Sequential Convex Programming

Moritz Diehl

University of Freiburg

Real-Time Sequential Convex Programming

Moritz Diehl

Joint work with Quoc Tran Dinh and Carlo Savorgnan

Optimization in Engineering Center OPTEC &

Electrical Engineering Department ESAT

KU Leuven, Belgium

KTH Stockholm, Oct 12, 2012

Overview

- Optimization in Engineering Center OPTEC
- Sequential Convex Programming
- Real-Time Sequential Convex Programming
- [Open Source Software ACADO]
- [Experiments with Tethered Airplanes]

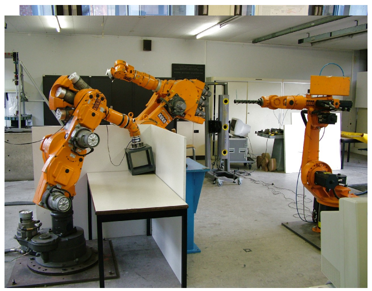
OPTEC - Optimization in Engineering Center

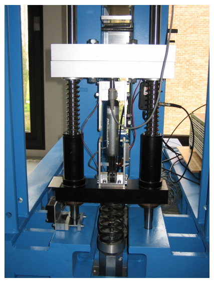
Center of Excellence of KU Leuven*, since 2005*

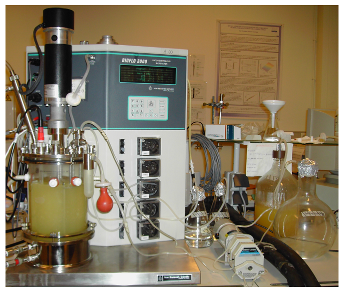
70 people, working jointly on ***methods and applications of optimization,*** *in 5 departments:*

- Electrical Engineering
- Mechanical Engineering
- Chemical Engineering
- Computer Science
- Civil Engineering

Many real world applications at OPTEC...







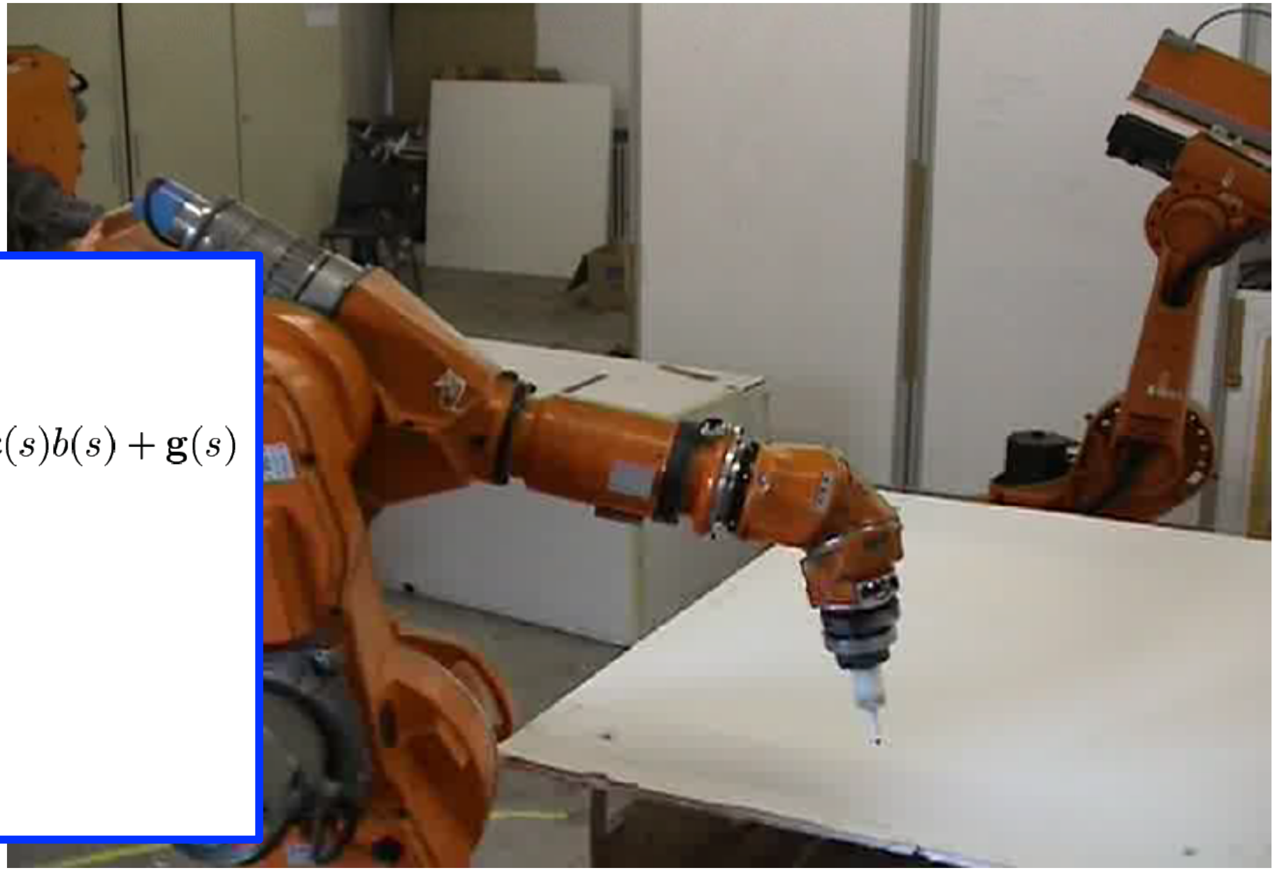
OPTEC Research Example: Time Optimal Robot Motion

Robot shall write as fast as possible.
Global solution found in 2 ms due to ***convex reformulation***

IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 54, NO. 10, OCTOBER 2009

Time-Optimal Path Tracking for Robots: A Convex Optimization Approach

Diederik Verscheure, Bram Demeulenaere, Jan Swevers, Joris De Schutter, and Moritz Diehl



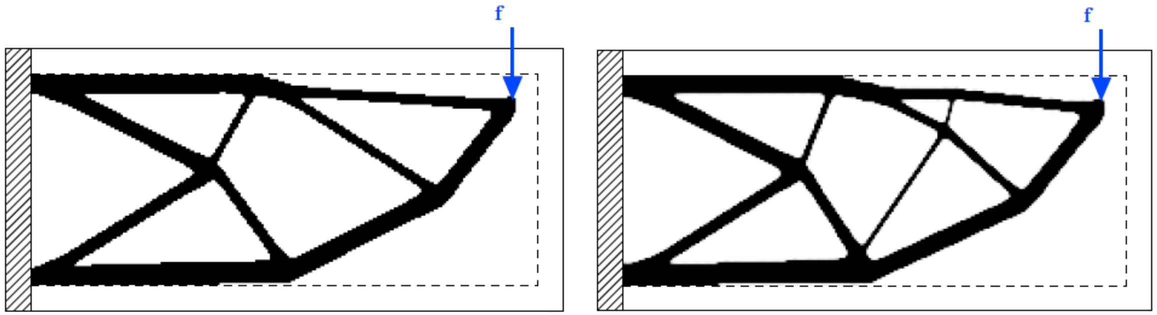
$$
\begin{aligned}
&\min_{a(\cdot),b(\cdot),\boldsymbol{\tau}(\cdot)} \quad \int_0^1 \frac{1}{\sqrt{b(s)}} ds \\
&\text{subject to} \quad \boldsymbol{\tau}(s) = \mathbf{m}(s)a(s) + \mathbf{c}(s)b(s) + \mathbf{g}(s) \\
&\qquad b(0) = \dot{s}_0^2 \\
&\qquad b(1) = \dot{s}_T^2 \\
&\qquad b'(s) = 2a(s) \\
&\qquad b(s) \geq 0 \\
&\qquad \underline{\boldsymbol{\tau}}(s) \leq \boldsymbol{\tau}(s) \leq \overline{\boldsymbol{\tau}}(s) \\
&\qquad \text{for } s \in [0,1].
\end{aligned}
$$

OPTEC Research Example: Topology Optimization

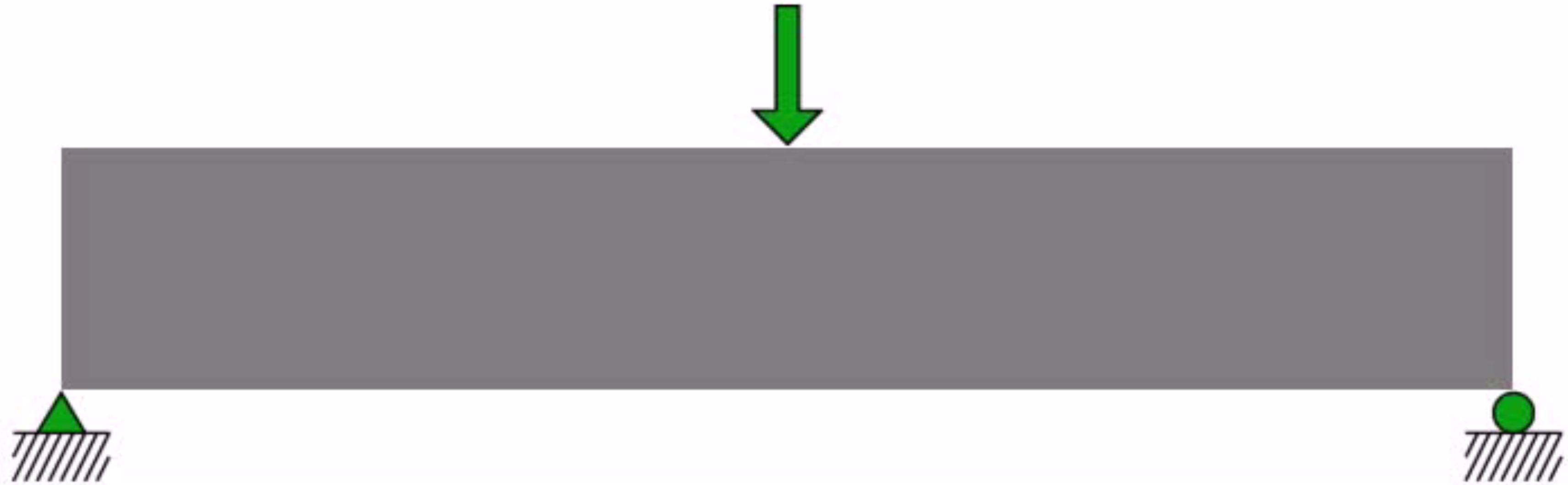
Miche Jansen
with 3 supervisors
from 3 departments

