Model Predictive Control application to spacecraft rendezvous in Mars Sample Return scenario

M. Saponara, V. Barrena**, A. Bemporad***, E. N. Hartley****, J. Maciejowski****, A. Richards*****, A. Tramutola*, P. Trodden******

** Thales Alenia Space Italia, Strada Antica di Collegno, 253 - 10146 Torino (Italy), email: massimiliano.saponara@thalesaleniaspace.com*
antonio.tramutola@thalesaleniaspace.com
*** GMV., Isaac Newton, 11. 28760 PTM Tres Cantos. Madrid. Spain,*
email: vbarrena@gmv.com
**** Department of Mechanical and Structural Engineering University of Trento, Italy,*
email: bemporad@ing.unitn.it
***** Department of Engineering, University of Cambridge,*
email: enh20@eng.cam.ac.uk
jmm@eng.cam.ac.uk
****** Department of Aerospace Engineering, University of Bristol*
email: Arthur.Richards@bristol.ac.uk
paul.trodden@ed.ac.uk

Abstract
Model Predictive Control (MPC) is an optimization-based control strategy that is considered extremely attractive in the autonomous space rendezvous scenarios. The ORCSAT study addresses its applicability in Mars Sample Return mission, including the implementation of the developed solution in a space representative avionic architecture system. With respect to a classical control solution (HARVD), MPC allows a significant performance improvement both in trajectory and in propellant save. Furthermore, thanks to the on-line 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 Mars Sample Return (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 ESA study named High integrity Autonomous RendezVous and Docking control system (HARVD). This study has been separated into two parallel activities, one of them lead by GMV in collaboration with Thales Alenia Space (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 pre-planned operations able to fulfill the MSR capture requirements.

Starting from HARVD experience, a further study has been defined, named On-line Reconfiguration Control System and Avionics Architecture (ORCSAT). The objective of the study is to improve the HARVD GNC by means of

optimization-based control strategies such as Model Predictive Control (MPC). The work on this study was supported by the European Space Agency under contract No. 22421.
MPC (e.g. [11], [19], [26]) is an advanced control technique which uses a prediction model and numerical optimisation methods to obtain a sequence of control inputs that minimises a function of the control inputs and predicted plant state trajectory over a given time horizon, subject to constraints. At each sampling instant, the optimisation 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 [22], there is increasing interest in its application to vehicle manœuvre problems ([4], [25], [28]), including spacecraft trajectory control ([6], [7], [8], [16]) and attitude control ([14], [20], [32]]). Essentially, the application of MPC builds upon the ideas of fuel and time-optimal trajectory planning by bringing the optimisation 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 a Model Predictive Control Framework software 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 HARVD ones.

2. The HARVD 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 on-board 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 (High Integrity Autonomous Rendezvous & Docking Control System), an ESA-funded activity implementing a GNC/Autonomous Mission Management/FDIR on-board software for rendezvous and docking/capture scenarios around Mars, Earth or potentially other planets ([1], [2] and [3]). HARVD, based on Radio Frequency (RF), camera and LIDAR 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 Avionics Test Benching and Dynamic Test Benching. Rendezvous and capture on an elliptic orbit has been specially addressed, demonstrating the technical feasibility and the potential propellant saving.
The HARVD step-wise development and verification approach is shown on Figure 1.

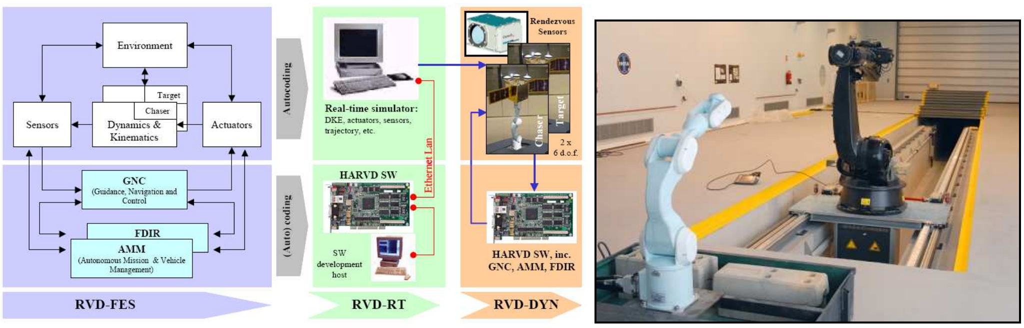


Figure 1: HARVD "Step wise" development and verification approach and GMV's PLATFORM dynamic test bench

