Surrogate optimization for mixed-integer nonlinear problems of black box type in engineering applications

Thomas Hemker

Simulation, Systemoptimierung und Robotik
Fachbereich Informatik
Technische Universität Darmstadt

Blockseminar „Wissenschaftliches Rechnen“,
23. Januar 2008, Mittelberg, Österreich

The research group

- Prof. Oskar von Stryk

The research group

- Prof. Oskar von Stryk
- 11 researchers, PhD students
- 2 external PhD students

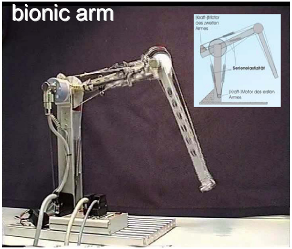
The research group

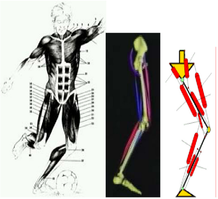
- Prof. Oskar von Stryk
- 11 researchers, PhD students
- 2 external PhD students
- 2 non scientific staff
  (secretary and IT infrastructure)

The research group

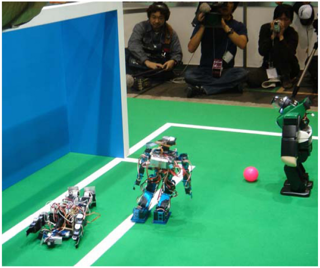
- Prof. Oskar von Stryk
- 11 researchers, PhD students
- 2 external PhD students
- 2 non scientific staff
  (secretary and IT infrastructure)
- about 10-20 students

Current research



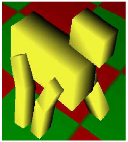


humaniod motor control

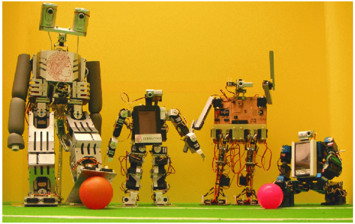


Robotics

cooperating teams of autonomous robots


four-legged robots

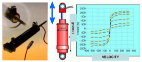


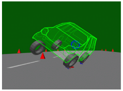
humanoid robots

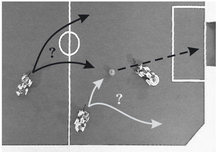
Current research

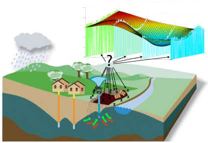
Optimization

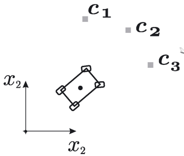
- Optimal control of dynamic systems
  - Gradient based
  - Mixed-integer
  - Including parameter estimation
- Parameter & simulation-based optimization
  - Mixed-integer and nonlinear
  - Gradient free
  - Robust, for noisy functions

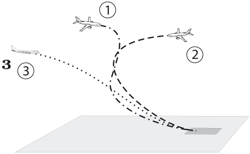










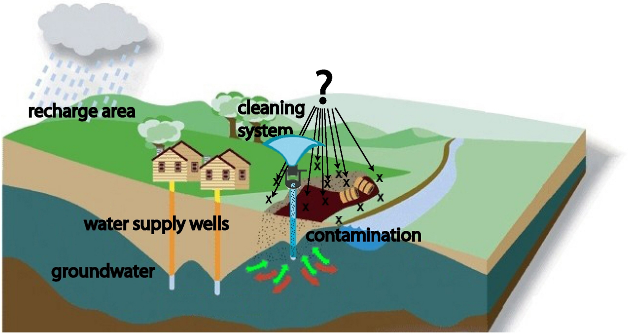


Education

TECHNISCHE
UNIVERSITÄT
DARMSTADT

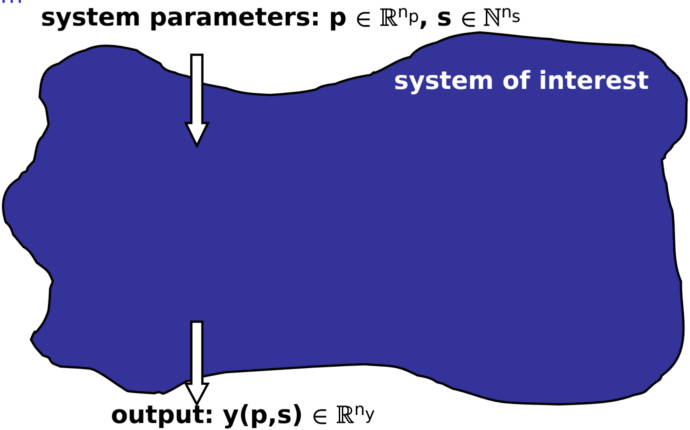
- Systems simulation and optimization:
  - principles of modeling and simulation
  - optimization of static and dynamic systems
- Robotics:
  - mobile and sensor guided robots
  - robotics 1 (foundations)
  - robotics 2 (mobility and autonomy)
- Practical courses, seminars and projects in robotics and optimization.