Robust topology optimization accounting for misplacement of material

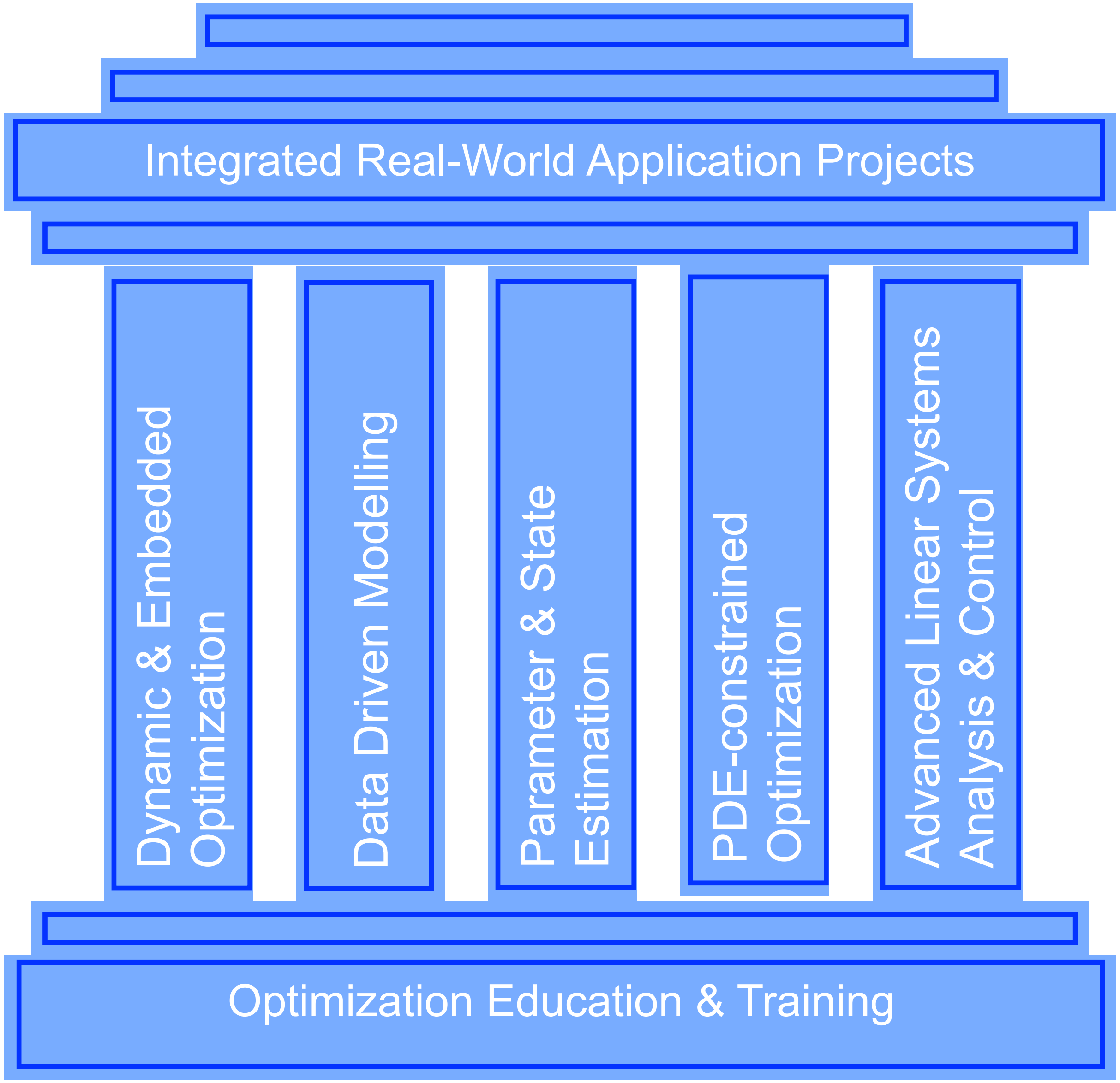
Miche Jansen · Geert Lombaert · Moritz Diehl · Boyan S. Lazarov ·
Ole Sigmund · Mattias Schevenels



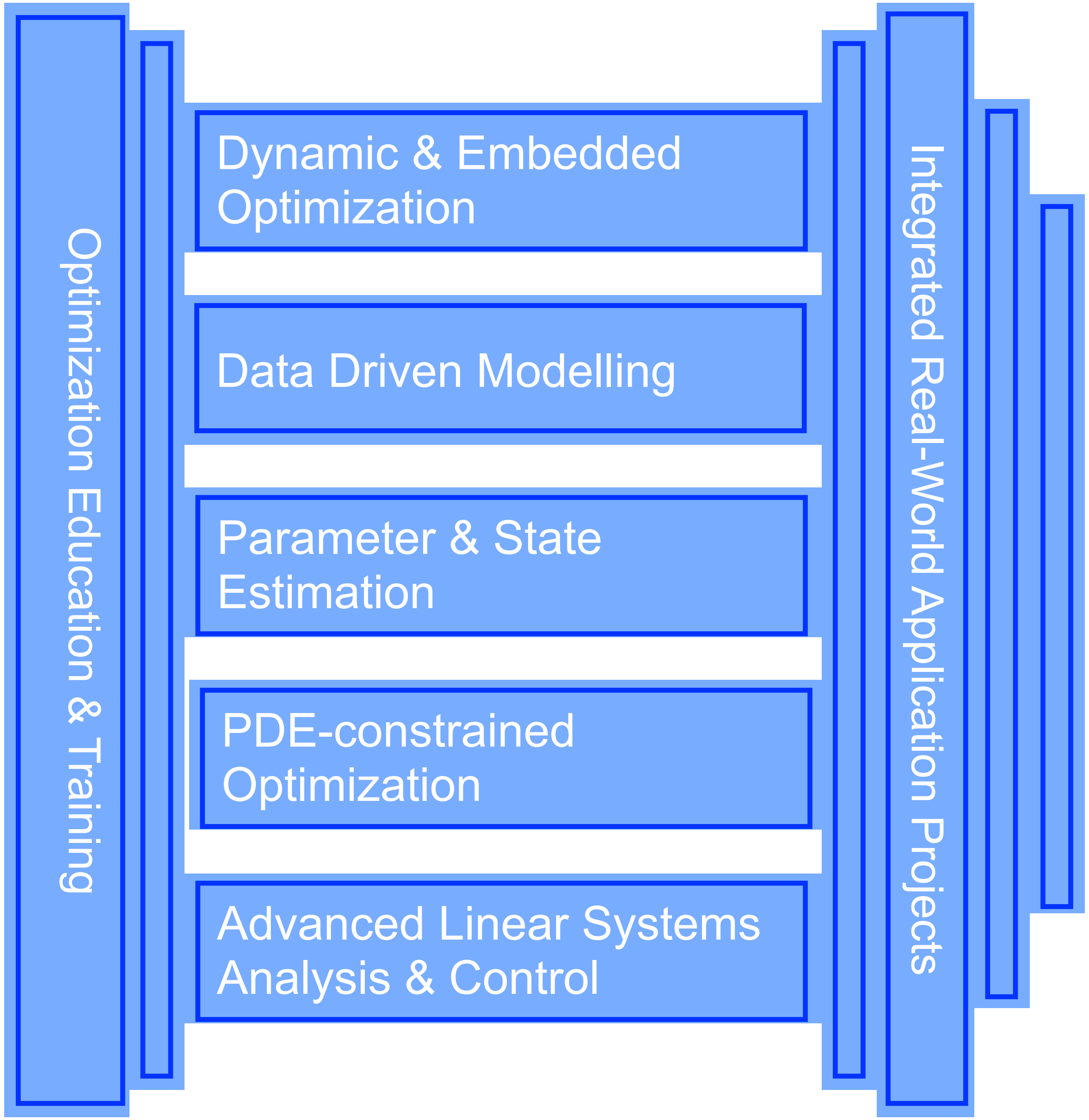
Deterministic optimized design   Robust optimized design



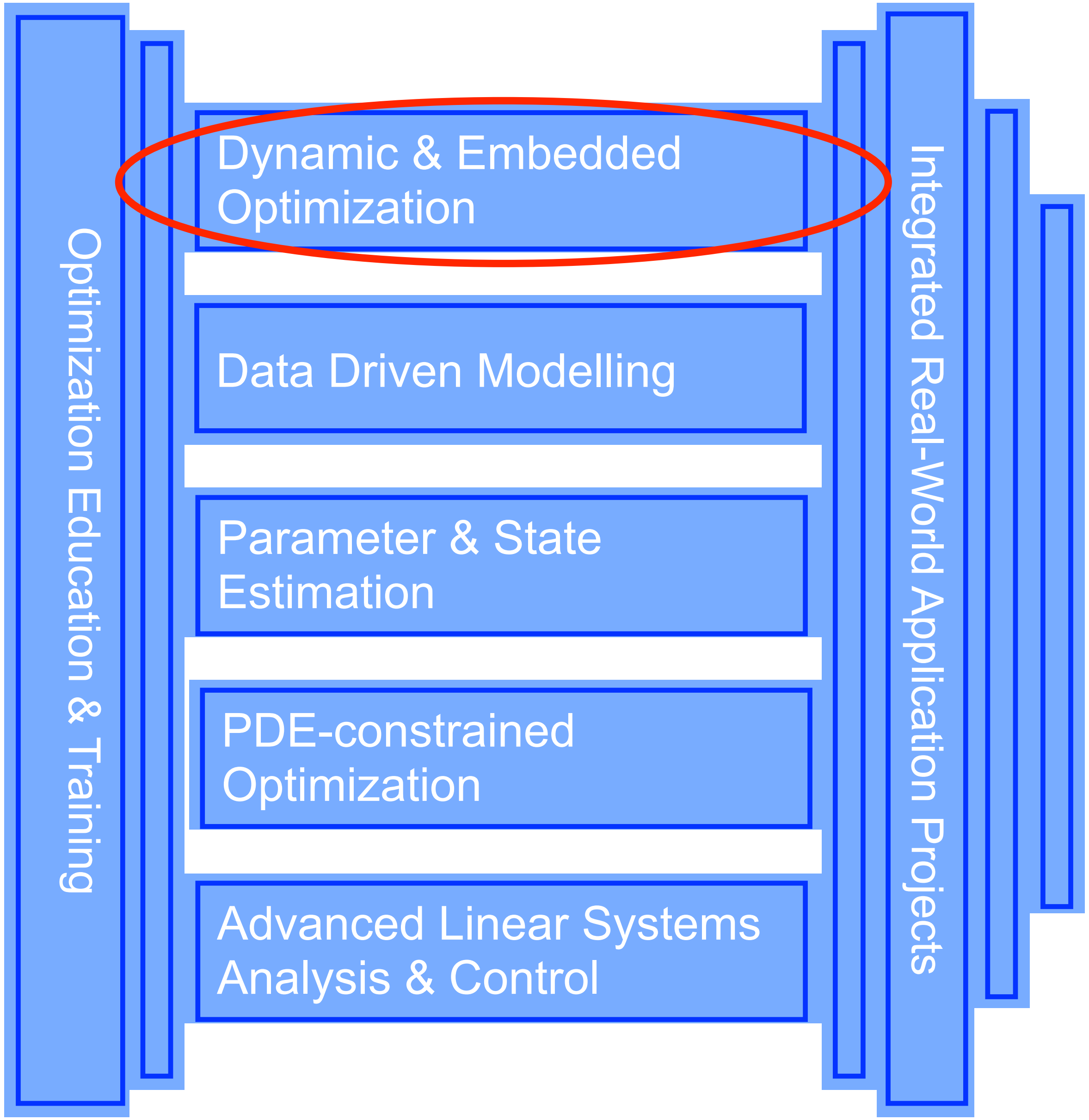
OPTEC: 70 people in five methodological working groups



