Model Predictive Control
(for Aerospace Applications)

Rafael Vazquez
Universidad de Sevilla

Milano, May 29 2019

Outline

1. Introduction to MPC (Slides by E.F. Camacho) ←
2. Application to Spacecraft Rendezvous (including PWM)
3. Rendezvous + Attitude Control
4. Soft Landing on an Asteroid
5. Guidance for UAVs

Model Predictive Control: an Introductory Survey

Eduardo F. Camacho

Universidad de Sevilla

MPC successful in industry.

- Many and very diverse and successful applications:
  - Refining, petrochemical, polymers,
  - Semiconductor production scheduling,
  - Air traffic control
  - Clinical anesthesia,
  - ….
  - Life Extending of Boiler-Turbine Systems via Model Predictive Methods, Li et al (2004)
- Many MPC vendors.

PREDICTIVE CONTROL IN COGNAC DISTILLATION

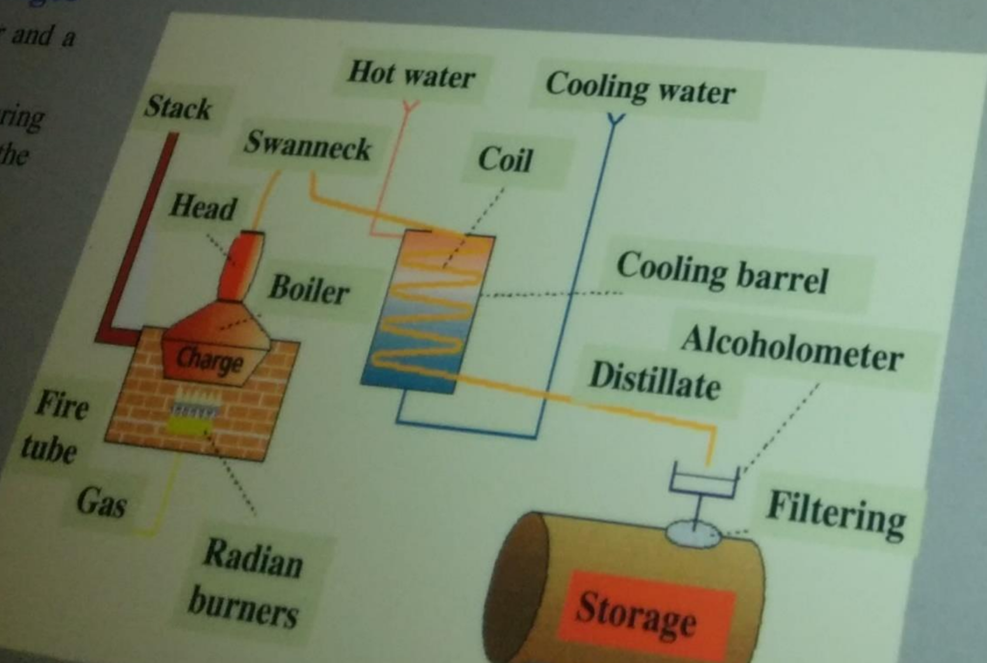
Patrick Boucher[1], Anne Boutillon[2], Didier Dumur[1],
Thierry Foix[3], Patrick Raguenaud[4], Christelle Rougebief[3]

[1] Supélec, Plateau de Moulon, F 91192 Gif-sur-Yvette Cedex, France
[2] Gaz de France Research and Development, F 93210 La Plaine Saint Denis, France
[3] ENSIA, 1 Avenue des Olympiades, F 91305 Massy Cedex, France
[4] Martell, BP 21, F 16101 Cognac Cedex, France

stages

ur and a
l,
during
the

e



✓ **Good "heat": broken down into**

- "Mise au courant", lasting roughly quarters, during which the liquid is bro
- "First run", lasting thirty to thirty-fi
- "Hearts" run, lasting five and a average, through to the cut at 60 (s alcohol content reaches 60%),
- "Seconds" run, which consists of the charge, for as short a time as po at 2 (the alcohol content reaches 2%)

→ The "hearts" make

MPC successful in Academia

- Many MPC sessions in control conferences and control journals, MPC workshops.
- 4/8 finalist papers for the *CEP best paper award* were MPC papers (2/3 finally awarded were MPC papers)

TABLE 1 A list of the survey results in order of industry impact as perceived by the committee members.

Rank and Technology	High-Impact Ratings	Low- or No-Impact Ratings
PID control	100%	0%
Model predictive control	78%	9%
System identification	61%	9%
Process data analytics	61%	17%
Soft sensing	52%	22%
Fault detection and identification	50%	18%
Decentralized and/or coordinated control	48%	30%
Intelligent control	35%	30%
Discrete-event systems	23%	32%
Nonlinear control	22%	35%
Adaptive control	17%	43%
Robust control	13%	43%
Hybrid dynamical systems	13%	43%

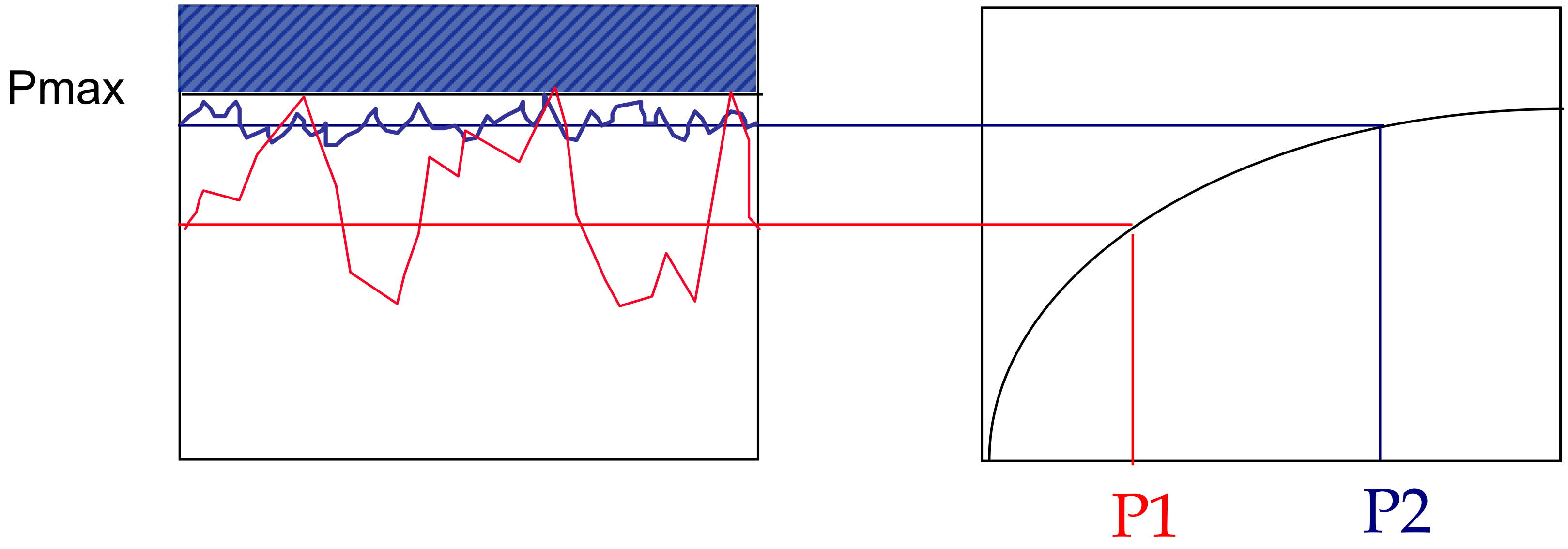
Tariq Samad, IEEE Control Systems Magazine, 2017

Why is MPC so successful ?

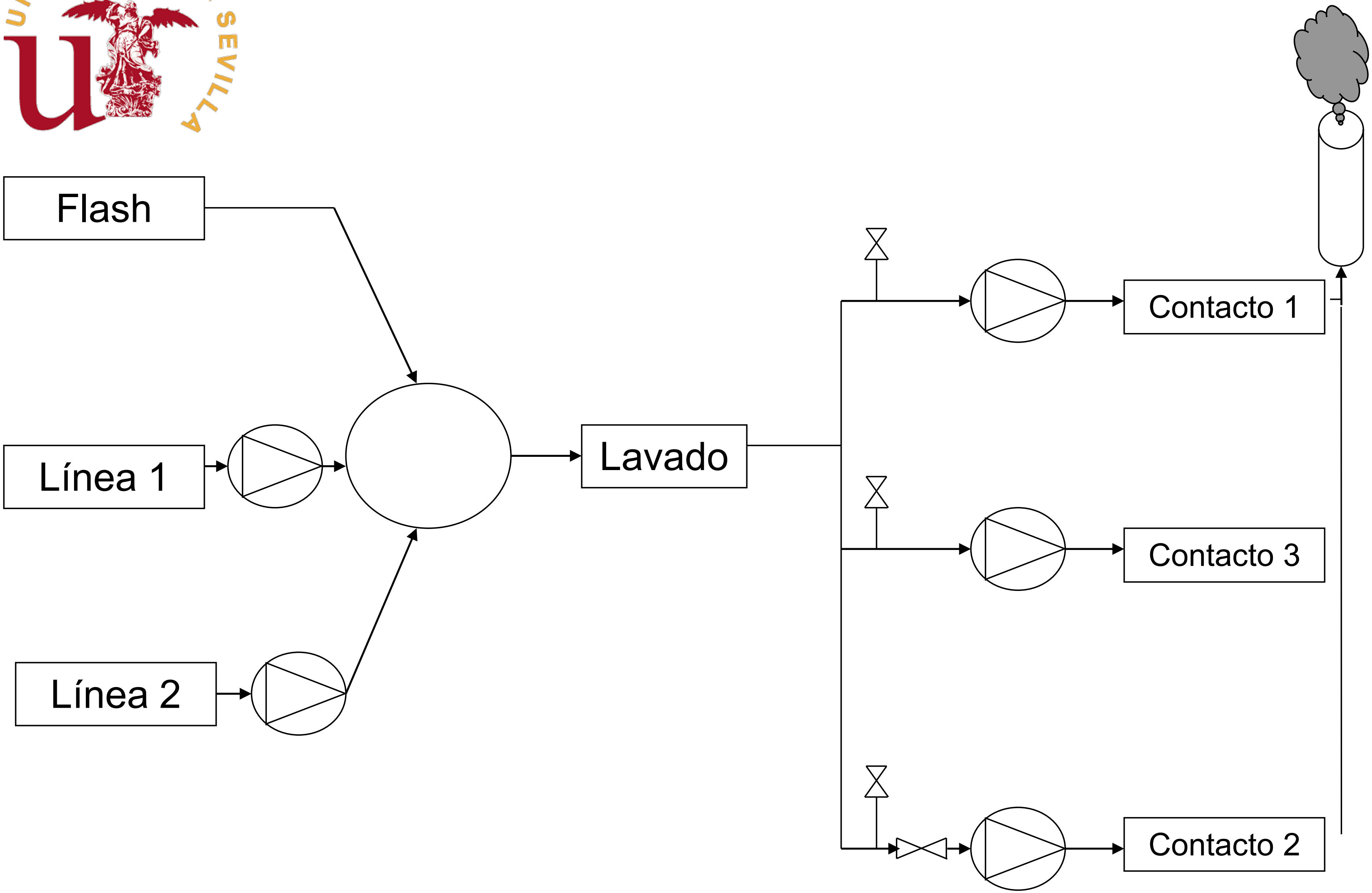
- MPC is Most general way of posing the control problem in the time domain:
  - Optimal control
  - Stochastic control
  - Known references
  - Measurable disturbances
  - Multivariable
  - Dead time
  - Constraints
  - Uncertainties

Real reason of success: Economics

- MPC can be used to optimize operating points (economic objectives). Optimum usually at the intersection of a set of constraints.
- Obtaining smaller variance and taking constraints into account allow to operate closer to constraints (and optimum).
- Repsol reported 2-6 months payback periods for new MPC applications.



ESQUEMA GENERAL CIRCUITO DE GASES



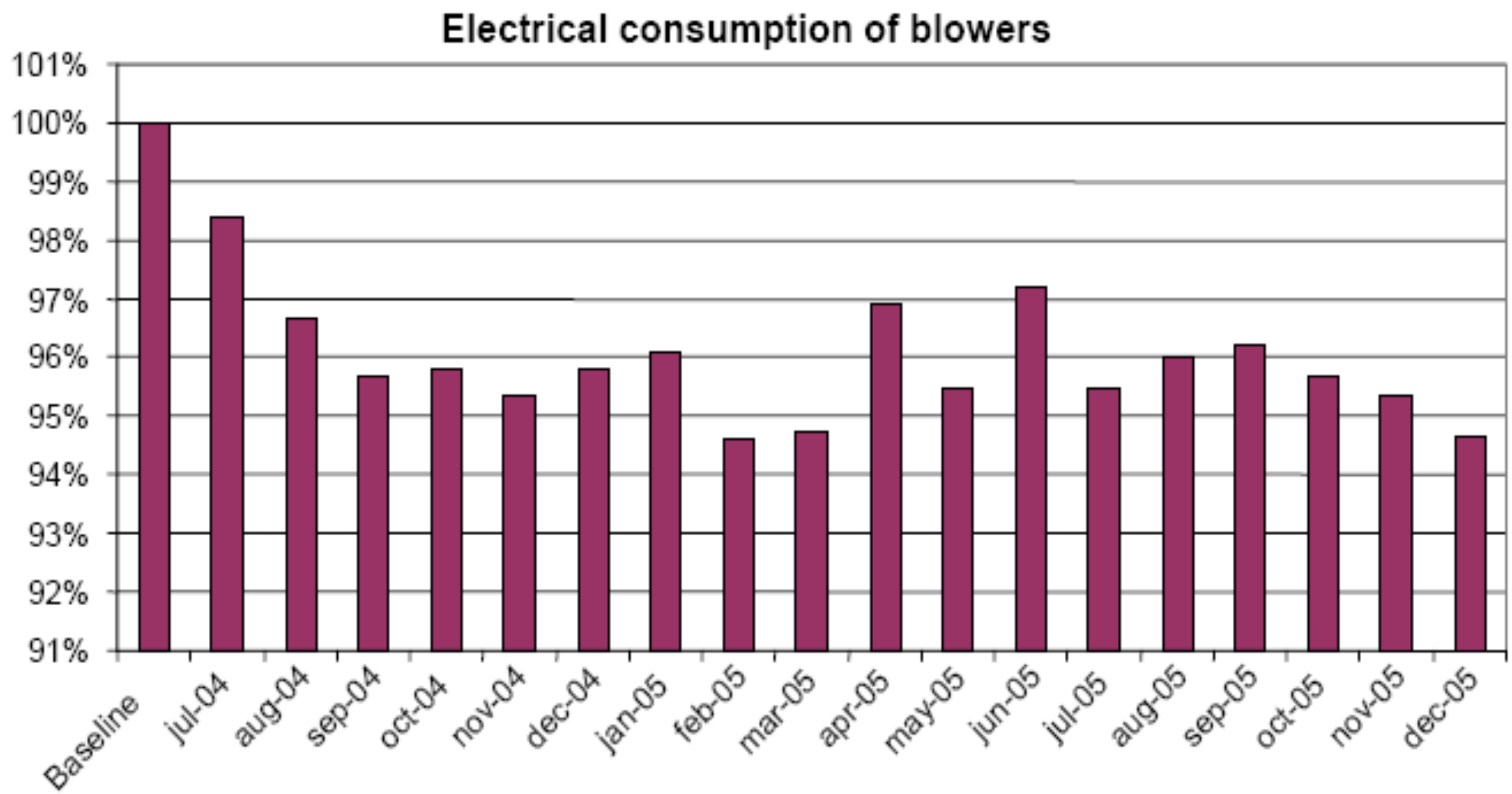


Fig. 14. Electrical consumption reduction.

Benefits

- Yearly saving of more that 1900 MWh
- Standard deviation of the mixing chamber pressure reduced from 0.94 to 0.66 mm water column.
- Operator's supervisory effort: percentage of time operating in auto mode raised from 27% to 84%.

Outline

- A little bit of history
- Model Predictive Control concepts
- Linear MPC
- Multivariable
- Constraints

Bibliografía

- R. Bitmead, Gevers M. and Werts W. “Adaptive Optimal Control: The Thinking man GPC”, Prentice Hall, 1990
- R. Soeteboek, Predictive Control: A Unified Approach, Prentice Hall, 1991.
- E.F. Camacho and C. Bordons, “Model Predictive Control in the Process Industry”, Springer-Verlag, 1995.
- J. Maciejowski, “Predictive Control with constraints”, Prentice Hall, 2002.
- A. Rossiter, “Model-Based Predictive Control: A Practical Approach”, CRC Press, 2003
- E.F. Camacho and C. Bordons, “Model Predictive Control”, Springer-Verlag, 1999 (Second edition 2004)

↳ 7412 Citations (Google Scholar)

MPC Objetive

- Compute at each time instant the sequence of future control moves that will make the future predicted controlled variables to best follow the reference over a finite horizon and taking into account the control effort.

- Only the first element of the sequence is used and the computation is done again at the next sampling time.

MPC basic concepts

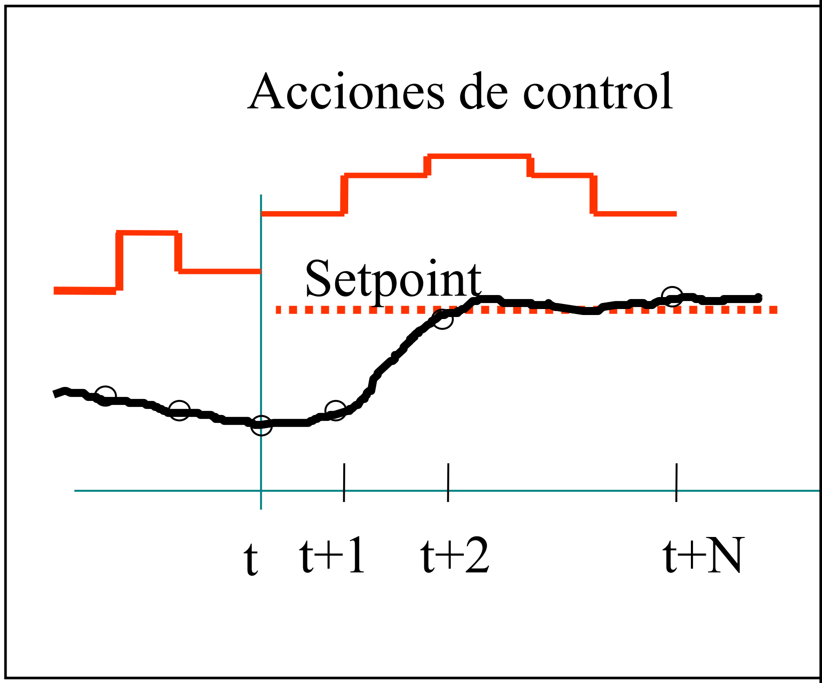
- Common ideas:
  - Explicit use of a model to predict output.
  - Compute the control moves minimizing an objective fuction.
  - Receding horizon strategy.
- The algorithms mainly differ in the type of model and objective function used.

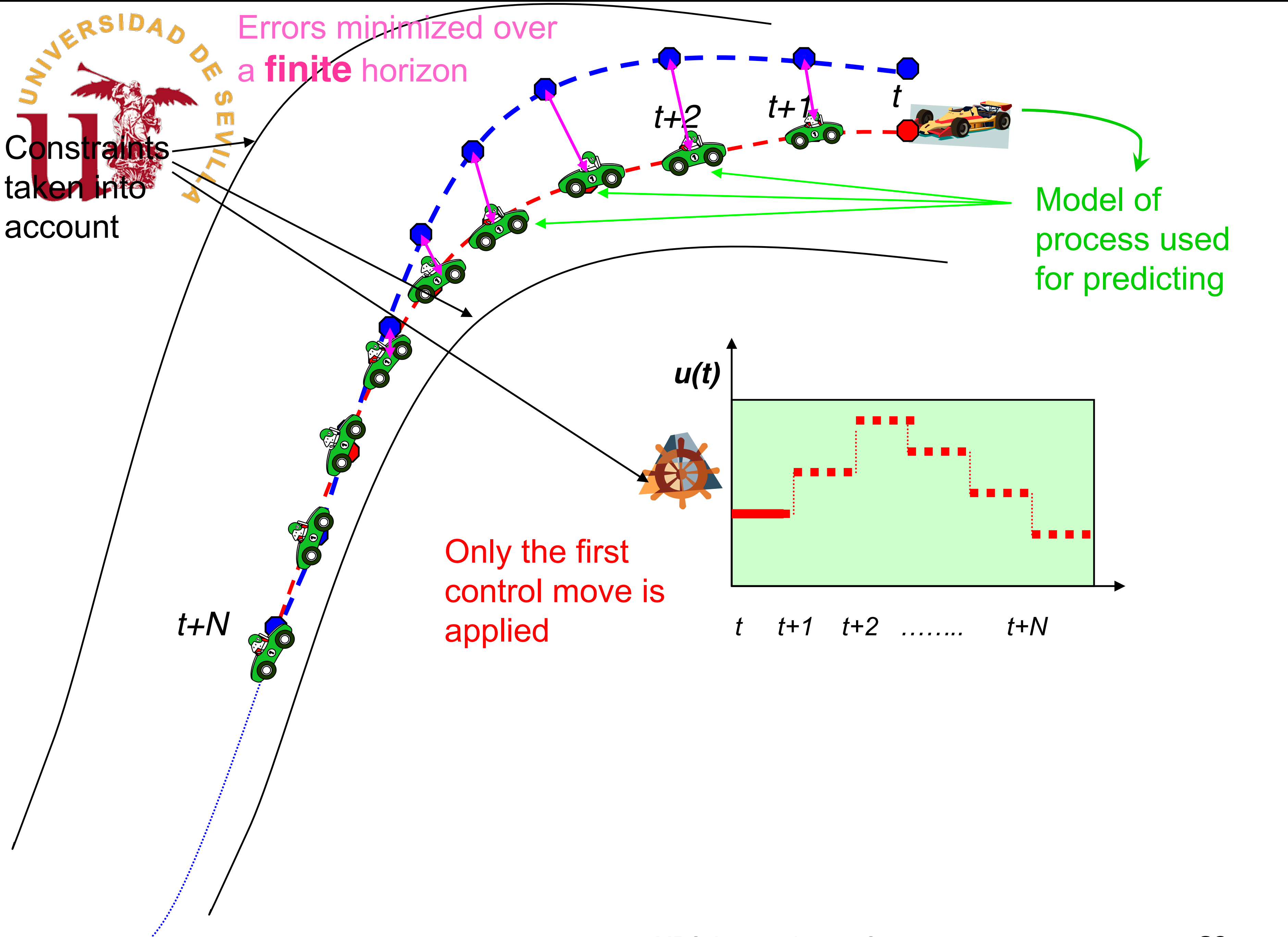
MPC strategy

- At sampling time *t* *the future control sequence is compute so that the future sequence of predicted output y(t+k/t) along a horizon N follows the future references as best as possible.*

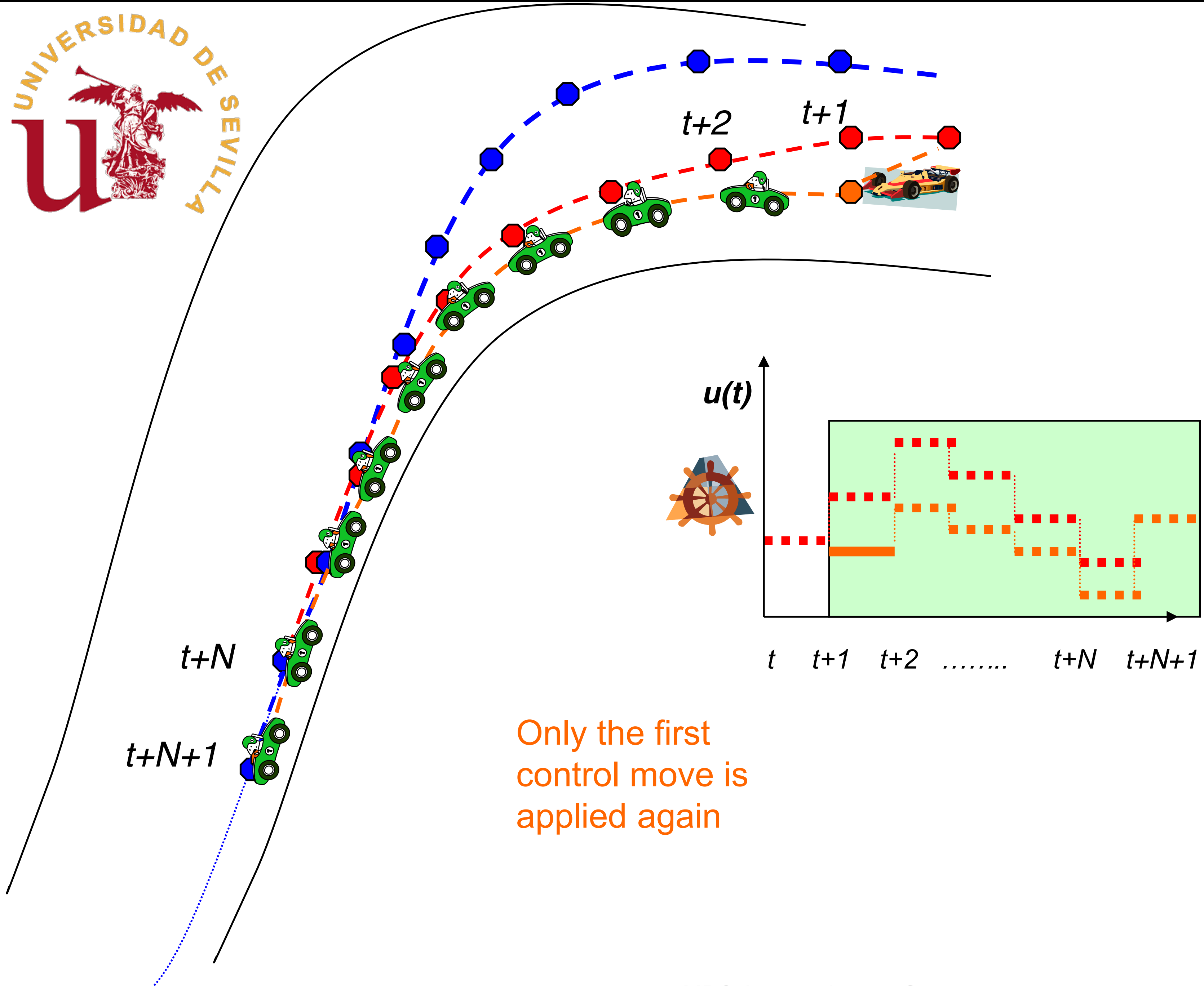
- The first control signal is used and the rest disregarded.

- The process is repeated at the next sampling instant *t+1*



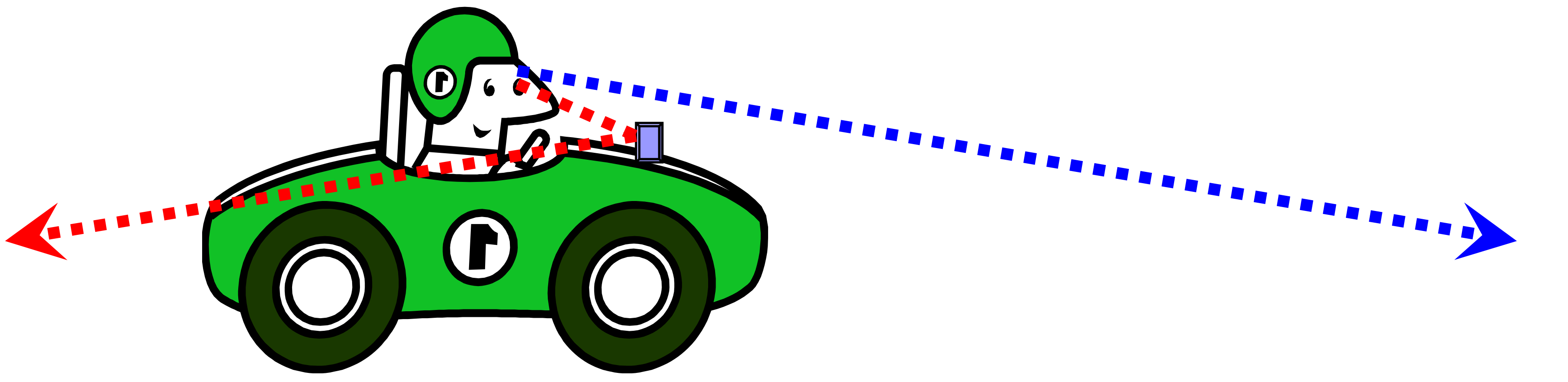


Errors minimized over
a **finite** horizon
Constraints taken into account
t+2
t+1
t
Model of process used for predicting
u(t)
Only the first control move is applied
t+N
t
t+1
t+2
……..
t+N



Only the first control move is applied again

MPC vs. PID



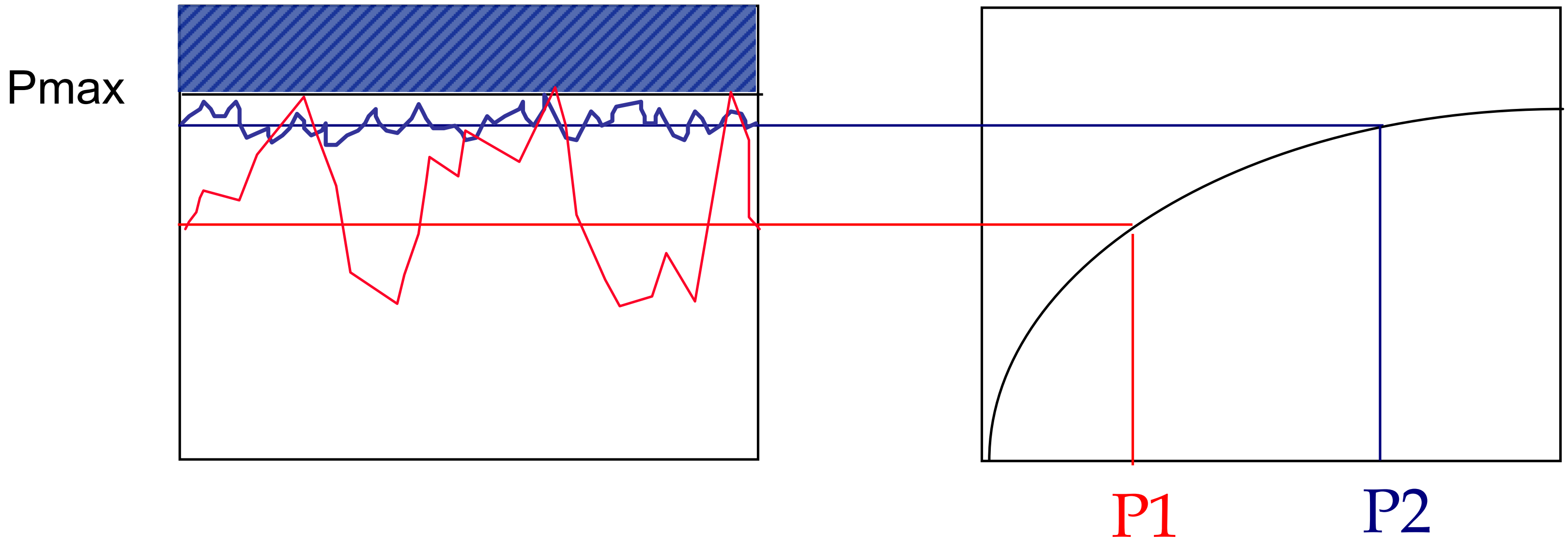
PID: $u(t)=u(t-1)+g_0\, e(t) + g_1\, e(t-1) + g_2\, e(t-2)$

Constraints in process control

- All process are constrained
- **Actuators** have a limited range and slew rate
- **Safety limits**: maximun pressure or temperature
- **Tecnological or quality requirements**
- **Enviromental legislation**

Real reason of success: Economics

- MPC can be used to optimize operating points (economic objectives). Optimum usually at the intersection of a set of constraints.
- Obtaining smaller variance and taking constraints into account allow to operate closer to constraints (and optimum).
- Repsol reported 2-6 months payback periods for new MPC applications.

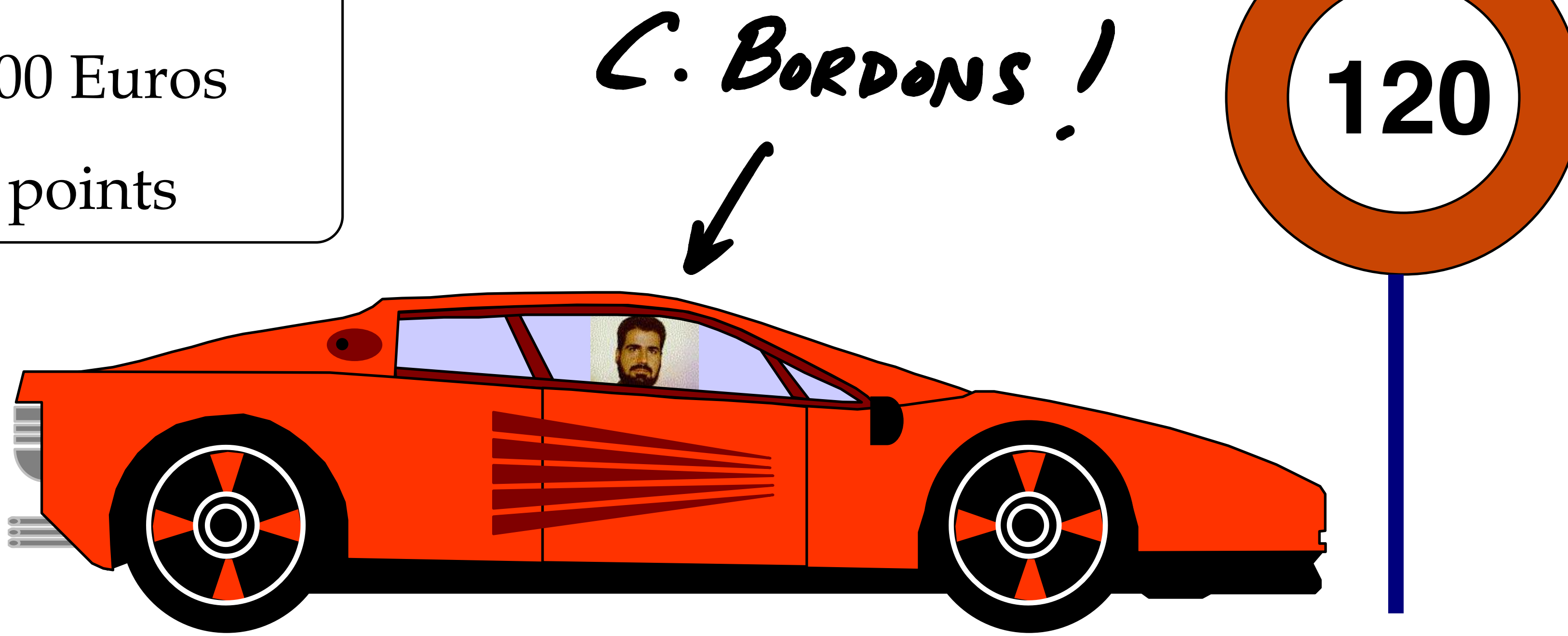


Work close to the optimal but not violating it

<u>Fine</u>

400 Euros

3 points



Control predictivo lineal

MODEL	COST FUNCTION	CONSTRAINTS	SOLUTION
Linear	Quadratic	None	Explicit
Linear	Quadratic	Linear	QP
Linear	Norm-1	Linear	LP

Constraints formulation

- Input constraints:
  - Amplitude in u
  - Slew-rate in u

In matrix form

$$\underline{U} \le u(t) \le \overline{U}$$

$$\mathbf{1}\underline{U} \le T\mathbf{u} + u(t-1)\mathbf{1} \le \mathbf{1}\overline{U}$$

$$\underline{u} \le u(t) - u(t-1) \le \overline{u}$$

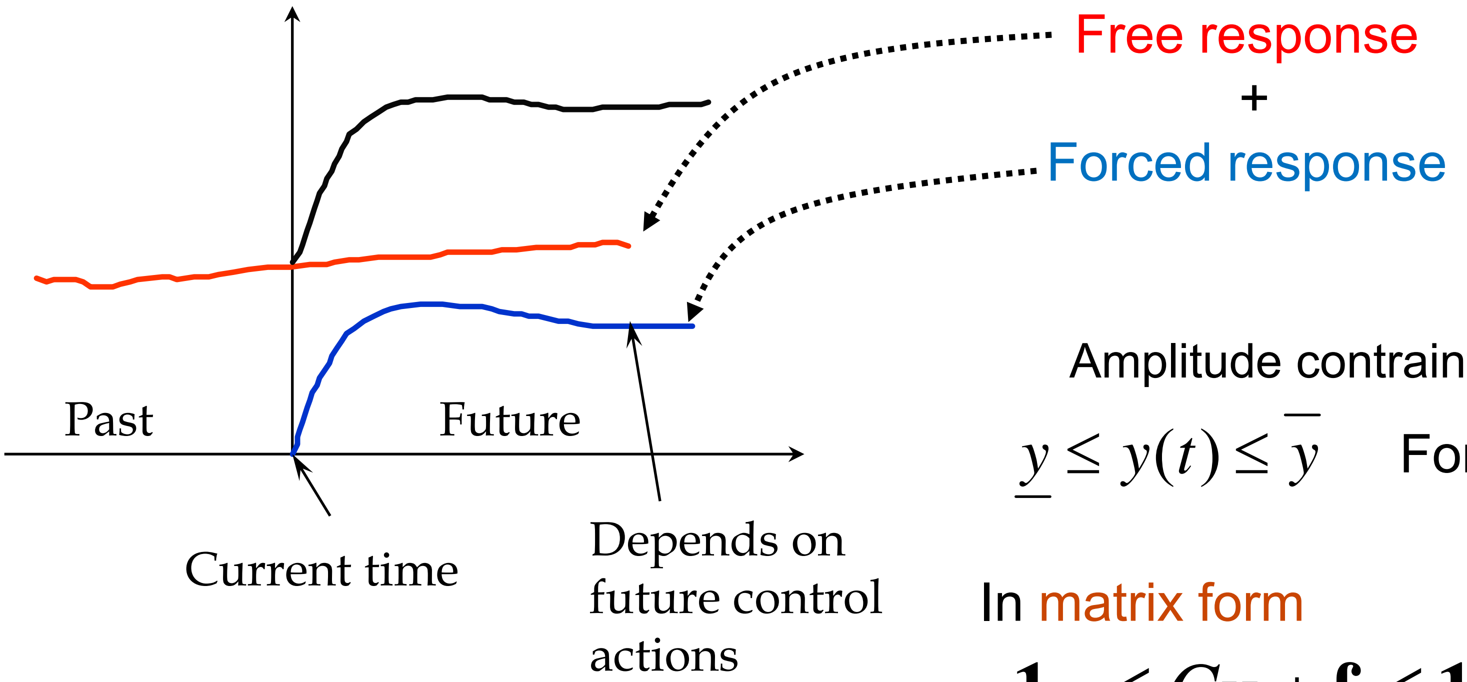
$$\mathbf{1}\underline{u} \le \mathbf{u} \le \mathbf{1}\overline{u}$$

For all t

Output constraints

- Output constraints must be expressed as functions of u using the **prediction equations**
- The prediction is computed as:

$$\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{f}$$



Amplitude contraints:

$$\underline{y} \leq y(t) \leq \overline{y} \quad \text{For all } t$$

In matrix form

$$\mathbf{1}\underline{y} \leq G\mathbf{u} + \mathbf{f} \leq \mathbf{1}\overline{y}$$

Constraints general form

Notice that these constraints are inequalities involving vector ***u*** (increment of the manipulated variables) and can be written in compact form as

$$Ru \leq c$$

with the following matrix and vector:

$$R = \begin{bmatrix} I_{N\times N} \\ -I_{N\times N} \\ T \\ -T \\ \mathbf{G} \\ -\mathbf{G} \end{bmatrix} \qquad \mathbf{c} = \begin{bmatrix} \mathbf{1}\,\overline{u} \\ -\mathbf{1}\,\underline{u} \\ \mathbf{1}\,\overline{U} - \mathbf{1}u(t-1) \\ -\mathbf{1}\,\underline{U} + \mathbf{1}u(t-1) \\ \mathbf{1}\,\overline{y} - \mathbf{f} \\ -\mathbf{1}\,\underline{y} + \mathbf{f} \end{bmatrix}$$

Formulation

All the constraints shown (except the dead zone) are inequalities depending on ***u*** that can be described in matrix form by

$$\mathbf{R}\mathbf{u} \leq \mathbf{r} + \mathbf{V}\mathbf{z}$$

where ***z*** is a vector composed of present and past signals. It is equal to the **current state** if a state-space representations if used, or composed of current output and past input and outputs in CARIMA models (a way of representing the state). Therefore:

$$\mathbf{R}\mathbf{u} \leq \mathbf{r} + \mathbf{V}x(t)$$

Decision variable

Depend on process parameters and signal bounds (**not frequent changes**)

State that changes at **every sampling time**

46

Solution

The implementation fo MPC with constraints involves the minimization of a **quadratic cost function** subjet to **linear inequalities**: Quadratic Programming (QP)

minimize $$J(\mathbf{u}) = \frac{1}{2}\,\mathbf{u}^T\mathbf{H}\mathbf{u} + \mathbf{b}\mathbf{u} + \mathbf{f}_0$$

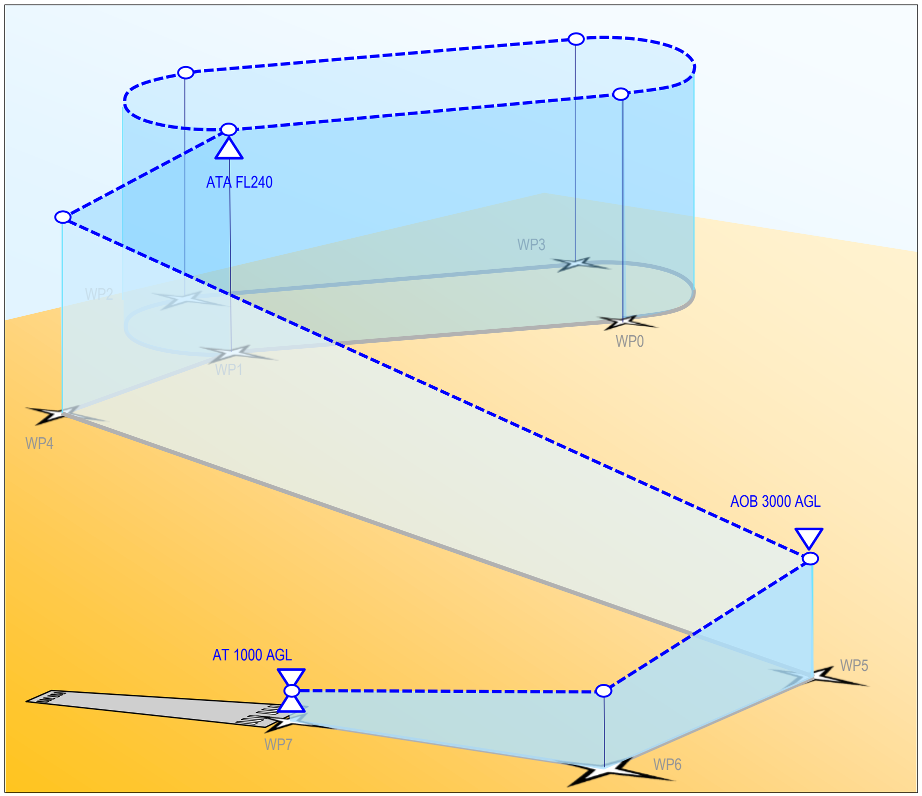
Subject to: $$\mathbf{R}\mathbf{u} \leq \mathbf{r} + \mathbf{V}x(t)$$

There are many reliable QP algorithms

- Active Set methods
- Feasible Direction methods
- Pivoting methods, etc.

