Spacecraft Rendezvous using Chance-Constrained Model Predictive Control and ON/OFF thrusters

Rafael Vázquez

Francisco Gavilán   Eduardo F. Camacho

Universidad de Sevilla

Workshop on Advances in Space Rendezvous Guidance
31 October 2013, Toulouse

Outline

1 Introduction
- MPC
- Rendezvous model, Constraints, Cost Function

2 MPC applied to Rendezvous
- MPC formulation for Spacecraft Rendezvous
- Robust and Chance-Constrained MPC with perturbation estimator
- Simulation Results for Chance-Constrained MPC

3 ON/OFF thrusters
- Model
- Algorithm
- Simulations

About MPC

- The main idea of MPC is to use, for each time instant, a control signal that is computed from an optimal plan that minimizes an objective function and verifies the constraints, in an *sliding time horizon*.
- A good references to start with MPC is Camacho, E. and Bordons, C. (2004). *Model Predictive Control*.
- How one does typically MPC:
  1. Discretize the system for a finite number of time intervals (time horizon), assuming inputs constant (ZOH).
  2. Predict the state, based on the actual state and the future inputs of the system (which are to be computed).
  3. Optimize the inputs for the time horizon such that a given objective function is minimized, and input, state and terminal constraints are.
  4. Apply the first input or inputs corresponding to the current time interval.
  5. When the next time interval begins, repeat (thus closing the loop!). This is called a receding or sliding horizon.

LTI example. Discretization.

- Consider:

$$\dot{x} = Ax + Bu$$

- Set N_p time intervals with duration of T, i.e. $[kT, (k+1)T]$ for $k = 0, \ldots, N_p$. Denote $t_k = kT$ and $x(k) = x(t_k)$.
- Assume u constant during t_k and equal to $u(k)$.
- Then:

$$x(k+1) = A_d x(k) + B_d u(k)$$

where the matrices A_d and B_d are computed as:

$$A_d = \mathrm{e}^{AT}, \quad B_d = \int_0^T \mathrm{e}^{A(T-\tau)} B d\tau$$

LTI example. Prediction of the state.

- From

$$x(k+1) = A_d x(k) + B_d u(k)$$

we predict $x(k+j)$:

$$x(k+j) = A_d^j x(k) + \sum_{i=0}^{j-1} A_d^{j-i-1} B_d u(k+i)$$

- This can be written as:

$$x(k+j) = F(j)x(k) + G(j) \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+j-1) \end{bmatrix}$$

LTI example. Optimization.

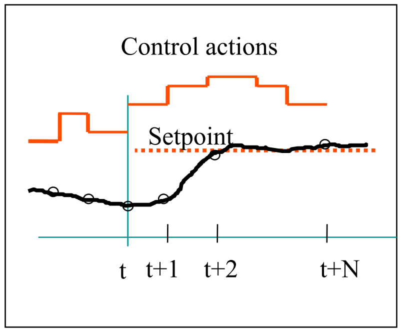
- Given inequality constraints

$$\forall k \in [0, N_p - 1], \quad A_i x(k) \leq b_i, \qquad A_u u \leq b_u$$

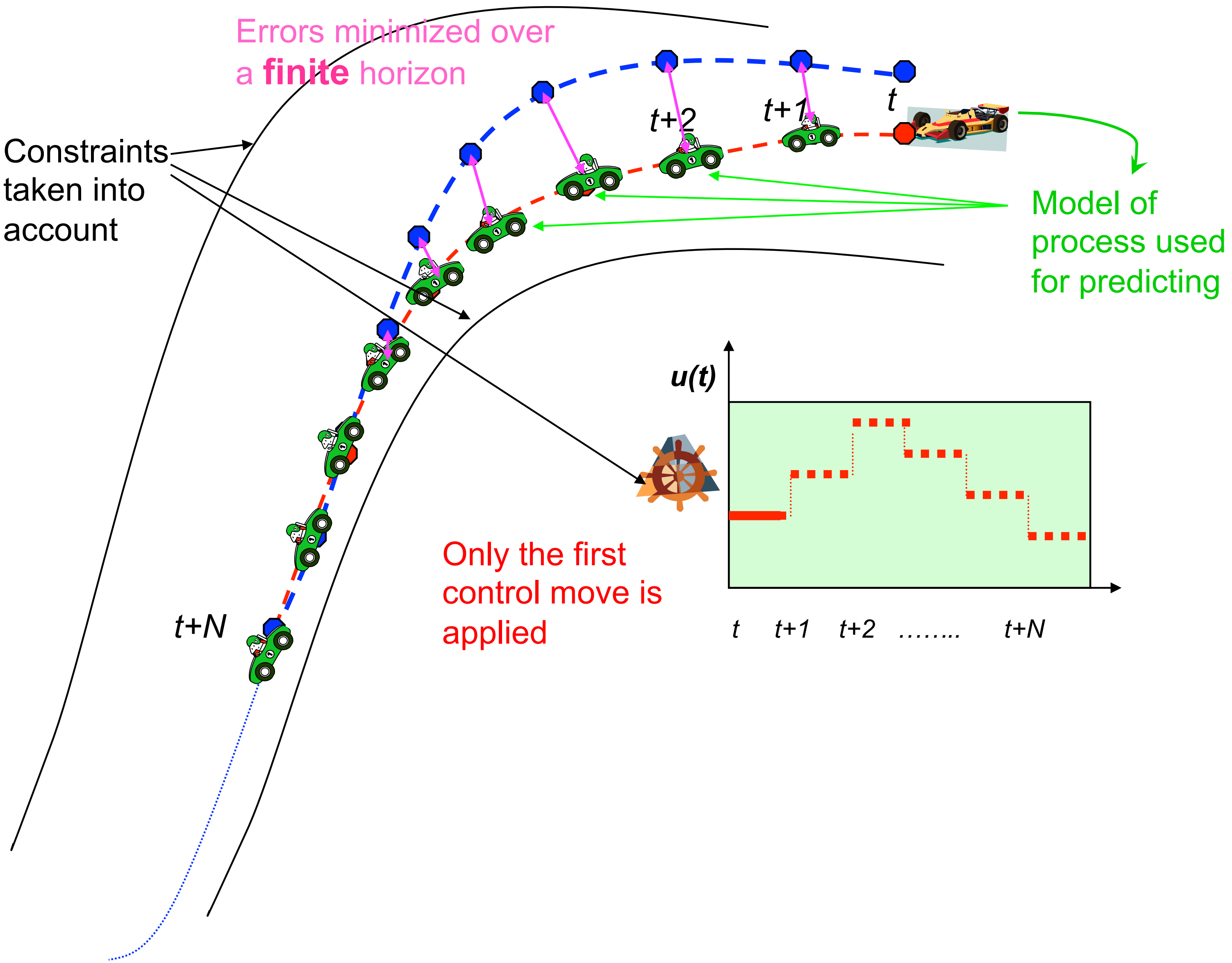
  and terminal constraints $A_t x(N_p) = b_t$.
- Given an objective function $J(x, u)$ to minimize over a finite horizon $\mathcal{K} \in [0, N_p]$.
- If we know $x(0)$, all constraints can be put in terms of $u(0)$, ..., $u(N_p - 1)$.
- Since the inputs are a discrete, finite set → finite-dimensional optimization problem. Easily solvable if the objective function is quadratic or linear!

LTI example. Receding horizon

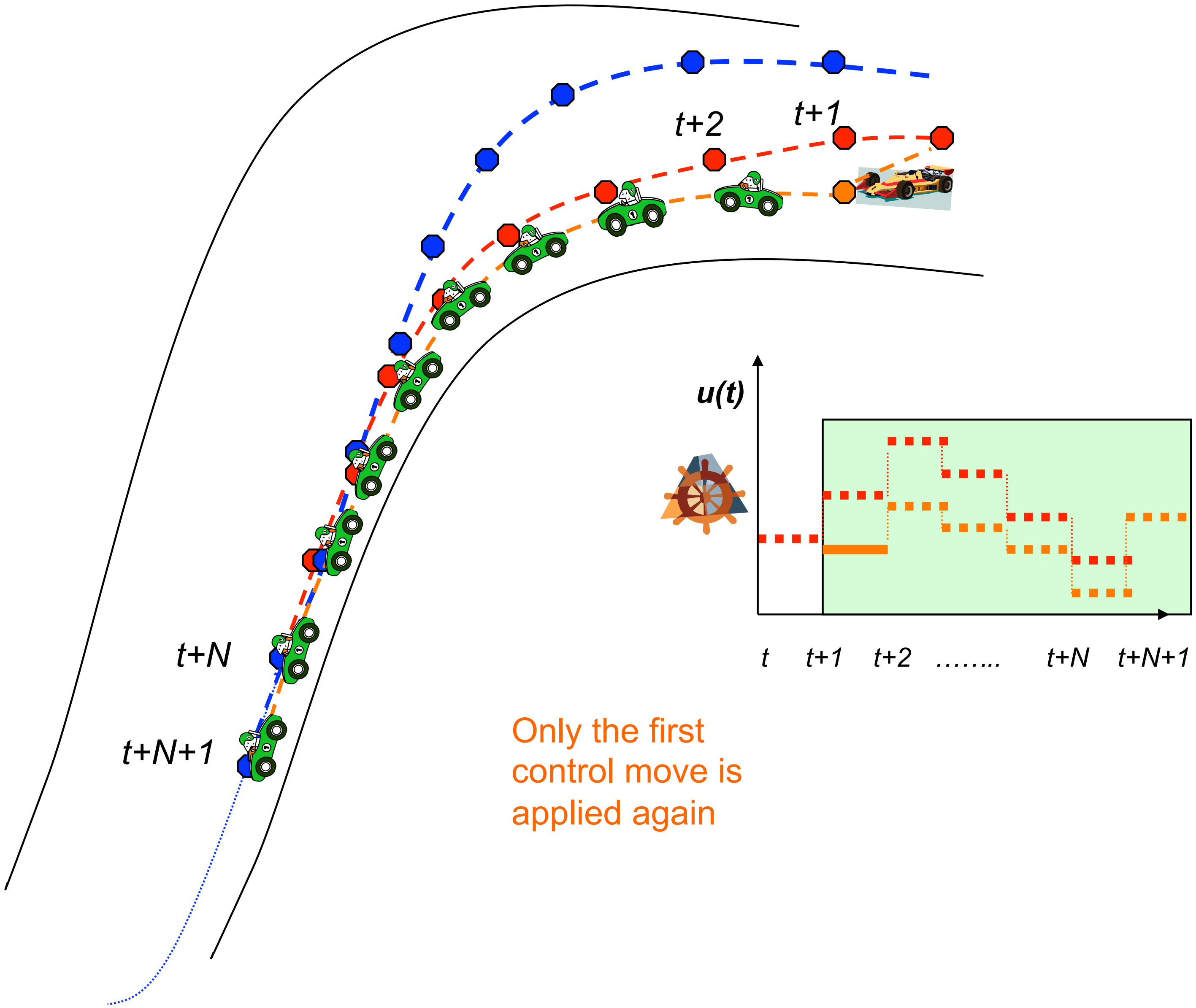
- We now apply the first control $u(0)$.
- Uncertainties/unmodelled dynamics might make the prediction to fail.
- That is the reason why open-loop optimal control usually does not work in practice (on its own).
- The approach of MPC is: "discard" the pre-computed values $u(1), \ldots, u(N_p - 1)$ and repeat the optimization process (using $x(1)$, which we know, as a new initial condition!).
- In the optimization process, we compute $u(1), \ldots, u(N_p - 1), u(N_p)$. Again we apply only $u(1)$ and when we reach $x(2)$ we repeat the process!
- Thus MPC is really closed-loop control!



