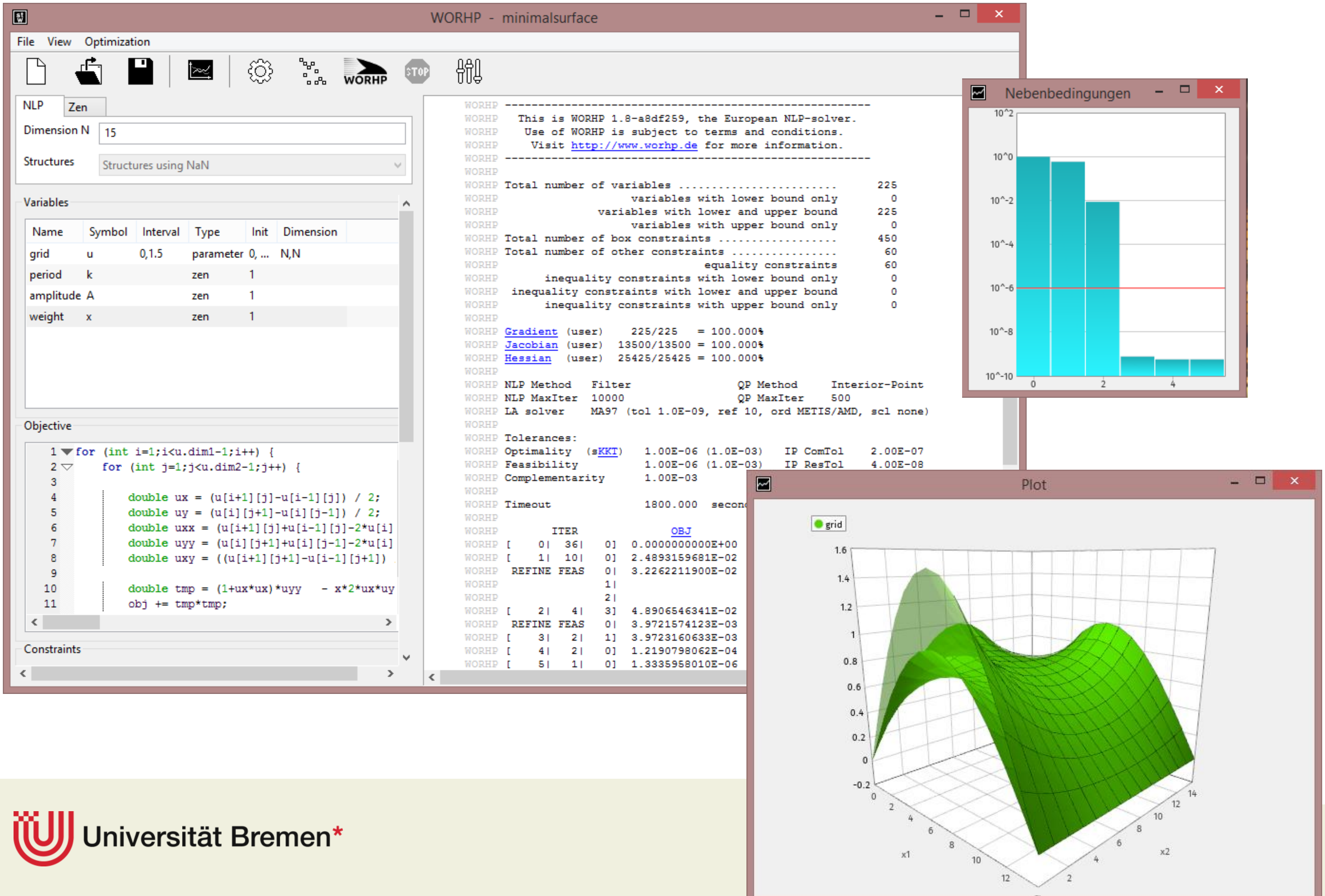
Understanding Concepts of Optimization and Optimal Control with WORHP Lab

M. Knauer, C. Büskens

Zentrum für Technomathematik
Universität Bremen

6th International Conference
on Astrodynamics Tools and Techniques
14th - 17th March 2016


WORHP - minimalsurface
File View Optimization
WORHP
STOP
NLP Zen
Dimension N 15
Structures Structures using NaN
Variables
Name Symbol Interval Type Init Dimension
grid u 0,1.5 parameter 0, ... N,N
period k zen 1
amplitude A zen 1
weight x zen 1
Objective
for (int i=1;i<u.dim1-1;i++) {
for (int j=1;j<u.dim2-1;j++) {
double ux = (u[i+1][j]-u[i-1][j]) / 2;
double uy = (u[i][j+1]-u[i][j-1]) / 2;
double uxx = (u[i+1][j]+u[i-1][j]-2*u[i]
double uyy = (u[i][j+1]+u[i][j-1]-2*u[i]
double uxy = ((u[i+1][j+1]-u[i-1][j+1])
double tmp = (1+ux*ux)*uyy - x*2*ux*uy
obj += tmp*tmp;
Constraints
This is WORHP 1.8-a8df259, the European NLP-solver.
Use of WORHP is subject to terms and conditions.
Visit http://www.worhp.de for more information.
Total number of variables 225
variables with lower bound only 0
variables with lower and upper bound 225
variables with upper bound only 0
Total number of box constraints 450
Total number of other constraints 60
equality constraints 60
inequality constraints with lower bound only 0
inequality constraints with lower and upper bound 0
inequality constraints with upper bound only 0
Gradient (user) 225/225 = 100.000%
Jacobian (user) 13500/13500 = 100.000%
Hessian (user) 25425/25425 = 100.000%
NLP Method Filter QP Method Interior-Point
NLP MaxIter 10000 QP MaxIter 500
LA solver MA97 (tol 1.0E-09, ref 10, ord METIS/AMD, scl none)
Tolerances:
Optimality (sKKT) 1.00E-06 (1.0E-03) IP ComTol 2.00E-07
Feasibility 1.00E-06 (1.0E-03) IP ResTol 4.00E-08
Complementarity 1.00E-03
Timeout 1800.000
ITER OBJ
[0| 36| 0] 0.0000000000E+00
[1| 10| 0] 2.4893159681E-02
REFINE FEAS 0| 3.2262211900E-02
[2| 4| 3] 4.8906546341E-02
REFINE FEAS 0| 3.9721574123E-03
[3| 2| 1] 3.9723160633E-03
[4| 2| 0] 1.2190798062E-04
[5| 1| 0] 1.3335958010E-06
Nebenbedingungen
10^2
10^0
10^-2
10^-4
10^-6
10^-8
10^-10
0 2 4
Plot
grid
1.6 1.4 1.2 1 0.8 0.6 0.4 0.2 0 -0.2
x1
x2
0 2 4 6 8 10 12 14

