Hardware-in-the-loop Demonstration of GPS-Based Autonomous Formation Flying

10-12 December 2008 at ESTEC, Noordwijk, the Netherlands

Toru Yamamoto [1], Simone D'Amico [2]

[1] *Japan Aerospace Exploration Agency (JAXA), 2-1-1 Sengen, Tsukuba-shi, Ibaraki, Japan, yamamoto.toru@jaxa.jp*

[2] *German Aerospace Center (DLR), Münchner Str. 20, 82234 Wessling, Germany, simone.damico@dlr.de*

ABSTRACT

GPS navigation is the essential technology for the autonomous formation flying in Low Earth Orbit (LEO). This paper presents a closed-loop testing system for hardware-in-the-loop (HIL) demonstration of GPS based autonomous formation flying. The developed closed-loop HIL simulation system allows an elaborate validation of GPS-based autonomous formation flying functionalities and performance in an integrated configuration including both navigation and guidance/control. The developed system is used to support a test campaign of the GPS-based autonomous formation flying experiment for the PRISMA mission which is being contributed by the German Aerospace Center (DLR). Key results of the tests are shown and discussed in this paper.

INTRODUCTION

Formation flying makes use of multiple satellites which work together in a coordinated manner to accomplish scientific and commercial mission objectives which normally need unrealistically large spacecraft structures. Many innovative formation flying missions such as virtual telescopes for astronomical observation, synthetic aperture radar interferometers and so on have been proposed in the last decades and are now under development.

GPS receivers are recognized as the most important source of onboard navigation for spacecraft orbits where GPS signals are available, especially in Low Earth Orbit (LEO). PRISMA [1], TanDEM-X [2], FFAST [3], and JC2SAT-FF [4] are typical examples of proposed autonomous formation flying missions in LEO. An elaborate Hardware-In-the-Loop (HIL) testing of Guidance, Navigation and Control (GNC) functionalities and performance is essential to the success of these missions.

A realistic HIL testing of a GPS-based onboard navigation system can be carried out by using a GPS Signal Simulator (GSS). GSS can produce artificial GPS Radio Frequency (RF) signals representative of those received by formation flying spacecraft. Normally it is possible to define an initial orbit of the GPS receiver which is subsequently propagated by a dedicated software which controls the GSS. This means that a complete orbital trajectory is defined a-priori and cannot reflect orbit and attitude maneuvers generated autonomously by the spacecraft on-board computer in real-time. This test typology is referred to as "open-loop" and does not allow a full validation of GNC algorithms in an integrated configuration. According to recent research [5], GPS relative navigation has the potential to achieve precise accuracy at the cm or even mm level. Therefore GPS-based navigation can serve the need of precise autonomous relative orbit control. Rather than the case of conventional orbit maintenance, characterized by 100m-order accuracy, and performed with ground-in-the-loop, precision control needs more frequent executions of maneuvers. In this latter case the expected thrust levels are relatively low and the actuation is performed in a continuous fashion. Generally maneuvers cause a temporal degradation of the navigation accuracy, especially if dynamic filtering is applied, and can cause undesired reactions of the autonomous guidance and control algorithms. The strong coupling between a navigation system and the guidance/control functions has to be reproduced by an efficient HIL test-bed through an integrated closed-loop configuration.

Leitner [6] firstly developed a closed-loop HIL simulation environment for GPS based formation flying which is called Formation Flying Test Bed (FFTB). Burns [7] did formation control demonstration using the FFTB and Orion GPS receivers. Gill [8] also used the FFTB to do closed-loop HIL simulation for autonomous GPS based formation flying. He used the Orion GPS receivers and a Power PC navigation processor for the real-time experiment. The control law was based on the Lyapunov's method to demonstrate autonomous formation acquisition and keeping.

This paper presents a closed-loop testing system for hardware-in-the-loop (HIL) demonstration of GPS based autonomous formation flying. The developed closed-loop HIL testing system allows an elaborate validation of GPS-based autonomous formation flying functionalities and performance in an integrated configuration including both navigation and guidance/control. In this context the PRISMA formation flying mission, to be launched in June 2009, offers the ideal background to test and validate GNC flight software. The developed closed-loop HIL testing system is used to support a test campaign of the GPS-based autonomous formation flying experiment for the PRISMA mission which is being contributed by the German Aerospace Center (DLR). Engineering models of the Phoenix-S GPS receivers to be flown on PRISMA are used for the test. Furthermore the prototype flight software for GPS-based GNC of PRISMA is validated by the closed-loop HIL simulation. Key results of the tests are shown and discussed in this paper.

HIL TESTING FOR GPS BASED FORMATION FLYING

In this section the general concept and the essential difference between "open-loop" and "closed-loop" HIL simulations for GPS-based autonomous formation flying is discussed. Fig. 1 depicts the typical configuration of an open-loop HIL simulation.

