MODEL PREDICTIVE CONTROL APPLICATION TO SPACECRAFT RENDEZVOUS IN MARS SAMPLE RETURN SCENARIO

M. Saponara[1], V. Barrena[2], A. Bemporad[3], E. N. Hartley[4], J. Maciejowski[4], A. Richards[5], A. Tramutola[1], and P. Trodden[5]

[1]Thales Alenia Space (TAS) Italia
253 Strada Antica di Collegno, Torino 10146, Italy
[2]GMV
11 Isaac Newton, PTM Tres Cantos, Madrid 28760, Spain
[3]Department of Mechanical and Structural Engineering University of Trento
Trento, Italy
[4]Department of Engineering, University of Cambridge
Cambridge, U.K.
[5]Department of Aerospace Engineering, University of Bristol
Bristol, U.K.

Model Predictive Control (MPC) is an optimization-based control strategy that is considered extremely attractive in the autonomous space rendezvous scenarios. The Online Reconfiguration Control System and Avionics Architecture (ORCSAT) study addresses its applicability in Mars Sample Return (MSR) mission, including the implementation of the developed solution in a space representative avionic architecture system. With respect to a classical control solution High-integrity Autonomous RendezVous and Docking control system (HARVD), MPC allows a significant performance improvement both in trajectory and in propellant save. Furthermore, thanks to the online optimization, it allows to identify improvements in other areas (i. e., at mission definition level) that could not be known *a priori*.

1 INTRODUCTION

Within AURORA programme, the MSR mission is the main planned objective in the international effort on the Solar System exploration. Its main goal is to bring back to the Earth a sample of Martian soil. A number of new technologies will be

required to carry out this pioneering mission and one of them is the rendezvous and capture system, which will be able to detect, approach, and capture the sample of Martian soil, previously put in a predefined orbit by the Mars Ascent Vehicle (MAV).

Although autonomous docking is now a well established technology, autonomous capture (with a poorly cooperative target) is more delicate. The development of a Guidance, Navigation and Control system (GNC) for rendezvous and capture has been addressed in the European Space Agency (ESA) study named HARVD. This study has been separated into two parallel activities, one of them lead by GMV in collaboration with TAS France and Italia. The developed solution shows that, with classical control techniques, it is possible to have an automated rendezvous and capture control system with preplanned operations able to fulfill the MSR capture requirements.

Starting from HARVD experience, a further study has been defined, named ORCSAT. The objective of the study is to improve the HARVD GNC by means of optimization-based control strategies such as MPC. The work on this study was supported by the ESA under contract No. 22421.

Model predictive control (see, for example, [1–3]) is an advanced control technique which uses a prediction model and numerical optimization methods to obtain a sequence of control inputs that minimizes a function of the control inputs and predicted plant state trajectory over a given time horizon, subject to constraints. At each sampling instant, the optimization performed based on new measurement data, and the first control input of the sequence is applied. The remainder of the sequence is discarded and the process is repeated at the next sampling instant in a "receding horizon" manner. Whilst MPC has its origins in the chemical process industries [4], there is increasing interest in its application to vehicle manoeuvre problems [5–7], including spacecraft trajectory control [8–11] and attitude control [12–14]. Essentially, the application of MPC builds upon the ideas of fuel and time-optimal trajectory planning by bringing the optimization onboard, providing a natural framework for increased autonomy and reconfigurability, whilst accounting for physical and operational constraints such as finite control authority, passive safety, and collision avoidance.

The ORCSAT study considers also the developing of an MPC Framework software (SW) tool (MPCTOOL) for supporting the design, analysis and simulation of MPC-based control systems as well as the development of embedded model predictive controller for autonomous rendezvous control systems. Furthermore, another key point of the ORCSAT study is the implementation of the developed MPC control system into a space representative avionic architecture system.

The paper will briefly present the HARVD study. Afterwards, it will concentrate on the MPCTOOL description, the MPC design, and the avionic architecture system. Finally, simulation results will be shown in comparison with the HARVD ones.

## 2 THE HIGH-INTEGRITY AUTONOMOUS RENDEZVOUS AND DOCKING CONTROL SYSTEM STUDY

In the last years, the number of studies considering rendezvous and docking/ capture missions around Mars or other planets/asteroids has significantly increased. As a consequence, it is surely worth dedicating effort to consolidate maturity of GNC technologies for such missions, in order to have onboard systems with a higher and higher level of autonomy, robustness, and safety, with the final objective of decreasing costs and increasing the probability of mission success. Following this tendency, a team led by GMV and including, among others, TAS, has developed HARVD, an ESA-funded activity implementing a GNC / Autonomous Mission Management (AMM) / Fault Detection, Isolation, and Recovery onboard SW for rendezvous and docking/capture scenarios around Mars, Earth, or potentially other planets [15–17]. The HARVD, based on radio frequency (RF), camera, and LIDAR (light detecting and ranging) measurements, includes design, prototyping, and verification at three different levels: algorithms design and verification in a High-Fidelity Functional Engineering Simulator, SW demonstrator to be verified in Real Time (RT) Avionics Test Benching and Dynamic Test Benching. Rendezvous and capture on an elliptic orbit have been specially addressed, demonstrating the technical feasibility and the potential propellant saving.

The HARVD stepwise development and verification approach is shown in Fig. 1.

The Development, Verification, and Validation (DVV) approach in the HARVD activity relies on the use of COTS SW tools:

– Matlab/Simulink/Stateflow from Mathworks, including associated toolboxes, for design, analysis, simulation, and validation of system models and algorithms;

– TargetLink from dSPACE for automatic generation of production code (C code) straight from the above graphical development environment; and

– dSPACE simulator for RT development/simulation environment.

The development and integration of the High-Fidelity Functional Engineering Simulator have been successfully completed, and an intensive test campaign has been carried out. Interesting results for different MSR scenarios have been obtained, demonstrating how the strict mission requirements on performances, autonomy, safety, and robustness have been fulfilled with high margins. A special attention has been dedicated to contingency scenarios (including different onboard system failures and collision risks detection and avoidance), for which