OPTEC: 70 people in five methodological working groups



OPTEC: 70 people in five methodological working groups



Dynamic Optimization Problem

$$
\begin{aligned}
\underset{x,\,z,\,u}{\text{minimize}} \quad & \sum_{i=0}^{N-1} L_i(x_i, z_i, u_i) \quad + \quad E\,(x_N) \\
\text{subject to} \quad & x_0 - \bar{x}_0 = 0, \\
& x_{i+1} - f_i(x_i, z_i, u_i) = 0, \quad i = 0, \ldots, N-1, \\
& g_i(x_i, z_i, u_i) = 0, \quad i = 0, \ldots, N-1, \\
& h_i(x_i, z_i, u_i) \leq 0, \quad i = 0, \ldots, N-1, \\
& r\,(x_N) \leq 0.
\end{aligned}
$$

Structured parametric Nonlinear Program (pNLP)

- Initial Value $\bar{x}_0$ is often not known beforehand (“online data” in MPC)
- Discrete time dynamics come from ODE simulation (“multiple shooting”)

Dynamic Optimization Problem

$$
\begin{array}{llcll}
\underset{x,\,z,\,u}{\text{minimize}} & \displaystyle\sum_{i=0}^{N-1} L_i(x_i, z_i, u_i) \quad + \quad E\,(x_N) & & & \\
\text{subject to} & x_0 - \bar{x}_0 & = & 0, & \\
 & x_{i+1} - f_i(x_i, z_i, u_i) & = & 0, & i = 0, \ldots, N-1, \\
 & g_i(x_i, z_i, u_i) & = & 0, & i = 0, \ldots, N-1, \\
 & h_i(x_i, z_i, u_i) & \leq & 0, & i = 0, \ldots, N-1, \\
 & r\,(x_N) & \leq & 0. &
\end{array}
$$

Summarize as

$$
\begin{array}{ll}
\displaystyle\min_{x \in \mathbb{R}^n} & f(x) \\
\text{s.t.} & g(x) + M\xi = 0, \\
 & x \in \Omega,
\end{array}
$$

with convex f and Ω

Dynamic Optimization Problem

$$
\begin{array}{llrcll}
\underset{x,z,u}{\text{minimize}} & \displaystyle\sum_{i=0}^{N-1} L_i(x_i, z_i, u_i) & + & & E(x_N) & \\
\text{subject to} & & x_0 - \bar{x}_0 & = & 0, & \\
 & & x_{i+1} - f_i(x_i, z_i, u_i) & = & 0, & i = 0, \ldots, N-1, \\
 & & g_i(x_i, z_i, u_i) & = & 0, & i = 0, \ldots, N-1, \\
 & & h_i(x_i, z_i, u_i) & \leq & 0, & i = 0, \ldots, N-1, \\
 & & r(x_N) & \leq & 0. &
\end{array}
$$

Summarize as

$$
\begin{array}{lll}
\underset{x \in \mathbb{R}^n}{\min} & f(x) & \\
\text{s.t.} & g(x) & = 0, \\
 & x \in \Omega, &
\end{array}
$$

with convex f and Ω

Overview

- Optimization in Engineering Center OPTEC
- **Sequential Convex Programming**
- Real-Time Sequential Convex Programming
- Open Source Software ACADO
- Experiments with Tethered Airplanes

Sequential Convex Programming (SCP)

Step 1: Linearize nonlinear constraints at x^k to obtain convex problem:

$$\boxed{\begin{array}{lll} \min\limits_{x \in \mathbb{R}^n} & f(x) & \\ \text{s.t.} & g(x^k) + g'(x^k)(x - x^k) & = 0, \\ & x \in \Omega. & \end{array}}$$

Step 2: Solve convex problem to obtain next iterate

Why could SCP be useful ?

- Generalizes Sequential Quadratic Programming (SQP) to non-polyhedral constraint sets
- Some constraints cannot easily be formulated as inequalities
- Convex subproblems reliably solvable
- Possibly better approximations of original NLP

Examples:

- Ellipsoids e.g. Terminal Regions in Model Predictive Control
- Second Order Cone Constraints e.g. in Robust Optimization
- Matrix Inequalities e.g. in Linear Controller Design

  → cf. Sequential Semidefinite Programming [Fares, Noll, Apkarian, SICON, 2002]

The Bad News on SCP

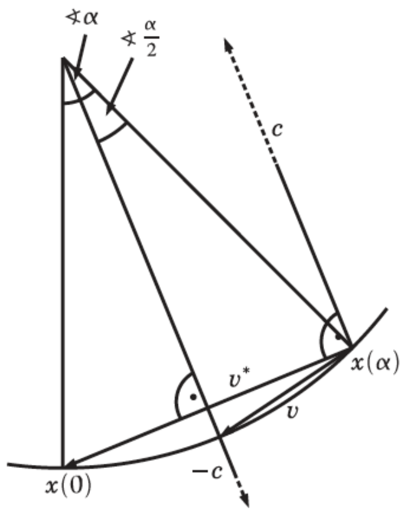
- SCP only has only **linear convergence** in general
- Cannot be improved by **any positive semi-definite bounded Hessian** approximation in subproblems:

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & \left\{ c^T x + \frac{1}{2}(x - x^k)^T H_k (x - x^k) \right\} \\ \text{s.t.} & g(x^k) + g'(x^k)(x - x^k) \qquad = 0, \\ & x \in \Omega. \end{array}$$

- Proof based on simple non-convex 2-D example:

$$\min \left\{ -x_1^2 - (x_2 - 1)^2 \mid \|x\|_2^2 \leq 1, \quad x \in \mathbb{R}^2 \right\}.$$

[Diehl, Jarre, Vogelbusch, SIOPT, 2006]



The Curse of Convexity

Convex sub-problems imply linear (or slower) convergence.

The Good News on SCP

Many widely and very successfully used methods are similar to SCP and in practice have **sufficiently fast (linear) convergence**:

- Method of Moving Asymptotes [Svanberg, 1987]
- Gauss-Newton Method and Variants* [Levenberg, 1944; Morrison, 1960; Marquardt, 1963]
- Generalized Gauss-Newton Method* [Bock, 1983]
- DC-Programming* e.g. [Pham, 1985]
- Sequential Linear Programming* [Palacios, Lasdon, Enquist, 1982] (in important special cases, SLP even has quadratic convergence)

* Can be regarded as special cases of this talk's SCP framework

Special SCP Method for Matrix Inequalities

Convex concave decompositions of Bilinear Matrix Inequalities:

$$\begin{array}{ll} \min\limits_{x} & f(x) \\ \text{s.t.} & G_i(x) - H_i(x) \preceq 0, \; i = 1, \ldots, l \\ & x \in \Omega \end{array}$$

Where G_i and H_i $(i = 1, \ldots, l)$ are psd-convex

SCP sub-problem is convex Generalized Semidefinite Program:

$$\begin{array}{ll} \min\limits_{x} & \left\{ f_k(x) := f(x) + \frac{\rho_k}{2} \|Q_k(x - x_k)\|_2^2 \right\} \\ \text{s.t.} & G_i(x) - H_i(x^k) - DH_i(x^k)(x - x^k) \preceq 0 \\ & i = 1, \ldots, l, \;\; x \in \Omega. \end{array}$$

No globalisation necessary. Easy and reliable algorithm for BMIs

[Tran Dinh, Gumussoy, Michiels, Diehl, IEEETAC, 2012]

Application to Sparse Linear Controller Design

- Regard non-convex problem with bilinear matrix inequalities (BMI)

$$
\begin{aligned}
\min_{\alpha,P,F} \quad & \left\{ f(\alpha,P,F) := -\sigma\alpha + \sum_{i=1}^{n_u}\sum_{j=1}^{n_y} |F_{ij}| \right\} \\
\text{s.t.} \quad & (A+BFC)^T P + P(A+BFC) + 2\alpha P \prec 0, \\
& P = P^T, \; P \succ 0.
\end{aligned}
$$

- Decompose BMI using one of several possible psd-DC decompositions, e.g.

$$
\begin{aligned}
X^T Y + Y^T X &= (X+Y)^T (X+Y) - (X^T X + Y^T Y) \\
&= X^T X + Y^T Y - (X-Y)^T (X-Y) \\
&= \frac{1}{2}[(X+Y)^T (X+Y) - (X-Y)^T (X-Y)]
\end{aligned}
$$

[Tran Dinh, Gumussoy, Michiels, Diehl, IEEETAC, 2012]