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

- dSPACE simulator for real-time 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 Mars Sample Return 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 on-board system failures and collision risks detection and avoidance), for which the results obtained are very encouraging for the consolidation of higher Technology Readiness Levels. MAV 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 @45MHz, 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 Figure 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 MPC controllers based on linear time-invariant (LTI) or linear time-varying (LTV) models, as well as for automatic code-generation of embedded MPC 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. [34] 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. MPC controllers designed for LTI systems can be converted to explicit form [35] via the direct link between MPCTOOL and the Hybrid Toolbox for MATLAB [36]. Furthermore, a new Dual-Simplex solver has been developed to manage optimization problems expressed as a Linear Programming (LP) problem ([37]). 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 on-line;
- 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;
- Return the optimal cost of MPC, for comparing and choosing the best action among a set of MPC controllers;
- Allow the specification of convex PWA stage costs (such as absolute values) on inputs and outputs;
- Handle arbitrary linear constraints on combinations of inputs and outputs;
- Handle arbitrary linear time-varying models, weights, constraints, and horizons, by providing two Simulink blocks based on EML code, supporting both 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-MPC controller relies on the following rather general linear time-varying prediction model

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

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, P are (possibly time-varying and state-dependent) matrices.

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

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

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

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

$$\begin{aligned} \min \quad & \rho_1\epsilon_1 + \rho_2\epsilon_2 + \sum_{k=0}^{N(t)-1} \sum_{i=1}^{\ell} d_i(j) \\ \text{s.t.} \quad & d_i(j) \ge \pm z_i(j),\ d_i(j) \ge 0 \\ & + \text{MPC constraints} \end{aligned} \tag{3}$$

which has $(m + \ell) \cdot N(t) + 1$ optimization variables and, besides the non-negativity constraints Δu - $\Delta u_{min} \ge 0$, $\epsilon \ge 0$, $d_i(j) \ge 0$, $2\ell \cdot N(t)$ 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 Figure 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 real-time varying signals, to vary on-line prediction and control horizons, to limit a priori the maximum number of LP iterations.

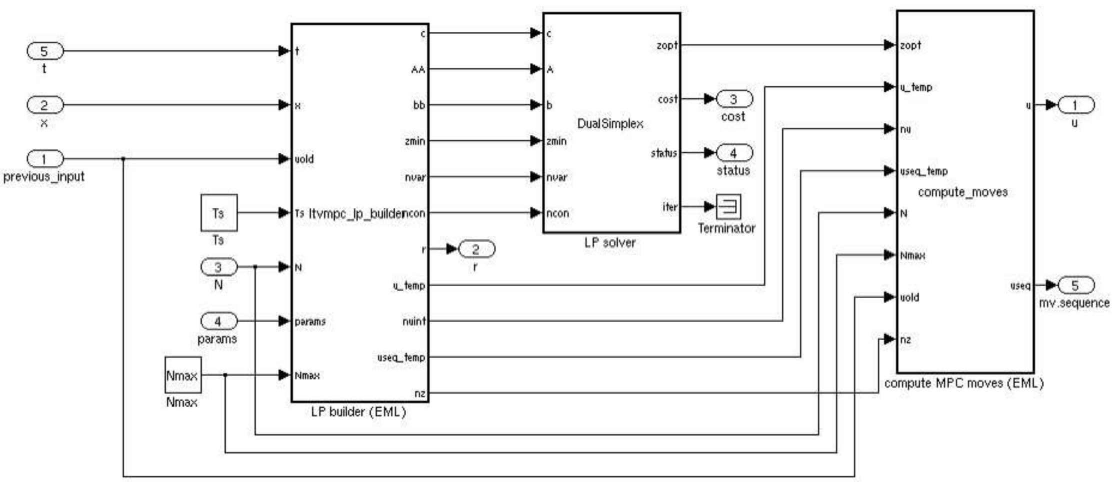


Figure 2: Simulink diagram underlying the LP-Based LTV-MPC block

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 ([5], [6], [10], [13], [12], [21], [29], [31], [33]). Whilst a non-linear 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 optimiser. 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. We therefore restrict our consideration to linear prediction models from which we can form the MPC optimisation problem as a convex quadratic or linear program. However, out of the models considered, only the Hill-Clohessy-Wiltshire equations [12] 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 is passive, ν_{tgt} can be calculated as a function of time using Kepler's equation [27], thus allowing a linear time-varying representation of the relative dynamics.

