Machine Learning: A change of paradigm in Flow Control?

Laurent CORDIER
Laurent.Cordier@univ-poitiers.fr

PPRIME Institute
Poitiers, France
1. Phenomenological approaches
   - Pros:
     - Physically based
     - Work experimentally
   - Cons:
     - Restricted to one type of flow physics

2. Model-based control
   (a) Based on identification: ARMAX, ERA, OKID
       - Pros:
         - Pure data driven
         - Work experimentally
       - Cons:
         - Restricted to one type of flow physics (linearized behaviour)
   (b) Based on first principle equations and (optionally) data
       - Pros:
         - Rigorous approach
       - Cons:
         - Purely numerical
         - Too fragile to work in most of the real configurations
1 – Model-based control
Riccati-based stabilization
Riccati-based stabilization

\[ V_{c} \]

\[ J_{\text{perf}} \]

\[ x \]

\[ y \]
Riccati-based stabilization

![Graph showing Riccati-based stabilization](image)

- **x** vs. **y**
- **Vc** vs. **t**
- **Jperf** vs. **t**

GDR CDD, Nantes, November 19, 2015 – p.5/29
Riccati-based stabilization

$V_c$ vs $t$

$J_{\text{perf}}$ vs $t$
2 – Machine Learning
Arthur Samuel (1959)

Field of study that gives computers the ability to learn without being explicitly programmed.

Tom Mitchell (1998) – Well-posed Learning Problem:

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

Applications:

Database mining, Games, Autonomous helicopter, handwriting recognition, Natural Language Processing,...
1. **Supervised Learning**
   Learn a mapping from inputs $x$ to outputs $y$ given a labeled set $\mathcal{D}_{SL} = \{x_i, y_i\}_{i=1}^N$.
   - Classification or pattern recognition
   - Regression

2. **Unsupervised Learning**
   Given only inputs $\mathcal{D}_{UL} = \{x_i\}_{i=1}^N$, discover “interesting patterns”
   - Clustering
   - Dimensionality Reduction: PCA

3. **Reinforcement Learning**
   How to take actions in an environment so as to maximize a cumulative reward.
3 – Genetic Programming Control
Genetic programming basics

**Step 1:**
1st generation with random nonlinear control laws

\[ b^1_m = K^1_m(s), \ m = 1, \ldots, 100 \]

**Steps 2...n:**
Biologically inspired optimization of the control laws based on the 'fitness grades'

\[ J [b = K(s)] \]

Optimization process

\[ b^1_1 = K^1_1 \rightarrow J^1_1 \]
\[ \cdot \]
\[ \cdot \]
\[ \cdot \]
\[ b^1_i = K^1_i \rightarrow J^1_i \]
\[ \cdot \]
\[ \cdot \]
\[ \cdot \]
\[ b^1_n = K^1_n \rightarrow J^1_n \]

Crossover

Recopy

Mutation

\[ b^2_1 = K^2_1 \rightarrow J^2_1 \]
\[ \cdot \]
\[ \cdot \]
\[ \cdot \]
\[ b^2_i = K^2_i \rightarrow J^2_i \]
\[ \cdot \]
\[ \cdot \]
\[ \cdot \]
\[ b^2_n = K^2_n \rightarrow J^2_n \]

Genetic programming: operations

**REPLICATION**

![Replication Diagram](image1)

**CROSS-OVER**

![Crossover Diagram](image2)

**MUTATION**

![Mutation Diagram](image3)
Genetic programming (GP) for closed-loop flow control

V. Parezanovic\textsuperscript{1},
L. Cordier\textsuperscript{1}, B. R. Noack\textsuperscript{1}, J.-P. Bonnet\textsuperscript{1},
T. Duriez\textsuperscript{2}, M. Segond\textsuperscript{3}, M. Abel\textsuperscript{3}, S. L. Brunton\textsuperscript{4}

\textsuperscript{1} Institut PPRIME, CNRS, Poitiers, FRANCE
\textsuperscript{2} CONICET, Buenos Aires, ARGENTINA
\textsuperscript{3} Ambrosys GmbH, Potsdam, GERMANY
\textsuperscript{4} University of Washington, Seattle, USA

