GROUND SIMULATION OF AN AUTONOMOUS SATELLITE RENDEZVOUS AND TRACKING SYSTEM USING DUAL ROBOTIC SYSTEMS

Matthew J. Strube[1], **Andrew M. Hyslop**[1,2], **Craig R. Carignan**[1,3], **and Joseph W. Easley**[1,4]

[1]*NASA Goddard Space Flight Center, Greenbelt, Maryland, USA*
[2]*Emergent Space Technologies, Greenbelt, Maryland, USA*
[3]*University of Maryland, College Park, Maryland, USA*
[4]*SGT, Greenbelt, Maryland, USA*

ABSTRACT

A hardware-in-the-loop ground system was developed for simulating a robotic servicer spacecraft tracking a target satellite at short range. A relative navigation sensor package "Argon" is mounted on the end-effector of a Fanuc 430 manipulator, which functions as the base platform of the robotic spacecraft servicer. Machine vision algorithms estimate the pose of the target spacecraft, mounted on a Rotopod R-2000 platform, relay the solution to a simulation of the servicer spacecraft running in "Freespace", which performs guidance, navigation and control functions, integrates dynamics, and issues motion commands to a Fanuc platform controller so that it tracks the simulated servicer spacecraft. Results will be reviewed for several satellite motion scenarios at different ranges.

Key words: robotics, satellite, servicing, guidance, navigation, tracking, control, docking.

1. NOMENCLATURE

${}^B_A\boldsymbol{q}$	=	quaternion from frame A to B, 4th element is scalar
${}^B_A R$	=	rotation matrix from frame A to B
${}^C\boldsymbol{r}^B_A$	=	position vector from point A to B, given in frame C
${}^B_A T$	=	homogeneous transform from frame A to B
θ	=	euler angle
τ	=	commanded servicer body torque
${}^C\boldsymbol{\omega}^B_A$	=	angular rate of frame B with respect to A, given in frame C

Subscripts/Superscripts

0	=	robot base frame
a	=	secondary servicer axis alignment direction (perpendicular to approach axis) for capture
d	=	desired servicer body frame
DA	=	origin of capture axis, fixed in target frame
dcm	=	desired position for servicer center of mass
F	=	target satellite body frame
G	=	target satellite center of mass frame
i	=	Earth-centered inertial frame
RIC	=	radial (zenith), in-track, cross-track (orbit normal) - Hill frame
s	=	servicer/Argon body frame
scm	=	servicer center of mass
T	=	robot tool frame

2. INTRODUCTION

The prospect of performing satellite servicing operations on-orbit has driven the need for ground simulation of "free-floating" robotic systems to simulate satellite rendezvous and capture using a robotic spacecraft. In the past, many researchers have used planar robots floating on air-bearing tables to simulate rendezvous operations such as for Japan's ETS-VII experiments (Yoshida 2003). However, these systems fail to capture the full 3D dynamics of satellite rendezvous and capture. More recently, researchers have begun using six-axis industrial robots to emulate the full six degrees of freedom (DOF) of the chaser and target spacecraft. For example, (Xu et al. 2007) developed a real-time 3D simulation system to emulate the capturing process in 3D space using a pair of industrial robots. The German Aerospace Center (DLR) has developed a hardware-in-the-loop system for simulating proximity operations for on-orbit servicing missions (Benninghoff et al. 2011). Their testbed consists of two Kuka robots mounted on a 25 m rail system to simulate relative motion between the chaser and target spacecraft. The US Naval Research Laboratory (Obermark et al. 2007) achieved a similar capability using two cartesian gantries with 20 m max relative range. The NASA Satellite Servicing Capabilities Office (SSCO) system described in this article (see Fig. 1) was developed to provide an extended capability to simulate the spin and precessional motion of a target satellite over extended periods of time at near-fixed ranges in order to test the capabilities of the relative navigation sensor package.

Figure 1. NASA/SSCO relative navigation satellite ground simulator.

3. SSCO GROUND SIMULATOR

To simulate relative motion between satellites, one motion platform is sufficient but two increases the size of the workspace, allowing for more complex motion profiles. The intent is for the ground motion platforms to be tied to frames of interest inside a space simulation such that relative navigation sensors (mounted on one of the motion platforms) are presented with the correct relative pose (target position and orientation with respect to the sensor) to match what they would see if they were in space. For intuitive debugging, the choice was made to fix the lab (and bases of the motion platforms) to the target's orbital frame (RIC/Hill), use the Fanuc robot to simulate motion of the servicer satellite in the Hill frame, and for the Rotopod to simulate motion of the target satellite. Since the Hill frame is located at the target center of mass, the Rotopod needs only to simulate attitude motion, whilst the Fanuc uses all of its 6 degrees-of-freedom to simulate translation and attitude. The greatest workspace restriction is the relative position of the satellites, which must stay within a 1 m^3 box.

