Examples

Synthesis of Collision Avoidance Protocol

Problem statement

In this scenario we have n UAVs, m altitude layers and q locations of interest. A UAV can ascend or descend to the altitude layer above or below its current altitude layer. A location of interest is a predefined point on an altitude layer. Our aim is to automatically synthesize a control protocol that guarantees the following:

1. Each UAV is able to visit, infinitely often, all of the locations of interest.
2. No Collision between UAVs.

Approach

Our approach to this problem is centered around the idea of UAVs' operating regions. An operating region for a UAV is a polygon on the UAV's current altitude layer. We assume That the UAV will only fly within its operating region. With this assumption, we can guarantee that no collision will happen if the UAV with intersected operating region remain still until the intersection is resolved.

Problem setup

We model this problem by assigning three output variables $l_i \in L$ where L is the set of indices of the altitude layers, $p_i, t_i \in P$ where P is the set of indices of locations of interest, and $(n-1)$ input variables $c_{ij} \in B$ where B is the boolean set and $c_{ij} = c_{ji}$ to each UAV$_i$ where $i, j \in U$, the set of indices for the UAVs. The variables are defined as follows:

- **l_i:** the altitude layer for UAV$_i$ to ascend or descend to.
- **p_i:** the location of interest for UAV$_i$ to head to.
- **t_i:** the intention of UAV$_i$.
- **c_{ij}:** true if the operating region for UAV$_i$ intersects with that of UAV$_j$ and false otherwise.

At each time instance, the synthesized controller takes the variables c_{ij} as inputs and outputs l_i, g_i, t_i for each UAV.

System specifications

- UAV$_i$ can only go to the locations that are in the same altitude layer
  - $(\bigwedge_{i \in U} \square(l_i = x \implies p_i = x \bigvee_{i \in P} p_i = stay))$
- UAV$_i$ and UAV$_j$ must not be at the same altitude layer if c_{ij} is true (their operating regions intersect).
  - $(\bigwedge_{i \in U} \bigwedge_{j \in U} \square(c_{ij} \implies l_i \neq l_j))$
- If UAV$_i$ is given a command to go to x then it must have signaled its intent at the previous step.
  - $(\bigwedge_{i \in U} \square(p_i = x \implies t_i = x))$
- UAV$_i$ will eventually be sent the command to go to x
  - $(\bigwedge_{i \in U} \square\Diamond(p_i = x))$

Environment assumptions

- We assume that if two UAVs have intersecting operating regions and one of the UAVs signaled an intention to go to a location then the intersection will be resolved.
  - $(\square\Diamond \bigwedge_{i \in U} \bigwedge_{j \in U} (\neg c_{ij}))$

Demo

`figs/ca.png`

Figure 1:

Synthesis of VIP Escort Protocol

Problem statement

In this scenario we have one main UAV we call "VIP", multiple "escort" UAVs, and multiple pre-defined locations of interest. Our aim is to automatically synthesize a control protocol that guarantees the following:

1. The VIP must not move from one location to another without being escorted by one of the escorts.

2. The VIP cannot visit certain locations until at least one of the escorts have inspected the location within the last x steps.
3. All the UAVs must go through a certain route to the location of interest. This is needed to prevent going through unsafe zones.

Problem setup

We model this problem by assigning two output variables $g_i \in P$ where P is the set of location indices and $t_i \in B$ where B is the boolean set and one input variable $s_i \in P$ to each escort UAV$_i$ where $i \in U$, the set of escort UAV indices. As for the VIP we assign one input variable $p \in P$ and one output variable $q \in P$. In addition, we assign a variable $r_i \in B$ where $i \in P$. The variables are defined as follows:

- g_i: the location for UAV$_i$ to go to.
- t_i: if true UAV$_i$ will start following the VIP.
- s_i: the current location of UAV$_i$.
- p: the current location of UAV$_i$.
- q: the location for UAV$_i$ to go to.
- r_i: true if location$_i$ was inspected in the past.

System specifications

- The escort must be at the same location as the VIP to be able to follow it.
- The VIP cannot move from a location to another without being escorted by another UAV.
- Mission commands:
  - The VIP must always eventually visits a set of locations.
  - A set of locations must eventually be inspected.

Environment assumptions

- The location is considered inspected when an escort visits the location.
- An inspected location remains inspected. It is also possible to set an expiration time.
- All UAVs will reach their destination they were commanded to go to by the next time step.

Demo

figs/vip.png

Figure 2: