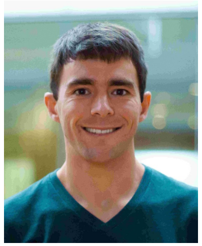
The Discrete-Time Internal Model Principle of Time-Varying Optimization

IEEE Conf. on Decision and Control – Rio de Janeiro, Brazil – Dec. 11, 2025

Gianluca Bianchin

UCLouvain

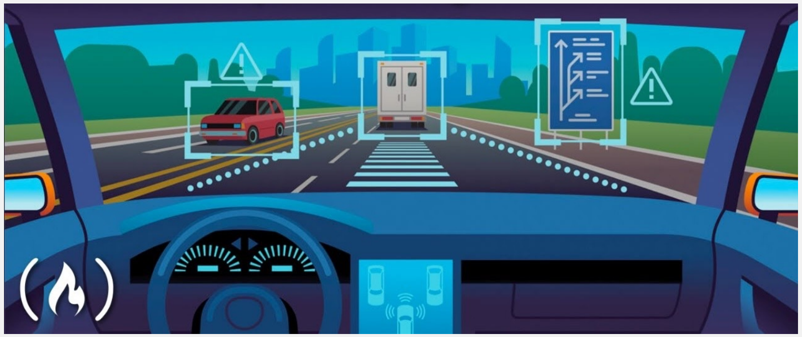
Bryan Van Scoy



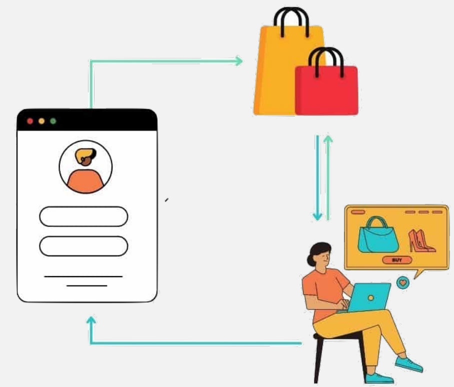
MIAMI UNIVERSITY

Motivation: making online decisions at sampled time instants

Online learning (eg perception)



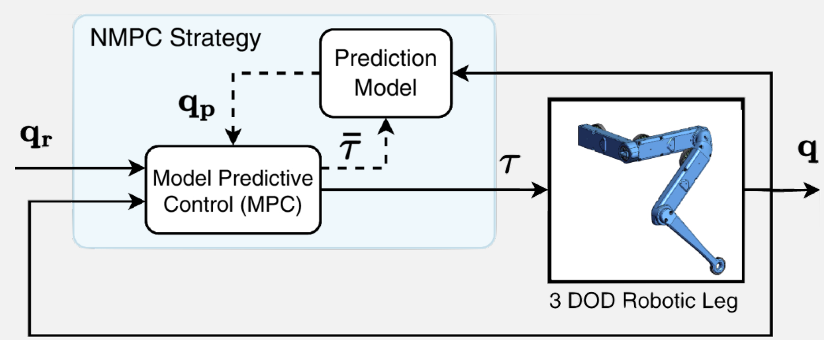
Online recommendation systems



$$\min_{x \in \mathbb{R}^n} f(x, \theta_k), \quad k \in \mathbb{Z}_{\geq 0}$$

Online ad placement

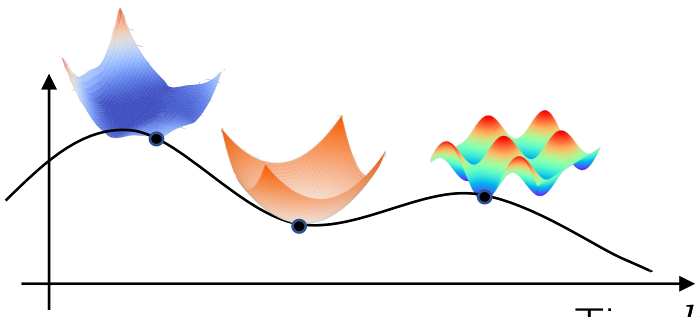
Online MPC



Objective: solve time-varying optimization

Objective:

$$\min_{x \in \mathbb{R}^n} f(x, \theta_k), \quad k \in \mathbb{Z}_{\geq 0}$$



- $\theta_k \in \Theta \subseteq \mathbb{R}^p$ $\rightarrow$ time-varying parameter vector
- $f(x, \theta) \rightarrow$ loss to be minimized

Assumption (Properties of the optimization):

- $x \mapsto f(x, \theta)$ is convex
- $x \mapsto \nabla_x f(x, \theta)$ is Lipschitz

Assumption (Properties of the temporal variability):

- There exists smooth $s : \Theta \rightarrow \Theta$ such that $\theta_{k+1} = s(\theta_k)$
- $\theta = 0$ is a locally stable equilibrium

Definition:

A **critical trajectory** is $x_k^\star$, given by: $0 = \nabla_x f(x_k^\star, \theta_k), \quad \forall k$

We search for algorithms within a class

We focus on methods that have access to first-order oracles:

$$y_k = \nabla_x f(x_k, \theta_k)$$

and consider the class of algorithms:

$$z_{k+1} = F_c(z_k, y_k)$$
$$x_k = G_c(z_k)$$

Examples:

- Online Gradient descent [Hazan 2016]: $F_c(z, y) = z - \alpha y, \quad G_c(z) = z$
- Prediction-correction [Simonetto 2017]

Problem 1 (Algorithm design):

Design $F_c(z, y)$ and $G_c(z)$ such that $x_k \to x_k^\star$

Problem 0 (Existence and required knowledge):

- When does an algorithm exist?
- What is the "minimal knowledge" required to design the algorithm?

Related work: developed under a variety of assumptions

Framework:

Continuous-time:

$$\min_x f(x, \theta(t)), \quad t \in \mathbb{R}_{\geq 0}$$

[Hall 2015, , Fazylab 2017, Bianchin 2024, ...]

Sampled-time:

$$\min_x f(x, \theta(t_k)), \quad k \in \mathbb{Z}_{\geq 0}$$

[Simonetto 2017, Simonetto 2020,...]

Discrete-time (this work):

$$\min_{x \in \mathbb{R}^n} f(x, \theta_k), \quad k \in \mathbb{Z}_{\geq 0}$$

[Hazan 2016, Bastianello 2024, Casti 2023, ...]

Properties of the temporal-variability:

(E1) $\theta(t)$ is measurable

[Hazan 2016, Hall 2015, ...]

(E2) $s(\theta)$ is known

[Bastianello 2024]

(E3) $\theta(t)$ and $s(\theta)$ are known

[Zhao 1998, Fazylab 2017, Raveendran 2022, ...]

....

Properties of the optimization:

(O1) Oracle gradient evaluations:

$$(t, x) \mapsto \nabla_x f(x, \theta(t))$$

[Hazan 2016, Hall 2015, ...]

(O2) Gradient $\nabla_x f(x, \theta)$

(O3) Sensitivity $\nabla_{x\theta} f(x, \theta)$

(O4) Hessian $\nabla^2_{xx} f(x, \theta)$

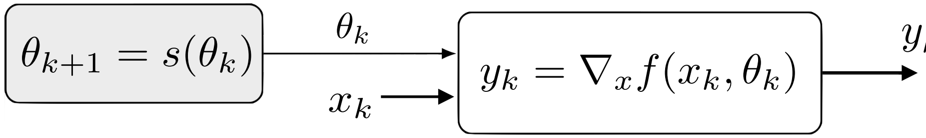
[Zhao 1998, Fazylab 2017, Raveendran 2022, ...]

(O5) Loss is quadratic

[Bastianello 2024, ...]

...

