Numerical Optimal Control and Airborne Wind Energy

Moritz Diehl

Control and Optimization Laboratory
Department of Microsystems Engineering (IMTEK)
Albert Ludwig University of Freiburg
und
Electrical Engineering Department (ESAT-STADIUS)
and Optimization in Engineering Center (OPTEC)
KU Leuven University, Belgium

Complex Sensor Actuator Systems

SENSORS

- GPS
- acceleration
- radar
- vision
- ...

→

How to connect ?

ACTUATORS

- flight surfaces
- steering wheel
- motor speeds
- joint torques
- ...

Aim: Optimally Operating Sensor Actuator Systems

SENSORS

- GPS
- acceleration
- radar
- vision
- ...

→

EMBEDDED OPTIMIZATION

→

ACTUATORS

- flight surfaces
- steering wheel
- motor speeds
- joint torques
- ...

Open Source Software Tools from the Control and Optimization Laboratory

under industry friendly LGPL license

- **qpOASES:** dense quadratic programming
  [Joachim Ferreau, …]
- **qpDUNES:** sparse quadratic programming
  [Janick Frasch, …]
- **ACADO:** nonlinear MPC [Boris Houska, Joachim Ferreau, Milan Vukov, Rien Quirynen, Robin Verscheuren, …]
- **CasADi:** modelling environment for dynamic optimization [Joel Andersson, Joris Gillis, Greg Horn, …]

ACADO - Computational Choices

1) Keep states in problem - use direct multiple shooting [1]

2) Exploit convexity via Generalized Gauss-Newton [2]

3) Use tangential predictors for short feedback delay [3]

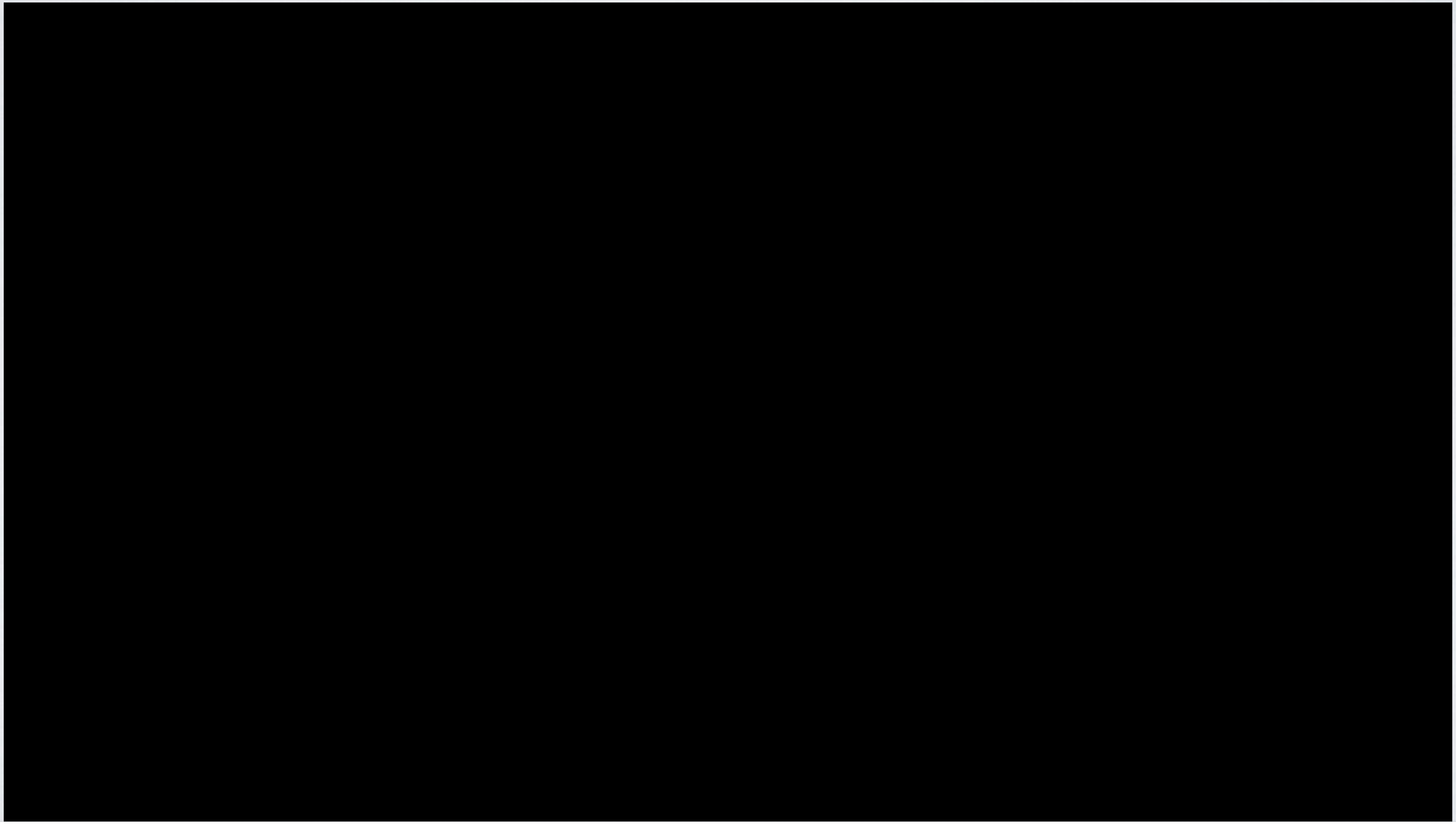
4) Iterate while problem changes (Real-Time Iterations) [4]

5) Auto-generate custom solvers in plain-C [5,6] (no `if`, no `malloc`)

[1] Bock & Plitt, IFAC WC, 1984
[2] Bock 1983
[3] Bock et al, , Ascona, 1999
[4] D. et al., J. Proc. Cont, 2002
[5] Mattingley & Boyd, Automatic code generation for real-time convex optimization, 2009

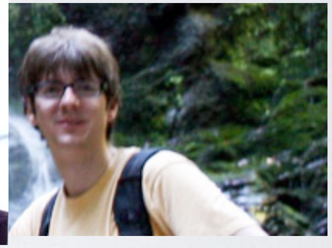
Time-optimal "racing" of model cars

Univ. Leuven/ETH & LMS [Robin Verscheuren] (ACADO)



CasADi

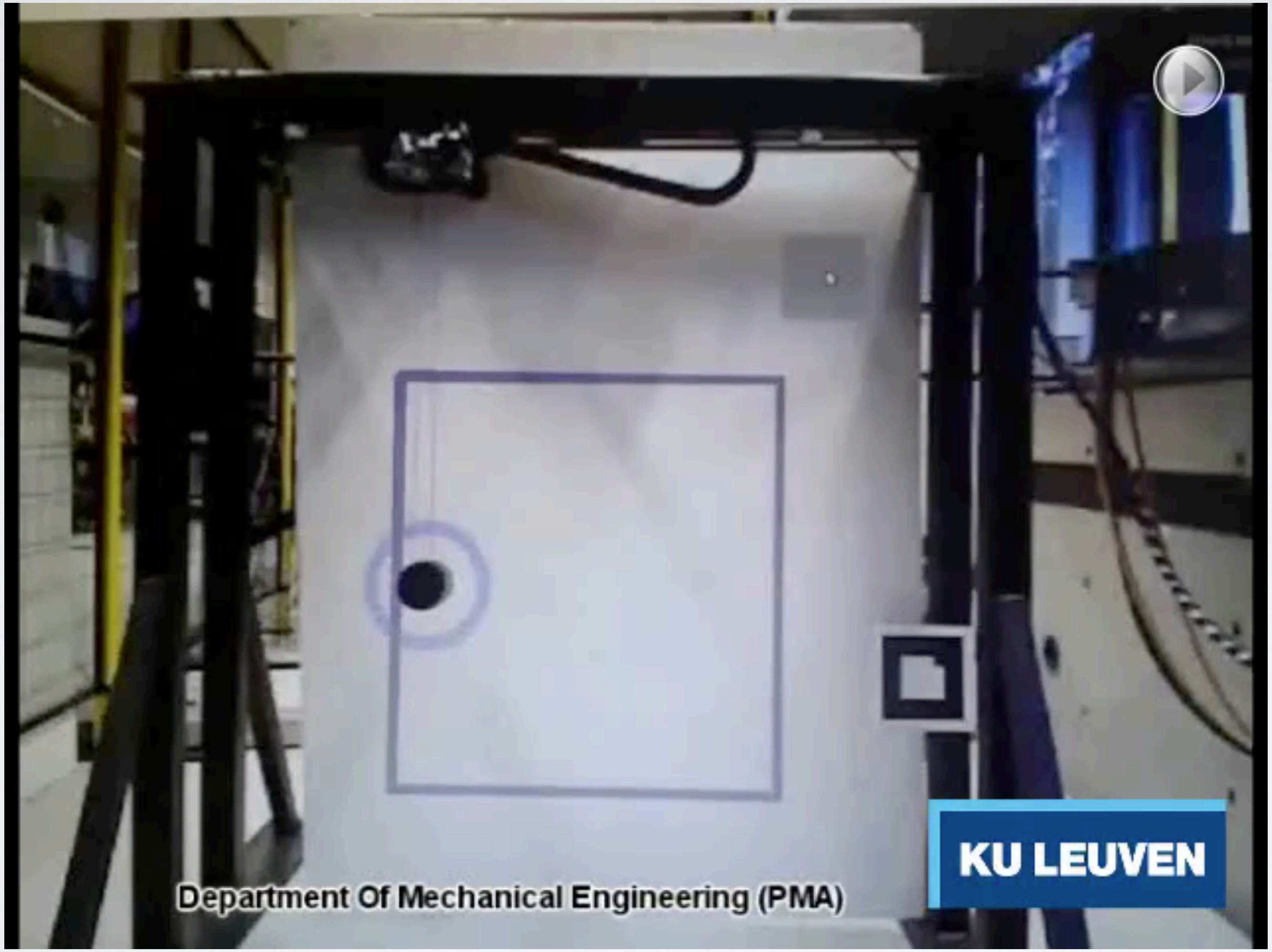
- "Computer Algebra System for Automatic Differentiation"
- Implements AD on sparse matrix-valued computational graphs
- Open-source tool (LGPL): www.casadi.org, developed by Joel Andersson and Joris Gillis

- Front-ends to C++, Python and Octave
- Symbolic model import from Modelica (via Jmodelica.org)
- Interfaces to: SUNDIALS, CPLEX, qpOASES, IPOPT, KNITRO,
- "Write efficient optimal control solver in a few lines"