Overview

WORHP Lab

WORHP

Introduction to GUI

Definition of problem

WORHP Zen

Sensitivity derivatives

Usage in

- WORHP
- TransWORHP

TransWORHP

Direct methods

Grid refinement

Sparsity Structures

MPC

Zentrum für Technomathematik

O^2C Optimization & Optimal Control

The WORHP Family

We Optimize Really Huge Problems

WORHP

- Sparse NLP solver
- **Official ESA NLP solver**
- Parallel linear algebra
- 99.8% of AMPL CUTEr testset
- Industrial standard
- Interfaces to Fortran/C/Matlab

Funding

esa, DLR, CLEANSKY, stw

Subpartner

THALES

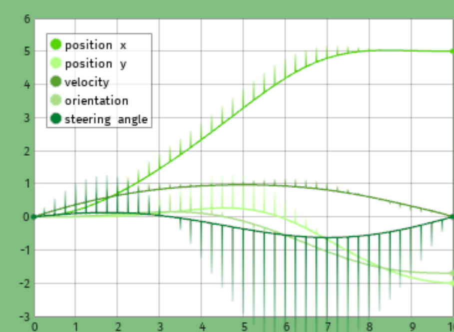
Development

Universität Bremen

Universität der Bundeswehr München

WORHP Zen

- Parametric sensitivity analysis
- Postoptimality analysis tool
- No computational cost