Good Performance for Convex Concave SCP

TABLE I
COMPUTATIONAL RESULTS FOR (24) IN COMPl$_e$ib

Problem		Other Results, $\alpha_0(A_F)$			Results and Performances		
Name	$\alpha_0(A)$	HIFOO	LMIRANK	PENBMI	$\alpha_0(A_F)$	Iter	time[s]
AC1	0.000	-0.2061	-8.4766	-7.0758	-0.8535	41	12.44
AC4	2.579	-0.0500	-0.0500	-0.0500	-0.0500	14	4.60
AC5[a]	0.999	-0.7746	-1.8001	-2.0438	-0.7389	28	63.33
AC7	0.172	-0.0322	-0.0204	0.0896	-0.0673	150	111.46
AC8	0.012	-0.1968	-0.4447	0.4447	-0.0755	24	21.95
AC9	0.012	-0.3389	-0.5230	-0.4450	-0.3256	78	74.57
AC11	5.451	-0.0003	-5.0577	-	-3.0244	61	38.44
AC12	0.580	-10.8645	-9.9658	-1.8757	-0.3414	150	86.72
HE1	0.276	-0.2457	-0.2071	-0.2468	-0.2202	150	87.64
HE3	0.087	-0.4621	-2.3009	-0.4063	-0.8702	47	48.80
HE4	0.234	-0.7446	-1.9221	-0.0909	-0.8647	63	71.66
HE5	0.234	-0.1823	-	-0.2932	-0.0587	150	178.71
HE6	0.234	-0.0050	-0.0050	-0.0050	-0.0050	12	41.00
REA1	1.991	-16.3918	-5.9736	-1.7984	-3.8599	77	79.23
REA2	2.011	-7.0152	-10.0292	-3.5928	-2.1778	40	39.15
REA3	0.000	-0.0207	-0.0207	-0.0207	-0.0207	150	362.21
DIS2	1.675	-6.8510	-10.1207	-8.3289	-8.4540	28	37.28
DIS4	1.442	-36.7203	-0.5420	-92.2842	-8.0989	72	124.23
WEC1	0.008	-8.9927	-8.7350	-0.9657	-0.8779	150	305.36
IH	0.000	-0.5000	-0.5000	-0.5000	-0.5000	7	39.41
CSE1	0.000	-0.4509	-0.4844	-0.4490	-0.2360	38	158.67
TF1	0.000	-	-	-0.0618	-0.1544	56	137.98
TF2	0.000	-	-	-1.0e-5	-1.0e-5	8	20.41
TF3	0.000	-	-	-0.0032	-0.0031	93	237.93
NN1	3.606	-3.0458	-4.4021	-4.3358	-0.8746	12	37.53
NN5[a]	0.420	-0.0942	-0.0057	-0.0942	-0.0913	11	42.62
NN9	3.281	-2.0789	-0.7048	-	-0.0279	33	111.41
NN13	1.945	-3.2513	-4.5310	-9.0741	-3.4318	150	572.50
NN15	0.000	-6.9983	-11.0743	-0.0278	-0.8353	150	524.80
NN17	1.170	-0.6110	-0.5130	-	-0.6008	99	342.67

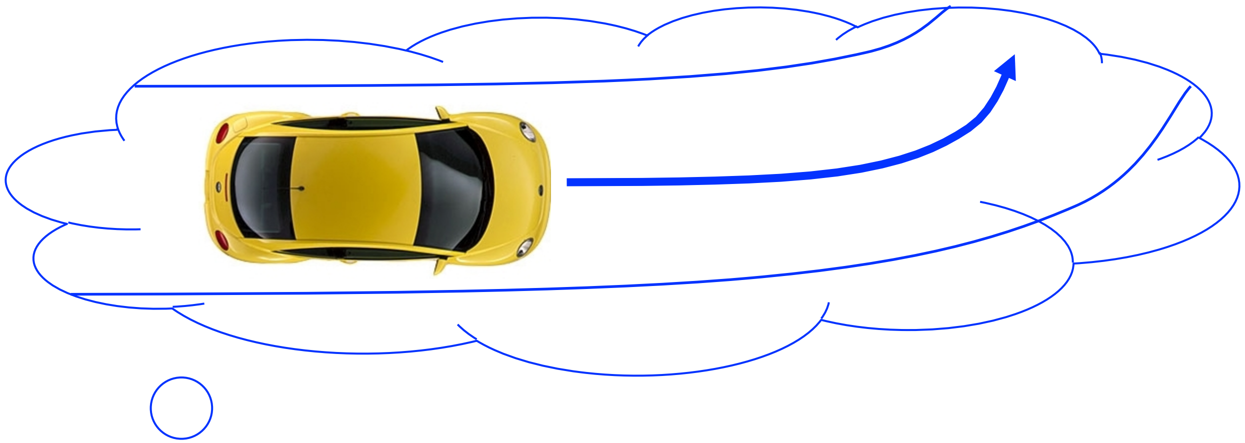
[Tran Dinh, Gumussoy, Michiels, Diehl, IEEETAC, 2012]

Overview

- Optimization in Engineering Center OPTEC
- Sequential Convex Programming
- **Real-Time Sequential Convex Programming**
- Open Source Software ACADO
- Experiments with Tethered Airplanes

Model Predictive Control (MPC)

Idea: always look a bit into the future.



Brain predicts and optimizes:
e.g. slow down **before** curve

Dynamic Optimization Problem

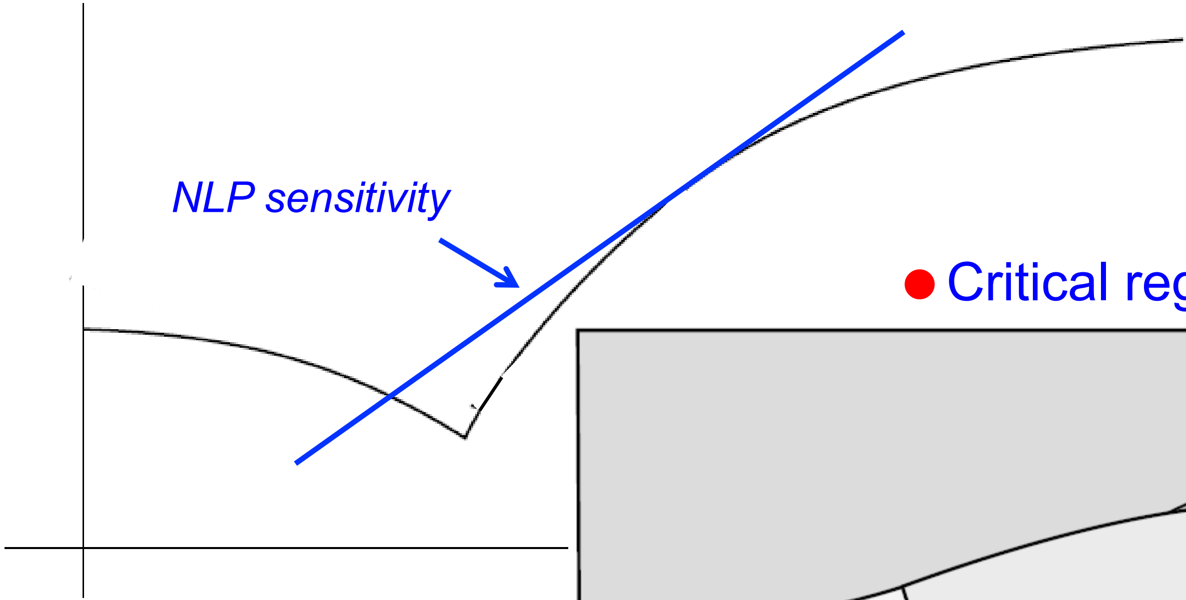
$$
\begin{array}{llcll}
\underset{x,\,z,\,u}{\text{minimize}} & \displaystyle\sum_{i=0}^{N-1} L_i(x_i, z_i, u_i) & + & E\,(x_N) & \\
\text{subject to} & x_0 - \bar{x}_0 & = & 0, & \\
& x_{i+1} - f_i(x_i, z_i, u_i) & = & 0, & i = 0, \ldots, N-1, \\
& g_i(x_i, z_i, u_i) & = & 0, & i = 0, \ldots, N-1, \\
& h_i(x_i, z_i, u_i) & \leq & 0, & i = 0, \ldots, N-1, \\
& r\,(x_N) & \leq & 0. &
\end{array}
$$

Parametric Nonlinear Program

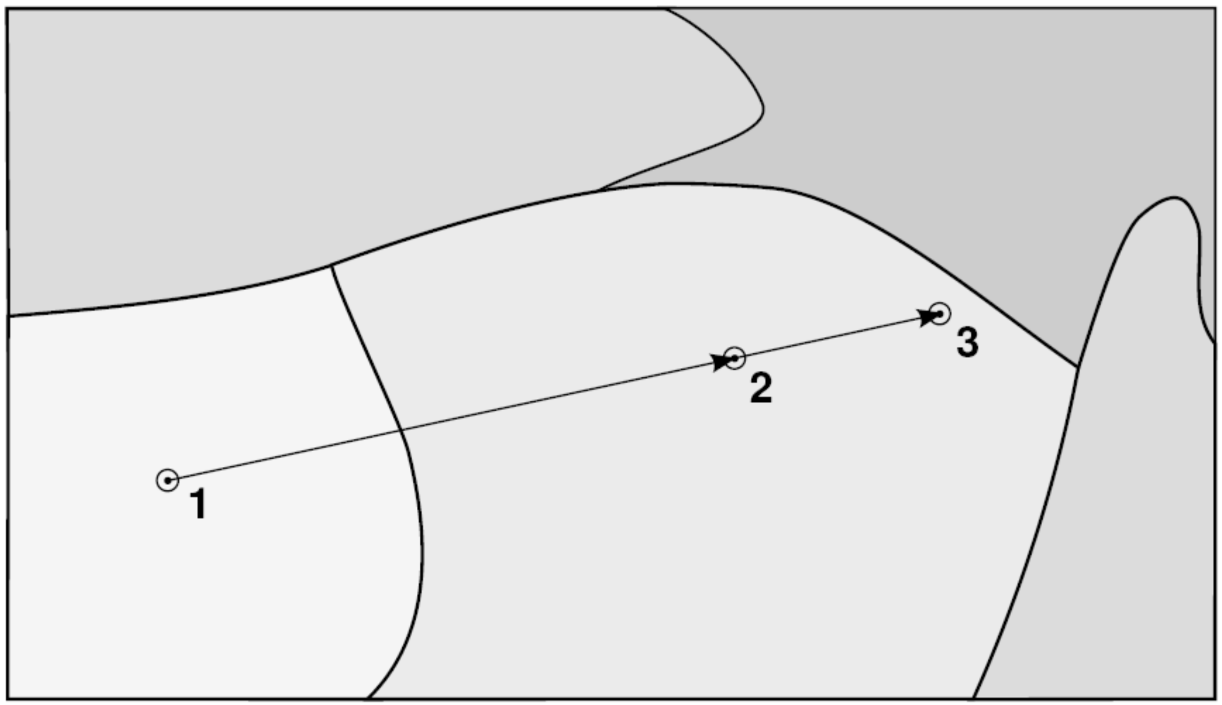
- Initial Value $\bar{x}_0$ is often not known beforehand ("online data" in MPC)

Nonlinear MPC = parametric NLP

- Solution manifold is piecewise differentiable (kinks at active set changes)



- Critical regions are non-polyhedral



How to deal with a sequence of large parameter changes?