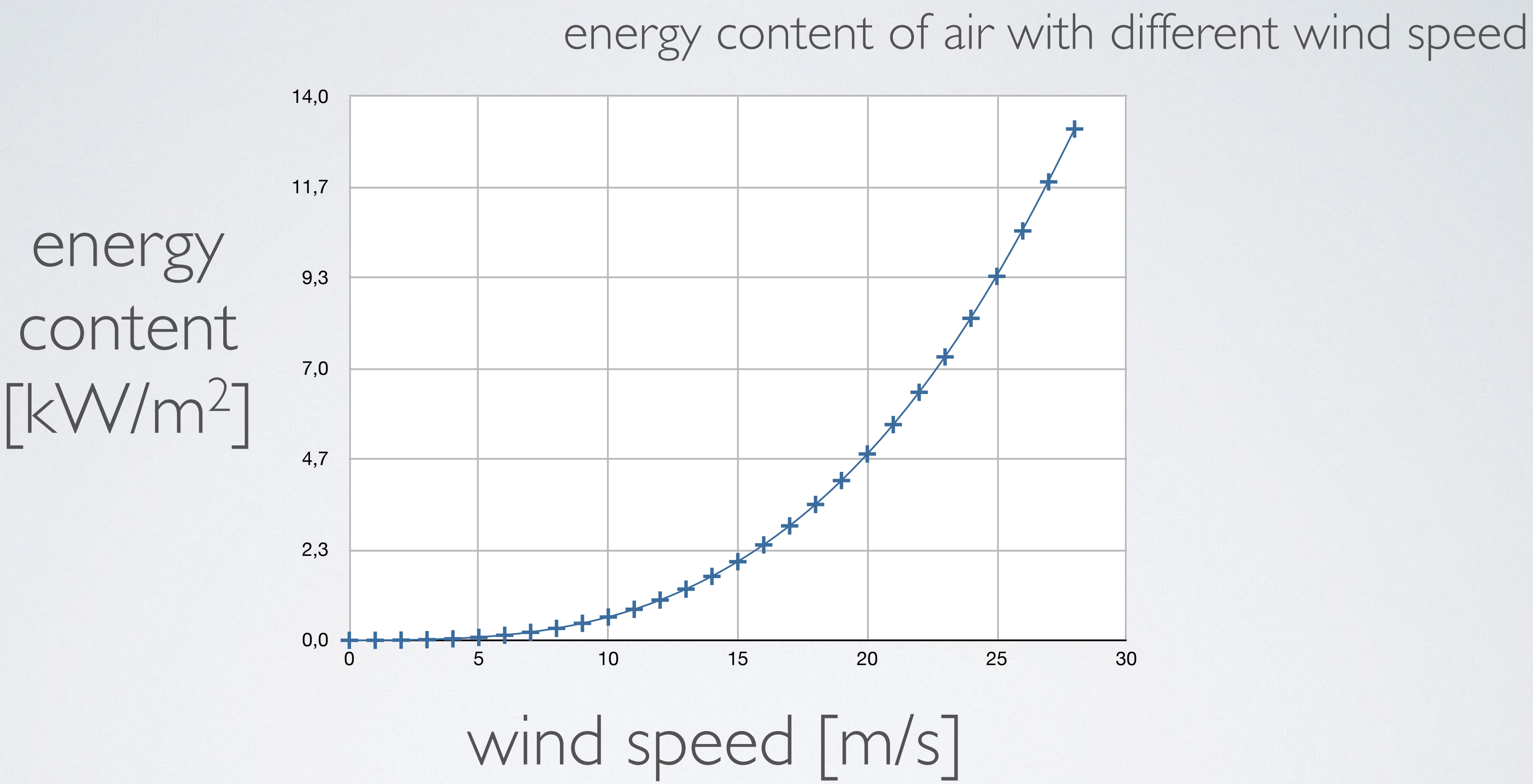
Time-optimal "drawing" by crane

Univ. Leuven [Wannes Van Loock et al.,] (CasADi)



AIRBORNE WIND ENERGY

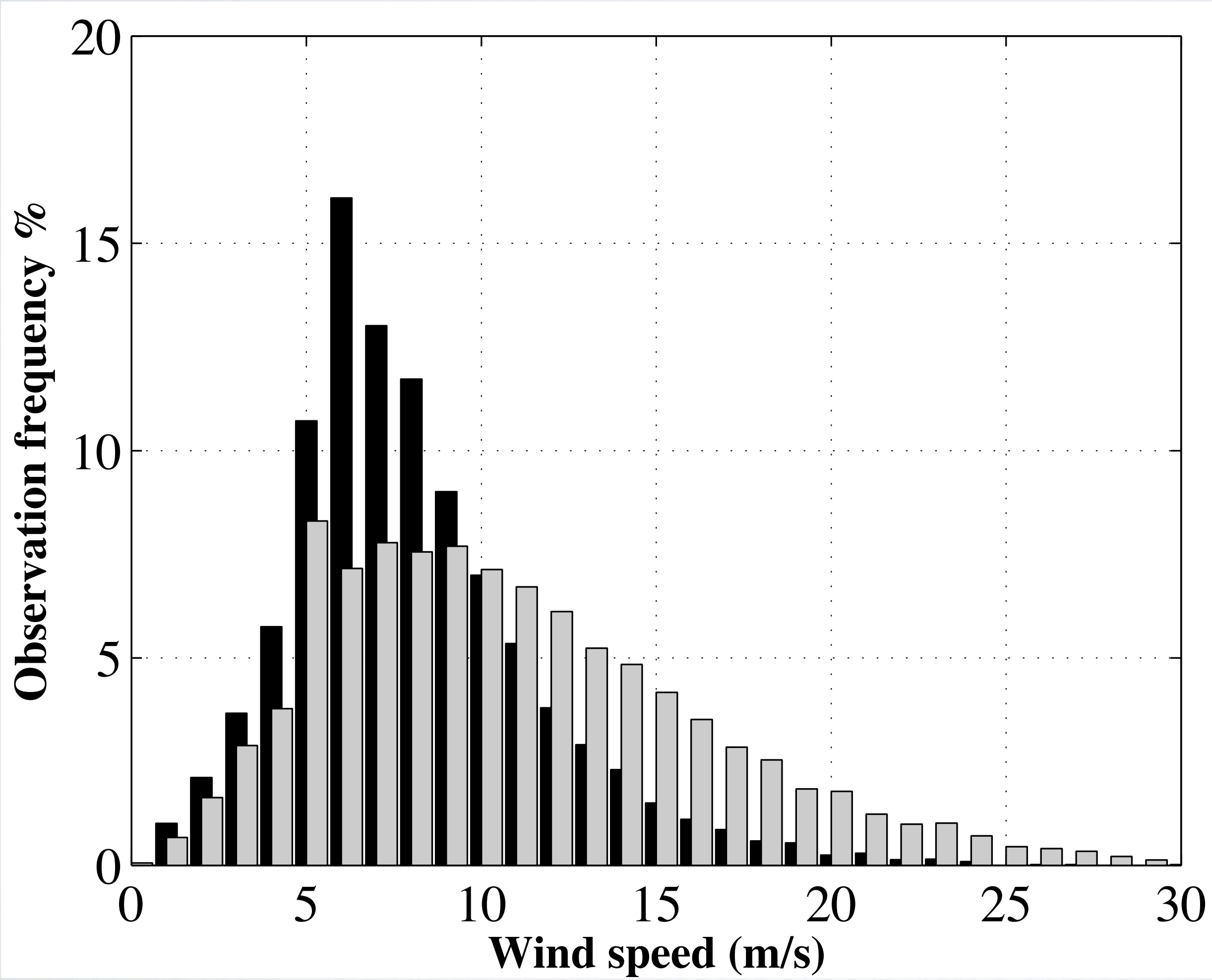
Wind power grows cubically with wind speed

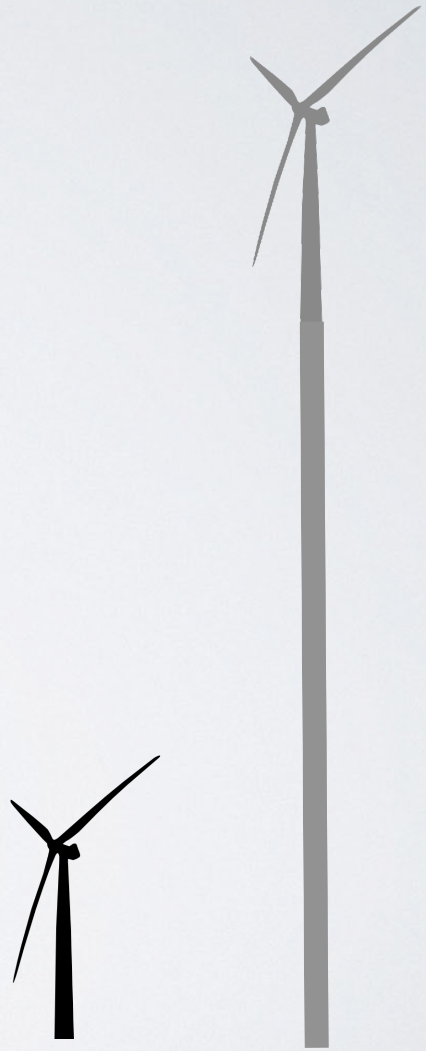


Doubling the wind speed leads to 8 x more power

Higher-up the wind is stronger

histogram of wind speeds at heights of 100m and 500m

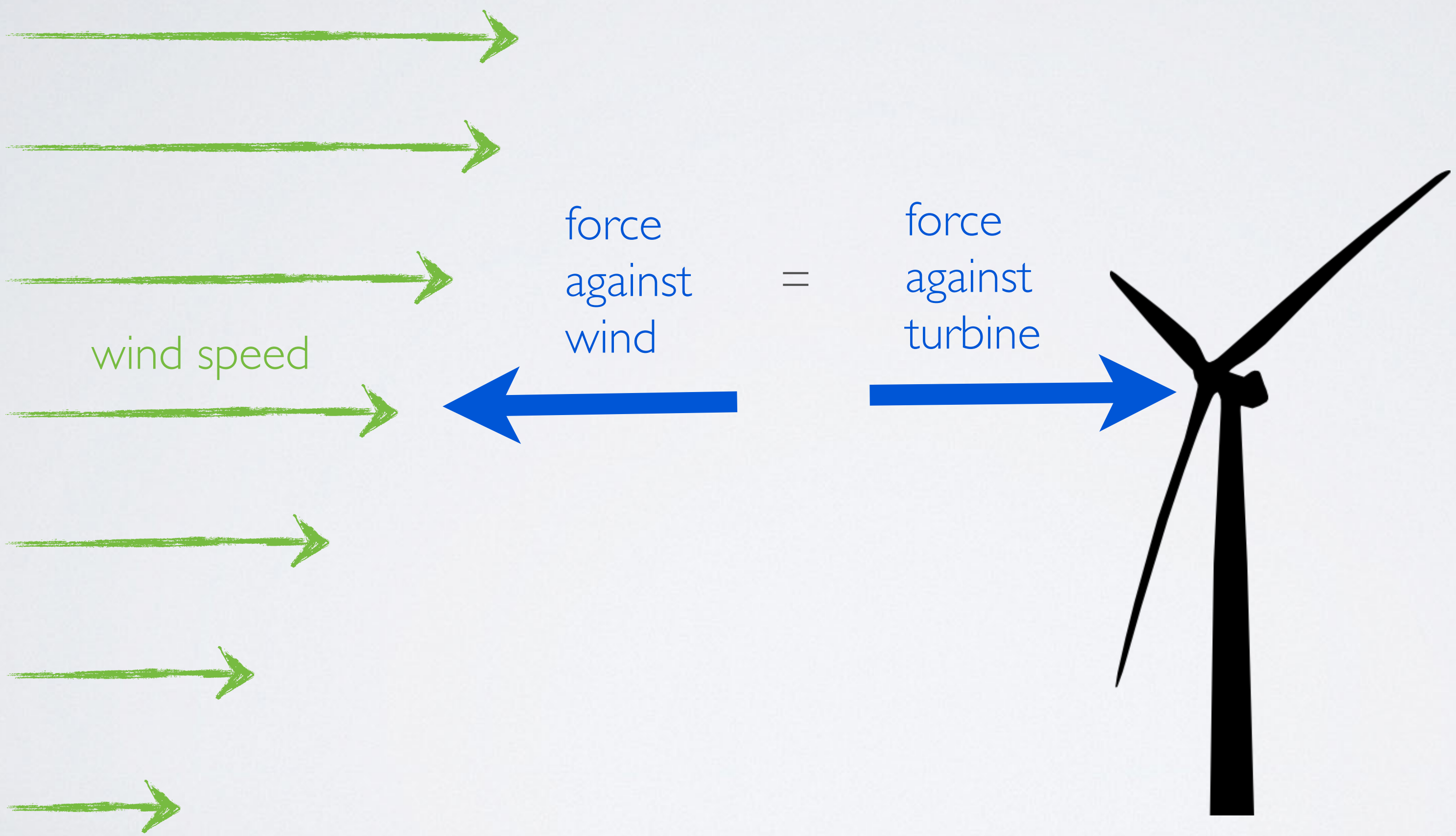




Wind speed in 100m [black] and 500m [grey] in De Bilt (NL)   [Fagiano 2009]

