KU LEUVEN

FLANDERS MAKE

DRIVING INNOVATION IN MANUFACTURING

MECO RESEARCH TEAM

A BRIEF HISTORY OF OPTIMIZATION-BASED ILC. WHAT'S NEXT?

17/03/2022 @ Syscop, University of Freiburg

Daniele Ronzani

MECO Research Team, Dept. of Mechanical Engineering, KU Leuven, Belgium

Core Lab DMMS-M, Decision & Control, Flanders Make, Leuven, Belgium

Outline

Introduction

Opt-ILC: a brief history

- Filter based an Model-inversion ILC
- Norm-Optimal ILC
- Optimization based ILC
- RoFaLT

Applications and developments

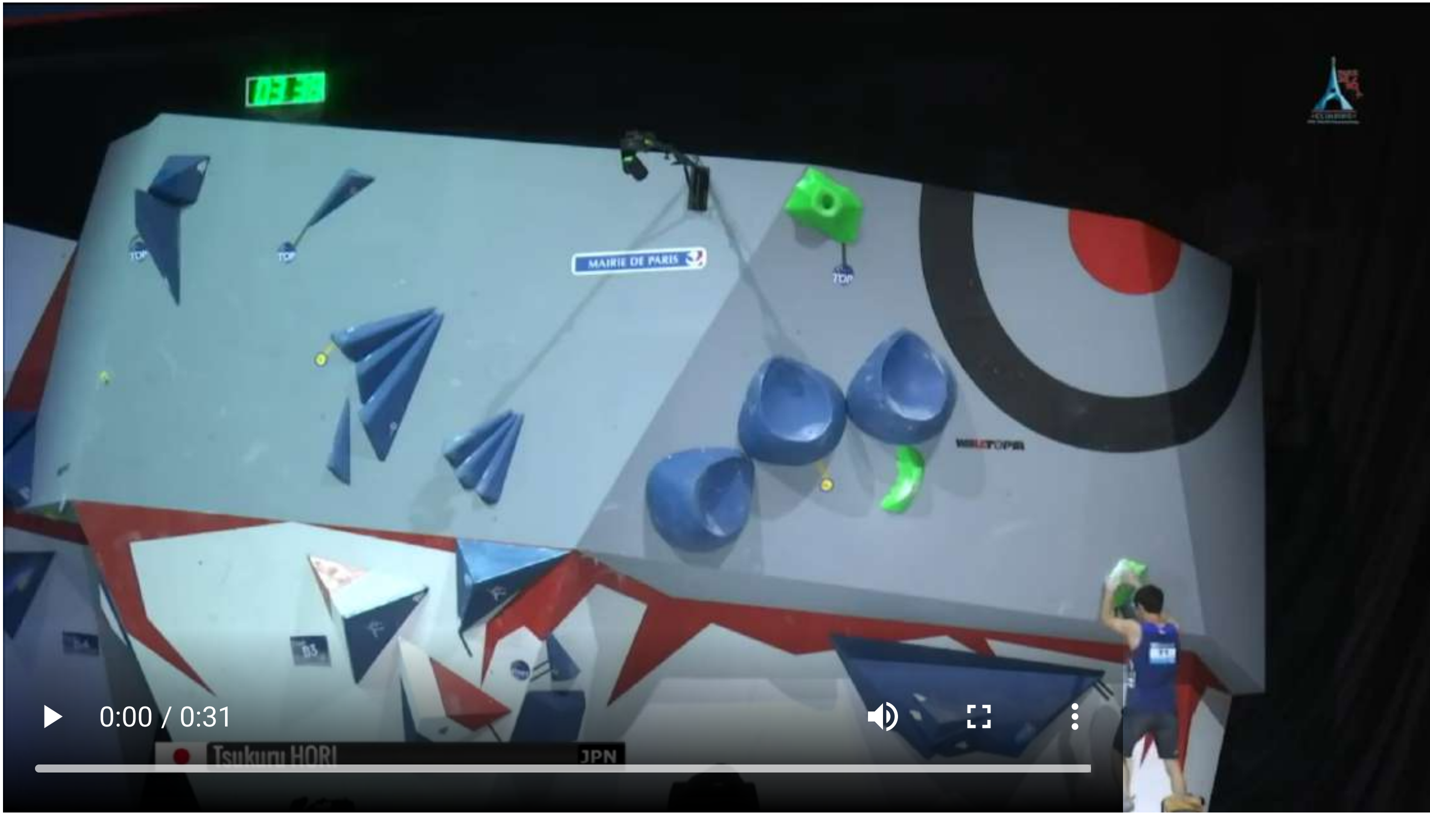
- Applications
- Multi-system ILC
- ECOset-ILC

Conlcusion

Outline

Introduction

What is ILC?



INTRODUCTION

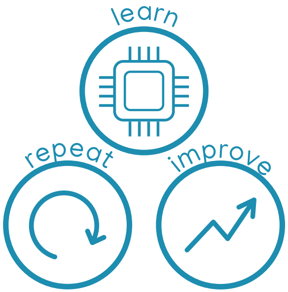
What is ILC?

- learn from previous trials and improve
- offline strategy
- Feed-foward/open-loop
- Plug-in to existent control system

Introduction

What is ILC?

- learn from previous trials and improve
- offline strategy
- Feed-foward/open-loop
- Plug-in to existent control system



INTRODUCTION

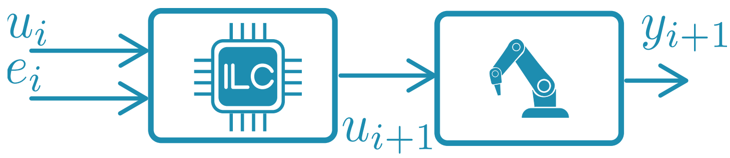
What is ILC?

- learn from previous trials and improve
- offline strategy
- Feed-foward/open-loop
- Plug-in to existent control system

INTRODUCTION

What is ILC?

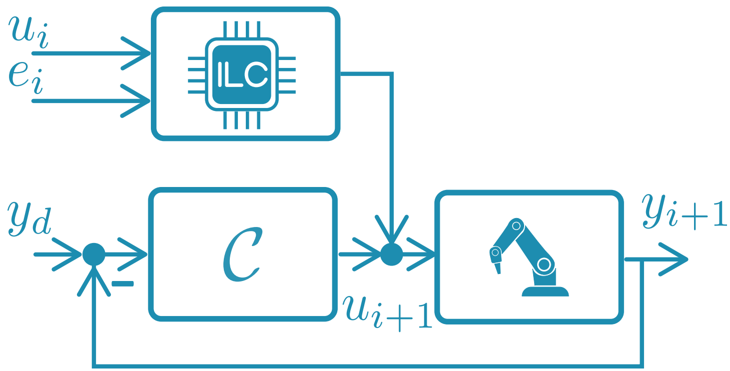
- learn from previous trials and improve
- offline strategy
- Feed-foward/open-loop
- Plug-in to existent control system



INTRODUCTION

What is ILC?

- learn from previous trials and improve
- offline strategy
- Feed-foward/open-loop
- Plug-in to existent control system



INTRODUCTION

When should I consider using ILC?

- the task is repetitive
- same initial conditions
- compensate repetitive errors
  - model-plant mismatch
  - exogenous disturbances

INTRODUCTION

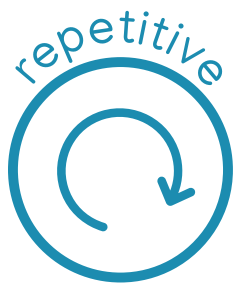
When should I consider using ILC?

- the task is repetitive
- same initial conditions
- compensate repetitive errors
  - model-plant mismatch
  - exogenous disturbances

INTRODUCTION

When should I consider using ILC?

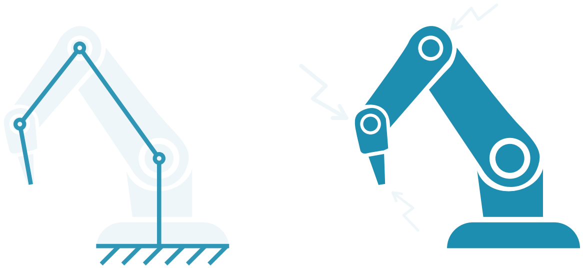
- the task is repetitive
- same initial conditions
- compensate repetitive errors
    - model-plant mismatch
    - exogenous disturbances



Introduction

When should I consider using ILC?