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 approximately 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 ms^{-1}. Target capture is then completed on a passive drift trajectory. The MPC system provides both guidance and control, and is not restricted to tracking pre-determined trajectories.

To achieve this objective using a single MPC controller would require a prediction horizon sufficient to predict a trajectory at least one orbital period into the future, a sufficiently complex prediction model to perform accurate trajectory propagations over long periods of time, and a sampling period short enough to allow target capture within a 20 cm 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 ([15], [17]), with an additional controller to perform a collision avoidance manœuvre (CAM) in case of a fault during the final moments of the rendezvous (Table 1).

Table 1: Rendezvous Phases

Phase	Requirement
Orbit Synchronisation Translational Guidance (OSTG)	To bring chaser from a distance of approximately 300 km into the same orbit as the target, with an in-track separation of between 5 km and 30 km on either side of the target
Impulsive Nominal Translational Guidance (INTG)	To perform passively safe impulsive transfers between a sequence of pre-defined holding points in the same orbit as the target until an in-track separation of 100 m is reached
Forced Terminal Translational Guidance (FTTG)	To track a straight-line trajectory from 100 m separation to 3 m separation from the target such that a subsequent free-drift trajectory captures the target with a 20 cm tolerance
Collision Avoidance Manœuvre (CAM)	To bring the chaser to a safe distance, further than 5 km from the target within 3 orbits, avoiding collision in the process

4.1.1 Orbit Synchronization Translational Guidance (OSTG)

The first phase, Orbit Synchronisation 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 km and 30 km on either side of the target, whilst minimising propellant consumption and manœuvre 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 manœuvres. For these reasons, the J_2-modified GVE prediction model of [6] 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 co-ordinate frame, whilst using the Gim-Alfriend [13] 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 [12], [33], 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 centred on the chaser.

4.1.2 Impulsive Nominal Translational Guidance (INTG)

The second phase, Impulsive Nominal Translational Guidance (INTG) must perform a sequence of passively safe impulsive transfers between a sequence of pre-defined 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 linearisation error on the Yamanaka-Ankersen [33] equations is no longer a problem, as long as a cylindrical relative coordinate system is used [21]. 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 linearised 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 (FTTG)

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 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 [33] equations are used for the trajectory prediction model.

In addition, to maintain target pointing, the MPC controller must also handle attitude regulation to an externally provided setpoint, using thrusters. A linearised quaternion-based prediction model [14], 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 ±180° in the quaternion representation [9]. Because the prediction matrices are re-built at each time step due to the LTV prediction model, the opportunity is taken to re-linearise the attitude dynamics about the current measured attitude at each time step.

4.1.3 Collision Avoidance Manœuvre (CAM)

The Collision Avoidance Manœuvre (CAM) must safely move the chaser away from the target, to a distance of 500m within three orbits without collision with the target. Essentially this objective is similar to that of INTG, except travelling 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 MPC subsystem design

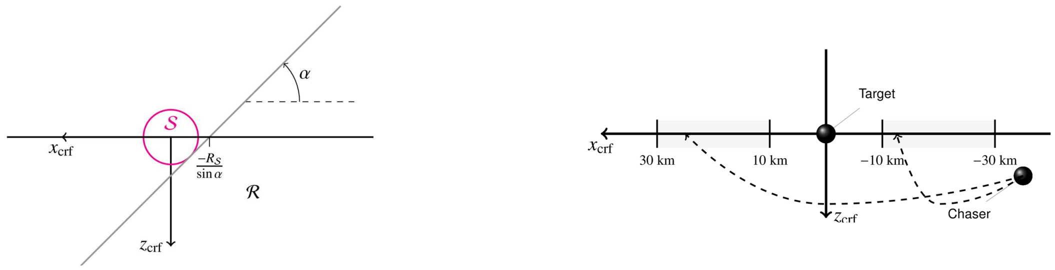
Each of the MPC controllers is designed independently, but with a common interface, and a common output function to convert the ΔV into finite-duration thrust pulses in the inertial frame. The core MPC function of each control subsystem is implemented using the blocks from the MPCTOOL, with the linear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

