MTR 99W000011

MITRE TECHNICAL REPORT

Integrating The Flight Management System With Air Traffic Control Functions

The Concept of Path Objects

July 1999

Dr. John N. Barrer

Dept. No.: F082

Project No.: 02MSR055-AF

MITRE

Center for Advanced Aviation System Development
McLean, Virginia

MITRE Department Approval: [signed]

Melvin J. Zeltser

MITRE Project Approval: [signed]

Dr. Glenn F. Roberts

Abstract

The concept of path objects (POs) is to store in aircrafts' flight management computers the instructions on how to create the shapes of frequently flown flight paths. The instructions for creating these paths are called path objects (POs). A particular PO is the definition of a shape, independent of its location. A specific aircraft's trajectory can then be specified by setting a few parameters and the starting and ending points of the PO, using universal latitude/longitude/altitude coordinates. This paper develops the concept, specifies a basic set of POs, and discusses many applications to air traffic control (ATC) situations.

The PO concept offers a common language that can compactly express complete information about the aircraft's intended path, which in turn offers potential benefits to both the ground control system and the aircraft operator. Using POs, most aircraft flight paths can be expressed with only the path's name and its starting and ending points. By simplifying the expression of flight paths, the PO concept potentially solves many practical problems related to ATC and offers increased flexibility to ATC functions. It addresses the issues of the integrity of aircraft intent information, the maintenance of navigation databases, the commonality of procedures for the pilot, the charting and cockpit display of information, and the efficiency of transmission.

Implementation of the concept requires that PO processing be incorporated into the aircraft's flight management computer system and into the ground ATC systems. However, once both sides of the system are aware of POs, any manufacturer's flight management system (FMS) that contains POs can communicate with any ATC system that also uses POs. This means that the development of air traffic management (ATM) functionality can proceed independently of FMS development. Any civil aviation system in the world can independently develop its proprietary ATM functions and incorporate the capabilities of the FMS by communicating in the language of POs. While the use of POs does not require data link (thereby allowing a transition from today's voice communication system to a full data-link system), the use of data link would enable the use of more complex POs.

KEYWORDS: Path Objects, Path Object Processing, Flight Management Systems, Air Traffic Control

Acknowledgments

The author would like to acknowledge the contributions made by George W. Flathers who initiated our search for a way to standardize routes in the terminal area. Walter Scales was the principal investigator on the internal MITRE Sponsored Research project that funded the development of this concept. Jerome Freedman and Joyce Forman contributed to the section on ATC benefits. Jonathan Schwartz made many useful suggestions concerning tactical functions that could benefit from POs. The document benefited from the comments of our internal peer reviewer, Vince Massimini.

Table of Contents

List of Figures

Section 1

Introduction

1.1 Background

The primary motivation for this report was the desire to bring the Flight Management System (FMS) into the Air Traffic Control (ATC) process to simplify the task of maintaining and using standard terminal area arrival routes (called STARs) when the Global Positioning System (GPS) becomes more widely available. With improvements in airborne computers, future FMSs, and ground ATC automation systems, it will be possible to store the information necessary to fly many different precise paths. Currently, the FMS is used to store paths for selected routes and terminal areas by storing each path as a series of waypoints. Consequently, a path near one airport may be nearly identical in shape to one near another airport, but the FMS treats these as two distinct sets of waypoints. Furthermore, when the aircraft arrives in the terminal area, its position determination system is currently based on the location of local navigation aids (navaids). This means, for example, that an aircraft, flying nonstop from Washington to Moscow, must carry with it all the waypoints for Moscow (and all points in between) as it departs Washington. The task of maintaining this database is enormous.

The initial question was how could this process be simplified by developing a standard set of terminal area routes for future systems. The first idea that came to mind was that of extending the idea of STARs, so that they would be based on GPS (rather than ground-based navaids) and using one set of STARs for all airports. However, current FMSs contain a fairly large and complex database, which raised concerns about making this database any larger. This quickly evolved to the idea of path objects (POs). This notion views flight paths as composed of segments of different shapes, each with a beginning, an end, and perhaps a *shape* parameter. Once the geographic coordinates of the two endpoints and the value of the shape parameter are specified, that segment of the route is determined. An aircraft's computer only has to store the definition of the shapes. Then, using the stored algorithm, it can construct the intermediate waypoints based on the two endpoints and pass those to the FMS, which would control the aircraft's flight path. In future generations of FMSs, the task would be fully integrated within the FMS.

In developing this concept, the author heard from a former controller who related an anecdote about early uses of this idea. In the early 1970s, controllers established a convention with pilots who were flying over Arizona into Los Angeles. It was understood that if the controller told the aircraft to "delay 30 degrees (deg) left," that the pilot would turn 30 deg left, fly for 1 minute, turn 60 deg to the right, fly for 2 minutes, turn 30 deg left, and rejoin the original course. So, the concept of having an unambiguous, shorthand way of communicating standard maneuvers is not new and could be greatly improved by having such path segments stored in the FMS.

The PO concept described in this report has evolved from the initial goal of simplifying the use of the FMS in terminal area operations to one that seeks to integrate the FMS with the ATC system for the purpose of improving the integrity of aircraft intent information that is exchanged between the aircraft and the ground. The use of POs permits the controller and pilot to exchange intent information reliably, unambiguously, and efficiently in a voice or data-link environment. In so doing, it helps provide a transition path to full ATC automation while aircraft are partially equipped with data link. In a data-link environment, it reduces the amount of information that has to be exchanged to maintain the integrity of intent information.

1.2 Purpose

This report is designed to define the concept of POs by providing a technical definition and then to explore a number of practical applications. It proposes a basic set of POs that would be incorporated into the FMS and ground systems.

1.3 Scope

The paper will concentrate on the definition of the concept of a PO and develop a set of basic objects. It provides a few examples to illustrate potential applications of the idea to ATC functions.

Section 2 describes path objects and defines a set of basic path objects. This section also describes *path object processing*, which is the mechanism for using POs.

Section 3 discusses application of POs to ATC problems. It also discusses the additional benefits to the maintenance of the navigation database.

Section 2

The Definition of Path Objects

2.1 The Path Object Concept

As a simple two-dimensional analogy, imagine a piece of stiff wire (analogous to a STAR) representing the aircraft's path over the ground, as shown in Figure 2-1. This path is an object that can be completely defined with respect to its endpoints, a single parameter, and a convenient coordinate system (such as GPS latitude/longitude). If the shape of the wire (path) is stored relative to its endpoints, then it can be located anywhere in the world by merely specifying the parameter, the endpoint coordinates, and the name of the object.

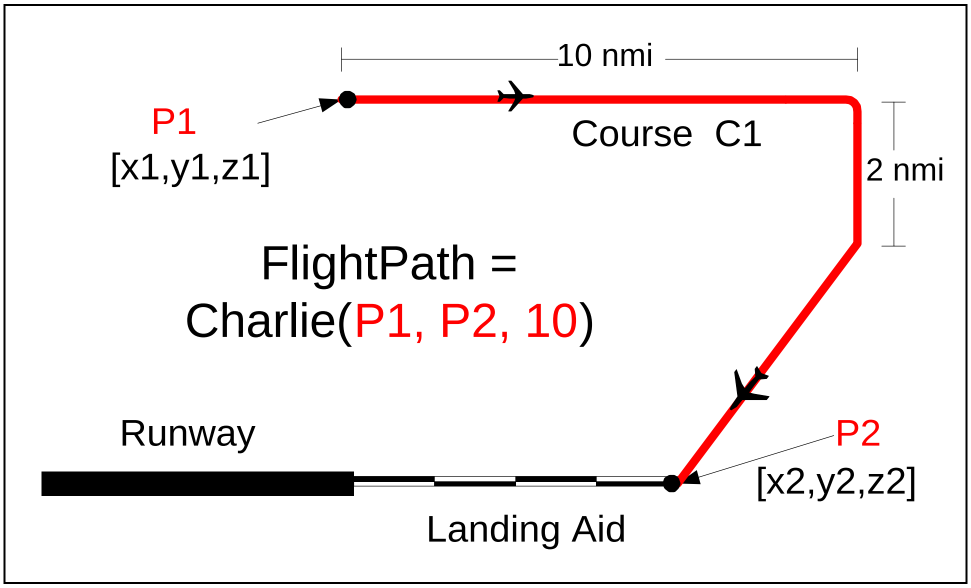


Figure 2-1. Hypothetical Example of a Stored Path

Figure 2-1 illustrates a hypothetical path that begins at point 1 (**P1**)[1], follows course C1 (calculated below based on **P1** and **P2**) for 10 nautical miles (nmi), makes one 90-deg turn, travels 2 nmi, and then turns to, and ends at point **P2**. It allows an altitude transition from z1

---

[1] The notation **P1** in bold letters will be used to indicate a triplet of the form (latitude, longitude, barometric altitude = (x1,y1,z1).

to z2 at the pilot's discretion. In this simple example, the PO (stored instructions on how to construct the path, given the input parameters) resides in a computer as path type *Charlie*. It is only necessary to specify, **P1** = [x1, y1, z1], **P2** = [x2, y2, z2] and **D =** 10, in order to completely define this path with the same precision, anywhere in the world. To define that path using waypoints would take at least four waypoints.

The instructions for *Charlie* **might** be something like this, based on Figure 2-2:

VALIDITY CHECK:

If the distance between **P1** and **P2** is less than **D** nmi or greater than (3×**D**) nmi, reject the request.

CALCULATE:

The initial course, C1, is the course between **P1** and **P2**, minus 45 deg.

FLIGHT PATH:

Fly along initial course C1, **D** nmi,

Turn right to course C1+90 deg and fly for 2 nmi, calculate the course to **P2**, and

Turn toward and fly to **P2** along that course.

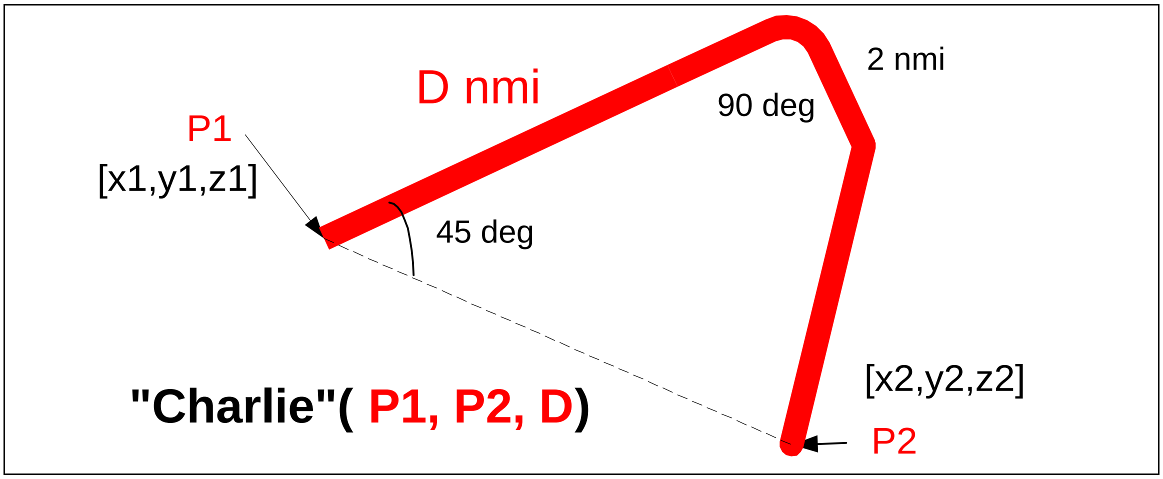


Figure 2-2. Calculations for *Charlie*

As seen in Figure 2-2, the shape *Charlie* is fairly rigid. The values of 2 nmi and 45 deg are stored as part of the definition of *Charlie*. By defining *Charlie* in such a rigid way, it reduces the amount of information necessary to specify a particular instance of *Charlie*, but it also reduces the range of shapes that are possible.

Having a compact notation is of use in an ATC application as shown in Figure 2-3 below. In this example, if there were two named geographical fixes, e.g., JUNNS and TANGY, the air traffic controller could instruct the pilot of a PO-equipped aircraft to "Fly object *Charlie* from JUNNS to TANGY with a 15-nmi leg." This means that, even in a voice communication environment, the ATC system can keep the FMS in the process of adhering to clearances, thus reducing the workload for both the controller and pilot.

