A hybrid approach to control: classical control theory meets data-driven methods

Mouhacine Benosman

Dynamical Systems Team
MERL - Mitsubishi Electric Research Labs, Cambridge, USA

Benelux Meeting on Systems and Control 2022
A hybrid approach to control: classical control theory meets data-driven methods


- MERL (ex-)interns (https://www.merl.com/internship/openings)

Big thanks to the organizers of this meeting for their warm welcome: Alain Vande Wouwer, Michel Kinnaert, Emanuele Garone, Laurent Dewasme, Guilherme Pimentel, Erjen Lefeber, William Van Hoeck…
Adaptation vs. Learning?

Before reviewing some results in the field of adaptation and learning, let us first define the two terms: Learn and adapt. Refereing to the Oxford dictionary we find these two definitions; Adapt is defined as: *to change something in order to make it suitable for a new use or situation, or to change your behavior in order to deal more successfully with a new situation.* As for learn, it is defined as: *to gain knowledge and skill by studying, from experience, or to gradually change your attitudes about something so that you behave in a different way.* [Benosman 2016]

Adaptation: change
Learning: gradual change by repetition
Main points of the talk

Part 1: Theory*

- Brief survey of adaptive control: model-based adaptation, data-driven (classical RL & control theory inspired RL, extremum seeking control), and learning-based adaptation (hybrid: model-based + data-driven)
- Learning-based adaptive control for nonlinear systems with constant/time-varying parametric uncertainties (ESC, GP-UCB, ADP, CBF)
- Learning-based feedback gains auto-tuning for nonlinear systems affine in the control (ESC)
- Indirect learning-based adaptive control for linear systems under constraints (MPC framework) (ESC)
- Learning-based adaptive PDEs stable model reduction and estimation (ESC, RL)

Part 2: Examples

- Mechatronics applications: Electromagnetic brakes, servo motors
- Fluid dynamics applications: Airflow modeling and estimation
- Robotics applications

A hybrid approach to control: classical control theory meets data-driven methods

Mouhacine Benosman
MERL - Mitsubishi Electric Research Labs, Cambridge, USA

Part I: Brief survey and some theoretical results
Adaptation in Control

Model of the system, e.g., law of physics or Input/Output models

Controller and filter are based on the model of the system

- Nonlinear model (direct vs. indirect adaptation), e.g., Krstic et al. 95, Slotine et al. 91, Spooner 2002, Astolfi et al. 2008, Fradkov et al. 99, Astolfi 2015, Guay et al. 2015, Taylor et al. 2020
- Infinite dimension and delays, e.g., Wen et al. 89, Smyshlyaev et al. 2010
- Constrained model (MPC type), e.g., Mosca 95, Guay et al. 2015
- Stochastic model, e.g., Sragovich 2006
- Multi-agent model, e.g., Lewis et al. 2014
Data-Driven Adaptation

- Reinforcement Learning (RL): **Stochastic Markov Decision Process (MDP)**
- RL: Control policies are designed from interaction with a simulator and/or with the real environment.

- Classical (CS) RL:
  - Model-based data generation (simulator-based/enhanced learning), e.g., Werbos 92, Bertsekas 96, Powell 2007, Busoniu 2010, Levine et al. 20, As et al., 2022
  - Model-free (real environment-based learning), e.g., Sutton et al. 98, Levine et al. 20
  - Multi-agent models, e.g., Oliehoek et al. 2016

- Control theory-‘inspired’ RL:
Data-Driven Adaptation

- Extremum seeking control (ESC)
- Data-driven optimization with estimation of the (higher order) derivatives of the cost function, i.e., ‘zero-order’ optimization
  - Deterministic, e.g., Leblanc 1922, Krstic et al. 2000, Ariyur et al. 03, Zhang et al. 12, Scheinker et al. 16, Feiling et al. 21, Dürr et al. 13, Nešic et al. 13, Tan et al. 2013, Guay et al. 15, Guay et al. 20, Benosman et al. 21a, Poveda et al. 21
  - Stochastic, e.g., Liu et al. 12, Manzie et al. 09, Radenkovic et al. 16
  - Infinite dimension, e.g., Oliveira et al. 20, Oliveira et al. 21, Feiling et al. 18
  - Hybrid, e.g., Poveda et al. 17, Poveda 2018
  - Multi-agent, e.g., Poveda 2018, Poveda 21a, Poveda 21b
Sur l'électrification des chemins de fer au moyen de courants alternatifs de fréquence élevée

Une légère modification à un dispositif permettant la transformation d’un courant continu en courant alternatif de fréquence élevée, décrit dans la « R. G. E. » du 19 août 1922, t. XII, p. 250-261, a permis à M. Maurice Leblanc d’envoisager l’alimentation d’une ligne de transmission d’énergie par la réaction par l’électricité. Le récepteur d’énergie électrique n’avait aucun point de départ.

9 mm d’épaisseur. Chacun s’étend au-dessous d’un des conducteurs de la ligne de transmission avec une distance, d’axe en axe, de 30 cm. Ces tubes sont portés par des isolateurs et reliés à ceux des voitures suivantes par des conducteurs.
Data-Driven Adaptation

- Iterative Learning Control (ILC), e.g., Owens 2015
- Genetic algorithms, e.g., Dracopoulos 2013
Learning-based Adaptation

- Learning-based (hybrid: model-based control + data-driven adaptation)

- Computer science
  - Data-driven learning

- Control theory
  - Model-based control

- Learning-based adaptive control
  - Merging model-based control and data-driven learning algorithms
Learning-based Adaptation

- Learning-based (hybrid: model-based control + data-driven adaptation)

ID-based (indirect adaptation):

- ESC *-based, e.g., Benosman 2016
- NN-based, e.g., Lewis et al. 99, Spooner et al. 02, Wang et al. 2010
- Control barrier functions (CBFs)-based, e.g., Lopez et al. 2020, Emam et al. 2021

‘Not’ ID–based (direct adaptation):

- Feedback controller tuning, e.g., Gain tuning, e.g., Hjalmarsson 02, Benosman 2016, Duivenvoorden et al. 2017, Benosman et al. 21b, MPC hyper-parameters tuning, e.g., Hewing et al. 2020

* ESC: Extremum seeking control, GP: Gaussian process, ADP: Adaptive dynamic programming, MPC: Model predictive control.
Survey References

Classical model-based adaptive control

Survey References

Classical model-based adaptive control

Survey References

Data-driven adaptive control: Classical reinforcement learning

- Oliehoek F. A. et al. 2016, A Concise Introduction to Decentralized POMDPs, doi.org/10.1007/978-3-319-28929-8, Springer.

Data-driven adaptive control: Control theory- inspired reinforcement learning

- Perkins T.J., Barto A.G., 2000, Lyapunov-constrained action sets for reinforcement learning. ICML.
Survey References

Data-driven adaptive control: Extremum seeking

- Dürr HB., Stanković MS., Ebenbauer C., Johansson KH., 2013, Lie bracket approximation of extremum seeking systems, Automatica 49 (6), 1538-1552
Survey References

Data-driven adaptive control: Extremum seeking

- Poveda J.I., 2018, Robust Hybrid Systems for Control, Learning, and Optimization in Networked Dynamical Systems, PhD thesis, UCSB.
Survey References

Data-driven adaptive control: Others

Survey References

Learning-based adaptive control:
Survey References

- Marvi et al., 2020, Safe Off-policy Reinforcement Learning Using Barrier Functions, IEEE American Control Conference.
Learning-based Adaptive Control for Nonlinear Systems*


© MERL
Main points

- Learning-based adaptive control for nonlinear systems with constant/time-varying parametric uncertainties (ESC, GP-UCB, ADP, CBF)*

- Learning-based iterative feedback gains tuning for nonlinear systems affine in the control (ESC)

- Indirect learning-based adaptive iterative control for linear systems under constraints (ESC-MPC framework)

- Learning-based adaptive PDE stable model reduction/estimation (ESC, RL)

* ESC: Extremum seeking control, GP-UCB: Gaussian process upper confidence bound, ADP: Adaptive dynamic programming, CBF: Control barrier function, RL: Reinforcement learning
Main points

- Learning-based adaptive control for nonlinear systems with constant/time-varying parametric uncertainties (ESC, GP-UCB, ADP, CBF)*

- Learning-based iterative feedback gains tuning for nonlinear systems affine in the control (ESC)

- Indirect learning-based adaptive iterative control for linear systems under constraints (ESC-MPC framework)

- Learning-based adaptive PDE stable model reduction/estimation (ESC, RL)

* ESC: Extremum seeking control, GP-UCB: Gaussian process upper confidence bound, ADP: Adaptive dynamic programming, CBF: Control barrier function, RL: Reinforcement learning
Learning-based adaptive control: a modular approach

Nonlinear model

\[ \dot{x}(t) = f(t,x,u,\Delta) \Rightarrow u = g(t,x,\tilde{\Delta}) \]

Nonlinear model-based controller

Model-Free

Learning-based parameters estimation

\[ \tilde{\Delta} = F(P) \]

Data-driven part

Performance evaluation

Convergence (Performance)

Model-based part

Real system

 boundedness (safety)

Parameters estimates / Gains tuning

Measurements
Learning-based indirect adaptive control for constant uncertainties

\[ \dot{x} = f(x, \Delta, u) \]

\( \Delta \in \mathbb{R}^p \) parametric uncertainties

the output vector \( y = h(x) \)

where \( h : \mathbb{R}^n \to \mathbb{R}^h \), with smoothness of \( f \), and \( h \).