A guidance example: first step



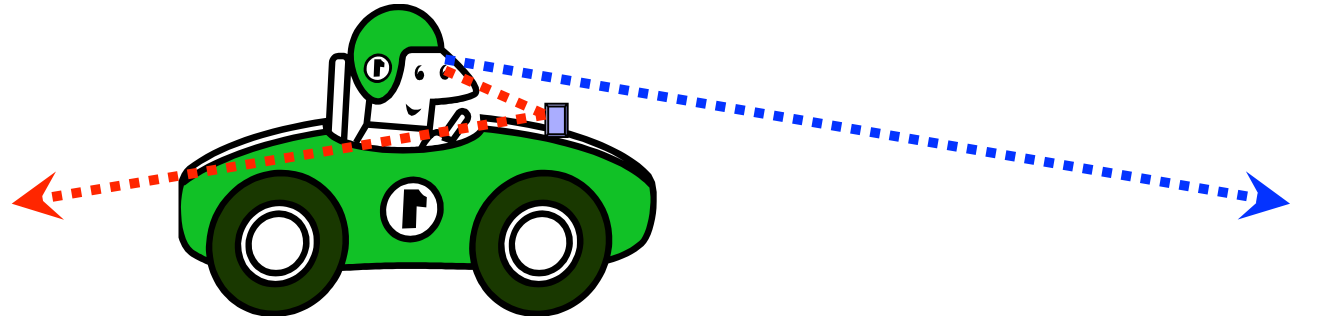
A guidance example: second step



Only the first control move is applied again

A guidance example: MPC vs PID

MPC vs. PID



PID: $u(t)=u(t-1)+g_0\, e(t) + g_1\, e(t-1) + g_2\, e(t-2)$

Advantages and Disadvantages of MPC

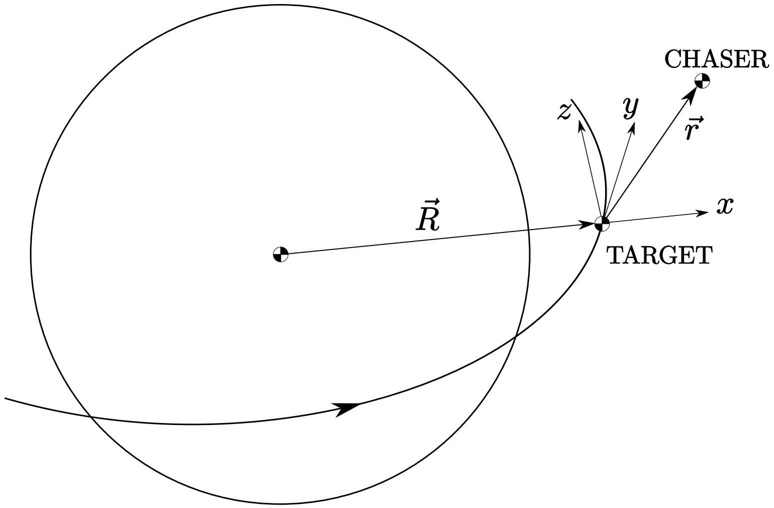
- Advantages: it looks into the future, it is optimal, it can treat many type of constraints, it guarantees a good performance of the system. It can also consider disturbances!
- Disadvantages: hard for nonlinear systems, requires some time for optimal input computation.
- It has been widely used in real life, for instance in chemical plants (there are companies specializing in MPC).
- However now that computational resources are cheap and more powerful, MPC is emerging as a feasible technique for many applications, for instance in the aerospace field.
- **Spacecraft rendezvous is an excellent example**, since it is very well described by linear equations and it is a slow system.

HCW model

- Under the usual assumptions (chaser close to the target, target in a keplerian orbit with zero eccentricity) we can use the Hill-Clohessy-Wiltshire (**HCW**) model:

$$\begin{aligned} \ddot{x} &= 3n^2x + 2n\dot{y} + u_x, \\ \ddot{y} &= -2n\dot{x} + u_y, \\ \ddot{z} &= -n^2z + u_z, \end{aligned}$$

in the LVLH frame, with n the mean orbital velocity.



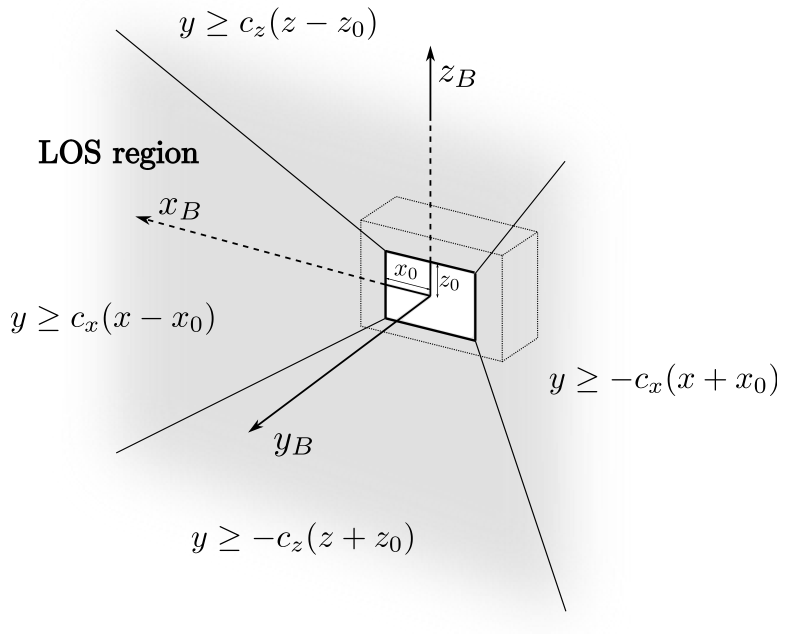
LVLH FRAME

Constraints of the problem

- Typical constraints:
  - Thruster limitations and mode of operation (PWM or PAM).
  - Avoid collisions between chaser and target (**safety**).
  - Typically, chaser must approach inside a previously designated safe zone.
  - If there are chaser engine failures, rendezvous should still be achieved, if possible (**fault tolerant control**).
  - If the target's attitude is changing with time (spinning target) the chaser should couple with that rotation to still guarantee rendezvous.
  - In case of total failure, collision probability should be as small as possible.
- Such constraints should be satisfied at the same time that fuel consumption is optimized (**economy**).

Safe zone

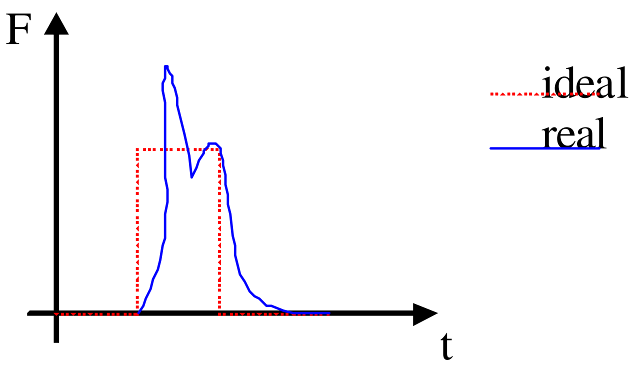
- In this work we will equal the safe zone with the "line of sight" (LOS)



- These LOS zone in the figure is described by the equations $y \geq c_x(x - x_0)$, $y \geq -c_x(x + x_0)$, $y \geq c_z(z - z_0)$, $y \geq -c_z(z + z_0)$ and $y > 0$.

Actuator constraints and Cost Function

- Typically there are two types of actuator:
  - Pulse-Amplitude Modulated (PAM): Any value of force in a given range can be used. $u_{min} \leq u(t) \leq u_{max}$. In spacecraft, this can be achieved by using electrical propulsion.
  - Pulse-Width Modulated (PWM): The value of force is fixed, only the start and duration of it can be set. In spacecraft, this is achieved by using conventional chemical thrusters (however it is far from perfect).



- Also, consumption of fuel should be minimized. Typically one seeks min $\int_0^{t_F} |\vec{u}(t)|^2 dt$ or min $\int_0^{t_F} |\vec{u}(t)| dt$.

HCW model in discrete time with perturbations

- Assuming that the control signal is constant for each sampling time T, we obtain the following discrete time version of the HCW equations:

$$\mathbf{x}(k+1) = A_T\mathbf{x}(k) + B_T\mathbf{u}(k) + \delta(k).$$

- A_T and B_T are:

$$A_T = \begin{bmatrix} 4-3C & 0 & 0 & \frac{S}{n} & \frac{2(1-C)}{n} & 0 \\ 6(S-nT) & 1 & 0 & -\frac{2(1-C)}{n} & \frac{4S-3nT}{n} & 0 \\ 0 & 0 & C & 0 & 0 & \frac{S}{n} \\ 3nS & 0 & 0 & C & 2S & 0 \\ -6n(1-C) & 0 & 0 & -2S & 4C-3 & 0 \\ 0 & 0 & -nS & 0 & 0 & C \end{bmatrix}$$

$$B_T = \begin{bmatrix} \frac{1-C}{n^2} & \frac{2nT-2S}{n^2} & 0 \\ \frac{2(S-nT)}{n^2} & -\frac{3T^2}{2}+4\frac{1-C}{n^2} & 0 \\ 0 & 0 & \frac{1-C}{n^2} \\ \frac{S}{n} & 2\frac{1-C}{n} & 0 \\ \frac{2(C-1)}{n} & -3T+4\frac{S}{n} & 0 \\ 0 & 0 & \frac{S}{n} \end{bmatrix}$$

where $S = \sin nT$ y $C = \cos nT$ ($T = 60\,\mathrm{s}$ is used in this work). We will drop the subindex T in A_T and B_T.

