Optimal control

Katja Mombaur
Institut für Technische Informatik, Universität Heidelberg

Robotics 2 - 24 June 2019

Contents of the lecture

- Formulation of optimal control problems
- Numerical solution of optimal control problems with the direct multiple shooting method
- Solution of example problems with MUSCOD

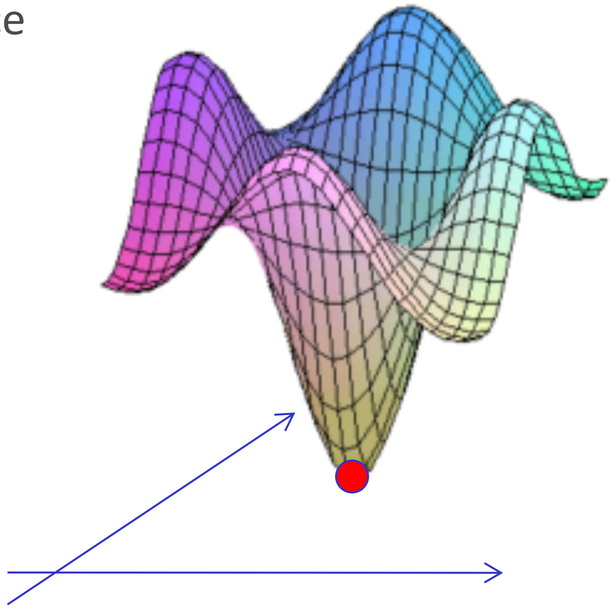
Properties of "classical" optimization problems, e.g. NLP

- Optimization variables are from some finite dimensional space

$$\text{e.g. } x \in \mathbb{R}^n$$

- The result is a point in finite dimensional space

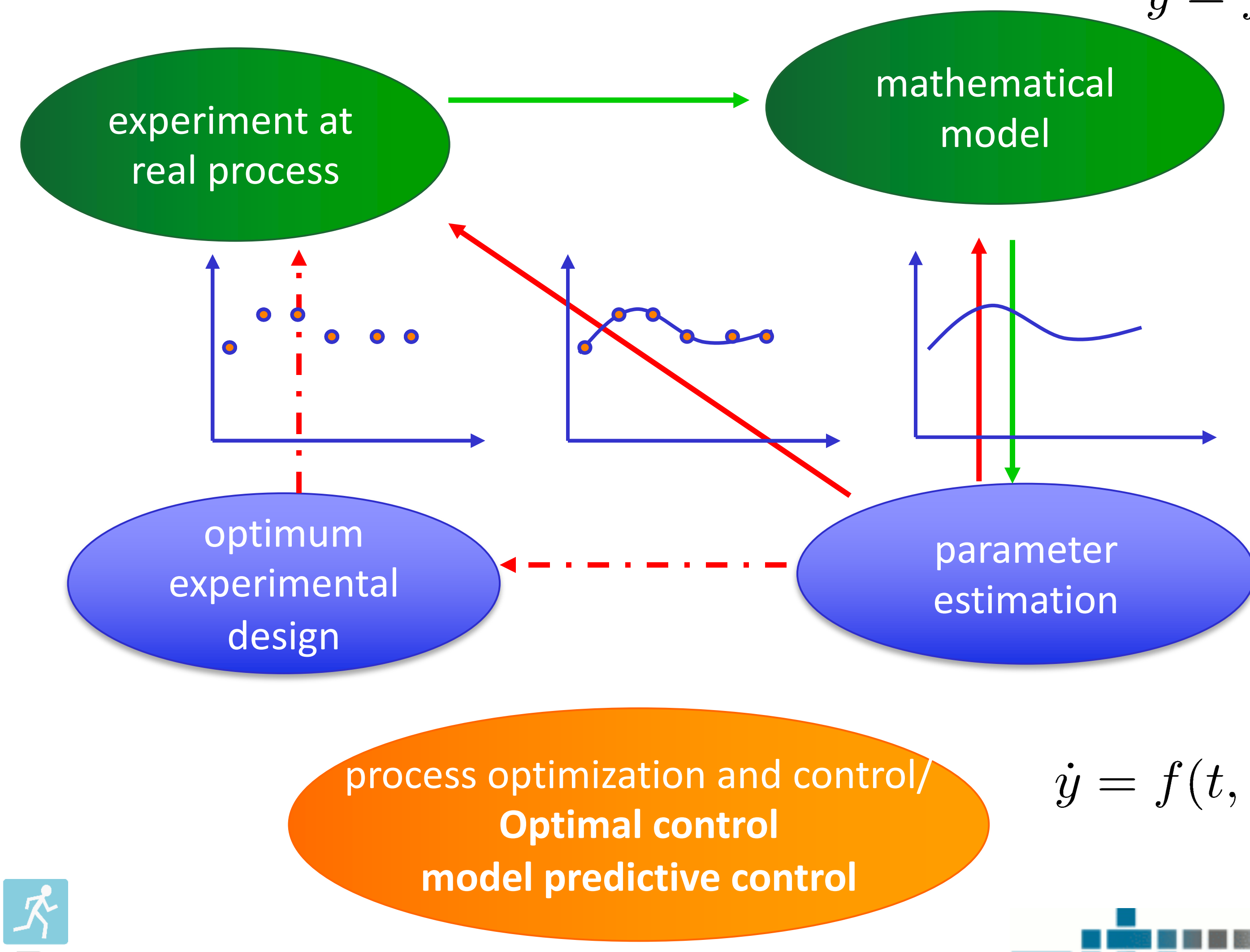
$$x^* = \begin{pmatrix} x_1^* \\ x_2^* \\ x_3^* \\ .. \\ x_n^* \end{pmatrix}$$



- This will change for the next classes of problems!

