)| \tag{2}$$

This is the Hamiltonian that was used in the $air3DHamFunc$ Matlab script.

1.2 MATLAB Function: PartialHamFunc

The second utility functions the Lax Friedrichs approximation requires for simulating the reachable set is a function which approximates:

$$\alpha_i(z) = \max_p \left| \frac{\delta}{\delta p_i} H^*(z,p) \right|$$

Although actually taking the maximum over p can be difficult (especially since our controls depend on p discontinuously), we can still approximate α_i by using an upper bound as an over approximation. Overapproximating this term corresponds to an artificial dissipation being added to the system, which is preferable to underestimation.

We will derive a general over approximation for any system in a control affine form (like the one we cast our system into).

$$H(x,p,u,d) = p^T f(x,u,d) = p^T F(x) + p^T F^u(x) u + p^T F^d(x) d$$

$$\frac{\delta}{\delta p_i} H(x,p,u,d) = f_i(x,u,d) = F_i(x) + F_i^u(x) u + F_i^d(x) d$$

$$
\begin{aligned}
\left|\frac{\delta}{\delta p_i}H(x,p,u,d)\right| &= |f_i(x,u,d)| = |F_i(x) + F_i^u(x)u + F_i^d(x)d| \\
&\leq |F_i(x)| + |F_i^u(x)||u| + |F_i^d(x)||d| \\
&\leq |F_i(x)| + |F_i^u(x)|U^{max} + |F_i^d(x)|D^{max}
\end{aligned}
$$

where $U^{max} = \max|c_u + o_u|, |c_u - o_u|$ and $D^{max} = \max|c_d + o_d|, |c_d - o_d|$

Plugging in our particular control-affine system we obtain upper-bound estimates of our α_i as:

$$
\alpha_1 \leq U^{max} + D^{max}|\cos(\psi_r)|
$$
$$
\alpha_2 \leq D^{max}|\sin(\psi_r)|
$$
$$
\alpha_3 = 0
$$

1.3 Results

With the requisite utility MATLAB functions defined corresponding to the particulars of our problem, our script is ready to roll. The generated sets that reach the unsafe set are shown in the following figures for four different relative angles.

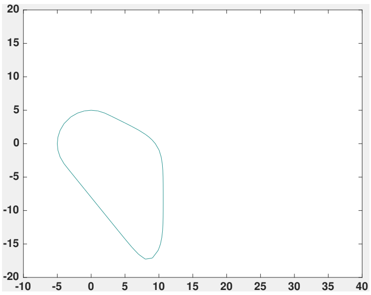

Figure 1: Unsafe set for Relative Angle $\psi_r = \frac{\pi}{2}$

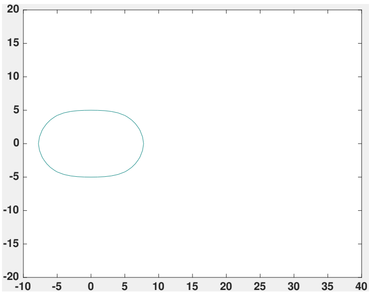

Figure 2: Unsafe set for Relative Angle $\psi_r = 0$

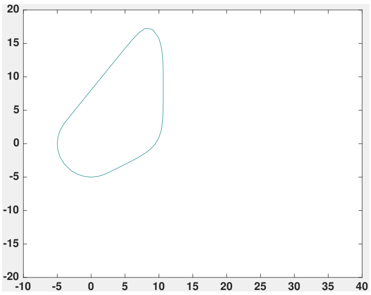

Figure 3: Unsafe set for Relative Angle $\psi_r = -\frac{\pi}{2}$

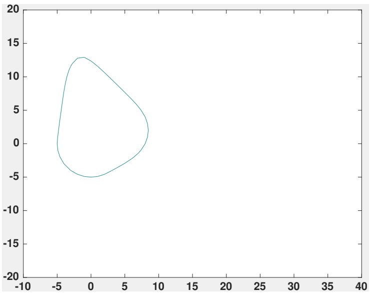

Figure 4: Unsafe set for Relative Angle $\psi_r = -\frac{\pi}{4}$

As the time horizon $T \rightarrow \infty$ the reachable set reached a fixed point and ceased to expand.

2 Mode Switching Control

Reachability analysis within mode 2 (turn mode) shows that for larger radii, the arc of the controllable predecessor's complement (hereby referred to in shorthand as the ”macaroni”) increases as well (see Figures 5-6).

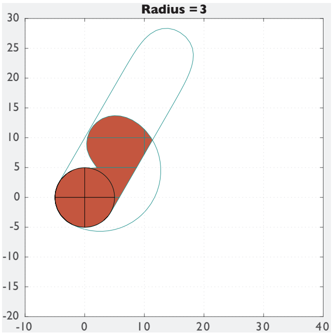

Figure 5: Unsafe set for Mode 2 (avoid mode) with Turning Radius 3

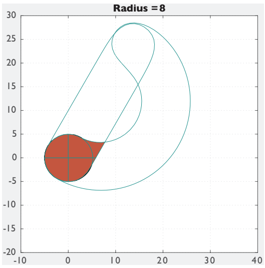

Figure 6: Unsafe set for Mode 2 (avoid mode) with Turning Radius 8

The final unsafe set is the set of states that can reach the unsafe disk while avoiding the controllable predecessor (that is, everything outside the ”macaroni”). This unsafe set is depicted in Figures 5 and 6 as the shaded region. Note that when the ”macaroni” bends more (due to the larger radius turns) it cuts off more of the unsafe set. Thus, the unsafe set shrinks with larger turn radius. This makes intuitive sense, since a larger turning arc gives the planes more of a gap between the planes creating more of a ”cushion”.