The control objective here is for \( y \) to asymptotically track a desired smooth vector time-dependent trajectory \( y_{ref} : [0, \infty) \to \mathbb{R}^h \).
Learning-based indirect adaptive control for constant uncertainties

Modularity through (ISS) robustness

\[ \dot{x} = f(t, x, u) \text{ is LiISS}^* \text{ if and only if there exist functions } \beta \in K\mathcal{L} \text{ and } \gamma_1, \gamma_2 \in \mathcal{K} \text{ such that } \]

\[ \|x(t, \xi, u)\| \leq \beta(\|\xi\|, t) + \gamma_1 \left( \int_{0}^{t} \gamma_2(\|u(s)\|)ds \right) \]

Learning-based indirect adaptive control for constant uncertainties

**Assumption 1:**

\[ e_y(t) = y(t) - y_{ref}(t). \]

\[ \exists u_{iss}(t, x, \hat{\Delta}): \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^m \]

\[ \dot{e}_y = f_{e_y}(t, e_y, e_\Delta) \]

is iISS from the input vector \( e_\Delta = \Delta - \hat{\Delta} \) to the state vector \( e_y \).
Concept of (dither-based) Extremum Seeking Control (ESC)*

**Advantages:**
- Model-free (zero-order) optimization
- Gradient implicit estimate using one measurement per learning iteration (good for real-time applications)
- Robustness to noise
- Robustness to initial conditions
- Input and state constraints

**Analysis:**
- Averaging theory
- Singular perturbation theory (for dynamic maps)

Learning-based indirect adaptive control for constant uncertainties

Basic intuition of ESC

Negligible changes in $y$

$\delta u \approx 0 \leftarrow \int \delta y \bigotimes d$ $\delta u \geq 0 \leftarrow \int \delta y \bigotimes d$ $\delta u < 0 \rightarrow \int$
Learning-based indirect adaptive control for constant uncertainties

ESC uncertainties estimator

cost function $Q(\hat{\Delta}) = F(e_y(\hat{\Delta}))$

where $F : \mathbb{R}^h \rightarrow \mathbb{R}$,

$F(0) = 0, \ F(e_y) > 0 \text{ for } e_y \neq 0$

Assumed to be well defined, i.e., for the same $\hat{\Delta}$, we obtain the same $Q(\hat{\Delta})$

If not intrinsically, it can be forced by an iterative or batch-to-batch implementation
Learning-based indirect adaptive control for constant uncertainties

Assumption 2:
$Q$ has a local minimum at $\hat{\Delta}^* = \Delta$

Assumption 3:
$e_\Delta(t_0)$ is sufficiently small

Assumption 4:
$Q$ is analytic $\| \frac{\partial Q}{\partial \Delta}(\tilde{\Delta}) \| \leq \xi_2$, $\xi_2 > 0$, $\tilde{\Delta} \in \mathcal{V}(\Delta^*)$
Learning-based indirect adaptive control for constant uncertainties

Lemma:

the system \( \dot{x} = f(x, \Delta, u) \) with the cost \( Q \)
under Assumptions 1, 2, 3, and 4
the control \( u_{iss} \), where \( \hat{\Delta} \) is estimated
with the multi-parameter extremum seeking

\[
\dot{x}_i = a_i \sin(\omega_i t + \frac{\pi}{2}) Q(\hat{\Delta}) \\
\hat{\Delta}_i = x_i + a_i \sin(\omega_i t - \frac{\pi}{2}), \; i \in \{1, \ldots, p\}
\]

Learning-based indirect adaptive control for constant uncertainties

Lemma: Cont.

with $\omega_i \neq \omega_j$, $\omega_i + \omega_j \neq \omega_k$, $i, j, k \in \{1, \ldots, p\}$ ensures that

$$\|e_y(t)\| \leq \beta(\|e_y(0)\|, t) + \alpha(\int_0^t \gamma(\tilde{\beta}(\|e_\Delta(0)\|, t) + \|e_\Delta\|_{\text{max}}))ds,$$

where $\|e_\Delta\|_{\text{max}} = \frac{\xi_1}{\omega_0} + \sqrt{\sum_{i=1}^{p} a_i^2}$, $\xi_1, \xi_2 > 0$, $e(0) \in D_e$, $\omega_0 = \max_{i\in\{1,\ldots,p\}} \omega_i$, $\alpha \in \mathcal{K}$, $\beta \in \mathcal{KL}$, $\tilde{\beta} \in \mathcal{KL}$ and $\gamma \in \mathcal{K}$. 
Learning-based indirect adaptive control for time-varying systems

\[ \dot{x} = f(t, x, \Delta, u) \]

\( \Delta \in \mathbb{R}^p \) parametric uncertainties

the output vector \( y = h(x) \)

where \( h : \mathbb{R}^n \rightarrow \mathbb{R}^h \), with \( f \) being piecewise continuous in \( t \) and (at least) locally Lipschitz in \( x, u \), uniformly in \( t \), \( h \) is smooth. The control objective here is for \( y \) to asymptotically track a desired smooth vector time-dependent trajectory \( y_{ref} : [0, \infty) \rightarrow \mathbb{R}^h \).
Learning-based indirect adaptive control for time-varying systems

Assumption 1:

\[ e_y(t) = y(t) - y_{ref}(t). \]

\[ \exists u_{iss}(t, x, \hat{\Delta}): \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^m \]

\[ \dot{e}_y = f_{e_y}(t, e_y, e_\Delta) \]

is iISS from the input vector \( e_\Delta = \Delta - \hat{\Delta} \) to the state vector \( e_y \).
Learning-based indirect adaptive control for time-varying systems

ESC (time-varying) uncertainties estimator

cost function

\[ Q(\hat{\Delta}, t) = F(e_y(\hat{\Delta}), t) \]

where \( F : \mathbb{R}^h \times \mathbb{R}^+ \rightarrow \mathbb{R}^+ \), \( F(0, t) = 0 \),

\[ F(e_y, t) > 0, e_y \neq 0 \]
Learning-based indirect adaptive control for time-varying systems

Assumption 2:

$Q$ has a local minimum at $\hat{\Delta}^* = \Delta$

Assumption 3:

$$\left| \frac{\partial Q(\hat{\Delta}, t)}{\partial t} \right| < \rho_Q, \forall t \in \mathbb{R}^+, \forall \hat{\Delta} \in \mathbb{R}^p.$$
Learning-based indirect adaptive control for time-varying systems

**Lemma**: the system $\dot{x} = f(t, x, \Delta, u)$ with the cost $Q$ then under Assumptions 1, 2, and 3, the control $u_{iss}$, where $\hat{\Delta}$ is estimated with the multi-parameter extremum seeking

$$\hat{\Delta}_i = a \sqrt{\omega_i} \cos(\omega_i t) - k \sqrt{\omega_i} \sin(\omega_i t) Q(\hat{\Delta}, t)$$

$i \in \{1, \ldots, p\}$

* M. Benosman, 2014, Extremum Seeking-based Indirect Adaptive Control for Nonlinear Systems, IFAC World Congress.
Lemma: Cont.

with $a > 0$, $k > 0$, $\omega_i \neq \omega_j$, $i, j, k \in \{1, \ldots, p\}$, $\omega_i > \omega^*$, $\forall i \in \{1, \ldots, p\}$, with $\omega^*$ large enough, ensures

$$\|e_y(t)\| \leq \beta(\|e_y(0)\|, t) + \alpha(\int_0^t \gamma(\|e_\Delta(s)\|)ds),$$

where $\alpha \in \mathcal{K}$, $\beta \in \mathcal{KL}$, $\gamma \in \mathcal{K}$, and $\|e_\Delta\|$ satisfies:
Learning-based indirect adaptive control for time-varying systems

**Lemma:** Cont.

1-$(\frac{1}{\omega}, d)$-Uniform Stability: For every $c_2 \in ]d, \infty[$, there exists $c_1 \in ]0, \infty[$ and $\hat{\omega} > 0$ such that for all $t_0 \in \mathbb{R}$ and for all $x_0 \in \mathbb{R}^n$ with $\|e(0)\| < c_1$ and for all $\omega > \hat{\omega}$,

$$\|e(t, e(0))\| < c_2, \forall t \in [t_0, \infty[$$
Lemma: Cont.

2-$\left(\frac{1}{\hat{\omega}}, d\right)$-Uniform ultimate boundedness: For every $c_1 \in ]0, \infty[$ there exists $c_2 \in ]d, \infty[$ and $\hat{\omega} > 0$ such that for all $t_0 \in \mathbb{R}$ and for all $x_0 \in \mathbb{R}^n$ with $\|e_{\Delta}(0)\| < c_1$ and for all $\omega > \hat{\omega}$,

$$\|e_{\Delta}(t, e_{\Delta}(0))\| < c_2, \forall t \in [t_0, \infty[$$
Learning-based indirect adaptive control for time-varying systems

**Lemma:** Cont.

3-(\(\frac{1}{\omega}, d\))-Global uniform attractivity: For all \(c_1, c_2 \in (d, \infty)\) there exists \(T \in \mathbb{R}\) and \(\hat{\omega} > 0\) such that for all \(t_0 \in \mathbb{R}\) and for all \(x_0 \in \mathbb{R}^n\) with \(\|e_\Delta(0)\| < c_1\) and for all \(\omega > \hat{\omega}\),

\[
\|e_\Delta(t, e_\Delta(0))\| < c_2, \quad \forall t \in [t_0 + T, \infty[,
\]

where \(d\) is given by: \(d = \min\{r \in \mathbb{R}^n : \Gamma_H \subset B(\Delta, r)\}\),

with \(\Gamma_H = \{\hat{\Delta} \in \mathbb{R}^n : \left\| \frac{\partial Q(\hat{\Delta}, t)}{\partial \hat{\Delta}} \right\| < \sqrt{\frac{2\rho_0}{k\alpha\beta_0}}\}, 0 < \beta_0 \leq 1\),

and \(B(\Delta, r) = \{\hat{\Delta} \in \mathbb{R}^n : \left\| \hat{\Delta} - \Delta \right\| < r\}\).
Learning-based indirect adaptive iterative control for nonlinear systems affine in the control variable


We consider an output tracking problem for systems that are affine in the control

\[
\dot{x} = f(x) + \Delta f(t, x) + g(x)u, \ x(0) = x_0, \quad (3)
\]

\[
y = h(x), \text{ with ref. trajectory } y_d(t). \]

under classical smoothness and relative degree assumptions, we can design an ISS controller satisfying,

\[
\|e_y(t)\| \leq \beta(\|e_y(t_0)\|, t - t_0) + \gamma(\sup_{t_0 \leq \tau \leq t} \|e_\Delta(\tau)\|),
\]

where \(e_y, e_\Delta\) denote the output tracking error and the uncertainties estimation error, respectively.
Learning-based indirect adaptive iterative control for nonlinear systems affine in the control variable