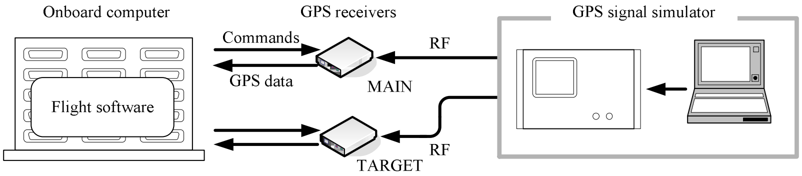

Fig. 1 Open-loop HIL simulation of GPS-based GNC system for autonomous formation flying.

GSS computes the spacecraft orbit and attitude in real-time and generates emulated GPS signals based on the computed motion data. GPS receivers are fed with the radio frequency signals and produce GPS navigation solutions, raw measurements (e.g., pseudorange, carrier phase, Doppler data) and broadcast ephemerides data of the GPS satellites. The GNC flight software installed in an onboard computer receives the data from the GPS receivers and performs filtering for on-board relative and absolute navigation. In addition the flight software carries out guidance and control tasks by means of the real-time navigation data. If a maneuver is considered to be necessary to maintain or reconfigure the formation according to predefined specifications, then thruster activation commands are typically generated to perform orbit corrections.

In general terms orbit control maneuvers affect the motion of formation flying spacecraft and in turn are reflected into the GPS measurements in a timely manner. As a consequence the subsequent GPS-based GNC computations are affected, with a global effect which circulates repeatedly through the closed chain of causality as time goes by.

Understanding how orbital maneuvers influence the navigation performance and how guidance/control algorithms react to the behaviour of the on-board real-time navigation is essential to validate a system which performs autonomous orbit control. This aspect is of special interest when control accuracy requirements are demanding and frequent orbital maneuvers have to be performed, resulting in a stronger coupling between on-board navigation and guidance/control.

In an open-loop test configuration the maneuver commands generated by the flight software do not have any effect on the actual orbit and attitude computed by GSS. Therefore onboard guidance/control functions can not see the effect of commanded maneuvers and react accordingly. An open-loop configuration is thus not suited to test and validate autonomous orbit control functionalities.

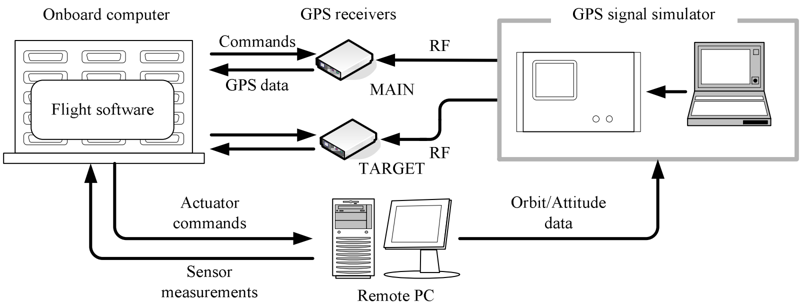

Fig. 2 Closed-loop HIL simulation of GPS-based GNC system for autonomous formation flying.

Fig. 2 shows the schematic of a closed-loop HIL simulation of a GPS-based GNC system. In this case the orbit/attitude computations are performed by a dedicated remote PC, and not by GSS. The remote PC provides real-time motion data to GSS. GSS generates emulated GPS signals based on the received motion data. Maneuver commands produced by the flight software are sent to the remote PC and are then reflected into the spacecraft dynamics in real-time.

The described closed-loop configuration is able to reproduce the real chain of causality. The orbit control maneuvers commanded by the guidance/control tasks affect the dynamic system, as a consequence new GPS measurements are collected in real-time and used by guidance/control algorithms like during the actual flight.

IMPLEMENTATION OF CLOSED HIL SYSTEM

In this section a detailed implementation of the developed closed-loop HIL simulation system is explained. Fig. 3 shows a schematic of the system under study.

The main elements of the HIL test-bed are a remote PC and a SPIRENT GSS [9] (cf. Fig. 3). Users of the formation flying test-bed can connect custom GPS receivers and integrate the GNC flight software to be tested. The software called Real-Time-Propagator (RTP) is the backbone of this HIL simulation system. The RTP is a simulation software which can compute orbit, attitude and model sensors and actuators of a multiple spacecraft system. The key feature of the RTP is its capability to perform step work of simulation at the timing controlled by a timer counter which is continuously synchronized with GSS.

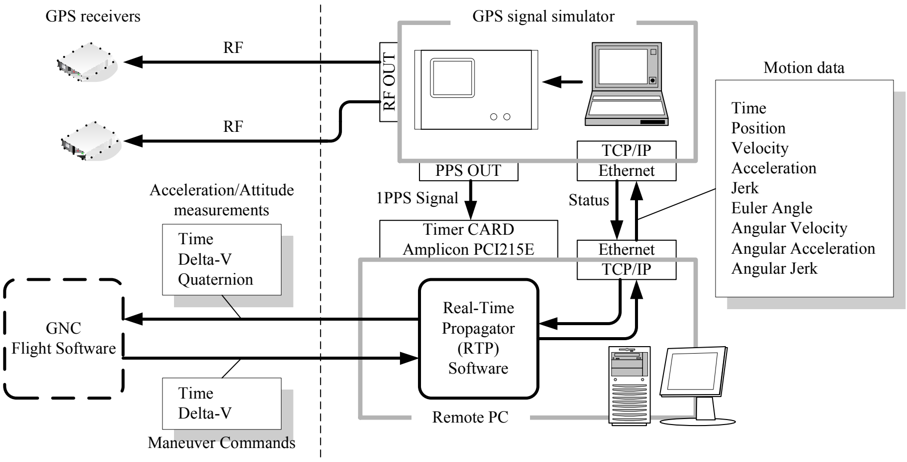

Fig. 3 Schematic of developed closed-loop HIL testing system.

