NASA'S AUTONOMOUS FORMATION FLYING TECHNOLOGY DEMONSTRATION, EARTH OBSERVING-1 (EO-1)

David FOLTA (david.folta@gsfc.nasa.gov)
John BRISTOW (john.bristow@gsfc.nasa.gov)
National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, MD 20771
(301)-286-6082

Albin HAWKINS (hawkins@ai-solutions.com)
Greg DELL (dell@ai-solutions.com)
a.i. solutions, Inc.
10001 Derekwood Lane Suite 215
Lanham, MD 20706

ABSTRACT *- NASA's first autonomous formation flying mission, the New Millennium Program's (NMP) Earth Observing-1 (EO-1) spacecraft, recently completed its principal goal of demonstrating advanced formation control technology. This paper provides an overview of the evolution of an onboard system that was developed originally as a ground mission planning and operations tool. We discuss the Goddard Space Flight Center's formation flying algorithm, the onboard flight design and its implementation, the interface and functionality of the onboard system, and the implementation of a Kalman filter based GPS data smoother. A number of safeguards that allow the incremental phasing in of autonomy and alleviate the potential for mission-impacting anomalies from the on-board autonomous system are discussed. A comparison of the maneuvers planned onboard using the EO-1 autonomous control system to those from the operational ground-based maneuver planning system is presented to quantify our success. The maneuvers discussed encompass reactionary and routine formation maintenance. Definitive orbital data is presented that verifies all formation flying requirements.*

1-INTRODUCTION

An innovative technical approach to autonomously maintain formations of spacecraft is essential, as scientific objectives become more challenging. New scientific research such as large-scale interferometry and co-scene comparisons has led many programs to recognize the advantage of flying multiple spacecraft in formation to achieve correlated instrument measurements. Beyond technologies that enable formation flying, the cost of on-orbit operations remains a significant and visible concern. Formations may require maneuvers so frequently that the mission itself would be cost prohibitive without automation. Advances in automation and technology by the Guidance Navigation and Control division at the Goddard Space Flight Center (GSFC) have resulted in the development and demonstration of an autonomous system, AutoCon™, to meet these new challenges. NASA GSFC teamed with a.i.-solutions, Inc. to fly AutoCon™ onboard the New Millennium Program's (NMP) Earth Observing-1 (EO-1). This technology automated EO-1's maneuver planning and formation control. AutoCon™ performs orbit maintenance of single spacecraft or constellations and formations. It can be applied to a variety of orbits ranging from low Earth orbits to non-Keplerian trajectories such as libration orbits.

2 - EO-1 MISSION REQUIREMENTS

NASA created the NMP to develop and validate the advanced technologies necessary to support space exploration in the 21st Century. The NMP's first earth observing mission is EO-1. EO-1 has as a principal mission requirement to successfully complete paired scene observations with Landsat-7 in order to validate the technologically advanced imagers on EO-1 [Mrre97]. To enable the paired scene process, the EO-1 spacecraft must fly over the current groundtrack of Landsat-7, a repeating groundtrack mission, within +/- three km. Also, in order to maintain a safety criterion, the nominal along-track separation is one-minute (450km), +/- six seconds (42.5km). The six-second tolerance is derived from a +/- 3-km groundtrack requirement. Maintaining this three-dimensional separation requirement is referred to as formation flying. Formation flying involves position maintenance of multiple spacecraft relative to measured separation errors. Since EO-1 and Landsat-7 have sizeable differences in their ballistic coefficients, the relative motion varies. Therefore, the formation maintenance requires an active control scheme to maintain the relative positions. Optimally, this process is performed autonomously on-board the spacecraft. An example of the orbit dynamics of the EO-1 and Landsat-7 formation in a rotation frame is shown in Figure 1. Maneuver algorithms were developed to provide EO-1 with the ability to adjust its orbit to maintain formation with Landsat-7. Two algorithms flew on EO-1: GSFC's algorithm known as the Folta-Quinn (FQ) algorithm [Folt98] discussed herein and a JPL-developed algorithm [Guin97].

