Intelligent control through learning and optimization

Emo Todorov

Applied Mathematics and Computer Science & Engineering

University of Washington

Winter 2014

Administrative information

- No fixed office hours or TA
- Feel free to contact me with any questions or to set up a meeting:
  - email: todorov@cs.washington.edu
  - office: CSE 422
- Grading will be based on 3 homeworks and one paper presentation
- Materials will be posted on the class website:
  http://www.cs.washington.edu/homes/todorov/courses/amath579

Background and readings

- Expected background:
  - Linear algebra
  - Vector calculus
  - Basic probability theory
  - Matlab programming
- Helpful background:
  - Numerical optimization
  - Stochastic processes
  - Dynamical systems, ODEs, PDEs

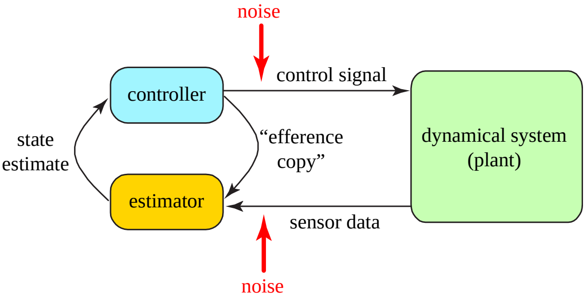
Background and readings

- Expected background:
  - Linear algebra
  - Vector calculus
  - Basic probability theory
  - Matlab programming
- Helpful background:
  - Numerical optimization
  - Stochastic processes
  - Dynamical systems, ODEs, PDEs
- There is no suitable textbook, but here are some useful books:
  - Sutton and Barto (1998) *Reinforcement Learning: An Introduction.* **online**
  - Bertsekas and Tsitsiklis (1996) *Neuro-dynamic programming.*
  - Bertsekas (2000) *Dynamic programming and optimal control.*
  - Stengel (1994) *Optimal control and estimation.*
- General-purpose readings (available on the class website)
  - Lecture notes, Pieter Abbeel (Berkeley)
  - Lecture notes/book in preparation, Russ Tedrake (MIT)
  - Lecture slides, Dimitri Bertsekas (MIT)
  - Lecture notes, Ben Van Roy (Stanford)
  - Book chapter on Optimal Control, Emo Todorov (UW)

Why learning and optimization?

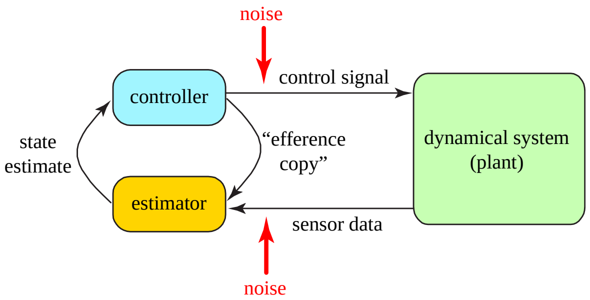
- It is good to be optimal. The real question is how to get there.
- This is how nature works – and it produces much more intelligent controllers than anything we have ever built.
- We are not smart enough to manually design really complex control systems, and analytical solutions are generally lacking (optimal or not). Indeed the vast majority of control systems used in industry are PID.
- If we are going to use a computer to design a control system, casting the problem in terms of numerical optimization makes sense.
- Learning and optimization have already produced very impressive control systems.
- The advent of massively parallel processors makes it possible to re-optimize control systems in real time – which may turn out to be the only way to solve really hard control problems.

The big picture



We will focus on problems where the controller has direct access to the state.

The big picture



We will focus on problems where the controller has direct access to the state.

Example: Linear quadratic Gaussian (LQG) system

dynamical system	$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \boldsymbol{\varepsilon}_t$
sensor data	$\mathbf{y}_t = H\mathbf{x}_t + \boldsymbol{\omega}_t$
optimal estimator	$\widehat{\mathbf{x}}_{t+1} = A\widehat{\mathbf{x}}_t + B\mathbf{u}_t + K\left(\mathbf{y}_t - H\widehat{\mathbf{x}}_t\right)$
optimal controller	$\mathbf{u}_t = L\widehat{\mathbf{x}}_t$

Direct and indirect methods

Direct methods

1. Choose a parametric form of the **control law**
2. Implement a function which can evaluate the performance of any control law (usually by extensive simulation)
3. Optimize this function with respect to the control law parameters (using generic optimization tools)

Robust but often slow.