The absolute time reference of the system is the oscillator of GSS. Therefore it is required that all other participants of the closed-loop system are synchronized with the clock of GSS. The GSS provides TTL level Pulse-Per-Sec (PPS) signal which can be used as the timing reference. To this end a dedicated timer-card is installed in the remote PC. It can provide a timer counter which synchronizes with the clock of GSS by receiving PPS signal and resetting its internal counter periodically. RTP can perform step work of simulation with continuous synchronization by using the timer counter.

RTP provides real-time motion data to GSS via TCP/IP interface. Motion data includes orbit/attitude information of each spacecraft and epoch time (cf. Fig. 3). The adopted force model for orbit propagation includes the Earth's gravity field (GGM02C model up to 70 in degree and order), atmospheric drag (Harris-Priester atmospheric density model), solar radiation pressure, gravity of Sun and Moon, and solid Earth tides.

RTP can receive maneuver commands from the GNC flight software. The maneuver command includes the delta-V and the execution time of the desired impulsive maneuver. RTP incorporates this information into the orbit propagation after adding maneuver execution errors in terms of net force and direction of the thrust vector. The effect of a maneuver is then reflected into the motion commands which are provided to the GSS in real-time.

On top of modeling the natural environment and actuators, RTP can also provide eventual sensor measurements generated on the spacecraft. In the context of the PRISMA mission, RTP generates delta-V information as measured by the accelerometers and attitude quaternions as measured by the star-trackers. In general this kind of sensor measurements are required by a flight software to perform navigation, guidance and control functionalities.

A good synchronization between RTP and GSS is mandatory in order to realize a rigorous real-time simulation. If a satisfactory level of synchronization is not achieved, two significant problems can occur as discussed in the sequel.

First of all GPS measurements induced by GSS and other sensor measurements produced by RTP could become inconsistent. Considering that the GNC flight software receives both types of measurements in real-time, different measurement epoch times could cause malfunctions. For example if GPS measurements and attitude measurements have time-tag differences up to several seconds, then eventual corrections of the GPS antenna offsets with respect to the center of mass could be affected by larger errors.

Second of all motion data exchanged between RTP and GSS could become inconsistent. In particular motion data generated by RTP represent the absolute "true" motion data of the HIL simulation system. On the other hand, GSS requires always "current" motion data since it must generate emulated GPS signals continuously in real-time. If RTP is delayed compared to GSS, then GSS performs an extrapolation by utilizing the most recent motion data provided by RTP to produce current GPS signals. Hence when the latency becomes large, the difference between "true" motion data (generated by RTP) and actually used "extrapolated" data (computed by GSS) can be unacceptably high.

Fig. 4 and Fig. 5 illustrates key verification results of the synchronization between RTP and GSS. Fig. 4 depicts the latency of RTP relative to GSS. The left plot shows the resulting drift when no synchronization by timer card is performed. In this case the delay of RTP gradually grows as time passes since both RTP and GSS run independently. The RTP and GSS clock difference amounts to about 5 seconds after a 24 hours simulation. This amount of delay is not acceptable for a rigorous validation of the GNC flight software. On the contrary, the right plot of fig. 4 shows an almost zero time delay when the synchronization is appropriately performed.

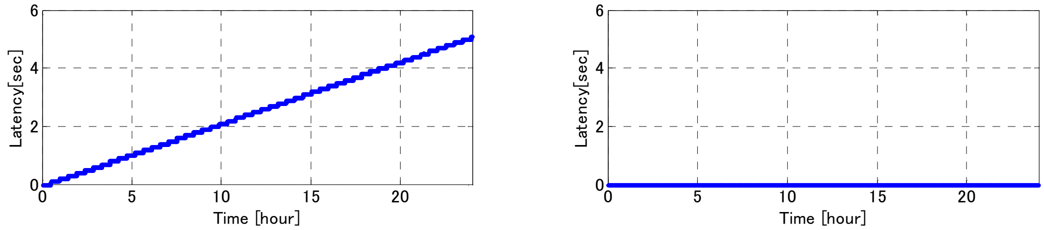

Fig. 4 Latency of RTP relative to GSS (Left: without sync., Right: with sync., 24Hour)