ISS controller (Model-based)

\[ y^{(r)}(t) = b(\xi(t)) + A(\xi(t))u(t) + \Delta b(t, \xi(t)), \quad \Delta b(t, \xi(t)) = E\, Q(\xi, t), \]  
(20)

\[ u_f = u_n + u_r, \]  
(14)

\[ u_n = A^{-1}(\xi)[v_s(t, \xi) - b(\xi)], \]  
(9)

\[ u_r = -A^{-1}(\xi)[\hat{B}^TP\hat{z}\|Q(\xi, t)\|^2 + \hat{E}(t)Q(\xi, t)]. \]  
(21)

\[ \hat{A}^TP + P\hat{A} = -I. \]  
(13)

\[ y^{(r)}(t) = [y_1^{(r_1)}(t), y_2^{(r_2)}(t), \ldots, y_m^{(r_m)}(t)]^T, \]

\[ \xi(t) = [\xi^1(t), \ldots, \xi^m(t)]^T, \]

\[ \xi^i(t) = [y_i(t), \ldots, y_i^{(r_i-1)}(t)]. \quad 1 \leq i \leq m \]

\[ v_{si} = y_i^{(r_i)}(t) - K_{r_i}^i(y_i^{(r_i-1)}(t) - y_{id}^{(r_i-1)}) - \cdots - K_1^i(y_i - y_{id}). \]

\( A, b \) are functions of \( f, g, \) and \( h), \) \( \hat{B} \) is a sparse matrix of 0s and 1s, \( \hat{A} \) is function of the feedback gains.
Learning-based indirect adaptive iterative control for nonlinear systems affine in the control variable

Multi-parametric ESC uncertainties estimator (Data-driven)

\[ J(\hat{\Delta}) = F(z(\hat{\Delta})), \hat{\Delta}(t) = [\hat{E}(1, 1), \ldots, \hat{E}(m, m)]^T \]

where \( F : \mathbb{R}^n \rightarrow \mathbb{R}, F(\mathbf{0}) = 0, F(z) > 0 \) for \( z \in \mathbb{R}^n - \{\mathbf{0}\} \).

\[ \dot{x}_i = a_i \sin(\omega_i t + \frac{\pi}{2})J(\hat{\Delta}), a_i > 0, \quad i \in \{1, 2, \ldots, m^2\} \]

\[ \hat{\Delta}_i(t) = \hat{\Delta}_i - \text{nominal} + \delta \Delta_i(t), \]

\[ \delta \Delta_i(t) = \hat{\delta} \Delta_i((I - 1)t_f), (I - 1)t_f \leq t \leq It_f, \]
Learning-based indirect adaptive iterative control for nonlinear systems affine in the control variable

Algorithm 1 MES-based Learning Adaptive Controller

- Initialize: $I = 1$, $x(0) = x_0$, $J_{th} > 0$, $\hat{\Delta} = \Delta_{nominal}$, $K^i_1, \ldots, K^i_{r_i}, i = 1, \ldots, m$.
- Solve (13).
- Apply the controller (9), (14), and (21), to (3), (20).

(Loop) – Evaluate the learning cost $J$ by (24).
- IF $J \leq J_{th}$ → Exit Loop, IF not:
  - I=I+1.
  - Estimate $\hat{\Delta}$ by (25).
  - Reset $t \in [(I - 1)t_f, It_f]$, $x((I - 1)t_f) = x_0$, then, apply the controller (9), (14), and (21), to (3), (20).
- Go to (Loop).
Learning-based indirect adaptive iterative control for nonlinear systems affine in the control variable

GP-UCB* uncertainties estimator (Data-driven)

- Gaussian process upper confidence bound GP-UCB* is used as the data-driven part of the controller
- Bayesian stochastic optimization, i.e., noisy observation of the cost function
- Global optimum on compact search sets

Learning-based indirect adaptive iterative control for nonlinear systems affine in the control variable

**GP-UCB uncertainties estimator (Data-driven)**

Let us assume that $\tilde J$ is a function sampled from a Gaussian Process (GP).

We recall that GP is defined by a mean function

$$\mu(\hat{\Delta}) = \mathbb{E} \left[ \tilde J(\hat{\Delta}) \right],$$

and its covariance function (or kernel)

$$k(\hat{\Delta}, \hat{\Delta}') = \text{Cov}(\tilde J(\hat{\Delta}), \tilde J(\hat{\Delta}')) = \mathbb{E} \left[ (\tilde J(\hat{\Delta}) - \mu(\hat{\Delta})) (\tilde J(\hat{\Delta}') - \mu(\hat{\Delta}'))^\top \right].$$

E.g., $k(\hat{\Delta}, \hat{\Delta}') = \exp \left( -\frac{\|\hat{\Delta} - \hat{\Delta}'\|^2}{2l^2} \right), \quad (32)$

as the squared exponential kernel with length scale $l > 0$. 
Learning-based indirect adaptive iterative control for nonlinear systems affine in the control variable

**GP-UCB uncertainties estimator (Data-driven)**

Let us first briefly describe how we can find the posterior distribution of a GP(0, K), i.e., a GP with zero prior mean. Suppose that for $\Delta_{I-1} \triangleq \{\hat{\Delta}_1, \hat{\Delta}_2, \ldots, \hat{\Delta}_{I-1}\} \subset D$, we have observed the noisy evaluation $y_i = \tilde{J}(\hat{\Delta}_i) = J(\hat{\Delta}_i) + \eta_i$ with $\eta_i \sim N(0, \sigma^2)$ being i.i.d. Gaussian noise. We can find the posterior mean and variance for a new point $\hat{\Delta}^* \in D$ as follows: Denote the vector of observed values by $y_{I-1} = [y_1, \ldots, y_{I-1}]^T \in \mathbb{R}^{I-1}$, and define the Grammian matrix $K \in \mathbb{R}^{I-1 \times I-1}$ with $[K]_{i,j} = K(\hat{\Delta}_i, \hat{\Delta}_j)$, and the vector $K_* = [K(\hat{\Delta}_1, \hat{\Delta}^*), \ldots, K(\hat{\Delta}_{I-1}, \hat{\Delta}^*)]$. The expected mean $\mu_I(\hat{\Delta}^*)$ and the variance $\sigma^2_I(\hat{\Delta}^*)$ of the posterior of the GP evaluated at $\hat{\Delta}^*$ are (cf. Section 2.2 of [63])

$$\mu_I(\hat{\Delta}^*) = K_* [K + \sigma^2 I]^{-1} y_{I-1},$$

(33)

$$\sigma^2_I(\hat{\Delta}^*) = K(\hat{\Delta}^*, \hat{\Delta}^*) - K_*^T [K + \sigma^2 I]^{-1} K_*.$$

(34)
Learning-based indirect adaptive iterative control for nonlinear systems affine in the control variable

GP-UCB uncertainties estimator (Data-driven)

At iteration $I$, the GP-UCB algorithm selects the next query point $\hat{\Delta}_I$ by solving the following optimization problem:

$$
\hat{\Delta}_I \leftarrow \arg\min_{\hat{\Delta} \in D} \mu_{I-1}(\hat{\Delta}) - \beta_I^{1/2} \sigma_{I-1}(\hat{\Delta}).
$$

(35)

**Remark 12.** The optimization problem (35) is often nonlinear and nonconvex. Nonetheless, solving it only requires querying the GP, which, in general, is much faster than querying the original dynamical system. This is important when the dynamical system is a real system and we would like to minimize the number of interactions with it before finding a $\hat{\Delta}$ with small $J(\hat{\Delta})$. One practical way to approximately solve (35) is to restrict the search to a finite subset $D'$ of $D$. The finite subset can be a uniform grid structure over $D$ or it might consist of randomly selected members of $D$.

Learning-based indirect adaptive iterative control for nonlinear systems affine in the control variable

Algorithm 2 GP-UCB-based Learning Adaptive Controller

- Initialize: $I = 1, \ x(0) = x_0, \ J_{th} > 0, \ \hat{\Delta} = \Delta_{\text{nominal}}$.
- Apply the controller (9), (14), and (21), to (3), (20).

(Loop) – Evaluate the learning cost $J$ by (24).

- IF $J \leq J_{th}$ → Exit Loop, IF not:
  - I=I+1.
  - Estimate $\hat{\Delta}$ by (32), (33), (34), (35), and (36).
  - Reset $t \in [(I - 1)t_f, \ t_f]$, $x((I - 1)t_f) = x_0$, then, apply the controller (9), (14), and (21), to (3), (20).
- Go to (Loop).

Adaptive dynamic programming

Linear time-invariant model

$$\dot{x} =Ax + Bu, \ x \in \mathbb{R}^n, \ u \in \mathbb{R}^m$$

where, $A$ is unknown.

The pair $(A, B)$ assumed to be stabilizable.

LQR-type cost function of the form

$$V(u) = \int_{t_0}^{\infty} (x^T(\tau)R_1x(\tau) + u^T(\tau)R_2u(\tau))d\tau,$$

$R_1 \geq 0, \ R_2 > 0$.

$$u^*(t) = -K x(t), \quad K = R_2^{-1} B^T P,$$


** More details about ADP algorithms can be found in these two talks by F. Lewis:

https://lewisgroup.uta.edu/FL%20talks%202017/2018%2005%20RL%20main.pdf
https://www3.nd.edu/~pantsakl/Archive/WolovichSymposium/files/Lewis_Presentation.pdf
Adaptive dynamic programming

$P$ solution of the Riccati equation

$$A^T P + PA - PBR_2^{-1}B^T P + Q = 0,$$

$A$ unknown! $\rightarrow$ Learning $P$

Integral reinforcement learning policy iteration algorithm (IRL-PIA):

$$x^T P_i x = \int_t^{t+T} x^T(\tau)(R_1 + K_i^T R_2 K_i)x(\tau)d\tau + x^T(t + T)P_i x(t + T),$$

$$K_{i+1} = R_2^{-1}B^T P_i, \ i = 1, 2, \ldots$$

where the initial gain $K_1$ is chosen such that $A - BK_1$ is stable.

Under conditions of stabilizability/detectability:

$$u^*(t) \rightarrow \text{argmin}_{u(t)} V(u), \ t \in [t_0, \infty[.$$
Control barrier function (CBF)-based learning control*, **

* Emam et al. 21, Safe Model-Based Reinforcement Learning using Robust Control Barrier Functions, arXiv:2110.05415v1.


Figure from [*] with the addition of the yellow part
Control barrier function (CBF)-based learning control*,**,

Model:
\[
\dot{x}(t) = f(x(t)) + g(x(t))u(x(t)) + d(x(t)), d \in D \\
D(x') = \text{co} \Psi(x') = \text{co}\{\psi_1(x') \ldots \psi_p(x')\}, \forall x' \in \mathbb{R}^n,
\]

Policy:
\[
u_{\text{RCBF}}(x') = u^*(x') + u^{\text{RL}}(x').
\]

Filter:
\[
u^{\text{RL}}(x') \sim \pi_\phi(\cdot|x') \\
u^*(x') = \arg \min_{u \in \mathbb{R}^m} \|u\|^2 + \epsilon^2 \\
s.t. \nabla h(x')^\top (f(x') + g(x')(u(x') + u^{\text{RL}}(x'))) \geq \\
- \alpha(h(x')) - \min \nabla h(x')^\top \Psi(x') + \epsilon
\]

* Emam et al. 21, Safe Model-Based Reinforcement Learning using Robust Control Barrier Functions, arXiv:2110.05415v1.