All methods use **iterative** algorithms (computation time)

MPC control of UA V (AIDL) trajectories
(Project funded by Boeing)



Conclusions

- Well established in industry and academia
- Great expectations for MPC
- Many contribution from the research community but …
- Many open issues
- Good hunting ground for PhD students.

Outline

1. ~~Introduction to MPC (Slides by E.F. Camacho)~~
2. Application to Spacecraft Rendezvous (including PWM) ←
3. Rendezvous + Attitude Control
4. Soft Landing on an Asteroid
5. Guidance for UAVs

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

Rafael Vázquez

Francisco Gavilán   Eduardo F. Camacho

Universidad de Sevilla

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 $k = 0, \dots, N_p$. 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.

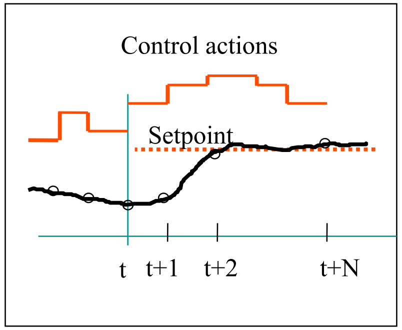
- Given inequality constraints

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

  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 $\rightarrow$ 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!

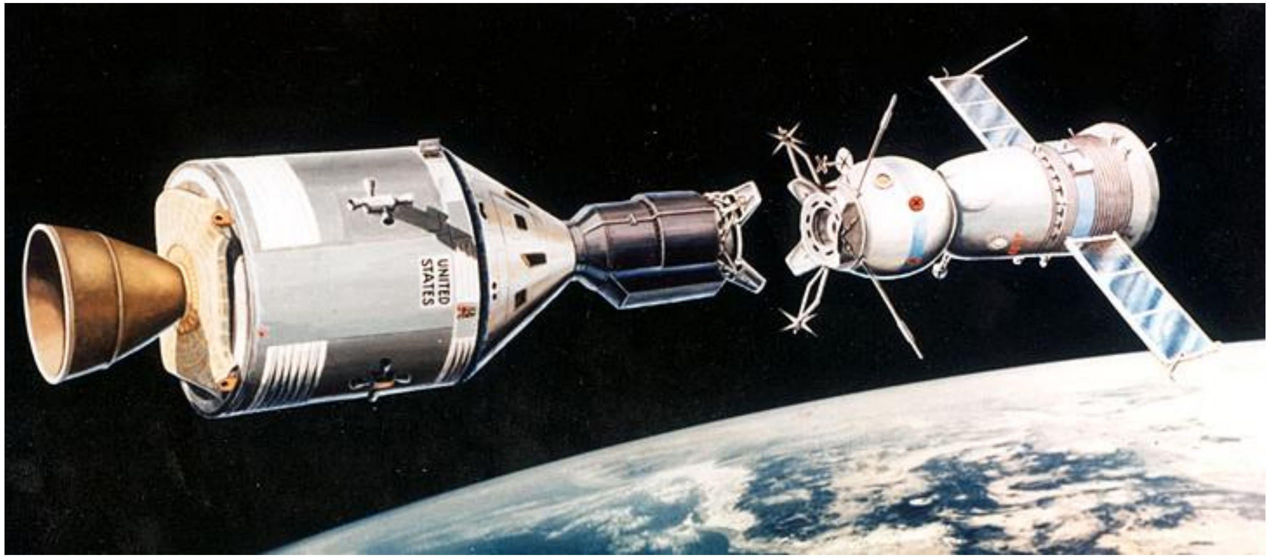


Advantages and Disadvantages of MPC

- 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.

Introduction

- For spacecraft, "rendezvous" is the **controlled** close encounter of two (or more) space vehicles.

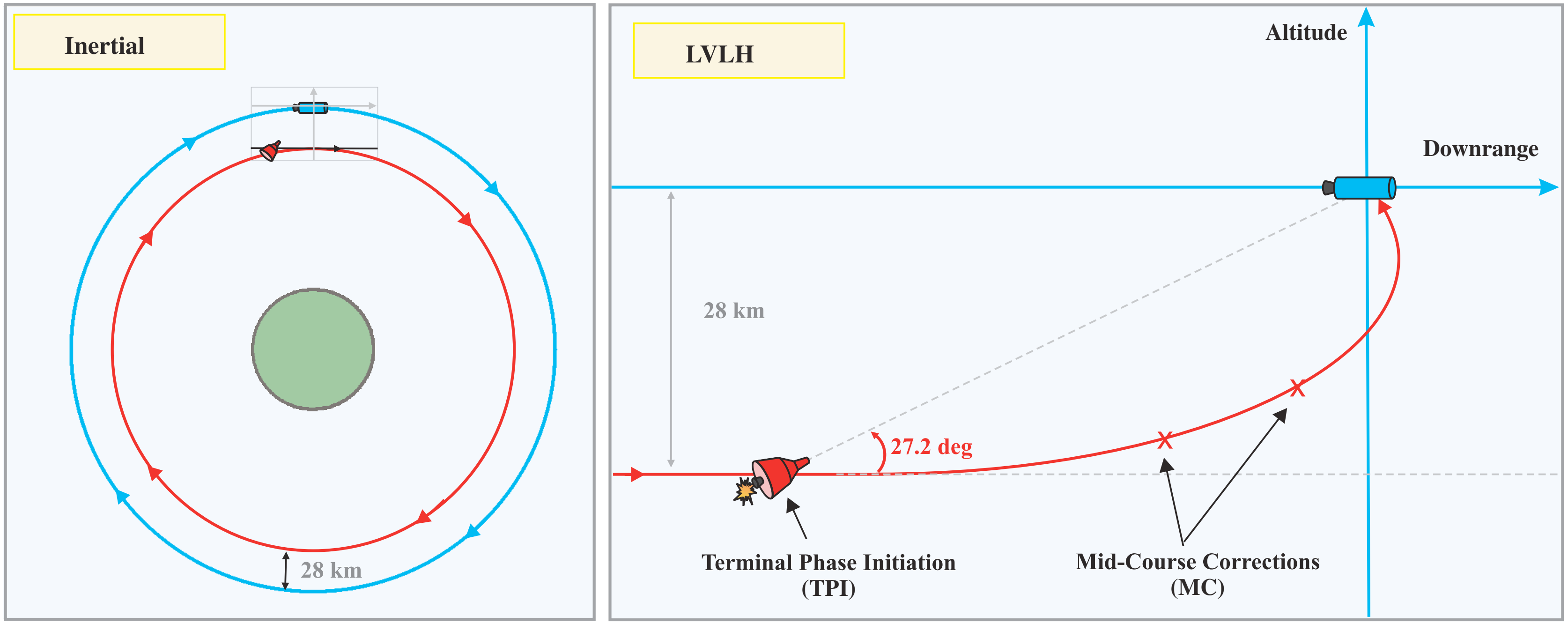


- Rendezvous between Apollo and Soyuz in 1975. First joint US/Soviet space flight mission. Docked during two days.

Introduction

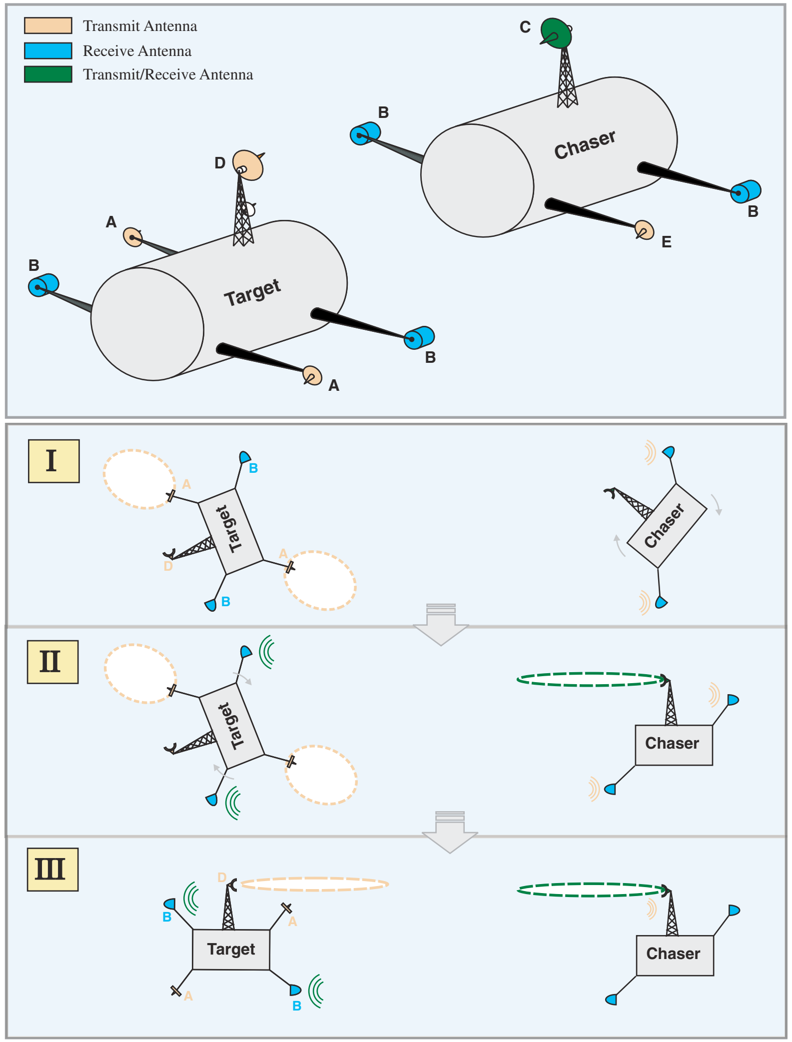
- We will consider the most usual case: two vehicles.
- One of the spacecraft is the "**target vehicle**" or just "**target**". Known orbit. It is considered passive.
- The other is the "**chaser spacecraft**" or just "**chaser**". Begins from a known position and maneuvers to target.
- Rendezvous must be done in a controlled fashion:
  - Control in position, to get the chaser close in position to the target.
  - Control in velocity, to get the chaser close in velocity to the target.
- Rendezvous and interception:
  - Rendezvous: as above.
  - Interception: Only looks to get close in position. Velocity can be different. Impact can be an objective (e.g. a missile).
- Both problems are studied using similar techniques.

Gemini: The first rendezvous mission



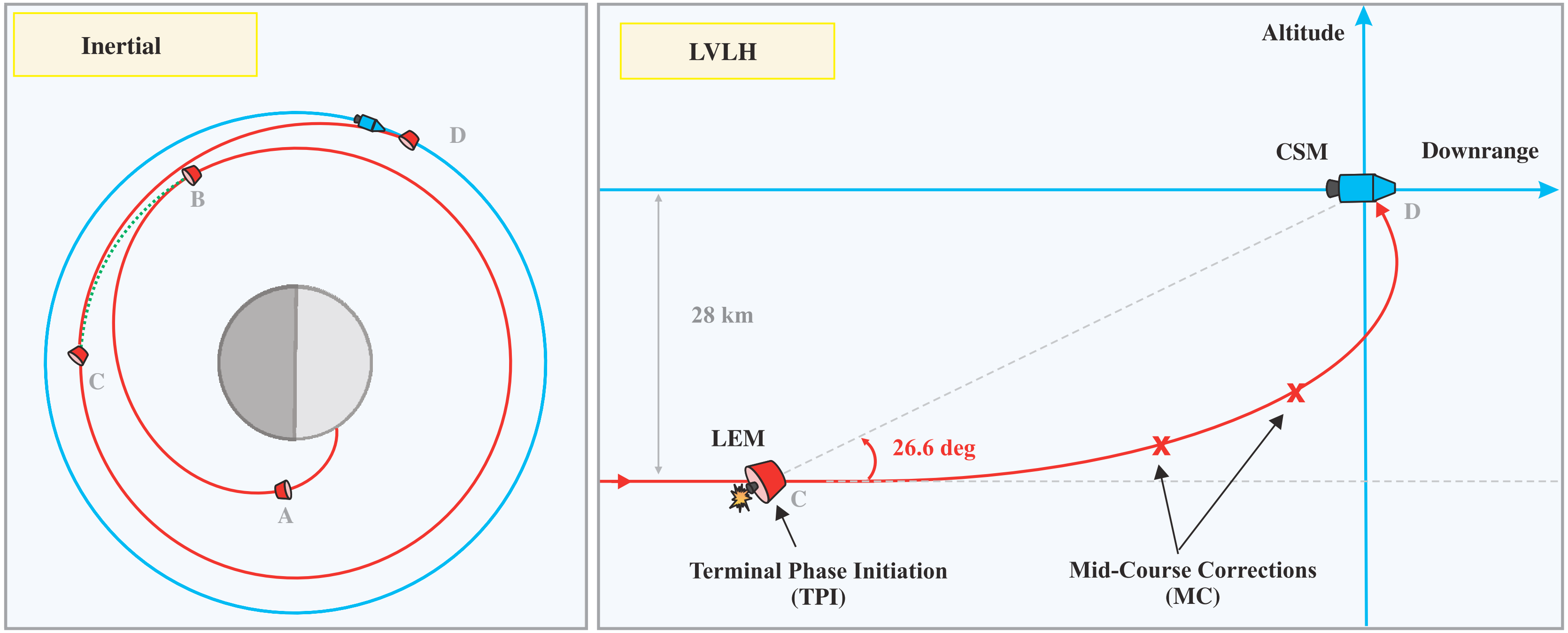
- Gemini missions (US) tested rendezvous technology in 1965.
- Rendezvous was performed **manually** by the astronauts on board the spacecraft.
- December 15, 1965 was the first succesful orbital rendezvous in history (between Gemini VI and Gemini VII).

Soyuz: the Russian approach



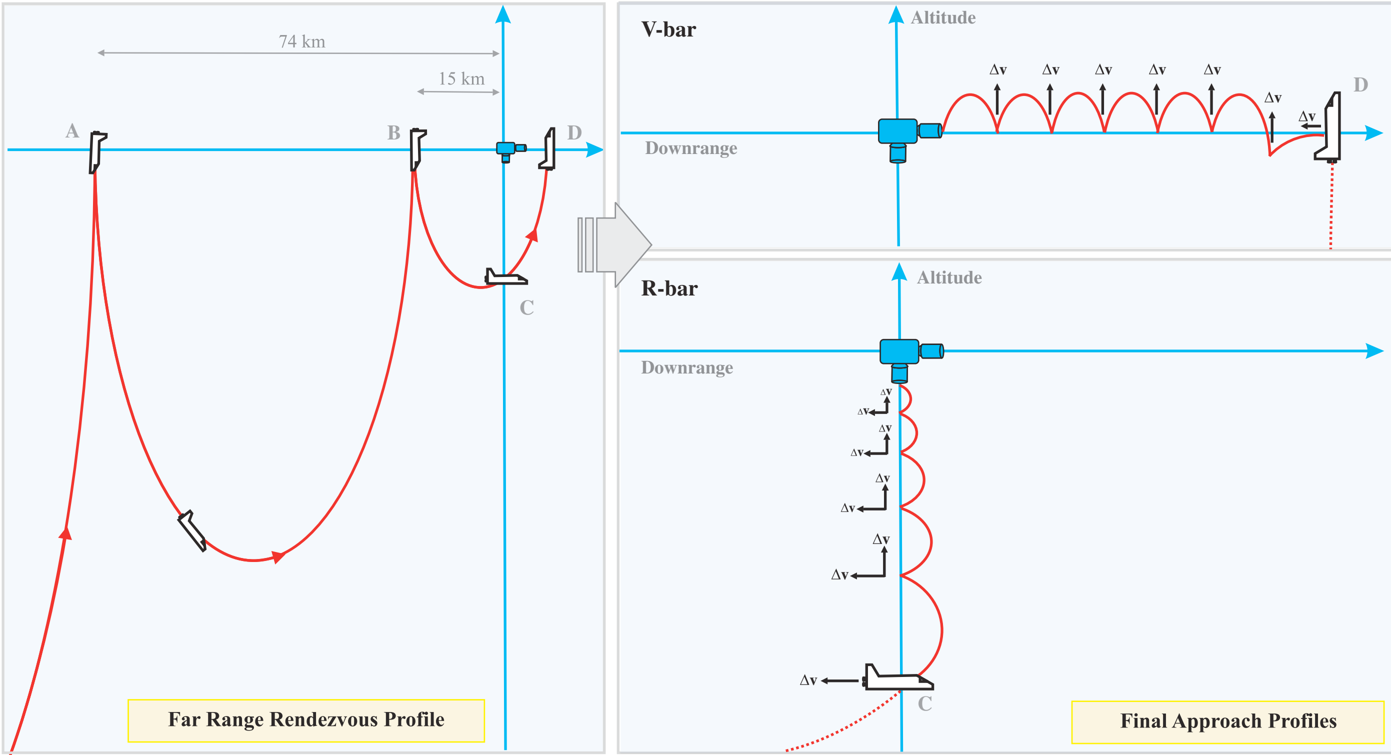
- In 1967 took place the first automated rendezvous between two unmanned space vehicles (two Soyuz spacecraft)
- Much more complex than the American system.
- Based on navigation system communication between the two vehicles, using several antennas they could obtain relative position, velocity and attitude.
- Requires a cooperative target.

Rendezvous in the Apollo mission



- For the American space program, the Apollo missions were the main reason to obtain rendezvous capacity.
- A critical stage in the mission to the Moon was the rendezvous between the Command Module and the Lunar Module. Performed manually (trained with simulator).

The Space Shuttle



- Profile of a rendezvous between the ISS and the Space Shuttle. Two options: V-bar approach and R-bar approach.
- The final phase is still manually performed!

Modern Russian rendezvous systems

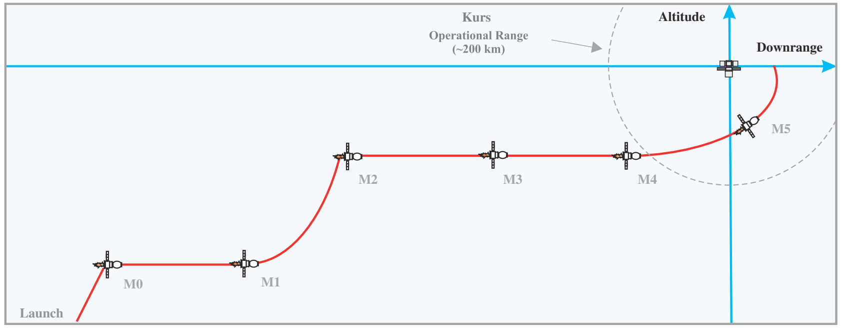
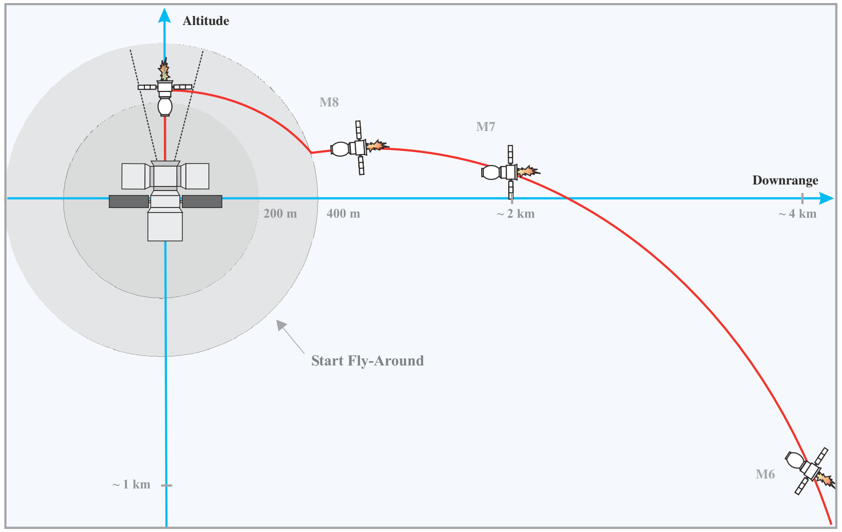
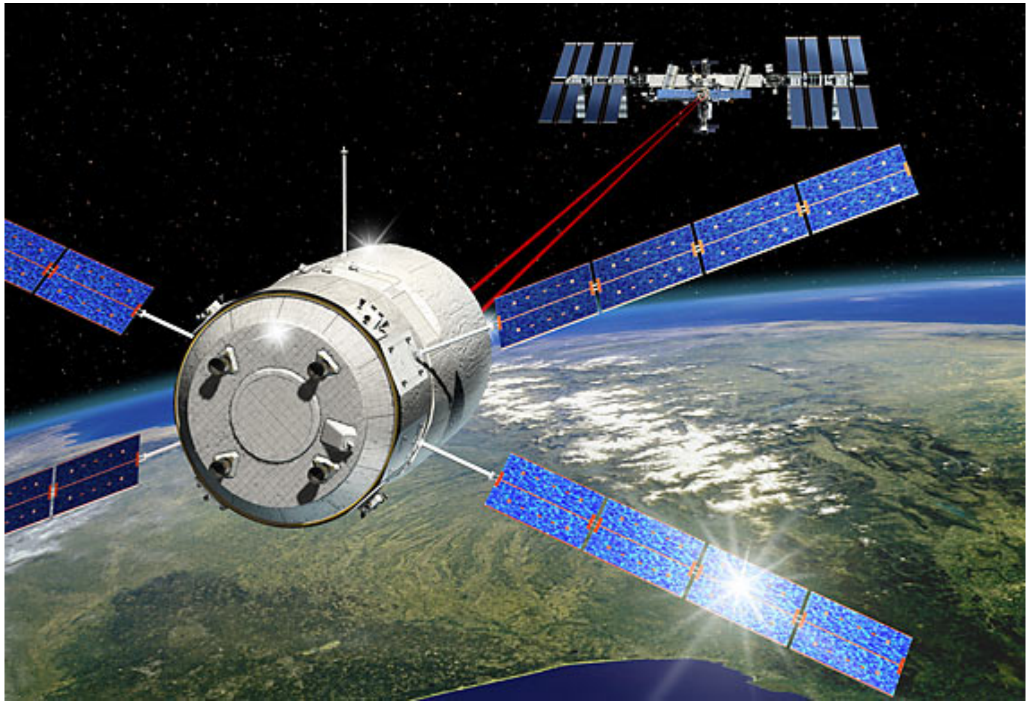


Fig. 6 Phasing and rendezvous sequence for Soyuz/Progress vehicles.



- The Russians developed the Kurs (course) system which allowed rendezvous between Soyuz and MIR.
- Also automatic but more precise and with more range than their older system.
- Does not require target cooperation.
- However, it weights a lot (85 kg.) and requires about 270 Watt (similarly in the target side).

What about Europe?



- ATV (Automated Transfer Vehicle) incorporates automated rendezvous capability with the ISS. Operative since 2008.
- Developed by EADS/Astrium.
- Does not require target cooperation, however uses specific equipment on both sides.

Rendezvous segments

- Typically, rendezvous problems are divided in several phases:
  1. **Orbital phase**: The chaser begins on Earth or in a different orbit from the target. Launch and orbital maneuvers have to be performed to approach the orbit of the target.
  2. **Far range rendezvous**: The chaser is “close” to the target ($\sim 10 - 100$ km), and must approach it ($\sim 100 - 1000$ m). Typically relative navigation is used.
  3. **Close range rendezvous**: Maneuvers are performed to get the target very close to the target (about 1 meter or less, relative speeds of cm/s). This is the phase considered in this talk.
  4. **Docking/berthing**: Smooth capture is performed followed by structural union among the spacecraft. Also an interesting control problem!!
- A good general reference for rendezvous: Fehse, W. (2003). *Automated Rendezvous and Docking of Spacecraft*.

(Close range) Rendezvous Model

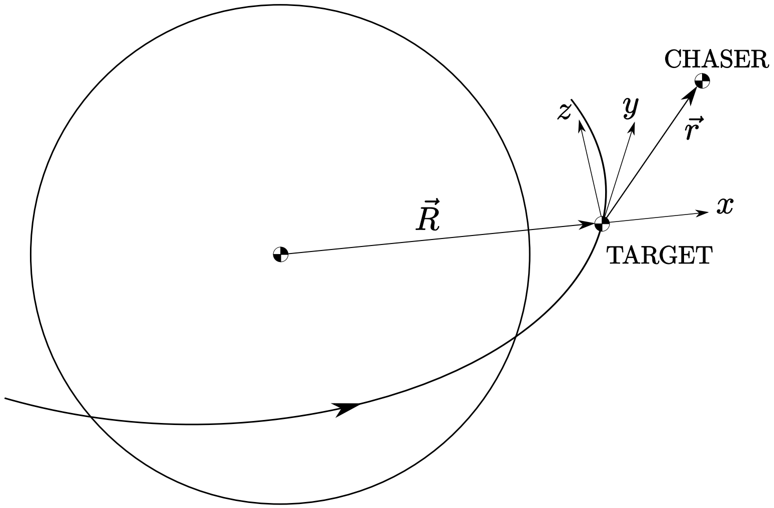
- There are many rendezvous models for spacecraft, according to which orbital perturbation model is used and the orbit of the target.
- The simplest possible case:
  - the target follows a circular keplerian orbit (i.e. zero eccentricity) around a central body (tipically the Earth).
  - the target is passive (does not perform maneuvers).
  - the chaser is very close (less than 1 kilometer).
- Call:
  - $\vec{R}$ vector from central body to target.
  - R: radius of the orbit of the target (given in kilometers).
  - μ: the gravitational parameter of the central body (for the Earth, $\mu = 398600.4\ \mathrm{km}^3/\mathrm{s}^2$).
  - Target mean (angular) velocity is $n = \sqrt{\frac{\mu}{R^3}}$.
  - $\vec{r}$ position of target with respect to chaser.

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.



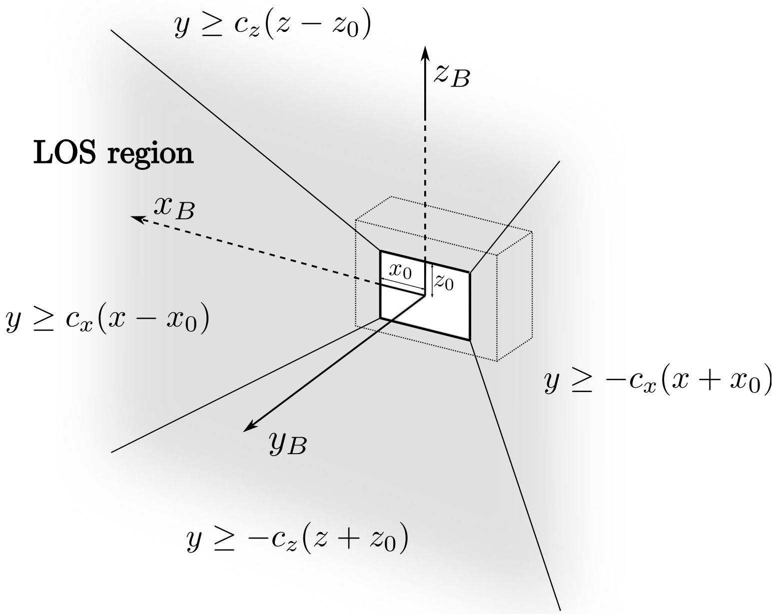
LVLH FRAME

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

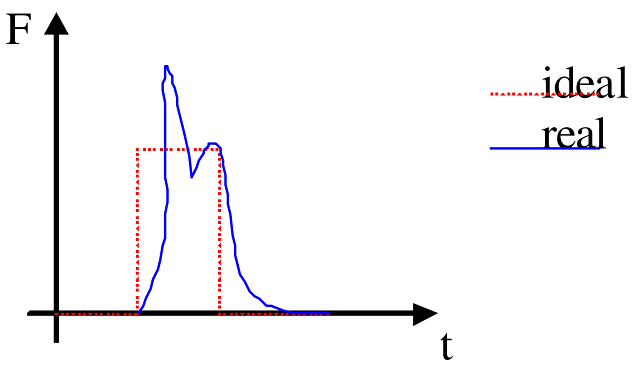
- In this work we will equal the safe zone with the "line of sight" (LOS)



- 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:

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

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

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

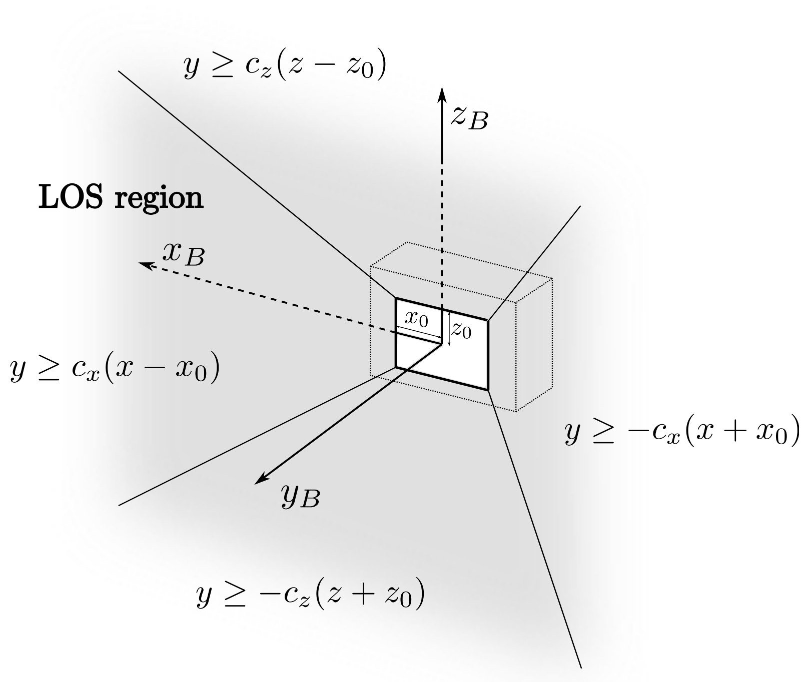
- Hence we can write the prediction equations as:

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

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`.