A system block diagram is shown in Fig. 2. The relative navigation sensor package to be tested, "Argon", uses visual and LIDAR sensors to feed pose estimation algorithms operating on flight-like processors, which estimate the position and orientation of the target with respect to the sensor package frame. These estimates are sent via raw ethernet to a ground station and relayed to a visualizer package which shows the compressed sensor image and overlays the on-board pose estimate with a desktop-based pose estimate for comparison. The estimates are relayed via ethernet to the Freespace simulator computer, which performs filtering, guidance and control, applies the forces and torques, and integrates the dynamics. Freespace then feeds the Hill-frame pose of the chaser satellite with respect to its pose at time zero to the Fanuc control station. It is assumed that the Fanuc is in the correct initial pose with respect to the Hill frame, using priori knowledge of the simulation's initial conditions. The Fanuc reacts to the pose requests and moves the chaser satellite accordingly. Motion requests are also produced for the target, but the interface with the Rotopod was not ready at the time of writing and so the target's motion was scripted (see Section 5). This separation of the target motion from the simulator is a flaw of the current setup, but the servicer spacecraft should respond to whatever relative motion it is presented (regardless of what the target is doing inside Freespace) by the virtue of loop closure around live lab measurements.

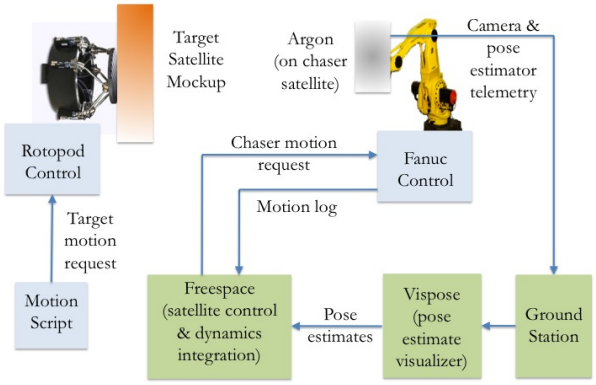

Figure 2. Block diagram of system.

The system control loop is shown in Fig. 3. Pose estimates are low-rate, 1-2 Hz (due to the computational intensity of the machine vision), but the satellite control gains are suitably low to maintain positive stability margins. The motion simulation part of the system is run at a much higher rate (100 Hz) with very stiff gains to achieve close agreement between the lab motion and motion inside the simulator. The setup allows for two important loop closure techniques; feedback of Freespace-generated truth-measurements (with option of artificial corruption) or measurements from the real hardware (Argon). Both techniques were used to debug Argon software and evaluate performance. To aid this process, sub-mm accuracy truth data was obtained with two Leica laser units.

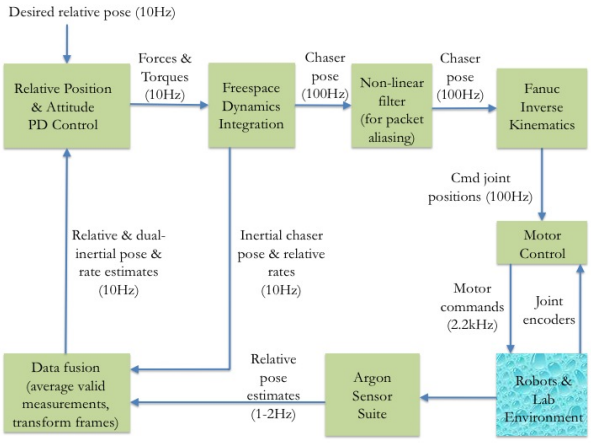

Figure 3. Block diagram of control loop.

4. SERVICER SPACECRAFT SIMULATOR

A Fanuc S-430iF 6-axis industrial manipulator was used for positioning Argon relative to the target satellite mockup. The unmodified robot has a reach of 2.643 m, payload capacity of 130 kg and repeatability of ± 0.3 mm. The stock controller has been replaced with a custom unit based on the Delta Tau PMAC architecture, and operates a joint position control loop on the servo card at

2.2 kHz. A second computer hosts the custom, soft-realtime, C-based, robot control software that normally generates trajectories, runs forward and inverse kinematics, software based compliance control and in this case, accepts and processes cartesian position commands from the Freespace computer. A third computer hosts a GUI and 3D visualization software for robot commanding and telemetry monitoring.

The robot control software sends joint position commands to the servo card over ethernet, triggered at 100 Hz (user selectable up to 500 Hz) via a synchronized clock pulse. It receives cartesian position updates from Freespace via an unsynchronized ethernet connection at approximately 100 Hz. Robot telemetry is also forwarded back to the Freespace computer via ethernet at 100 Hz allowing fast correlation between robot, Freespace and Argon data at test time. Since the communications path between Freespace and the robot control computer is unsynchronized, packets may show up one or more cycles late. To ensure that a smooth trajectory is sent to the servo card, a non-linear tracking differentiator (Jian-liang et al. 2009) is applied to the incoming pose vector. The filter was originally intended for use in joint-space to reduce chatter, but works well in Cartesian space provided Euler singularities are handled or avoided. For each new position command, the desired servicer pose is transformed to robot base frame by the robot control software. Inverse kinematics and some safety checks are performed and the resulting joint position vector is forwarded to the servo card.

5. TARGET SATELLITE SIMULATOR

A mock-up of the aft end of a GOES-12 satellite (FSAB) was mounted to a Rotopod R-2000 parallel robotic platform manufactured by Mikrolar, Inc. The R-2000 has six struts of fixed length 0.394 m that are attached via ball joints between the top plate (diameter 0.84 m) and "trucks" moving around a circular rail of radius 0.514 m at the base, allowing 360° of rotation of the top plate. Full six-axis motion of the top platform is achieved by moving the trucks around the track with azimuth specified by inverse kinematics.