State, perturbation and control variables

- $\mathbf{x}(k)$, $\mathbf{u}(k)$ y $\delta(k)$ denote respectively the state (position and velocity), control effort (propulsive force per unit mass) and perturbation for time $t = k$, where:

$$\begin{aligned} \mathbf{x} &= [x\ y\ z\ \dot{x}\ \dot{y}\ \dot{z}]^T, \mathbf{u} = [u_x\ u_y\ u_z]^T, \\ \delta &= [\delta_x\ \delta_y\ \delta_z\ \delta_{\dot{x}}\ \delta_{\dot{y}}\ \delta_{\dot{z}}]^T. \end{aligned}$$

- x, y, and z are position in the LVLH local frame about the center of gravity of the target.
- x is radial position, y is position along the orbit and z is perpendicular to the orbit.
- Velocity, control $\mathbf{u}(k)$ and perturbations $\delta(k)$ are also written in the LVLH frame.
- Perturbations are unknown, hence $\delta(k)$ is a 6-D random variable, of mean $\bar{\delta}$ and covariance matrix Σ also unknown.

Prediction of state and compact notation

- The state at $t = k + j$ is predicted from the past state $\mathbf{x}(k)$ and control and disturbances at times from $t = k$ to time $t = k + j - 1$ as:

$$\mathbf{x}(k+j) = A^j\mathbf{x}(k) + \sum_{i=0}^{j-1} A^{j-i-1}B\mathbf{u}(k+i) + \sum_{i=0}^{j-1} A^{j-i-1}\delta(k+i).$$

- We use a compact (stack) notation where we denote:

$$\mathbf{x_S}(k) = \begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{x}(k+2) \\ \vdots \\ \mathbf{x}(k+N_p) \end{bmatrix}, \mathbf{u_S}(k) = \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \vdots \\ \mathbf{u}(k+N_p-1) \end{bmatrix}, \delta_\mathbf{S}(k) = \begin{bmatrix} \delta(k) \\ \delta(k+1) \\ \vdots \\ \delta(k+N_p-1) \end{bmatrix}.$$

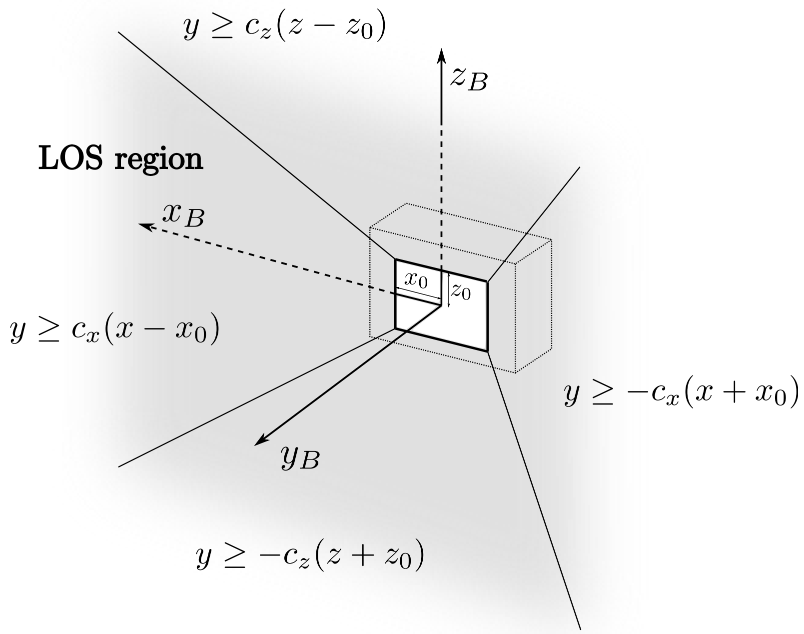
- Hence we can write the prediction equations as:

$$\mathbf{x_S}(k) = \mathbf{F}\mathbf{x}(k) + \mathbf{G_u}\mathbf{u_S}(k) + \mathbf{G}_\delta\delta_\mathbf{S}(k),$$

where $\mathbf{F}$, $\mathbf{G_u}$ and $\mathbf{G}_\delta$ are defined from the model matrices A and B.

Constraints

- Two kind of constraints have been included. Other constraints could be included as well.



- In the first place, it is required that the chaser is always inside a Line of Sight zone (LOS) with respect to the target.
- We write the restriction as $A_{LOS}\mathbf{x}(k) \leq b_{LOS}$.

$$A_{LOS} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ c_x & -1 & 0 & 0 & 0 & 0 \\ -c_x & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & c_z & 0 & 0 & 0 \\ 0 & -1 & -c_z & 0 & 0 & 0 \end{bmatrix}$$

$$b_{LOS} = \begin{bmatrix} 0 & c_x x_0 & c_x x_0 & c_z z_0 & c_z z_0 \end{bmatrix}^T$$

- Restrictions in the control signal: $\mathbf{u}_{min} \leq \mathbf{u}(k) \leq \mathbf{u}_{max}$

Objective function

- Taking expectation we define: $\hat{\mathbf{x}}(k+j|k) = E[\mathbf{x}(k+j)|\mathbf{x}(k)]$
- Similary $\hat{\mathbf{x}}_{\mathbf{S}}(k+j|k) = E[\mathbf{x}_{\mathbf{S}}(k+j)|\mathbf{x}(k)]$.
- Objective function:

$$J(k) = \sum_{i=1}^{N_p} \left[\hat{\mathbf{x}}^T(k+i|k)R(k+i)\hat{\mathbf{x}}(k+i|k)\right] + \sum_{i=1}^{N_p} \left[\mathbf{u}^T(k+i-1)Q\mathbf{u}(k+i-1)\right],$$

where N_p is the control horizon.

- $Q = \mathrm{Id}_{3\times 3}$ and $R(k)$ is defined as:

$$R(k) = \gamma h(k-k_a) \begin{bmatrix} \mathrm{Id}_{3\times 3} & \Theta_{3\times 3} \\ \Theta_{3\times 3} & \Theta_{3\times 3} \end{bmatrix}.$$

where h is the step function, k_a is the desired arrival time and γ is a large number. Hence $R = 0$ before the arrival time, and after arrival time it gives a large weight to the error in position (distance from the origin).

Objective function and constraints in compact notation

- The objective function can be written as:

$$J(k) = (\mathbf{G_u u_S}(k) + \mathbf{Fx}(k) + \mathbf{G}_\delta \bar{\delta}_\mathbf{S})^T \mathbf{R_S} (\mathbf{G_u u_S}(k) + \mathbf{Fx}(k) + \mathbf{G}_\delta \bar{\delta}_\mathbf{S}) + \mathbf{u_S}^T \mathbf{Q_S u_S}$$

  where prediction of the state has been used. Note that it depends on the state at $t = k$ and the control and disturbances up to the control horizon. The matrices $\mathbf{R_S}$ and $\mathbf{Q_S}$ appearing in the expression are defined from R and Q respectively. The compact variable $\bar{\delta}_\mathbf{S}$ contains the disturbances mean.
- Similarly the LOS constraints are written as:

$$\mathbf{A}_c \mathbf{x_S} \leq \mathbf{b}_c,$$

  and using prediction of the state :

$$\mathbf{A}_c \mathbf{G_u u_S} \leq \mathbf{b}_c - \mathbf{A}_c \mathbf{Fx}(k) - \mathbf{A}_c \mathbf{G}_\delta \delta_\mathbf{S}$$

- Control signal restriction are written as $\mathbf{u}_{min} \leq \mathbf{u_S} \leq \mathbf{u}_{max}$.

Computation of control signal

- For $t = k$, the MPC problem is formulated as:

$$\begin{aligned} \min_{\mathbf{u_S}} \quad & J(\mathbf{x}(k), \mathbf{u_S}, \bar{\delta}_{\mathbf{S}}) \\ \text{subject to} \quad & \mathbf{A}_c \mathbf{G_u} \mathbf{u_S} \leq \mathbf{b}_c - \mathbf{A}_c \mathbf{F} \mathbf{x}(k) - \mathbf{A}_c \mathbf{G}_\delta \delta_{\mathbf{S}}, \ \forall \delta_{\mathbf{S}} \\ & \mathbf{u}_{min} \leq \mathbf{u_S} \leq \mathbf{u}_{max} \end{aligned}$$

- It is a quadratic cost function with linear constraints; $\mathbf{x}(k)$ is known, $\mathbf{u_S}$ has to be found.
- If perturbations $\delta_{\mathbf{S}}$ were known (or e.g. zero) the problem is easily solved. For instance, in MATLAB, using `quadprog`.
- The problem is solved for a time instante $t = k$, and one computes a complete history of future control signals from the state $\mathbf{x}(k)$. However only the control signal $\mathbf{u}(k)$ is used and the rest are discarded. The next time instant $t = k + 1$ the solution of the problem is recomputed using the new state $\mathbf{x}(k + 1)$, thus closing the loop.

Robust MPC with known perturbation bounds

- If perturbations are unknown, the previous problem is not solvable.
- Assume instead that we just know perturbation bounds: $\mathbf{A}_\delta \delta_\mathbf{S} \leq \mathbf{c}_\delta$ (admissible perturbations) and perturbation means $\bar{\delta}_\mathbf{S}$.
- A control system that achieves its objective for *all* admissible perturbations is called **robust**.
- To accommodate all admissible perturbations, we bound $-\mathbf{A}_c \mathbf{G}_\delta \delta_\mathbf{S}$ which appears in the minimization constraints, *for all admissible perturbations*.
- This procedure is always possible for bounded perturbations (with known bounds).

Computation of control (known perturbation bounds)

- Hence to compute the control signal in $t = k$ we solve:

$$\begin{aligned} \min_{\mathbf{u_S}} \quad & J(\mathbf{x}(k), \mathbf{u_S}, \bar{\delta}_{\mathbf{S}}) \\ \text{subject to} \quad & \mathbf{A}_c \mathbf{G_u} \mathbf{u_S} \leq \mathbf{b}_c - \mathbf{A}_c \mathbf{F} \mathbf{x}(k) + \mathbf{b}_\delta \\ & \mathbf{u}_{min} \leq \mathbf{u_S} \leq \mathbf{u}_{max} \end{aligned}$$

where $\mathbf{b}_\delta$ is a column vector, whose i-th terms $(\mathbf{b}_\delta)_i$ is given by

$$(\mathbf{b}_\delta)_i = \min_{\text{s.t. } \mathbf{A}_\delta \delta_{\mathbf{S}} \leq \mathbf{c}_\delta} a_i \delta_{\mathbf{S}}$$

and where a_i is the i-th row of the matrix $-\mathbf{A}_c \mathbf{G}_\delta$

- Hence for each time $t = k$ a minimization subproblem has to be solved before computing the control signal from the main minimization problem.

Some Remarks about Robust MPC

- When solving the minimization subproblem for the constraints, we get the constraints computed for the *worst* case scenario for admissible perturbations.
- Hence, since constraints are verified for that case, they are *robustly* verified, i.e., verified for any perturbation from the set of admissible perturbations.
- The minimization subproblem consists on a minimization problem for every row for the matrix $-\mathbf{A}_c\mathbf{G}_\delta$. However, being a linear optimization problem with linear restrictions, it can be efficiently solved in numerical form. For instance, in MATLAB, using the command `linprog`.