139
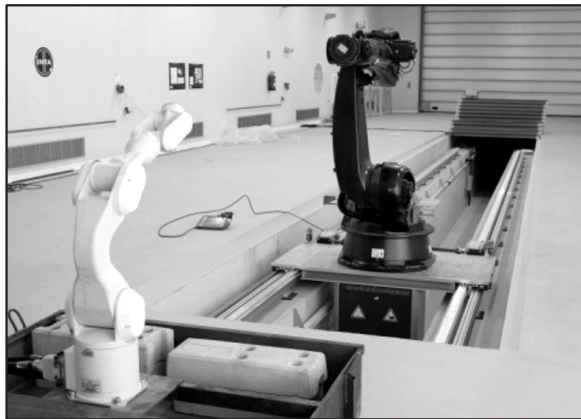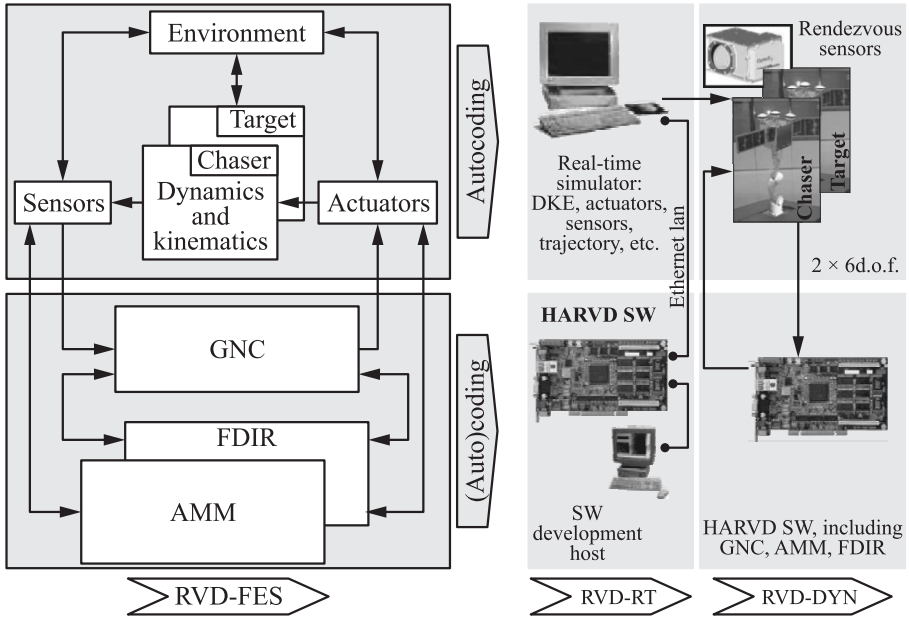
Figure 1 The HARVD stepwise development and verification approach and GMV's PLATFORM dynamic test bench (DKE — domain knowledge engine)

the results obtained are very encouraging for the consolidation of higher Technology Readiness Levels. Mars ascent vehicle circularization failures have been also taken into account, resulting in a number of elliptic target orbit rendezvous scenarios for which HARVD has demonstrated to be fully ready.

The development of RT test bench has been concluded and the acceptance RT test campaign has been successfully completed. The RT test bench is based on a LEON board GR-PCI-XC2V @45 MHz, and computational load margins of 32% have been achieved for the Worst Case Execution Time (WCET).

Recently, the tailoring of the GMV Dynamic Test Bench (PLATFORM, see Fig. 1) has already started, and the dynamic tests are foreseen to be executed in the next few months.

3 THE MPCTOOL

MPCTOOL is a MATLAB/Simulink toolbox providing all major features for the design, analysis, and simulation of model predictive controllers based on linear time-invariant (LTI) or linear time-varying (LTV) models, as well as for automatic code-generation of embedded model predictive controllers. MPCTOOL is tailored (although not limited) to the synthesis of autonomous rendezvous control systems. The inclusion of LTV capability is a key enabler for rendezvous, since elliptical orbits and J2 effects introduce time variation into the dynamics.

MPCTOOL extends the Model Predictive Control Toolbox from The Mathworks, Inc. [18] to introduce new features, modifying existing MATLAB objects, adding new functions (MATLAB methods) based on them, introducing new objects and their methods, extending the C code of the S-Function behind the basic LTI-MPC controller, and introducing new Simulink blocks coded in Embedded MATLAB (EML) for LTV-MPC. Model predictive controllers designed for LTI systems can be converted to explicit form [19] via the direct link between MPCTOOL and the Hybrid Toolbox for MATLAB [20]. Furthermore, a new Dual-Simplex solver has been developed to manage optimization problems expressed as a Linear Programming (LP) problem [21]. The new features introduced by MPCTOOL on top of the existing MPC Toolbox are the following:

- the ability to set terminal weights and constraints in LTI-MPC (including infinite-horizon MPC);

- handle variable-horizon MPC problems in which the horizon length is optimized online;

- handle quantized inputs in LTI-MPC problems;

- return the optimal sequence of MPC (both in MATLAB scripts and in Simulink), for example, to check *a posteriori* the enforcement of complex constraints not modeled in the MPC optimization model;

141

- return the optimal cost of MPC for comparing and choosing the best action among a set of model predictive controllers;

- allow the specification of convex piecewise affine stage costs (such as absolute values) on inputs and outputs;

- handle arbitrary linear constraints on combinations of inputs and outputs; and

- handle arbitrary linear time-varying models, weights, constraints, and horizons by providing two Simulink blocks based on EML code, supporting both quadratic programming (QP) and LP problem formulations.

The latter feature, namely, LTV-MPC based on LP, was employed in the studies described in this paper and will be detailed next.

The LTV model predictive controller relies on the following rather general linear time-varying prediction model:

$$
\left.
\begin{aligned}
x(j + T_s) &= A(j, x(t))x(j) + B(j, x(t))u(j) + f(j, x(t)) \,; \\
y(j) &= C(j, x(t))x(j) + D(j, x(t))u(j) + g(j, x(t)) \,; \\
z(j) &= E_z(j, x(t))(y(j) - r(j)) + H_z(j, x(t))(u(j) - u_r(j)) \\
&\qquad + P_z(j, x(t))\Delta u(j) \,; \\
c(j) &= E_c(j, x(t))x(j) + H_c(j, x(t))u(j) + P_c(j, x(t))\Delta u(j)
\end{aligned}
\right\}
\quad (1)
$$

where T_s is the sampling time; k is the prediction step; t is the current time, $j = t + kT_s$ is the prediction time; x is the state vector; u is the input vector; y is the output vector; $\Delta u(j) = u(j) - u(j - T_s)$ is the input increment; r is the output reference vector; u_r is the input reference; z is the "performance vector" to be optimized; c is the "constrained vector;" and A, B, f, C, D, g, E, H, and P are (possibly time-varying and state-dependent) matrices.

The MPC optimal problem to be optimized at each time t is

$$
\left.
\begin{aligned}
\min \quad & \rho_1 \epsilon_1 + \rho_2 \epsilon_2 + \sum_{k=0}^{N(t)-1} \| z(j) \|_1 \,; \\
\text{s.t.} \quad & \Delta u_{\min}(j) \le \Delta u(j), \quad k = 0, \dots, N(t) - 1 \,; \\
& c(j) \le c_{\max}(j) + V_c \rho_1, \quad k = 0, \dots, N(t) - 1 \,; \\
& C_N(t)x(t + N(t)T_s) \le d_N(t) + V_N \rho_2
\end{aligned}
\right\}
\quad (2)
$$

where $N(t) \le N_{\max}$ is the prediction horizon; and ρ_1 and ρ_2 are the slack variables used to soft constraints. Constraints are hardened by zeroing the corresponding entry in vectors $V_c \ge 0$ and $V_N \ge 0$.