4.2.1 OSTG MPC controller

The OSTG MPC controller must bring the chaser into the same orbit as the target in a timely manner, whilst minimising propellant consumption. Rather than using the more commonly used quadratic cost function, to correctly encode the minimisation of total propellant consumption, the MPC controller must minimise the absolute sum of ΔV applied over the prediction horizon ([30]). 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 ([23][24][25]). Letting N be the prediction horizon, $\mathbf{u} = \left[u(t+T_S \mid t)^T, ..., u(t+(N-1)T_S \mid t)^T\right]^T$ and α be a parameter determining constraints that will be described later, the cost function is:

$$J_{\mathrm{OSTG}}(\alpha, \mathbf{u}, N) = N + \sum_{k=1}^{N-1} \left\| w_u u(t+kT_S \mid t) \right\|_1 \tag{4}$$

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 that 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 by [7], unforced drift trajectories emanating from each point in the prediction horizon are also constrained to ensure passive safety. Collision avoidance is a manifestly non-convex constraint, but it is approximated by a half-space constraint with angle relative to the in-track direction parameterised by α (Figure 3). The value of α then determines on which side of the target the terminal constraint places the end of the predicted trajectory.



(a) Safety constraint (b) Terminal constraint

Figure 3: OSTG safety constraint and terminal constraint

The optimisation is implemented using the "LP-based LTV-MPC 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 the current angle between the chaser and the z_{crf} axis, rounded to the nearest 45°, by solving $3N$ convex optimisations, 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 minimises the overall cost function. A sampling period T_S = 600s was chosen, along with a maximum prediction horizon $N_{\max}$ = 25.

4.2.2 INTG MPC controller

The INTG MPC controller must transfer the chaser between a sequence of invariant holding points on $\bar{V}$ (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, parameterised by the distance from the target of the next holding point.
The design is similar to that of the OSTG MPC controller in that a 1-norm cost function is used in conjunction with a variable horizon implemented by solving multiple convex optimisations. However, the cost function includes distance instead of time to reflect that fuel consumption is proportional to distance travelled rather than time when carrying out passively safe hopping trajectories. The holding points are scheduled by an external algorithm, and parameterised by distance x_{hp}. The cost function is

$$J_{\mathrm{INTG}}(\mathbf{u}, N) = \sum_{k=1}^{N-1} \left\| E_C \left(x_{\mathrm{crf}}(t+kT_S \mid t) - r(t+kT_S)\right) \right\|_1 + \left\| w_u u(t+kT_S \mid t) \right\|_1 \tag{5}$$

where $r(t+kT_S)=\left[\pm x_{hp}\left(1+e_{\mathrm{tgt}}\cos\nu_{\mathrm{tgt}}(t+kT_S)\right),0,0,0,0,0\right]^T$ dependent 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

$$E_C=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\end{bmatrix} \tag{6}$$

As for the OSTG MPC 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:

$$\left[-\operatorname{sgn}(x_{\mathrm{crf}}(t))\quad 0\quad 0\quad 0\quad 0\quad 0\right]\left(A_{\mathrm{orb}}\left(\nu_{\mathrm{tgt}}(t+kT_S)\right)\right)x_{\mathrm{crf}}(t+kT_S\mid t)\le 0 \tag{7}$$

The terminal set for the INTG MPC controller is defined as a box with side-length $2(e_{\mathrm{tgt}}+0.1)$, centred 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 ¼, ½ and ¾ orbits of free drift. A sampling period T_S = 300s and a maximum prediction horizon of $N_{\max}$ = 20 was chosen.

4.2.3 FTTG MPC controller

During the FTTG phase, trajectory and attitude tracking accuracy becomes more important than long-term fuel minimisation. 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-MPC Controller" block from MPCTOOL, with a sampling period of T_S = 3s, and a prediction horizon N = 15. Letting $x(j\mid t)$ be the combined position, velocity, attitude quaternion and angular velocity states, $r(j)$ the corresponding reference setpoint, and $u(j\mid t)$ the vector of thruster inputs, the cost function is:

$$J_{\mathrm{FTTG}}=\sum_{k=1}^{N-1}\left(x(t+kT_S\mid t)-r(t+kT_S)\right)^T Q\left(x(t+kT_S\mid t)-r(t+kT_S)\right)+\Delta u(t+kT_S\mid t)^T R\Delta u(t+kT_S\mid t) \tag{8}$$

