

On-Board Rendezvous Targeting for Orion

Michael W. Weeks* and Christopher N. D'Souza†

NASA Johnson Space Center, Houston, TX, 77058, USA

The Orion On-board GNC system is among the most complex ever developed for a space mission. It is designed to operate autonomously (independent of the ground). The rendezvous system in particular was designed to operate on the far side of the moon, and in the case of loss-of-communications with the ground. The vehicle GNC system is designed to retarget the rendezvous maneuvers, given a mission plan. As such, all the maneuvers which will be performed by Orion, have been designed and are being incorporated into the flight code. This paper will describe the rendezvous maneuvers and the architecture with which the on-board

Nomenclature

<i>NSR</i>	Nominal Slow Rate (Coelliptic Maneuver)
<i>NPC</i>	Nominal Plane Change
<i>NH</i>	Nominal Height Adjust (Altitude Raising Maneuver)
<i>NC</i>	Nominal Catchup (Phasing Maneuver)

I. Introduction

A number of vehicles have successfully performed automated rendezvous and docking with target vehicles. These include: Progress, Soyuz, ETS-VII, Orbital Express, HTV, and ATV. The Space Shuttle does perform limited on-board targeting.

With regard to on-board targeting, early in the Space Shuttle program, a decision was made to give the vehicle the capability to perform on-board targeting, including the early rendezvous maneuvers. However, due to computational throughput constraints and limitations, this far-field rendezvous targeting capability was deleted in favor of performing the far-field targeting on the ground.

Orion, because of the requirement to perform the rendezvous independent of the ground (either due to loss-of-communications with the ground or on the far side of the Moon). Algorithms and software have been designed and developed to perform the targeting functions, including the far-field rendezvous on-board the vehicle. This decision became all the more important because the relative navigation is performed on-board.

For this purpose, maneuver planning involves creating a plan which takes into account all effective constraints (lighting, communications) whereas targeting involves taking the maneuver plan and retargeting the specified maneuvers at the specified times based upon relative navigation states.

This paper describes the design of the algorithms and software to perform the Orion on-board targeting function. It is organized as followed. In the next section, the rendezvous targeting architecture will be described. Section 3 contains a description of the phasing maneuver (called the NC maneuver). Section 4 contains a description of the height adjust maneuver (called NH). Section 5 contains a description of the co-elliptic maneuver (called NSR). Section 6 contains the description of the plane change maneuver (called NPC). In Section 7, results of these algorithms are presented. Finally, Section 8 contains a few concluding remarks.

*Orion Guidance and Targeting Subsystem Manager

†Orion Orbit GNC Deputy System Manager

II. The Rendezvous Targeting Architecture

The Orion is required to perform a wide variety of rendezvous scenarios: LEO with ISS, LEO with the EDS/Altair and LLO with Altair. As such the targeting algorithms and software must be flexible and robust enough to retarget and refine the maneuvers in order to successfully complete the rendezvous. This functionality is performed by a CSU (Computer Software Unit) known as “*RTARG*”. *RTARG* is required to retarget not only the upcoming maneuver but, if necessary, all future maneuvers. This was designed and ‘coded’ into the Matlab/Simulink/Stateflow environment – it will be autocoded – and has been working in the FSW architecture designed for Orion. This is seen in Figure 1.

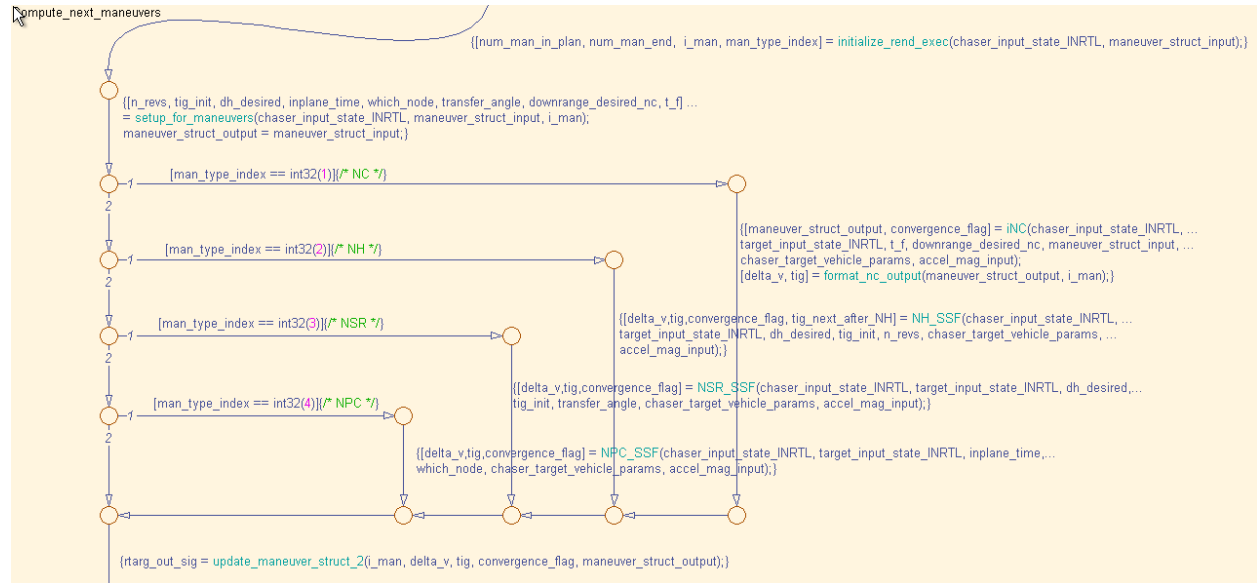


Figure 1. Top Level *RTARG* Architecture

This particular CSU is called from the GNC executive, and its inputs include the maneuver plan, the current Orion state and the current target state. In addition, parameters, such as the degree and order and coefficients of the gravity field, the drag model, etc are part of the inputs to this CSU.

The *RTARG* CSU calls the other maneuvers according to the prescribed input maneuver plan. The updated maneuver plan, with the times of ignition (TIGs) and ΔV s are sent out of *RTARG*.