The coordinate frames used to determine the rotopod kinematics are shown in Fig. 4: $\{G\}$ is located at the CG of the satellite; $\{F\}$ is attached to the FSAB; $\{T\}$ is attached to the top plate of the Rotopod; and $\{0\}$ is the base frame of the Rotopod. The 4x4 T transform matrix (Craig 1989) that gives the forward kinematics of the top plate of the Rotopod is

$$ {}^{0}_{T}T = {}^{0}_{G}T\,{}^{G}_{F}T\,{}^{F}_{T}T \tag{1} $$

The pose of the FSAB mockup with respect to the CG frame of the satellite ${}^{G}_{F}T$ is found using the ZYZ Euler model depicted in Fig. 5. The satellite CG is located at the origin of frame 0, and frame 3 is attached to the body

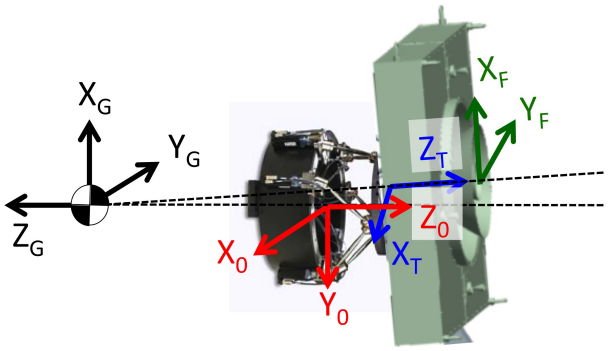

Figure 4. Coordinate frames used for Rotopod.

of the satellite. The resultant satellite pose is given by

$$ {}^{G}\boldsymbol{r}^{F}_{G} = \begin{bmatrix} d_3 c_1 s_2 \\ d_3 s_1 s_2 \\ d_3 c_2 \end{bmatrix} \tag{2} $$

$$ {}^{G}_{F}R = \begin{bmatrix} c_1c_2c_3 - s_1s_3 & -c_1c_2s_3 - s_1c_3 & c_1s_2 \\ s_1c_2c_3 + c_1s_3 & -s_1c_2s_3 + c_1c_3 & s_1s_2 \\ -s_2c_3 & s_2s_3 & c_2 \end{bmatrix} \tag{3} $$

where frame 0 is $\{G\}$, frame 3 is $\{F\}$, $s_i \equiv sin\theta_i$ and $c_i \equiv cos\theta_i$. The Cartesian axis rotational rates for a constant precession angle θ_2 are related to the satellite spin and precession rates via

$$ {}^{0}\boldsymbol{\omega}^{3}_{0} = \begin{bmatrix} c_1 s_2 \dot{\theta}_3 \\ s_1 s_2 \dot{\theta}_3 \\ \dot{\theta}_1 + c_2 \dot{\theta}_3 \end{bmatrix} \tag{4} $$

The desired precession angle θ_2, precession rate $\dot{\theta}_1$, and spin rate $\dot{\theta}_3$ were input to (4) to determine the FSAB pose ${}^{G}_{F}T$ and then (1) was used to obtain the Rotopod pose ${}^{0}_{T}T$. The desired trajectory is then converted into a Cartesian position and roll-pitch-yaw command script, which is executed by the Rotopod Parasol6 controller at 100 Hz.

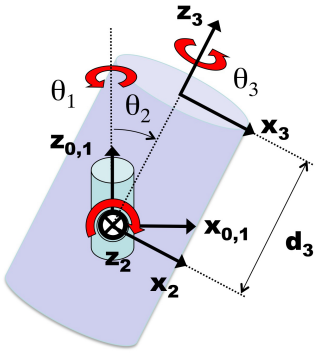

Figure 5. ZYZ Euler model used to model satellite.

6. MISSION SIMULATION USING FREESPACE

Freespace is a multi-body dynamics simulation platform developed at NASA/GSFC. It is based upon momentum-space dynamics (Hughes 1986) and allows the user to

select from pre-existing or user-defined C-code modules (gravity, sensors, actuators, flight software, etc.) to comprise a simulation. Simulation parameters are setup using scripts that conform to a Matlab-like interface. Integration uses the Dormand-Prince 853 method and soft real-time simulation is achieved by using the system clock available on the real-time Linux kernel. An OpenGL-based 3D viewer allows the user to monitor the simulation in real-time.

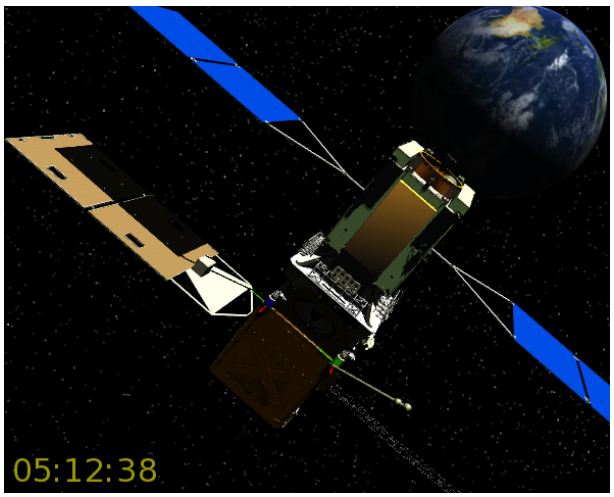

Figure 6. The Freespace simulation viewer utility, showing a geo-satellite servicer spacecraft at a 2 m separation from the GOES12 weather satellite target

For the hardware-in-the-loop tests, the simulator model included Moon and Sun gravity, and Earth gravity harmonics up to 10th order. Solar pressure is a significant low-frequency torque disturbance in geosynchronous orbit, but was neglected for these short tests. The target is modeled as a free-drifting body and the servicer spacecraft is under translation and attitude control as specified below.