(I use GUROBI nowadays)

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:

$$\hat{\delta}(k) = \frac{e^{-\lambda}}{\gamma_k}\left(\gamma_{k-1}\hat{\delta}(k-1) + \delta(k-1)\right),$$

$$\hat{\Sigma}(k) = \frac{e^{-\lambda}}{\gamma_k}\Big(\gamma_{k-1}\hat{\Sigma}(k-1) + \left(\delta(k-1) - \hat{\delta}(k)\right)\left(\delta(k-1) - \hat{\delta}(k)\right)^T\Big),$$

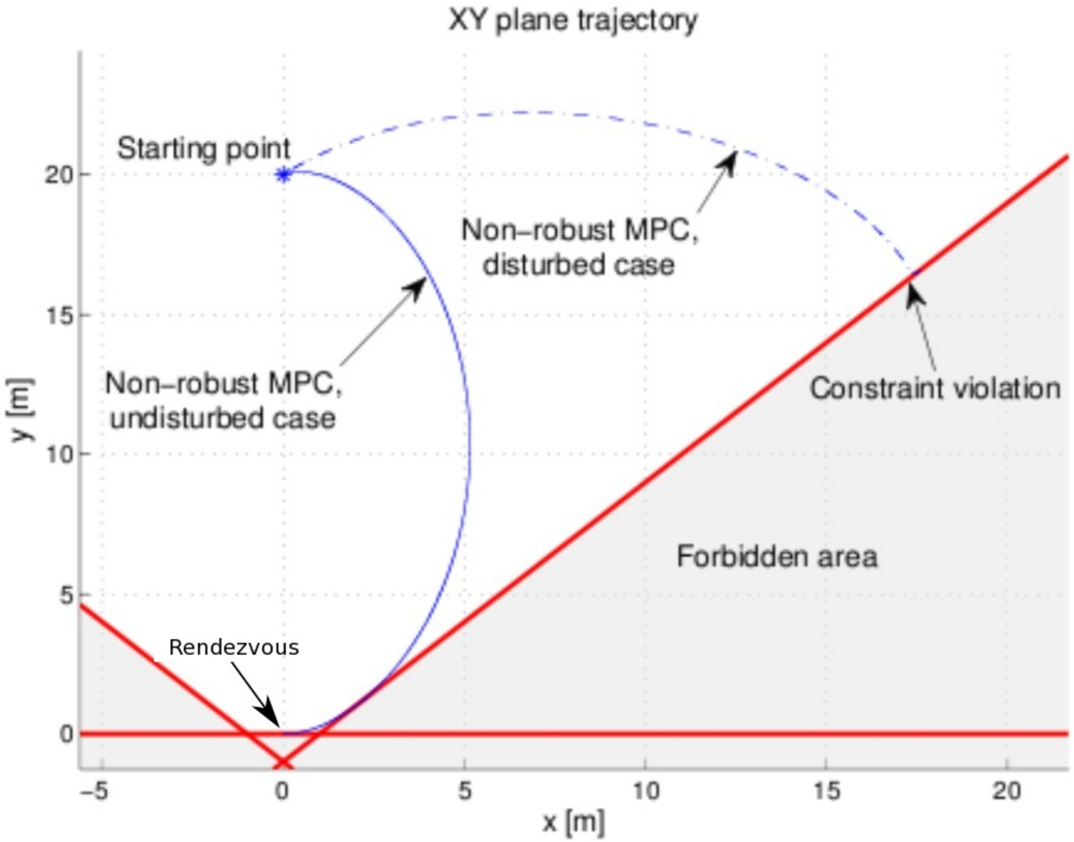
where $\gamma_k = \frac{e^{-\lambda}(1-e^{-\lambda k})}{1-e^{-\lambda}}$

- 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

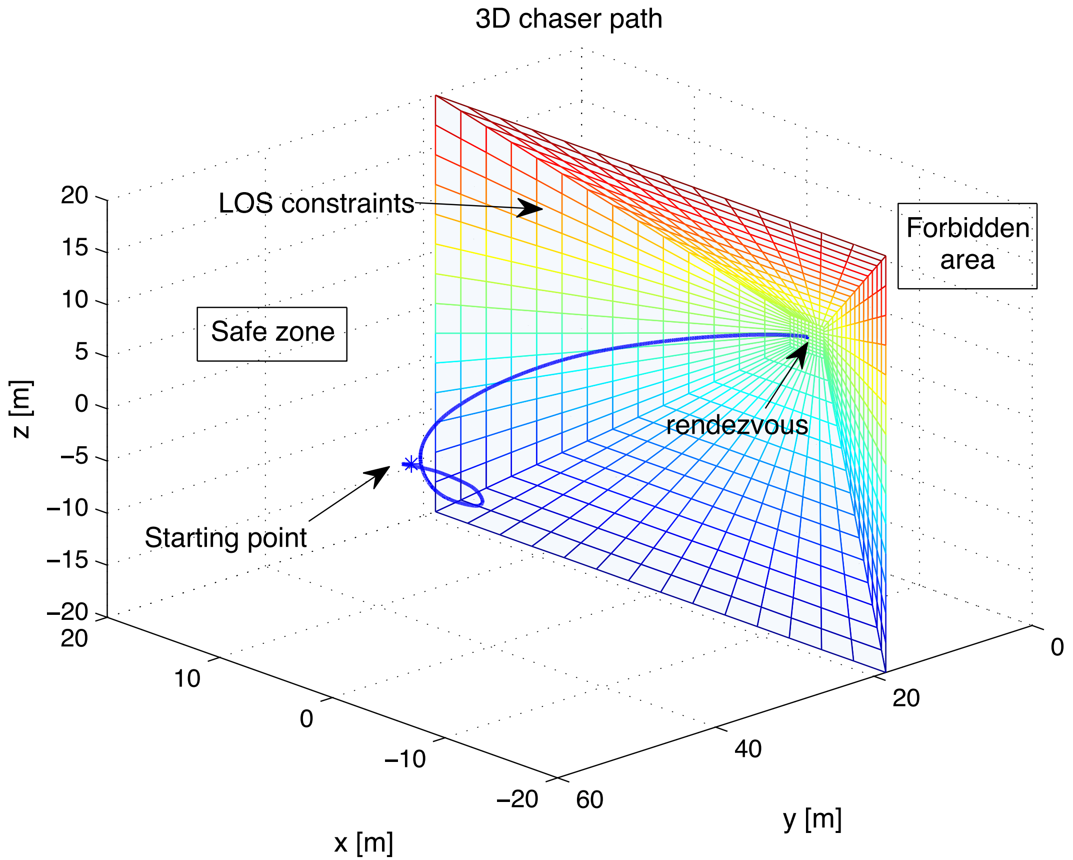
- For numerical simulations, several scenarios have been considered with and without perturbations.
- Parameters used: $R_0 = 6878\,\mathrm{km}$, $n = 1.1068 \cdot 10^{-3}\,\mathrm{rad/s}$, and LOS constraint parameters: $x_0 = z_0 = 1.5\,\mathrm{m}$ and $c_x = c_z = 1$.
- We included propulsive perturbations 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



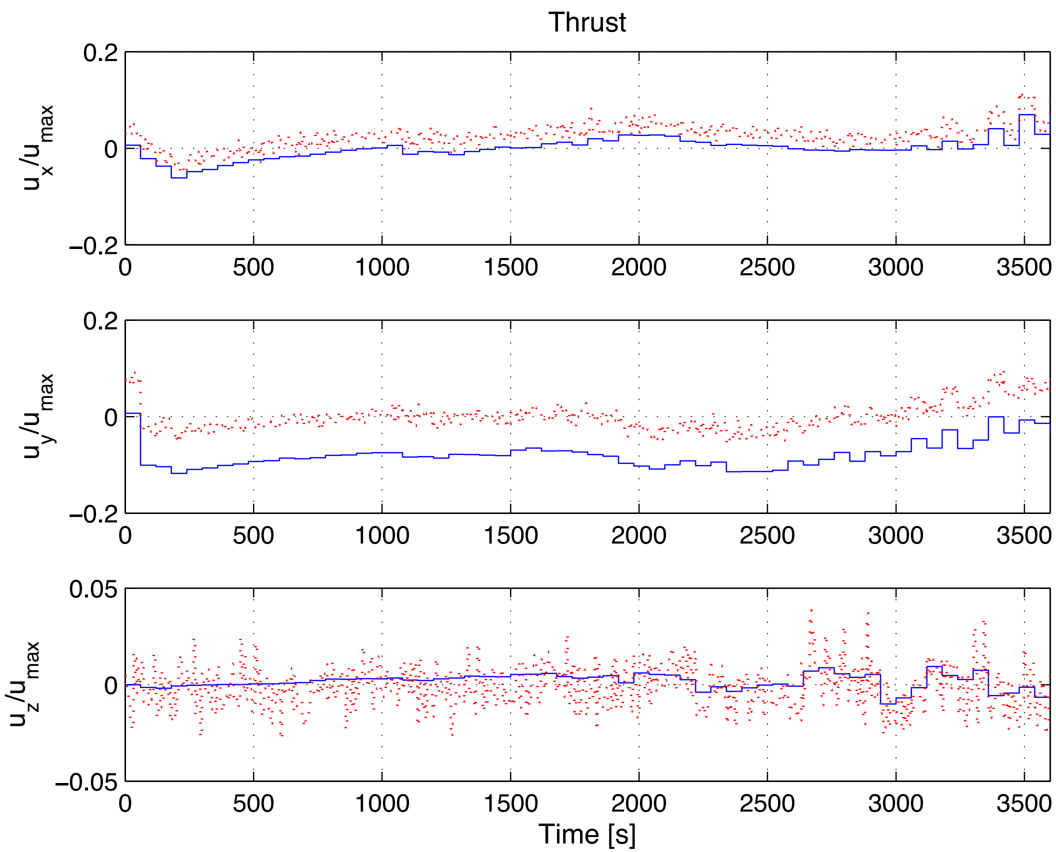
- Good results without perturbations (solid line).
- Fails when perturbations are present (dashed line). However if perturbations are small, still works.

Chance Constrained MPC controller with perturbations



- Includes perturbations. Good results!

Chance Constrained MPC controller with perturbations



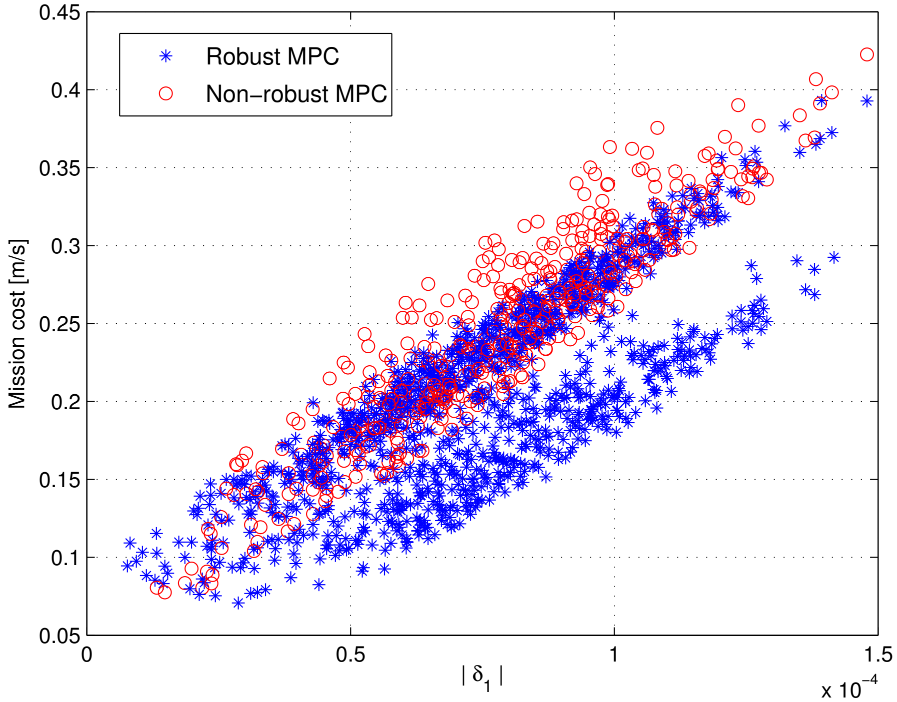
- Commanded control (solid) and applied control (dotted).

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.

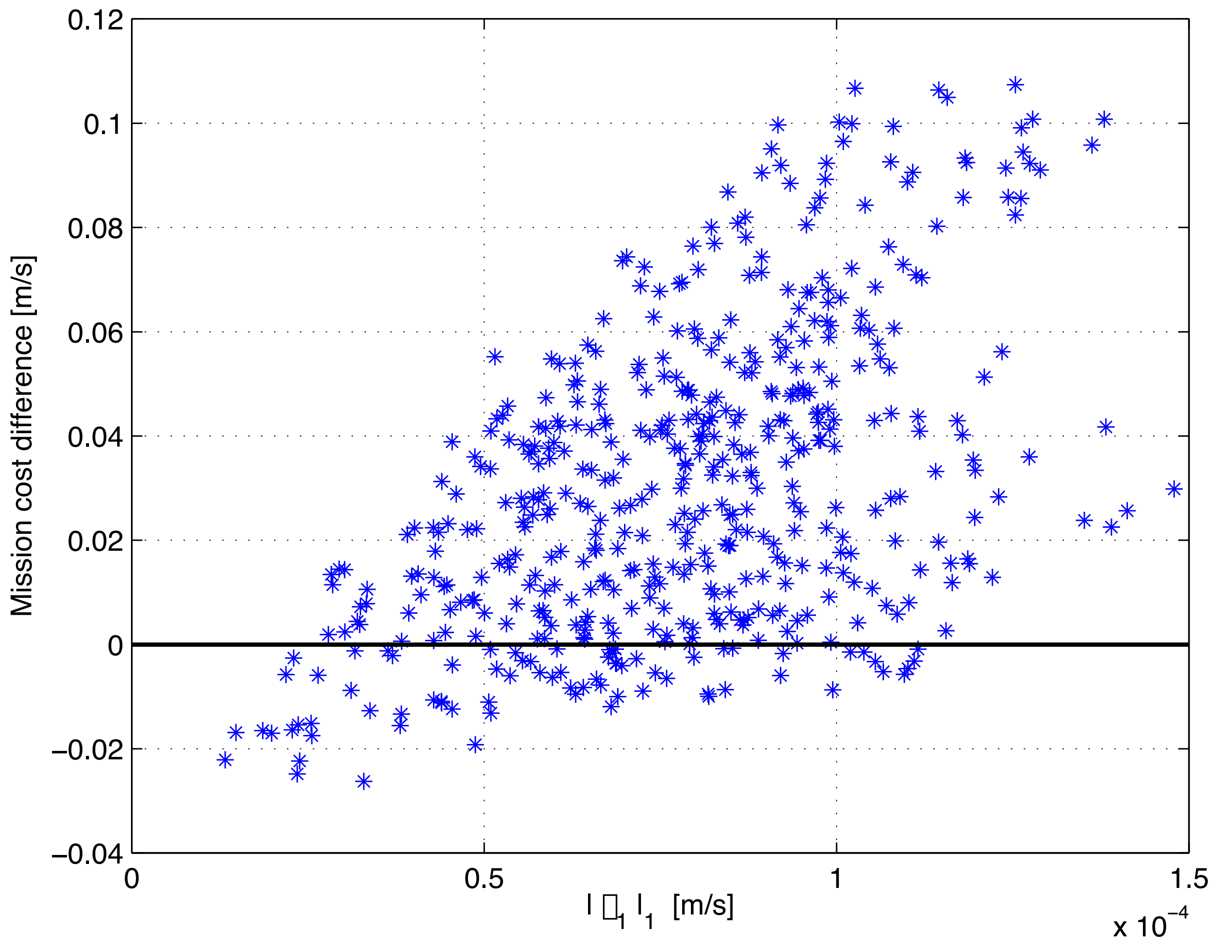
	Non-robust MPC	Robust MPC
Constraint violations	59%	0%
$d \leq 0.2\,\mathrm{m}$	19%	100%
$0.2\,\mathrm{m} \leq d \leq 0.5\,\mathrm{m}$	22%	0%
$0.5\,\mathrm{m} \leq d$	0%	0%
Mean cost (m/s) of successful missions	0.2444	0.2039

Monte Carlo simulations



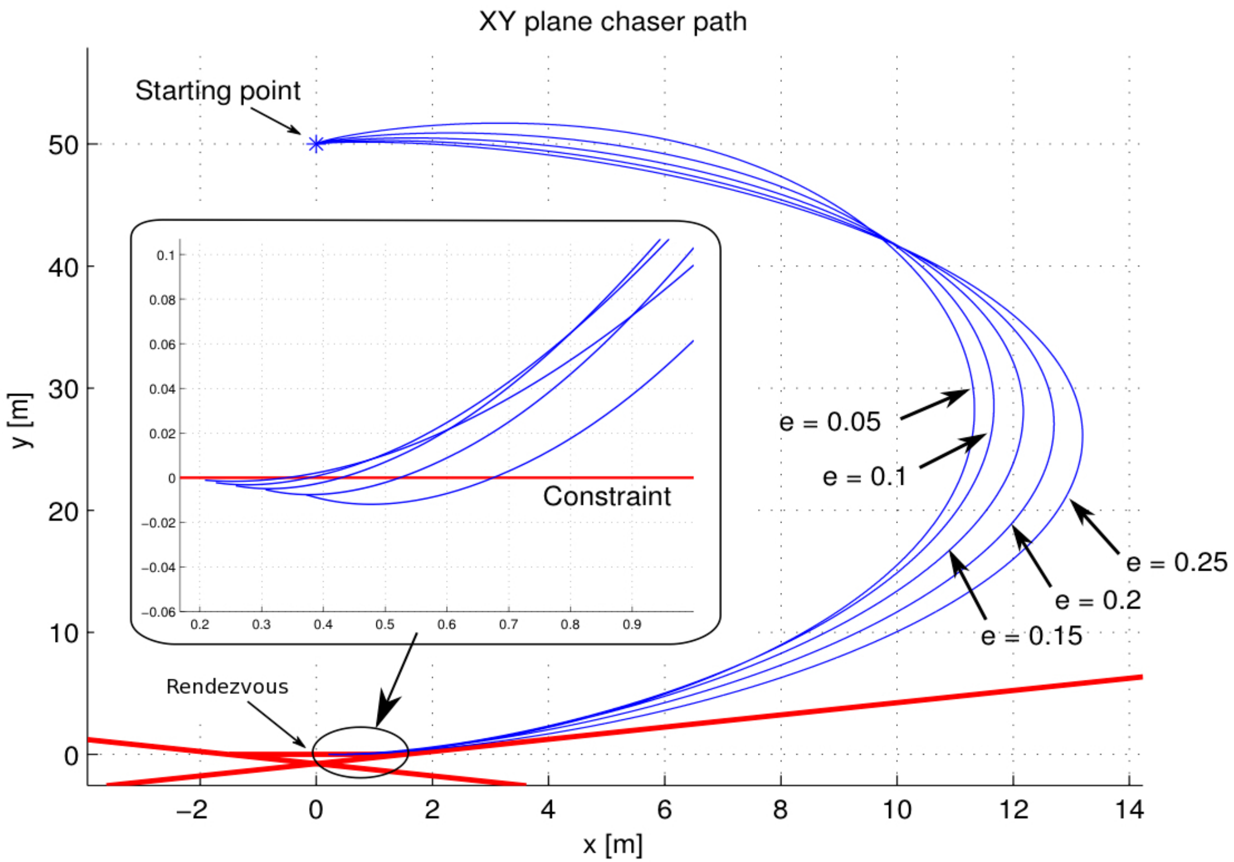
- Plot of total cost of successful missions for both robust and non-robust approach, against L_1 norm of the mean of the disturbances.
- It can be found that using the non-robust controller implies a 15% of cost increment.

Monte Carlo simulations



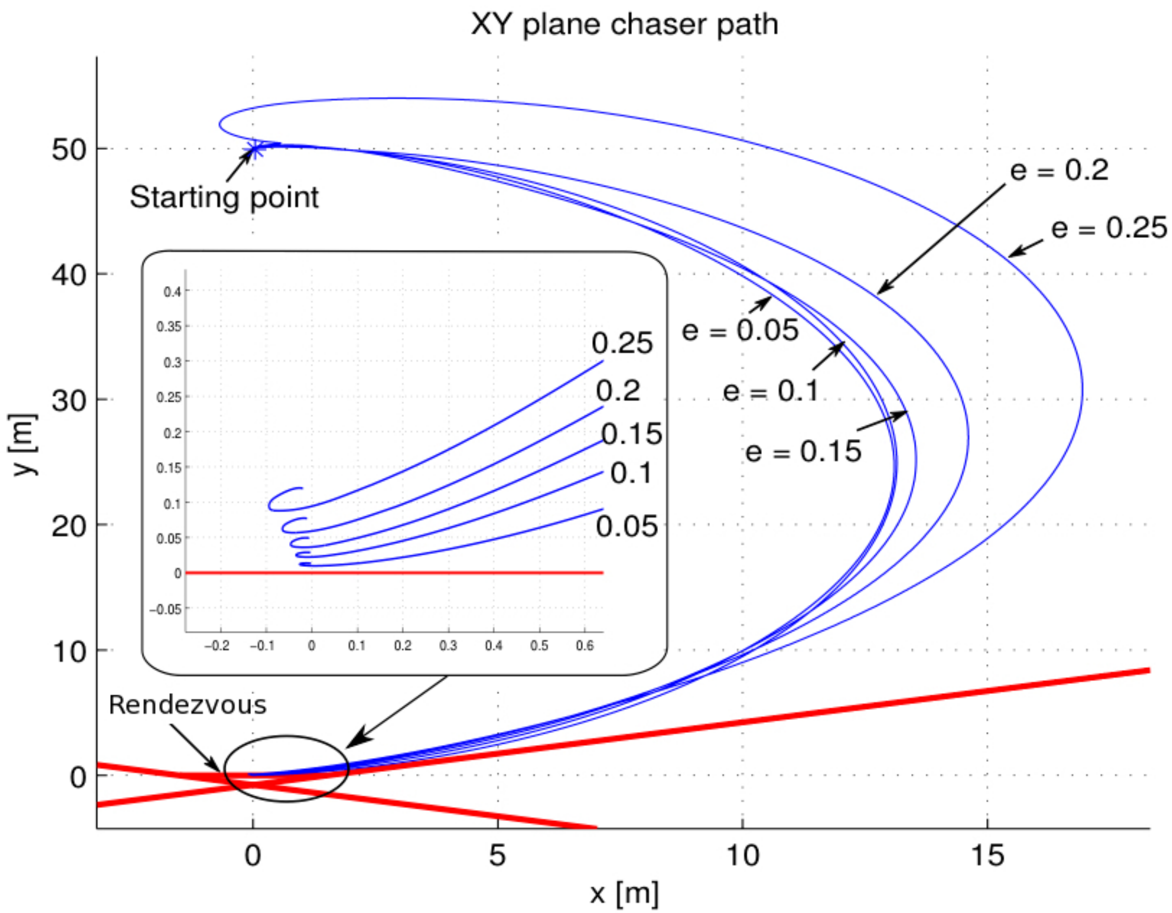
- Increase in cost of the non-robust MPC with respect to the chance constrained MPC.

Non-robust MPC controller with unmodeled dynamics



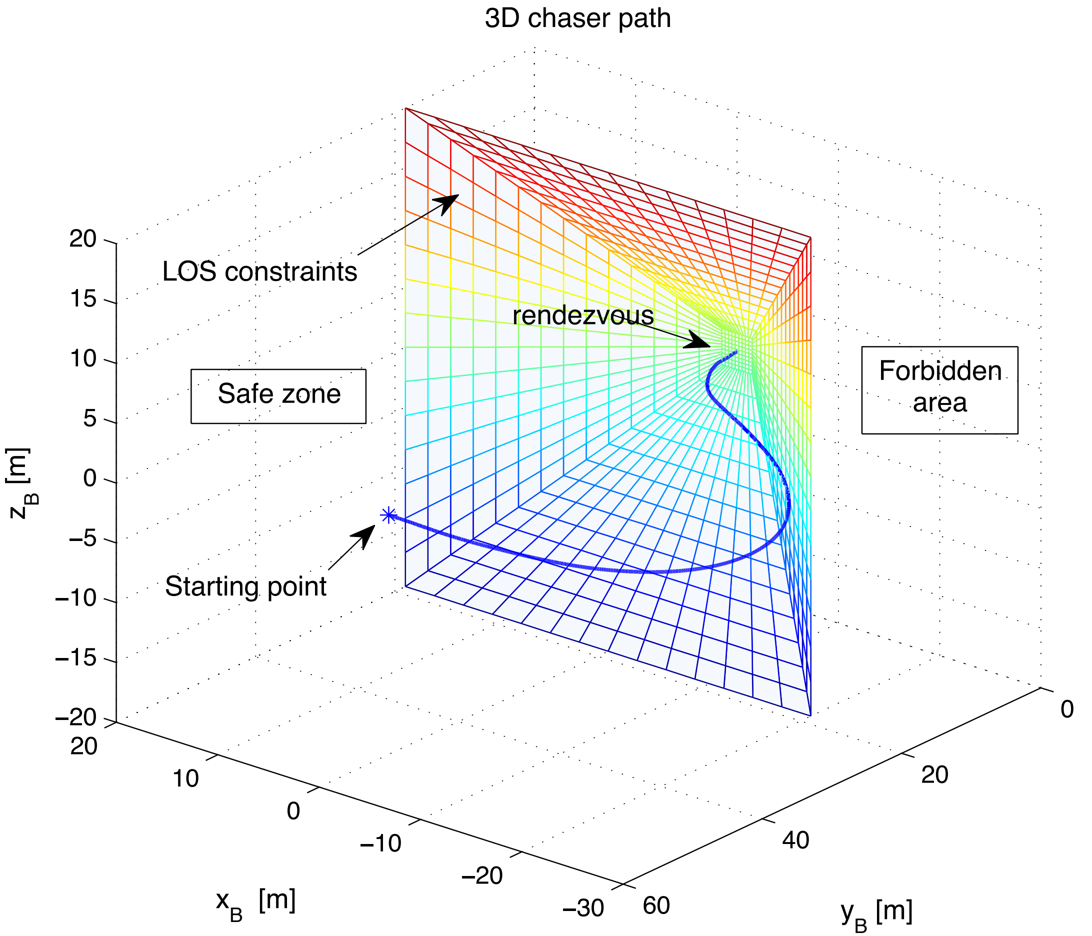
- Assume that the target orbit is elliptic (i.e. has some eccentricity e) instead of circular: unmodeled dynamics.
- Non-robust MPC is able to rendezvous, however it violates the constraints at the end.

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.

Trajectory Planning with On/Off (PWM) Thrusters

- Lots of previous results, but most consider impulsive or continuous thrust.
- Normally thrusters are pulsed: fixed amount of propulsion for a variable time of actuation (PWM).
- There are some previous results with PWM thrusters using filters or multirate approaches, but do not directly include the PWM constraints.

Trajectory Planning with On/Off (PWM) Thrusters

- Lots of previous results, but most consider impulsive or continuous thrust.
- Normally thrusters are pulsed: fixed amount of propulsion for a variable time of actuation (PWM).
- There are some previous results with PWM thrusters using filters or multirate approaches, but do not directly include the PWM constraints.
- Our previous result (IFAC WC 2011) considered circular orbits (LTI system).
- **In this work we present a close-range rendezvous planning algorithm for the more realistic PWM (On/Off) thruster case for elliptical orbits.**
- Since the orbits are elliptical, the models is LTV and more nonlinear in PWM variables. **Basic transformations from PAM or impulsive actuation to PWM do not work very well**.

Rendezvous Model (free motion)

- Elliptical orbits: Tschauner-Hempel model.
  - Target is passive and follows an elliptical keplerian orbit of eccentricity e and semi-major axis a, with starting eccentric anomaly E_0 at t_0.
  - The chaser is close (kilometers) compared with target orbital radius (thousands of kilometers).
  - The model is linear but time-varying, and time-discrete (sampling time T).

Rendezvous Model (free motion)

- Elliptical orbits: Tschauner-Hempel model.
  - Target is passive and follows an elliptical keplerian orbit of eccentricity e and semi-major axis a, with starting eccentric anomaly E_0 at t_0.
  - The chaser is close (kilometers) compared with target orbital radius (thousands of kilometers).
  - The model is linear but time-varying, and time-discrete (sampling time T).

Tschauner-Hempel model (free motion)

$$\mathbf{x}_{k+1} = A(t_{k+1}, t_k)\mathbf{x}_k, \; \mathbf{x}_k = [x_k \; y_k \; z_k \; v_{x,k} \; v_{y,k} \; v_{z,k}]^T,$$

- The matrix $A(t_{k+1}, t_k)$ can be written explicitly if instead of time t_k, true anomaly (θ_k) or eccentric anomaly (E_k) is used. They are related through Kepler's equation and E_0 so that, for instance, $E_k = K(t_k)$.
- A convenient form was expressed by Yamanaka and Ankersen, where $A(t_{k+1}, t_k) = Y_{K(t_{k+1})} Y^{-1}_{K(t_k)}$, with Y, Y^{-1} explicit.

Rendezvous Model (impulsive thrust)

- A typical actuator model considers impulsive thrust, such that the velocity is instantaneously changed.
- Impulses are placed at the beginning of the time interval.
- Good model if the impulses are high and short, not so good if the impulses are low and maintained for a certain interval of time.

Rendezvous Model (impulsive thrust)

- A typical actuator model considers impulsive thrust, such that the velocity is instantaneously changed.
- Impulses are placed at the beginning of the time interval.
- Good model if the impulses are high and short, not so good if the impulses are low and maintained for a certain interval of time.

Tschauner-Hempel model (impulsive thrust)

$$\mathbf{x}_{k+1} = A(t_{k+1}, t_k)\mathbf{x}_k + B(t_{k+1}, t_k)\mathbf{u}_k, \ \mathbf{u}_k = [u_{x,k} \ u_{y,k} \ u_{z,k}]^T$$

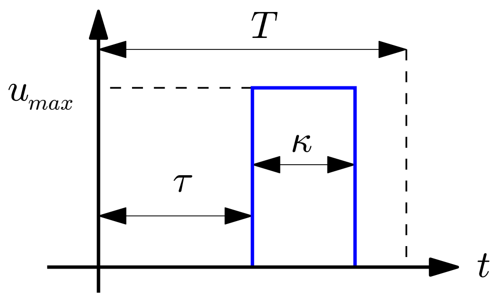
- The vector $\mathbf{u}_k$ represent the impulses (ΔV).
- The matrix $B(t_{k+1}, t_k)$ is explicitly found from $A(t_{k+1}, t_k)$.

Rendezvous Model (ON/OFF thrust)

- Thrusters typically can be only switched on or off and produce a fixed amount of force: PWM control.
- Assume an aligned pair of thrusters for each direction $i = 1, 2, 3$ with opposing orientation.Positive and negative are denoted as u_i^+ and u_i^-.
- The (fixed) value of thrust is $\bar{u}_i^+$ and $\bar{u}_i^-$, respectively.
- During each sample time each thruster fires only once.

Rendezvous Model (ON/OFF thrust)

- Thrusters typically can be only switched on or off and produce a fixed amount of force: PWM control.
- Assume an aligned pair of thrusters for each direction $i = 1, 2, 3$ with opposing orientation.Positive and negative are denoted as u_i^+ and u_i^-.
- The (fixed) value of thrust is $\bar{u}_i^+$ and $\bar{u}_i^-$, respectively.
- During each sample time each thruster fires only once.
- 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, and one for each direction (denoted by $+$ and -).
- 12 control variables for each k: $\kappa_1^+(k)$, $\kappa_2^+(k)$, $\kappa_3^+(k)$, $\kappa_1^+(k)$, $\kappa_2^-(k)$, $\kappa_3^-(k)$, $\tau_1^+(k)$, $\tau_2^+(k)$, $\tau_3^+(k)$, $\tau_1^+(k)$, $\tau_2^-(k)$, $\tau_3^-(k)$

Rendezvous Model (ON/OFF thrust)

- Call $\mathbf{u}_k^P = \left[\; \tau_{1,k}^+ \, \kappa_{1,k}^+ \, \tau_{1,k}^- \, \kappa_{1,k}^- \, \tau_{2,k}^+ \, \kappa_{2,k}^+ \, \tau_{2,k}^- \, \kappa_{2,k}^- \, \tau_{3,k}^+ \, \kappa_{3,k}^+ \, \tau_{3,k}^- \, \kappa_{3,k}^- \; \right]^T$
- $\mathbf{u}_k^P$ contains all the PWM control variables.
- Denote

$$b_i(t, \tau_i, \kappa_i) = \int_{K(t+\tau_i)}^{K(t+\tau_i+\kappa_i)} Y_E^{-1} C_{i+3} \frac{1 - e \cos E}{n} dE$$

  where C_i is a column vector of zeros with a 1 in the i-th row.
- The integrals in b_i can be carried out explicitly. The nonlinear dependence of the system on the PWM parameters is contained in b_i.

Rendezvous Model (ON/OFF thrust)

- Call $\mathbf{u}_k^P = \left[\ \tau_{1,k}^+ \ \kappa_{1,k}^+ \ \tau_{1,k}^- \ \kappa_{1,k}^- \ \tau_{2,k}^+ \ \kappa_{2,k}^+ \ \tau_{2,k}^- \ \kappa_{2,k}^- \ \tau_{3,k}^+ \ \kappa_{3,k}^+ \ \tau_{3,k}^- \ \kappa_{3,k}^-\ \right]^T$
- $\mathbf{u}_k^P$ contains all the PWM control variables.
- Denote

$$b_i(t, \tau_i, \kappa_i) = \int_{K(t+\tau_i)}^{K(t+\tau_i+\kappa_i)} Y_E^{-1} C_{i+3} \frac{1 - e\cos E}{n} dE$$

where C_i is a column vector of zeros with a 1 in the i-th row.
- The integrals in b_i can be carried out explicitly. The nonlinear dependence of the system on the PWM parameters is contained in b_i.

Tschauner-Hempel model (ON/OFF thrust)

$$\mathbf{x}_{k+1} = A(t_{k+1}, t_k)\mathbf{x}_k + B_{PWM}(t_{k+1}, t_k, \mathbf{u}_k^P)$$

- In the equation $B_{PWM} = \sum_{i=1}^{i=3} B_i^+ \bar{u}_i^+ + \sum_{i=1}^{i=3} B_i^- \bar{u}_i^-$, with $B_i^\pm(t_{k+1}, t_k, \mathbf{u}_k^P) = Y(t_{k+1}) b_i(t, \tau_{i,k}^\pm, \kappa_{i,k}^\pm)$.

Objectives and State Constraints

- Time of rendezvous T_R is usually fixed beforehand.
- A sampling time T is chosen for discretization such that $T_R = NT$, where N is the discrete time of rendezvous.

Objectives and State Constraints

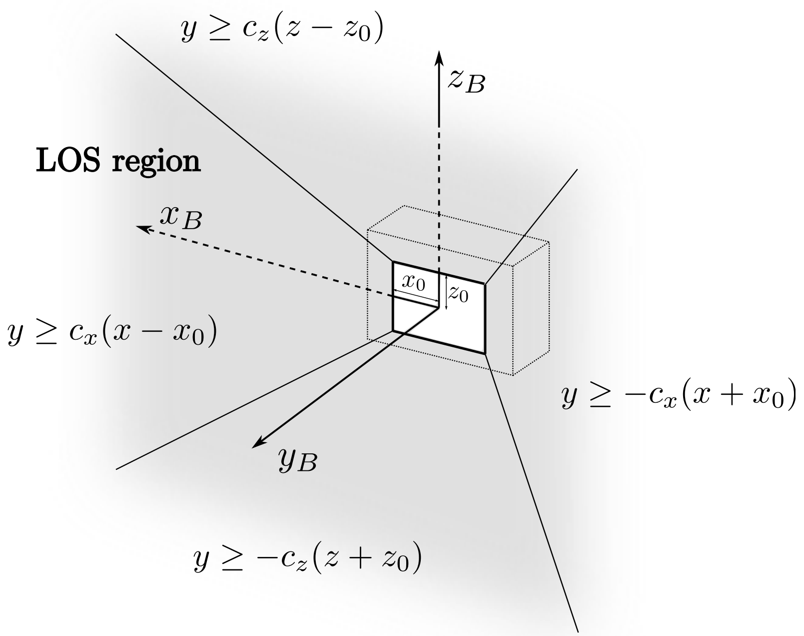
- Time of rendezvous T_R is usually fixed beforehand.
- A sampling time T is chosen for discretization such that $T_R = NT$, where N is the discrete time of rendezvous.
- Consumption of fuel should be minimized:

$$\min \int_0^{t_F} \left(|u_x(t)| + |u_y(t)| + |u_z(t)| \right) dt$$

Objectives and State Constraints

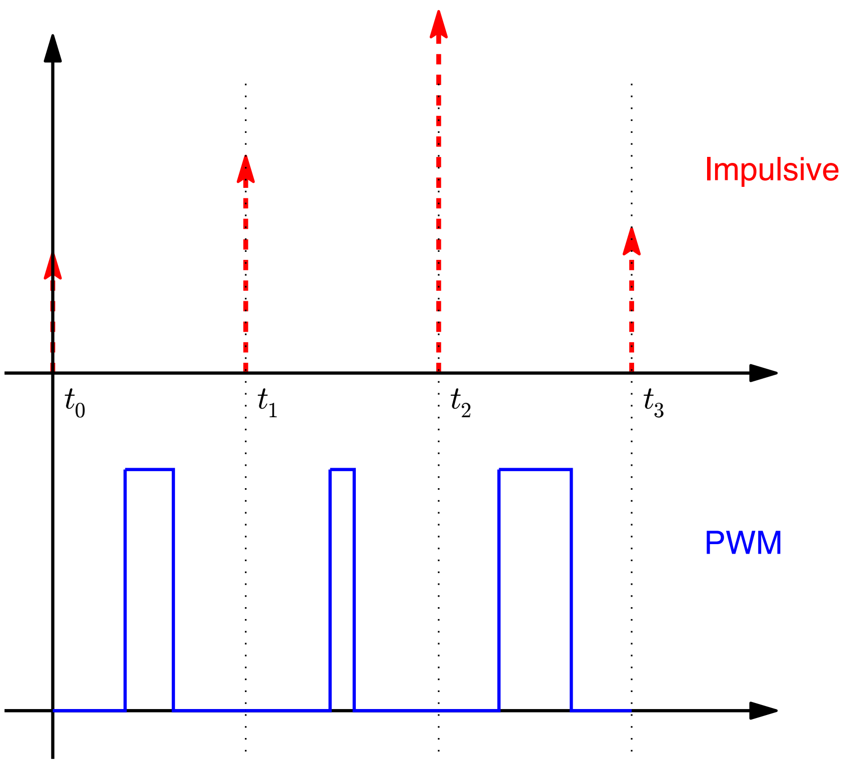
- Time of rendezvous T_R is usually fixed beforehand.
- A sampling time T is chosen for discretization such that $T_R = NT$, where N is the discrete time of rendezvous.
- Consumption of fuel should be minimized:

$$\min \int_0^{t_F} (|u_x(t)| + |u_y(t)| + |u_z(t)|)\, dt$$



- State should remain in safe zone for security and sensing purposes: "line of sight" (LOS) region

Constraints of the problem: Actuator constraints



- Implusive:
  - Any value of impulse in a given range can be used, i.e. $u_{min} \leq u(t) \leq u_{max}$.
  - In spacecraft, high-force thrusters actuating for a short time can be modeled as impulsive.
  - Not realistic for small spacecraft!
- Pulse-Width Modulated (PWM):
  - The value of force is fixed to a value $\bar{u}$, only the start and duration of it can be set.
  - Conventional chemical thrusters.
  - We will consider this constraint.

Planning algorithm for ON/OFF thrusters

- As seen, equations are **highly nonlinear** and not explicit in PWM control variables (pulse start point and width).

Planning algorithm for ON/OFF thrusters

- As seen, equations are **highly nonlinear** and not explicit in PWM control variables (pulse start point and width).
- The following algorithm is applied:

PWM Rendezvous Planning Algorithm

1. Initially solve the rendezvous problem for standard impulsive control.
2. From the impulsive solution find an initial starting guess for the PWM solution.
3. Linearize around PWM solution and find small increments in the PWM controls improving the solution.
4. Repeat previous step until convergence or time is up.

Planning algorithm for ON/OFF thrusters

- As seen, equations are **highly nonlinear** and not explicit in PWM control variables (pulse start point and width).
- The following algorithm is applied:

PWM Rendezvous Planning Algorithm

1. Initially solve the rendezvous problem for standard impulsive control.
2. From the impulsive solution find an initial starting guess for the PWM solution.
3. Linearize around PWM solution and find small increments in the PWM controls improving the solution.
4. Repeat previous step until convergence or time is up.

- Linearization explicit and easy to compute.
- Since we have a reasonable initial guess the algorithm works well.

Step 1. Finding a solution with impulsive actuation

- Using the impulsive Tschauner-Hempel model, and iterating:
$$\mathbf{x}_{k+1} = A(t_{k+1}, t_0)\mathbf{x}(0) + \sum_{j=0}^{k} A(t_{k+1}, t_{j+1})B(t_{j+1}, t_j)\mathbf{u}_j$$
- We have used the property
$A(t_{i+1}, t_i)A(t_i, t_{i-1}) = A(t_{i+1}, t_{i-1})$.
- Compact (stack) notation for the whole planning horizon:

$$\mathbf{x}_S = \begin{bmatrix} \mathbf{x}(1) \\ \mathbf{x}(2) \\ \vdots \\ \mathbf{x}(N) \end{bmatrix}, \quad \mathbf{u}_S = \begin{bmatrix} \mathbf{u}(0) \\ \mathbf{u}(1) \\ \vdots \\ \mathbf{u}(N-1) \end{bmatrix}$$

- Compact propagation equation:

$$\mathbf{x}_S = \mathbf{F}\mathbf{x}(0) + \mathbf{G}_\mathbf{u}\mathbf{u}_S$$

$\mathbf{F}$ and $\mathbf{G}_\mathbf{u}$ defined in the paper.

Step 1. Finding a solution with impulsive actuation

- The objective function (fuel consumption) can be written as:

$$J = T\|\mathbf{u_S}\|_{L^1}$$

- Using the compact notation, the LOS constraints are written

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

  and using propagation of the state in terms of $\mathbf{u_S}$:

$$\mathbf{A}_c\mathbf{G_u}\mathbf{u_S} \leq \mathbf{b}_c - \mathbf{A}_c\mathbf{F}\mathbf{x}(0)$$

- Similarly, terminal constraints ($\mathbf{x}(N) = \mathbf{0}$) are written as $\mathbf{A}_e\mathbf{x_S} = 0$, thus in terms of $\mathbf{u_S}$:

$$\mathbf{A}_e\mathbf{G_u}\mathbf{u_S} = -\mathbf{A}_e\mathbf{F}\mathbf{x}(0)$$

- Control signal restriction are written as $-T\bar{\mathbf{u}}^- \leq \mathbf{u_S} \leq T\bar{\mathbf{u}}^+$ (see step 2).