Power: Force x Wind Speed

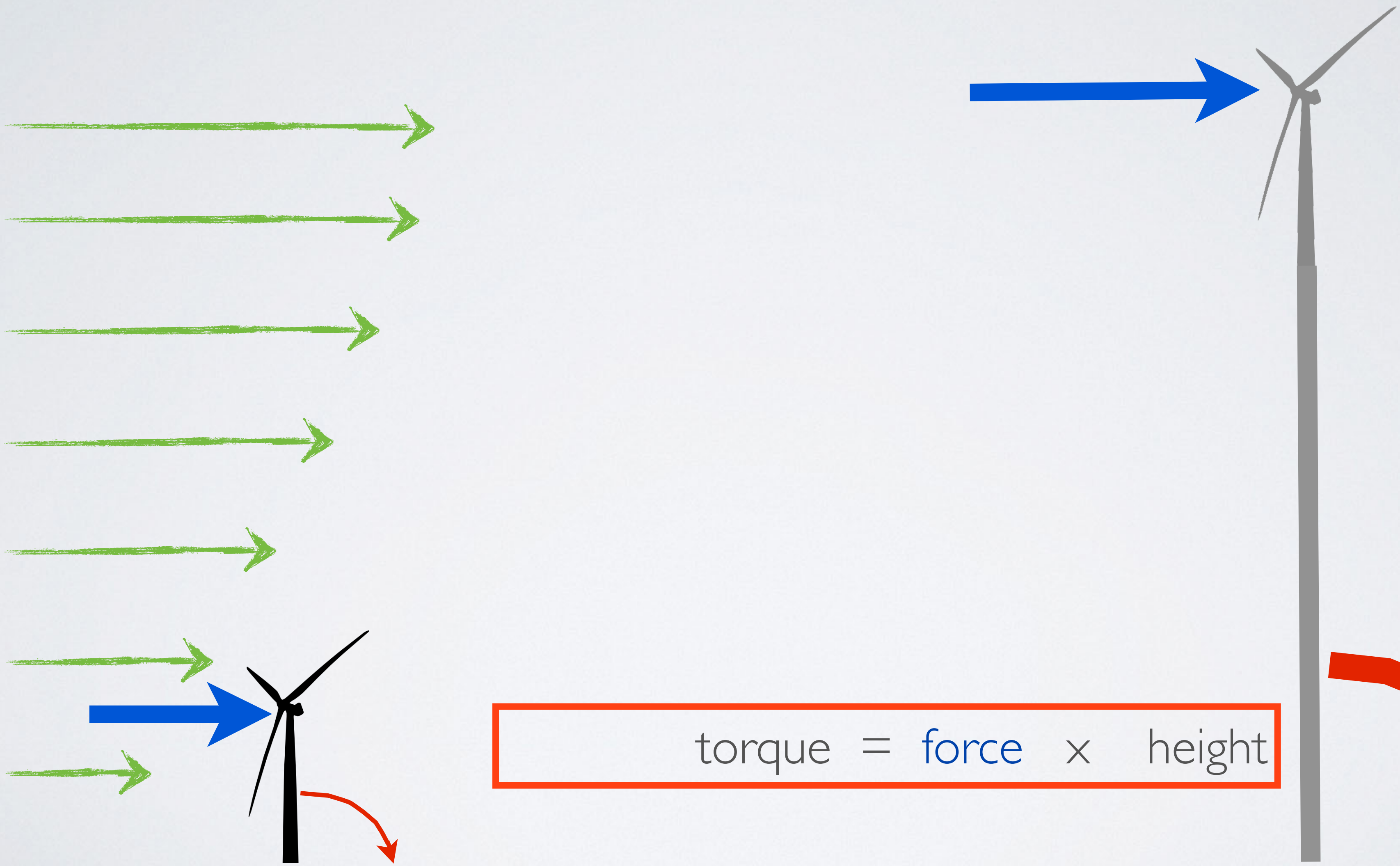
"No wind power without force against the wind"



Power extracted from the wind = Force x Wind Speed

A turbine of 500m height is difficult to build

Long lever arm leads to large torque

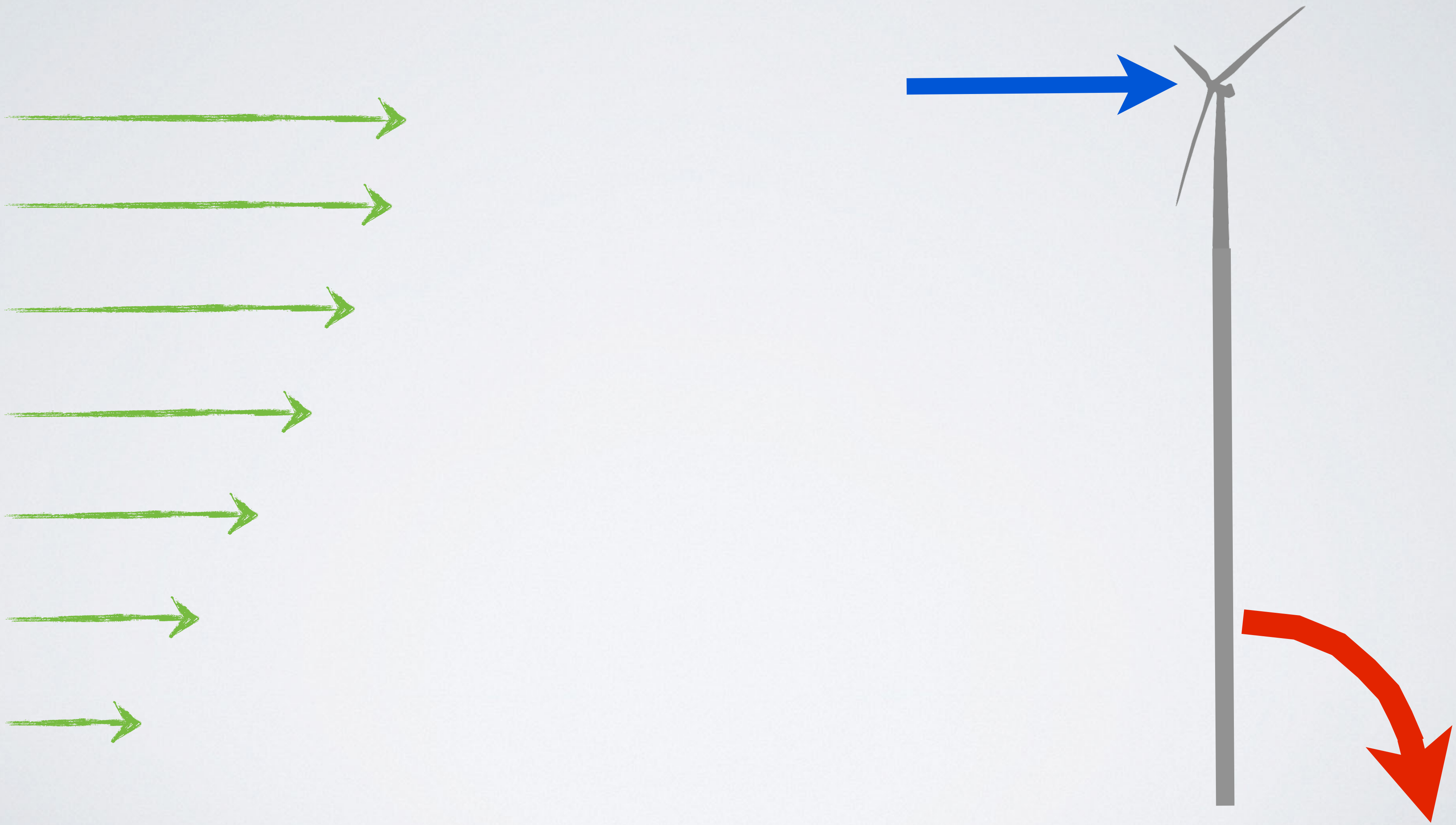


torque = force × height

Without force no wind energy can be harvested

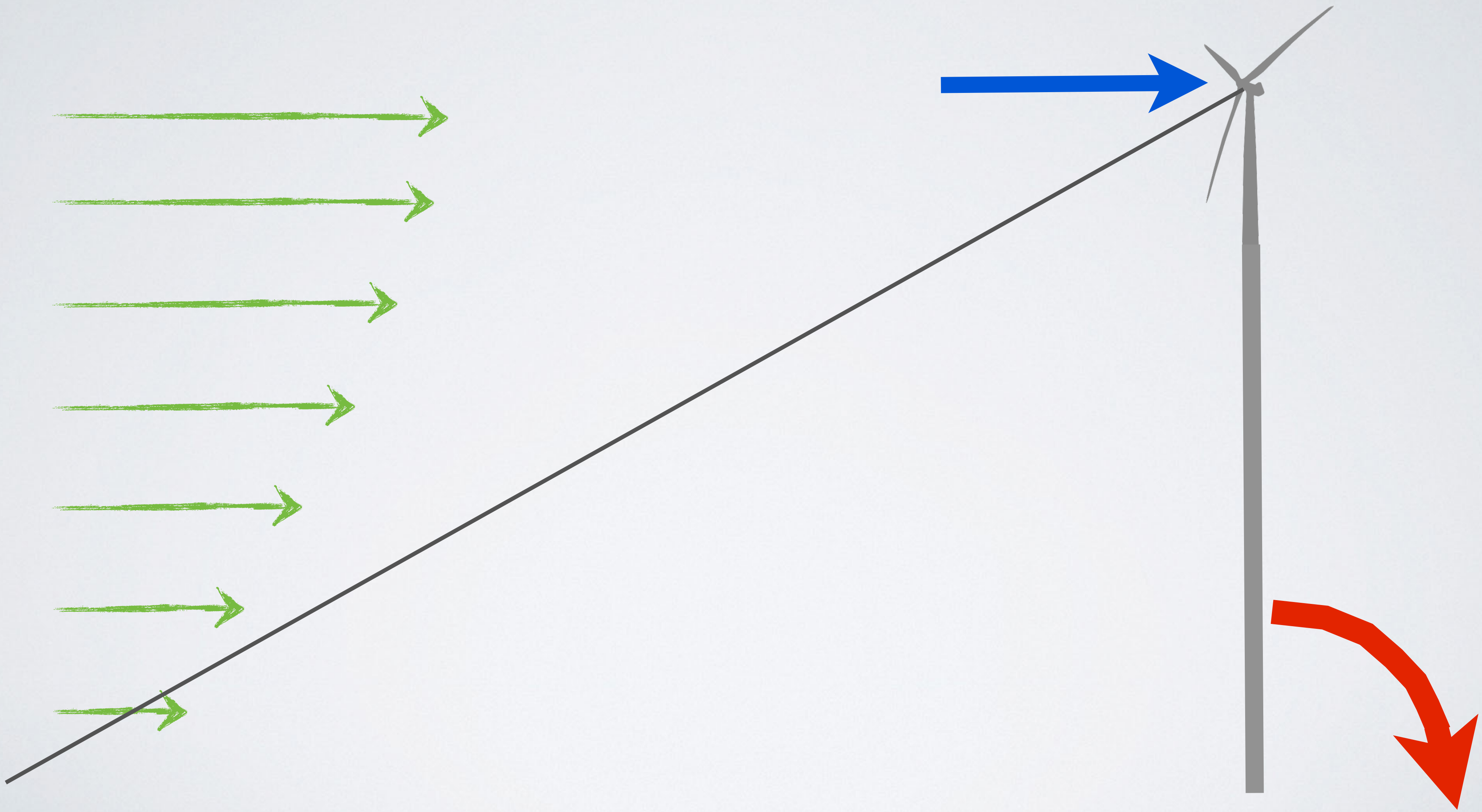
A turbine of 500m height is difficult to build