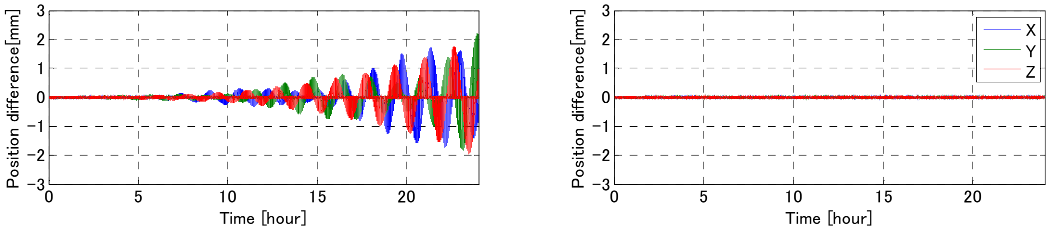

Fig. 5 Position difference between RTP and GSS (Left: without sync., Right: with sync., 24Hour)

Fig. 5 shows the position difference between the "true" trajectory generated by RTP and the trajectory actually used to generate GPS signals by GSS. On the left plot growing position differences can be seen when the synchronization is not performed. Even if RTP has 5 seconds latency relative to GSS, the position difference remains less than 3 mm after 24 hours. This is due to the fact that extrapolation is carried out by using not only position/velocity but also acceleration/jerk. Thus inconsistency of motion data is not a serious problem for short term simulations. The right plot of fig. 5 shows that there is no growing position difference when the synchronization is performed.

HIL TEST FOR PRISMA FORMATION FLYING

The developed closed-loop HIL simulation system is used to demonstrate and validate the functionalities and performance of the DLR contributions to the PRISMA formation flying mission. In this section, details of testing scenario and results are shown and discussed.

PRISMA Formation Flying Mission

Fig. 6 shows an illustration of PRISMA and the architecture of the redundant GPS sensor system contributed by DLR. PRISMA is a technology demonstration mission for satellite formation flying and in-orbit servicing. It comprises the fully maneuverable Main satellite and the smaller passive Target satellite. The German Aerospace Center (DLR) provides various key contributions to the PRISMA formation flying. These comprise a redundant GPS hardware architecture for the two spacecrafts, a real-time navigation software to support formation flying during all phases, and dedicated experiments for absolute and relative orbit control [1].

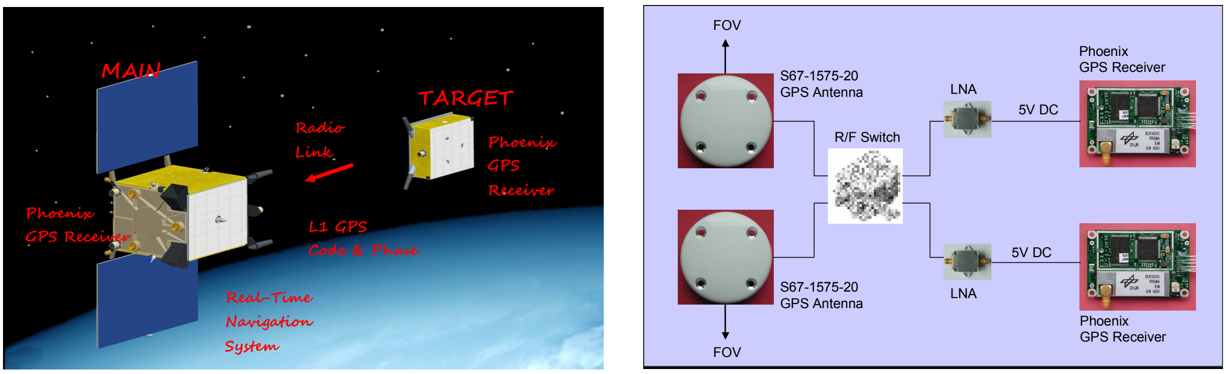

Fig. 6 Illustration of PRISMA formation flying (left) and PRISMA GPS receiver system (right)

PRISMA GPS-based GNC software

Fig. 7 shows a simplified architecture and data interface of the GPS-based autonomous formation flying software for PRISMA developed by DLR [5].

The GPS interface (GIF) is directly fed with GPS messages issued by the Phoenix-S GPS receivers on-board MAIN and TARGET. GIF handles GPS raw data formats and ephemerides, and performs data sampling as well as coarse editing prior to the GPS-based orbit determination.

The GPS-based Orbit Determination (GOD) implements an extended Kalman filter to process GRAPHIC observables as well as single difference carrier phase measurements from MAIN and TARGET. Attitude data from both spacecraft are applied to correct for the GPS receivers antenna offset with respect to the spacecraft center of mass. Furthermore, a history of maneuver data is provided to GOD and taken into account in the orbit determination task. GOD performs a numerical orbit propagation which is invoked after the measurement update and provides orbit coefficients for interpolation to GOP for both spacecraft.

The GPS-based Orbit Prediction (GOP) module interpolates the orbit coefficients provided by GOD and finally supplies the various GNC core functions as well as the PRISMA payload with continuous position and velocity data of MAIN and TARGET. Due to the different data rates of the GPS-based navigation modules, orbit maneuver data have to be taken into account in both GOD and GOP. In particular at each GNC step, the GOP task accounts for maneuvers which have not been considered by GOD in the last orbit determination/prediction process.