Funding

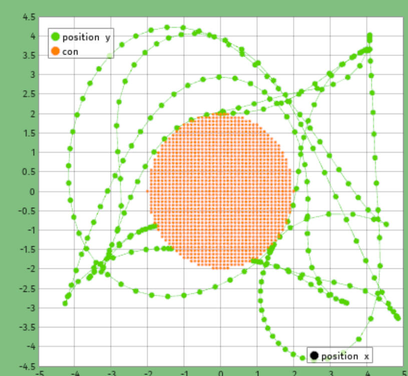
stw

Development

Universität Bremen

TransWORHP

- Sparse OCP solver
- Full discretization or shooting
- Multiple Phases
- MPC



Funding

stw

Development

Universität Bremen

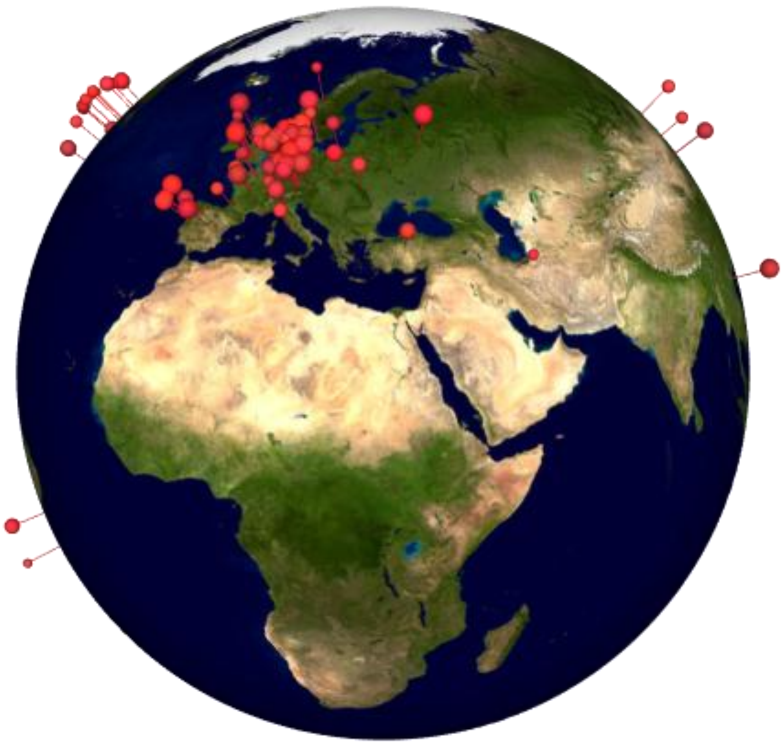
Universität Bremen*

Free for academics and testing

www.worhp.de

6TH ICATT International Conference on Astrodynamics Tools and Techniques

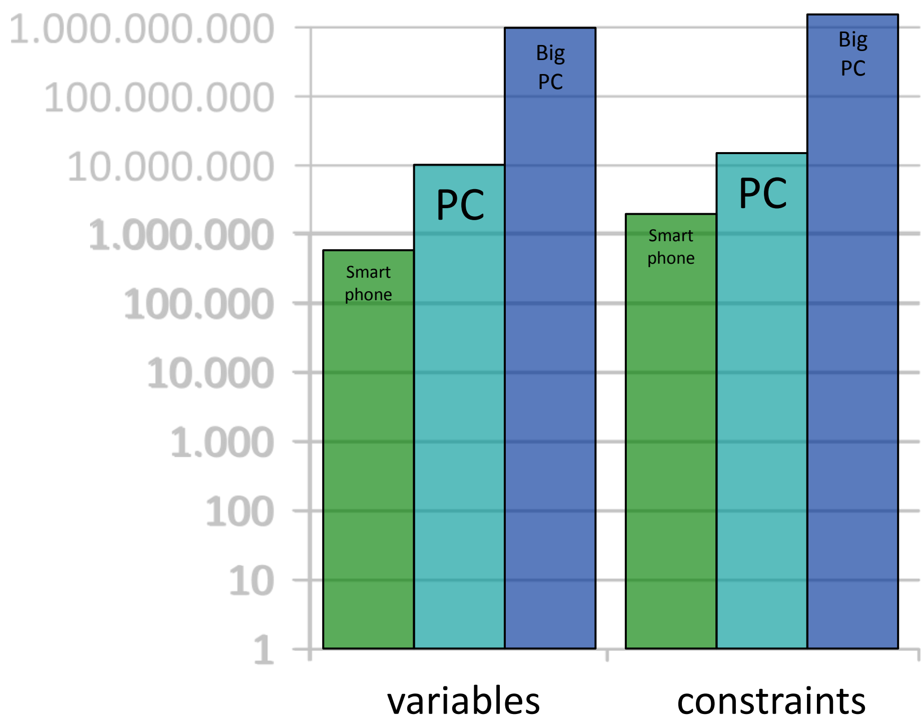
WORHP Community

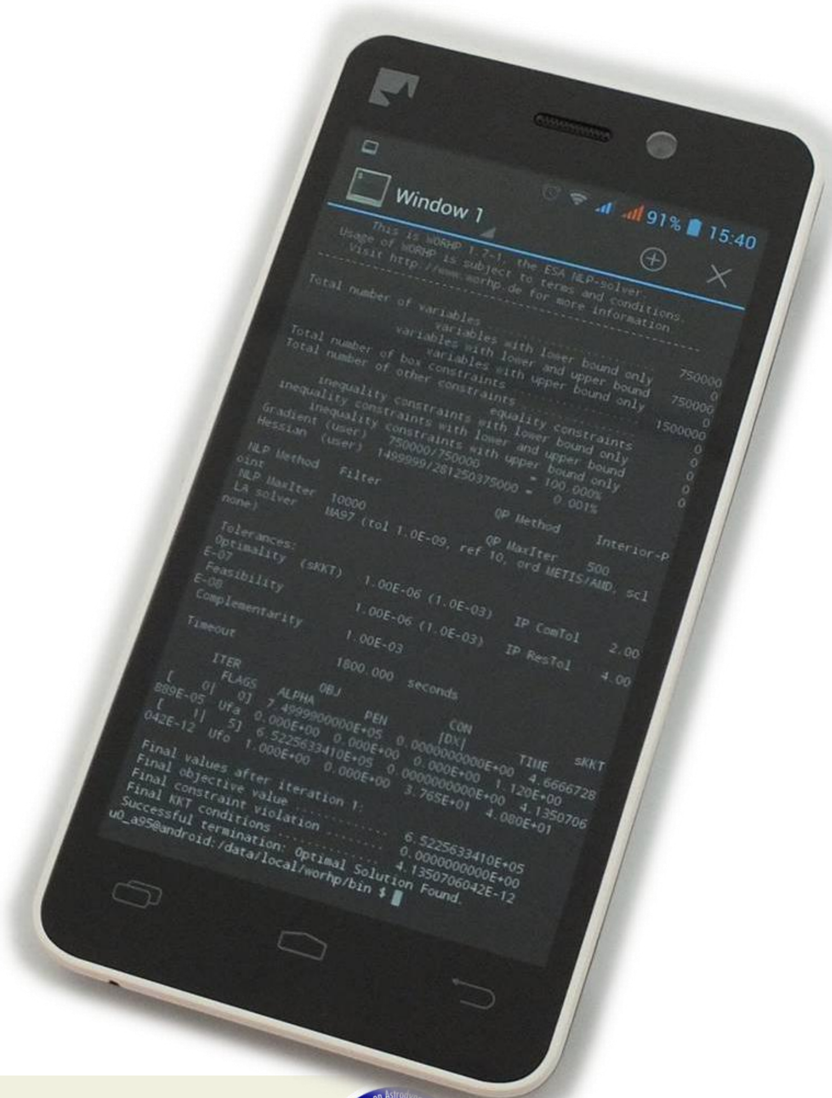


>600 users worldwide
>150 international users

WORHP

Developed at Universität Bremen*





Nonlinear Optimization

Problem formulation

Standard formulation

$$\begin{aligned} \min_{z \in \mathbb{R}^n} \quad & F(z) \\ \text{s.t.} \quad & G(z) \leq 0 \end{aligned}$$

WORHP formulation

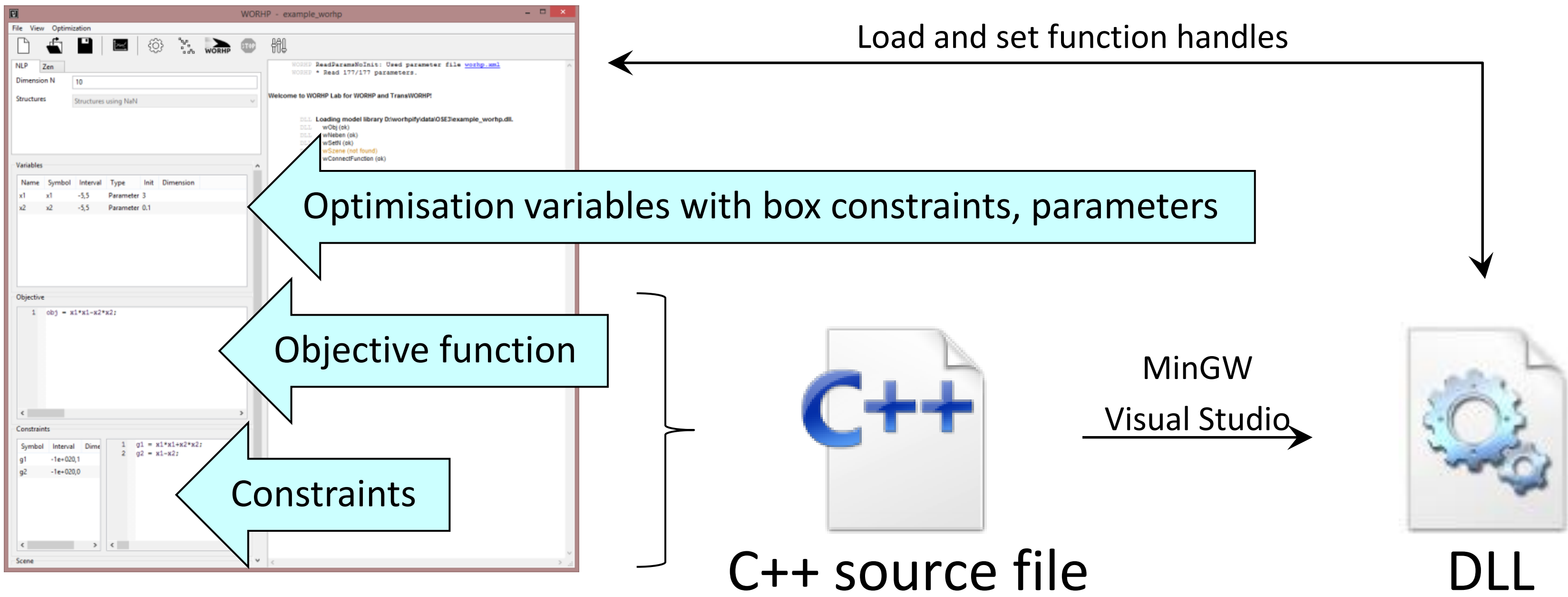
$$\begin{aligned} \min_{z \in \mathbb{R}^n} \quad & F(z) \\ \text{s.t.} \quad & l \leq z \leq u \\ & L \leq G(z) \leq U \end{aligned}$$

