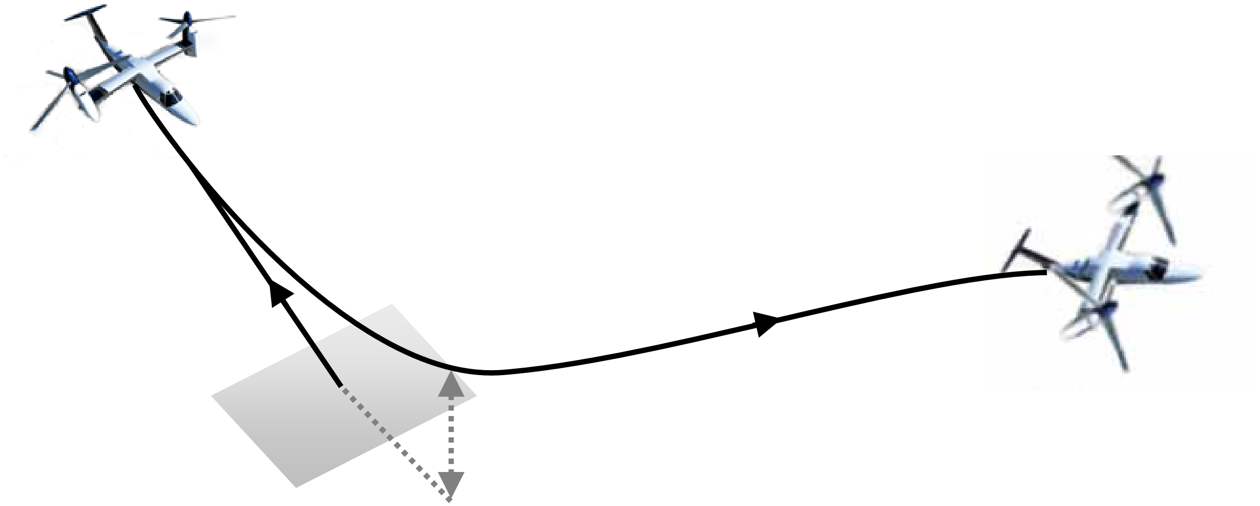
Aero-servo-elastic Multibody Dynamics: Applications in Rotorcraft Flight Mechanics and Wind Turbine Modeling

Carlo L. Bottasso
Politecnico di Milano



University of Zagreb
March 31, 2009

Outline

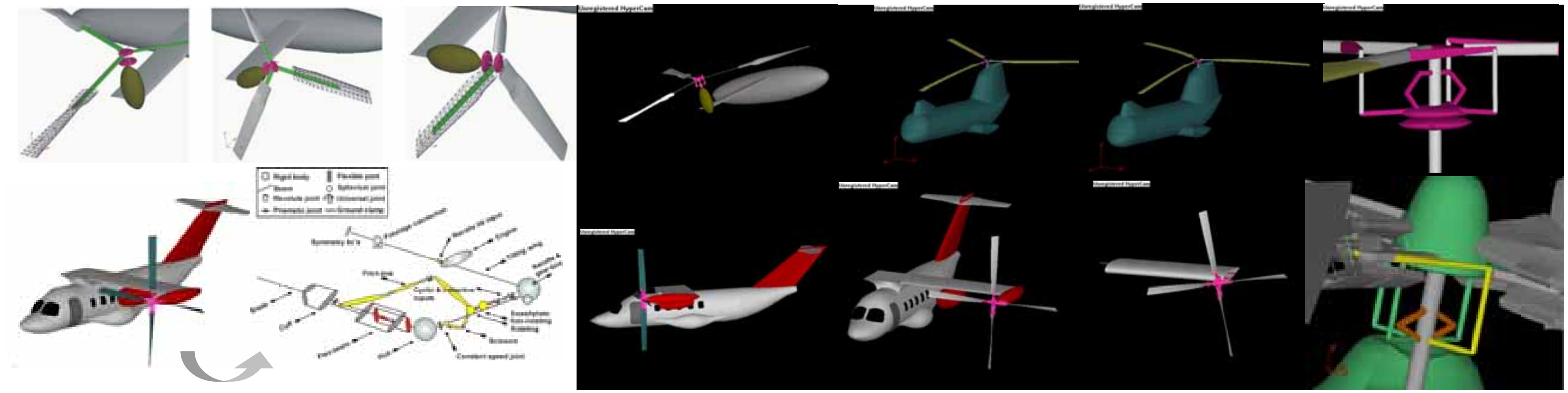
- Multidisciplinary FEM-based multibody modeling
- Applications to complex aero-servo-elastic problems and corollary supporting technologies:
  - Maneuvering multibody dynamics for rotorcraft vehicles in emergency conditions
  - Parameter estimation from flight testing
  - Advanced control laws for wind turbines
- Conclusions and outlook

Overview of a Comprehensive Multibody Aero-servo-elastic Code

Aero-servo-elastic Models

FEM multibody code, extensively validated for rotorcraft applications:

(Bauchau, Bottasso, Nikishkov, *MCM* 2001)



Wind-energy version:

CpLambda (Code for Performance, Loads and Aeroelasticity by Multi-Body Dynamic Analysis)

Aero-servo-elastic Models

Classical modeling philosophy:

Ad-hoc codes developed in-house by manufacturers, tailored to specific configurations (e.g. for wind turbines: horizontal axis, three bladed, etc.)

FEM multibody approach:

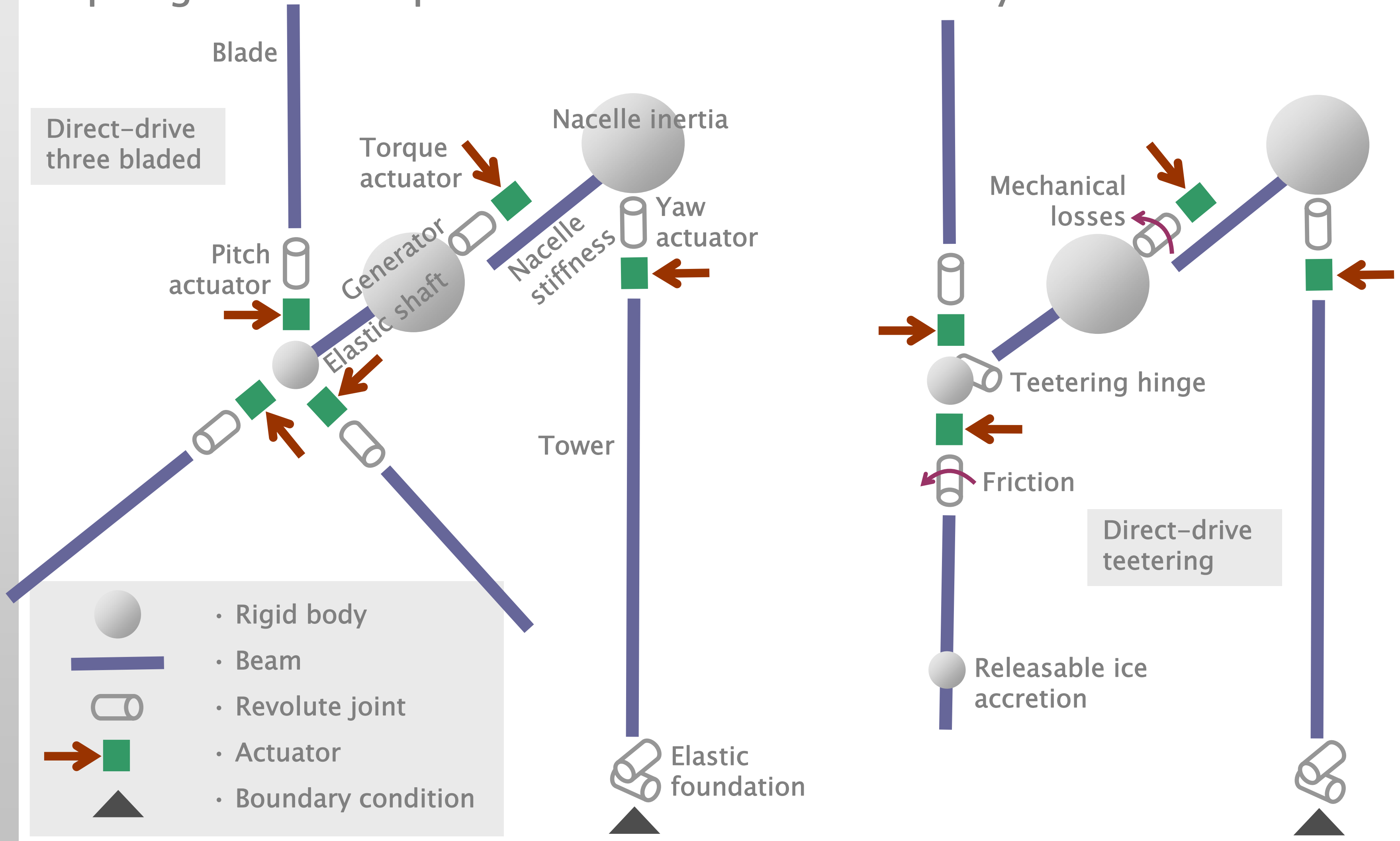
- System is viewed as a complex flexible mechanism
- Model novel configurations of arbitrary topology by assembling basic components chosen from an extensive library of elements

Advantages:

- Simulation software tools are modular and expandable
- Applicable to configurations with arbitrary topologies, including those not yet foreseen

Example: Wind Turbine Models

Topological view of possible wind turbine multibody models:



Direct-drive three bladed

Blade

Torque actuator

Nacelle inertia

Pitch actuator

Generator

Elastic shaft

Nacelle stiffness

Yaw actuator

Tower

Elastic foundation

Mechanical losses

Teetering hinge

Friction

Direct-drive teetering

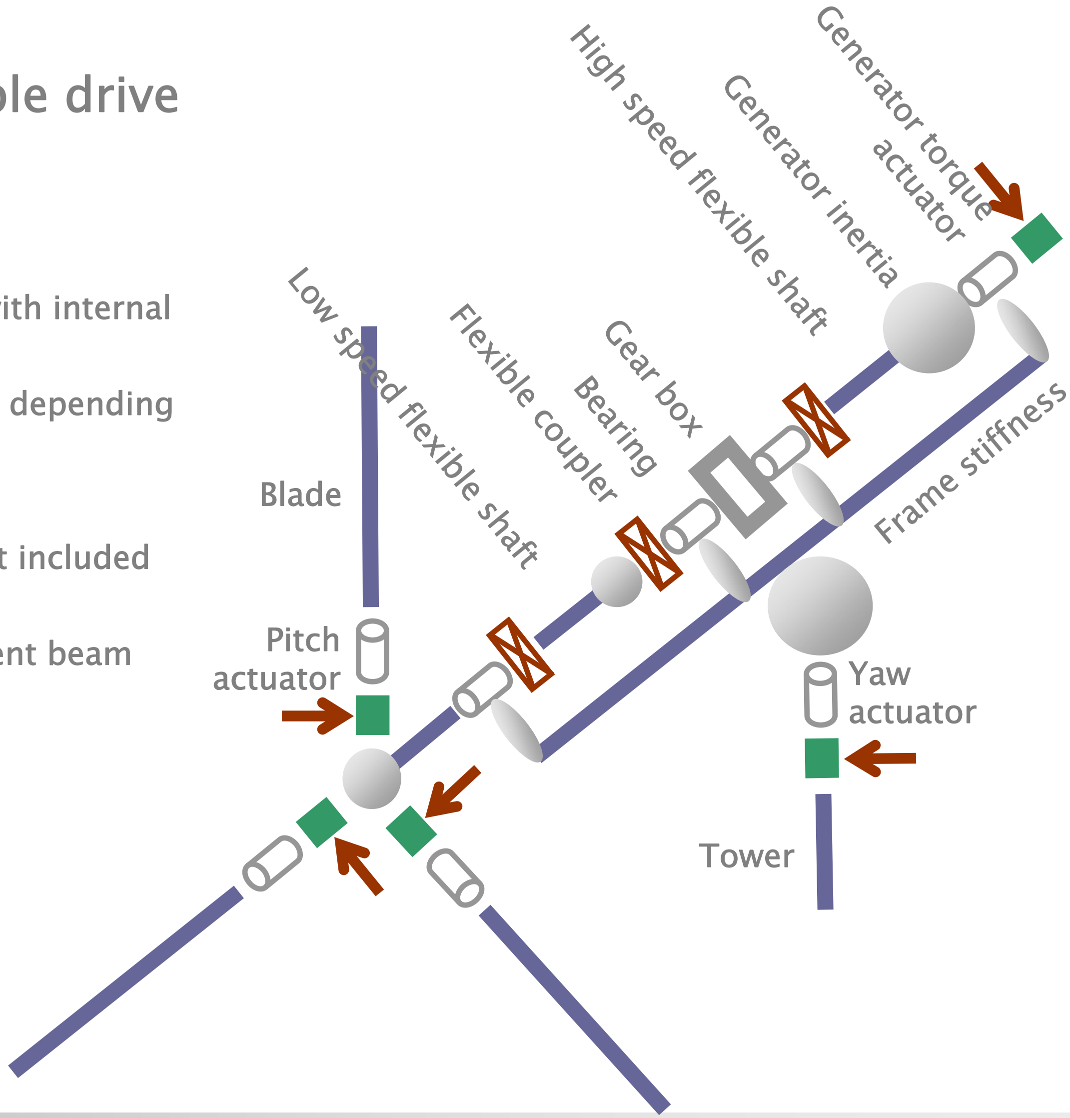
Releasable ice accretion

- Rigid body
- Beam
- Revolute joint
- Actuator
- Boundary condition

Example: Wind Turbine Models

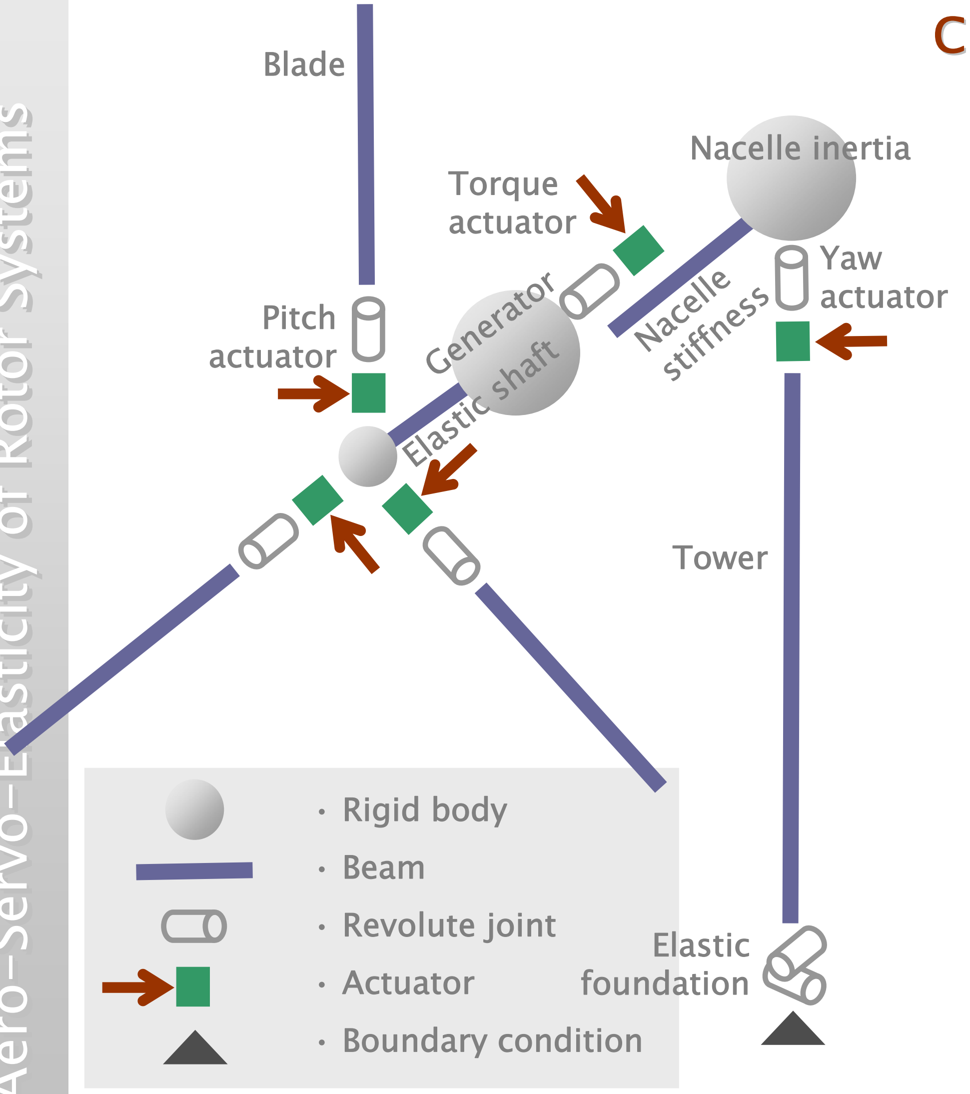
Topological view of possible drive train multibody model:

- Couplers modeled as flexible joints with internal equivalent springs and dampers
- Bearings with internal friction models depending on internal reaction components
- Shafts modeled with beam elements
- Rigid bodies to account for inertia not included in beam elements
- Nacelle frame modeled as an equivalent beam



- ● Rigid body
- ▬ Beam
- ⊂⊃ Revolute joint
- ☒ Flexible joint
- → 🟩 Actuator

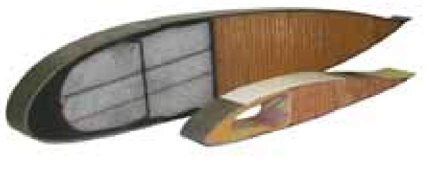
Example: Wind Turbine Models



CpLambda structural element library:

▶ **Beams**:

- Geometrically exact, composite-ready beams
- Curved and twisted NURBS reference lines
- Fully populated 6x6 stiffness (aeroelastic couplings)



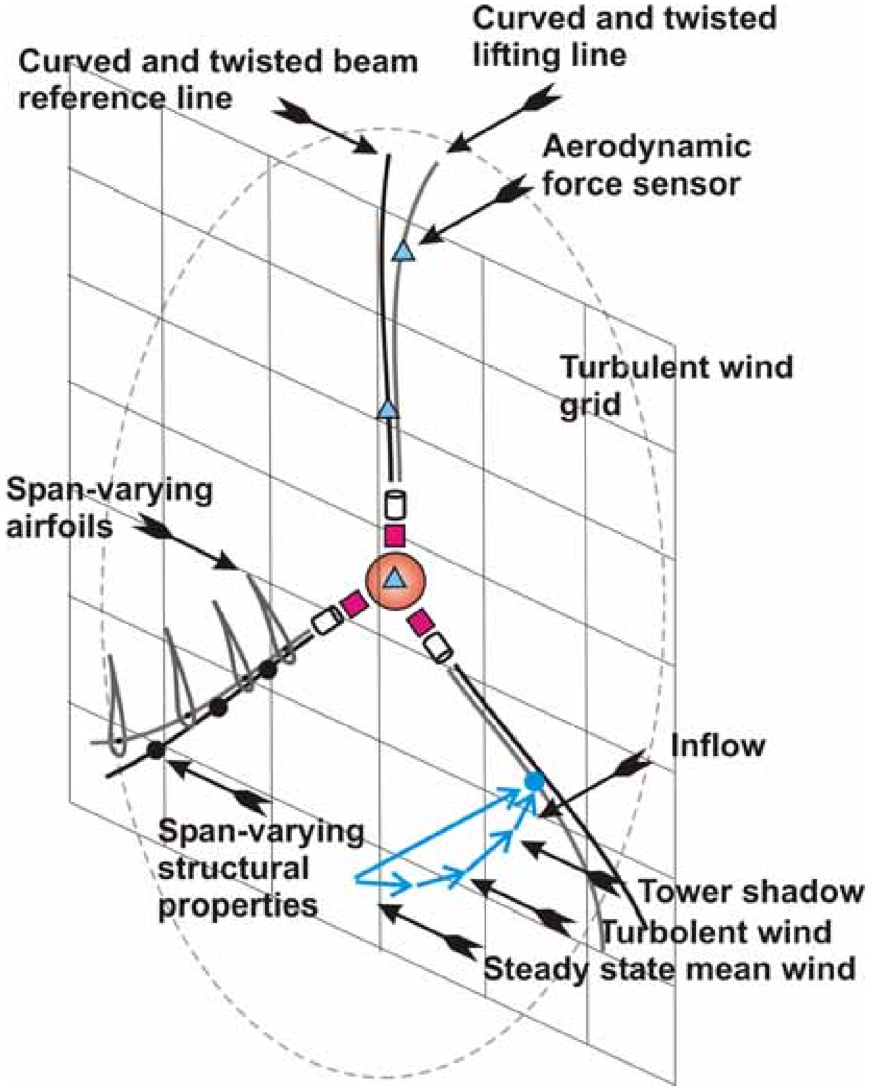
▶ **Joints**:

- Enforced by Lagrange multipliers (DAE formulation)
- Spring, damper, backlash and friction in all joints
- Flexible joints (contact beam-cylindrical, prismatic, screw)
- Unilateral joints (contact-impact analysis)

▶ **Actuators**: first and second order linear and rotational models, refined actuator models

▶ **Sensors** and **control elements**

Example: Wind Turbine Models



▶ **Aerodynamic model**:

- Lifting lines (two-dimensional strip theory)
- Tip losses, radial & unsteady flow, dynamic stall
- Inflow models (Dynamic Pitt-Peters & Peters-He)
- Generic interface to external CFD or free wake
- Tower shadow
- **Wind models** (according to IEC 61400-1):
  - Deterministic gusts (EOG1, ECG)
  - 3D stochastic turbolent wind
  - Wind shear (exponential and logarithmic)
- **Hydrodynamic models** (according to Morrison):
  - Fatigue loads: irregular waves based on linear Airy theory, with JONSWAP wave spectrum
  - Extreme loads: regular wave train computed using stream function theory

▶ **Analysis types**:

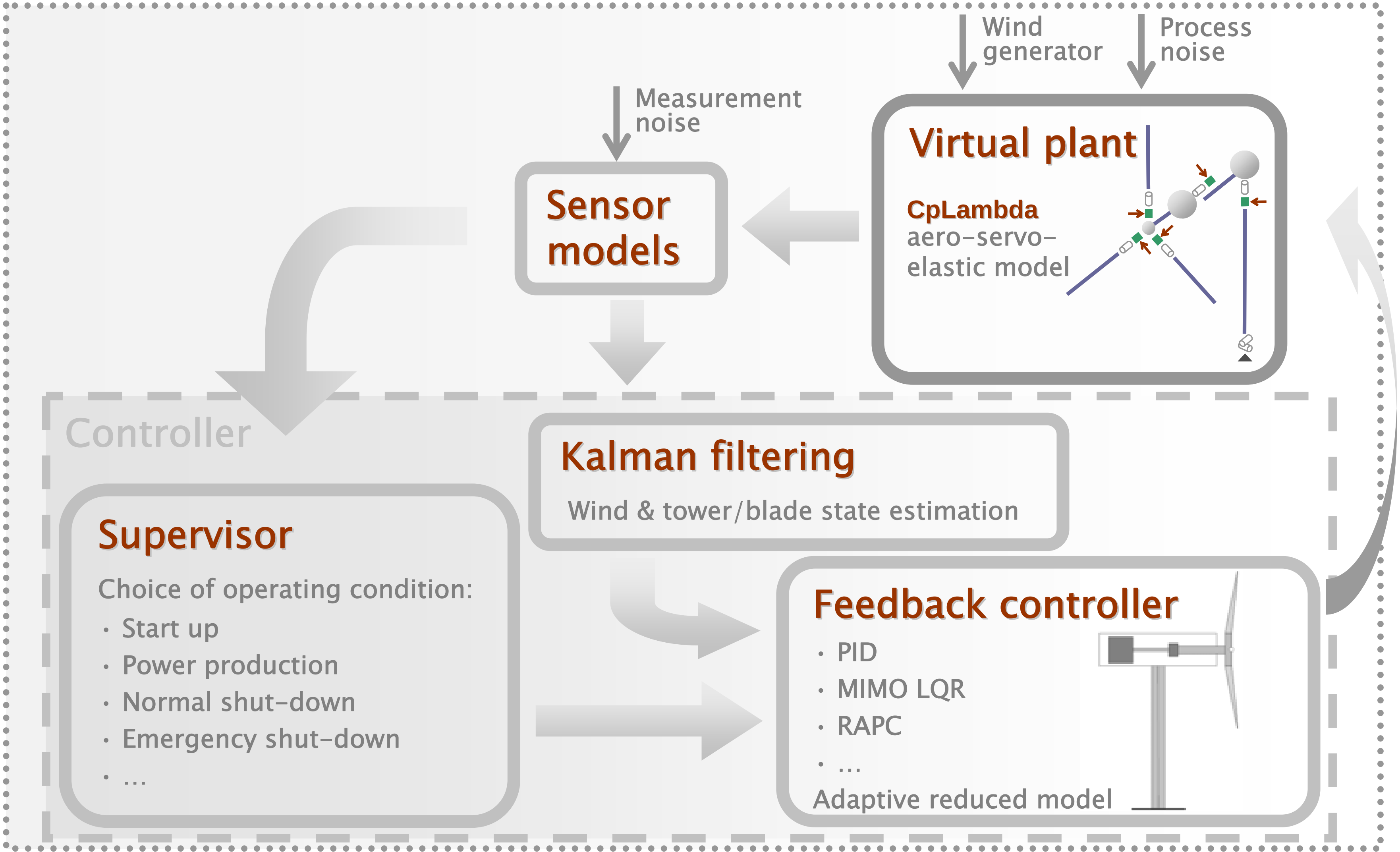
- Static analysis
- Eigenanalysis
- Dynamic response analysis
- Stability analysis (implicit Floquet or by excitation)

Virtual Testing Environment

Simulation of wind turbine operations in a high fidelity environment:

- Compute extreme loads due to gusts
- Evaluate fatigue damage due to turbulence
- Evaluate response spectra
- Judge performance and suitability of control laws
- Simulate failures and off-design conditions
- Etc.

Virtual Testing Environment



Wind generator

Process noise

Measurement noise

Virtual plant

CpLambda
aero-servo-elastic model

Sensor models

Controller

Kalman filtering

Wind & tower/blade state estimation

Supervisor

Choice of operating condition:

- Start up
- Power production
- Normal shut-down
- Emergency shut-down
- …

Feedback controller

- PID
- MIMO LQR
- RAPC
- …

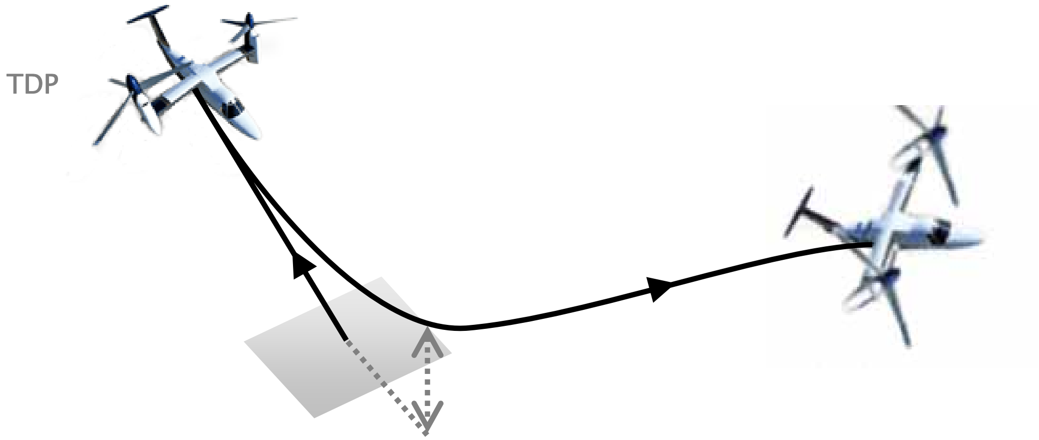
Adaptive reduced model

Applications and Corollary Technologies I: Maneuvering Multibody Dynamics

Computing Maneuvers

Goal: modeling of critical maneuvers of helicopters and tilt-rotors at the **boundaries of the flight envelope**

Examples: Cat-A certification (Continued TO, Rejected TO), ADS-33, autorotation, tail rotor loss, mountain rescue operations, etc.



Applicability:
- Vehicle design
- Design of procedures, flight trials, certification

Related work: *Okuno & Kawachi 1993*, *Carlson & Zhao 2002*, *Bottasso et al. 2004*

Maneuvers as Optimal Control Problems

Tools:

- Mathematical models of maneuvers
- Mathematical models of vehicle
- Numerical solution strategy

Maneuvers are here defined as **optimal control problems**, whose ingredients are:

- A **cost function** (index of performance)
- **Constraints**:
  - Vehicle equations of motion
  - Physical limitations (limited control authority, flight envelope boundaries, etc.);
  - Procedural limitations

Solution yields: **trajectory** and **controls** that fly the vehicle along it

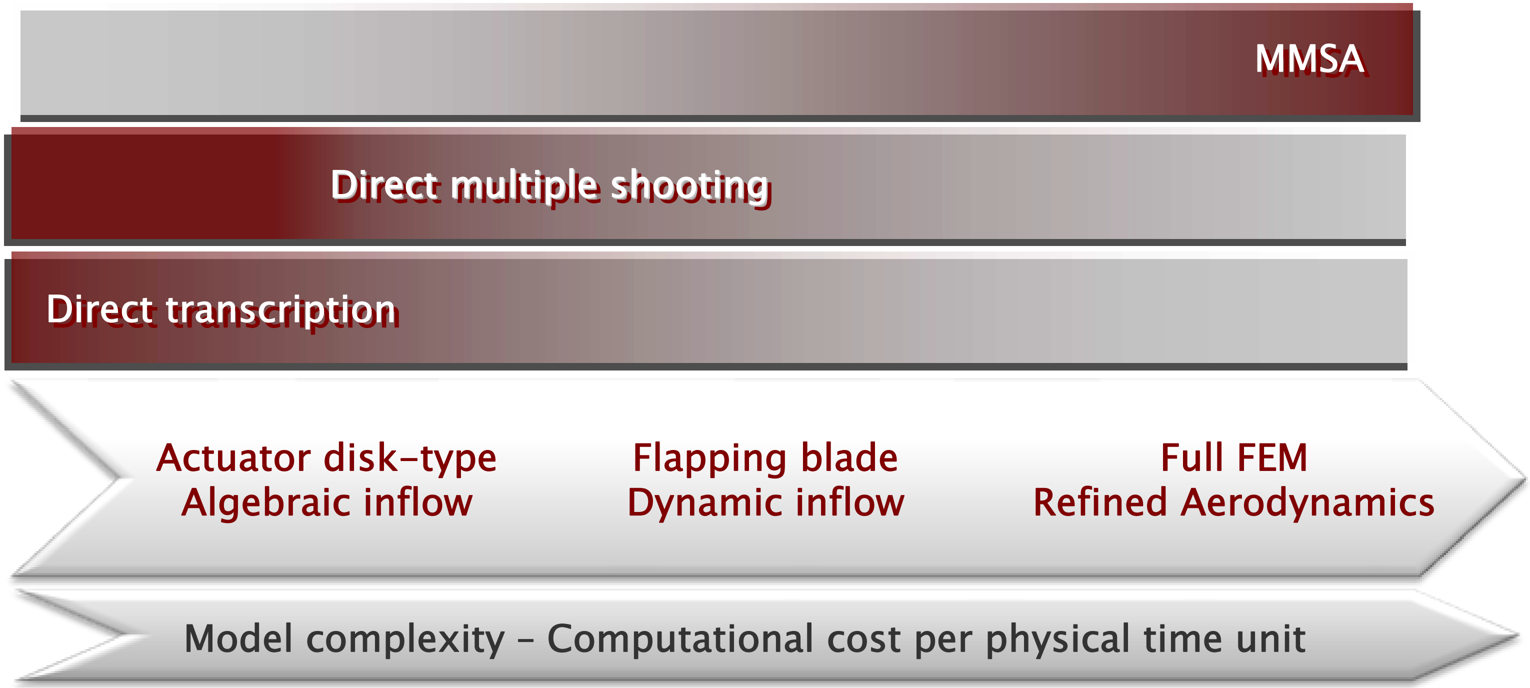
Vehicle Models

- **Flight mechanics** helicopter and tilt-rotor models

- **Comprehensive aeroelastic** multibody-based models (Bottasso et al. 2005-2008)

ADS-33 sidestep & Category A CTO - multibody model (full-FEM flexible main and tail rotors, main rotor control linkages, Peters-He aerodynamics):

Preferred Methods for Vehicle Models of Increasing Complexity



MMSA

Direct multiple shooting

Direct transcription

Actuator disk-type
Algebraic inflow

Flapping blade
Dynamic inflow

Full FEM
Refined Aerodynamics

Model complexity – Computational cost per physical time unit

(Bottasso 2008, Bottasso et al. 2009)

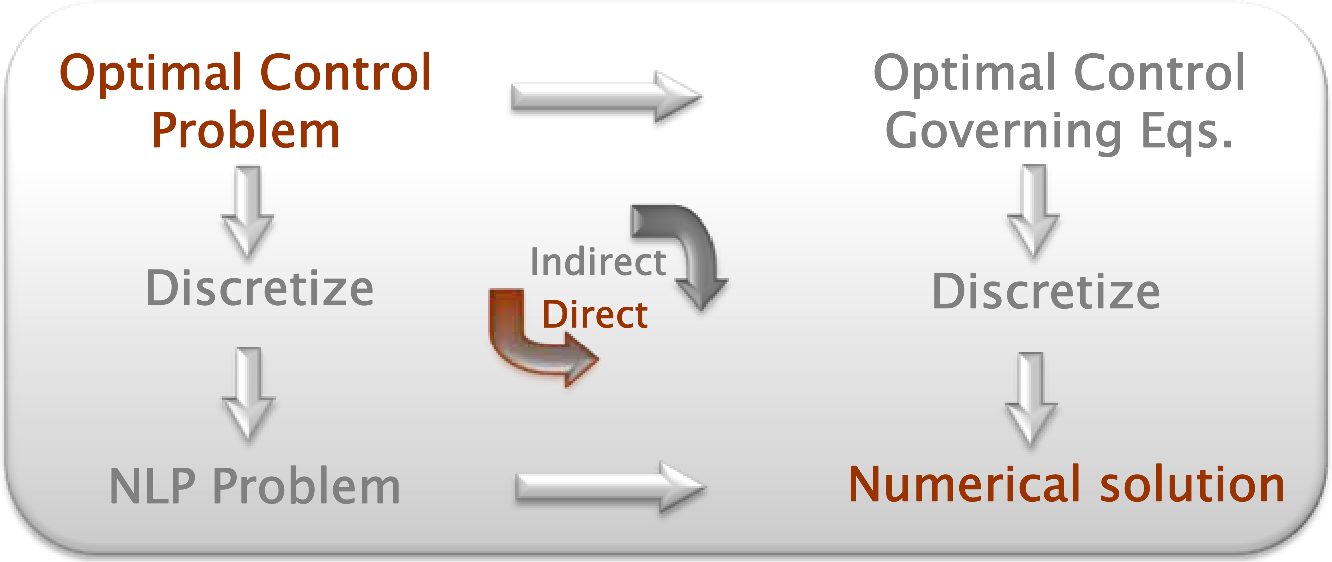
Trajectory Optimization

Maneuver Optimal Control Problem (MOCP):

- Cost function $J = \phi(\boldsymbol{y}, \boldsymbol{u}, t)\big|_\Gamma + \int_\Omega L(\boldsymbol{y}, \boldsymbol{u}, t)\,\mathrm{d}t$
- Vehicle model $\dot{\boldsymbol{y}} - \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u}, t) = 0$
- Boundary conditions $\boldsymbol{\psi}(\boldsymbol{y}(T_0)) \in [\boldsymbol{\psi}_{0_\mathrm{min}}, \boldsymbol{\psi}_{0_\mathrm{max}}]$ (initial)
  $\boldsymbol{\psi}(\boldsymbol{y}(T)) \in [\boldsymbol{\psi}_{T_\mathrm{min}}, \boldsymbol{\psi}_{T_\mathrm{max}}]$ (final)
- Constraints
  point: $\boldsymbol{g}(\boldsymbol{y}, \boldsymbol{u}, t) \in [\boldsymbol{g}_\mathrm{min}, \boldsymbol{g}_\mathrm{max}]$ integral: $\int_\Omega \boldsymbol{h}(\boldsymbol{y}, \boldsymbol{u}, t)\,\mathrm{d}t \in [\boldsymbol{h}_\mathrm{min}, \boldsymbol{h}_\mathrm{max}]$
- Bounds $\boldsymbol{y} \in [\boldsymbol{y}_\mathrm{min}, \boldsymbol{y}_\mathrm{max}]$ (state bounds)
  $\boldsymbol{u} \in [\boldsymbol{u}_\mathrm{min}, \boldsymbol{u}_\mathrm{max}]$ (control bounds)

Remark: cost function, constraints and bounds collectively define in a compact and mathematically clear way a maneuver

Numerical Solution of Maneuver Optimal Control Problems



Indirect approach:

- Need to derive optimal control governing equations (impossible for third-party black-box vehicle models)
- Need to provide initial guesses for co-states
- For state inequality constraints, need to define a priori constrained and unconstrained sub-arcs

Direct approach: all above drawbacks are avoided

TOP: Trajectory Optimization Program for Rotorcraft Vehicles

Supported vehicle models:

- External model:
  - Full-FEM multibody models (Cp-Lambda (PoliMI), Dymore (GaTech))
  - FLIGHTLAB©, Europa or other black-box initial value solvers
- Internal model, based on:
  - Blade element and inflow theory (Prouty, Peters)
  - Quasi-steady flapping dynamics, aerodynamic damping correction
  - Look-up tables for aerodynamic coefficients of lifting surfaces
  - Compressibility effect and downwash at tail due to main rotor

Implemented direct solution strategies:

- Direct transcription
- Direct multiple shooting

Direct Transcription

- **Transcribe** equations of dynamic equilibrium using suitable time marching scheme:

$$\dot{\boldsymbol{y}} - \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{u}, t) = 0 \quad \Rightarrow \quad \boldsymbol{\xi}_h(\boldsymbol{y}_h, \boldsymbol{u}_h, T) = 0$$

Time finite element method *(Bottasso 1997)*:

$$\int_K (\boldsymbol{y}_h \cdot \dot{\boldsymbol{w}}_h + \boldsymbol{f}(\boldsymbol{y}_h, \boldsymbol{u}_h, t) \cdot \boldsymbol{w}_h)\,\mathrm{d}t = \boldsymbol{\lambda}_h \cdot \boldsymbol{w}_h|_{\partial K}, \quad \forall \boldsymbol{w}_h \in \boldsymbol{W}_h(K)$$

- **Discretize cost function** and **constraints**
- Solve resulting **NLP problem** using a SQP or IP method:

$$\min_{\boldsymbol{y}_h, \boldsymbol{u}_h, T} J_h(\boldsymbol{y}_h, \boldsymbol{u}_h, T)$$

$$\text{s.t.: } \boldsymbol{\phi}_h(\boldsymbol{y}_h, \boldsymbol{u}_h, T) \in [\boldsymbol{\phi}_{\min}, \boldsymbol{\phi}_{\max}]$$

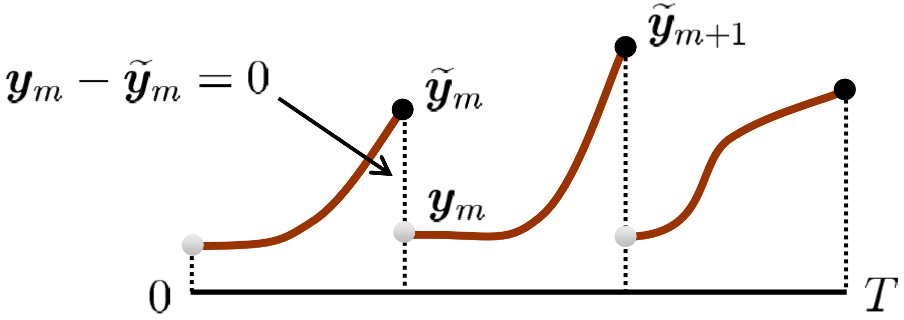
Problem is **large** but highly **sparse** and **banded**

Direct Transcription

Remarks:

- Rigorous and trivial treatment of state and control constraints
- Optimality of NLP problem converges to optimality of OC problem as grid id refined
- Two-point boundary value solver: unstable solution modes do not lead to catastrophic failures as with shooting (ideal for unstable rotorcraft problems)
- Models with very fast solution components need very small time steps: very large problems, excessive computational cost (size of NLP dictated by time step)
- Typically **best method**, but applicable only to models of low-moderate complexity with slow solution components

Direct Multiple Shooting



- Partition time domain into shooting segments: $0 = t_1 < t_2 < \ldots < t_M = T$
- Discretize controls as: $\boldsymbol{u}^m = \sum_{i=1}^{n_m} h_i(\xi)\boldsymbol{u}_i^m$
- Advance solution from $\boldsymbol{y}_m$ to $\tilde{\boldsymbol{y}}_{m+1}$ using N^m time steps.
- Glue segments together with NLP constraints:

$$\boldsymbol{y}_m - \tilde{\boldsymbol{y}}_m = 0 \quad m = 2, \ldots, M-1$$

- **NLP unknowns**:

$$\boldsymbol{x} = (\boldsymbol{y}_{m=(1,M)}; \boldsymbol{u}_{n=(1,n_m)}^{m=(1,M-1)}; T)$$

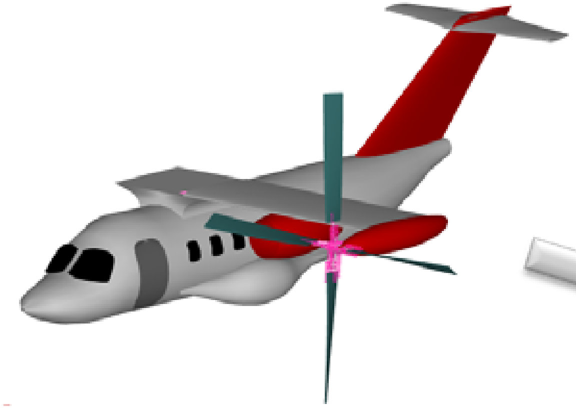
Direct Multiple Shooting

Remarks:

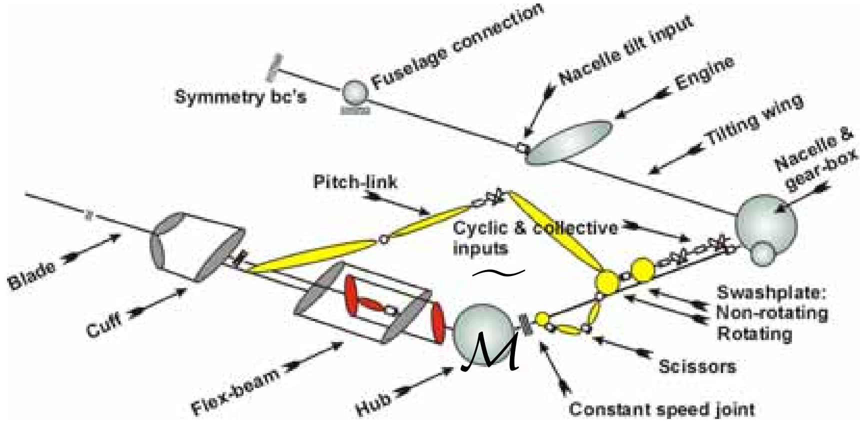
- Can handle models with fast solution components (size of NLP unrelated to time step)
- Need special techniques to handle state constraints within shooting segments
- For state constrained problems, it does not approximate the optimal control problem when the grid is refined (no state constraints within shooting arcs)
- Need care when dealing with unstable problems: multiple segments necessary for curing (alleviating) instability of single shooting

New feature: hybrid single/multiple shooting strategy (single for slow states, multiple for fast ones)

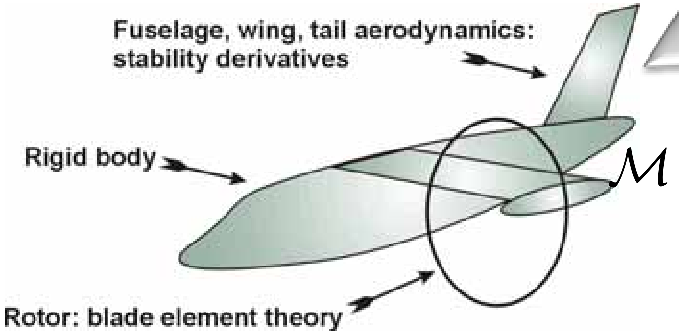
Reduced Models



Comprehensive model: many dofs, captures fine-scale solution details



Model Reduction

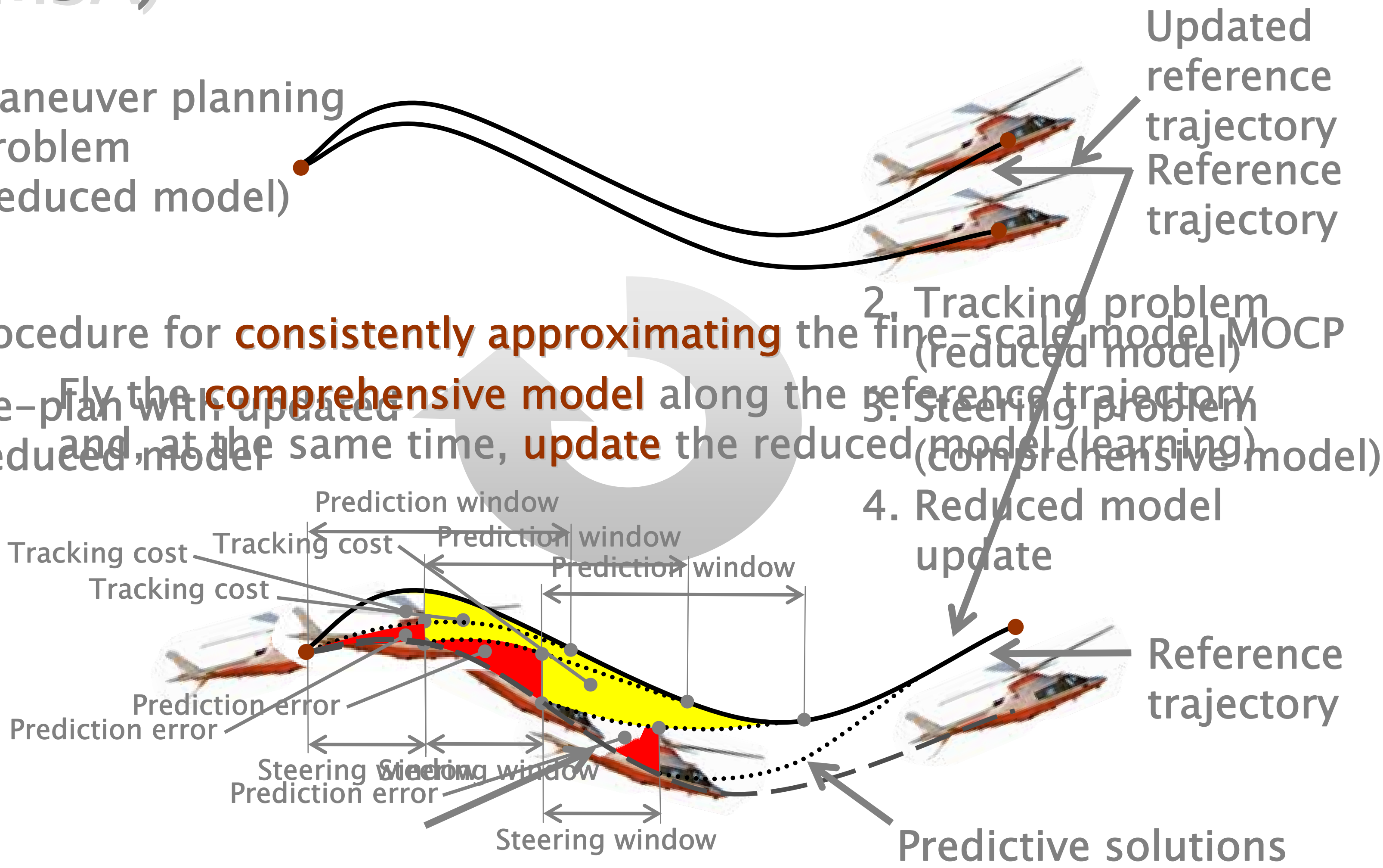


Reduced model: few dofs, captures output response

The Multi-Model Steering Algorithm (MMSA)

A procedure for **consistently approximating** the fine-scale model MOCP

Fly the **comprehensive model** along the reference trajectory and, at the same time, **update** the reduced model (learning)



Applications: ADS-33 MTEs

Mission Task Elements (MTE): assessment of ability to perform critical tasks

All MTEs can be formulated as **constrained Optimal Control problems**

Example: lateral translation MTE

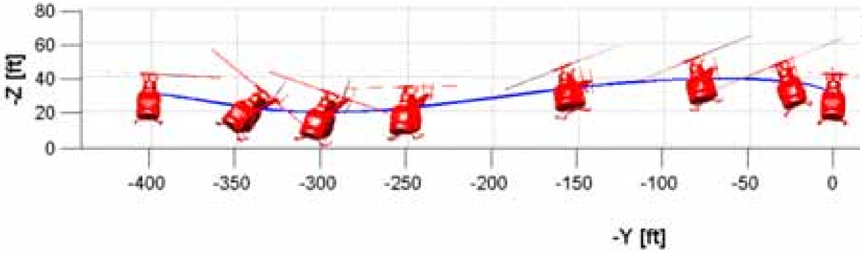
Merit function:

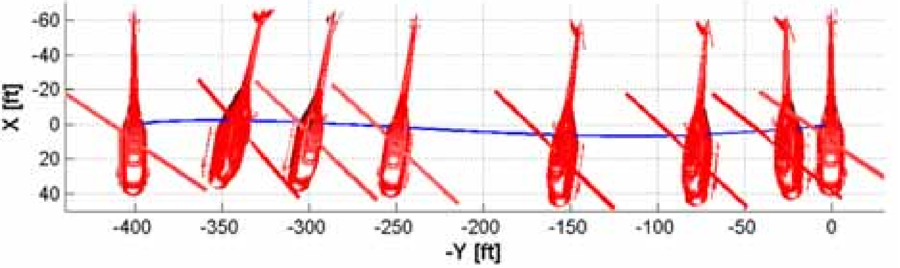
$$J = T + \frac{1}{T}\int_0^T \dot{\boldsymbol{u}} \cdot \boldsymbol{W} \dot{\boldsymbol{u}}\, \mathrm{d}t$$

Good Visual Environment, cargo/utility:

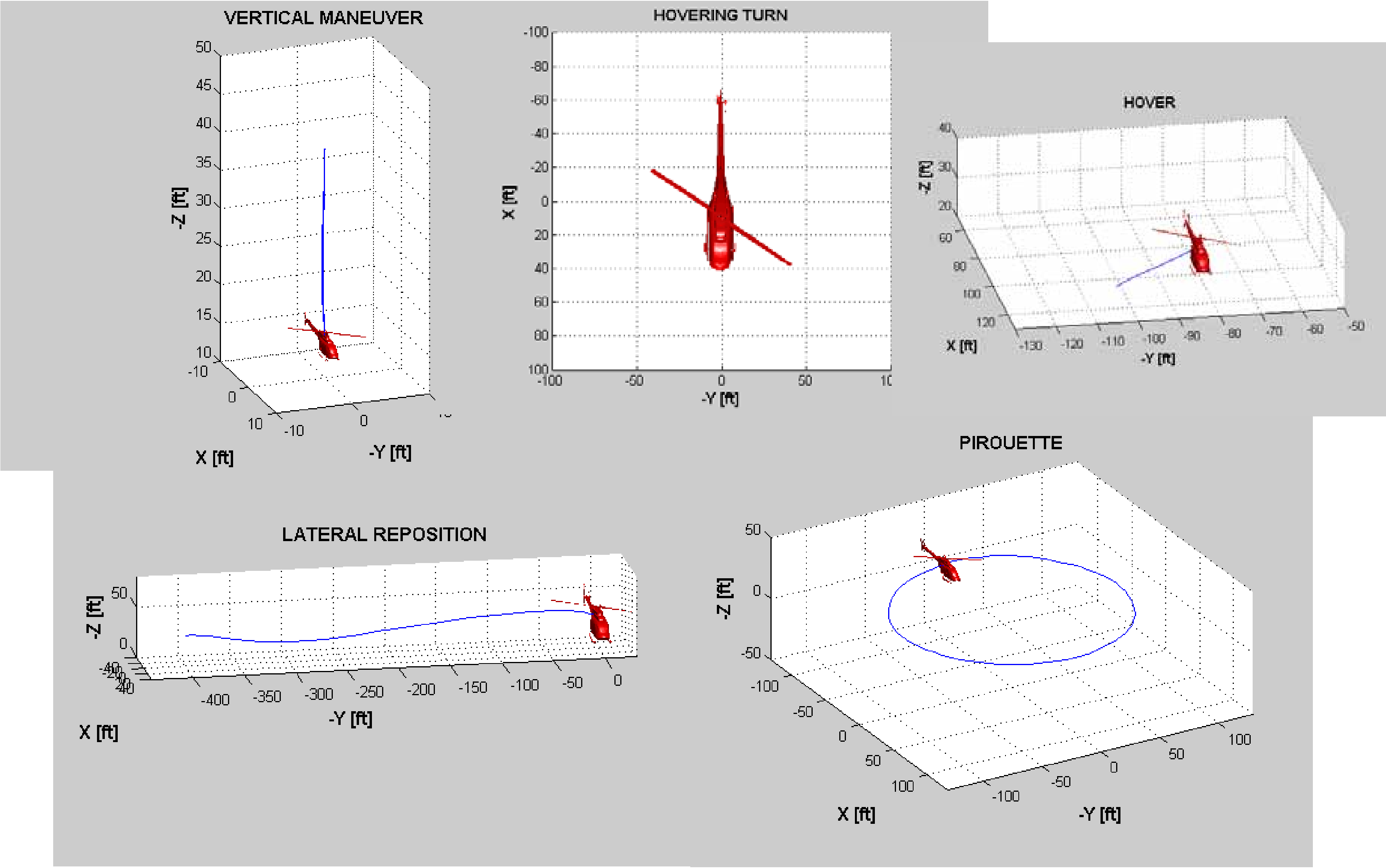
Longitudinal and lateral tracking error of ±10 ft, heading error ±10 deg

Maneuver duration $T \leq 18$ sec

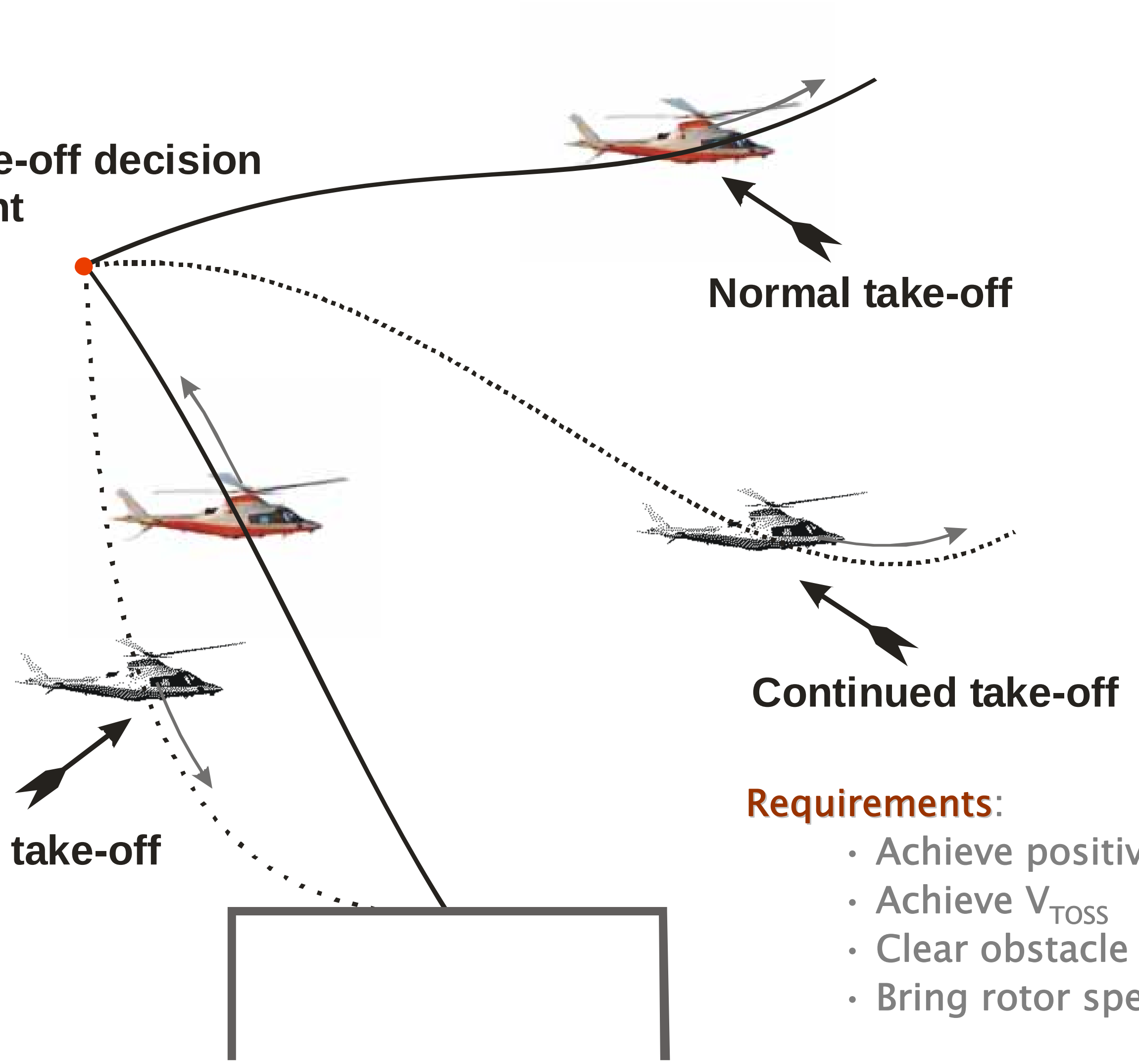




Applications: ADS-33 MTEs



Optimal Helicopter Multi-Phase CTO



Take-off decision point

Normal take-off

Continued take-off

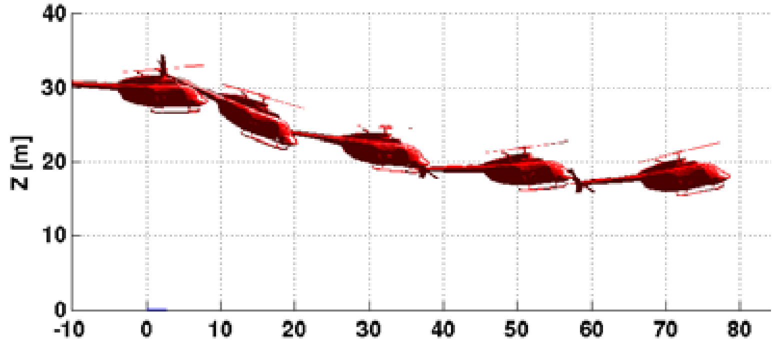
Rejected take-off

Requirements:

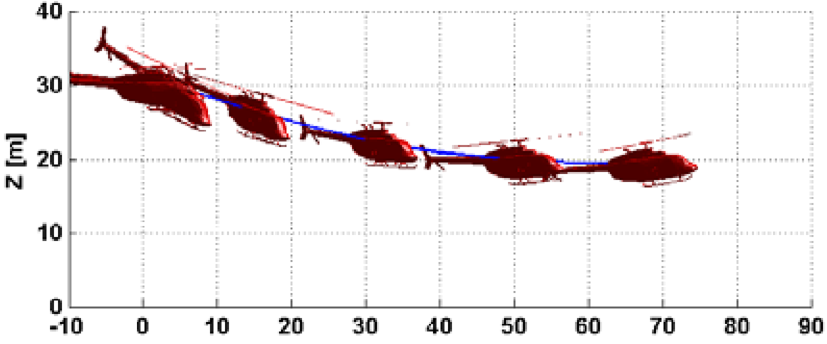
- Achieve positive rate of climb
- Achieve V_{TOSS}
- Clear obstacle of given height
- Bring rotor speed back to nominal