A. The Trajectory Predictor/Integrator

The Orion vehicle will use a common integrator for all predictive functions, i.e. to compute the state of a vehicle at a future time, given the current state. A trade study was initiated to select the integrator and a modified fourth-order fixed step Encke-Nystrom integrator was selected. The Encke-Nystrom integrator is a second-order integrator, and the fourth-order version calls for three function evaluation per time step.¹ The major reason for the selection is that because it corrects the trajectory based upon a conic due to perturbative accelerations, it can operate to high accuracy while using long step sizes. Since the Orion vehicle was also required to operate in low earth orbit, where the drag forces cannot be ignored, a modification to the Encke-Nystrom was made which allows for a second-order formulation while still including the atmospheric effects.

B. The Fidelity of *RTARG* Dynamic Models

The models used in *RTARG* can be divided into environment models and actuator models

1. The Environment Models

The planetary gravity models assumes that the degree, the order and the coefficients of the gravity field are specified. In addition, the perturbations of the third bodies are included. These include the Sun, the Moon (while orbiting the Earth) and the Earth (while orbiting the Earth).

Drag is modeled as

$$\mathbf{a}_d = -\frac{1}{2}\rho|\mathbf{V}_{rel}|S_{ref}C_D\mathbf{V}_{rel} \quad (1)$$

where S_{ref} is the reference surface area, ρ is the atmospheric density, C_D is the co-efficient of drag, $\mathbf{V}_{rel} = \mathbf{V} - \mathbf{V}_{atm}$ where \mathbf{V} is the inertial vehicle velocity and \mathbf{V}_{atm} is the velocity of the atmosphere at altitude. The atmosphere is modeled after a Babb-Mueller approximation.²

2. The Actuator Model

RTARG has the capability of targeting the maneuvers accounting for the finite burn effects. As such, it models the maneuvers as taking a finite (as opposed to an infinitesimal) amount of time allowing the inefficiency of the maneuvers to be taken into account in targeting the maneuvers. It is assumed that the direction of the maneuver is fixed in inertial space. Hence, during a maneuver, a small step size is taken with the expected acceleration from the selected engine (or group of engines). The engine/thrust magnitude is an input to *RTARG*.

Tables 1 and 2 contain the input and output parameters to *RTARG*.

Input	Description
maneuver_struct_input	The “burn-plan” of all maneuvers to be targeted between TIG and t.f
r_chaser	Chaser Planet-centered Inertial Position
v_chaser	Chaser Planet-centered Inertial Velocity
t_chaser	Time of chaser state
r_target	Target Planet-centered Inertial Position
v_target	Target Planet-centered Inertial Velocity
t_target	Time of target state
accel	Thrust acceleration for specified effector

Table 1. *RTARG* Input Parameters

Input	Description
maneuver_struct_output	The “burn-plan” of all maneuvers to be targeted between TIG and t.f and the associated delta-V for each burn

Table 2. *RTARG* Input Parameters

III. The Altitude Raise Maneuver – NH

This maneuver is fairly straightforward. It’s purpose is to increase the energy so as to achieve a particular delta-altitude relative to the target. Since it is a purely energy raising (or lowering if appropriate) maneuver, the direction is parallel to the velocity vector. The algorithm provides flexibility to achieve the desired change in altitude in a specified transfer angle (specified in degrees). This allows, say, a 135 degree transfer allowing a node to be created, if required. This maneuver does not need to be performed at an apse. It

is designed to be initiated at a specified time and will achieve the desired altitude at the specified transfer angle.

Whereas, because of the non-sphericity of the gravity fields involved, not to mention the non-conservative nature of drag, iterations will be required, experience garnered over thousands of runs indicate that it converges within 3 iterations. It does not require an initial guess as an input, generating it's own guess.

This algorithm is sometimes used in conjunction with the NSR maneuver to perform a coelliptic approach to a target or with the phasing maneuver (NC) to target for a particular altitude and downrange, both of which are energy raising. Figure 2 depicts a Stateflow design of this maneuver.

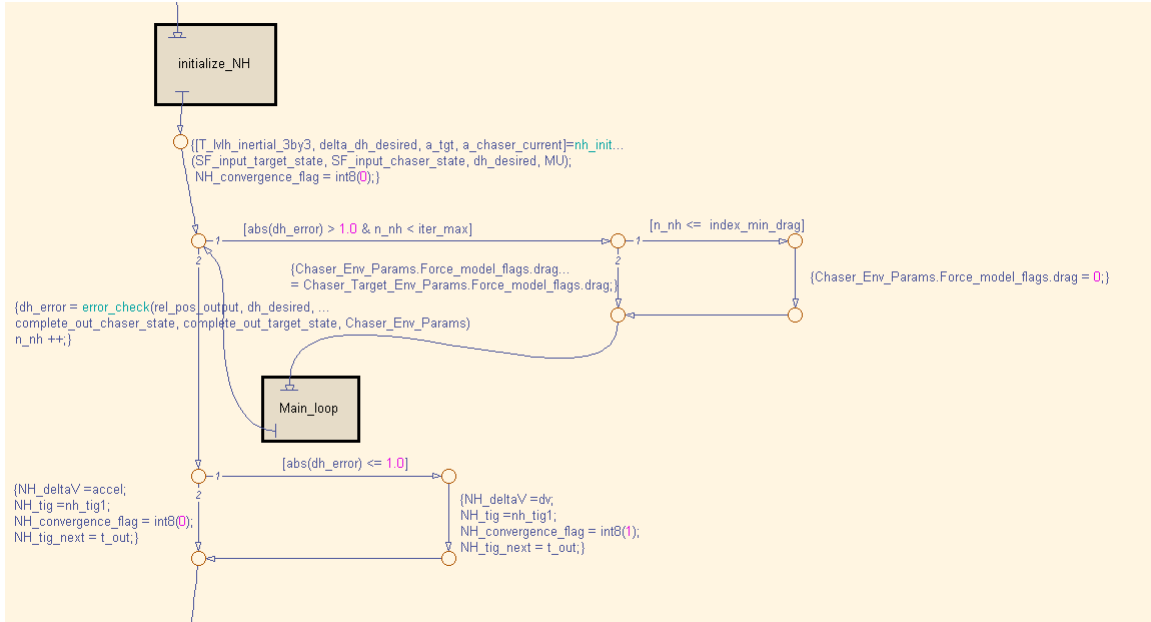


Figure 2. NH Stateflow Architecture

Tables 3 and 4 contain the input and output descriptions of the NH maneuver design.