Robust MPC: Chance Constrained approach

- However, perturbation bounds are not always known a priori. Or they are too conservative. Then we can model the perturbations as random variables.
- Assumption: $\delta \sim N_6(\bar{\delta}, \Sigma)$. (Non-Gaussian models can also be used, however then the formulation is more complicated)
- Assume for the moment we know the mean $\bar{\delta}$ and the covariance matrix Σ of the perturbations.
- A **chance constrained robust control law** is one that achieves its objective with a certain given probability.
- Thus, we find a bound for the term $-\mathbf{A}_c\mathbf{G}_\delta\delta_\mathbf{S}$ which appears in the minimization constraints, verified with a probability p.
- Since $\delta \sim N_6(\bar{\delta}, \Sigma)$, for a given p, one can find a confidence region (ellipsoid), i.e., compute α such that

$$\left(\delta - \bar{\delta}\right)^T \Sigma^{-1} \left(\delta - \bar{\delta}\right) \leq \alpha$$

is verified with probability p.

Computation of control (Chance Constrained approach)

- To compute the control signal in $t = k$ we solve:

$$\begin{aligned} \min_{\mathbf{u_S}} \quad & J(\mathbf{x}(k), \mathbf{u_S}, \bar{\delta}_\mathbf{S}) \\ \text{subject to} \quad & \mathbf{A}_c \mathbf{G_u} \mathbf{u_S} \leq \mathbf{b}_c - \mathbf{A}_c \mathbf{F} \mathbf{x}(k) + \mathbf{b}_\delta \\ & \mathbf{u}_{min} \leq \mathbf{u_S} \leq \mathbf{u}_{max} \end{aligned}$$

  where $\mathbf{b}_\delta$ is a column vector, whose i-th terms $(\mathbf{b}_\delta)_i$ is given by

$$(\mathbf{b}_\delta)_i = \min_{\text{s.t. } (\delta - \bar{\delta})^T \Sigma^{-1} (\delta - \bar{\delta}) \leq \alpha} a_i \delta_\mathbf{S}$$

  and where a_i is the i-th row of the matrix $-\mathbf{A}_c \mathbf{G}_\delta$
- Again for each time $t = k$ a minimization subproblem has to be solved. However, this time it has an explicit solution:

$$(\mathbf{b}_\delta(k))_i = \sum_{j=0}^{N_p - 1} \left(-\sqrt{\alpha} \sqrt{a_{ij} \Sigma a_{ij}^T} + a_{ij} \bar{\delta} \right)$$

Some Remarks about the Chance Constrained approach

- Since the minimization subproblem is explicitly solved, this approach gives an algorithm as fast as the non-robust MPC.
- However:
  - Needs estimation of statistical properties.
  - The normal distribution is unbounded: cannot choose the probability p of constraint satisfaction too large: conservativeness or even unfeasibility.
  - Each constraint satisfied with probability p: global probability smaller. However compensated with the receding horizon of MPC!

Algorithm for estimating perturbations

- The Chance Constrained Robust MPC, as it has been formulated, requires knowing the mean and covariance of the perturbations.
- Frequently, perturbations are totally unknown and these data has to be obtained online using an estimator.
- Then, for each $t = k$ we estimate $\bar{\delta}$ y Σ taking into account past perturbations, using:

$$\delta(i) = \mathbf{x}(i+1) - A\mathbf{x}(i) - B\mathbf{u}(i),$$

for $i = 1, \ldots, k-1$.

Estimating mean and covariance

- Denoting by $\hat{\delta}(k)$ y $\hat{\Sigma}(k)$ the estimations of $\bar{\delta}$ y Σ at $t = k$:

$$\hat{\delta}(k) = \frac{\sum_{i=0}^{k-1} e^{-\lambda(k-i)} \delta(i)}{\sum_{i=0}^{k-1} e^{-\lambda(k-i)}},$$

$$\hat{\Sigma}(k) = \frac{\sum_{i=0}^{k-1} e^{-\lambda(k-i)} \left(\delta(i) - \hat{\delta}(i)\right)\left(\delta(i) - \hat{\delta}(i)\right)^T}{\sum_{i=0}^{k-1} e^{-\lambda(k-i)}},$$

- The function $e^{-\lambda i}$ weights in the value of $\delta(i)$ in the sum, where $\lambda > 0$ is a forgetting factor.
- This is done to give more importance to the recent values of δ than to its past history.
- This weighting is useful is properties of the perturbations change with time, i.e., perturbations are not only random variables but stochastic processes.

Recursive formulae

- It is possible to use recursive formulae for the previous computations of mean and covariance:

$$
\begin{aligned}
\hat{\delta}(k) &= \frac{e^{-\lambda}}{\gamma_k}\left(\gamma_{k-1}\hat{\delta}(k-1) + \delta(k-1)\right), \\
\hat{\Sigma}(k) &= \frac{e^{-\lambda}}{\gamma_k}\left(\gamma_{k-1}\hat{\Sigma}(k-1)\right. \\
&\quad \left. + \left(\delta(k-1) - \hat{\delta}(k)\right)\left(\delta(k-1) - \hat{\delta}(k)\right)^T\right),
\end{aligned}
$$

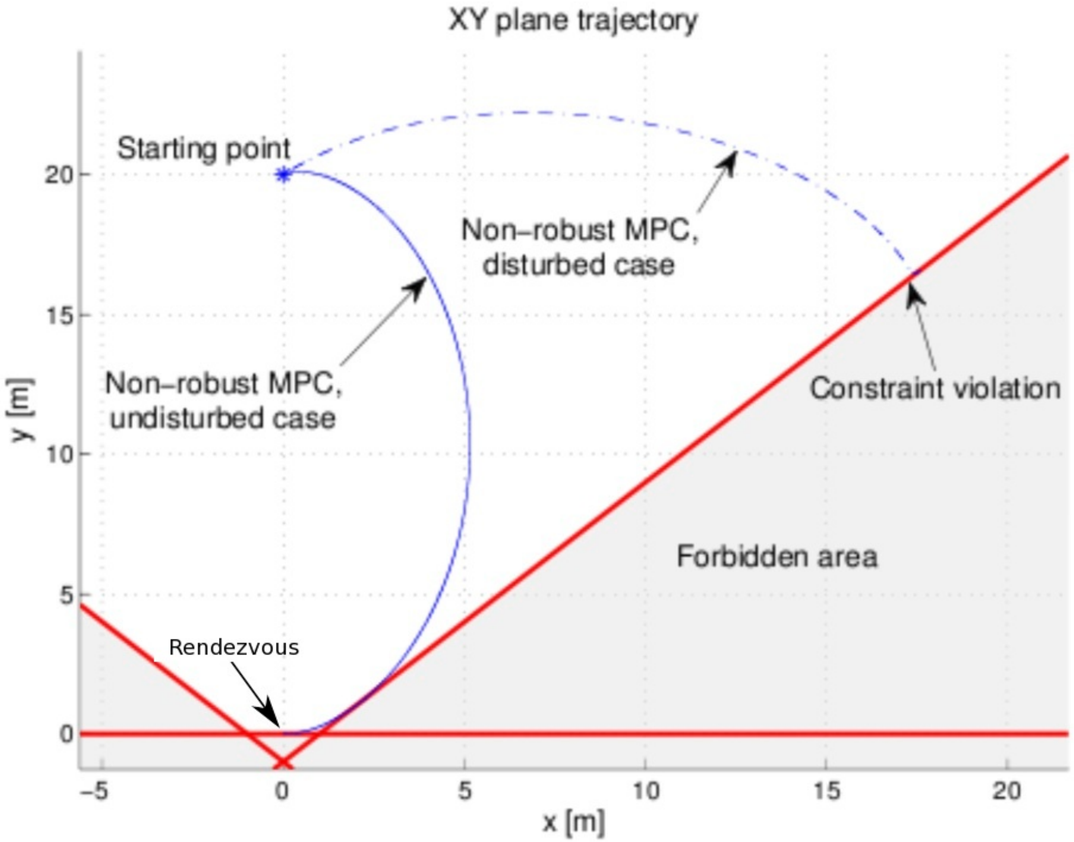
  where $\gamma_k = \frac{e^{-\lambda}\left(1-e^{-\lambda k}\right)}{1-e^{-\lambda}}$

- These allow to discard past values of δ and save memory.
- Once mean and covariance are obtained, it is possible to get the confidence region for disturbances that was used in the chance constrained approach.

Simulations

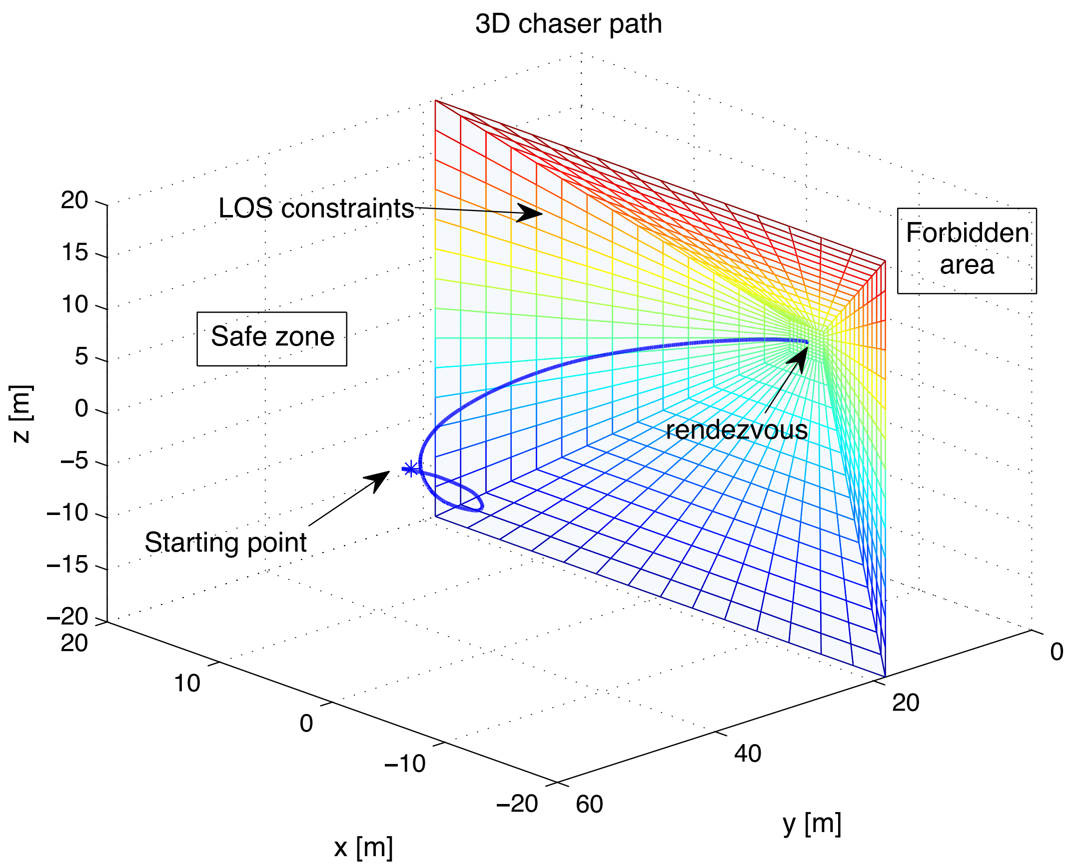
- For numerical simulations, several scenarios have been considered with and without perturbations.
- Parameters used: $R_0 = 6878\,\mathrm{km}$, $n = 1.1068 \cdot 10^{-3}\,\mathrm{rad/s}$, and LOS constraint parameters: $x_0 = z_0 = 1.5\,\mathrm{m}$ and $c_x = c_z = 1$.
- We included propulsive perturbations in the form: $\mathbf{u}_{\mathrm{real}} = (1 + \delta_1)T(\delta\theta)\mathbf{u}$, where:
  - $\mathbf{u}_{\mathrm{real}}$ is the real control signal given by the propulsive system.
  - $\mathbf{u}$ is the computed (desired) control signal.
  - δ_1 is a normally distributed random variable. Physically, δ_1 represents errors in the actuators.
  - $T(\delta\theta)$ is a rotation matrix with rotation angles given by $\delta\theta$, which is a normally distributed random vector of (small) angles. Physically, it comes from small errors in attitude that cause the engines to be slightly off course.
- Much more complex than nominal model.