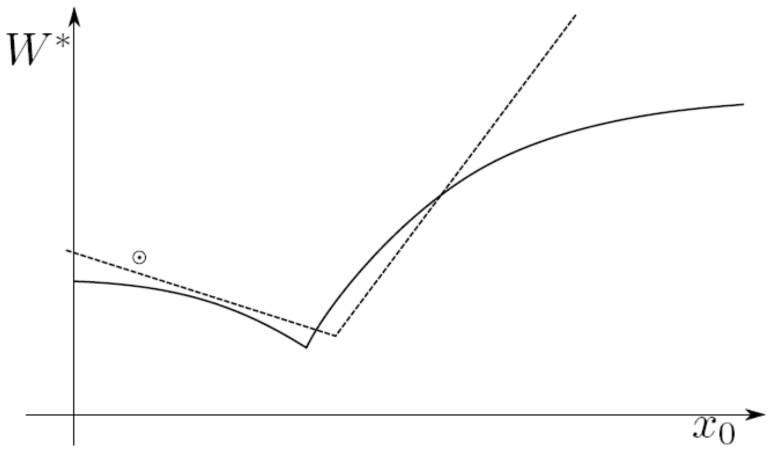
NLP Pathfollowing!

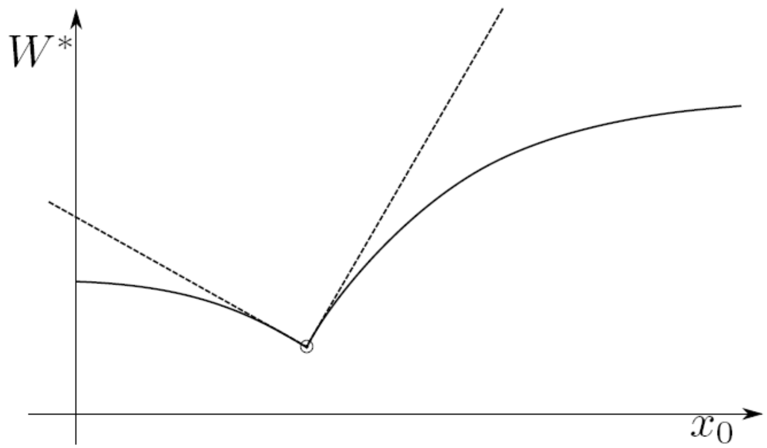
Sequential Quadratic Programming for p-NLP

- In each iteration, linearize and solve *parametric* QP with inequalities

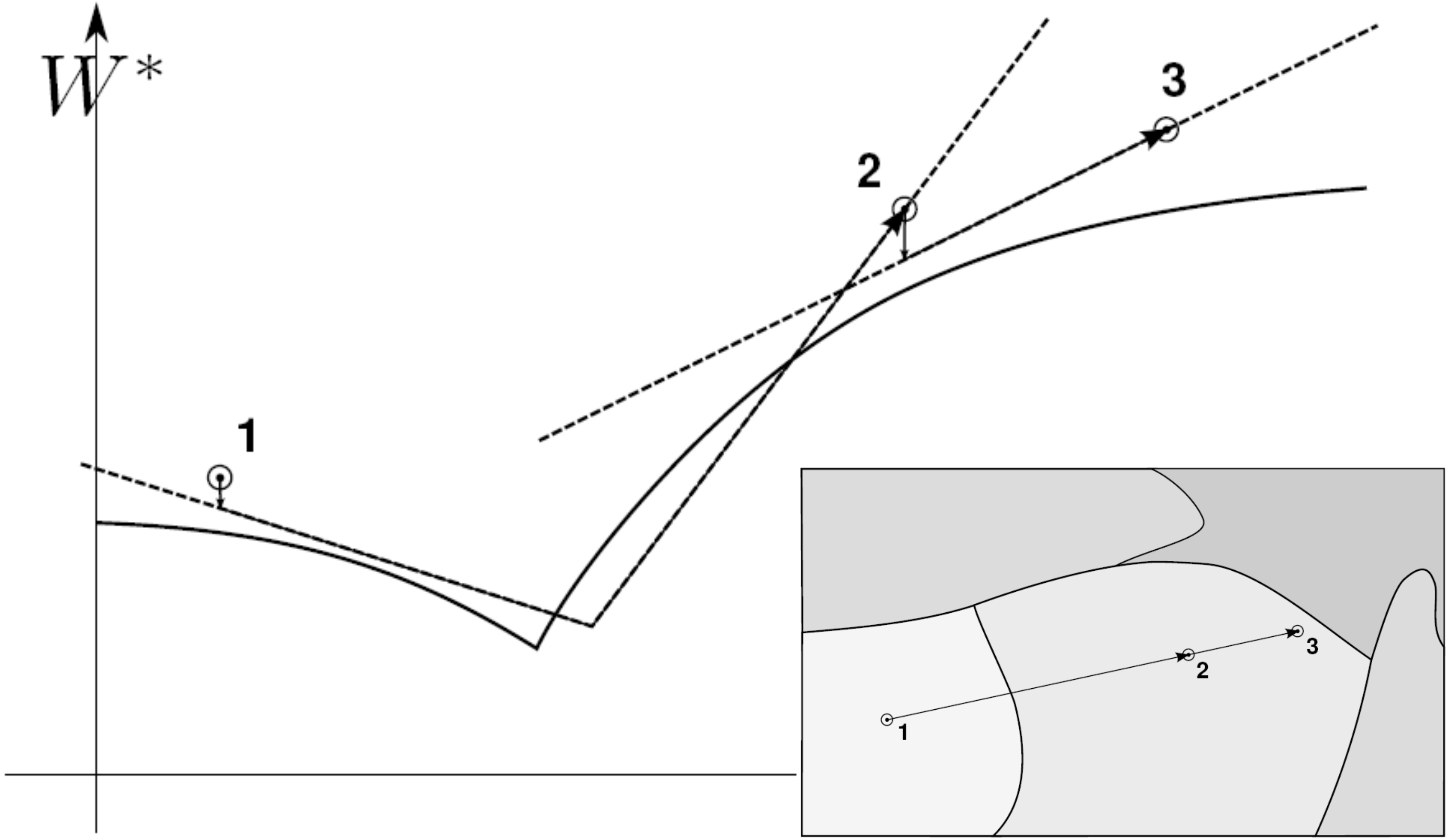
$$\begin{array}{llcll} \underset{x,\,u}{\text{minimize}} & \sum_{i=0}^{N-1} L_{\text{redQP},i}(x_i, u_i) \quad + \quad E_{\text{QP}}(x_N) & & & \\ \text{subject to} & x_0 - \bar{x}_0 & = & 0, & \\ & x_{i+1} - c_i - A_i x_i - B_i u_i & = & 0, & i = 0, \ldots, N-1, \\ & \bar{h}_i + \bar{H}_i^x x_i + \bar{H}_i^u u_i & \leq & 0, & i = 0, \ldots, N-1, \\ & r' + R x_N & \leq & 0. & \end{array}$$

- This “Initial Value Embedding” delivers first order prediction also at active set changes [D. 2001].





SQP Real-Time Iteration [D. 2001]



- long "preparation phase" for linearization
- fast "feedback phase" (QP solution once $\bar{x}_0$ is known)

Real-Time Sequential Convex Programming

Step 1: Linearize nonlinear constraints at x^k to obtain convex problem:

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t.} & g(x^k) + g'(x^k)(x - x^k) + M\xi = 0, \\ & x \in \Omega. \end{array}$$

Step 2: Solve convex problem to obtain next iterate.
Obtain new value of parameter ξ and go to step 1.

[Diehl, Bock, Schloeder, Findeisen, Nagy, Allgower, JPC, 2002]
[Zavala, Anitescu, SICON, 2010]
[Tran Dinh, Savorgnan, Diehl, SIOPT, 2013]

Adjoint Based SCP Variant

- SCP also works with inexact Jacobian matrices…

$$\boxed{\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n} & \left\{f(x) + (s^k)^T(x-x^k) + \frac{1}{2}(x-x^k)^T H_k (x-x^k)\right\} \\ \text{s.t.} & A_k(x-x^k) + g(x^k) + M\xi = 0, \\ & x \in \Omega, \end{array}}$$

$$\mathcal{L}(x, y; \xi) := f(x) + (g(x) + M\xi)^T y,$$

- …if a gradient correction based on multiplier guess y^k is used:

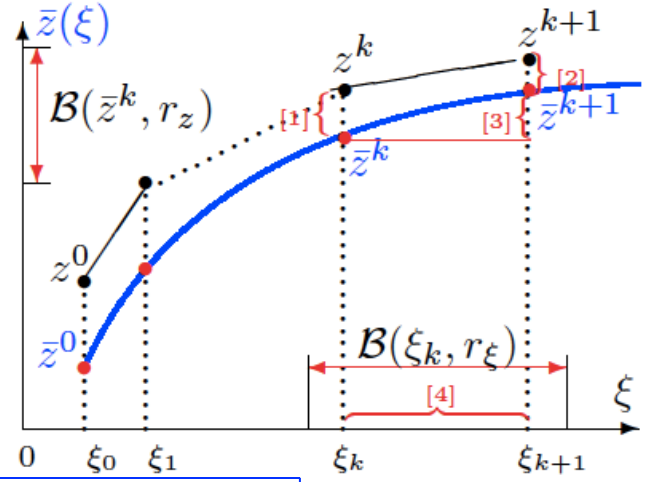
$$s^k := \left(g'(x^k) - A_k\right)^T y^k$$

(should be computed by adjoint mode of automatic differentiation)

- Still linear convergence [Tran Dinh, Savorgnan, Diehl, SIOPT, 2013] cf. [Griewank, Walther, 2002; D., Walther, Bock, Kostina, OMS, 2010]

Real-Time SCP Contraction Estimate

$$z^k := (x^k, y^k)$$



Contraction estimate for primal dual errors:

$$\boxed{\begin{aligned}\left\|z^{k+1} - \bar{z}^{k+1}\right\| \le &\left(\alpha + c_1 \left\|z^k - \bar{z}^k\right\|\right)\left\|z^k - \bar{z}^k\right\| \\ &+ \left(c_2 + c_3 \left\|\xi_{k+1} - \xi_k\right\|\right)\left\|\xi_{k+1} - \xi_k\right\|\end{aligned}}$$

Contraction depends on bounds on nonlinearity, Jacobian error, and on strong regularity. Contraction rate independent of active set changes!