Design of a groundwater remediation system



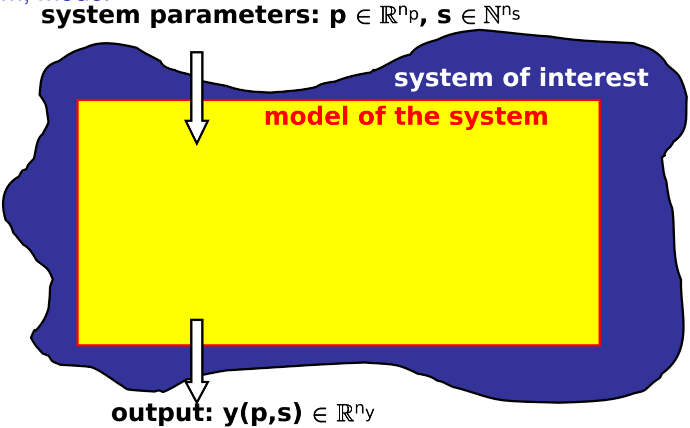
Problem introduction

System



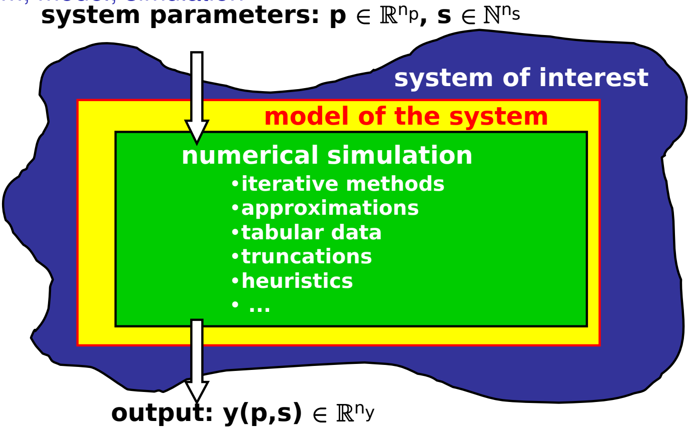
- Definition of the system boundary & surrounding.
- Determination of system properties.
- Part of interest.

System, model



- Formulation of the parts of interest.
- Resulting in an analytical description of the system.
- Always build for a special purpose (level of detail).

System, model, simulation



- Black box system, simulator either too much source code.
- Non-smooth/discontinuous underlying model.
- Simulation calls consume a lot computational time.

Model based optimization:

- All-at-once methods: tailored coupling of simulation and optimization methods.

White and grey box approaches:

- Simplified (physical) models (model replacement), e.g., space mapping techniques.

Code based optimization:

- Iterative simulation and optimization with ext./int. differentiation.

Robust black box optimization approaches:

- Interactive in-a-loop coupling.

Black box optimization

- Objective function arises from output $\boldsymbol{y}(\boldsymbol{p}, \boldsymbol{s})$.
- No additional information on process details is available.

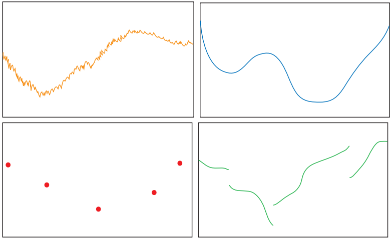
Robust
- Noisy black box output $\boldsymbol{y}(\boldsymbol{p}, \boldsymbol{s})$
- Errors, black box based constraints

Global
- Non-convex objective function f
- Multiple local minima

Direct
- No accurate gradient information
- Expensive objective function evaluations

Mixed-Integer
- Continuous, real-valued parameters
- Binary, integer-valued, categorical parameters

Minimize: $f(\boldsymbol{p}, \boldsymbol{s}, \boldsymbol{y}(\boldsymbol{p}, \boldsymbol{s}))$,
examples for f:



Common MINLP techniques

Methods: Generalized Benders decomposition,
Outer approximation, Extended cutting plane methods,
Branch-and-Bound, ... [e.g. Grossmann, Floudas].

Conditions: Corresponding gradient information,
relaxation of integer variable domain,
efficient solvers for underlying
MIPs and NLPs [e.g. CPLEX, SNOPT].

Problem: Not appropriate for the considered cases.

Gradient-free optimization approaches (1)

Random search methods: Genetic/Evolutionary Algorithms, Simulated Annealing, Tabu Search, Particle Swarm Methods, ...

- Improvement or not? Only qualitatively, not quantitatively. Not using the objective function values.
- New candidate by meta-heuristics out of evaluated ones.
- Easy to apply, without much problem information.
- No free lunch - often consuming very many function evaluations.

Gradient-free optimization approaches (2)

Sampling methods: Pattern Search (APPSPACK), Nelder-Mead, Implicit Filtering (IFFCO), Dividing Rectangulars (DIRECT), ...

- Grid based or grid free methods, with or without stencil.
- Some include function values explicitly, others are only qualitatively, not quantitatively.
- Often faster than random search - if they work.

Gradient-free optimization approaches (3)

Surrogate optimization methods:

process of abstraction:

real system → model and simulation → surrogate function

1. Approximation	Response surface, splines, kriging [Sacks], radial basis functions [Schoemaker], ...
2. Optimization	Fast Newton-type/gradient-based methods.
3. Design	One shot (DOE) and/or update strategies [Jones, Schonlau, Sasena].