- the task is repetitive
- same initial conditions
- compensate repetitive errors
  - model-plant mismatch
  - exogenous disturbances



OUTLINE

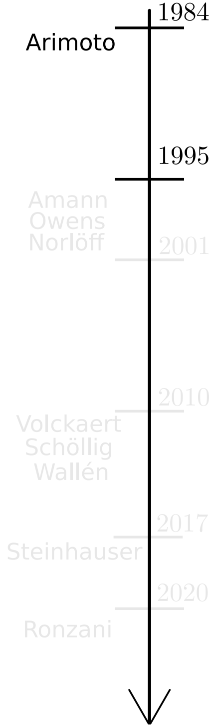
Outline

Opt-ILC: a brief history

From $\mathcal{LQ}$ -filter to norm-optimal ILC

How to get fast convergence and high performance?

Model-inversion



OPT-ILC: A BRIEF HISTORY

From $\mathcal{L}\mathcal{Q}$ -filter to norm-optimal ILC

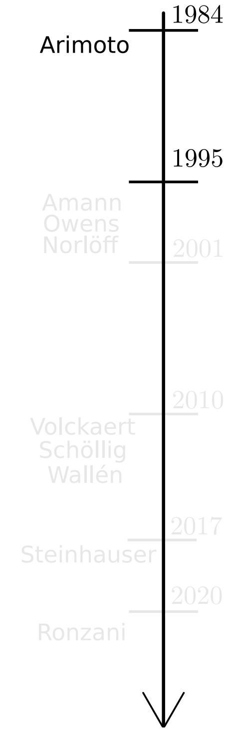
$$u_{i+1} = \mathcal{Q}\big(u_i + \mathcal{L}\,e_i\big)$$

$$e_i = y_d - y_i$$

How to get fast convergence and high performance?

⇩

Model-inversion



OPT-ILC: A BRIEF HISTORY

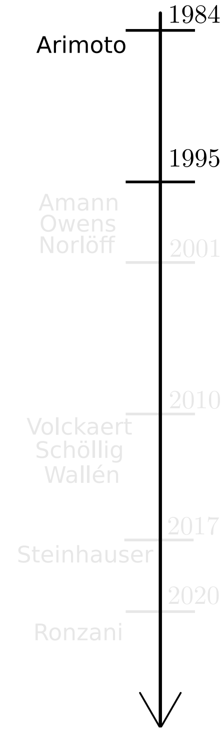
From $\mathcal{L}\mathcal{Q}$ -filter to norm-optimal ILC

$$u_{i+1} = \mathcal{Q}\big(u_i + \mathcal{L}\, e_i\big)$$

robustness learning filter

How to get fast convergence and high performance?

Model-inversion



OPT-ILC: A BRIEF HISTORY

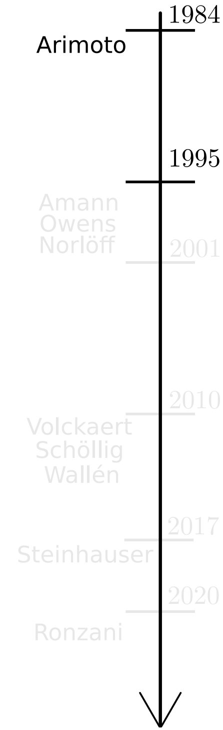
From $\mathcal{L}\mathcal{Q}$ -filter to norm-optimal ILC

$$u_{i+1} = \mathcal{Q}\big(u_i + \mathcal{L}\, e_i\big)$$

$$e_i = y_d - y_i$$

How to get fast convergence and high performance?

Model-inversion



OPT-ILC: A BRIEF HISTORY

From $\mathcal{L}\mathcal{Q}$ -filter to norm-optimal ILC

$$u_{i+1} = \mathcal{Q}\big(u_i + \mathcal{L}\, e_i\big)$$

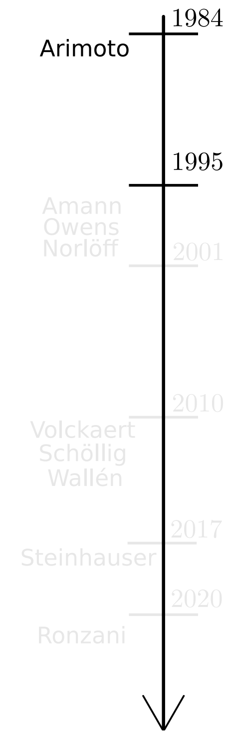
$$e_i = y_d - y_i$$

How to get fast convergence and high performance?

⇩

Model-inversion

$$\mathcal{Q} = I \qquad \mathcal{L} = P^{-1}$$



OPT-ILC: A BRIEF HISTORY

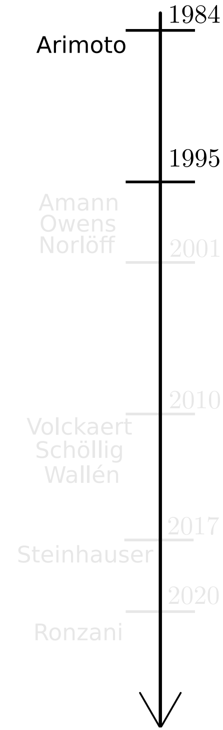
From $\mathcal{LQ}$ -filter to norm-optimal ILC

$$u_0 = 0$$

$$y_0 = 0$$

How to get fast convergence and high performance?

Model-inversion



OPT-ILC: A BRIEF HISTORY

From $\mathcal{LQ}$ -filter to norm-optimal ILC

$$u_1 = I\left(0 \ + P^{-1} y_d\right)$$

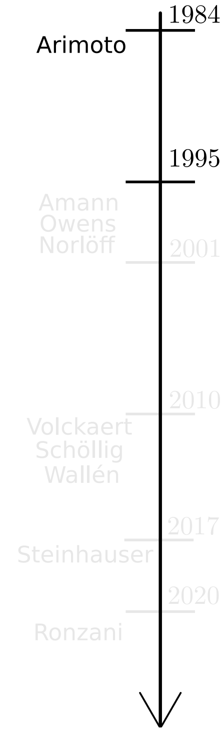
$$e_0 = y_d - 0$$

How to get fast convergence and high performance?

⇩

Model-inversion

$$\mathcal{Q} = I \qquad\qquad \mathcal{L} = P^{-1}$$



Opt-ILC: a brief history

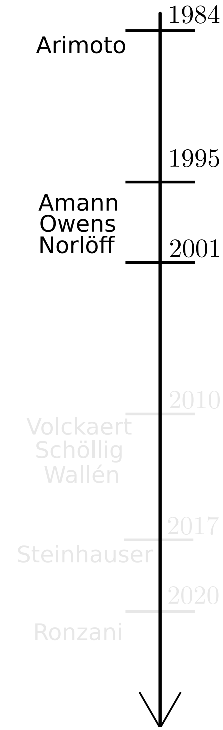
From $\mathcal{L}\mathcal{Q}$ -filter to norm-optimal ILC

- Optimal control approach
- In practice:
- Equivalence in the filter design framework:

$$\mathcal{Q} = \left(\hat{P}^T Q \hat{P} + R_1 + R_2\right)^{-1} \left(\hat{P}^T Q \hat{P} + R_2\right)$$

$$\mathcal{L} = \left(\hat{P}^T Q \hat{P} + R_2\right)^{-1} \hat{P}^T Q$$

Can this be generalized even further?



Opt-ILC: a brief history

From $\mathcal{LQ}$ -filter to norm-optimal ILC

- Optimal control approach

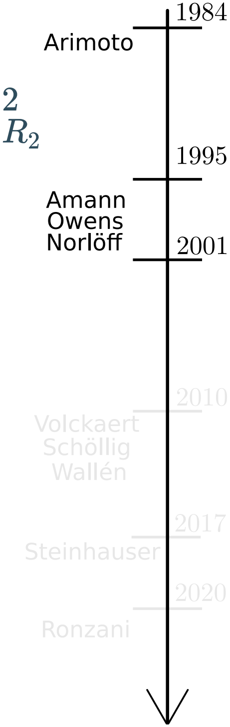
$$u_{i+1} = \arg\min_u \|y_d - Pu\|_Q^2 + \|u\|_{R_1}^2 + \|u - u_i\|_{R_2}^2$$

- In practice:

- Equivalence in the filter design framework:

$$\mathcal{Q} = \left(\hat{P}^T Q \hat{P} + R_1 + R_2\right)^{-1} \left(\hat{P}^T Q \hat{P} + R_2\right)$$
$$\mathcal{L} = \left(\hat{P}^T Q \hat{P} + R_2\right)^{-1} \hat{P}^T Q$$

Can this be generalized even further?