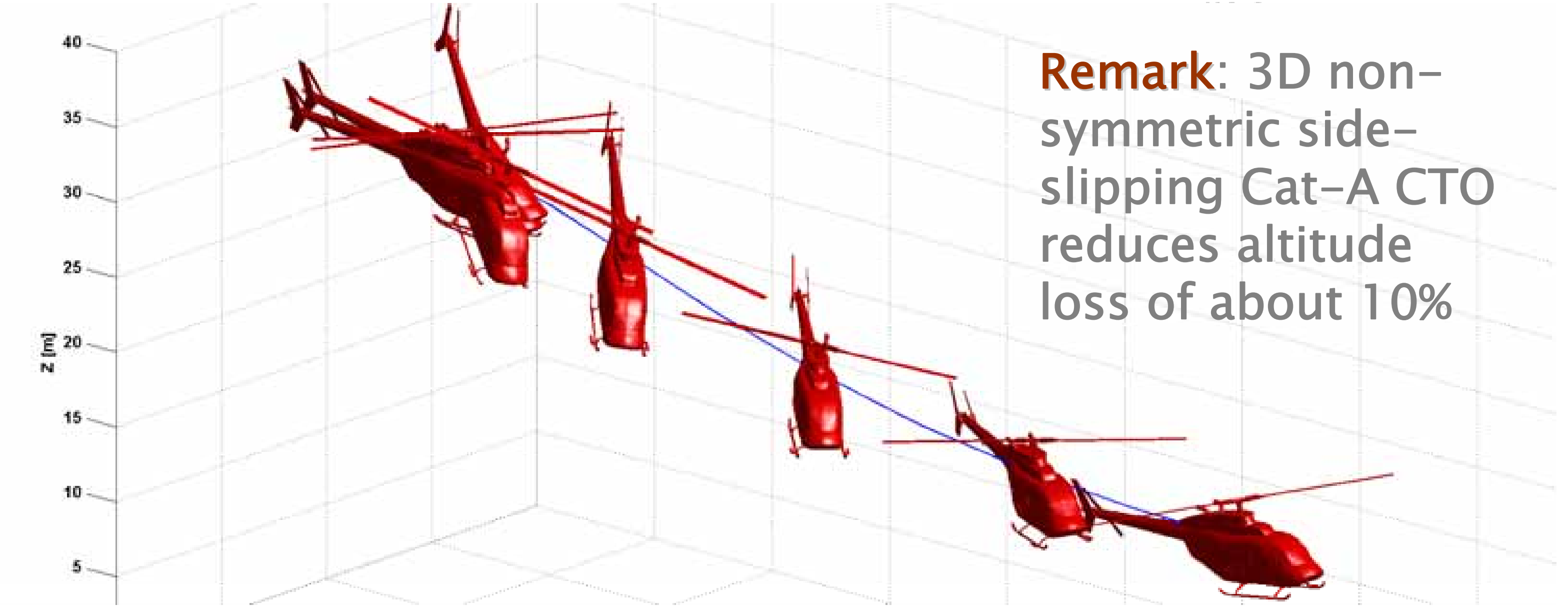
Optimal Helicopter Multi-Phase CTO

Maneuver flown symmetrically (2D):



Side-slipping (3D):





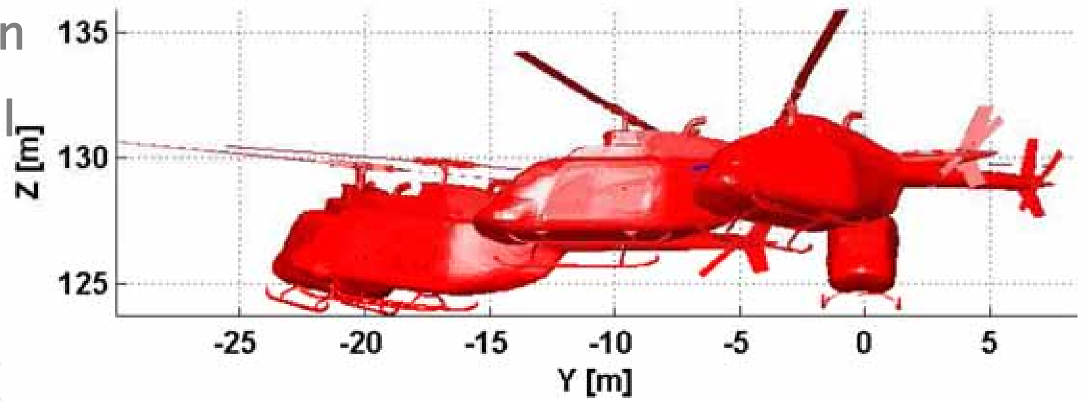
Remark: 3D non-symmetric side-slipping Cat-A CTO reduces altitude loss of about 10%

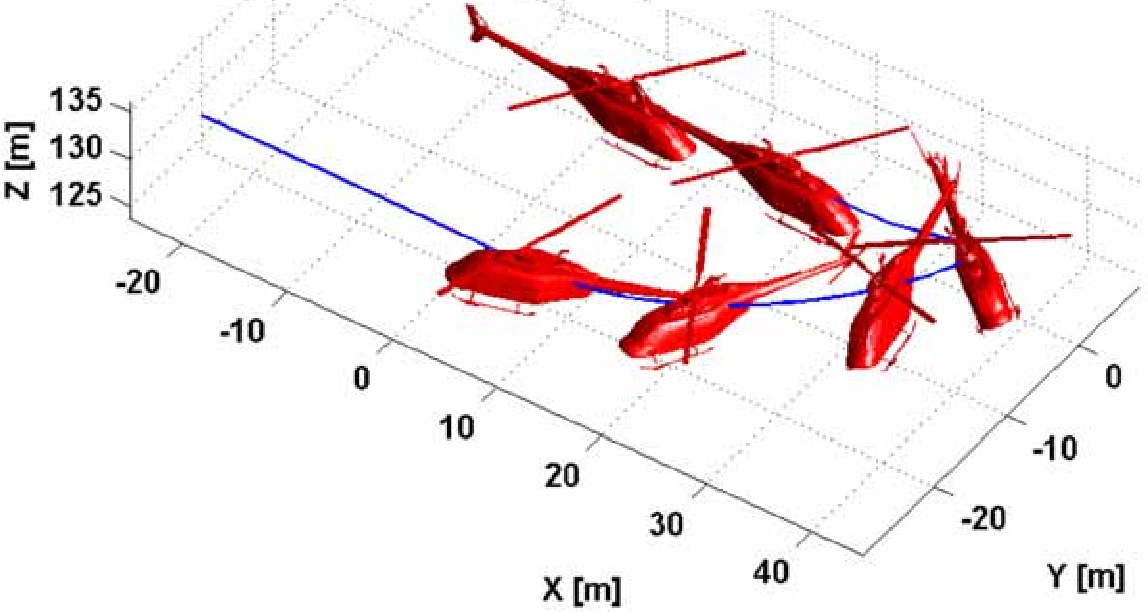
Applications: Minimum Time Turn

Minimum time 180-deg turn

FLIGHTLAB helicopter model

Direct transcription





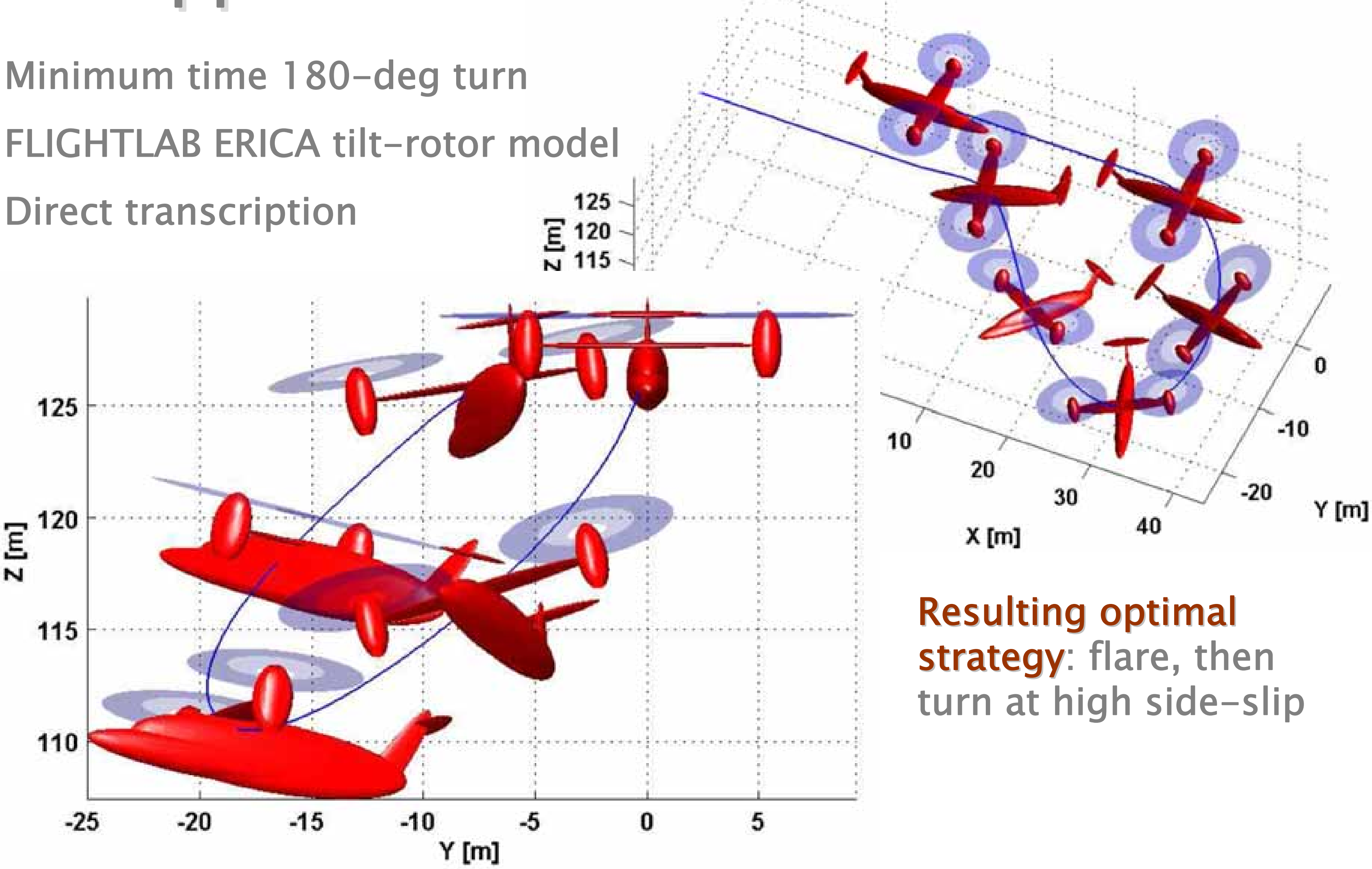
Resulting optimal strategy: classical bank and turn

Applications: Minimum Time Turn

Minimum time 180-deg turn

FLIGHTLAB ERICA tilt-rotor model

Direct transcription

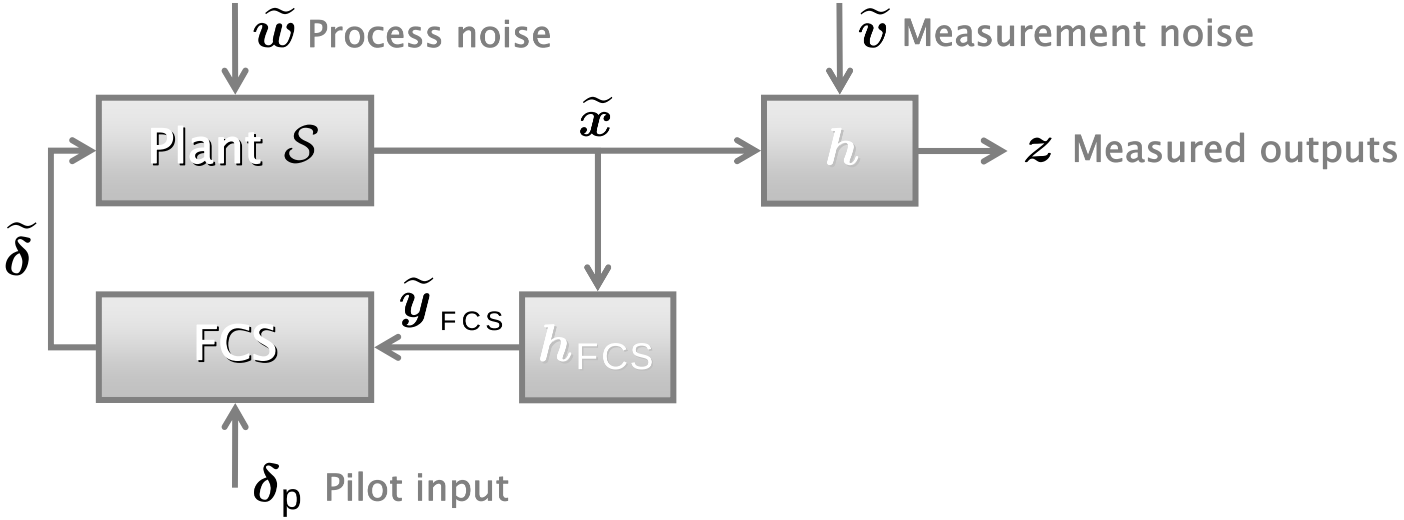


Resulting optimal strategy: flare, then turn at high side-slip

Applications and Corollary Technologies II: Time-Domain Parameter Estimation from Experimental Observations

Identification of Unstable Systems

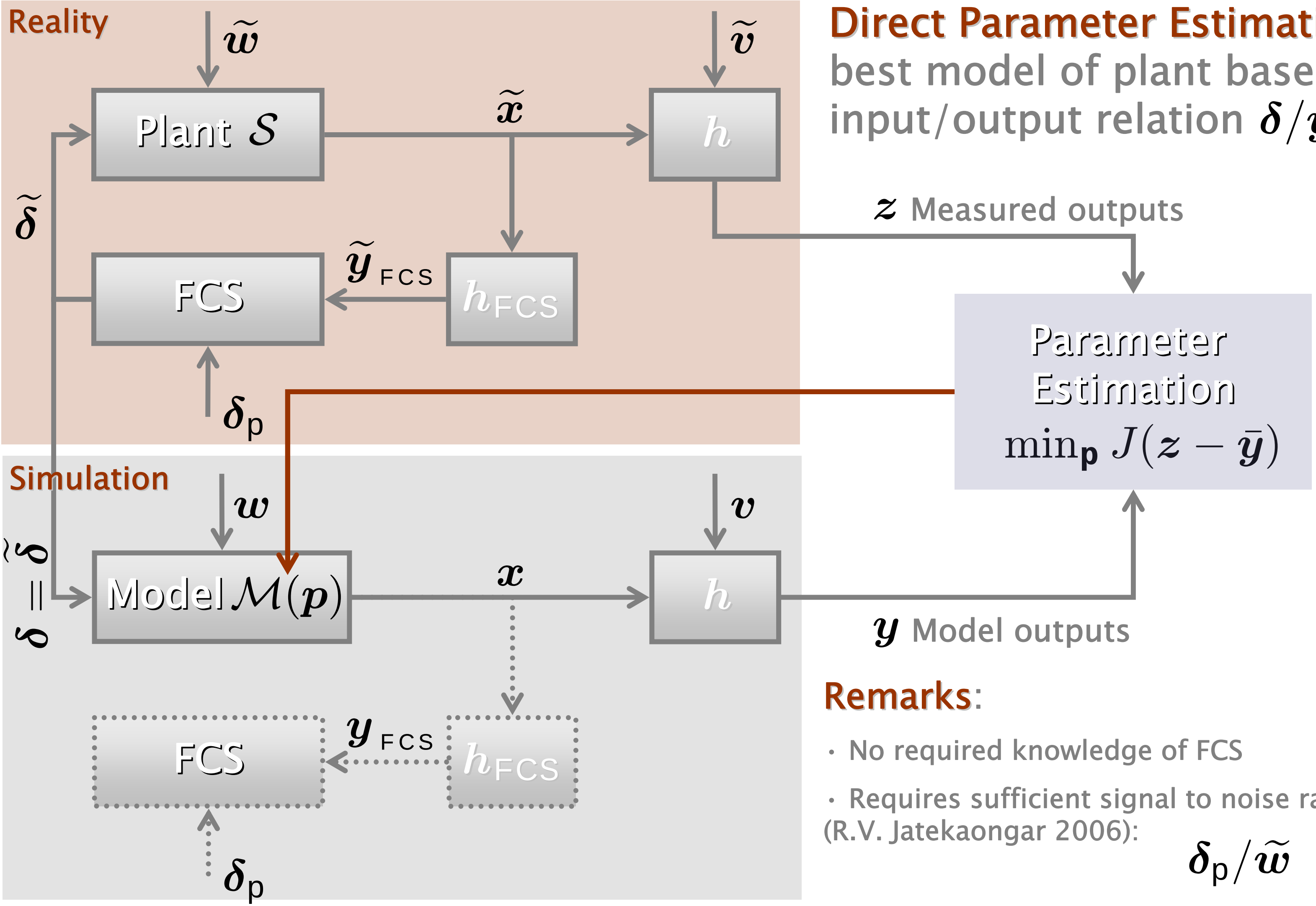
Closed-loop identification: data collected under closed-loop operation