Main points

- Learning-based adaptive control for nonlinear systems with constant/time-varying parametric uncertainties (ESC, GP-UCB, ADP, CBF)*

- Learning-based iterative feedback gains tuning for nonlinear systems affine in the control (ESC)

- Indirect learning-based adaptive iterative control for linear systems under constraints (ESC-MPC framework)

- Learning-based adaptive PDE stable model reduction/estimation (ESC, RL)

* ESC: Extremum seeking control, GP-UCB: Gaussian process upper confidence bound, ADP: Adaptive dynamic programming, CBF: Control barrier function, RL: Reinforcement learning
Learning-based iterative feedback gains auto-tuning for nonlinear systems *

\[
\dot{x} = f(x) + \Delta f(x) + g(x)u, \quad x(0) = x_0
\]
\[
y = h(x),
\]

where \(x \in \mathbb{R}^n, u \in \mathbb{R}^{n_a}, y \in \mathbb{R}^m\) \((n_a \geq m)\).

Assumption 1: \(f : \mathbb{R}^n \to \mathbb{R}^n\) and the columns of \(g : \mathbb{R}^n \to \mathbb{R}^{n \times n_a}\) are \(C^\infty\) vector fields on a bounded set \(X\) of \(\mathbb{R}^n\) and \(h(x)\) is a \(C^\infty\) function on \(X\). The vector field \(\Delta f(x)\) is \(C^1\) on \(X\).

* M. Benosman, 2016, Multi-Parametric Extremum Seeking-based Auto-Tuning for Robust Input-Output Linearization Control", International Journal of Robust and Nonlinear Control, 26(18), 4035-4055.
Learning-based iterative feedback gains
auto-tuning for nonlinear systems

Assumption 2: System (1) has a well-defined (vector) relative degree \( \{r_1, \ldots, r_m\} \) at each point \( x^0 \in X \), and the system is linearizable, i.e. \( \sum_{i=1}^{i=m} ri = n \)

Assumption 3: The uncertainty vector \( \Delta f \) is s.t. \( |\Delta f(x)| \leq d(x) \ \forall x \in X \), where \( d : X \rightarrow \mathbb{R} \) is a smooth nonnegative function.
Learning-based iterative feedback gains
auto-tuning for nonlinear systems

Assumption 4: The desired output trajectories $y_{id}$ are smooth functions of time, relating desired initial points $y_{i0}$ at $t = 0$ to desired final points $y_{if}$ at $t = t_f$, and s.t. $y_{id}(t) = y_{if}$, $\forall t \geq t_f$, $t_f > 0$, $i \in \{1, ..., m\}$.

Control objectives

- uniform boundedness of a tracking error,
- feedback gains vector $K$ is iteratively auto-tuned, to optimize a desired performance
Learning-based iterative feedback gains auto-tuning for nonlinear systems

**Controller design:** (using I/O linearization and Lyapunov reconstruction)

**Step one: Passive robust control design**

\[
    u = A^{-1}(\xi)(v_s(t, \xi) - b(\xi)) - A^{-1}(\xi)\frac{\partial V^T}{\partial z}k_{d_2}(e), \quad k > 0, \quad v_s = (v_{s1}, ..., v_{sm})^T,
\]

\[
v_{si}(t, \xi) = y_{id}^{(ri)} - K_{ri}^{i} \left(y_{i}^{(ri-1)} - y_{id}^{(ri-1)}\right) - ... - K_{1}^{i}(y_{i} - y_{id}).
\]

\[
    V = z^T P z, \quad P > 0 \quad P \tilde{A} + \tilde{A}^T P = -I
\]

\[
z = (z^1, ..., z^m)^T, \quad z^i = (e_i, ..., e_i^{ri-1}), \quad \tilde{z} = (z^1(r_1), ..., z^m(r_m))^T \in \mathbb{R}^m
\]

\[
e_i(t) = y_i(t) - y_{id}(t)
\]

\[d_2(.)\text{ is an upper bound of the uncertainty}
\]
\[\tilde{A}\text{ is a block diagonal matrix of the feedback gains}
\]
Learning-based iterative feedback gains auto-tuning for nonlinear systems

**Controller design**

Step two: *Iterative tuning of the feedback gains*

\[ Q(z(\beta)) = \int_{(I-1)t_f}^{It_f} z^T(t)C_1z(t)dt + \int_{(I-1)t_f}^{It_f} u^T(t)C_2u(t)dt, \]

\[ I = 1, 2, 3..., \quad C_1, \quad C_2 > 0 \]
Learning-based iterative feedback gains auto-tuning for nonlinear systems

Step two: Iterative tuning of the feedback gains

$$\beta = [\delta K^1_1, ..., \delta K^1_{r_1}, ..., \delta K^m_1, ..., \delta K^m_{r_m}, \delta k]^T$$

$$K^i_j = K^i_j - \text{nominal} + \delta K^i_j, \ j = 1, \ldots, r_i, \ i = 1, \ldots, m.$$  
$$k = k_{\text{nominal}} + \delta k, \ k_{\text{nominal}} > 0$$

$$\dot{x}_{K^i_j} = a_{K^i_j} \sin(\omega_{K^i_j} t - \frac{\pi}{2}) Q(z(\beta))$$

$$\delta \dot{K}^i_j(t) = x_{K^i_j}(t) + a_{K^i_j} \sin(\omega_{K^i_j} t + \frac{\pi}{2}), \ j = 1, \ldots, r_i, \ i = 1, \ldots, m$$

$$\dot{x}_k = a_k \sin(\omega_k t - \frac{\pi}{2}) Q(z(\beta))$$

$$\delta \dot{k}(t) = x_k(t) + a_k \sin(\omega_k t + \frac{\pi}{2})$$,
Learning-based iterative feedback gains auto-tuning for nonlinear systems

Step two: *Iterative tuning of the feedback gains*

\[ \omega_1 + \omega_2 \neq \omega_3, \text{ for } \omega_1 \neq \omega_2 \neq \omega_3, \]
\[ \forall \omega_1, \omega_2, \omega_3 \in \{\omega_{K_i}^j, \omega_k, j = 1, \ldots ri, i = 1, \ldots, m\}, \]

with \( \omega_i > \omega^* \), \( \forall \omega_i \in \{\omega_{K_i}^j, \omega_k, j = 1, \ldots ri, i = 1, \ldots, m\}, \omega^* \)

large enough.
Learning-based iterative feedback gains auto-tuning for nonlinear systems

Step two: **Iterative tuning of the feedback gains**

**Assumption 7**: We assume that the cost function $Q$ has a local minimum at $\beta^*$.

**Assumption 8**: We consider that the initial gain vector $\beta$ is sufficiently close to the optimal gain vector $\beta^*$.

**Assumption 9**: The cost function is analytic and its variation with respect to the gains is bounded in the neighborhood of $\beta^*$, i.e. $|\frac{\partial Q}{\partial \beta}(\tilde{\beta})| \leq \Theta_2$, $\Theta_2 > 0$, $\tilde{\beta} \in \mathcal{V}(\beta^*)$, where $\mathcal{V}(\beta^*)$ denotes a compact neighborhood of $\beta^*$. 
Learning-based iterative feedback
gains auto-tuning for nonlinear systems

Put together: Robust controller + ESC tuning

\[
\begin{align*}
\dot{y}_{id}(t) &= y_{id}(t - (I - 1)t_f), \quad (I - 1)t_f \leq t < It_f, \quad I \in \{1, 2, \ldots\}, \\
K^i_j(t) &= K^i_{j-nominal} + \delta K^i_j(t) \\
\delta K^i_j(t) &= \delta \hat{K}^i_j((I - 1)t_f), \quad (I - 1)t_f \leq t < It_f, \\
k(t) &= k_{nominal} + \delta k(t), \quad k_{nominal} > 0 \\
\delta k(t) &= \delta \hat{k}((I - 1)t_f), \quad (I - 1)t_f \leq t < It_f, \quad I = 1, 2, 3, \ldots \\
\hat{K}^i_j, \delta \hat{k} & \text{ are estimated by the MES algorithm.}
\end{align*}
\]
Learning-based iterative feedback gains auto-tuning for nonlinear systems

- the obtained closed-loop impulsive time-dependent dynamic system is well posed,
- the tracking error $z$ is uniformly bounded
- $z$ is steered at each iteration $I$ towards the positive invariant set $S_I = \{ z \in \mathbb{R}^n \mid 1 - k_I \left| \frac{\partial V}{\partial z_{in}} \right| \geq 0 \}$

$$k_I = \beta_I(n + 1)$$

$$|Q(\beta(I t_f)) - Q(\beta^*)| \leq \Theta_2 \left( \frac{\Theta_1}{\omega_0} + \sqrt{\sum_{i=1,\ldots,m} \sum_{j=1,\ldots,r_i} a_{K_j}^2 + a_k^2} \right)$$

$\Theta_1, \Theta_2 > 0$, for $I \to \infty, \omega_0 = \text{Max}(\omega_{K_1}, \ldots, \omega_{K_{r_m}}, \omega_k)$. 