Opt-ILC: a brief history

From $\mathcal{LQ}$ -filter to norm-optimal ILC

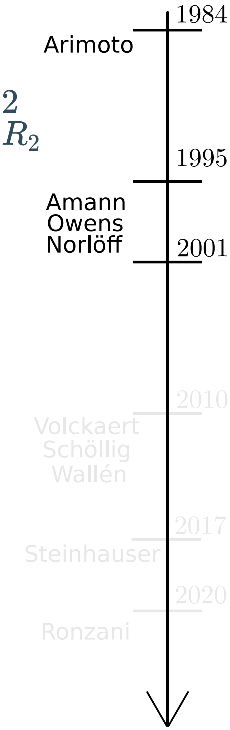
- Optimal control approach

$$u_{i+1} = \arg\min_{u} \|y_d - Pu\|_Q^2 + \|u\|_{R_1}^2 + \|u - u_i\|_{R_2}^2$$

- In practice: $Pu \approx \hat{P}u + (y_i - \hat{P}u_i)$
- Equivalence in the filter design framework:

$$\mathcal{Q} = \left(\hat{P}^T Q \hat{P} + R_1 + R_2\right)^{-1}\left(\hat{P}^T Q \hat{P} + R_2\right)$$
$$\mathcal{L} = \left(\hat{P}^T Q \hat{P} + R_2\right)^{-1} \hat{P}^T Q$$

Can this be generalized even further?



Opt-ILC: a brief history

From $\mathcal{L}\mathcal{Q}$ -filter to norm-optimal ILC

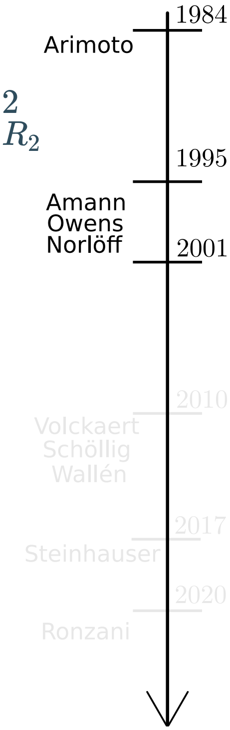
- Optimal control approach

$$u_{i+1} = \arg\min_{u} \|y_d - Pu\|_Q^2 + \|u\|_{R_1}^2 + \|u - u_i\|_{R_2}^2$$

- In practice: $Pu \approx \hat{P}u + \left(y_i - \hat{P}u_i\right)$
- Equivalence in the filter design framework:

$$\mathcal{Q} = \left(\hat{P}^T Q \hat{P} + R_1 + R_2\right)^{-1} \left(\hat{P}^T Q \hat{P} + R_2\right)$$
$$\mathcal{L} = \left(\hat{P}^T Q \hat{P} + R_2\right)^{-1} \hat{P}^T Q$$

Can this be generalized even further?



OPT-ILC: A BRIEF HISTORY

From $\mathcal{L}\mathcal{Q}$ -filter to norm-optimal ILC

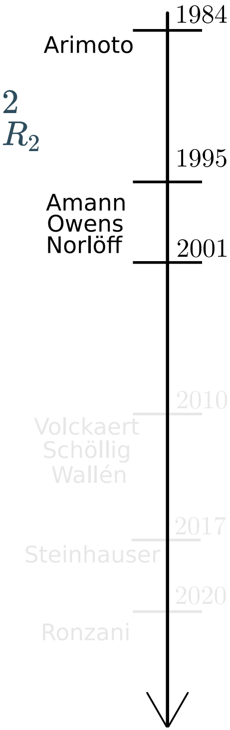
- Optimal control approach

$$u_{i+1} = \arg\min_{u} \|y_d - Pu\|_Q^2 + \|u\|_{R_1}^2 + \|u - u_i\|_{R_2}^2$$

- In practice: $Pu \approx \hat{P}u + (y_i - \hat{P}u_i)$
- Equivalence in the filter design framework:

$$\mathcal{Q} = (\hat{P}^T Q \hat{P} + R_1 + R_2)^{-1} (\hat{P}^T Q \hat{P} + R_2)$$
$$\mathcal{L} = (\hat{P}^T Q \hat{P} + R_2)^{-1} \hat{P}^T Q$$

Can this be generalized even further?



Opt-ILC: a brief history

From $\mathcal{L}\mathcal{Q}$ -filter to norm-optimal ILC

- Optimal control approach

ocp

$$u_{i+1} = \arg\min_{u} \|y_d - Pu\|_Q^2 + \|u\|_{R_1}^2 + \|u - u_i\|_{R_2}^2$$

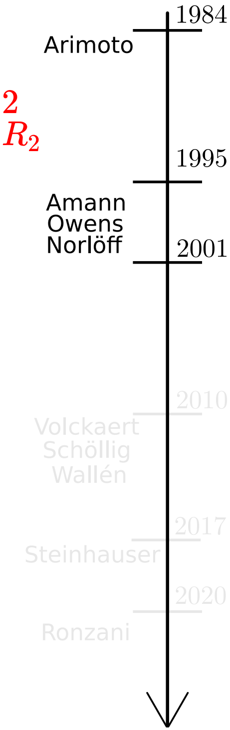
- In practice: $Pu \approx \hat{P}u + (y_i - \hat{P}u_i)$ model correction

- Equivalence in the filter design framework:

$$\mathcal{Q} = \left(\hat{P}^T Q \hat{P} + R_1 + R_2\right)^{-1} \left(\hat{P}^T Q \hat{P} + R_2\right)$$

$$\mathcal{L} = \left(\hat{P}^T Q \hat{P} + R_2\right)^{-1} \hat{P}^T Q$$

Can this be generalized even further?



OPT-ILC: A BRIEF HISTORY

Optimization-based ILC

- Model correction

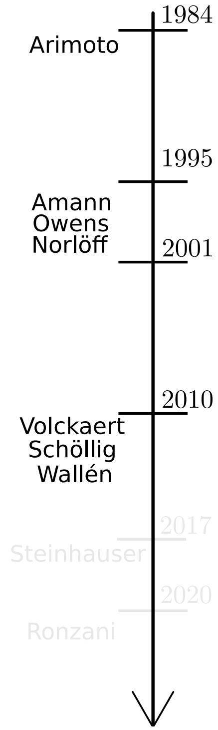
$$\beta_i = \arg\min_{\beta} \|y_i - \hat{\mathcal{P}}(u, \beta)\|_V^2 + \|\beta\|_{W_1}^2 + \|\beta - \beta_{i-1}\|_{W_2}^2$$

  - Observer-based/Kalman filter estimation
  - Disturbances/parameter estimation

- Model inversion

$$u_{i+1} = \arg\min_{u} \|y_d - \hat{\mathcal{P}}(u, \beta_i)\|_Q^2 + \|u\|_{R_1}^2 + \|u - u_i\|_{R_2}^2$$

  - other type of tasks can be considered, e.g. time-optimal motion
  - Constraints can be easily added, e.g input saturations



OPT-ILC: A BRIEF HISTORY

Optimization-based ILC

- Model correction

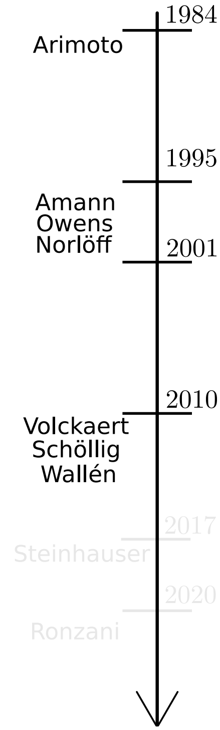
$$\beta_i = \arg\min_{\beta} \|y_i - \hat{\mathcal{P}}(u,\beta)\|_V^2 + \|\beta\|_{W_1}^2 + \|\beta - \beta_{i-1}\|_{W_2}^2$$

  - Observer-based/Kalman filter estimation
  - Disturbances/parameter estimation

- Model inversion

$$u_{i+1} = \arg\min_{u} \|y_d - \hat{\mathcal{P}}(u,\beta_i)\|_Q^2 + \|u\|_{R_1}^2 + \|u - u_i\|_{R_2}^2$$

  - other type of tasks can be considered, e.g. time-optimal motion
  - Constraints can be easily added, e.g input saturations