Changes in input (Δu) are penalised instead of the absolute input value to enable offset-free tracking of forced-equilibrium setpoints ([18]). Positivity and saturation constraints are applied to inputs. The reference trajectory $r(j)$ and cost function weightings $Q\ge 0$ and $R\ge 0$ 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 CAM MPC controller

The CAM MPC controller is based on a modified version of the INTG MPC 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 MPC 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 ORCSAT Avionic 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 control 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 co-processor has to be considered as baseline (distributed architecture) taking into account available space qualified processor computational performances. This solution allows a distribution of the complete on-board software on the two processors leaving to the co-processor 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.

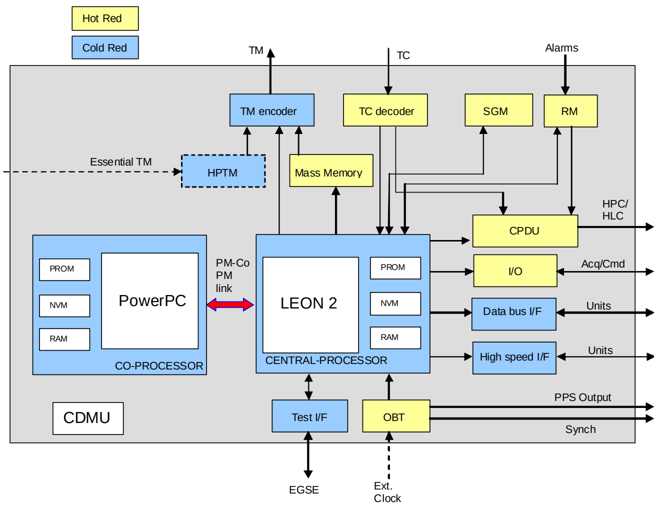


Figure 4: CDMU with distributed architecture

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 @ 100 MHz: taking into account also the HARVD experience, it has been considered adequate for the central processor of the CDMU.
Regarding the co-processor, 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 @ 733MHz which has been used to design space-qualified boards like the Maxwell SCS750.

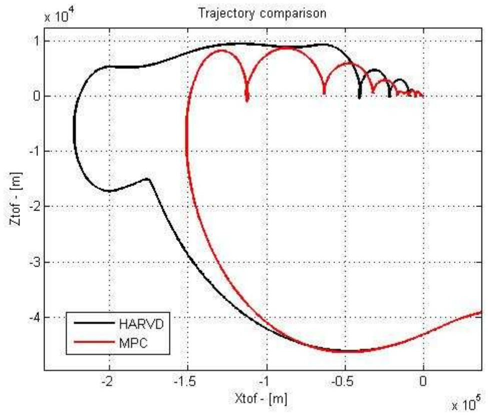
Table 2: MPC profiling

Mode	*Workstation* *Max time [s]*	*Scaling to LEON2 FT* *Time [s]*	*Scaling to PowerPC 750FX* *Time [s]*
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

The profiling results are summarized in Table 2. It can be seen that the LEON2 FT cannot be selected as co-processor, since the FTTG would take more than 4 seconds for the computation of the control action against a theoretical control step of 3 seconds. 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 co-processor.

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 results is the propellant save, which in this case is about 35 kg.



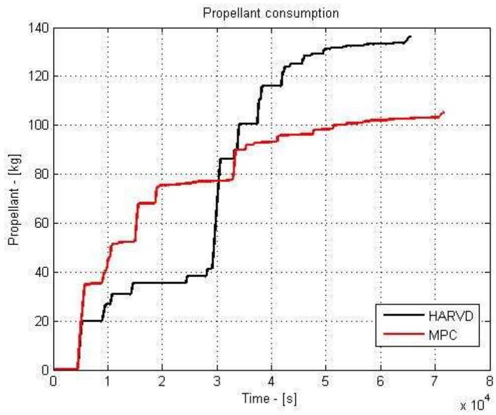
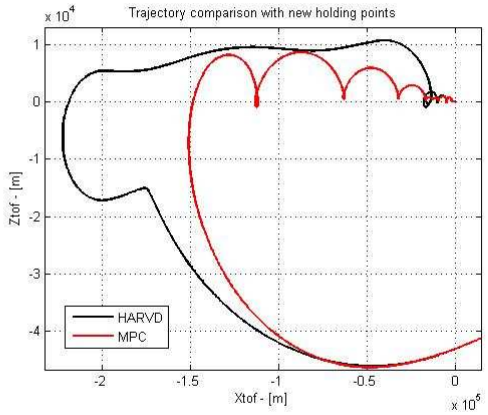


Figure 5: HARVD vs MPC performance comparisons

The main differences can be found analysing 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: therefore, 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.