Learning-based iterative feedback gains auto-tuning for nonlinear systems

- $\beta$ remains bounded over the iterations s.t.

$$|\beta((I + 1)t_f) - \beta(It_f)| \leq 0.5t_f Max(a_{K_1}^2, \ldots, a_{K_m}^2, a_k^2)\Theta_2 + tf\omega_0 \sqrt{\sum_{i=1,\ldots,m} \sum_{j=1,\ldots,ri} a_{K_i}^2 + a_k^2}, \quad I \in \{1, 2, \ldots\}$$

- satisfies asymptotically the bound

$$|\beta(It_f) - \beta^*| \leq \frac{\Theta_1}{\omega_0} + \sqrt{\sum_{i=1,\ldots,m} \sum_{j=1,\ldots,ri} a_{K_i}^2 + a_k^2}, \quad \Theta_1 > 0, \text{ for } I \to \infty$$
Main points

- Learning-based adaptive control for nonlinear systems with constant/time-varying parametric uncertainties (ESC, GP-UCB, ADP, CBF)*

- Learning-based iterative feedback gains tuning for nonlinear systems affine in the control (ESC)

- Indirect learning-based adaptive iterative control for linear systems under constraints (ESC-MPC framework)

- Learning-based adaptive PDE stable model reduction/estimation (ESC, RL)

* ESC: Extremum seeking control, GP-UCB: Gaussian process upper confidence bound, ADP: Adaptive dynamic programming, CBF: Control barrier function, RL: Reinforcement learning
Indirect learning-based adaptive iterative control for linear systems under constraints (MPC framework) *, **

\[
x(k + 1) = (A + \Delta A)x(k) + (B + \Delta B)u(k)
\]
\[
y(k) = (C + \Delta C)x(k) + (D + \Delta D)u(k),
\]
\[x \in \mathbb{R}^n, \ u \in \mathbb{R}^m, \ y \in \mathbb{R}^p\]
\[x_{\text{min}} \leq x(k) \leq x_{\text{max}},\]
\[u_{\text{min}} \leq u(k) \leq u_{\text{max}},\]
\[y_{\text{min}} \leq y(k) \leq y_{\text{max}},\]
\[r_r(k + 1) = A_r r_r(k), \ y_e(k) = C x(k) - C_r r_r(k),\]


** Subbaraman S., Benosman M., 2016, Extremum Seeking-based Iterative Learning Model Predictive Control (ESILC-MPC), IFAC International Workshop on Adaptation and Learning in Control and Signal Processing (follow up paper with convergence proofs).
Indirect learning-based adaptive iterative control for linear systems under constraints (MPC framework) *, **

Assumption 1: The constant uncertainty matrices $\Delta A$, $\Delta B$, $\Delta C$ and $\Delta D$, are bounded, s.t. $\|\Delta A\|_2 \leq l_A$, $\|\Delta B\|_2 \leq l_B$, $\|\Delta C\|_2 \leq l_C$, $\|\Delta D\|_2 \leq l_D$, with $l_A$, $l_B$, $l_C$, $l_D > 0$.

Assumption 2: There exists non empty convex sets $\mathcal{K}_a \subset \mathbb{R}^{n \times n}$, $\mathcal{K}_b \subset \mathbb{R}^{n \times m}$, $\mathcal{K}_c \subset \mathbb{R}^{p \times n}$, and $\mathcal{K}_d \subset \mathbb{R}^{p \times m}$, such that $A + \Delta A \in \mathcal{K}_a$ for all $\Delta A$ such that $\|\Delta A\|_2 \leq l_A$, $B + \Delta B \in \mathcal{K}_b$ for all $\Delta B$ such that $\|\Delta B\|_2 \leq l_B$, $C + \Delta C \in \mathcal{K}_c$ for all $\Delta C$ such that $\|\Delta C\|_2 \leq l_C$, $D + \Delta D \in \mathcal{K}_d$ for all $\Delta D$ such that $\|\Delta D\|_2 \leq l_D$.

Assumption 3: The iterative learning MPC problem (and the associated reference tracking extension), is a well-posed optimization problem for any matrices $A + \Delta A \in \mathcal{K}_a$, $B + \Delta B \in \mathcal{K}_b$, $C + \Delta C \in \mathcal{K}_c$, $D + \Delta D \in \mathcal{K}_d$. 
Indirect learning-based adaptive iterative control for linear systems under constraints (MPC framework) *, **

\[ Q(\hat{\Delta}) = F(y_e(\hat{\Delta})) \text{, different from the MPC cost} \]

where \( \hat{\Delta} \) is the vector obtained by concatenating the estimated uncertainty matrices \( \Delta \hat{A}, \Delta \hat{B}, \Delta \hat{C} \) and \( \Delta \hat{D} \),

\[ F : \mathbb{R}^p \rightarrow \mathbb{R}, \; F(0) = 0, \; F(y_e) > 0 \text{ for } y_e \neq 0. \]

Assumption 4: The cost function \( Q \) has a local minimum at \( \hat{\Delta}^* = \Delta \).

Assumption 5: The original parameter estimate vector \( \hat{\Delta} \) is close enough to the actual parameters vector \( \Delta \).

Assumption 6: The cost function is analytic and its variation with respect to the uncertain variables is bounded in the neighborhood of \( \Delta^* \), i.e., there exists \( \xi_2 > 0 \), s.t. \( \left\| \frac{\partial Q}{\partial \Delta}(\tilde{\Delta}) \right\| \leq \xi_2 \) for all \( \tilde{\Delta} \in \mathcal{N}(\Delta^*) \), where \( \mathcal{N}(\Delta^*) \) denotes a compact neighborhood of \( \Delta^* \).
Indirect learning-based adaptive iterative control for linear systems under constraints (MPC framework) *, **

\[
\dot{z}_i = a_i \sin(\omega_i t + \frac{\pi}{2}) Q(\hat{\Delta}) \\
\hat{\Delta}_i = z_i + a_i \sin(\omega_i t - \frac{\pi}{2}), \quad i \in \{1, \ldots, N_p\}
\]
with \( N_p \leq nn + nm + pn + pm \) is the number of uncertain elements, \( \omega_i \neq \omega_j, \omega_i +\omega_j \neq \omega_k, \ i, j, k \in \{1, \ldots, N_p\} \), and \( \omega_i > \omega^*, \ \forall i \in \{1, \ldots, N_p\} \), with \( \omega^* \) large enough, converges to the local minima of \( Q \).

Main points

- Learning-based adaptive control for nonlinear systems with constant/time-varying parametric uncertainties (ESC, GP-UCB, ADP, CBF)*

- Learning-based iterative feedback gains tuning for nonlinear systems affine in the control (ESC)

- Indirect learning-based adaptive iterative control for linear systems under constraints (ESC-MPC framework)

- Learning-based adaptive PDE stable model reduction/estimation (ESC, RL)

* ESC: Extremum seeking control, GP-UCB: Gaussian process upper confidence bound, ADP: Adaptive dynamic programming, CBF: Control barrier function, RL: Reinforcement learning
Learning-based PDE stable model reduction *

Consider a stable dynamical system modeled by a nonlinear partial differential equation of the form

\[ \dot{z} = \mathcal{F}(\mathbf{z}, \mu) \in \mathcal{Z}, \mu \in \mathbb{R} \]

where \( \mathcal{Z} \) is an infinite-dimension Hilbert space.

\[ P_n \mathbf{z}(t, x) \approx \Phi \mathbf{z}_r(t) = \sum_{i=1}^{r} z_{r_i}(t) \phi_i(x) \in \mathbb{R}^n \]

where \( P_n \) is the projection of \( \mathbf{z}(t, x) \) onto \( \mathbb{R}^n \).

\[ \dot{\mathbf{z}}_r(t) = F(\mathbf{z}_r(t), \mu) \]

The function \( F : \mathbb{R}^r \rightarrow \mathbb{R}^r \) is obtained from the weak form of the original PDE (through Galerkin projection).

The Closure-Model Concept for ROMs Stabilization

\[ \frac{\dot{q}(t)}{q(t)} = F(q(t), \mu) + H(t, q(t)) \]

1) Closure models with constant eddy viscosity coefficients:

\( \mu \) is substituted by a virtual viscosity coefficient \( \mu_{cl} \), \( \mu_{cl} = \mu + \mu_{e} \), Heisenberg ROM

2) Closure models with time and space varying eddy viscosity coefficients:

\[ H_{nev}(\mu_{e}, q(t)) = \mu_{e} \sqrt{\frac{V(q(t))}{V_{\infty}(\lambda)}} \text{diag}(d_{11}, \ldots, d_{rr}) q(t), \]

\( V(q) = \frac{1}{2} \sum_{i=1}^{i=r} q_i^2, \ V_{\infty}(\lambda) = \frac{1}{2} \sum_{i=1}^{i=r} \lambda_i, \)

the \( \lambda_i \) are the selected POD eigenvalues

where \( D \in \mathbb{R}^{r \times r} \) represents a constant viscosity damping matrix,
A Lyapunov-based closure-Model for Robust ROMs

Stabilization

original PDE $\rightarrow$ Using POD $\rightarrow$ \[
\begin{aligned}
\dot{q}_{\text{pod}}(t) &= \tilde{F}(q_{\text{pod}}(t)) + \mu \, D q_{\text{pod}}, \quad D < 0 \\
z_{\text{pod}}(t, x) &= \sum_{i=1}^{N_{\text{pod}}} \phi_i^\text{pod}(x) q_i^\text{pod}(t),
\end{aligned}
\]

Assumption 1 The norm of the vector field $\tilde{F}$ is bounded by a known function of $q^\text{pod}$, i.e., $\|\tilde{F}(q^\text{pod})\| \leq \tilde{f}(q^\text{pod})$.

Assumption 2 The solutions of the original PDE model are assumed to be in $L^2([0, \infty); \mathbb{Z})$.

Then, the nonlinear closure model

\[
H_{nl} = \mu_{nl} \tilde{f}(q^\text{pod}) \, \text{diag}(d_1, \ldots, d_{N_{\text{pod}}N_{\text{pod}}}) q^\text{pod}, \quad \mu_{nl} > 0
\]

stabilizes the solutions of the ROM to the invariant set

\[
S = \{q^\text{pod} \in \mathbb{R}^{N_{\text{pod}}} \text{ s.t. } \mu \frac{\lambda(D)_{\text{max}}}{\tilde{f}} q^\text{pod} + \mu_{nl} q^\text{pod} \|\text{Max}(d_1, \ldots, d_{N_{\text{pod}}N_{\text{pod}}}) + 1 \geq 0\}.
\]
An Extremum Seeking–based Auto-Tuning of Closure Models for ROMs

\[ \mu \text{ is substituted by a virtual viscosity coefficient } \mu_{cl}, \quad \mu_{cl} = \mu + \mu_e, \text{ Heisenberg ROM} \]

if the closure models amplitudes \( \mu_e, \mu_{nl} \) are tuned using the MES algorithm

\[
\begin{align*}
\dot{y}_1 &= a_1 \sin(\omega_1 t + \frac{\pi}{2})Q(\hat{\mu}_e, \hat{\mu}_{nl}) \\
\dot{\mu}_e &= y_1 + a_1 \sin(\omega_1 t - \frac{\pi}{2}) \\
\dot{y}_2 &= a_2 \sin(\omega_2 t + \frac{\pi}{2})Q(\hat{\mu}_e, \hat{\mu}_{nl}) \\
\dot{\mu}_{nl} &= y_2 + a_2 \sin(\omega_2 t - \frac{\pi}{2}),
\end{align*}
\]

where \( \omega_{max} = \max(\omega_1, \omega_2) > \omega^*, \omega^* \) large enough, and \( Q \) the learning cost function

\[
Q(\hat{\mu}) = H(e_z(\hat{\mu})), \quad \hat{\mu} = (\hat{\mu}_e, \hat{\mu}_{nl})
\]

\[
e_z(t) = z^{pod}(t, x) - z(t, x), \quad H \text{ is a positive definite function of } e_z
\]

Assumption 3 The learning cost function \( Q \) has a local minimum at \( \hat{\mu} = \mu^* \).