Overview: Optimization problems with dynamic process models

$$\dot{y} = f(t, y(t), p)$$



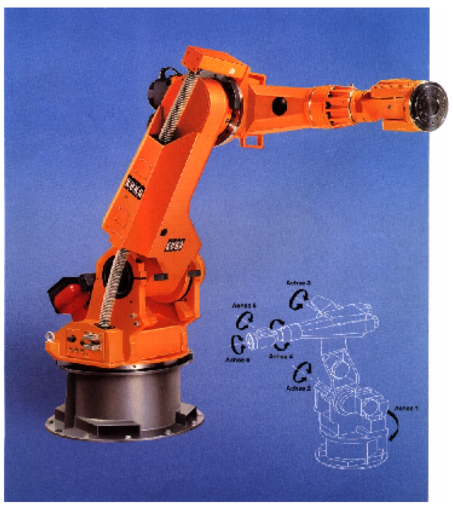
$$\dot{y} = f(t, y(t), u(t), p)$$

Reminder: general form of optimal control problem

Optimal control problems =
Optimal choice of entry variables of dynamic processes / systems

System dynamics can be manipulated by controls and parameters:

$$\boxed{\dot{y} = f(t, y(t), u(t), p)}$$



- simulation interval: $[t_0, t_{\text{end}}]$
- time $t \in [t_0, t_{\text{end}}]$
- state $y(t) \in \mathbb{R}^{n_z}$
- controls $u(t) \in \mathbb{R}^{n_u}$
- design parameters $p \in \mathbb{R}^{n_p}$

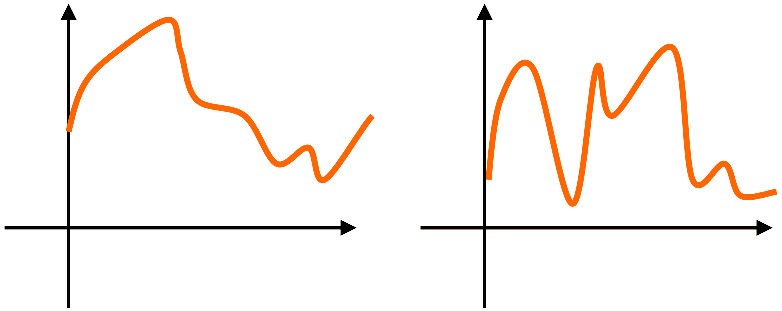
Characteristics of optimal control problems

- **Optimal control problems**

$$
\begin{aligned}
\min_{x(\cdot),u(\cdot),p,T} \quad & \int_0^T \phi(x(t),u(t),p)\,dt + \ \Phi(T,x(T),p) \\
\text{s. t.} \quad & \dot{x}(t) = (t, x(t), u(t), p) \\
& g(t, x(t), u(t), p) \geq 0 \\
& r_{eq}(x(0), .., x(T), p) = 0 \\
& r_{ineq}(x(0), .., x(T), p) \geq 0.
\end{aligned}
$$

- **State and control variables are functions in time**

  (infinite dimensional variables)

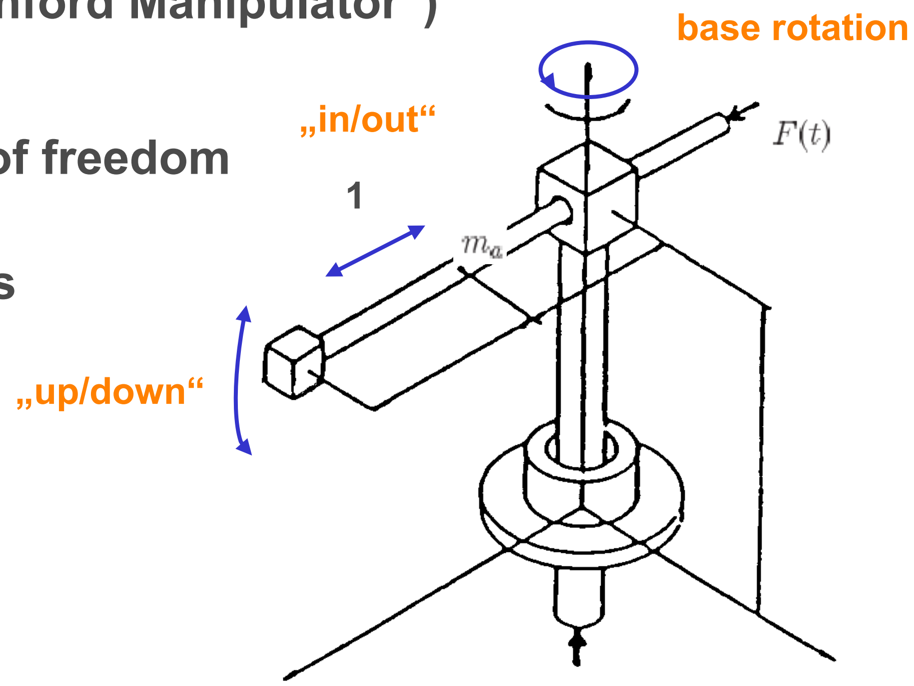


A simple example robot

Polar coordinate robot ("Stanford Manipulator")