Long lever arm leads to large torque



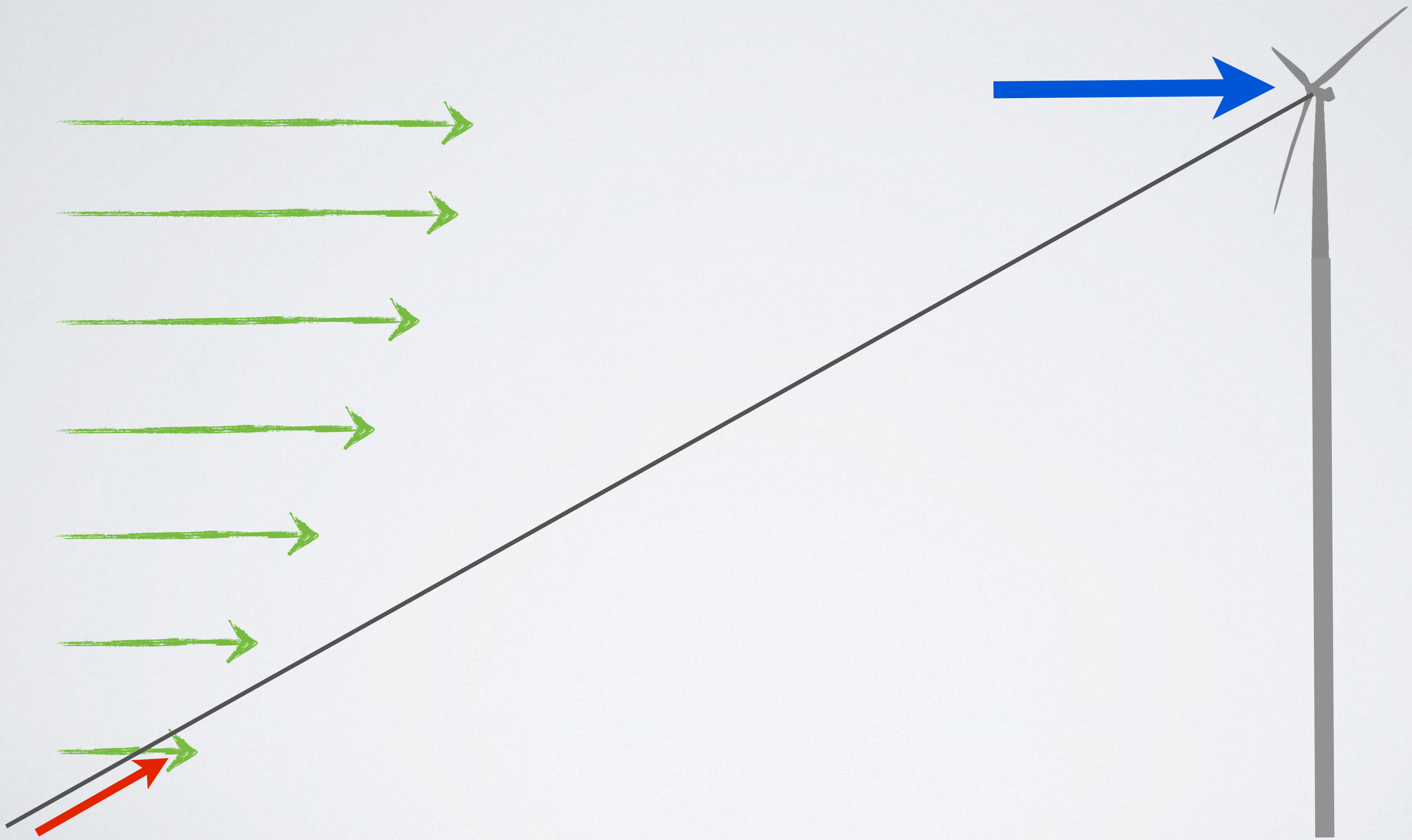
Without force no wind energy can be harvested

A turbine of 500m height is difficult to build



Without force no wind energy can be harvested

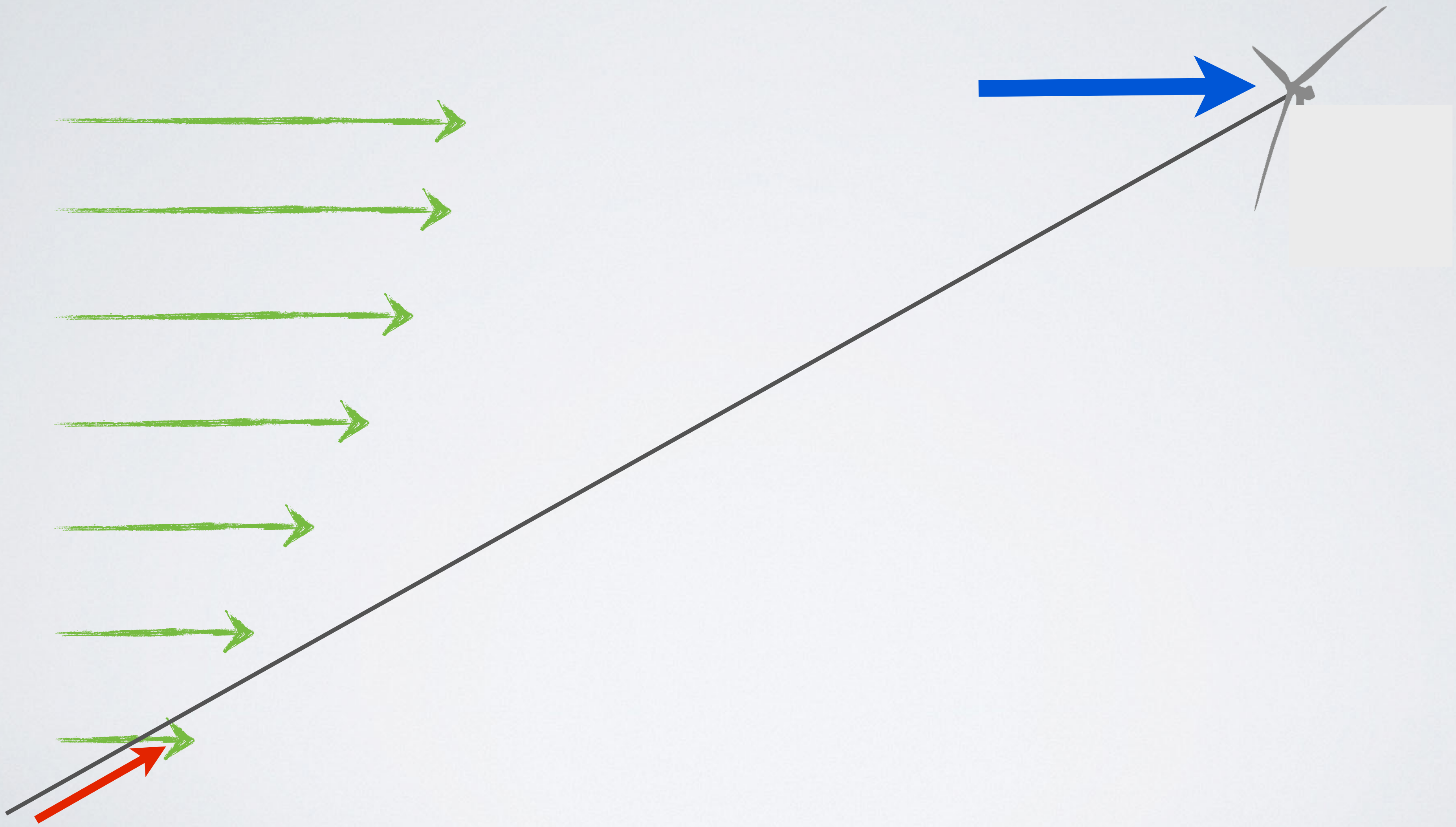
A turbine of 500m height is difficult to build



Without force no wind energy can be harvested

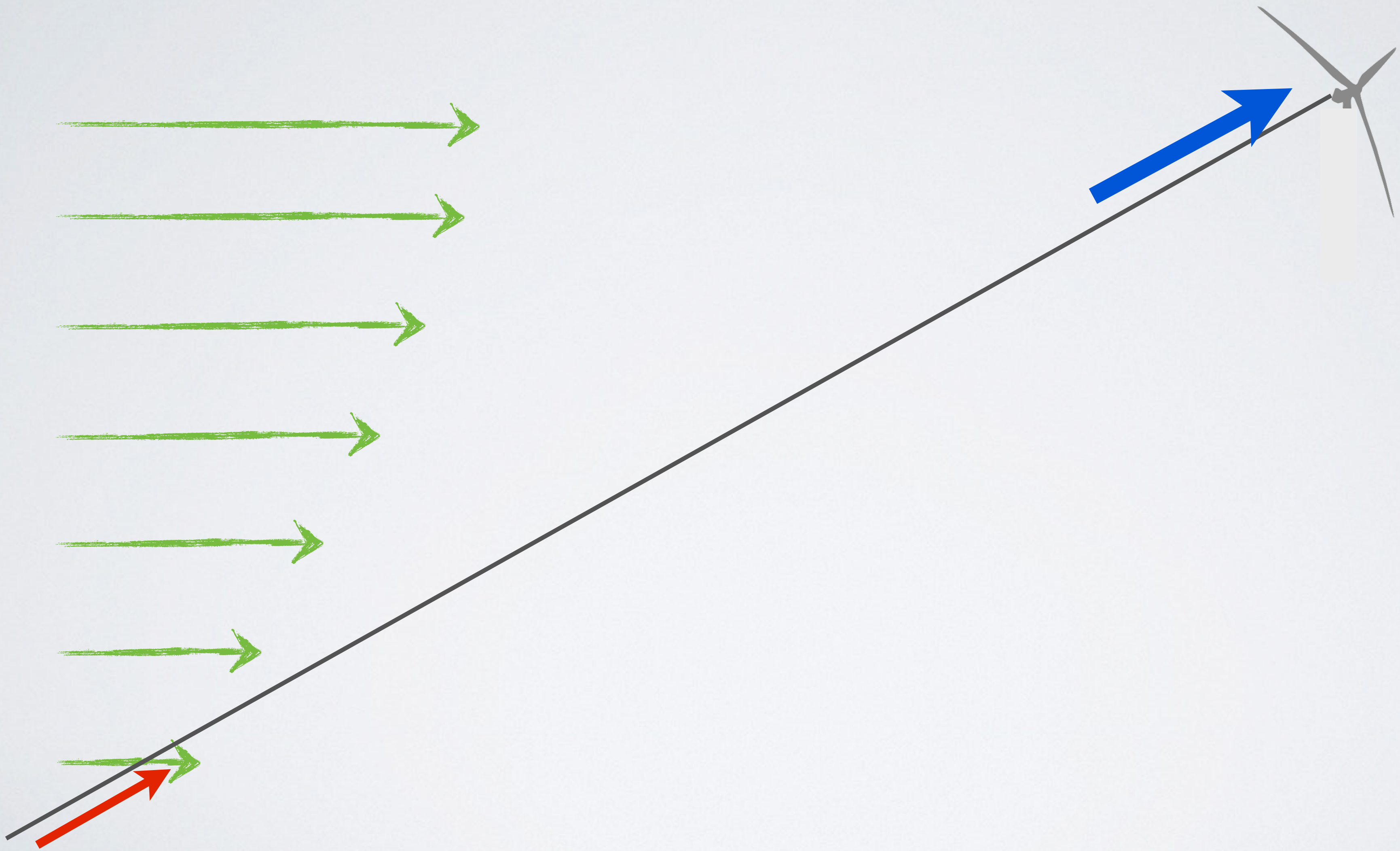
A turbine of 500m height is difficult to build