Remarks:

- Noise introduces a stochastic input into the system: $\widetilde{\boldsymbol{\delta}} = \widetilde{\boldsymbol{\delta}}(\widetilde{\boldsymbol{\delta}}_{\mathrm{p}}, \widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{w}})$
- Correlated inputs and outputs
- FCS suppresses excitation

Direct Approach



Direct Parameter Estimation: best model of plant based on input/output relation $\boldsymbol{\delta}/\boldsymbol{y}$

Remarks:

- No required knowledge of FCS
- Requires sufficient signal to noise ratio (R.V. Jatekaongar 2006): $\boldsymbol{\delta}_{\mathrm{p}}/\widetilde{\boldsymbol{w}}$

Time Domain Parameter Estimation

Estimation by optimization: formally very similar to maneuver optimization

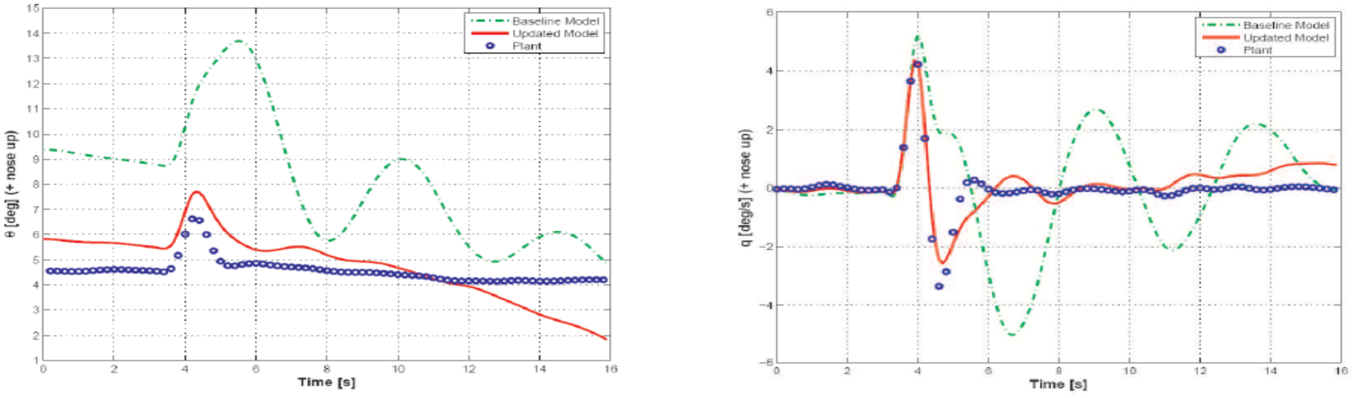
- Various possible versions: least-squares, output error (maximum likelihood), filter error

- Integrated software for trajectory optimization and parameter estimation from flight test data (common data structures, discretization methods, NLP solvers, vehicle models and interfaces, over 95% of the lines of code)

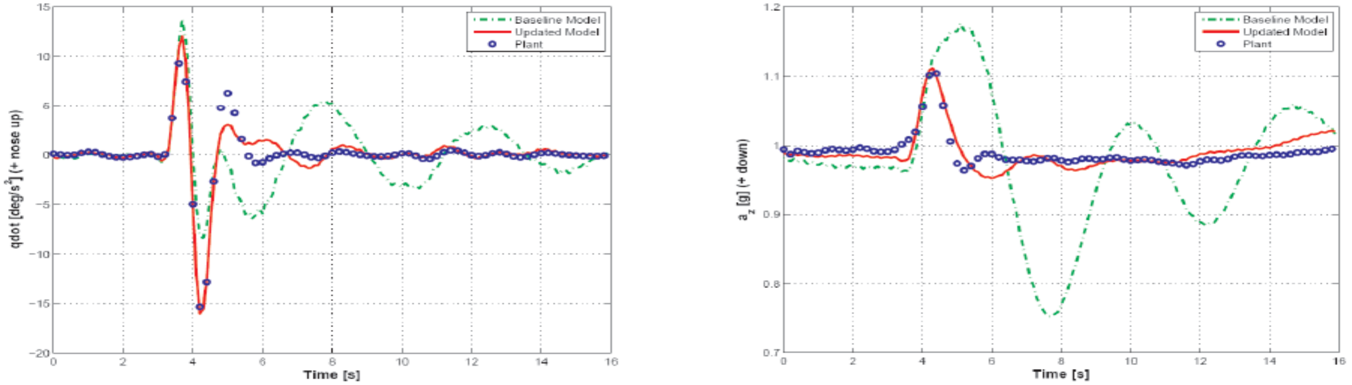
- Maneuvers for optimal estimation of model parameters

Time Domain Parameter Estimation

Rotorcraft parameter estimation from flight testing



(a) Pitch Attitude. (b) Pitch Rate



(c) Pitch Acceleration (d) Z-axis Acceleration

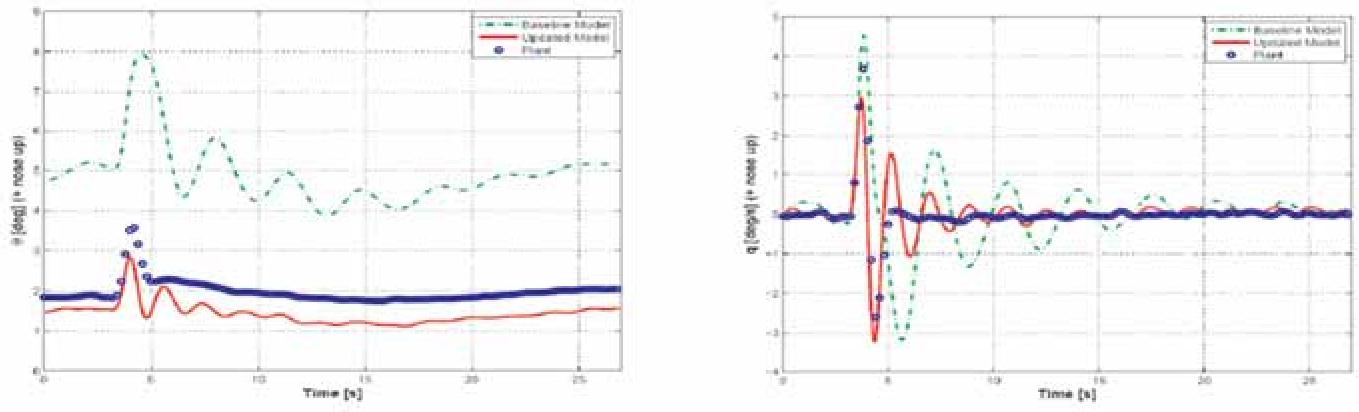
Baseline model

Updated model

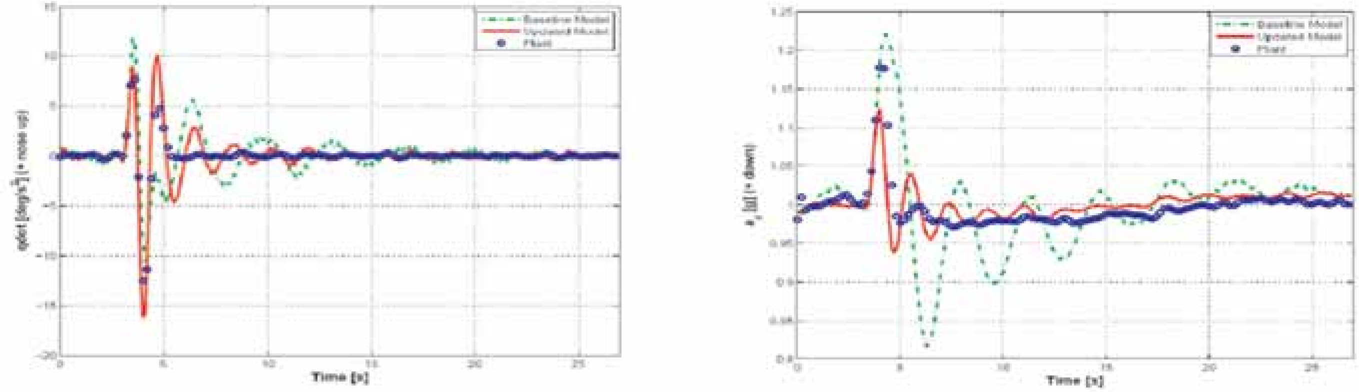
Flight data

Time Domain Parameter Estimation

Rotorcraft parameter estimation from flight testing



(a) Pitch Attitude. (b) Pitch Rate



(c) Pitch Acceleration (d) Z-axis Acceleration

- - · Baseline model

— Updated model

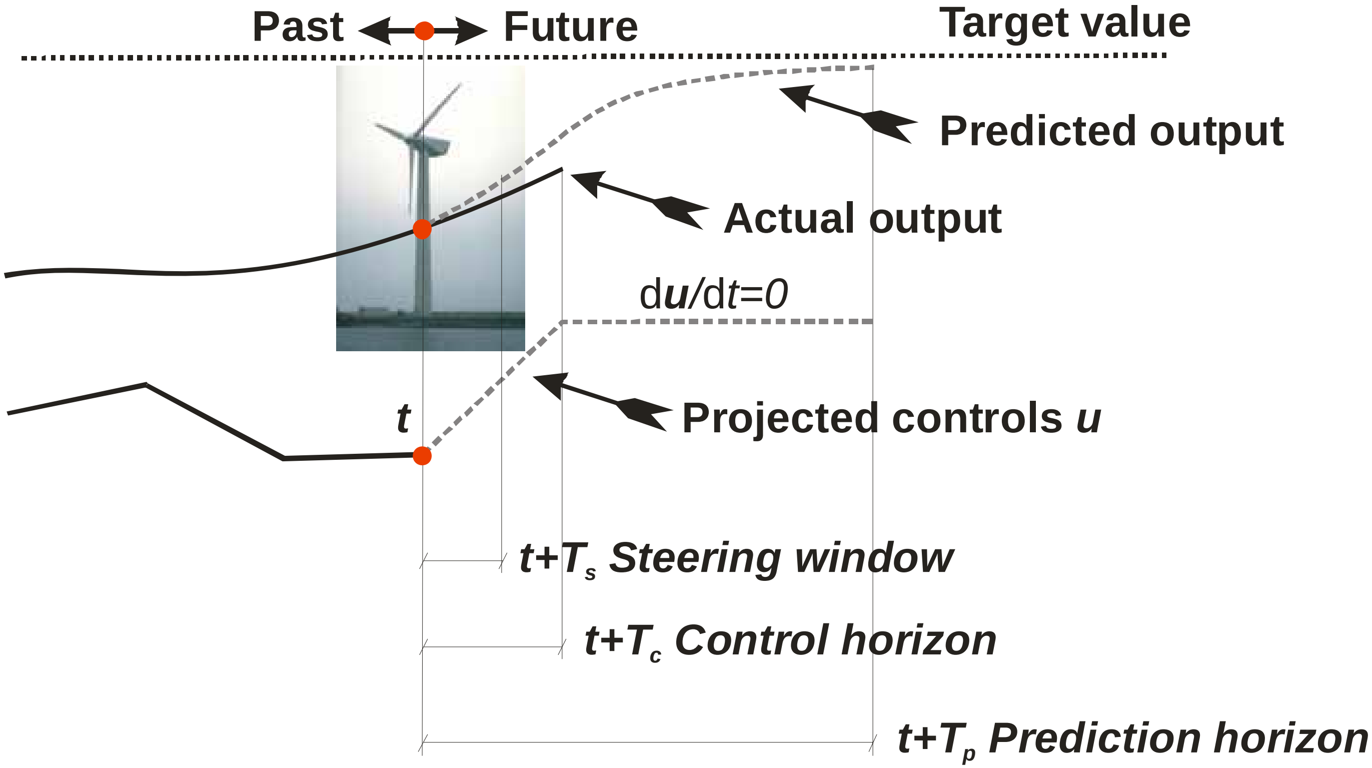
○ Flight data

Applications and Corollary Technologies III:
Advanced Control Laws for Wind Turbines

Control of Wind Turbine Generators

Goals:

- Regulate wind turbine by adjusting blade pitch (and possibly generator torque) to react against **wind turbulence** and **gusts**
- Minimize **fatigue damage** and maximize **power output**