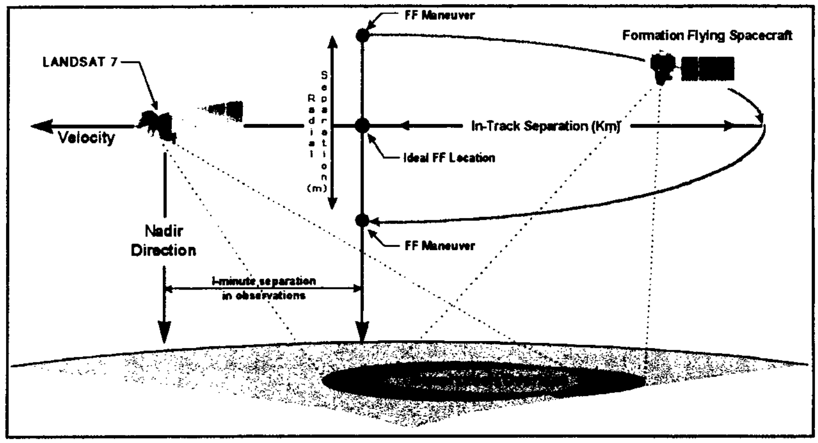

Figure 1. EO-1 Formation Relative Motion

2.1 – AutoCon Background

AutoCon™, a ground-based mission planning tool, was originally developed to satisfy automation needs of the mission analyst [Chap01]. Ground based AutoCon™ includes a user friendly graphical interface, plots and report generation. It uses fuzzy logic to resolve multiple conflicting constraints and plan maneuvers. Fuzzy logic can be used to control mission planning through a rule-based scheme by combining constraints such as orbit true anomaly and shadow events. Mission and instrument constraints can also be incorporated into a flexible maneuver-planning scenario. The on-board flight version of AutoCon™, developed for EO-1, consists of a subset of the ground based AutoCon™ plus a flight software interface. Table 1 shows the functionality included in the flight and ground versions of AutoCon™. The flight interface connects directly with the Command and Data Handling (C&DH) system to retrieve all required data, including GPS position information, and to create command loads for computed burn times and durations. Only the objects and methods needed to support EO-1 formation flying are incorporated in the AutoCon™ system conserving on-board resources. AutoCon™ inherits from its ground-based counterpart its object-oriented C++ design. Scaling the existing ground software for on-board use not only saves money in porting, but also saves in testing, since the development path automatically provides a ground reference system. AutoCon™ is designed to be flexible and extendable. It provides the architecture to facilitate interchangeable formation flying algorithms. AutoCon™ is built around a structure called an *event.*

Table 1. AutoCon Functions

Function	Ground	Flight
Multiple Spacecraft States	Yes	Yes
Time Conversion	Yes	UTC-TAI
Event Calculations	>100	10
Integrators	6	1
Coordinate Conversion	Multiple	EO-1
Maneuver Decision	Fuzzy Logic	EO-1 Specific Fuzz Logic
Force Models	Multiple	EO-1
C&DH Interface	None	EO-1

Events can be added to AutoCon™ as necessary to support new algorithms or capabilities, thus providing extendibility. To be flexible, AutoCon™ uses natural language scripting. The scripting provides the EO-1 flow control as shown in Figure 2. A new algorithm can be defined by events that are scripted to represent the algorithmic process. As long as all the necessary events exist, a new algorithm can be uploaded and executed on-board without changing the flight software.

3 - ONBOARD DEVELOPMENT

The AutoCon™ flight control system is designed to be compatible with various onboard navigation systems (i.e. GPS, uploaded ground-based ephemeris, etc). One interface to the Command and Data Handling (C&DH) system is employed to obtain all onboard attitude and propulsion system data. This C&DH interface is used to ingest GPS state information, and command tables, and output telemetry and maneuver commands. The maneuver algorithm input data are provided internally though propagation of the initial states. EO-1 autonomous formation control requires that a known control regime be established consistent with mission parameters. That data was provided once to the spacecraft as a set of relative formation limits. When orbital differential perturbations carry the spacecraft close to any of the established boundaries, the spacecraft reacts (via maneuver) to maintain itself within its error box. The EO-1 system is currently set to check the tolerance requirements every 12 hours. From this point, AutoCon™ propagates the states for 48 hours (a commandable setting) and will execute a maneuver plan if needed.