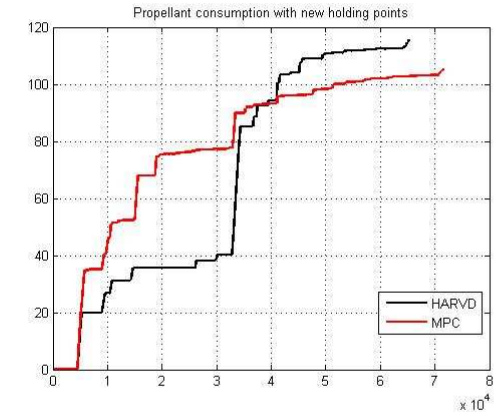


Figure 6: HARVD vs MPC performance comparisons with new holding points definition

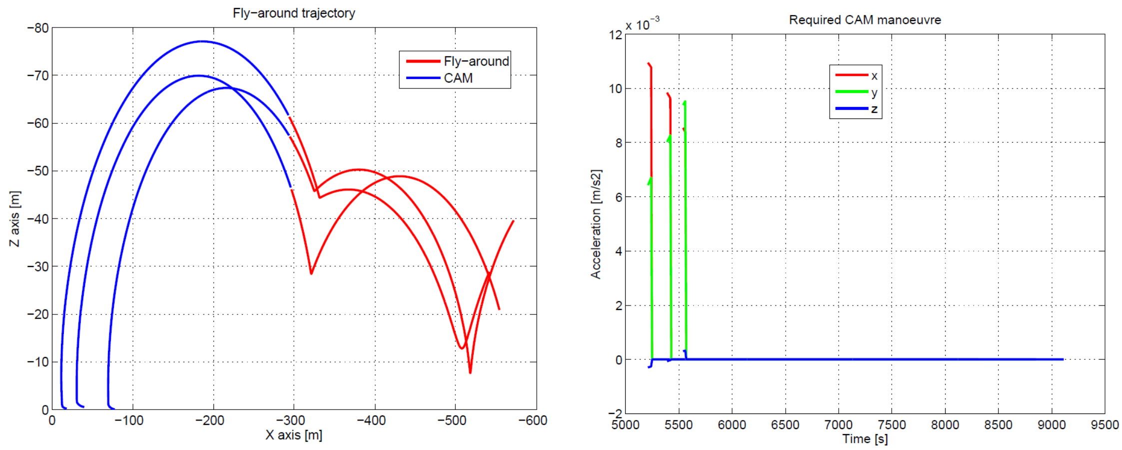


Figure 7: CAM simulation results

This aspect is quite important: the on-line 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 (10m) with a single manoeuvre.

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, Figure 9 shows the aggregated capture accuracy results of 400 cases.

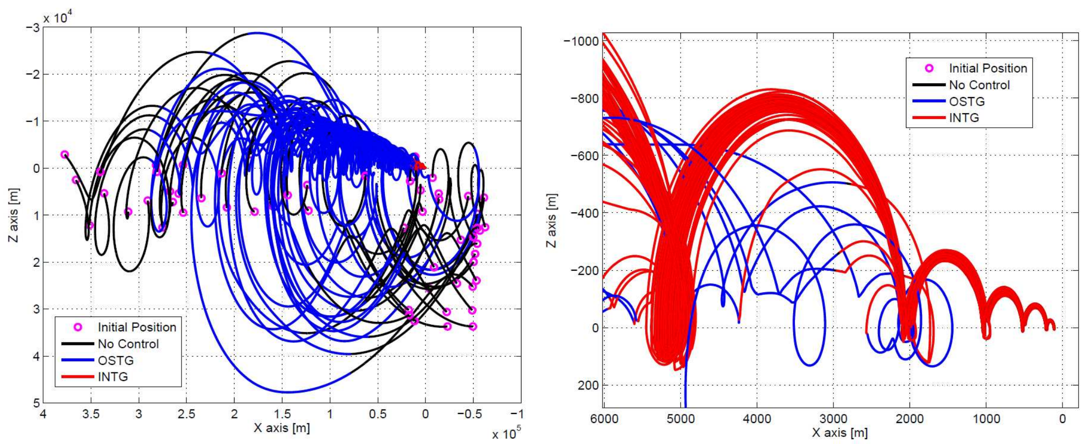


Figure 8: Overall trajectories during OSTG and INTG of 50 cases of the Monte-Carlo campaign