Past ⟷ **Future**

Target value

Predicted output

Actual output

d*u*/d*t*=0

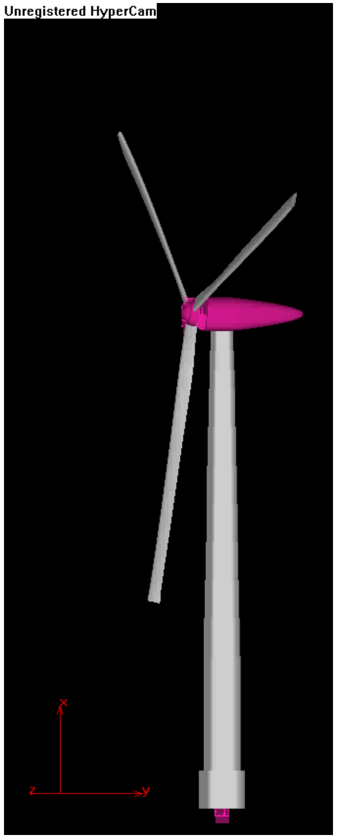
Projected controls *u*

t

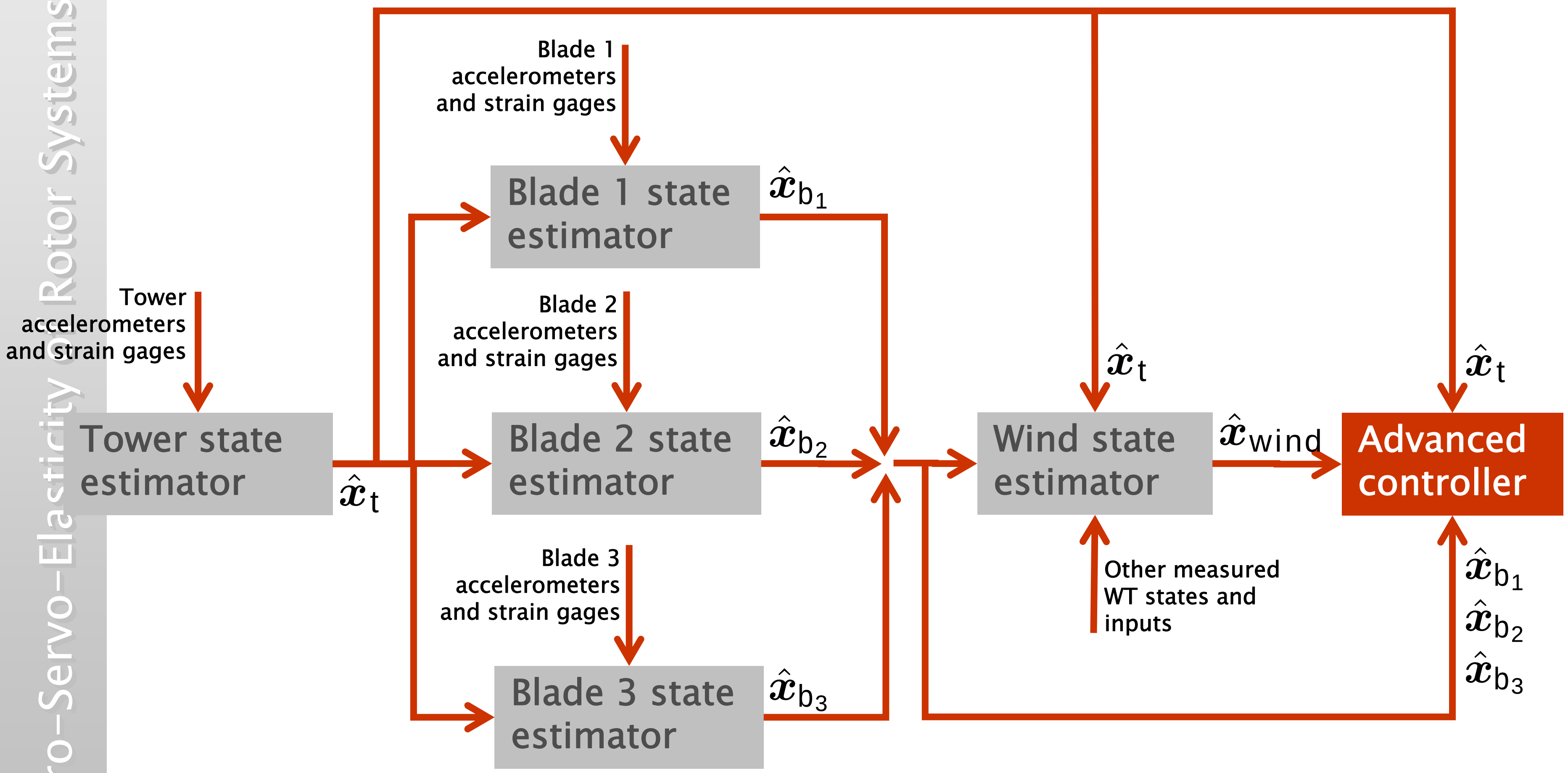
***$t+T_s$* Steering window**

***$t+T_c$* Control horizon**

***$t+T_p$* Prediction horizon**

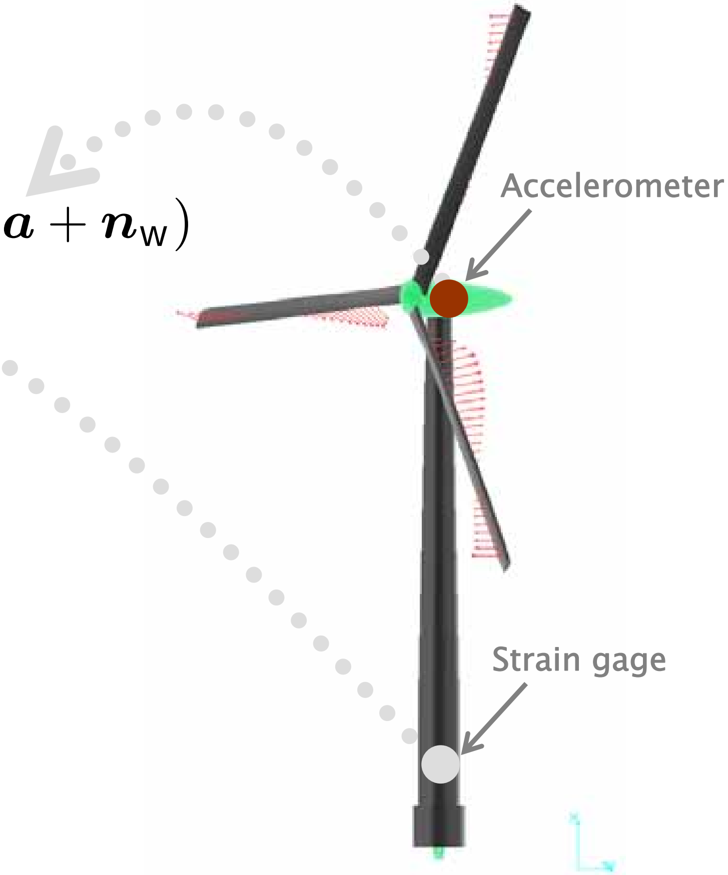


Advanced Control Laws and Cascading Kalman Observers



Tower State Observer

Kalman modal-based tower observer:



Accelerations:
$$\begin{cases} \dot{\boldsymbol{q}} &= \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= (\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathsf{T}}(\boldsymbol{a}+\boldsymbol{n}_{\mathsf{W}}) \end{cases}$$

Curvatures:
$$\boldsymbol{c} = \boldsymbol{\Phi}''\boldsymbol{q} - \boldsymbol{n}_{\mathsf{V}}$$

- Unknown modal amplitudes: $\boldsymbol{q}$
- Modal bases: $\boldsymbol{\Phi}$
- Process & measurement noise: $\boldsymbol{n}_{\mathsf{W}}, \boldsymbol{n}_{\mathsf{V}}$

▶ **Remarks**:

- Fore-aft and side-side identification
- Multiple modal ampl. (sensor number and position for observability)
- Formulation applicable also to identification of flap-lag blade states

Tower State Observer

State space form:

$$\left\{ \begin{array}{rcl} \dot{\boldsymbol{x}} & = & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{W}\boldsymbol{n}_{\mathsf{w}} \\ \boldsymbol{y} & = & \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} + \boldsymbol{V}\boldsymbol{n}_{\mathsf{v}} \end{array} \right.$$

with
$$\boldsymbol{x} = (\boldsymbol{q}^{\mathsf{T}}, \boldsymbol{v}^{\mathsf{T}})^{\mathsf{T}} \quad \boldsymbol{u} = \boldsymbol{a} \quad \boldsymbol{y} = \boldsymbol{c}$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\Psi} \end{bmatrix} \quad \boldsymbol{C} = \begin{bmatrix} \boldsymbol{\Phi}'' & \boldsymbol{0} \end{bmatrix} \quad \boldsymbol{D} = \begin{bmatrix} \boldsymbol{0} \end{bmatrix}$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\Psi} \end{bmatrix} \quad \boldsymbol{V} = \begin{bmatrix} \boldsymbol{I} \end{bmatrix}$$

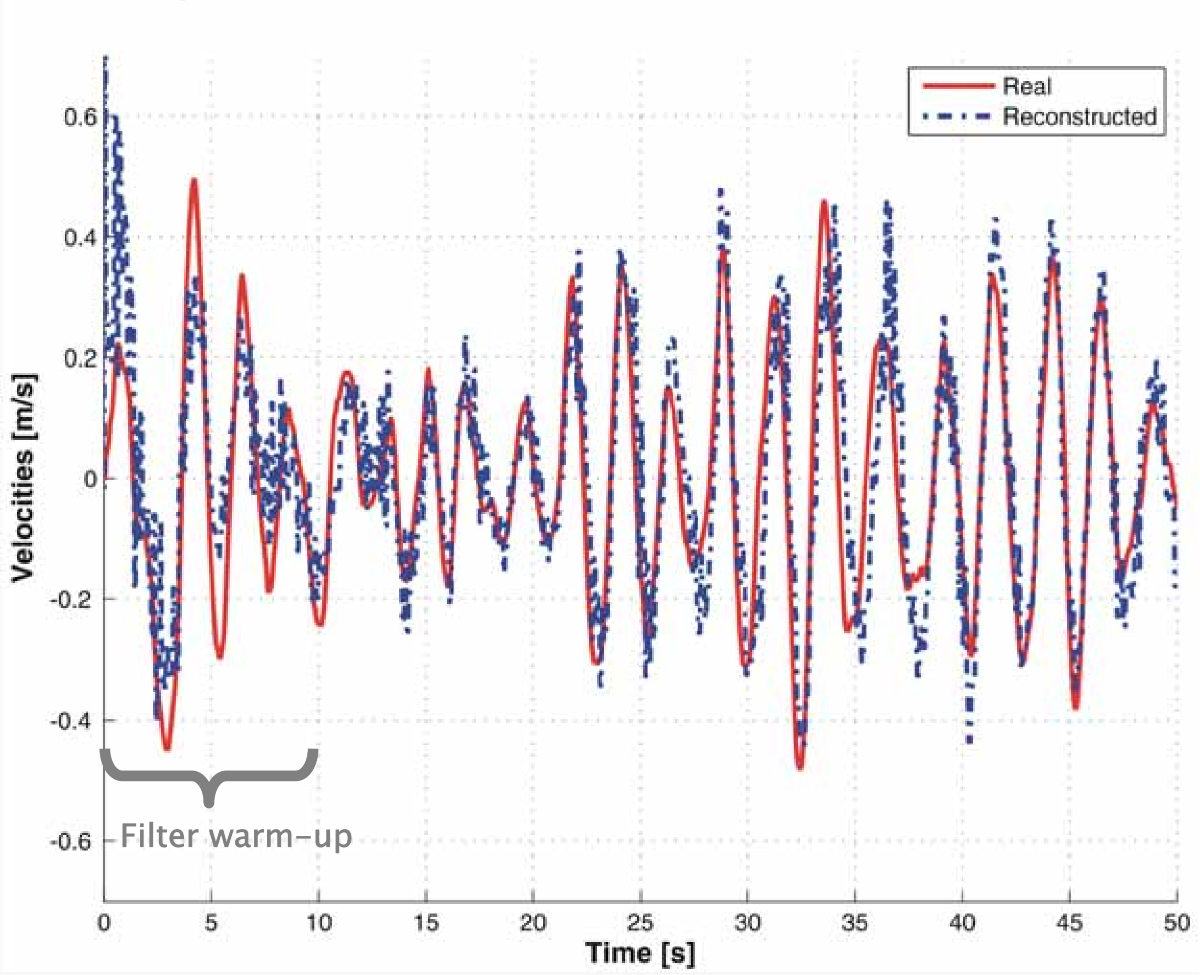
Optimal Kalman state estimate:

$$\boldsymbol{x}_{\mathsf{k}} = \boldsymbol{x}_{\mathsf{k}}^{-} + \boldsymbol{K}_{\mathsf{k}}(\hat{\boldsymbol{y}}_{\mathsf{k}} - \boldsymbol{y}_{\mathsf{k}}^{-})$$

- Filter gain matrix $\boldsymbol{K}_{\mathsf{k}}$
- Propagated states and outputs based on accelerometric reading: $\boldsymbol{x}_{\mathsf{k}}^{-}, \boldsymbol{y}_{\mathsf{k}}^{-}$
- Curvature reading: $\hat{\boldsymbol{y}}_{\mathsf{k}}$

Tower State Observer

Tower tip velocity estimation:

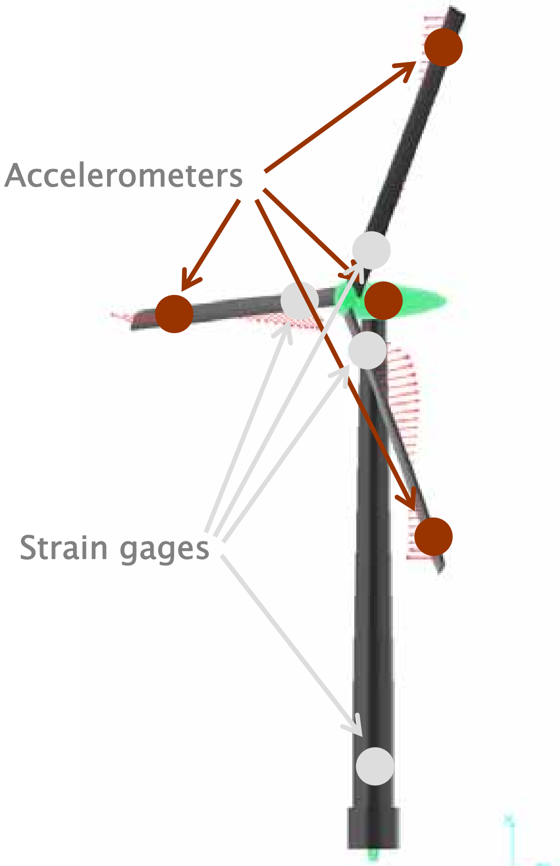


Tower and Blade State Observer

Kalman modal-based tower and blade state observer:

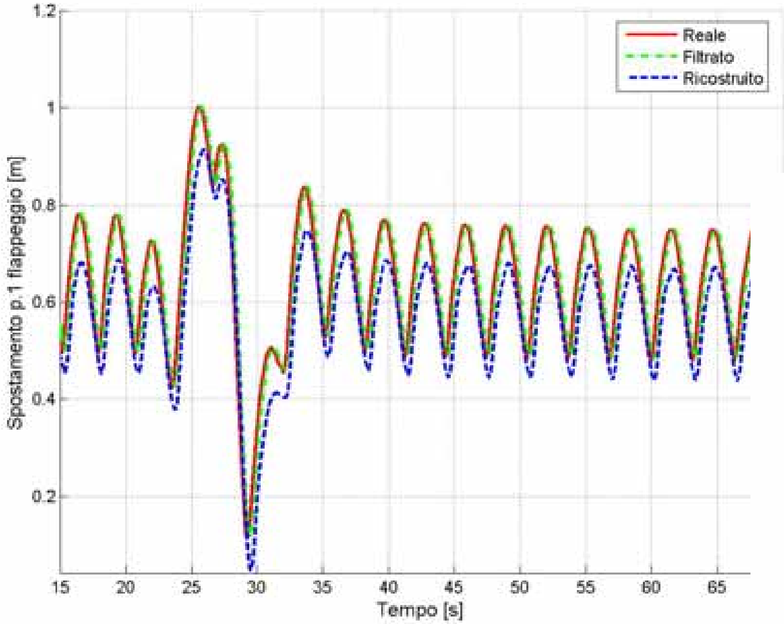
Compute or measure modal bases for blades and tower

- Integrate tower kinematic equations from accelerations
- Correct with tower strain gage curvature readings
- Integrate blade kinematic equations from blade and tower accelerations
- Correct with blade strain gage curvature readings



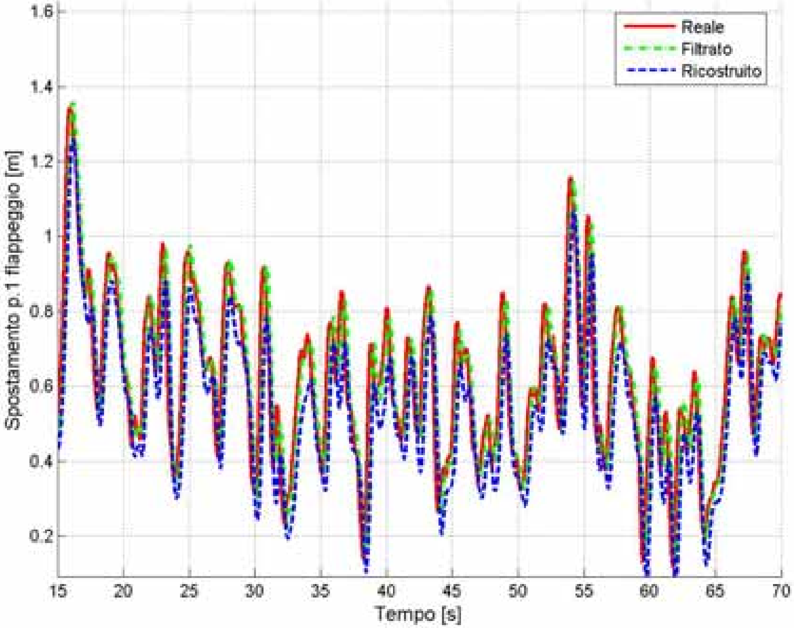
Tower and Blade State Observer

Real and reconstructed flap displacement:



▲ EOG$_1$-13 case

▼ Turbulent wind ($V_\mathrm{m} = 15$m/sec)



Wind Observer

Anemometer:

- Cup, but also laser, ultrasonic, etc.
- Measurements highly inaccurate because of
  - Rotor wake
  - Wake turbulence
  - Nacelle disturbance
- Sufficient accuracy for supervision tasks and yaw alignment
- Not sufficient for sophisticated control law implementation

Need ways to **reconstruct** wind blowing on rotor from **reliable measurements** (pitch setting, rotor speed, etc.)