Opt-ILC: A brief history

Optimization-based ILC

- Model correction

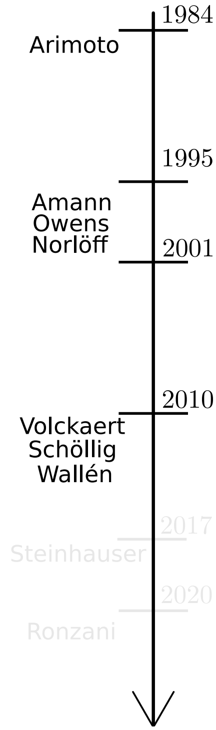
$$\beta_i = \arg\min_{\beta} \|y_i - \hat{\mathcal{P}}(u,\beta)\|_V^2 + \|\beta\|_{W_1}^2 + \|\beta - \beta_{i-1}\|_{W_2}^2$$

  - Observer-based/Kalman filter estimation
  - Disturbances/parameter estimation

- Model inversion

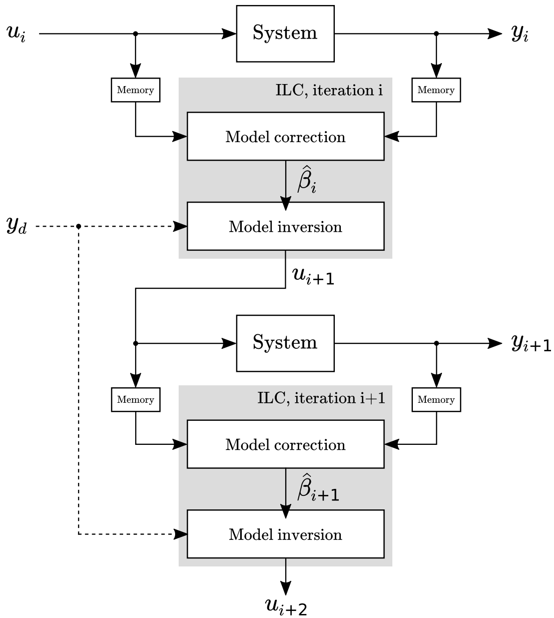
$$u_{i+1} = \arg\min_{u} \|y_d - \hat{\mathcal{P}}(u,\beta_i)\|_Q^2 + \|u\|_{R_1}^2 + \|u - u_i\|_{R_2}^2$$

  - other type of tasks can be considered, e.g. time-optimal motion
  - Constraints can be easily added, e.g input saturations

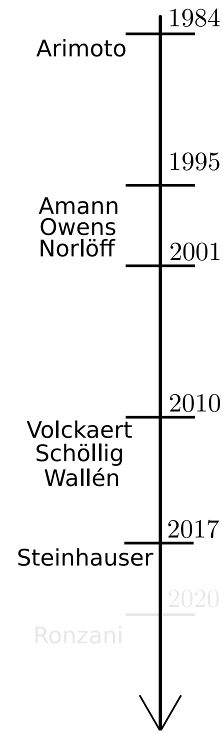


OPT-ILC: A BRIEF HISTORY

RoFaLT: two-step nonlinear ILC




```
myRoFaLT = rofalt.RoFaLT();

doc rofalt.RoFaLT

model = myRoFaLT.setModel('ts', 0.01, ...
                          'ode', odeFcn, ...
                          'output', yFcn);

ilc = myRoFaLT.setController('ilc');

ilc.setReference(ref);

ilc.control.addInequalityConstraints(usym, ...
                                     umin, umax)

for i = 1:1:nTrials

  umeas = ilc.runIteration(struct('u', umeas, ...
                                  'y', ymeas));

  ymeas = executeMeasurements(umeas);
end

results = ilc.getResults();
```



Outline

Outline

Introduction

Opt-ILC: a brief history

- Filter based an Model-inversion ILC
- Norm-Optimal ILC
- Optimization based ILC
- RoFaLT

Applications and developments

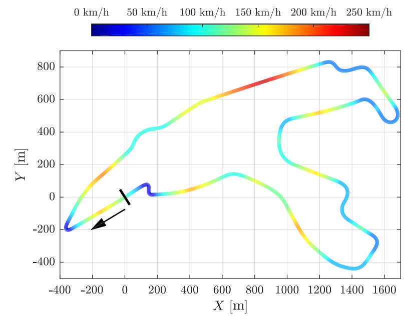
- Applications
- Multi-system ILC
- ECOset-ILC

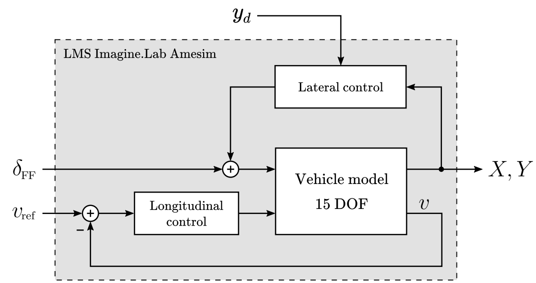
Conlcusion

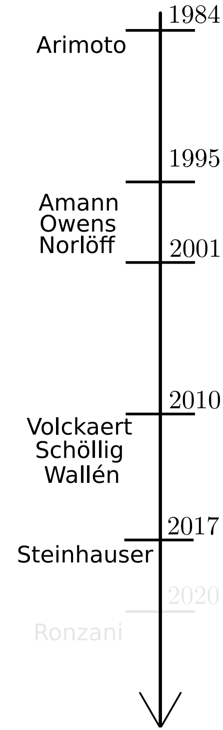
APPLICATIONS

Car-race simulation

- Model correction
  - Kinematic bicycle model
  - Input correction
- Model inversion
  - Trajectory tracking
  - Constraints: input saturation



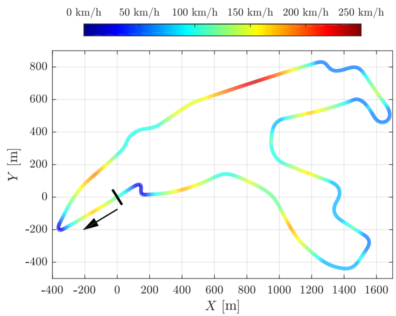


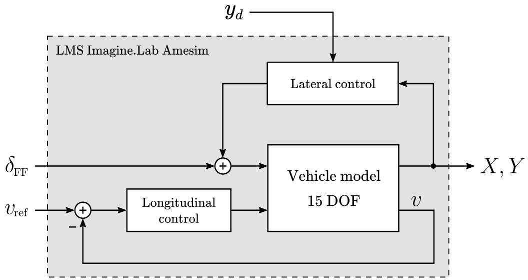


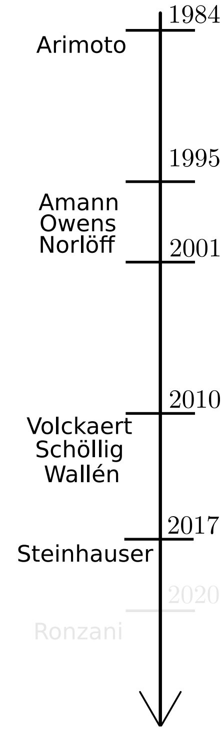
APPLICATIONS

Car-race simulation

- Model correction
  - Kinematic bicycle model
  - Input correction
- Model inversion
  - Trajectory tracking
  - Constraints: input saturation



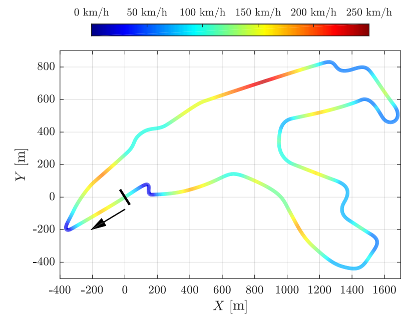


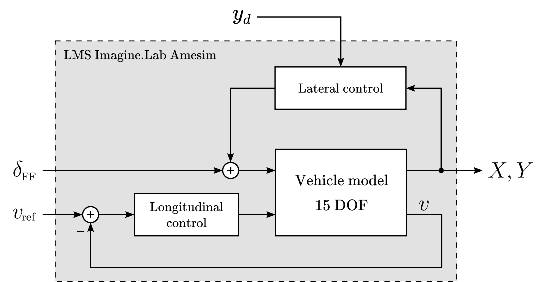


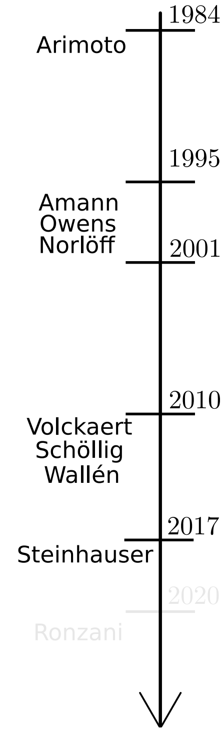
APPLICATIONS

Car-race simulation

- Model correction
  - Kinematic bicycle model
  - Input correction
- Model inversion
  - Trajectory tracking
  - Constraints: input saturation



