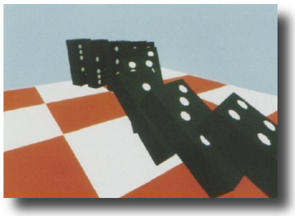
# Control Methods
# ( for movement skills)

CPSC 526

Sept-Dec 2017

Past → Future



methods

lessons

"Can you fly that thing?"
“Not yet …”

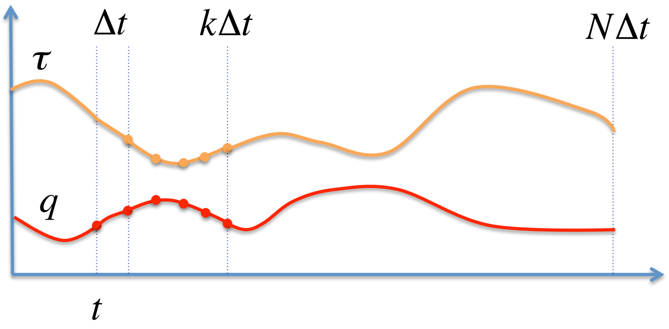
# Overview

- Trajectory Optimization
  - spacetime constraints (offline)
  - model-predictive control (online)
- QP-based inverse dynamics
  - QP to solve for torques
  - many humanoid robots
- Direct policy search
  - derivative-free optimization to find the best controller
- Value-based RL
  - value function and policy learned over all states
- Guided Learning
  - supervised learning of policy
  - iterate data-collection & learning as needed

# Trajectory Optimization

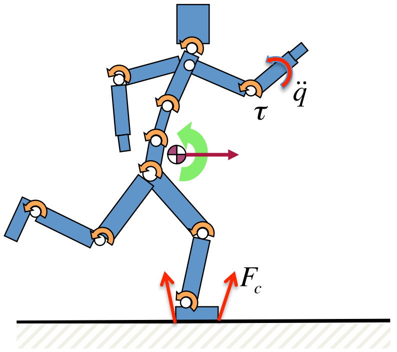
- Spacetime Constraints (1988)
  - simple system; regular discrete time samples (later: splines)
- Physically Based Motion Transformation (1999)
  - human motions (with reduced dynamics models)
- Adaptation of Performed Ballistic Motion (2005)
  - full dynamics models, scaling of unknowns, stay close to a reference motion
- Online Trajectory Optimization (2012)
  - finite horizon lookahead, differentiable dynamics through contacts
- Contact Invariant Optimization (2012)
  - can discover best contact phases to use
- Online Motion Synthesis Using Sequential Monte Carlo (2015)
  - finite horizon method, model-free

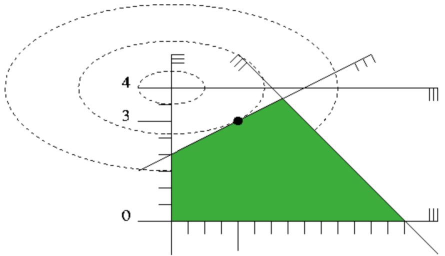
# Trajectory Optimization – Online Version (MPC: Model Predictive Control)



optimize *k* step "finite horizon",
then execute *1* step

# QP-based inverse dynamics





[https://doc.cgal.org/latest/QP_solver/index.html ]

Online optimization problem
(typically: quadratic program)

unknowns: $\tau, \ddot{q}, F_c$

objectives – quadratic in:
- joint accel
- COM accel
- angular accel
- minimize sq torques