a cable can deliver the necessary force efficiently



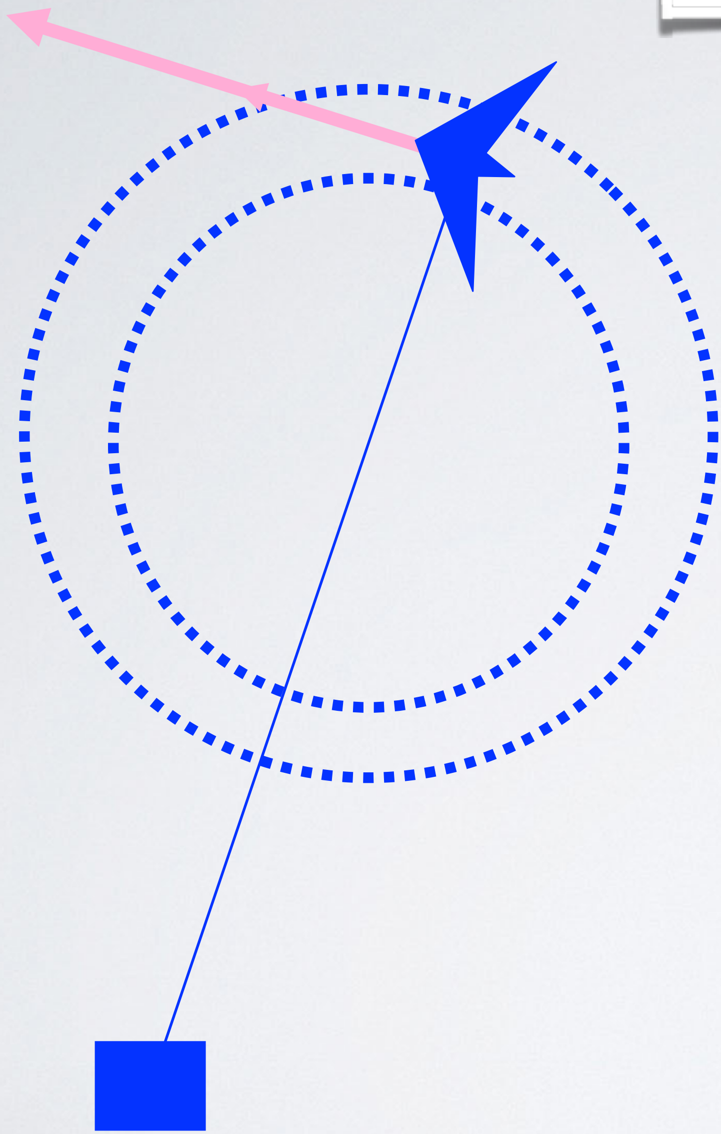
A turbine of 500m height is difficult to build

a cable can deliver the necessary force efficiently



Metamorphosis of a wind turbine

The Crosswind Effect

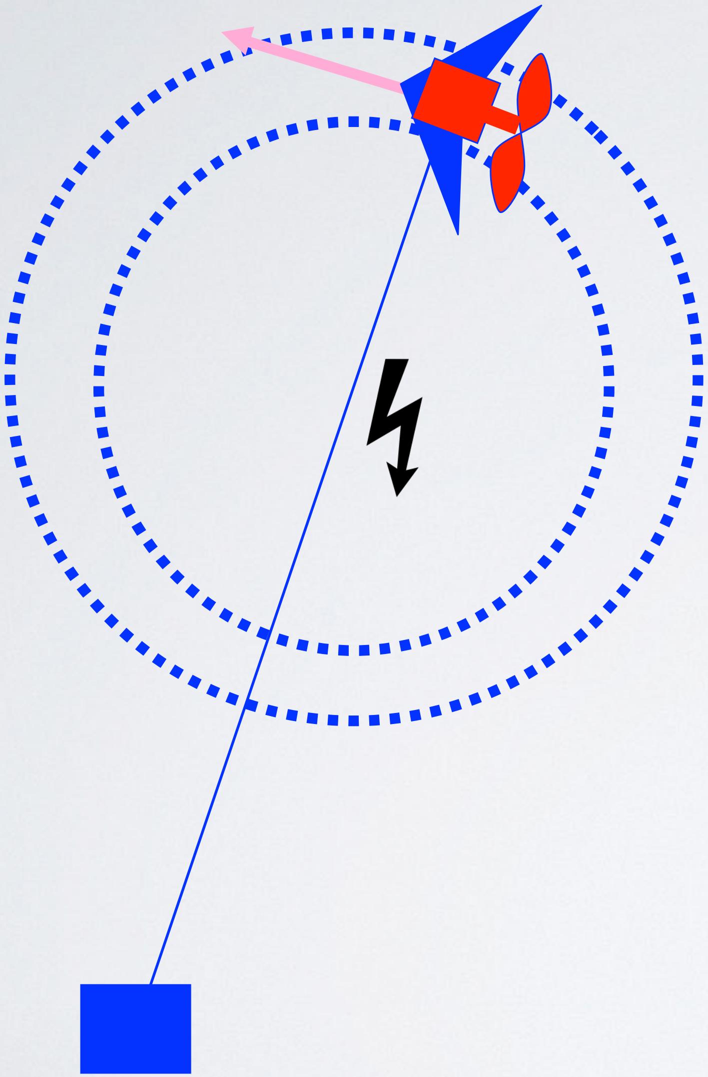


- wind lets kite fly fast circles
- kite flies **orthogonal to wind**
- strong tension force in tether, growing quadratically with kite speed

But where can we place a generator ?

("kite" = any flying object on a tether)

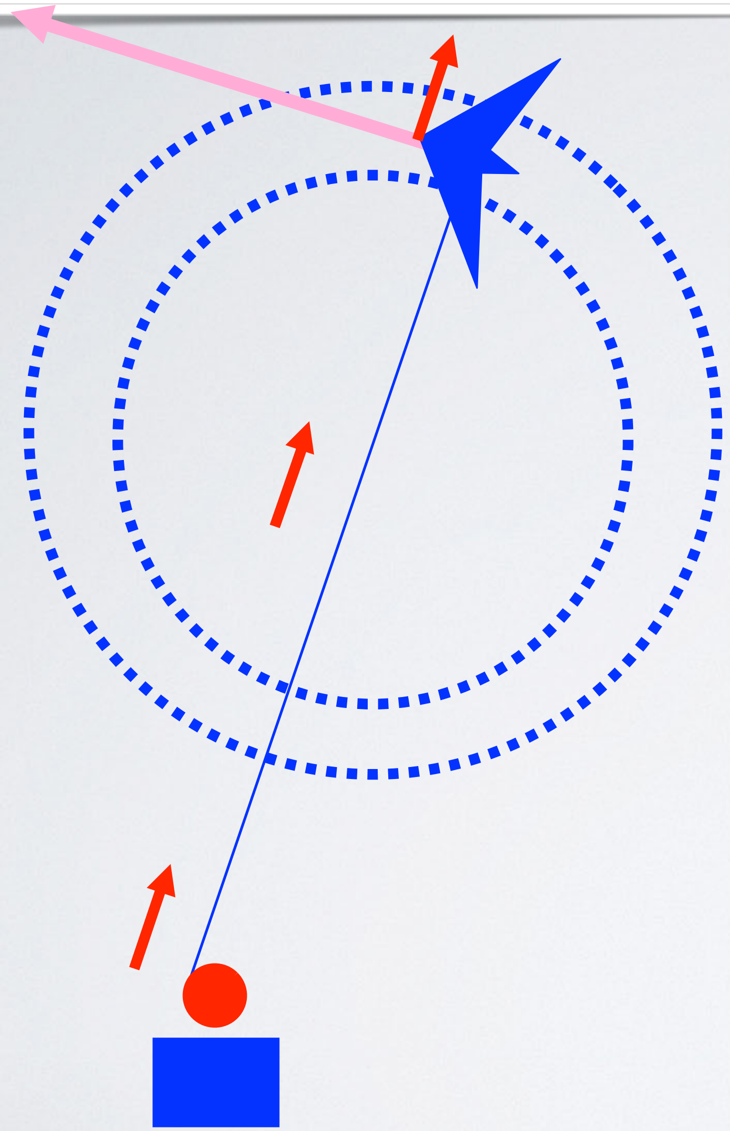
Variant 1: Drag Mode (airborne generation)



- relative wind drives small wind turbine on board of the kite
- cable transmits electricity to ground

Advantage: small, fast spinning generator
Disadvantage: high voltage cable needed

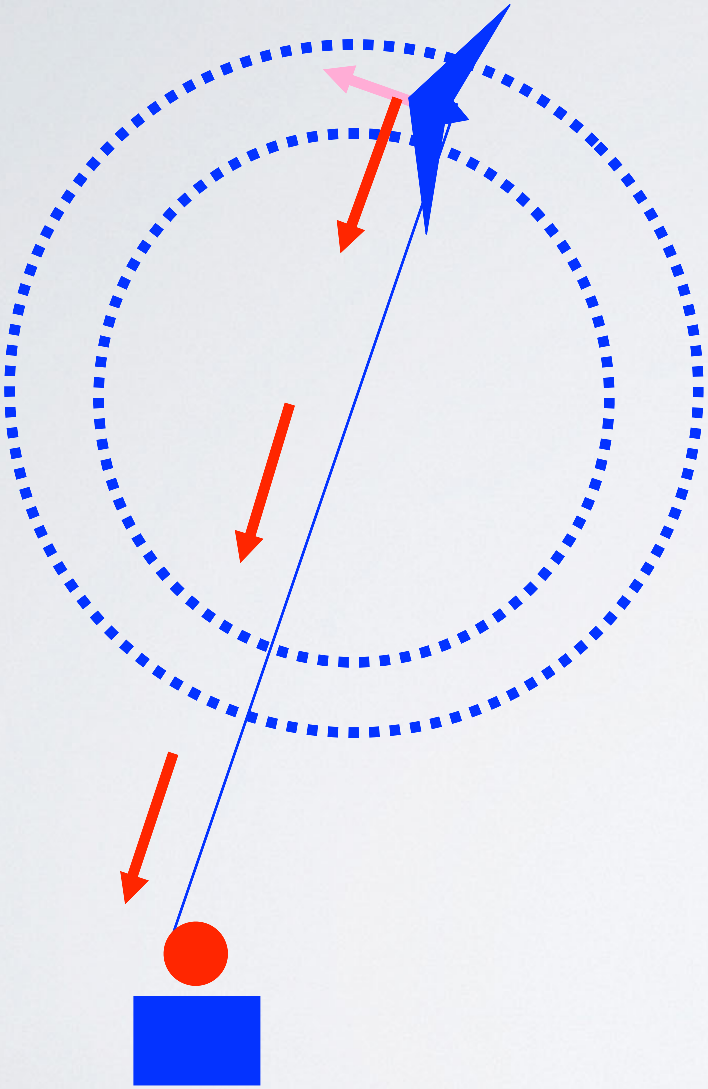
Variant 2: Lift Mode (ground-based generation)



Pumping cycle with two phases:

- Production phase:
    - fly kite fast, get high cable tension
    - unroll cable from drum
    - drum drives electric generator

Variant 2: Lift Mode (ground-based generation)



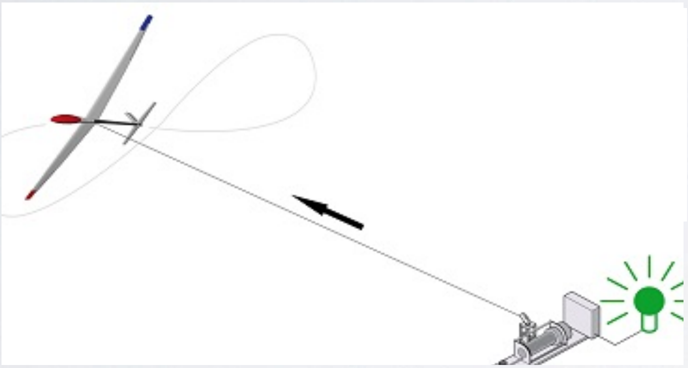
Pumping cycle with two phases:

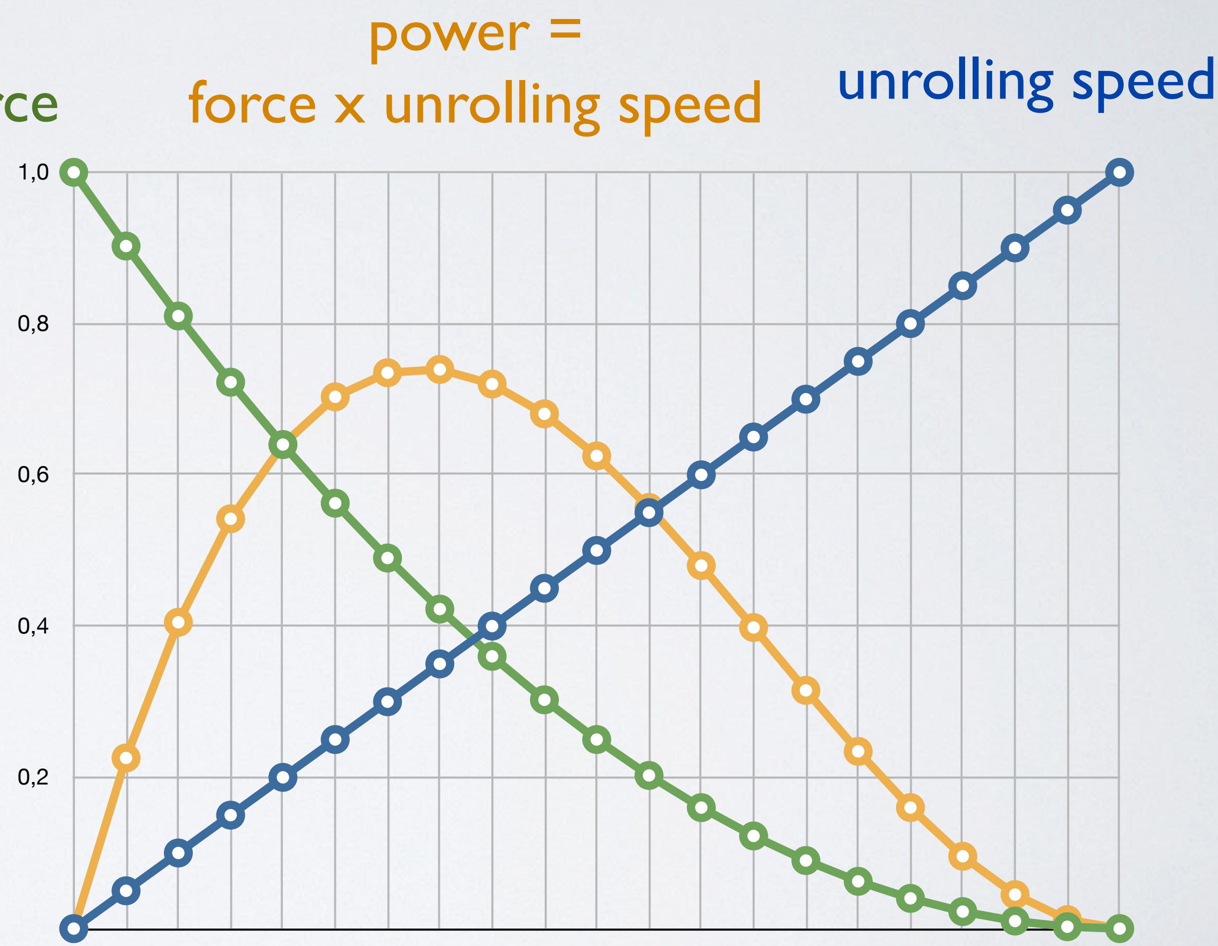
- Production phase:
  - fly kite fast, get high cable tension
  - unroll cable from drum
  - drum drives electric generator
- Retraction phase:
  - fly kite slow, get low cable tension
  - roll-in cable

Advantage: electric machine on ground
Disadvantage: slow, heavy generator needed
(like conventional wind turbines)

Which cable unrolling speed is optimal ?

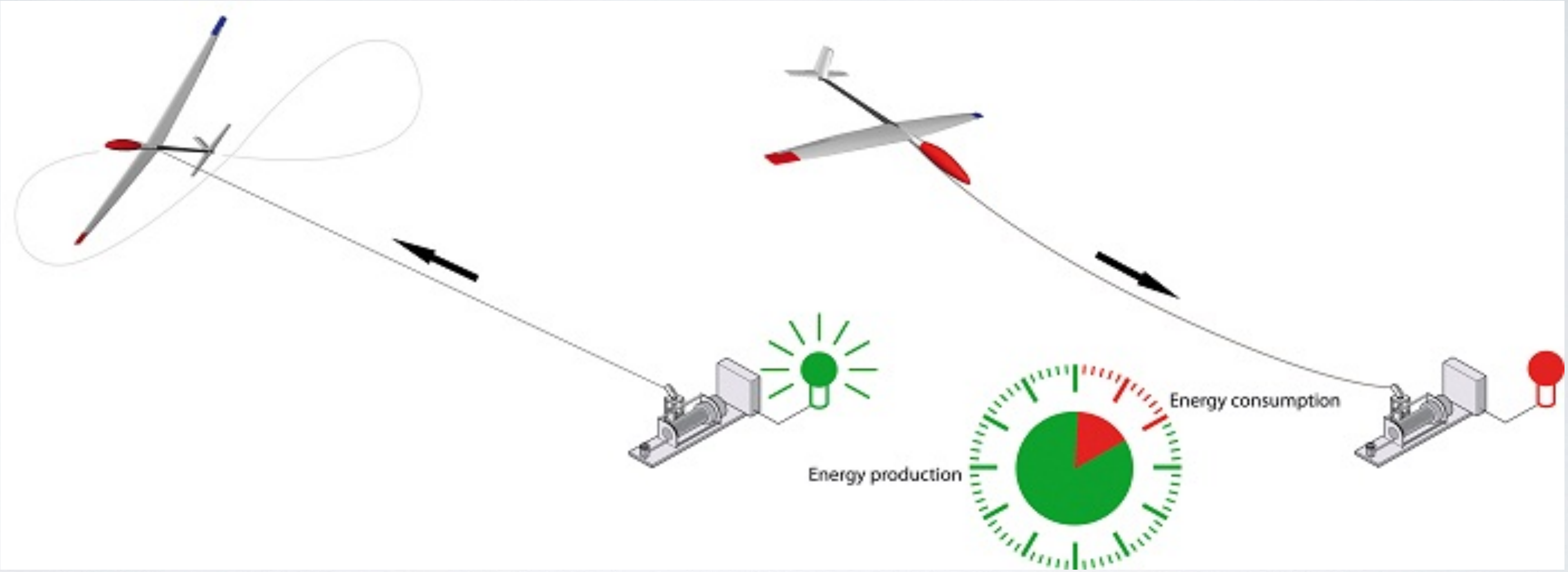
Maximal power reached at 1/3 of wind speed



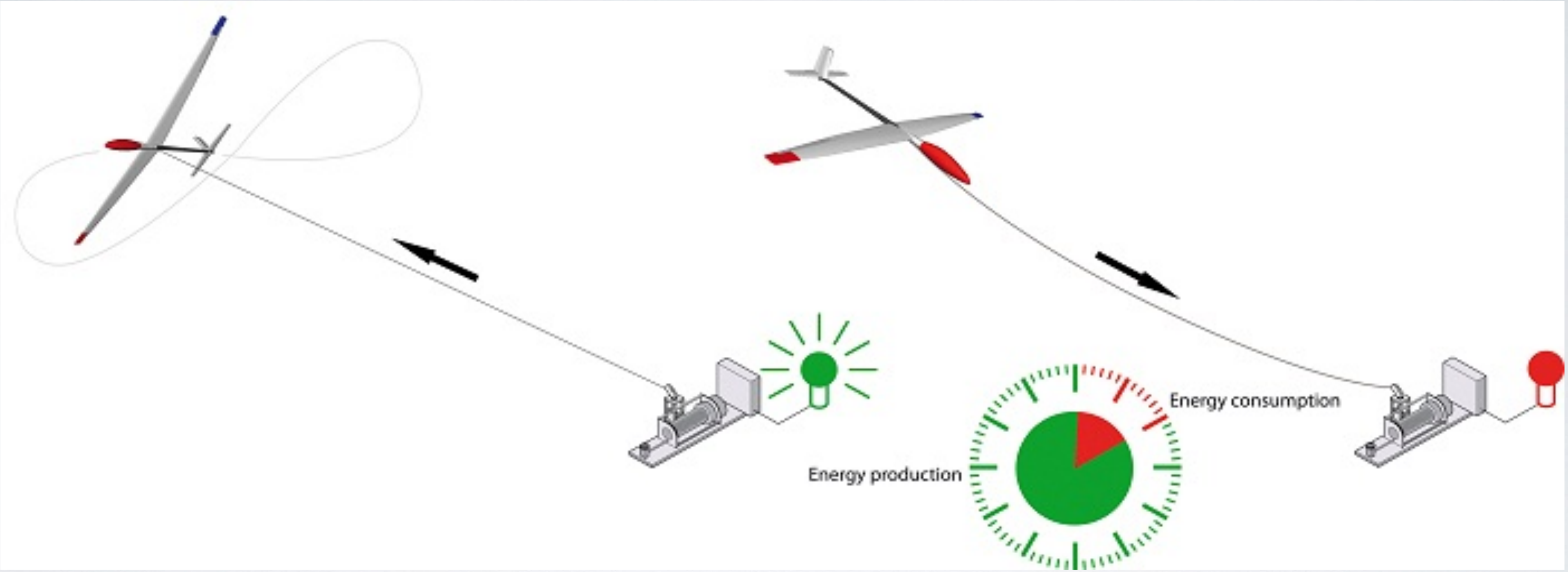


Note: kite flies much faster than wind speed, depending on its aerodynamic efficiency. The faster, the more tension force in the line.

Pumping Cycle to Harvest Wind Power



Pumping Cycle to Harvest Wind Power



Which cable unrolling speed is optimal ?

too slow: generator does not turn

too fast: kite "sees" relatively less wind, cable tension too small

optimum: 1/3 of wind speed (note: plane flies much faster)

[AmpyxPower]

Loyd's Formula

J. ENERGY

VOL. 4, NO. 3

ARTICLE NO. 80-4075

Crosswind Kite Power

Miles L. Loyd*

Lawrence Livermore National Laboratory, Livermore, Calif.

$$P = \frac{2}{27} \rho A w^3 C_L \left(\frac{C_L}{C_D}\right)^2$$

power P

air density ρ

wing area A

wind speed w

Lift-over-drag ratio (L/D) $\left(\frac{C_L}{C_D}\right)$

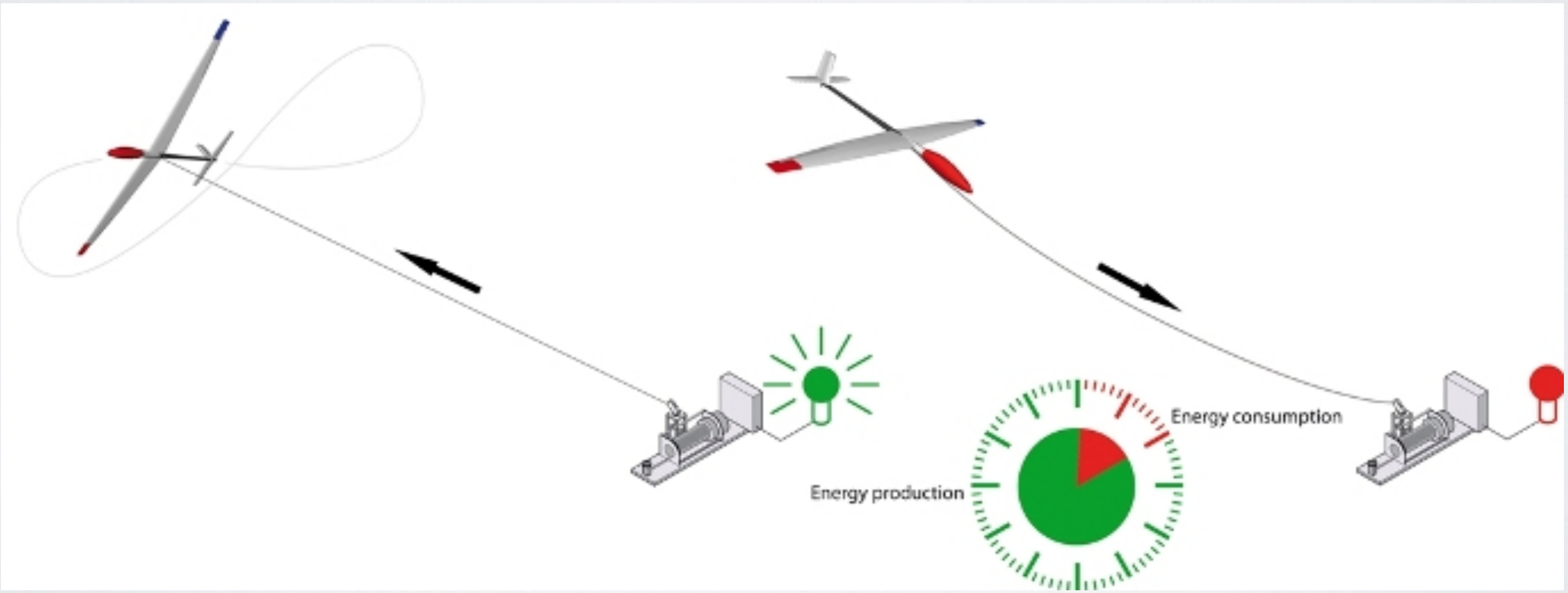
wing area of **1 m^2** generates **40 kW** power (at 13 m/s wind speed and L/D of 15). Same efficiency for drag and lift mode.