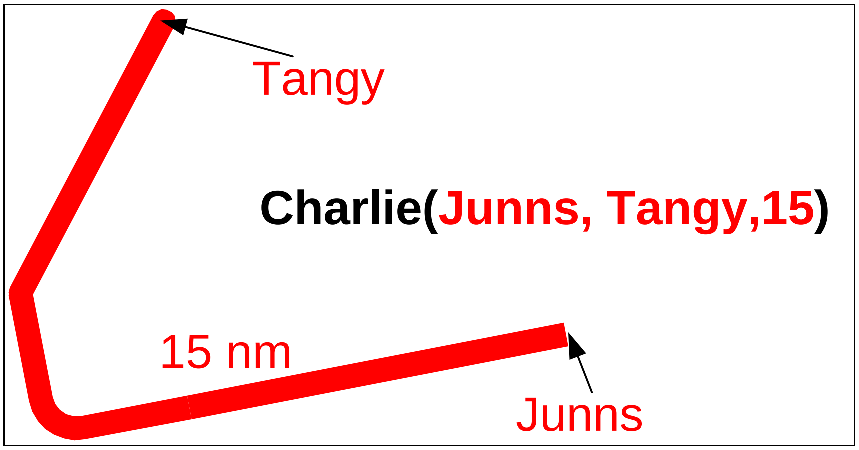


Figure 2-3. ATC Application in Voice Environment

2.1.1 Path Objects and the Path Object Processor

The POs are paths in space that are independent of the locations of the aircraft or the ATC system using them. POs can either be very rigid like a piece of stiff wire, or they can have a parameter associated with them that allows the user to create objects that are fundamentally alike, but different in size or curvature.

There must be a path object processor (POP) that would convert POs to the form required by the specific user. For example, there would be a POP in the aircraft that receives path object requests and converts them to waypoints that the FMS can use to calculate a flight path. The POP must also evaluate if a given path object is feasible for the user. For example, it may be possible for a helicopter to fly a path that is infeasible for a large jet transport aircraft. In addition, there may be certain sequences of POs that are incompatible, and the POP must be able to detect them and prevent them from causing an infeasible path.

It is envisioned that the POP system would have an airborne component and a ground component. It is anticipated that, initially, the airborne POP would be a separate device that interfaces with the existing FMS systems. The airborne unit would transform the POs, using the stored instructions, to flight path coordinates (waypoints) that could then be flown by the FMS. Eventually, the POP would become an integral part of the FMS. Also packaged with the POP would be the interface device for the pilot. The pilot needs a means to display POs, modify their parameters, select them, and confirm (to ATC) which PO the aircraft is following.

The ground-based POP unit would convert POs to radar flight data information in Flight Data Processing format. Packaged with that would be the device that the controller could use to display POs, modify their parameters, select them, and transmit (voice or data link) the path object to the aircraft. Figure 2-4 illustrates the flow of information using the POP interface.

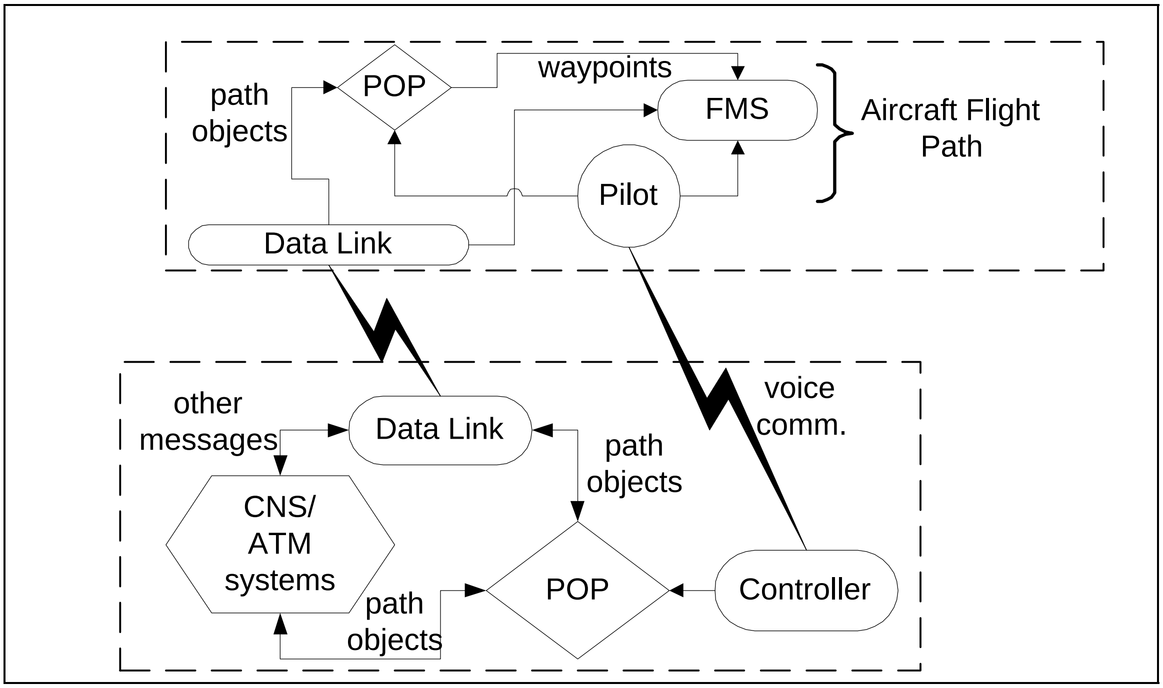


Figure 2-4. Information Flow for Path Objects

2.1.2 Using Path Objects for ATC

Air Traffic Management (ATM) functions would rely on the POP to reliably transfer complex flight paths in the terminal and en route domains. When there is a need for a path change, the revised path would be communicated to the aircraft (via data link or voice communication) using PO terminology.

Either the controller or CNS/ATM system will determine the desired ATC objective and develop the flight profile for a single aircraft or group of aircraft that meets the objective. The desired flight path(s) would then be converted to path object notation by the POP and sent to the aircraft via voice or data link. The PO would then be entered into the POP in the aircraft and the result presented to the pilot for approval and verification. Once the new flight path is accepted, the information would be passed to the FMS for execution.

In a free flight scenario, the pilot may initiate the request for a flight path change. In a similar fashion, the aircraft's POP would be used to generate the new path object that would be transmitted to the ground controller or automation system. The ground would perform the necessary functions such as separation assurance and traffic flow management before approving the request.

2.2 Precise Definitions of Path Objects

The future vision of this concept is that an onboard computer would store the definitions of the POs and interface with (or preferably be totally integrated with) the FMS, providing it the necessary information to perform the maneuver. Some POs would be flexible with associated parameters, and others would be inflexible and completely determined by just their two endpoints. An inflexible path object is derived from a flexible path object by specifying a value for each of the parameters. There may be hundreds of such inflexible POs, but at this stage of the development it is believed there would be no more than 50 flexible POs. Unlike current routes, POs do not require updating so there can be a large number of inflexible paths, if necessary, without creating a maintenance burden.

There would also be a database containing the values of the default parameters for specific airports, terminal areas, and other sections of airspace. For example, a STAR at one particular airport may require the use of path object "T32" with starting point (x1, y1, z1), ending point (x2, y2, z2), and D1 = 10.4 nmi. Another airport may also use T32, but with different parameters. The database would contain the mapping between the airport's procedures and the POs, together with default parameters. As with today's system, this database would require regular updates, but the complexity and size would be greatly reduced through the use of POs. This is because it currently takes several waypoints to define flight paths that can be defined with POs using only the two endpoints.

2.2.1 Flexible Path Objects

The basic flexible POs are *line segment*, *turn*, and *course reversal*. Most POs have at least three parameters: a label, a (three-dimensional) beginning point, and an ending point. These are always the first three parameters in the encoding.

All points are three dimensional of the form (latitude, longitude, barometric altitude). Because altitudes can be optional, hard requirements, or bounds, an additional character is required to describe the nature of the altitude. Therefore, altitudes are designated:

- *o* (optional)
- *e* (exact)
- *a* (at or above)
- *b* (at or below)

In a voice communications environment, the points P1 and P2 would be named fixes stored in the geographical database.

2.2.1.1 Line Segment

The simplest PO is a segment of a straight line (technically a great circle), defined by its end points as shown in Figure 2-5, below. In fact, an FMS is currently programmed to fly such a path. The added value of the PO concept is that the coordinates are 3-D and based on latitude and longitude.

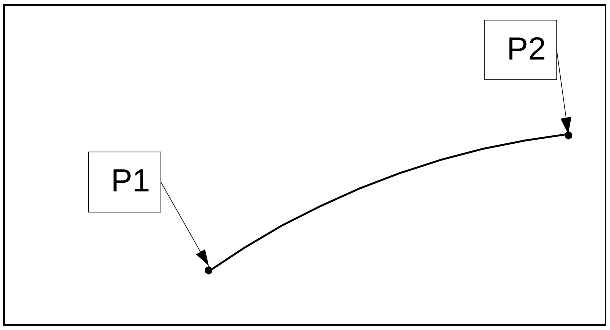


Figure 2-5. Great Circle Segment

If this path is called “L1”, then it only needs two points to implement it. For example, the notation,

[L1, **P1**, **P2**],

where **P1** = (x1,y1,z1) and **P2** = (x2,y2,z2) completely define the path. The PO in this case consists of the instructions (stored in the computer under the label ‘L1’), and the two parameters, **P1** and **P2**. The distance from **P1** to **P2** is calculated by the relationship,

$$d = \sqrt{|P1|^2 + |P2|^2 - 2 \times P1 \bullet P2}$$

However, any algorithm that produces the same path is acceptable. The philosophy for developing POs is that that there should be alternative representations for the same path so that the ATC application can choose the one that is most convenient. An alternative way to express this is in the form of a course, “a,” from **P1** for a distance “d,” with altitude change “c.”

[L2,**P1**,a,d,c].

The instructions in the POP would be to fly from **P1** along course “a” for a distance “d,” and change altitude by an amount “c.” There is a functional relationship between L1 and L2, so that the same path can be expressed in either notation.

2.2.1.2 Turns

The next simplest PO is a ‘turn’ as depicted in Figure 2-6. There are several ways this PO could be specified.

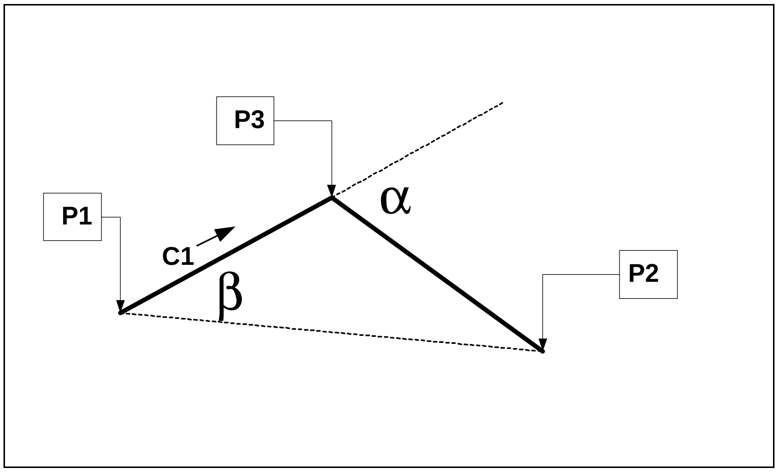


Figure 2-6. “Turn” Path Objects T1 and T2

It would be possible to specify **P1**, **P3**, and **P2**, such as [T1, **P1, P2**, **P3**]. The instructions in the computer would then turn the aircraft at **P3** with a sufficient angle to fly directly (great circle) to **P2**.

Another method would be to specify **P1**, the angle β, **P2**, and the angle, α.

[T2, **P1, P2**, β, α]

The angle, α, could be within +/- 90 deg. The FMS would then fly from **P1** along the course C1 (calculated from the course **P1**-**P2** and β) until the relative angle between the aircraft and **P2** was α, at which point it would turn toward **P2**. The choice of definition T1 or T2 would depend on whether the user wanted additional altitude control at point **P3**.

Analogous to the course-distance representation of a line where the altitude is constant, a turn could be specified by providing a starting point, **P1**, an initial course α, for a distance d1, and a second course, β, for a distance d2.

[T3, **P1**, α, d1, β, d2].

In situations where the ATC system wanted tighter specification of the area where the aircraft turns, the turn PO could take on other parameters as shown in Figure 2-7.