Indirect methods

1. Choose a parametric form of the **value function**
2. Solve the equation which this function is supposed to satisfy (Bellman equation, Pontryagin maximum principle)
3. Derive the corresponding control law by greedy optimization of the value function (often analytically)

Fast but not always robust.

Local and global methods

Local methods

1. Represent the solution along a **trajectory**
2. Find a direction of improvement in trajectory space, do line-search

No feedback (but see MPC)

Global methods

1. Represent the solution as a **globally-defined function**
2. Improve the solution (globally) by modifying the function

Curse of dimensionality

Specific methods

	Local	**Global**
Direct	space-time optimization: $L\left(x_0, x_1, \cdots x_N\right)$	policy gradient: $E\left[\sum_t \ell\left(x_t, u_t = \pi\left(x_t; w\right)\right)\right]$
Indirect	DDP: $v_t\left(x\right) \approx c_t + x^T S_t x$	dynamic programming: $v_t\left(x\right) = \min_u \left\{\ell_t\left(x, u\right) + v_{t+1}\left(x'\right)\right\}$

Optimal control in the context of optimization

- generic optimization problem: $\min_w L(w)$
- learning problem: $\min_w L(w)$ where

$$L(w) = \frac{1}{N} \sum_{n=1}^{N} \ell(y_n - f(x_n; w))$$

- optimal control problem: $\min_w L(w)$ where

$$L(w) = \frac{1}{N} \sum_{t=1}^{N} \ell(x_t; w), \quad \text{and} \quad x_{t+1} = f(x_t; w)$$

Part I: Stochastic optimal control theory

- Stochastic optimal control in discrete space and time
  - Markov Decision Processes (MDPs)
  - Bellman equations for different problem formulations
  - Policy iteration and value iteration, contraction mappings
  - Linear programming view of MDPs
- Stochastic optimal control in continuous space and time
  - Controlled diffusions
  - Numerical discretization
  - Hamilton-Jacobi-Bellman equations
  - Exact solutions for Linear-quadratic-Gaussian (LQG) problems
- Stochastic optimal control problems with linear Bellman equations
  - Discrete and continuous problem classes with linear (Hamilton-Jacobi) Bellman equations
  - Compositionality of optimal control laws
  - Embedding of generic optimal control problems
  - Path integrals

Part II: Numerical methods for optimal control

- Robot dynamics and control
  - Multi-joint kinematics and dynamics
  - Dynamics in the presence of frictional contacts
  - Computed torque control, PID control, sliding mode control
  - Hierarchical and operational-space control
- Approximate dynamic programming and Reinforcement Learning
  - Dynamic programming with function approximation
  - Function approximation in the linear Bellman equation framework
  - Monte Carlo methods, temporal difference methods (TD-lambda)
  - Off-policy and on-policy methods (Q-learning and Sarsa)
  - Policy gradient methods
- Local trajectory-based methods for optimal control
  - Maximum principle for deterministic and stochastic systems
  - ODE methods, pseudo-spectral methods, space-time optimization
  - Differential dynamic programming and iterative LQG
  - Model-predictive and receding-horizon control, rollout policies

Part III: Other topics

- Data-driven methods
  - Using motion capture to design controllers
  - Imitation learning
  - Unsupervised learning from motion data
- Inverse optimal control
  - Kalman's work in the LQG case
  - Inverse Reinforcement Learning
  - Inverse optimal control in the linear Bellman equation framework
  - Design of control-Lyapunov functions
- Duality between optimal control and Bayesian estimation
  - Kalman filter, information filter and LQG control
  - General duality in the linear Bellman equation framework
  - Using approximate inference methods to solve optimal control problems