Project support:
ANR Chair of Excellence "Closed-loop control of turbulent shear flows using reduced-order models (TUCOROM)"
Experimental setup

**TUCOROM demonstrator for control of the mixing layer**

- **Wind tunnel**: long test section, independently driven streams, velocity range [0:12m/s]
Machine Learning Control design

Mixing layer plant

Control law $b = K(s)$

LEARNING PHASE (Genetic Programming)

$J_1 < J_2 < \ldots < J_n$
Machine learning control design

Mixing layer plant

Control law \( b = K(s) \)

Heaviside function

INDEPENDENT REAL-TIME CONTROLLER

Learning module is disconnected...

\[ J_1 < J_2 < \cdots < J_n \]
MLC results (I) – Frequency selection

Max W (x=200mm)
(mixing layer thickness)

Max K (x=200mm)
(mixing layer fluctuation energy)
4 – Cluster Reduced-Order Model
CROM
Cluster-based reduced-order modeling
Liouville equation based method

Navier-Stokes equation
\[ \frac{\partial u}{\partial t} + (u \cdot \nabla) u = -\frac{1}{\rho} \nabla p + \nu \Delta u \]

POD expansion
\[ u(x, t) = \sum_{i=0}^{N} a_i(t) u_i(x) \]

Galerkin projection
\[ (u_k, \mathcal{R}[u])_\Omega = 0 \]

\[ \frac{da}{dt} = f(a) \]

POD Galerkin model

Liouville equation
\[ \frac{\partial p(a, t)}{\partial t} + \nabla_a (f(a)p(a, t)) = 0 \]

Galerkin expansion
\[ p(a, t) = \sum_{k=1}^{K} \rho_k(t) \Psi_k(a) \]

Ulam-Galerkin method
\[ (\Psi_j, \mathcal{R}[p])_\Omega = 0 \]

\[ \frac{dp}{dt} = \mathbf{P} \rho \]

Markov model for CROM

State space
\[ u(x, t) \]

State space
\[ p(a, t) \]
CROM approach

How can one identify physical mechanisms in an unsupervised manner based on given data?

Data → Discretised state space (cluster analysis) → Construct cluster Transition matrix → Markov model

Discrete snapshots of a limit cycle

Time-resolved velocity snapshots
\( u^m = u(x, t_m) \)

Cluster analysis (k-means) (Steinhaus 1956, MacQueen 1967)

\[ V = \sum_{k=1}^{K} \sum_{u^m \in c_k} ||c_k - u^m||^2 \rightarrow \text{Minimum} \]

Discrete-time Markov model (Time-homogeneous, memoryless)

\[ p^l = p^l p^0 \quad \text{with} \quad P_{jk} = \frac{n_{jk}}{n_k} \]

Refined analysis: Cluster sociology Entropy
Clustering of mixing layer data

K-means applied to POD coefficient vectors for K=10, M=2000 snapshots

- Compression of all snapshots into a few modes
- Uncertainty due to cluster size
CROM of a mixing layer

Cluster transition matrix

Simplified cluster transitions

Markov chain: Weighted directed graph
5 – Reinforcement Learning
Agent interacts with environment to gain knowledge

- Explores and receives rewards
- Actions change the state of the environment
- Choose actions to maximize long-term reward
**Markov Decision Process**

- $S$: State space (finite); $s_k \in S$
- $A$: Action space (finite); $a_k \in A$
- Transition probability $p(s_{k+1}|s_k, a_k)$
- $r$: Reward function
- $\gamma \in [0, 1[$: Discount factor
- $\Pi$: Policy
  - Deterministic: $a = \Pi(s)$
  - Stochastic: $p_{\Pi}(a|s) = \Pi(a|s)$

**Objective:**

Find a policy $\pi^*$ that maximizes the expected long-term reward

$$V^\Pi(s) = \mathbb{E} \left[ \sum_{k=0}^{+\infty} \gamma^k r_{k+1} | s_0 = s, \Pi \right]$$

$$r_{k+1} = r_{k+1}(s_k, a_k, s_{k+1})$$
Hash functions (2)
Left: \( a(t) \). Right: \( C_d(t) \) under three different control policies: 0-command (black), best known command (“ora”), and present approach.
Questions ???