Last September in Berlin.

The Company AmpyxPower

AmpyxPower

- startup from TU Delft, since 2008
- 10 permanent staff
- financed via venture capital, KLM, statkraft, ...



AmpyxPower:
Autonomous Energy Harvesting Flight

MODEL BASED CONTROL OF TETHERED AIRPLANES (LEUVEN / FREIBURG)

Differential Algebraic Equation (DAE) Model of Tethered Airplane

Translational: $$\begin{bmatrix} m & 0 & 0 & x \\ 0 & m & 0 & y \\ 0 & 0 & m & z \\ x & y & z & 0 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \lambda \end{bmatrix} = \begin{bmatrix} F_x + m\left(\dot{\delta}^2 r_A + \dot{\delta}^2 x + 2\dot{\delta}\dot{y} + \ddot{\delta} y\right) \\ F_y + m\left(y\dot{\delta}^2 - 2\dot{x}\dot{\delta} - \ddot{\delta}(rA + x)\right) \\ F_z - gm \\ -\dot{x}^2 - \dot{y}^2 - \dot{z}^2 \end{bmatrix}$$

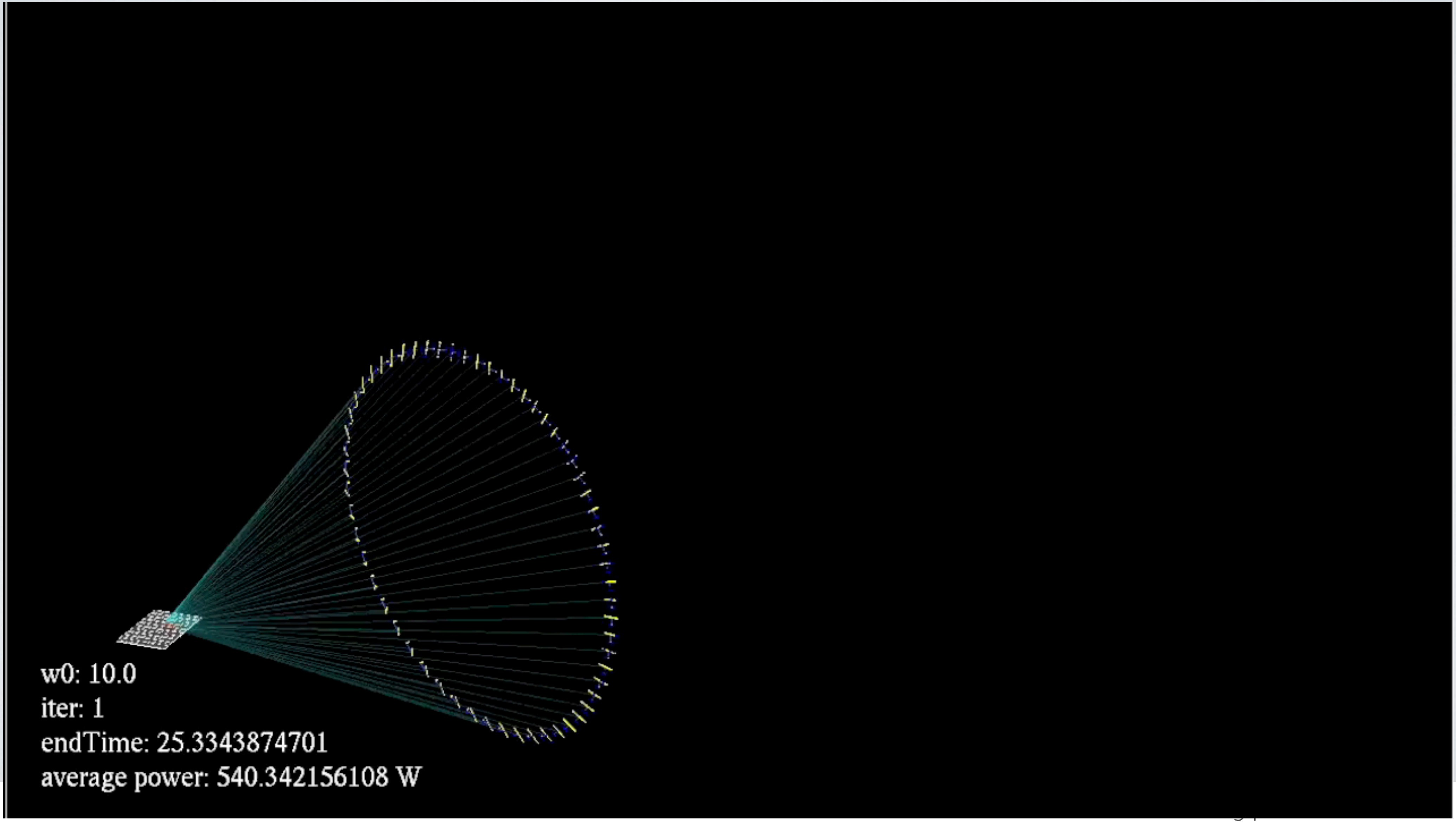
Rotational: $$\dot{R} = R\omega_\times - R^T \begin{bmatrix} 0 \\ 0 \\ \dot{\delta} \end{bmatrix}, \qquad J\dot{\omega} = T - \omega \times J\omega, \qquad R = \begin{bmatrix} \vec{E}_x & \vec{E}_y & \vec{E}_z \end{bmatrix}$$

Aero. coefficients: $$\vec{v} = \begin{bmatrix} \dot{x} - \dot{\delta} y \\ \dot{y} + \dot{\delta}(r_\mathrm{A} + x) \\ \dot{z} \end{bmatrix} - \vec{w}(x, y, z, \delta, t), \qquad \alpha = -\frac{\vec{E}_z^T \vec{v}}{\vec{E}_x^T \vec{v}}, \qquad \beta = \frac{\vec{E}_y^T \vec{v}}{\vec{E}_x^T \vec{v}}$$

Aero. forces/torques: $$\vec{F}_\mathrm{A} = \frac{1}{2}\rho A \|\vec{v}\| (C_\mathrm{L} \vec{v} \times \vec{E}_y - C_\mathrm{D} \vec{v}), \quad \vec{T}_\mathrm{A} = \frac{1}{2}\rho A \|\vec{v}\|^2 \begin{bmatrix} C_\mathrm{R} \\ C_\mathrm{P} \\ C_\mathrm{Y} \end{bmatrix}$$

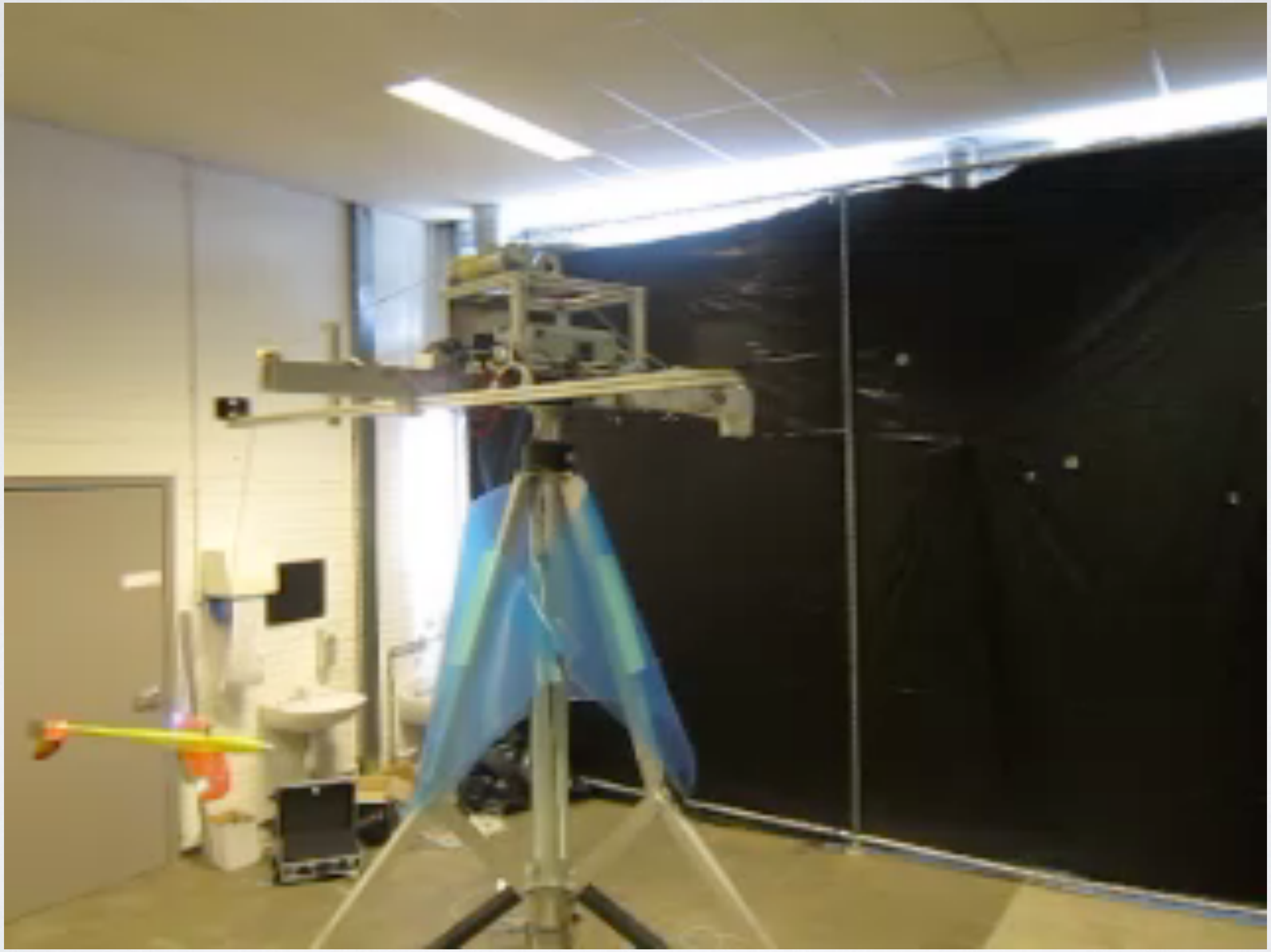
Computing a Power Optimal Orbit

optimization algorithm at work [Greg Horn]



Experiments with Predictive Flight Control

[mit Milan Vukov, Kurt Geebelen, Andrew Wagner, Mario Zanon, Sebastien Gros, Greg Horn, Jan Swevers]



Aim: Transition from Rotation to Power Orbit

Model Predictive Control in Practice

2x EMBEDDED OPTIMIZATION

SENSORS

- acceleration
- gyros
- vision
- ...