142

The optimal control problem (2) is mapped into the LP

$$\left.\begin{array}{l} \min \quad \rho_1 \epsilon_1 + \rho_2 \epsilon_2 + \sum_{k=0}^{N(t)-1} \sum_{i=1}^{l} d_i(j) \, ; \\[2mm] \text{s.t.} \quad d_i(j) \geq \pm z_i(j) \, , \quad d_i(j) \geq 0 \, , \\[2mm] \quad + \text{MPC constraints} \end{array}\right\} \qquad (3)$$

which has $(m+l)N(t)+1$ optimization variables and, besides the nonnegativity constraints $\Delta u - \Delta u_{\min} \geq 0$, $\epsilon \geq 0$, $d_i(j) \geq 0$, $2lN(t)$ constraints to express the 1-norm in (3), plus as many constraints as the ones that are optionally defined in (2).

The user can exploit the maximum flexibility offered by the EML language to define the prediction model (1) and all the parameters appearing in the MPC optimization problem (2) in an EML module, which is then used by the LTV-MPC Simulink block to construct and solve problem (3). Accordingly, as depicted in Fig. 2, the block contains an LP builder function and a Dual Simplex LP solver coded in EML code, implementing the LTV-MPC formulation described above. The block is flexible enough to allow an arbitrary number of parameters entering the EML prediction model from the Simulink diagram as RT varying signals, to vary online prediction and control horizons, to limit *a priori* the maximum number of LP iterations.

4 THE ORCSAT MPC DESIGN

4.1 Control System Architecture and Choice of Prediction Model

There exist a large number of well-researched models for the prediction of the relative dynamics of one spacecraft with respect to another [9, 22–29]. Whilst a nonlinear model would provide the highest fidelity predictions, for the purpose of this study, it was judged that the possible gains would not be worth the additional complexity in the optimizer. Similarly, integer decision variables have been avoided as the resulting integer program would also have excessive complexity. Discrete decision making is instead handled by solving multiple instances of continuous optimizations at each control step. Therefore, the consideration is restricted to linear prediction models from which one can form the MPC optimization problem as a convex quadratic or linear program. However, out of the models considered, only the Hill–Clohessy–Wiltshire equations [22] are linear time-invariant, and these only apply to circular, or very-near circular orbits. The other models are linear parameter-varying with respect to the true anomaly of the target ν_{tgt}. However, because the target
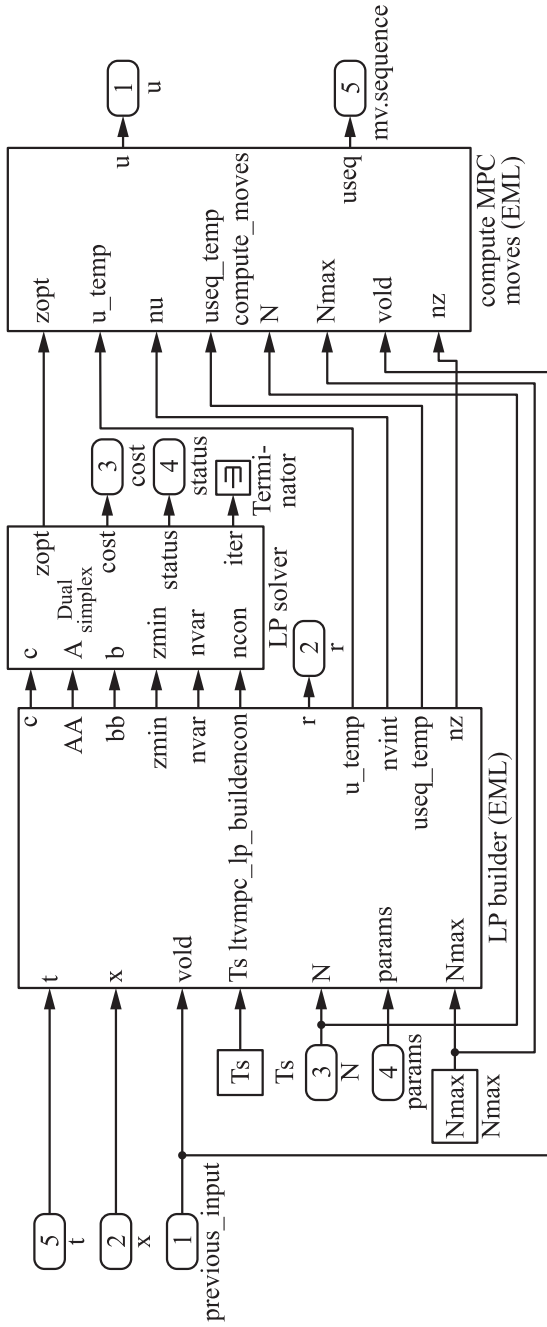
Figure 2 Simulink diagram underlying the LP-based LTV-MPC block

Table 1 Rendezvous phases

Phase	Requirement
Orbit Synchronization Transla-tional Guidance (OSTG)	To bring chaser from a distance of approx-imately 300 km into the same orbit as the target, with an in-track separation of be-tween 5 and 30 km on either side of the target
Impulsive Nominal Translational Guidance (INTG)	To perform passively safe impulsive trans-fers between a sequence of predefined hold-ing points in the same orbit as the tar-get until an in-track separation of 100 m is reached
Forced Terminal Translational Guidance (FTTG)	To track a straight-line trajectory from 100- to 3-meter separation from the target such that a subsequent free-drift trajectory captures the target with a 20-centimeter tolerance
Collision Avoidance Manoeuvre (CAM)	To bring the chaser to a safe distance, fur-ther than 5 km from the target within 3 or-bits, avoiding collision in the process

is passive, ν_{tgt} can be calculated as a function of time using Kepler's equa-tion [30], thus allowing a linear time-varying representation of the relative dy-namics.

The objective of the MPC control system designed during this study is to bring the chaser craft from the point of target detection at a range of approxi-mately 300 km, via a sequence of holding points in the same orbit as the target, to a "blinding point" approximately 3 m from the target, at which point it should be moving towards the target at an in-track velocity of 0.1 m/s. Target capture is then completed on a passive drift trajectory. The MPC system pro-vides both guidance and control and is not restricted to tracking predetermined trajectories.

To achieve this objective using a single model predictive controller would re-quire a prediction horizon sufficient to predict a trajectory at least one orbital period into the future, a sufficiently complex prediction model to perform accu-rate trajectory propagations over long periods of time, and a sampling period short enough to allow target capture within a 20-centimeter tolerance. Given finite computational resources, this is not a practical solution. The rendezvous is, therefore, divided into three phases similar to those used in HARVD [31, 32], with an additional controller to perform a collision avoidance manoeuvre (CAM) in case of a fault during the final moments of the rendezvous (Table 1).