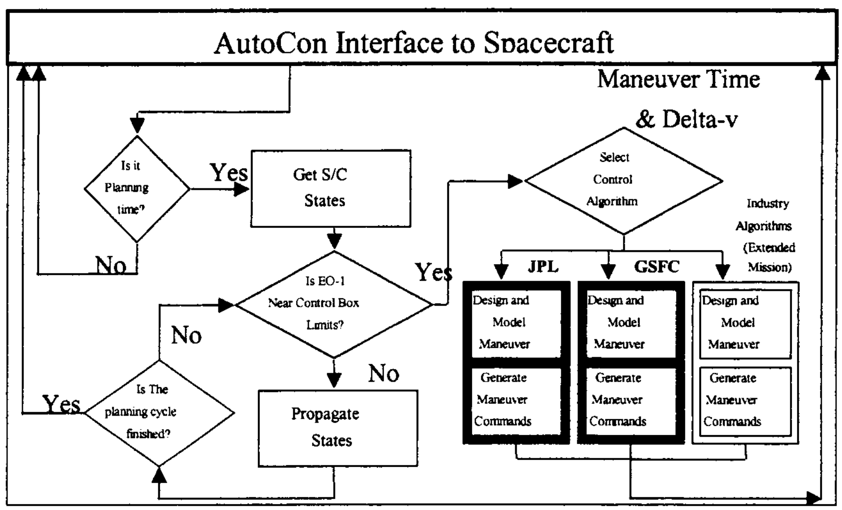

Figure 2. AutoCon Control Flow

3.1-Flight Conversion

The ground version of AutoCon™ was developed under Windows NT. The flight system is built on a Mongoose V (MGV) processor running VxWorks. The system is built with the Tornado compiler, which is a derivative of GNU. Since a MGV system was unavailable at the time of the initial port, AutoCon™ was initially ported to HP UNIX and built with the GNU compiler. The following issues were addressed during this port, all filenames and references to filenames were changed to lower case; and the system was built for the environment, a single executable was built by generating an all-inclusive makefile. The major porting issues occurred during the compiling and linking of the system. The compilation issues with the largest scope were the use of the GNU string and math libraries. To resolve the string class issues, a simplified string class specific to AutoCon™ was developed to override the system string class. To support the change in the math library, AutoCon™ required its own definition of PI and the re-evaluation of error handling for the math functions. For example, the fmod function provided in the Windows environment returned a value of zero when one of the two passed arguments was zero. In the flight environment, a +NAN (not a number) was returned from the same method if a zero was passed as the second argument. The largest challenge when porting to the flight system was the restriction on dynamic memory allocation. While the object-oriented design is based on the creation of objects at run time, the flight operating system forbade the use of dynamic memory allocation. To overcome this hurdle, AutoCon™ was fitted with a memory manager. The memory manager contained in its data segment a 1.5MB block of space and overrode the C++ new and delete operators to manage the space. A number of redesigns were implemented before the memory manager operators would compile without conflicts with other parts of the flight system.

3.2 -Size Reduction

The next challenge was to fit into the available space on-board. When first ported to UNIX, the executable size was over 7MB. The spacecraft requirement was be under 500 KB in the flight environment. The first five months include seven builds that were focused solely on reducing the size of the system. Once the size requirement was achieved, subsequent builds focused on modifying the capabilities of AutoCon™ to support the flight system interfaces and mission requirements. As new capabilities were added, the code size was re-evaluated and reduction efforts were implemented. The first step for size reduction was to remove unnecessary capabilities. Since AutoCon™ is object-oriented this simply entailed removing whole objects from the build of the system. The first two builds removed a total of 160 of the 265 classes that were not necessary for the flight requirements. To remove additional objects, AutoCon™ was modified to use the math functions of the Attitude Control System (ACS) in place of its own math classes. The final core system includes 85 C++ classes. The next step in code reduction was to eliminate code methods from within the remaining classes. These include file input and debug output and methods associated with unnecessary coordinate transformations and flight regimes. Another size reduction technique explored was compiler flag settings. The flag for compiling with debug was turned off for the 7th build; it was activated for all previous builds. Static allocation and initialization of arrays provided significant savings. The change in initialization saved over 36 KB of space.