WORHP Lab

- Simplified interface to WORHP
  - hide initialization and reverse communication from user
  - for common tasks (e.g. data interpolation)
  - easy access to solution

- Showcase for features of WORHP
  - Industrial workshops
  - Education (university, school)

WORHP Lab


Load and set function handles
Optimisation variables with box constraints, parameters
Objective function
Constraints
MinGW
Visual Studio
C++ source file
DLL

Smooth surfaces

- Minimise difference of adjacent matrix entries where some entries are fixed

$$\min_{x \in \mathbb{R}^{n \times n}} \alpha \sum_{i=2}^{n} \sum_{j=1}^{n} (x_{i,j} - x_{i-1,j})^2 + \sum_{i=1}^{n} \sum_{j=2}^{n} (x_{i,j} - x_{i,j-1})^2$$

$$x_{i,j} \text{ fixed for } (i,j) \in I \subset \{1, \dots, n\}^2$$

WORHP Zen

Impact of small perturbations on optimal solution

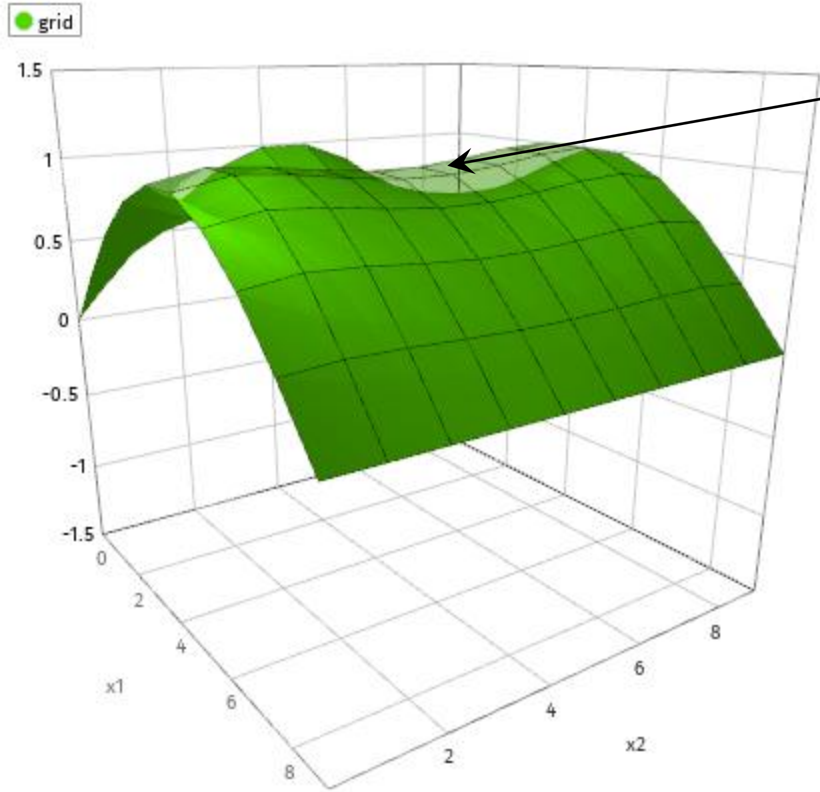
- Parametric Nonlinear Programming $(p_0, r_0, q_0) = (p_0, 0, 0)$

$$\min_{z \in \mathbb{R}^n} \quad F(z,p) + r^T z$$
$$\text{s.t.} \quad G(z,p) + q \leq 0$$

- Sensitivity derivatives of solution $\frac{\mathrm{d}z}{\mathrm{d}p}(p_0)$
- Extension of sensitivity theorem of Fiacco, Robinson
  - Eliminating NLP fractals (relaxation, regularisation)
  - Without measurable computational cost
  - For free: Additional sensitivity derivatives $\frac{\mathrm{d}z}{\mathrm{d}q}(0)$ and $\frac{\mathrm{d}z}{\mathrm{d}r}(0)$

$$\begin{pmatrix} \nabla_z{}^2 L & \nabla_z G^T \\ \Delta \nabla_z G & \Gamma \end{pmatrix} \cdot \begin{pmatrix} \frac{\mathrm{d}z}{\mathrm{d}p}(p_0) \\ \frac{\mathrm{d}\lambda}{\mathrm{d}p}(p_0) \end{pmatrix} = - \begin{pmatrix} \nabla_{zp} L \\ \Delta \nabla_p G \end{pmatrix}$$