Step 1. Finding a solution with impulsive actuation

- The trajectory planning problem with impulsive actuation is formulated as:

$$\begin{aligned} \min_{\mathbf{u_S}} \quad & J(\mathbf{u_S}) \\ \text{subject to} \quad & \mathbf{A}_c\mathbf{G_u}\mathbf{u_S} \leq \mathbf{b}_c - \mathbf{A}_c\mathbf{F}\mathbf{x}(0) \\ & -T\bar{\mathbf{u}}^- \leq \mathbf{u_S} \leq T\bar{\mathbf{u}}^+ \\ & \mathbf{A}_e\mathbf{G_u}\mathbf{u_S} = -\mathbf{A}_e\mathbf{F}\mathbf{x}(0) \end{aligned}$$

Step 1. Finding a solution with impulsive actuation

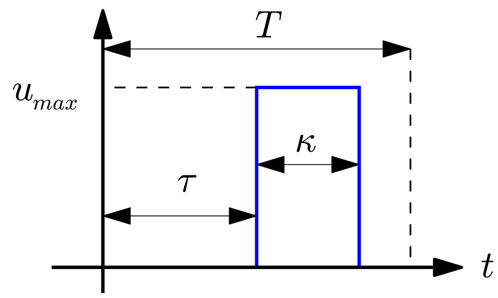
- The trajectory planning problem with impulsive actuation is formulated as:

$$\begin{aligned} \min_{\mathbf{u}_S} \quad & J(\mathbf{u}_S) \\ \text{subject to} \quad & \mathbf{A}_c \mathbf{G}_\mathbf{u} \mathbf{u}_S \leq \mathbf{b}_c - \mathbf{A}_c \mathbf{F} \mathbf{x}(0) \\ & -T\bar{\mathbf{u}}^- \leq \mathbf{u}_S \leq T\bar{\mathbf{u}}^+ \\ & \mathbf{A}_e \mathbf{G}_\mathbf{u} \mathbf{u}_S = -\mathbf{A}_e \mathbf{F} \mathbf{x}(0) \end{aligned}$$

- L^1-norm optimization with linear inequality and equality constraints; $\mathbf{x}(0)$ is known, $\mathbf{u}_S$ has to be found.
- Easily solvable, for instance, in MATLAB, using `linprog`.

Step 2. A fist PWM solution

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



- To find an initial guess of the PWM control variables from the impulsive actuation, we use:
  1. Use a positive or negative thruster according to the sign of $u_{i,k}$.
  2. The pulse width has an area equal to the impulse value: $\kappa^{\pm}_{i,k} = \frac{|u_{i,k}|}{\bar{u}^{\pm}_i}$, where $\bar{u}^{\pm}_i$ is the maximum level of the (positive or negative) thruster i (since $-T\bar{\mathbf{u}}^{-} \leq \mathbf{u_S} \leq T\bar{\mathbf{u}}^{+}$, $\kappa^{\pm}_{i,k} \leq T$).
  3. Since the impulse was modeled to start at the beginning of a time sample, $\tau^{\pm}_{i,k} = 0$.
- $\mathbf{u}^P_k$ constructed by this method is not optimal and might not even verify the constraints or reach the target. However it is close to a PWM solution.

Step 3. Linearization of the PWM model

- The linearized model is written as

$$\mathbf{x}_{k+1} = A(t_{k+1}, t_k)\mathbf{x}_k + B_{PWM}(t_{k+1}, t_k, \mathbf{u}_k^P) + B^{\Delta}(t_{k+1}, t_k, \mathbf{u}_k^P)\boldsymbol{\Delta}\mathbf{u}_k^P,$$

- $\boldsymbol{\Delta}\mathbf{u}_k^P$ are the increments in the PWM signals and the matrix $B^{\Delta}(\tau, \kappa(k))$ is defined as

$$(B^{\Delta})_{i,j} = \frac{\partial(B_{PWM}(t_{k+1}, t_k, \mathbf{u}_k^P))_i}{\partial(\mathbf{u}_k^P)_j},$$

which is explicit (derivative of an integral). See the paper.

- Constraints:

$$\begin{aligned} -\Delta\kappa_i^{\pm}(k) &\leq \kappa_i^{\pm}(k), -\Delta\tau_i^{\pm}(k) \leq \tau_i^{\pm}(k) \\ \Delta\tau_i^{\pm}(k) + \Delta\kappa_i^{\pm}(k) &\leq T - \tau_i^{\pm}(k) - \kappa_i^{\pm}(k) \end{aligned}$$

Step 3. Linearization of the PWM model

- The linearized model is written as

$$\mathbf{x}_{k+1} = A(t_{k+1}, t_k)\mathbf{x}_k + B_{PWM}(t_{k+1}, t_k, \mathbf{u}_k^P) + B^{\Delta}(t_{k+1}, t_k, \mathbf{u}_k^P)\mathbf{\Delta u}_k^P,$$

- $\mathbf{\Delta u}_k^P$ are the increments in the PWM signals and the matrix $B^{\Delta}(\tau, \kappa(k))$ is defined as

$$(B^{\Delta})_{i,j} = \frac{\partial(B_{PWM}(t_{k+1}, t_k, \mathbf{u}_k^P))_i}{\partial(\mathbf{u}_k^P)_j},$$

which is explicit (derivative of an integral). See the paper.

- Constraints:

$$\begin{aligned} -\Delta\kappa_i^{\pm}(k) &\leq \kappa_i^{\pm}(k), -\Delta\tau_i^{\pm}(k) \leq \tau_i^{\pm}(k) \\ \Delta\tau_i^{\pm}(k) + \Delta\kappa_i^{\pm}(k) &\leq T - \tau_i^{\pm}(k) - \kappa_i^{\pm}(k) \end{aligned}$$

- Add additional constraint on $\mathbf{\Delta u}_k^P$ size to avoid going too far away from linearization point: $|\mathbf{\Delta u}_k^P| \leq \mathbf{\Delta}^{MAX}$.

Step 3. Linearization of the PWM model

- Compact/stack notation: $\mathbf{u}_\mathbf{S}^\mathbf{P}$ for the PWM variables, $\mathbf{\Delta u}_\mathbf{S}^\mathbf{P}$ for the increments.
- PWM Compact formulation around the linearized point:

$$\mathbf{x_S} = \mathbf{F}\mathbf{x}(0) + \mathbf{G_{PWM}}(\mathbf{u}_\mathbf{S}^\mathbf{P})\bar{\mathbf{u}}_\mathbf{S} + \mathbf{G_\Delta}(\mathbf{u}_\mathbf{S}^\mathbf{P})\mathbf{\Delta u}_\mathbf{S}^\mathbf{P},$$

- State constraints written in terms of $\mathbf{\Delta u}_\mathbf{S}^\mathbf{P}$:

$$\begin{aligned} \mathbf{A}_c\mathbf{G_\Delta}(\mathbf{u}_\mathbf{S}^\mathbf{P})\mathbf{\Delta u}_\mathbf{S}^\mathbf{P} &\leq \mathbf{b}_c - \mathbf{A}_c\mathbf{F}\mathbf{x}(0) - \mathbf{A}_c\mathbf{G_{PWM}}(\mathbf{u}_\mathbf{S}^\mathbf{P})\bar{\mathbf{u}}_\mathbf{S} \\ \mathbf{A}_e\mathbf{G_\Delta}(\mathbf{u}_\mathbf{S}^\mathbf{P})\mathbf{\Delta u}_\mathbf{S}^\mathbf{P} &= -\mathbf{A}_e\mathbf{F}\mathbf{x}(0) - \mathbf{A}_e\mathbf{G_{PWM}}(\mathbf{u}_\mathbf{S}^\mathbf{P})\bar{\mathbf{u}}_\mathbf{S} \end{aligned}$$

- Summarize PWM actuation constraints as $\mathbf{A_\Delta}\mathbf{\Delta u}_\mathbf{S}^\mathbf{P} \leq \mathbf{b_\Delta}$.

Step 3. Linearization of the PWM model

- Objective function $J = J_{PWM}(\mathbf{u_S^P}) + \mathbf{J^\Delta}(\mathbf{\Delta u_S^P})$, where

$$J^\Delta(\mathbf{\Delta u_S^P}) = \sum_{k=0}^{N_p-1}\sum_{i=1}^{3}\left(\bar{u}_i^+\Delta\kappa_i^+(k) + \bar{u}_i^-\Delta\kappa_i^-(k)\right)$$

Step 3. Linearization of the PWM model

- Objective function $J = J_{PWM}(\mathbf{u_S^P}) + \mathbf{J^\Delta}(\mathbf{\Delta u_S^P})$, where

$$J^\Delta(\mathbf{\Delta u_S^P}) = \sum_{k=0}^{N_p-1}\sum_{i=1}^{3}\left(\bar{u}_i^+\Delta\kappa_i^+(k) + \bar{u}_i^-\Delta\kappa_i^-(k)\right)$$

- Optimization on increment $\mathbf{\Delta u_S^P}$:

$$\begin{aligned} \min_{\mathbf{\Delta u_S^P}} \quad & J^\Delta(\mathbf{\Delta u_S^P}) \\ \text{subject to:} \quad & \mathbf{A}_c\mathbf{G_\Delta}\mathbf{\Delta u_S^P} \le \mathbf{b}_c - \mathbf{A}_c\mathbf{F}\mathbf{x}(0) - \mathbf{A}_c\mathbf{G_{PWM}}(\mathbf{u_S^P})\mathbf{\bar{u}_S} \\ & \mathbf{A_\Delta}\mathbf{\Delta u_S^P} \le \mathbf{b_\Delta} \\ & \mathbf{A}_e\mathbf{G_\Delta}\mathbf{\Delta u_S^P} = -\mathbf{A}_e\mathbf{F}\mathbf{x}(0) - \mathbf{A}_e\mathbf{G_{PWM}}(\mathbf{u_S^P})\mathbf{\bar{u}_S} \end{aligned}$$

Step 3. Linearization of the PWM model

- Objective function $J = J_{PWM}(\mathbf{u_S^P}) + \mathbf{J^\Delta}(\mathbf{\Delta u_S^P})$, where

$$J^\Delta(\mathbf{\Delta u_S^P}) = \sum_{k=0}^{N_p-1}\sum_{i=1}^{3}\left(\bar{u}_i^+\Delta\kappa_i^+(k) + \bar{u}_i^-\Delta\kappa_i^-(k)\right)$$

- Optimization on increment $\mathbf{\Delta u_S^P}$:

$$\min_{\mathbf{\Delta u_S^P}} \quad J^\Delta(\mathbf{\Delta u_S^P})$$

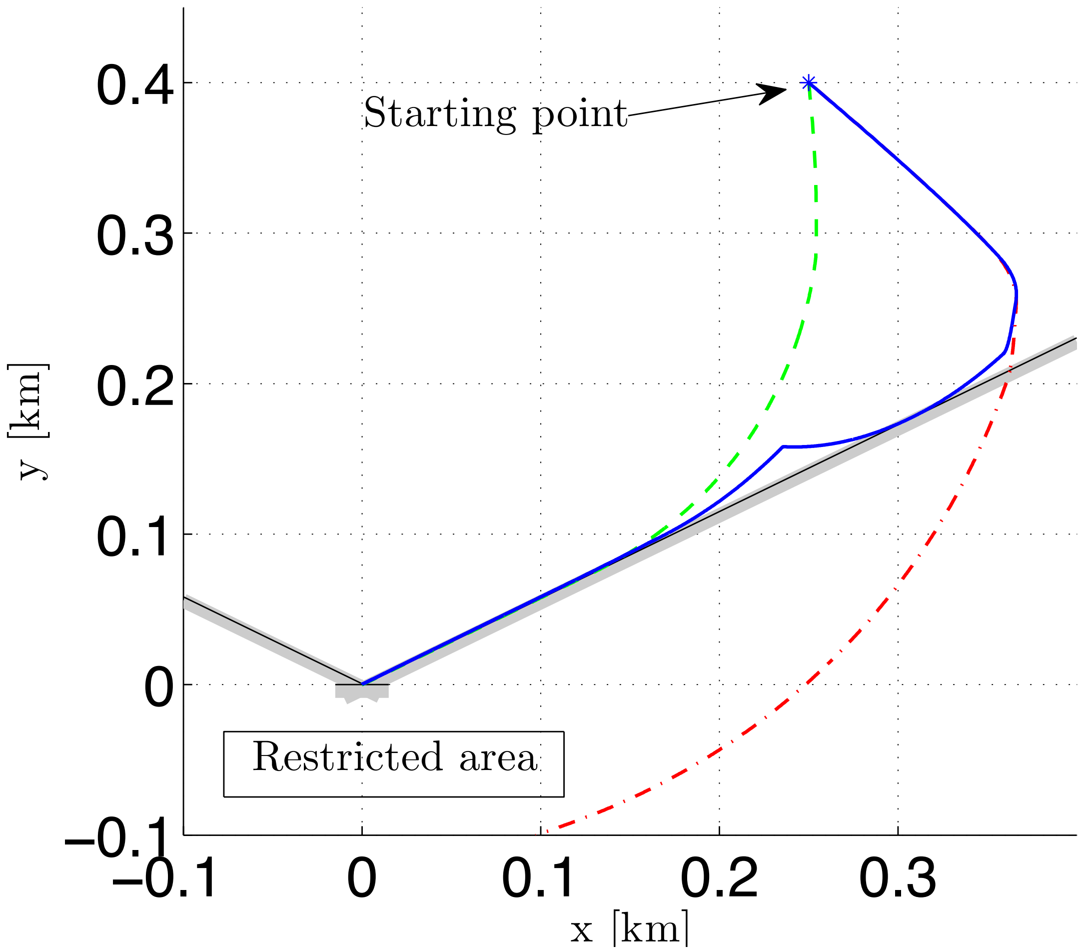
$$\text{subject to:} \quad \begin{aligned} \mathbf{A}_c\mathbf{G_\Delta}\mathbf{\Delta u_S^P} &\leq \mathbf{b}_c - \mathbf{A}_c\mathbf{F}\mathbf{x}(0) - \mathbf{A}_c\mathbf{G_{PWM}}(\mathbf{u_S^P})\mathbf{\bar{u}_S} \\ \mathbf{A_\Delta}\mathbf{\Delta u_S^P} &\leq \mathbf{b_\Delta} \\ \mathbf{A}_e\mathbf{G_\Delta}\mathbf{\Delta u_S^P} &= -\mathbf{A}_e\mathbf{F}\mathbf{x}(0) - \mathbf{A}_e\mathbf{G_{PWM}}(\mathbf{u_S^P})\mathbf{\bar{u}_S} \end{aligned}$$

- Linear cost function with linear inequality and equality constraints; very fast solution!
- Add solution $\mathbf{\Delta u_S^P}$ to previous linearization point $\mathbf{u_S^P}$ to find new PWM values $\mathbf{u_S^P}(\mathbf{new})$: new linearization point.
- Linearize around new solution and iterate until cost function does not improve or time is up!

Simulation results for the PWM algorithm

- Matlab simulation of a high eccentricity case ($e = 0.7$).
- Parameters: $N_p = 50$ as planning horizon, $T = 60$ s, and $\bar{u} = 10^{-1}$ N/kg. The target orbit has perigee altitude $h_p = 500$ km.
- Initial conditions were $\theta_0 = 45^{\circ}$, $\mathbf{r}_0 = [0.25\ 0.4\ \ -0.2]^T$ km, $\mathbf{v}_0 = [0.005\ \ -0.005\ \ -0.005]^T$ km/s. The LOS constraint is defined by $x_0 = 0.001$ km and $C_{LOS} = \tan 30^{\circ}$.
- Impulsive initial cost: 14.6 m/s.
- After 6 iterations, the solution converges. Each iteration took about 1 second to compute.
- Final PWM cost: 15.5 m/s

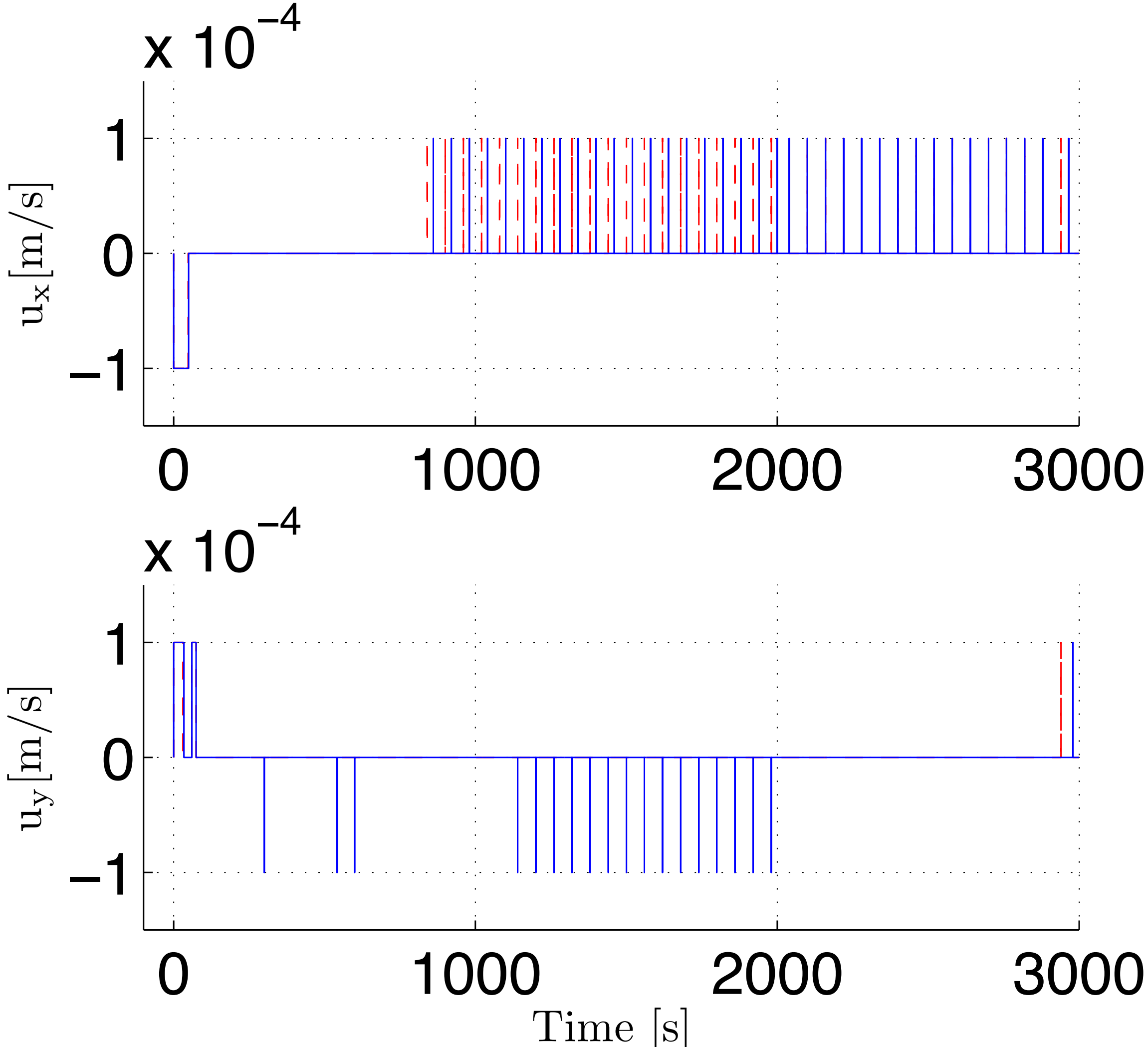
Simulation results for the PWM algorithm



- Trajectories: impulsive (green), PWM computed from impulsive (red), final computed PWM(blue)

Simulation results for the PWM algorithm

- Comparison between PWM computed from impulsive (red) and final computed PWM control signals (blue).



Conclusions

- We have presented a planning algorithm to solve the problem of automatic spacecraft rendezvousfor elliptical target orbits.
- Line-of-sight state constraints and PWM control constraints are included in the model.
- To overcome nonlinear optimization, algorithm uses a **hot start** obtained from impulsive actuation and refines it using **explicit linearization**.
- In simulations it is shown that the algorithm converges.

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. ⟶ also IFAC 2014, CEP 2017
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.

Outline

1. ~~Introduction to MPC (Slides by E.F. Camacho)~~
2. ~~Application to Spacecraft Rendezvous (including PWM)~~
3. Rendezvous + Attitude Control ←
4. Soft Landing on an Asteroid
5. Guidance for UAVs

A Flatness-Based Trajectory Planning Algorithm for Rendezvous of Single-Thruster Spacecraft

Julio Cesar Sanchez Merino[1]
Francisco Gavilan Jimenez[1]
Rafael Vazquez Valenzuela[1]
Christophe Louembet[2]

[1]Universidad de Sevilla. Sevilla, Spain.
jsanchezm@us.es, fgavilan@us.es, rvazquez1@us.es

[2]LAAS/CNRS & Université de Toulouse. Toulouse, France.
louembet@laas.fr

May 29, 2019

Table of contents

Outline

Introduction

Objective

Generate optimal **rendezvous** trajectories for a **single-thruster**[1,2] **spacecraft** equipped with an **ACS**.

Methodology

Exploit the **state transition matrix** for translational motion and the **flatness property**[3] for angular motion. Then, **discretize** the problem to obtain a tractable static program.

[1] Oland, E., et al. Aerospace Conference (2013).
[2] Moon, G.H., et al. European Control Conference (2016).
[3] Louembet, C., et al. IET Control Theory and Applications (2009).

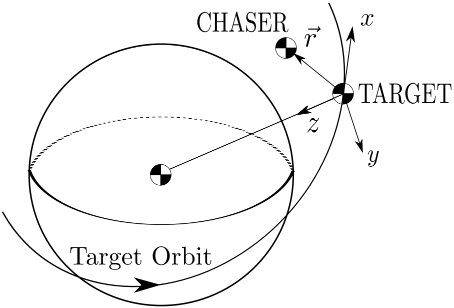
Outline

Translational motion I

Target in circular obit.
Target & chaser close ($\sim$1 km).
HCW equations[1] in LVLH frame

$$\begin{cases} \ddot{x} &= 2n\dot{z}, \\ \ddot{y} &= -n^2 y, \\ \ddot{z} &= 3n^2 z - 2n\dot{x}. \end{cases}$$

$$n = \sqrt{\frac{\mu_\oplus}{R^3}}$$



[1] Clohessy, W., et al. Journal of Aerospace Sciences (1960).

Translational motion II

- Propulsion modelled as discrete **impulses**:

$$\mathbf{u}(t) = \sum_{k=1}^{N_p} \mathbf{u}_k \delta(t - t_k).$$

- From HCW equations to **state transition matrix**

$$\boxed{\mathbf{x}(t) = \mathbf{A}(t, t_0)\mathbf{x_0} + \mathbf{B}\mathbf{u}(t),}$$

where

$$\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T,$$
$$\mathbf{u} = [u_x, u_y, u_z]^T.$$

Rotational motion I

Modified Rodrigues parameters[1], MRP, for attitude representation wrt LVLH frame

- MRP are a minimal attitude representation (no unit-norm quaternion constraint)
- Denoted as $\boldsymbol{\sigma}=[\sigma_1, \sigma_2, \sigma_3]^T$, related with rotation axis $\mathbf{e}$ and angle θ_{rot} as $\boldsymbol{\sigma}=\mathbf{e}\tan(\theta_{rot}/4)$.
- Singularities at $\theta_{rot}=\pm 2\pi$ avoided constraining $\theta_{rot}\in(-2\pi, 2\pi)$.
- Rotation (DCM) matrix:

$$\mathbf{R}(\boldsymbol{\sigma}) = \mathbf{Id} + \frac{8\boldsymbol{\sigma}^{\times}\boldsymbol{\sigma}^{\times} - 4(1-\|\boldsymbol{\sigma}\|_2^2)\boldsymbol{\sigma}^{\times}}{\left(1+\|\boldsymbol{\sigma}\|_2^2\right)^2}.$$

[1] Marandi, S., et al. Acta Astronautica (1987).

Rotational motion II

Rotational kinematics

$$\begin{bmatrix}\dot\sigma_1\\ \dot\sigma_2\\ \dot\sigma_3\end{bmatrix} = \begin{bmatrix}1+\sigma_1^2-\sigma_2^2-\sigma_3^2 & 2(\sigma_1\sigma_2-\sigma_3) & 2(\sigma_1\sigma_3+\sigma_2)\\ 2(\sigma_1\sigma_2+\sigma_3) & 1-\sigma_1^2+\sigma_2^2-\sigma_3^2 & 2(\sigma_2\sigma_3-\sigma_1)\\ 2(\sigma_1\sigma_3-\sigma_2) & 2(\sigma_2\sigma_3+\sigma_1) & 1-\sigma_1^2-\sigma_2^2+\sigma_3^2\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix} \rightarrow \dot{\boldsymbol\sigma}(t) = \mathbf{C}(\boldsymbol\sigma(t))\boldsymbol\omega(t)$$

Rotational dynamics (body axes chosen as principal axes)

$$\begin{cases} I_1\dot\omega_1 = M_1 - (I_3 - I_2)\omega_2\omega_3,\\ I_2\dot\omega_2 = M_2 - (I_1 - I_3)\omega_1\omega_3,\\ I_3\dot\omega_3 = M_3 - (I_2 - I_1)\omega_1\omega_2.\end{cases}$$

ACS w/ reaction wheels is being considered but **torque M** taken as control input for simplicity

Coupled motion

Single-thruster pointing at the $\mathbf{v}$ direction (in body axes). **Projection** of impulse $u(t)$ on **LVLH frame** is

$$\mathbf{u}(t) = \mathbf{R}(\boldsymbol{\sigma}(t))\mathbf{v}u(t).$$

Coupled 6 DoF system is

$$\begin{cases} \mathbf{x}(t) &= \mathbf{A}(t, t_0)\mathbf{x}(t_0) + \mathbf{B}\mathbf{R}(\boldsymbol{\sigma}(t))\mathbf{v}u(t), \\ \dot{\boldsymbol{\sigma}}(t) &= \mathbf{C}(\boldsymbol{\sigma}(t))\boldsymbol{\omega}(t), \\ \mathbf{I}\dot{\boldsymbol{\omega}}(t) &= \mathbf{M}(t) - \boldsymbol{\omega}(t) \times \mathbf{I}\boldsymbol{\omega}(t). \end{cases}$$

Coupling arises through the **propulsion term** of the translational equation (gravity gradient effects neglected).

Outline

Constraints

- **Line of sight** (LOS):

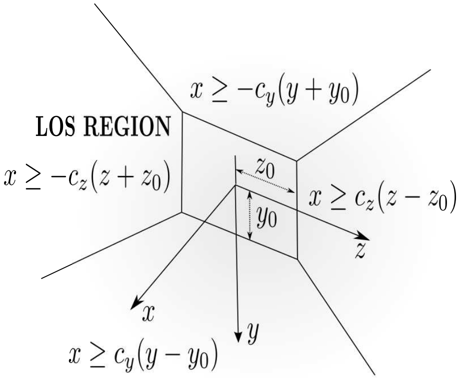
$$\mathbf{A}_L \mathbf{x}(t) \leq \mathbf{b}_L.$$

- **Control input bounds**:

$$0 \leq u(t) \leq u_{max},$$
$$-M_{max} \leq M_i(t) \leq M_{max}.$$

- **Terminal conditions**:

$$\mathbf{x}(t_f) = \mathbf{0},$$
$$\boldsymbol{\sigma}(t_f) = \boldsymbol{\sigma}_f,$$
$$\boldsymbol{\omega}(t_f) = \mathbf{0}.$$



Objective function

Minimize fuel consumption $\longrightarrow$ minimize the L1-norm of impulses

$$
\begin{array}{ll}
\underset{u(t),\mathbf{M}(t)}{\text{minimize}} & \int_{t_0}^{t_f} \|u(t)\|_1 \, dt, \\
\text{subject to} & \mathbf{x}(t) = \mathbf{A}(t, t_0)\mathbf{x}_0 + \mathbf{B}\mathbf{R}(\boldsymbol{\sigma}(t))\mathbf{v}u(t), \\
& \dot{\boldsymbol{\sigma}}(t) = \mathbf{C}(\boldsymbol{\sigma}(t))\boldsymbol{\omega}(t), \\
& \mathbf{I}\dot{\boldsymbol{\omega}}(t) = \mathbf{M}(t) - \boldsymbol{\omega}(t) \times \mathbf{I}\boldsymbol{\omega}(t), \\
& \mathbf{A}_L\mathbf{x}(t) \leq \mathbf{b}_L, \\
& 0 \leq u(t) \leq u_{max}, \\
& -M_{max} \leq M_i(t) \leq M_{max}, \quad i = 1, 2, 3, \\
& \mathbf{x}(t_f) = \mathbf{0}, \\
& \boldsymbol{\sigma}(t_f) = \boldsymbol{\sigma}_f, \\
& \boldsymbol{\omega}(t_f) = \mathbf{0}.
\end{array}
$$

Outline

Attitude flatness property

Flatness property

A **Flat system**[1] has a **flat output**, which can be used to explicitly express all states and inputs in terms of the flat output and a finite number of its derivatives.

Attitude flatness

Attitude dynamics has the flatness property. Flat output $\longrightarrow$ MRP.

$$\begin{cases} \omega(t) = \mathbf{C}^{-1}(\sigma)\dot{\sigma}, \\ \dot{\omega}(t) = \mathbf{C}^{-1}(\sigma)\ddot{\sigma} + \dot{\mathbf{C}}^{-1}(\dot{\sigma},\sigma)\dot{\sigma}, \end{cases}$$

and torque is **parameterized** with the MRP

$$\boxed{\mathbf{M}(t) = \mathbf{I}[\dot{\mathbf{C}}^{-1}(\dot{\sigma},\sigma)\dot{\sigma} + \mathbf{C}^{-1}(\sigma)\ddot{\sigma}] + [\mathbf{C}^{-1}(\sigma)\dot{\sigma}] \times \mathbf{I}\mathbf{C}^{-1}(\sigma)\dot{\sigma}.}$$

[1] Fliess, M., et al. Journal of Guidance Control and Dynamics (1995).

NLP description I

- Manoeuvre **divided** into N_p **intervals** of duration $T = (t_f - t_0)/N_p$.
- MRP parameterization[1] based on mth **degree splines**

$$\boxed{\begin{array}{l}\sigma_i(t) = \sum_{j=0}^{m} a_{i,j,k}(t - t_{k-1}),\ \ i = 1, 2, 3, \\ t \in [t_{k-1}, t_k],\ \ t_k = t_0 + kT,\ \ k = 1 \dots N_p.\end{array}}$$

- C^2 **continuity** at the **nodes**

$$\begin{cases} \boldsymbol{\sigma}(t_k, \mathbf{a}_k) = \boldsymbol{\sigma}(t_k, \mathbf{a}_{k-1}), & k = 2 \dots N_p, \\ \dot{\boldsymbol{\sigma}}(t_k, \mathbf{a}_k) = \dot{\boldsymbol{\sigma}}(t_k, \mathbf{a}_{k-1}), & k = 2 \dots N_p, \\ \ddot{\boldsymbol{\sigma}}(t_k, \mathbf{a}_k) = \ddot{\boldsymbol{\sigma}}(t_k, \mathbf{a}_{k-1}), & k = 2 \dots N_p, \end{cases}$$

where $\mathbf{a}_k = [a_{1,0,k} \dots a_{1,m,k}, a_{2,0,k} \dots a_{2,m,k}, a_{3,0,k} \dots a_{3,m,k}]^T$.

[1] Louembet, C., et al. IET Control Theory and Applications (2009).

NLP description II

- **Minimal rotation path**: between consecutive nodes $\theta_{rot} \in [-\pi, \pi]$
- **Torque constraint discretization**: grid each interval k with n_M subintervals of duration $T_M = T/n_M$

$$\begin{aligned} -M_{max} \leq M_i(t_{k,l}, \mathbf{a}_k) \leq M_{max}, \qquad & i = 1, 2, 3, \\ t_{k,l} = t_0 + (k-1)T + lT_M, \qquad & l = 0 \ldots n_M. \end{aligned}$$

Compact formulation

Compact formulation[1]: **stack vectors**

$$\mathbf{x_S} = [\mathbf{x}_1^T,\ \mathbf{x}_2^T, \dots, \mathbf{x}_{N_p}^T]^T,$$
$$\mathbf{u_S} = [u_1,\ u_2, \dots, u_{N_p}]^T,$$
$$\mathbf{a_S} = [\mathbf{a}_1^T,\ \mathbf{a}_2^T, \dots, \mathbf{a}_{N_p}^T]^T.$$

and **stack matrices**

$$\mathbf{F} = [\mathbf{A}^T,\ (\mathbf{A}^2)^T, \dots, (\mathbf{A}^{N_p})^T]^T,\ \ \mathbf{G}_{ik} = \mathbf{A}^{i-k}\mathbf{B}\mathbf{R}_{\mathbf{a}_k}\mathbf{v}.$$

Dynamics compactly expressed as:

$$\boxed{\mathbf{x_S} = \mathbf{F}\mathbf{x}_0 + \mathbf{G}(\mathbf{a_S})\mathbf{u_S}}$$

[1]Vazquez, R., et al. Control Engineering Practice (2017).

Discrete optimization problem I

Finite dimension **static program** in compact formulation (NLP)

$$
\begin{aligned}
\underset{\mathbf{u_S},\,\mathbf{a_S}}{\text{minimize}} \quad & \|\mathbf{u_S}\|_1, \\
\text{subject to} \quad & \mathbf{A}_{LS}\mathbf{G}(\mathbf{a_S})\mathbf{u_S} \leq \mathbf{b}_{LS} - \mathbf{F}\mathbf{x}_0, \\
& \mathbf{0} \leq \mathbf{u_S} \leq \mathbf{u_S}_{max}, \\
& -M_{max} \leq M_i(t_{k,l}, \mathbf{a}_k) \leq M_{max}, \\
& \mathbf{A}_{rend}\mathbf{G}(\mathbf{a_S})\mathbf{u_S} = -\mathbf{A}_{rend}\mathbf{F}\mathbf{x}_0, \\
& \boldsymbol{\sigma}(t_0, \mathbf{a}_1) = \boldsymbol{\sigma}_0, \\
& \dot{\boldsymbol{\sigma}}(t_0, \mathbf{a}_1) = \dot{\boldsymbol{\sigma}}_0, \\
& \boldsymbol{\sigma}(t_f, \mathbf{a}_{N_p}) = \boldsymbol{\sigma}_f, \\
& \dot{\boldsymbol{\sigma}}(t_f, \mathbf{a}_{N_p}) = \mathbf{0}, \\
& \mathbf{A}_{C2}\mathbf{a_S} = \mathbf{0}, \\
& \mathbf{f}_{rot}(\mathbf{a_S}) \leq \mathbf{0}.
\end{aligned}
$$

Discrete optimization problem II

Initial guess (hotstart):

1. 3DoF rendezvous posed as a linear programming (LP) problem (6 thusters assumed).
2. LP solution converted to NLP decision variables $\mathbf{u_S}$ & $\mathbf{a_S}$:
   1. $u_k = \|\mathbf{u}_{\mathrm{LP},k}\|_2 \rightarrow \mathbf{u_S}$
   2. Attitude coefficients from the required u_k impulse orientation at the nodes

$$\mathbf{v}_{k_i} = [u_{x,k_i}, u_{y,k_i}, u_{z,k_i}]^T / \|\mathbf{u}_{\mathrm{LP},k}\|_2, \quad \text{if} \quad \|\mathbf{u}_{\mathrm{LP},k}\|_2 > 0,$$

   3. Rotation angle and axis:

$$\theta_{k_i} = \mathrm{acos}(\mathbf{v}_{k_i} \cdot \mathbf{v}_{k_{i-1}}), \quad \mathbf{e}_{k_i} = \frac{\mathbf{v}_{k_i} \times \mathbf{v}_{k_{i-1}}}{\|\mathbf{v}_{k_i} \times \mathbf{v}_{k_{i-1}}\|_2}.$$

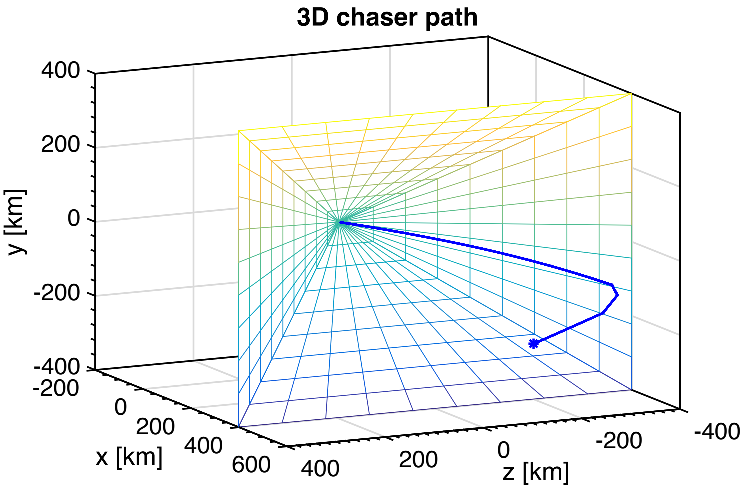
   4. From θ_{k_i} and $\mathbf{e}_{k_i}$ obtain MRP $\rightarrow \mathbf{a_S}$ (see details in paper)
   5. If u_k=0 attitude interpolated between non-zero impulses.

