POLITECNICO MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

Executive Summary of the Thesis

# Model Discovery Predictive Control for Satellite Relative Range Bounding

Laurea Magistrale in Space Engineering - Ingegneria Spaziale

**Author:** Hariharan Venkatesh Vitaladevuni

**Advisor:** Prof. Mauro Massari

**Academic year:** 2021-2022

## 1. Introduction

The new-age space exploration demands higher autonomy in the relative orbital control of small satellites. Applications range from low-Earth orbit formations, orbital inspections, tandem satellites for deep space missions, etc. The fundamental challenges in increasing the autonomy of the control systems are communication latency and dynamics modelling inaccuracy. The controller must adapt to spatio-temporal dependent dynamics, deviations from idealised models and perform impromptu manoeuvres.
The thesis employs a novel method of onboard sensor measurement based local dynamics identification. The identified model is utilised in the model predictive controller (MPC) subject to certain physical and configuration constraints. The combined model discovery and predictive controller (MDPC) performance is simulated at various orbital environments and control system parameters to quantise a baseline.

## 2. Problem Description

The specific case utilised for investigation is a tandem spacecraft formation, consisting of target and chaser spacecraft. The control system must keep the chaser within a bounded relative range of the target.

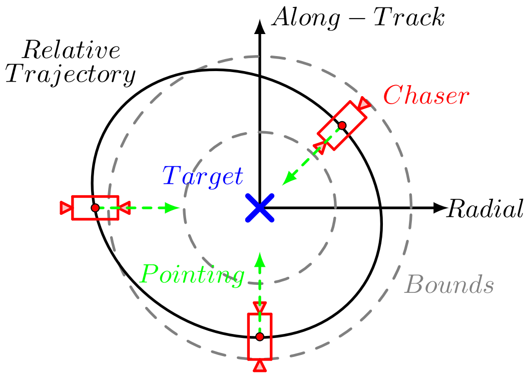

Figure 1: Range bounds and chaser pointing.

Other mission constraints are the persistent pointing of the chaser towards the target and the presence of a pair of opposite-facing thrusters on the chaser, aligned with its pointing. The various possible chaser positions and its pointing is represented in Fig. 1, as an example.

## 3. Proposed Solution

The MDPC framework is capable of discovering the local dynamics using SINDy algorithm, based on onboard radar ranging (PRISMA) [8] sensor measurement, and use it for the MPC. The SINDy algorithm is faster and more efficient than alternatives based on neural networks [5]. SINDy algorithm offers process transparency and intuition, reflecting the actual dynamics, compared to a "black-box" neural network [5].

## 3.1. Mathematical Models

The relative motion reference frame is oriented along radial-out ($\hat{i}_x$), along-track ($\hat{i}_y$) and orbit-normal ($\hat{i}_z$) directions of the target spacecraft. The set of coupled differential equations can be represented as a matrix as follows:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} \left(\omega^2 + 2k\omega^{\frac{3}{2}}\right) x + \dot{\omega} y + 2\omega \dot{y} \\ -\dot{\omega} x + \left(\omega^2 - k\omega^{\frac{3}{2}}\right) y - 2\omega \dot{x} \\ -k\omega^{\frac{3}{2}} z \end{bmatrix} + \boldsymbol{a}_d$$

$$k = \frac{\mu}{h^{\frac{3}{2}}} = const; \; \frac{\mu}{r_t^3} = k\omega^{\frac{3}{2}} \tag{1}$$

where $\boldsymbol{a}_d$ is the external acceleration vector, formulated as: $\boldsymbol{a}_d = \boldsymbol{a}_f + \boldsymbol{a}_p$. The $\boldsymbol{a}_f$ and $\boldsymbol{a}_p$ represent the acceleration vectors due to chaser actuation and orbital perturbation respectively. The validity of these equations hold under the assumption that the separation between the spacecrafts is much smaller than the orbital radius, $\rho \ll r$. The complete derivation of these equations can be found in Ref. [7] however, the reference frame basis vectors have been modified for convenience.