The tests presented in this paper were intended to demonstrate stable system performance in the final phases of autonomous rendezvous. In the final approach, the mission design has the servicer move down a capture axis, pre-defined in the body frame of the target, in a series of hops between hold points. Tests were conducted around the 12 m and 2 m hold points by physically relocating the Rotopod (target motion simulator) in the lab.

In a robust system, the sensor measurements would be filtered by an observer to provide smooth state estimates with minimal latency - via use of a model to predict system behavior. For this test campaign, in the interest of simplicity, relative state estimates (${}^s\boldsymbol{r}_s^F$, ${}^s_F\boldsymbol{q}$) were generated from a simple average of any valid relative-pose measurement channels. These state estimates are converted to inertial values, using the inertial state of the servicer. Since the filter did not produce rate estimates from the pose measurements, truth inertial rates were used in the guidance and control algorithms. Future tests will include artificial corruption of inertial state estimates as well as a Kalman filter for estimation of relative rates.

During this final approach phase, the desired servicer attitude was to point the center of the servicer sensor face at the origin of the approach axis (pre-specified point on the target capture face) and to simultaneously align the solar panel axis of the servicer to match the solar panel axis of the target. The first objective helps minimize motion of the target in the sensor images and the second objective sets a pre-specified relative orientation of servicer and target to maintain power positive condition during the orbit. To find the desired pointing axes, the docking axis aim point must be found with respect to the servicer and transformed to the inertial frame. The secondary alignment axis is found from a simple frame transform from the target frame to inertial.

$$\begin{aligned} {}^i\boldsymbol{r}_s^{DA} &= {}^i_sR{}^s\boldsymbol{r}_s^F + {}^i_F\,R^F\boldsymbol{r}_F^{DA} \\ {}^i\hat{\boldsymbol{r}}_1 &= {}^i\boldsymbol{r}_s^{DA}/\|{}^i\boldsymbol{r}_s^{DA}\| \\ {}^i\hat{\boldsymbol{r}}_2 &= {}^i\hat{\boldsymbol{r}}_a = {}^i_FR^F\hat{\boldsymbol{r}}_a \end{aligned} \tag{5}$$

The primary and secondary pointing axes are also specified in the servicer frame, hence the *qfast* function (Markley 2002) can be used to return a quaternion that defines the servicer frame with respect to the the inertial frame.

$${}^d_i\boldsymbol{q} = \text{qfast}\left({}^i\hat{\boldsymbol{r}}_1, {}^i\hat{\boldsymbol{r}}_2, {}^s\hat{\boldsymbol{r}}_1, {}^s\hat{\boldsymbol{r}}_2\right) \tag{6}$$

Attitude control is achieved with simple PD feedback of quaternion attitude and rate errors in the servicer frame.

$${}^s_d\boldsymbol{q} = {}^s_i\boldsymbol{q} \otimes \left({}^d_i\boldsymbol{q}\right)^{-1} \tag{7}$$

$$\tau = K_p\,{}^s_d\boldsymbol{q}\,\text{sgn}\left({}^s_dq_4\right) + K_d\left({}^s\boldsymbol{w}_i^s - {}^s\,\boldsymbol{w}_i^d\right) \tag{8}$$

Translational tracking of a capture axis can be done with simple PD control, but greater accuracy (and reduced burn frequency) can be achieved by accounting for axes coupling due to the orbit motion using the well-known Clohessy-Wiltshire (CW) equations, which describe the relative motion between two bodies in a similar circular orbit. To take advantage of these equations, the desired and current positions of the servicer center of mass are first converted to RIC-frame quantities.

$$\begin{aligned} {}^{RIC}\boldsymbol{r}_G^{dcm} &= {}^i_F\,R\left({}^F\boldsymbol{r}_F^{DA} + {}^F\,\boldsymbol{r}_{DA}^d - {}^F\,\boldsymbol{r}_F^G\right) + {}^i_s\,R^s\boldsymbol{r}_s^{scm} \\ {}^{RIC}\boldsymbol{r}_G^{scm} &= {}^{RIC}_i\,R\left({}^i_sR\left({}^s\boldsymbol{r}_s^{scm} - {}^s\,\boldsymbol{r}_s^F\right) - {}^i_F\,R^F\boldsymbol{r}_F^G\right) \end{aligned}$$

The desired velocity, ${}^{RIC}v_G^{dcm}$, to transfer the servicer from ${}^{RIC}\boldsymbol{r}_G^{scm}$ to ${}^{RIC}\boldsymbol{r}_G^{dcm}$ in a given amount of time is then computed from CW equations.

The translational control loop is closed by regular application of velocity impulses, ΔV:

$$\Delta V = {}^{RIC}\,\boldsymbol{v}_G^{dcm} - {}^{RIC}\,\boldsymbol{v}_G^{scm}, \tag{9}$$

which force the servicer to follow a specific velocity profile in the RIC frame as computed from the CW guidance.