145

4.1.1 Orbit synchronization translational guidance

The first phase, Orbit Synchronization Translational Guidance (OSTG), has the objective of bringing the chaser from a distance of approximately 300 km into the same orbit as the target using thrusters, with an in-track separation of between 5 and 30 km on either side of the target, whilst minimizing propellant consumption and manoeuvre time. At these ranges, short-term control accuracy is not critical; so, a relatively long prediction time can be used. However, long-term prediction accuracy is important in order to perform optimal manoeuvres. For these reasons, the J_2-modified Gauss's variational equation (GVE) prediction model of [9] is chosen. This predicts the relative trajectory between the chaser and target in terms of the relative Keplerian orbital elements rather than relative positions and velocities in a rectangular or cylindrical coordinate frame, whilst using the Gim–Alfriend [29] approach of incorporating the effects of J_2 to account for variations in gravity due to the oblateness of the central body of the orbit. Because the relative orbital elements are small, despite large Euclidean separations, the effects of linearisation error are small in comparison to prediction models such as those of [22, 28], which use rectangular or cylindrical relative coordinates. The system input is assumed to be an impulsive change in velocity (ΔV) in a local orbital reference frame centered on the chaser.

4.1.2 Impulsive nominal translational guidance

The second phase, Impulsive Nominal Translational Guidance (INTG), must perform a sequence of passively safe impulsive transfers between a sequence of predefined holding points until an in-track separation of 100 m is reached. Greater control accuracy is required during this phase, necessitating a shorter sampling period. In addition, collision avoidance constraints must be more fine-grained. However, as the OSTG phase will have reduced much of the radial and out-of-plane separation between chaser and target, the effect of linearization error on the Yamanaka–Ankersen [28] equations is no longer a problem, as long as a cylindrical relative coordinate system is used [26]. This model is less complex than the J_2-modified GVEs and allows objectives and constraints to be directly specified in the cylindrical frame without requiring a linearized geometric transformation (with inevitable loss of accuracy) from the relative orbital elements. The prediction model input is assumed to be an impulsive ΔV in the cylindrical target orbital frame.

4.1.3 Forced terminal translational guidance

The third phase, Forced Terminal Translational Guidance (FTTG), is tasked with bringing the chaser from its final holding point at 100 m from the target

146

to a position 3 m from the target from where it can capture the target on a free drift trajectory. Radial, in-track, and out-of-plane separation are small during this phase. Control accuracy is critical due to the tight capture tolerances, and a much higher sampling rate is required than for other phases. As for the INTG phase, the Yamanaka–Ankersen [28] equations are used for the trajectory prediction model.

In addition, to maintain target pointing, the model predictive controller must also handle attitude regulation to an externally provided setpoint, using thrusters. A linearized quaternion-based prediction model [13] extended to consider the elliptical orbital dynamics is used for the relative attitude control. The attitude reference frame used for control is chosen depending on the direction of approach, and the attitude setpoint in the inertial frame to avoid the predicted trajectory crossing the discontinuity at $\pm 180°$ in the quaternion representation [33]. Because the prediction matrices are rebuilt at each time step due to the LTV prediction model, the opportunity is taken to relinearize the attitude dynamics about the current measured attitude at each time step.

4.1.4 Collision avoidance manoeuvre

The CAM must safely move the chaser away from the target, to a distance of 500 m within three orbits without collision with the target. Essentially, this objective is similar to that of INTG, except traveling away from the target instead of towards it, and with a less specific terminal objective. It, therefore, makes sense to use the Yamanaka–Ankersen prediction model for this phase also.

4.2   Model Predictive Control Subsystem Design

Each of the model predictive controllers is designed independently, but with a common interface and a common output function to convert the ΔV into finite-duration thrust pulses in the inertial frame. The core MPC function of each control subsystem is implemented using the blocks from the MPCTOOL, with the linear time-varying prediction models implemented as EML functions called by the MPCTOOL blocks. Any additional logic or reference-frame changes are implemented using Simulink blocks.

4.2.1 Orbit sinchronization translational guidance model predictive controller

The OSTG model predictive controller must bring the chaser into the same orbit as the target in a timely manner, whilst minimizing propellant consumption. Rather than using the more commonly used quadratic cost function, to correctly
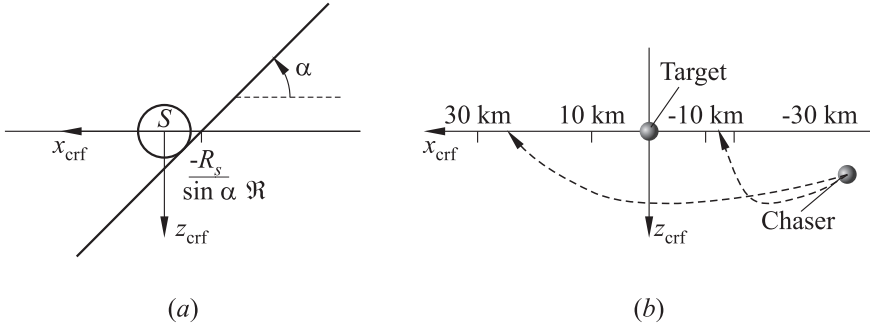
Figure 3 The OSTG safety (*a*) and terminal (*b*) constraints

encode the minimization of total propellant consumption, the model predictive controller must minimize the absolute sum of ΔV applied over the prediction horizon [34]. Furthermore, to balance this with time to completion, a terminal constraint enforcing the completion criteria is imposed at the end of the prediction horizon, and the prediction horizon itself is included as a decision variable in the cost function [6, 35, 36]. Letting N be the prediction horizon, $\mathbf{u} = [u(t + T_s|t)^{\mathrm{T}}, \cdots, u(t + (N-1)T_s|t)^{\mathrm{T}}]^{\mathrm{T}}$ and α be a parameter determining constraints that will be described later, the cost function is:

$$J_{\mathrm{OSTG}}(\alpha, \mathbf{u}, N) = N + \sum_{k=1}^{N-1} \| w_u u(t + kT_s|t) \|_1 \ .$$

Note that the summation is from $k = 1$ not $k = 0$, implying that the input calculated at the current time step is applied at the next time step to allow sufficient time duration for computation to occur. The terminal constraint, which will be described later, ensures that the predicted trajectory ends in the correct orbit, with an acceptable separation from the target.

In order the predicted trajectories do not collide with the target, constraints are placed on the predicted trajectories to ensure that they do not enter a safety sphere of radius $R_s(t)$, surrounding the target. In addition, as proposed in [10], unforced drift trajectories emanating from each point in the prediction horizon are also constrained to ensure passive safety. Collision avoidance is a manifestly nonconvex constraint, but it is approximated by a half-space constraint with angle relative to the in-track direction parameterized by α (Fig. 3). The value of α then determines on which side of the target the terminal constraint places the end of the predicted trajectory.