Gradient-free optimization approaches (3)

Surrogate optimization methods:

process of abstraction:

real system → model and simulation → surrogate function

1. Approximation	Response surface, splines, kriging [Sacks], radial basis functions [Schoemaker], ...
2. Optimization	Fast Newton-type/gradient-based methods.
3. Design	One shot (DOE) and/or update strategies [Jones, Schonlau, Sasena].

- Smooth out noise, cheap to minimize.
- May not catch the main characteristics of the original problem.
- *They are not*: surrogate models, metamodels, model replacements, low-fidelity models, or space mapping techniques (→ grey/white box approaches).

... and for the here considered problems?

- MINLP methods do not work for our kind of problems.
- GA & EA by high computational costs.
- Pattern search with user-defined neighborhoods (MVP).

Idea: Combining surrogate optimization approaches and analytically/numerically approved MINLP-methods.

Surrogate optimization for mixed-integer nonlinear problems including black boxes

Resulting optimization problem

Variable space: $\boldsymbol{p} \in \Omega_p \subset \mathbb{R}^{n_p}$, and $\boldsymbol{s} \in \Omega_s \subset \mathbb{N}^{n_s}$.

Numerical simulation: $\boldsymbol{y}(\boldsymbol{p}, \boldsymbol{s}) : \Omega_p \times \Omega_s \to \Omega_y \subset \mathbb{R}^{n_y}$.

- objective function:

$$\min f(\boldsymbol{p}, \boldsymbol{s}, \boldsymbol{y}(\boldsymbol{p}, \boldsymbol{s})), \quad f : \Omega_p \times \Omega_s \times \Omega_y \to \mathbb{R}.$$

- Linear and bound constraints:

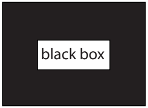
$$\boldsymbol{A} \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{s} \end{pmatrix} \leq \boldsymbol{b}, \quad \boldsymbol{A} \in \mathbb{R}^{n_b \times (n_p + n_s)}, \quad \boldsymbol{b} \in \mathbb{R}^{n_b}.$$

- Nonlinear constraints:

$$\boldsymbol{g}(\boldsymbol{p}, \boldsymbol{s}, \boldsymbol{y}(\boldsymbol{p}, \boldsymbol{s})) \leq 0, \quad g : \Omega_p \times \Omega_s \times \Omega_y \to \mathbb{R}^{n_g}$$

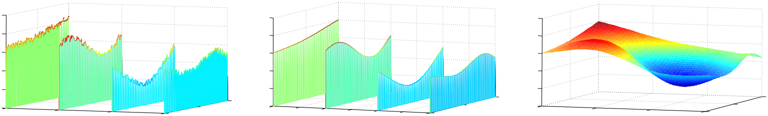
$$\text{with:} \quad \begin{array}{ll} g_i(\boldsymbol{p}, \boldsymbol{s}) & \leq 0, \text{ for } i \in I_{ex} \\ g_i(\boldsymbol{p}, \boldsymbol{s}, \boldsymbol{y}(\boldsymbol{p}, \boldsymbol{s})) & \leq 0, \text{ for } i \in I_{im} \end{array}$$

Surrogate functions for mixed-integer problems



surrogate function

$$\boldsymbol{y} : \Omega_p \times \Omega_s \rightarrow \Omega_y \subset \mathbb{R}^{n_y} \quad \Rightarrow \quad \hat{\boldsymbol{y}}(\boldsymbol{p}, \boldsymbol{s}) : \mathbb{R}^{n_p} \times \mathbb{R}^{n_s} \rightarrow \mathbb{R}^{n_y}$$



$\Rightarrow$

DACE - for mixed-integer problems

- Design and analysis of computer experiments (DACE):

$$\hat{\boldsymbol{y}} : \mathrm{IR}^{n_p} \times \mathrm{IR}^{n_s} \to \mathrm{IR}^{n_y}, \qquad \hat{\boldsymbol{y}}(\boldsymbol{p}, \boldsymbol{s}) = \boldsymbol{v}(\boldsymbol{p}, \boldsymbol{s})^T \beta + Z(\boldsymbol{p}, \boldsymbol{s}),$$

$$\text{with} \quad \boldsymbol{y}(\boldsymbol{p}_i, \boldsymbol{s}_i) = \hat{\boldsymbol{y}}(\boldsymbol{p}_i, \boldsymbol{s}_i), \quad \forall i = 1, ..., n.$$

- Linear combination of basis functions: $\boldsymbol{v}^T \beta$.
- $Z(\boldsymbol{p}, \boldsymbol{s})$ is a stationary Gaussian random process, mean zero:

$$Cov[Z(\boldsymbol{p}_l, \boldsymbol{s}_l), Z(\boldsymbol{p}_k, \boldsymbol{s}_k)] = \sigma^2 R((\boldsymbol{p}_l, \boldsymbol{s}_l), (\boldsymbol{p}_k, \boldsymbol{s}_k)).$$

- $MSE(\hat{\boldsymbol{y}})$ describes the expected mean square error.