[Tran Dinh, Savorgnan, Diehl, SIOPT, 2013]

Overview

- Optimization in Engineering Center OPTEC
- Sequential Convex Programming
- Real-Time Sequential Convex Programming
- **Open Source Software ACADO**
- Experiments with Tethered Airplanes

ACADO Toolkit [1]

- ACADO = Automatic Control and Dynamic Optimization

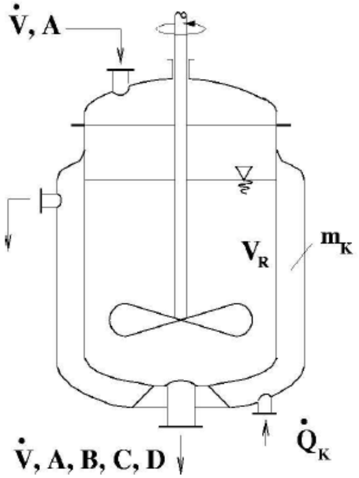
- Open source (LGPL) C++: www. acadotoolkit. org
- Implements direct multiple shooting [2] and real-time iterations [3]
- User interface close to mathematical syntax
- *Automatic C-Code Export for Microsecond Nonlinear MPC* [4]

- Developed at OPTEC by B. Houska, H.J. Ferreau, M. Vukov, ...
- ~3000 downloads since first release in 2009

[1] Houska, Ferreau, D., *OCAM,* 2011
[2] Bock, Plitt, *IFAC WC*, 1984
[3] D., Bock, Schloder, Findeisen, Nagy, Allgower, *JPC*, 2002
[4] Houska, Ferreau, D., *Automatica,* 2011



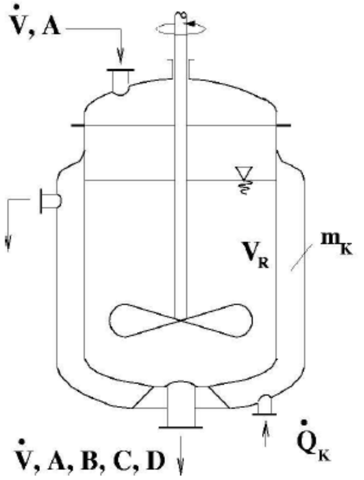
ACADO Code Generation for Benchmark CSTR



$$
\begin{aligned}
\dot{c}_A(t) &= u_1(c_{A0} - c_A(t)) - k_1(\vartheta(t))c_A(t) - k_3(\vartheta(t))(c_A(t))^2 \\
\dot{c}_B(t) &= -u_1 c_B(t) + k_1(\vartheta(t))c_A(t) - k_2(\vartheta(t))c_B(t) \\
\dot{\vartheta}(t) &= u_1(\vartheta_0 - \vartheta(t)) + \frac{k_w A_R}{\rho C_p V_R}(\vartheta_K(t) - \vartheta(t)) \\
&\quad - \frac{1}{\rho C_p}\big[k_1(\vartheta(t))c_A(t)H_1 + k_2(\vartheta(t))c_B(t)H_2 \\
&\quad + k_3(\vartheta(t))(c_A(t))^2 H_3\big] \\
\dot{\vartheta}_K(t) &= \frac{1}{m_K C_{PK}}(u_2 + k_w A_R(\vartheta(t) - \vartheta_K(t))) \,.
\end{aligned}
$$

CSTR Benchmark by [Klatt & Engell 1993]

ACADO Code Generation for Benchmark CSTR



$$\dot{c}_A(t) = u_1(c_{A0} - c_A(t)) - k_1(\vartheta(t))c_A(t) - k_3(\vartheta(t))(c_A(t))^2$$
$$\dot{c}_B(t) = -u_1 c_B(t) + k_1(\vartheta(t))c_A(t) - k_2(\vartheta(t))c_B(t)$$
$$\begin{aligned}\dot{\vartheta}(t) &= u_1(\vartheta_0 - \vartheta(t)) + \frac{k_w A_R}{\rho C_p V_R}(\vartheta_K(t) - \vartheta(t)) \\ &\quad - \frac{1}{\rho C_p}\left[k_1(\vartheta(t))c_A(t)H_1 + k_2(\vartheta(t))c_B(t)H_2 \right.\\ &\quad \left. + k_3(\vartheta(t))(c_A(t))^2 H_3\right]\end{aligned}$$
$$\dot{\vartheta}_K(t) = \frac{1}{m_K C_{PK}}(u_2 + k_w A_R(\vartheta(t) - \vartheta_K(t))).$$

CSTR Benchmark by [Klatt & Engell 1993]

CPU Times for ACADO:

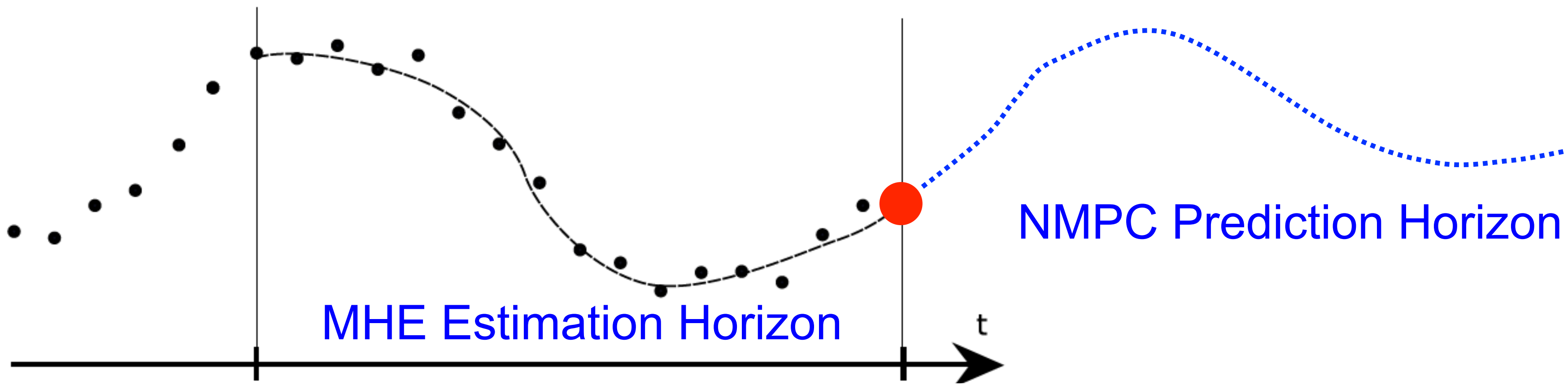
	CPU time (μs)	%
Integration & sensitivities	121	30
Condensing	98	24
QP solution (with qpOASES)[a]	180	44
Remaining operations	<5	<2
A complete real-time iteration	404	100

From [Houska, Ferreau, D., *Automatica,* 2011]

NMPC now 100 000x faster than 1997 (200x by CPU, 500x by algorithms)

NMPC Practice: Estimation AND Optimization

- Moving Horizon Estimation (MHE): Get State by Least Squares Optimization
- Noninear Model Predictive Control (NMPC): Solve Optimal Control Problem

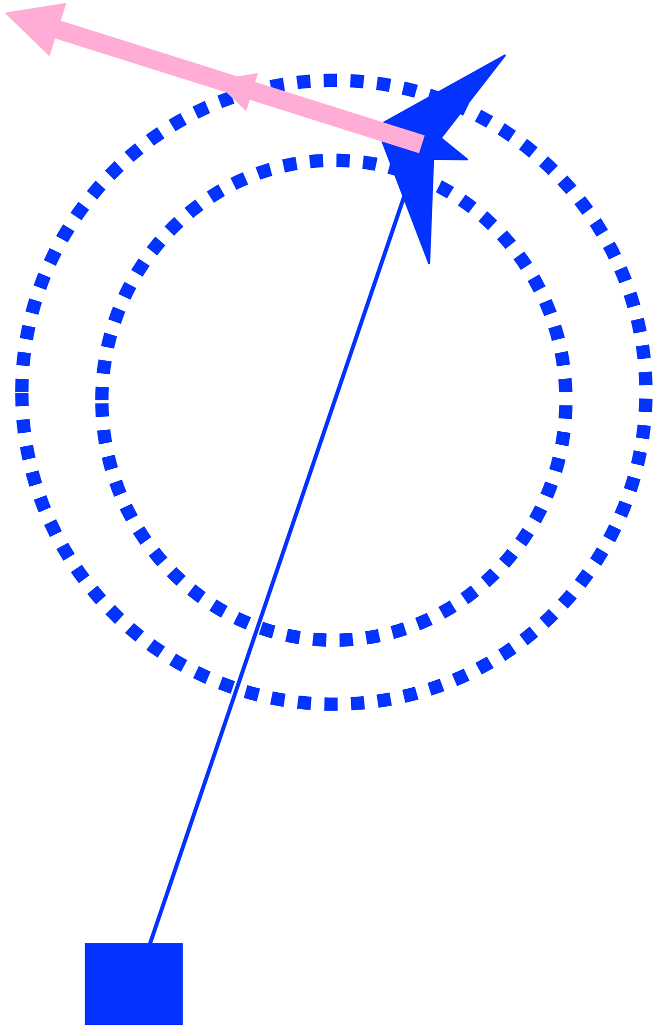


Overview

- Optimization in Engineering Center OPTEC
- Sequential Convex Programming
- Real-Time Sequential Convex Programming
- Open Source Software ACADO
- **Experiments with Tethered Airplanes**

What is the Optimal Wind Turbine ?

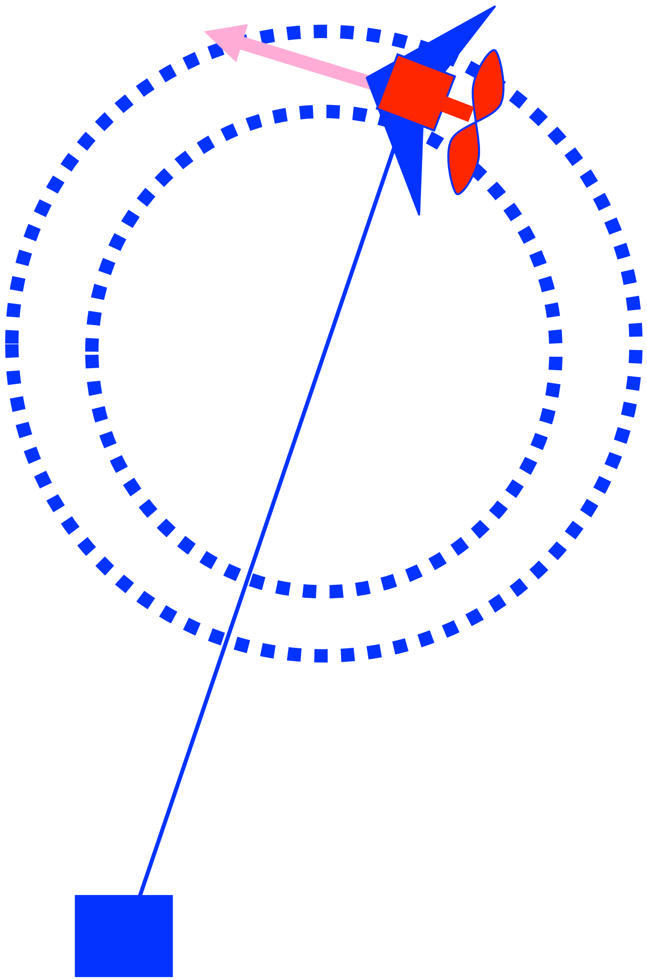
Crosswind Kite Power



- Fly kite fast in crosswind direction
- Very strong force.