The optimization is implemented using the "LP-based LTV model predictive controller" block from MPCTOOL, which allows prediction horizon N and user-defined parameters to be passed in as signals. Given an angle α_0 calculated as

148

the current angle between the chaser and the z_{crf} axis, rounded to the nearest $45°$, by solving $3N$ convex optimizations, varying N between 1 and $N_{\max}$, for each $\alpha \in \{\alpha_0 - 45°, \alpha_0, \alpha_0 + 45°\}$ using two nested Simulink "For-iterator" subsystems, the control sequence can be found that minimizes the overall cost function. A sampling period $T_S = 600$ s was chosen, along with a maximum prediction horizon $N_{\max} = 25$.

4.2.2 Impulsive nominal translational guidance model predictive controller

The INTG model predictive controller must transfer the chaser between a sequence of invariant holding points on $\overline{V}$ (i. e., the in-track axis in the cylindrical orbital frame) until a separation of 100 m is achieved. Because release from these holding points must be governed by an external signal, there is no point predicting further ahead than the end of a single transfer. It is sufficient to design a controller to perform a transfer, parameterized by the distance from the target of the next holding point.

The design is similar to that of the OSTG model predictive controller in that a 1-norm cost function is used in conjunction with a variable horizon implemented by solving multiple convex optimizations. However, the cost function includes distance instead of time to reflect that fuel consumption is proportional to distance traveled rather than time when carrying out passively safe hopping trajectories. The holding points are scheduled by an external algorithm and parameterized by distance x_{hp}. The cost function is

$$J_{\mathrm{INTG}}(\mathbf{u}, N) = \sum_{k=1}^{N-1} \| E_c(x_{\mathrm{crf}}(t + kT_s|t) - r(t + kT_s)) \|_1 + \| w_u u(t + kT_s|t) \|_1$$

where $r(t + kT_s) = [\pm x_{\mathrm{hp}}(1 + e_{\mathrm{tgt}} \cos \nu_{\mathrm{tgt}}(t + kT_s)), \ 0, \ 0, \ 0, \ 0, \ 0]^{\mathrm{T}}$ depends on the direction of approach; x_{crf} is the state vector in the cylindrical reference frame; e_{tgt} is the eccentricity of the target orbit; ν_{tgt} is the true anomaly of the target; and

$$E_c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

As for the OSTG model predictive controller, passive safety constraints are imposed over a period of one orbit from each prediction in the control horizon. In addition, to ensure passive safety over a longer period, an additional passive drift constraint is imposed to make sure that long-term secular drift is away from the target (thus avoiding collision in subsequent orbits). Letting $A_{\mathrm{orb}}(\nu_{\mathrm{tgt}})$ be the propagation matrix for a whole orbit,

$$[-\mathrm{sgn}\,(x_{\mathrm{crf}}(t)) \ \ 0 \ 0 \ 0 \ 0 \ 0]\,(A_{\mathrm{orb}}\,(\nu_{\mathrm{tgt}}(t + kT_s)))\,x_{\mathrm{crf}}(t + kT_s|t) \leq 0\,.$$

149

The terminal set for the INTG model predicted controller is defined as a box with side-length $2(e_{\text{tgt}}+0.1)$, centered on a point x_{hp} away from the target on $\overline{V}$, with an additional constraint that the chaser should be on a periodic trajectory and also be inside the box after $^{1}/_{4}$, $^{1}/_{2}$, and $^{3}/_{4}$ orbits of free drift. A sampling period $T_s = 300$ s and a maximum prediction horizon of $N_{\text{max}} = 20$ were chosen.

4.2.3 Forced terminal translational guidance model predictive controller

During the FTTG phase, trajectory and attitude tracking accuracy becomes more important than long-term fuel minimization. The navigation uncertainty is of a similar order of magnitude to the expected tracking errors; so, a conventional quadratic cost function is appropriate. The controller is implemented using the "QP-based LTV model predictive controller" block from MPCTOOL, with a sampling period $T_s = 3$ s and a prediction horizon $N = 15$. Letting $x(j|t)$ be the combined position, velocity, attitude quaternion, and angular velocity states, $r(j)$ the corresponding reference setpoint, and $u(j|t)$ the vector of thruster inputs, the cost function is:

$$J_{\text{FTTG}} = \sum_{k=1}^{N-1} (x(t + kT_s|t) - r(t + kT_s))^{\text{T}} Q(x(t + kT_s|t) - r(t + kT_s))$$
$$+ \Delta u(t + kT_s|t)^{T} R \Delta u(t + kT_s|t) .$$

Changes in input (Δu) are penalized instead of the absolute input value to enable offsetfree tracking of forced-equilibrium setpoints [37]. Positivity and saturation constraints are applied to inputs. The reference trajectory $r(j)$ and cost function weightings $Q \geq 0$ and $R \geq 0$ are chosen so that the controller tracks an attitude setpoint, a position in the radial and out-of-plane directions, and an approach velocity in the in-track direction.

4.2.4 Collision avoidance manoeuvre model predictive controller

The CAM model predictive controller is based on a modified version of the INTG model predictive controller. However, in order for rapid response, a delay of T_s is not assumed in the model. Instead, it is assumed that the calculation of the control move will complete as fast as possible. To facilitate the fast computation, a variable horizon is not used for CAM, and the trajectory is constrained so that only one impulsive ΔV may be applied at the beginning of the prediction horizon. This is applied open-loop on the assumption that navigation error may increase outside nominal operational levels following the fault triggering the CAM, especially if attitude pointing is lost. The terminal constraint is

chosen so that under the specified worst-case navigation error, the chaser will be further than 500 m away from the target in three orbits. The INTG model predictive controller can then hold the chaser in a periodic fly-around orbit or restart approach via its sequence of holding points once the CAM is complete.

5 THE ONLINE RECONFIGURATION CONTROL SYSTEM AND AVIONICS ARCHITECTURE

The avionic architecture considered in HARVD is based on the "Aurora Avionics Architecture" ESA study, which is the avionics reference for future exploration vehicles. From this starting point, the ORCSAT study includes the design of an Avionic Architecture System allowing the implementation of embedded MPC-based systems. The MPC concept is based on the optimization of a cost function under some constraints, which usually is carried out using quite complex iterative algorithms, requiring high computational capability. Therefore, the main challenge of the avionic architecture design is to define a Central Data Management Unit (CDMU) able to cope with the MPC needs. In particular, the selection of the Central Processing Unit (CPU) is the key for the MPC embedded implementation.

Since the beginning of the design, it was evident that a CPU composed of a processor and a coprocessor has to be considered as baseline (distributed architecture) taking into account available space qualified processor computational performances (Fig. 4). This solution allows a distribution of the complete on-board SW on the two processors leaving to the coprocessor the execution of GNC algorithms requiring significant computational throughput (MPC) and to the processor the handling of the system units, the other parts of the GNC, etc.