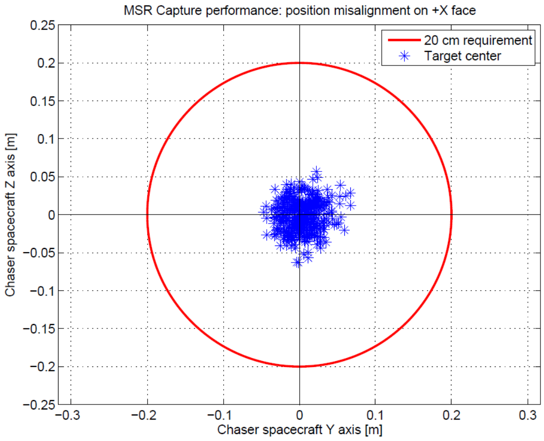


Figure 9: Capture performance of 400 test cases of the Monte-Carlo campaign

7. Conclusions

Optimization-based control techniques like Model Predictive Control 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 Mars Sample Return 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, on-line 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 real-time performance of the developed solution on a flight-representative avionic. In the end, the GMV Dynamic Test Bench will be enhanced with the MPC based control and the selected avionics for the final dynamic test campaign.

References

[1] P. Colmenarejo et al., “HARVD Development, Verification and Validation Approach (from Traditional GNC Design/V&V Framework Simulator to Real-Time Dynamic Testing)”, 7th International ESA Conference on Guidance, Navigation & Control Systems, 2-5 June 2008, Tralee, County Kerry, Ireland

[2] V. Barrena et al., “Integrated Development, Verification and Validation Approach for Space Systems Using Autocoding Techniques”, Data System in Aerospace Conference (DASIA 2008), 27-30 May 2008, Palma Majorca, Spain

[3] Strippoli, L et al., “High Integrity Control System For Generic Autonomous RVD”, 61st International Astronautical Congress, September 19–25, 2010, Prague, CZ

[4] Almeida, F.A., 2008. Waypoint navigation using constrained infinite horizon model predictive control. In: Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit. Honolulu, Hawaii.

[5] Battin, R.H., 1999. An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition. AIAA Education Series. American Institute of Aeronautics and Astronautics.

[6] Breger, L. and How, J.P., 2007. Gauss’s variational equation-based dynamics and control for formation flying spacecraft. Journal of Guidance, Control, and Dynamics, 30(2):437–448.

[7] Breger, L. and How, J.P., 2008. Safe trajectories for autonomous rendezvous of spacecraft. Journal of Guidance Control and Dynamics, 31(5):1478–1489.

[8] Bodin, P., Noteborn, R., Larsson, R., and Chasset, C., 2011. System test results from the GNC experiments on the PRISMA in-orbit test bed. Acta Astronautica, 68(7–8):862–872.

[9] Bach, R. and Paielli, R., 1993. Linearization of attitude-control error dynamics. IEEE Transactions on Automatic Control, 38(10):1521–1525.

[10] Carter, T.E., 1998. State transition matrices for terminal rendezvous studies: brief survey and new example. Journal of Guidance Control and Dynamics, 21(1):148–155.

[11] Camacho, E.F. and Bordons, C., 2004. Model predictive control. Springer-Verlag, London.

[12] Clohessy, W.H. and Wiltshire, R.S., 1960. Terminal guidance system for satellite rendezvous. Journal of the Aerospace Sciences, 27(9):653–658.

[13] Gim, D. and Alfriend, K.T., 2003. State transition matrix of relative motion for the perturbed noncircular reference orbit. Journal of Guidance, Control, and Dynamics, 26(6):956–971.