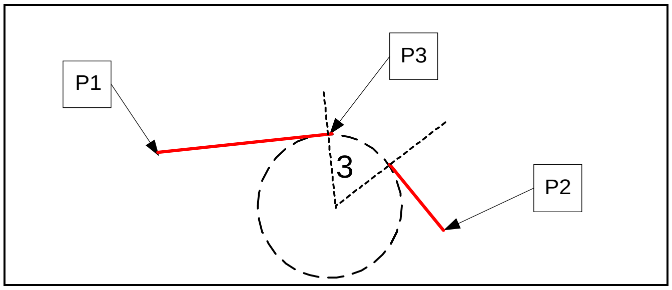


Figure 2-7. "Turn" Path Object T4

This turn PO is specified as [T4, **P1, P2**, **P3**]. The path begins at **P1**, then follows a path to **P3** where the path turns along a (flat earth) circle of radius 3 nmi, with center on the normal to **P1-P3** at **P3**. The center is on the same side of the line **P1-P3** as is **P2**. It then follows the path emanating from **P2** that is tangent to the circle.

If the system wanted control over the turn radius, then that parameter could be added to the specification as shown in Figure 2-8.

[T5, **P1, P2, P3**, r]

The instructions would be to fly from **P1** to **P3**, calculate the center of a circle of radius r, such that the line **P1–P3** is tangent to the circle at **P3** (the center is on the same side of the line **P1–P3** as is **P2)**, fly along the circumference until meeting a tangent line that goes through **P2**, then direct to **P2**.

It may be useful to have just the arc segment of the turn available to construct other POs. A circular arc segment can be represented by the endpoints, **P1** and **P2**, and the diameter of a circle as [T6, **P1, P2**, r] and [T7, **P1, P2**, r]. T6 is a right turn, and T7 is a left turn. These are illustrated in Figures 2-9 and 2-10, respectively. The aircraft begins turning toward **P2** at

P1 along a circle of radius r until it reaches **P2**. However, there is more responsibility on the user to ensure that the turn is compatible with the POs that join it.

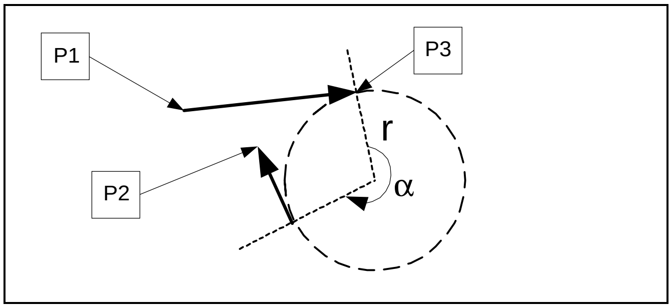


Figure 2-8. Large-Angle Turns

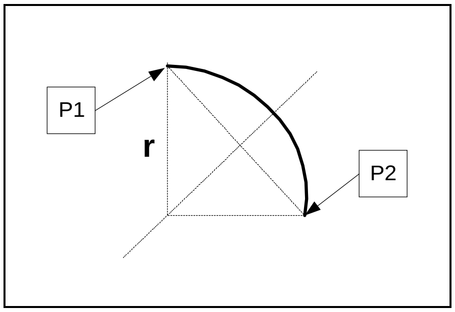


Figure 2-9. T6, Right Turn Segment of a Circle

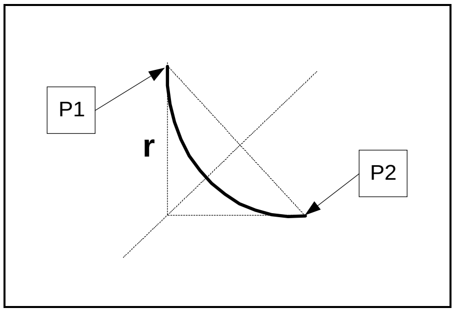


Figure 2-10. T7, Left Turn Segment of a Circle

2.2.1.3 Course Reversal

A course reversal is a combination of a line segment and a turn, so it could be viewed as just two separate POs. However, because the objective is to fly a 180-deg turn, the notation can be made more compact by incorporating that knowledge in the instruction set rather than the parameter set.

A very common course reversal pattern shown in Figure 2-11 is called the *traffic pattern* or *circuit*. It consists of a straight course parallel to the runway, followed by a 180-deg descending turn to a course parallel to the starting course, usually ending at the runway. There are a number of ways of specifying this path. Because the distances are relatively small, a flat-earth model is used. The simplest notation is to specify **P1, P2**, and **P3**. The computer would store instructions to:

- Calculate D1 based on the course **P1**-**P3** and the point **P2**
- Fly from **P1** to **P3**
- Turn toward **P2** and fly a 180-deg turn of diameter D1
- Fly the reverse course to **P2**

It is encoded as [C1,**P1,P2**,**P3**].

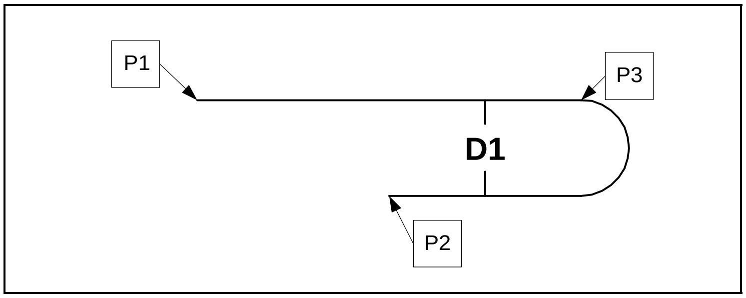


Figure 2-11. C1 Course Reversal (Traffic Pattern)

An alternative way to express a course reversal, shown in Figure 2-12, is to specify the starting and ending points and the initial course:

[C2,**P1,P2**,a]

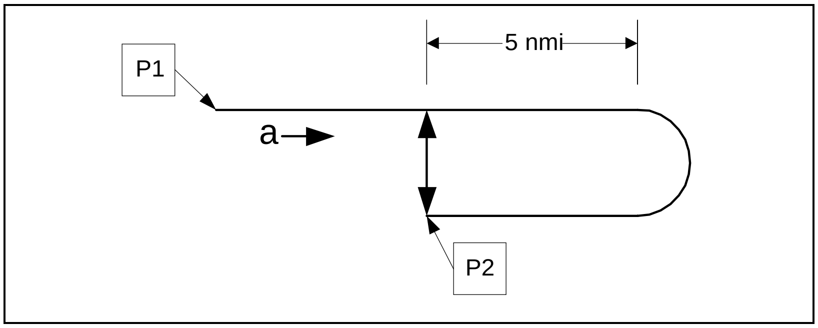


Figure 2-12. Simplified Traffic Pattern

The instructions in the computer would:

- Calculate a course from **P1** along course "a"
- Calculate the perpendicular between the course and the point **P2** and find the intersection
- Continue the course for 5 nmi past perpendicular to **P2**
- Turn 180 deg toward **P2**
- Fly the reverse course to **P2**

As with all POs, the POP must validate the given parameters to ensure that they are feasible. In this case, if **P2** were too close to the initial course, the aircraft could not conduct the maneuver. An extension of this is to make the 5 nmi distance a variable, using

[C3,**P1,P2**,a,d].

2.2.2 Compound Path Objects

Using the POs just defined, the ATC systems of the world and the aircraft can exchange very precise information on the intended path of the aircraft in a very succinct manner. This is then an enabling technology for many of the ATC Traffic Flow Management (TFM) concepts under consideration. In the terminal area, routes that are more complex can be formed by combining elementary shapes.

2.2.2.1 Example of a STAR

For example, a typical STAR would consist of a T1 turn to a course reversal, C2. By creating a compound PO such as S1 below, designers of terminal procedures would have access to precise specifications for more complex arrival patterns. Because the last point of one elementary PO is the starting point for the other, it might be possible to reduce the number of parameters necessary to specify the two POs as one.

This maneuver shown in Figure 2-13 could be expressed by

[S1, **P1**, **P3**, **P2**, a, d],

using the T1 and C3 encodings described above.

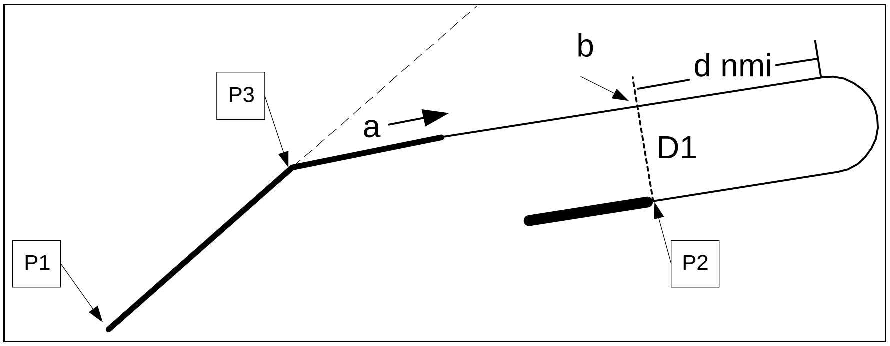


Figure 2-13. Compound Path Object S1, using T1 and C3

The instructions would be to fly from **P1** to **P3**, then along course “a,” d nmi past the intersection with the perpendicular from **P2**, turn 180 deg toward **P2** and fly the reverse course of “a” to **P2**.

2.2.2.2 Holding Pattern

Another useful compound object is the ‘race track’ holding pattern shown in Figure 2-14. This can be expressed as [R1,**P1**,**P2,**a].

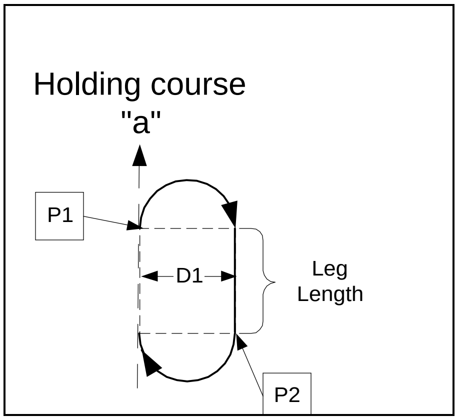


Figure 2-14. Holding Pattern Object

The instructions for R1 calculate the rectangle having corners at **P1** and **P2** and fly two 180-deg course reversals at the ends. The pattern is oriented so that the aircraft is on course “a” as it approaches **P1**.

Many modern flight management systems already have a holding function. The pilot only has to specify the point P1, the course, and the length of the legs. This is done using a keypad entry method. The parameter D1 is not specified and the assumption is that if the aircraft executes a standard rate turn at P1, it would stay within reasonable limits.

2.2.2.3 Procedure Turn

As part of an instrument approach, aircraft are sometimes required to fly outbound from the runway on the approach course and then perform a course reversal maneuver that positions them on a 30-deg intercept angle to the final approach course. Such a maneuver is shown in Figure 2-15. The leg of the procedure turn is set at 2 nmi and the turn radius is set at 1.5 nmi. Consequently, the procedure can be calculated once the endpoints, **P1** and **P2,** are known.

[PT1, **P1**, **P2]**

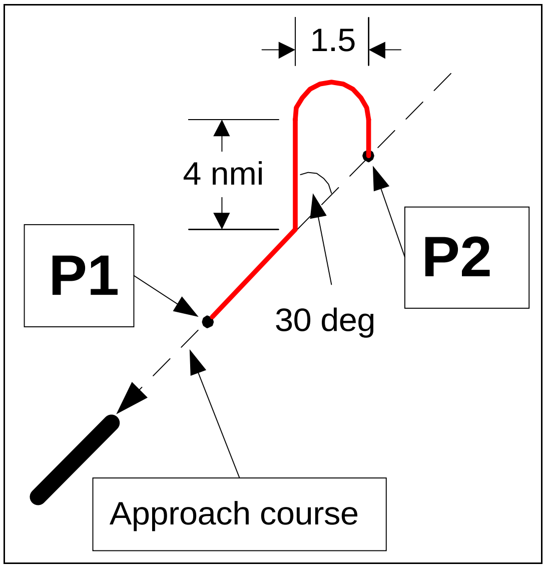


Figure 2-15. Procedure Turn Object

2.2.2.4 Delay Pattern

The following example in Figure 2-16 probably has no practical application. Nevertheless, it illustrates the power of POs to compactly specify a rather complicated maneuver. Four semicircles are concatenated to produce a *figure eight* turn, which is a potential maneuver for delaying the arrival of an aircraft to meet a fix-crossing time. The resulting path adds a distance of (2π X D1) to the path. Knowing the aircraft's speed, the FMS can calculate the induced delay. The line P1-P2 determines the orientation, starting point, ending point, and the diameter D1. So this could be encoded as simply,

[F8,**P1**,**P2**].

The instructions would calculate the distance D1 from **P1**–**P2,** calculate the collinear point **P3**, and use the PO T6 to construct the four semicircles along the line **P1**-**P3**, ending at **P1**.