The next step was the selection of the processors. Currently space-qualified processors are based on LEON2 FT, with performance of 86 MIPS, 23 MFLOPS at 100 MHz: taking into account also the HARVD experience, it has been considered adequate for the central processor of the CDMU.

Regarding the coprocessor, the MPC computational throughput is the driver for the selection. To support this task, profiling of the MPC algorithm was performed using Simulink features, to evaluate the time needed for the execution of the MPC algorithms. Afterwards, these timing values have been scaled to the selected processor exploiting the Whetstone benchmark. The processors selected for the trade-off are the LEON2 FT and PowerPC750FX, able to perform 1650MIPS at 733 MHz which has been used to design space-qualified boards like the Maxwell SCS750.

The profiling results are summarized in Table 2. It can be seen that the LEON2 FT cannot be selected as coprocessor, since the FTTG would take more than 4 s for the computation of the control action against a theoretical con-
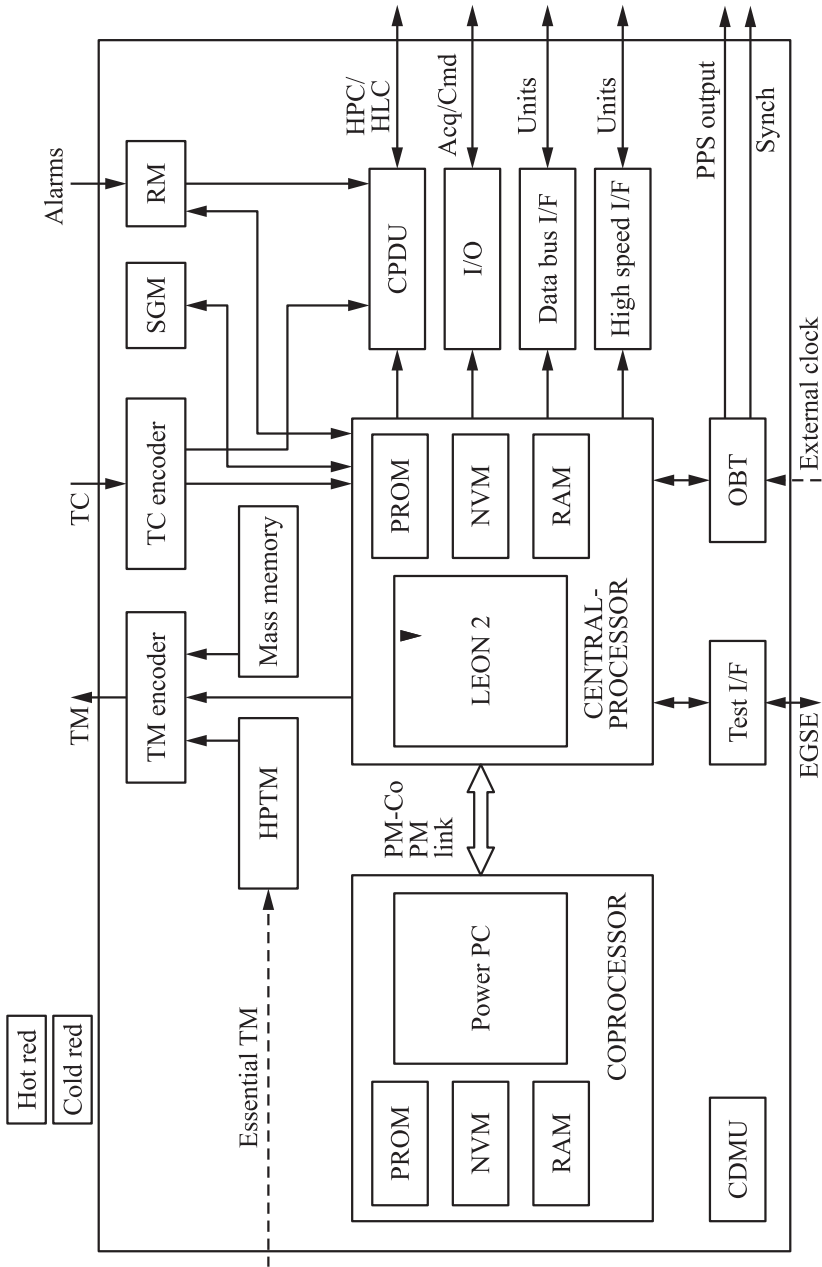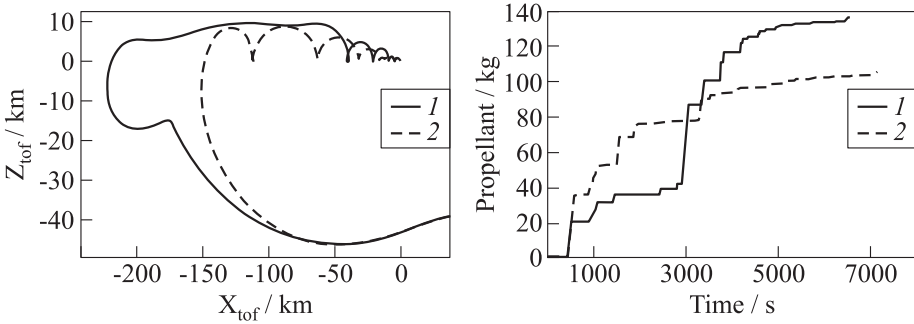
Figure 4 Central data management unit with distributed architecture

Table 2 Model predictive control profiling

| Mode	Maximal time, s	Time, s	
	Workstation	Scaling to LEON2 FT	Scaling to PowerPC 750FX
OSTG	7.8575	410.9473	23.7401
INTG	1.9028	99.5164	5.7490
FTTG	0.0885	4.6286	0.2674
CAM	0.1249	6.5323	0.3774

Figure 5 The HARVD (*1*) vs. MPC (*2*) performance comparisons

trol step of 3 s. Instead, the PowerPC 750FX shows timings which are widely compatible with the MPC design and expected computational capabilities and, therefore, it has been selected as baseline for the CPU coprocessor.

6   SIMULATION RESULTS AND COMPARISONS

Figure 5 shows the comparison between the simulation results obtained with the HARVD GNC solution and the ones obtained with the MPC in the case of rendezvous circular orbit. Differences are visible since the beginning of the rendezvous, where the MPC trajectory remains closer to the target with respect to HARVD, but the most significant result is the propellant save, which in this case is about 35 kg.