Wind Observer

Extended Kalman wind observer:

- Wind equation: $\dot{V}_{\mathsf{w}} = n_{\mathsf{w}}$
- Output measurement torque-balance equation:

$$y = (J_{\mathsf{R}} + J_{\mathsf{G}})\dot{\Omega} + T_{\mathsf{l}}(\Omega) + T_{\mathsf{el_e}} - T_{\mathsf{a}}(\Omega, \beta_{\mathsf{e}}, V_{\mathsf{w}} - \dot{d}, V_{\mathsf{m}}) + n_{\mathsf{v}}$$

Non-linear state-space form:

$$\begin{cases} \dot{x} & = & f(x, \boldsymbol{u}, n_{\mathsf{w}}) \\ y & = & h(x, \boldsymbol{u}, n_{\mathsf{v}}) \end{cases}$$

with $x = V_{\mathsf{w}} \quad \boldsymbol{u} = (\dot{\Omega}, \Omega, \beta_{\mathsf{e}}, \dot{d}, V_{\mathsf{m}})^{\mathsf{T}}$

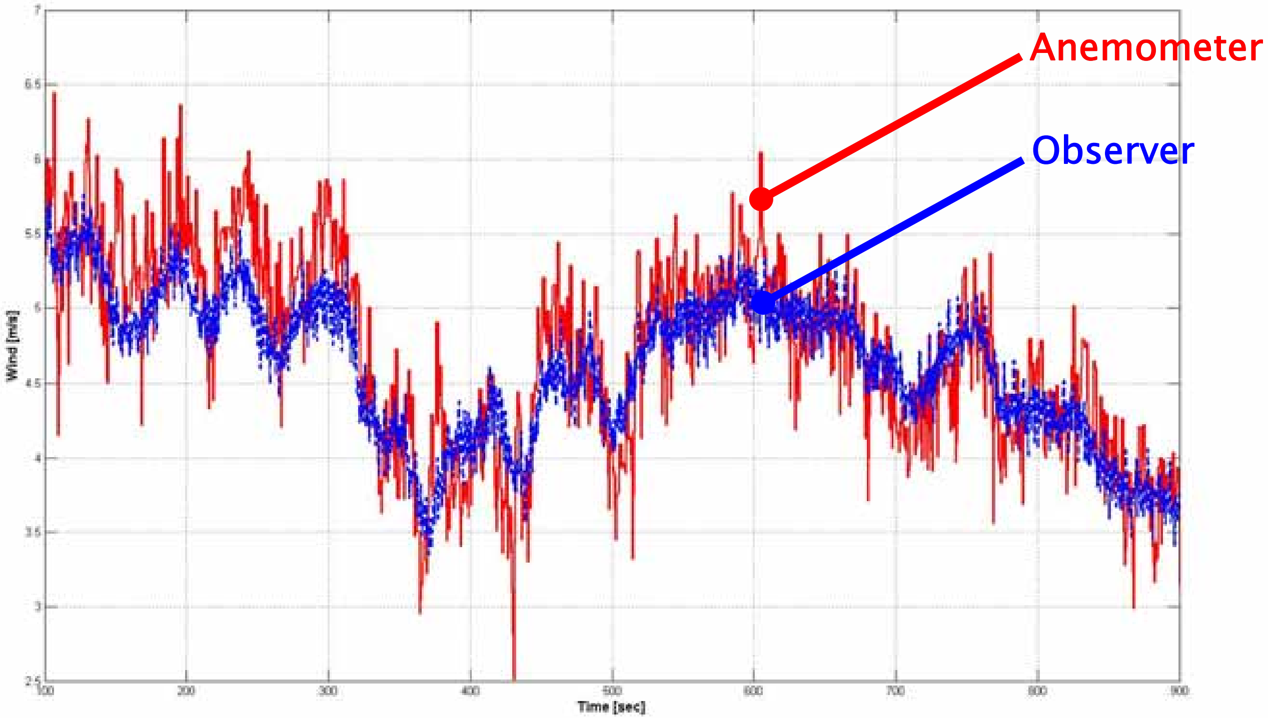
Extended Kalman estimate $x_{\mathsf{k}} = x_{\mathsf{k}}^{-} + K_{\mathsf{k}}(\hat{y}_{\mathsf{k}} - y_{\mathsf{k}}^{-})$

with measured output $\hat{y}_{\mathsf{k}} = 0$ to enforce torque-balance equation

Mean wind V_{m} reconstructed with moving average on 10 sec window

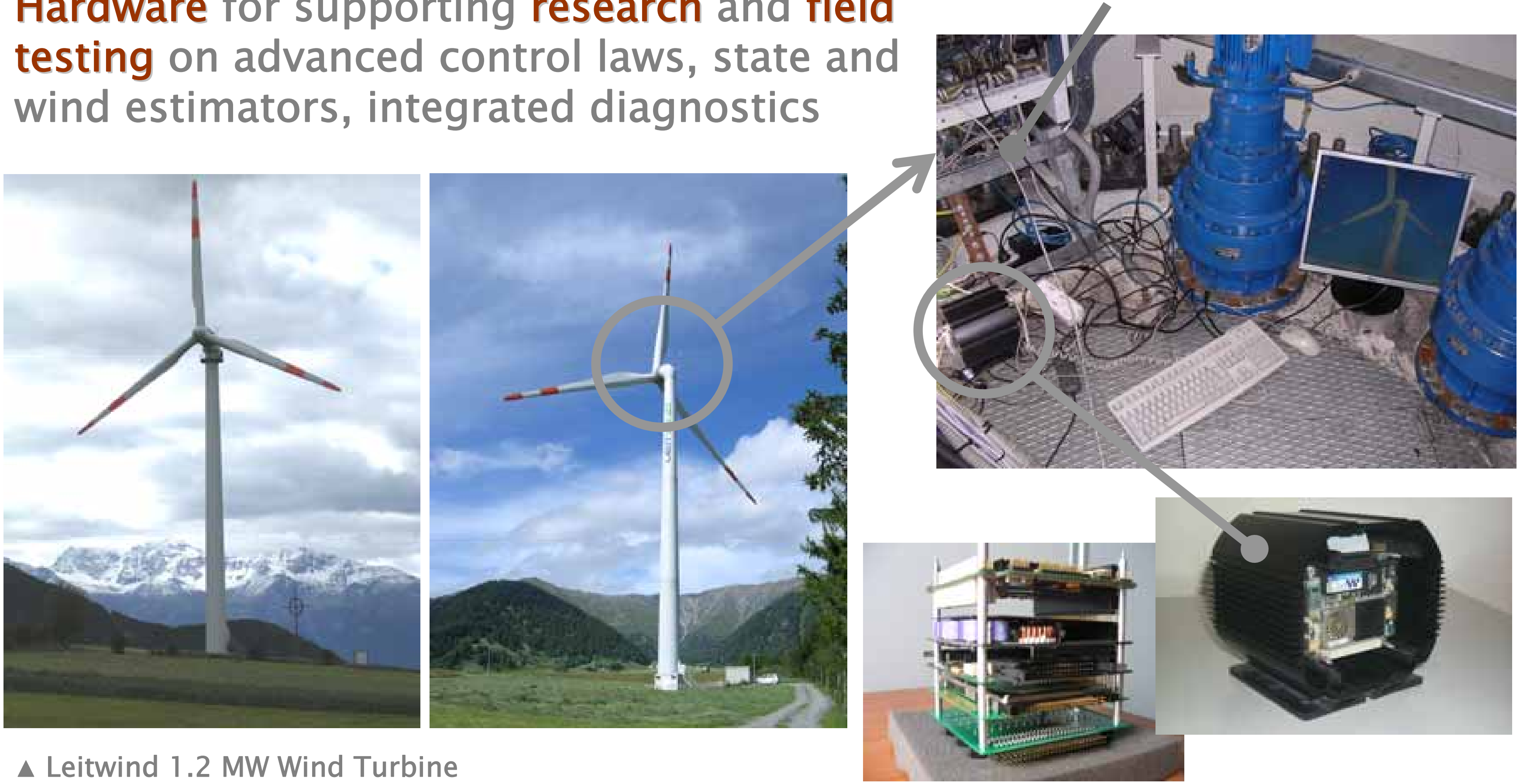
Wind Observer

Field testing on 1.2 MW wind turbine:



PoliMi Control Research Platform

Hardware for supporting **research** and **field testing** on advanced control laws, state and wind estimators, integrated diagnostics



▲ Leitwind 1.2 MW Wind Turbine

- Hub height 65m
- Rotor radius 38m

- PC/104 architecture, Pentium M 1.6 GHz
- Linux real-time operative system

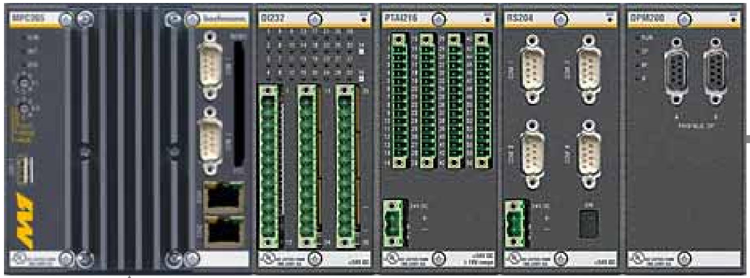
PoliMi Control Research Platform

From sensors:

Anemometer, inverter, pitch regulator, yaw

To servos:

Pitch, yaw, torque setpoints



◄ Collect data, interface with servos, compute yaw control

Serial communication RS485 @1Hz

- Torque
- Rotor speed
- Azimuth
- Blade pitch angle
- Wind

- Pitch control
- Torque control

- Complete compatibility with and minimum impact on existing on-board system
- Substantial computing power
- On-board system can give control to and regain control from research platform at any time

Analog inputs:

Tower accelerations and strain gages

▲ Controller and observer algorithms, interface with on-board industrial controller

Wind Observer

Simple mean hub wind reconstruction from torque balance equation

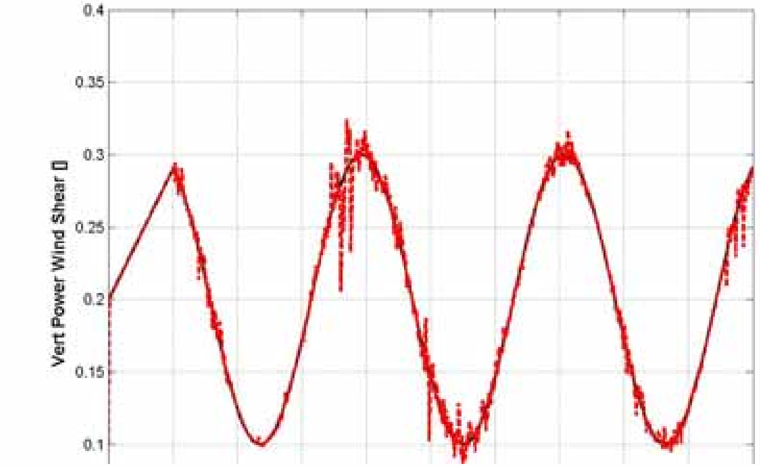
More in general:

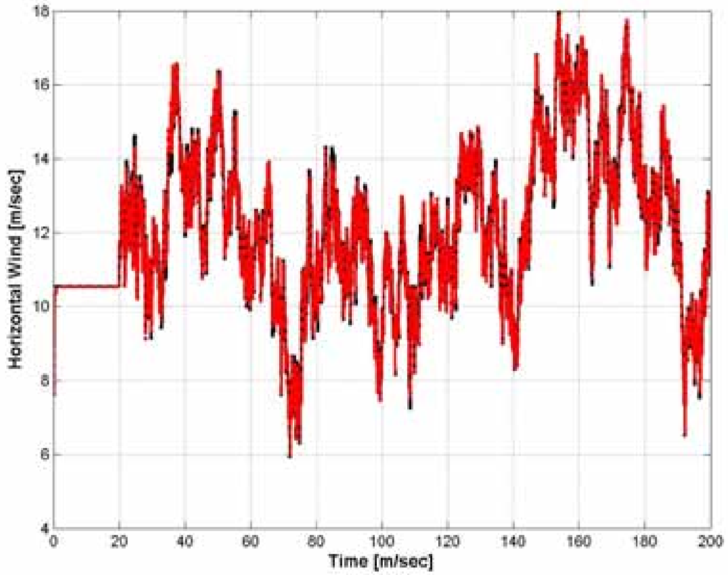
The rotor system is a **sensor** which responds to **temporal** as well as **spatial** wind variations

Model-based interpretation of response can be used for reconstructing **wind states** (e.g. vertical and horizontal wind shear parameters, vertical and horizontal wind components) for improved rotor control

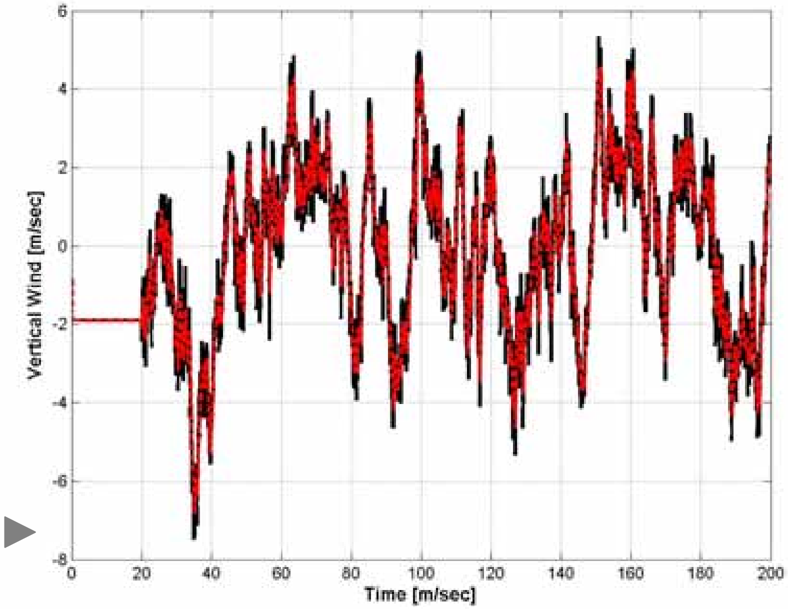
Wind Observer

Vertical wind shear parameter ▶





▲ Horizontal wind component



Vertical wind component ▶

Conclusions and Outlook

Conclusions and Outlook

Observations:

- Computational procedures now blend traditionally separate disciplines, e.g. **aero-servo-elasticity** with **flight mechanics**
- High fidelity virtual models permeate the **design** and **verification** of complex engineering systems
- Maturity of the simulation tools is pushing **corollary supporting technologies** (efficient solution of very large optimal control problems, system identification, model reduction, etc. etc.)
- Mathematical models are becoming **so complex** that there is a trend to use methods for analyzing **experimental data** (e.g. stability analysis, etc.)

Conclusions and Outlook

Outlook:

- These trends will continue (**virtual lab**)
- The importance of **system identification** (reliable, statistically-based methods, elimination of manual tuning)
- **Improved models**: aerodynamics, other coupled fields (e.g. pilot models for maneuver simulations with pilot-in-the-loop effects)
- **Improved efficiency**: real-time simulation

Acknowledgements

Work in collaboration with:

A. Croce, D. Leonello, G. Maisano

Work supported by:

AgustaWestland, US Army Research Office, EU (NICETRIP), Leitwind, Vestas Wind Systems A/S, TREVI, Italian Ministry of Industrial Activities

More info, papers and presentations at:

- www.aero.polimi.it/~bottasso
- www.aero.polimi.it/~bottasso/POLI-Rotorcraft.htm
- www.aero.polimi.it/~bottasso/POLI-Wind.htm