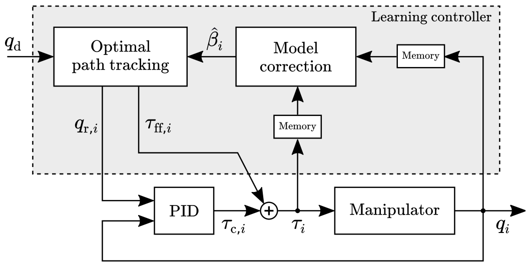


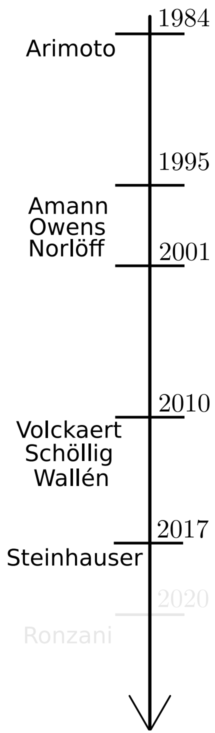


Applications

Time-optimal path tracking

- Model correction
  - Manipulator inverse dynamics
  - Torque input correction
- Model inversion
  - Time-optimal
  - Path-tracking

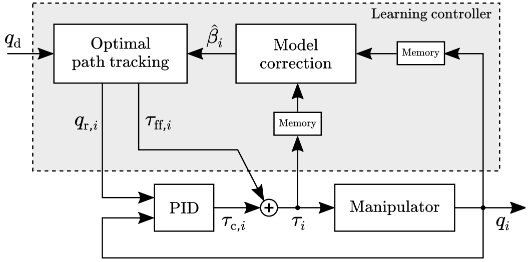


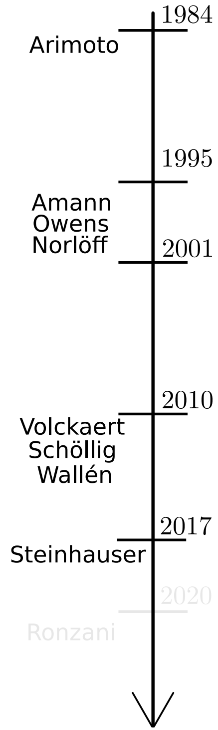


APPLICATIONS

Time-optimal path tracking

- Model correction
  - Manipulator inverse dynamics
  - Torque input correction
- Model inversion
  - Time-optimal
  - Path-tracking

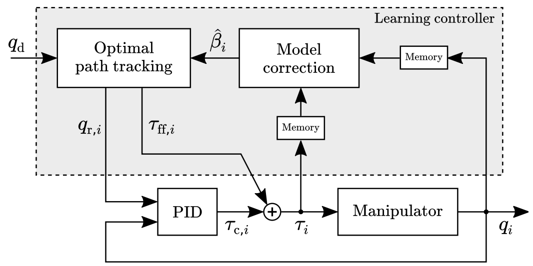


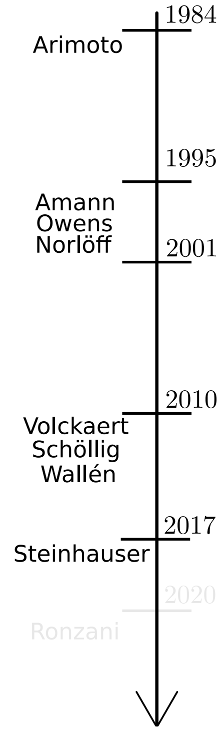


APPLICATIONS

Time-optimal path tracking

- Model correction
  - Manipulator inverse dynamics
  - Torque input correction
- Model inversion
  - Time-optimal
  - Path-tracking





DEVELOPMENTS

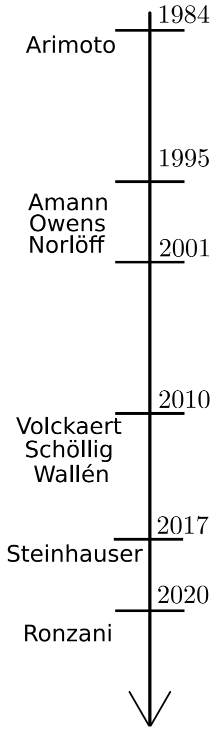
- Fleet of mechatronic systems
- Repetitive task
- Iterative Learning Control

+

- Industry 4.0: interconnected systems

⇩

- Multi-System Learning Control



DEVELOPMENTS

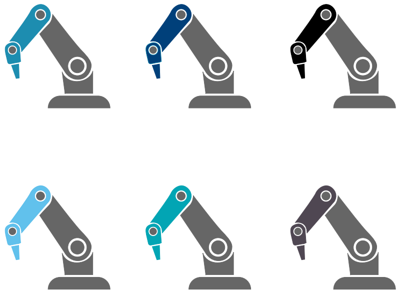
- Fleet of mechatronic systems
- Repetitive task
- Iterative Learning Control

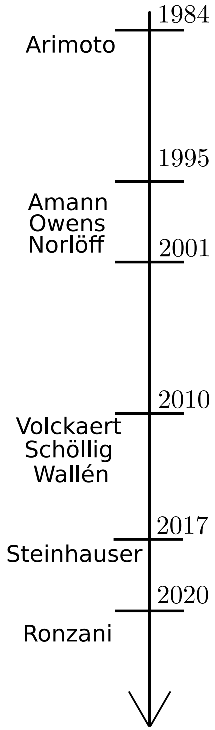
+

- Industry 4.0: interconnected systems

⇩

- Multi-System Learning Control





Developments

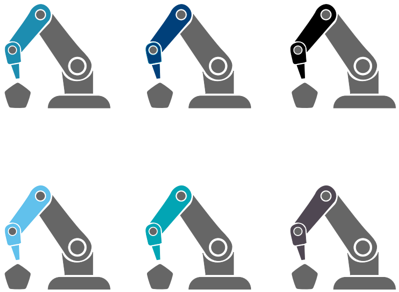
- Fleet of mechatronic systems
- Repetitive task
- Iterative Learning Control

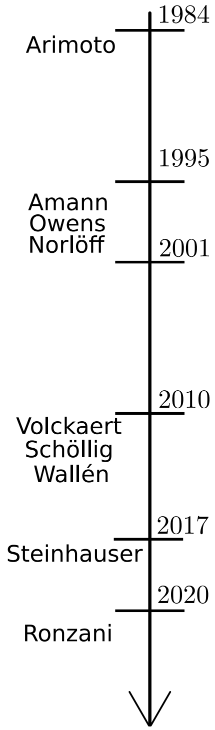
+

- Industry 4.0: interconnected systems

⇩

- Multi-System Learning Control





Developments

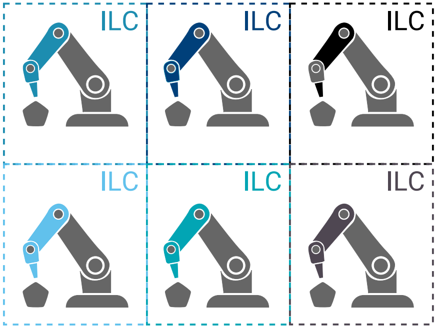
- Fleet of mechatronic systems
- Repetitive task
- Iterative Learning Control

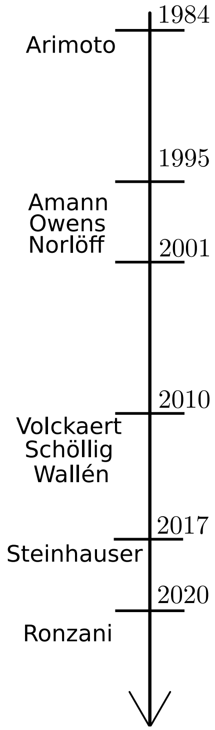
+

- Industry 4.0: interconnected systems

⇩

- Multi-System Learning Control





DEVELOPMENTS

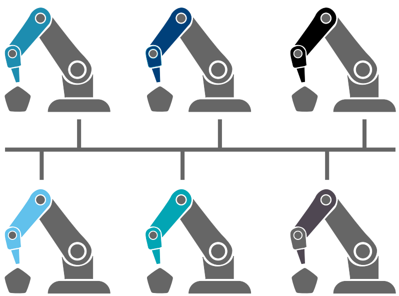
- Fleet of mechatronic systems
- Repetitive task
- Iterative Learning Control