A state-space representation can be formulated for eq. 1 by assuming a state-vector and an control input vector as:

$$\mathbf{x} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T; \; \mathbf{u} = \boldsymbol{a}_f \tag{2}$$

The system equation can be written as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_I \mathbf{u} + \boldsymbol{\Upsilon} \tag{3}$$

where $\boldsymbol{\Upsilon}$ denotes the vector of corresponding contributions due to $\boldsymbol{a}_p$. The system matrix $\mathbf{A}$, input matrix $\mathbf{B}_I$ and $\boldsymbol{\Upsilon}$ can be expanded as:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \begin{pmatrix} \omega^2 \\ +2k\omega^{\frac{3}{2}} \end{pmatrix} & \dot{\omega} & 0 & 0 & 2\omega & 0 \\ -\dot{\omega} & \begin{pmatrix} \omega^2 \\ -k\omega^{\frac{3}{2}} \end{pmatrix} & 0 & -2\omega & 0 & 0 \\ 0 & 0 & -k\omega^{\frac{3}{2}} & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

$$\mathbf{B}_I = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{5}$$

$$\boldsymbol{\Upsilon} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & a_{px} & a_{py} & a_{pz} \end{bmatrix}^T \tag{6}$$

The eq. 3 can be rearranged for convenience:

$$\dot{\mathbf{x}}^T = \begin{bmatrix} \mathbf{x}^T & \mathbf{a}_f^T \end{bmatrix} \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} + \boldsymbol{\Upsilon}^T \tag{7}$$

The eq. 7 is referred to as the augmented state equation and it forms the basis of the formulation of the model discovery framework.

## 3.2. Model Discovery: SINDy

The SINDy framework utilises numerical regression problem formulations to obtain the discovered dynamics models. This is performed by utilising the regression equation formulation, borrowed from Ref. [1], shown below:

$$\dot{\mathbf{X}} = \boldsymbol{\Theta}(\mathbf{X})\,\boldsymbol{\Xi} \tag{8}$$

where $\dot{\mathbf{X}}$ is the targets matrix, $\boldsymbol{\Theta}$ is the features matrix and $\boldsymbol{\Xi}$ is the coefficients matrix.

The matrices $\dot{\mathbf{X}}$ and $\boldsymbol{\Theta}$ are populated by sensor measurements in time. To adopt our mathematical models of relative motion into a SINDy regression problem, we can compare eq. 7 and eq. 8, by assuming $\boldsymbol{\Upsilon} = 0$, to obtain:

$$\mathbf{X} \equiv \begin{bmatrix} \mathbf{x}^T & \mathbf{a}_f^T \end{bmatrix} \tag{9}$$

$$\boldsymbol{\Theta}(\mathbf{X}) \equiv \begin{bmatrix} \mathbf{x}^T & \mathbf{a}_f^T \end{bmatrix} \Rightarrow \boldsymbol{\Theta}(\mathbf{X}) = \mathbf{X} \tag{10}$$

$$\boldsymbol{\Xi} \equiv \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} \tag{11}$$

To discover the system model, the coefficient matrix $\boldsymbol{\Xi}$ must be found through regression of the known or estimated matrices $\dot{\mathbf{X}}$ and $\boldsymbol{\Theta}$.

### 3.2.1 Regression Algorithm

The chosen regression optimiser is called the Sparse Relaxed Regularised Regression (SR3) optimiser [1]. The SR3 optimiser offers additional flexibility to impose constraints and employ coefficient specific thresholding during regression. These tools can be exploited to "inform" the regression optimiser about the feasibility and the preexisting knowledge about the dynamics. The SR3 optimiser can be mathematically formulated as:

$$\min_{\boldsymbol{\Xi}} \frac{\parallel \dot{\mathbf{X}} - \boldsymbol{\Theta}(\mathbf{X})\,\boldsymbol{\Xi} \parallel}{2} + \lambda ||\boldsymbol{\Xi}||_1 + \frac{\parallel \boldsymbol{\Xi} - \mathbf{W} \parallel}{2\nu}$$

where $\lambda$ is the a hyper-parameter associated with tuning the thresholding during each regression iteration. $\mathbf{W}$ represents a coefficient matrix generated through relaxation of $\boldsymbol{\Xi}$. Relaxation is the elimination of coefficients which correspond to a weak correlation, resulting in a sparse model [1]. The regression objective function employs the $L1$ norm for promoting sparse coefficients during every iteration [3].