3 degrees of freedom

3 motors

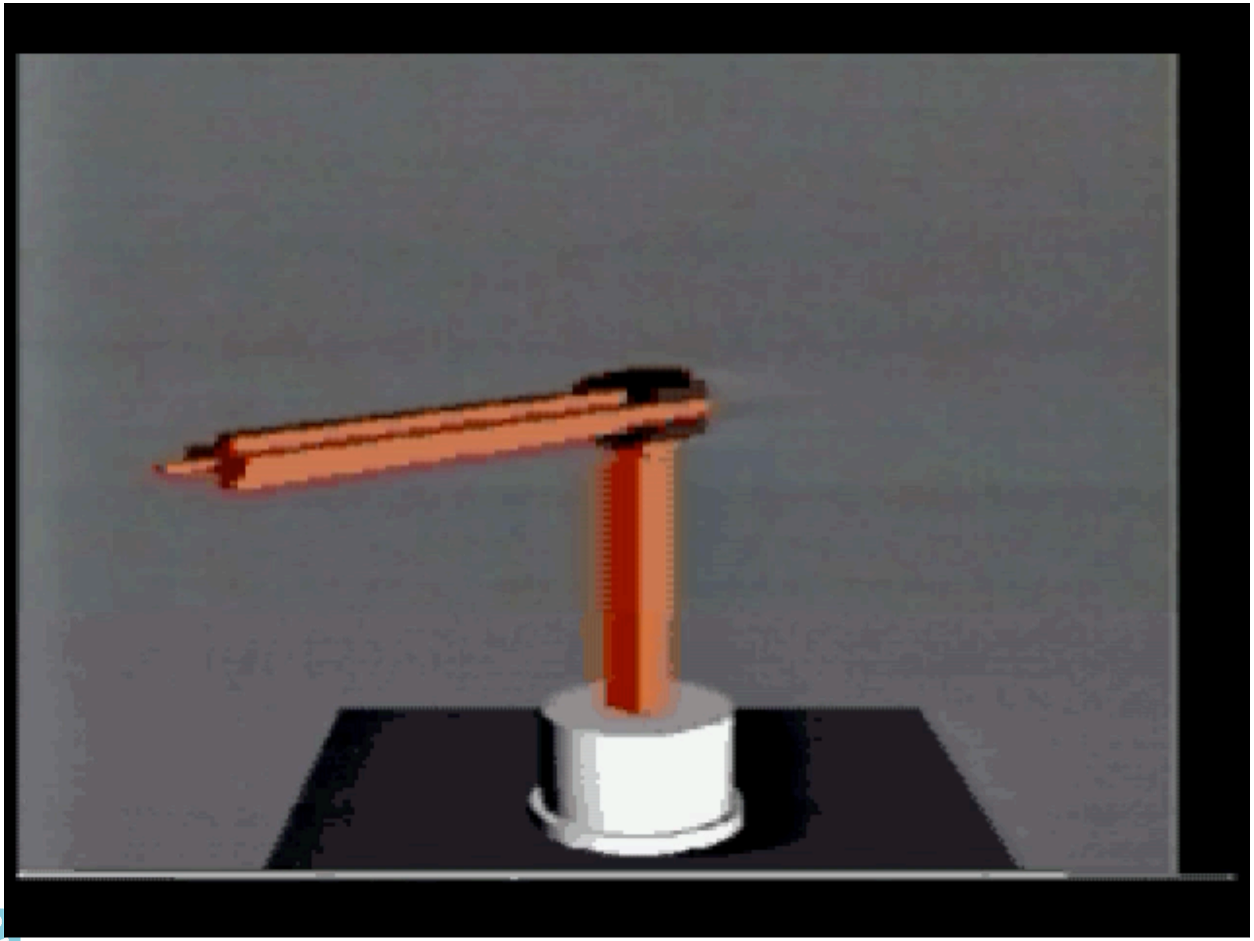


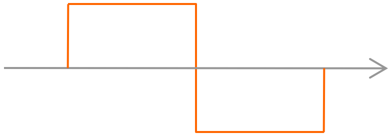
Example maneuver :

Retract arm in minimal time

Time optimal robot motions: base motion

- Classical solution: only „in/out“-motor (which reaches its limits)

