NPI: Real-Time Adaptive Entry Guidance

Final Presentation – ESTEC – 08.04.2016

David Seelbinder ✉ david.seelbinder@dlr.de

Prof. Dr. Christof Büskens ✉ bueskens@math.uni-bremen.de

Dr. Stephan Theil ✉ stephan.theil@dlr.de

esa

University Bremen

DLR

Agenda

- **Motivation and introduction**
- **Trajectory computation**
  - Parametric sensitivity analysis of nonlinear programs (offline)
  - Fast solution approximation (online)
  - Repeated online trajectory computation
  - Region of convergence
- **Trajectory tracking**
  - Feedback linearization
  - Drag energy dynamics
- **Guidance system overview**
- **Monte Carlo campaign**
  - Perturbed environment
  - Guidance performance results
- **Processor-in-the-loop test**
- **Software tools**
- **Summary out outlook**

Guided Entry Phase

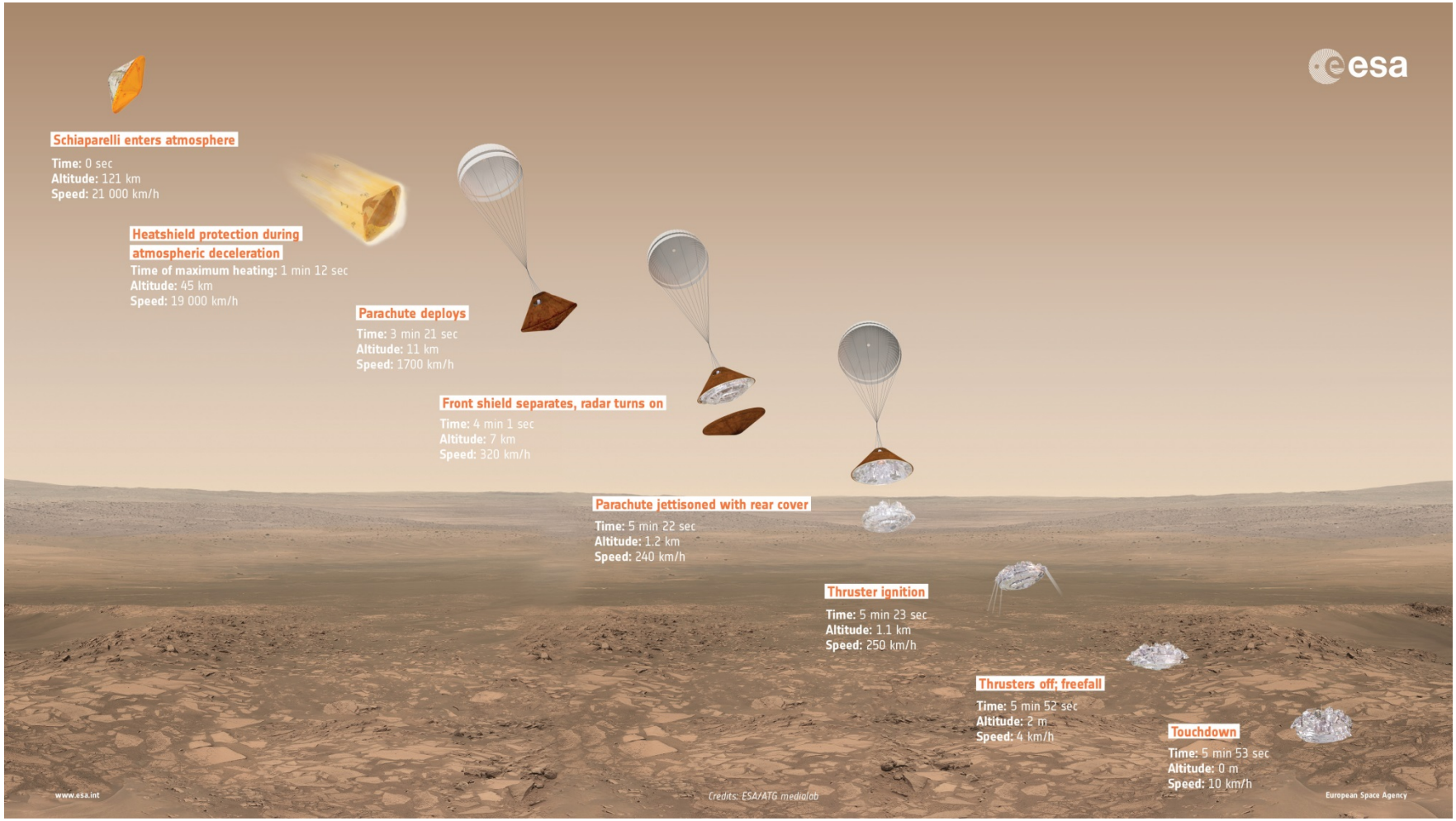


Image credit: ESA

Guided Entry Phase

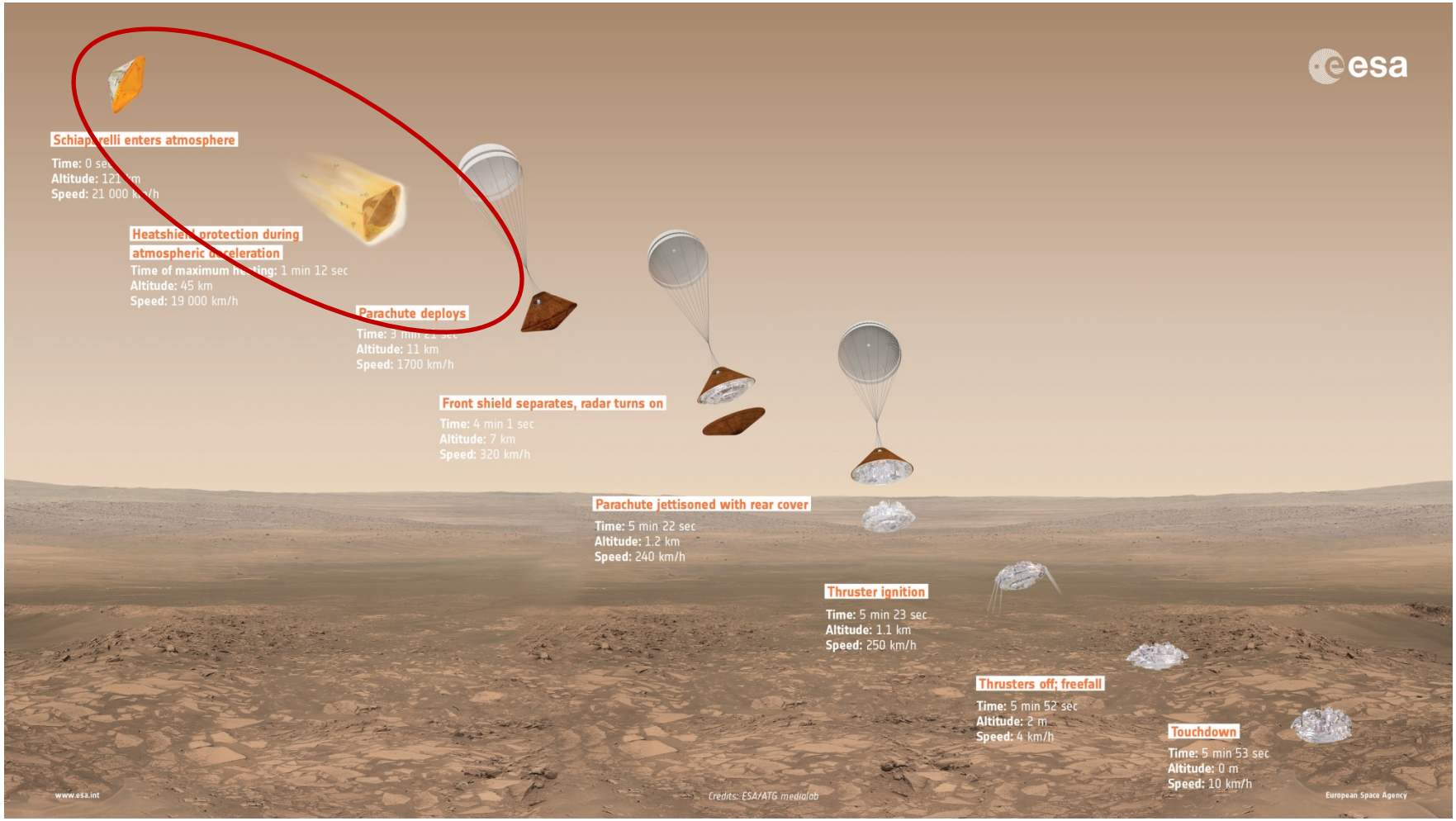


Image credit: ESA

Guidance Task

Find an admissible control sequence that steers the vehicle on a feasible trajectory form the current state to the desired terminal state.

Problem: Nonlinear dynamic system with state and control constraints

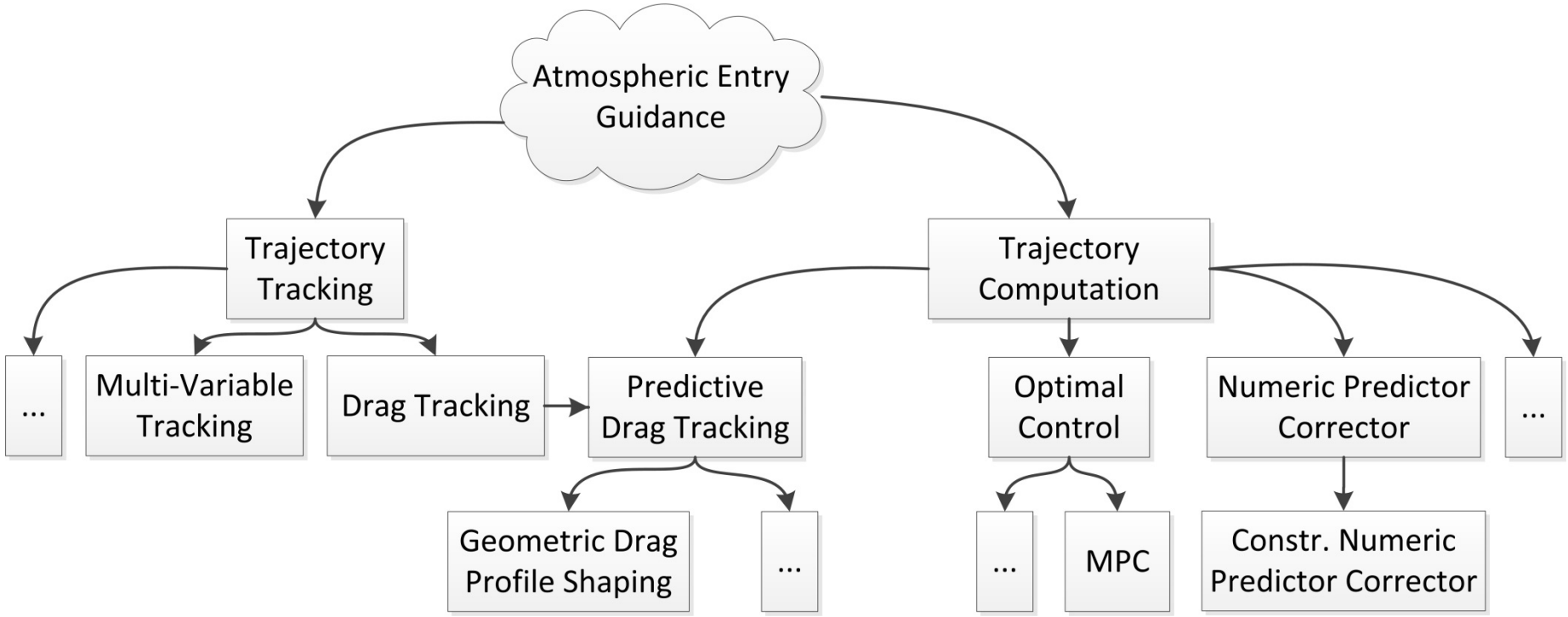
Req.: Planning of the control sequence and state trajectory over the entire future path

Req.: Mathematical model

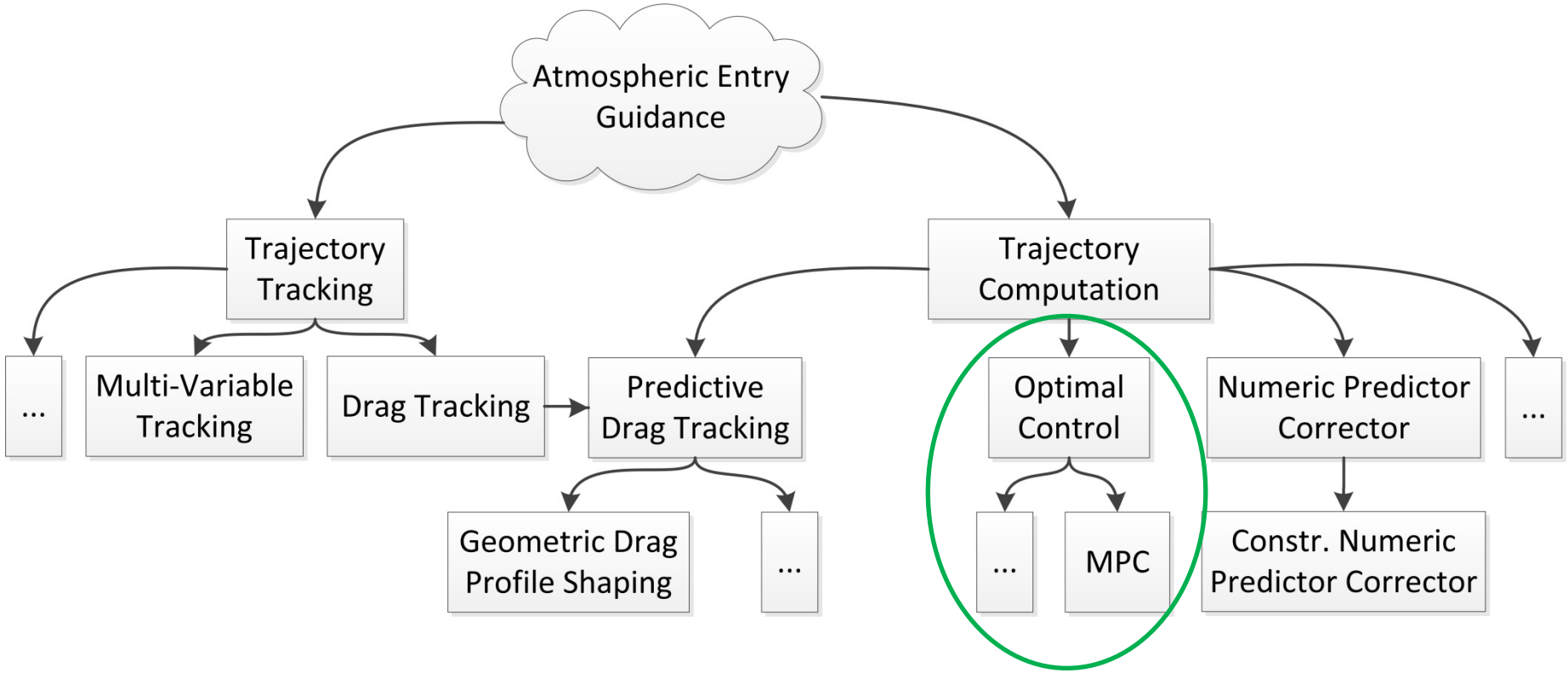
Challenges:

1. Constrained nonconvex optimization problem
2. High computational power demand
3. Model dependency

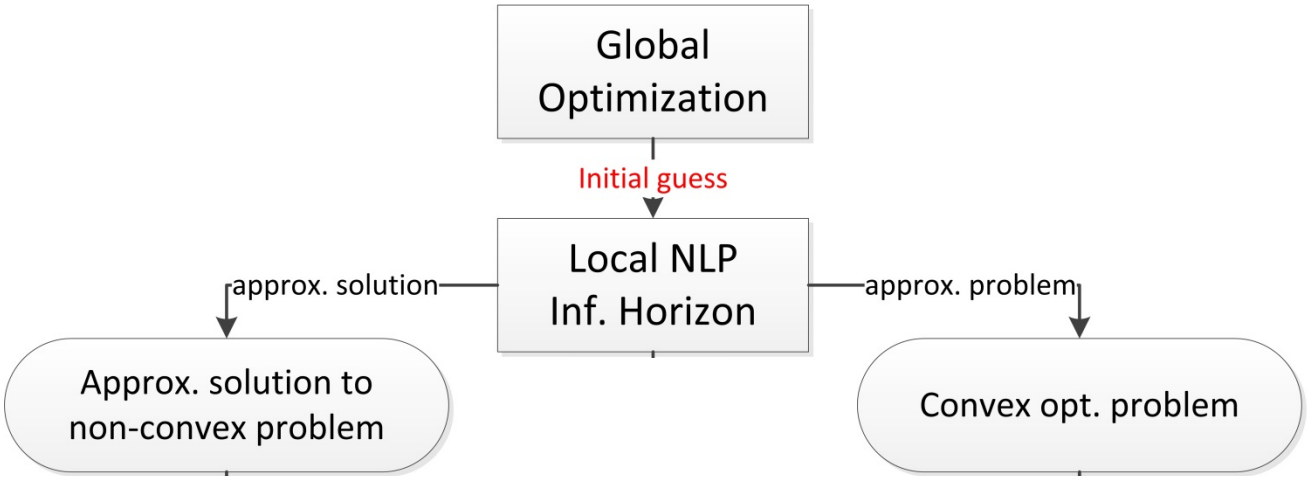
Approaches to Entry Guidance



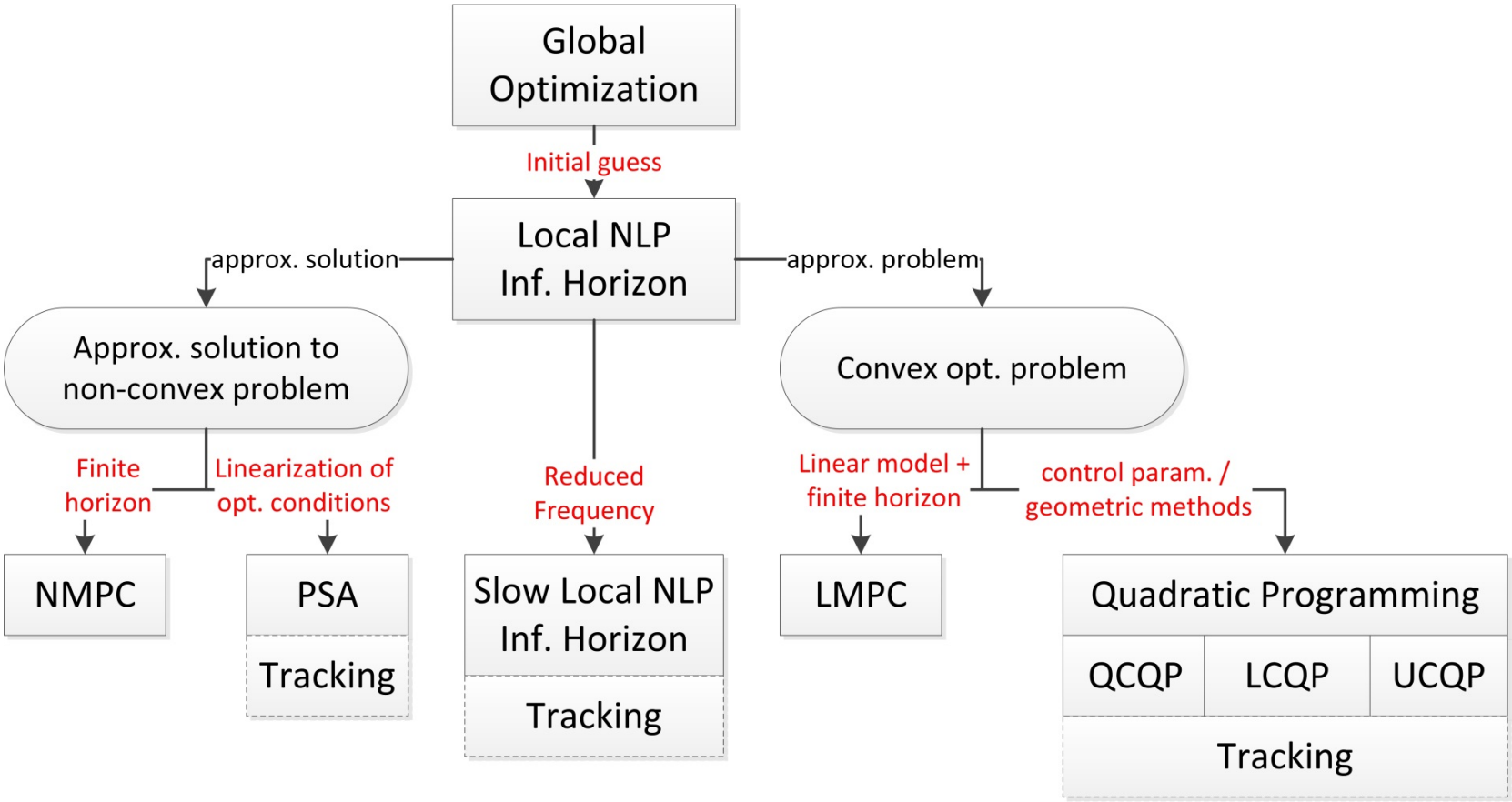
Approaches to Entry Guidance



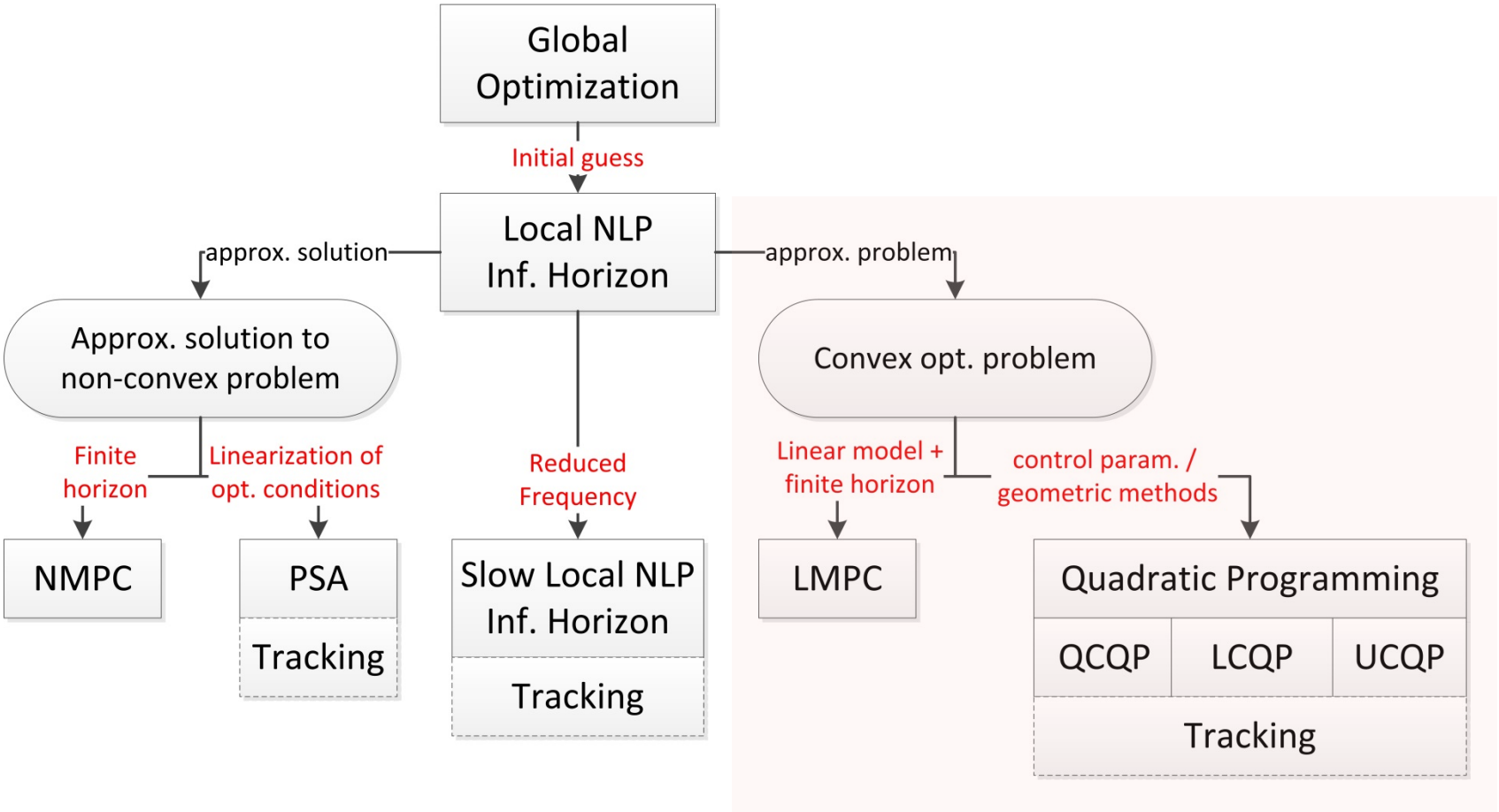
Optimization based Guidance



Optimization based Guidance



Optimization based Guidance



Optimization based Guidance

Property	NMPC Inf. Horizon	NMPC
Computational power required	--	-
OBS complexity	--	--
Respect path constraint	++	++
Region of attraction	++	+
Optimality	++	+
Method universality	++	++
Online optimization	yes	yes

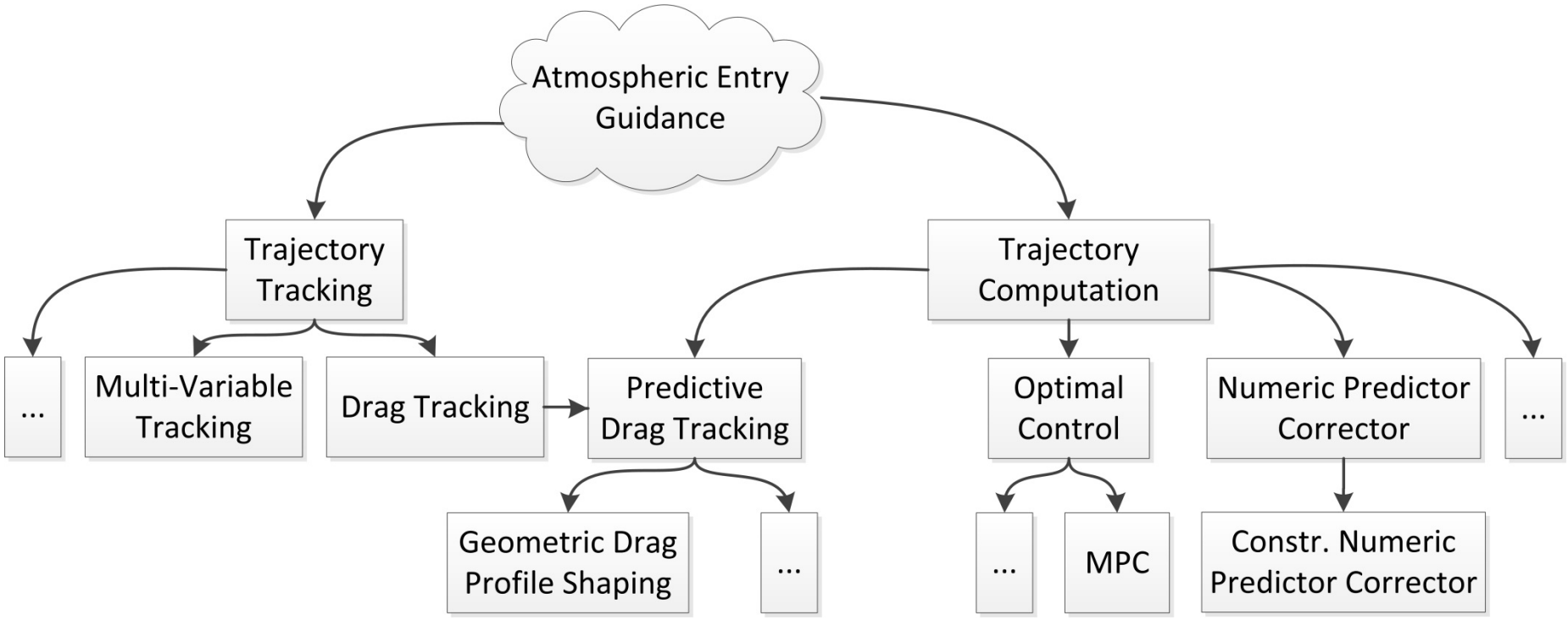
Optimization based Guidance

Property	NMPC Inf. Horizon	NMPC	Slow NLP / Tracking	PSA / Tracking
Computational power required	--	-	o/+	+/+
OBS complexity	--	--	--	+
Respect path constraint	++	++	+/-	o/-
Region of attraction	++	+	+/?	?/?
Optimality	++	+	+/?	+/?
Method universality	++	++	+/o	o/o
Online optimization	yes	yes	yes	no

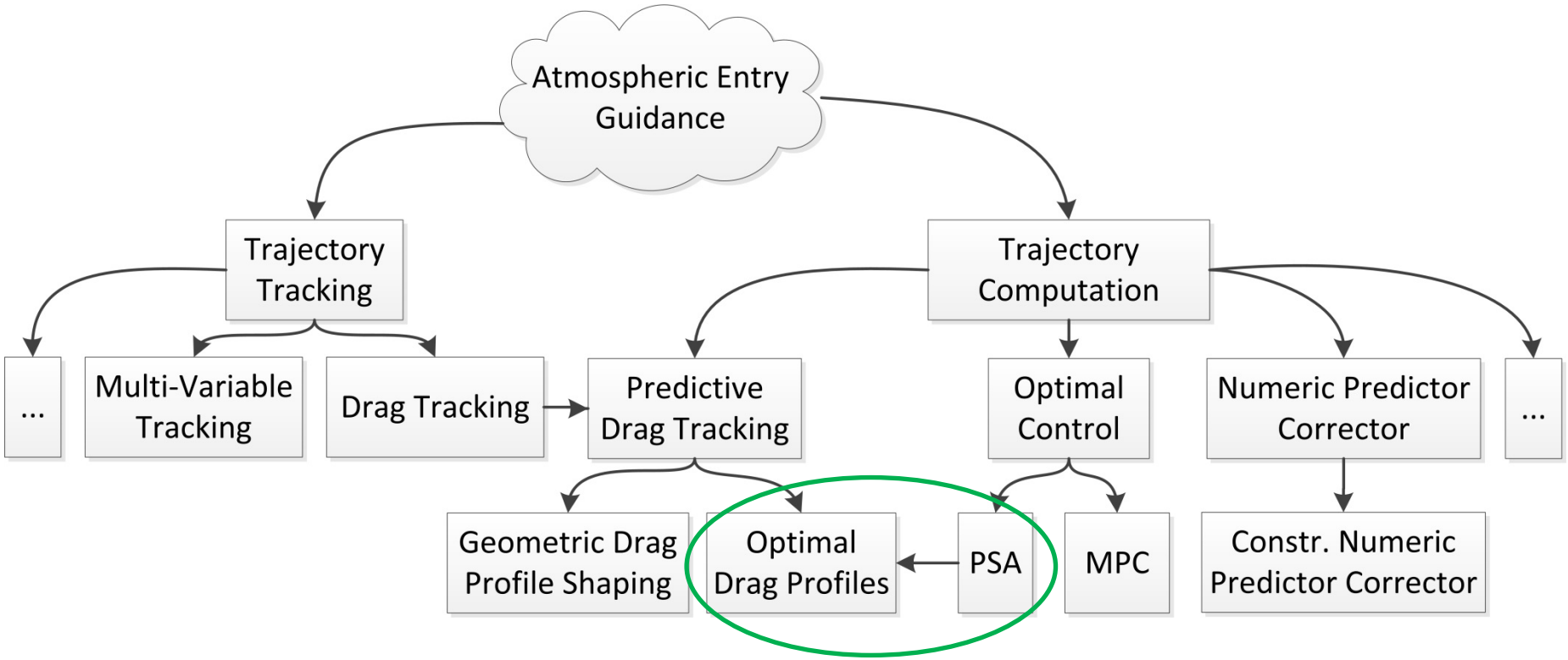
Optimization based Guidance

Property	NMPC Inf. Horizon	NMPC	Slow NLP / Tracking	PSA / Tracking
Computational power required	--	-	o/+	+/+
OBS complexity	--	--	--	+
Respect path constraint	++	++	+/-	o/-
Region of attraction	++	+	+/?	?/?
Optimality	++	+	+/?	+/?
Method universality	++	++	+/o	o/o
Online optimization	yes	yes	yes	no

Approaches to Entry Guidance



Approaches to Entry Guidance



- Optimal solution approximation based on parametric sensitivity analysis to obtain a fast update of the optimal control sequence
- Combination with drag tracking

Trajectory computation

- ❑ Parametric sensitivity analysis of nonlinear programs (offline)
- ❑ Fast solution approximation (online)
- ❑ Repeated online trajectory computation
- ❑ Region of attraction

esa

University Bremen

DLR

Offline Phase: Problem Formulation and Transcription

OCP(p)

Formulation as parametric Optimal Control Problem:

$$\min \quad J(x,u,p) = g\big(x(t_0), x(t_f), p\big) + \int_{t_o}^{t_f} l(x(t),u(t),p)\ dt$$

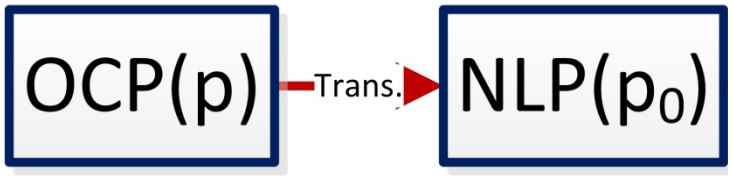
$$\text{w.r.t.} \quad \dot{x}(t) = f(x(t),u(t),p)$$

$$\psi_0(x(t_0),p) = 0$$

$$\psi_f\big(x(t_f),p\big) = 0$$

$$C(x(t),u(t),p) \leq 0 \qquad t\epsilon[t_0,t_f]$$

Offline Phase: Problem Formulation and Transcription



- Direct optimization
  - Discretization (Runge-Kutta-4, Linear control interpolation)
  - Transcription into a parametric nonlinear program (using single or multiple shooting)

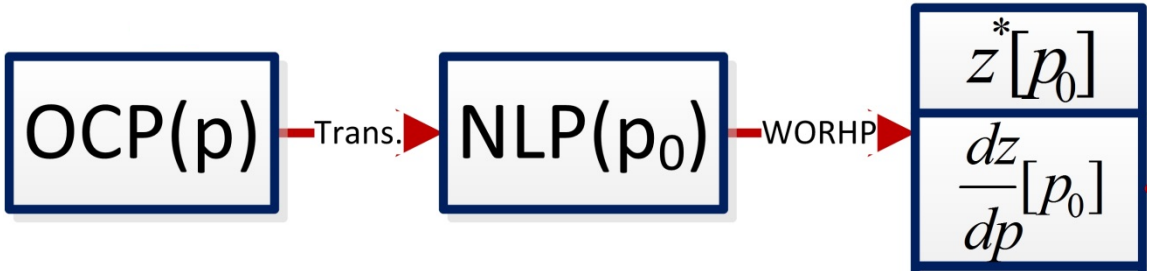
- Choice of a nominal parameter set $p_0 = 0$

$$\begin{aligned} \min_z \quad & F(z, p_0) \\ \text{w.r.t.} \quad & g_l \leq G(z, p_0) \leq g_u \end{aligned}$$

Offline Phase: Sensitivity Analysis I

- Obtain nominal optimal solution $z_0 = z^*(p_0)$

Offline Phase: Sensitivity Analysis I



- Obtain nominal optimal solution $z_0 = z^*(p_0)$
- Obtain parametric sensitivity $\frac{dz}{dp}[p_0]$

Linearization of the Necessary Optimality Conditions

Active constraints: $G^a(z,p)$

Lagrange multipliers: η^a

Lagrange function: $L(z,\eta^a,p) = F(z,p) + (\eta^a)^T G^a(z,p)$

Linearization of the Necessary Optimality Conditions

Active constraints: $G^a(z,p)$

Lagrange multipliers: η^a

Lagrange function: $L(z,\eta^a,p) = F(z,p) + (\eta^a)^T G^a(z,p)$

Necessary optimality conditions (KKT):

$$K(z(p),\eta^a(p),p) = \begin{bmatrix} \nabla_z L(z,\eta^a,p) \\ G^a(z,p) \end{bmatrix} = \begin{bmatrix} \nabla_z F(z,p) + (\eta^a)^T \nabla_z G^a(z,p) \\ G^a(z,p) \end{bmatrix} = 0$$

Linearization of the Necessary Optimality Conditions

Active constraints: $G^a(z,p)$

Lagrange multipliers: η^a

Lagrange function: $L(z,\eta^a,p) = F(z,p) + (\eta^a)^T G^a(z,p)$

Necessary optimality conditions (KKT):

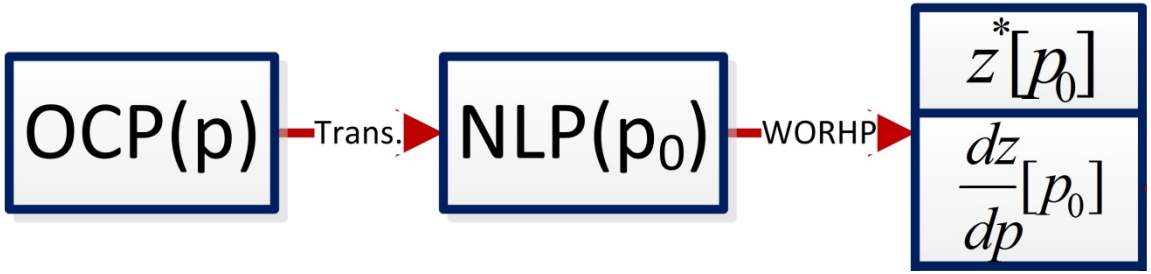
$$K(z(p),\eta^a(p),p) = \begin{bmatrix} \nabla_z L(z,\eta^a,p) \\ G^a(z,p) \end{bmatrix} = \begin{bmatrix} \nabla_z F(z,p) + (\eta^a)^T \nabla_z G^a(z,p) \\ G^a(z,p) \end{bmatrix} = 0$$

Differentiation of $K(z(p),\eta^a(p),p) \equiv 0$ with respect to p at point p_0:

$$\begin{bmatrix} \nabla_z{}^2 L(z_0,\eta_0^a,p_0) & \nabla_z G^a(z_0,p_0)^T \\ \nabla_z G^a(z_0,p_0) & 0 \end{bmatrix} \begin{bmatrix} \frac{dz}{dp}[p_0] \\ \frac{d\eta^a}{dp}[p_0] \end{bmatrix} + \begin{bmatrix} \nabla_{zp}{}^2 L(z_0,\eta_0^a,p_0) \\ \nabla_p G^a(z_0,p_0) \end{bmatrix} = 0$$

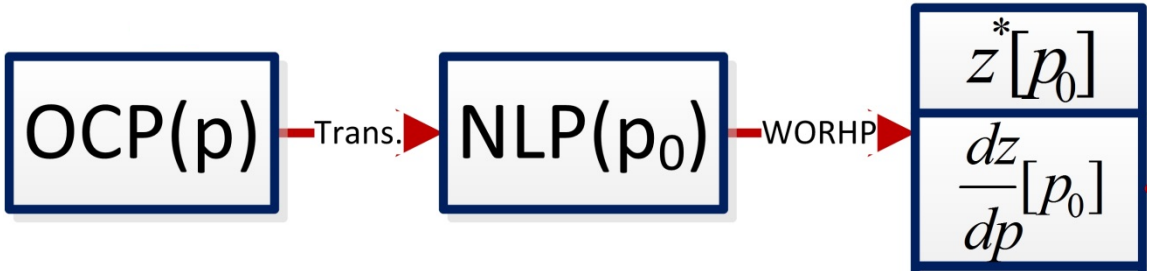
Literature: [Fiacco] [Büskens]

Offline Phase: Sensitivity Analysis I



- Obtain nominal optimal solution $z_0 = z^*(p_0)$
- Obtain parametric sensitivity $\frac{dz}{dp}[p_0]$

Offline Phase: Sensitivity Analysis I



- Obtain nominal optimal solution $z_0 = z^*(p_0)$
- Obtain parametric sensitivity $\frac{dz}{dp}[p_0]$
- If z_0 fulfills strong second order sufficient conditions: Kuhn-Tucker matrix is invertible

$$\begin{bmatrix} \boldsymbol{\frac{dz}{dp}[p_0]} \\ \frac{d\eta^a}{dp}[p_0] \end{bmatrix} = -\begin{bmatrix} \nabla_z{}^2 L(z_0, \eta_0^a, p_0) & \nabla_z G^a(z_0, p_0)^T \\ \nabla_z G^a(z_0, p_0) & 0 \end{bmatrix}^{-1} \begin{bmatrix} \nabla_{zp}{}^2 L(z_0, \eta_0^a, p_0) \\ \nabla_p G^a(z_0, p_0) \end{bmatrix}$$

Online Solution Approximation: p-Step

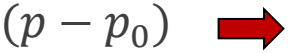
- Approximation of optimal solution for disturbed parameters p using Taylor expansion

$$z^*(p) \approx z_1 := z_0 + \frac{dz}{dp}[p_0] \cdot (p - p_0)$$

Online Solution Approximation: p-Step

- Approximation of optimal solution for disturbed parameters p using Taylor expansion

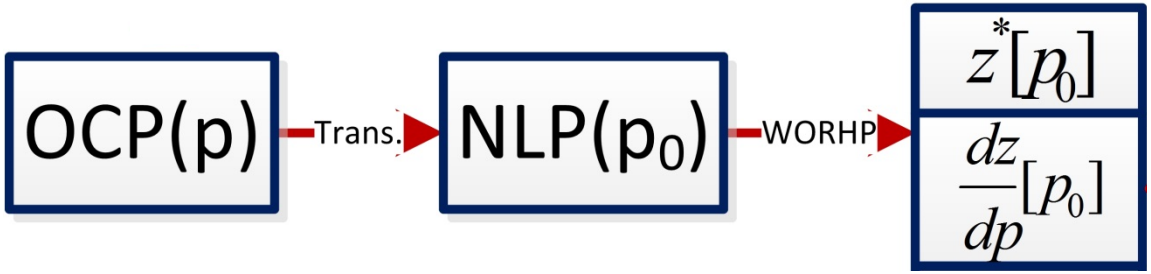
$$z^*(p) \approx z_1 := z_0 + \frac{dz}{dp}[p_0] \cdot (p - p_0)$$



error in the active constraints

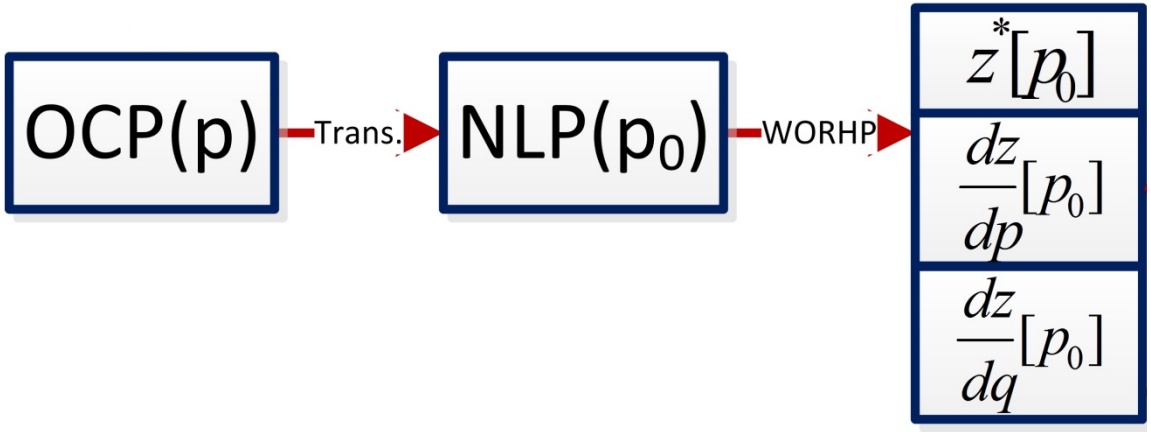
$$\|G^a(z_1, p)\| > 0$$

Offline Phase: Sensitivity Analysis II



$$\begin{aligned} \min_z \quad & F(z, p_0) \\ \text{w.r.t.} \quad & g_l \leq G(z, p_0) \leq g_u \end{aligned}$$

Offline Phase: Sensitivity Analysis II



- Additional parameter vector q with nominal value $q_0 = 0$

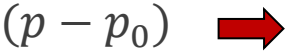
$$\begin{aligned} \min_z \quad & F(z, p_0) \\ \text{w.r.t.} \quad & g_l \leq G(z, p_0) - q_0 \leq g_u \end{aligned}$$

- $\frac{dz}{dq}$ can be computed analog to $\frac{dz}{dp}$

Online Solution Approximation: p-Step and q-Step

- Approximation of optimal solution for disturbed parameters p using Taylor expansion

$$z^*(p) \approx z_1 := z_0 + \frac{dz}{dp}[p_0] \cdot (p - p_0)$$



error in the active constraints

$$\|G^a(z_1, p)\| > 0$$

Online Solution Approximation: p-Step and q-Step

- Approximation of optimal solution for disturbed parameters p using Taylor expansion

$$z^*(p) \approx z_1 := z_0 + \frac{dz}{dp}[p_0] \cdot (p - p_0)$$

error in the active constraints

$$\|G^a(z_1, p)\| > 0$$

- $\frac{dz}{dq}$ is used to iteratively correct the constraint error and at the same time improve the optimality of the approximation

while $\|G^a(z_i, p)\| > \varepsilon$

$$q_i = G^a(z_i, p)$$

$$z_{i+1} := z_i + \frac{dz}{dq^a}[p_0] \cdot q_i$$

For $\|q - q_0\| < \delta$ iteration converges against a fixpoint z_∞ at which

$$\|G^a(z_\infty, p)\| = 0$$

[Büskens]

Example

- Scenario: Entry of a small capsule into the Martian atmosphere
- Control: Bank angle μ
- Assumptions: Constant AoA (capsule statically and dynamically stable)
- Dynamic model: Translational motion over a spherical, rotating planet (nonlinear, coupled, first order ODE system)

Example

- Scenario: Entry of a small capsule into the Martian atmosphere
- Control: Bank angle μ
- Assumptions: Constant AoA (capsule statically and dynamically stable)
- Dynamic model: Translational motion over a spherical, rotating planet (nonlinear, coupled, first order ODE system)

State	$x(t_0)$	$x(t_f)$
Altitude	$h_0 = 120000\ m$	$h_f = 10000\ m$
Longitude	$\lambda_0 = 0°$	$\lambda_f = 11.3°$
Latitude	$\varphi_0 = 25°$	$\varphi_f = 23.3°$
Velocity	$v_0 = 5440.8\ {}^{m}/_{s}$	$v_f \leq 450\ {}^{m}/_{s}$
FPA	$\gamma_0 = -14.5°$	$\gamma_f : free$
Heading	$\chi_0 = 97.4°$	$\chi_f : free$

Path Constraint	**Limit**
Heat flux	$\dot{Q} \leq 1600$ kW/m^2
Dynamic pressure	$q \leq 17$ kPa
Load factor	$n \leq 15$

Example

Parametrization

Initial condition: $\psi_0 = x(t_0) + p_I$

Vehicle mass: $m = m_0 + p_m$

Lift coefficient: $C_L = C_{L_0} + p_L C_{L_0}$

Drag coefficient: $C_D = C_{D_0} + p_D C_{D_0}$

Parameter: $p = (p_I, p_m, p_L, p_D)^T$

Nominal value: $p_0 = (0,0,0,0,0,0,0,0,0)^T = 0$

Example

Parametrization

Initial condition: $\psi_0 = x(t_0) + p_I$

Vehicle mass: $m = m_0 + p_m$

Lift coefficient: $C_L = C_{L_0} + p_L C_{L_0}$

Drag coefficient: $C_D = C_{D_0} + p_D C_{D_0}$

Parameter: $p = (p_I, p_m, p_L, p_D)^T$

Nominal value: $p_0 = (0,0,0,0,0,0,0,0)^T = 0$

Performance index

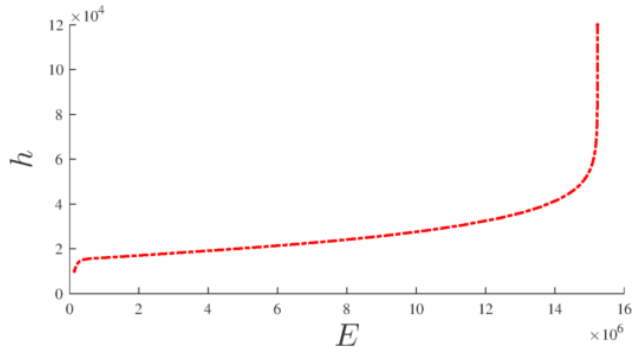
- Minimize terminal velocity v_f
- Avoid flight at max/min lift
- “Smooth” control function

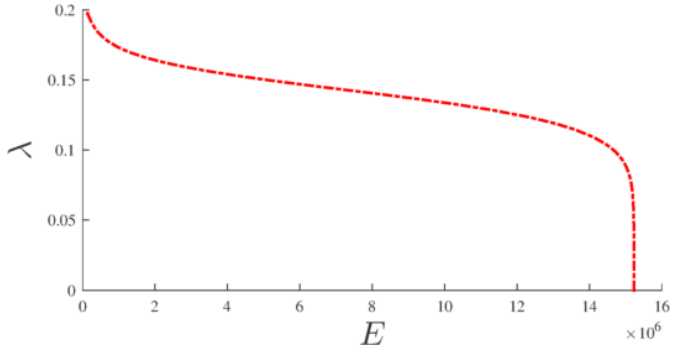
Objective function: $F(x,u,p) = w_1 v_f^2 + w_2 \int_{t_0}^{t_f} w_e(t)[w_3(\cos\mu)^2 + w_4\dot{\mu}^2]\,dt$

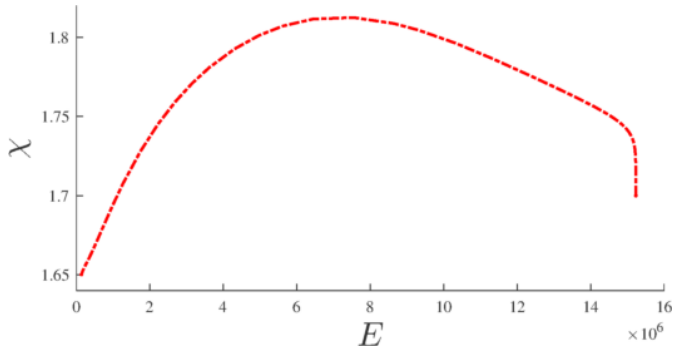
w_1, w_2, w_3, w_4: positive, constant

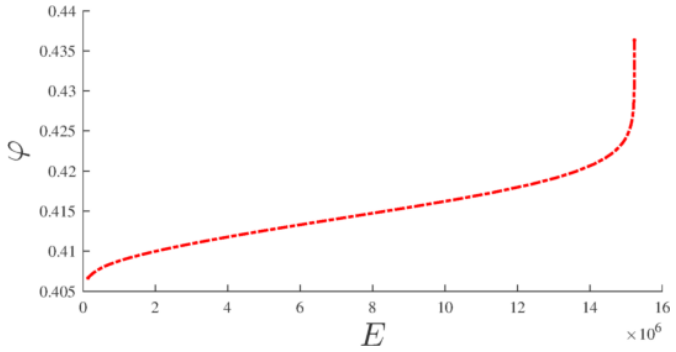
$w_e(t)$: positive function

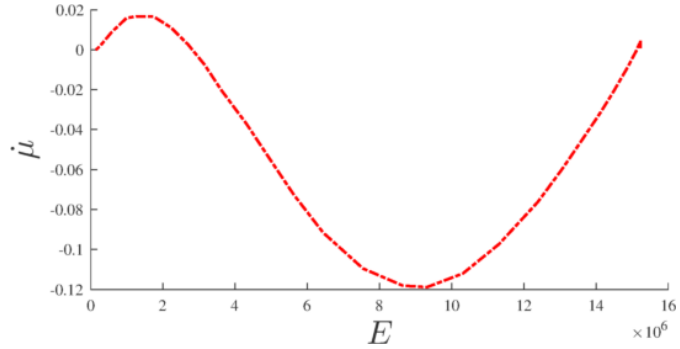
Nominal Solution ($p := p_0$)

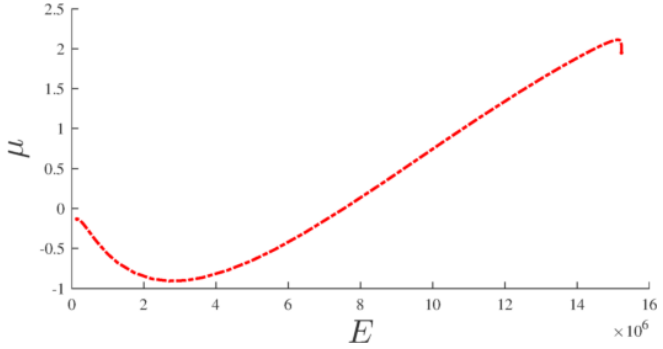












Perturbed Environment

Example: Estimated conditions at flight time t_0:

Perturbed initial conditions:

$\bar{x}_0(t_0)$	$\bar{p}_I$
$\overline{h_0} = h_0 +$	2000
$\overline{\lambda_0} = \lambda_0 -$	0.001
$\overline{\varphi_0} = \varphi_0 +$	0.005
$\overline{v_0} = v_0 -$	100
$\overline{\gamma_0} = \gamma_0 -$	0.0012
$\overline{\chi_0} = \chi_0 -$	0.0005

Perturbed model:

$\bar{p}_m = 9$

$\bar{p}_L = 0.03$

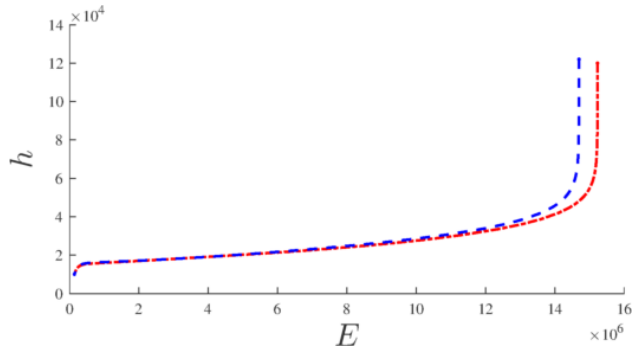
$\bar{p}_D = -0.04$

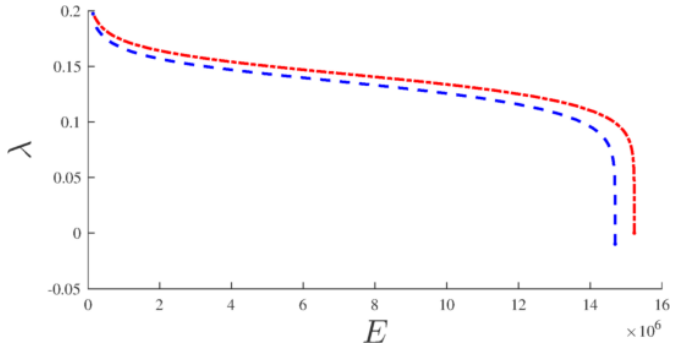
Perturbed parameter set:

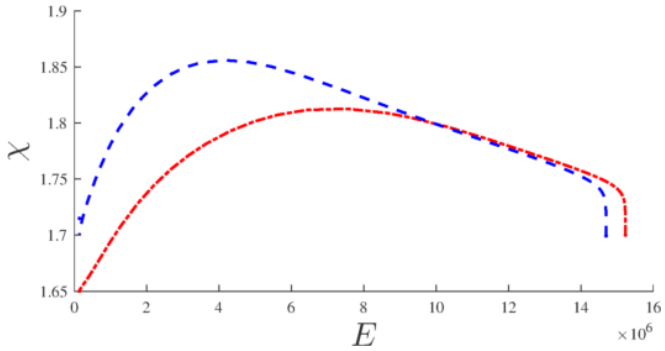
$$\bar{p} = (\bar{p}_I, \bar{p}_m, \bar{p}_L, \bar{p}_D)^T$$

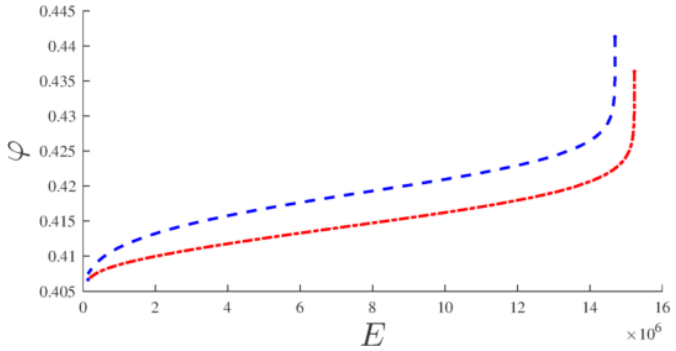
Approximate optimal control sequence and trajectory for perturbed parameters $\bar{p}$

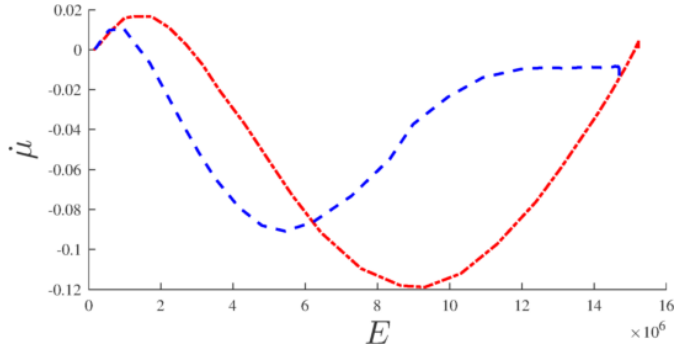
Solution Approximation for $p := \overline{p}$ (p-Step)

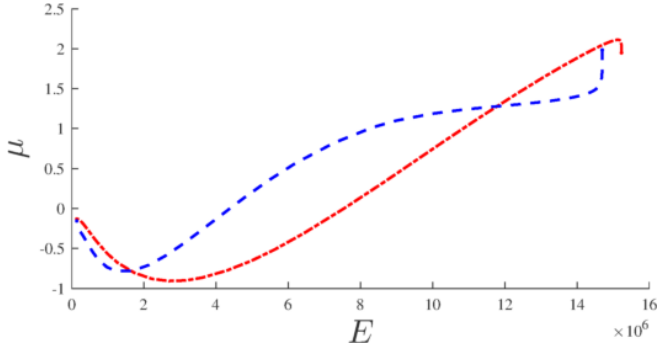




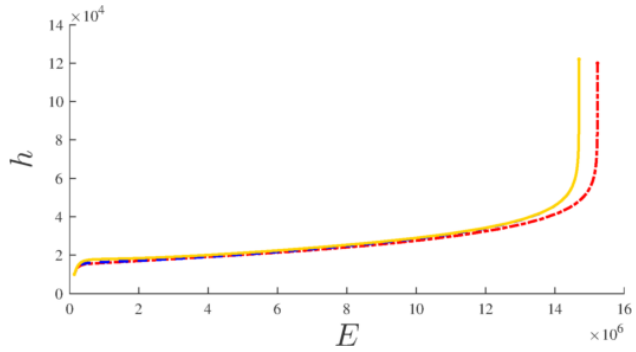


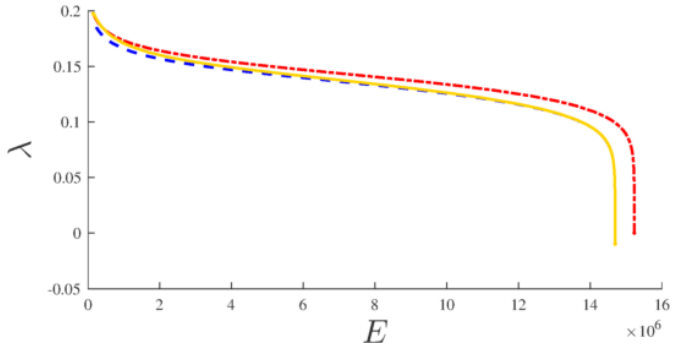


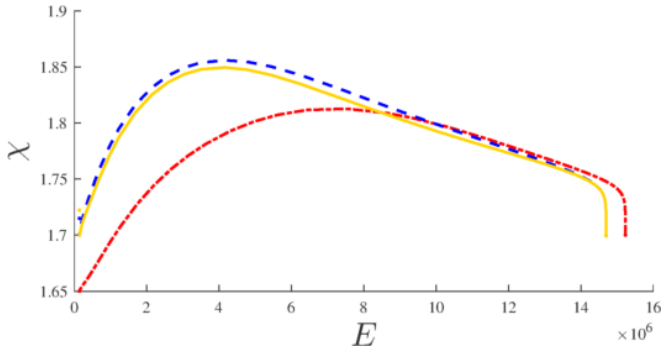


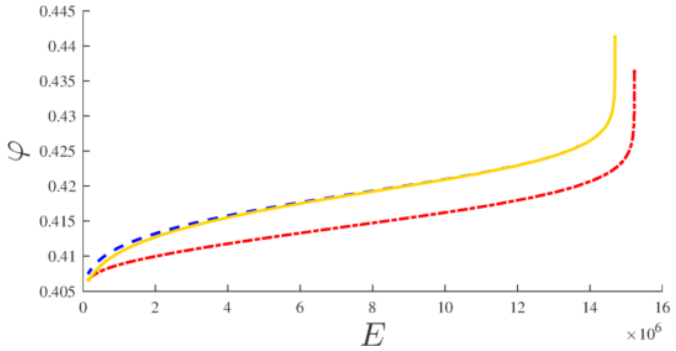


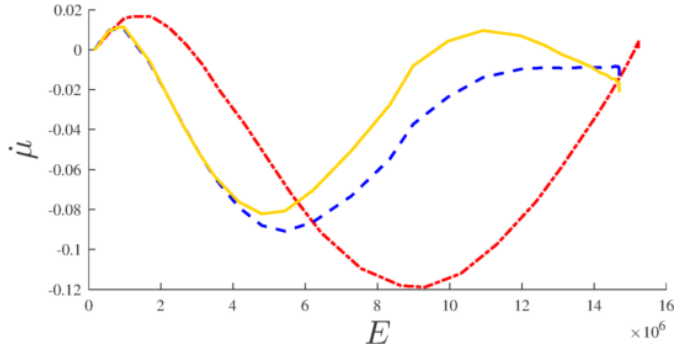
Solution Approximation for $p := \overline{p}$ (q-Step 1)

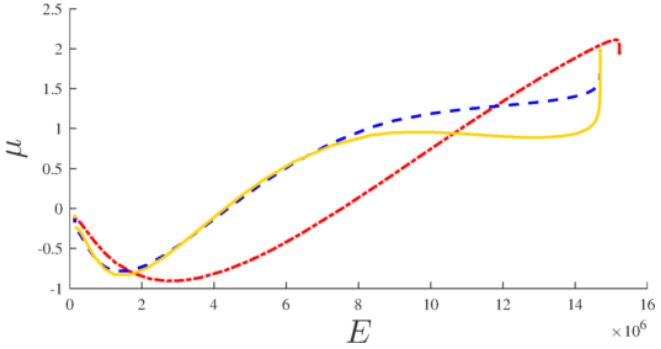




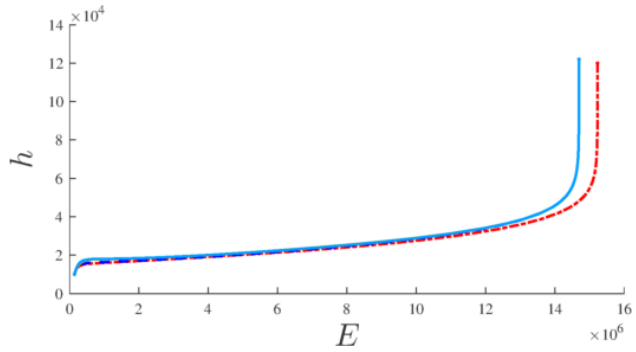


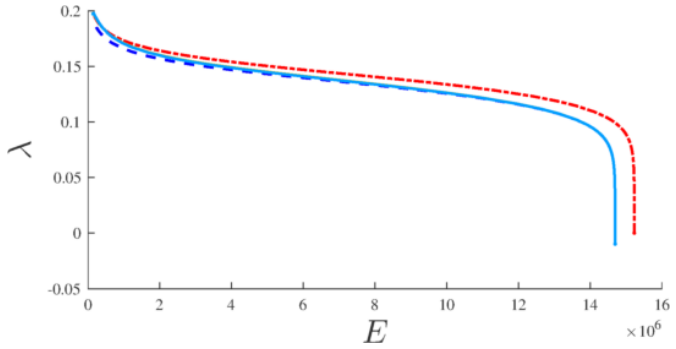


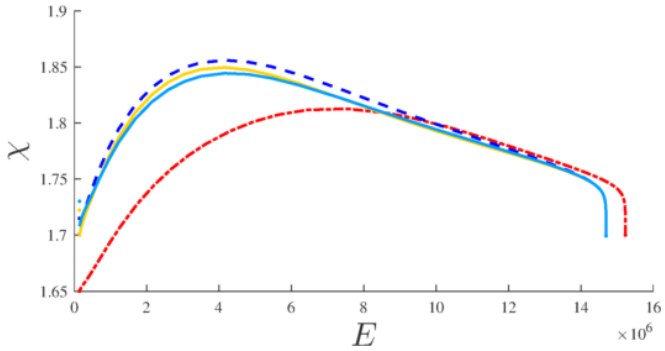


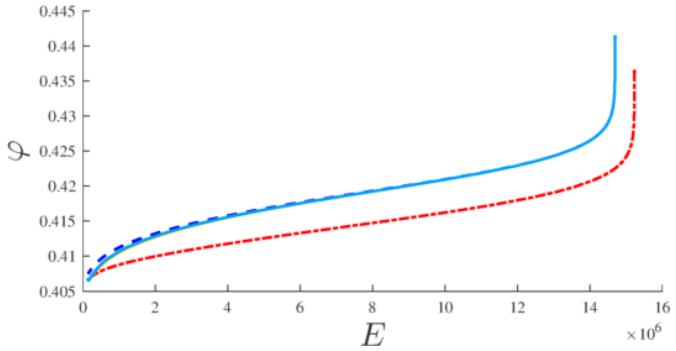


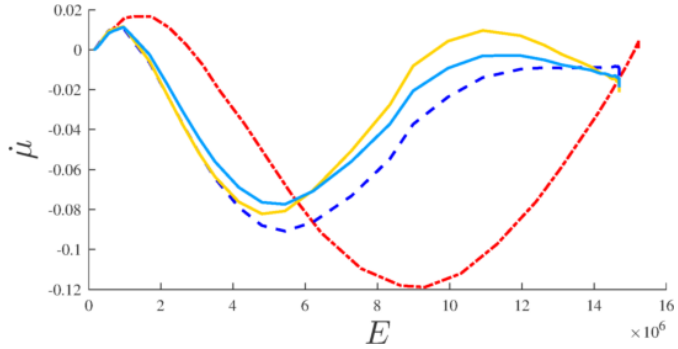
Solution Approximation for $p := \overline{p}$ (q-Step 2)

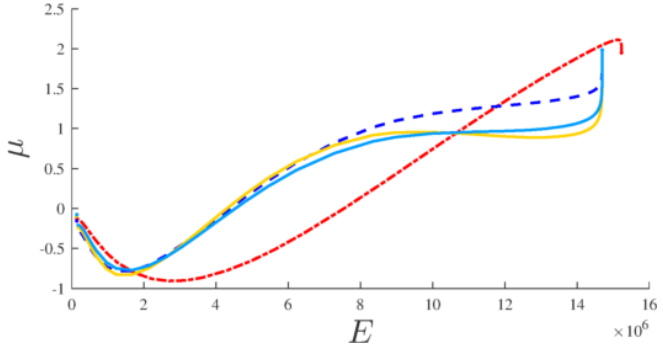




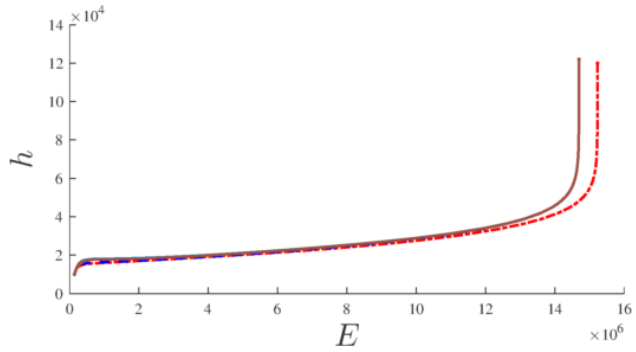


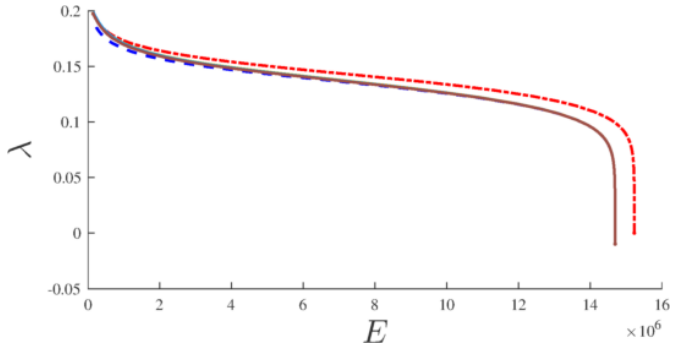


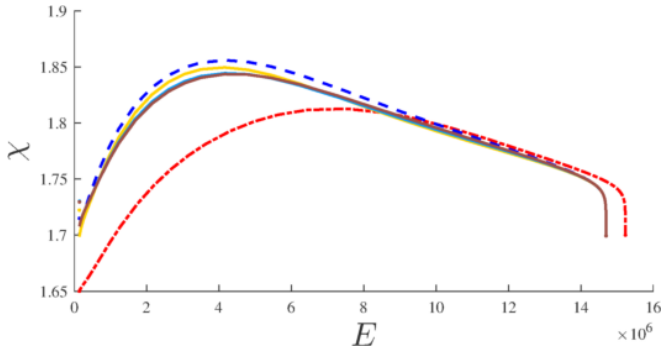


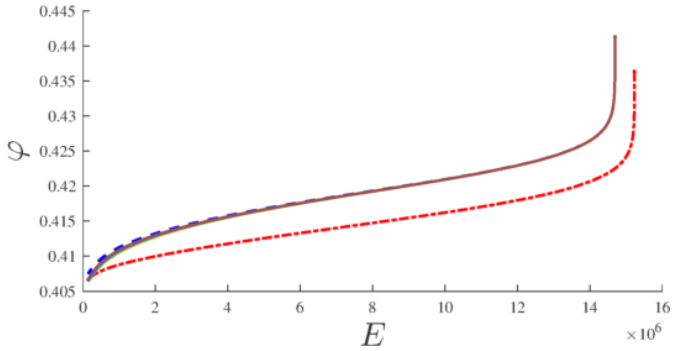


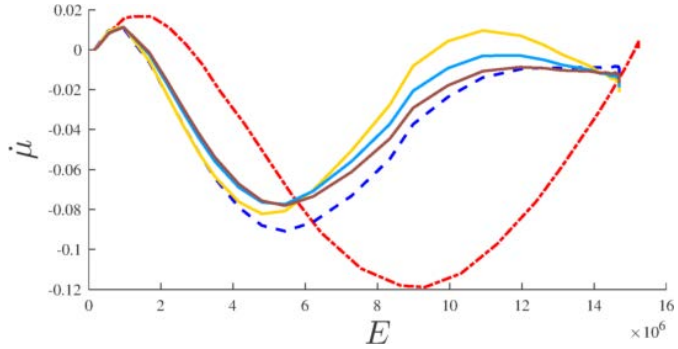
Solution Approximation for $p := \overline{p}$ (q-Step 3)

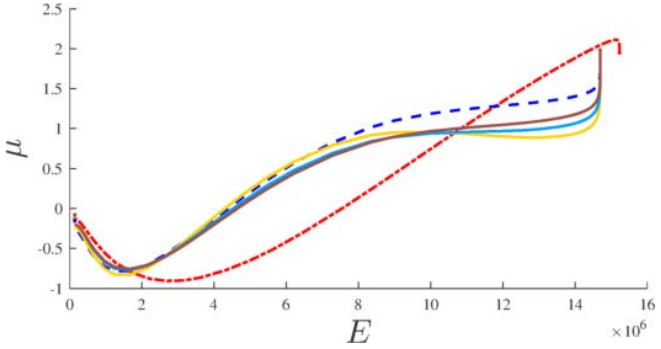




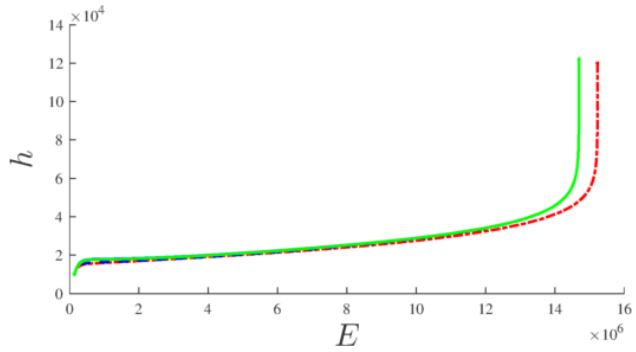


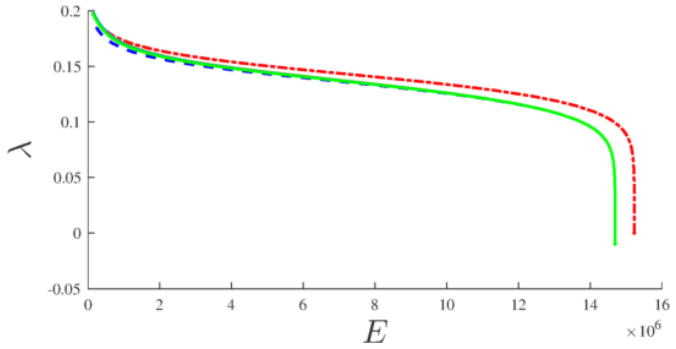


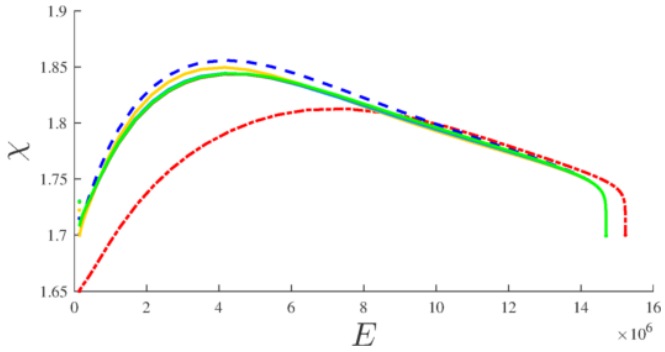


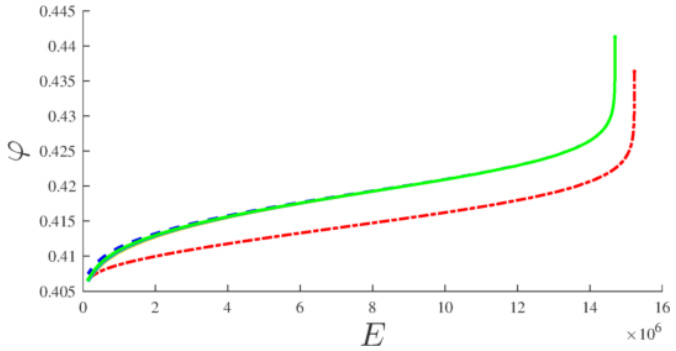


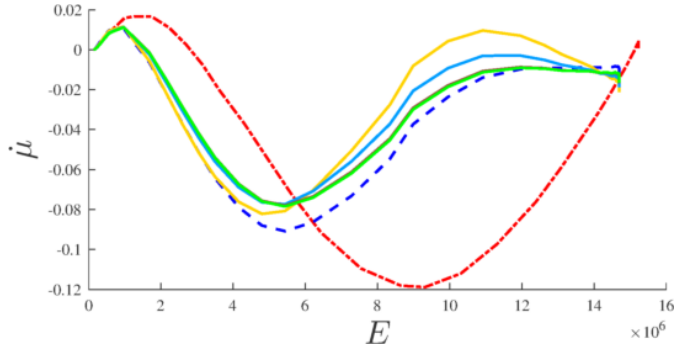
Solution Approximation for $p := \overline{p}$ (q-Step 4) (Termination)

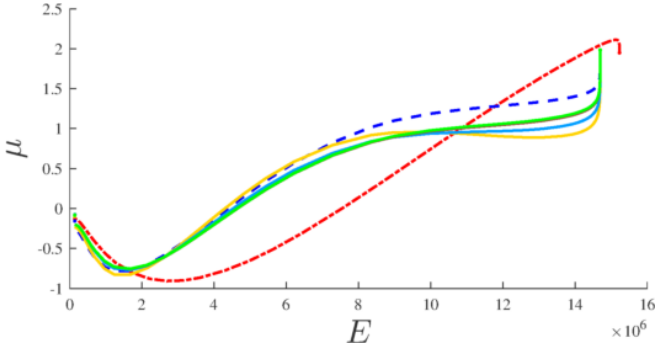






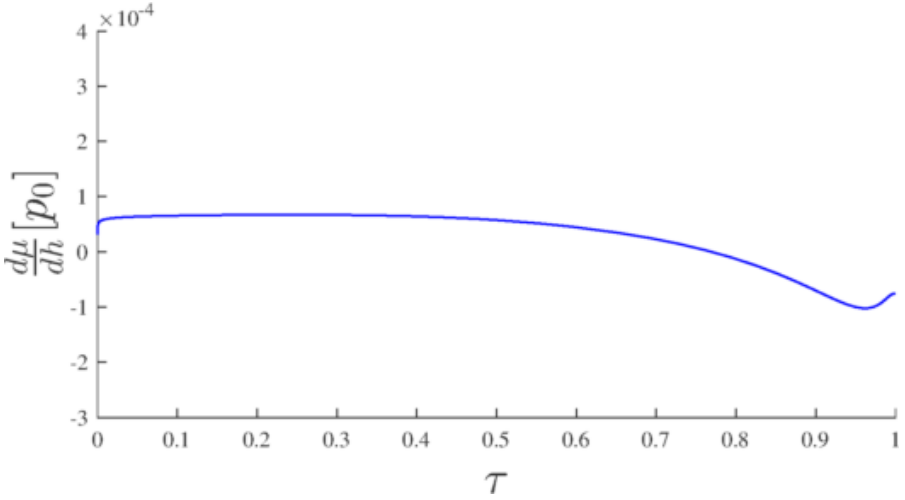






Parametric Sensitivities on Discrete Points of the Nominal Trajectory

Image 1: Sensitivity of μ against perturbations in h at $\psi_0^0 = x^*(t_0)$



Parametric Sensitivities on Discrete Points of the Nominal Trajectory

Image 1: Sensitivity of μ against perturbations in h at $\psi_0^0 = x^*(t_0)$

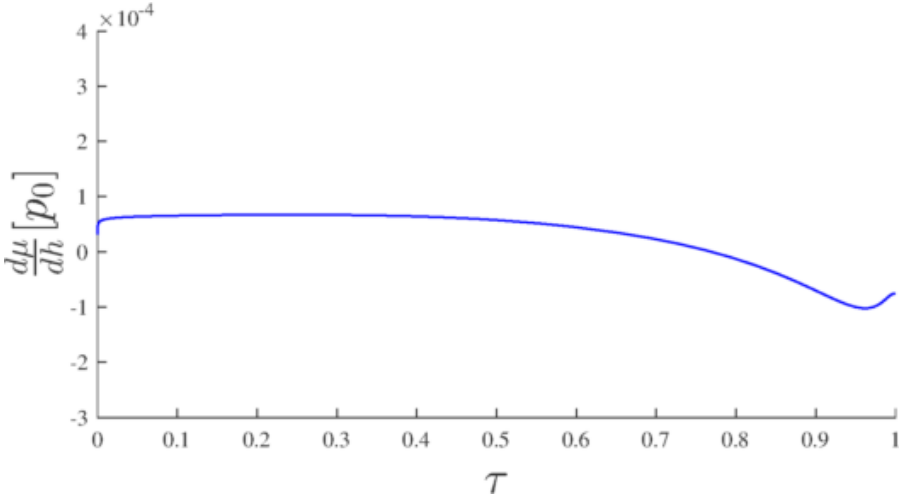
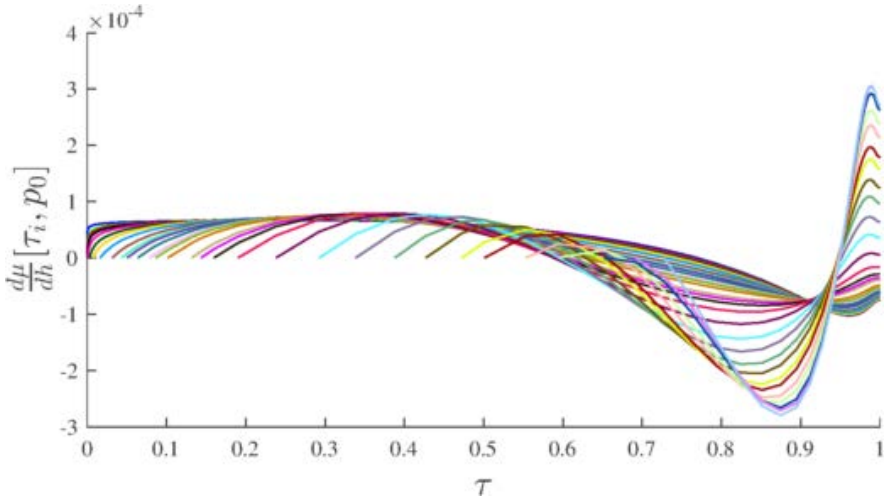
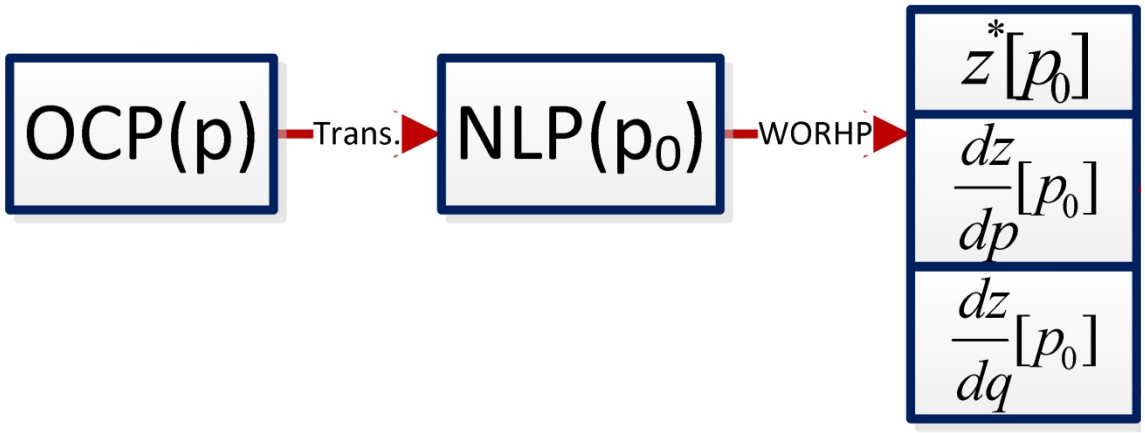


Image 2: Sensitivity of μ against perturbations in h at $\psi_0^{t_i} = x^*(t_i),\ 0 \leq i \leq l$

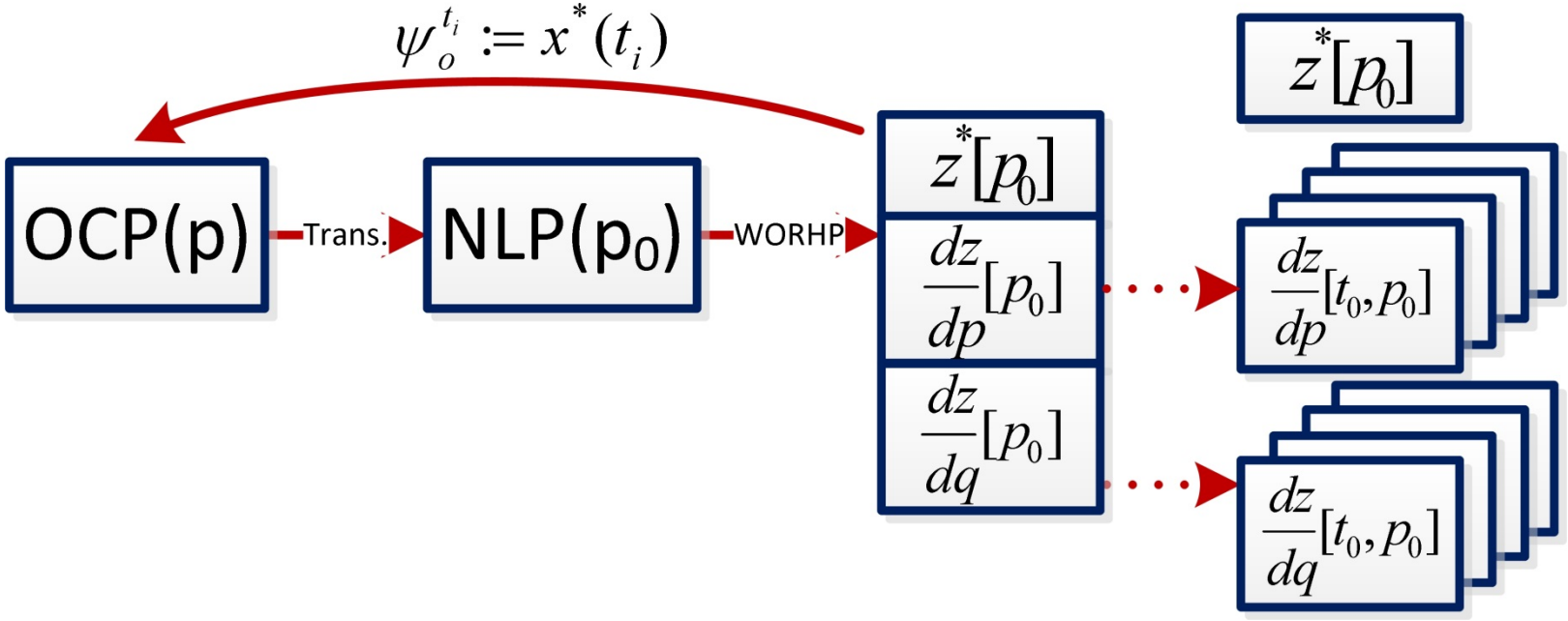


Offline Phase: Sensitivity Catalog



- Trajectory computation at a fixed time t requires sensitivity differentials for initial condition $\psi_0^t := x^*(t),\ t \in [t_0, t_f)$

Offline Phase: Sensitivity Catalog

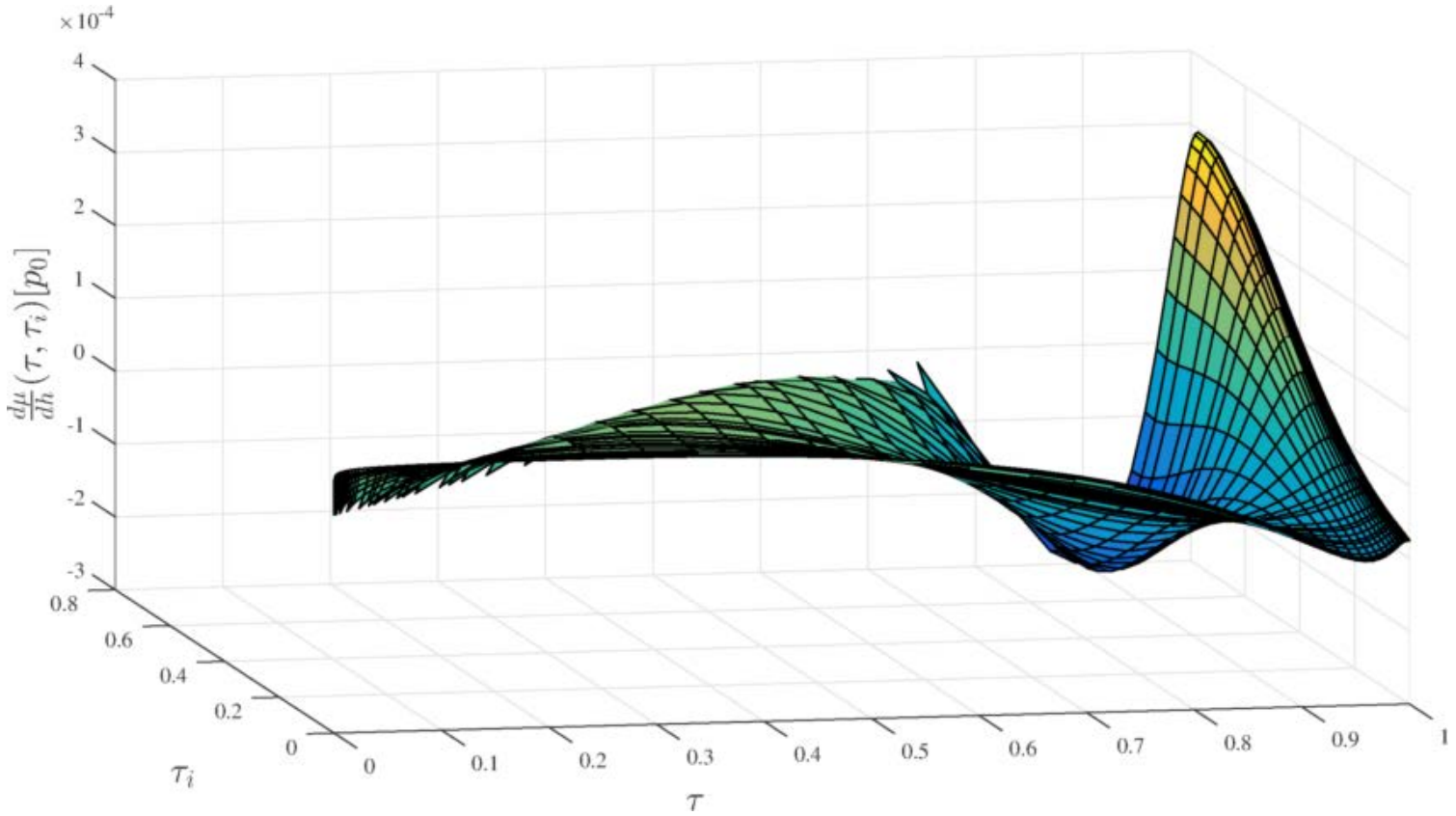


- Trajectory computation at a fixed time t requires sensitivity differentials for initial condition $\psi_0^t := x^*(t)$, $t \in [t_0, t_f)$

- Sensitivity analysis is repeated at discrete points $t_i \in [t_0, t_f)$, $0 < i \leq l$ of the nominal trajectory $x^*(t)$

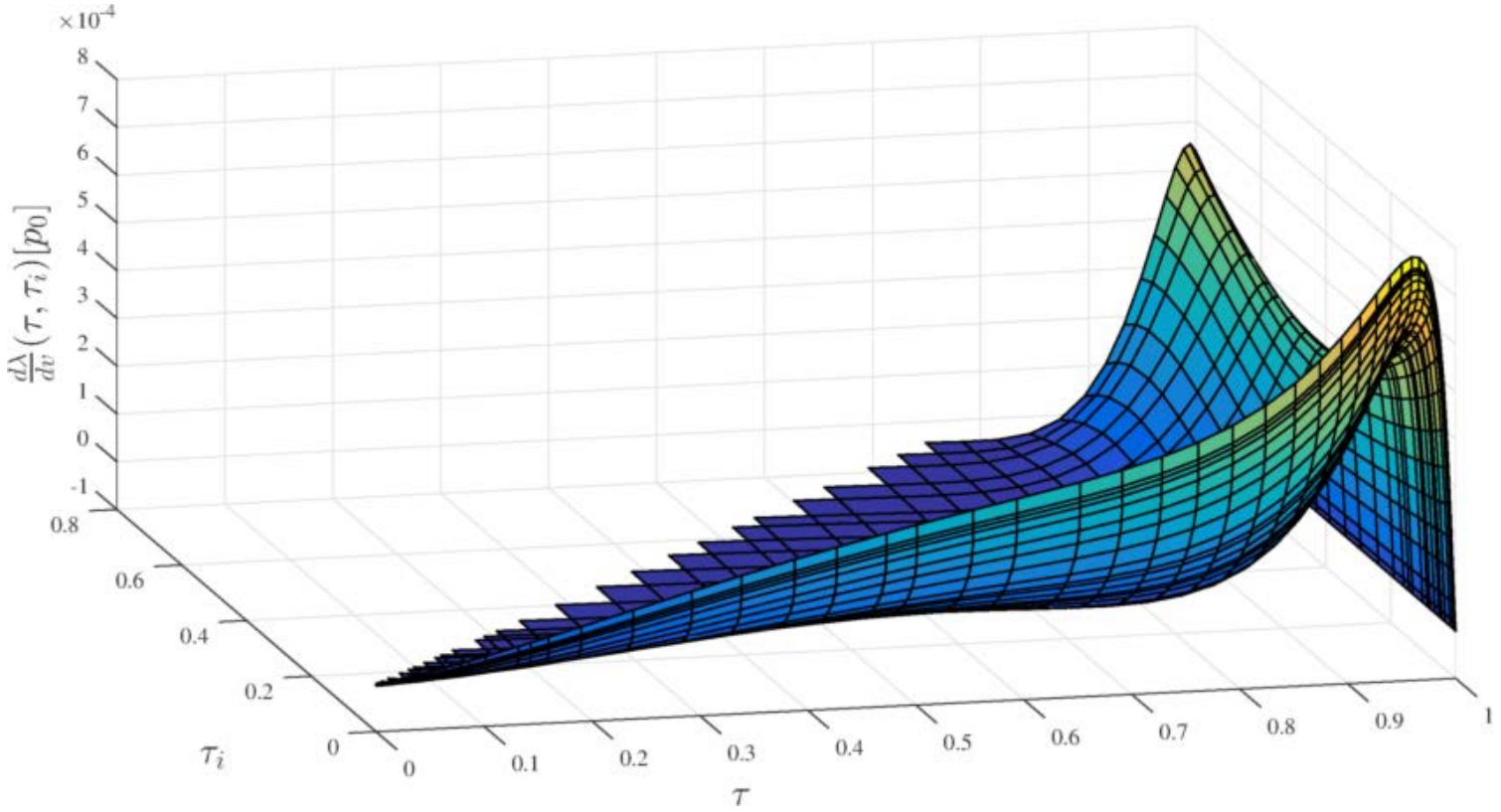
Interpolation of Parametric Sensitivity Differentials

- Interpolate parametric sensitivities between sufficiently close initial conditions (required for all combinations of states/controls and perturbation parameters)



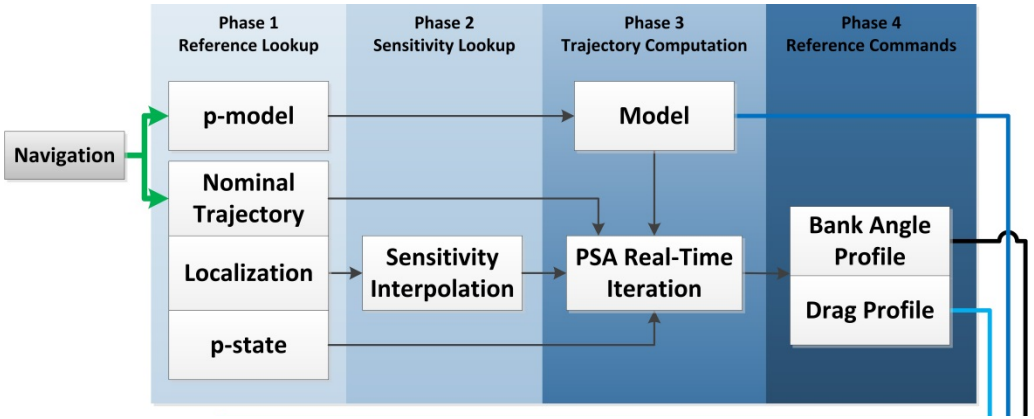
Interpolation of Parametric Sensitivity Differentials

- Interpolate parametric sensitivities between sufficiently close initial conditions (required for all combinations of states/controls and perturbation parameters)



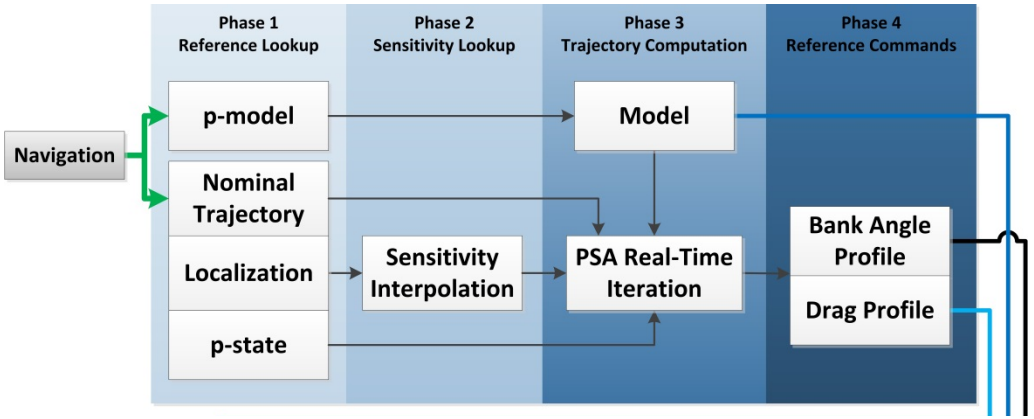
Procedure: Repeated Online Trajectory Computation

1. Estimate current state $\bar{x}$ and model perturbations $\bar{p}_m, \bar{p}_L, \bar{p}_D$



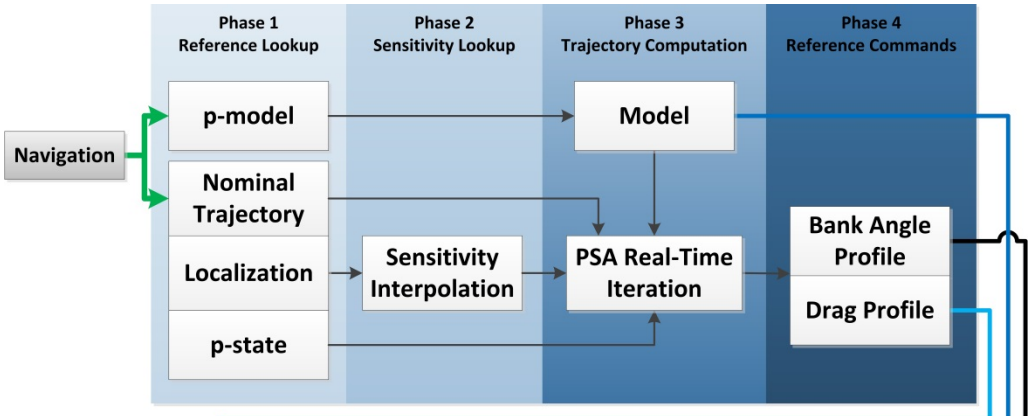
Procedure: Repeated Online Trajectory Computation

1. Estimate current state $\bar{x}$ and model perturbations $\bar{p}_m, \bar{p}_L, \bar{p}_D$
2. Determine $\bar{t}$ such that $\|\bar{p}_I\|$ is sufficiently small with $\bar{p}_I = \bar{x} - x^*(\bar{t}, p_0)$



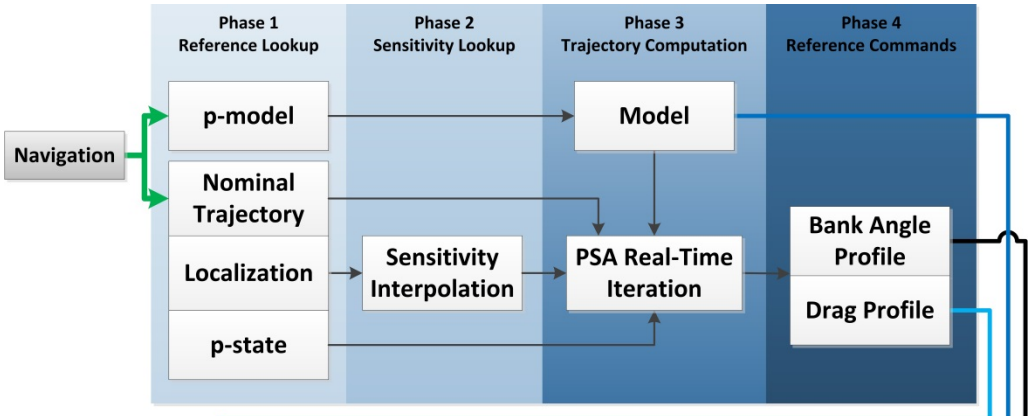
Procedure: Repeated Online Trajectory Computation

1. Estimate current state $\bar{x}$ and model perturbations $\bar{p}_m, \bar{p}_L, \bar{p}_D$
2. Determine $\bar{t}$ such that $\|\bar{p}_I\|$ is sufficiently small with $\bar{p}_I = \bar{x} - x^*(\bar{t}, p_0)$
3. Set $p := (\bar{p}_I, \bar{p}_m, \bar{p}_L, \bar{p}_D)^T$



Procedure: Repeated Online Trajectory Computation

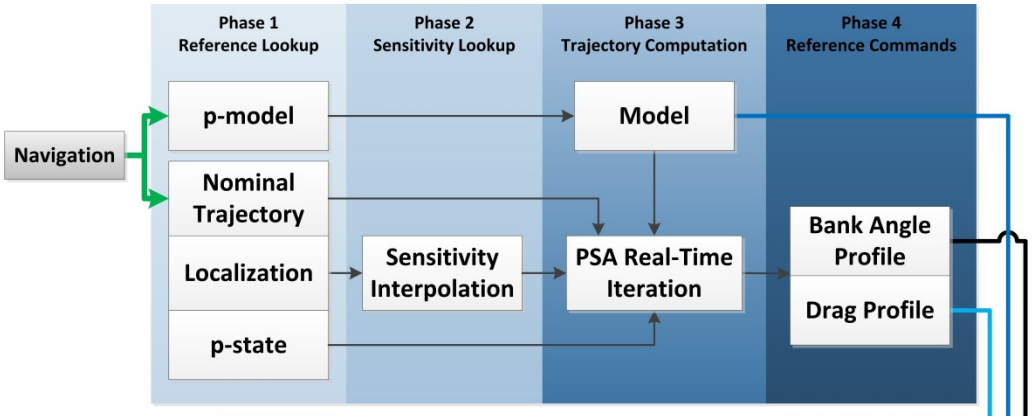
1. Estimate current state $\bar{x}$ and model perturbations $\bar{p}_m, \bar{p}_L, \bar{p}_D$
2. Determine $\bar{t}$ such that $\|\bar{p}_I\|$ is sufficiently small with $\bar{p}_I = \bar{x} - x^*(\bar{t}, p_0)$
3. Set $p := (\bar{p}_I, \bar{p}_m, \bar{p}_L, \bar{p}_D)^T$



4. Evaluate sensitivity surfaces at $\bar{t}$ to obtain parametric sensitivities approximations $\frac{\widetilde{dz}}{dp}[\bar{t}, p_0]$, $\frac{\widetilde{dz}}{dq}[\bar{t}, q_0]$ (corresponding to initial condition $\psi_0^{\bar{t}} := x^*(\bar{t}, p_0)$)

Procedure: Repeated Online Trajectory Computation

1. Estimate current state $\bar{x}$ and model perturbations $\bar{p}_m, \bar{p}_L, \bar{p}_D$
2. Determine $\bar{t}$ such that $\|\bar{p}_I\|$ is sufficiently small with $\bar{p}_I = \bar{x} - x^*(\bar{t}, p_0)$
3. Set $p := (\bar{p}_I, \bar{p}_m, \bar{p}_L, \bar{p}_D)^T$



4. Evaluate sensitivity surfaces at $\bar{t}$ to obtain parametric sensitivities approximations $\frac{\widetilde{dz}}{dp}[\bar{t}, p_0], \frac{\widetilde{dz}}{dq}[\bar{t}, q_0]$ (corresponding to initial condition $\psi_0^{\bar{t}} := x^*(\bar{t}, p_0)$)
5. Execute real-time iteration scheme with arguments
   a. $x^*(t, p_0), t \in [\bar{t}, t_f]$ (remaining nominal trajectory)
   b. $\frac{\widetilde{dz}}{dp}[\bar{t}, p_0], \frac{\widetilde{dz}}{dq}[\bar{t}, q_0]$ (interpolated parametric sensitivities)
   c. $p := (\bar{p}_I, \bar{p}_m, \bar{p}_L, \bar{p}_D)^T$ (perturbation parameters)

Convergence Region

What are the maximal perturbations that can be compensated? ⟺ Convergence region of the real-time iteration scheme?

- If the functions F and G are three times continuously differentiable and strong sufficient conditions of optimality hold:

  $\exists\, U(p_0)$ of p_0 such that $\forall\, p = p_0 + \Delta p \,\epsilon\, U$ it holds that $\|G^a(z_\infty, p)\| = 0$

- Extend of U: No analytic answer for nonlinear problems
- U is defined for a fixed set of active constraints G^a
- If G^a changes for a perturbation $p \quad \Rightarrow \quad p \notin U$

Convergence Region

What are the maximal perturbations that can be compensated? ⟺ Convergence region of the real-time iteration scheme?

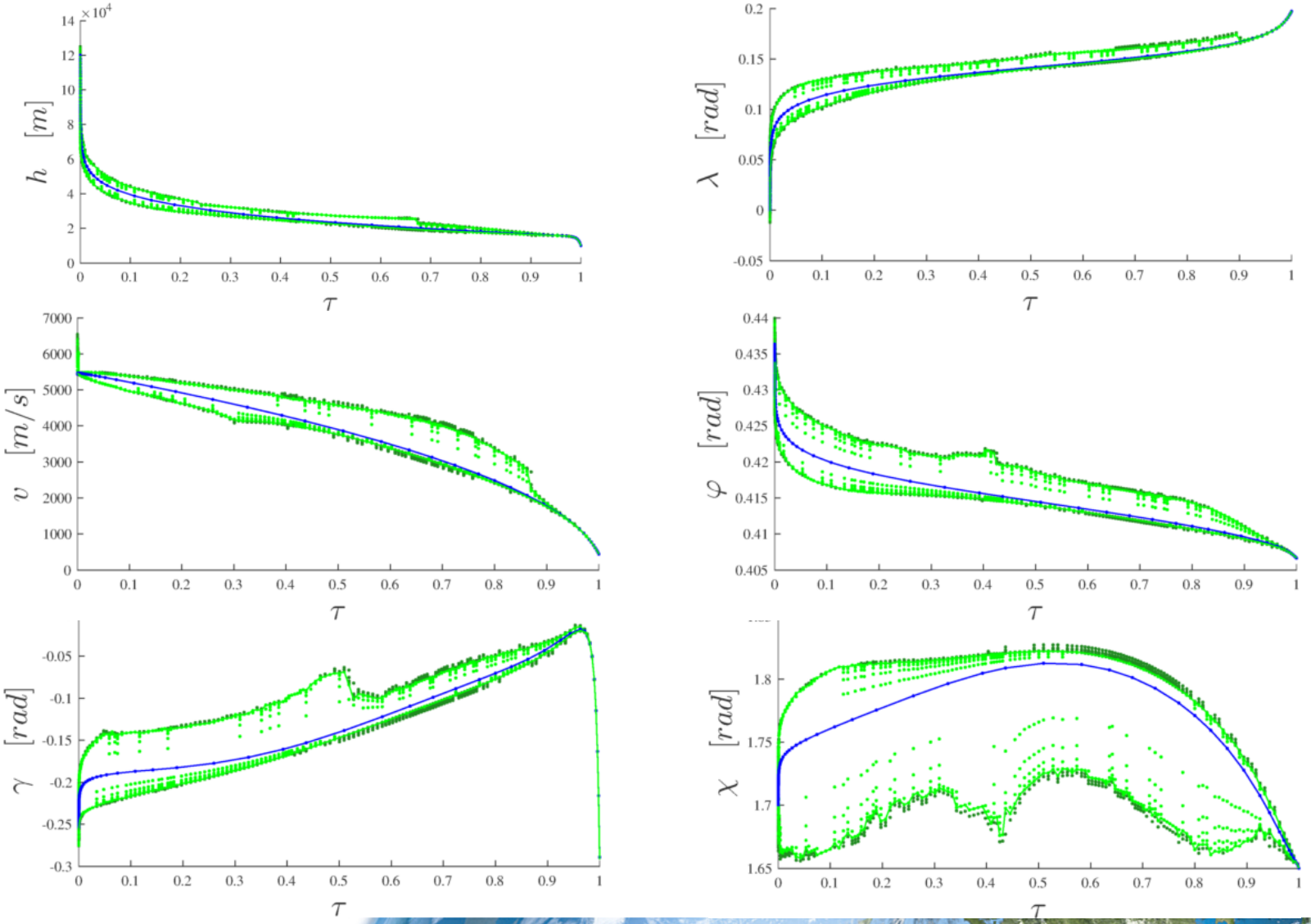
- If the functions F and G are three times continuously differentiable and strong sufficient conditions of optimality hold:

  $\exists\, U(p_0)$ of p_0 such that $\forall\, p = p_0 + \Delta p \,\epsilon\, U$ it holds that $\|G^a(z_\infty, p)\| = 0$

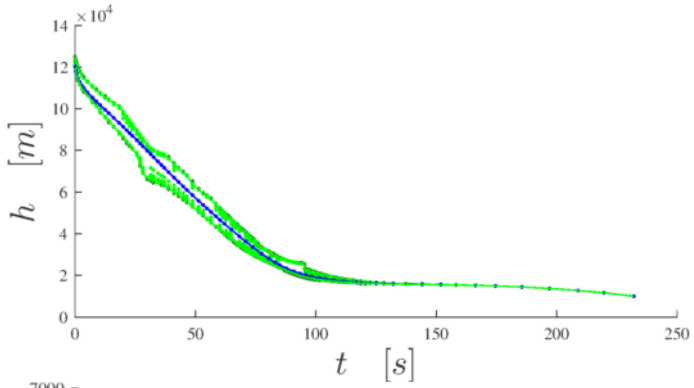
- Extend of U: No analytic answer for nonlinear problems
- U is defined for a fixed set of active constraints G^a
- If G^a changes for a perturbation $p \quad \Rightarrow \quad p \notin U$

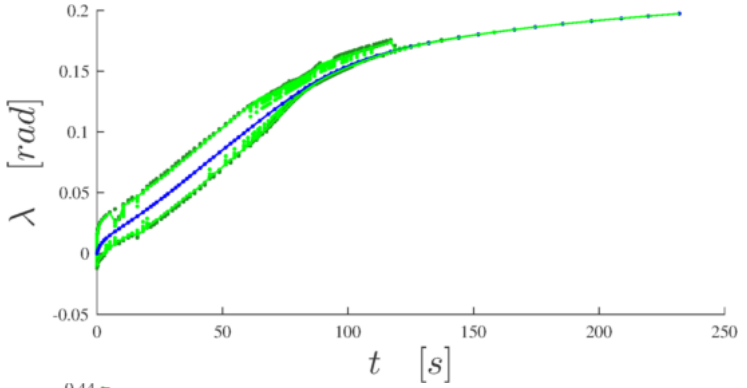
Investigate convergence region!
(for selected perturbations)

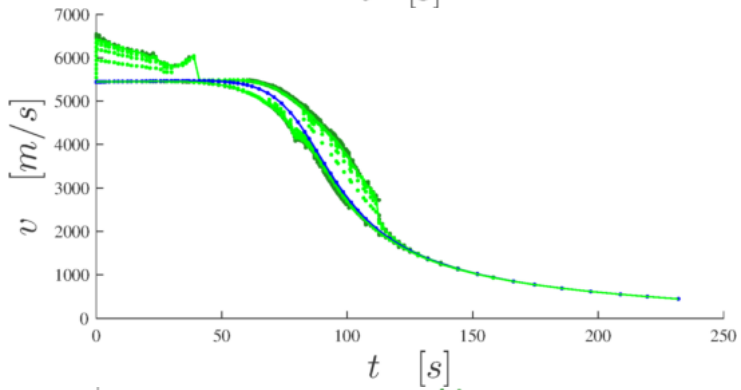
Convergence Region : Single State Error (vs. Norm. Energy)

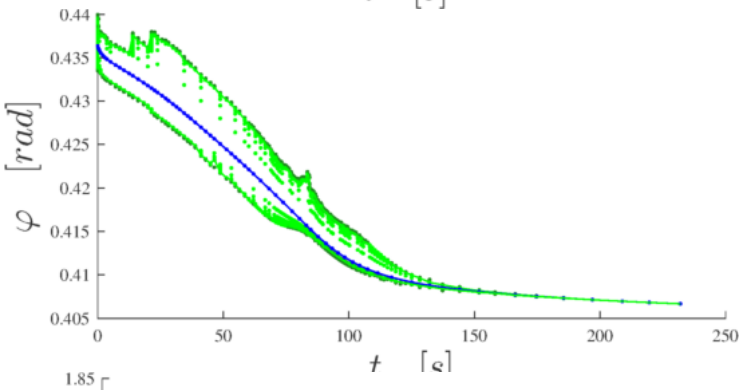


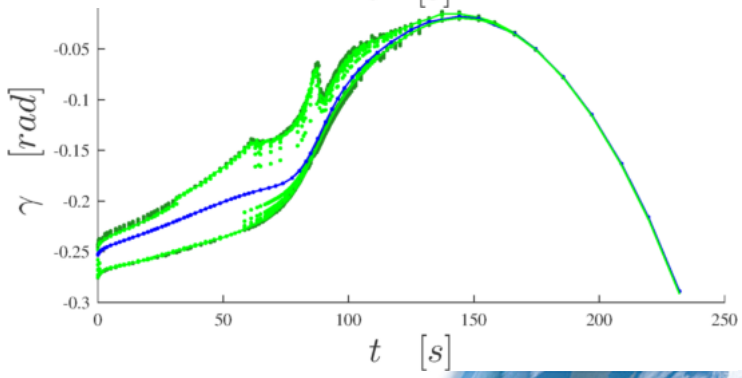
Convergence Region : Single State Error (vs. Time)

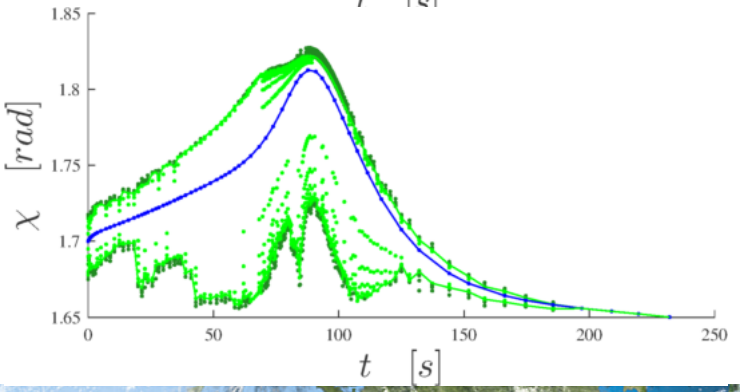












Convergence Region : Some Conclusions

- The convergence region $U(p_0)$ is large enough to cover expected errors at the entry interface point

- The choice of the reference point $x^*(\bar{t}, p_0)$ is important! Metric $|.|$ desirable such that:
  - $\bar{p}_I = min_{\bar{t}\in[t_0,t_f]} \; |\bar{x} - x^*(\bar{t}, p_0)| \;\; \Rightarrow \;\; \bar{p}_I \in U$

- U shrinks after peak deceleration such that successful trajectory computation cannot be guaranteed

Convergence Region : Some Conclusions

- The convergence region $U(p_0)$ is large enough to cover expected errors at the entry interface point

- The choice of the reference point $x^*(\bar{t}, p_0)$ is important! Metric $|.|$ desirable such that:
  - $\bar{p}_I = min_{\bar{t}\in[t_0,t_f]} \; |\bar{x} - x^*(\bar{t}, p_0)| \;\; \Rightarrow \;\; \bar{p}_I \in U$

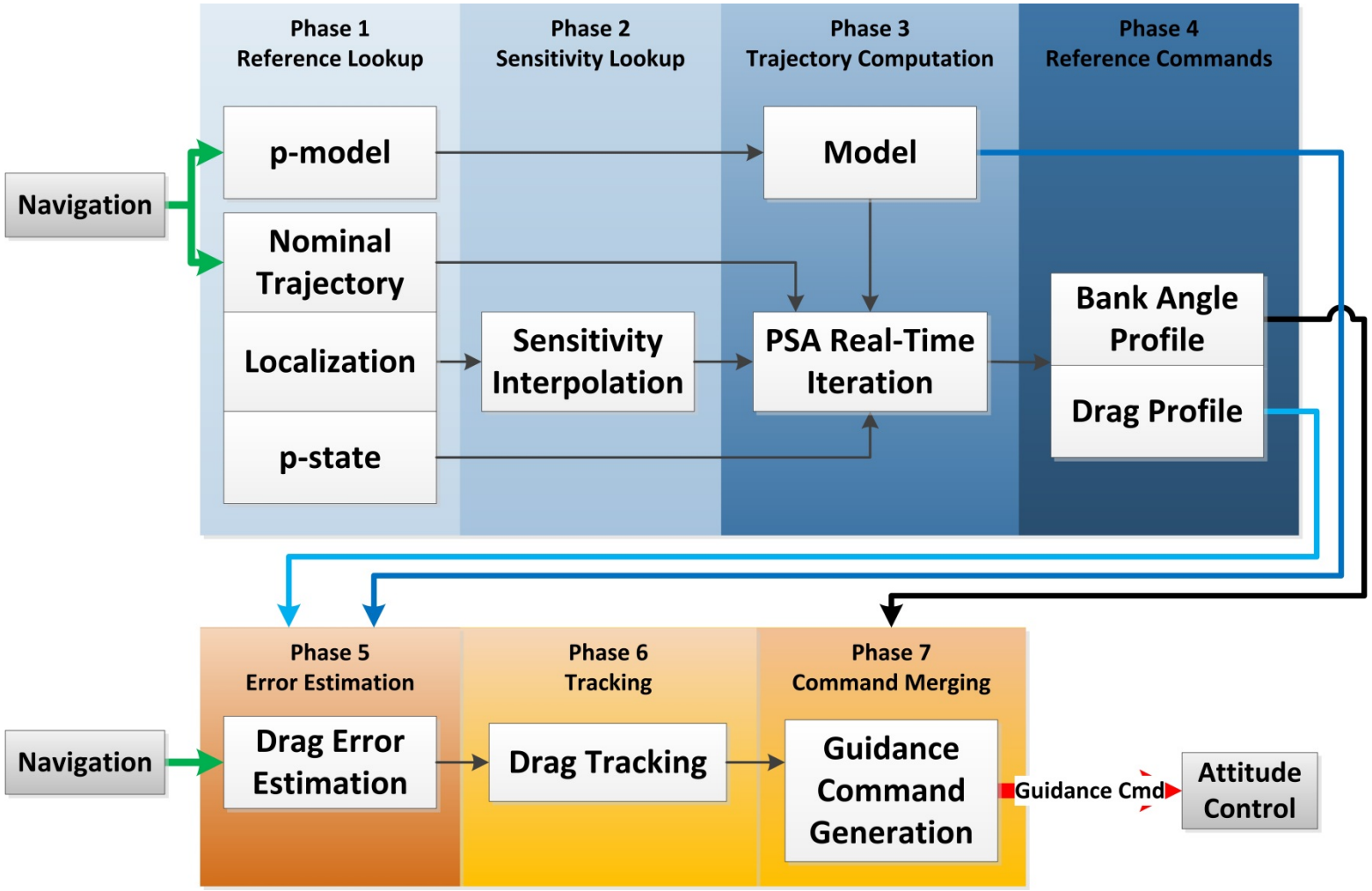
- U shrinks after peak deceleration such that successful trajectory computation cannot be guaranteed

Additional guidance strategy required for “low” velocity flight

Trajectory tracking

- Feedback linearization
- Drag-energy dynamics

Guidance Stages



Feedback Linearization 1

Feedback linearization can be applied to nonlinear systems of the form

$\dot{x} = f(x) + g(x)u$ with state vector $x \in \mathbb{R}^n$, control $u \in \mathbb{R}^p$, output $y \in \mathbb{R}^m$

$y = h(x)$

Goal: Input-Output linearization

Method: Transform $\dot{x}$ into a new system whose states are the output $y \in \mathbb{R}^m$ and its first (*n*-1) 'time' derivatives.

$$
\begin{aligned}
y &= h(x) \\
\dot{y} &= L_f h(x) \\
\ddot{y} &= L_f^2 h(x) \\
&\vdots \\
y^{(n-1)} &= L_f^{n-1} h(x) \\
y^{(n)} &= L_f^n h(x) + L_g L_f^{n-1} h(x) u
\end{aligned}
$$

Taking Lie derivatives of the output

$$L_f h(x) = \frac{\mathrm{d}\, h(x)}{\mathrm{d}\, x} f(x),$$

$$L_g h(x) = \frac{\mathrm{d}\, h(x)}{\mathrm{d}\, x} g(x).$$

Feedback Linearization 2

Derivatives give the transformation T and the transformed system z

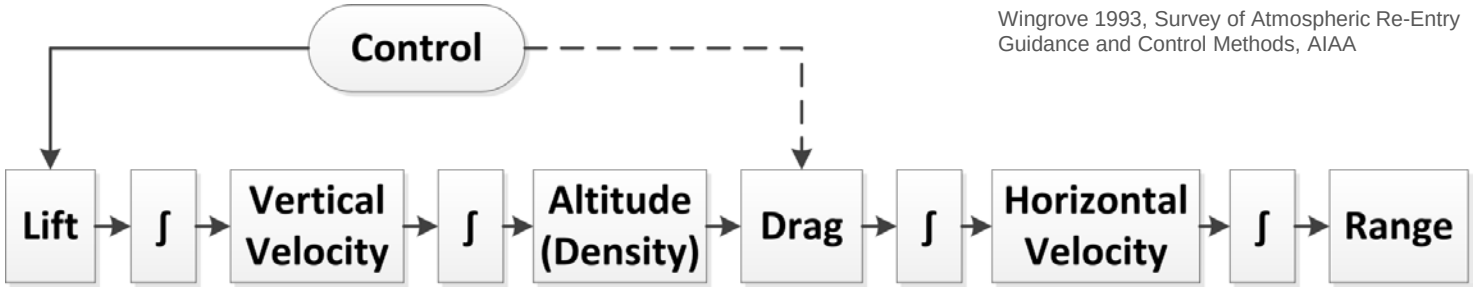
$$z = T(x) = \begin{bmatrix} z_1(x) \\ z_2(x) \\ \vdots \\ z_n(x) \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \\ \vdots \\ y^{(n-1)} \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{bmatrix}$$

$$\begin{aligned} \dot{z}_1 &= L_f h(x) = z_2(x) \\ \dot{z}_2 &= L_f^2 h(x) = z_3(x) \\ &\vdots \\ \dot{z}_n &= L_f^n h(x) + L_g L_f^{n-1} h(x) u \end{aligned} .$$

The feedback law $$u = \frac{1}{L_g L_f^{n-1} h(x)} (-L_f^n h(x) + v)$$

- Creates a linear input-output map from v to $z_1 = y$
- System z is a cascade of n integrators
- Outer-loop control v can be chosen using linear system methods

Downrange Control

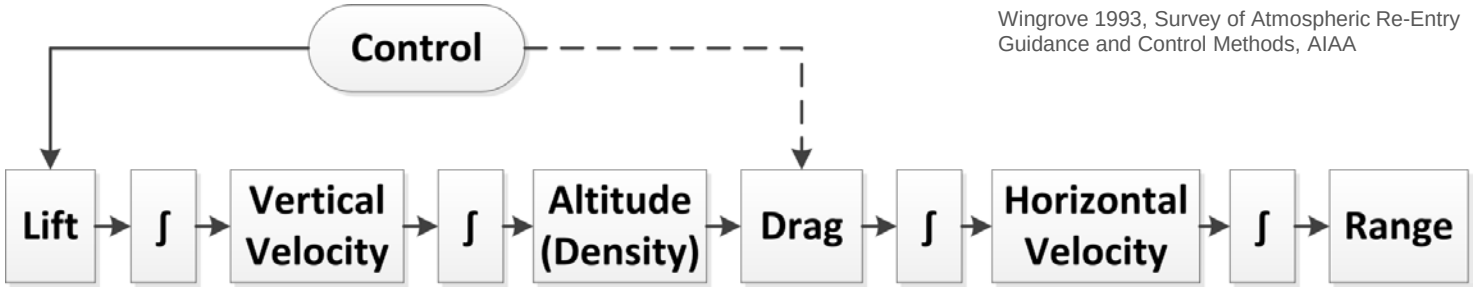


Wingrove 1993, Survey of Atmospheric Re-Entry Guidance and Control Methods, AIAA

Drag is the deciding factor!

- Entry capsule cannot control drag directly
- Control lift via bank angle rotation

Downrange Control



Wingrove 1993, Survey of Atmospheric Re-Entry Guidance and Control Methods, AIAA

Drag is the deciding factor!

- Entry capsule cannot control drag directly
- Control lift via bank angle rotation

$$D[\alpha, r, V_{rel}] = \frac{1}{2} \cdot \rho[r] \cdot V_{rel}^2 \cdot C_D\,[\alpha, M[V, c[r]]] \cdot S$$

Chose drag as system output.

ρ atmospheric density
C_D drag coefficient
α angle of attack
M Mach number
c speed of sound
S aerodynamic reference area

Application of FBL to Entry Guidance

- Controller design in energy domain
- Separation of dynamics: Consider only longitudinal motion in v-r plane!

$$h' = -\frac{V \sin\gamma}{D}$$

$$v' = \frac{D + g\sin\gamma}{Dv}$$

$$\gamma' = -\frac{L}{Dv^2}\cos\boldsymbol{\mu} + \left(\frac{g}{Dv^2} - \frac{1}{h + r_p D}\right)\cos\gamma$$

Assumptions:
- Spherical, nonrotating planet
- No side slip
- Constant angle of attack

Independent variable:

$$E = \frac{1}{2}v^2 - \left(\frac{GM}{r} - \frac{GM}{r_P}\right)$$

Energy as independent variable causes a system order reduction compared to the time domain: A minimal energy domain representation needs only retain either h or v!

Drag-Energy Derivatives

- Take output derivatives and apply feedback linearization mechanism

$$D(r, V_{rel}) = \frac{1}{2} \cdot \rho(r) \cdot V_{rel}^2 \cdot C_D\,(M(V, c(r))) \cdot S$$

$$D' = \frac{1}{2} S v \left(v\, \rho(h)\, \mathrm{Cd}'(M) \frac{dM}{d\mathrm{En}} + \mathrm{Cd}(M) \left(v \frac{dh}{d\mathrm{En}} \rho'(h) + 2\, \rho(h) \frac{dv}{d\mathrm{En}} \right) \right)$$

$$D'' = \frac{1}{2} S \left(v \left(v\, \rho(h)\, \mathrm{Cd}''(M) \left(\frac{dM}{d\mathrm{En}} \right)^2 + \mathrm{Cd}'(M) \left(v\, \rho(h) \frac{d^2 M}{d\mathrm{En}^2} + 2 \frac{dM}{d\mathrm{En}} \left(v \frac{dh}{d\mathrm{En}} \rho'(h) + 2\, \rho(h) \frac{dv}{d\mathrm{En}} \right) \right) \right) + \right.$$
$$\left. \mathrm{Cd}(M) \left(v \left(v \frac{d^2 h}{d\mathrm{En}^2} \rho'(h) + 2\, \rho(h) \frac{d^2 v}{d\mathrm{En}^2} + v \left(\frac{dh}{d\mathrm{En}} \right)^2 \rho''(h) \right) + 4\, v \frac{dh}{d\mathrm{En}} \frac{dv}{d\mathrm{En}} \rho'(h) + 2\, \rho(h) \left(\frac{dv}{d\mathrm{En}} \right)^2 \right) \right)$$

$$D'' = a + bu$$

2nd Drag-Energy Derivative

$$D'' = \boldsymbol{a}(D, \dot{D}, E) + \boldsymbol{b}(D, \dot{D}, E)\boldsymbol{u}(D, \dot{D}, E)$$

$$\begin{aligned}
\boldsymbol{a} = \; & \frac{1}{4\,D^2\,v\,c(h)^4\,(h+\mathrm{rM})}\,S\left(2\,D^2\,v^5\,\rho(h)\,(h+\mathrm{rM})\,c'(h)^2\left(\frac{dh}{d\mathrm{En}}\right)^2\mathrm{Cd}''\left(\frac{v}{c(h)}\right)+4\,D^2\,v^4\,c(h)\,\rho(h)\,(h+\mathrm{rM})\,c'(h)\,\frac{dh}{d\mathrm{En}}\left(c'(h)\,\frac{dh}{d\mathrm{En}}\,\mathrm{Cd}'\left(\frac{v}{c(h)}\right)-\frac{dv}{d\mathrm{En}}\,\mathrm{Cd}''\left(\frac{v}{c(h)}\right)\right)-\right.\\
& 2\,v^3\,c(h)^2\left(\mathrm{Cd}'\left(\frac{v}{c(h)}\right)\left(D^2\,v\,\rho(h)\,(h+\mathrm{rM})\,c''(h)\left(\frac{dh}{d\mathrm{En}}\right)^2+c'(h)\left(\rho(h)\left((h+\mathrm{rM})\left(6\,D^2\,\frac{dh}{d\mathrm{En}}\,\frac{dv}{d\mathrm{En}}+v\sin(y)\,\frac{dD}{d\mathrm{En}}\right)+D\cos^2(y)\left(g\,(h+\mathrm{rM})-v^2\right)\right)+2\,D^2\,v\,(h+\mathrm{rM})\left(\frac{dh}{d\mathrm{En}}\right)^2\rho'(h)\right)\right)-\\
& D^2\,\rho(h)\,(h+\mathrm{rM})\left(\frac{dv}{d\mathrm{En}}\right)^2\mathrm{Cd}''\left(\frac{v}{c(h)}\right)\Bigg)+\\
& v\,c(h)^3\,\mathrm{Cd}'\left(\frac{v}{c(h)}\right)\left(\rho(h)\left(2\,(h+\mathrm{rM})\left(D^2\,\frac{dv}{d\mathrm{En}}\left(4\,v\,\frac{dv}{d\mathrm{En}}-1\right)-g\sin(y)\left(v\,\frac{dD}{d\mathrm{En}}+D\,\frac{dv}{d\mathrm{En}}\right)\right)-2\,D\,g\cos^2(y)\left(g\,(h+\mathrm{rM})-v^2\right)\right)+4\,D^2\,v^2\,(h+\mathrm{rM})\,\frac{dh}{d\mathrm{En}}\,\frac{dv}{d\mathrm{En}}\,\rho'(h)\right)+\\
& c(h)^4\,\mathrm{Cd}\left(\frac{v}{c(h)}\right)\left(v^2\left(2\,\rho'(h)\left((h+\mathrm{rM})\left(4\,D^2\,\frac{dh}{d\mathrm{En}}\,\frac{dv}{d\mathrm{En}}+v\sin(y)\,\frac{dD}{d\mathrm{En}}\right)+D\cos^2(y)\left(g\,(h+\mathrm{rM})-v^2\right)\right)+2\,D^2\,v\,(h+\mathrm{rM})\left(\frac{dh}{d\mathrm{En}}\right)^2\rho''(h)\right)+\\
& \left.4\,\rho(h)\left((h+\mathrm{rM})\left(D^2\,\frac{dv}{d\mathrm{En}}\left(v\,\frac{dv}{d\mathrm{En}}-1\right)-g\sin(y)\left(v\,\frac{dD}{d\mathrm{En}}+D\,\frac{dv}{d\mathrm{En}}\right)\right)+D\,g\cos^2(y)\left(v^2-g\,(h+\mathrm{rM})\right)\right)\right)\Bigg)
\end{aligned}$$

$$\boldsymbol{b} = \frac{S\,\mathrm{uT}\cos(y)\left(v^3\,\rho(h)\,c'(h)\,\mathrm{Cd}'\left(\frac{v}{c(h)}\right)+g\,v\,c(h)\,\rho(h)\,\mathrm{Cd}'\left(\frac{v}{c(h)}\right)+c(h)^2\,\mathrm{Cd}\left(\frac{v}{c(h)}\right)\left(2\,g\,\rho(h)-v^2\,\rho'(h)\right)\right)}{2\,v\,c(h)^2}$$

$\boldsymbol{u} = \dfrac{\boldsymbol{L}}{\boldsymbol{D}}\cos\boldsymbol{\beta}$ State dependent control transformation!

Drag Tracking Law

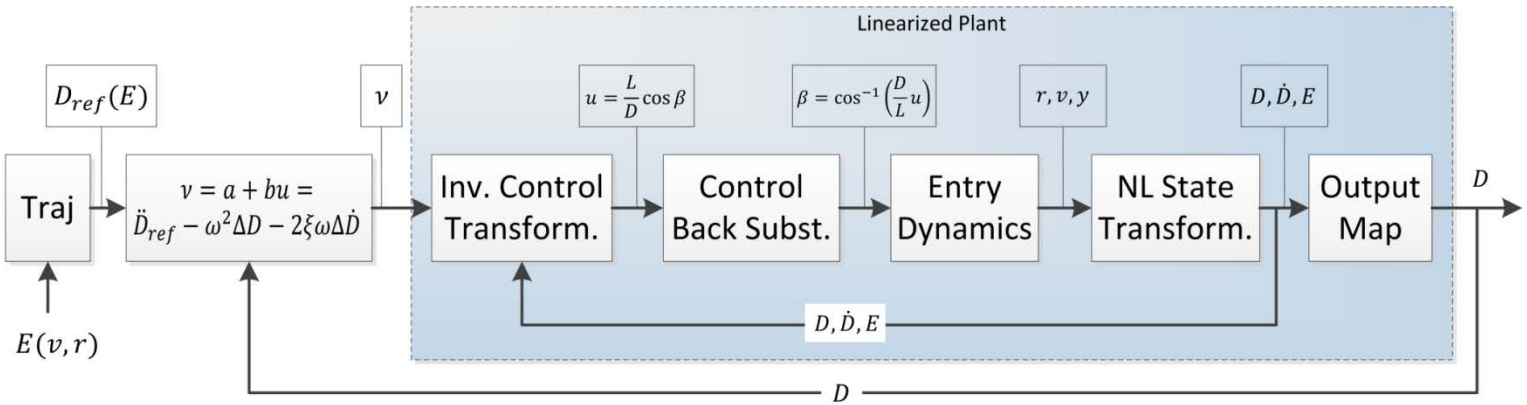
Bijective transformation $T_E: [r, v, y] \rightarrow [D, \dot{D}, E]$ *(Drag – Energy Space)*

System order: 2
Output relative degree: 2 ➔ No internal dynamics!

Linearized plant should have error dynamics of 2nd order linear system.

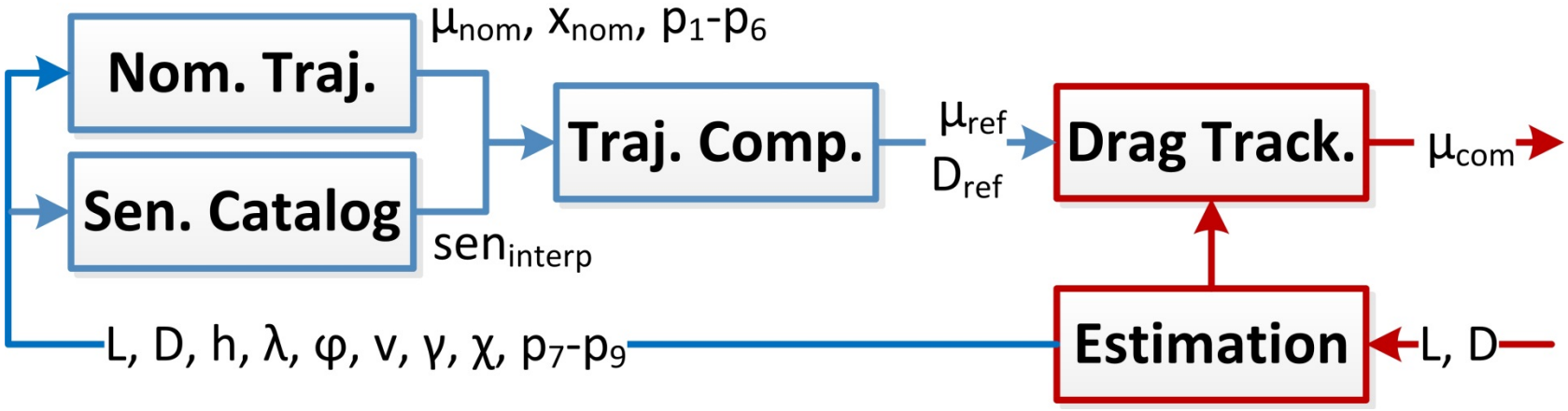
Obtain control law:

$$\mathbf{u} = \frac{1}{b}(-a + \ddot{D}_{ref} - \omega^2 \Delta D - 2\xi\omega\Delta\dot{D})$$



Two-Degree of Freedom Guidance System

Guidance System Overview



- Two-degree-of-freedom design
  - Fast inner tracking loop (20 Hz)
  - Slow outer trajectory loop (0.05 Hz)

- Trajectory computation outputs
  - near optimal discrete u^*, x^* for the entire remaining process
  - optimal bank angle profile μ_{ref} and drag profile D_{ref} obtained from u^*, x^*

- Drag tracking controller based on Mease et. al.

PSA + Drag Tracking

Drag Tracking

✓ Strength

- Fast feedback based directly on physical measurement
- Robust against atmospheric disturbances
- Performant during high-drag flight

o Weakness

- No planning / prediction
- Based on separated and simplified dynamics
- Over-sensitive during low-drag flight

PSA + Drag Tracking

Drag Tracking

✓ Strength

- Fast feedback based directly on physical measurement
- Robust against atmospheric disturbances
- Performant during high-drag flight

o Weakness

- No planning / prediction
- Based on separated and simplified dynamics
- Over-sensitive during low-drag flight

PSA Update

o Weakness

- Strongly reliant on dynamic model and perturbation model
- Not applicable after peak deceleration

✓ Strength

- Near-optimal solution for states and control over entire flight path
- Large initial correction space
- Unified solution for DR and CR

PSA + Drag Tracking

Drag Tracking

✓ Strength

- Fast feedback based directly on physical measurement
- Robust against atmospheric disturbances
- Performant during high-drag flight

o Weakness

- No planning / prediction
- Based on separated and simplified dynamics
- Over-sensitive during low-drag flight

PSA Update

o Weakness

- Strongly reliant on dynamic model and perturbation model
- Not applicable after peak deceleration

✓ Strength

- Near-optimal solution for states and control over entire flight path
- Large initial correction space
- Unified solution for DR and CR

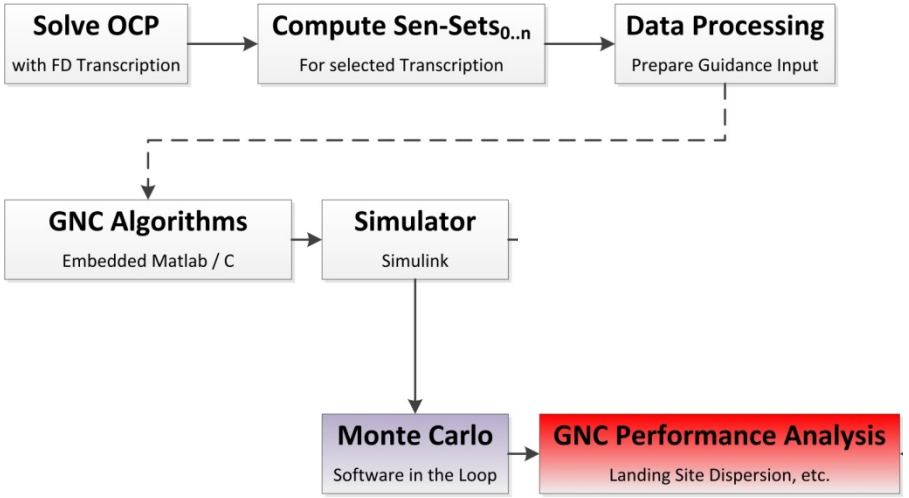
PSA update and drag tracking compliment each other well!

Monte Carlo Campaign

Monte Carlo Campaign

- Large EIP state errors (uniform error distribution)

State	Pert.
h_0	+- 3 km
λ_0	+- 0.3265°
φ_0	+- 0.1632°
v_0	+- 200 m/s
γ_0	+- 1°
χ_0	+- 1°

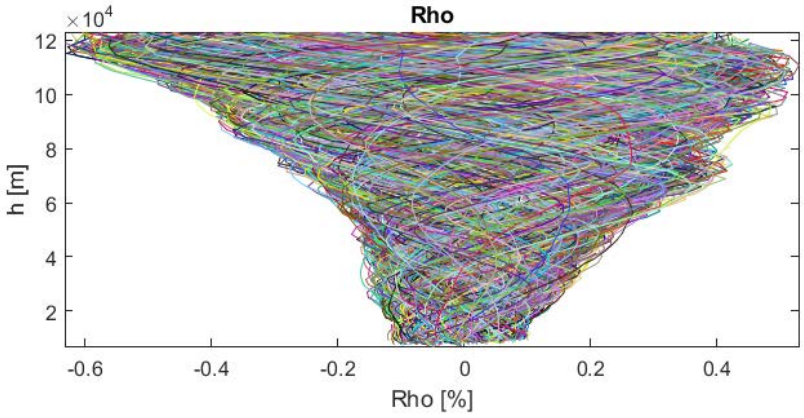


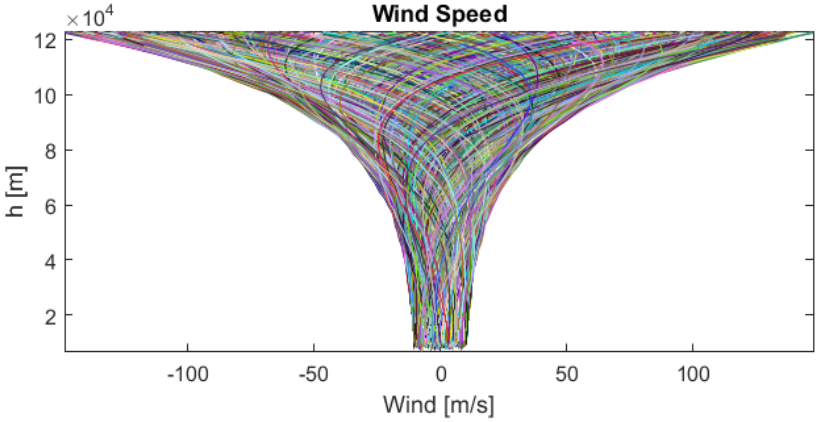
- Guidance input
  - true state, lift and drag falsified with white noise
  - Perturbation parameters are estimated using an extended Kalman filter

Monte Carlo Campaign: Perturbed Environment

- Atmosphere perturbations
  - Random temperature profile between warm and cold conditions
  - Random sinusoidal density perturbations of up to 50% amplitude
- Aerodynamic coefficients perturbed by up to 10%

Param	Pert.
ρ	Temp. +- 50%
c_L	+- 10 %
c_D	+- 10 %
wind	+- 200 m/s
mass	+- 20 kg





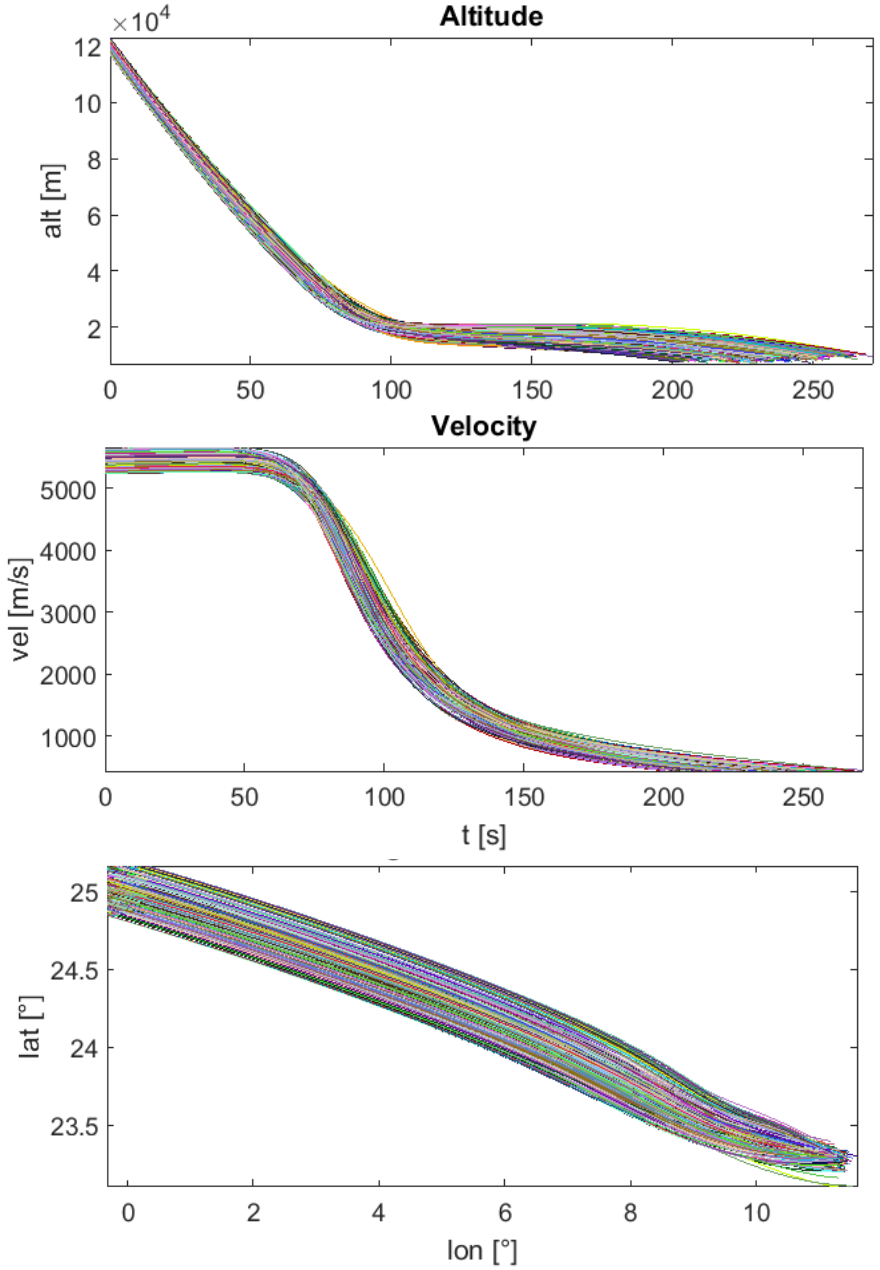
Monte Carlo Campaign: Results

($|\mu| + 3\sigma$) hori. dist.: **12.3 km**

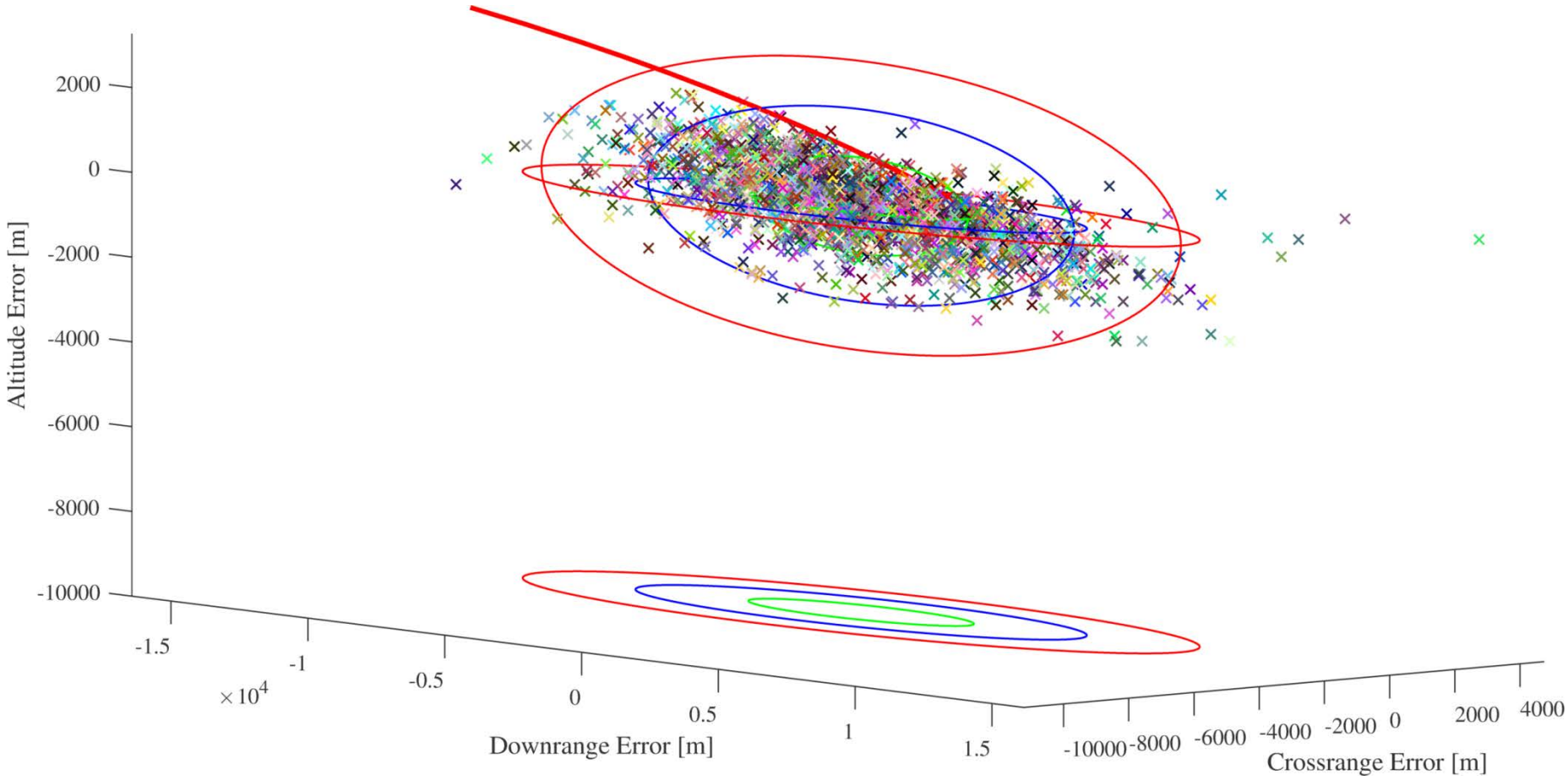
($|\mu| + 3\sigma$) alt. error: **2.4 km**

(3.5 DoF, 2500 MC cases)

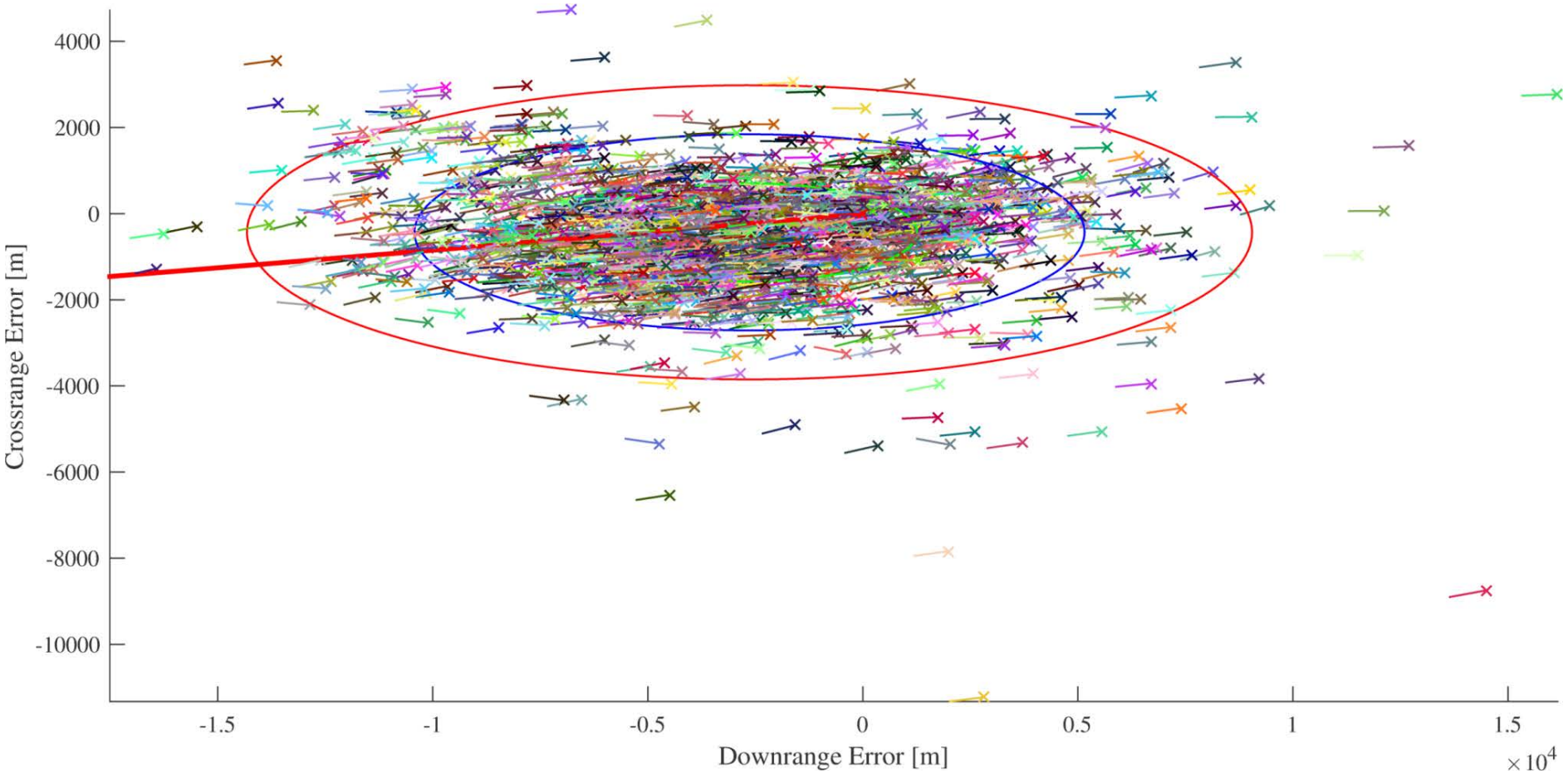
Result	Mean (μ)	Std. Dev. (σ)
Eucl. dist.	4.1 km	2.6 km
Hori. dist.	4 km	2.7 km
DR error	- 2.6 km	3.8 km
CR error	- 0.4 km	1.1 km
Alt. error	- 0.4 km	0.7 km
Vel. error	4 m/s	6 m/s



Parachute Opening Zone



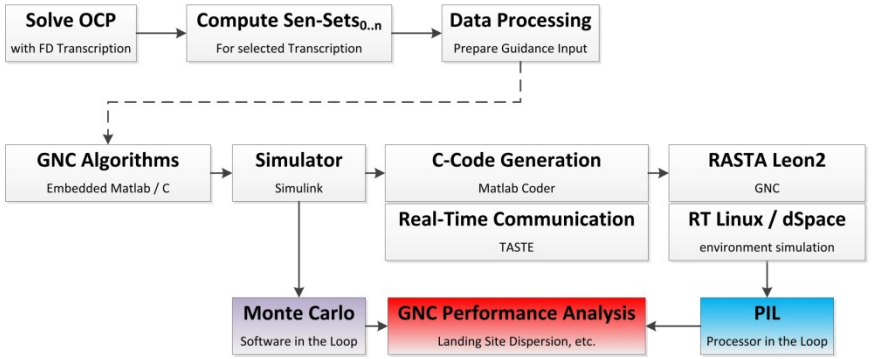
Parachute Opening Zone



LEON2 Processor-in-the-Loop

Processor-in-the-loop

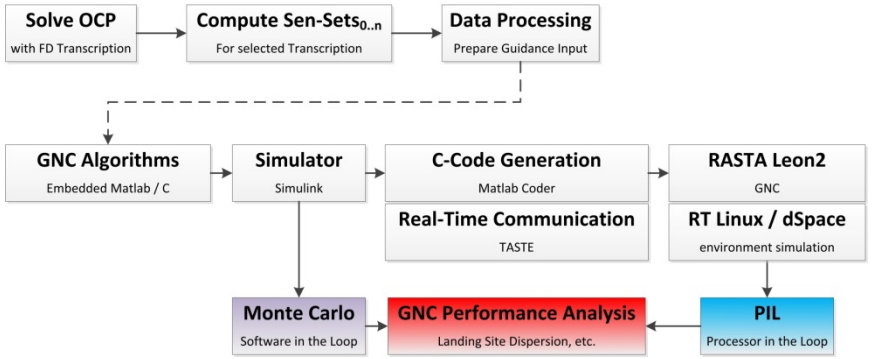
- Test on RASTA-101 with 80 MHz LEON2 processor
- GNC c-code from autocoding from Embedded Matlab
- TASTE toolset: Onboard SW interface definition, communication setup and target compilation
- Problem sizing
  - dense NLP formulation
  - grid length 70
  - ~25 MB sensitivity data



Processor-in-the-loop

- Test on RASTA-101 with 80 MHz LEON2 processor
- GNC c-code from autocoding from Embedded Matlab
- TASTE toolset: Onboard SW interface definition, communication setup and target compilation
- Problem sizing
  - dense NLP formulation
  - grid length 70
  - ~25 MB sensitivity data

Trajectory computation time: **< 1 sec.**



Developed Software Tools

Sensitivity Analysis Framework (SAF)

Software tools to enable or support:

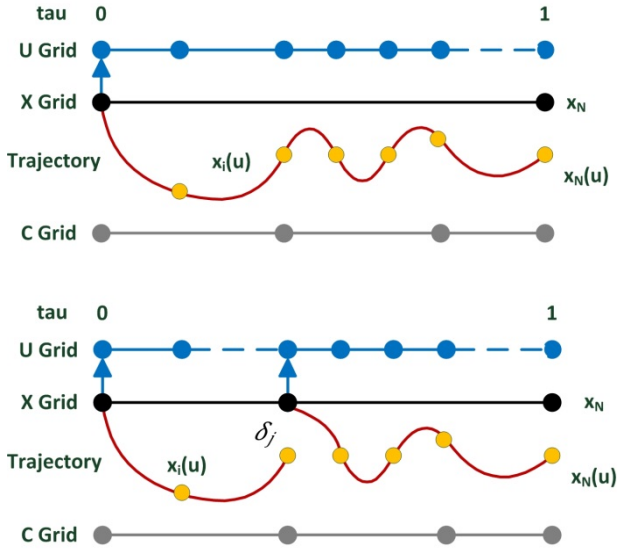
- Transcription of the OCP(p) into a NLP(p)
- Solution of multiphase OCP(p) (WORHP, IPOPT)
- High precision derivative computation (ADOL-C)
- Analysis, visualization and processing of the optimal solution and the sensitivity differentials
- Analysis and test of the real-time iteration scheme

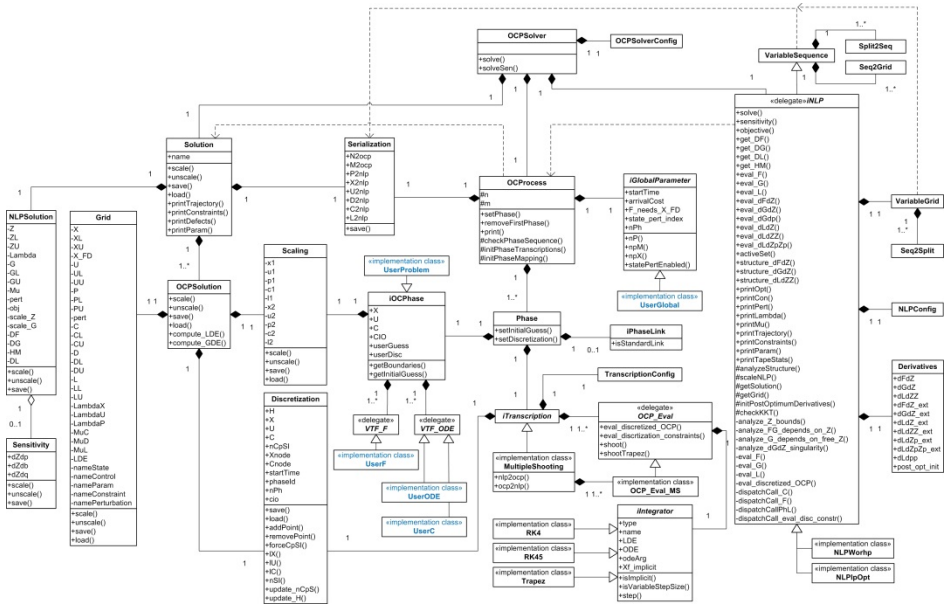
SAF components:

- OCP transcription layer (C++)
- Analysis toolbox (Matlab)
- Mars entry simulator (Embedded Matlab/Simulink)
- Mars guided entry GNC (Embedded Matlab, autocoding capable)

SAF Transcription Layer

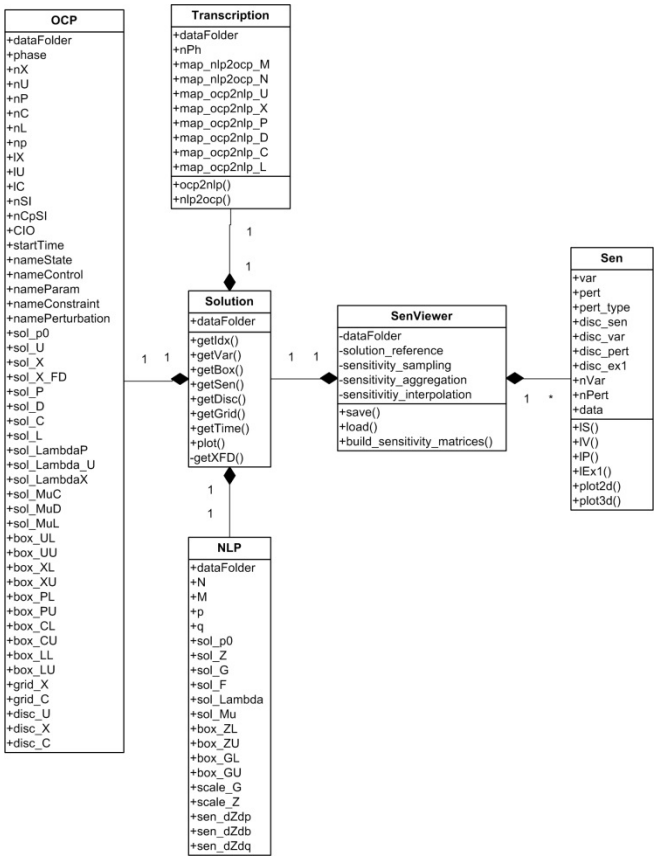
- Supports formulation of multi-phase parametric optimal control problems
- Uses automatic derivative computation with ADOL-C
- Enables parametric sensitivity computation using WORHP
- Generic multiple shooting transcription
- Automatic grid adaption
- Automatic problem scaling
- Generic interface is adaptable to most NLP solvers (interfaces existing for WORHP and IPOPT)

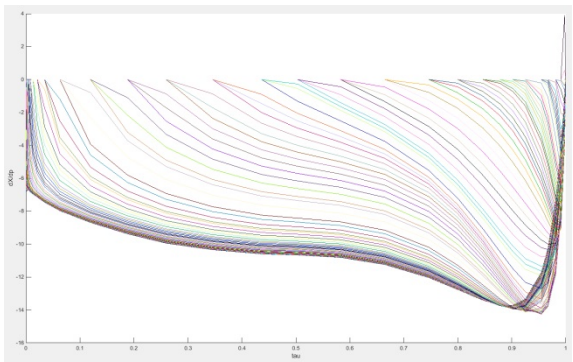


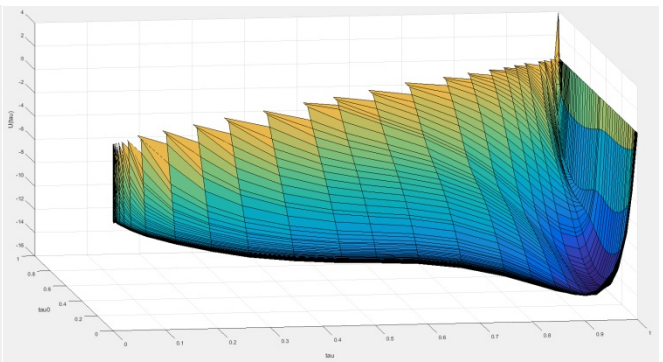


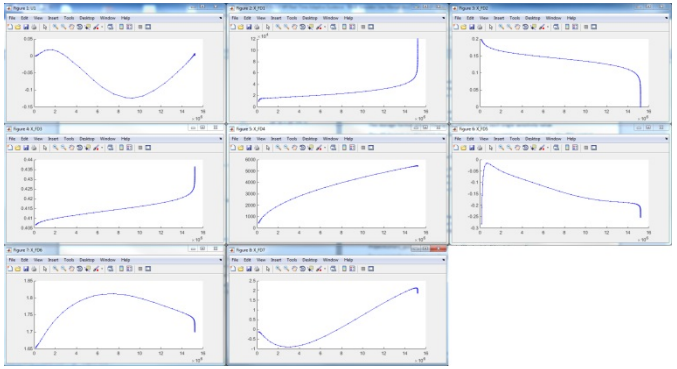
Sensitivity Analysis Toolbox

- Modular, object oriented design
- No commercial software quality, but
- Fairly tested and documented R&D tools for expert users
- User manual and architectural design guide provided







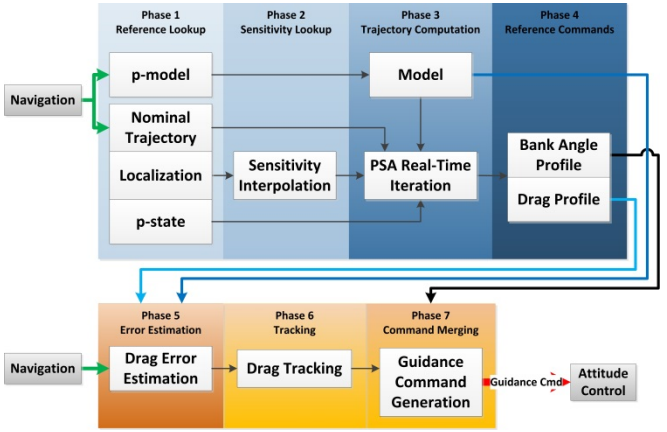


Summary and Outlook

Summary

- Sensitivity analysis of discretized optimal control process
- Parametrization of dynamic model and initial conditions
- Repeated online trajectory computation
  - Parametric sensitivities + interpolation
  - Taylor expansion and iterative constraint correction
- Two degree of freedom guidance system
  - PSA real-time iteration
  - Drag tracking
- Promising results in 3.5 DoF Monte Carlo campaign
- Real-time capability proven by PIL test on LEON2 processor

Tackling the Difficult Problems



Non-convexity and path constraints

- ✓ Offline analysis and trajectory design
- ✓ Reference trajectories and sensitivity catalog
- ✓ Path constr. can be represented in drag domain

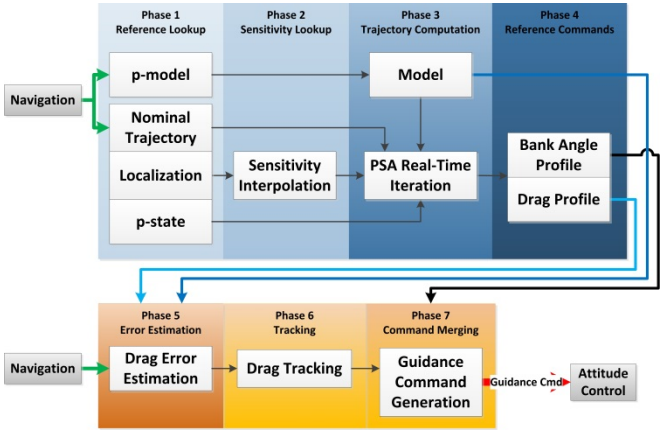
Computational load

- ✓ Cascaded loop structure schedules and staggers expensive tasks
- ✓ PSA: fast online adaption relies on precomputed optimal solution derivative
- ✓ Drag feedback law

Model dependency

- ✓ Disturbance estimation and online model adaption
- ✓ Drag guidance law based directly on physical measurement

Tackling the Difficult Problems



Non-convexity and path constraints

- ✓ Offline analysis and trajectory design
- ✓ Reference trajectories and sensitivity catalog
- ✓ Path constr. can be represented in drag domain

Computational load

- ✓ Cascaded loop structure schedules and staggers expensive tasks
- ✓ PSA: fast online adaption relies on precomputed optimal solution derivative
- ✓ Drag feedback law

Model dependency

- ✓ Disturbance estimation and online model adaption
- ✓ Drag guidance law based directly on physical measurement

What could be done next?

- ❑ Embedded NLP solver is critical enabling technology
- ❑ NLP solver as FPGA?
- ❑ Hybrid: Online optimization and sensitivity computation + PSA update

Thank you for your attention!

david.seelbinder@dlr.de

This research was conducted under No. 4000107257/12 NL/GLC/al of the ESA Networking and Partnering Initiative.