### 3.2.2 Constraints and Thresholding

Several constraints can be imposed on the $\boldsymbol{\Xi}$ matrix, from pre-existing knowledge. By substituting eq. 5 and eq. 4 into eq. 11, we can formulate the analytical expression of $\boldsymbol{\Xi}$ as:

$$\boldsymbol{\Xi} = \begin{bmatrix} 0 & 0 & 0 & \begin{pmatrix} \omega^2 \\ +2k\omega^{\frac{3}{2}} \end{pmatrix} & -\dot{\omega} & 0 \\ 0 & 0 & 0 & \dot{\omega} & \begin{pmatrix} \omega^2 \\ -k\omega^{\frac{3}{2}} \end{pmatrix} & 0 \\ 0 & 0 & 0 & 0 & 0 & -k\omega^{\frac{3}{2}} \\ 1 & 0 & 0 & 0 & -2\omega & 0 \\ 0 & 1 & 0 & 2\omega & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

This formulation can be exploited to impose certain constraints on the elements $\boldsymbol{\xi}_{i,j}$ in $\boldsymbol{\Xi}$ as:

$$\begin{aligned} \boldsymbol{\xi}_{1,4} - \boldsymbol{\xi}_{2,5} + 3\boldsymbol{\xi}_{3,6} &= 0 \\ \boldsymbol{\xi}_{1,5} + \boldsymbol{\xi}_{2,4} &= 0 \\ \boldsymbol{\xi}_{4,5} + \boldsymbol{\xi}_{5,4} &= 0 \end{aligned} \tag{13}$$

$$\boldsymbol{\xi}_{4,1} = \boldsymbol{\xi}_{5,2} = \boldsymbol{\xi}_{6,3} = \boldsymbol{\xi}_{7,4} = \boldsymbol{\xi}_{8,5} = \boldsymbol{\xi}_{9,6} = 1$$

The constraints "inform" the regression optimiser about the mandatory mutual relationships. The feasibility of the discovered model can be determined by the values $\boldsymbol{\xi}$. To eliminate impractical values of $\boldsymbol{\xi}$, coefficient specific thresholding can be formulated as follows:

$$\boldsymbol{\eta} = \mathcal{T} \begin{bmatrix} \infty & \infty & \infty & \boldsymbol{\xi}_{1,4} & \boldsymbol{\xi}_{1,5} & \infty \\ \infty & \infty & \infty & \boldsymbol{\xi}_{2,4} & \boldsymbol{\xi}_{2,5} & \infty \\ \infty & \infty & \infty & \infty & \infty & \boldsymbol{\xi}_{3,6} \\ 0 & \infty & \infty & \infty & \boldsymbol{\xi}_{4,5} & \infty \\ \infty & 0 & \infty & \boldsymbol{\xi}_{5,4} & \infty & \infty \\ \infty & \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 \end{bmatrix} \tag{14}$$

where $\mathcal{T} = 10^{-2}$. The matrix $\boldsymbol{\eta}$ provides thresholds for its corresponding elements mapped to matrix $\boldsymbol{\Xi}$. During the regression iterations, any coefficients in $\boldsymbol{\Xi}$ which are smaller in absolute value compared to their mapped threshold in $\boldsymbol{\eta}$ will be eliminated due to SR3 relaxation.

### 3.2.3 Initial Guess

To obtain accurate and fast regression convergence, an initial guess value of $\boldsymbol{\Xi}$ is provided. The equations used to estimate the variables in $\boldsymbol{\Xi}$ namely, $k\omega^{\frac{3}{2}}, \omega$ and $\dot{\omega}$, is are follows:

$$\widehat{k\omega^{\frac{3}{2}}} \approx \frac{\widehat{a_{fz}} - \hat{\ddot{z}}}{\hat{z}};\ \hat{\omega} \approx \hat{h}\left(\frac{\widehat{a_{fz}} - \hat{\ddot{z}}}{\mu\hat{z}}\right)^{\frac{2}{3}} \tag{15}$$