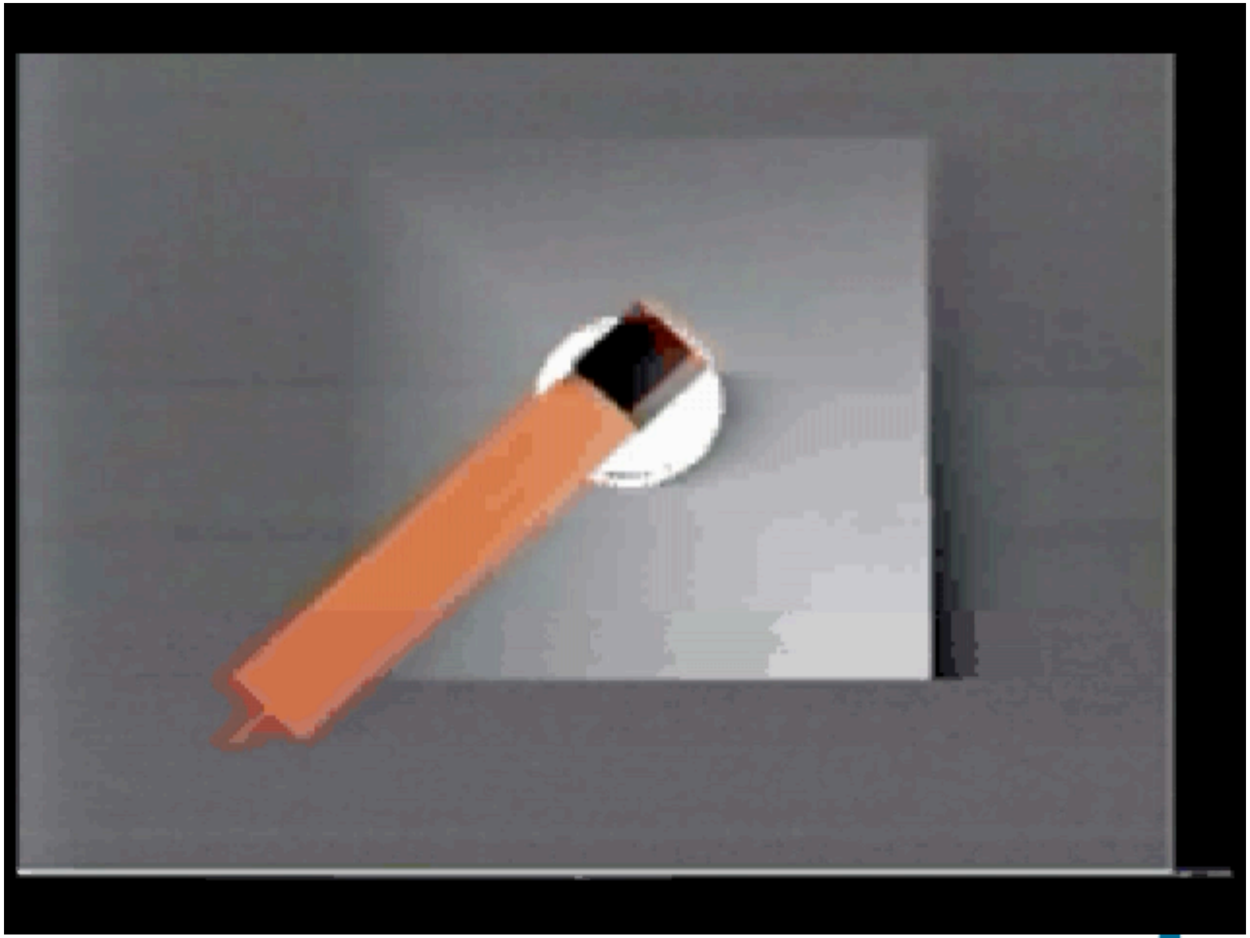




„Bang-bang“

Time optimal robot motions

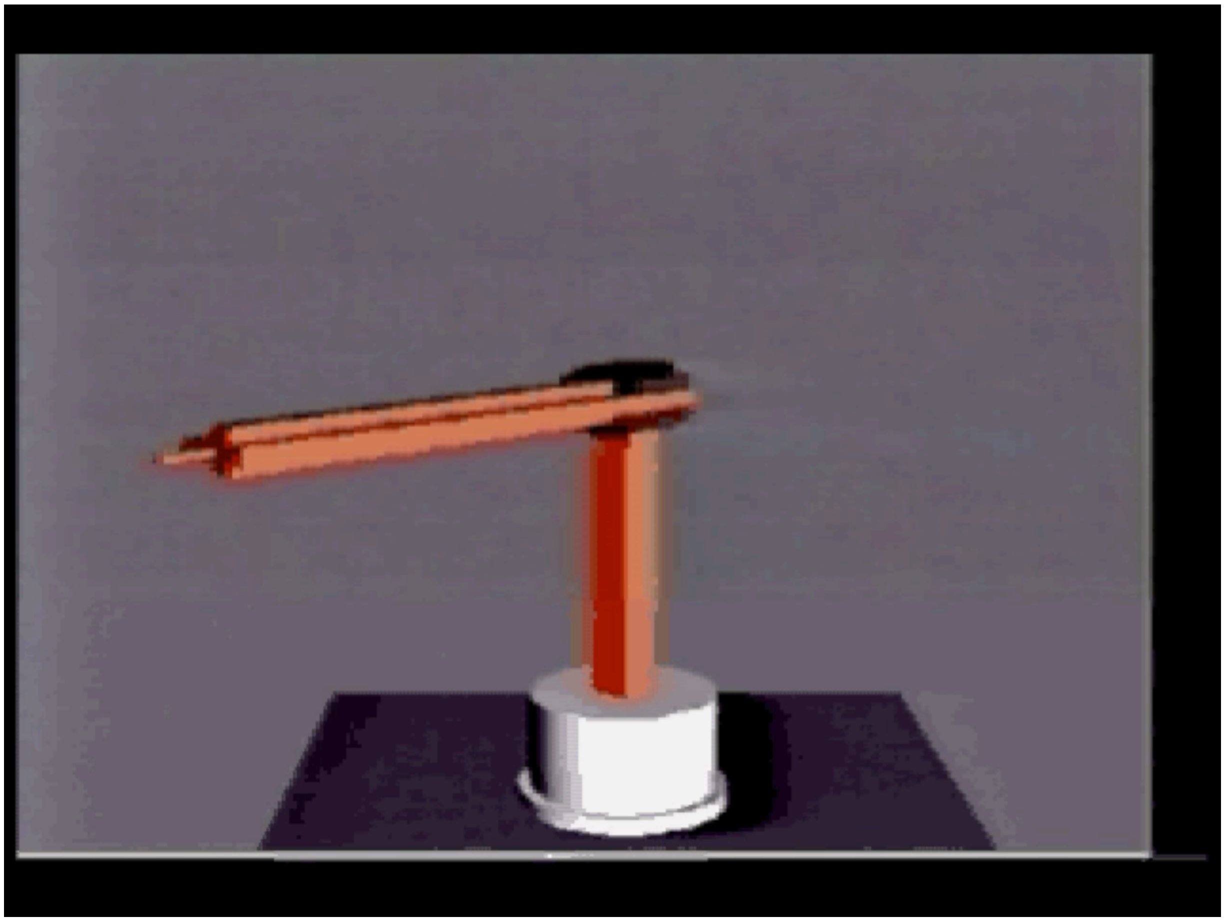
Support of „in/out“-motor by base rotation motor