DACE - correlation function

For an analytic response and differentiability of $\hat{\boldsymbol{y}}$:

- R is product of 1-dimensional stationary functions

$$R\left(\begin{pmatrix} \boldsymbol{p}_l \\ \boldsymbol{s}_l \end{pmatrix}, \begin{pmatrix} \boldsymbol{p}_k \\ \boldsymbol{s}_k \end{pmatrix}\right) = \prod_{j=1}^{(n_p+n_s)} R(d_j) = \prod_{j=1}^{(n_p+n_s)} e^{-\theta_j d_j^2},$$

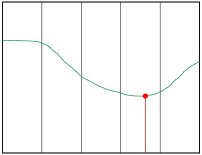
$$\text{with } d_j = \begin{cases} |p_{l,j} - p_{k,j}| & \text{for } j = 1, \ldots, n_p \\ |s_{l,j} - s_{k,j}| & \text{for } j = n_p + 1, \ldots, n_p + n_s \end{cases}$$

Parameter estimation:

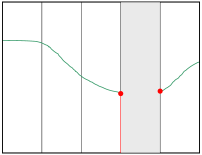
- θ and σ are maximum likelihood estimates.

MINLP on the surrogate problem

- Basic idea of Branch-and-Bound methods:



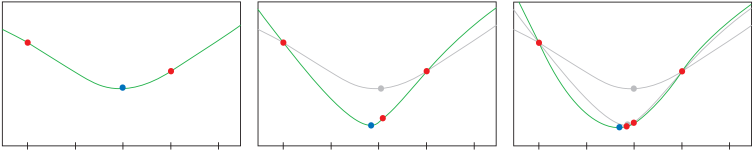
Solve the relaxed real-valued problem



Add constraints and solve again

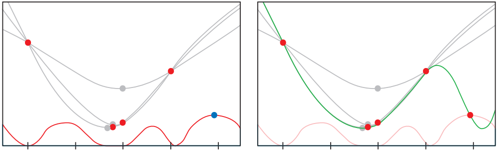
- How to solve the relaxed problems fast?
  - Sequential quadratic programming, highly effective for NLP.
  - Efficient implementations available, e.g., SNOPT.
  - Includes box-, linear- and nonlinear constraints.

Sequential update procedure

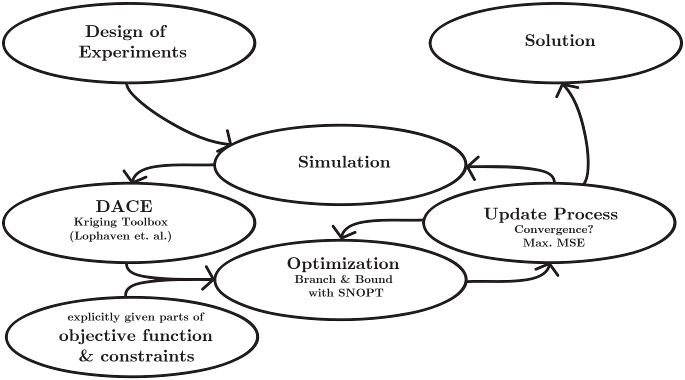


Update: Minimize until: $\left| \begin{pmatrix} \boldsymbol{p}^* \\ \boldsymbol{s}^* \end{pmatrix} - \begin{pmatrix} \boldsymbol{p}_i \\ \boldsymbol{s}_i \end{pmatrix} \right| \leq \epsilon$ for $i \in \{1, ..., n\}$.

Global: Maximize the MSE to get the next candidate.



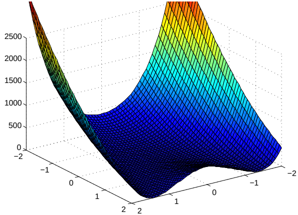
Resulting implementation layout

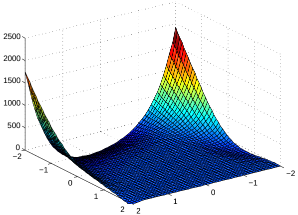


Example - Rosenbrock's function

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2, \; f(\boldsymbol{x}^*) = 0, \; \boldsymbol{x}^* = (1, 1)^T$$

The first 50 iterations:



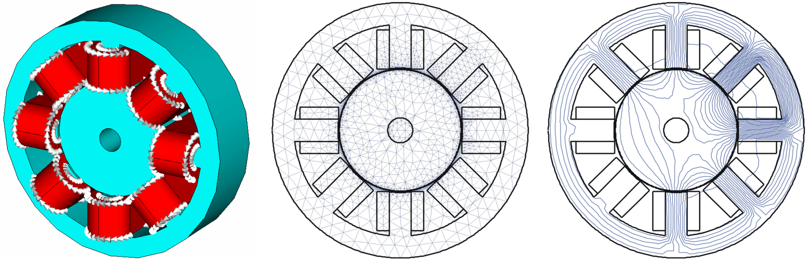


Original | Surrogate | Difference

Result: $f(\boldsymbol{x}^{(39)}) = 1.581 \cdot 10^{-5}, \; \boldsymbol{x}^{(39)} = (1.0031, \; 1.0064)^T$

Optimization results for engineering applications

Design of a magnetic bearing



1. 3D model defined by width and length of the poles.
2. Meshing of the construction (of CAD data).
3. Forces by electromagnetic field simulation.