3.3 - Parsed Flight CPU Execution

The EO-1 flight environment system requires that the executing tasks use CPU time in slices. The AutoCon™ system receives a CPU slice every two seconds and is expected to complete processing within a fraction of a second. Failure to complete processing within 5 seconds results in the flight system performing a warm restart. AutoCon™ which operates through the sequential execution of scripted commands, retained its original design as a parsed execution system for UI messaging with a script command executing and returning process to the controlling UI one command per time slice. The commands, however, took varying amounts of time to execute. Some commands exceeded the allotted time slice. The commands that exceeded the time slice were segmented to complete a portion of the command, return processing to the main system, and continue with the next part of the command segment at the next time slice. To make the best use of the available time slice, the commands that used very little processing time were identified and grouped with the next command within the same time slice. Near the final development stages a new capability to provide burn durations and burn start times on the whole second was implemented. It required only a few lines of additional code, but caused AutoCon™ to consistently use too much CPU time when targeting. The customer supplied fmod math function was replaced with a less CPU intensive algorithm to resolve this issue.

3.4 - Testing

Upon successful compilation of AutoCon™ in the flight environment, testing began. A series of benchmark tests using AutoCon™ in the Windows NT environment were defined. The same tests were then duplicated in the flight environment and the results compared. The flight environment tests were designed to exercise table uplink, telemetry downlink, commanding, as well as computational accuracy. Initially, the results between the Windows NT and the Mongoose V were unacceptably different. AutoCon™ is required to propagate two spacecraft states (EO-1 and Landsat-7) into the future and plan any maneuvers required during that time to maintain the formation. The propagation differences between the benchmark NT result and the flight environment result were 540 meters RSS after 36 hours of propagation, to large to plan a maneuver. Coding of spacecraft drag caused the model to return an atmospheric density of zero without returning a processing error. The problem was found to be in a conditional statement in the Jacchia-Roberts drag model class, where a variable was set to the result of a function call. While the code complied with ANSI standards, the compiler did not handle the syntax properly. The problem was fixed by adding an interceding function call. After resolution of this problem, there was still approximately 36 meters of propagation difference using the full force model. To investigate this difference, separate results were produced for each force in the force model. This force-by-force testing showed that the largest discrepancy was related to the effects of the moon. Further tests revealed that on the Mongoose, after the initial calculation of the moon's position, the moon's position remained static while the epoch was being advanced. Inspection of the code revealed the same type of structure found in the drag problem- the calculation and testing of a value in a conditional statement. Further analysis revealed that the error was being caused by a chained assignment. The correct path was being executed, but the variables were not properly updated, causing the subsequent test to incorrectly bypass re-calculating the position of the sun and moon. The correction required breaking the chained assignments into separate statements. Chained assignments were subsequently broken up in all other parts of the code even though they were not currently experiencing problems. Once the compiler issues were resolved, the comparisons between the Windows NT and the Mongoose V results agreed.