Sensitivity analysis in objective

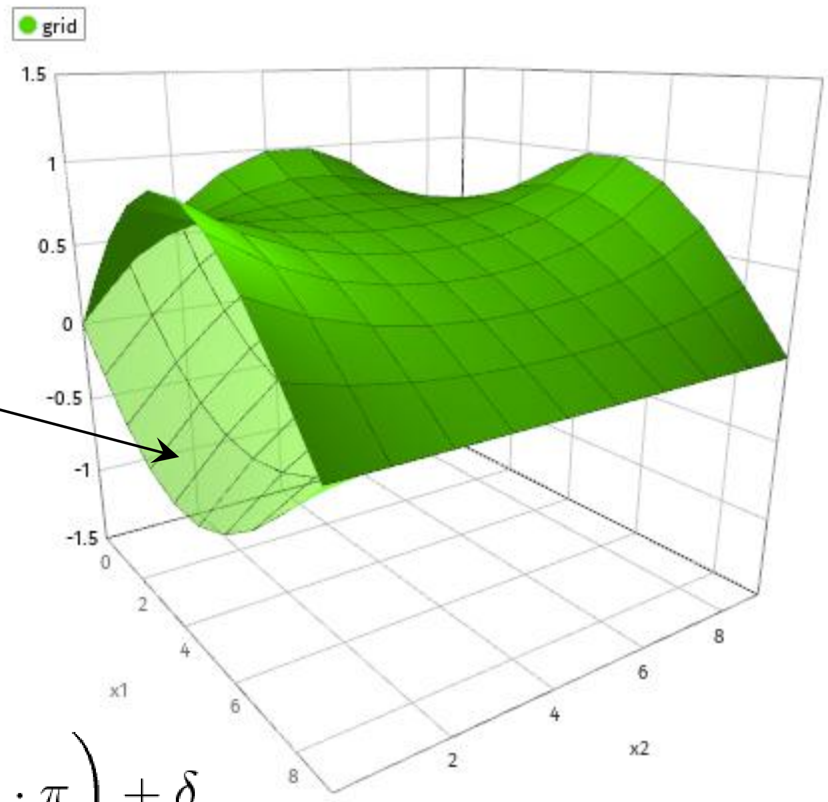


$$x + t\frac{dx}{d\alpha}(\alpha_0)$$

Change weighting parameters in objective

Linear approximation of Pareto front

Sensitivity analysis in constraints



$$x + t\frac{dx}{dA}(A_0)$$

Change amplitude of boundary

$$x_{i,1} = A\sin\left(\frac{i-1}{n-1}\cdot f\cdot\pi\right) + \delta$$

Mathematical Task

General problem formulation of optimal control

How do I have to control the motors, so that

a system is brought from an initial position

to a final position as good as possible

without overstressing?

Mathematical Task

General problem formulation of optimal control

$$\min_{u,x,t_f} \int_0^{t_f} g(x(t),u(t),t)\,\mathrm{d}t$$

$$\begin{aligned}
\text{s.t.} \quad \dot{x}(t) &= f(x(t),u(t),t), \quad && t \in [0; t_f] \\
\omega(x(0),x(t_f)) &= 0 && \\
C(x(t),u(t),t) &\le 0, && t \in [0; t_f]
\end{aligned}$$

Optimal Control Problem

Standard formulation and methods

$$\min_{u,x,t_f} \int_0^{t_f} g(x(t), u(t), t)\,\mathrm{d}t$$

$$\begin{aligned} \text{s.t.} \quad \dot{x}(t) &= f(x(t), u(t), t), \quad t \in [0; t_f] \\ \omega(x(0), x(t_f)) &= 0 \\ C(x(t), u(t), t) &\leq 0, \quad t \in [0; t_f] \end{aligned}$$

Indirect Methods

(First optimize, then discretize)

- Reformulation to a boundary value problem using necessary conditions

Direct Methods

(First discretize, then optimize)

- Reformulation to a nonlinear optimization problem by discretization

Direct Methods

Numerical solution by discretization of time variable

$$t \in [0, t_f] \to t \in \{0 = t_1 \leq t_2 \leq \dots t_N = t_f\}$$

$$h_i = t_{i+1} - t_i, \quad i = 1, \dots, N-1$$

Single Shooting/Multiple Shooting

Free Variables

$$u^i \approx u(t_i) \qquad x^0 \approx x(0)$$

Recursive integration

$$x^i = x^i(u^1, \dots, u^{i-1})$$

Small + dense

Full Discretization

Free Variables

$$u^i \approx u(t_i) \qquad x^i \approx x(t_i)$$

Integration scheme as constraints

$$0 = x^{i+1} - x^i - h_i \cdot f(x^i, u^i, t_i)$$

Large + sparse

Direct Methods

Numerical Solution by discretization of time variable and Euler's method, e.g.

$$t \in [0, t_f] \rightarrow t \in \{0 = t_1 \leq t_2 \leq \ldots t_N = t_f\}$$

$$h_i = t_{i+1} - t_i, \quad i = 1, \ldots, N-1$$

$$u^i \approx u(t_i) \qquad x^i \approx x(t_i)$$

f_i

$$\begin{aligned} \min_{u,x} \quad & \sum_{i=1}^{N-1} h_i \cdot g(x^i, u^i, t_i) \\ \text{s.t.} \quad & x^{i+1} = x^i + h_i \cdot f(x^i, u^i, t_i), \quad i = 1, \ldots, N-1 \\ & \omega(x^1, x^N) = 0 \\ & C(x^i, u^i, t_i) \leq 0, \quad i = 1, \ldots, N \end{aligned}$$

High dimensional NLP problem

Methods of Higher Order

Implicit methods for free

Euler's Method

$$0 = x^{i+1} - x^i - h_i f_i$$

Trapezoidal rule

$$0 = x^{i+1} - x^i - \frac{h_i}{2}\left(f_{i+1} + f_i\right)$$

Hermite-Simpson: additional point $t_{i+\frac{1}{2}} = \frac{1}{2}(t_{i+1} + t_i)$

$$0 = x^{i+\frac{1}{2}} - \frac{1}{2}(x^{i+1} + x^i) - \frac{h_i}{8}\left(f_i - f_{i+1}\right)$$

$$0 = x^{i+1} - x^i - \frac{h_i}{6}\left(f_{i+1} - 4f_{i+\frac{1}{2}} + f_i\right)$$