The main differences can be found analyzing the trajectory during the OSTG. In this phase, the MPC design is such that the chaser is left at a relative distance with respect to the target between 5 and 30 km: with the final MPC tuning, it has been noted that the chaser is left at the end of OSTG usually at 15 km from the target. The latter finding suggested a different definition of the holding points, which in the HARVD initial solution started from 50 km: there-

153
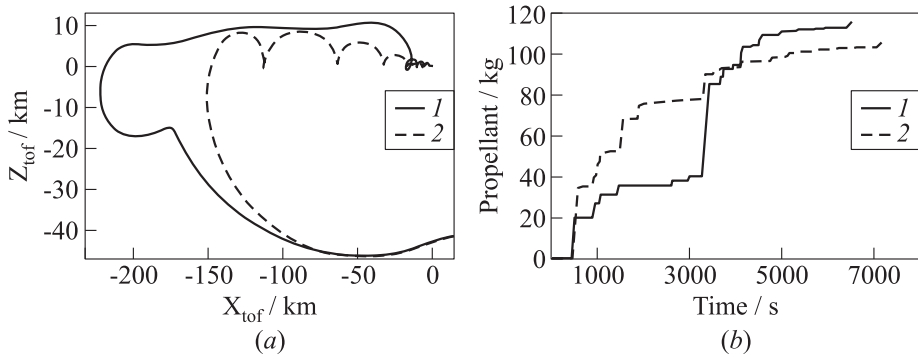
Figure 6 The HARVD (*1*) vs. MPC (*2*) performance comparisons with new holding points definition
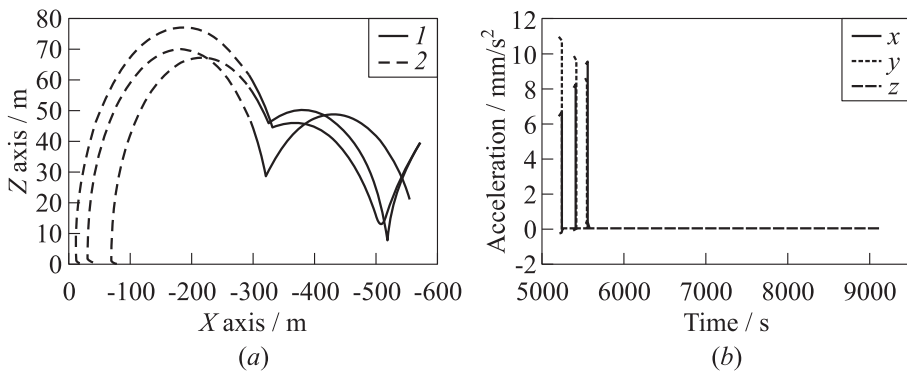

Figure 7 The CAM simulation results: *1* — fly-around and *2* — CAM

fore, the HARVD simulation has been repeated with the first holding point at 20 km. Figure 6 shows that HARVD performance improved a lot, in particular, on the propellant consumption: in this case, the difference is reduced to about 10 kg.

This aspect is quite important: the online optimization performed by MPC has permitted the detection of a possible improvement in the nominal mission scenario (i. e., definition of the holding points) that would be difficult to clearly identify *a priori*.

Figure 7 shows the trajectories obtained in three simulations where the CAM is triggered at different relative distances from the target and the following fly-around. Performances are very good, since MPC allows avoiding the collision also at very short distance (10 m) with a single manoeuvre.

154
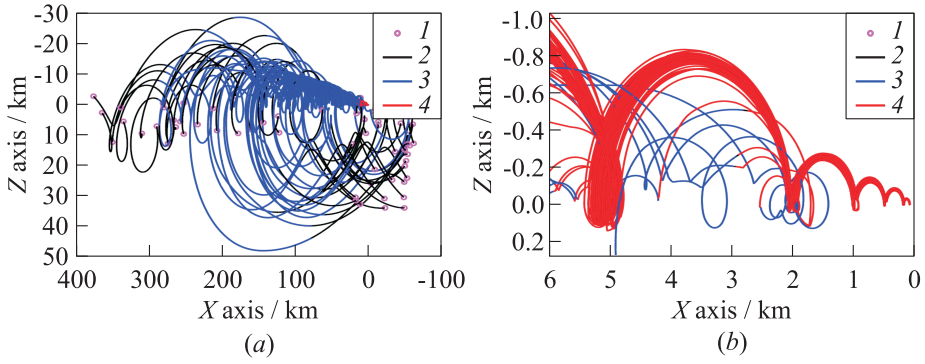
Figure 8 Overall trajectories during OSTG and INTG of 50 cases of the Monte-Carlo campaign: *1* — initial position; *2* — no control; *3* — OSTG; and *4* — INTG.
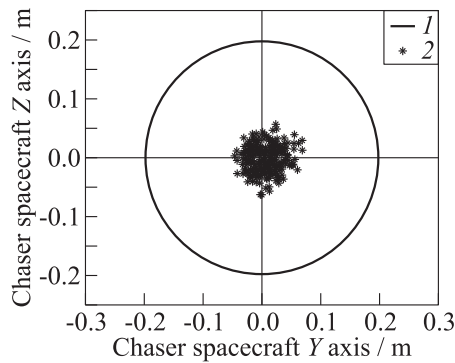
Figure 9 Capture performance of 400 test cases of the Monte-Carlo campaign: *1* — 20-centimeter requirement; and *2* — target center

The MPC solution has been also validated and verified by means of a Monte-Carlo simulation campaign composed by 800 test cases, in order to test the performance of the control in different scenarios (circular and elliptic orbit) and starting from different initial relative positions and dynamics with respect to the target. The obtained results are very good, since the capture has been always achieved with margins.

Figure 8 shows a typical result of this campaign, summarizing 50 test cases trajectories during OSTG and INTG. Instead, Fig. 9 shows the aggregated capture accuracy results of 400 cases.

155

7   CONCLUDING REMARKS

Optimization-based control techniques like MPC are considered extremely attractive for applications which require high level of autonomy, optimal path planning, and dynamic safety margins. The ORCSAT study is addressing the usage of the MPC techniques on the rendezvous and capture scenarios of the MSR mission. The results obtained after the design phase are encouraging, since with respect to classical control techniques (HARVD), it is possible to have a significant improvement, in particular, in the propellant consumption. As side effect, but not less important, online optimization could drive the definition of higher level mission aspects that could not be easy to address in the earlier phase of GNC design.

The MPC design has been verified and validated throughout a wide Monte-Carlo simulation campaign which considers plant mismatch, sensors and actuators failures, and different initial dynamic conditions. Obtained results confirm the robustness of the design and the very good performance.

The next step of the ORCSAT study will be the implementation of the MPC algorithms in the selected avionic architecture, with the objective to test the RT performance of the developed solution on a flight-representative avionic. In the end, the GMV Dynamic Test Bench will be enhanced with the model predictive based control and the selected avionics for the final dynamic test campaign.