Input	Description
tig	“Time of Ignition ” for <i>NH</i> maneuver (sec)
dh_desired	Desired delta-height after maneuver
n_revs	Desired number of revs from TIG when dh_desired will be achieved (integer or non-integer)
r_chaser	Chaser Planet-centered Inertial Position
v_chaser	Chaser Planet-centered Inertial Velocity
t_chaser	Time of chaser state
r_target	Target Planet-centered Inertial Position
v_target	Target Planet-centered Inertial Velocity
t_target	Time of target state
accel	Thrust acceleration for specified effector

Table 3. NH Input Parameters

Input	Description
tig	“Time of Ignition ” for <i>NH</i> maneuver (sec)
dv	<i>NH</i> Planet-centered Inertial Delta-V Vector
status_flag	Success flag; Scheduling Failure

Table 4. *NH* Output Parameters

IV. The Coelliptic Maneuver – NSR

The NSR maneuver seeks to establish a chaser orbit which is coelliptic with that of the target. Mathematically, that is described as:

$$a_C e_C = a_T e_T \quad (2)$$

where a and e denote the semi-major axis and eccentricity, respectively, of the orbit and the subscripts C and T refer to the chaser and target vehicles, respectively. Simply put, two vehicles have orbits which are coelliptic if they have a constant delta-height through out the orbit. Or, put a third way, the arguments of periapses are coincident. This targeting algorithm seeks to coellipticize the chaser orbit with respect to the target orbit. Therefore, unless the maneuver occurs at an apsis, there will be a radial component to the maneuver. In general, radial maneuvers are not efficient; hence if one varies the times of earlier maneuvers, one can adjust the argument of periapsis of the chaser orbit, thereby minimizing the radial component of the NSR maneuver. The maneuver design places no restriction on where in the orbit it must occur. It merely coellipticizes the orbit at the delta altitude at which the maneuver occurs.

As expected, it involves iterations, but experience to date over thousands of runs have uncovered no convergence issues. In addition, it usually converges in less than three iterations. It does not require an initial guess as an input, generating it's own guess. Figure 3 depicts a Stateflow design of this maneuver. Tables 5 and 6 contain the input and output descriptions of the NSR maneuver design.

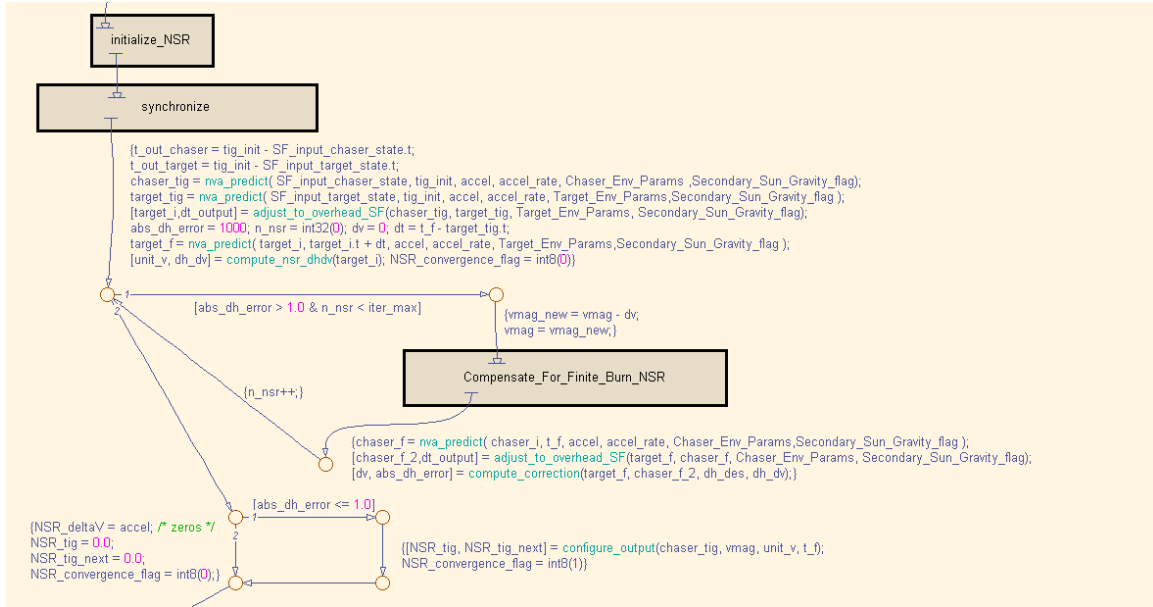


Figure 3. *NSR* Stateflow Architecture

Input	Description
tig	“Time of Ignition ” for <i>NSR</i> maneuver (sec)
dh_desired	Desired delta-height after maneuver
transfer_angle	The transfer angle from TIG to the point at which the coelliptic orbit and dh_desired are achieved
r_chaser	Chaser Planet-centered Inertial Position
v_chaser	Chaser Planet-centered Inertial Velocity
t_chaser	Time of chaser state
r_target	Target Planet-centered Inertial Position
v_target	Target Planet-centered Inertial Velocity
t_target	Time of target state
accel	Thrust acceleration for specified effector

Table 5. *NSR* Input Parameters

Input	Description
tig	“Time of Ignition ” for <i>NSR</i> maneuver (sec)
dv	<i>NSR</i> Planet-centered Inertial Delta-V Vector
status_flag	Success flag; Scheduling Failure

Table 6. *NSR* Input Parameters

V. The Plane Change Maneuver – NPC

The plane change (NPC) targeting algorithm seeks to eliminate cross-track error with respect to the specified target at a specified future time, the in-plane time, by performing an out-of-plane burn. The algorithm schedules the burn at a nodal crossinga place where the vehicle and target orbit planes cross. Non-spherical gravity causes the orbit planes to precess. The algorithm accounts for differing precession rates due to different orbit characteristics by direct propagation through a high-order gravity field. Because the NPC occurs at a nodal crossing, the rendezvous maneuver schedule must allow half an orbital period for the NPC maneuver plus time for burn solution validation and orienting the vehicle to burn attitude. To account for orbital precession, the algorithm propagates the vehicle and target forward to the specified in-plane time and places a phantom vehicle in the targets orbit plane by removing the cross-track position and velocity error from the vehicles state while preserving its velocity magnitude. When the algorithm then performs a plane-change maneuver into the phantom vehi-cles plane, the vehicle will remove its cross-track error with the target at the in-plane time. This numerical solution is much simpler than analytical calculations of precession.

