Definitions of important terms in Direct-To algorithm

October 10, 2000
Michelle Eshow

Flight Plan

A flight plan provides information about a flight and the aircraft that will fly it; it indicates the intention of the aircraft. However, it does not enable one to completely specify the 4-dimensional path that the aircraft will fly. Some but not all of the information in a flight plan is necessary but not sufficient to determine a path. For example, the flight plan does not specify the ending altitude of a route. Also, the tokens in the flight plan are meaningless without adaptation data to interpret what they mean (eg, where a fix is or what characteristics an aircraft type implies).

The flight plan for a flight is received by CTAS anywhere from 1/2 hour to several hours before its track information begins to arrive. That is, the flight plan is received before the aircraft enters the Center where CTAS is running.

When CTAS is first brought up, it receives all of the flight plans currently in the Center's Host computer. Some number of these plans will be for aircraft that are already being tracked. CTAS processes the flight plan before processing the first track for any aircraft, so this does not pose a problem.

Any number of updated flight plans, called flight plan amendments, can be received from the time the original flight plan arrives until the last track information is received. Each flight plan amendment contains instructions to change one or more fields of the flight plan. The amended fields of the CTAS flight plan are overwritten with the updated fields. No history of the previous fields is maintained in the system (although they are recorded in the data recording system).

In general, all fields of a flight plan can be amended, including the aircraft flight identifier. CTAS has logic to uniquely associate a flight plan amendment with a flight that it already knows about.

Within a few minutes to an hour after CTAS receives the last track information for the flight, it receives a message to delete the flight plan. This causes CTAS to delete the aircraft and all data associated with it from its internal memory.

Below are the primary flight plan fields as received from the Host. CTAS adds derived fields to the internal flight plan object -- only those important to Direct-To are shown here.

Field	Definition	Example	Required for Trajectory?
Airline and flight number	Flight identifier (non-unique)	AAL108	no
Host computer identifier	Host's id number for this aircraft	817	no
Aircraft Type	Code indicating model	B757	yes
Aircraft equipage code	Auto-pilot sophistication	/G	no
Beacon code	Radar transponder id	0245	no
Assigned altitude	Planned cruise altitude/100	370	yes
True airspeed	Planned speed in knots	280	Before 1st track
Departure fix	Origination	SFO	yes
Coordination fix	Where aircraft will first appear in this Center	TXO	Before 1st track

Coordination time	Time at which it will appear, GMT	1335	Before 1st track
Status	Type of flight plan	Proposed	Before 1st track
Route		See following	yes
Category	Type of route (derived field)	Arrival, Departure, Overflight, etc	yes

Flight Plan Route

A flight plan route is a sequence of instructions that indicate the horizontal path that the aircraft will fly. That is, it can be parsed to give a series of fixes. The aircraft will fly from one fix to the next, in order, and in as straight a fashion as possible.

(Note that this rule only holds in a Center. The TRACON portion of the flight plan does not give an accurate indication of the fixes that the aircraft will fly through, because controllers determine those fixes dynamically based on the traffic situation.)

Also, for two fixes very far apart (say over 800 miles), the aircraft should not be expected to fly a straight path when displayed on a planar map. Rather it would fly a "great circle" route, which represents the shortest path over the curved surface of the earth. Since CTAS thinks the earth is flat, this consideration is not present, but it does limit the length of Direct-To paths that can be generated accurately.

The rules for parsing the route into a series of fixes are very complex. The Host does its own parsing, and CTAS receives that parsed list of fixes. However, the list only spans fixes in the current Center, because the Host only has detailed adaptation data for its own Center. For this and other reasons, the Host's parsed route can't be used for Direct-To. CTAS has its own stripped-down version of Host route parsing. This is done via a library call. The output of this flight plan route parsing is a sequence of horizontal route segments whose endpoints represent the fixes in the planned route of flight. This processing should probably be considered outside the scope of the re-design.

In addition, each time a track update is received, CTAS builds the horizontal path from the current position through the remainder of the planned route. This is known as route generation, and it produces a sequence or horizontal route segments similar to the flight plan route parsing. This processing is also done in a library that could be considered outside the scope of the re-design, or not.