The Autonomous Formation Control (AFC) module receives the current MAIN and TARGET spacecraft position and velocity from the GOP module. Navigation data are used to detect deadband violations and plan maneuvers to maintain or reconfigure the geometry of the formation. If a maneuver is necessary, appropriate commands containing maneuver time and size are sent to the actuators. Details on the orbit control algorithms can be found in [10].

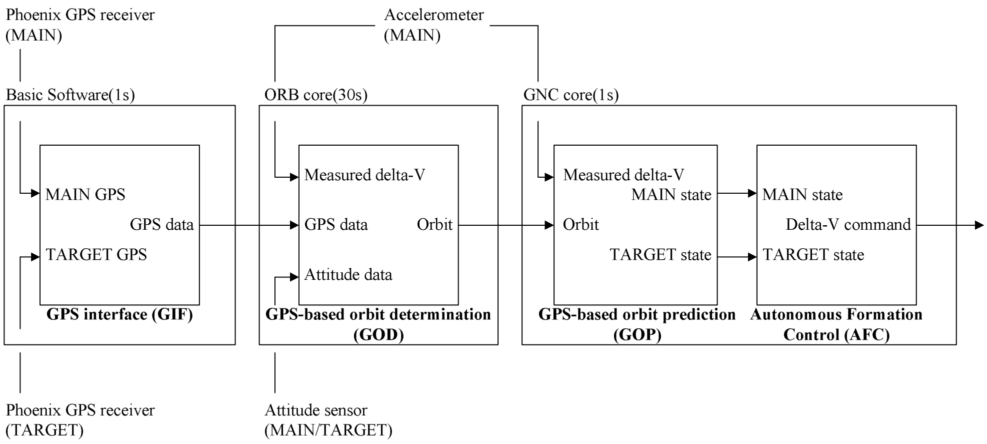

Fig. 7 PRISMA GPS-based autonomous formation flying GNC software architecture

Closed-loop HIL System for PRISMA GPS-based formation flying

Fig. 8 is a schematic of the closed-loop HIL simulation configuration used for PRISMA. Two Phoenix-S GPS receivers get emulated GPS signals from GSS and output GPS navigation solutions, raw measurements, and broadcast ephemeredes data of the GPS satellites. The PRISMA prototype flight software is executed in a Matlab/Simulink environment on the remote PC. The flight software receives data from the GPS receivers via a RS-232C interface, and performs GNC tasks as outlined in Fig. 7. Maneuver commands produced by the flight software and sensor measurements generated by RTP are shared and exchanged via shared memory I/F in the remote PC.

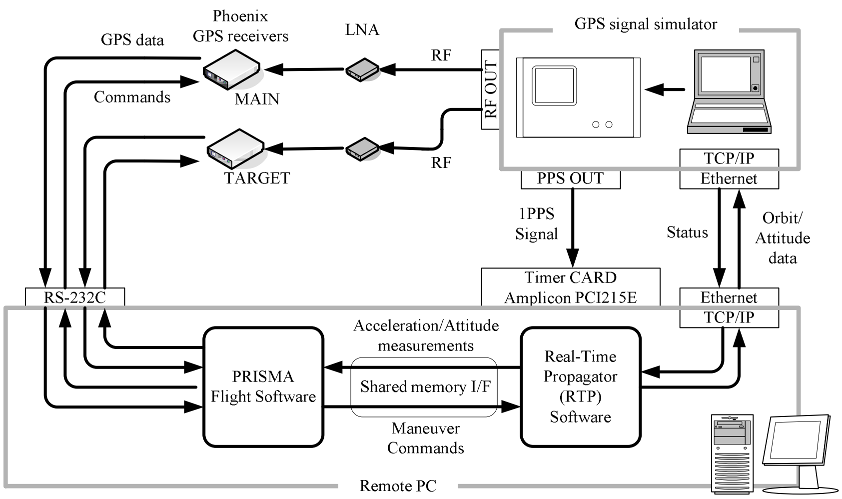

Fig. 8 Closed-loop HIL test configuration for PRISMA formation flying

In this configuration, the closed-loop chain of causality is realistically reproduced. Maneuver commands produced by the flight software affect the PRISMA spacecraft orbit, and new GPS measurements affect the guidance/control functions of the flight software in real-time. Such a simulation system enables a sophisticate validation of autonomous orbit control functionality and performance using actual flight GPS receivers and flight software.

Test Scenarios

Two representative test scenarios are simulated to validate the DLR flight software for PRISMA through the developed closed-HIL simulation system. Table 1 shows the major parameters for each scenario.