Outline

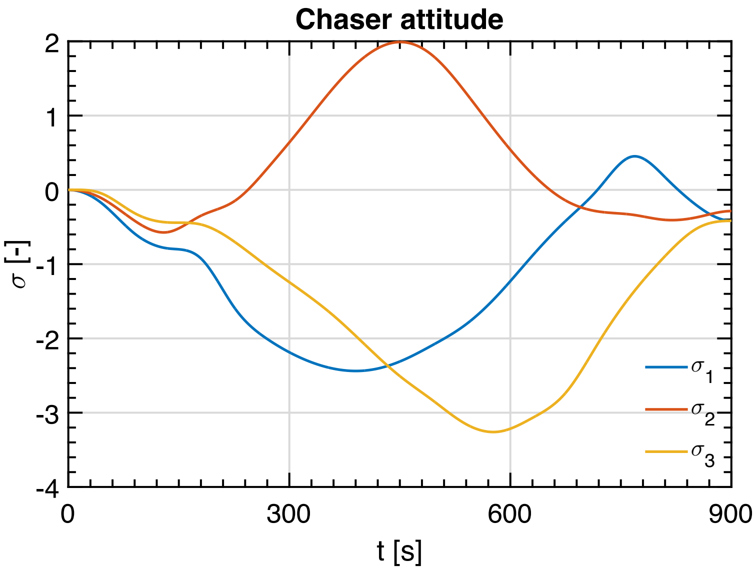
Simulation parameters

- **Target parameters**: h=600 km, y_0=z_0=2.5 m, c_y=c_z=$1/\tan(\pi/4)$.
- **Chaser parameters**: $\mathbf{I} = \mathrm{diag}(28, 45, 49)\text{kg} \cdot \text{m}^2$,$u_{max} = 1$ m/s, $M_{max} = 0.02$ N·m,$\mathbf{v} = [0, 0, -1]^T$.
- **Manoeuvre conditions**: t_f=900 s, $\mathbf{x}(0)$=$[400, -250, -200]^T$ m, $\dot{\mathbf{x}}(0) = [1, 1, -1]^T$ m/s, $\boldsymbol{\omega}(0) = [0, 0, 0]^T s^{-1}$, $\theta_1(0) = \theta_2(0) = \theta_3(0) = 0$, $\theta_1(t_f)$=0, $\theta_2(t_f)$=-$\pi/2$, $\theta_3(t_f) \equiv$ free. (1,2,3 Euler angles sequence).
  (Thruster nozzle pointing towards the $+x$ axis at the end to avoid plume impingement).
- **Planning parameters**: N_p=20, T=45 s, n_M=12, T_M=3.75 s, m=3.
- **Linear solver**: GUROBI (<1 second). **Nonlinear solver**: IPOPT (1.5 minutes; 260 decision variables and $\sim$ 1700 constraints). Routines integrated in Matlab.

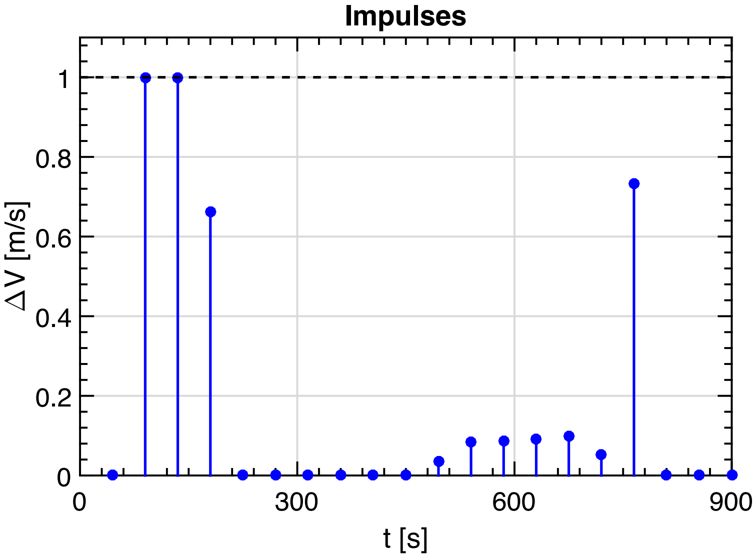
Simulation results I



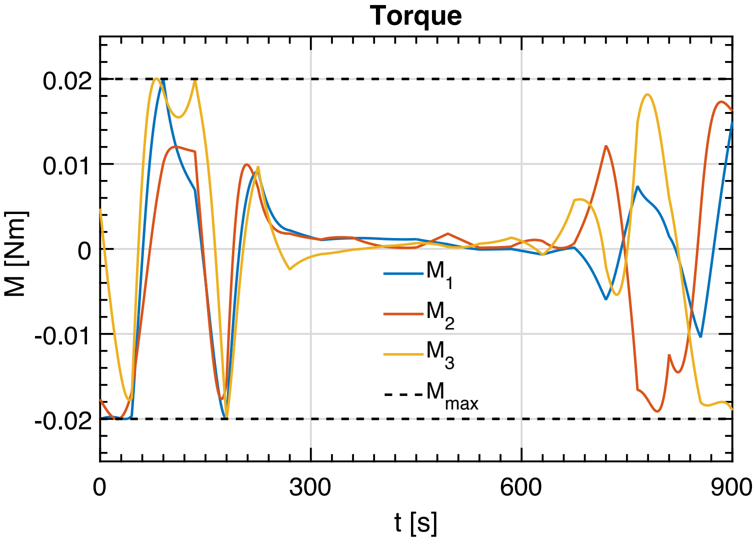
Simulation results II



Simulation results III



Simulation results IV



Outline

Conclusions

- We have presented a **rendezvous trajectory planning algorithm for a single-thruster spacecraft** equipped with **ACS**.
- Solution based on **translational state transition matrix** + **attitude flatness property** $\longrightarrow$ exact description.
- Problem is **discretized** and posed as NLP. No need of numerical integration.
- Formulation **extendeds** to **arbitrary number of thrusters**.
- As future work, **MPC scheme** based on linearization around the computed solution $\rightarrow$ deal with unmodelled dynamics and disturbances.
- Extension to **constellation formation flying** w/ relative attitude objectives also possible

Outline

1. ~~Introduction to MPC (Slides by E.F. Camacho)~~
2. ~~Application to Spacecraft Rendezvous (including PWM)~~
3. ~~Rendezvous + Attitude Control~~
4. Soft Landing on an Asteroid
5. Guidance for UAVs

A Predictive Guidance Algorithm for Autonomous Asteroid Soft Landing

Julio Cesar Sanchez Merino
Francisco Gavilan Jimenez
Rafael Vazquez Valenzuela

Universidad de Sevilla. Sevilla, Spain

jsanchezm@us.es, fgavilan@us.es, rvazquez1@us.es

May 29, 2019

Table of contents

Outline

1. Introduction
2. Asteroid modelling
3. Landing problem
4. Optimal control computation
5. MPC Guidance
6. Results
7. Conclusions

Introduction

Soft Landing

A **soft landing** is any type of aircraft, rocket or spacecraft landing that does not result in damages to the vehicle or anything on board.

Objective

The objective of this work is to present an **autonomous guidance algorithm** for soft-landing on an asteroid.

Methodology

The resolution approach is based on constraints **convexification**[1], **discretization** and an **iterative method**[2] . Then, this approach is embedded in a decreasing horizon MPC scheme.

[1] Acikmese, B., et al. Journal of Guidance, Control and Dynamics (2007).
[2] Pinson, R., et al. AAS/AIAA Astrodynamics Specialist Conference (2015).

Outline

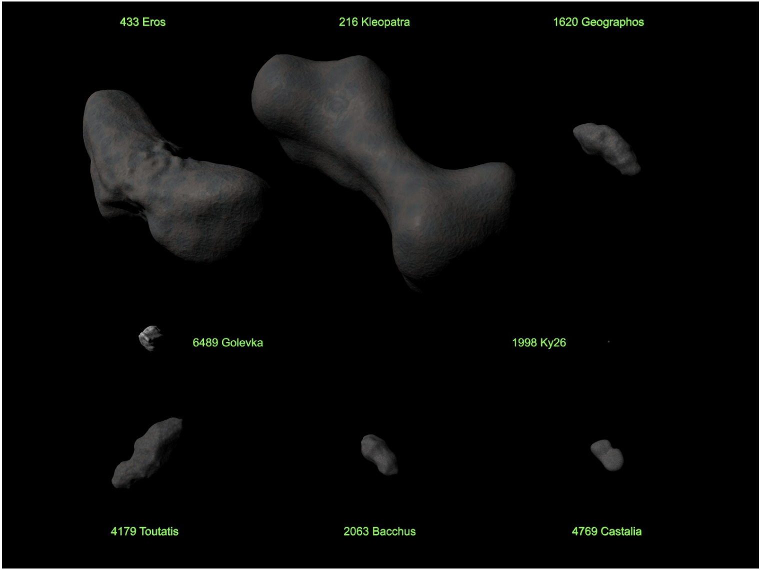
Asteroid modelling I

- **Asteroid fixed frame** in principal inertia axes (z major axis, x minor)

$$\begin{cases} \ddot{\mathbf{r}} & = -\dot{\boldsymbol{\omega}} \times \mathbf{r} - 2\boldsymbol{\omega} \times \dot{\mathbf{r}} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) + (\mathbf{F} + \mathbf{T})/m, \\ \dot{m} & = -\|\mathbf{T}\|_2 / v_{ex}, \end{cases}$$

  where $\mathbf{r}=[x, y, z]^T$ relative position, m lander mass, $\boldsymbol{\omega}$ asteroid rotation rate, $\mathbf{T}$ thrust, $\mathbf{F}$ external forces on the lander, v_{ex} the escape gases velocity.
- Most relevant external force: **asteroid central gravity field**, $\mathbf{F}=\mathbf{F}_g=m\nabla U_g$.

Asteroid modelling II



Asteroid modelling III

- **Polyhedron model**[1]: exact potential of a polyhedron shape body w/ constant density

$$U_g = \frac{G\rho}{2}\left(\sum_{e\in\text{edges}} \mathbf{r}_e^T \mathbf{E}_e \mathbf{r}_e L_e - \sum_{f\in\text{faces}} \mathbf{r}_f^T \mathbf{F}_f^T \mathbf{r}_f \omega_f\right).$$

("reality" in the simulation)

- **Mass-concentrations model**[2]: discrete masses

$$U_g = \sum_{i=1}^{n} \frac{Gm_i}{\|\mathbf{r} - \mathbf{r}_i\|_2}.$$

used to compute controls (lower computational load)

[1] Werner, R.A., et al. Celestial Mechanics and Dynamical Astronomy (1996).
[2] Kubota, T., et al. ISAS 16^{th} Workshop on Astrodynamics and Flight Mechanics (2006).

Outline

Constraints I

- **Thrust bounds**: engine cannot be shut down when turned on ($T_{min}>0$)

$$T_{min} \leq \|\mathbf{T}(t)\|_2 \leq T_{max}.$$

- **Fuel consumption**:

$$m(t) \geq m_{dry}.$$

- **Surface avoidance**:
Circumnavigation phase[1] (rotating tangent plane to the minimum volume ellipsoid)

$$(\mathbf{r}(t) - \mathbf{r}_t(t))^T \mathbf{n}_t^T \geq 0, \quad t \in [t_0, t_0 + t_{circ}].$$

Landing phase (line of sight form landing point)

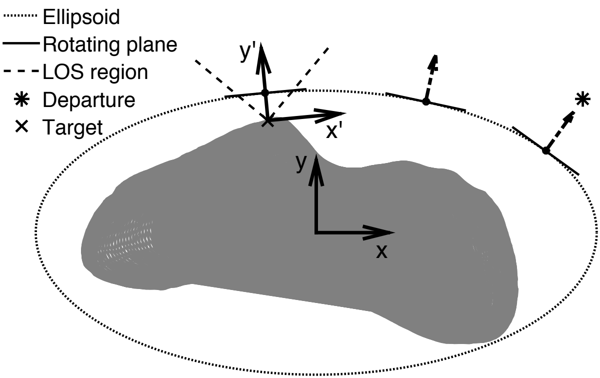
$$\mathbf{A}_L(\mathbf{r}(t) - \mathbf{r}_F) \leq \mathbf{b}_L, \quad t \in (t_0 + t_{circ}, t_f].$$

- **Terminal constraints**:

$$\mathbf{r}(t_f) = \mathbf{r}_f, \ \ \mathbf{v}(t_f) = \mathbf{0}.$$

[1] Dunham, W., et al. American Control Conference (2016).

Asteroid surface avoidance constraint[1] illustration



[1]Dunham, W., et al. American Control Conference (2016).

Objective function

Minimize fuel consumption (maximize the final mass value)

$$
\begin{array}{llcll}
\min\limits_{\mathbf{T}(t)} & -m(t_f), & & & \\
\text{s.t.} & \dot{\mathbf{r}}(t) & = & \mathbf{v}, & \\
 & \dot{\mathbf{v}}(t) & = & -2\omega \times \mathbf{v} - \omega \times (\omega \times \mathbf{r}) + \mathbf{T}/m & \\
 & & & +\nabla U_g(\mathbf{r}), & \\
 & \dot{m}(t) & = & -\|\mathbf{T}\|_2 / v_{ex}, & \\
 & \|\mathbf{T}(t)\|_2 & \leq & T_{max}, & \\
 & \|\mathbf{T}(t)\|_2 & \geq & T_{min}, & \\
 & m(t) & \geq & m_{dry}, & \\
 & \mathbf{r}_t^T(t)\mathbf{n}_t(t) & \leq & \mathbf{r}^T(t)\mathbf{n}_t(t), & t \in [t_0, t_0 + t_{circ}], \\
 & \mathbf{A}_L \mathbf{r}(t) & \leq & \mathbf{b}_L - \mathbf{A}_L \mathbf{r}_F, & t \in (t_0 + t_{circ}, t_f], \\
 & \mathbf{r}(t_f) & = & \mathbf{r}_F, & \\
 & \mathbf{v}(t_f) & = & \mathbf{0}. &
\end{array}
$$

Outline

Change of variables I

- Non-convex thrust constraint:

$$\boxed{\mathbf{a}_t = \mathbf{T}/m, \;\; a_{tm} = \|\mathbf{T}\|_2/m.}$$

- Mass variable:

$$\boxed{q = \ln(m) \;\longrightarrow\; \dot{q} = -a_{tm}/v_{ex} \quad \text{linear!}}$$

This change of variables **relaxes**[1] the non-convex thrust lower bound:

$$T_{min}e^{-q} \leq a_{tm} \leq T_{max}e^{-q},$$
$$\|\mathbf{a}_t\|_2 \leq a_{tm} \;\longrightarrow\; \text{SOCP!}$$

The mass term can be linearized as $e^{-q} \approx e^{-q_r}[1 - (q - q_r)]$.

[1] Acikmese, B., et al. Journal of Guidance, Control and Dynamics (2007).

Change of variables II

$$
\begin{array}{rrcl}
\min\limits_{\mathbf{a}_t, a_{tm}} & -q(t_f), & & \\
\text{s.t.} & \dot{\mathbf{r}}(t) & = & \mathbf{v}, \\
 & \dot{\mathbf{v}}(t) & = & -2\omega \times \mathbf{v} - \omega \times (\omega \times \mathbf{r}) + \mathbf{a}_t \\
 & & & +\nabla U_g(\mathbf{r}), \\
 & \dot{q}(t) & = & -a_{tm}/v_{ex}, \\
 & q(t) & \geq & q_{dry}, \\
 & a_{tm}(t) & \geq & T_{min} e^{-q_r(t)}[1 - (q(t) - q_r(t))], \\
 & a_{tm}(t) & \leq & T_{max} e^{-q_r(t)}[1 - (q(t) - q_r(t))], \\
 & \|\mathbf{a}_t(t)\|_2 & \leq & a_{tm}(t), \\
 & \mathbf{r}_t^T(t)\mathbf{n}_t(t) & \leq & \mathbf{r}^T(t)\mathbf{n}_t(t), \quad t \in [t_0, t_0 + t_{circ}], \\
 & \mathbf{A}_L \mathbf{r}(t) & \leq & \mathbf{b}_L - \mathbf{A}_L \mathbf{r}_F, \quad t \in (t_0 + t_{circ}, t_f], \\
 & \mathbf{r}(t_f) & = & \mathbf{r}_F, \\
 & \mathbf{v}(t_f) & = & \mathbf{0}.
\end{array}
$$

Note that the constraints are **linear or second-order cones**. The asteroid gravity field is the sole non-linearity of the model.

Discretization I

The asteroid gravity field non-linearities are tackled using an **iterative process** where the gravity terms are evaluated with the last iteration data

$$\boxed{\dot{\mathbf{x}}^{[j]} = \mathbf{A}\mathbf{x}^{[j]} + \mathbf{B}\mathbf{u}^{[j]} + \mathbf{c}(\mathbf{r}^{[j-1]}),\ \ \mathbf{x} = [\mathbf{r}^T, \mathbf{v}^T, q]^T,\ \ \mathbf{u} = [\mathbf{a}_t^T, a_{tm}]^T,}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \omega^2 & 0 & 0 & 0 & 2\omega & 0 & 0 \\ 0 & \omega^2 & 0 & -2\omega & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},\ \ \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -v_{ex}^{-1} \end{bmatrix},$$

$$\mathbf{c} = -\sum_{i=1}^{n} \frac{Gm_i}{\|\mathbf{r} - \mathbf{r}_i\|_2^3} [0,\ 0,\ 0,\ (x - x_i),\ (y - y_i),\ (z - z_i),\ 0]^T.$$

Discretization II

Manoeuvre is **discretized** into N intervals of duration $\Delta T = (t_f - t_0)/N$
Trapezoidal integration rule to obtain the states at the nodes

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta T[\mathbf{A}(\mathbf{x}_k + \mathbf{x}_{k-1}) + \mathbf{B}(\mathbf{u}_k + \mathbf{u}_{k-1}) + \mathbf{c}_k + \mathbf{c}_{k-1}]/2,$$

solving for $\mathbf{x}_k$

$$\boxed{\mathbf{x}_k = \mathbf{C}\mathbf{x}_{k-1} + \mathbf{D}(\mathbf{u}_k + \mathbf{u}_{k-1}) + \mathbf{E}(\mathbf{c}_k + \mathbf{c}_{k-1}),}$$

where

$$\begin{aligned} \mathbf{C} &= (\mathbf{I} - \Delta T\mathbf{A}/2)^{-1}(\mathbf{I} + \Delta T\mathbf{A}/2), \\ \mathbf{D} &= (\mathbf{I} - \Delta T\mathbf{A}/2)^{-1}\Delta T\mathbf{B}/2, \\ \mathbf{E} &= (\mathbf{I} - \Delta T\mathbf{A}/2)^{-1}\Delta T/2. \end{aligned}$$

Discretization III

Compact formulation[1]

$$\mathbf{x_S} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T]^T, \;\; \mathbf{u_S} = [\mathbf{u}_0^T, \ldots, \mathbf{u}_N^T]^T,$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}^2 \\ \vdots \\ \mathbf{C}^N \end{bmatrix}, \;\; \mathbf{H} = \begin{bmatrix} \mathbf{E}(\mathbf{c}_1 + \mathbf{c}_0) \\ \mathbf{E}(\mathbf{c}_2 + \mathbf{c}_1) + \mathbf{CE}(\mathbf{c}_1 + \mathbf{c}_0) \\ \vdots \\ \sum_{j=1}^N \mathbf{C}^{N-j}\mathbf{E}(\mathbf{c}_j + \mathbf{c}_{j-1}) \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{D} & \mathbf{D} & \mathbf{\Theta}_{7\times4} & \ldots & \mathbf{\Theta}_{7\times4} \\ \mathbf{CD} & (\mathbf{I}+\mathbf{C})\mathbf{D} & \mathbf{D} & \ldots & \mathbf{\Theta}_{7\times4} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}^{N-1}\mathbf{D} & \mathbf{C}^{N-2}(\mathbf{I}+\mathbf{C})\mathbf{D} & \mathbf{C}^{N-3}(\mathbf{I}+\mathbf{C})\mathbf{D} & \ldots & \mathbf{D} \end{bmatrix}.$$

[1] Vazquez, R., et al. Control Engineering Practice (2017).

Discretization IV

Dynamics in compact formulation:

$$\mathbf{x_S} = \mathbf{F}\mathbf{x}_0 + \mathbf{G}\mathbf{u_S} + \mathbf{H}.$$

Discrete optimization problem (SOCP) in compact formulation

$$\begin{array}{lrcll}
\min\limits_{\mathbf{u_S}} & -q_N, & & & \\
\text{s.t.} & \mathbf{A}_{Tmin}\mathbf{u_S} & \geq & \mathbf{b}_{Tmin}, & \\
 & \mathbf{A}_{Tmax}\mathbf{u_S} & \leq & \mathbf{b}_{Tmax}, & \\
 & \|\mathbf{a}_{t,k}\|_2 & \leq & a_{tm,k}, & k = 1 \dots N, \\
 & \mathbf{A}_{CS}\mathbf{x_S} & \leq & \mathbf{b}_{CS}, & \\
 & \mathbf{A}_{LS}\mathbf{x_S} & \leq & \mathbf{b}_{LS}, & \\
 & \mathbf{A}_{M}\mathbf{x_S} & \leq & \mathbf{b}_{M}, & \\
 & \mathbf{r}_N & = & \mathbf{r}_F, & \\
 & \mathbf{v}_N & = & \mathbf{0}. &
\end{array}$$

Iterative process

Iterative algorithm[1]:

1. Evaluate asteroid gravity with the initial spacecraft position, $\mathbf{r}_0$. Consider the vehicle flying at minimum thrust so initial mass reference is $m_{r,k}=m_0 - k\Delta T(T_{min}/v_{ex})$.
2. Compute a solution of the SOCP problem, $\mathbf{u_S}^{[j]} \longrightarrow \mathbf{r}_k^{[j]}, \mathbf{v}_k^{[j]}, m_k^{[j]}$.
3. Go back to Step 2, using $\mathbf{r}_k^{[j-1]}$ and $m_k^{[j-1]}$ to update asteroid gravity and mass, until $\max(\mathbf{r}_k^{[j-1]} - \mathbf{r}_k^{[j-2]}) <$Tol or $j > j_{max}$.

[1] Pinson, R., et al. AAS/AIAA Astrodynamics Specialist Conference (2015).

Outline

MPC Guidance

Autonomous landing requires a **closed-loop scheme** to cope with model uncertainties and disturbances.
A **decreasing horizon MPC**, relaxing terminal constraints to costs, is proposed

$$J_{MPC} = -q_N + \gamma_r(\mathbf{r}_N - \mathbf{r}_F)^T \mathbf{I}(\mathbf{r}_N - \mathbf{r}_F) + \gamma_v \mathbf{v}_N^T \mathbf{I} \mathbf{v}_N.$$

1. Use the presented iterative algorithm to start at $k = 0$ and planning horizon N.
2. Apply the commanded thrust for the current interval k. Decrease the planning horizon by one.
3. Since disturbances perturb the planned path, from the reached point recompute control using J_{MPC} and without terminal constraints. Go back to Step 2 until the planning horizon ends.

Outline

- **Asteroid 433 Eros parameters**: ρ=2.67 g/cm^3, T_{rot}=5.27 h.
- **Lander parameters**[1]: m_0=600 kg, m_{dry}=487 kg, T_{max}=80 N, T_{min}=20 N, v_{ex}=2000 m/s.
- **Manoeuvre parameters**: $\mathbf{r}_F=[-0.5114,\ -2.836,\ 1.443]^T$ km, $\mathbf{r}_0=[0,\ 35,\ 0]^T$ km, $\mathbf{v}_0=[-3.5709,\ 0,\ 0]^T$ m/s, t_f=2000 s, t_{circ}=1500 s, $x'_0=z'_0$=10 m, $c_{x'}=c_{z'}=1/\tan(\pi/4)$.
- **Mascons model parameters**: n=4841 (equidistant), $m_i=\rho V/n$.
- **Polyhedron model parameters**[2]: 25350 vertexes and 49152 faces.
- **Controller parameters**: N=100, ΔT=20 s, N_C=75, $\gamma_r=\gamma_v$=100, j_{max}=6, Tol$=0.02\|\mathbf{r}_F\|_2$.

[1] Lantoine, G., AE8900 MS Special Problems Report (2006).
[2] Gaskell, R.W., NASA Planetary Data System (2008).

Simulation parameters II

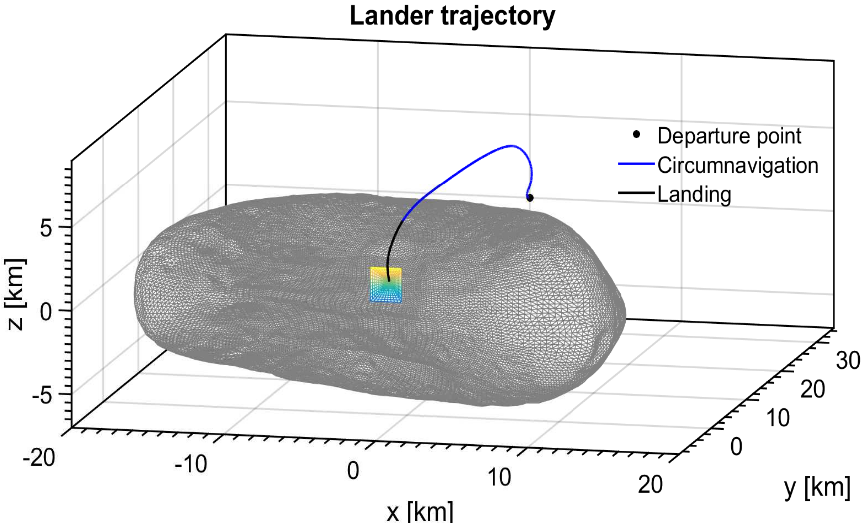
- Disturbances[1] on each **thruster component** are added as

$$\mathbf{T}_{real} = \mathbf{\Omega}(\boldsymbol{\delta\theta})[\mathbf{T}_{comm}(1+\boldsymbol{\delta}) + \boldsymbol{\delta}\mathbf{T}],$$

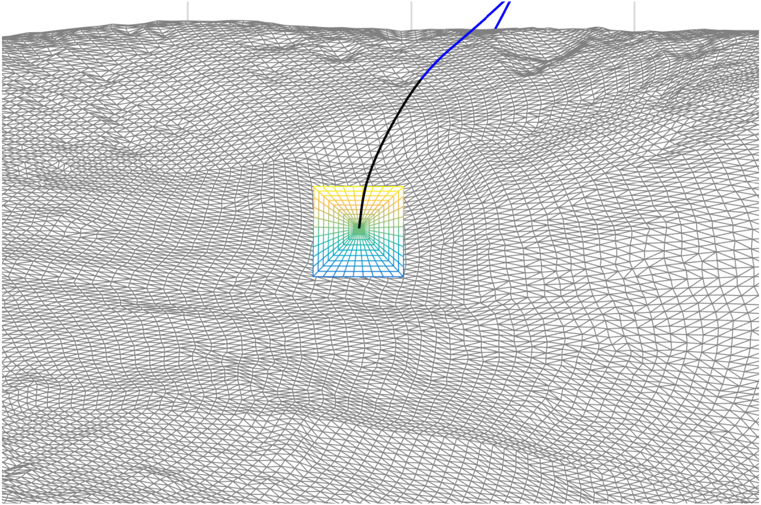
  where $\boldsymbol{\delta\theta}$ is a vector of random small angles, $\boldsymbol{\delta}$ is a vector of random multiplicative noises and $\boldsymbol{\delta}\mathbf{T}$ is a vector of additive noises.
- The disturbances **model several physical aspects** such as alignment errors, thrust noises or even unmodeled dynamics as SRP, sun gravity, etc.
- **Disturbances parameters** (normal distributions): $\bar{\boldsymbol{\delta\theta}}=\mathbf{0}$, $\bar{\boldsymbol{\delta}}=[0.01, 0.01, 0.01]^T$, $\bar{\boldsymbol{\delta}\mathbf{T}}=[0.01, 0.01, 0.01]^T T_{max}$, $\mathbf{\Sigma}_{\delta\theta,ij}=0.0436\delta_{ij}$, $\mathbf{\Sigma}_{\delta,ij}=0.05\delta_{ij}$, $\mathbf{\Sigma}_{\delta T,ij}=0.02 T_{max}\delta_{ij}$.

[1] Gavilan, F., et al. Control Engineering Practice (2012).

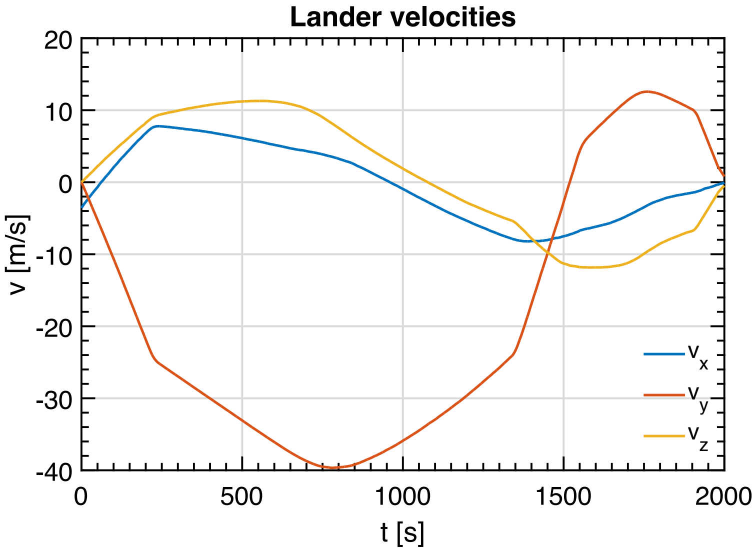
Simulation results I



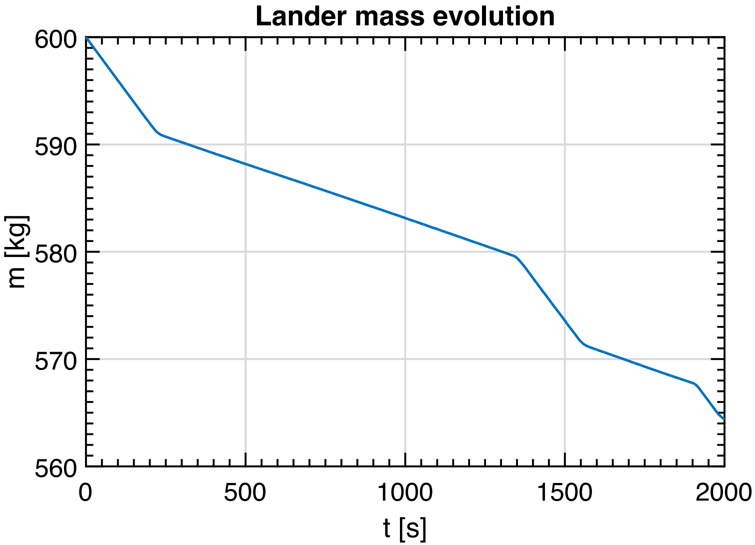
Simulation results II



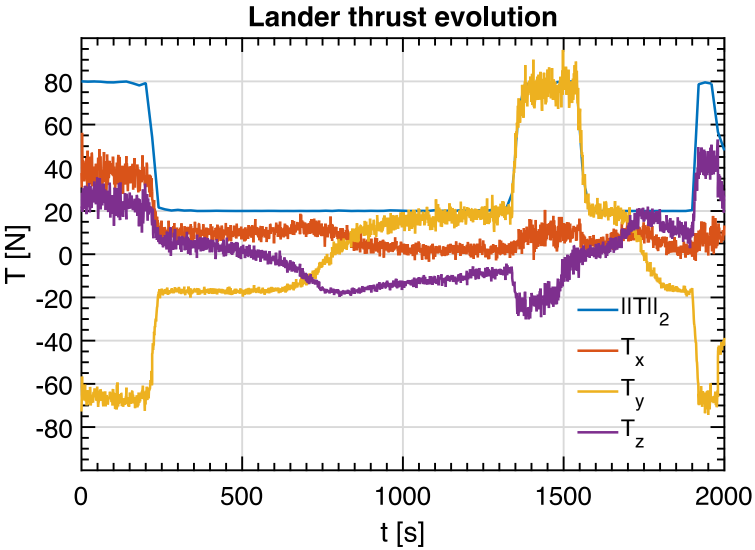
Simulation results III



Simulation results IV



Simulation results V



Outline

Conclusions

- A **MPC guidance algorithm** to autonomously land powered probes on small bodies while handling with unmodelled dynamics and disturbances has been presented.
- **Lossless convexification**, **discretization** and a **successive solution method** were features of the solution.
- Future work may include **comparisons** with other state of the art methods, a detailed **sensitivity** analysis with problem parameters as well as including the circumnavigation and landing durations as decision variables.
- Additionally a **six-degrees of freedom** model shall be considered. The lander would have an ACS (e.g. reaction wheels or a RCS) to control its orientation.

Outline

1. ~~Introduction to MPC (Slides by E.F. Camacho)~~
2. ~~Application to Spacecraft Rendezvous (including PWM)~~
3. ~~Rendezvous + Attitude Control~~
4. ~~Soft Landing on an Asteroid~~
5. Guidance for UAVs ←

A High-Level Model Predictive Control Guidance Law for Unmanned Aerial Vehicles

Model Predictive Control for Aerospace Applications

Francisco Gavilán, Rafael Vazquez and Eduardo F. Camacho

Departamento de Ingeniería Aeroespacial y Mecánica de Fluidos

Universidad de Sevilla

Outline

1. Introduction
2. High level Guidance System
3. Simulations
4. Conclusions

Introduction

Our goal:

- Design a path-following guidance system for airplane autonomous operation. Main features:
  - Follows a reference
  - Prescribed flying times
  - Must take wind into account

The challenge:

- Nonlinear model
- Disturbances entering the system (wind)
- Feasible control solution must be available at any sampling time

Proposed solution:

- Hierarchical control architecture to handle system complexity
- High level control: airplane guidance
  - Makes the airplane follow a reference trajectory, computing high level commands (velocity/flight path angle/bank angle)
  - Iterative Model Predictive Control: uses robust backup L1 navigation to compute a "hotstart" solution, refined in a iterative optimization process
- Low level control: airplane stabilization and high level (velocity/flight path angle/bank angle) reference seeking (outside of the scope of this presentation)

Outline

Model Predictive Guidance for UAVs

Model Predictive Control

Using a prediction law $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)$, compute the sequence of control signals along the prediction horizon $\mathbf{u}_k, \mathbf{u}_{k+1}, \ldots, \mathbf{u}_{k+N_p-1}$, which optimizes the desired cost function.

MPC for UAV guidance

1. Discretization.
   - Sampling time $T_s = 1\,\mathrm{s}$
2. Prediction law:

$$\mathbf{x}_\mathrm{S} = f(\mathbf{u}_\mathrm{S}, \mathbf{x}_0) \qquad \mathbf{x}_\mathrm{S} = [\mathbf{x}_1\,\mathbf{x}_2\,\ldots\,\mathbf{x}_{N_p}]^T \qquad \mathbf{u}_\mathrm{S} = [\mathbf{u}_0\,\mathbf{u}_1\,\ldots\,\mathbf{u}_{N_p-1}]^T$$

3. Computation of the optimal control sequence:

$$\min_{\mathbf{u}_\mathrm{S}} J(\mathbf{x}_\mathrm{S}(\mathbf{u}_\mathrm{S}) - \mathbf{x}_{\mathrm{ref},\mathrm{S}}, \mathbf{u}_\mathrm{S})$$

3 DoF airplane model:

$$\frac{\mathrm{d}\,x}{\mathrm{d}\,t} = V\cos\gamma\cos\chi + w_x,$$
$$\frac{\mathrm{d}\,y}{\mathrm{d}\,t} = V\cos\gamma\sin\chi + w_y,$$
$$\frac{\mathrm{d}\,z}{\mathrm{d}\,t} = -V\sin\gamma,$$

Nonlinear optimization problem

Model Predictive Guidance for UAVs

Model Predictive Control

Using a prediction law $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)$, compute the sequence of control signals along the prediction horizon $\mathbf{u}_k, \mathbf{u}_{k+1}, \ldots, \mathbf{u}_{k+N_p-1}$, which optimizes the desired cost function.