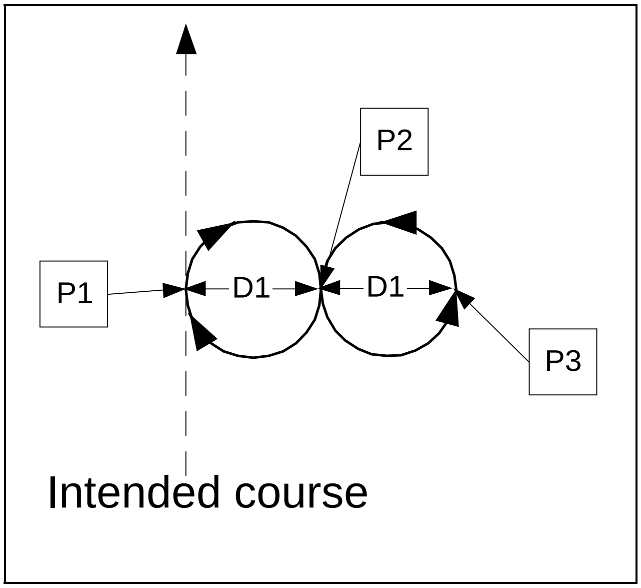


Figure 2-16. ***Figure-Eight*** **Delay Maneuver**

2.2.2.5 Path Stretching Maneuver

Similar to the delay maneuver, but of more practical interest, is the notion of a path stretching PO shown in Figure 2-17. The idea is that an aircraft traveling from P1 along course "C" must insert a *dogleg* maneuver along the course that adds "N" nmi to the path within certain bounds.

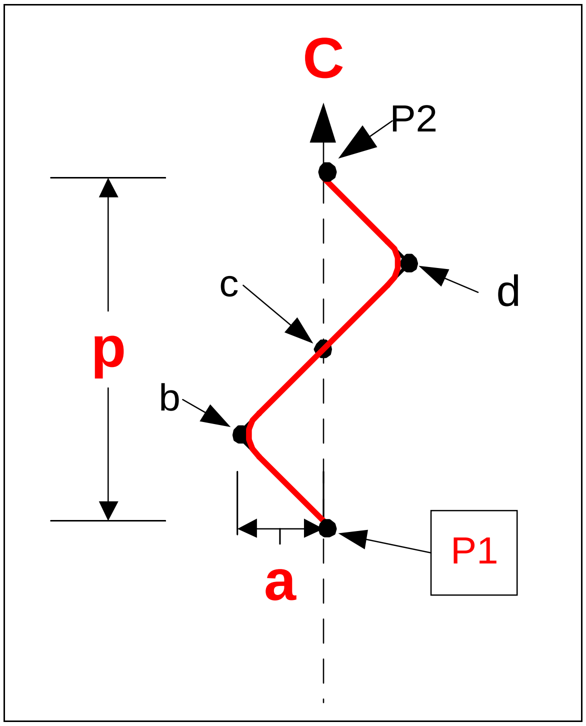


Figure 2-17. Path Stretching Maneuver "PS1"

The notation is [PS1, P1, C, a, p]. The instructions are then to calculate the path of an *S-turn* that has an amplitude of “a” and a period of “p” along the course “C,” starting at P1. The legs P1-b, b-c, c-d and d-P2 are all the same length. The length from P1 to b is

$$\sqrt{a^2 + \left(p^2 / 16\right)}$$

The length L of the path P1-b-c-d-P2 is 4 times that, or:

$$N = \sqrt{(4a)^2 + p^2}$$

Perhaps a more useful way to express this is to specify “N” rather than “a.” The amplitude of the deviations, “a,” would then be calculated from the relationship

$$a = \sqrt{\frac{N^2 - p^2}{16}}$$

The controller could then issue a command such as “Extend your path by N nmi beginning at fix P1 within P nmi along course C.”

2.2.2.6 Traffic Pattern Object

In Figure 2-18, a two-dimensional PO is used to define the standard traffic pattern. The parameters of the PO have meaning in the traffic pattern. For example the dimension, d3, is the length of the final leg, and the dimension d1 is the length of the base leg. If this PO were stored in the FMS, the pilot would only have to enter the dimensions, which are all single-digit entries. Modifying the pattern would then be equally efficient. The figure shows how changes in the parameter d1 affects the width of the pattern. Of course, the same pattern can be created with the course reversal POs defined above, providing increased flexibility. This PO is encoded as [TB1, **P1**, c,d1,d2,d3]. The path is constructed from P1, by having the course line, C, pass by P1 at a distance, d1. From that, the dimension d2 determines the starting point and the distance d3 determines the turn to base and the length of the final approach.

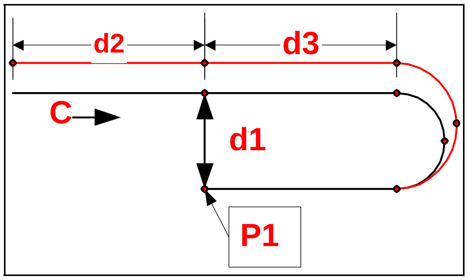


Figure 2-18. Traffic Pattern Path Object

2.2.3 Inflexible Path Objects

Inflexible POs are derived from the flexible POs by fixing the associated parameters so that only the endpoints are needed to determine the exact flight path. For example, in the case of a turn, T2, specified by [T2, **P1, P2**, β, α], the two angles could be set as 45 deg and 30 deg. The resulting path could be stored as T78, where the '78' has no significance. The values for β and α would be stored under the label T78. The user could then refer to the path [T78, **P1, P2**] and the computer would convert that to [T2, **P1, P2**, 45, 30].

Computer memory is inexpensive, so there could be thousands of inflexible paths stored in the POP. For example, for the T2 example above, there could be 81 inflexible paths representing each combination of β and α between 0 and 90 deg in increments of 10 deg. Unlike the entries in the current navigation database, these would not change; therefore, it is feasible to think in terms of having thousands of these rigid definitions stored permanently in the navigation computer. The advantage is that all of the commonly used paths could be specified by just their label and the two endpoints, **P1, P2**.

Since two of the objectives in creating POs are to reduce the effort involved in maintaining the navigation database and to simplify the amount of transmitted data required to specify a change in flight paths, it is worthwhile to have a larger number of POs that are more rigid, but require fewer parameters. For example, the traffic pattern course reversal, C1, could be made more rigid in the following way:

Using the notation

[C1_5, **P1**, **P2],**

the instructions related to Figure 2-19 below, would be to

- Calculate the angle "a" such that "a" is arcsine(5/ **|P1-P2|)**
- Fly from **P1** along the course determined by "a" and **P1-P2** for 5 nmi past the perpendicular between the course and **P2**,
- Turn 180 deg towards **P2**, and fly the reverse course to **P2.**

The designer of the path, naturally, would want the course to be parallel to the runway. This can easily be achieved by choosing **P2** and then calculating the point **P1** that will cause the above set of instructions to produce the desired value of "a."

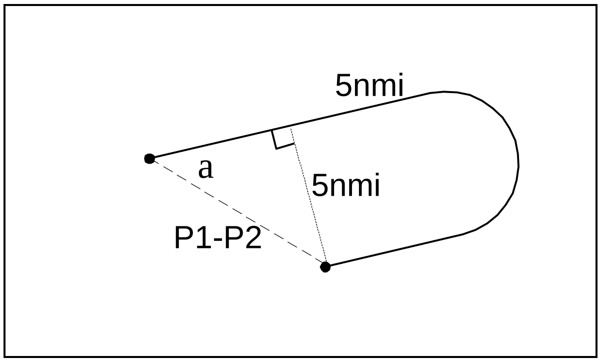


Figure 2-19. Traffic Pattern with 5 nmi Downwind

2.3 Creating a Set of Basic Path Objects

The flexible POs included in this paper form a core set of POs. These are summarized in Appendix A. Even these few cover a tremendous number of actual arrival and departure situations at the world's airports. They vary from a completely rigid PO such as C1_5, to a very flexible shape such as S1. There is an obvious tradeoff between rigidity and the number of parameters required to specify the PO.

When implemented, the POP or FMS would contain the set of flexible POs so that any path could be constructed using a series of flexible paths. In addition, there would be perhaps thousands of inflexible paths that are derived from the flexible paths by fixing one or more of the parameters. The list of inflexible paths may be updated annually, but any path not in the list of inflexible paths could be constructed by specifying the parameter values of the stored flexible PO.

The set of initial POs proposed here can serve as a starting point for any standards committee that takes on the task of developing an international standard for POs.

Section 3

Applications of Path Objects

In this section, several applications of the PO concept are discussed. The primary objectives of the PO concepts are to:

- Improve the reliability and integrity of aircraft intent information
- Improve the efficiency with which intentions are communicated
- Improve the standardization of procedures for pilots
- Simplify the task of maintaining the navigation database

In doing those things, the concept offers advantages to systems that manage and control air traffic.

3.1 Standardized Paths

As discussed in the introduction, one of the main motivations for this work was the desire to bring commonality to Departure Procedures (DPs) and STARs. The term *standardized flight path* is used in this document to describe a flight path that follows a commonly used trajectory taken by aircraft in terminal areas. POs are the building blocks of standardized flight paths, of which examples are:

- Departure procedures
- Procedure turns
- Final approach courses

Standardized flight paths would be expressed as POs, and, in some cases, complex POs would be created to match a standardized path.

3.2 Flexibility and Vectoring

Currently, when an aircraft must deviate from a published DP or STAR, e.g., because of weather, the controller must give radar vectors to the pilot. A *radar vector* is simply an instruction to the aircraft to fly along a certain magnetic heading until told otherwise. Often, the controller has to repeat the same instructions for several aircraft.

Several problems with vectoring are solved with the use of POs. When an aircraft is sent on a vector, the length of an aircraft's intended flight path is no longer known to either the pilot or the control system. Once the aircraft is off a known route, the accuracy of the system is significantly degraded. The time at which the aircraft rejoins the route is uncertain to both the pilot and the control algorithms. The pilot has less situational awareness because the aircraft is not flying toward a known fix. While speed and altitude clearances can be used to prevent timing conflicts, there is a limited amount of controllability using vectoring. Another primary concern with vectoring is the dependency on radar to know the location of the traffic. In case of radar failures, aircraft on vectors are at a higher risk than those on a defined and mutually understood path.

As seen in Figure 3-1 below, there is a decreased workload when using POs rather than vectors. The control system (automated or manual) must issue three heading changes to instruct the aircraft how to fly an S-turn. In addition, the instructions must be timed so that they are issued when needed to fly the turns. With POs, the control system need only issue one PO command of the form [PS1, **P1**, C, a, p] and this can be at its convenience, anytime before the aircraft reaches P1. The control systems on the ground and in the aircraft retain full knowledge about the aircraft's intent.

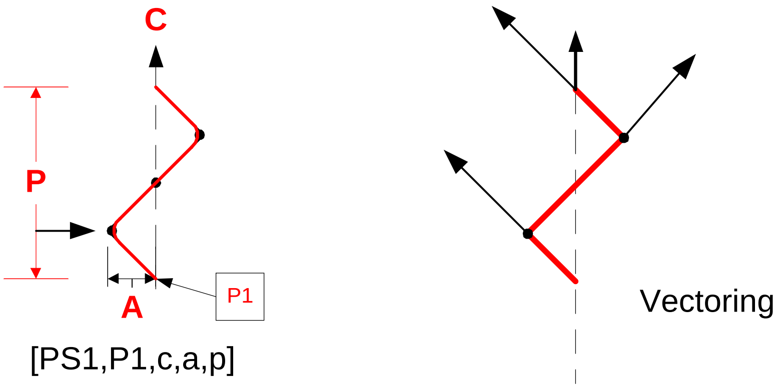


Figure 3-1. Defining an S-Turn with POs

The PO concept offers the capability to add flexibility to the control system while maintaining certainty about the aircraft's intended course. The current system of having paper-based DPs and STARs inhibits flexibility. From the ATC side, it is usually the case that an aircraft must be vectored off the DP or STAR to achieve flexibility. With POs, an aircraft can remain on a path type while its dimensions are slightly varied for ATC purposes. Standardization comes from having a manageable number of types of paths.

By certifying the algorithms in the POP, many variations on standardized routes can be created safely, thereby allowing increased flexibility for the ATC system and users. For example, the distance between the downwind leg and the runway can be varied during the day to account for winds and traffic demand. Precise paths can be defined that lead the aircraft all the way to the intercept with the landing aid.

3.3 Range of Potential Benefits of Path Objects

Figure 3-2 is an attempt to capture the variety of benefits to the different segments of the air transportation community from POs. With respect to the notion of flexible FMS-based terminal routes, airlines may realize benefits in several ways. With standardized routings, airlines would be better able to predict aircraft movement into or out of the terminal area, thereby enabling better preparation of ground staff and resources for aircraft servicing.