Table 1 Test scenarios

<table>
<tr><th colspan="3">Scenario Name</th><th>Formation Keeping</th><th colspan="3">Formation Reconfiguration</th></tr>
<tr><td colspan="3">Ionospheric Delay Model</td><td>Constant TEC, 1.0e17[electrons/m^2]</td><td colspan="3">Constant TEC, 1.0e17[electrons/m^2]</td></tr>
<tr><td colspan="3">GPS Ephemerides Error</td><td>3D RMS = 1.5[m]</td><td colspan="3">3D RMS = 1.5[m]</td></tr>
<tr><td colspan="3">Spacecraft Attitude</td><td>Earth Pointing</td><td colspan="3">Earth Pointing</td></tr>
<tr><td rowspan="6">Initial Osculating Orbital Element of MAIN</td><td>a_M</td><td>[m]</td><td>7087297.557</td><td colspan="3">7087297.557</td></tr>
<tr><td>e_M</td><td></td><td>0.00145443</td><td colspan="3">0.00145443</td></tr>
<tr><td>i_M</td><td>[deg]</td><td>98.185286</td><td colspan="3">98.185286</td></tr>
<tr><td>Ω_M</td><td>[deg]</td><td>189.891385</td><td colspan="3">189.891385</td></tr>
<tr><td>ω_M</td><td>[deg]</td><td>1.093764</td><td colspan="3">1.093764</td></tr>
<tr><td>M_M</td><td>[deg]</td><td>-1.093691</td><td colspan="3">-1.093691</td></tr>
<tr><td rowspan="7">Mean Relative Orbital Element</td><td colspan="2">Formation</td><td>Formation A</td><td>Formation A</td><td rowspan="7">$\Rightarrow$</td><td>Formation B</td></tr>
<tr><td>Da</td><td>[m]</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>aDe_x</td><td>[m]</td><td>$200\times\cos(100^\circ)$</td><td>$200\times\cos(100^\circ)$</td><td>$150\times\cos(90^\circ)$</td></tr>
<tr><td>aDe_y</td><td>[m]</td><td>$200\times\sin(100^\circ)$</td><td>$200\times\sin(100^\circ)$</td><td>$150\times\sin(90^\circ)$</td></tr>
<tr><td>aDi_x</td><td>[m]</td><td>$100\times\cos(40^\circ)$</td><td>$100\times\cos(40^\circ)$</td><td>$150\times\cos(60^\circ)$</td></tr>
<tr><td>aDi_y</td><td>[m]</td><td>$100\times\sin(40^\circ)$</td><td>$100\times\sin(40^\circ)$</td><td>$150\times\sin(60^\circ)$</td></tr>
<tr><td>aDu</td><td>[m]</td><td>9.3</td><td>9.3</td><td>12.5</td></tr>
<tr><td rowspan="4">Control Deadband</td><td>aDe_x *Soft*</td><td>[m]</td><td>0.5</td><td colspan="3">0.5</td></tr>
<tr><td>aDe_y *Hard*</td><td>[m]</td><td>2.0</td><td colspan="3">2.0</td></tr>
<tr><td>aDi_x *Soft*</td><td>[m]</td><td>0.5</td><td colspan="3">0.5</td></tr>
<tr><td>aDi_y *Hard*</td><td>[m]</td><td>2.0</td><td colspan="3">2.0</td></tr>
</table>

The formation is parameterized in terms of relative orbital elements ($a, aDe_x, aDe_y, aDi_x, aDi_y, aDu$) as described in [11]. In the formation keeping scenario these relative orbital elements have to be maintained by AFC within the prescribed deadbands as listed in Table 1. In the formation reconfiguration scenario, the relative eccentricity vector and the relative inclination vector have to be changed by AFC so to transfer the formation geometry from “Fromation A” to “Fromation B”.

Test Results and Discussion

Fig. 9 shows key results out of the formation keeping scenario.

The control tracking error is depicted in Fig. 9 (upper four plots) as the difference between the desired relative orbital elements and the actual values. The absolute and relative navigation errors are shown in 5th and 6th plots. The bottom (7th) plot shows the delta-V for each in-plain and out-of-plain maneuver. The orbit control deadbands (Soft and Hard) are also shown on the plots for the relative eccentricity vector (aDe_x, aDe_y) and the relative inclination vector (aDi_x, aDi_y). The relative semi-major axis (Da), the relative eccentricity vector and the relative mean argument of latitude (aDu) are controlled by pairs of in-plain maneuvers in radial direction separated by half an orbital period interval. The relative inclination vector is controlled by out-of-plain maneuvers only.

The relative semi-major axis tracking error is affected by a small bias which is intentionally introduced by AFC to compensate the drift of the relative mean argument of latitude due to differential accelerations caused by the J2 gravity term and the atmospheric drag. The relative eccentricity vector and the relative inclination vectors are properly moved from one side of the deadband to the opposite one in order to exploit their natural drift caused by J2. Both control tracking errors remain well within the hard deadband thresholds. The relative mean argument of latitude experiences shifts up to 2 meters in between the execution of the manuever pair due to the radial thrust direction. The deterministic feedback control law is shown to work properly and demands sparse impulsive maneuver executions every three orbital revolutions.

Overall statistical performance of absolute and relative navigation is 2.7m and 5.5cm (3D RMS), respectively. The accuracy is temporary degraded due to the effect of maneuvers. However the amount of the effect remains small enough since the degradation is degreased by accounting the measurement information of the accelerometer into dynamic filtering of the navigation software. Hence the guidance/control algorithms are not puzzled by the temporal degradation.