MPC for UAV guidance

1. Discretization.
   - Sampling time $T_s = 1\,\mathrm{s}$
2. Prediction law:

$$\mathbf{x_S} = f(\mathbf{u_S}, \mathbf{x}_0) \qquad \mathbf{x_S} = [\mathbf{x}_1\,\mathbf{x}_2\,\ldots\,\mathbf{x}_{N_p}]^T \qquad \mathbf{u_S} = [\mathbf{u}_0\,\mathbf{u}_1\,\ldots\,\mathbf{u}_{N_p-1}]^T$$

3. Computation of the optimal control sequence:

$$\min_{\mathbf{u_S}} J(\mathbf{x_S}(\mathbf{u_S}) - \mathbf{x}_{\mathbf{ref},\mathbf{S}}, \mathbf{u_S})$$

3 DoF airplane model:

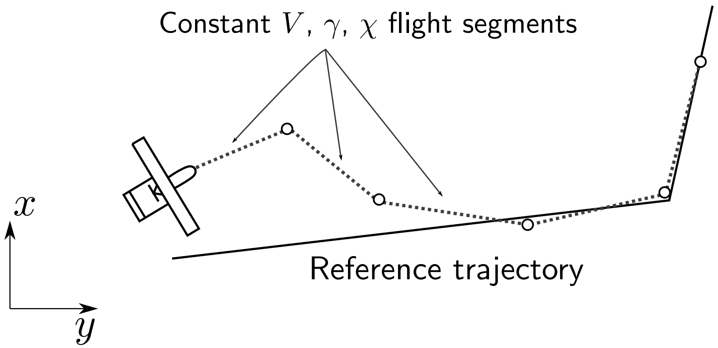
$$\frac{\mathrm{d}\,x}{\mathrm{d}\,t} = V\cos\gamma\cos\chi + w_x,$$
$$\frac{\mathrm{d}\,y}{\mathrm{d}\,t} = V\cos\gamma\sin\chi + w_y,$$
$$\frac{\mathrm{d}\,z}{\mathrm{d}\,t} = -V\sin\gamma,$$

Nonlinear optimization problem

Discretization of equations of motion

Classic approaches:

$$
\begin{aligned}
x_{k+1} &= V_k \cos\gamma_k \cos\chi_k + x_k, \\
y_{k+1} &= V_k \cos\gamma_k \sin\chi_k + y_k, \\
z_{k+1} &= -V_k \sin\gamma_k + z_k.
\end{aligned}
$$



- Constant heading flight segments, inputs V_k, γ_k, χ_k
- Instantaneous turns: *not feasible*

Proposed discretization

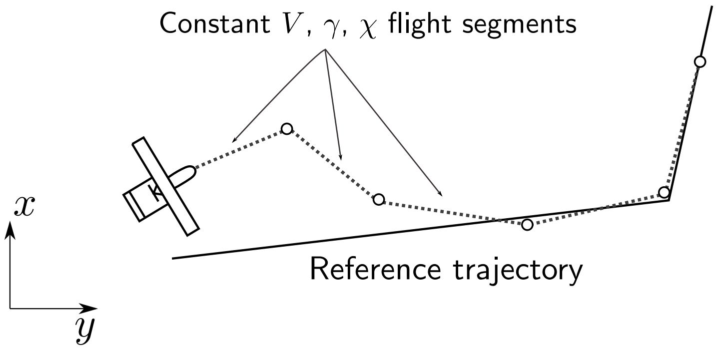
$$
\begin{aligned}
x_{k+1} &= \frac{V_k \cos\gamma_k T_s}{\kappa_k}\left(\sin(\kappa_k + \chi_k) - \sin\chi_k\right) + x_k, \\
y_{k+1} &= \frac{V_k \cos\gamma_k T_s}{\kappa_k}\left(\cos\chi_k - \cos(\kappa_k + \chi_k)\right) + y_k, \\
z_{k+1} &= -V_k \sin\gamma_k T_s + z_k, \\
\chi_{k+1} &= \kappa_k + \chi_k, \qquad \kappa_k = \frac{g\tan\phi_k}{V_k}.
\end{aligned}
$$

- Constant curvature flight segments: *realistic approach*
- The guidance algorithm handles turn control, inputs V_k, γ_k, $\kappa_k(\phi_k)$
- Quite more complex optimization problem

Discretization of equations of motion

Classic approaches:

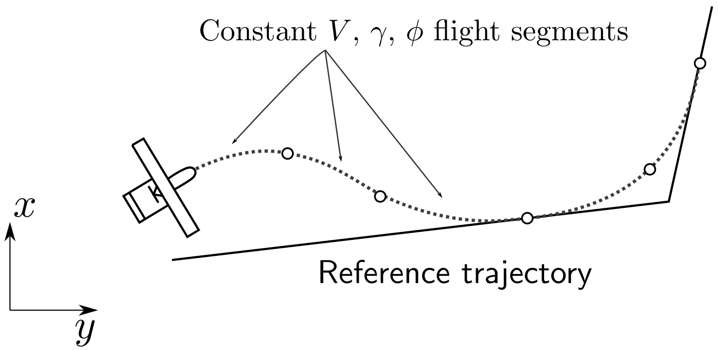
$$
\begin{aligned}
x_{k+1} &= V_k \cos\gamma_k \cos\chi_k + x_k, \\
y_{k+1} &= V_k \cos\gamma_k \sin\chi_k + y_k, \\
z_{k+1} &= -V_k \sin\gamma_k + z_k.
\end{aligned}
$$



- Constant heading flight segments, inputs V_k, γ_k, χ_k
- Instantaneous turns: *not feasible*

Proposed discretization

$$
\begin{aligned}
x_{k+1} &= \frac{V_k \cos\gamma_k T_s}{\kappa_k}\left(\sin\left(\kappa_k + \chi_k\right) - \sin\chi_k\right) + x_k, \\
y_{k+1} &= \frac{V_k \cos\gamma_k T_s}{\kappa_k}\left(\cos\chi_k - \cos\left(\kappa_k + \chi_k\right)\right) + y_k, \\
z_{k+1} &= -V_k \sin\gamma_k T_s + z_k, \\
\chi_{k+1} &= \kappa_k + \chi_k, \qquad \kappa_k = \frac{g\tan\phi_k}{V_k}.
\end{aligned}
$$

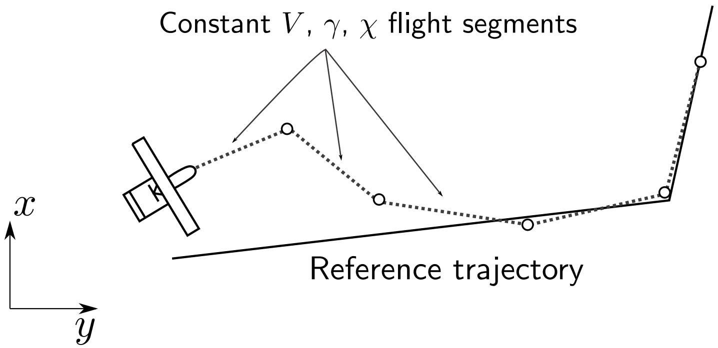


- Constant curvature flight segments: *realistic approach*
- The guidance algorithm handles turn control, inputs V_k, γ_k, $\kappa_k(\phi_k)$
- Quite more complex optimization problem

Discretization of equations of motion

Classic approaches:

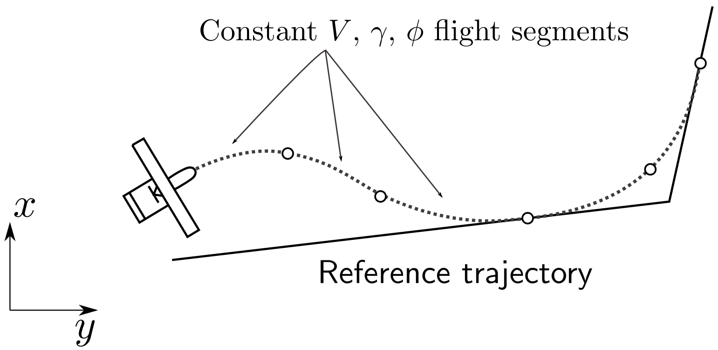
$$
\begin{aligned}
x_{k+1} &= V_k \cos\gamma_k \cos\chi_k + x_k,\\
y_{k+1} &= V_k \cos\gamma_k \sin\chi_k + y_k,\\
z_{k+1} &= -V_k \sin\gamma_k + z_k.
\end{aligned}
$$



- Constant heading flight segments, inputs V_k, γ_k, χ_k
- Instantaneous turns: *not feasible*

Proposed discretization

$$
\begin{aligned}
x_{k+1} &= \frac{V_k \cos\gamma_k T_s}{\kappa_k}\left(\sin(\kappa_k + \chi_k) - \sin\chi_k\right) + x_k,\\
y_{k+1} &= \frac{V_k \cos\gamma_k T_s}{\kappa_k}\left(\cos\chi_k - \cos(\kappa_k + \chi_k)\right) + y_k,\\
z_{k+1} &= -V_k \sin\gamma_k T_s + z_k,\\
\chi_{k+1} &= \kappa_k + \chi_k, \qquad \kappa_k = \frac{g\tan\phi_k}{V_k}.
\end{aligned}
$$

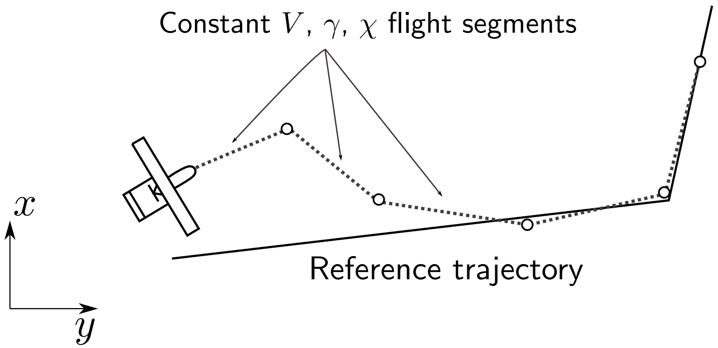


- Constant curvature flight segments: *realistic approach*
- The guidance algorithm handles turn control, inputs V_k, γ_k, $\kappa_k(\phi_k)$
- Quite more complex optimization problem

Discretization of equations of motion

Classic approaches:

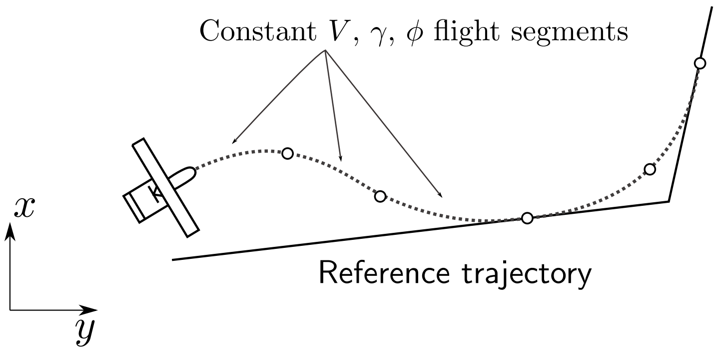
$$
\begin{aligned}
x_{k+1} &= V_k \cos\gamma_k \cos\chi_k + x_k,\\
y_{k+1} &= V_k \cos\gamma_k \sin\chi_k + y_k,\\
z_{k+1} &= -V_k \sin\gamma_k + z_k.
\end{aligned}
$$



- Constant heading flight segments, inputs V_k, γ_k, χ_k
- Instantaneous turns: *not feasible*

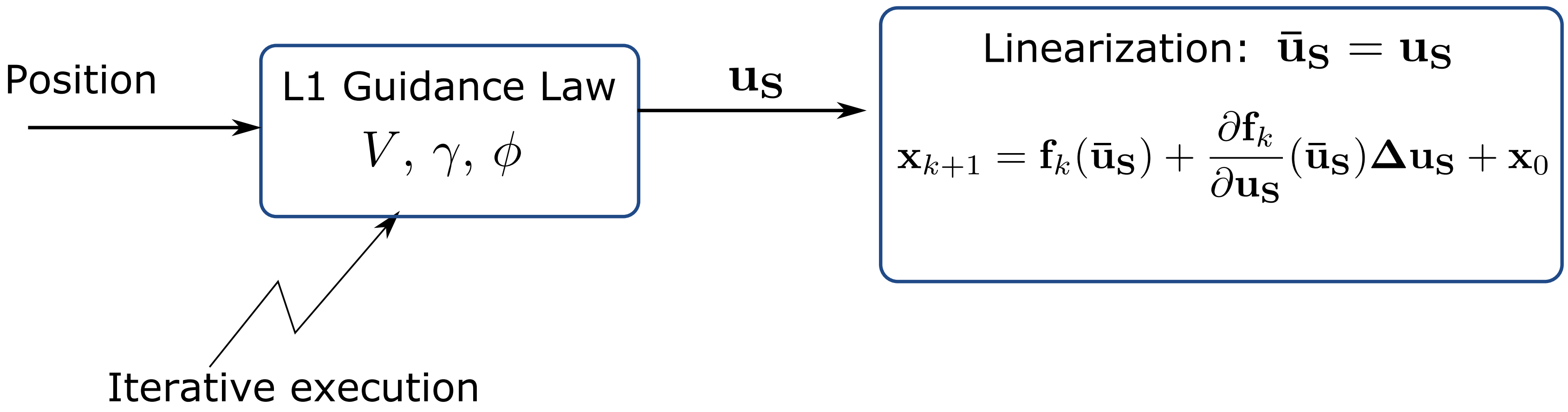
Proposed discretization

$$
\begin{aligned}
x_{k+1} &= \frac{V_k \cos\gamma_k T_s}{\kappa_k}\left(\sin(\kappa_k+\chi_k) - \sin\chi_k\right) + x_k,\\
y_{k+1} &= \frac{V_k \cos\gamma_k T_s}{\kappa_k}\left(\cos\chi_k - \cos(\kappa_k+\chi_k)\right) + y_k,\\
z_{k+1} &= -V_k \sin\gamma_k T_s + z_k,\\
\chi_{k+1} &= \kappa_k + \chi_k, \qquad \kappa_k = \frac{g\tan\phi_k}{V_k}.
\end{aligned}
$$

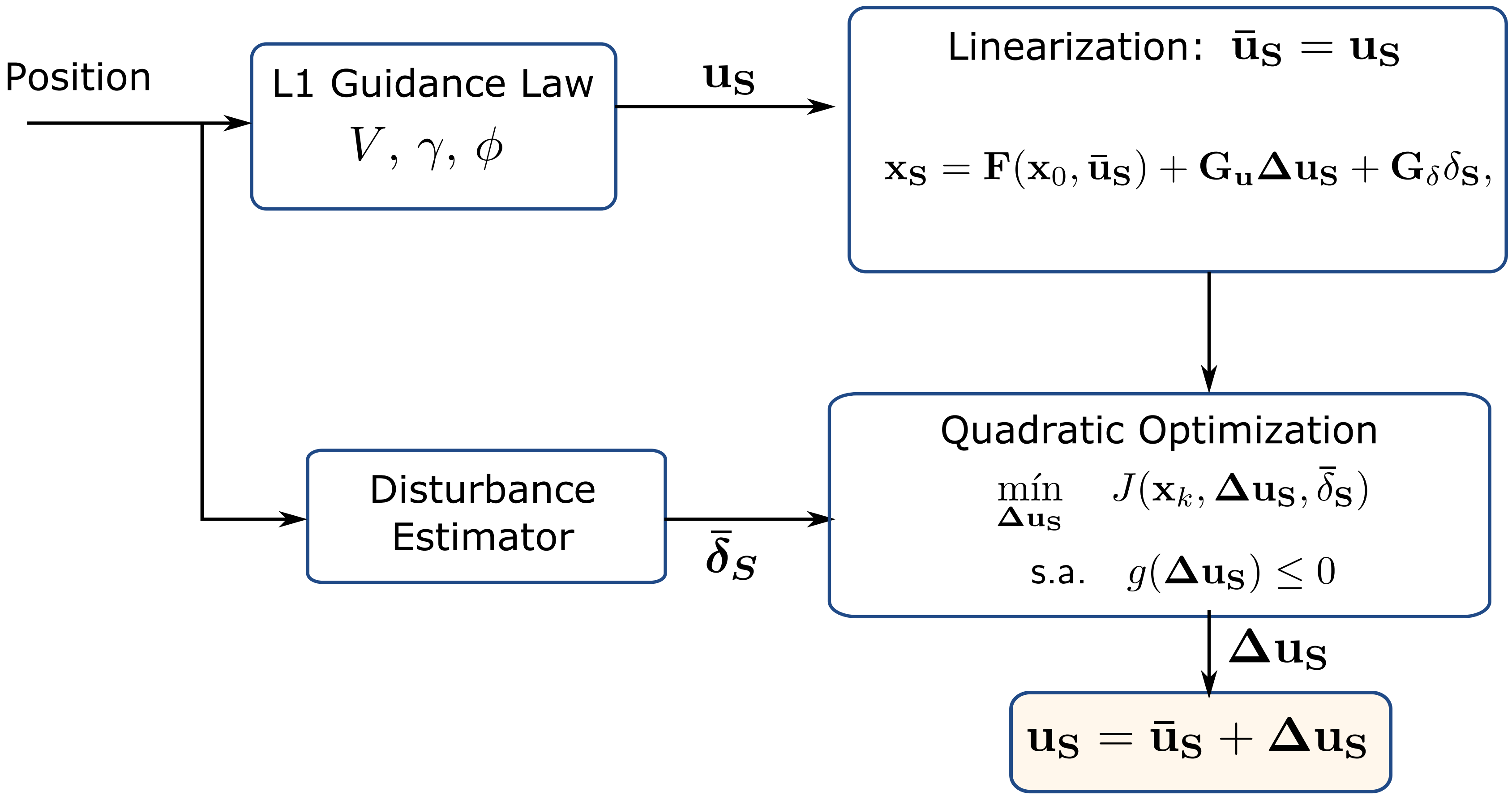


- Constant curvature flight segments: *realistic approach*
- The guidance algorithm handles turn control, inputs V_k, γ_k, $\kappa_k(\phi_k)$
- Quite more complex optimization problem

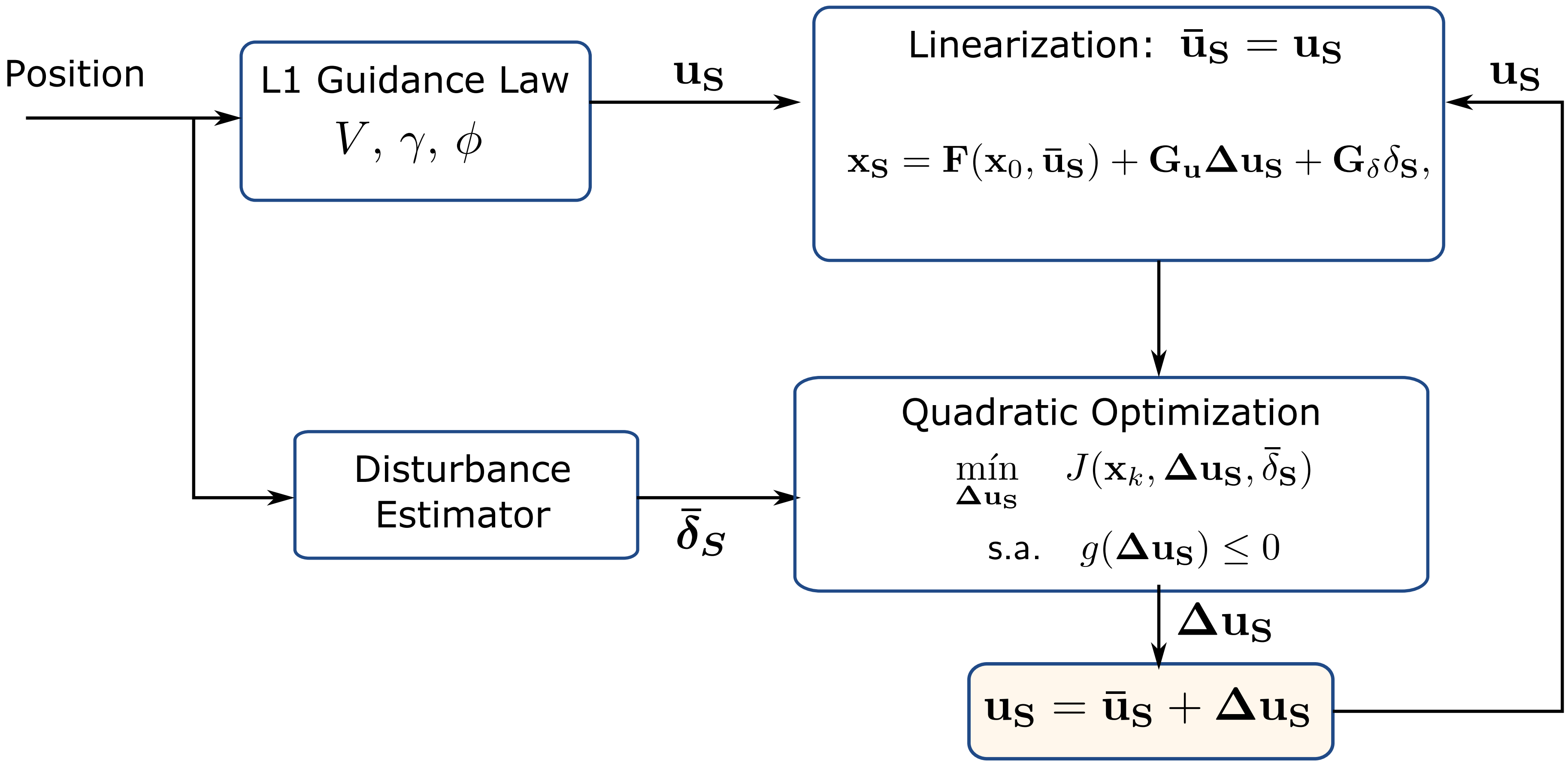
Proposed guidance strategy



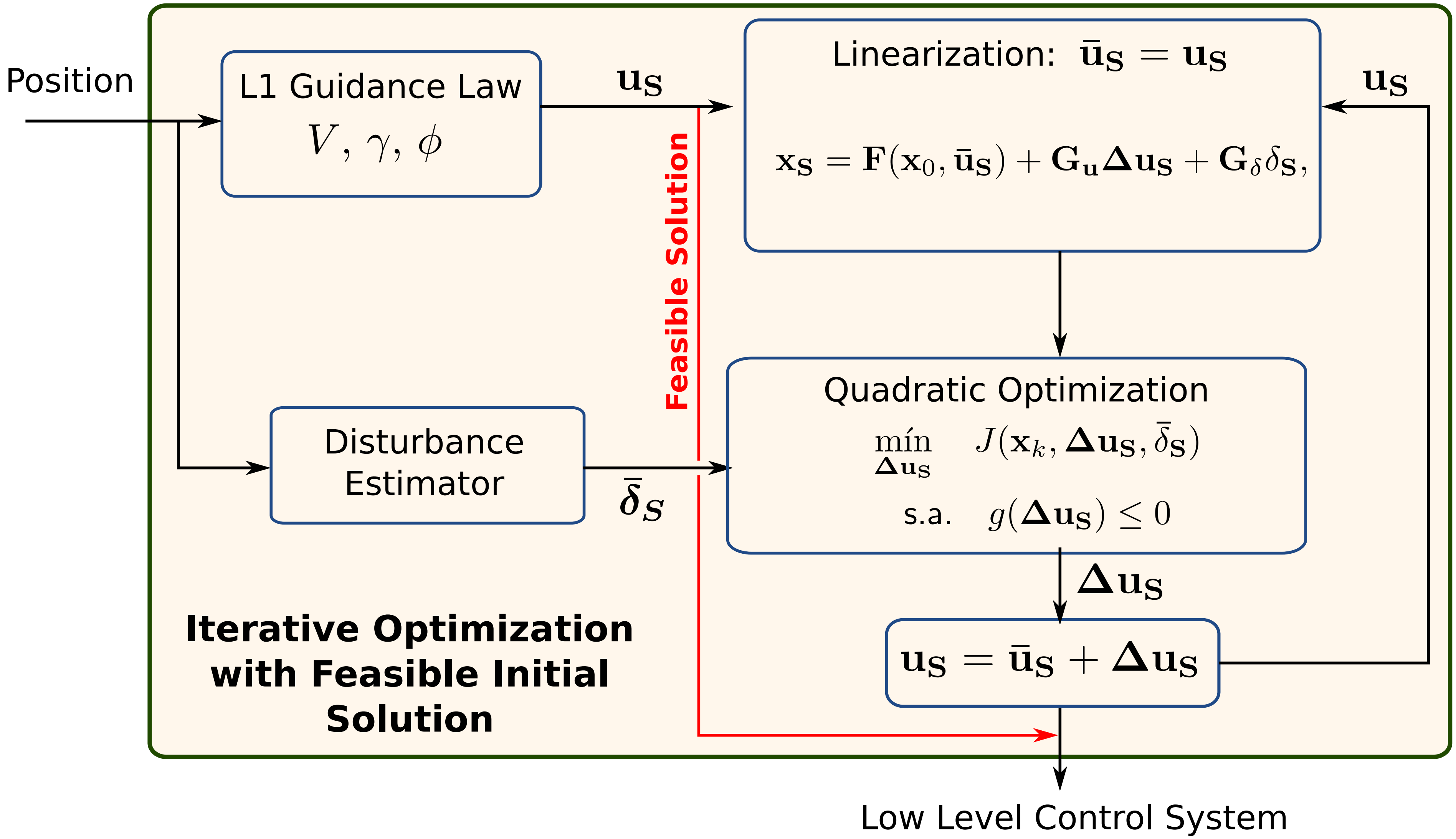
Proposed guidance strategy



Proposed guidance strategy

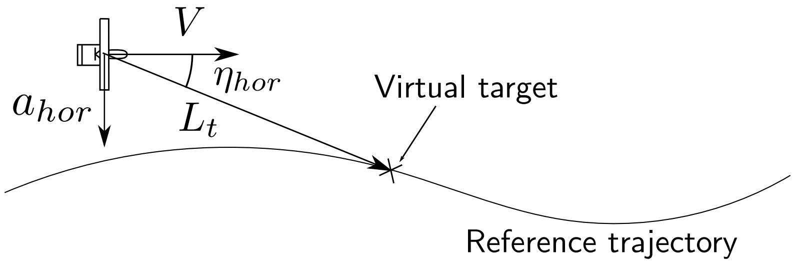


Proposed guidance strategy



Hotstart: L1 Navigation

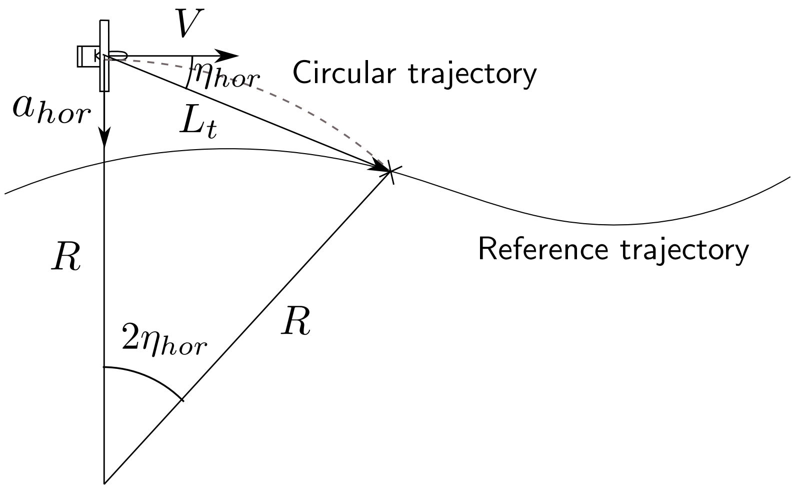
L1 Navigation



$$
\begin{aligned}
a_{hor} &= N_{hor}\frac{V^2}{L_t}\sin\eta_{hor},\\
a_{ver} &= N_{ver}\frac{V^2}{L_t}\sin\eta_{ver}.
\end{aligned}
$$

$$\Delta\chi \approx \frac{a_{hor}}{V}T_s,\quad \Delta\gamma \approx \frac{a_{ver}}{V}T_s,\quad \Delta V = V_{ref} - V.$$

L1 modification to include load factors at turns n



For circular flight segments:

$$
\begin{aligned}
n &= \sqrt{\left(\frac{2V^2\sin\eta_{hor}}{gL_t}\right)^2 + 1},\\
\phi &= \operatorname{sgn}(\sin\eta_{hor})\arccos\frac{1}{n}.
\end{aligned}
$$

Linearized model

Linearized prediction law:

$$\mathbf{x_S} = \mathbf{F}(\mathbf{x_0}, \bar{\mathbf{u}}_\mathbf{S}) + \mathbf{G_u}(\mathbf{x_0}, \bar{\mathbf{u}}_\mathbf{S})\boldsymbol{\Delta}\mathbf{u_S} + \mathbf{G}_\delta \delta_\mathbf{S}$$

$$\mathbf{x_S} = \begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T & \cdots & \mathbf{x}_{N_p}^T \end{bmatrix}^T$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_0(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) + \mathbf{x}_0 \\ \mathbf{f}_1(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) + \mathbf{f}_0(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) + \mathbf{x}_0 \\ \vdots \\ \mathbf{f}_{N_p-1}(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) + \cdots + \mathbf{f}_0(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) + \mathbf{x}_0 \end{bmatrix}, \quad \mathbf{G_u} = \begin{bmatrix} \frac{\partial \mathbf{f}_0}{\partial \mathbf{u_S}}(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) \\ \frac{\partial \mathbf{f}_1}{\partial \mathbf{u_S}}(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) + \frac{\partial \mathbf{f}_0}{\partial \mathbf{u_S}}(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) \\ \vdots \\ \frac{\partial \mathbf{f}_{N_p-1}}{\partial \mathbf{u_S}}(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) + \cdots + \frac{\partial \mathbf{f}_0}{\partial \mathbf{u_S}}(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) \end{bmatrix}.$$

- Explicit computation of $\mathbf{F}(\mathbf{x_0}, \bar{\mathbf{u}}_\mathbf{S})$ and $\mathbf{G_u}(\mathbf{x_0}, \bar{\mathbf{u}}_\mathbf{S})$
- Additive disturbances included

Constraints:

1. **Airplane limitations:** lower and upper bounds of the airspeed, flight path angle and bank angle. $\mathbf{u} = \begin{bmatrix} V & \gamma & \kappa \end{bmatrix}^T$
2. **Linearization constraints:** control signals are bounded to ensure that the linearization holds

$$-\delta\mathbf{u} \leq \boldsymbol{\Delta}\mathbf{u}_k \leq \delta\mathbf{u}$$

with $\delta u = [\delta V,\ \delta\gamma,\ \delta\kappa]^T$

Disturbance estimator

- The prediction law requires values for $\bar{\boldsymbol{\delta}}_i$.
- Simple approach to compute from past disturbances:

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

- Where $\hat{\boldsymbol{\delta}}_k$ is the estimate of $\bar{\boldsymbol{\delta}}_k$ and $\lambda > 0$ is a *forgetting factor*.
- This can be written recursively by defining $\hat{\boldsymbol{\delta}}_0 = 0$,

$$\hat{\boldsymbol{\delta}}_k = \frac{\mathrm{e}^{-\lambda}}{\rho_k} \left(\rho_{k-1} \hat{\boldsymbol{\delta}}_{k-1} + \boldsymbol{\delta}_{k-1} \right).$$

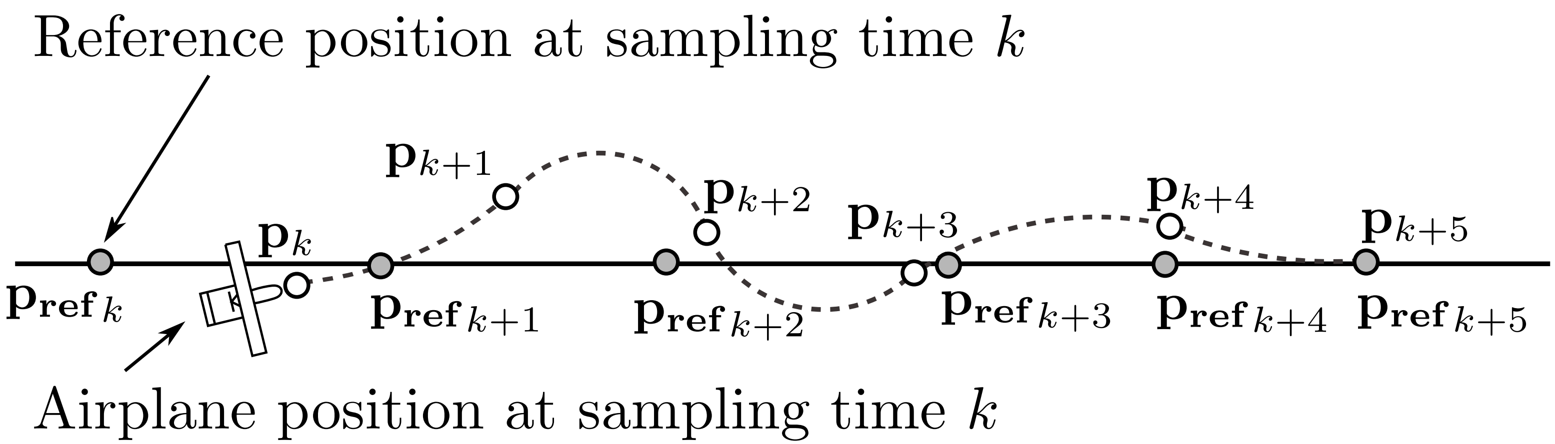
  where $\rho_k = \frac{\mathrm{e}^{-\lambda}\left(1-\mathrm{e}^{-\lambda k}\right)}{1-\mathrm{e}^{-\lambda}}$.
- Past disturbances are computed (approximately) by comparing the real airplane state at each sampling time and the expected state from the prediction in the previous sampling time:

$$\boldsymbol{\delta}_{k-1} = \mathbf{x}_k - \mathbf{f}_k(V_{k-1}, \gamma_{k-1}, \kappa_{k-1}, \chi_{k-1}) - \mathbf{x}_{k-1}.$$

- It is convenient to sample disturbances are sampled at a higher frequency than the main guidance law.

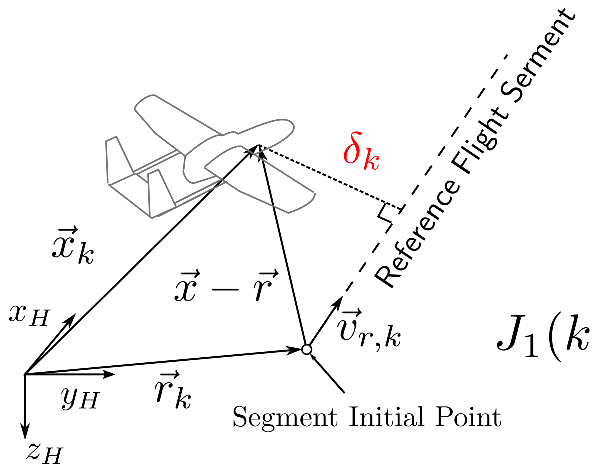
Cost Function

- If a standard quadratic cost penalizing the position error at each sampling time is used, we would minimize the difference between the trajectory and virtual waypoints.
- However, this approach might lead to an oscillatory trajectory:



- Thus, we propose an alternative approach, combining 3 different cost functions.

Cost Function

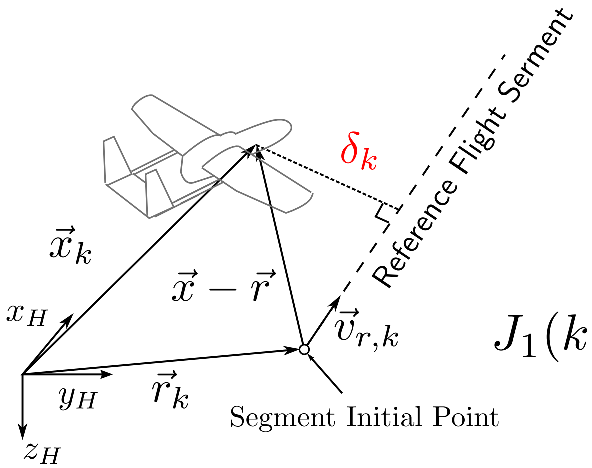


① Distance to the reference flight segment

$$\delta_k = \frac{\vec{v} \wedge (\vec{x}_k - \vec{r}_k)}{|\vec{v}_{r,k}|}$$

$$J_1(k) = \sum_{i=1}^{N_p} \left(\mathbf{V}_{k+i}(\hat{\mathbf{x}}_{k+i|k} - \mathbf{r}_{k+i})\right)^T \mathbf{R}_{1,k+i} \left(\mathbf{V}_{k+i}(\hat{\mathbf{x}}_{k+i|k} - \mathbf{r}_{k+i})\right)$$

Cost Function

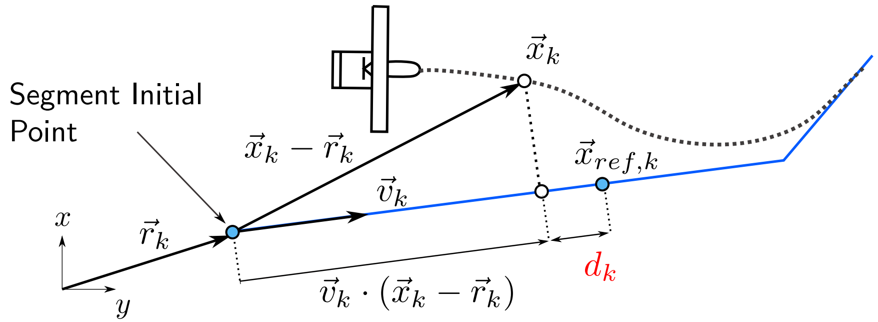


① Distance to the reference flight segment

$$\delta_k = \frac{\vec{v} \wedge (\vec{x}_k - \vec{r}_k)}{|\vec{v}_{r,k}|}$$

$$J_1(k) = \sum_{i=1}^{N_p} \left(\mathbf{V}_{k+i}(\hat{\mathbf{x}}_{k+i|k} - \mathbf{r}_{k+i})\right)^T \mathbf{R}_{1,k+i} \left(\mathbf{V}_{k+i}(\hat{\mathbf{x}}_{k+i|k} - \mathbf{r}_{k+i})\right)$$

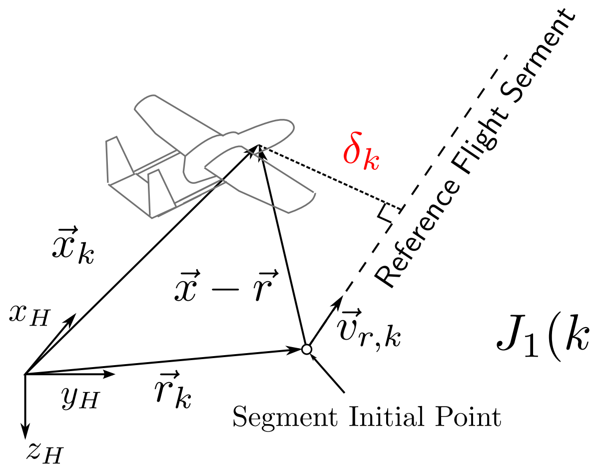
② Time synchronization



$$d_k = \vec{v}_{r,k} \cdot (\vec{x}_k - \vec{x}_{ref,k})$$

$$J_2(k) = \sum_{i=1}^{N_p} \left(\mathbf{v}_{k+i}(\hat{\mathbf{x}}_{k+i|k} - \mathbf{x}_{\mathbf{ref}\,k+i})\right)^T \mathbf{R}_{2,k+i} \left(\mathbf{v}_{k+i}(\hat{\mathbf{x}}_{k+i|k} - \mathbf{x}_{\mathbf{ref}\,k+i})\right)$$

Cost Function

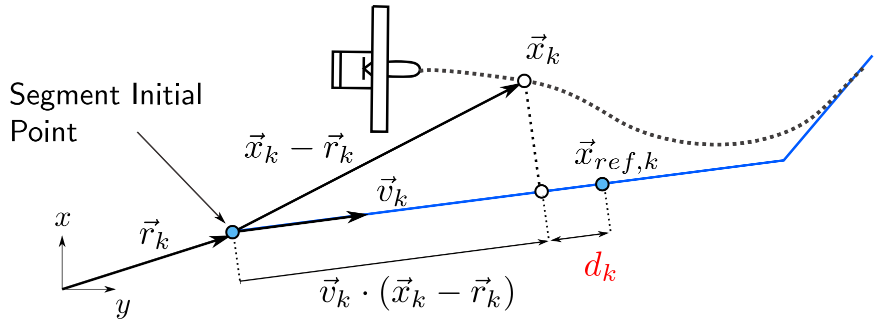


① Distance to the reference flight segment

$$\delta_k = \frac{\vec{v} \wedge (\vec{x}_k - \vec{r}_k)}{|\vec{v}_{r,k}|}$$

$$J_1(k) = \sum_{i=1}^{N_p} \left(\mathbf{V}_{k+i}(\hat{\mathbf{x}}_{k+i|k} - \mathbf{r}_{k+i})\right)^T \mathbf{R}_{1,k+i} \left(\mathbf{V}_{k+i}(\hat{\mathbf{x}}_{k+i|k} - \mathbf{r}_{k+i})\right)$$

② Time synchronization



$$d_k = \vec{v}_{r,k} \cdot (\vec{x}_k - \vec{x}_{ref,k})$$

$$J_2(k) = \sum_{i=1}^{N_p} \left(\mathbf{v}_{k+i}(\hat{\mathbf{x}}_{k+i|k} - \mathbf{x}_{\mathbf{ref}\,k+i})\right)^T \mathbf{R}_{2,k+i} \left(\mathbf{v}_{k+i}(\hat{\mathbf{x}}_{k+i|k} - \mathbf{x}_{\mathbf{ref}\,k+i})\right)$$

③ Avoid excessive control use

$$J_3(k) = \sum_{i=1}^{N_p-1} (\mathbf{u}_{k+i} - \mathbf{u}_{k+i-1})^T \mathbf{Q}_{k+i} (\mathbf{u}_{k+i} - \mathbf{u}_{k+i-1}) + (\mathbf{u}_k - \hat{\mathbf{u}}_k)^T \mathbf{Q}_k (\mathbf{u}_k - \hat{\mathbf{u}}_k)$$

Cost Function: how to choose weights

- The total cost function is a combination of the three cost functions

$$J(\mathbf{x}_k, \mathbf{\Delta u_S}) = J_{1,k} + J_{2,k} + J_{3,k}.$$

- The weights are chosen as

$$\begin{aligned} \mathbf{Q}_i &= k_Q \operatorname{diag}\left(\frac{1}{\delta V^2}, \frac{1}{\delta\gamma^2}, \frac{1}{\delta\kappa^2}\right), \\ \mathbf{R}_{1,i} &= k_{R_1}\zeta_i \operatorname{diag}(1,\ 1,\ 1,\ 0), \\ R_{2,i} &= k_{R_2}\zeta_i, \end{aligned}$$

where δV, $\delta\gamma$ and $\delta\kappa$ are the input bounds used in the constraints.