4 - GPS DATA SMOOTHER

Ensuring an accurate input state to AutoCon™ is crucial to EO-1 formation control. On EO-1, the GPS TENSOR™ receiver software using a Kalman filter processes raw GPS data that consists of pseudorange and Doppler measurements. Orbital states obtained from the GPS TENSOR™ have RMS position and velocity errors of 35.7 m and 5.2 cm/s, respectively [Dibb01]. The requirement for an input state for the GSFC maneuver algorithm is that the errors in radial position and velocity be no larger than 5 m and 2 cm/s, respectively. Thus an additional stage of optimization was provided. This optimization has been implemented as a discrete fixed interval data smoother, which uses the Rauch, Tung and Striebel algorithm [Rauc65]. The Kalman filter underlying the smoother is adapted from the GSFC GPS Enhanced Orbit Determination Experiment (GEODE)-lite software [Carp97]. This model has been enhanced to incorporate the velocity components that are also provided by the GPS TENSOR™ software, and to incorporate the Jacchia-Roberts drag model, required by AutoCon™. The GPS data smoother is implemented as an object that holds the final smoothed state. This smoothed state is the input state for maneuver planning. In testing scenarios, the definitive smoother state has proven to be nearly always better in comparison to a reference ephemeris than the Kalman estimate. Despite the lag in time between the Kalman and smoothed estimates, multi-day propagation of the smoothed solutions are comparable to, and often much better than, the propagated Kalman estimates.

5- MANEUVER CONTROL ALGORITHM DESCRIPTION

The GSFC FQ algorithm for formation flying solves the formation maintenance problem by combining a modified Lambert problem and Battin's 'C*' matrix [Batt87]. The algorithm enables the spacecraft to autonomously execute complex three-axis orbital maneuvers [Folt97]. For minimum fuel use, the EO-1 maneuvers were restricted to in-plane components. EO-1 monitors the Landsat-7 position and performs maneuvers designed to maintain the relative position imposed by the formation requirements. The FQ algorithm plans maneuvers by determining a Keplerian path which will transfer the EO-1 spacecraft from some initial state, (r_0, v_0), at a given time, t_0, to a target state, (r_t, v_t), at a later time, t_t so as to maintain the formation. A desired state is also computed by back propagating the target state to find the non-maneuvered initial state (r_d, v_d) that

EO-1 would need at time t_0. These states give rise to differenced states, δr and δv. The FQ algorithm computes state transition matrices for calculation of the maneuver ΔVs. Selecting initial conditions prescribed at a time t_0 a state transition matrix, $\Phi(t_1,t_0)$, can be constructed such that it will be a function of both t and t_0 and satisfy matrix differential equation relationships. Having partitioned the state transition matrix, $\Phi(t_1,t_0)$ for time $t_0 < t_1$ we use symplectic properties in equation 1 (assuming a reversible Keplerian path) to find the inverse where the matrix $\Phi(t_0,t_1)$ is based on a propagation forward in time from t_0 to t_1 and is sometimes referred to as the navigation matrix, and $\Phi(t_1,t_0)$ is based on a propagation backward in time from t_1 to t_0 and is sometimes referred to as the guidance matrix. When the fundamental matrix C* is defined as $C^* \equiv V^* R^{*-1}$, see equation 2, it can be found using the submatrices that $C^*\delta r = \delta v_0$ becomes the velocity deviation required at time t_0 as a function of the measured position error δr at time t_0 if the spacecraft is to arrive at the reference position. With parameters derived from the Gauss problem of planar motion, the target and desired states, and the F and G series using universal variables, R and V are defined. From sub-matrices, the C* matrix is then computed and the expression for the impulsive maneuver generated, see equation 3. For EO-1's orbit a long, iterative window requiring many small burns is not necessary, and ΔV maneuvers resemble a Hohmann transfer performed over 1½ revolution.

$$\phi^{-1}(t_1,t_0) \equiv \phi(t_0,t_1) \equiv \begin{bmatrix} \phi_1(t_0,t_1)\phi_2(t_0,t_1) \\ \phi_3(t_0,t_1)\phi_4(t_0,t_1) \end{bmatrix} \qquad \phi^{-1}(t_1,t_0) \equiv \begin{bmatrix} \phi^T{}_4(t_1,t_0)\phi^T{}_2(t_1,t_0) \\ \phi^T{}_3(t_1,t_0)\phi^T{}_1(t_1,t_0) \end{bmatrix} \tag{1}$$