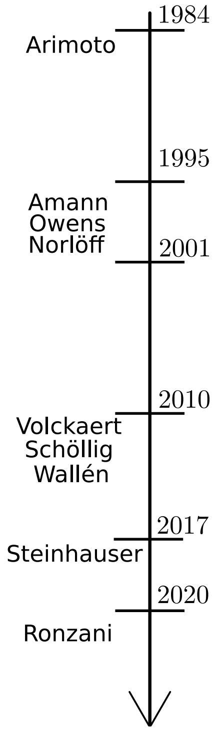
+

- Industry 4.0: interconnected systems

⇩

- Multi-System Learning Control





Developments

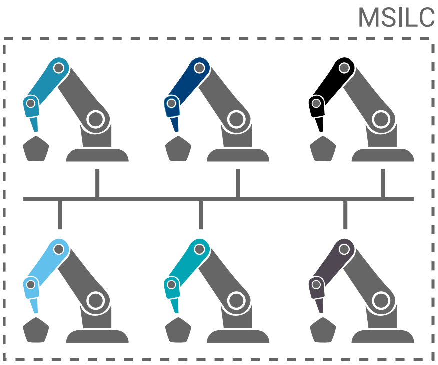
- Fleet of mechatronic systems
- Repetitive task
- Iterative Learning Control

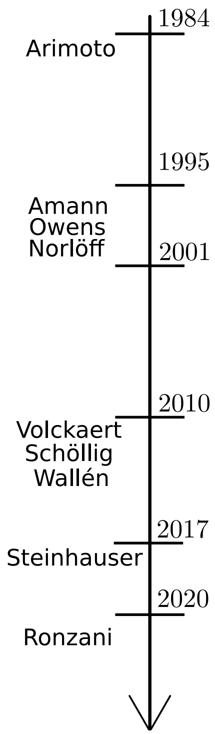
+

- Industry 4.0: interconnected systems

⇩

- Multi-System Learning Control

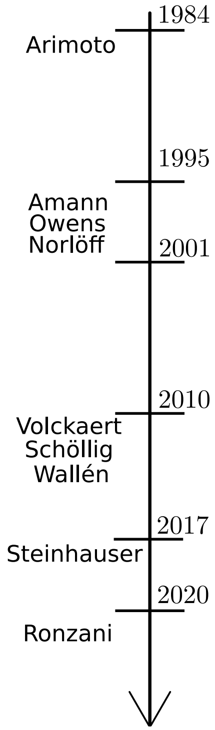




Developments

Multi-system ILC

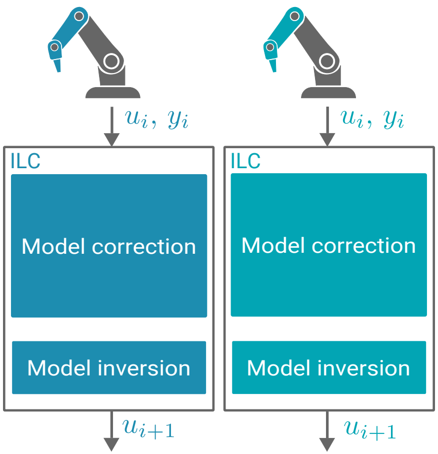
- Model correction: common+delta
- Common ILC vs. Single ILC update
- Performances:
  - $\propto$ similarity of fleet
  - Common ILC reduces iteration-varying disturbances

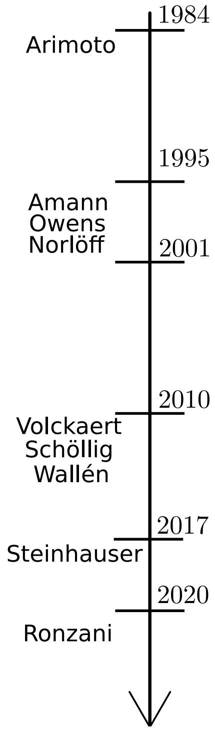


Developments

Multi-system ILC

- Model correction: common+delta
- Common ILC vs. Single ILC update
- Performances:
  - $\propto$ similarity of fleet
  - Common ILC reduces iteration-varying disturbances

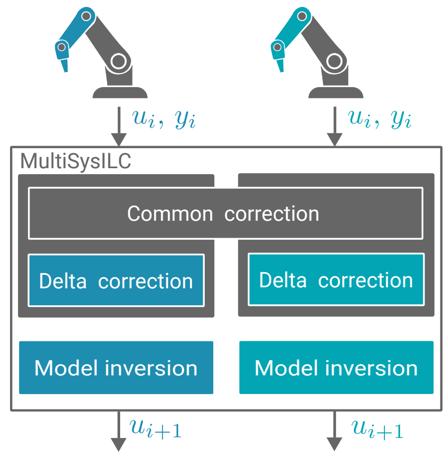


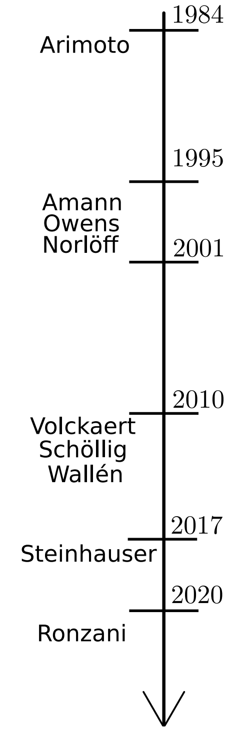


DEVELOPMENTS

Multi-system ILC

- Model correction: common+delta
- Common ILC vs. Single ILC update
- Performances:
  - $\propto$ similarity of fleet
  - Common ILC reduces iteration-varying disturbances

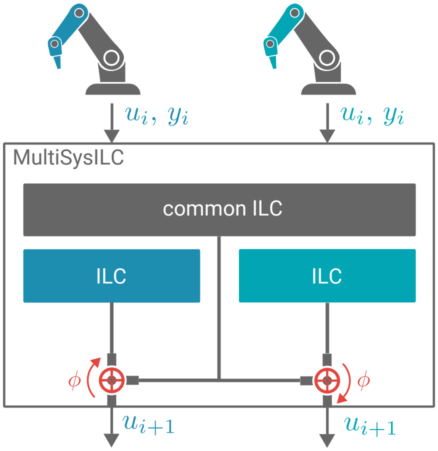


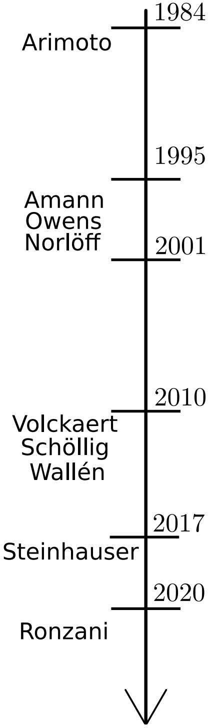


DEVELOPMENTS

Multi-system ILC

- Model correction: common+delta
- Common ILC vs. Single ILC update
- Performances:
  - $\propto$ similarity of fleet
  - Common ILC reduces iteration-varying disturbances

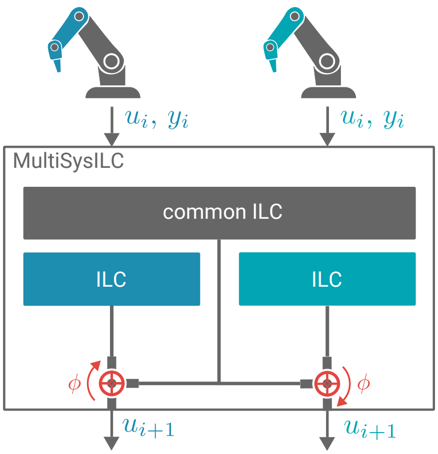


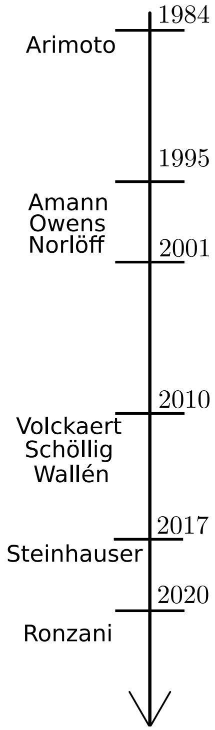


Developments

Multi-system ILC

- Model correction: common+delta
- Common ILC vs. Single ILC update
- Performances:
  - $\propto$ similarity of fleet
  - Common ILC reduces iteration-varying disturbances