$$\widehat{\dot{\omega}} \approx -3\frac{\widehat{a_{fz}} - \hat{\ddot{z}}}{\hat{z}} \tag{16}$$

where each term is the estimated or measured value from the sensor data or numerical differentiation of the sensor measurements. The accented variables represent the measured or estimated versions of the analytical variables. An accurate estimate of $\dot{\omega}$ is not required because this is just an initial value for the regression. A final constraint is imposed to ensure that the regression problem has no free DoFs, by imposing, $|\boldsymbol{\xi}_{5,4} - 2\hat{\omega}| = \epsilon$, where $\epsilon$ is the machine precision.

### 3.2.4 Model Discovery Robustness

The discovered model is linearised in time and spatial coordinates due to performing the model discovery based on a limited time-interval $T_{lin}$ of sensor measurements. Regression has problems with repeatability of results, resulting in divergence from the "ground-truth" when propagated in time beyond $T_{lin}$. A Monte-Carlo Multiple Shooting based method is implemented to remedy this [4]. The measurements within $T_{lin}$ are divided into $n$ sub-sets by sub-sampling with a lower sampling frequency. These sub-sets simultaneously undergo SINDy regression to discover $n$ simultaneous models $\boldsymbol{\Xi}_n$ of the same underlying dynamics, as in eq. 17.

$$\dot{\mathbf{X}}_n = \boldsymbol{\Theta}_n\left(\mathbf{X}_n\right)\boldsymbol{\Xi}_n \tag{17}$$

Then each model is propagated within the linearisation interval $T_{lin}$ to compute the predicted relative trajectory, as shown in Fig. 2. Every model propagation is compared with the "ground-truth" measurements from the original data-set to compute the Root Normalised Mean Squared Error $E_{RNMSE,\ n}$. An effective discovered model is obtained by computing the weighted average of all the $\boldsymbol{\Xi}_n$, see eq. 18. This process is called model assembly or bagging.

$$\boldsymbol{\Xi}_{eff} = \frac{\sum_{i=1}^{n}\left(\frac{1}{E_{RNMSE,\ i}}\right)\boldsymbol{\Xi}_i}{\sum_{i=1}^{n}\left(\frac{1}{E_{RNMSE,\ i}}\right)} \tag{18}$$

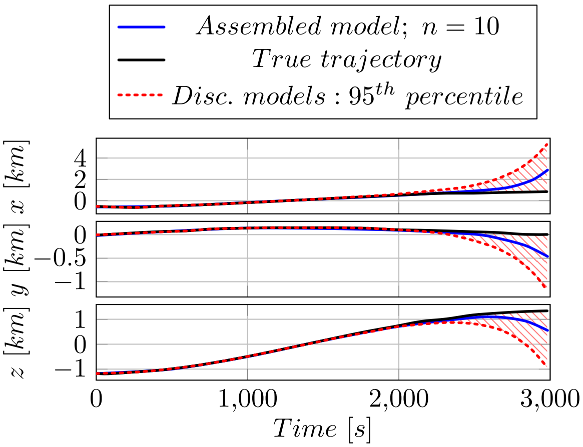

Figure 2: Simultaneous models propagation.

This results in a more robust and stable model, as evidenced in Fig. 2. To test the model prediction performance, various simulations in different orbital environments were performed for $T_{sim} = 300s$. The resulting prediction errors are within acceptable margins even for highly eccentric orbit with $e = 0.6$, as shown in Fig. 3.

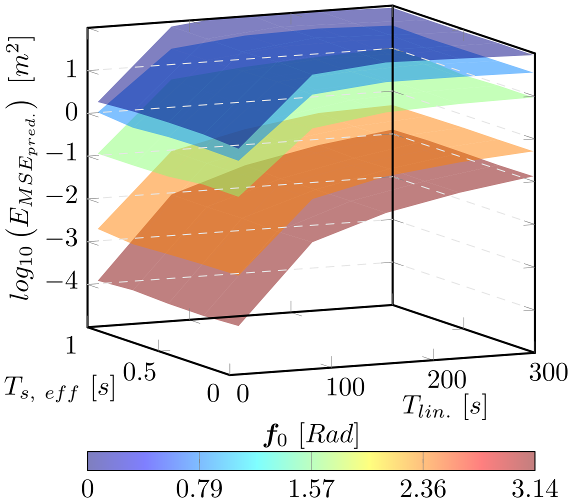

Figure 3: Model prediction test, $e = 0.6$.