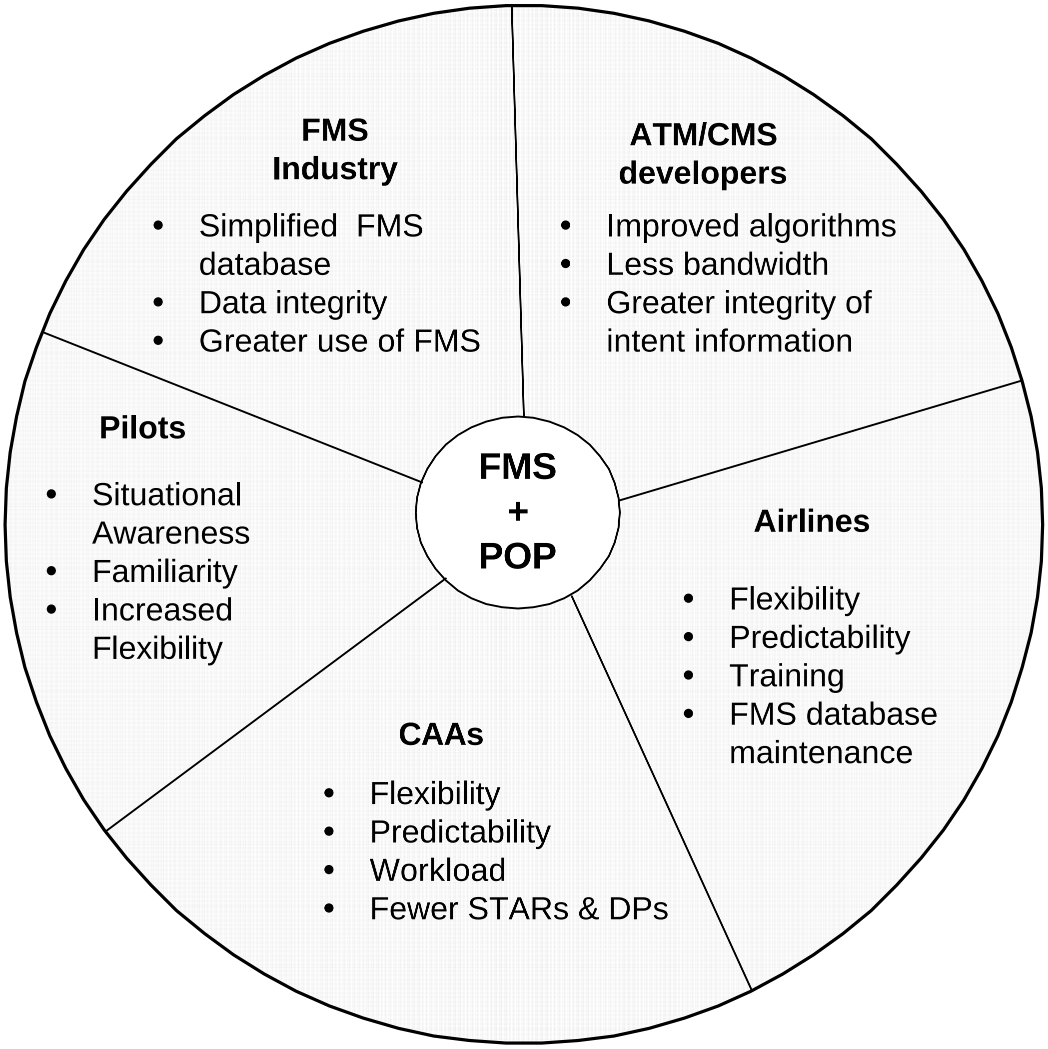


Figure 3-2. Range of Benefits of Path Objects

Training costs for cockpit crews would be incrementally lower in proportion to the simplicity of the new procedures. In a qualitative sense, safety would also be enhanced by standardized paths as the familiarity of the procedures would reduce operational errors, and also because such standardization lends itself to better flight deck automation support.

There may be reduced terminal area and en route delays from voice communications congestion since clearances are shorter and need to be repeated less. The evaluation of data-link benefits for terminal[2] and en route ATM applications estimate the airline operating cost savings potential from eliminating voice communications congestion to be over $590M annually for terminal and en route domains nationwide[3]. If frequency congestion reduction from using PO technology were only ten percent of this, the benefit would still be millions of dollars.

3.4 Potential Benefits to Pilots

3.4.1 Standardization

The benefits of flexible FMS routes to the pilot have been mentioned in earlier sections. The advantages to pilots come from the fact that there are only going to be a small number of flexible POs that the pilots will have to understand. Instead of hundreds of routes and terminal area FMS routes, the pilot will only have to study how to fly POs.

When arriving in the terminal area, he will be presented with one or more POs to fly, each of which is familiar to him. This factor, alone, cascades into several other advantages for the pilot making way for a safer and more efficient operation:

- The burden of training is reduced
- The pilot is better able to plan for the operation in terms of configuration and speed management
- The communication workload is decreased
- The pilot's situational awareness is enhanced as greater use is made of the FMS's ability to navigate all the way to the runway without radar vectors.

Most of the advantages of PO processing would be transparent to the pilot; however, there would be some associated benefits. The segments of the individual POs can be named

---

[2] *User Benefits of Two-Way Data Link ATC Communications,* U.S. Department of Transportation Federal Aviation Administration, DOT/FAA/CT-95/4, February 1995.

[3] *Benefits of Controller-Pilot Data Link ATC Communications in Terminal Airspace,* Federal Aviation Administration, William J. Hughes Technical Center, DOT/FAA/CT-96/3, September 1996.

in consistent, operationally meaningful ways. The naming of key points in compound POs can be standardized. Just as today the traffic pattern has components called (in the USA) *downwind, base, final,* and *crosswind,* the key features of a holding pattern could also have meaningful names.

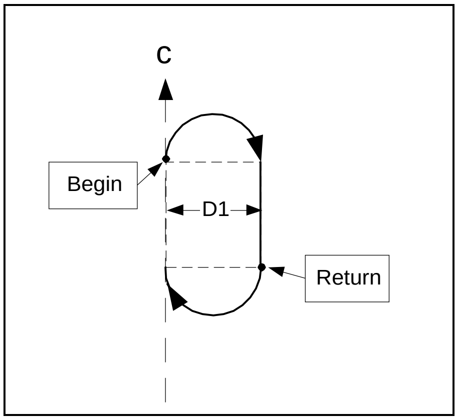


Figure 3-3. Standardized Names for Waypoints

For example, as seen in Figure 3-3 in the racetrack PO R1, the point **P1** could be called *Begin* and the point **P2** could be called *Return*. Then, regardless of where the path R1 was located, the pilot would refer to these two fixes by name, and the aircraft's moving map display would label these *Begin* and *Return*.

In addition, using POs, the paths can be specified without any discontinuities. In the current system, en route airways do not always join with STARS, and STARS do not always join with the approach course. Flight paths can be completely defined from take-off to touchdown. The radius of arcs, for example, could be completely defined which would facilitate provision of steering cues to the pilot through the autopilot/flight-director system.

The use of generic route structures and an improved computer-human interface for viewing route layout are features provided by the PO technology that would improve pilots' ability to rapidly familiarize themselves with approaches to unfamiliar airports.

The simplified method for defining a route and its variants results in less "heads down" time for reviewing and selecting routing options and initiating an amendment request, thereby reducing workload and enhancing safety.

3.4.2 Situational Awareness Between Aircraft

Another potential benefit of POs for future-generation avionics is improved inter-aircraft separation assurance. One emerging cockpit technology, Automatic Dependent

Surveillance-Broadcast (ADS-B), involves having suitably equipped aircraft transmit their position, heading and altitude on a *broadcast* channel out to a range of about 200 nmi. Other aircraft that 'hear' this broadcast would then know if their courses were converging, based on the assumption that the aircraft would continue on their respective headings and altitudes. Using the PO concept, aircraft could succinctly broadcast their exact course intentions. That information, coupled with airspeed, would allow an improved estimation of whether (or not) the two aircraft were on conflicting courses (and consequently decrease the number of warnings and false alerts).

3.5 Potential Benefits to Air Traffic Control Services

3.5.1 Increased Flexibility

Flexible FMS routes, using POs, can be applied to enhance ATM services and will be especially useful for the terminal domain operations. The net impact of the PO concept is that it increases the variety of available arrival and departure routes, providing economic savings and flexibility to the user and ATC.

By making feasible an unlimited set of flexible FMS arrival routes extending all the way to the runway threshold, PO technology enables increased flexibility and predictability in National Airspace System operations. ATM tools can be designed to automatically select alternate defined routes of known length, thus avoiding weather hazards while maintaining the time of arrival predictability needed to fully use the available runway capacity.

This technology preserves the free-flight attribute of *freedom of route* by offering multiple arrival routes to ensure that one can always be found that closely matches the user-preferred path. It also makes it easier for the pilot to communicate requests for changes to an aircraft's route.

Flexible FMS routes can be used for both arrivals and departures. The use of FMS routes allows aircraft intentions to be known. This will improve conflict probe performance, especially with respect to a reduction in false alerts. Predictable trajectories will allow enhanced safety while affording freedom to choose a preferred route from the many options made possible by the PO routing technology.

FMS routes, using PO technology, can reduce controller workload associated with spacing aircraft in the terminal area. Replacing control by vectoring with the ability to select a flexible route reduces the vectoring task to one of issuing only occasional speed adjustment clearances. For example, a controller could easily select a maneuver expressed in terms of a PO that would increase the aircraft's path length by exactly 3 miles, allowing the controller to achieve more precise spacings.

While it is possible to store many fixed FMS routes in the database, doing so would still provide only limited flexibility. Figure 3-4 shows a set of fixed, predefined FMS routes.

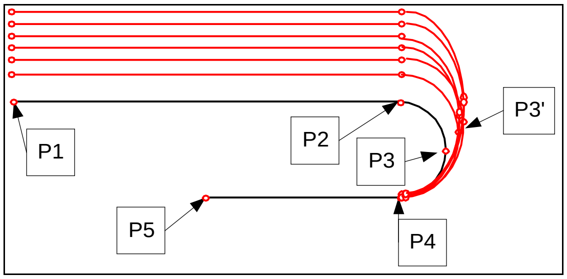


Figure 3-4. Set of Fixed FMS Routes

With POs a single flexible PO, as shown in Figure 3-5, could generate all of these fixed routes as well as many more. This arrival pattern is specified by [TB1, **P1**, c,d1,d2,d3].

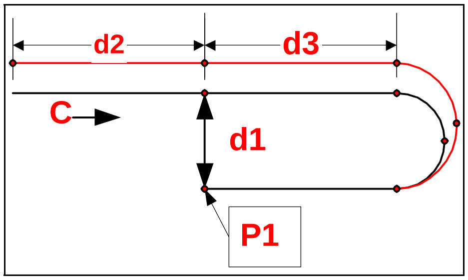


Figure 3-5. Flexible FMS Routes

In this example, the PO is defined with parameters that have operational meaning. The length, d1, for example is the width of the *base* leg of the traffic pattern. Having such a PO stored in the FMS would mean that the controller and pilot could communicate in meaningful terms and still keep the FMS in the process of controlling the aircraft's path. This would allow more precise path control even when not all aircraft are equipped with data-link capability.

3.5.2 Improved Situational Awareness for Controllers

A more orderly flow along FMS routes, rather than having ad hoc maneuvering will improve controller situation awareness and make conflict alert tools more reliable. This, along with reduced voice channel congestion resulting from reduced reliance on vectoring, allows the controller to maintain situational awareness, reduces distractions and thereby increases safety while alleviating controller workload and stress.

Because the aircraft have well-defined routes throughout the terminal area, the controller will have higher confidence that the aircraft will adhered to the assigned routes. This leaves the controller more time to manage the exceptional circumstances.

3.5.3 Application to Free-flight

In the free-flight concept, the PO technology will permit aircraft to reliably and efficiently exchange intent information with the ATM systems. Using the FMS and POs, the pilot can create the desired route. The POP can then express it in terms of POs and efficiently transmit that to the ground. This means that as the needs of the aircraft operator change, the aircraft can quickly and reliably notify ATC of its changed intentions. This will speed up the process of verifying separation assurance, allowing for more flexibility in congested airspace.

3.5.4 Potential for Improved ATM Algorithms

3.5.4.1 Potential for Better Planning Algorithms

The PO concept allows the creation of a flexible, yet discrete, set of arrival paths. On a given day at a particular airport, the set of feasible paths to the runway would be relatively small. It would be possible to calculate the cost (time) of assigning an aircraft to each of the possible arrival paths and then choose the assignments to minimize the overall sum of the costs. Discretizing the set of paths opens the way for new classes of ATM algorithms that would select a preferred path from the set of POs, rather than from a continuum of paths. In general, it may be possible to use a variety of combinatorial optimization methods to better control traffic. Such options for control mechanisms are not possible without POs because of the uncertainty in aircraft intentions and the large number of possible paths that would have to be evaluated.