Optimization problem

- Objective function: costs for steel and copper

$$\min \quad f(p_l, p_w) := y_{costs}.$$

- Implicit constraints: forces

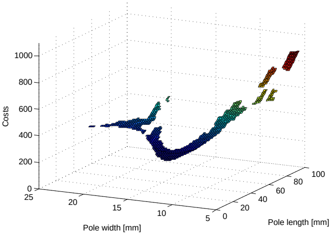
$$\begin{aligned} g_1(p_l, p_w) &= (y_{direct} - y_{quad})^2 &\leq& \quad 10^4, \\ g_2(p_l, p_w) &= -y_{direct} - y_{quad} &\leq& \quad -200. \end{aligned}$$

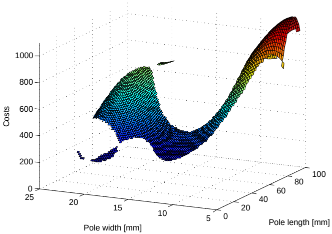
- Explicit constraints: geometric requirement

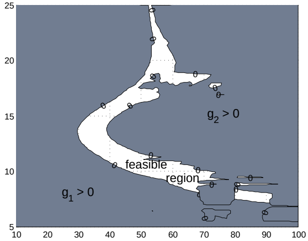
$$\begin{aligned} g_3(p_l, p_w) &= -r_s \sin(\tfrac{p_p}{2}) + \tfrac{p_w}{2} + \epsilon_{tol} &\leq& \quad 0, \\ g_4(p_l, p_w) &= \epsilon_{tol} - p_l &\leq& \quad 0, \end{aligned}$$

$$10 \leq \quad p_l \quad \leq 100, \qquad 5 \leq \quad p_w \quad \leq 25.$$

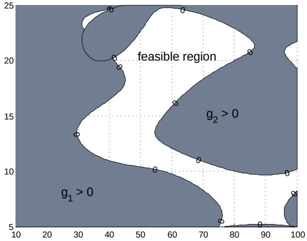
Simulation output (100x100 grid) vs. surrogate functions after 36 simulations







min: 206.6279

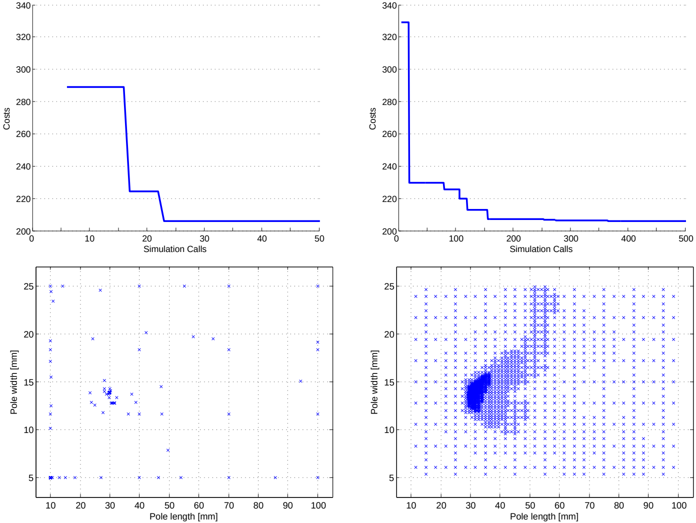


min: 205.8395

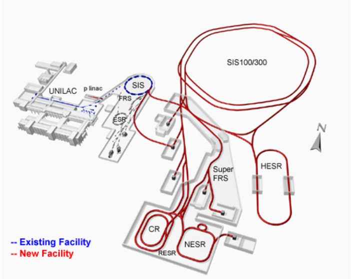
Surrogate optimization vs. DIRECT (1)

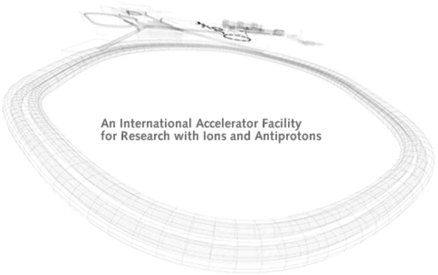
Number of iteration	(p_l, p_w) Surr. Opt.	Costs Surr. Opt.	(p_l, p_w) DIRECT	Costs DIRECT
6	(40, 11.6667)	289.1745	(55, 21.6667)	329.1087
17	(32.2393, 13.3958)	224.2539	(55, 21.6667)	329.1087
23	(29.8582, 13.8585)	205.9889	(35, 15)	229.7703
24	(29.8796, 13.8954)	205.8434	(35, 15)	229.7703
33	(29.8794, 13.8955)	205.8407	(35, 15)	229.7703
36	(29.8792, 13.8956)	205.8395	(35, 15)	229.7703
⋮			⋮	⋮
419			(29.8148, 13.8294)	205.9421
⋮			⋮	⋮
2579			(29.8880, 13.9026)	205.8404

Surrogate optimization vs. DIRECT (2)



Design optimization of superconductive magnets for particle accelerators

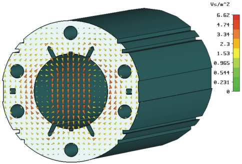


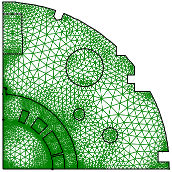


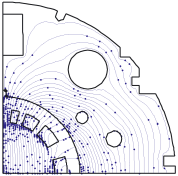
Background:

- New accelerator facility at GSI, Darmstadt, [Darmstadtium 110, 1994].
- Design process for the devices of the magnets is not finished.
- Objective: Homogenous magnetic field for particle tracking.