The NPC algorithm contains two parts: a part that finds the plane change delta-velocity to put the vehicle in-plane with the target at the next node crossing, and a part that accounts for precession by propagating the vehicle and target forward to the specified in-plane time. The first portion finds the nodal crossing and subtracts the vehicles velocity from the desired in-plane ve-locity to find the impulsive delta-velocity vector. The second portion propagates the vehicle and target to the in-plane time and creates a phantom vehicle state by moving the vehicles state in plane with the target. Figure 4 depicts a Stateflow design of this maneuver.

Tables 7 and 8 contain the input and output descriptions of the NPC maneuver design.

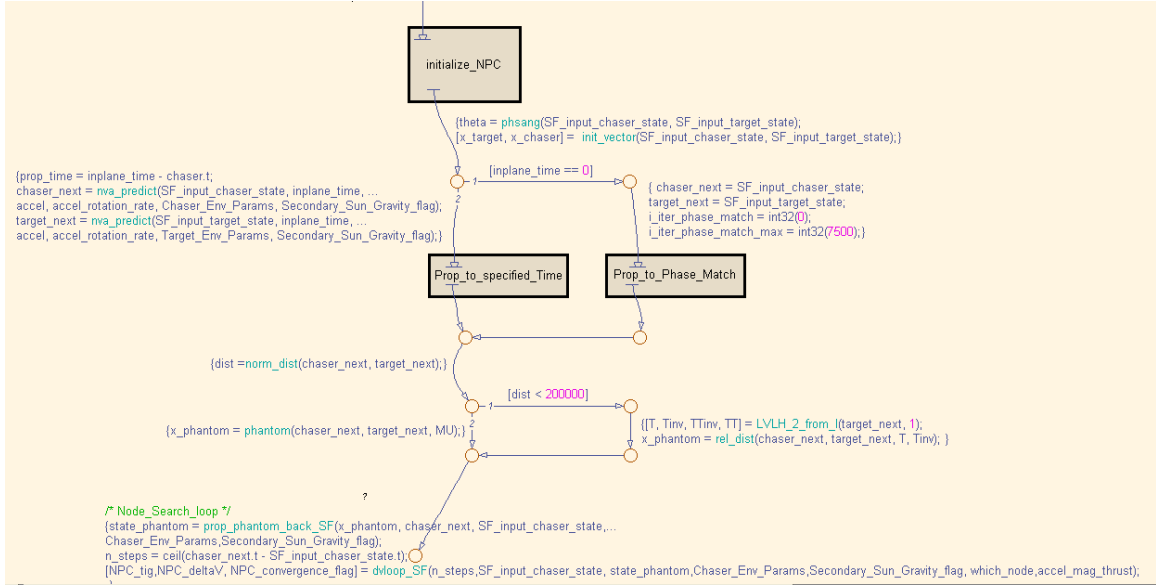


Figure 4. NPC Stateflow Architecture

Input	Description
dt_setup	Time from t_chaser to earliest allowable TIG
dt_inplane	Time from t_chaser to desired in-plane time
r_chaser	Chaser Planet-centered Inertial Position
v_chaser	Chaser Planet-centered Inertial Velocity
t_chaser	Time of chaser state
r_target	Target Planet-centered Inertial Position
v_target	Target Planet-centered Inertial Velocity
t_target	Time of target state
accel	Thrust acceleration for specified effector

Table 7. NPC Input Parameters

Input	Description
tig	“Time of Ignition ” for NPC maneuver (sec)
dv	NH Planet-centered Inertial Delta-V Vector
status_flag	Success flag; Scheduling Failure

Table 8. NPC Output Parameters

VI. The Phasing Maneuver – NC

This maneuver is among the most complex maneuvers in RTARG. In fact, it is the most complex algorithm in the Orion flight software. The reason for it is simple. It is designed to use all the prior algorithms. It is designed to achieve a specified downrange position relative to the target at a specified time and this time and downrange position are far enough in the future to include future maneuvers. For example, the insertion maneuver can be an NC maneuver and this maneuver could be directed to a point a few hundred meters away from the target several orbits later, which would have several maneuvers between the insertion maneuver and the final time. Thus, the algorithm has to account for all future maneuvers to the final time.

This maneuver is designed to be an energy raising maneuver. Hence it is optimal.

If the future maneuvers include additional phasing maneuvers, the algorithm expects a nominal DV to be specified along with the TIG of said maneuver. The other maneuvers are the ones previously described and are called accordingly. One can easily see why this maneuver is so complex.

If the maneuver is targeted such that it includes other maneuvers to the terminal time, the TIGs and the types and the objectives of the future maneuvers need to be specified. Figures 5 and ?? depict the Stateflow design of this maneuver. . .