3.5.4.2 Greater Accuracy in Estimating Trajectories

Greater path predictability in terminal areas will improve the accuracy of trajectory modeling, reducing the arrival time dispersion at key nodes such as merge points and runway thresholds. Even though speed control can be used for metering with reasonable success, there is a limited amount of adjustment to metering times that can be accomplished with speed control alone over a fixed path length. Since path extensions using POs can be accomplished by varying the path parameters, vectors off the route will not be required and greater predictability will increase the achievable throughput for the airport. A combination of path control and speed control will enable the ATM system to meet any metering objective.

Currently, ATM algorithms must rely on waypoints to define and communicate information about intended paths. In order to define a curve, many waypoints must be specified to indicate a piece-wise linear approximation to the desired curve. With POs, a curve can be specified (to the accuracy of the FMS's ability to fly it) with only a few parameters. This provides increased fidelity to the path definitions within the terminal area. In addition, when the path must be changed, it may only require a small number of parameter changes, rather than many waypoints. In the example below, the traffic pattern must be widened by specifying new waypoints. If the pattern is expressed by waypoints, rather than POs, several waypoints must be changed. With the traffic pattern PO mentioned above, only one parameter change is required.

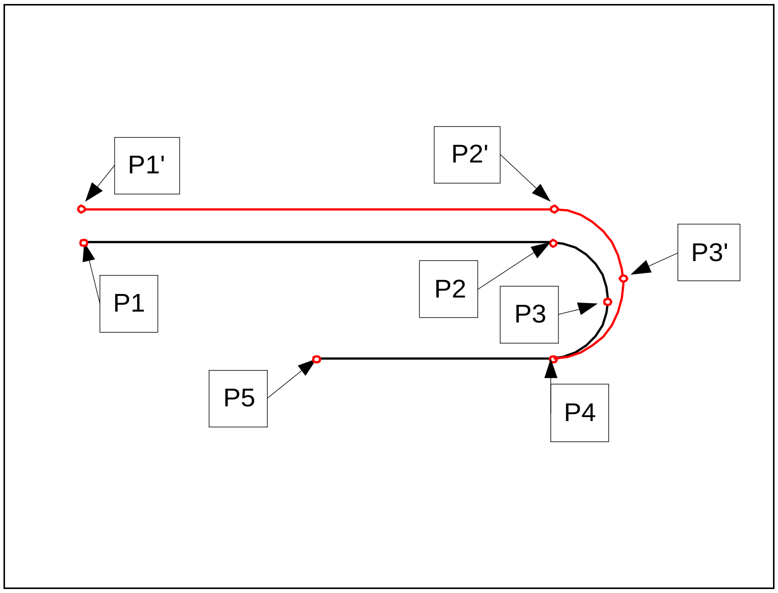


Figure 3-6. Widening Traffic Pattern

3.5.4.3 Reduced Demand on Data Link for Routine Operations

In the future, a large segment of the data-link message traffic will be composed of request/clearance messages. Simplified path definition technology will result in reduced demand on data-link capacity, allowing lower cost data-link networks. This will also allow ATM control algorithms for metering or spacing to respond more efficiently to operational environment changes (such as the appearance of a storm cell blocking an arrival path) and produce better solutions.

3.5.4.4 Use of a Common Language of Path Definitions

ATM functions are being developed by countries all over the world. If a particular function intends to utilize the capability of the FMS, it must currently plan to send a sequence of waypoints to vector the aircraft according to its calculations. There is nothing technically wrong with that approach and if POs are never adopted, that is what will happen. However, data link will be required to process the sequence of waypoints.

POs, especially compound POs, allow standard maneuvers to be stored, processed, and exchanged easily. There are many types of paths that can easily be transmitted by voice, not requiring data link. Because a PO specifies a maneuver such as a turn or arrival pattern, the FMS can have functionality that capitalizes on that. With a string of waypoints, the FMS can only process them sequentially. For example, the FMS could, in principle, calculate the most fuel-efficient way to fly a holding pattern expressed as a PO because it would 'know' that it was trying to fly a holding pattern. In contrast, if just presented with a string of waypoints, it would probably not even 'recognize' that it was in a holding pattern. There is more to *intent* than just a string of waypoints.

With the use of POs, individual countries can develop their desired control systems, taking advantage of the FMS's capability, without having to change the architecture of the FMS. Without POs, the only way that a new ATM function could utilize the FMS is to have the FMS pre-programmed for the particular application. This would add months or years to the development process. For example, one country may have an advanced sequencing and metering system based on proprietary information and research. With POs, the system would merely convert its desired courses to POs and send that information to each aircraft. The aircraft system and the FMS need not have any knowledge of the underlying ATM functions because they can fly the requested POs. An FMS system made in the United States could be in an aircraft flown to Europe where it is controlled by an algorithm developed in France, without the US manufacturer's having any knowledge about the French ATM manufacturer.

This type of open standard permits more rapid development of innovative ATM strategies.

3.5.5 Avionics and Navigational Charts Industries

A primary benefit of the PO concept is that it reduces the work necessary to maintain the FMS's navigation database. While this may seem less important than the flight benefits to the aircraft, it is because the problem is not yet that severe. The worldwide navigation data base must be updated every 28 days, a task that gets more tedious and vulnerable to errors as the number of worldwide routes proliferate. For example, there are now approximately 1500 STARs, each defined by roughly 5 to 10 latitude/longitude coordinate points that are included in this data base along with DPs and other types of routes. It is anticipated that with PO technology, by allowing standardized routes with relatively few parameters, there would be a reduction in the number of characters that must be maintained and a simplified database, reducing cost and risk of erroneous entries.

Currently, the aircraft must carry the most current database of navigation information in the FMS computer. An aircraft on international routes, where it may be routed over many different countries, must carry this information for all airports, in the event it has an en route emergency of some kind and must land. The basic problem is that, the more numbers in the database, the greater the chance for error. There are numerous sources of errors that can occur in the process, as all these data elements are moved from their source to each aircraft's FMS. These errors can occur in the source data (the VOR is not where the coordinates say it is), the master database, copies of the master database, and as random errors during each transfer. This is an enormously labor-intensive effort when all of the participants are considered. A group called the RTCA Committee SC-181 is studying how FMS procedures should be designed to account for the fact that there may be errors in the data.

With the PO concept, the size of this database will initially shrink to a more manageable level. This is because multiple FMS routes will be replaced by single POs. However, the pressure to add FMS-based routes will increase the number of routes. The PO concept will minimize the size of the database and make it possible to add flexibility while maintaining data integrity. The simplified method of route definition with fewer data entry steps that is offered by PO technology increases the data integrity. By using parameters to define the route, there is no limit to the number of paths that can be generated for the FMS. This will increase the role of the FMS in ATC systems making them more responsive to user needs, while maintaining safety.

3.5.6 Potential Benefits to Civil Aviation Authorities

Flexible FMS routes have the potential to positively impact the NAS performance metrics. Flexibility and predictability in terminal operations can be significantly improved if aircraft are adhering to FMS-defined paths. These paths can be extended or shortened to allow for system flexibility without compromising system predictability.

Another benefit that can be realized is a reduction in the total number of STARs and DPs that need to be stored in the worldwide navigation database. A limited set of path types will be stored and a mapped to each airport. This concept eliminates the need to define navaid-based routes for each site. In place of a large database of unique routes, a small set of standardized paths can be mapped to the world's airports. The Federal Aviation Administration (FAA) is moving toward standardization of RNAV approaches (including GPS), per FAA Order 8260.45. The order describes a standardized approach course for the terminal area, which is compatible with the concepts described in this document.

In the example below, a generic approach route is shown. There are points along the approach that are for altitude control and are specified distances from the starting and ending points. Such an object is very general and can be used to construct a wide variety of terminal area procedures.

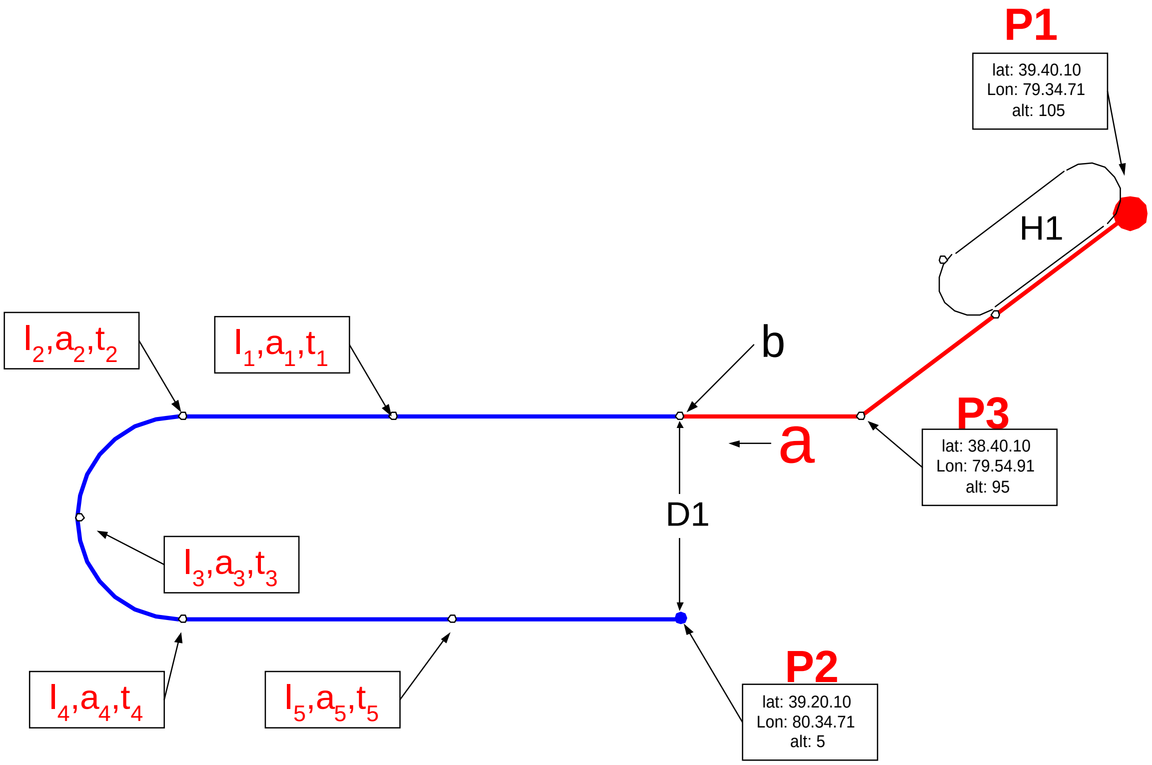


Figure 3-7. Generic Terminal Arrival Route

The PO is encoded as [ST2,P1,P2,P3,a,d,h,l_1,a_1,t_1, l_2,a_2,t_2, ... l_n,a_n,t_n]. P1 and P2 are the beginning and ending points. P3 determines the width of the pattern and the start of the ‘downwind’ segment of the path. The downwind course is given as "a." Using P3 and “a”, the point “b” can be determined and the distance D1 can be calculated. The parameter “d” is the distance from “b”, on course “a” at which the aircraft starts a 180 deg turn of radius D1 toward P2. The parameter h would be given value 1 if the holding pattern is to be used, otherwise it is “0.” The altitude crossing points are given as triples,[l_i , a_i , t_i]. There are “n” altitude restrictions. The l_i’s are distances measured along the path from P1. The a_i’s are altitudes and the t_i’s are types. The types are: 1- “at”, 2- “at or above”, 3- “at or below”, 4- “suggested.” The corresponding algorithm would be:

- Proceed to P1
- If h=1, fly holding pattern 1. When finished,
- Fly directly to P3

- Calculate Distance D1 between P2 and course “a” through P3
- Calculate point “b” on perpendicular to course “a” through P2
- Turn and fly along course “a” to point “b”, then for distance “d” along course “a”
- Turn 180 deg towards P2 along circle of diameter D1,
- Fly directly to P2

Section 4

Conclusions

Implementing the PO concept has the potential to solve many problems that are now facing planners of the world's future ATC systems. By incorporating the aircraft's flight management computer into the architecture of the traffic control system, the PO concept offers the capability to maintain a high degree of certainty about an aircraft's intended path regardless of its complexity.

