Brief Overview of Adaptive and Learning Control

Tuomas J. Lukka

1.10.2007

Outline

Outline

Outline

Definition of Adaptive Control

Definition of Adaptive Control

- Zames (reported by Dumont&Huzmezan): “A non-adaptive controller is based solely on a-priori information whereas an adaptive controller is based also on a posteriori information”

But isn't feedback control *a posteriori*?

- Unified view: system state and parameters are all just state, anyway
- Stochastic control solves everything

But isn't feedback control *a posteriori*?

- Unified view: system state and parameters are all just state, anyway
- Stochastic control solves everything
- Not possible in practice
  $\rightarrow$ need approximations and optimizations
- The terminology and conceptual organization of the field is based on a long history
  - Analog components, ...
  - Analytic proofs, ...

Approximations and optimizations

- For example,
  - Fixed-structure controller with parameters, simple laws to alter those parameters
  - The system is periodic, and works the same every time

Definition of Adaptive Control 2

- Zames (reported by Dumont&Huzmezan): "A non-adaptive controller is based solely on a-priori information whereas an adaptive controller is based also on a posteriori information"
- Sastry&Bodson: "direct aggregation of a (non-adaptive) control methodology with some form of recursive system identification"

Definition of Learning Control

- On an abstract level, arbitrary: combination of timescales and conceptual difference

Definition of Learning Control

- On an abstract level, arbitrary: combination of timescales and conceptual difference
- Adaptive controller depends on very recent history
  - No memory
  - Reacts to current state only
- Learning controller depends on long-term history
  - Memory
  - Remembers previous states and appropriate responses

Definition of Learning Control

- On an abstract level, arbitrary: combination of timescales and conceptual difference
- Adaptive controller depends on very recent history
  - No memory
  - Reacts to current state only
- Learning controller depends on long-term history
  - Memory
  - Remembers previous states and appropriate responses
- Again, from the grand unified stochastic control perspective, these are the same

Direct — Indirect

- Direct = adapt or identify controller parameters directly
- Indirect = adapt model of system, calculate controller parameters from model

Direct — Indirect

- Direct = adapt or identify controller parameters directly
- Indirect = adapt model of system, calculate controller parameters from model
- Even here, the only difference is conceptual

Outline

Motivating example

- Two-armed bandit
- System has two actions
- Each action gives a reward from an unknown but constant distribution
- How to maximize the wins?

Motivating example

- Two-armed bandit
- System has two actions
- Each action gives a reward from an unknown but constant distribution
- How to maximize the wins?
- Must take into account information from the system when making the decision, but also the uncertainty of the information, and optimize both.

Mathematical Methods

- Laplace transform
  - Used all through control theory
- Lyapunov functions
  - Can show convergence of certain controllers, *provided assumptions hold*

History

- 50s: Initial algorithms
- 60s: Dynamic Programming and Dual Control — intractable
- 70s and 80s: Convergence proofs
- 80s-90s: Reinforcement learning and neural methods

Outline

Algorithms

- Classical Adaptive
  - Gain Scheduling
  - MRAC (Model Reference Adaptive Control)
  - Self-tuning regulator
  - SOAS (Self-Oscillating Adaptive Systems)
- Periodic Adaptive / Learning
  - ILC (Iterative Learning Control)
  - RC (Repetitive Control)
- Machine Learning
  - Reinforcement Learning

Outline

Gain Scheduling

- Determine controller parameters directly from measurements “unrelated” to the process
- Example: use measured air pressure and velocity to determine feedback gain in an aeroplane pitch controller — otherwise, trouble
- Usually linear interpolation between controllers designed for particular parameter values
- Works well in certain problems

Gain Scheduling

- Determine controller parameters directly from measurements "unrelated" to the process
- Example: use measured air pressure and velocity to determine feedback gain in an aeroplane pitch controller — otherwise, trouble
- Usually linear interpolation between controllers designed for particular parameter values
- Works well in certain problems
- Difficulties:
  - Need to find good variables to measure
  - May need lots of controllers to interpolate between

MRAC (Model Reference Adaptive Control)

- Drive difference between plant and reference model to zero by adapting controller parameters directly
- Ad hoc rules: e.g., high-gain servo
- MIT rule: gradient descent + assume unknown parameters are the estimated values for calculating gradient
  - Can be unstable
- Many variants

Self Tuning Regulators

- Use parameterized control design equations for a plant
- Identify parameters on-line
- Apply controller for those parameters — "Certainty Equivalence"
- Harder to analyze
  - Design equations usually nonlinear

SOAS - Self-Oscillating Adaptive Systems

- Use relay to discretize control signal
- Use dithered error to control relay
- Adapt gain of relay based on limit cycle amplitude
- Instance of MRAS, with constant excitation designed into the system

Outline

Periodic systems

- Premise: fixed operation cycle
  - Robot arm doing repetitive operation
  - Adjusting voltage to particle accelerator magnet
- Premise: Error has periodic part

ILC and RC

- ILC = Iterative Learning Control, RC = Repetitive Control
- Eliminate periodic error
- Record error as a function of time, use on the subsequent cycles to improve control
  - Heuristically: "At 2.54 seconds, the robot arm usually goes too far left, so use force to the right at that point"
- Works well with non-linear and difficult-to-model systems
- Stability sometimes difficult to obtain, need various filters
- Difference between schemes:
  - ILC assumes known initial state for each period
  - RC lets end of previous period affect start of next (transients)

Outline

Reinforcement Learning

- System = states, actions
- Step = act, get reward
- Goal: maximize reward over time
  - Decide what to do (policy)
  - Update policy over time to reflect reward obtained (learning rule), directly or indirectly
- Main variability
  - Policy (e.g., α-greedy)
  - Learning method (e.g., Q-learning: $Q : \text{states} \times \text{actions} \rightarrow R$)
  - Function approximation and generalization
    - Convergence guaranteed only if when there is no extrapolation (details in books)

Outline

Conclusion

- Control theory: roots in era of real, analog components
- On an abstract enough level, it is all just approximations to stochastic control
- Big changes from neural computation: nonlinear function approximation
- Methods are difficult to compare since the range of systems to be controlled is huge
- In the industry, the simplest system that works is good

Issues with Adaptive Control

- Unmodeled dynamics cause bad behaviour
- If a controller regulates well, knowledge of plant's behaviour decays
  - Need constant or intermittent excitation to know system behaviour
  - Stochastic Control actually does generate excitations

Methods not discussed here

- Neurofuzzy control
  - Adapt "rules-of-thumb" with data
- Neural augmentation of classical control methods
  - Adapting feedforward control
  - Treating nonlinearities in some area of the control problem using backpropagation neural networks
- Countless others — the literature is huge

Hidden Bonus: Model-Predictive Control (MPC)

- Popular practical control method
- Original motivation: constraints
- Uses explicit model
- Explicitly optimizes control several time steps forwards (sliding horizon)
- Computationally intensive
  - Enabled by digital computers