where $T_{s,\ eff}$ is the effective sampling interval of the measurement sub-sets. The initial conditions (Keplerian elements) for the test are:

$$K_t = [7106.14\ km, 0.6, 98.3°, 0°, 270°, f_0]$$

$$K_c = [7106.14\ km, 0.601, 98.31°, 0°, 270°, f_0]$$

### 3.3. MPC Design

The MPC is designed to implement an optimised control sequence which minimises a cost function, subjected to constraints, within a limited time-horizon. The prediction horizon of the MPC is limited by the model discovery model accuracy, hence it is set as $T_{PH} = T_{lin}$. A parametric thrust profile is used to speed-up the optimisation and introduce a bias in the MPC. An exponential decay function is used as the thrust profile, see eq. 19, which ensures that maximal control effort is prioritised at the earliest, improving the MPC "response time".

$$u(t) = ae^{-bt};\ \{0 \leq t \leq T_{CH}\} \quad (19)$$

where $T_{CH}$ is the control horizon, $a$ is the initial thrust and $b$ is the thrust decay factor. The limits of the chaser thruster dictate the value of $a \in [-1, 1]\ N$. Negative thrust indicates a thrust opposite to the chaser pointing. To satisfy the objectives of a global minima,the cost function is modelled as a convex function, shown below:

$$C_{net}(a, b) = C_R + G_{rel}C_U \quad (20)$$

where $C_{net}$ is the net cost as a function of thrust profile parameters $(a, b)$. $C_R$ and $C_U$ denote the range bounds violation cost and the control effort cost, respectively. A relative scaling gain $G_{rel}$ is introduced to tune the MPC performance. The control effort cost is:

$$C_U(t) = u(t)^2 \quad (21)$$

The net control effort cost is found by using eq. 19 and integrating then normalising eq. 21:

$$C_U = \frac{\int_{t=0}^{T_{CH}} u(t)^2 dt}{T_{CH}} = \frac{a^2\left(1 - e^{-2bT_{CH}}\right)}{2bT_{CH}} \quad (22)$$

The range bounds violation cost function must be continuous and have a positive second derivative. This is necessary for convexity of the function. The cost of range bounds violation must be null within the bounds and linearly increase beyond the bounds. The range bounds violation cost function is inspired from ELU function [2]:

$$C_R(\rho) = \frac{\rho_{min} - \rho}{\frac{1}{d_{min}} + e^{(\rho - \rho_{min})}} + \frac{\rho - \rho_{max}}{\frac{1}{d_{max}} + e^{(\rho_{max} - \rho)}}$$

where $\rho$ is the relative range and $\rho_{max}, \rho_{min}$ represent the maximum and minimum range bounds. The $d_{max}$ and $d_{min}$ are the slope scaling variables for controlling the slope of the function beyond the bounds $\rho_{max}, \rho_{min}$. An example of this function is shown in Fig. 4:

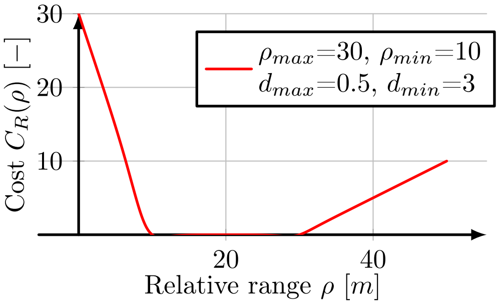

Figure 4: Range violation cost function.