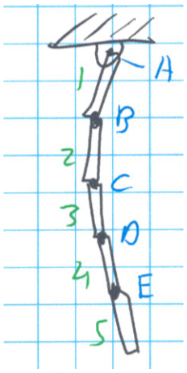
task

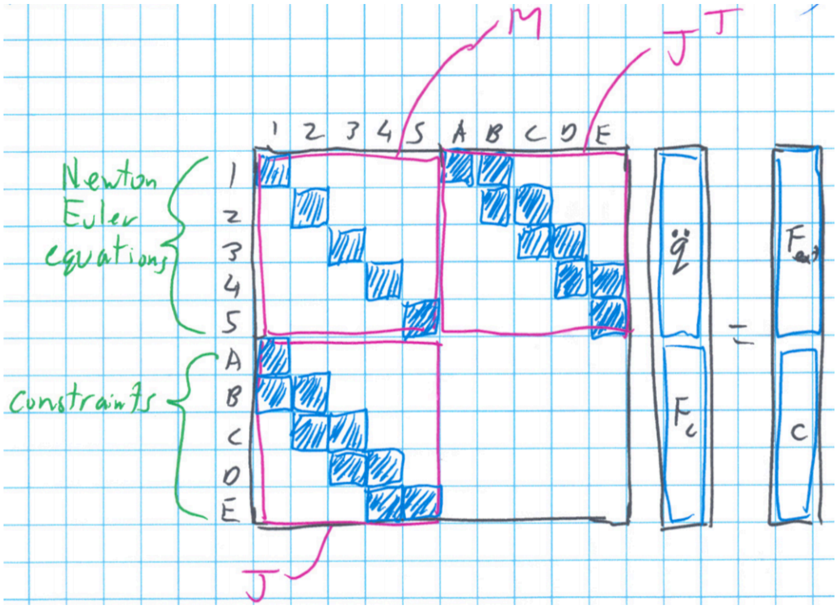
constraints

equality constraints (eqns of motion):

$$Ax - b = 0$$

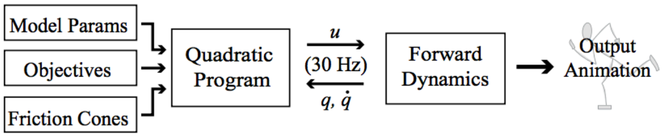
inequality constraints:
- contact force limits
- torque limits





Multiobjective Control with Frictional Contacts, SCA 2007   (video)

***System Overview***



$$\min_{\boldsymbol{a},\boldsymbol{f},\boldsymbol{u}} \quad \{g^{(1)},\ldots,g^{(\ell)}\}$$

$$\text{subject to} \quad \boldsymbol{Ma}+\boldsymbol{n}+\boldsymbol{G}^\top\boldsymbol{f}=\begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix}\boldsymbol{u}$$

$$\boldsymbol{f}\in \mathrm{K}, \quad \boldsymbol{u}\in \mathrm{L}$$

$$\boldsymbol{Ga}+\dot{\boldsymbol{G}}\dot{\boldsymbol{q}}=\mathbf{0}$$

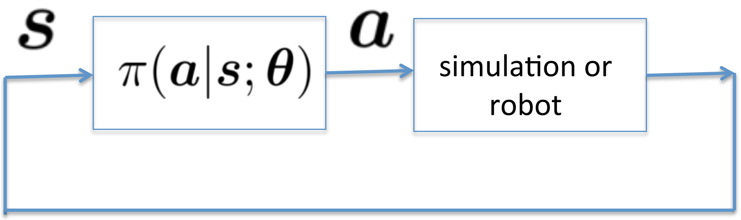
Feature-Based Locomotion Control, SIGGRAPH 2010

# DARPA Robotics Challenge (DRC, 2015)

- Multi-level optimization:
  - Footstep Optimization (Discrete + Continuous)
  - Trajectory Optimization (Continuous)
  - Optimization-Based Inverse Dynamics: Greedy continuous optimization (Quadratic Program = QP) for full body at the current instant.

[ DARPA Robotics Challenge, www.cs.cmu.edu/~cga/dw ]

# Direct Policy Search



$\boldsymbol{s}$

$\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta})$

$\boldsymbol{a}$

simulation or robot

$$R^{[i]} = \sum_{t=1}^{T} r_t$$

$$\mathcal{D} = \left\{\boldsymbol{\theta}^{[i]}, R^{[i]}\right\}$$

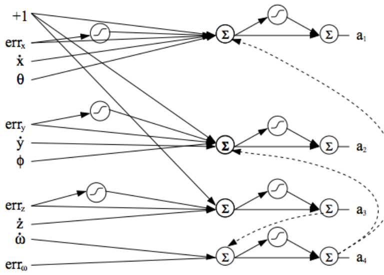
Repeat

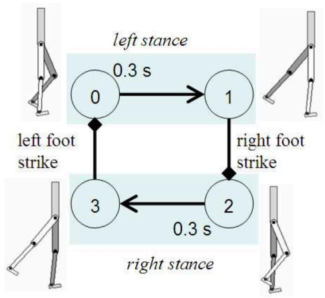
1. Explore: Generate trajectories $\boldsymbol{\tau}^{[i]}$ following the current policy $\pi_k$
2. Evaluate: Assess quality of trajectory or actions
3. Update: Compute new policy $\pi_{k+1}$ from trajectories and evaluations

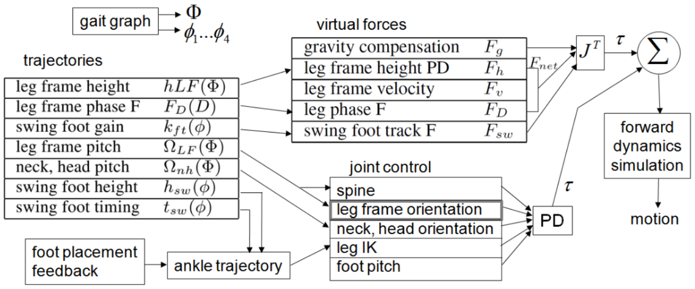
Until convergence

[Schaal 2015: http://icml.cc/2015/tutorials/PolicySearch.pdf ]