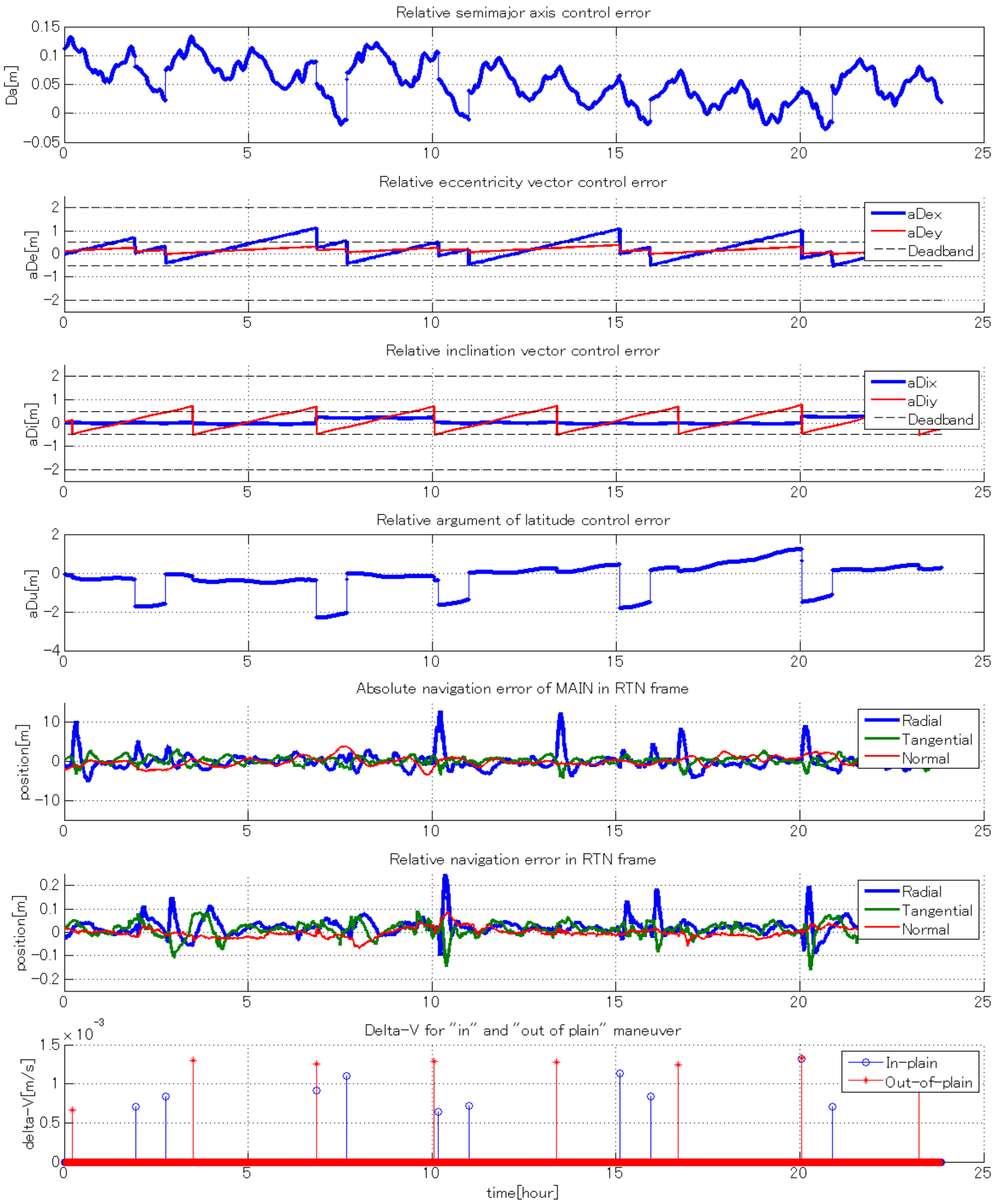

Fig. 9 Key results of formation keeping scenario
Control tracking error expressed in terms of relative orbital elements (1st ~ 4th plots),
Absolute and relative navigation error expressed in RTN frame (5th and 6th plots),
Delta-V for each maneuver (7th plot)

Fig. 10 shows the results of the formation reconfiguration scenario.

This scenario demonstrates an autonomous formation reconfiguration experiment where the constellation geometry is modified from "Fromation A" to "Formation B". The arrows in Fig. 10 show how the relative eccentricity vector and relative inclination vector are changed by the reconfiguration maneuvers. In fact the new relative orbital elements match the expected values as prescribed via telecommand in Table 1. In the right plot of fig. 10 the transformation from the relative motion ellipse of "Formation A" to the target ellipse of "Formation B" is shown.

Throughout these two representative test scenarios, functionalities and performance of formation keeping and formation reconfiguration for the PRISMA formation flying are investigated with realistic HIL test-bed in an integrated closed-loop configuration.