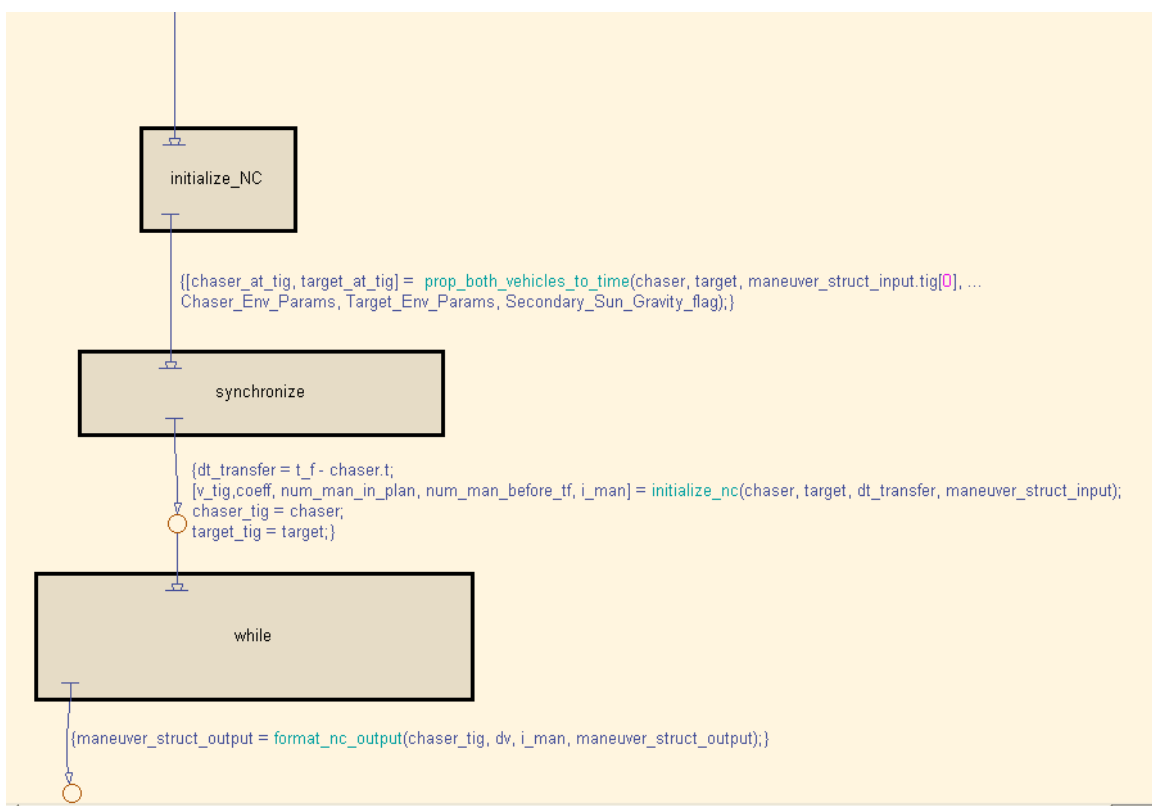


Figure 5. Top Level NC Stateflow Architecture

Tables 9 and 10 contain the input and output descriptions of the NH maneuver design.

VII. Results

Each of these algorithms has been subjected to rigorous testing prior to incorporation into the FSW process. To wit, they were tested well past $3 - \sigma$ conditions to $10 - \sigma$ conditions. The most stressful cases were the so-called Flight-Day 3 rendezvous. The nominal trajectory for the Flight Day 3 rendezvous is presented in Figure 7. We can see that this flight profile exercises all the rendezvous algorithms.

To further demonstrate the performance and robustness of the algorithms, the expected injection dispersions, which were significantly larger than the Shuttle dispersions, were selectively sampled, so that only

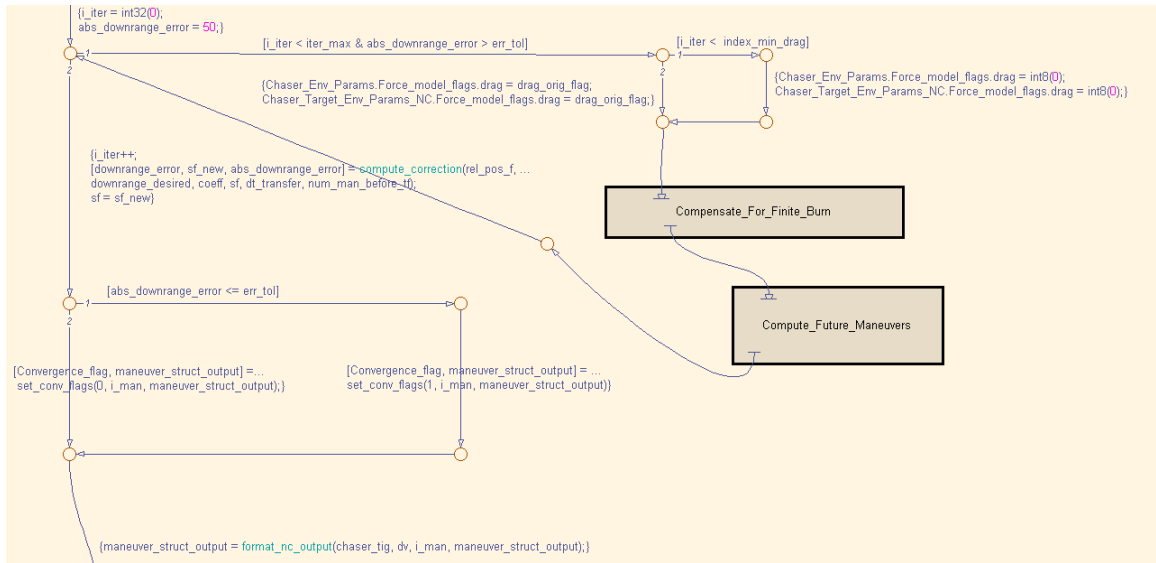


Figure 6. More Detailed NC Stateflow Architecture

Input	Description
tig	“Time of Ignition ” for <i>NC</i> maneuver (sec)
downrange_desired	Desired downrange distance at the specified t_f time relative to the target spacecraft
t_f	The final time at which it is desired to have the specified downrange desired distance
maneuver_struct_input	The “burn-plan” of all maneuvers to be targeted between TIG and t_f
r_chaser	Chaser Planet-centered Inertial Position
v_chaser	Chaser Planet-centered Inertial Velocity
t_chaser	Time of chaser state
r_target	Target Planet-centered Inertial Position
v_target	Target Planet-centered Inertial Velocity
t_target	Time of target state
accel	Thrust acceleration for specified effector