$$C^*(t_0) = V^*(t_0)\left[R^*(t_0)\right]^{-1} \tag{2}$$

$$\Delta V = \left[C^*(t_0)\right]\delta r_0 - \delta v_0 \tag{3}$$

6 – Safety

One of the major concerns of the EO-1 mission is to make sure that the autonomous maneuver system is as safe as possible. There was considerable concern that an autonomous system would enable a command that would result in an extremely long maneuver duration and jeopardize the mission. Several safeguards were created to alleviate such concerns. These include a standard of 48 hours notice before any planned maneuver (the time length is adjustable) and a phased in approach to autonomy. The 48-hour notice gives the ground time to review the planned maneuver before its execution. These include a monitor mode, which allows burn plans to be generated and reviewed, a manual mode, which allows maneuvers to be predicted but not implemented and a semi-autonomous mode, which allows burn plans and the resulting command to be verified by the ground before execution. The autonomous mode allows the generation and execution of a maneuver, but the maneuver information is still available two days prior to a maneuver event. The autonomous mode can be interrupted by ground command. Also, the autonomous mode is limited to a specified number of burns before it automatically transitions back to manual mode. In addition to AutoCon™'s built-in safety features, the attitude control system (ACS) limits all burns to 60 seconds or less. The stored command sequence also limits burn duration. Additionally, EO-1 has a watchdog timer to make sure no task, such as AutoCon™ exceeds CPU utilization, depriving other critical tasks processing time. Finally, the spacecraft has a safehold mode that can disable AutoCon™, if necessary.

7- OPERATIONS AND VALIDATION

Validation of the safety modes to ensure proper execution of AutoCon™ included six weeks of evaluation of the monitor mode and several weeks of evaluation of the smoother. This early validation proved the functionality and capabilities of AutoCon™ and the FQ algorithm. AutoCon™ was run in a continuous mode to provide continuous maneuver planning and data ingest. Over 300 onboard maneuver test plans were generated in that time frame. After this early confirmation, the remainder of the mission was dedicated to the validation of executed EO-1 maneuvers. A total of nine maneuvers were planned and validated onboard in the manual, semi-autonomous, and fully autonomous modes using the FQ algorithm. All were used to plan a formation flying maintenance maneuver. The commands generated onboard in semi-autonomous mode were enabled via ground command, while the fully autonomous mode were placed in the absolute time sequence with other spacecraft commands at approximately 12 hours before the maneuver execution. The locations and epochs of these maneuvers were chosen to meet the EO-1 orbit and science requirements in response to Landsat-7 maneuvers or to an EO-1 maneuver to maintain formation. Table 2 presents the maneuver mode, absolute ΔV difference, and absolute percentage difference in the quantized maneuvers. It shows that there is excellent agreement between the onboard system and the ground validation system. Note that the percent error of the first ΔV computed from the Folta-Quinn algorithm ($\Delta V1$) ranges from 0.000154% to 1.569%, the larger difference being the result of differences in the input target and desired states after propagation. The larger residual of the second velocity matching ($\Delta V2$) is due to the spacecraft 1-second quantization of a 6 second velocity-matching maneuver. This difference is due to the spacecraft system yielding a maneuver duration near the mid-point that rounded down while the ground system rounded up. Additionally, a comparison was performed against the original algorithm in Matlab with excellent results as the percentage differences were all under 0.005%. Final autonomous formation flying maneuvers were completed on June 27 and November 14, 2001 and used GPS as the input state. Evaluation of these maneuvers show that the quantized maneuver ΔV was only 0.0005% and 0.001% respectively different from the ground. Figures 3 and 4 are derived from the definitive ephemerides of EO-1 and Landsat-7 and were used as an independent check to verify that the formation requirements of 450km with a tolerance of +/- 42.5km and the relative ground track of +/-3km are met. Additionally, relative eccentricity and semi-major axis of the frozen orbit eccentricity were also maintained as a result of the formation flying maneuvers. Figure 3 shows the general formation flying evolution of the alongtrack and radial components presented in a Landsat-7 centered rotating coordinate system with the radial direction (ordinate) being the difference in radius magnitude and the alongtrack direction (abscissa) being the arc between the position vectors. . Figure 4 shows effect on the mission groundtrack by the formation flying maneuvers and that it meets NMP requirements. The time span is over the duration of the formation flying demonstration of 5 months from February 2001 to June 2001. Performance tests of the GPS TENSOR™ were conducted early in the mission. In an attempt to characterize the behavior of the smoother to real input data, GPS TENSOR™ states and smoother states were compared to ground based S-Band orbital solutions. Figure 5 presents the position comparisons,