[14] Hegrenæs, O., Gravdahl, J.T., and Tondel, P., 2005. Spacecraft attitude control using explicit model predictive control. Automatica, 41(12):2107–2114.

[15] Kerambrun, S., Despré, N., Frapard, B., Hyounet, P., Polle, B., Ganet, M., Silva, N., Cropp, A., and Philippe, C., 2008. Autonomous rendezvous system: The HARVD solution. In: Proceedings of the 7th International ESA Conference on Guidance, Navigation & Control Systems. Tralee, Ireland.

[16] Larsson, R., Berge, S., Bodin, P., and Jönsson, U., 2006. Fuel efficient relative orbit control strategies for formation flying and rendezvous within PRISMA. In: Proceedings of the 29th Annual AAS Guidance and Control Conference.

[17] Le Peuvédic, C., Colmenarejo, P., and Guiotto, A., 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.

[18] Maciejowski, J.M., 1998. The implicit daisy-chaining property of constrained predictive control. Applied Mathematics and Computer Science, 8(4):101–117.

[19] Maciejowski, J.M., 2002. Predictive Control with Constraints. Pearson Education.

[20] Manikonda, V., Arambel, P.O., Gopinathan, M., Mehra, R.K., and Hadaegh, F. Y., 1999. A model predictive control-based approach for spacecraft formation keeping and attitude control. In: Proceedings of the American Control Conference, volume 6, 4258–4262. San Diego, CA.

[21] Melton, R.G., 2000. Time-explicit representation of relative motion between elliptical orbits. Journal of Guidance, Control, and Dynamics, 23(4):604–610.

[22] Qin, S.J. and Badgwell, T.A., 2003. A survey of industrial model predictive control technology. Control Engineering Practice, 11(7):733–764.

[23] Richards, A. and How, J., 2003. Performance evaluation of rendezvous using model predictive control. In: AIAA Guidance, Navigation and Control Conference and Exhibit. Austin, Texas.

[24] Richards, A.G. and How, J.P., 2003. Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility. In: Proceedings of the 2003 American Control Conference, volume 5, 4034–4040. Denver, Colorado.

[25] Richards, A. and How, J.P., 2006. Robust variable horizon model predictive control for vehicle maneuvering. International Journal of Robust and Nonlinear Control, 16(7):333–351.

[26] Rawlings, J. B. and Mayne, D.Q., 2009. Model predictive control: Theory and design. Nob Hill Publishing.

[27] Sidi, M. J., 1997. Spacecraft dynamics and control: A practical engineering approach. Cambridge University Press.

[28] Shim, D.H., Kim, H.J., and Sastry, S., 2003. Decentralized nonlinear model predictive control of multiple flying robots. In: Proceedings of the 42nd IEEE Conference on Decision and Control, volume 4, 3621–3626. Maui, Hawaii USA.

[29] Schaub, H., Vadali, S. R., Junkins, J.L., and Alfriend, K.T., 2000. Spacecraft formation flying control using mean orbital elements. Journal of the Astronautical Sciences, 48:69–87.

[30] Tillerson, M., Inalhan, G., and How, J. P., 2002. Co-ordination and control of distributed spacecraft systems using convex optimization techniques. International Journal of Robust and Nonlinear Control, 12(2-3):207–242.

[31] Tschauner, J., 1967. Elliptical orbit rendezvous. AIAA Journal, 5(6):1110–1113.

[32] Wood, M. and Chen, W.H., 2008. Model predictive control of low Earth orbiting satellites using magnetic actuation. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 222(6):619–631.

[33] Yamanaka, K. and Ankersen, F., 2002. New state transition matrix for relative motion on an arbitrary elliptical orbit. Journal of Guidance Control and Dynamics, 25(1):60–66.

[34] A. Bemporad, M. Morari, and N. L. Ricker, “Model Predictive Control ToolboxTM 3 – User’s Guide”, The Mathworks, Inc., 2009, http://www.mathworks.com/access/helpdesk/help/ toolbox/mpc/.

[35] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica*, vol. 38, no. 1, pp. 3–20, 2002

[36] A. Bemporad, “Hybrid Toolbox v1.2.2 – User’s Guide”, Dec, 2009, http://www.ing.unitn.it/~bemporad/ hybrid/toolbox.

[37] Bertsimas, D. and Tsitsiklis, J. N. Introduction to Linear Optimization. Athena Scientific, 1997.