Non-robust MPC controller



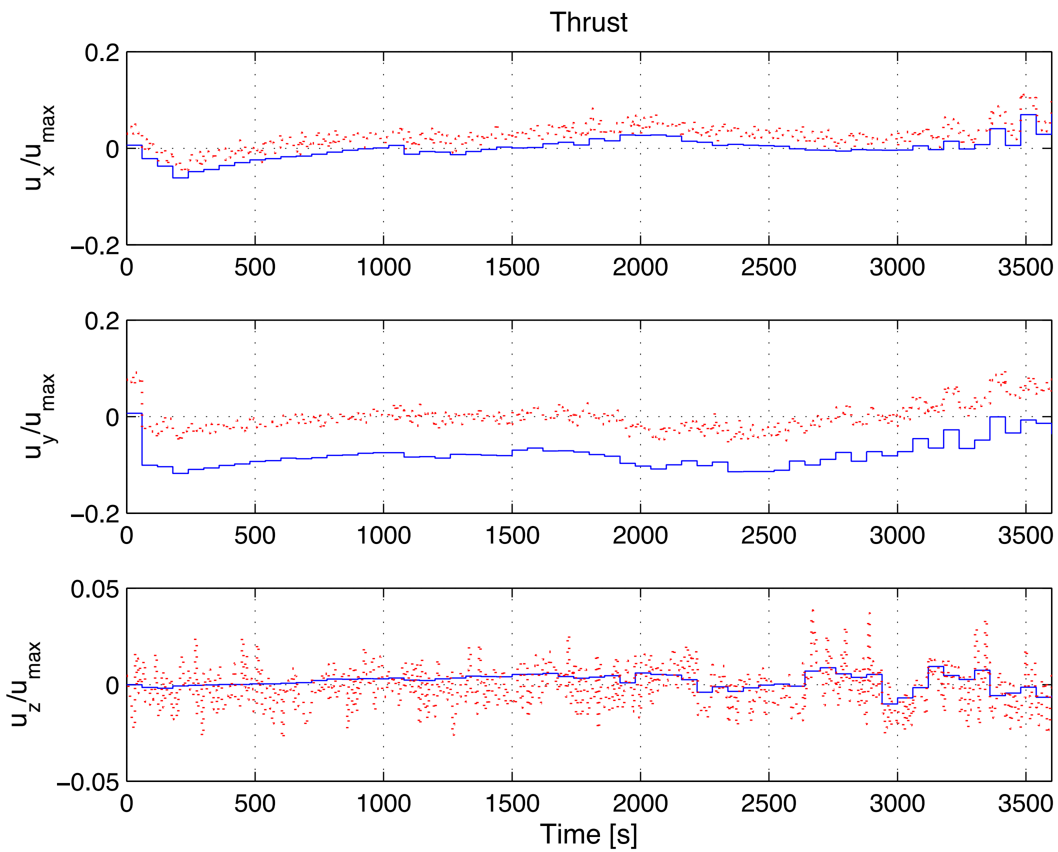
- Good results without perturbations (solid line).
- Fails when perturbations are present (dashed line). However if perturbations are small, still works.

Chance Constrained MPC controller with perturbations



- Includes perturbations. Good results!

Chance Constrained MPC controller with perturbations



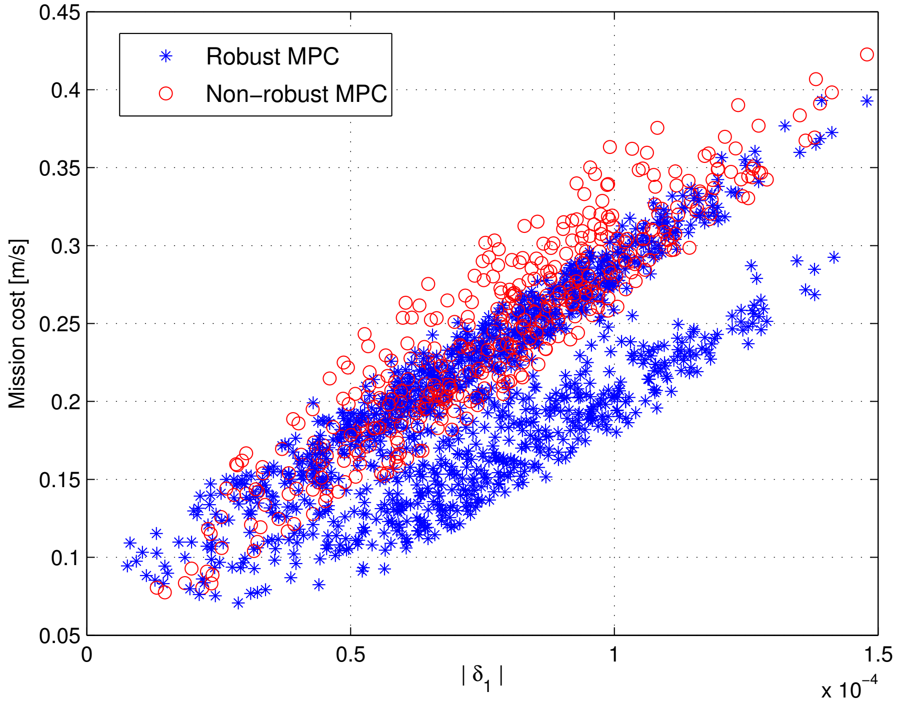
- Commanded control (solid) and applied control (dotted).

Monte Carlo simulations

- Simulated 1220 cases (with different disturbances). For each case we perform a simulation with the non-robust and another with the robust (chance constrained) approach.
- In the table d is the relative distance at the desired arrival time.

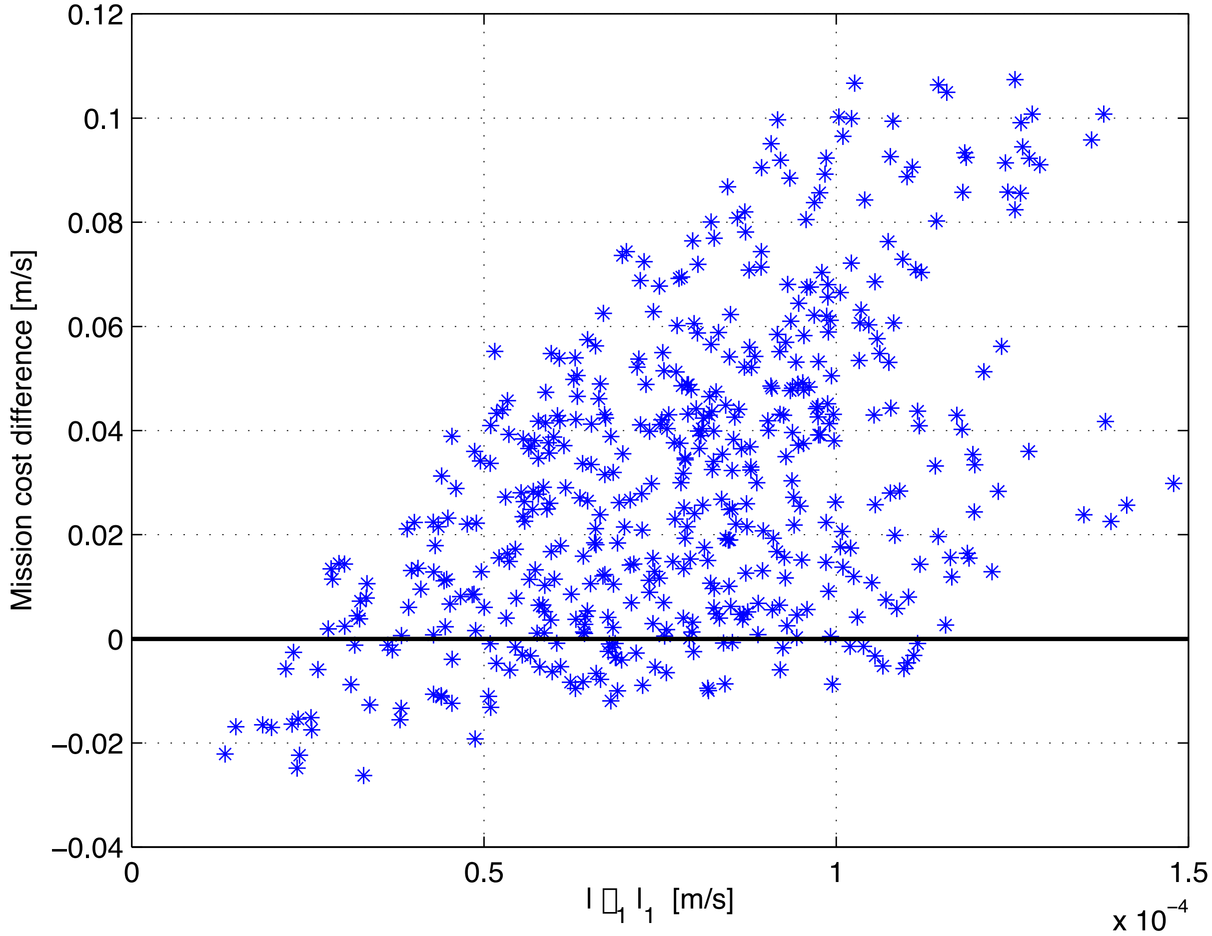
	Non-robust MPC	Robust MPC
Constraint violations	59%	0%
$d \leq 0.2\,\mathrm{m}$	19%	100%
$0.2\,\mathrm{m} \leq d \leq 0.5\,\mathrm{m}$	22%	0%
$0.5\,\mathrm{m} \leq d$	0%	0%
Mean cost (m/s) of successful missions	0.2444	0.2039