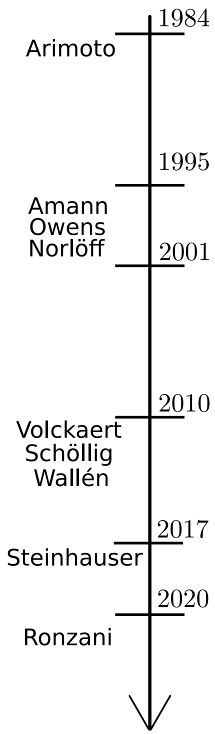
DEVELOPMENTS

ECOset-ILC

- Mechatronic system
- Repetitive task
- Iterative Learning Control

+

- Low-informative sensor (binary/multi-valued)
  - Hall-effect sensors
  - Photoelectric sensors sensors



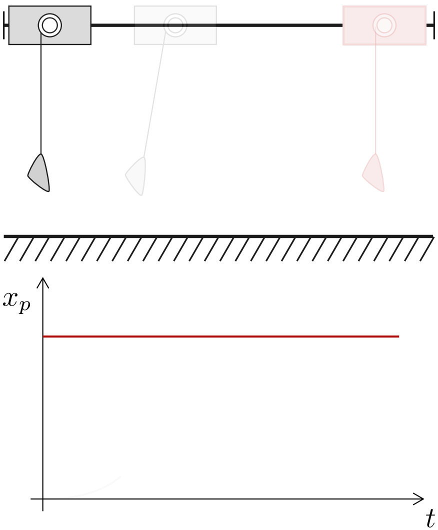
Developments

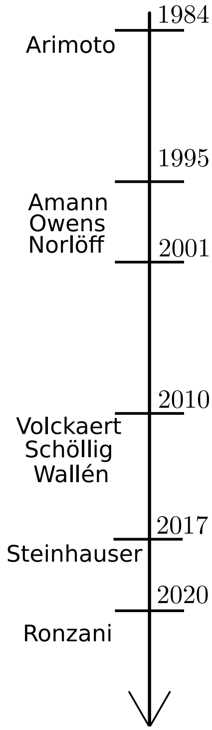
ECOset-ILC

- Mechatronic system
- Repetitive task
- Iterative Learning Control

+

- Low-informative sensor (binary/multi-valued)
  - Hall-effect sensors
  - Photoelectric sensors sensors





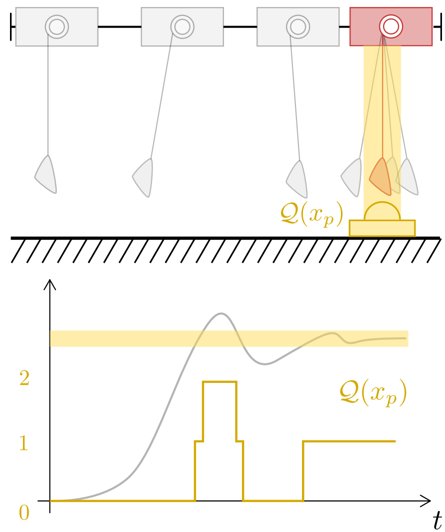
Developments

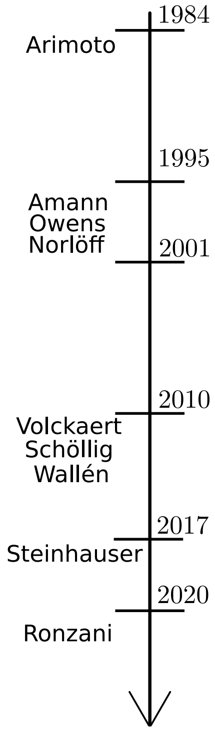
ECOset-ILC

- Mechatronic system
- Repetitive task
- Iterative Learning Control

+

- Low-informative sensor (binary/multi-valued)
    - Hall-effect sensors
    - Photoelectric sensors sensors

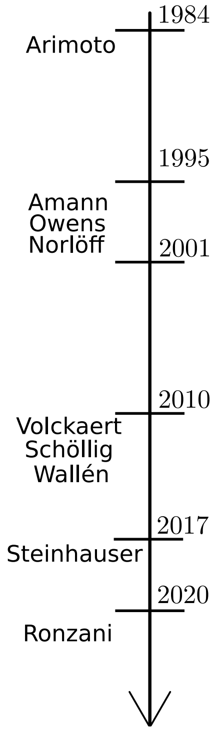




DEVELOPMENTS

ECOset-ILC

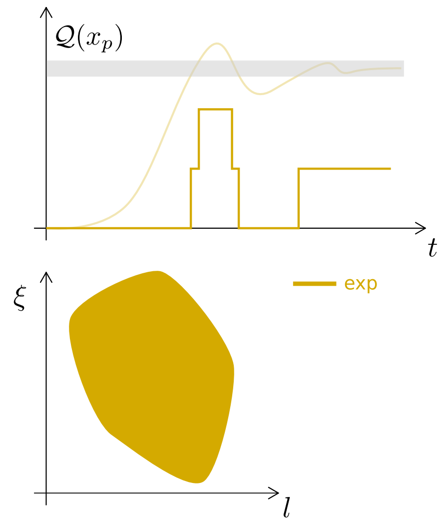
- Set-Membership Estimation
  - Binary/multi-valued output as bounded-noise
  - Parameter set estimate
  - Semi-infinite programming
- Optimal control problem
  - Control Performance (tracking, time-optimal)
  - Optimal Design of Experiments
  - Bilevel Optimization Problem

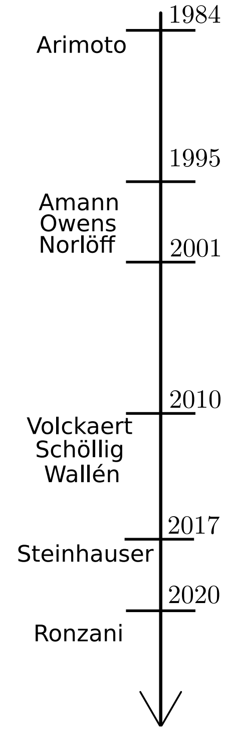


DEVELOPMENTS

ECOset-ILC

- Set-Membership Estimation
  - Binary/multi-valued output as bounded-noise
  - Parameter set estimate
  - Semi-infinite programming
- Optimal control problem
  - Control Performance (tracking, time-optimal)
  - Optimal Design of Experiments
  - Bilevel Optimization Problem

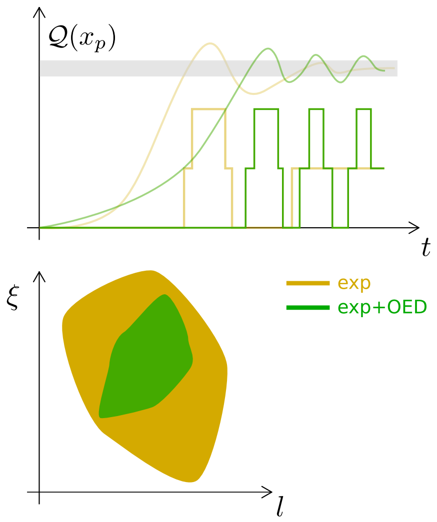


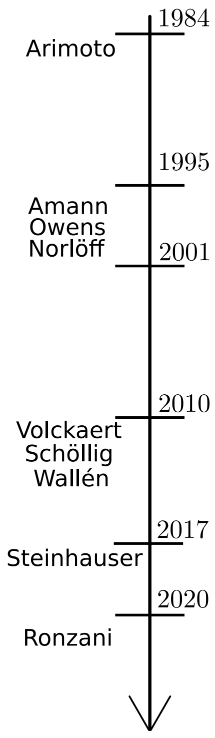


Developments

ECOset-ILC

- Set-Membership Estimation
  - Binary/multi-valued output as bounded-noise
  - Parameter set estimate
  - Semi-infinite programming
- Optimal control problem
  - Control Performance (tracking, time-optimal)
  - Optimal Design of Experiments
  - Bilevel Optimization Problem

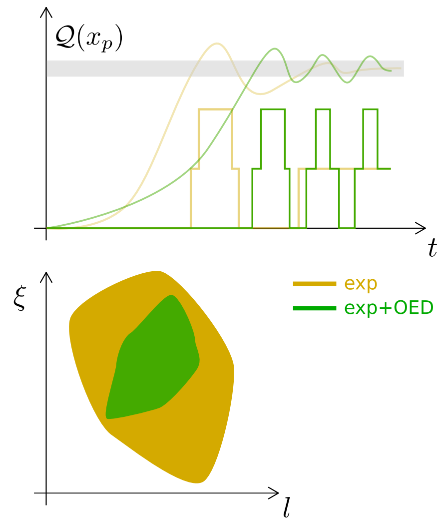


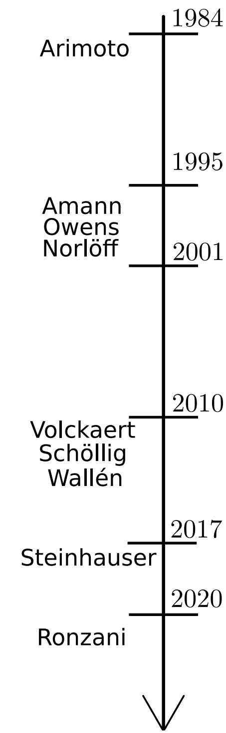


Developments

ECOset-ILC

- Set-Membership Estimation
  - Binary/multi-valued output as bounded-noise
  - Parameter set estimate
  - Semi-infinite programming
- Optimal control problem
  - Control Performance (tracking, time-optimal)
  - Optimal Design of Experiments
  - Bilevel Optimization Problem





Next?