Precision of solution

$\mathcal{O}(h)$

$\mathcal{O}(h^2)$

$\mathcal{O}(h^4)$

Parking

Dynamic system

- States:
  - Position *x*, *y*
  - Orientation of car θ
  - Steering angle φ
  - Velocity *v*
- Controls
  - Rate of steering angle σ
  - Acceleration *u*
- Constants
  - Gear ratio *b*

$$
\begin{aligned}
\dot{x} &= v\cos\theta \\
\dot{y} &= v\sin\theta \\
\dot{v} &= u \\
\dot{\theta} &= \frac{v}{b}\sin\varphi \\
\dot{\varphi} &= \sigma
\end{aligned}
$$

Parking

Optimal control problem

- Objective: Free process time and "energy consumption"

$$\min t_f + \int_0^{t_f} u^2 + \sigma^2 \, dt$$

- Boundary conditions: Halt at initial and terminal point

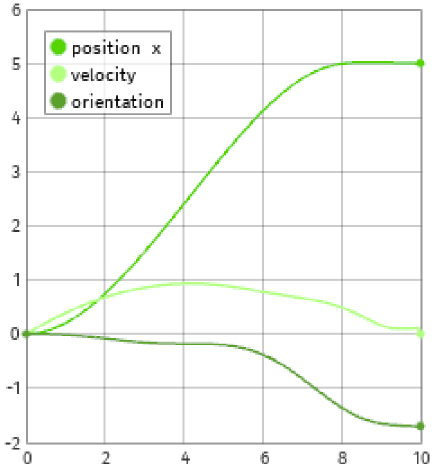
$$\begin{array}{ll} x(0) = 0 & x(t_f) = 5 \\ y(0) = 0 & y(t_f) = 2 \\ v(0) = 0 & v(t_f) = 0 \\ \theta(0) = 0 & \theta(t_f) = -\frac{\pi}{2} \\ \varphi(0) = 0 & \varphi(t_f) = 0 \end{array}$$

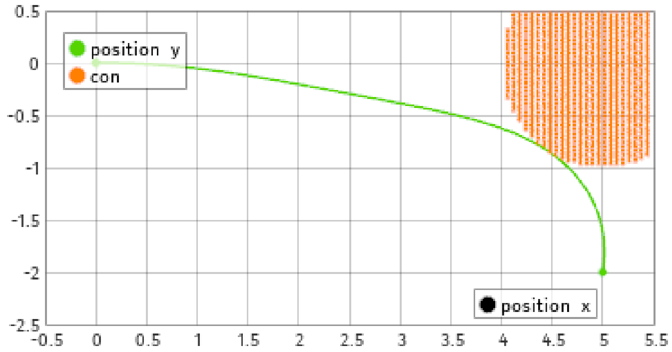
- Path constraints: Avoid (*x*,*y*) = (5,0)

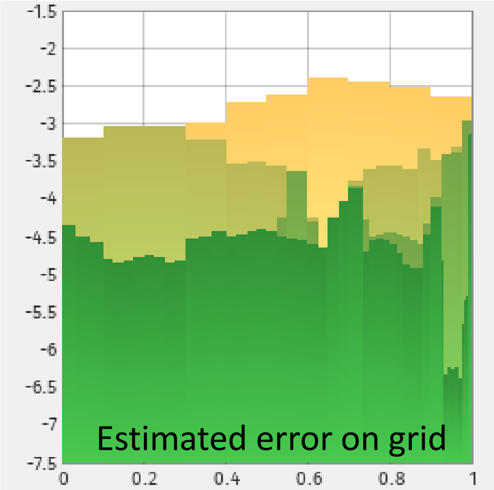
$$(x-5)^2 + y^2 \geq 1$$

Parking

Solution



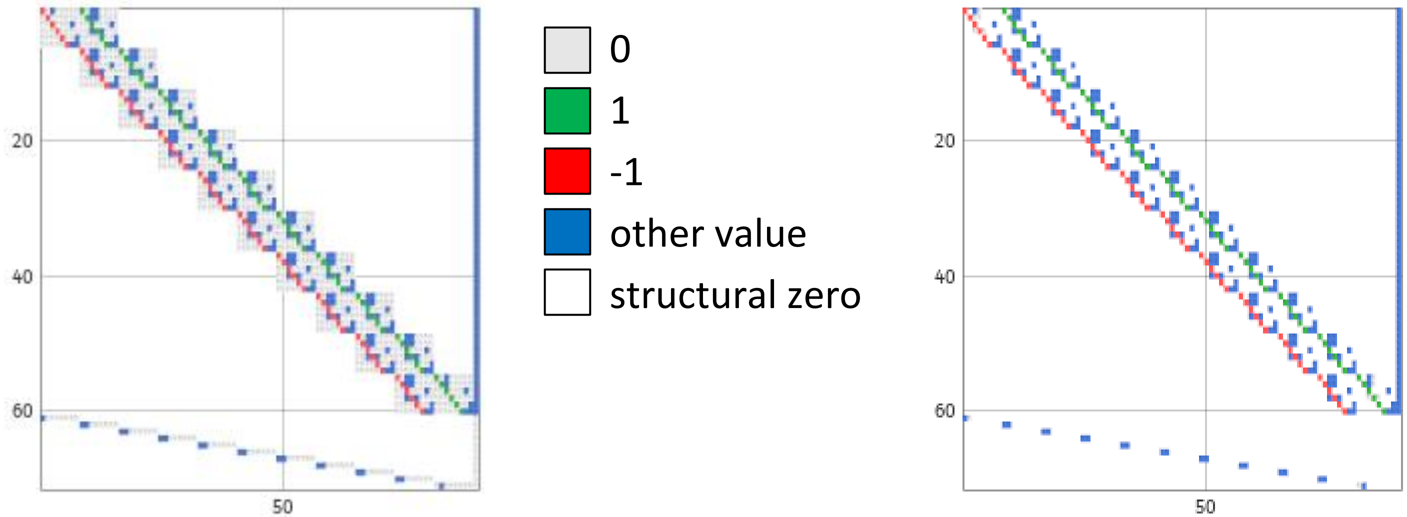




- Full discretization: Euler, Trapezoidal, Hermite Simpson
  - Adaptive grid points
- Multiple shooting

