Marabou input bounds for verifying the safety of a neural network representation of ACAS Xa

Kyle Julian

May 24, 2018

1 Advisories

The variable $w = \pm 1$ is used to specify whether the bound is an upper or lower bound. If the bound is a lower bound, $w = -1$ and $v \in (-\infty, v_{\text{lo}}]$, and if the bound is an upper bound, $w = 1$ and $v \in [v_{\text{lo}}, +\infty)$. Each advisory has an associated w and $v_l o$.

There are 11 advisories in Run 15 of the Acas Xa vertical logic:

Table 1: ACAS Xa Advisories

Advisory	Description	w	v_{lo} (ft/min)
COC	Clear of conflict	N/A	N/A
DNC	Do not climb	-1	0
DND	Do not descend	+1	0
Maintain	Maintain vertical rate	N/A	N/A
DES1500	Descend at least 1500 ft/min	-1	-1500
CL1500	Climb at least 1500 ft/min	+1	+1500
SDES1500	Strengthen descent to at least 1500 ft/min	-1	-1500
SCL1500	Strengthen climb to at least 1500 ft/min	+1	+1500
SDES2500	Strengthen descent to at least 2500 ft/min	-1	-2500
SCL2500	Strengthen climb to at least 2500 ft/min	+1	+2500
MTLO	Multi-threat level-off	N/A	N/A

2 In terms of τ

The reduced dimensionality network uses a τ variable rather than r and r_v, where

$$\tau = \frac{r - r_p}{r_v} \tag{1}$$

Bounds 0 and 2 can be ignored since the network only considers states where $\tau > 6$ seconds. Assuming $v_{\text{I}} = 0$ and $a_{\text{lo}} = \infty$, the equations reduce to

$$\text{bound}_4(r, h, w, v_{\text{lo}}) \equiv w r_v h < w v_{\text{lo}}(r - r_p) - \frac{r_v v_{\text{lo}}^2}{2a_{\text{lo}}} - r_v h_p \tag{2}$$

$$\equiv w r_v h < w v_{\text{lo}}(r - r_p) - r_v h_p \tag{3}$$

$$\equiv w h < w v_{\text{lo}} \tau - h_p \tag{4}$$

$$\tag{5}$$

3 Marabou bounds

The safe region can be defined as

$$\Omega_{\text{safe}}(h, \tau, w, v_{\text{lo}}) \equiv w h < w v_{\text{lo}} \tau - h_p \tag{6}$$

Marabou will search the region outside the safe region to see if an advisory is ever issued outside of its safe region. If a satisfying point is found, then an unsafe advisory is found. Otherwise Marabou returns UNSAT, and we know that the advisory is only given in its safe region.

We can define the unsafe region we need to check as:

$$\Omega_{\text{unsafe}}(h, \tau, w, v_{\text{lo}}) \equiv \neg \Omega_{\text{safe}}(h, \tau, w, v_{\text{lo}}) \tag{7}$$

$$\equiv w h \geq w v_{\text{lo}} \tau - h_p \tag{8}$$

Equation (7) imposes a linear bound involving two of the state variables, h and τ. We also have additional bounds on the state variables:

$$-8000\,\text{ft} \leq h \leq 8000\,\text{ft} \tag{9}$$

$$w v_{\text{lo}} \leq w v \tag{10}$$

$$v_{\text{I}} = 0\,\text{ft/s} \tag{11}$$

$$6\,\text{s} \leq \tau \leq 40\,\text{s} \tag{12}$$

However, the networks were trained with normalized inputs rather than the original state variables variables. The normalization ensures that the training data is zero mean and unit range for each input, which helps the network to train more quickly. The relationship between the variables X and their normalized values $\bar{X}$ takes the form

$$\bar{X} = (X - \mu_X)/R_X \tag{13}$$

$$X = R_X \bar{X} + \mu_X \tag{14}$$

Table 2: Neural network normalization constants

Variable	μ	R
h	0 ft	16 000 ft
v	0 ft/s	200 ft/s
v_I	0 ft/s	200 ft/s
τ	23.8421 s	34 s

where μ_X is the mean value of X and R_X is the range of values for X. These normalization values are

Substituting in the above variable bound equations yields neural network input bounds

$$-0.5 \leq \bar{h} \leq 0.5 \tag{15}$$

$$\frac{wv_{lo}}{R_v} \leq w\bar{v} \tag{16}$$

$$\bar{v}_I = 0.0 \tag{17}$$

$$\frac{6\,\text{s} - \mu_\tau}{R_\tau} \leq \bar{\tau} \leq \frac{40\,\text{s} - \mu_\tau}{R_\tau} \tag{18}$$

$$-wR_h\bar{h} + v_{lo}wR_\tau\bar{\tau} \leq h_p - v_{lo}w\mu_\tau \tag{19}$$

Note that these equations have been simplified slightly by removing μ_h and μ_v since their values are 0. Equation (19) defines a hyperplane bound of the form $a^T x <= b$ with

$$x = [\bar{h}, \bar{\tau}]$$
$$a = [-wR_h, v_{lo}wR_\tau]$$
$$b = h_p - v_{lo}w\mu_\tau$$

If $v_{lo} = 0$, then Eq. (19) simplifies to

$$-wR_h\bar{h} \leq h_p \tag{20}$$

For each advisory, we can impose these constraints and search for points where the network would issue the advisory. If no points are found, then we are guarantee that the network is safe under the assumptions made (no pilot delay, instantaneous advisory compliance, intruder maintains steady level flight).