Table 9. NC Input Parameters

Input	Description
tig	“Time of Ignition ” for <i>NC</i> maneuver (sec)
dv	<i>NC</i> Planet-centered Inertial Delta-V Vector
status_flag	Success flag; Scheduling Failure
maneuver_struct_output	The “burn-plan” of all maneuvers to be targeted between TIG and t_f and the associated delta-V for each burn

Table 10. NC Input Parameters

samples within the desired σ limits were generated. These dispersions were then used to perturb the injection state and then this was used to target the NC maneuver to the TPI point (2 days later) for downrange (and altitude). The results for the $3\text{-}4\sigma$ cases (for 30 samples) are provided in Figure 8. It should be pointed out that these trajectories have no navigation errors; they are merely the result of the targeting and the resulting trajectories were flown out to the TPI point. But they demonstrate the ability of the RTARG algorithms to handle dispersions.

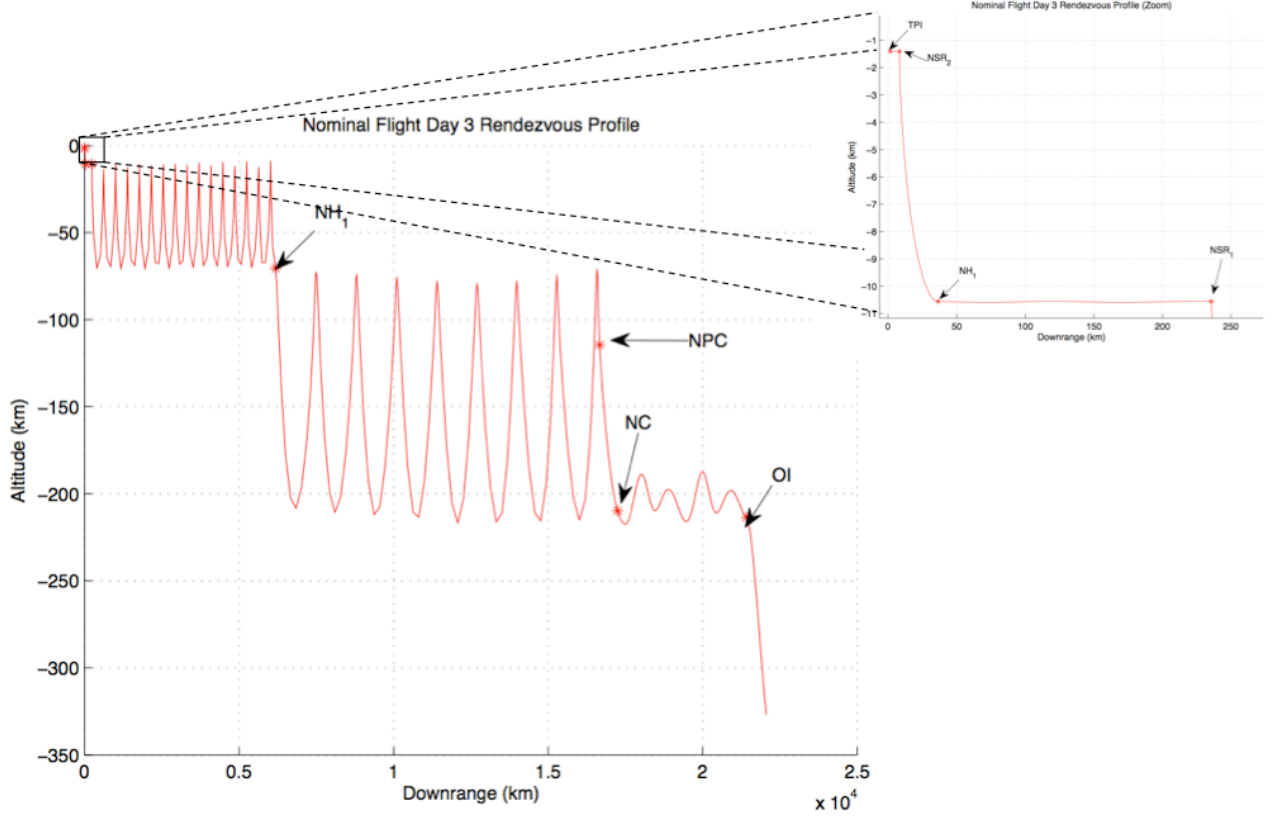


Figure 7. Nominal Orion Flight Day 3 Rendezvous Profile

The Flight Day 3 $7\text{--}8\sigma$ 20-sample dispersions and $10\text{--}11\sigma$ 10-sample dispersions are depicted in Figures 9 and 10, respectively. They show that the RTARG algorithms are very robust, even to unexpectedly large initial dispersions.

VIII. Conclusion

This paper has endeavored to describe the on-board targeting algorithms for the Orion vehicle. These algorithms have been extensively tested to date and have not experienced any convergence issues. They are currently being converted to flight code and are being tested in a 6-DOF simulation under realistic rendezvous and docking scenarios.

Appendix

We are interested in finding the optimal maneuver direction to increase the energy of the orbit. As such, the problem can be formulated as follows:

$$\min J = \Delta \mathbf{v}^T \Delta \mathbf{v} \quad (3)$$

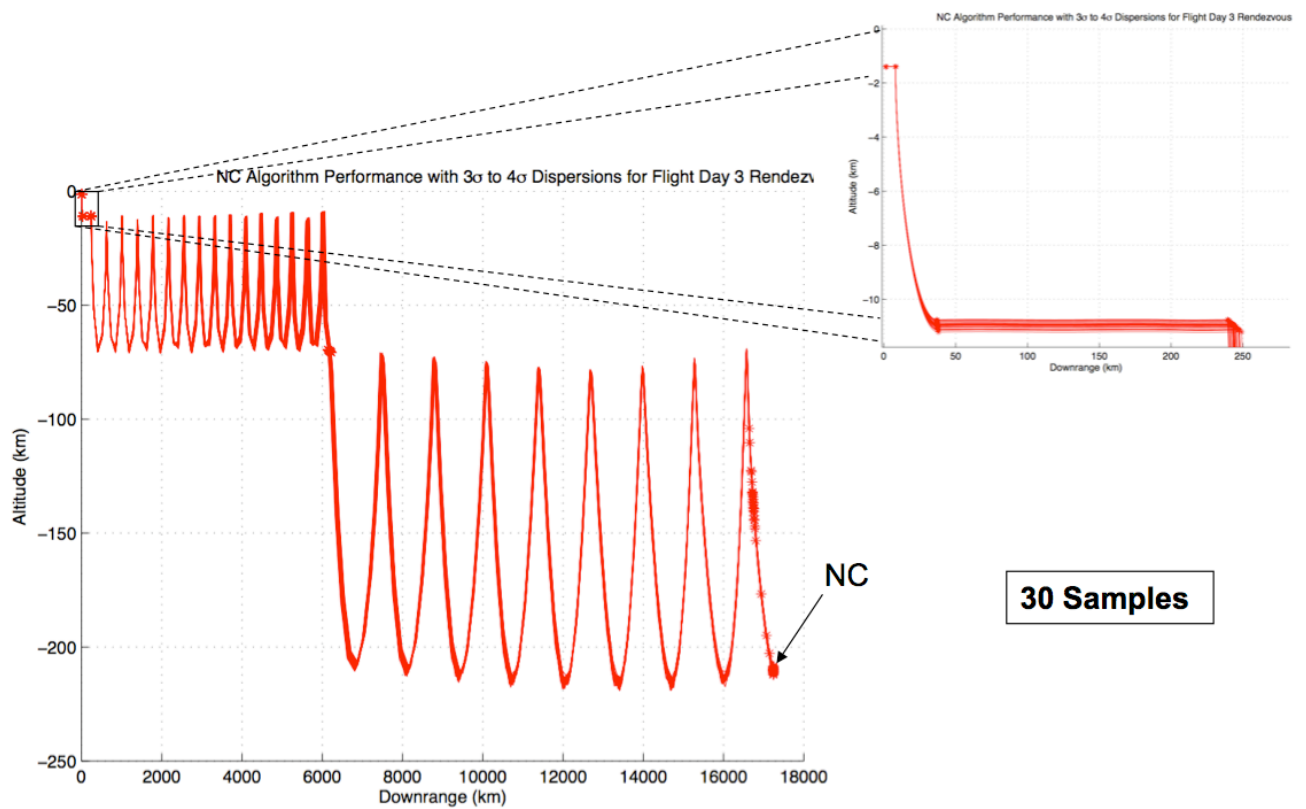


Figure 8. RTARG Performance for Flight Day 3 Rendezvous with 7-8 σ dispersions for 20 samples

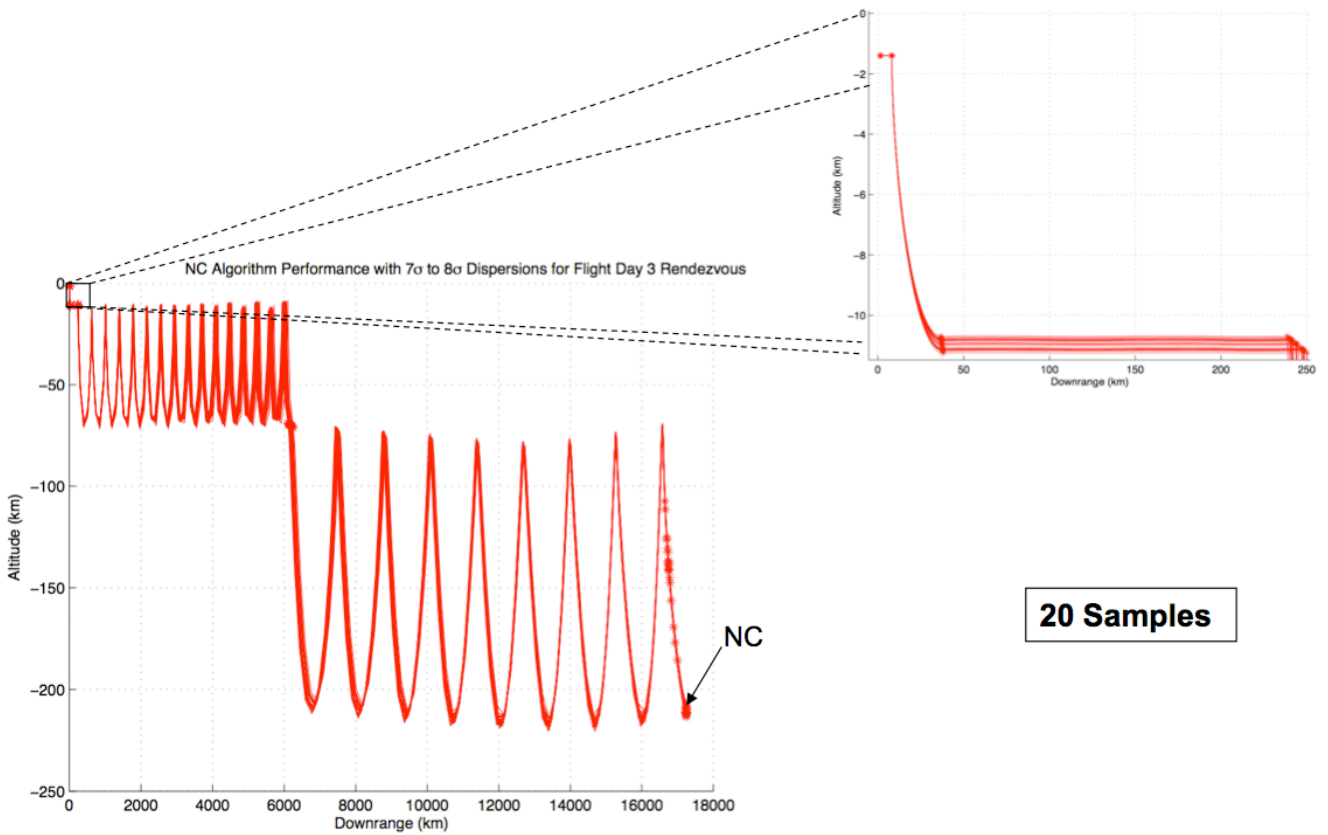


Figure 9. RTARG Performance for Flight Day 3 Rendezvous with 7-8 σ dispersions for 20 samples