Problem Details







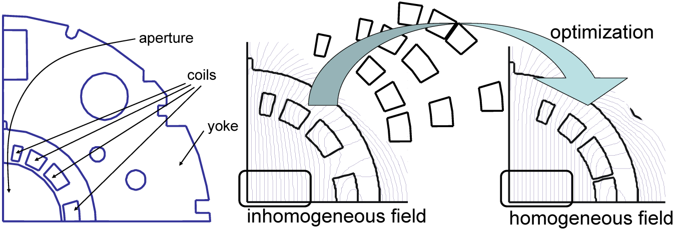
- Objective function:

$$\min f(\boldsymbol{p}, \boldsymbol{s}) = y(\boldsymbol{p}, \boldsymbol{s}), \; M : \mathbb{R}^{n_p} \times \mathbb{N}^{n_s} \rightarrow \mathbb{R}.$$

- y($\boldsymbol{p}$,$\boldsymbol{s}$) is a 2D FE-simulation output, describing level of homogenity of the magnetic field of the aperture field.

Optimization problem formulation (1)

Distribute a certain number of turns, collected in turn blocks such that the dipole field in the magnet aperture is homogeneous.



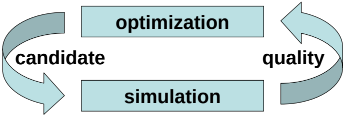
- Vector $\boldsymbol{p} \in \mathbb{R}^3$ the azimuthal position of each block.
- Vector $\boldsymbol{s} \in \mathbb{N}^3$ describes the number of turns on each coil.
- $\boldsymbol{s}$ is a non-relaxable integer design parameter vector.

Optimization problem formulation (2)

- Box and linear constraints: $\boldsymbol{A}\begin{pmatrix}\boldsymbol{p}\\ \boldsymbol{s}\end{pmatrix} \leq \boldsymbol{b}$

$$\text{with: } \boldsymbol{p} \in \mathbb{R}^{n_p},\ \boldsymbol{s} \in \mathbb{N}^{n_s},\ \boldsymbol{b} \in \mathbb{R}^{n_b},\ \boldsymbol{A} \in \mathbb{R}^{(n_p+n_s)\times n_b},$$
$$n_p, n_s, n_b \in \mathbb{N}.$$

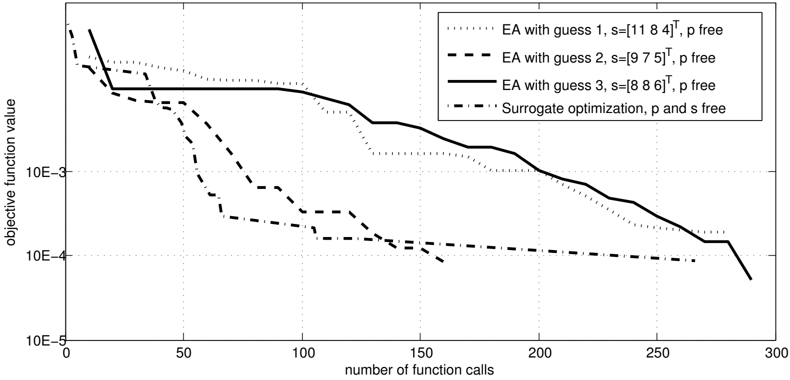
- Electromagnetic field simulation and optimization are running in a loop.



- Same simulation software package as used for the magnetic bearing problem.

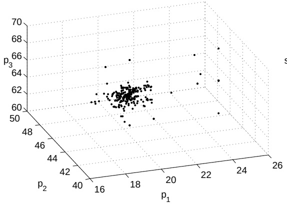
Numerical results (1)

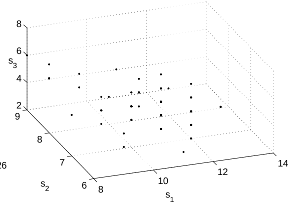
Surrogate approach vs. EA without any discrete variable.



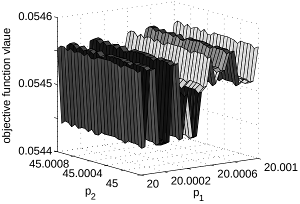
Numerical results (2)

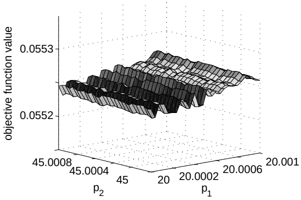
Distribution of evaluated designs:





Noise without and with mesh-adaption:



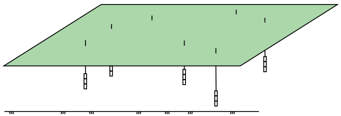


The Community Problems

- Set of typical scenarios for sub-surface flow simulations.
- Guideline for simulation and optimization studies.