7. CLIENT PACKAGE (ARGON)

Originally formulated as a payload to the International Space Station (ISS) (Naasz et al. 2011), the Argon system, shown in Fig. 7, includes all of the major component elements that would be part of rendezvous and proximity operation subsystem on a servicing spacecraft. The Argon sensors include 1 mega-pixel optical cameras built by MacDonald, Dettwiler and Associates Ltd., flown previously on the Hubble Space Telescope (HST) Servicing Mission 4 Relative Navigation Sensor (RNS) experiment, where they successfully captured images used to compute a real-time, on-board pose solution to HST (Naasz et al. 2010). Argon also includes a flash-based LIDAR: the Vision Navigation System (VNS) built by Ball Aerospace Technologies Corp., designed and built to be the primary relative navigation sensor for the Orion Multi-Purpose Crew Vehicle (MPCV). Instead of scanning the scene with a single beam, the VNS flashes the scene once and collects multiple returns via a 256x256 pixelated detector.

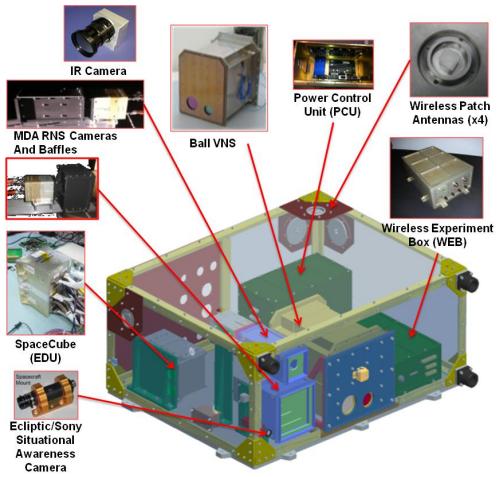

Figure 7. Argon internal components.

GNFIR **-** Argon utilizes the GNFIR (Goddard Natural Feature Image Recognition) algorithm to return its primary pose measurement (Naasz et al. 2010). Visual features are matched to a model (formulated as a set of edges) as shown in Fig. 8.

Goddard FlashPose **-** Argon also hosts the GSFC FlashPose algorithm, which processes real-time flash LIDAR frames to produce a 6-DOF pose estimate. The algorithm processes range images as a point cloud, subsamples the cloud based on certain quality metrics, and uses a custom Iterative Closest Point (ICP) algorithm to determine an optimal estimate of the relative position and attitude.

8. HARDWARE-IN-THE-LOOP TEST RESULTS

Table 1 shows the tracking accuracy for several different tests at the Freespace to Fanuc interface, for the robot itself, and for the combined motion platform system. The closed-loop performance of the Argon sensor package is also provided. Despite several large error sources in the motion simulation system (see Section 9) the servicer

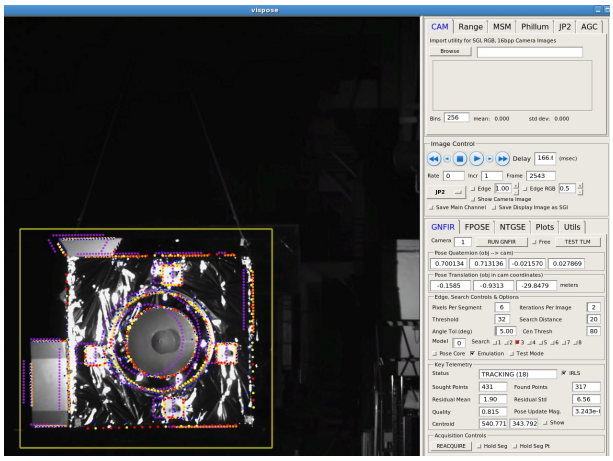

Figure 8. GNFIR algorithm tracking GOES-12 mockup during testing.

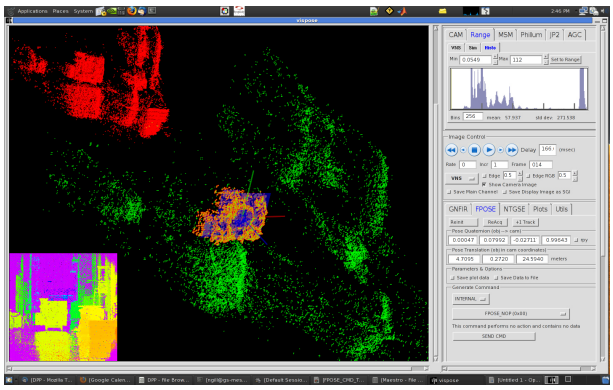

Figure 9. Flashpose algorithm traclomg GOES-12 during Argon testing

should be able to correct for these as it is closing the loop around live measurements from the lab scene. Indeed, Table 1 indicates worse tracking performance for the tests using Freespace-truth data for feedback as opposed to Argon data. Lab misalignments between the two robots present as fixed offsets to the initial condition and tool transform knowledge errors provide a pose-dependent disturbance during the test. Argon's performance was evaluated by comparing the actual pose from the Leica data to the desired pose over the second half of the test - allowing the servicer spacecraft time to recover from initial perturbations. The desired position is a function of time (aim for a specified point on capture axis), and desired attitude is a function of position (to keep the sensor face pointed at the target). For cases where the target was static or spinning about the capture axis, the servicer was able to track to within the accuracy of Argon's visual-based pose estimation technique. Attitude bias is less than a degree and translational bias about lateral axes is on the order of a centimeter. Range error is larger (up to 20 cm), due to errors in assumed camera parameters and subtle mismatches between the CAD model and the target. In addition, time-varying bias occurs due to the tracking of false edges in the camera image. Standard deviation of pose estimate noise is around 0.1 cm and 0.1°. For cases where the capture axis was moving due to tar-