# Control Policy Parameterizations



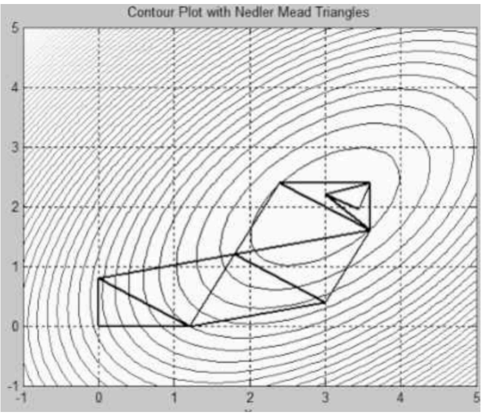




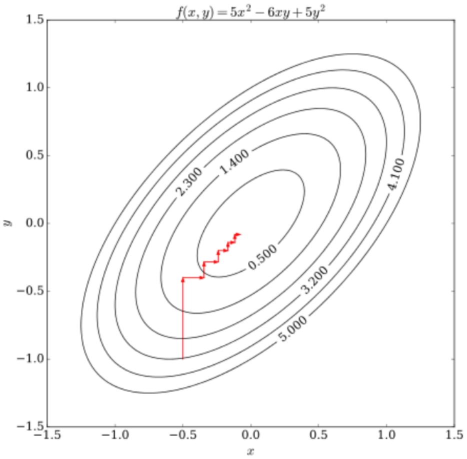
$$u - u_{ref} = F\,(x - x_{ref})$$

# Derivative Free Optimization (“Black box optimization”)

Nelder-Mead / Downhill Simplex Method

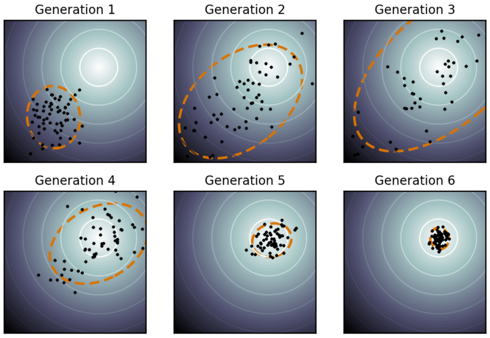


Coordinate Descent

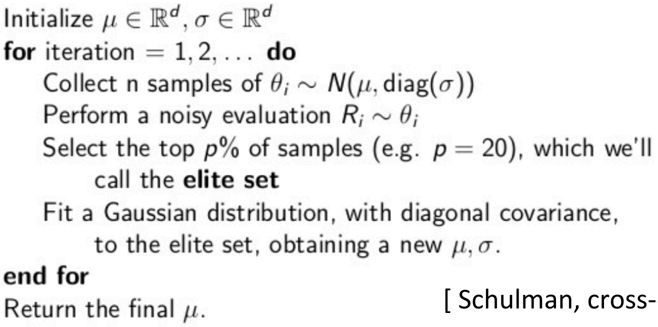


Others: Simultaneous Perturbation Stochastic Approximation (SPSA),
Parameter Exploring Policy Gradient
Simulated Annealing

- Covariance Matrix Adaptation (CMA) / Cross Entropy method (CEM)



[ wikipedia ]



```
Initialize $\mu \in \mathbb{R}^d, \sigma \in \mathbb{R}^d$
**for** iteration $= 1, 2, \ldots$ **do**
    Collect n samples of $\theta_i \sim N(\mu, \text{diag}(\sigma))$
    Perform a noisy evaluation $R_i \sim \theta_i$
    Select the top $p\%$ of samples (e.g. $p = 20$), which we'll
        call the **elite set**
    Fit a Gaussian distribution, with diagonal covariance,
        to the elite set, obtaining a new $\mu, \sigma$.
**end for**
Return the final $\mu$.
```

[ Schulman, cross-entropy method ]

# Overview

- Trajectory Optimization
  - spacetime constraints (offline)
  - model-predictive control (online)
- QP-based inverse dynamics
  - QP to solve for torques
  - many humanoid robots
- Direct policy search
  - derivative-free optimization to find the best controller
- Value-based RL
  - value function and policy learned over all states
- Guided Learning
  - supervised learning of policy
  - iterate data-collection & learning as needed