[Mayer et al., 2002]

The general scenarios are:



- Well field design,
- Saltwater intrusion,
- Remediation by pump and treat systems,
- Remediation by hydraulic capturing.

3 Examples are available for simulation with MODFLOW:
`http://www4.ncsu.edu/~ctk/community.html`

Objective function

Total costs of ownership:

$$f(h,s,x,y,Q) = \underbrace{\sum_{i=1}^{n} s_i c_0 d_i^{b_0} + \sum_{Q_i<0} s_i c_1 |1.5 Q_i|^{b_1} (z_{gs} - h^{min})^{b_2}}_{f^{ex1}}$$

$$+ \overbrace{\int_0^{t_f} \sum_{i,Q_i \geq 0}^{n} s_i c_3 Q_i \, dt}^{f^{ex2}} + \underbrace{\int_0^{t_f} \sum_{i,Q_i \leq 0}^{n} s_i c_2 Q_i (h_i - z_{gs}) \, dt}_{f^{im}}$$

- h is simulation-based, all other elements are explicitly given.
- It follows: $\min f = f^{ex1} + f^{ex2} + f^{im}$.

Constraints

- Box:

$$(s_{min}, x_{min}, y_{min}, Q_{min}) \leq (s_i, x_i, y_i, Q_i) \leq (s_{max}, x_{max}, y_{max}, Q_{max}).$$

- Explicitly given:

$$Q_{T,max} \leq Q_T = \sum_{i=1}^{n} s_i Q_i \leq Q_{T,min},$$

$$\min(\max(|\overline{x}_i - \overline{x}_j|, |\overline{y}_i - \overline{y}_j|)) \geq s_i s_j \delta, \quad \forall\, i,j = 1,...,n \;\wedge\; i < j.$$

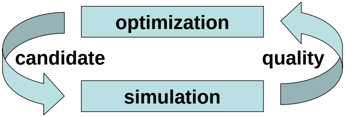
- Implicit, depending on simulation output h:

$$h_{min} \leq h_i \leq h_{max}, \qquad i = 1,...,n,$$

$$h_i - h_{i+l} \geq d, \qquad i = n+j,\; j = 1,...,l.$$

Implementation details

- Flow simulation by MODFLOW (the expensive part).
- Simulation and optimization running in a loop.



- 2 well-field design scenarios, 1 hydraulic capturing scenario.
- Initial designs as proposed by Fowler (reference solution).

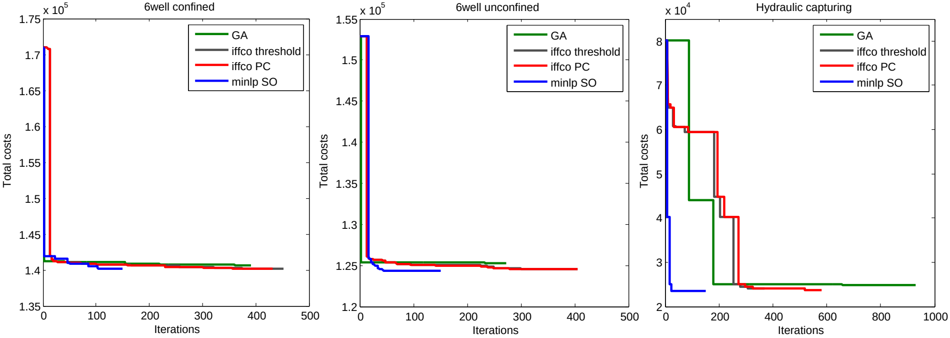
Numerical results (1)

- Dimension: well-field design: $n_s = 6$, hydraulic capturing: $n_s = 4$.
- Costs for best found system design (number of Modflow calls):

Problem formulation	Optimization method	well-field confined	well-field unconfined	hydraulic capturing
Inact.-well threshold	IFFCO	140,175 (362)	124,527 (320)	24,032 (363)
Penalty coefficients	IFFCO	140,190 (433)	124,512 (316)	23,640 (574)
Mixed-integer	NSGA-II	140,610 (391)	125,226 (273)	24,854 (659)
Mixed-integer	Sur Opt	140,159 (113)	124,387 (87)	23,491 (22)

- All approaches found feasible final designs.
- Almost no simulation errors.

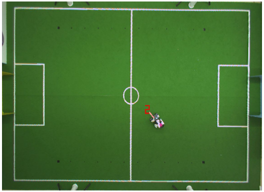
Numerical results (2)



- All approaches found solutions of the same structure.
- A lot unfeasible simulations calls regarding to the implicit constraints.

Walking optimization of 4-legged Sony Aibos

- Aim: Improvement of forward walking and turning speed.
- Hardware in the loop optimization with APPSPACK.
- Optimization problem dimension: 31.