Table 1. RMS tracking errors for several tests

Test No.	10	11	12	13	91	93	94	96	98
argon (A) or fsp truth (T) feedback	T	T	A	A	A	A	A	A	A
average range (m)	2.5	2.5	2.5	2.5	11.5	11.5	11.5	11.5	11.5
initial translation offset (m)						$y = 1$	$y = 1$		$y = 1$
initial attitude offset (deg)						$\theta_x = 5.7$	$\theta_x = 5.7$		
tgt. cap. axis spin (deg/s)					-2.2	-0.1	-2.2		-2.2
tgt. cap. axis precession (deg/s)					2.3		2.3		2.3
RMS errors (cm and deg)									
pos. fsp. cmd. vs fanuc cmd.	0.05	0.03	0.04	0.03	0.04	0.03	0.03	0.05	0.03
pos. fanuc cmd. vs fanuc meas.	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.01	0.01
pos. fanuc meas. vs leica meas.	6.34	4.61	6.03	5.99	10.2	10.7	14.7	0.83	7.19
att. fsp. cmd. vs fanuc cmd.	0.04	0.01	0.01	0.01	0.05	0.01	0.01	0.04	0.01
att. fanuc cmd. vs fanuc meas.	0.01	0.01	0.02	0.03	0.01	0.03	0.03	0.02	0.01
att. fanuc meas. vs leica meas.	0.55	1.17	1.11	1.03	1.25	1.04	1.66	0.20	1.02
pos. argon lateral tracking (leica meas.)	5.66	5.30	1.26	2.99	27.4	12.3	33.6	7.52	60.50
att. argon tracking (leica meas.)	3.99	2.75	3.04	2.25	1.23	1.32	1.92	0.44	2.59

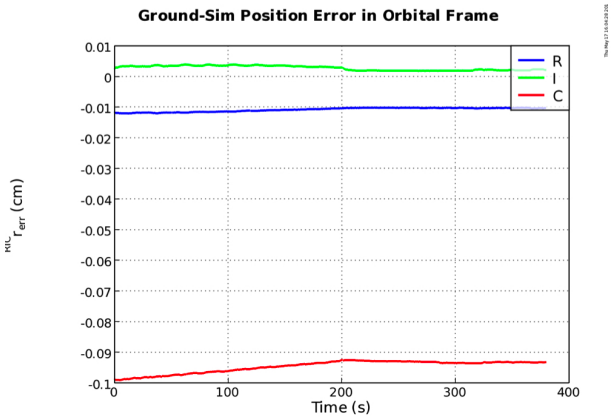

Figure 10. Relative position measurement error for test 13, range 2.5 m; encoder-derived relative measurement vs Leica relative measurement. Large inaccuracy in cross-track direction could be reduced by iterating Fanuc or Rotopod nominal pose, taking cues from the Leica.

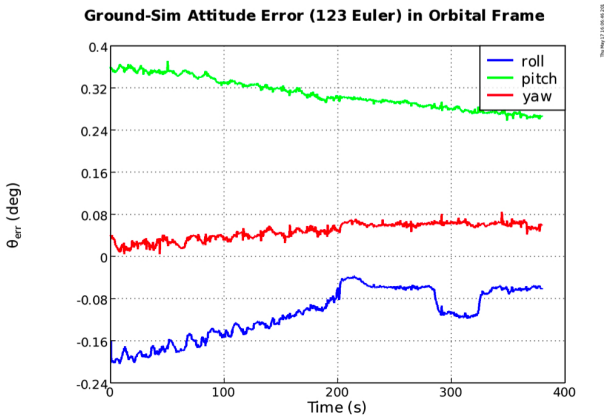

Figure 11. Relative attitude measurement error, test 13, range 2.5 m; encoder-derived relative measurement vs Leica relative measurement.

get precession, the translational control gains were too low to track the relatively fast wobble (period of 160 sec) and therefore position tracking errors were significantly higher. In future test campaigns, an Extended Kalman Filter will be used to improve state estimate accuracy and reduce latencies thus allowing for higher control gains. The filter will also provide rate estimates, eliminating the need to use Freespace-truth rates in the control loop.

9. DISCUSSION

The ability of the motion platforms to reproduce the motion inside the space simulation is affected by several error sources:

1. Misalignment and position error between Fanuc and Rotopod.
2. Errors in knowledge of transforms from end effector to tools (outer face of satellites).
3. Mismatch between target motion inside Freespace and Rotopod motion script.
4. Fanuc tracking of Freespace commands
   (a) Tracking latency / transients
   (b) Absolute encoder error / joint slop / flexibility
5. Rotopod tracking of script commands (or Freespace commands in future setup)
   (a) Tracking latency / transients
   (b) Absolute encoder error / joint slop / flexibility

Error source 1 is significant, as relative positioning of the robots is achieved by manually moving the Rotopod on a hover platform and correcting for some amount of error