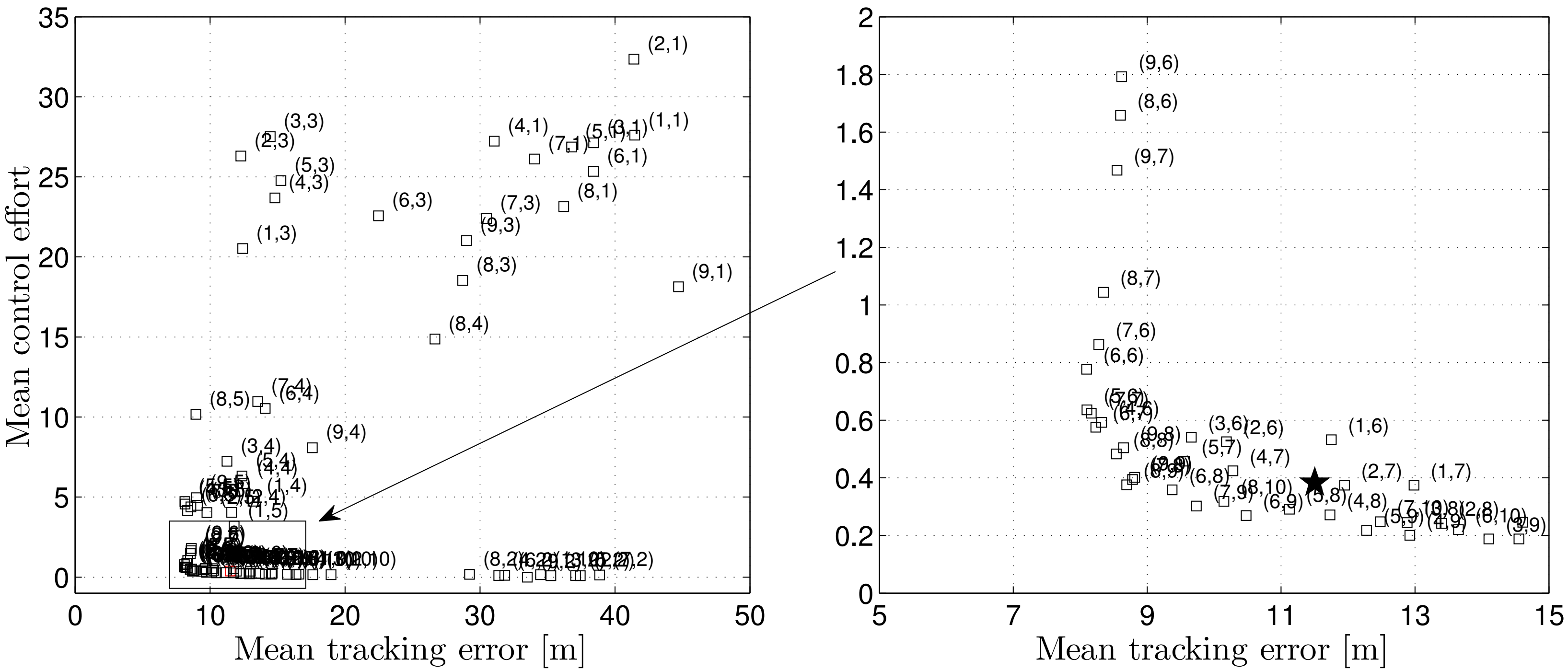
- ζ_i is a function introduced to avoid penalizing errors during the first sampling times:

$$\zeta_i = \begin{cases} 0, & \text{If } i \leq 3, \\ 1, & \text{If } i \in [4, N_p]. \end{cases}$$

- The scalar weights k_Q, k_{R_1} and k_{R_2} were chosen performing a Pareto analysis.

Cost function: how to choose weights

Pareto Analysis to choose relative weights.



Mean Control Effort: $CE = \frac{1}{N_{CE}} \sum_k \left[\left(\mathbf{u}_k - \hat{\mathbf{u}}_k\right)^T \bar{\mathbf{Q}}_k \left(\mathbf{u}_k - \hat{\mathbf{u}}_k\right) \right]^{\frac{1}{2}}$

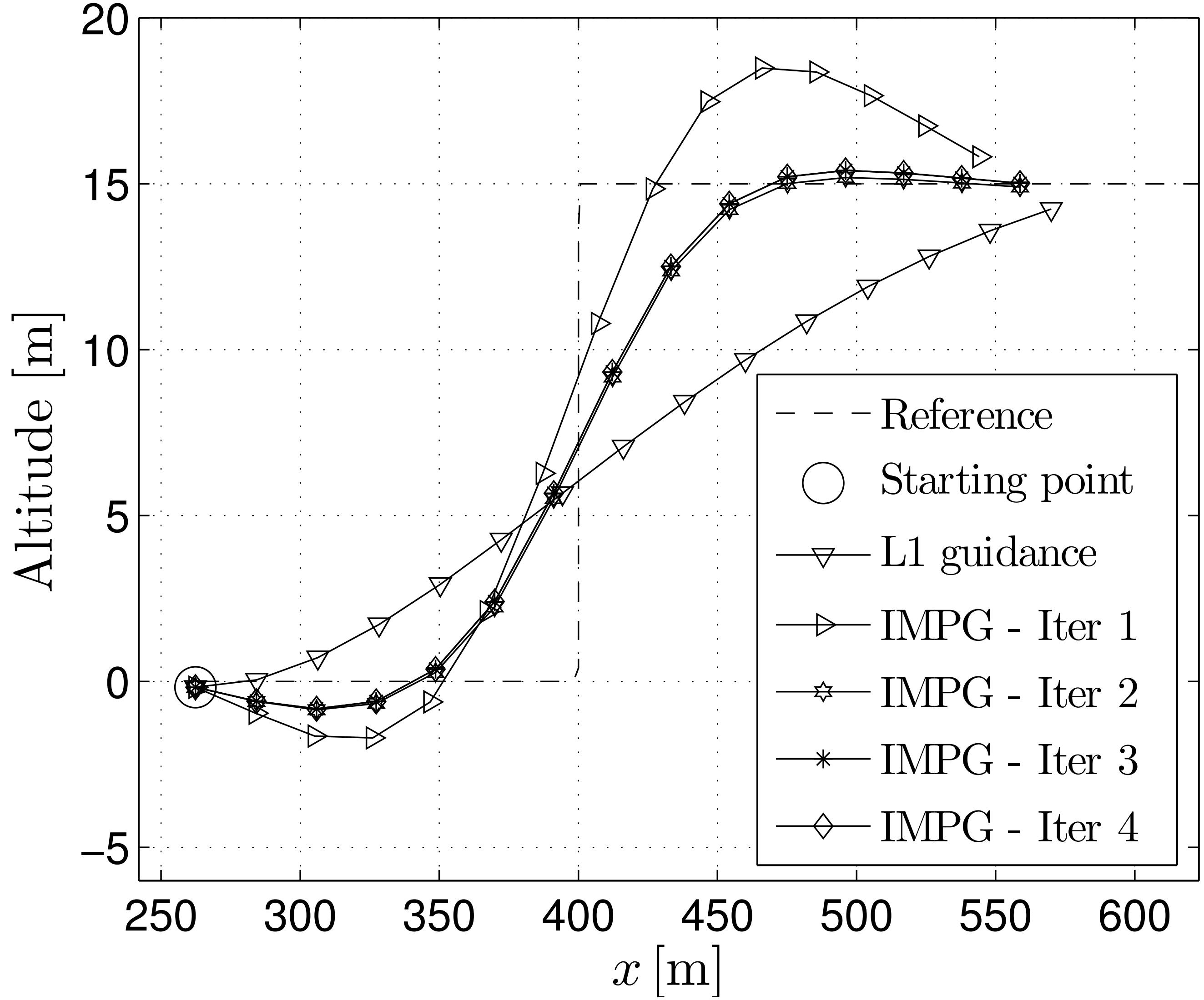
Mean Tracking performance: $TE = \frac{1}{N_{TE}} \sum_{t \in S} ||\mathbf{p}(t) - \mathbf{p_{ref}}(t)||_2$

Outline

1. Introduction
2. High level Guidance System
3. Simulations
4. Conclusions

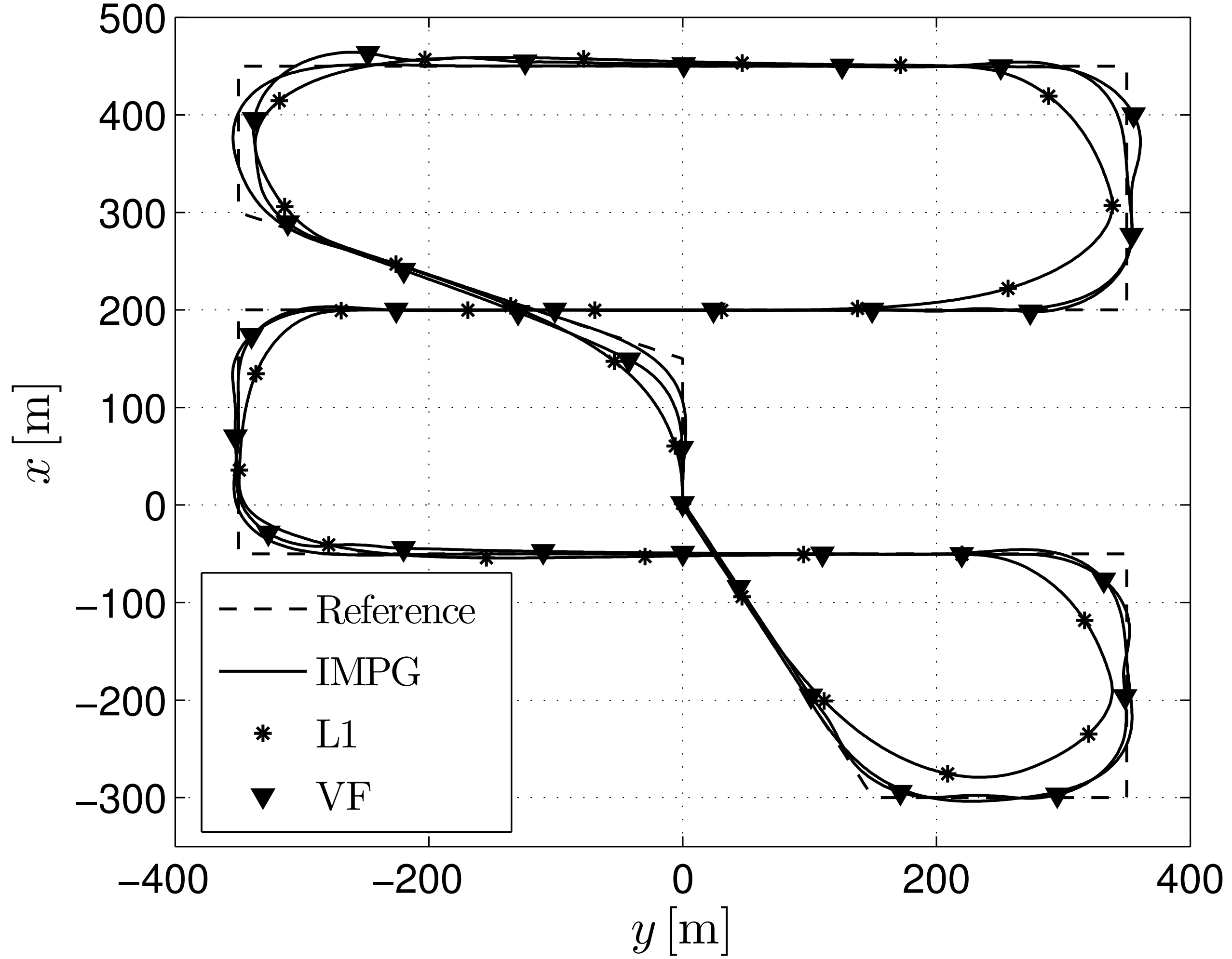
Simulations

Vertical profile of successive trajectories computed along the iterations.



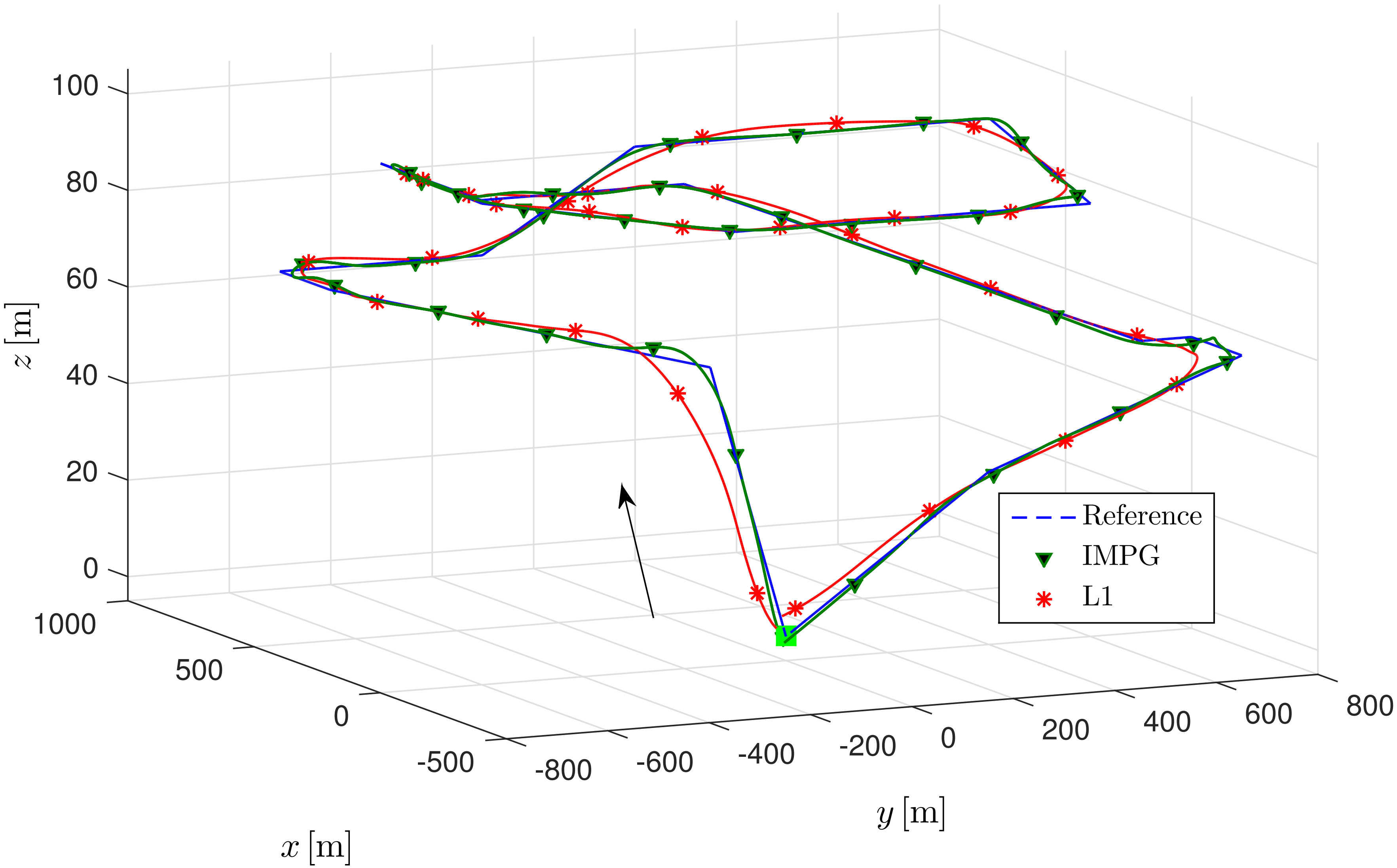
Simulations I

Comparison with vector field (VF) and L1 guidance for a plane mission



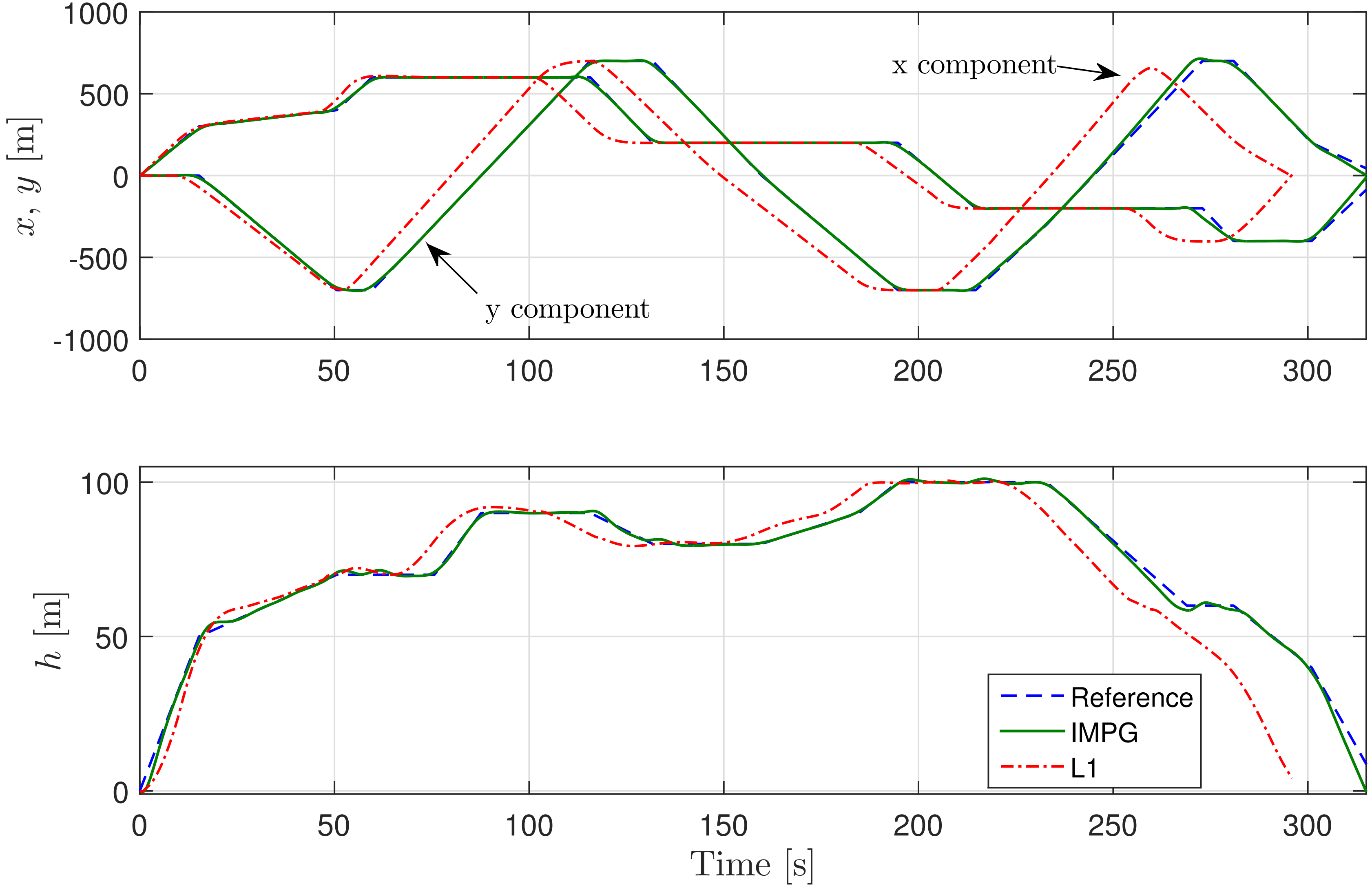
Simulations II

Trajectory for a 3-D surveillance mission (with wind)



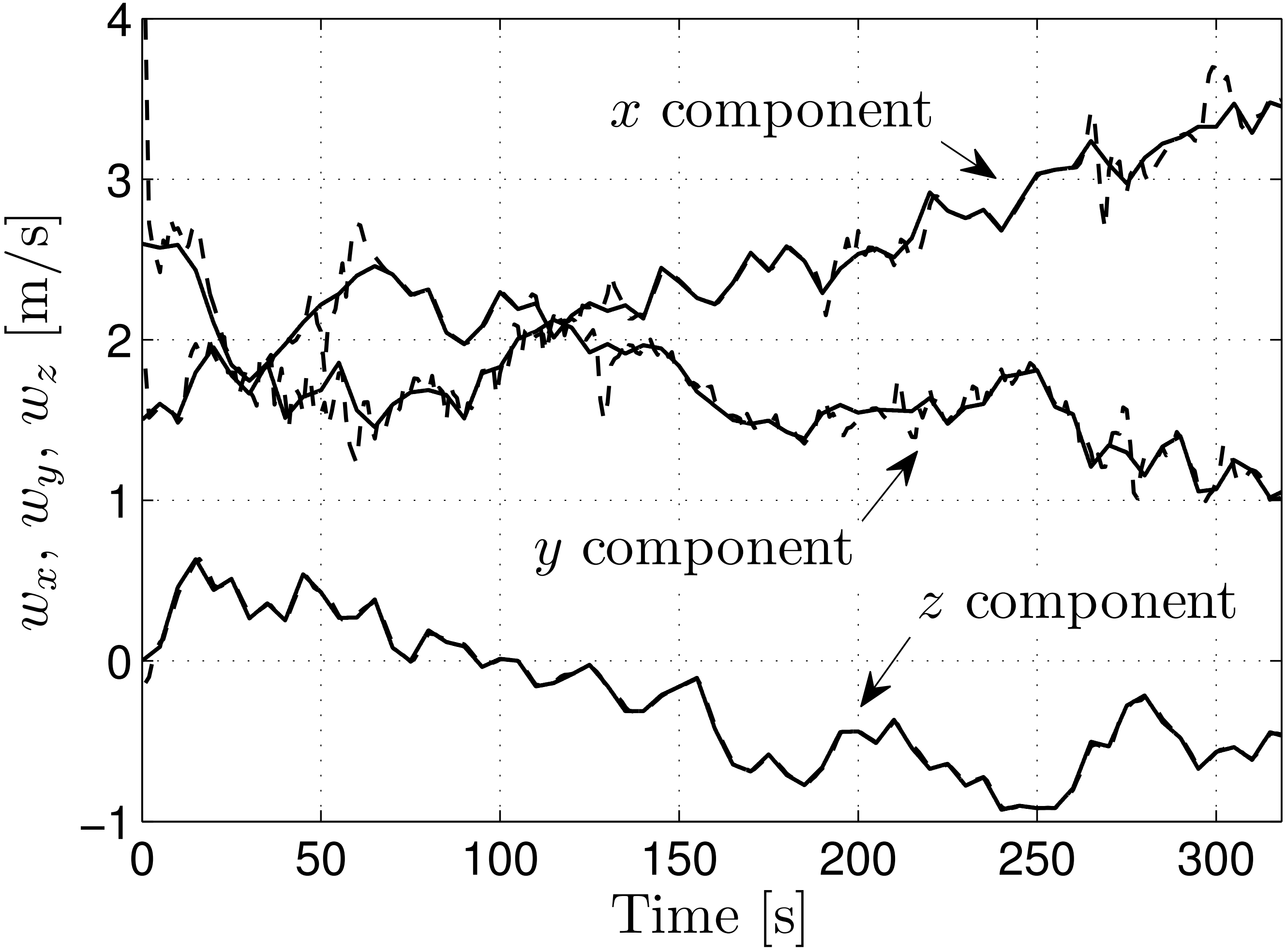
Simulations III

Time synchronization for a 3-D surveillance mission (with wind)



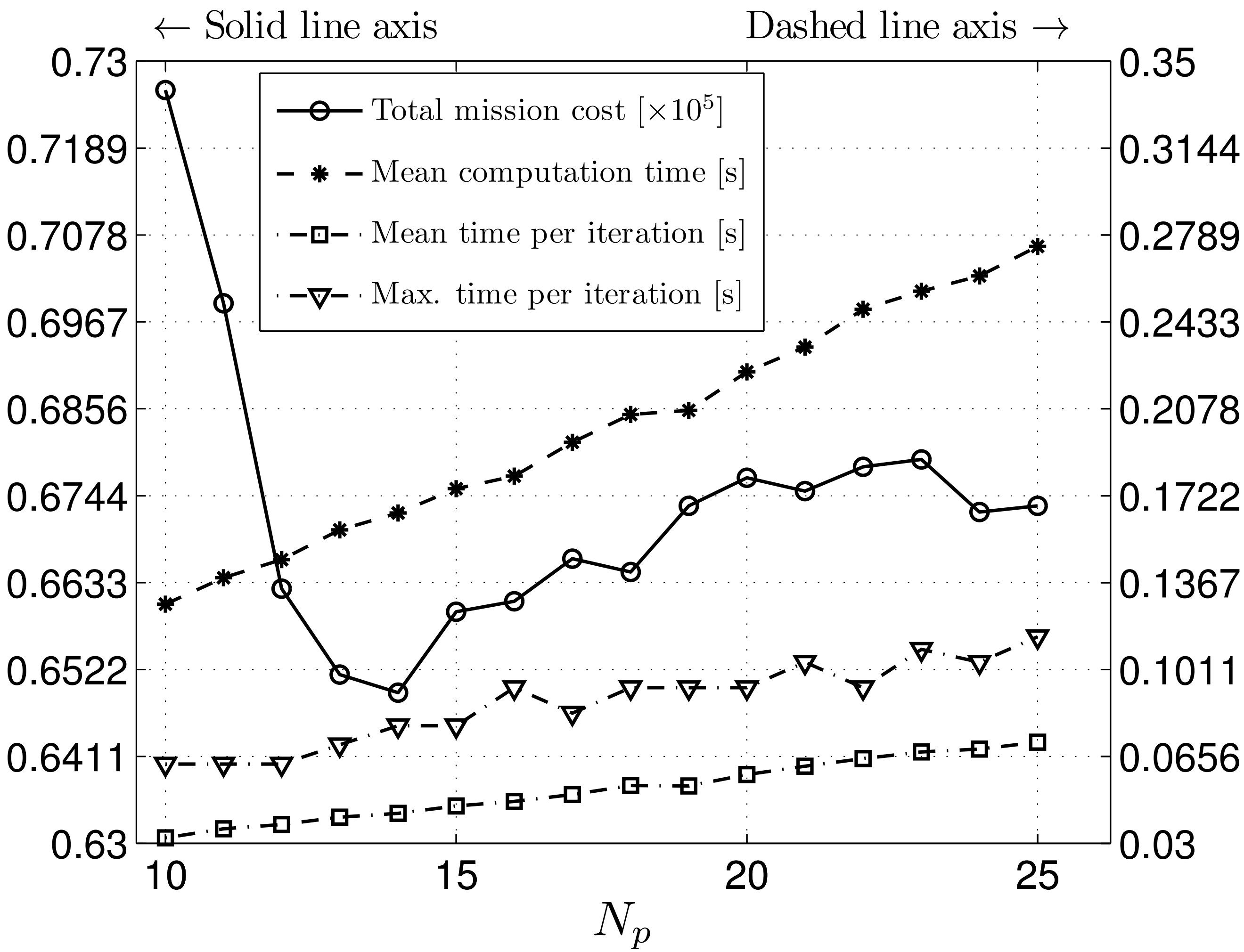
Simulations IV

Wind estimation



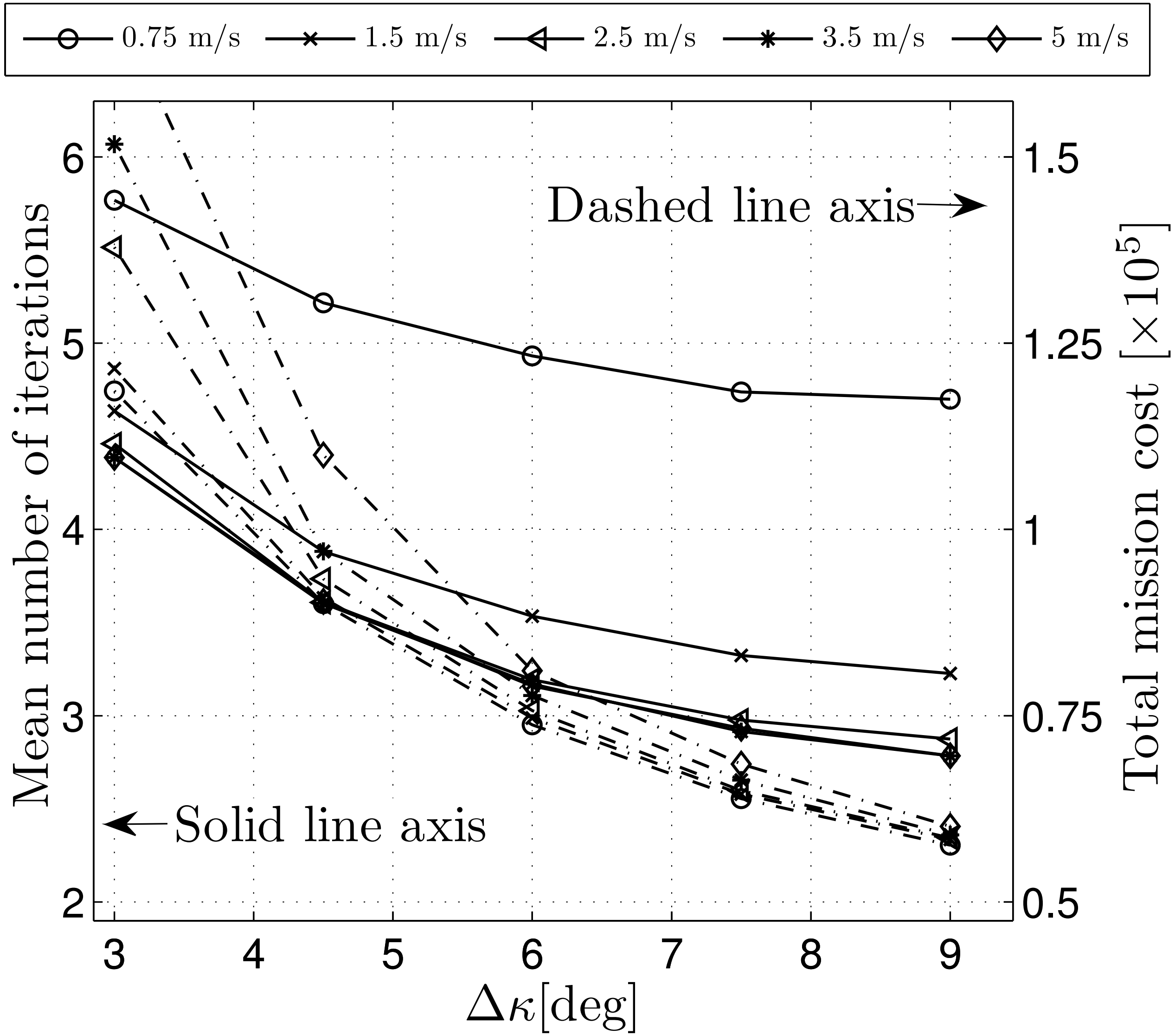
Parametric study I

Influence of N_p



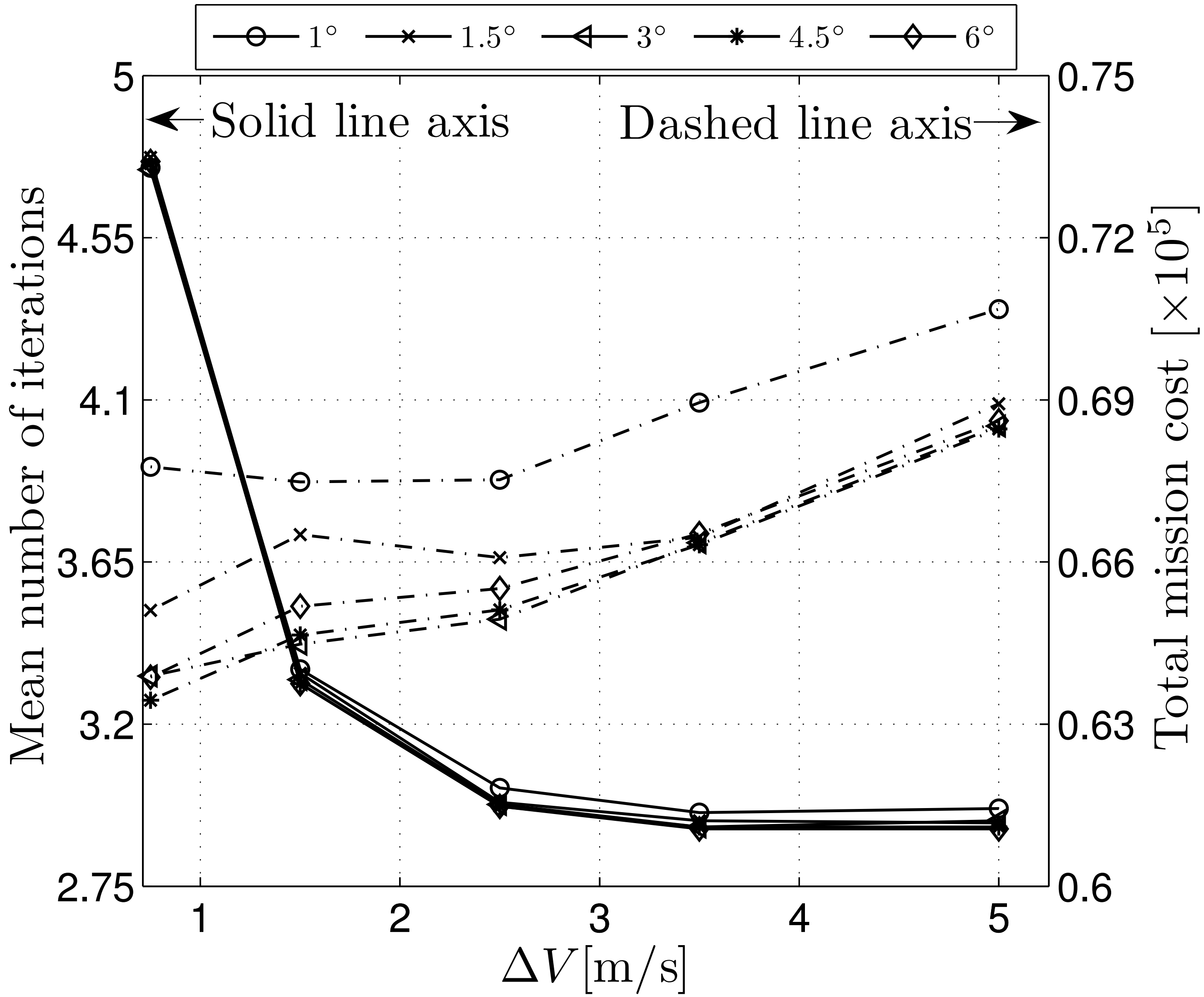
Parametric study II

Influence of $\Delta\kappa$



Parametric study III

Influence of ΔV



Outline

Conclusions

We have presented a flight control system based on a hierarchical architecture:

- Top level: Iterative model predictive guidance
- Low level: Flight controller.

✓ Main features

- Robust "hotstart" guidance algorithm. Feasibility assessment
- Disturbance (wind) estimator.
- Good performances in an accurate simulation model

✓ Future work

- Improve the prediction law.
- Optimization of the guidance algorithm, towards a realtime implementation.
- Extend guidance algorithm for formation flying.
- Develop a flight test campaign.

Conclusions

We have presented a flight control system based on a hierarchical architecture:

- Top level: Iterative model predictive guidance
- Low level: Flight controller.

✓ Main features

- Robust "hotstart" guidance algorithm. Feasibility assessment
- Disturbance (wind) estimator.
- Good performances in an accurate simulation model

✓ Future work

- Improve the prediction law.
- Optimization of the guidance algorithm, towards a realtime implementation.
- Extend guidance algorithm for formation flying.
- Develop a flight test campaign.

Engineering, Operations & Technology
Boeing Research & Technology

Increasing Predictability and Performance in UAS Flight Contingencies using AIDL and MPC

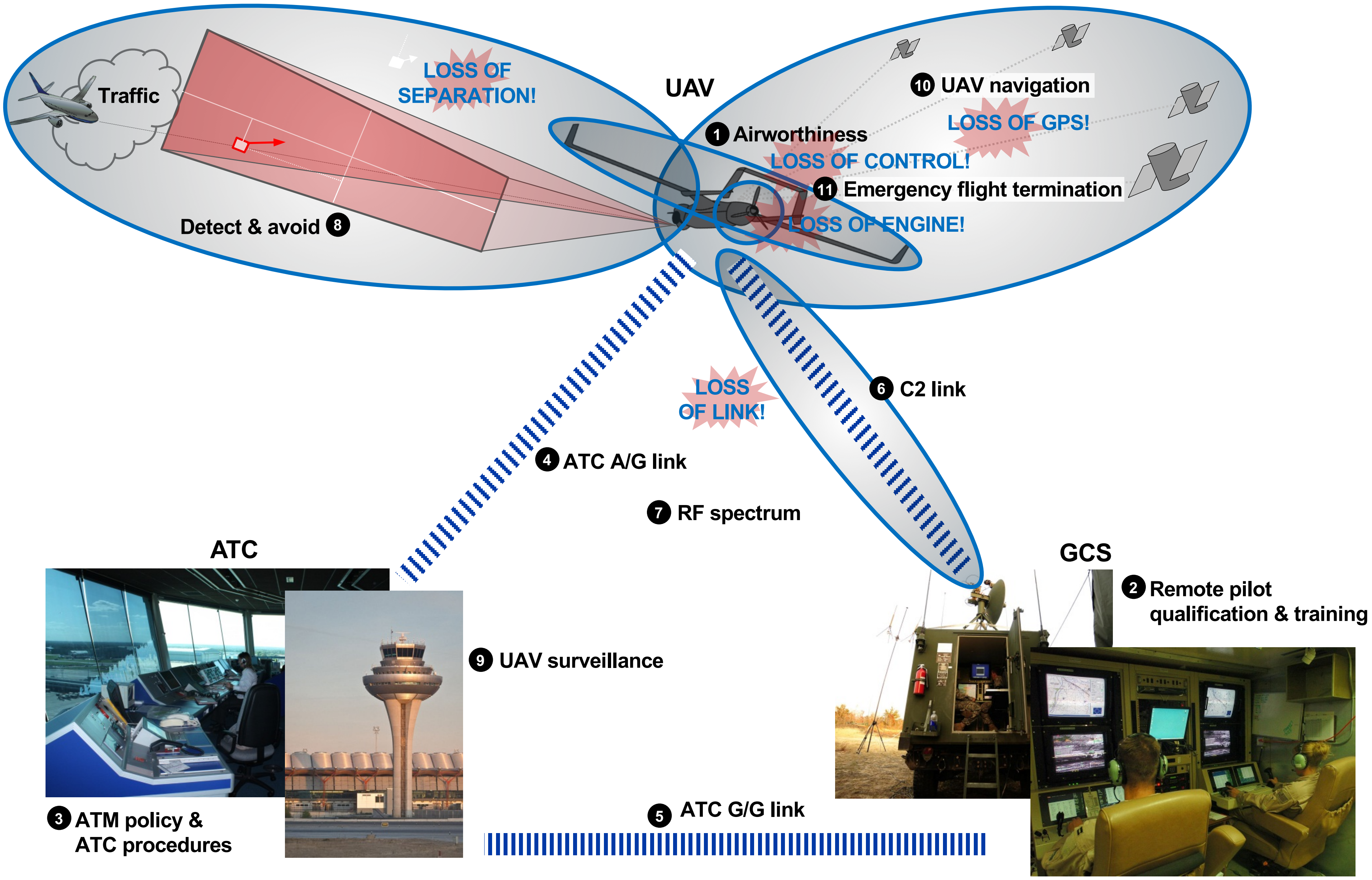
F. Gavilan, R. Vazquez, A. Lobato, M. de la Rosa, A. Gallego, E.F. Camacho (*Univ. of Seville*)

M. Hardt, F. Navarro *(Boeing Research & Technology – Europe)*

Outline

- UAV Contingencies
- Why AIDL?
- Why MPC?
- Control Architecture & Design
- Simulation & HW-in-loop Testing

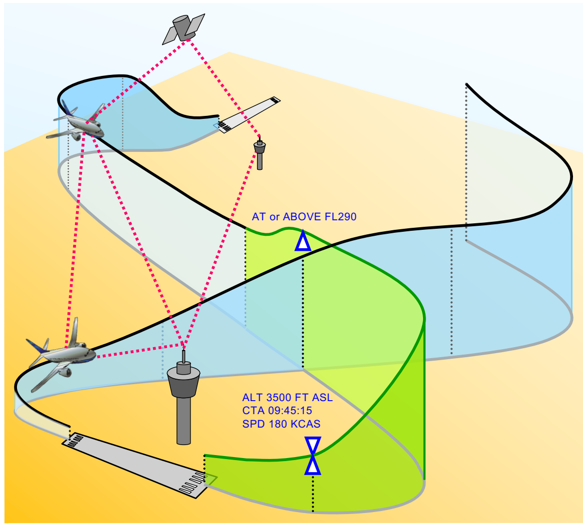
Key UAS/RPAS Integration Requirements



Key UAV Contingencies

- **Loss of Separation (LoS):** Requires UAV to (*autonomously)* generate and execute collision avoidance maneuvers and resume back to the original flight plan. Key prerequisite to enable safe operations of UAS in crowded non-segregated environments. Stringent time, spatial and dynamic constraints.
- **Loss of Link (LoL):** Requires UAV to generate and execute lost-link trajectories to reestablish command contact. Critical that the UAV's behavior and position be accurately estimated until contact is reestablished. High awareness of flight specifications and constraints desirable.
- **Loss of Engine (LoE):** Requires UAV to generate and execute emergency landing maneuver. Stringent time, spatial and dynamic constraints.
- **Loss of Control (LoC):** Requires UAV to robustly return to flight envelope, and/or adapt to actuator and sensor failures, estimate and compensate for severe wind conditions.

Trajectory-Based Operations



Model-Based Control

Maximize Predictability

AIDL - Aircraft Intent Description Language

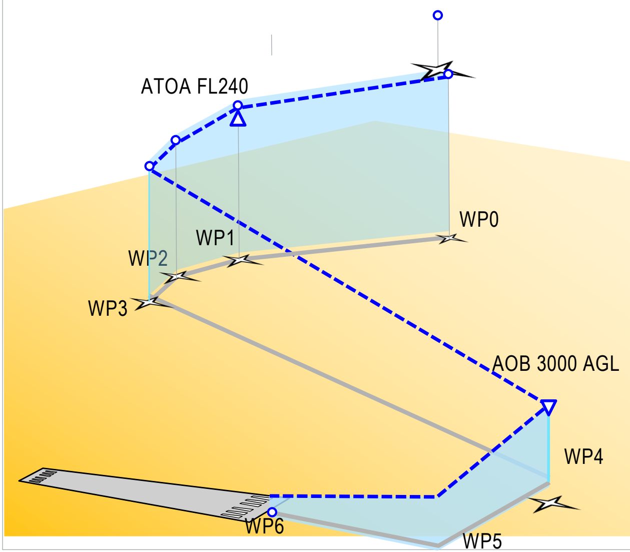
A method to formally capture the necessary and sufficient information that determines the trajectory of an aerial vehicle (AV), i.e. the *aircraft intent*

AIDL is a domain-specific formal language (DSL) composed of

- An ***alphabet*** (set of *"instructions"* or atomic ways of describing aircraft behavior)
- A ***lexicon*** (set of rules that govern the legal/meaningful combination of elements from the alphabet)
- A ***sequence control mechanism*** (set of *"triggers"* that switch behavioral changes upon reaching conditions)

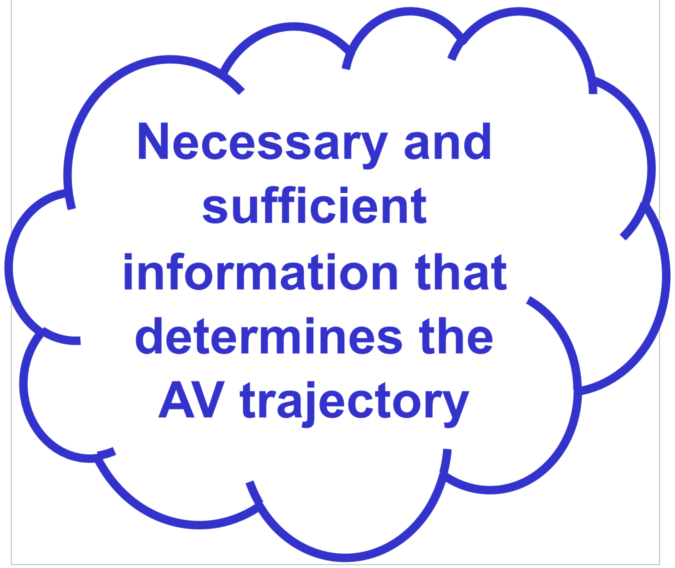
Requirements

What to fly and why



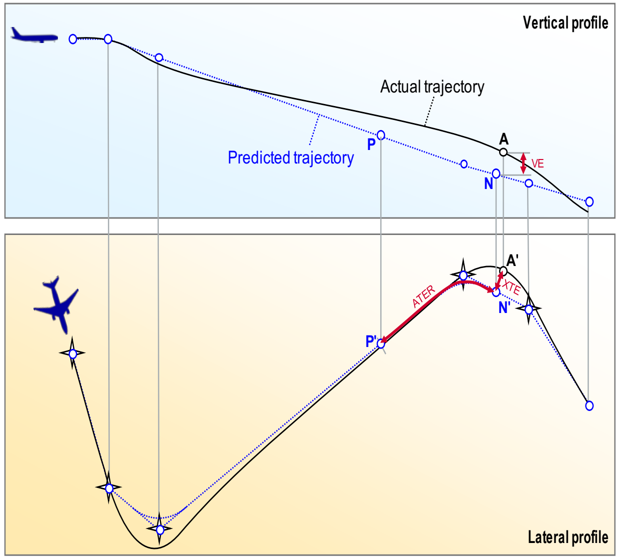
Recipe

How to operate the AV

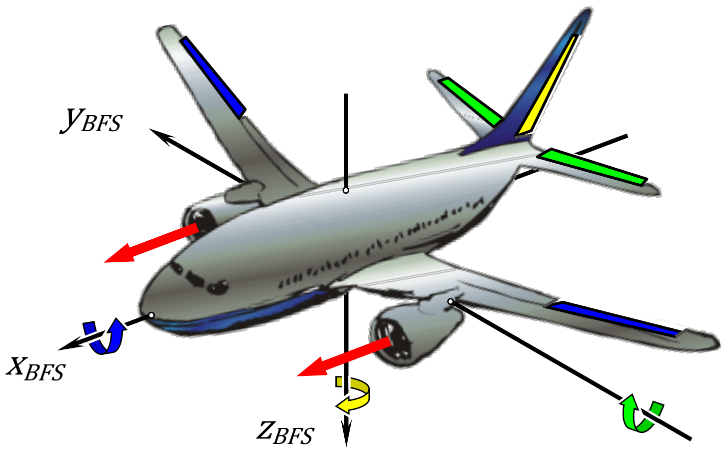


Results

What comes out



AIDL Aircraft Motion Model



Motion DOFs:

1st DOF – Coordinated lateral control (ailerons + rudder)

2nd DOF – Longitudinal control (elevators)

3rd DOF – Thrust control (throttle)

Configuration DOFs:

1st DOF – High lift devices

2nd DOF – Speed brakes

3rd DOF – Landing gear

4th DOF – Altitude reference (baroaltimeter setting)

Momentum Equations
Mass variation
Navigation Equations
} Motion model

\+

Aircraft performance model

\+

Earth model
(Geodetic, Geopotential, Atmosphere)

\+

$\delta_{Lateral} = q_1(t)$
$\delta_{Longitudinal} = q_2(t)$
$\delta_{Thrust} = q_3(t)$
} Control/guidance model

$\delta_{High\text{-}Lift\ Devices} = c_1(t)$
$\delta_{Speed\ Brakes} = c_2(t)$
$\delta_{Landing\ Gear} = c_3(t)$
$\delta_{Altitude\ Reference} = c_4(t)$
} Configuration model

\+

Initial conditions

AIDL Aircraft Motion Model

Universal equations governing any possible aircraft motion ⇨

- Momentum Equations
- Mass variation
- Navigation Equations

} Motion model

+

Aircraft-specific characteristics ⇨ Aircraft performance model

+

Environment-specific characteristics ⇨ Earth model (Geodetic, Geopotential, Atmosphere)

+

Characteristics specific to the particular aircraft motion at hands ⇨

- $\delta_{\text{Lateral}} = q_1(t)$
- $\delta_{\text{Longitudinal}} = q_2(t)$
- $\delta_{\text{Thrust}} = q_3(t)$

} Control/guidance model

- $\delta_{\text{High-Lift Devices}} = c_1(t)$
- $\delta_{\text{Speed Brakes}} = c_2(t)$
- $\delta_{\text{Landing Gear}} = c_3(t)$
- $\delta_{\text{Altitude Reference}} = c_4(t)$

} Configuration model

+

Initial conditions

AIDL Aircraft Motion Model

Universal equations governing any possible aircraft motion ⇨

- Momentum Equations
- Mass variation
- Navigation Equations

} Motion model

\+

aircraft-specific characteristics ⇨ Aircraft performance model

\+

Environment-specific characteristics ⇨ Earth model (Geodetic, Geopotential, Atmosphere)

\+

Trajectory Recipe ⇨

Aircraft Intent

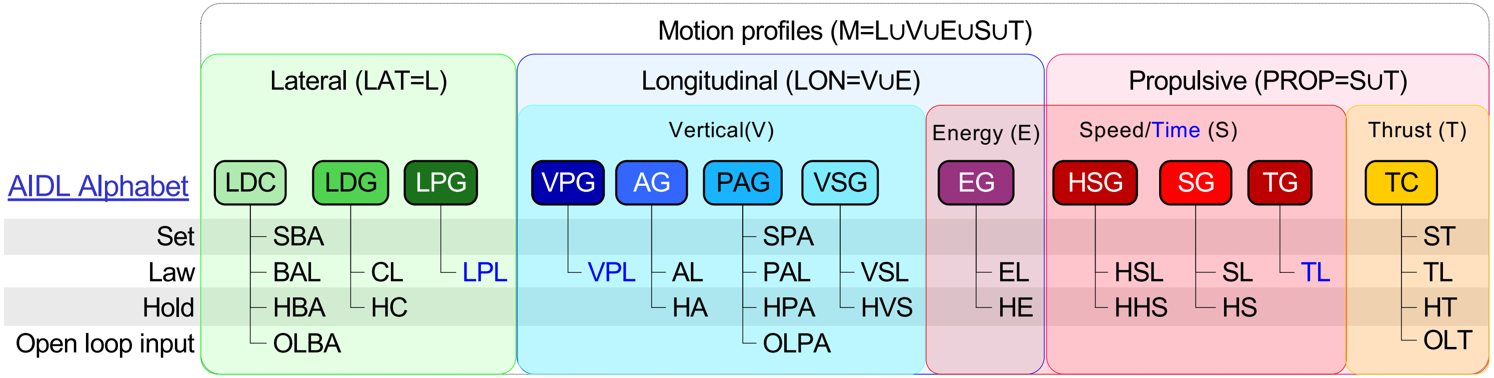
Description of how the configuration and motion degrees of freedom shall be governed along the time so the resulting aircraft motion gets univocally determined

i.e. anticipated unambiguous description on how the aircraft is to be operated over time

\+

Initial conditions

AIDL Alphabet and Lexicon



#	Keyword	Instruction	Effect
1	SBA	Set Bank Angle	$g(\mu_{TAS})=f(\mathbf{X},\mathbf{E},t)$
2	BAL	Bank Angle Law	
3	HBA	Hold Bank Angle	$g(\mu_{TAS})=0$
4	OLBA	Open Loop Bank Angle	$g(\mu_{TAS})=f(t)$
5	CL	Course Law	$g(\chi_{TAS},\mathbf{E})=f(\mathbf{X},\mathbf{E},t)$
6	HC	Hold Course	$g(\chi_{TAS},\mathbf{E})=0$
7	LPL	Lateral Path Law	$f(\lambda,\varphi,t)=0$
8	VPL	Vertical Path Law	$h=f(\lambda,\varphi,t)$
9	AL	Altitude Law	$g(h,\mathbf{E})=f(\mathbf{X},\mathbf{E},t)$
10	HA	Hold Altitude	$g(h,\mathbf{E})=0$
11	SPA	Set Path Angle	$g(\gamma_{TAS},\mathbf{E})=f(\mathbf{X},\mathbf{E},t)$
12	PAL	Path Angle Law	
13	HPA	Hold Path Angle	$g(\gamma_{TAS},\mathbf{E})=0$
14	OLPA	Open Loop Path Angle	$g(\gamma_{TAS},\mathbf{E})=f(t)$

#	Keyword	Instruction	Effect
15	VSL	Vertical Speed Law	$g(v_{TAS}\sin\gamma_{TAS},\mathbf{E})=f(\mathbf{X},\mathbf{E},t)$
16	HVS	Hold Vertical Speed	$g(v_{TAS}\sin\gamma_{TAS},\mathbf{E})=0$
17	EL	Energy Law	$g(dv_{TAS}/dh,\mathbf{E})=f(\mathbf{X},\mathbf{E},t)$
18	HE	Hold Energy	$g(dv_{TAS}/dh,\mathbf{E})=0$
19	HSL	Horizontal Speed Law	$g(v_{TAS}\cos\gamma_{TAS},\mathbf{E})=f(\mathbf{X},\mathbf{E},t)$
20	HHS	Hold Horizontal Speed	$g(v_{TAS}\cos\gamma_{TAS},\mathbf{E})=0$
21	SL	Speed Law	$g(v_{TAS},\mathbf{E})=f(\mathbf{X},\mathbf{E},t)$
22	HS	Hold Speed	$g(v_{TAS},\mathbf{E})=0$
23	TL	Time Law	$t=g(v_{TAS}\cos\gamma_{TAS},\ \mathbf{E})$
24	STC	Set Throttle Control	$g(\delta_T)=f(\mathbf{X},\mathbf{E},t)$
25	TCL	Throttle Control Law	
26	HTC	Hold Throttle Control	$g(\delta_T)=0$
27	OLTC	Open Loop Throttle Control	$g(\delta_T)=f(t)$

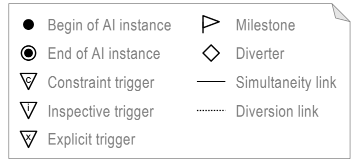
Allowed combinations of motion profiles

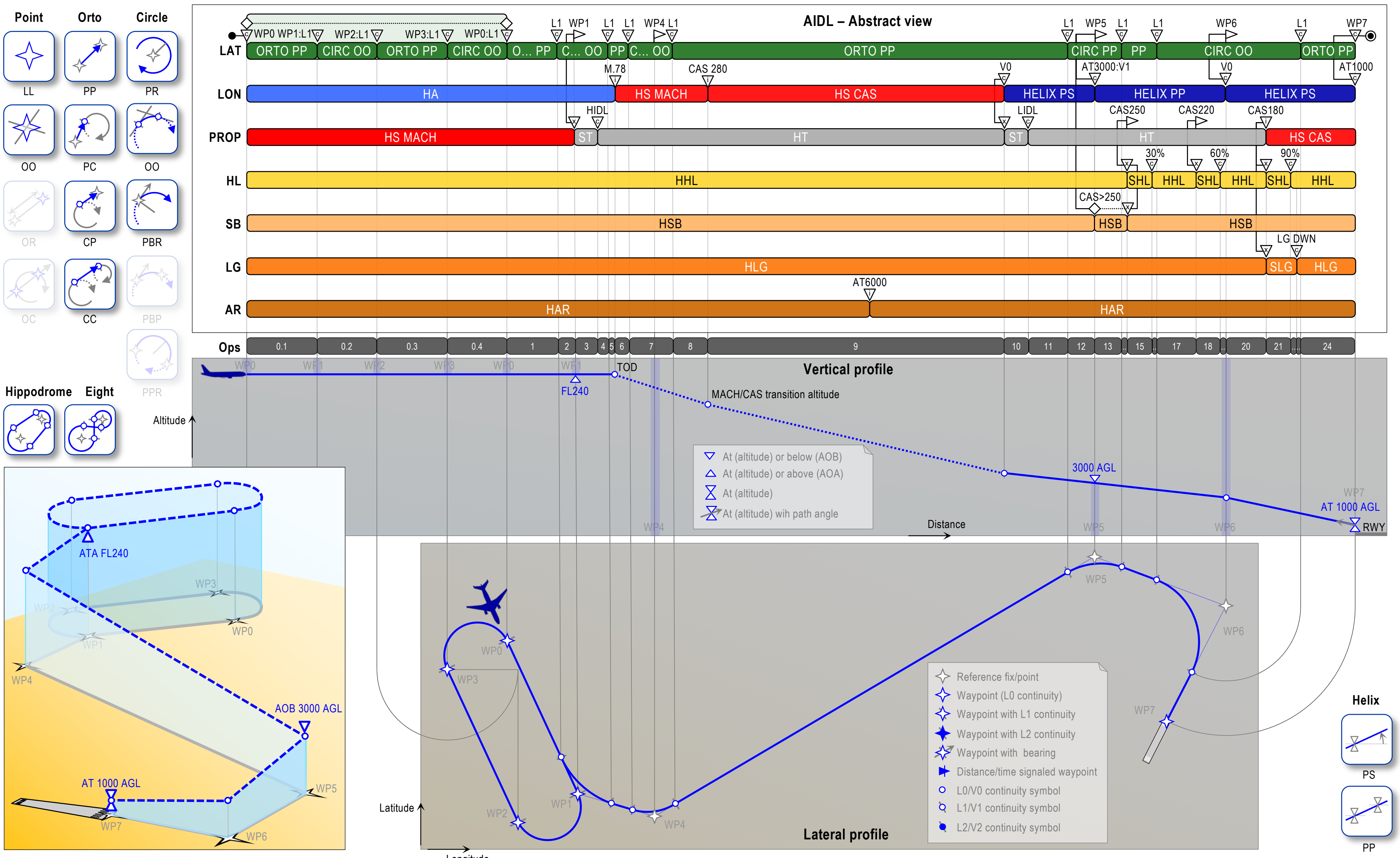
1st DOF	L	L	L	L	Lateral profile (LAT)
2nd DOF	V	V	E	S	Longitudinal profile (LON)
3rd DOF	S	T	T	T	Propulsive profile (PROP)

AIDL Lexicon

- 7 instructions, each from a different group
- Of the 7, 3 must belong to motion profiles and 4 to the configuration profile
- The 3 motion instructions must belong to different motion profiles (L, V, S, T)
- Of the 3 motion instructions, 1 must come from the lateral profile (L)

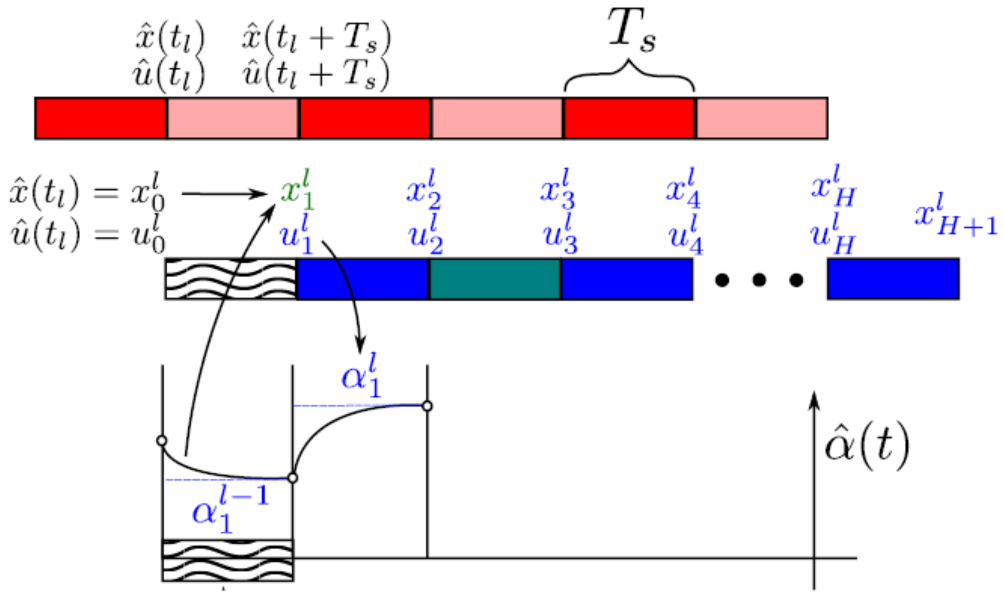
AIDL Sample Trajectory





Why MPC?

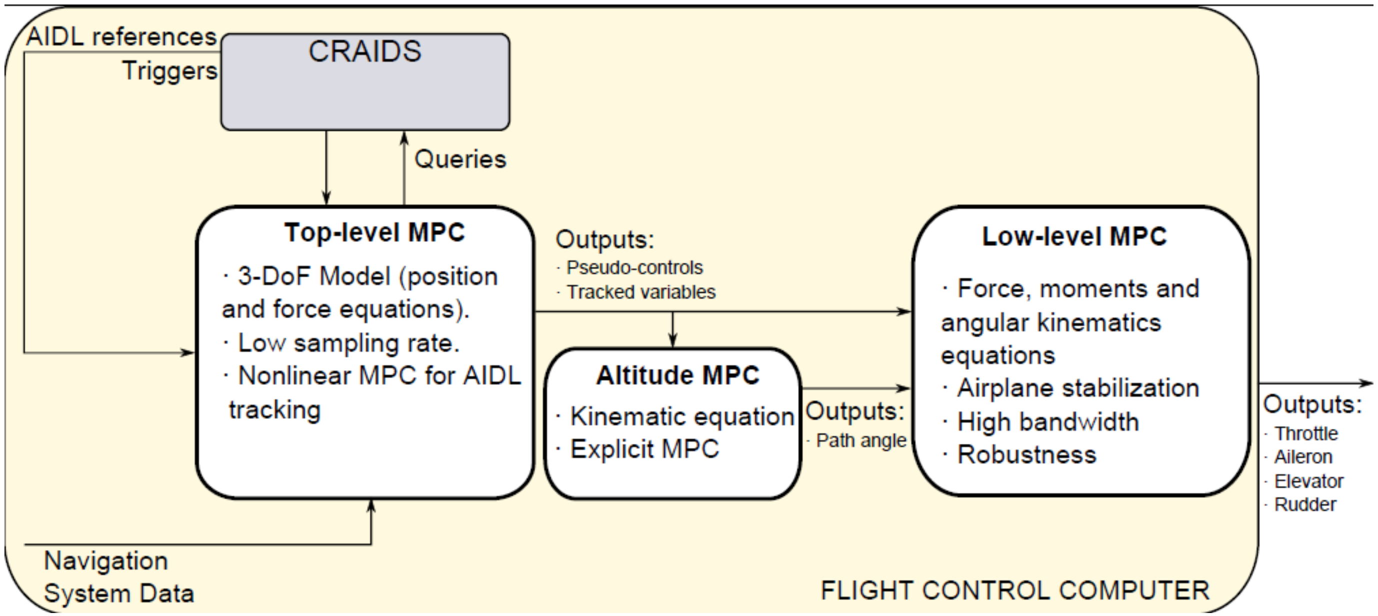
- **Model Predictive Control** is a family of control methods which make explicit use of a process model to obtain the control by repeated optimization of an objective function over a receding time horizon



- Fits naturally into AIDL framework by expressing cost function as minimization of error to motion constraints associated to AIDL instructions
- Handle hybrid nature by considering trigger activation within receding horizon
- Implement on low SWaP in real-time

AIDL-MPC Flight Control Architecture

- Top Level MPC:
  - Follows three AIDL threads (lateral, longitudinal, propulsive) by enforcing their respective constraints
  - Checks for trigger activation during horizon and in each thread
  - Considers estimated wind
- Low Level MPC:
  - Receives virtual setpoints from high-level, which may vary depending upon active AIDL instructions, and defines four actuation inputs
  - Maintains symmetric flight
  - Operates at 50Hz



AIDL-MPC Flight Control Architecture

- Top Level MPC:
  - State: $x_k = [\varphi, \lambda, h, V_a, \chi_{tas}, \gamma_{tas}]_k$
  - Control: $u_k = [\alpha, \mu_{tas}, \delta_T]_k$ (virtual)
  - Problem formulation:

$$\min_u \; J(x, u, x_{ref}, u_{ref})$$

$$s.a. \quad x_k = f(x_{k-1}, x_{k-1})$$

$$u_{min} \leq u \leq u_{max}$$

$$Ax = b$$

  - Iterative, discrete state propagation for entire horizon is defined from nonlinear model

$$\Delta x = M \cdot \Delta u$$

  - Repeated linearization in each control cycle together with Sequential Quadratic Programming (SQP) strategy are used
  - Prediction horizon: 7.5s, Control horizon: 5s, Frequency: 2Hz.
  - Repeat calculation of longitudinal degree of freedom in separate MPC to capture faster altitude dynamics

AIDL-MPC Flight Control Architecture

- Top Level MPC objective function:

$$J = (x - x_{ref})^T R (x - x_{ref}) + (u - u_{ref})^T Q (u - u_{ref})$$

- Substitute $\Delta x = M \cdot \Delta u$ and use $u_{ref,k} = u_{k-1}$, $J = J_x + J_u$

$$J_x = \Delta u^T (M^T R M) \Delta u + [2\bar{x}^T R M - 2x_{ref}^T R M] \Delta u$$

$$J_u = \Delta u^T (Q - 2Q_B + Q_C) \Delta u + (2(\bar{u} - \bar{u}_D)^T (Q - Q_B)) \Delta u$$

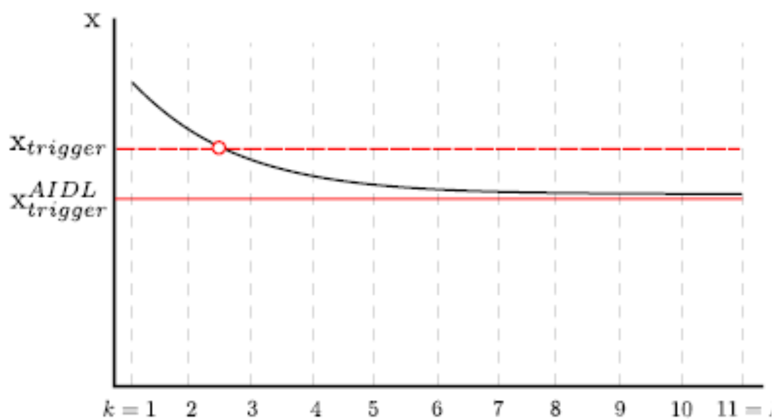
$$R = \begin{bmatrix} R_2 & 0 & \cdots & 0 \\ 0 & R_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{H+1} \end{bmatrix}, \quad Q = \begin{bmatrix} Q_1 & 0 & \cdots & 0 \\ 0 & Q_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_H \end{bmatrix}$$

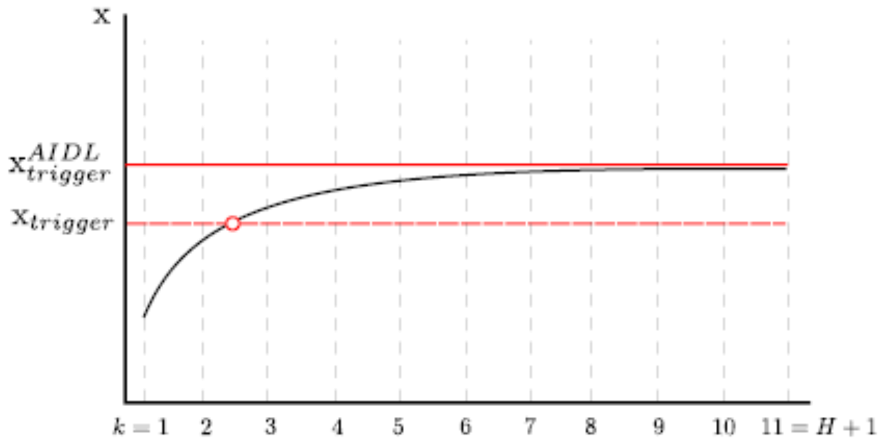
- Coefficients take on 0 or 1 for time step i depending upon whether corresponding AIDL instruction is active or control not specifically set

$$R_i = \begin{bmatrix} \frac{R_{\varphi i}}{\delta_\varphi^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{R_{\lambda i}}{\delta_\lambda^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{R_{hi}}{\delta_h^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{R_{Vi}}{\delta_v^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{R_{\chi i}}{\delta_\chi^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{R_{\gamma i}}{\delta_\gamma^2} \end{bmatrix} \quad Q_i = \begin{bmatrix} \frac{R_{\alpha i}}{\delta_\alpha^2} & 0 & 0 \\ 0 & \frac{R_{\mu i}}{\delta_\mu^2} & 0 \\ 0 & 0 & \frac{R_{\delta_T i}}{\delta_{\delta_T}^2} \end{bmatrix}$$

AIDL-MPC Trigger Handling

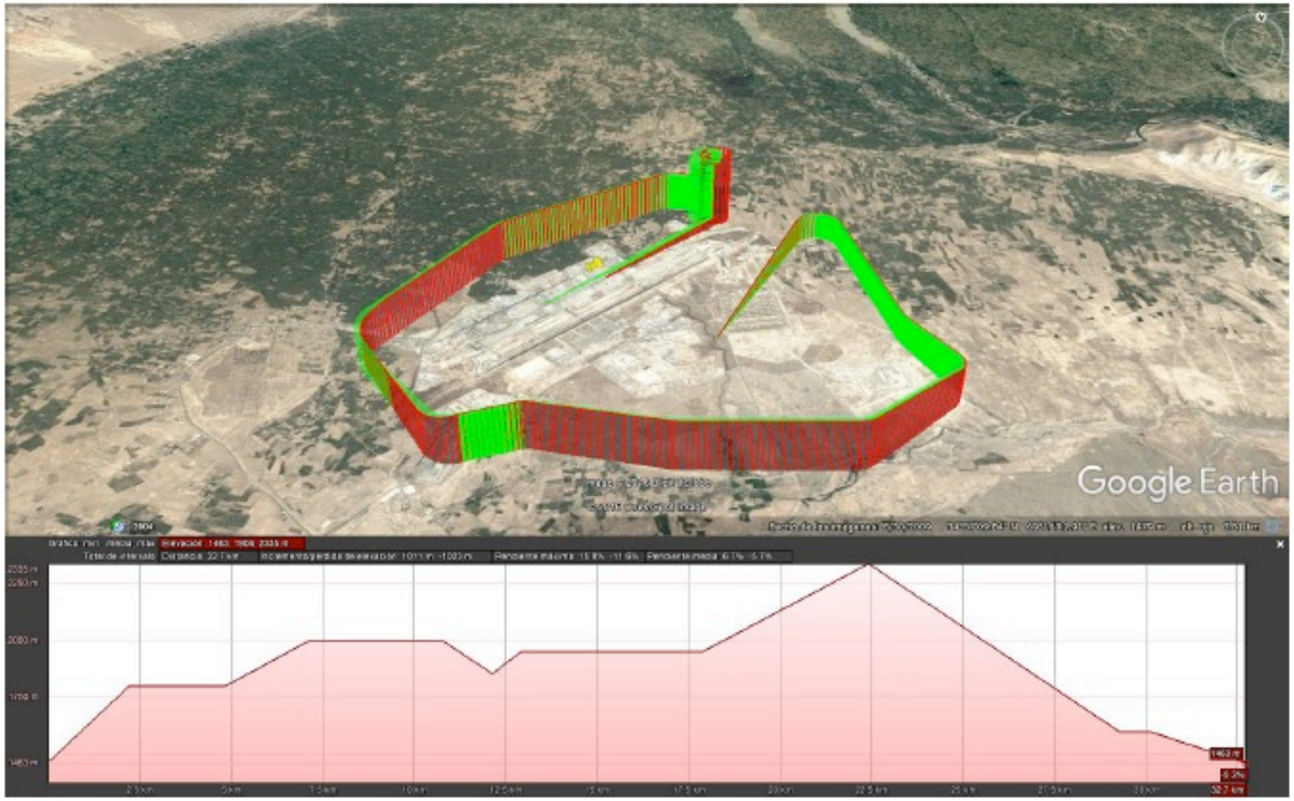
- Triggers indicate transitions from one AIDL instruction to another
- They are typically time or state dependent
- The assumption is made that at most one trigger may appear in the prediction horizon in each motion thread
- Trigger identification and decoupling from optimization:
  - At each iteration, zero detection algorithm is run to check if and when trigger condition is satisfied
  - Once time is identified, objective function is generated appropriately along prediction horizon
  - After optimization, predicted state and subsequently trigger are updated
- Numerical issues obligate the modification of the trigger conditions into detection of interval crossing





AIDL-MPC Experimental Testing

- Simulation scenario of flying around fictitious base with takeoff, contingency, and landing maneuvers
- Wide range of wind relative and absolute instructions and trigger conditions have been tested.
- Control performance achieves near exact theoretical trajectory prediction, even in the presence of wind turbulence.



Longitudinal 1 Thread							
Inst	SpecVar	InstCode	TargVal	TrigCode	TrigVar	TrigVal	TrigID
HS	v_{TAS}	1021	25 [mps]	0	ID	tH	—
HTC	δ_T	1231	0.3 [-]	0	ID	tdescend01	—
HTC	δ_T	1231	0.5 [-]	12	v_{TAS}	25 [mps]	—
HS	v_{TAS}	1021	25 [mps]	0	ID	end	—

AIDL-MPC HW-in-loop Testing

- Flight hardware from Skylife Engineering with Gumstix DuoVero Crystal (Clock speed 1GHz, Dual Core, 1 Gb RAM, Linux OS
- One core calculates:
  - AHRS/EKF Navigation based upon sensor inputs from 9-DOF IMU, wind vanes, air data sensors
  - AIDL-based Guidance system which feeds control with formulation of current AIDL instructions, next trigger conditions and following set of AIDL instructions
  - Low-level MPC
- Second core calculates:
  - Top-level MPC
- Connected by usb to PC upon which runs:
  - Full flight and actuator dynamics
  - Environmental model including wind turbulence
  - Sensor models

Case	N_p	N_c	Gumstix	Notebook
1	50	10	0.03766	0.06070
2	10	7	0.01563	0.02470
3	10	5	0.00520	0.02280
4	8	7	0.01564	0.03090
5	8	4	0.00339	0.02160

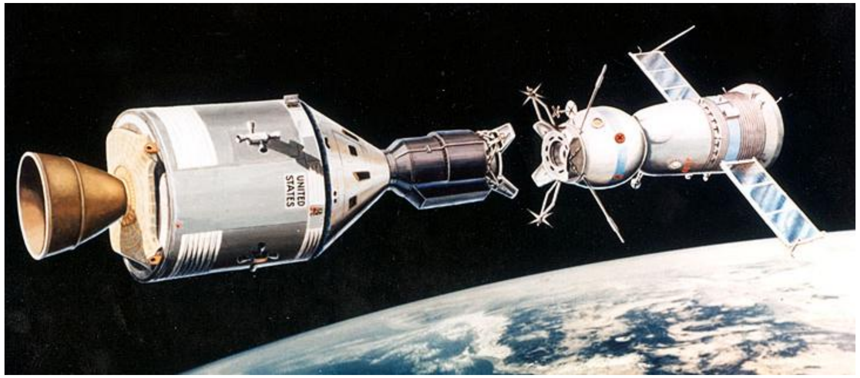
MPC dimensioning

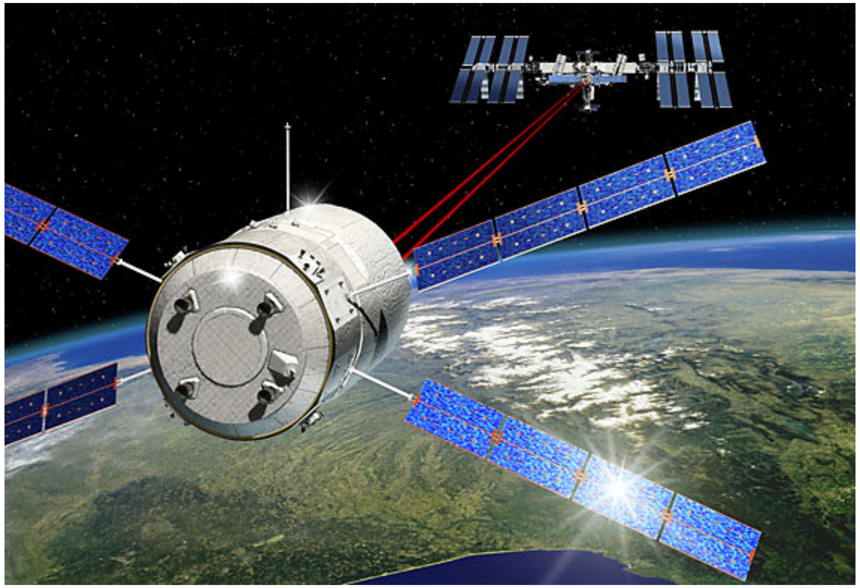
Outline

1. ~~Introduction to MPC (Slides by E.F. Camacho)~~
2. ~~Application to Spacecraft Rendezvous (including PWM)~~
3. ~~Rendezvous + Attitude Control~~
4. ~~Soft Landing on an Asteroid~~
5. ~~Guidance for UAVs~~

That's all folks!

GRAZIE MILLE





Thank you!

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

rvazquez1@us.es