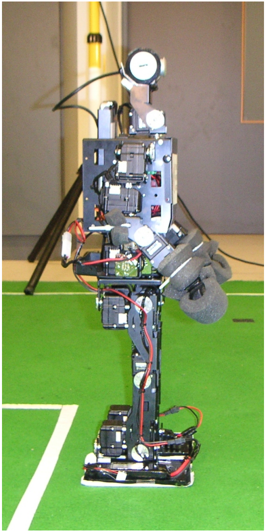


Walking optimization of 4-legged Sony Aibos

- Aim: Improvement of forward walking and turning speed.
- Hardware in the loop optimization with APPSPACK.
- Optimization problem dimension: 31.

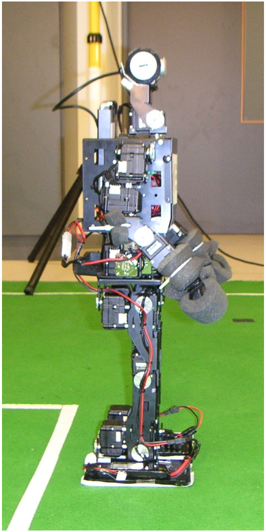
  - Forward: from 40 to 43 cm/sec with 83 Iterations.
  - Turning: from 120 to 180 deg/sec in 206 iterations.

The humanoid robot Bruno



- 55 cm tall, 21 joints, 13 directly relevant for the walking motion, 6 in each leg.
- Equipped with a pocket PC and two cameras.
- Trajectories generated from a supplied parameter vector by inverse kinematics.
- Gyros, accelerometers, and joint controller directly used for correction/stabilization.

The humanoid robot Bruno



- 55 cm tall, 21 joints, 13 directly relevant for the walking motion, 6 in each leg.
- Equipped with a pocket PC and two cameras.
- Trajectories generated from a supplied parameter vector by inverse kinematics.
- Gyros, accelerometers, and joint controller directly used for correction/stabilization.

5 dim optimization problem:

- Relation of center of mass to front and rear leg.
- Lateral position, roll angle and height of the foot during swinging phase.
- Pitch of the upper body.

Interactive hardware in the loop optimization

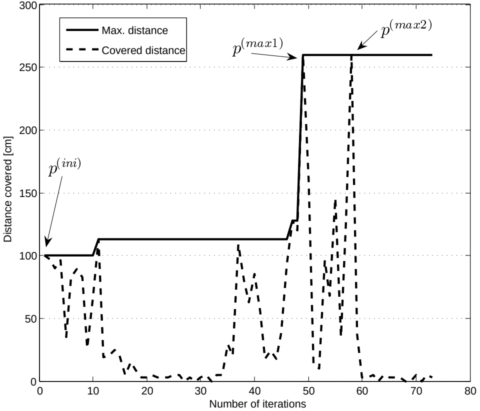
Objective function: Measured distance that the robot covers,until it stops or falls.

- Only box constraints for the parameter domain, robustness is included implicitly.

- Bruno starts with a small step length.
- Every two steps increased by 5 *mm*.
- After 52 steps the max. step length is reached.
- The optimization is done with the proposed SO approach.

Experimental results

- After 58 walking experiments 2 good sets were found, covered distance of 260 *cm*.
- Step frequency is maximized afterwards. Max. speed $> 30\ cm/sec$.



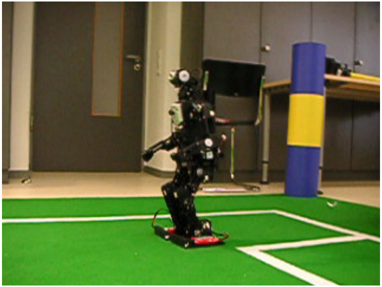
Experimental results

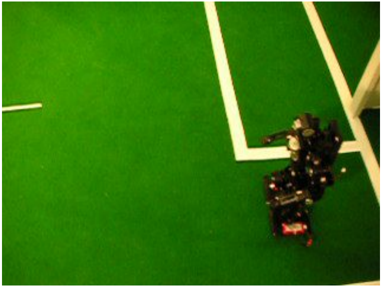
- After 58 walking experiments 2 good sets were found, covered distance of 260 *cm*.
- Step frequency is maximized afterwards. Max. speed $> 30\ cm/sec$.

Initial parameter set → Optimized parameter set

After design modifications

- Measured average speed for a 2 m distance: about 44 cm/sec.
- Emphasizes stability for the determined parameter set.





Summary

- A general optimization problem formulation for increasingly important black box or simulation-based problems.
- Sequential optimization approach for mixed-integer non relaxable problems.
- Proved successful with benchmark and real world optimization problems including black box parts.

Acknowledgement - lots of help from:

H. De Gersem, T. Weiland	Institut für elektromagnetische Felder, Technische Universität Darmstadt
K. Fowler	Dept. of Mathematics and Computer Science, Clarkson University
M. Farthing	Environmental Sciences and Engineering, The University of North Carolina at Chapel Hill
H. Sakamoto	Hajime Research Institute, Osaka

Parts of this research were supported by the Computational Engineering Research Center of the Technische Universität Darmstadt

Contact:

Thomas Hemker hemker@sim.tu-darmstadt.de

Simulation, Systems Optimization and Robotics

Department of Computer Science

Technische Universität Darmstadt

www.sim-opt.de