Monte Carlo simulations



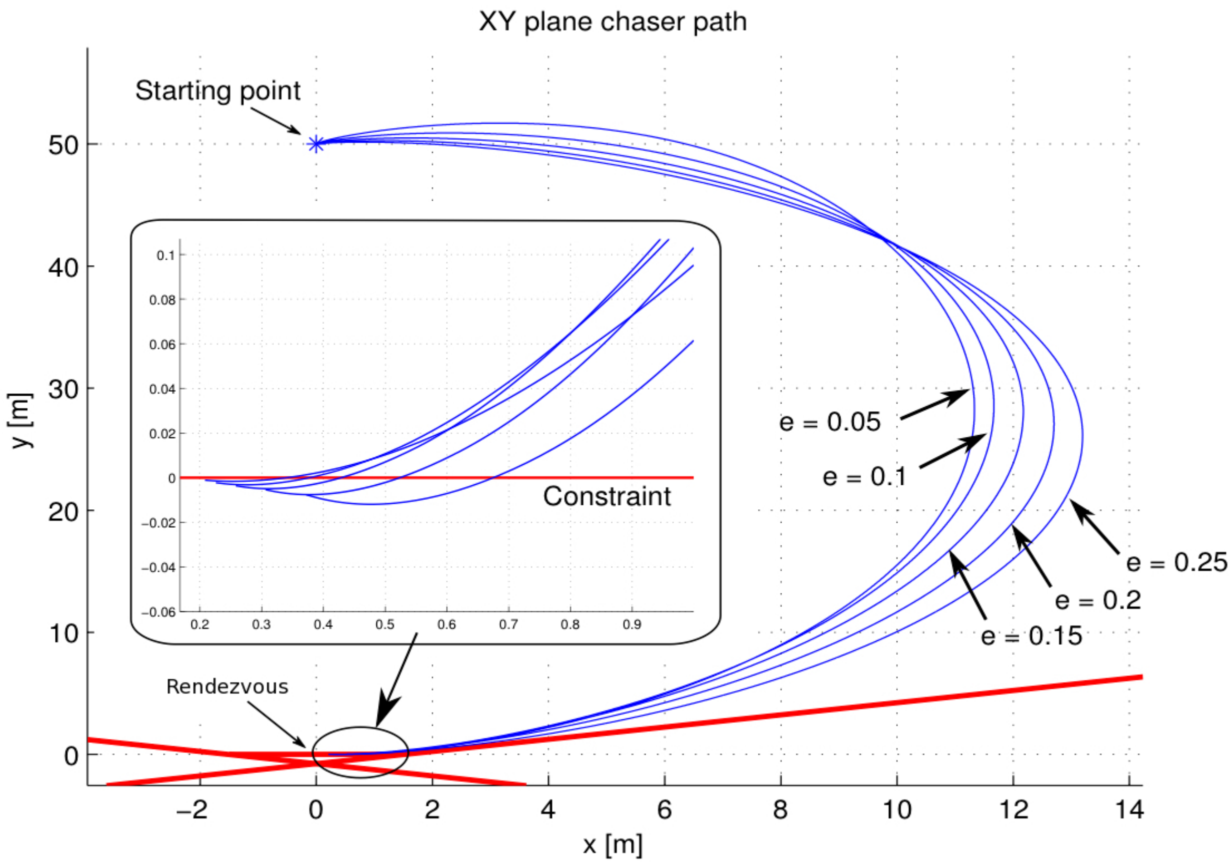
- Plot of total cost of successful missions for both robust and non-robust approach, against L_1 norm of the mean of the disturbances.
- It can be found that using the non-robust controller implies a 15% of cost increment.

Monte Carlo simulations



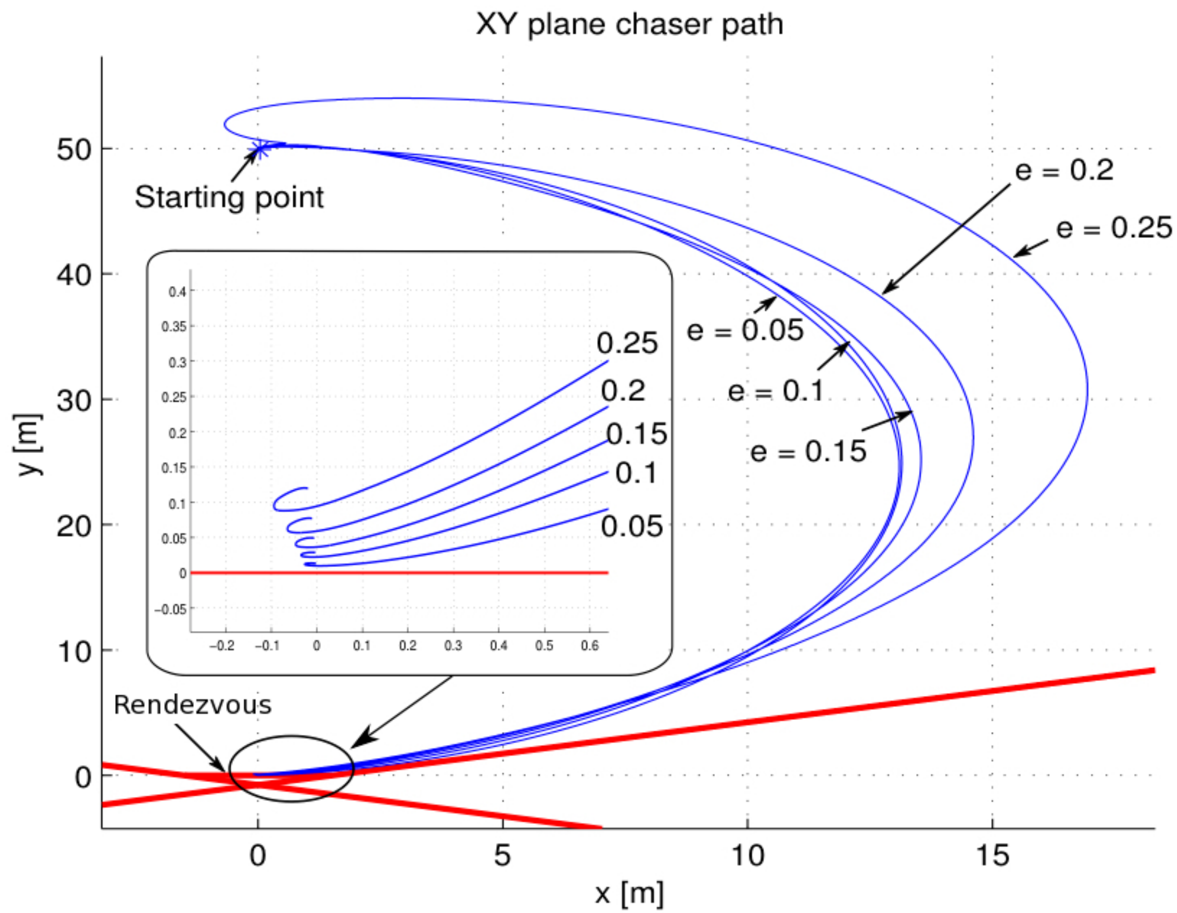
- Increase in cost of the non-robust MPC with respect to the chance constrained MPC.

Non-robust MPC controller with unmodeled dynamics



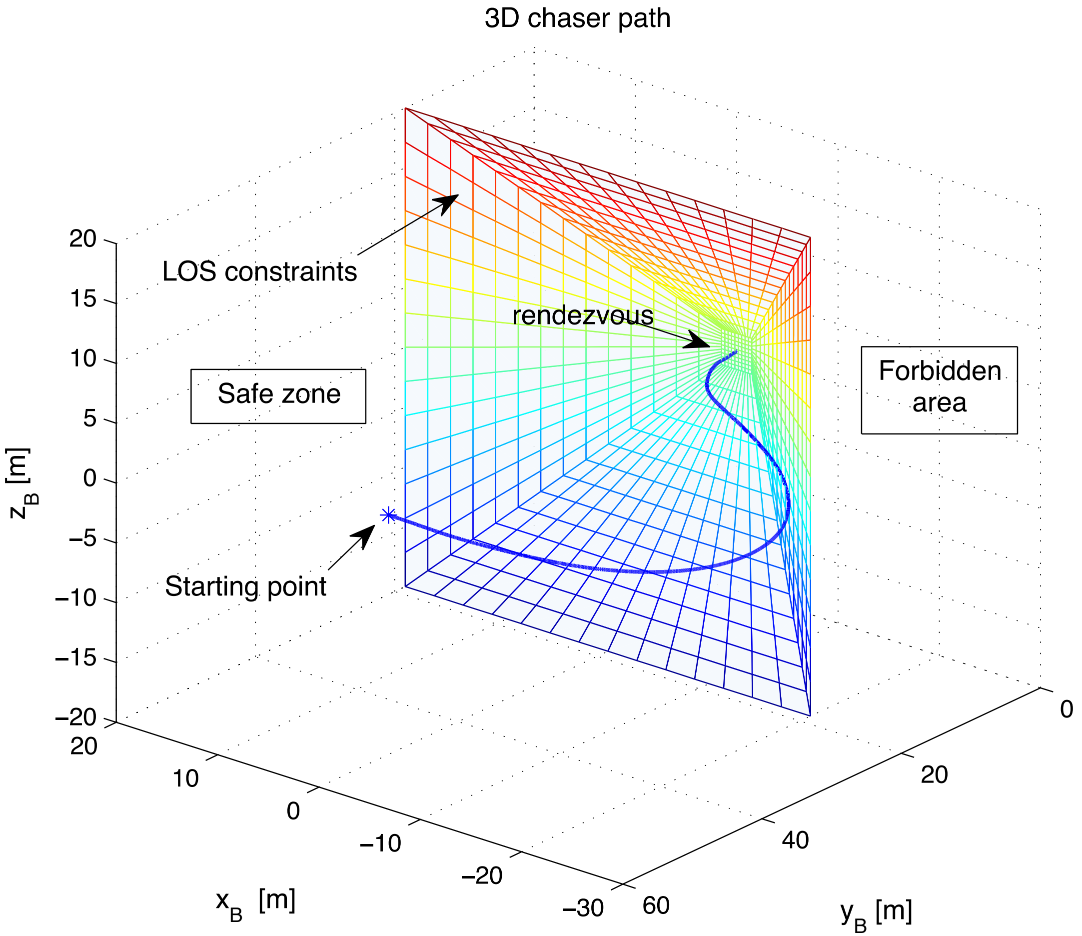
- Assume that the target orbit is elliptic (i.e. has some eccentricity e) instead of circular: unmodeled dynamics.
- Non-robust MPC is able to rendezvous, however it violates the constraints at the end.