Table 2. Quantized Maneuver Results

Date	*Mode*	*Onboard ΔV1* cm/s	*Onboard ΔV2* cm/s	*Ground ΔV1 Difference* cm/s	*Ground ΔV2 Difference* cm/s	*% Diff ΔV1 vs. Ground* %	*% Diff ΔV2 vs. Ground* %
11/14/02	Auto (GPS)	4.162500	1.8500000	0.0000045	0.2313035	.00108159	12.502891
06/27/02	Auto (GPS)	5.359500	4.3387000	-0.000003	0.2552395	-0.0005535	5.8828571
06/07/02	Auto	4.9854078	0.0000000	0.0000001	0.0000000	0.00015645	0.00000000
05/17/02	Auto	2.4376271	3.7919202	0.0000003	0.0000002	0.00111324	0.00053176
05/10/02	Semi-Auto	1.0831335	1.6247106	0.0000063	-.0026969	0.05852198	-14.2361365
05/03/02	Semi-Auto	2.3841027	0.2649020	0.0000000	0.0000000	0.00011329	0.00073822
04/26/02	Semi-Auto	5.2980985	1.8543658	-0.0008450	-0.0002963	-1.56990117	-1.57294248
04/08/02	Manual	2.1915358	5.2049883	0.0000004	-0.0332099	0.00163366	-0.00022414
04/05/02	Manual	3.5555711	7.9318735	-0.0000003	-0.0272687	-0.00081327	3.57089537

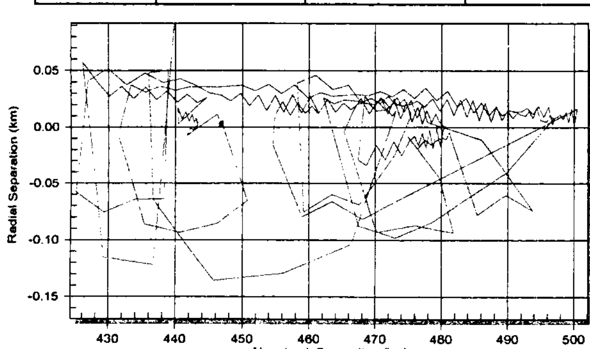

Figure 3. EO-1 Formation Evolution

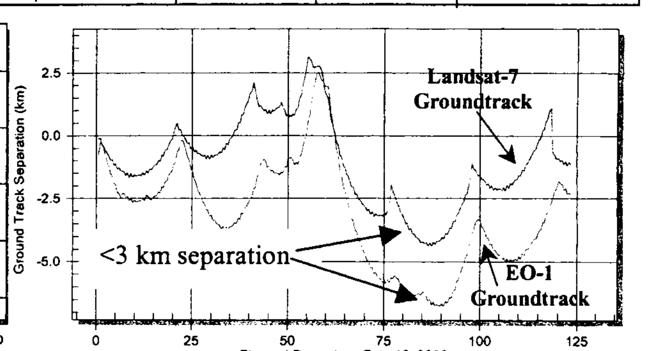

Figure 4. EO-1 and Landsat-7 Relative Groundtrack

including overall magnitude, and broken down into radial, along-track and cross-track components. The lines connecting the smoother points are not intended to indicate the errors between points, but only to call attention to the individual points. Except for a few outlying points the smoother seems to be performing well, eliminating the large variations. [Dibb01]