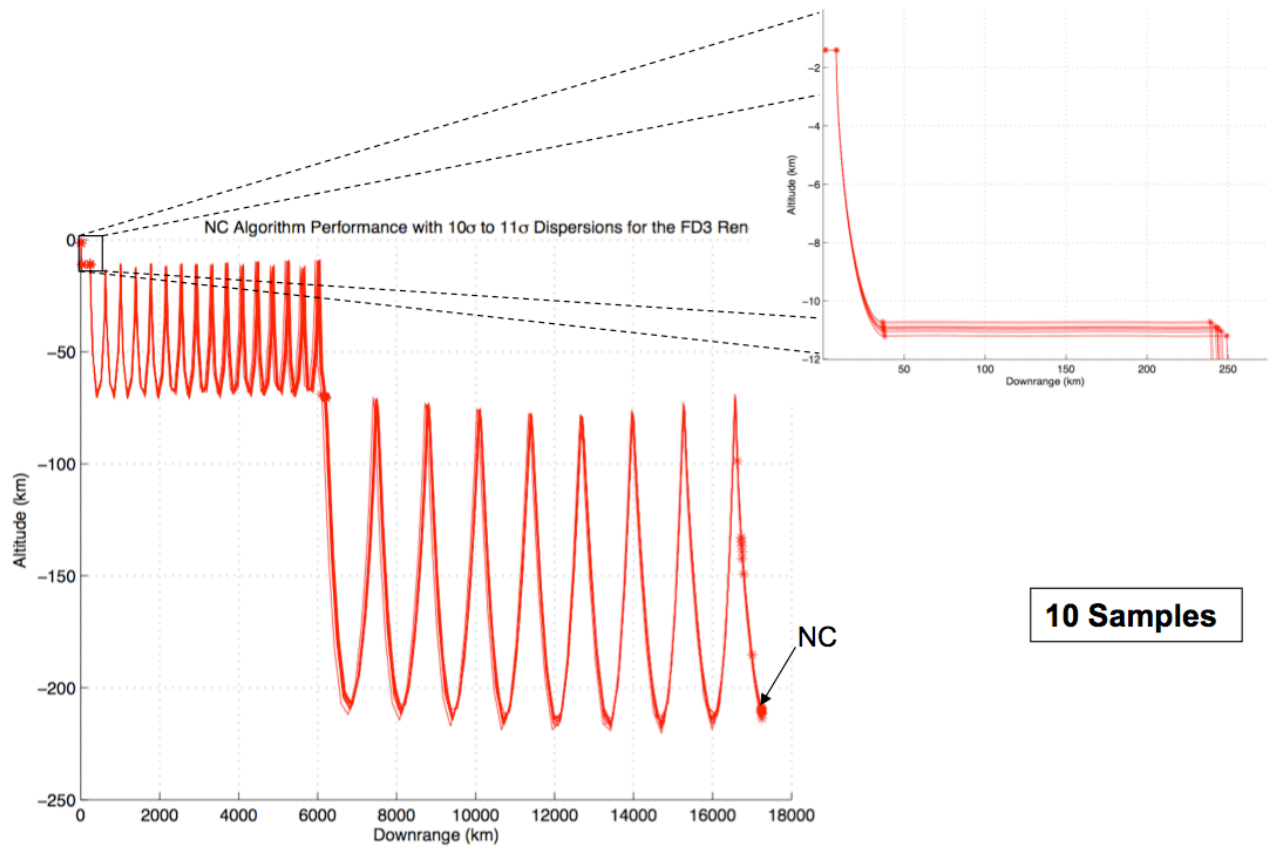


Figure 10. RTARG Performance for Flight Day 3 Rendezvous with 10-11 σ dispersions for 10 samples

subject to the following vis-viva equation

$$2\mathcal{E}_{new} - (\mathbf{v} + \Delta\mathbf{v})^T (\mathbf{v} + \Delta\mathbf{v}) = 0 \quad (4)$$

Adjoining the constraint (the vis-viva equation) to the performance index using a (scalar) Lagrange multiplier, λ , as follows

$$J' = \Delta\mathbf{v}^T \Delta\mathbf{v} + \lambda \left[2\mathcal{E}'_{new} - (\mathbf{v} + \Delta\mathbf{v})^T (\mathbf{v} + \Delta\mathbf{v}) \right] \quad (5)$$

the first variation necessary condition yields the following equation

$$\Delta\mathbf{v} = \frac{\lambda}{1 + \lambda} \mathbf{v} \quad (6)$$

Substituting into the vis-viva equation and simplifying yields

$$\Delta\mathbf{v} = \pm \left(\frac{\sqrt{2\mathcal{E}'_{new}} \mp v}{v} \right) \mathbf{v} \quad (7)$$

Since we are interested in energy *raising* maneuvers ($\mathcal{E}'_{new} > \mathcal{E}'$)

$$2\mathcal{E}'_{new} > v^2 \quad (8)$$

the upper sign is applicable and is chosen. As an aside, for energy lowering maneuvers ($2\mathcal{E}'_{new} < v^2$), the lower sign is chosen. Therefore, the optimal energy raising maneuver at any point in an elliptical orbit is

$$\Delta\mathbf{v}_{optimal} = \frac{\sqrt{2\mathcal{E}'_{new}} - v}{v} \mathbf{v} \quad (9)$$

Acknowledgments

References

- ¹R. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics*. Reston, VA: AIAA, 1999.
- ²W. Bean and R. A. Price, “Babb–Mueller Atmospheric Density Model – Calibration And Interface With The Shuttle Onorbit Navigation Flight Software,” tech. rep., Mission Planning and Analysis Division, IN–814–FM–59, JSC–16931, August 1982.