22 % faster

Time optimal robot motions

Support of „in/out“-motor by „up/down“motor



51 % faster

Optimal control problem (simplifed)

Lagrange type | Mayer type

$$\min \int_0^T L(t, x(t), u(t), p)\,dt + E(x(T))$$

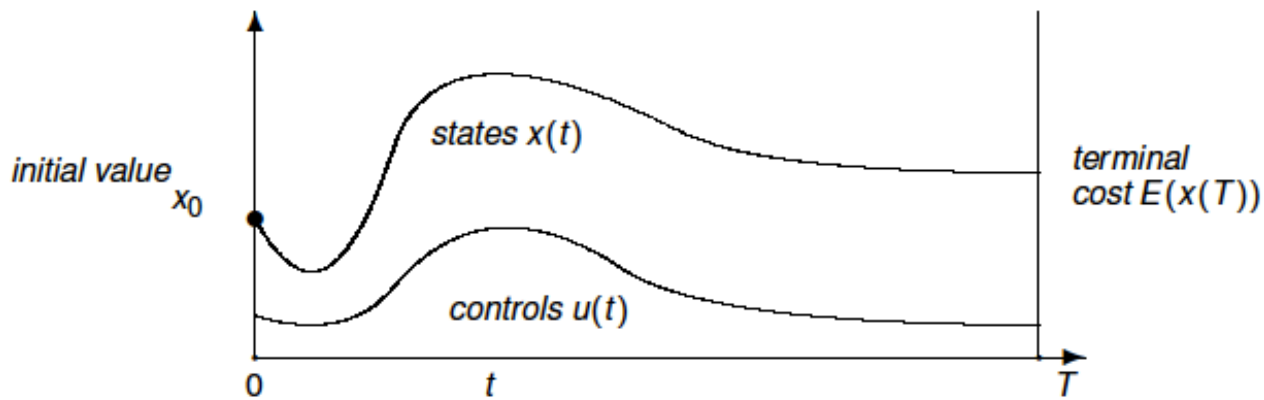
Objective function

$$s.t. \quad \dot{x} = f(t, x(t), u(t), p)$$

Process dynamics

$$\bar{x}(t_0) = \bar{x}_0 \quad \bar{x}(T) = \bar{x}_{end}$$

Initial and final constraints



Optimal control problem (more complex)

e.g. for the optimization of dynamic robot gaits:
Multiphase problems with many additional constraints

$$\min_{x,u,T} \int_0^T \phi(x(t), u(t), p)\, dt + \Phi(T, x(T), p)\,,$$

$$\begin{aligned}
\text{s. t.} \qquad \dot{x}(t) &= f_j(t, x(t), u(t), p) \quad \text{for} \quad t \in [\tau_{j-1}, \tau_j]\,, \\
& \qquad j = 1, \ldots, n_{\text{ph}}\,, \quad \tau_0 = 0\,, \quad \tau_{n_{\text{ph}}} = T \\
x(\tau_j^+) &= \tilde{J}(x(\tau_j^-), p) \quad \text{for} \quad j = 1, \ldots, n_{\text{ph}}\,, \\
g_j(t, x(t), u(t), p) &\geq 0 \quad \text{for} \quad t \in [\tau_{j-1}, \tau_j]\,, \\
r_{\text{eq}}(x(0), \ldots, x(T), p) &= 0\,, \\
r_{\text{ineq}}(x(0), \ldots, x(T), p) &\geq 0\,.
\end{aligned}$$

But we will stick with the simple problems for now

Formulation and solution of optimal control problems – Ingredients required

Process dynamics

Solution of optimization problems

Optimal control problems

$$
\begin{aligned}
\min_{x,u,T} \int_0^T & \phi(x(t),u(t),p)\, dt + \Phi(T,x(T),p)\,, \\
\text{s. t.} \qquad & \dot{x}(t) = f_j(t,x(t),u(t),p) \quad \text{for} \quad t \in [\tau_{j-1},\tau_j]\,, \\
& j = 1,\ldots,n_{\text{ph}}\,, \quad \tau_0 = 0\,, \quad \tau_{n_{\text{ph}}} = T \\
& x(\tau_j^+) = J(x(\tau_j^-),p) \quad \text{for} \quad j = 1,\ldots,n_{\text{ph}}\,, \\
& g_j(t,x(t),u(t),p) \geq 0 \quad \text{for} \quad t \in [\tau_{j-1},\tau_j]\,, \\
& r_{\text{eq}}(x(0),\ldots,x(T),p) = 0\,, \\
& r_{\text{ineq}}(x(0),\ldots,x(T),p) \geq 0\,.
\end{aligned}
$$

In the case of mechanical systems: Multibody system modeling

Solution of boundary value problems

Solution of optimal control problems with the direct multiple shooting methods

- Combines many topics we have discussed in previous lectures:
  - Formulation of dynamic process dynamics
  - Boundary value problem solution by multiple shooting
  - Nonlinear optimization problem solution by SQP methods
  - Sensitivity generation of trajectories by IND

3 different approaches for optimal control problems

Reminder – Key problem:

How to handle infinite dimensionality of states x(t) and controls u(t)?