But where could a generator be driven?

Variant 1: On-Board Generator

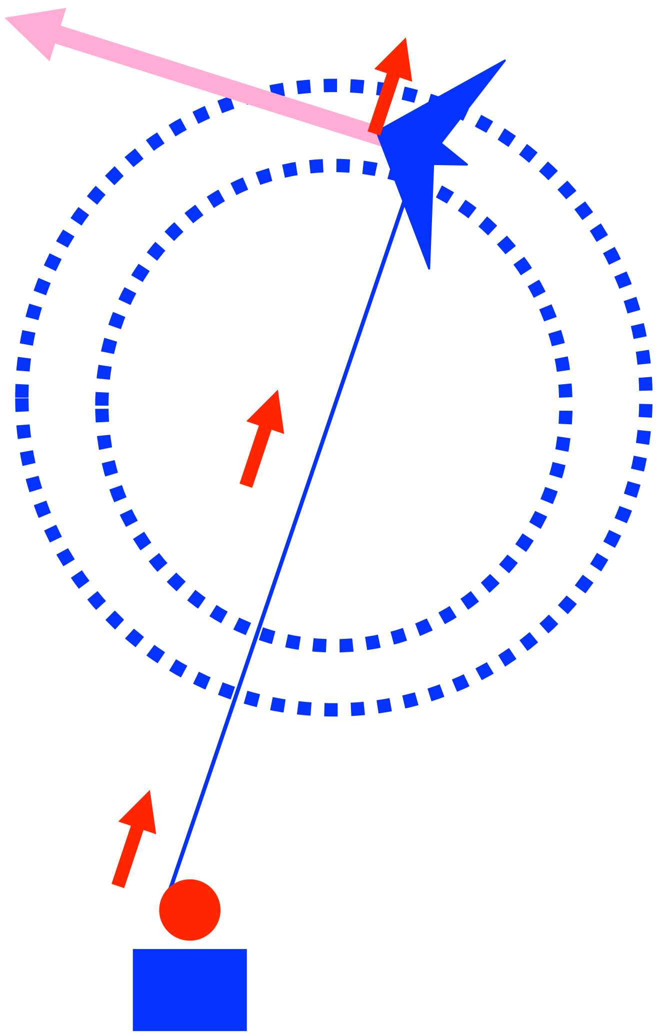


- attach ***small wind turbines*** to kite
- cable transmits power

Pro: light, high speed generators

Con: high voltage power transmission

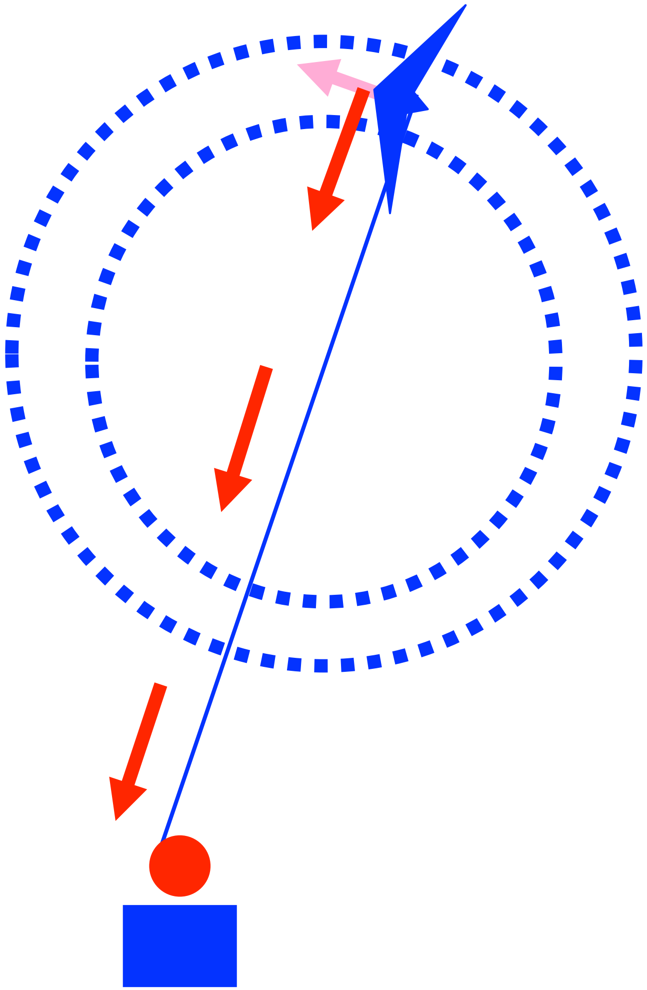
Variant 2: Generator on Ground



Cycle consists of two phases:

- ***Power generation phase:***
  - Fly kite fast, have high force
  - ***unwind cable***
  - generate power

Variant 2: Generator on Ground



Cycle consists of two phases:

- ***Power generation phase:***
  - Fly kite fast, have high force
  - ***unwind cable***
  - generate power

- ***Retraction phase:***
  - Slow down kite, reduce force
  - pull back line

Pro: all electric parts on ground

Con: heavy, slow generator

Seminal Paper by Miles Loyd in 1980

J. ENERGY VOL. 4, NO. 3
ARTICLE NO. 80-4075

Crosswind Kite Power

Miles L. Loyd*
Lawrence Livermore National Laboratory, Livermore, Calif.

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

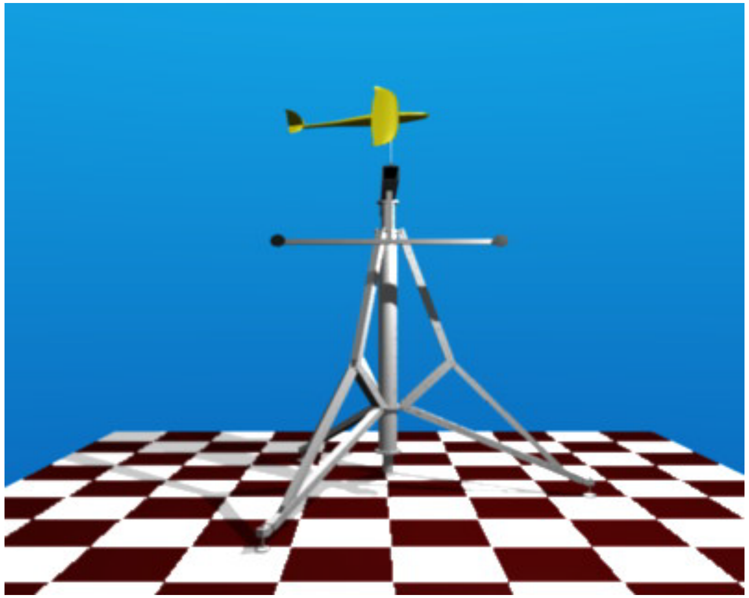
- Wing area of $1 m^2$ can deliver up to 40 kW electric power (200 x higher power density than PV)

European Research Council Project, 2011-2016

erc **HIGHWIND**

Simulation, Optimization, and Control of High Altitude Wind Power Generators

Aim: Guide the development of high altitude wind power, focus on *modeling, optimization, and control*, plus small scale experiments.



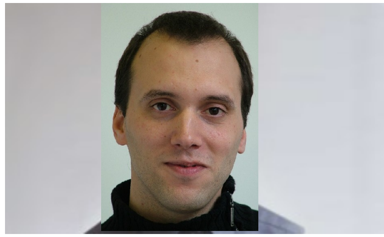
ERC
HIGHWIND
SIMULATION, OPTIMIZATION & CONTROL OF
HIGH-ALTITUDE WIND POWER GENERATORS