In any case, the flight plan route can be considered a series of tokens to be interpreted:

Fix
Route
. (single dot)
.. (double dot)
/ (slash)
\ (backslash)

For example, here are some real flight plans routes:

SJC./.TXO..SJT.CUGAR6.IAH
DFW.DALL5.TXK..LIT.J131.PXV.J29.JHW.J82.ALB.GDM2.BOS
LAS./.3441N/10332W..DFW
PWA..IRW..IRW263039..GCC

Rules for interpreting tokens:

- 4 letters + 1 number: SID or STAR (TRACON route)
- 3 letters: fix (also indicates VOR nav aid)
- 5 letters: fix that is an intersection between 2 jet routes with no nav aid
- 3 letters with nnnmmm: fix on the nnn radial from 3-letter fix, mmm miles away (IRW263039)
- nnnnN/nnnnnW (N and W optional): fix that is a longitude/latitude point (3856N/11726W)
- nnn\nnn : fix that is an x\y location in Center (245\469)
- J + number: jet route (V + number: victor/vfr route)
- ".." separates 2 like elements (ABC..DEF)
- "." separates 2 un-like elements (ABC.J52)
- "./." indicates some part of the route was left out because the aircraft already passed those segments

Trajectory

A trajectory is a 4-dimensional path (in space and time) that satisfies a set of inputs and constraints. The Trajectory Synthesizer (TS) module computes trajectories based on accepted kinematics of flight using 4 degrees of freedom (x, y, z, heading).

In order to generate a trajectory, TS needs the following inputs:

- Weather data, which it reads from a file based on a file name passed in from the Direct-To process
- Aircraft model data for all known aircraft types, which it reads in at start-up
- A variety of data passed in from the client process. The important items when D2 is the client are:
  - Desired horizontal path. The horizontal path is a series of waypoints through which the aircraft should fly. D2 attaches to each waypoint a mode that is a series of bits giving TS instructions on how to make the turn around the waypoint, and whether or not it is the waypoint where the end constraints should be applied.
  - Initial track data of the aircraft (altitude, ground speed, heading, x, y)
  - Optional command to capture a new altitude
  - Optional command to capture a new airspeed
  - Desired ending altitude
  - Desired ending airspeed

In a pure cruise trajectory, in which the aircraft is already flying at its cleared altitude, and the flight plan extends beyond the limit where D2 would compute the descent, the ending altitude and speed are set the same as the initial. Note that most of the logic for setting the constraint speeds and altitudes is in a library and may be outside of the scope of the re-design.

The trajectory as computed by the TS consists of the following elements:

- Several sequences of data that represent the segments of the processed horizontal path, including: path distance, heading, total turn angle, turn radius, x/y location. Segments are not of uniform length. Whenever there is a turn, TS starts a new segment.
- Several sequences of data that represent the segments of the processed vertical path, including: time, path distance, altitude, ground speed, airspeed. Vertical segments are not of uniform length. Whenever a new vertical parameter needs to be varied, a new segment is started, eg, when transitioning from fixed engine control and vertical speed to fixed cruise speed and flight path angle. Note that vertical segments and horizontal segments are not coincident in path distance or time.
- Various flags indicating whether there was a climb or descent, whether the course is curved, etc.

- Some data that is necessary to form an advisory is returned with the trajectory. The advisories give an indication of what the aircraft should be instructed to do to achieve the trajectory. Some of these data re-state the inputs that were passed in, but may reflect unit conversions and provide more detail. They include: new cruise speed, new climb speed, new descent speed, location and time of top of climb and/or descent, wind-corrected heading of the first horizontal segment, etc.

The above data is good for extracting advisories that controllers might issue. However, the path description is difficult to use for conflict analysis because it is split into the disparate horizontal and vertical components. To address this, there is a library function that combines the horizontal and vertical components into a unified sequence of segments through space. The segments are 10 seconds long, and for each the following information is generated in the library call: time, x, y, altitude, path distance, ground speed, airspeed, heading. For the purposes of the D2 algorithm, each 10 second segment can be considered a point. That is, if segments from 2 aircraft trajectories intersect in time and space, then that would be a prediction of collision of the aircraft.

Trajectories are generated for each aircraft as new tracks are received, or approximately every 12 seconds.

Advisory

In general, an advisory is a suggestion to a controller on how to meet the CTAS plan. An advisory can be a scheduled time to cross a certain fix or a runway assignment, for example. Advisories arise in D2 for 2 reasons: 1) it is helpful to the controller to be able to tell the pilot what heading to fly to get to the D2 fix, and 2) there is a lot of legacy functionality in the D2 module for another tool, Descent Advisor, whose intent is to tell controllers/pilots how to achieve a given arrival trajectory to the meter fix.

So for the purposes of D2, we can say that the only advisory of interest is the heading that the aircraft should fly to the next fix in the route, whether it is a D2 fix or not. This heading is pulled from the Trajectory.

The remaining uses of Trajectory-based advisories in the D2 module should probably be ignored, but it would be nice if the design could be extended to handle them easily.