- Dynamic Programming / Hamilton-Jacobi-Bellman equation
- Indirect Methods/ calculus of variations/ Pontryagin Maximum Principle
- Direct Methods (discretization of controls)

Only approach treated in this lecture

Direct optimal control methods

Optimal control method

Transformation from optimal control to nonlinear optimization problem

↓

Nonlinear Optimization Methods

Direct methods for optimal control problems

The control functions are approximated by a finite dimensional representation on a grid:

$$t_a = t_0 < t_1 < \ldots < t_{m-1} < t_m = t_b$$

$$u(t) = \varphi_j(t, q_j), q_j \in \mathbb{R}^{k_j}, t \in I_j = [t_j, t_{j+1}] \text{ for } j = 0, 1, \ldots, m-1$$

Base functions $\varphi_j(t, q_j)$:

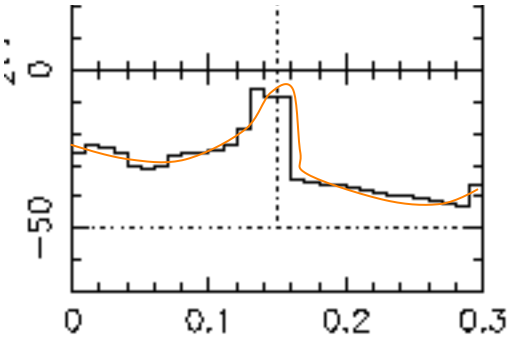
- piecewise constant: $\varphi_j(t, q_j) = q_j$
- piecewise linear:

$$\varphi_j(t, q_j) = q_j^1 + \frac{t - t_j}{t_{j+1} - t_j}(q_j^2 - q_j^1), \;\; q_j = \binom{q_j^1}{q_j^2}$$

- splines ...

Direct methods for optimal control problems

- Common idea: replace control functions u by a discretization (= a finite-dimensional parameter vector)
- Are also called first-discretize-then-optimize methods



➢ Infinite dimensionality of controls resolved

➢ But what about states x(t)?

Three different methods for state discretization

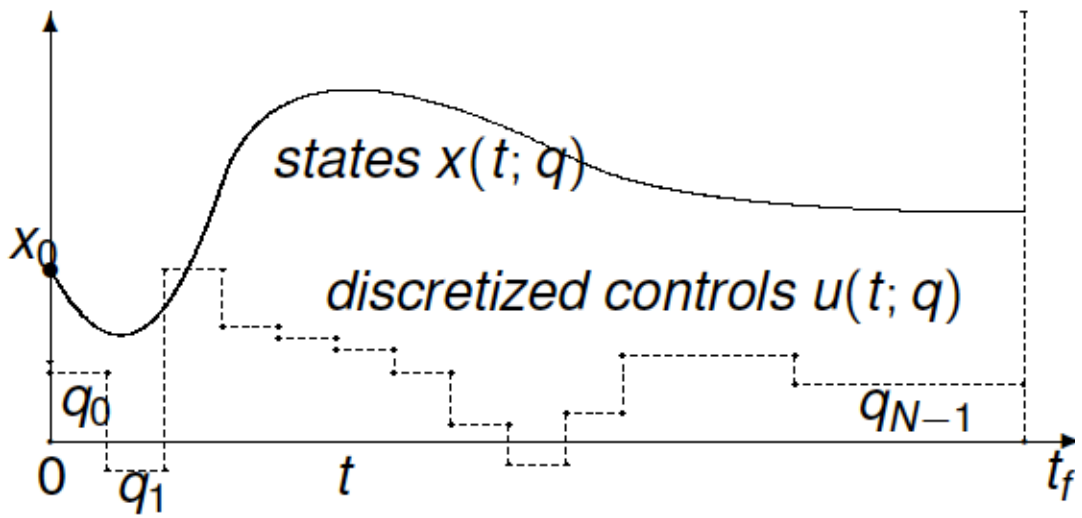
- Direct Collocation
- Direct Single Shooting
- Direct Multiple Shooting

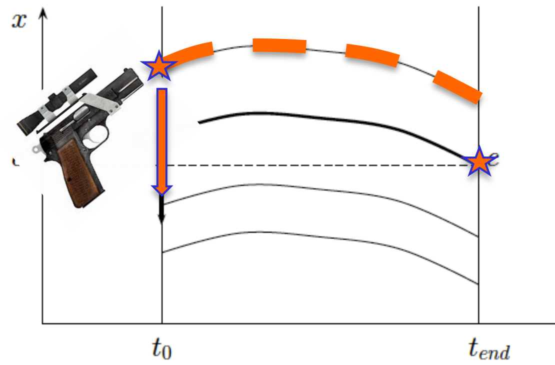
See lecture on boundary value problems

Direct Single Shooting

[Hicks, Ray 1971; Sargent, Sullivan 1977]

Discretize controls $u(t)$ on fixed grid $0 = t_0 < t_1 < \ldots < t_N = t_f$.





Regard states $x(t)$ on $[t_0, t_f]$ as dependent variables.
Use numerical integration to obtain state as function $x(t; q, x_0)$ of finitely many control parameters $q = (q_0, q_1, \ldots, q_{N-1})$ and the initial value x_0.

NLP in Direct Single Shooting

$$\min_{x_0, q_i, t_f} \quad \tilde{L}(x_0, q_i, t_f) + E(x(t_f; x_0))$$

subject to:

$$\begin{aligned} x(t_f; x_0, q_i) &= x_f \\ q_{min} \leq \quad & q_i \quad \leq q_{max} \\ x_{min} \leq \quad & x_0 \quad \leq x_{max} \end{aligned}$$

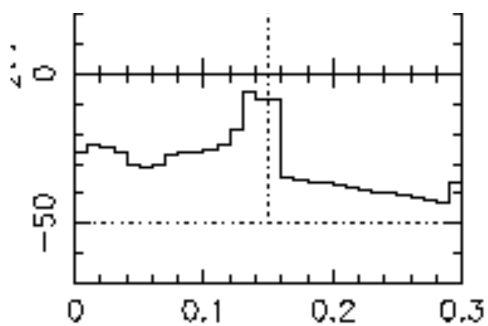
Direct Multiple Shooting

Bock, Plitt, 1981

- Discretize controls piecewise on a coarse grid
- Use functions which have only local support, e.g. piecewise constant