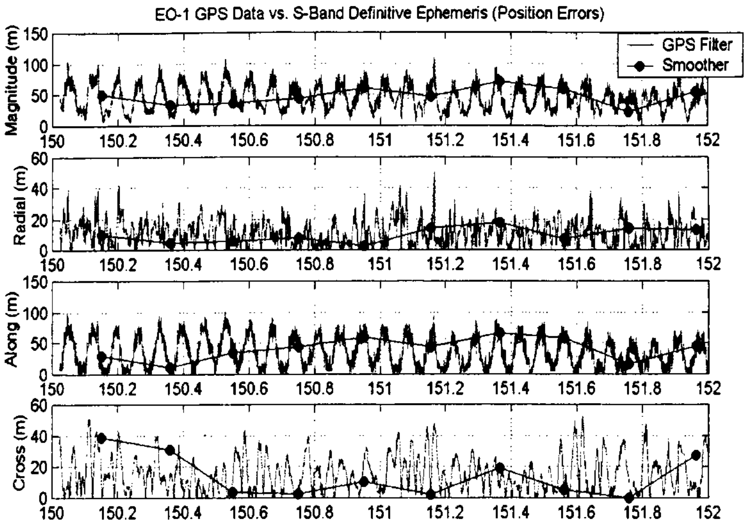

Figure 5 Comparisons of GPS States

8 - CONCLUSIONS

The need for low mission cost and multiple spacecraft support has made satellite orbit control autonomy a priority. True savings in cost and error prevention require onboard autonomy. New mission concepts that require spacecraft formations, tight orbit control, and constellations, all necessitate onboard maneuver capabilities.

Without formation control, many future science missions would not be possible. The AutoCon™ system onboard EO-1, NASA's first autonomous formation flying mission, demonstrated that autonomous control and formation flying technologies can be implemented. Additionally, on-orbit validation has demonstrated that the EO-1 formation flying requirements can be easily met using AutoCon™ and the GSFC formation flying algorithm. This system is envisioned for use on other NASA missions including the Global Precipitation Measurement mission (GPM), Earth Observing System (EOS) missions, Geosynchronous missions, and others. The EO-1 formation flying experiment establishes a demonstrated, validated fully non-linear autonomous system for formation flying, a reliable and tested flight software suite, a universal algorithm that can handle any orbit constraint, a proven GPS state smoother, a system that incorporates fuzzy logic for multiple constraint resolution, and use of multiple navigation inputs. Autonomous formation flying is here.

9 – REFERENCES

[Mrre97] Mission Requirements Request (MRR) for the Earth Orbiter-1 (EO-1), 501-NMP/EO-1 MRR01, NASA GSFC Code 501, January 1997.

[Folt97] D. Folta and D. Quinn, "A Universal three-axis Method for Controlling the Relative Motion of Multiple Spacecraft in Any Orbit," Proceedings of the AIAA/AAS Astrodynamics Specialists Conference, August 10-12, Boston, MA. 1998

[Guin97] J.Guinn, "Autonomous Navigation for the New Millennium Program EO-1 Mission", AIAA Guidance Navigation and Control Conference, AIAA 97-3816, August 1997.

[Chap01] K. Chapman, et al, "Autonomous Formation Flying from the Ground to Flight", AAS/AIAA Astrodynamics Specialist Conference, Quebec, Canada, July 2001.

[Batt87] R. Battin, An Introduction to the Mathematics and Methods of Astrodynamics, AIAA Education Series, (1987) Chapters 9 and 11.

[Rauc65] H.E. Rauch, F Tung, C Striebel, [RTS], AIAA J., vol. 3. no. 8, 1965

[Carp97] R. Carpenter et al., "Global Positioning System (GPS) Enhanced Orbit Determination Experiment (GEODE) Mathematical Specifications", Revision 1 March 1997, NASA GSFC Flight Dynamics Division/Code 550, Greenbelt, Maryland

[Dibb01] J. Dibble and M. Nicholson, " A Decentralized Kalman Filter and Smoother for Formation Flying Control of the Earth Observing -1 (EO-1) Satellite, AAS/AIAA Astrodynamics Specialist Conference, Quebec, Canada, July 2001.