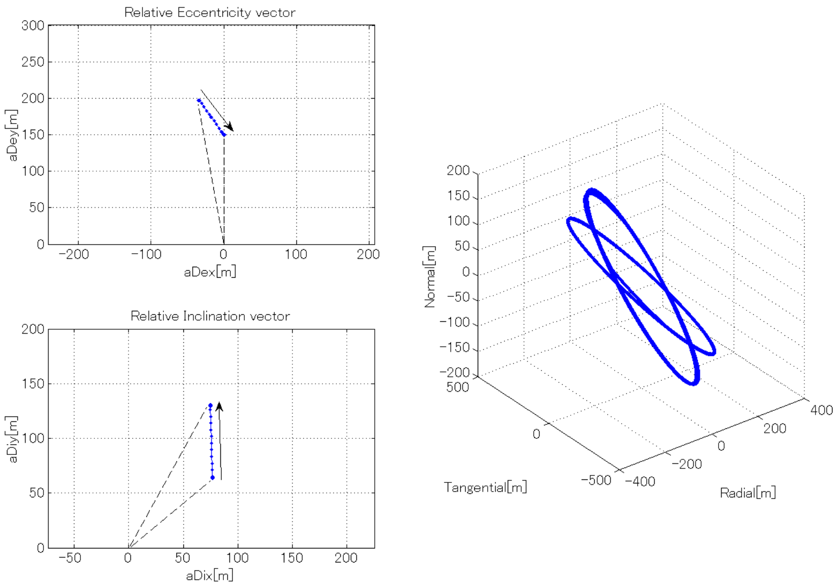

Fig. 10 Illustration of the formation reconfiguration test executed through the closed-loop HIL testbed. The relative eccentricity vector (left-top) and the relative inclination vector (left-bottom) are modified as desired by the orbit correction maneuvers. The resulting 3D relative motion is depicted on the right.

SUMMARY AND WAY FORWARD

A closed-loop test-bed for HIL simulation of GPS-based autonomous formation flying has been presented in this paper. The developed closed-loop HIL simulation system allows an elaborate validation in an integrated configuration including both navigation and guidance/control functionalities. The system is utilized to demonstrate and validate the flight software for the PRISMA autonomous formation flying mission. Test results from the realistic HIL simulation environment are good evidence to support performance and quality of the flight software. This work is the first opportunity a closed-loop HIL simulation system is applied for the test of the actual flight software of GPS-based autonomous formation flying.

In this paper the flight software simulations are conducted in a Matlab/Simulink environment. In the near future, the integrated closed-loop HIL test which is conducted on the PRISMA spacecraft with the flight software installed in the actual flight GNC computer is planned.

REFERENCES

[1] S. Persson, B.Jakobsson, E.Gill, “PRISMA – Demonstration Mission for Advanced Rendezvous and Formation Flying Technologies and Sensors”, IAC-05-B56B07, 56th IAC, Fukuoka, Japan (2005)

[2] A. Moreira et al., “TanDEM-X: A TerraSAR-X Add-on Satellite for Single-Pass SAR Interforometry”, IGARSS, Achorage, USA (2004)

[3] Tsunemi et al.: Formation Flight All Sky Telescope (FFAST), Proc. SPIE, Vol. 5900 (2005)

[4] Yoshihara K., Mierlo M., Ng A., Shankar Kumar B., De Ruiter A., Komatsu Y., Horiguchi H., Hashimoto H., “JC2Sat-FF: An International Collaboration Nano-Sat Project, Overview of the Project, System Analyses and Design”, Small Satellites Systems and Services - The 4S Symposium (2008)

[5] D'Amico S., De Florio S., Ardaens J.S., Yamamoto T., “Offline and Hardware-in-the-loop Validation of the GPS-based Real-Time Navigation System for the PRISMA Formation Flying Mission”, 3rd International Symposium on Formation Flying, Missions and Technologies, Noordwijk, The Netherlands (2008)

[6] Leitner J. “A Hardware-in-the-Loop Testbed for Spacecraft Formation Flying Applications,” IEEE Aerospace Conference, Big Sky, MT (2001)

[7] R. Burns, B. Naasz, D. Gaylor, J. Higinbotham, “An Environment for Hardware-in-the-Loop Formation Navigation and Control”, AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Providence, USA, AIAA 2004-4735 (2004)

[8] E. Gill, B. Naasz, T. Ebinuma, “First Results from a Hardware-in-the-Loop Demonstration of Closed-Loop Autonomous Formation Flying”, 26th Annual AAS Guidance and Control Conference, Breckenridge, USA, AAS 03-040 (2003)

[9] “SimGEN (Including SimLocate) User Manual – Software for the Spirent Range of Satellite Navigation Simulator Products”, DGP00686AAA, Issue 1-20 (2006)

[10] D'Amico S., Gill E., Montenbruck O., “Relative Orbit Control Design for the PRISMA Formation Flying Mission” AIAA Guidance, Navigation, and Control Conference, 21-24 Aug. 2006, Keystone, Colorado (2006)

[11] D'Amico S., Montenbruck O., “Proximity Operations of Formation Flying Spacecraft using an Eccentricity/Inclination Vector Separation”, AIAA Journal of Guidance, Control and Dynamics, 29/3 554-563 (2006)