The net range violation cost is the sum of all range violation costs normalised in $T_{PH}$:

$$C_R = \frac{\sum_{k=0}^{N_{PH}} C_R(\rho_k)}{T_{PH}} \quad (23)$$

where $k$ is the sampling instance of the predicted trajectory propagated from the discovered model using RK4 method. $\rho_k$ is the $k^{th}$ predicted range. $N_{PH}$ is the total sampling instances in the prediction horizon $T_{PH}$. To guarantee the cost function convexity, $b > 0$ for a continuous positive second derivative of the control cost function $C_U$. Thrust profile resembles an impulse for high values of $b$. Hence, the range of $b$ is $b \in (0, 5]$. An example of the cost function optimisation domain is shown in Fig. 5:

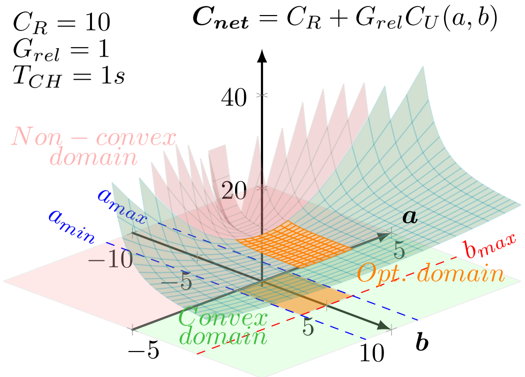

Figure 5: MPC cost function example.

The MPC objective function now becomes:

$$\min_{\boldsymbol{a,b}} C_{net}(a, b); \; a \in [-1, \; 1]; \; b \in (0, \; 5] \quad (24)$$

The MPC objective function is optimised using the SQP algorithm [6], which guarantees constraint non-violation at every iteration . Even if the optimisation is interrupted due to time-out of the controller, it will provide a sub-optimal control sequence respecting the constraints.

### 3.3.1 Control System: MDPC

Until the current discovered model expires after $T_{lin}$ interval since its discovery, the same model is used by the MPC. At model expiry, a new model discovery cycle updates the model, using previous $T_{lin}$ measurements, see Fig. 6.

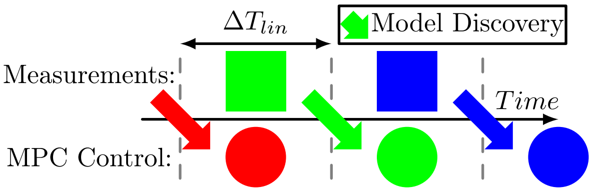

Figure 6: MDPC cycle schematic.

## 4. Control System Performance

The MDPC is simulated for various orbital environments ranging from beyond GEO to LEO orbits. The actual motion is simulated using the two-body problem and $J_2$ perturbations. The MPC is disengaged in first $T_{lin}$ of the simulation for first model discovery (blue line, Fig. 8). The worst case for the MDPC is simulated i.e. the highly nonlinear orbital environments with maximum perturbation e.g. highly eccentric LEO. The simulation notations are: $N_H : N_{PH}/N_{CH}$ is the total sampling instances within $T_{PH}$ and $T_{CH}$, and $\rho_{lim} : \rho_{min}/\rho_{max}$. The orbit under investigation is physically impossible due to the perigee inside the Earth. It is still simulated to test the MDPC performance limits.

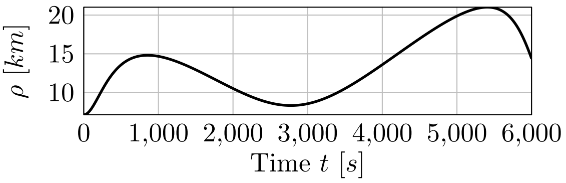

Figure 7: Simulation: MPC disengaged.

In Fig. 7, an uncontrolled trajectory baseline is established for comparison. The simulation corresponds to a complete orbital period and starts at perigee, which is the most challenging orbital environment for MDPC. The drift in the relative range in Fig. 7 is due to the $J_2$ perturbations. The MDPC must compensate for this drift and the periodic relative motion. The MPC is very effective at keeping the chaser within bounds, see Fig. 8. However, due to the challenging environment, the thruster is saturated at the end.