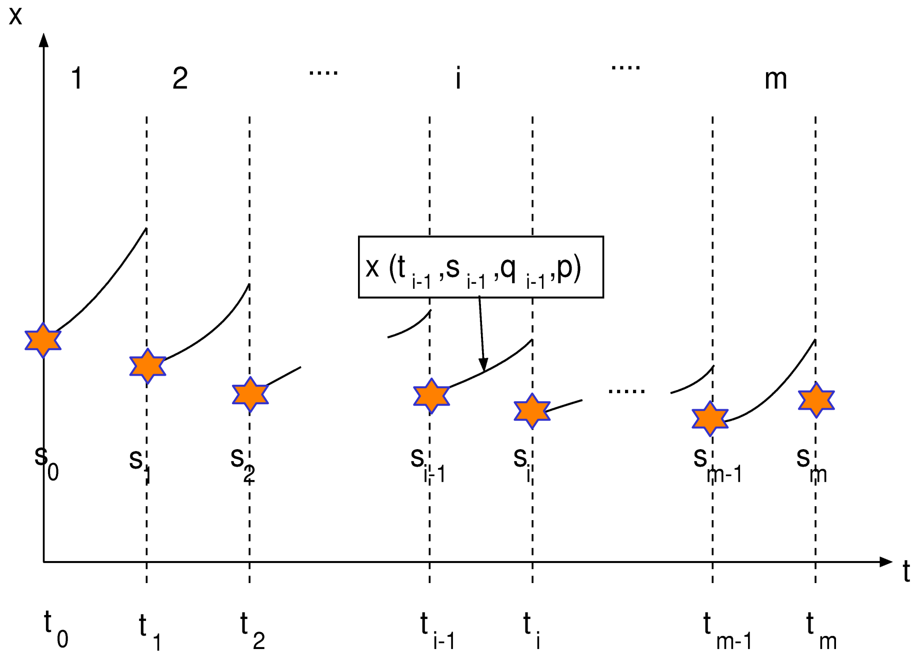
$$u(t) = q_i \quad \text{for} \quad t \in [t_i, t_{i+1}]$$

or piecewise linear functions



Direct Multiple Shooting

Bock, Plitt, 1981

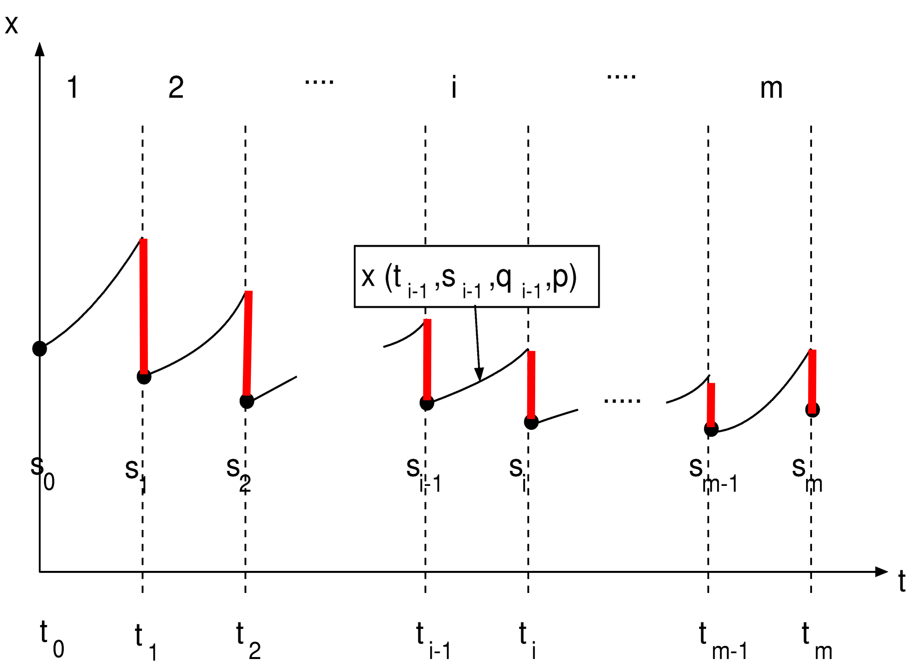


- Split long integration interval into many shorter ones
- Use initial values of all integration intervals as free variables and „shoot“ a new integration from each initial value

$$
\begin{aligned}
\dot{x}_i(t; s_i, q_i) &= f(x_i(t; s_i, q_i), q_i), \quad t \in [t_i, t_{i+1}], \\
x_i(t_i; s_i, q_i) &= s_i .
\end{aligned}
$$

Direct Multiple Shooting

Bock, Plitt, 1981



- Introduce constraints to close gaps („continuity conditions“)
- Also use integrators to compute objective function

$$\int_0^T L(t, x(t), u(t), p)dt$$

MUSCOD discretization summary

- **Direct method** = control discretization by piecewise constant, linear or cubic functions

- State parameterization by **multiple shooting**

- **Meshes for control discretization and multiple shooting:**
  - are chosen to be identical in practice (MUSCOD)
  - MS points could be chosen as a subset of control nodes (or vice versa)