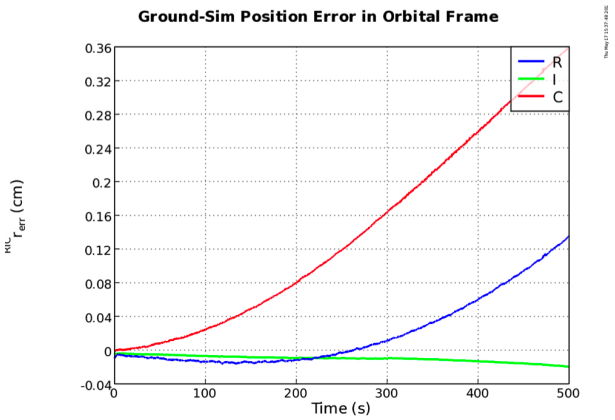

Figure 12. Relative position measurement error, test 93, range 11.5 m and spinning target; encoder-derived relative measurement vs Leica relative measurement. Divergence is due to mismatch between Rotopod motion script and true target dynamics inside Freespace.

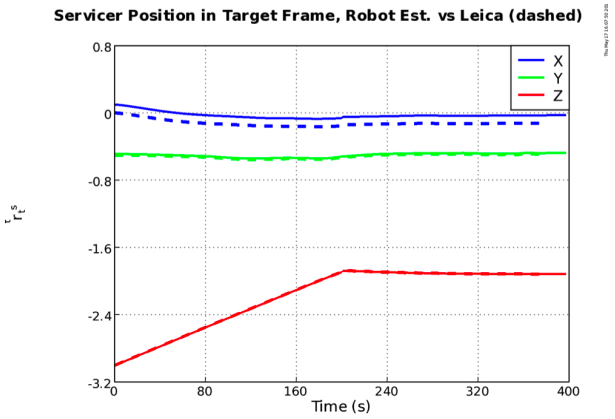

Figure 13. Argon position in target frame, test 13; Leica relative measurement. Desired y-offset is -0.5 m.

with the Fanuc nominal pose. In the Argon test campaign there was not sufficient time to iterate the Fanuc pose with verification from the Leica unit. Leica measurements indicated that the tests were run with per axis errors of 0.5-7 cm and 0.1-1°. Iteration of the Rotopod or Fanuc nominal pose could reduce these to ∼0.5 cm and ∼0.1°.

Error source 2 could be eliminated by measurement but this was not performed for the Argon campaign. Instead a combination of hand measurements and CAD models were used. Associated errors are estimated to be ∼1 cm and ∼0.5°.

Error source 3 can be evaluated by comparison of Freespace and Rotopod motion logs. Aside from a negligible error in the different start times for the motion, there is also a slow divergence due to the orbital pitch rate of the target satellite coupling into the spin dynamics. The Rotopod script derivation assumed that the initial spin and precession motion relative to the orbital frame was decoupled from the orbital spin rate. The consequence is a divergence of about 0.25°/min, as the target satellite becomes inertially spin stabilized when biased with high spin rates about one axis. This is significant for all the

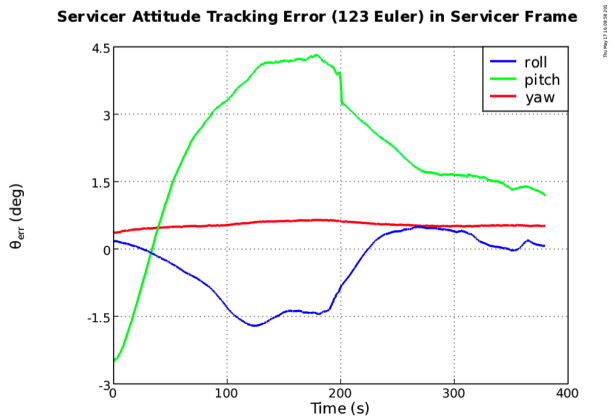

Figure 14. Argon capture attitude tracking error, test 13; Leica relative measurement.

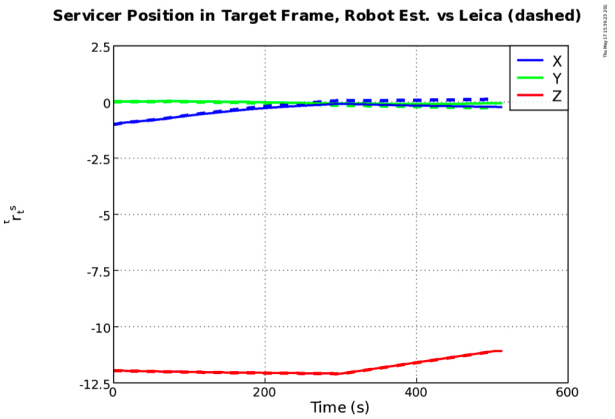

Figure 15. Argon position in target frame, test 93, spinning target; Leica relative measurement.

tests where the target is spinning (see Fig. 12, where a 2° error developed over 500 sec creates a capture axis lateral offset of 40 cm at 11 m range). Error 3 will be irrelevant in future campaigns where the Rotopod will take commands directly from Freespace.

The combination of errors 4b, 1 and 2 can be evaluated by comparing the robot-estimated pose with the Leica-measured relative pose, see example Figs. 10-11. The RMS errors in this table tend to capture slow varying biases or diverging bias (as mentioned above) rather than high frequency noise. Examining the statistics from several different tests, RMS errors of 0.8-14.7 cm and 0.2-1.7° were observed (see Table 1). Error 1 should be constant amongst tests where the Rotopod base was in the same position, with time-varying, pose-dependent errors attributable to 4b and 2. The low RMS error in test 96 is largely due to finer alignment practices in re-locating the Rotopod base compared to early tests. Tests 91-94 and 98 additionally include error sources 5b and 3, the latter being the prime cause of the larger RMS errors compared with the static Rotopod tests. Note that time synchronization of data sources for this comparison relies upon two users pressing a button simultaneously with an audio cue. Due to the relatively slow nature of the motion, this is not a significant issue.