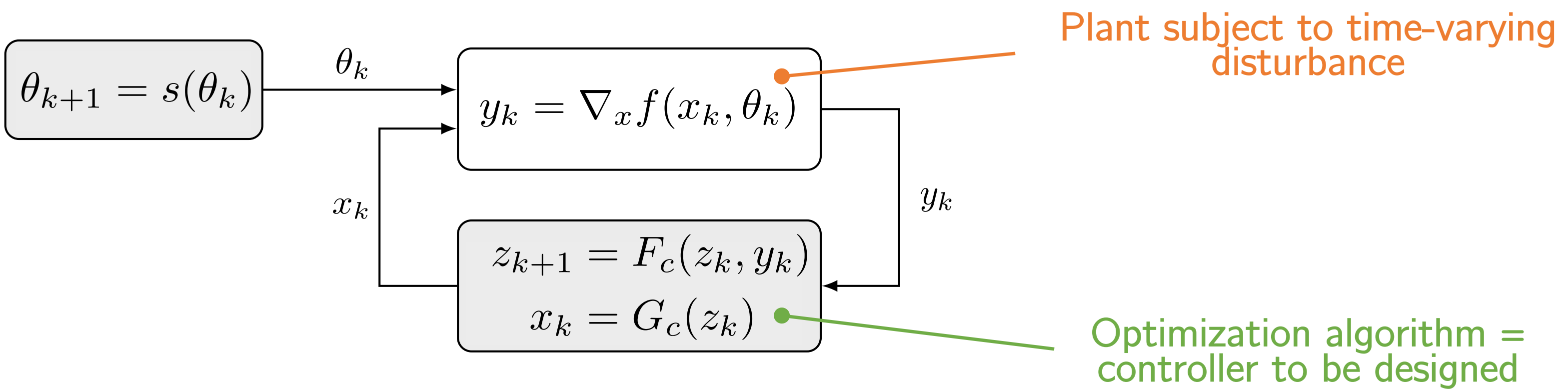
Time-varying optimization = output regulation problem



exact tracking means:

$y_k \to 0$

Time-varying optimization = output regulation problem



The interconnection has the form of a **nonlinear autonomous system:**

$$
\begin{aligned}
z_{k+1} &= F_c(z_k, y_k) \\
\theta_{k+1} &= s(\theta_k) \\
y_k &= \nabla_x f(G_c(z_k), \theta_k)
\end{aligned}
$$

Definition:

The algorithm **exactly asymptotically tracks a critical trajectory** if there exists $\mathcal{Z}_\circ \times \Theta_\circ$ such that, for all initializations in this set, the closed-loop satisfies $y_k \to 0$ as $k \to \infty$

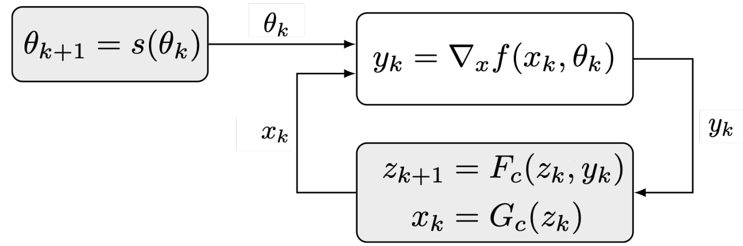
Assumption (Local detectability):

θ_k is locally exponentially detectable from y_k: $\quad \|\hat{\theta}_k - \theta_k\| \leq M a^k \|\hat{\theta}_0 - \theta_0\|$

Lack of detectability = certain modes do not affect the gradient, thus can be removed

The internal model principle of optimization

Recall (regulation framework):



Main theorem:

Suppose $z_\circ^\star$ is a locally exponentially stable equilibrium when $\theta \equiv 0$.

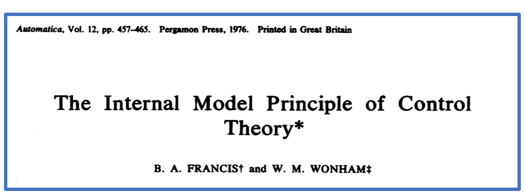
Then, the algorithm achieves exact asymptotic tracking if and only if there exists a C^2 mapping $z = \sigma(\theta)$ (with $\sigma(0) = z_\circ^\star$) such that

$$\sigma(s(\theta_\omega)) = F_c(\sigma(\theta_\omega), 0)$$
$$0 = \nabla_x f(G_c(\sigma(\theta_\omega)), \theta_\omega)$$

hold at all limit points $\theta_\omega \in \Omega(\Theta_\circ)$

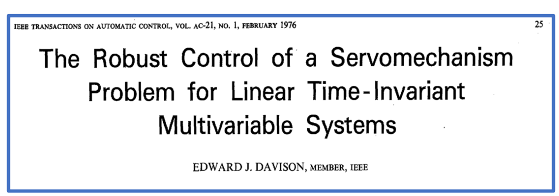
In words:

θ_k and z_k must be related, at all limit points of θ_k, by a change of variables: $z_k = \sigma(\theta_k)$


Automatica, Vol. 12, pp. 457–465. Pergamon Press, 1976. Printed in Great Britain

The Internal Model Principle of Control Theory*

B. A. FRANCIS† and W. M. WONHAM‡


IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. AC-21, NO. 1, FEBRUARY 1976 25

The Robust Control of a Servomechanism Problem for Linear Time-Invariant Multivariable Systems

EDWARD J. DAVISON, MEMBER, IEEE

The "internal model principle"

an algorithm can track time-varying optimizers if and only if it incorporates a model of the temporal variability

Answers to the questions we posed

Problem 0 (Existence and required knowledge):

- When does an algorithm exist?
- What is the "minimal knowledge" required to design the algorithm?

Answer to Problem 0:

- An algorithm exists if and only there exists a function $H_c(\theta)$ s.t. $0 = \nabla_x f(H_c(\theta_\omega), \theta_\omega)$.
- The minimal knowledge required for design are the functions involved in

$$\begin{aligned} \sigma(s(\theta_\omega)) &= F_c(\sigma(\theta_\omega), 0) \\ 0 &= \nabla_x f(G_c(\sigma(\theta_\omega)), \theta_\omega) \end{aligned}$$

Problem 1 (Algorithm design):

Design $F_c(z, y)$ and $G_c(z)$ such that $x_k \to x_k^\star$

Answer to Problem 1:

$\to$ Up next!

Optimization algorithm design

Recall (IMP conditions):

$$\sigma(s(\theta_\omega)) = F_c(\sigma(\theta_\omega), 0)$$
$$0 = \nabla_x f(G_c(\sigma(\theta_\omega)), \theta_\omega)$$

Optimization algorith design procedure:

1. Select $\sigma(\theta) = Id$ (identity operator). The IMP conditions simplify to:

$$s(\theta_\omega) = F_c(\theta_\omega, 0)$$
$$0 = \nabla_x f(G_c(\theta_\omega), \theta_\omega)$$

2. Select function $G_c(\theta)$ such that: $0 = \nabla_x f(G_c(\theta_\omega), \theta_\omega)$

3. Select function $F_c(z, \theta)$ as: $F_c(z, y) = s(z) + L(y - \nabla_x f(G_c(z), z))$

where L is chosen s.t. the linerization around $\theta \equiv 0$ is exponentially stable

The optimization algorithm acts as an exponential observer for θ_k

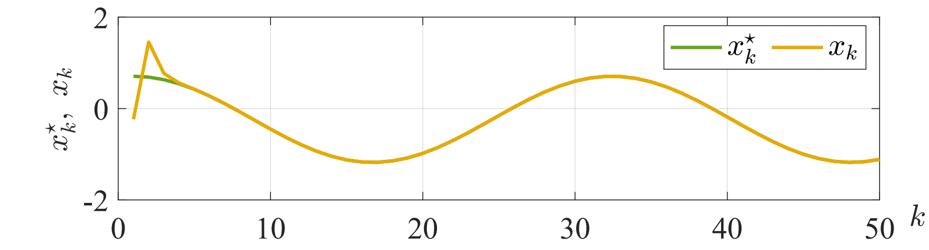
Numerical simulations support the approach

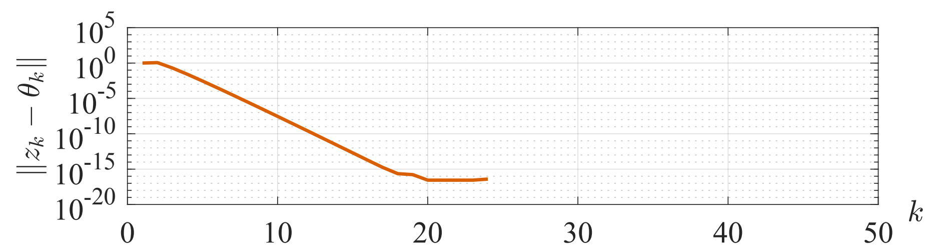
Optimization:

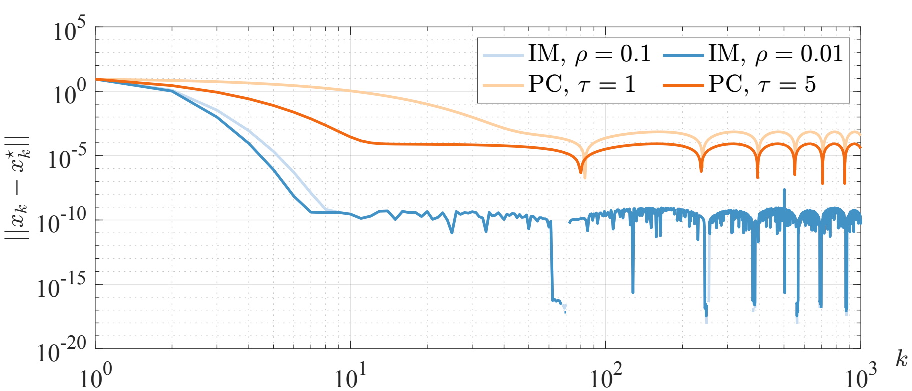
$$\min_{x \in \mathbb{R}} f(x, \theta_k) := \frac{1}{2}\left(x - \theta_k^{(1)}\right)^2 + \kappa \log(1 + e^{\mu x})$$

Temporal variability:

$$\begin{bmatrix} \theta_{k+1}^{(1)} \\ \theta_{k+1}^{(2)} \end{bmatrix} = \begin{bmatrix} 0.9801 & 0.9933 \\ -0.03977 & 0.9801 \end{bmatrix} \begin{bmatrix} \theta_k^{(1)} \\ \theta_k^{(2)} \end{bmatrix}$$







PC: [Simonetto 2017]

Comparison with literature:

Numerics suggest that, for this problem

- Our approach outperforms PC in both in convergence rate and in asymptotic precision
- Rate of convergence can be tuned arbitrarily by tuning the observer's eigenvalues

Conclusions

- Time-varying optimization design = output regulation problem

- **Fundamental result:** exact tracking is possible only if the algorithm contains a reduplicated copy of the temporal variability

- We proposed a design procedure: algorithm acts as exponential observer of the time variability

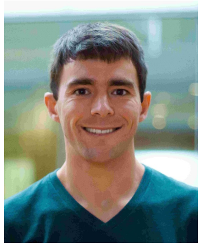
- Future directions: alternative algorithm designs, constrained optimization, devising algorithms with global convergence guarantees, consider rates of convergence, ...

Thank you

Gianluca Bianchin

UCLouvain

Bryan Van Scoy



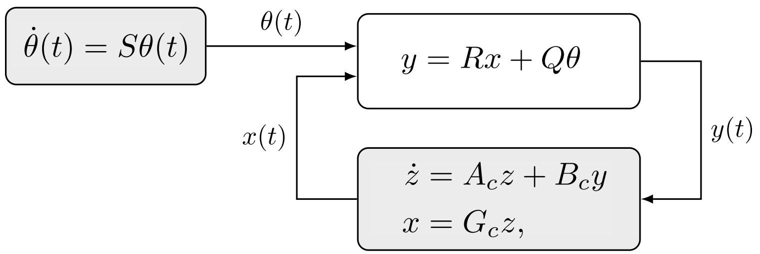
MIAMI UNIVERSITY

Backup slides

Example

Optimization: $\min_{x \in \mathbb{R}^n} f(x(t), \theta(t)) = \frac{1}{2} x(t)^\mathsf{T} R x(t) + x(t)^\mathsf{T} Q \theta(t)$

Temporal variability: $\theta_{k+1} = S\theta_k$



Mapping zeroing the gradient: $G_c(\theta) = -R^\dagger Q \cdot \theta$

IMP conditions: $0 = (\Sigma S - A_c \Sigma)\theta \quad 0 = (R G_c \Sigma + Q)\theta$

Optimization algorithm: $F_c(z, y) = Sz + Ly, \quad G_c(\theta) = -R^\dagger Q \cdot \theta$