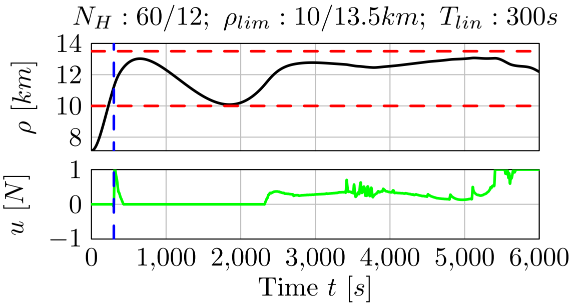

Figure 8: Simulation: MPC engaged.

The effect of changing the MPC horizons, which are doubled compared to Fig. 8, are visible in Fig. 9. There are more range bound violations, however, the control input and the trajectory are smoother. The bounds violation is due to the "dampening" effect caused by normalisation of the cost functions, as per eq. 22 and eq. 23.

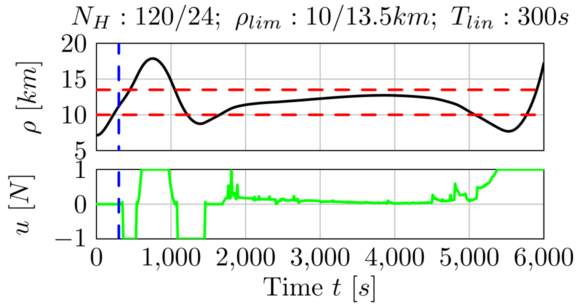

Figure 9: Simulation: MPC horizons change.

In Fig. 9, when the MDPC predicts the future trajectory at $t = 300s$ with a longer horizon, it predicts the chaser moving out of bounds and also returning back into the bounds due to the corrective control effort. This reduces the normalised cost, hence behaving as control effort "dampener". To make the MPC more responsive, an alternative cost function can be formulated without normalisation. Thruster saturation can be evidenced even in Fig. 9, because the orbital environment under investigation is a worst-case scenario and physically impossible. Real-life applications will encounter easier orbital environments for the MDPC.

## 5. Limitations

The main limitations of the MDPC are:

1. Constant system matrix $\mathbf{A}$ within $T_{lin}$, instead of time-variant system matrix.
2. For manoeuvres altering the orbit substantially, MDPC needs the data of specific orbital momentum more than once per orbit.
3. The controller prediction horizon is implicitly linked to the measurement interval ($T_{lin}$) utilised in model discovery, in the context of the implemented solution.
4. Cannot account for substantial deviation from two-body problem e.g. asteroids.
5. Cannot account and compensate for actuator saturation in the control optimisation.

## 6. Conclusions

The important conclusions from the thesis are:

1. MDPC is effective in highly eccentric orbits.
2. Valid for initial separations around 100$km$.
3. Relative range prediction is accurate and stable even under large perturbations and time-varying nonlinear dynamics.
4. MDPC is computationally efficient, robust and simple.
5. Utilises a single actuator for range control.
6. More autonomy, without ground support.
7. Faster than neural networks based methods.

## References

[1] Kathleen Champion, Peng Zheng, Aleksandr Y Aravkin, Steven L Brunton, and J Nathan Kutz. A unified sparse optimization framework to learn parsimonious physics-informed models from data. *IEEE Access*, 8:169259–169271, 2020.

[2] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus), 2015.

[3] Alexandre Cortiella, Kwang-Chun Park, and Alireza Doostan. Sparse identification of nonlinear dynamical systems via reweighted l1-regularized least squares. *Computer Methods in Applied Mechanics and Engineering*, 376:113620, 2021.

[4] U. Fasel, J. N. Kutz, B. W. Brunton, and S. L. Brunton. Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478(2260), apr 2022.

[5] E. Kaiser, J. N. Kutz, and S. L. Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2219):20180335, nov 2018.

[6] Klaus Schittkowski and Christian Zillober. Nonlinear programming: Algorithms, software, and applications. volume 166, pages 73–107, 01 2006.

[7] K. Yamanaka and F. Ankersen. “new state transition matrix for relative motion on an arbitrary elliptical orbit, ” Journal of Guidance, Control, and Dynamics. 25:60–66, January 2002.

[8] Zhang Zhang, Lei Deng, Jiaqi Feng, Liang Chang, Dong Li, and Yilin Qin. A survey of precision formation relative state measurement technology for distributed spacecraft. *Aerospace*, 9(7):362, 2022.