Current ERC HIGHWIND Team

Prof. Moritz Diehl
principal investigator

Prof. Jan Swevers
control systems

Prof. Dirk Vandepitte
mechanical design

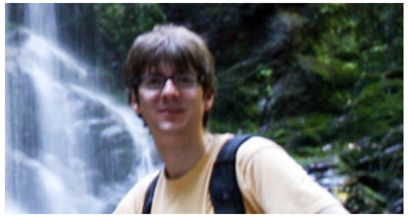


Prof. Johan Meyers
CFD simulation

Dr. Sébastien Gros
coordinator, simulation

Dr. Andrew Wagner
vision & hardware

Greg Horn
large system optimization



Joris Gillis
stability optimization

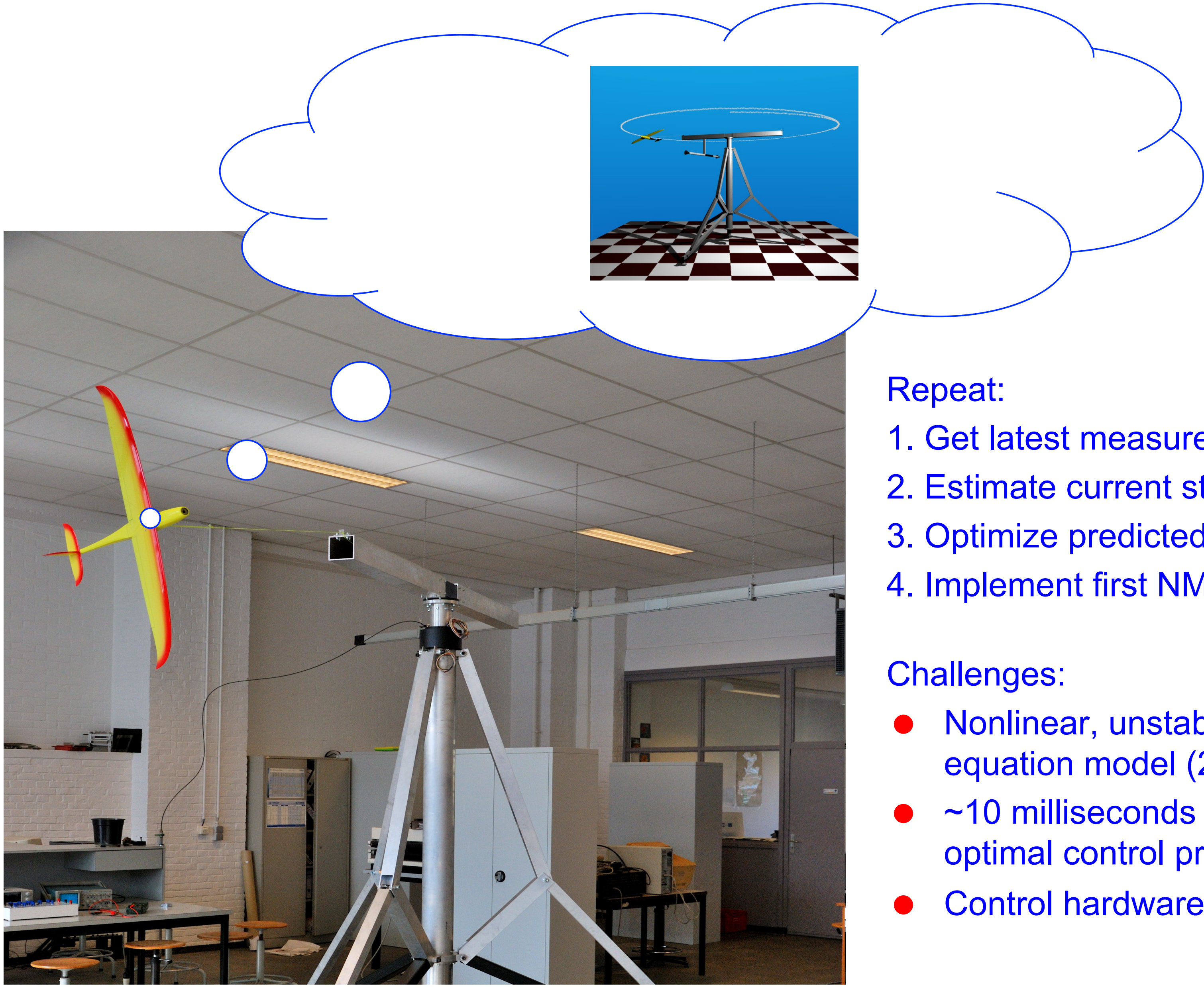
Kurt Geebelen
control experiments

Mario Zanon
predictive control

Milan Vukov
embedded optimization

ERC HIGHWIND
SIMULATION, OPTIMIZATION & CONTROL OF HIGH-ALTITUDE WIND POWER GENERATORS

Nonlinear MPC and MHE of Leuven Power Plane



Repeat:

1. Get latest measurements
2. Estimate current state (MHE)
3. Optimize predicted trajector (NMPC)
4. Implement first NMPC control

Challenges:

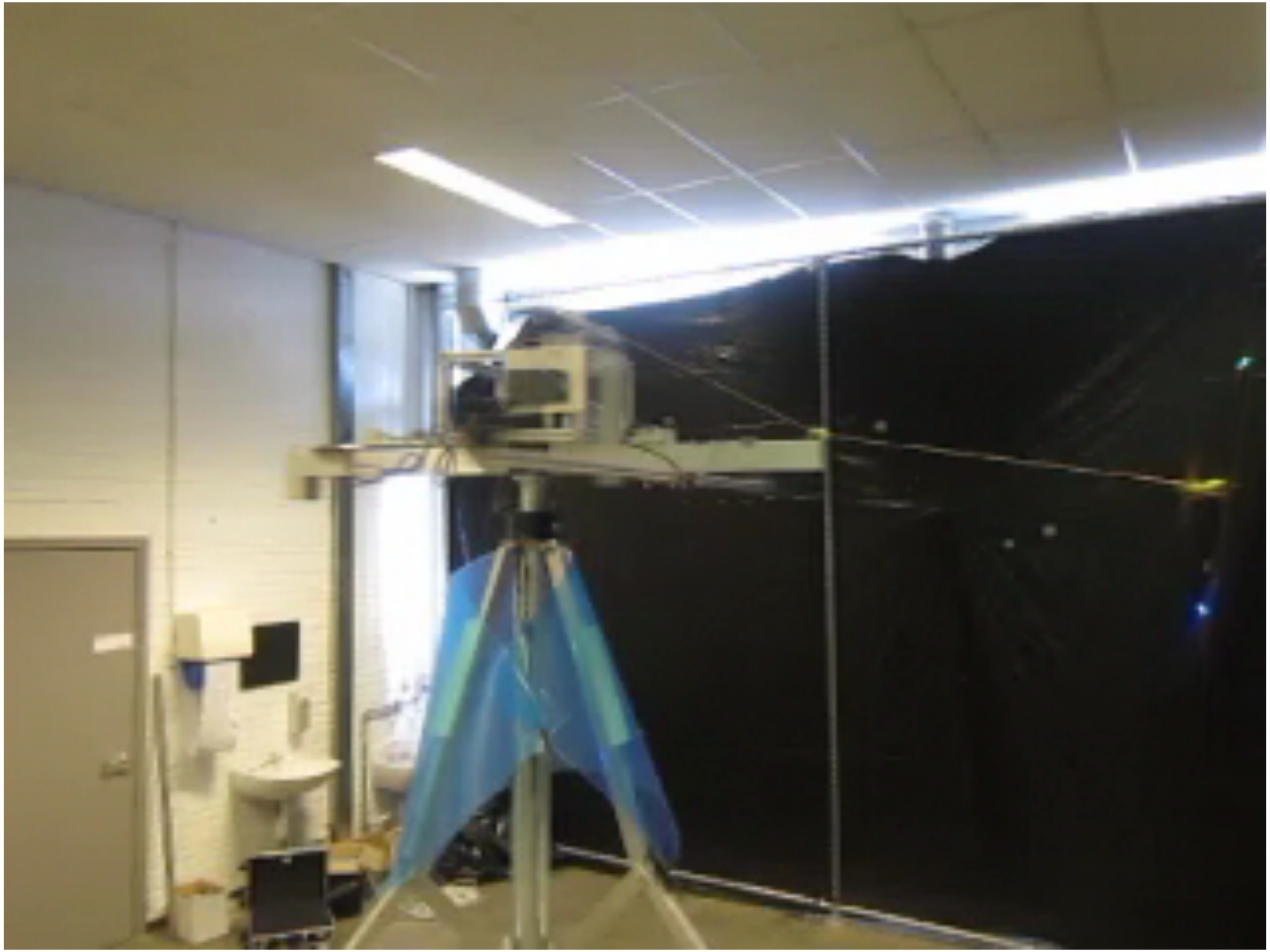
- Nonlinear, unstable differential equation model (22 states)
- ~10 milliseconds to solve two optimal control problems
- Control hardware, not desktop PC

ACADO Code Generation for Tethered Airplanes

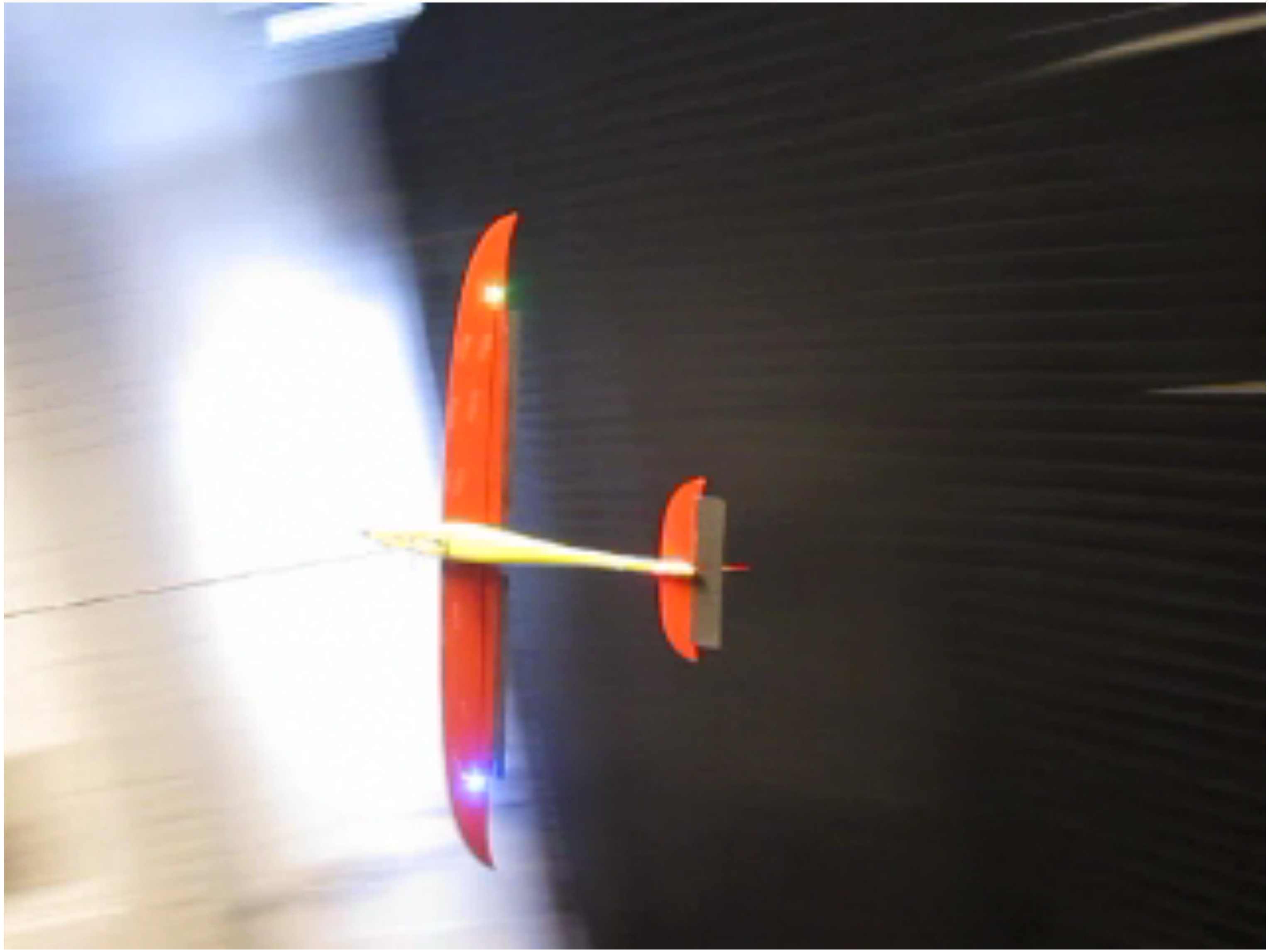
- 22 states
- 2 controls
- 1 s Prediction/Estimation Horizon
- Implicit RK Integrator (by R. Quirynen)

4 ms Execution times for one optimization problem (on i7 2.5 GHz)

MHE+NMPC Experiments (Aug 22, 2012)



MHE+NMPC Experiments (Aug 22, 2012)




ERC HIGHWIND Vision: replace steel by intelligent control

ERC HIGHWIND Vision: replace steel by intelligent control