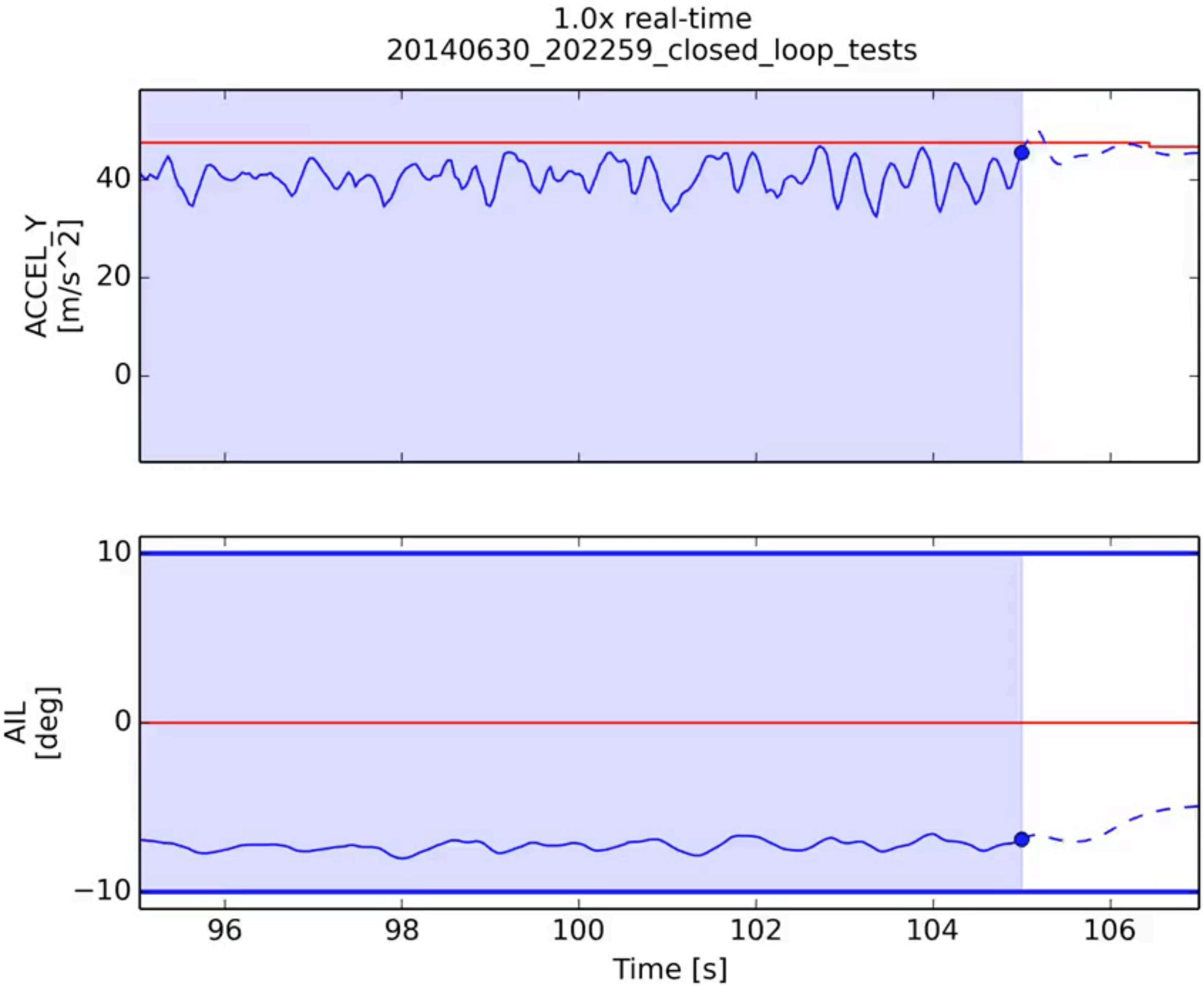
STATE-ESTIMATION
Data Fitting on Moving Horizon

MODEL PREDICTIVE CONTROL

ACTUATORS

- elevator
- aileron
- winch motor

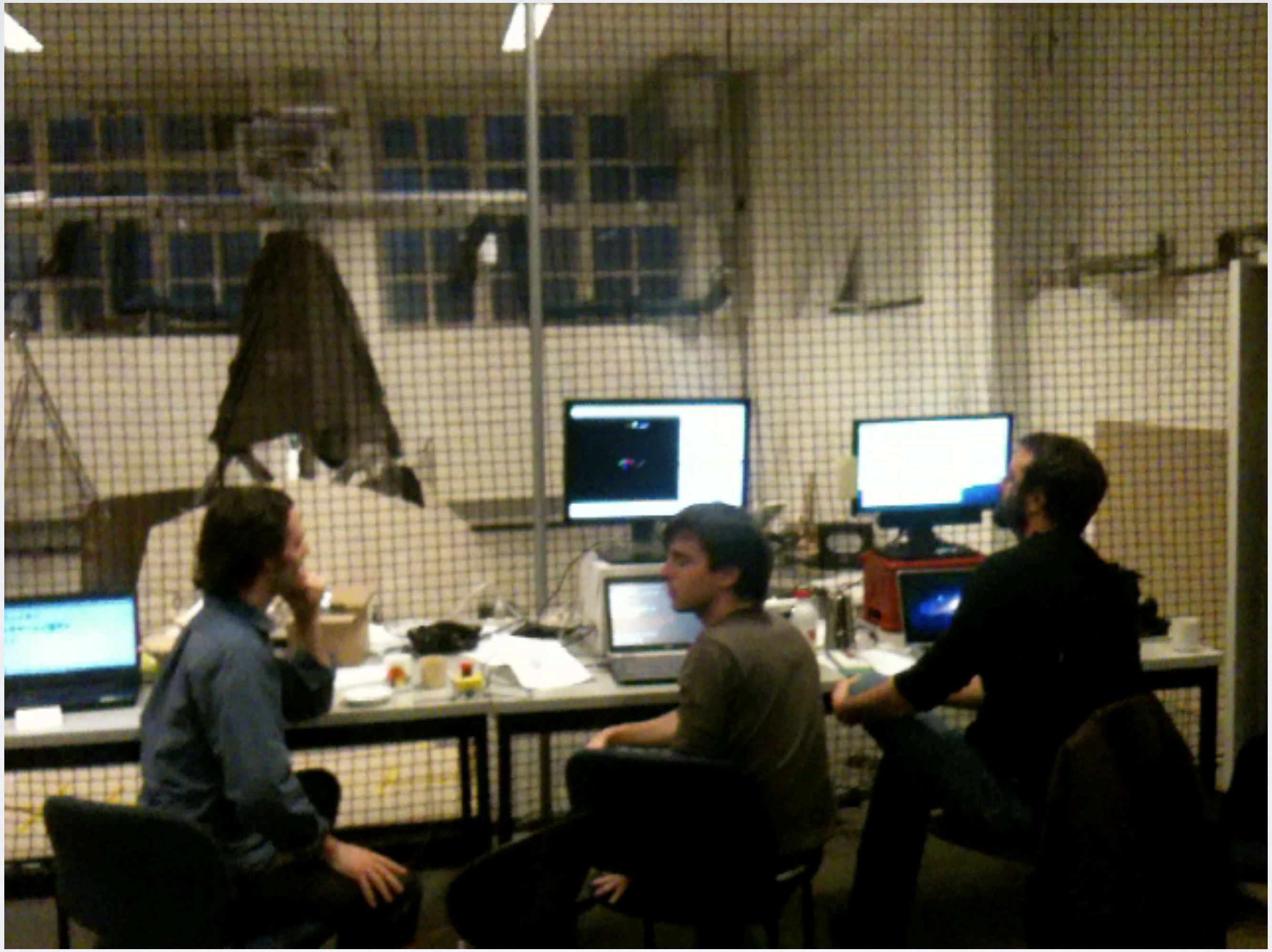
Prediction & Estimation Horizons in Flight Experiment (ACADO)



What do we see here on the slide, in one sentence

Experimente mit Moving Horizon Estimation

subtitle



SUMMARY

- embedded optimization uses more CPU time than classical filters, but allows the development of more powerful nonlinear control and estimation algorithms

- examples are time-optimal or power-optimal model predictive control (of cranes, robots, cars, tethered airplanes…)

- good numerical methods can solve nonlinear optimal control problems at millisecond sampling rates

- need accurate and differentiable ODE or DAE models

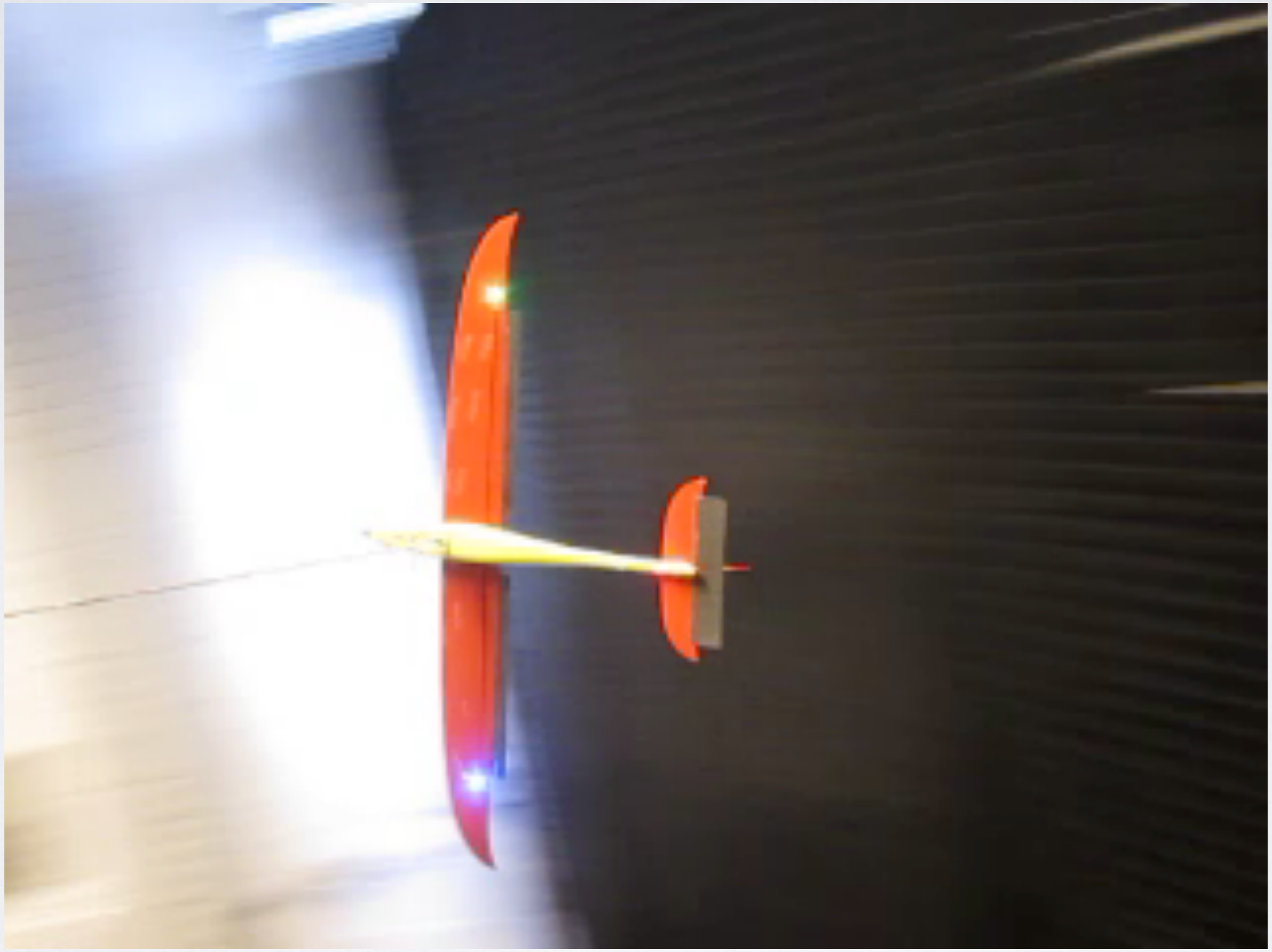
Thank you

Our Vision: replace tons of steel and concrete...

Our Vision: replace tons of steel and concrete...
...by a cable and intelligent control

Laborexperimente mit prädiktiver Flugregelung

[mit Kurt Geebelen, Andrew Wagner, Milan Vukov, Mario Zanon, Sebastien Gros, Greg Horn, Jan Swevers]



Outdoors Carousel Tests