REFERENCES

1. Maciejowski, J. M. 2002. *Predictive control with constraints*. Pearson Education.
2. Camacho, E. F., and C. Bordons. 2004. *Model predictive control*. London: Springer-Verlag.
3. Rawlings, J. B., and D. Q. Mayne. 2009. *Model predictive control: Theory and design*. Nob Hill Publishing.
4. Qin, S. J., and T. A. Badgwell. 2003. A survey of industrial model predictive control technology. *Control Eng. Practice* 11(7):733–64.
5. Shim, D. H., H. J. Kim, and S. Sastry. 2003. Decentralized nonlinear model predictive control of multiple flying robots. *42nd IEEE Conference on Decision and Control Proceedings*. 4:3621–26. Maui, Hawaii, USA.
6. Richards, A., and J. P. How. 2006. Robust variable horizon model predictive control for vehicle maneuvering. *Int. J. Robust Nonlinear Control* 16(7):333–51.
7. Almeida, F. A. 2008. Waypoint navigation using constrained infinite horizon model predictive control. *AIAA Guidance, Navigation and Control Conference and Exhibit Proceedings*. Honolulu, Hawaii.
8. Larsson, R., S. Berge, P. Bodin, and U. Jönsson. 2006. Fuel efficient relative orbit control strategies for formation flying and rendezvous within PRISMA. *29th Annual AAS Guidance and Control Conference Proceedings*.

9. Breger, L., and J. P. How. 2007. Gauss's variational equation-based dynamics and control for formation flying spacecraft. *J. Guidance Control Dyn.* 30(2):437–48.

10. Breger, L., and J. P. How. 2008. Safe trajectories for autonomous rendezvous of spacecraft. *J. Guidance Control Dyn.* 31(5):1478–89.

11. Bodin, P., R. Noteborn, R. Larsson, and C. Chasset. 2011. System test results from the GNC experiments on the PRISMA in-orbit test bed. *Acta Astronautica* 68(7-8):862–72.

12. Manikonda, V., P. O. Arambel, M. Gopinathan, R. K. Mehra, and F. Y. Hadaegh. 1999. A model predictive control-based approach for spacecraft formation keeping and attitude control. *American Control Conference Proceedings*. San Diego, CA. 6:4258–62.

13. Hegrenæs, O., J. T. Gravdahl, and P. Tondel. 2005. Spacecraft attitude control using explicit model predictive control. *Automatica* 41(12):2107–14.

14. Wood, M., and W. H. Chen. 2008. Model predictive control of low Earth orbiting satellites using magnetic actuation. *Proc. of the Institution of Mechanical Engineers, Part I: J. Syst. Control Eng.* 222(6):619–31.

15. Colmenarejo, P., L. Tarabini, C. Le Peuvédic, and A. Guiotto. 2008. HARVD development, verification and validation approach (from traditional GNC design/V&V framework simulator to real-time dynamic testing). *7th ESA Conference (International) on Guidance, Navigation and Control Systems*. Tralee, County Kerry, Ireland.

16. Barrena, V., P. Colmenarejo-Matellano, D. Modrego-Contreras, C. Le Peuvédic, and A. Guiotto. 2008. Integrated development, verification and validation approach for space systems using autocoding techniques. *Data System in Aerospace Conference (DASIA 2008)*. Palma Majorca, Spain.

17. Strippoli, L., P. Colmenarejo, T. V. Peters, C. Le Peuvédic, and T. Voirin. 2010. High integrity control system for generic autonomous RVD. *61st Astronautical Congress (Internatioanl)*. Prague, CZ.

18. Bemporad, A., M. Morari, and N. L. Ricker. 2009. *Model predictive control ToolboxTM 3 — user's guide*. The Mathworks, Inc. http://www.mathworks.com/access/helpdesk/help/toolbox/mpc/.

19. Bemporad, A., M. Morari, V. Dua, and E. N. Pistikopoulos. 2002. The explicit linear quadratic regulator for constrained systems. *Automatica* 38(1):3–20.

20. Bemporad, A. 2009. *Hybrid Toolbox v1.2.2 — user's guide*. Dec. http://www.ing.unitn.it/∼bemporad/hybrid/toolbox.

21. Bertsimas, D., and J. N. Tsitsiklis.1997. *Introduction to linear optimization*. Athena Scientific.

22. Clohessy, W. H., and R. S. Wiltshire. 1960. Terminal guidance system for satellite rendezvous. *J. Aerospace Sci.* 27(9):653–58.

23. Tschauner, J. 1967. Elliptical orbit rendezvous. *AIAA J.* 5(6):1110–13.

24. Carter, T. E. 1998. State transition matrices for terminal rendezvous studies: Brief survey and new example. *J. Guidance Control Dyn.* 21(1):148–55.

25. Battin, R. H. 1999. *An introduction to the mathematics and methods of astrodynamics*. Revised edn. AIAA education ser. American Institute of Aeronautics and Astronautics.

157

26. Melton, R. G. 2000. Time-explicit representation of relative motion between elliptical orbits. *J. Guidance Control Dyn.* 23(4):604–10.

27. Schaub, H., S. R. Vadali, J. L. Junkins, and K. T. Alfriend. 2000. Spacecraft formation flying control using mean orbital elements. *J. Astronautical Sci.* 48:69–87.

28. Yamanaka, K., and F. Ankersen. 2002. New state transition matrix for relative motion on an arbitrary elliptical orbit. *J. Guidance Control Dyn.* 25(1):60–66.

29. Gim, D., and K. T. Alfriend. 2003. State transition matrix of relative motion for the perturbed noncircular reference orbit. *J. Guidance Control Dyn.* 26(6):956–71.

30. Sidi, M. J. 1997. *Spacecraft dynamics and control: A practical engineering approach.* Cambridge University Press.

31. Kerambrun, S., N. Despré, B. Frapard, P. Hyounet, B. Polle, M. Ganet, N. Silva, A. Cropp, and C. Philippe. 2008. Autonomous rendezvous system: The HARVD solution. *7th ESA Conference (International) on Guidance, Navigation and Control Systems Proceedings.* Tralee, Ireland.

32. Le Peuvédic, C., P. Colmenarejo, and A. Guiotto. 2008. Integrated multi-range RDV control system — autonomous RDV GNC test facility — HARVD control system trade-off analysis and baseline solution. Technical Report GMV-HARVD-TN06. GMV.

33. Bach, R., and R. Paielli. 1993. Linearization of attitude-control error dynamics. *IEEE Trans. Automatic Control* 38(10):1521–25.

34. Tillerson, M., G. Inalhan, and J. P. How. 2002. Co-ordination and control of distributed spacecraft systems using convex optimization techniques. *Int. J. Robust Nonlinear Control* 12(2-3):207–42.

35. Richards, A., and J. How. 2003. Performance evaluation of rendezvous using model predictive control. *AIAA Guidance, Navigation and Control Conference and Exhibit.* Austin, Texas.

36. Richards, A. G., and J. P. How. 2003. Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility. *2003 American Control Conference Proceedings.* Denver, Colorado. 5:4034–40.

37. Maciejowski, J. M. 1998. The implicit daisy-chaining property of constrained predictive control. *Appl. Math. Computer Sci.* 8(4):101–17.