Matrix Structures

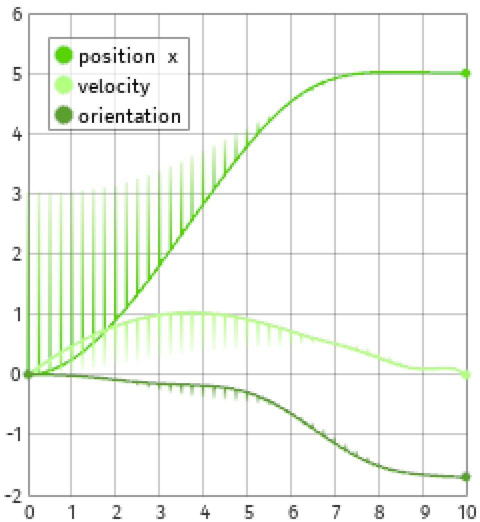
Automatic sparsity detection

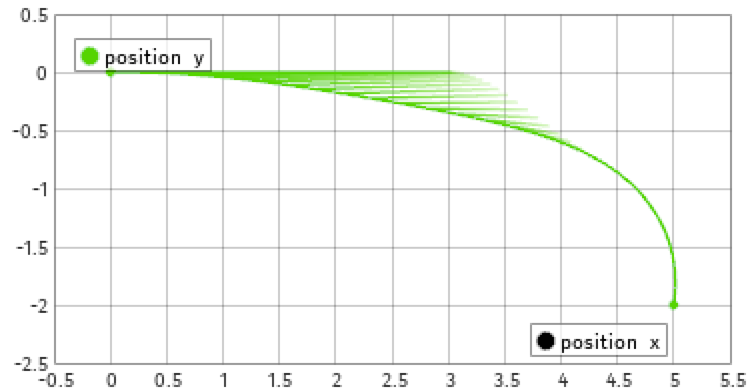


- Structure due to integration method
- Structure due to
  - sparse ode system
  - sparse path constraints

Sensitivity Analysis for Parking

Result from WORHP Zen





- Sensitivity of gear ratio
- Sensitivity of initial position x

Real-Time Optimal Control

Using sensitivity differentials of ode

1. Update control and state for $i = 1$

$$u^{[1]}(p) = u(p_0) + \frac{\mathrm{d}u}{\mathrm{d}p}(p_0) \cdot (p - p_0) \qquad x^{[1]}(p) = x(p_0) + \frac{\mathrm{d}x}{\mathrm{d}p}(p_0) \cdot (p - p_0)$$

2. Measure deviations $\Delta q^i := \left(x^{[i],j+1} - x^{[i],j} - h_j \cdot f\left(x^{[i],j}, u^{[i],j}, t_j\right)\right)_{j=1,\ldots,N-1}$

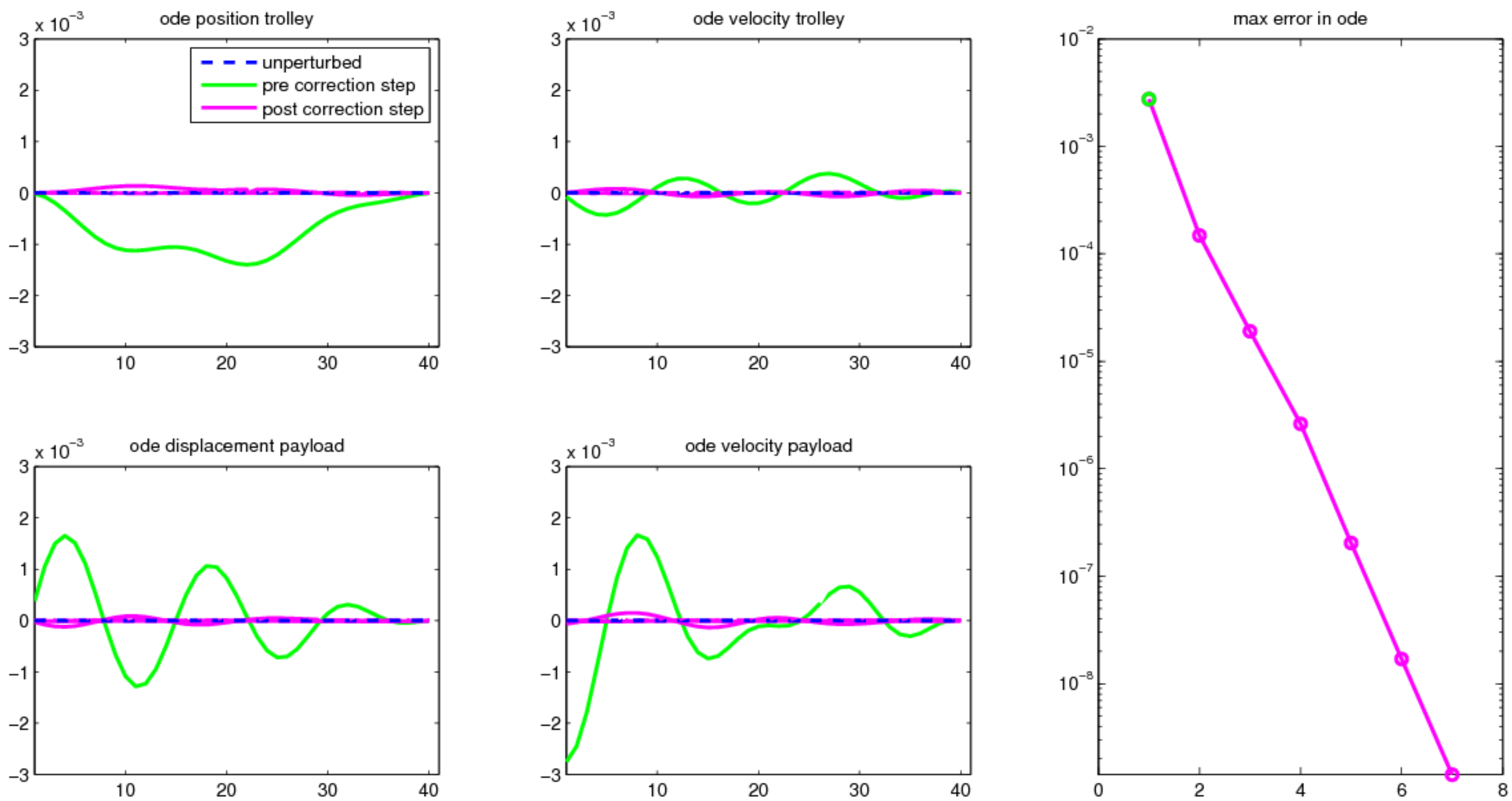
3. While $\Delta q^i > \varepsilon$

   a. Update control and state

$$u^{[i+1]}(p) = u^{[i]}(p) + \frac{\mathrm{d}u}{\mathrm{d}q}(0) \cdot \Delta q^i \qquad x^{[i+1]}(p) = x^{[i]}(p) + \frac{\mathrm{d}x}{\mathrm{d}q}(0) \cdot \Delta q^i$$

   b. Update deviation, $i = i + 1$

Real-Time Optimal Control



Tracking Problems

Industrial Application

Simulation environments
- real-time reaction to user input
- robot-based flight simulations