- **Objective function is separable**

$$\Phi = \sum_{i=0}^{m-1} \Phi_i(s_i, w_i)$$

Discretized Optimal Control Problem

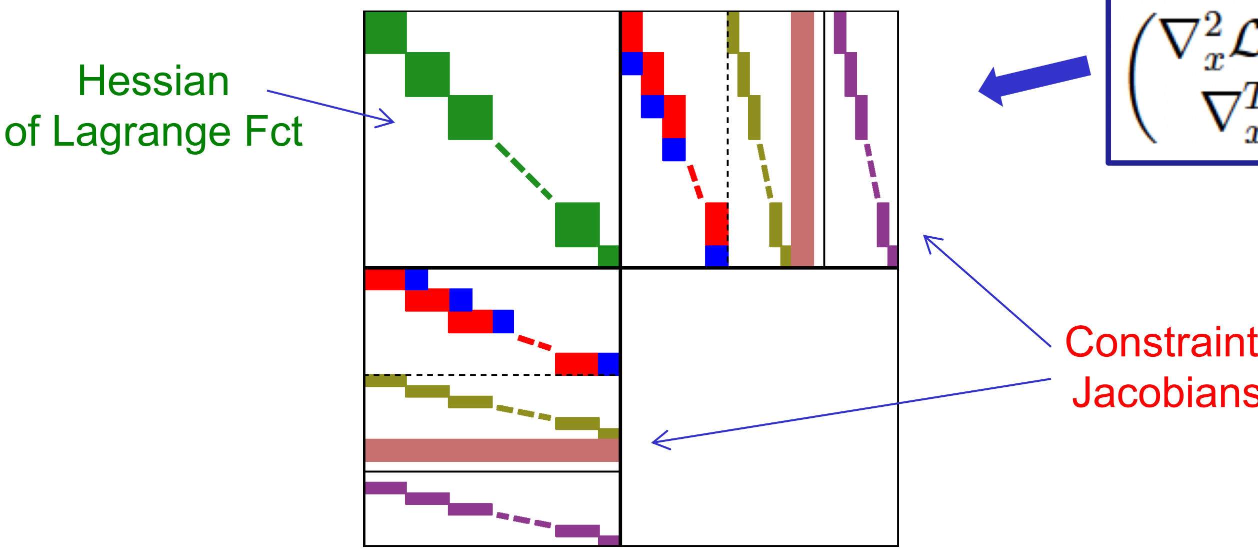
$$
\begin{aligned}
& \min_{y} \quad \tilde{\Phi}(y) \\
\text{s.t.}\quad & h_i = x(t_{i+1}, s_i, q_i) - s_{i+1} \quad = \quad 0 \quad \text{for } i = 0, ..., m-1 \\
& \tilde{r}_{eq,i}(s_0, ..., s_m) \quad = \quad 0 \quad \text{for } i = 0, ..., m \\
& \tilde{r}_{ineq,i}(s_0, ..., s_m) \quad \geq \quad 0 \quad \text{for } i = 0, ..., m \\
& y^T = (s_0, q_0, s_1, q_1, ..., s_m, T_p)^T
\end{aligned}
$$

UNIVERSITÄT HEIDELBERG
ZUKUNFT
SEIT 1386

Direct Multiple Shooting – Resulting NLP

Bock, Plitt, 1981

- Result of discretization:
  Large-scale nonlinear programming problem (NLP)
- Special structure originating from discretization (in KKT matrix)



$$\begin{pmatrix} \nabla_x^2 \mathcal{L}(x_k, \lambda_k) & -\nabla_x g(x_k) \\ \nabla_x^T g(x_k) & 0 \end{pmatrix}$$

- Solution with structure-exploiting, tailored sequential quadratic programming (SQP) method, special condensing techniques

Direct multiple shooting

Bock, Plitt, 1981

- **Simultanuous optimization and simulation**

Discretized optimal control problem

(NLP)

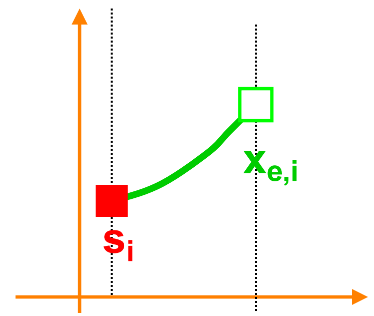
Variables:
discretized states s_i,
discretized controls u_i,
parameters p,
Phase durations h_i

Solution by efficient SQP

s_i, u_i, p, h →

← **$x_{e,i}$ + sens.**

Initial value problem solution (Integration + derivatives) on each multiple shooting interval



Solution by efficient integrators

Demonstration of optimal control problem solution with MUSCOD-II

A simple example

Katja Mombaur

A simple optimal control problem

- A car is supposed to travel 300 m in 32 seconds
- Its initial and final velocity are zero
- The acceleration can vary between -2 m/s^2 and 1m/s^2
- The maximum velocity is 30 m/s

- Optimization goal: minimize "energy" in terms of accelerations squared

A simple optimal control problem

- Mathematical formulation:

$$\min_{u,q} \int_0^{t_f} u^2(t)dt$$

Objective function (Lagrange type) **lfcn**

subject to the constraints:

$$\dot{q} = v$$
$$\dot{v} = u$$

Dynamic process model **ffcn**

$$q(0) = 0, q(t_f) = 300$$
$$v(0) = 0, v(t_f) = 0$$

Initial and final constraints **rdfcn**

$$-2 \leq u(t) \leq 1$$
$$0 \leq q(t) \leq 300$$
$$0 \leq v(t) \leq 30$$

Bounds (data file)

MUSCOD files overview

Sourcefile:
Projectname/SRC/~

Model
- Dimension
- Objective function
- Differential equation
- „Complex“ constraints
- (Order of phases)

Datafile:
Projectname/DAT/~

- Time discretization
- Initial values (x,u,p)
- Upper and lower bounds
- Scaling
- Algorithmic settings

CMakeLists:
Projectname/~


Thank you very much for your attention!