Robust MPC controller with unmodeled dynamics



- Robust (chance-constrained) MPC does not violate constraints at the end.

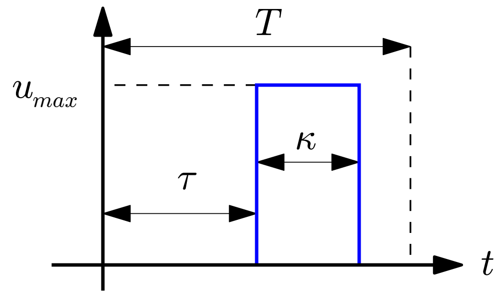
Rotating target, chance constrained MPC



- Rotating target (trajectory shown for axes fixed in target). Rendezvous is achieved.

Rendezvous with ON/OFF thrusters

- PWM control variables:
  - The pulse width κ.
  - The pulse start time τ.



- For simplification, consider only one pulse per time interval.
- Need six thrusters, one for each axis (denoted by x, y and z), and one for each direction (denoted by $+$ and -).
- 12 control variables for each k: $\kappa_1^+(k)$, $\kappa_2^+(k)$, $\kappa_3^+(k)$, $\kappa_1^+(k)$, $\kappa_2^-(k)$, $\kappa_3^-(k)$, $\tau_1^+(k)$, $\tau_2^+(k)$, $\tau_3^+(k)$, $\tau_1^+(k)$, $\tau_2^-(k)$, $\tau_3^-(k)$.
- The new variables control variables verify $\kappa_i^+(k) > 0$, $\tau_i^+(k) > 0$ and $\tau_i^+(k) + \kappa_i^+(k) < T$ (to prevent the PWM signal to spill over to the next time interval).

Rendezvous with ON/OFF thrusters: model

- Define

$$b_t^1 = \begin{bmatrix} \frac{1-C}{n^2} \\ \frac{2(S-nt)}{n^2} \\ 0 \\ \frac{S}{n} \\ \frac{2(C-1)}{n} \\ 0 \end{bmatrix}, \quad b_t^2 = \begin{bmatrix} \frac{2nt-2S}{n^2} \\ -\frac{3t^2}{2} + 4\frac{1-C}{n^2} \\ 0 \\ 2\frac{1-C}{n} \\ -3t + 4\frac{S}{n} \\ 0 \end{bmatrix}, \quad b_t^3 = \begin{bmatrix} 0 \\ 0 \\ \frac{1-C}{n^2} \\ 0 \\ 0 \\ \frac{S}{n} \end{bmatrix}$$

- The HCW equations are replaced in the PWM case by

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B_{PWM}(\mathbf{u_P}(k))\mathbf{u_{max}}$$

where:

$$B_{PWM}(\mathbf{u_P}(k)) = \begin{bmatrix} A_{T-\tau_1^+(k)-\kappa_1^+(k)} b^1_{\kappa_1^+(k)} \\ A_{T-\tau_1^-(k)-\kappa_1^-(k)} b^1_{\kappa_1^-(k)} \\ A_{T-\tau_2^+(k)-\kappa_2^+(k)} b^2_{\kappa_2^+(k)} \\ A_{T-\tau_2^-(k)-\kappa_2^-(k)} b^2_{\kappa_2^-(k)} \\ A_{T-\tau_3^+(k)-\kappa_3^+(k)} b^3_{\kappa_3^+(k)} \\ A_{T-\tau_3^-(k)-\kappa_3^-(k)} b^3_{\kappa_3^-(k)} \end{bmatrix}^T, \quad \mathbf{u_{max}} = \begin{bmatrix} u^+_{1max} \\ -u^-_{1max} \\ u^+_{2max} \\ -u^-_{2max} \\ u^+_{3max} \\ -u^-_{3max} \end{bmatrix}$$

Rendezvous with ON/OFF thrusters: model

- The equations are highly nonlinear in the control!
- The procedure with PAM control cannot be applied. We use linearization, applying the following algorithm:
    1. Solve the problem using the normal algorithm for PAM control.
    2. Use the optimal PAM-PWM filter to get a initial starting guess of the PWM solution (see next slide).
    3. Linearize around the actual PWM solution and find small increments in the PWM controls that improve the objective function and satisfy the constraints.
    4. Repeat previous step until it converges or time is up.
- Linearization is explicit and easy to compute (since the matrices come from a discretized continuous system).
- Since we have a very good initial guess the algorithm works well.

Optimal PAM-PWM filter

- Since we are linearizing it is crucial to have a good initial guess.
- The optimal PAM-PWM filter is an algorithm that takes a sequence of PAM control inputs and produces a sequence of PWM control inputs, such that both system outputs are very close.
- Found in the literature: e.g. Shieh et al, "Design of PAM and PWM controllers for sampled-data interval systems," J Dyn Syst Meas Contr., 118.
- Simple and system independent, works specially well for linear systems. Based on two rules:
  - The law of areas: both PWM and PAM control inputs must produce, for each sample interval, the same area.
  - The pulse (when there is only one) must be centered in the sample interval.

Linearization of the PWM model

- The linearized model is written as

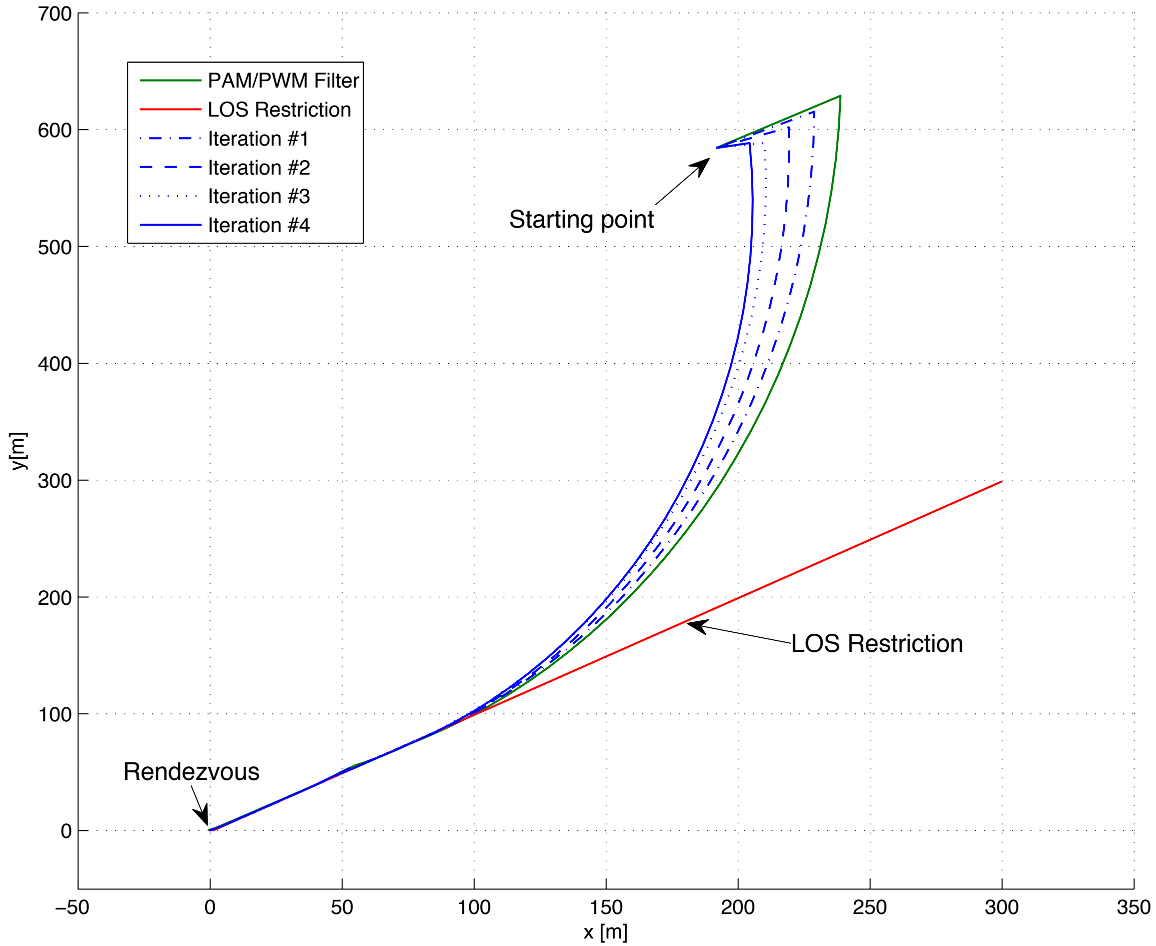
$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B_{PWM}(\mathbf{u_P}(k))\mathbf{u_{max}} + B^{\Delta}(\mathbf{u_P}(k))\mathbf{\Delta}(k)$$

- $\mathbf{\Delta}(k)$ are the increments in the PWM signals and the matrix $B^{\Delta}(\mathbf{u_P}(k))$ is defined explicitly as:

$$B^{\Delta} = \begin{bmatrix} -A'_{T-\tau_1^+-\kappa_1^+} b^1_{\kappa_1^+} u^+_{1max} \\ \left(-A'_{T-\tau_1^+-\kappa_1^+} b^1_{\kappa_1^+} + A_{T-\tau_1^+-\kappa_1^+} b^{1'}_{\kappa_1^+}\right) u^+_{1max} \\ \vdots \\ \left(A'_{T-\tau_3^--\kappa_3^-} b^3_{\kappa_3^-} - A_{T-\tau_3^--\kappa_3^-} b^{3'}_{\kappa_3^-}\right) u^-_{3max} \end{bmatrix}^T,$$