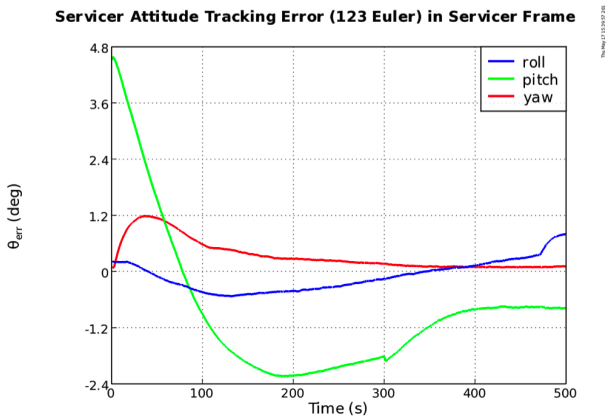

Figure 16. Argon capture attitude tracking error, test 93, spinning target; Leica relative measurement.

Error source 4a is small (RMS error $<$ 0.6 mm and 0.06°), both for noisy Freespace commands (when the servicer uses Argon measurement feedback) and smooth Freespace commands (using truth measurements). In the error plots, the sharp jump halfway through the test coincides with the change in desired motion and servicer translational gain from hold on capture axis to move along capture axis. According to manufacturer specifications, the absolute accuracy of the Rotopod and Fanuc are sub-mm, thus the robot contribution to the tracking error (errors 4 and 5) is quite small.

Figs. 10-11 and 13 show the roughly constant error due to lab misalignments of the two robots. Fig. 12 and Fig. 15 show divergence of error, due to the aforementioned problem associated with error source 3. Fig. 13 shows the servicer attempting to move from 3 m to 2 m and hold (with a laterally offset capture axis, -0.5 m in y). Tracking of attitude (Fig. 14) is less accurate at this short range due to the coupling between desired attitude and position offset from capture axis. Therefore, a decoupled attitude control scheme may be advisable for these latter stages of approach. The performance plots show stable behavior and recovery from an initial offset of 2.5° induced due to lab robot misalignments. Similar stable behavior is shown in Figs. 15-16, when the target is spinning at 0.1°/s about the capture axis and the servicer has an initial offset from the capture axis of 1 m. The goal of this test was to hold at 11.95 m for 300 sec, whilst matching the target's attitude and spin rate, then move forward along the capture axis. Note that a large component of the lateral steady state bias at this longer range is due to a bug in the entry of a frame transform that biased measurements by $\sim$1° in attitude and $\sim$20 cm in lateral position.

10. CONCLUSIONS

The SSCO ground simulator was found to be an effective platform for testing the Argon relative navigation system. The Fanuc manipulator was able to track the servicer position commands output from Freespace, while the Rotopod was able to produce a variety of spin/precession rates for the satellite mockup. RMS tracking errors in the ground hardware were 0.8-14.7 cm and 0.2-1.7°, largely due to script mismatch between Freespace and the Rotopod, as well as initial misalignments between the Fanuc and Rotopod. Important future improvements to the system include using Freespace to generate the motion commands for the Rotopod and iterating on the home positions for the Rotopod and Fanuc to achieve near-zero misalignments.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Argon test team, GSFC Robotics team, and the GSFC Satellite Servicing Capabilities Office for supporting this research.

REFERENCES

Benninghoff, H., Tzschichholz, T., Boge, T., & Rupp, T. 2011, in Proc. of the 28th International Symposium on Space Technology and Science, Okinawa

Craig, J. 1989, Introduction to Robotics: Mechanics and Control, 2nd ed. (Reading, Mass.: Addison-Wesley)

Hughes, P. C. 1986, Spacecraft Attitude Dynamics (New York: Wiley)

Jian-liang, P., Xiu-xia, S., Wen-han, D., & Shu-guang, L. 2009, in Proc. of the IEEE International Conference on Intelligent Computing and Intelligent Systems, IEEE, Shanghai, 848–852

Markley, F. L. 2002, Journal of Guidance, Control and Dynamics, 25, 411

Naasz, B. J., Strube, M. J., Van Eepoel, J., Barbee, B. W., & Getzandanner, K. M. 2011, in Proceedings of the 34th AAS Guidance and Control Conference No. AAS 11-072, Breckenridge Colorado

Naasz, B. J., Van Eepoel, J., Queen, S. Z., II, C. M. S., & Hannah, J. 2010, in Proceedings of the 33rd AAS Guidance and Control Conference No. AAS 10-086, Breckenridge Colorado

Obermark, J., Creamer, G., Kelm, B. E., Wagner, W., & Henshaw, G. C. 2007, in Proceedings of the SPIE - Sensors and Systems for Space Applications, Vol. 6555

Xu, W., Liang, B., Xu, Y., Li, C., & Qiang, W. 2007, Journal of Intelligent and Robotic systems, 48, 187

Yoshida, K. 2003, The International Journal of Robotics Research, 22, 321