Task flexibility in ILC

- Model
- Explicit corrections terms
  - non-parametric corrections: β represents a disturbance
  - parametric corrections: β adapts physical parameters
- How to combine this correction to trade-off performance vs flexibility?

Next?

Task flexibility in ILC

- Model

$$\hat{\mathcal{P}} : \begin{cases} x(k+1) = f(x(k), u(k)) \\ y(t) = h(x(k), u(k)) \end{cases}$$

- Explicit corrections terms
  - non-parametric corrections: β represents a disturbance
  - parametric corrections: β adapts physical parameters
- How to combine this correction to trade-off performance vs flexibility?

NEXT?

Task flexibility in ILC

- Model

$$\hat{\mathcal{P}} : \begin{cases} x(k+1) = f(x(k), u(k)) \\ y(t) = h(x(k), u(k)) + \beta(k) \end{cases}$$

- Explicit corrections terms
    - non-parametric corrections: β represents a disturbance
    - parametric corrections: β adapts physical parameters
- How to combine this correction to trade-off performance vs flexibility?

Next?

Task flexibility in ILC

- Model

$$\hat{\mathcal{P}} : \begin{cases} x(k+1) = f(x(k), u(k), \beta) \\ y(t) = h(x(k), u(k), \beta) \end{cases}$$

- Explicit corrections terms
    - non-parametric corrections: β represents a disturbance
    - parametric corrections: β adapts physical parameters
- How to combine this correction to trade-off performance vs flexibility?

NEXT?

Task flexibility in ILC

- Model

$$\hat{\mathcal{P}} : \begin{cases} x(k+1) = f(x(k), u(k), \beta) \\ y(t) = h(x(k), u(k), \beta) \end{cases}$$

- Explicit corrections terms
  - non-parametric corrections: β represents a disturbance
  - parametric corrections: β adapts physical parameters
- How to combine this correction to trade-off performance vs flexibility?

Outline

Introduction

Opt-ILC: a brief history

- Filter based an Model-inversion ILC
- Norm-Optimal ILC
- Optimization based ILC
- RoFaLT

Applications and developments

- Applications
- Multi-system ILC
- ECOset-ILC

Conlcusion

Outline

Introduction

Opt-ILC: a brief history

- Filter based an Model-inversion ILC
- Norm-Optimal ILC
- Optimization based ILC
- RoFaLT

Applications and developments

- Applications
- Multi-system ILC
- ECOset-ILC

Conlcusion

Conclusion

Optimization-based ILC

- Model correction: disturbance/parameter estimation
- Model inversion: Optimal Control problem
- Flexible formulation for nonlinear systems $\Longrightarrow$ NLPs

What about stability and convergence?

Conclusion

Optimization-based ILC

- Model correction: disturbance/parameter estimation
- Model inversion: Optimal Control problem
- Flexible formulation for nonlinear systems $\Longrightarrow$ NLPs

What about stability and convergence?

Conclusion

Optimization-based ILC

- Model correction: disturbance/parameter estimation
- Model inversion: Optimal Control problem
- Flexible formulation for nonlinear systems $\Longrightarrow$ NLPs

What about stability and convergence?

Conclusion

Optimization-based ILC

- Model correction: disturbance/parameter estimation
- Model inversion: Optimal Control problem
- Flexible formulation for nonlinear systems $\Longrightarrow$ NLPs

What about stability and convergence?

Conclusion

Optimization-based ILC

- Model correction: disturbance/parameter estimation
- Model inversion: Optimal Control problem
- Flexible formulation for nonlinear systems $\Rightarrow$ NLPs

What about stability and convergence?

Conclusion

Stability and convergence analisys

CONCLUSION

Stability and convergence analisys

- Model correction: additive disturbance (no optimization problem)
- Model inversion: Tracking error minimization
- Linear system dynamics

$\Downarrow$

$\mathcal{LQ}$ framework analysis

Conclusion

Stability and convergence analisys

- Model correction: additive disturbance (no optimization problem)
- Model inversion: Optimal Control Problem (NLP)
- Nonlinear system dynamics

⇩

ZOO-ILC (Baumgärtner, Diehl 2020)

Conclusion

Stability and convergence analisys

- Model correction: Disturbance/parameter Estimation (NLP)
- Model inversion: Optimal Control Problem (NLP)
- Nonlinear system dynamics

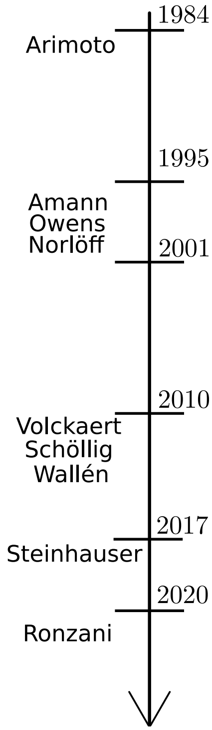
$$\Downarrow$$

?????????????????

References

A brief history of pptimization-based ILC

1. S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of Robots by learning: A new control theory for servomechanism or mechatronics systems," J. Robot. Syst., 1984.
2. N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control for discrete-time systems with exponential rate of convergence," IEE Proc. Control Theory Appl., vol. 143, no. 2, pp. 217–224, 1996.
3. S. Gunnarsson and M. Norrlöf, "On the design of ILC algorithms using optimization," Automatica, vol. 37, no. 12, pp. 2011–2016, Dec. 2001.
4. A. P. Schoellig and R. D'Andrea, "Optimization-based iterative learning control for trajectory tracking," in 2009 European Control Conference, ECC 2009, 2009, pp. 1505–1510.
5. M. Volckaert, J. Swevers, and M. Diehl, "A Two Step Optimization Based Iterative Learning Control Algorithm," ASME 2010 Dyn. Syst. Control Conf. Vol. 1, pp. 579–581, Jan. 2010.
6. J. Wallén, M. Norrlöf, and S. Gunnarsson, "A framework for analysis of observer-based ILC," Asian J. Control, vol. 13, no. 1, pp. 3–14, Jan. 2011.
7. S. Gunnarsson and M. Norrlöf, "On the design of ILC algorithms using optimization," Automatica, vol. 37, no. 12, pp. 2011–2016, Dec. 2001.
8. A. Steinhauser, T. D. Son, E. Hostens, and J. Swevers, "RoFaLT : An Optimization-based Learning Control Tool for Nonlinear Systems," 2018 IEEE 15th Int. Work. Adv. Motion Control, pp. 198–203, Mar. 2018.
9. A. Steinhauser and J. Swevers, "An efficient iterative learning approach to time-optimal path tracking for industrial robots," IEEE Trans. Ind. Informatics, vol. 14, no. 11, pp. 5200–5207, Nov. 2018.
10. D. Ronzani, A. Steinhauser, and J. Swevers, "Multi-System Iterative Learning Control: an Extension of ILC for Interconnected Systems," in 2020 IEEE 16th International Workshop on Advanced Motion Control (AMC), 2020, pp. 79–84.
11. K. Baumgartner and M. Diehl, "Zero-Order Optimization-Based Iterative Learning Control," in Proceedings of the IEEE Conference on Decision and Control, 2020, vol. 2020-Decem, pp. 3751–3757.
12. D. Ronzani, J. Gillis, G. Pipeleers, and J. Swevers, "ECOset-ILC: an Iterative Learning Control Approach with Set-membership Uncertainty," 2022 IEEE 17th Int. Conf. Adv. Motion Control, pp. 68–75, Feb. 2022.



Thank you for your attention.

Questions? daniele.ronzani@kuleuven.be