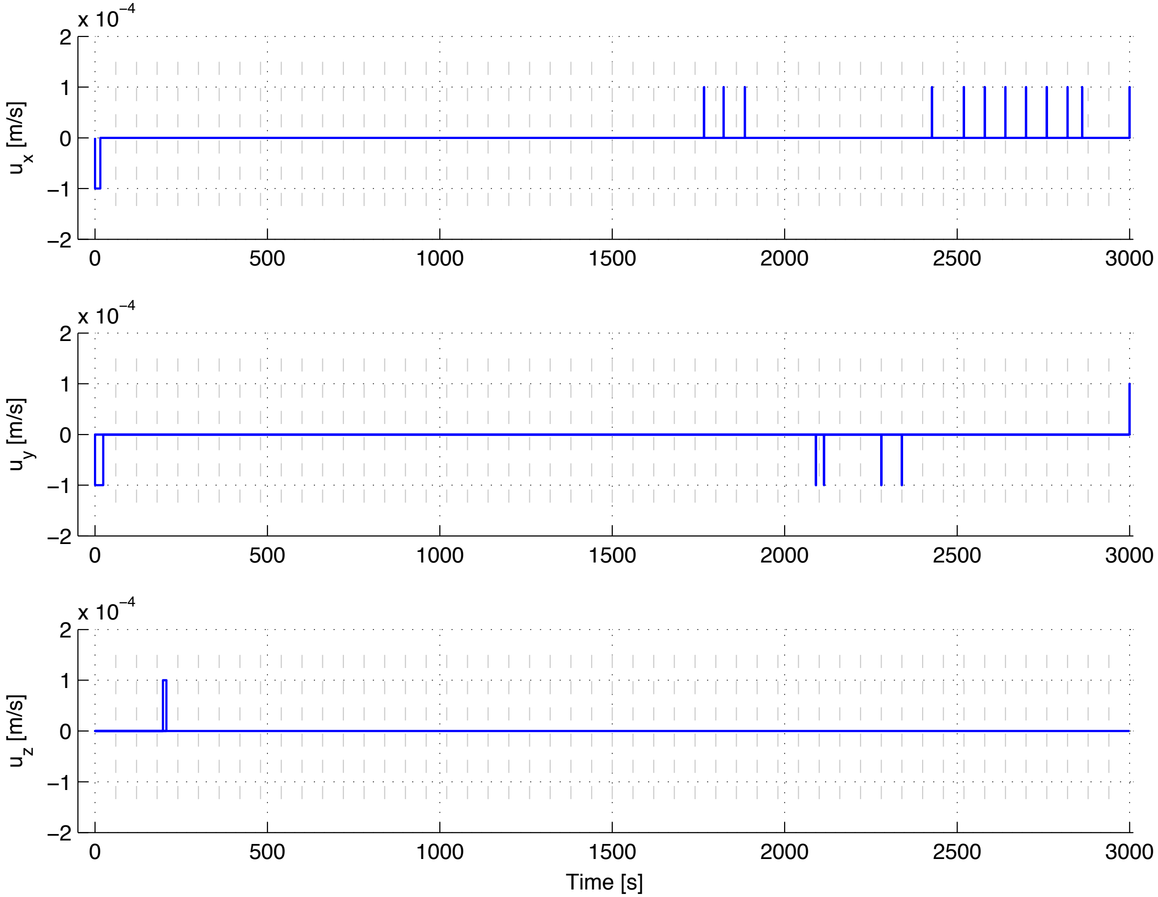
- In the matrix, $A'_t = \frac{d}{dt}A_t$, $b^{i'}_t = \frac{d}{dt}b^i_t$.
- Since the model is now linear, optimization is fast (even in Matlab!).

Simulation results for the PWM algorithm



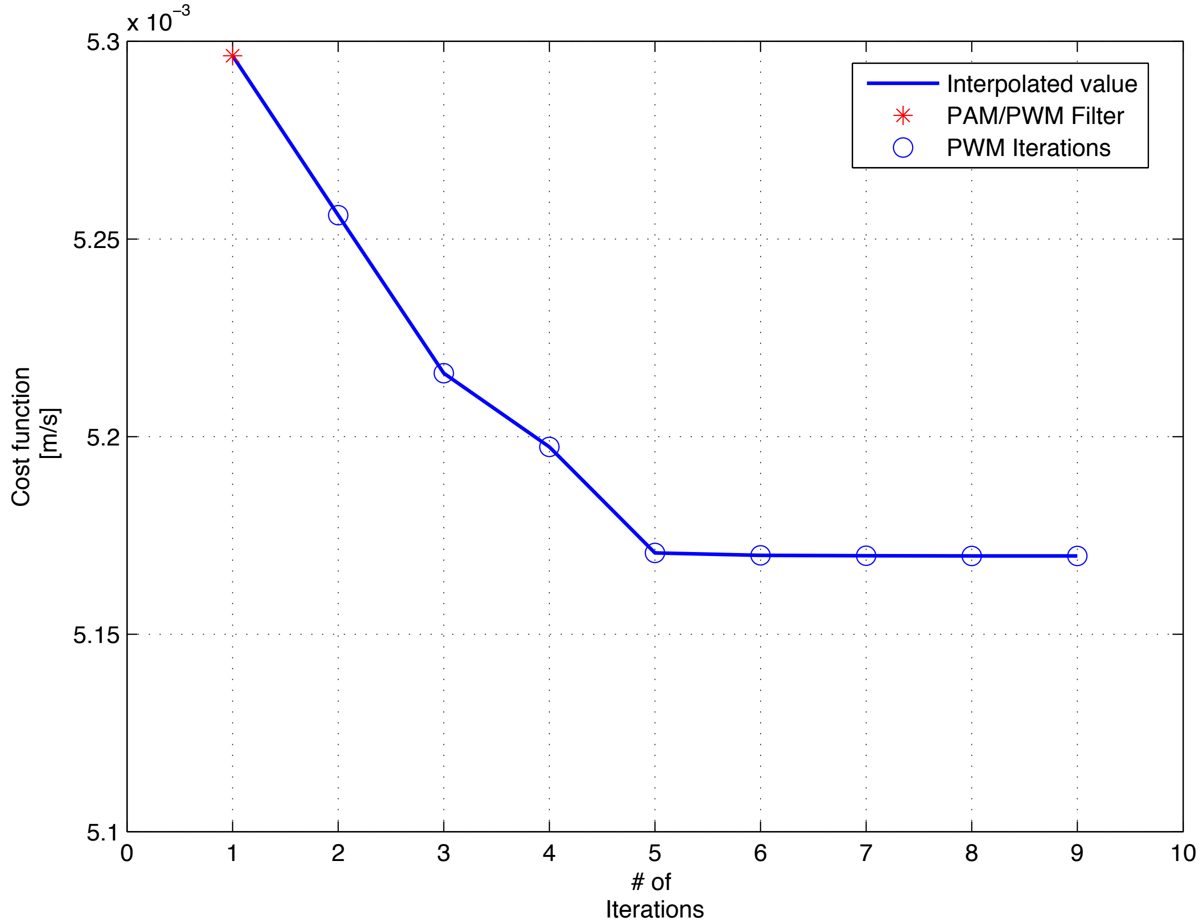
- Comparison between a PAM and PWM trajectory applying the algorithm. Without disturbances.

Simulation results for the PWM algorithm



- Resulting PWM control sequence.

Simulation results for the PWM algorithm

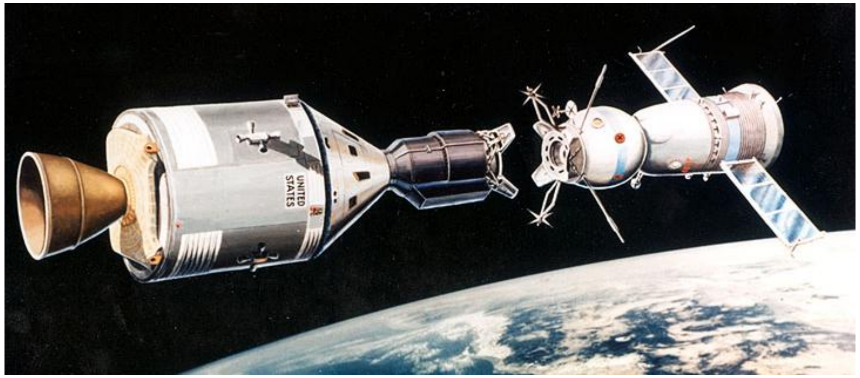


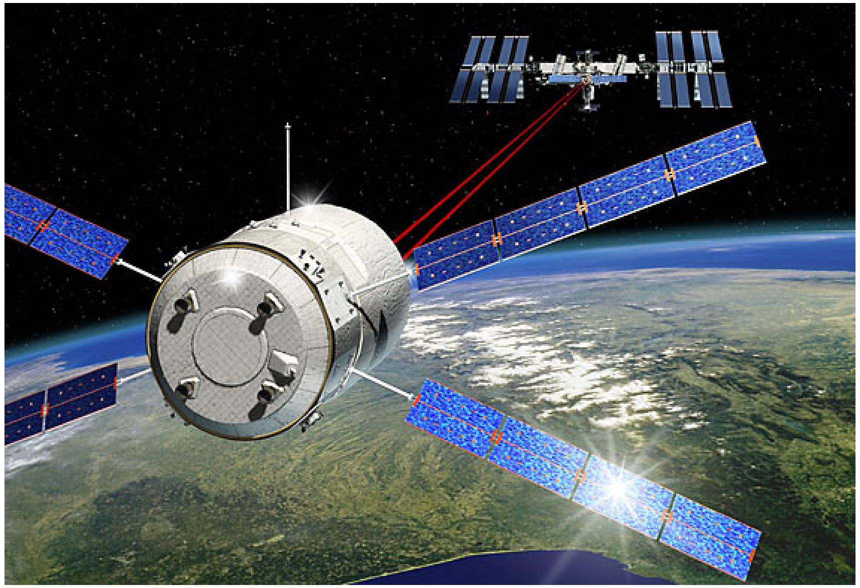
- Improvement in the cost function for each iteration.
- After 5-6 iterations, it converges.
- Slight improvement in cost.

Conclusions

- We have presented a robust MPC controller to solve the problem of automatic spacecraft rendezvous.
- Perturbations are estimated online and accommodated.
- In simulations it is shown that the method can overcome large disturbance and unmodeled dynamics.
- PWM control constraints have been included in the model.
- Future work:
  - Include eccentricity and orbital perturbations.
  - Add an state estimator (based e.g. on observations from target).
  - Include fault-tolerant schemes and safety constraints.
  - Use more sophisticated disturbance estimation techniques.
  - Study stability of the closed loop system.
  - Reduce # of actuators, include attitude dynamics (nonlinear).
- References:

1. F. Gavilan, R. Vazquez, E. F. Camacho, "Robust Model Predictive Control for Spacecraft Rendezvous with Online Prediction of Disturbance Bounds," IFAC AGNFCS'09, Samara, Russia, 2009.
2. R. Vazquez, F. Gavilan, E. F. Camacho, "Trajectory Planning for Spacecraft Rendezvous with On/Off Thrusters," IFAC World Congress, 2011.
3. F. Gavilan, R. Vazquez and E. F. Camacho,"Chance-constrained Model Predictive Control for Spacecraft Rendezvous with Disturbance Estimation," Control Engineering Practice, 20 (2), 111-122, 2012.





Thank you!

http://aero.us.es/rvazquez/research.htm