The PO concept offers a common language that can express complete information about the aircraft's intended path, which in turn offers benefits to both the ground control system and the aircraft operator. By simplifying the expression of flight paths, it solves many practical problems related to the maintenance of navigation databases, commonality of procedures for the pilot, charting and cockpit display of information, and efficiency of transmission.

POs that are defined by parameters having 'meaning' to the controller and pilot, such as the length of the base leg, would allow the exchange of intent information even in a voice environment. This then allows the FMS to be incorporated into the ATC system during the transition to a fully equipped data-link environment. The issue of how to deal with partially equipped aircraft has been a difficult one for automation developers. The concept also can be used during the transition in a mixed-equipage environment using voice and data-link communications.

While no single benefit would probably justify the transition to this concept, the fact that it addresses so many major issues at once, makes it a very desirable concept to both the users of the airspace and the providers of ATC services.

POs can be slowly integrated into the system over 5 to 10 years as the avionics are replaced and upgraded. To implement this concept, the FMS computers must be enhanced to store and process POs. While this can be done unilaterally by a single avionics manufacturer, an international standard is preferable. Both airborne and ground-based POPs must be developed that interface with the aircraft systems and the ground-control systems. In fact, the most advanced FMS systems already have two POs; the line segment and the holding pattern.

Consequently, POs offer a feasible solution to many problems now facing the developers of the next generation of traffic management systems.

Appendix A

Summary of Flexible Path Object Definitions

Type	Key	Form	Description **[P1 =(x1,y1,z1)]**
Line Segment	L1	[l1,**P1,P2**]	Great Circle Segment between **P1** and **P2**
	L2	[l2,**P1**,a,d]	Great Circle Segment from **P1**, on course a, for a distance of d, using constant altitude z1
Turn	T1	[t1,**P1,P2**,**P3**]	3 waypoint turn **P1**-**P2**-**P3**
	T2	[t2,**P1,P2**,β,α]	Turn with relative bearings from course **P1**-**P2**
	T3	[t3,**P1,**α,d1,β,d2]	Fly from **P1** along course α for distance d1, then course β for distance d2
	T4	[t4,**P1,P2**,**P3**]	From **P1**, turn at **P3**, following circle having radius of 3 nmi, direct to **P2**.
	T5	[t5,**P1,P2,P3**,r]	Same as T4, except turn radius is r
	T6	[t6,**P1,P2**,r]	Right Turn along a radius of r from **P1** to **P2**.
	T7	[t7,**P1,P2**,r]	Left turn along a radius of r from **P1** to **P2**.

Type	**Key**	**Form**	**Description [P1 =(x1,y1,z1)]**
Course Reversal	C1	[c1,**P1,P2**,**P3**]	Calculate course **P1**-**P3**, distance d from **P2** and **P1**-**P3**, turn at **P3** 180 deg along circle of diameter d, to **P2**
	C2	[c2,**P1,P2**,a]	From **P1** fly course a, for 5 nmi past perpendicular to **P2**, turn 180 deg, to **P2**
	C3	[c2,**P1,P2**,a,d]	Same as C2, except fly d miles past perpendicular
Compound Objects	S1	[s1, ,**P1,P3,P2**, a]	Combine T1 and C2 to develop a STAR.
	PT1	[PT1, **P1**, **P2]**	Procedure turn on course **P1**-**P2,** 2 nmi leg, 30 deg intercept
	F8	[f8, **P1,P2**]	Figure 8, oriented on line **P1**-**P2**, starting and ending at **P1**
	TB1	[Tb1,P1,C,d1,d2,d3]	Traffic pattern starting on course C, ending at P1, for distance d2 + d3, d1 away from P1.
	ST1	[STL1, **P1,**C,a,p]	S-turn to the left along course C starting at **P1,** with amplitude a and period P.
	R1	[R1, C,**P1,P2**]	'Race track' holding pattern on course C, using **P1** and **P2** as corners of rectangle.

Glossary

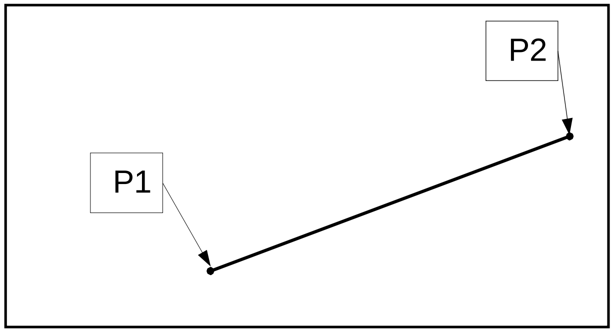
ADS-B	Automatic Dependent Surveillance-Broadcast
ATC	Air Traffic Control
ATM	Air Traffic Management
CNS	Communication, Navigation, Surveillance

Appendix B
Initial Set of Path Objects

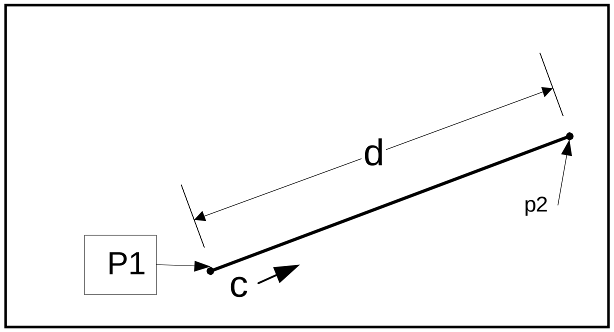
Initial Path Objects

- Line Segments
  - [L1,P1,P2], [L2,P1,c,d]
- Turns
  - [T1,P1,P2,P3], [T2,P1,P2, β,a]
  - [T3,P1, a, d1, b, d2], [T4, P1, P2, P3, r]
  - [Dly2, P1, P2, D] Path Extension
  - [ST1,P1,A,P,C] S-Turn
- Patterns
  - [R1,P1,P2,a] Holding
  - [PT1, P1, P2] Procedure Turn
  - [TP,P1,a,D1,D2,D3] Traffic Pattern
  - [ST1,P1,P2,P3,a,d,h,l_1,a_1,t_1, l_2,a_2,t_2, ... l_n,a_n,t_n] STAR
- In conjunction with these objects, there must be basic editing functions such as ADD, DELETE, INSERT, MOVE, and MODIFY

Line Segment

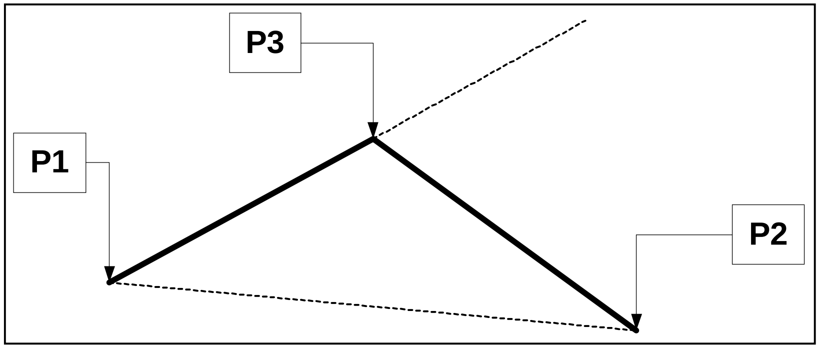


- [L1,P1,P2]
- Algorithm:
  - Proceed to P1,
  - Fly directly to P2



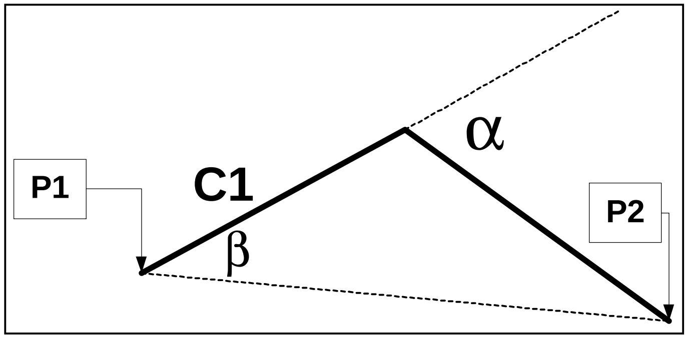
- [L2,P1,c,d]   (constant altitude)
- Algorithm:
  - Proceed to P1
  - Fly course “c” for “d” nm
    - calculate P2

Turn T1



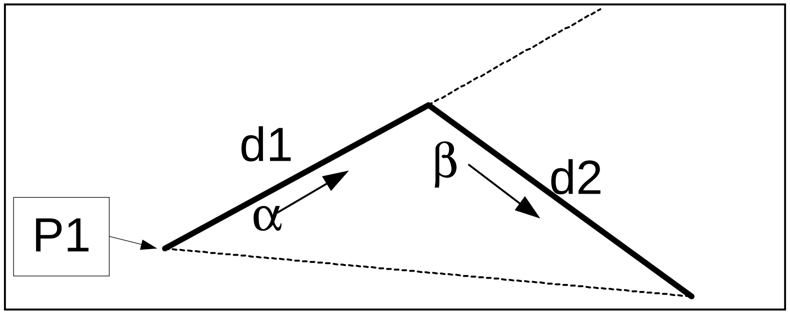
- [T1,P1,P2,P3]
- Algorithm:
  - – Proceed to P1,
  - – Fly directly to P3,
  - – At P3, turn toward and
  - – Fly directly to P2

Turn T2



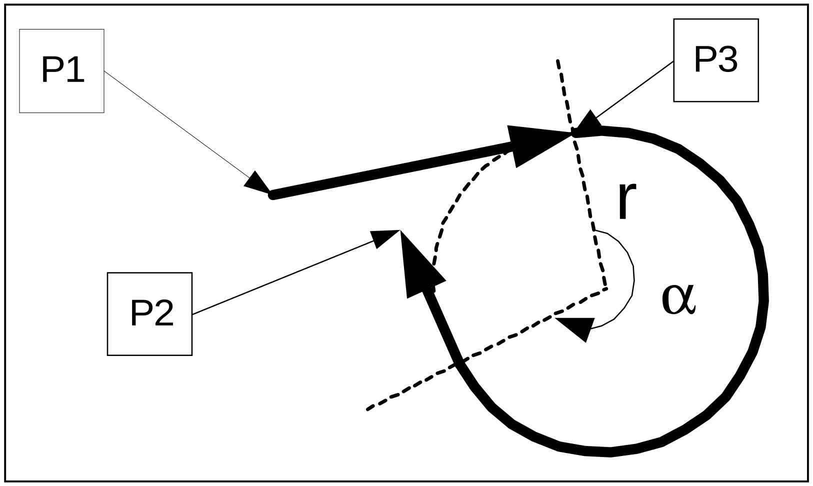
- [T2,P1,P2, β,α]
- Algorithm:
  - Proceed to P1,
  - Calculate course C1 as, course P1-P2, minus β
  - Fly from P1 along the course C1 until the relative angle between the aircraft and P2 is α,
  - Turn toward, and fly directly to P2.

Turn T3



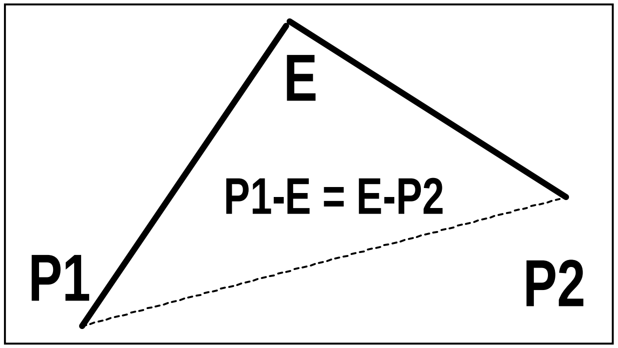
- [T3,P1, α, d1, β, d2]
  (Constant altitude)
- Algorithm:
  - Proceed to P1
  - Turn to magnetic course α
  - Fly a distance d1
  - Turn to magnetic course β,
  - Fly distance d2.