Assistance systems
- controller to simplify handling

- Hexapod-Robot
  - control=acceleration
  - 15 state functions
  - 7 control functions
- Predicted Pseudo-Tracking
- Optimization cycle <200 ms

Task unknown in advance

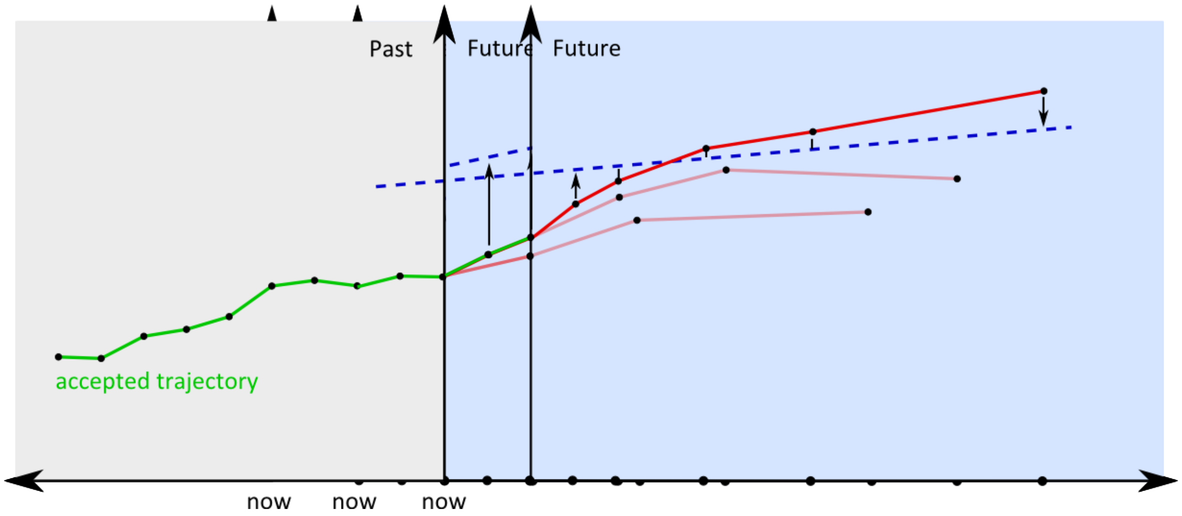
Task too complex for feedback controllers

- Crane System
- Tracking of user input

MPC with TransWORHP

Nonlinear Model Predictive Control

- Finite prediction horizon
- Accept only first part of iterative solution



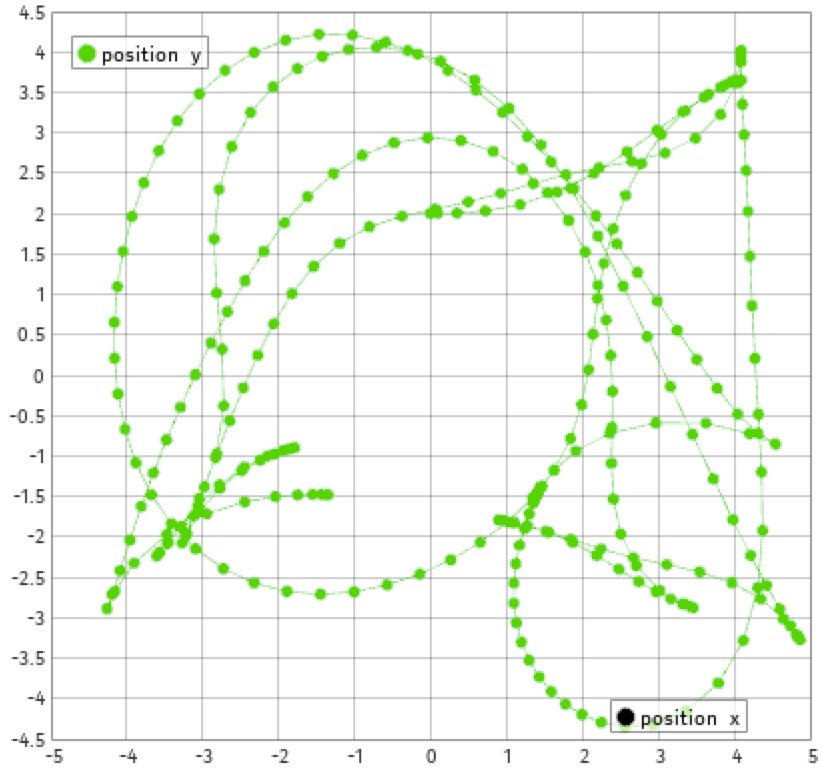
Chasing with a car

MPC

- Objective:
  - Reach target
  - “Stable” final position
- No boundary conditions

- Prediction time: 2 seconds
- Accepted time: 0.5 seconds

- Target moves on Lissajous figure



Conclusions

- WORHP Family
  - WORHP - TransWORHP - WORHP Zen - WORHP Lab
- Online and real-time solutions of OCP
  - WORHP/TransWORHP < 1000 ms
  - MPC for tracking < 100 ms
  - Real-time correction < 1 ms
- **Current Work.** Representative examples
  Automatic derivatives using operator overloading
  3d visualization
  Support of various user input devices (Joystick, Kinect)
- **Transfer.** e.g. Automotive, Logistics, Aerospace, Energy