Assumption 4 The learning cost function \( Q \) is analytic and its variation with respect to \( \mu \) is bounded in the neighborhood of \( \mu^* \), i.e., \( \| \frac{\partial Q}{\partial \mu}(\tilde{\mu}) \| \leq \xi_2, \xi_2 > 0, \tilde{\mu} \in \mathcal{V}(\mu^*), \) where \( \mathcal{V}(\mu^*) \) denotes a compact neighborhood of \( \mu^* \).
An Extremum Seeking–based Auto-Tuning of Closure Models for ROMs

Then, the norm of the vector

\[ e_\mu = (\mu_e^* - \hat{\mu}_e(t), \mu_{nl}^* - \hat{\mu}_{nl}(t)) \]

admits the following bound

\[ \|e_\mu(t)\| \leq \frac{\xi_1}{\omega_{\max}} + \sqrt{a_1^2 + a_2^2}, \quad t \to \infty \]

where \( a_1, \ a_2 > 0, \ \xi_1 > 0 \), and the learning cost function approaches its optimal value within the following upper-bound

\[ \|Q(\hat{\mu}_e, \hat{\mu}_{nl}) - Q(\mu_e^*, \mu_{nl}^*)\| \leq \xi_2 \left( \frac{\xi_1}{\omega} + \sqrt{a_1^2 + a_2^2} \right), \quad t \to \infty \]

where \( \xi_2 = \max_{(\mu_1, \mu_2)} \in V(\mu^*) \left| \frac{\partial Q}{\partial \mu} \right| \).
Learning-based observers

RL-based observer

Full-order model

\[
\begin{align*}
z_{k+1} &= f(z_k) & \text{full state } z_k & \in \mathbb{R}^n \\
y_k &= Cz_k & \text{measurement } y_k & \in \mathbb{R}^p
\end{align*}
\]

Reduced-order model (ROM)

\[
\begin{align*}
x_{k+1} &= Ax_k & \text{reduced state } x_k & \in \mathbb{R}^r \\
y_k &= C_rx_k & \text{measurement } y_k & \in \mathbb{R}^p
\end{align*}
\]

\[\begin{align*}
x_k &= U^Tz_k, \\
z_{k+1} &\approx Az_k
\end{align*}\]

dimensionality reduction

dynamics approximation

How to do the data assimilation?

Kalman filter (conventional approach)

\[
\hat{x}_k = A_r\hat{x}_{k-1} + K_k(y_k - C_rA_r\hat{x}_{k-1})
\]

Challenge: performs poorly when \( A_r \) is not a good model

Reinforcement learning-trained filter (what we propose)

\[
\hat{x}_k = A_r\hat{x}_{k-1} + a_k \quad \text{where} \quad a_k \sim \pi_{\theta}(\cdot | y_k, \hat{x}_{k-1})
\]

Flexibility of nonlinear policy \( \pi_{\theta} \) allows to compensate for errors in \( A_r \)


** Benosman et al., 2020, Reinforcement Learning-based Model Reduction for Partial Differential Equations, World Congress of the International Federation of Automatic Control (IFAC).
Open theoretical problems?

- Robustness to hyper-parameters tuning
- Large scale systems and high dimensional systems, e.g., PDE models, delays
- Robustness and safety (state/input constraints) of ML algorithms from control theory perspective (e.g., stability and robustness of (CS-)RL algorithms using dynamical systems theory tools, neural ODEs from dynamical systems perspective (useful/scalable ?))
- Sampling efficiency/data constraints
- Real-time computational constraints
- …

* CS: Computer science, RL: Reinforcement learning, ODEs: Ordinary diff. equations, PDEs: Partial diff. equations
A hybrid approach to control: classical control theory meets data-driven methods

Mouhacine Benosman
MERL - Mitsubishi Electric Research Labs, Cambridge, USA

Part II: Examples
Mechatronics Examples: Electromagnetic brakes*
Mechatronics Examples: Electromagnetic brakes*

Mechatronics Examples: Electromagnetic brakes

Trajectory generation

\( x_{\text{ref}}(t), v_{\text{ref}}(t), a_{\text{ref}}(t) \)

Electromagnetic actuator system

\( u(t) \)

Nonlinear Model

\[
\begin{align*}
\dot{x}(t) &= v(t) \\
\dot{v}(t) &= F_1(x(t), v(t), i(t)) \\
\frac{di(t)}{dt} &= F_2(v(t), u(t), i(t))
\end{align*}
\]

Nonlinear model-base controller

\[ u(t) = F_3(x, v, i, x_{\text{ref}}, v_{\text{ref}}, a_{\text{ref}}, \theta) \]

Learning-based parameters estimation

\[ \hat{\theta} = F(t, x, v, x_{\text{ref}}, v_{\text{ref}}) \]

Parameters estimates

Coil control voltage

measurements

measurements
Mechatronics Examples: Electromagnetic brakes

- Mechanical part

\[ m\ddot{x} = k(x_0 - x) - 0.5 \frac{a}{(b + x)^2} i^2 - \eta \dot{x} - f_d \]

- Electrical part

\[ u = Ri + L(x) \frac{di}{dt} - \frac{a}{(a + x)^2} i \frac{dx}{dt}, \quad L(x) = \frac{a}{b + x} \]
Mechatronics Examples: Electromagnetic brakes

\[
m \frac{d^2 x}{dt^2} = k(x_0 - x) - \eta \frac{dx}{dt} - \frac{a i^2}{2(b+x)^2} + f_d
\]

\[
u = R i + \frac{a}{b+x} \frac{di}{dt} - \frac{a i}{(b+x)^2} \frac{dx}{dt}, \quad 0 \leq x \leq x_f,
\]

\[
z := \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}^T = \begin{bmatrix} x & \dot{x} & i \end{bmatrix}^T
\]

\[
\dot{z}_1 = z_2
\]

\[
\dot{z}_2 = \frac{k}{m}(x_0 - z_1) - \frac{\eta}{m} z_2 - \frac{a}{2m(b+z_1)^2} z_3^2 + \frac{f_d}{m}
\]

\[
\dot{z}_3 = -\frac{R}{a} z_3 + \frac{z_3}{b+z_1} z_2 + \frac{u}{b+z_1}.
\]

(2)

\[
z_{1 \text{ ref}}(t_0) = z_{1 \text{ int}}, \quad z_{1 \text{ ref}}(t_f) = z_{1 f},
\]

\[
\dot{z}_{1 \text{ ref}}(t_0) = \dot{z}_{1 \text{ ref}}(t_f) = 0,
\]

\[
\ddot{z}_{1 \text{ ref}}(t_0) = \ddot{z}_{1 \text{ ref}}(t_f) = 0,
\]
Based on (i)ISS back-stepping approach

\[ u_{iss} \]

the cost function

\[ Q(\Delta) = \int_0^{t_f} q_1(z_1(s) - z_1^\text{ref}(s))^2 ds + \int_0^{t_f} q_2(z_2(s) - z_2^\text{ref}(s))^2 ds \]

\[ q_1, q_2 > 0. \]
Mechatronics Examples: Electromagnetic brakes

\[
\hat{k}(t) = k_{\text{nominal}} + \hat{\Delta}_k(t)
\]

\[
\hat{\eta}(t) = \eta_{\text{nominal}} + \hat{\Delta}_\eta(t)
\]

\[
\hat{f}_d(t) = f_{d-\text{nominal}} + \hat{\Delta}_f(t)
\]

\[
x_k(k') + 1 = x_k(k') + a_k t f \sin(\omega_k k' t_f + \frac{\pi}{2}) Q \]

\[
\hat{\Delta}_k(k') + 1 = x_k(k') + 1 + a_k \sin(\omega_k k' t_f - \frac{\pi}{2}),
\]

\[
x_{\eta}(k') + 1 = x_{\eta}(k') + a_{\eta} t f \sin(\omega_{\eta} k' t_f + \frac{\pi}{2}) Q \]

\[
\hat{\Delta}_{\eta}(k') + 1 = x_{\eta}(k') + 1 + a_{\eta} \sin(\omega_{\eta} k' t_f - \frac{\pi}{2}),
\]

\[
x_{f_d}(k') + 1 = x_{f_d}(k') + a_{f_d} t f \sin(\omega_{f_d} k' t_f + \frac{\pi}{2}) Q \]

\[
\hat{\Delta}_{f_d}(k') + 1 = x_{f_d}(k') + 1 + a_{f_d} \sin(\omega_{f_d} k' t_f - \frac{\pi}{2})
\]
Mechatronics Examples: Electromagnetic brakes

\[ \Delta k = -4.5, \quad \Delta \eta = -0.7 \quad \Delta f_d = -7.5 \]
\[
\Delta k = -4.5, \quad \Delta \eta = -0.7 \quad \Delta f_d = -7.5
\]
Mechatronics Examples: Electromagnetic brakes

\[
m \frac{d^2 x_a}{dt^2} = k(x_0 - x_a) - \eta \frac{dx_a}{dt} - \frac{a i^2}{2(b+x_a)^2}
\]

\[
u = Ri + \frac{a}{b+x_a} \frac{di}{dt} - \frac{ai}{(b+x_a)^2} \frac{dx_a}{dt}, \quad 0 \leq x_a \leq x_f,
\]

\[
x_{ref} \quad \text{a desired armature position trajectory, s.t.}
\]

\[
x_{ref}(0) = 0, \quad x_{ref}(t_f) = x_f, \quad \dot{x}_{ref}(0) = 0, \quad \dot{x}_{ref}(t_f) = 0.
\]

bounded parametric uncertainties

- spring coefficient \( k = k_{nominal} + \delta k, |\delta k| \leq \delta k_{max} \)
- the damping coefficient \( \eta = \eta_{nominal} + \delta \eta, \quad |\delta \eta| \leq \delta \eta_{max} \)
Passive robust controller:

\[
\begin{align*}
    u &= -\frac{m(b+x_a)}{i} \left( v_s + \frac{k_{\text{nominal}}}{m} \ddot{x}_a + \frac{\eta_{\text{nominal}}}{m} \dddot{x}_a - \frac{Ri^2}{(b+x_a)m} \right) + \\
    &+ \frac{m(b+x_a)}{i} \frac{\partial V}{\partial z_3} k \left( \frac{\delta k_{\text{max}}}{m} |\dot{x}_a| + \frac{\delta \eta_{\text{max}}}{m} |\dddot{x}_a| \right), \quad k > 0 \\
    v_s &= x^{(3)}_{\text{ref}}(t) + K_3 (x^{(2)}_a - x^{(2)}_{\text{ref}}(t)) + K_2 (x^{(1)}_a - x^{(1)}_{\text{ref}}(t)) \quad + K_1 (x_a - x_{\text{ref}}(t)), \quad K_i < 0, \ i = 1, 2, 3.
\end{align*}
\]

\[
V = z^TPz, \quad P > 0 \quad P \tilde{A} + \tilde{A}^TP = -I,
\]

\[
\tilde{A} = \begin{pmatrix}
    0 & 1 & 0 \\
    0 & 0 & 1 \\
    K_1 & K_2 & K_3
\end{pmatrix},
\]

where \(K_1, K_2, K_3\) are chosen such that \(\tilde{A}\) is Hurwitz.
Learning-based auto-tuning of the controller gains:

\[ Q(z(\beta)) = C_1 z_1 (It_f)^2 + C_2 z_2 (It_f)^2 + C_3 z_3 (It_f)^2, \]

\[ I = 1, 2, 3... \text{ is the number of iterations, } C_1, C_2 > 0, C_3 > 0, \]

\[ \beta = (\delta K_1, \delta K_2, \delta K_3, \delta k)', \]

\[ K_1 = K_{1\text{nominal}} + \delta K_1 \]

\[ K_2 = K_{2\text{nominal}} + \delta K_2 \]

\[ K_3 = K_{3\text{nominal}} + \delta K_3 \]

\[ k = k_{\text{nominal}} + \delta k, \]
Learning-based auto-tuning of the controller gains:

\[ \dot{x}_{K_1} = a_{K_1} \sin(\omega_1 t - \frac{\pi}{2})Q(z(\beta)) \]
\[ \delta\dot{K}_1(t) = x_{K_1}(t) + a_{K_1} \sin(\omega_1 t + \frac{\pi}{2}) \]
\[ \dot{x}_{K_2} = a_{K_2} \sin(\omega_2 t - \frac{\pi}{2})Q(z(\beta)) \]
\[ \delta\dot{K}_2(t) = x_{K_2}(t) + a_{K_2} \sin(\omega_2 t + \frac{\pi}{2}) \]
\[ \dot{x}_{K_3} = a_{K_3} \sin(\omega_3 t - \frac{\pi}{2})Q(z(\beta)) \]
\[ \delta\dot{K}_3(t) = x_{K_3}(t) + a_{K_3} \sin(\omega_3 t + \frac{\pi}{2}) \]
\[ \dot{x}_k = a_k \sin(\omega_4 t - \frac{\pi}{2})Q(z(\beta)) \]
\[ \delta\dot{k}(t) = x_k(t) + a_k \sin(\omega_4 t + \frac{\pi}{2}) \]
\[ \delta K_j(t) = \delta\dot{K}_j((I - 1)t_f), \quad (I - 1)t_f \leq t < It_f, \]
\[ j \in \{1, 2, 3\}, \quad I = 1, 2, 3... \]
\[ \delta k(t) = \delta\dot{k}((I - 1)t_f), \quad (I - 1)t_f \leq t < It_f, \quad I = 1, 2, 3... \]
Mechatronics Examples: Electromagnetic brakes
Mechatronics Examples: Electromagnetic brakes
Mechatronics Examples: DC- Servo motor with flexible shaft

The example studied here is about the angular position control of a load connected by a flexible shaft to a voltage actuated DC servo motor.

The states: the load angle, angular rate, the motor angle and angular rate

The input is the motor voltage

The outputs: the load angle and the torque acting on the flexible shaft

uncertainties $\delta \beta_l = -70, \ [Nms/\text{rad}], \delta J_l = -0.2, \ [kgm^2]$
Mechatronics Examples: DC- Servo motor with flexible shaft

uncertainties $\delta \beta_l = -70$, $[Nms/rad]$, $\delta J_l = -0.2$, $[kgm^2]$

$J_l = 25 kgm^2$, $\beta_l = 25 Nms/rad,$
Mechatronics Examples: DC- Servo motor with flexible shaft

uncertainties $\delta \beta_l = -70, \ [Nms/\text{rad}], \ \delta J_l = -0.2, \ [kgm^2]$

$J_l = 25kgm^2, \ \beta_l = 25Nms/\text{rad},$
Mechatronics Examples: DC-Servo motor with flexible shaft

uncertainties \( \delta \beta_l = -70, \ [Nms/\text{rad}], \ \delta J_l = -0.2, \ [kgm^2] \)

\[ J_l = 25kgm^2, \ \beta_l = 25Nms/\text{rad}, \]
Fluid dynamics applications

Efficient energy management in buildings
- HVAC (Heating, Ventilation, and Air Conditioning)


Optimizing HVAC performance is linked to modelling/controlling temperature and airflow in the room.

Hygiene applications: non-optimal vs optimal HVAC airflow to flush the virus out of the built environment.

How can we model airflow and temperature in a room with models that are precise and computationally trackable for real-time estimation and control?
Fluid dynamics applications:
The Coupled Burgers’ Equation*

\[
\begin{align*}
\frac{\partial w(t, x)}{\partial t} + w(t, x) \frac{\partial w(t, x)}{\partial x} &= \mu \frac{\partial^2 w(t, x)}{\partial x^2} - \kappa \frac{\partial T(t, x)}{\partial x}, \\
\frac{\partial T(t, x)}{\partial t} + w(t, x) \frac{\partial T(t, x)}{\partial x} &= c \frac{\partial^2 T(t, x)}{\partial x^2} + f(t, x).
\end{align*}
\]

Where \( w \) is the velocity variable, \( T \) is the temperature variable, \( f \) is a forcing disturbance function, \( \mu = \frac{1}{Re} \), where \( Re \) is the Reynolds number, \( c \) is the thermal conductivity, and \( \kappa \) is the thermal expansion coefficient. The notations \( F_x, F_{xx} \) stand for first and second partial derivatives of \( F \) w.r.t. \( x \), respectively. The forcing \( f \) is assumed to be at least \( L^2 \) in space and time.

\[
\begin{align*}
w(t, 0) &= w_L, \quad \frac{\partial w(t, 1)}{\partial x} = w_R, \\
T(t, 0) &= T_L, \quad T(t, 1) = T_R, \quad \omega_L, \omega_R, T_L, T_R \text{ are positive constants.}
\end{align*}
\]

The initial condition for the velocity is given by

\[
w(0, x) = w_0(x) \quad \in L^2([0, 1])
\]

and the initial condition for the temperature is

\[
T(0, x) = T_0(x) \quad \in L^2([0, 1])
\]

Fluid dynamics applications: 
The Coupled Burgers’ Equation

Following a Galerkin projection onto the subspace spanned by the POD basis functions, 
the coupled Burgers’ equation is reduced to a POD ROM with the following structure

\[
\begin{pmatrix}
\dot{q}_w \\
\dot{\hat{q}}_T
\end{pmatrix} = B_1 + \mu B_2 + \mu D q + \tilde{D}q + Cqq^T,
\]

\[\omega_n^{pod}(x, t) = \omega_{av}(x) + \sum_{i=1}^{i=r} \phi wi(x)q_{wi}(t),\]

\[T_n^{pod}(x, t) = T_{av}(x) + \sum_{i=1}^{i=r} \phi Ti(x)q_{Ti}(t),\]

in the form

\[
\begin{cases}
\dot{q}^{pod}(t) = F(q^{pod}(t), \mu) = \tilde{F}(q^{pod}(t)) + \mu Dq^{pod} \\
z^{pod}(t, x) = \sum_{i=1}^{i=N_{pod}} \phi_i^{pod}(x)q_i^{pod}(t),
\end{cases}
\]

\[\tilde{F} = B_1 + \mu B_2 + \tilde{D}q^{pod} + Cq^{pod}q^{podT},\]
Fluid dynamics applications:
The Coupled Burgers’ Equation

which can be upper-bounded by

\[
\tilde{F} \leq b_{1_{\text{max}}} + \mu_{\text{max}}b_{2_{\text{max}}} + \tilde{d}_{\text{max}}\|q^{\text{pod}}\| + c_{\text{max}}\|q^{\text{pod}}\|^2,
\]

where \(\|B_1 + \Delta B_1\|_F \leq b_{1_{\text{max}}}\), \(\|B_2 + \Delta B_2\|_F \leq b_{2_{\text{max}}}\), \(\mu \leq \mu_{\text{max}}\), \(\|\tilde{D} + \Delta \tilde{D}\|_F \leq \tilde{d}_{\text{max}}\), and \(\|C + \Delta C\|_F \leq c_{\text{max}}\).