Turn T4



- **[T4, P1, P2, P3, r]**
- **Algorithm:**
  - **Proceed to P1**
  - **Fly directly to P3**
  - **Calculate circle of radius, r, with center on perpendicular to P1-P3, at P3**
  - **Fly along circle until on course heading directly to P2,**
  - **Fly directly to P2**

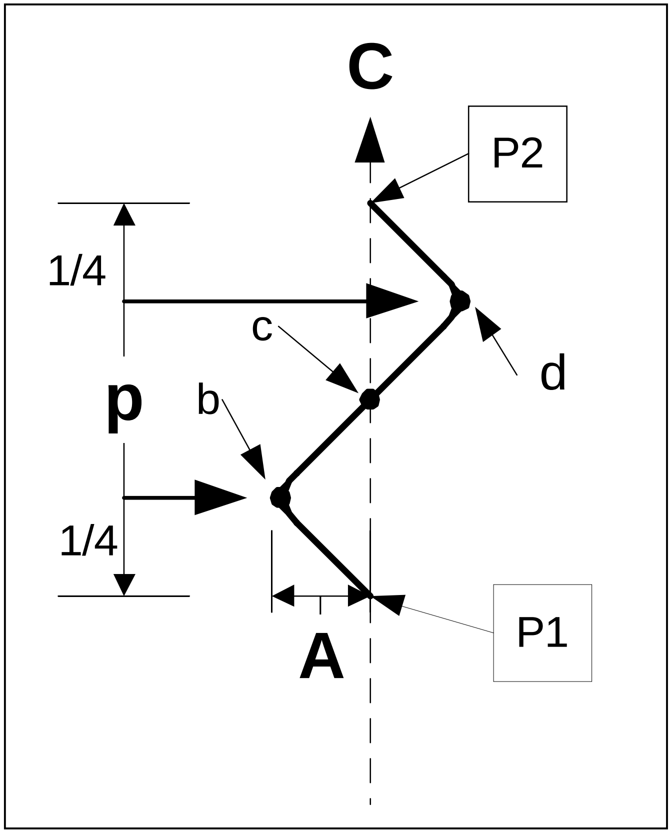
Path Extension 2



[Dly2, P1, P2, D]

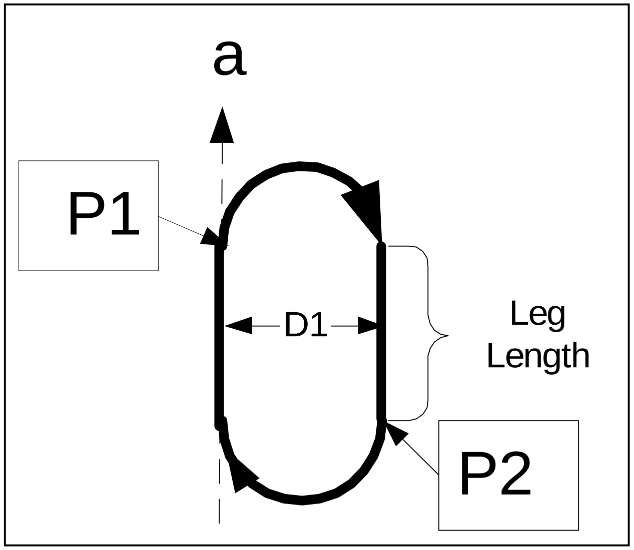
- Fly to P1
- Calculate point E so that :
  - |P1- E| = |E-P2|
  - |P1- E| = [D+ |P1-P2|] / 2
- Fly to E and then to P2

S-Turn Maneuver



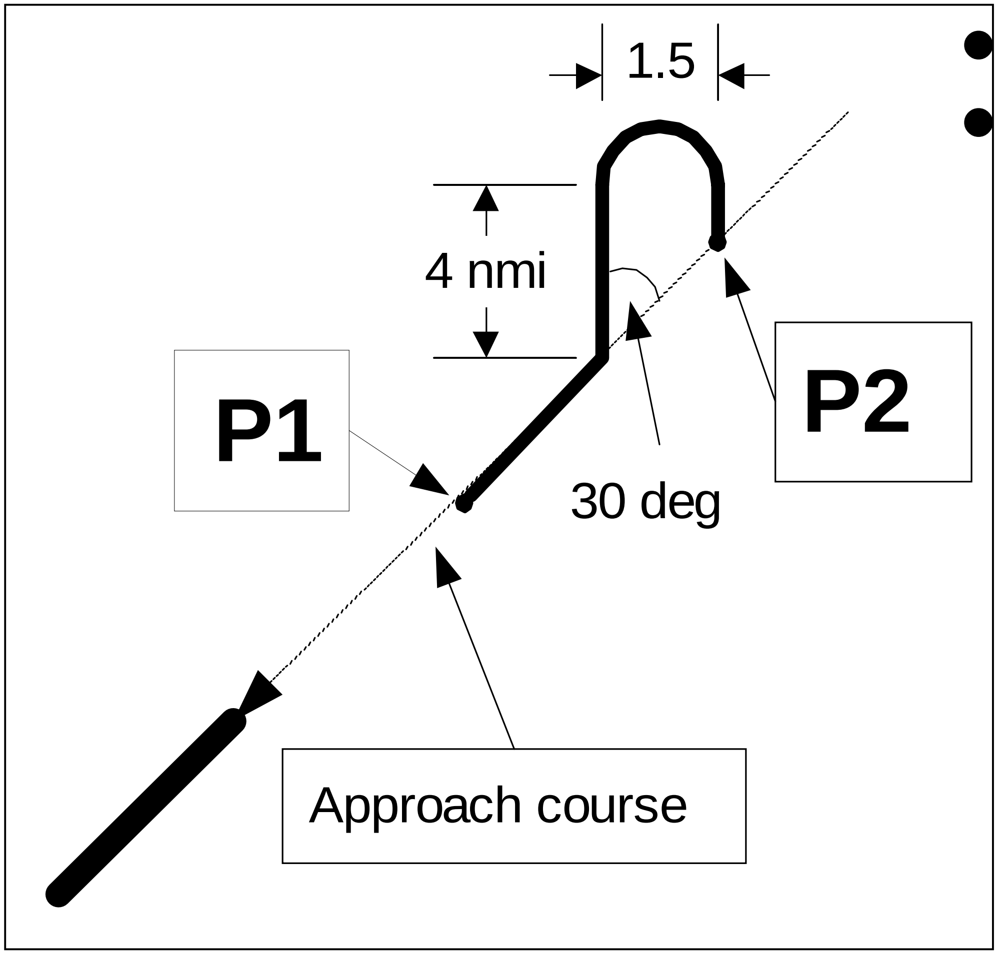
- [ST1,P1,P2,A]
- Algorithm:
  - Fly to P1
  - Calculate course C and distance, p, from P1 to P2
  - Calculate points b, c, and d, 1/4-, 1/2- and 3/4-P, from P1
  - At P1, Turn <u>left</u> (<u>right</u>, if a is < 0), fly smooth turns through b, c, and d
  - Return to course C.

Holding Pattern R1



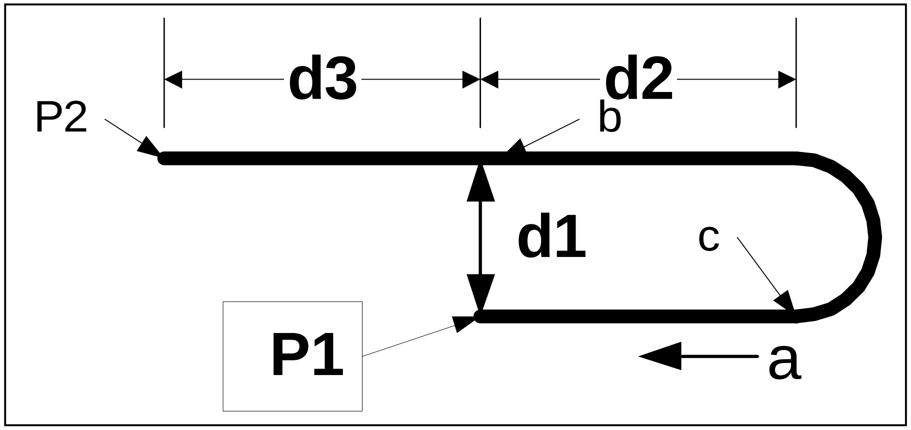
- [R1,P1,P2,a]
- Algorithm:
  - Proceed to P1 along course “a”
  - Calculate corners of rectangle having P1 and P2 on opposite corners
  - Calculate D1, the distance from P1 to the side containing P2
  - At P1 turn toward P2 on circular path of diameter D1,
  - fly directly to P2 on reverse course, a+180
  - At P2, turn toward P1 on circular path of diameter D1
  - Repeat algorithm

Procedure Turn P1



- [PT1, P1, P2]
- Algorithm:
  - Fly to P1
  - Calculate Course P1-P2
  - Fly along P1-P2 until 3 nm from P2
  - Turn left 30 degrees, fly 4 nm
  - Turn right 180 degrees along arc of circle having 1.5 nm diameter
  - Fly to P2
  - Turn right to reciprocal course, P2-P1,
  - Fly to P1

Traffic Pattern

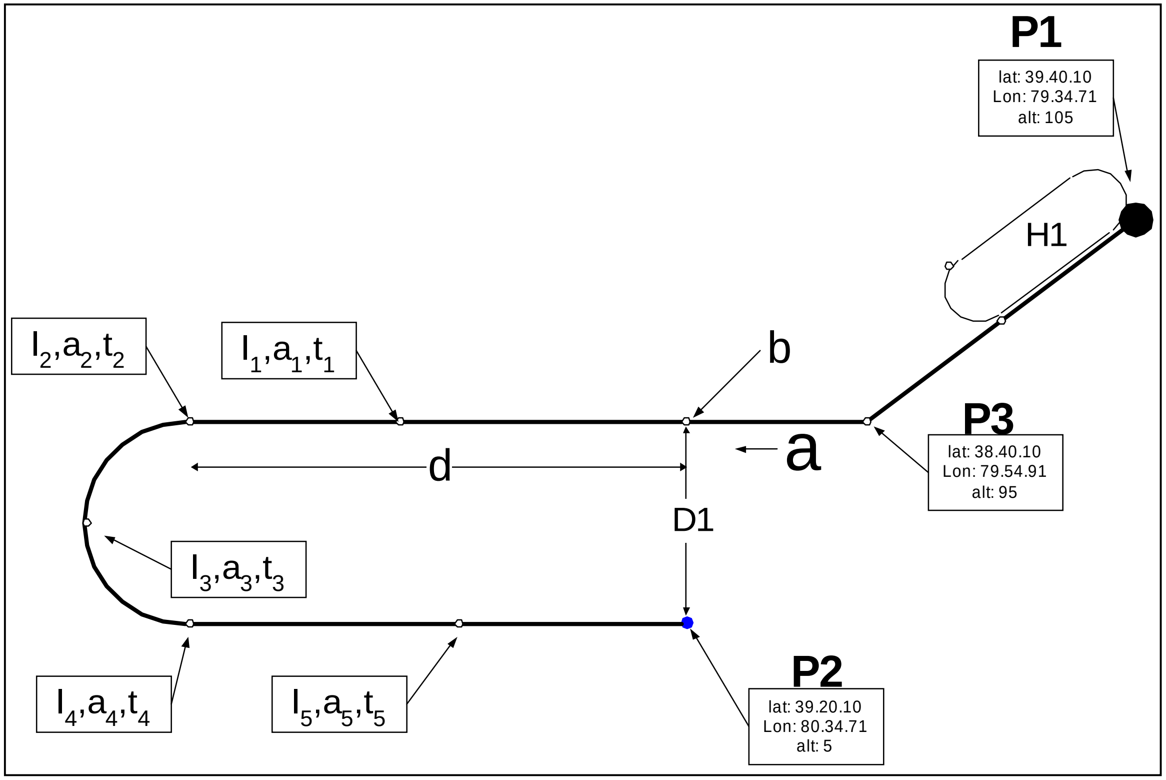


- [TP,P1,a,d1,d2,d3]

Algorithm:

- Calculate:
  - b from (P1,a,d1)
  - p2 from (b,a,d3)
- Fly to p2
- Fly from p2 on course a+180 for distance d3+d2
- Turn toward P1 on circular arc of diameter d1
- Fly to P1 on course “a”

STAR ST1



- **A generalized STAR is a combined turn and course reversal, with added altitude requirements**
- **Described on following page**

$[ST1,P1,P2,P3,a,d,h,l_1,a_1,t_1,\ l_2,a_2,t_2,\ \ldots\ l_n,a_n,t_n]$

STAR Construction

- $[ST1, P1, P2, P3, a, d, h, l_1, a_1, t_1, l_2, a_2, t_2, \ldots l_n, a_n, t_n]$
- Proceed to P1
- If h=1, fly holding pattern 1. When finished,
- Fly directly to P3
  - Calculate Distance D1 between P2 and course “a” through P3
  - Calculate point “b” on perpendicular to course “a” through P2
- Turn and fly along course “a” to point “b”, then for distance “d” along course “a”
- Turn towards P2 along circle of diameter D1,
- Fly directly to P2
- There are “n” altitude restrictions. The l_i’s are distances measured along the path from P1. The a_i’s are altitudes and the t_i’s are types. The types are: 1- “at”, 2- “at or above”, 3- “at or below”, 4- “suggested”