This leads to the nonlinear closure model

\[
H_{nl} = \mu_{nl}(b_{1_{\text{max}}} + \mu_{\text{max}}b_{2_{\text{max}}} + \tilde{d}_{\text{max}}\|q^{\text{pod}}\| + c_{\text{max}}\|q^{\text{pod}}\|^2)\text{diag}(d_1, \ldots, d_{N_{\text{pod}}N_{\text{pod}}})q^{\text{pod}}
\]

Compete reduced order model

\[
\dot{q}(t) = F(q(t), \mu) + H(t, q(t)).
\]

\(H \rightarrow H_{nl}\)

\(\mu \rightarrow \mu_{\text{cl}} = \mu + \mu_e,\)
Fluid dynamics applications:
The Coupled Burgers’ Equation

boundary conditions \( w_L = w_R = 0, \ T_L = T_R = 0, \)

\[
  w(x, 0) = \begin{cases} 
    1, & \text{if } x \in [0, 0.5] \\
    0, & \text{if } x \in (0.5, 1],
  \end{cases}
\]

\[
  T(x, 0) = \begin{cases} 
    1, & \text{if } x \in [0, 0.5] \\
    0, & \text{if } x \in (0.5, 1],
  \end{cases}
\]

Learning cost

\[
  Q(\mu) = Q_1 \int_0^{t_f} < e_T, e_T >_X dt + Q_2 \int_0^{t_f} < e_w, e_w >_X dt, \quad Q_1 = Q_2 = 1
\]

\[
  \mu = (\mu_e, \mu_n l)^T
\]

\[
  e_T = T - T_n^{pod}, \quad e_w = w - w_n^{pod}
\]

Fluid dynamics applications: The Coupled Burgers' Equation

Velocity - True

Temperature - True

Velocity - POD ROM - No learning

Temperature - POD ROM - No learning

With 10 PODs
Fluid dynamics applications:  
The Coupled Burgers’ Equation

Error between ROM and systems’ measurements before learning
Fluid dynamics applications: The Coupled Burgers’ Equation
Fluid dynamics applications: The Coupled Burgers' Equation
Fluid dynamics applications:
The Coupled Burgers’ Equation
Fluid dynamics applications:
The 3D Boussinesq Equation

3D incompressible Boussinesq equations

\[ \rho \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \nabla \cdot \tau (\mathbf{v}) + \rho \mathbf{g} \]
\[ \nabla \cdot \mathbf{v} = 0 \]
\[ \rho c_p \left( \frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T \right) = \nabla (\kappa \nabla T), \]
\[ \tau (\mathbf{v}) = \rho \nu (\nabla \mathbf{v} + \nabla \mathbf{v}^T) \]

POD ROM with the following structure,

\[ q^{\text{pod}} = \mu \ D \ q^{\text{pod}} + \ C q^{\text{pod}} q^{\text{pod}T} + \ P q^{\text{pod}} q^{\text{pod}T}, \]
\[ v(x, t) = v_0(x) + \sum_{i=1}^{i=\text{N}_{\text{pod-v}}} \phi(x)_i^{\text{pod-v}} q_i^{\text{pod-v}}(t), \]
\[ T(x, t) = T_0(x) + \sum_{i=1}^{i=\text{N}_{\text{pod-T}}} \phi(x)_i^{\text{pod-T}} q_i^{\text{pod-T}}(t), \]
Fluid dynamics applications:

The 3D Boussinesq Equation

in the form

\[
\begin{align*}
\dot{q}^{\text{pod}}(t) &= F(q^{\text{pod}}(t), \mu) = \tilde{F}(q^{\text{pod}}(t)) + \mu Dq^{\text{pod}} \\
\phi^{\text{pod}}(t, x) &= \sum_{i=1}^{N_{\text{pod}}} \phi_i(x) q_i^{\text{pod}}(t),
\end{align*}
\]

\[
\tilde{F} = Cq^{\text{pod}} q^{\text{pod}}^T + P q^{\text{pod}} q^{\text{pod}}^T,
\]

where, \( C, P \) are kept separate to track the impact of different physical uncertainties on the ROM.

If we consider bounded parametric uncertainties on the coefficients of \( C \) and \( P \),

\[
\tilde{F} = (C + \Delta C)q^{\text{pod}} q^{\text{pod}}^T + (P + \Delta P)q^{\text{pod}} q^{\text{pod}}^T,
\]

where \( \| C + \Delta C \|_F \leq c_{\text{max}} \), and \( \| P + \Delta P \|_F \leq p_{\text{max}} \),

\[
\tilde{F} \leq c_{\text{max}} \| q^{\text{pod}} \|^2 + p_{\text{max}} \| q^{\text{pod}} \|^2
\]

This leads to the nonlinear closure model

\[
H_{nl} = \mu_{nl}(c_{\text{max}} \| q^{\text{pod}} \|^2 + p_{\text{max}} \| q^{\text{pod}} \|^2) \text{diag}(d_{11}, ..., d_{N_{\text{pod}}N_{\text{pod}}})q^{\text{pod}}, \quad \mu_{nl} > 0
\]

\[
\mu \rightarrow \mu_{cl} = \mu + \mu_e
\]

Learning cost

\[
Q(\mu) = \int_0^{t_f} \langle e_T, e_T \rangle_{\mathcal{H}} dt + \int_0^{t_f} \langle e_v, e_v \rangle_{(\mathcal{H})^3} dt.
\]

\[
\mu = (\mu_e, \mu_{nl})^T
\]
Numerical Results: The 3D Boussinesq Equation (Rayleigh-Benard modified problem)

Exact temperature at $t=0$

Exact temperature at $t=50$ sec

Temperature was specified at ±0.5 on the $x$-faces and taken as homogeneous Neumann on the remaining faces. $Re = 4.964 \times 10^4$, $Pr = 0.712$, and $Gr = 7.369 \times 10^7$

The simulation was run from zero velocity and temperature and snapshots were collected for 78 seconds. In this case we use 8PODs for the Galerkin ROM (ROM-G)
Numerical Results: The 3D Boussinesq Equation
Numerical Results: The 3D Boussinesq Equation (Rayleigh-Benard modified problem)

ROM-G velocity clip no learning at t=50 sec

ROM-G velocity clip with learning at t=50 sec

True velocity clip at t=50 sec
Numerical Results: The 3D Boussinesq Equation

Clip of the velocity error at $t=50$ sec.
ROM-G (no learning-8 PODs)

Clip of the velocity error at $t=50$ sec.
ROM-GL (with learning-8 PODs)
Numerical Results: The 2D Boussinesq Equation - Unsteady lock exchange flow problem

2D flow video
Numerical Results: The 2D Boussinesq Equation- Unsteady lock-exchange flow problem

Reconstruction error ROM-G (no learning)  Reconstruction error ROM-G (with learning)
Robotics examples: Rigid manipulators

We consider here a two-link robot manipulator:

\[ H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau, \]

where \( q \triangleq [q_1, q_2]^T \) denote the two joint angles and \( \tau \triangleq [\tau_1, \tau_2]^T \) denote the two joint inputs.

Now we assume uncertainties in the model

\[ \ddot{q} = H^{-1}(q)\tau - H^{-1}(q) [C(q, \dot{q})\dot{q} + G(q)] + \Delta b(q, t) \]

Robotics examples: Rigid manipulators

The reference trajectory

\[ q_{id}(t) = \frac{1}{1 + \exp(-t)}, \quad i = 1, 2 \]

The extremum seeking algorithm

\[ x_i(k + 1) = x_i(k) + t_f \alpha_i \sqrt{\omega_i} \cos(\omega_i t_f k) \]
\[ - t_f \kappa_i \sqrt{\omega_i} \sin(\omega_i t_f k) J, \quad i = 1, 2 \]
Robotics examples: Rigid manipulators

Case 1: \[ \Delta b = [\Delta b_1(t), \Delta b_2(t)]^T \]

\[ \begin{align*} 
\Delta b_1(t) &= 1 - 0.14 \sin(0.01 \ t), \\
\Delta b_2(t) &= 1 - 0.12 \cos(0.01 \ t). 
\end{align*} \]

the cost function

\[ J = Q_1 \int_{(N-1)t_f}^{Nt_f} (q - q_d)^T (q - q_d) \, dt \]
\[ + Q_2 \int_{(N-1)t_f}^{Nt_f} (\dot{q} - \dot{q}_d)^T (\dot{q} - \dot{q}_d) \, dt, \]

\[ Q_1 > 0, \ Q_2 > 0 \text{ and } N = 1, 2, \ldots \]
Robotics examples: Rigid manipulators

Case 1: $\Delta b = [\Delta b_1(t), \Delta b_2(t)]^T$
Robotics examples: Rigid manipulators

Case 1: $\Delta b = [\Delta b_1(t), \Delta b_2(t)]^T$
Robotics examples: Rigid manipulators

Case 2:

\[ \Delta b(q, t) = \Delta(t) \times (D\dot{q}) \]

\[ \Delta b_1(t) = -1 - 0.04 \sin(0.24 t), \]

\[ \Delta b_2(t) = -2 - 0.13 \sin(0.17 t). \]
Robotics examples: Rigid manipulators

Case 2: \[ \Delta b(q, t) = \Delta(t) \times (Dq) \]
Robotics applications: Maze mounted on a servo-motor*

Circular Maze Environment (CME)
- Tip and Tilt the maze so that the marble moves from the outer ring into the inner-most circle
- Intuitive to humans; most humans can solve very quickly
- Complex for RL agent due to constrained geometry, underactuated control, nonlinear dynamics, etc.

Robotics applications: Maze mounted on a servo-motor

Goal: obtain accurate model and exploit it for model-based RL

1. Collect real trajectories $X_{\text{real}} \sim f_{\text{real}}$ in the real system
2. Estimate physical parameters $\mu^*$ to obtain a more accurate physics engine
3. Learn residual model using Gaussian Process
4. Use the estimated model to control the real system with NMPC policy

Slide from: Vetro A.@MERL, keynote at IEEE Conference on Autonomous Systems (ICAS), 2021.
Robotics applications:
Maze mounted on a rigid arm

Experiments: Comparison with Human Performance

- 15 participants were asked to solve the maze by looking at the video feed of the marble movement.
- To familiarize them with the controls, they were given 1 minute to play with the maze using a joystick, but no marble.
- Then, they were asked to solve the maze five times.

- Can move the marble to goal within minutes of interaction.
- Consistently improve performance with larger amount of data.

Slide from: Vetro A.@MERL, keynote at IEEE Conference on Autonomous Systems (ICAS), 2021.
**Other applications**

**Batteries estimation**

**Gains auto-tuning for PV systems**

**Multi-robots source seeking and trajectory planning**

**RF power amplifiers auto- tunning and automated design**
- Kantana, C., Ma, R., Benosman, M., Komatsuzaki, Y., Yamanaka, K., A Hybrid Heuristic Search Control Assisted Optimization of Dual-Input Doherty Power Amplifier, European Microwave Conference 2021
- Cao, W., Benosman, M., Zhang, X., Ma, R., Domain Knowledge-Based Automated Analog Circuit Design with Deep Reinforcement Learning, AAAI Conference on Artificial Intelligence, February 2022 (nominated for best paper award).
What next?

Sentient meat by Terry Bisson’s: https://www.wnycstudios.org/podcasts/studio/segments/168264-theyre-made-out-of-meat

Learning paradigms inspired from:
- Cognitive psychology (mind)
- Neuro-science and brain physiology (brain)

e.g., See the course ‘Brains, minds and machines’ summer course: https://ocw.mit.edu/courses/res-9-003-brains-minds-and-machines-summer-course-summer-2015/pages/syllabus/course-instructors-guest-speakers-and-icub-team/

General AI ?!