Poisoning Attacks on Data-Driven Control Methods

Alessio Russo and Alexandre Proutiere

American Control Conference (ACC), 2021

KTH, Royal Institute of Technology, Stockholm

Problem motivation and background

Problem motivation

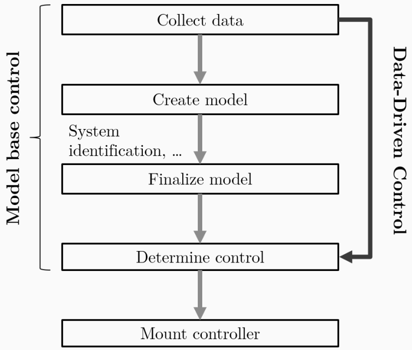
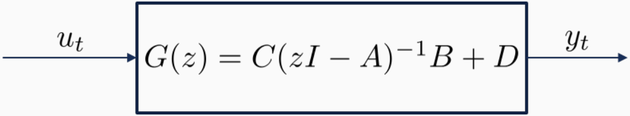


Figure 1: Model-based control vs Data-Driven control paradigms.

- **Data-Driven control: type of offline direct adaptive control.**
- **Offline**: suitable for offline optimization
- **Direct**: directly designs a control law using the gathered data. Two main techniques
  1. Model-reference based methods: Virtual Reference Feedback Tuning (VRFT) [4], Iterative Feedback Tuning [2], correlation-based [3]...
  2. Methods based on Willems' et al. lemma [1,5].
- The data can be poisoned.
- We will focus on VRFT, a popular model-reference based method[1].

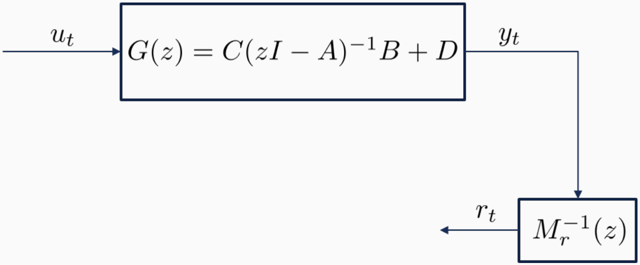
[1]You can find online also an analysis of LMI methods that exploit Willems' et al. lemma.

Background: Virtual Reference Feedback Tuning [4]



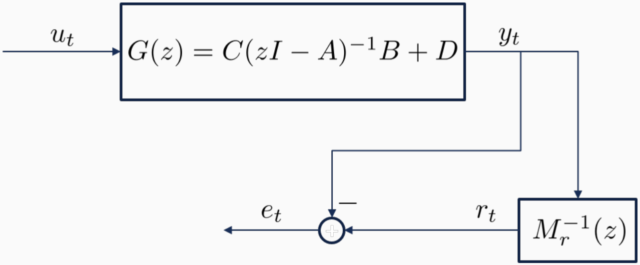
1. Feed a pre-designed signal u_t and measure y_t.

Background: Virtual Reference Feedback Tuning [4]



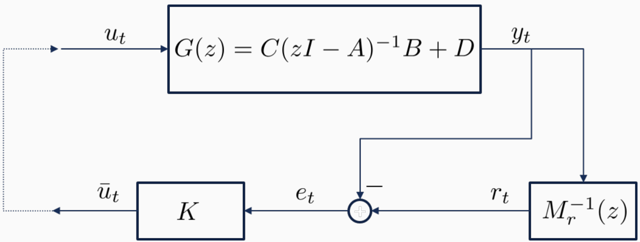
1. Feed a pre-designed signal u_t and measure y_t.
2. Given a reference model $M_r(z)$, compute the reference signal r_t.

Background: Virtual Reference Feedback Tuning [4]



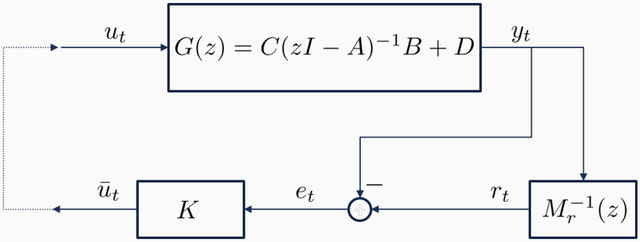
1. Feed a pre-designed signal u_t and measure y_t.
2. Given a reference model $M_r(z)$, compute the reference signal r_t.
3. Compute the *virtual error* $e_t = r_t - y_t$.

Background: Virtual Reference Feedback Tuning [4]



1. Feed a pre-designed signal u_t and measure y_t.
2. Given a reference model $M_r(z)$, compute the reference signal r_t.
3. Compute the *virtual error* $e_t = r_t - y_t$.
4. Design a control law K that outputs a signal $\bar{u}_t$ that is *close* to u_t.

Background: Virtual Reference Feedback Tuning [4]



1. Feed a pre-designed signal u_t and measure y_t.
2. Given a reference model $M_r(z)$, compute the reference signal r_t.
3. Compute the *virtual error* $e_t = r_t - y_t$.
4. Design a control law K that outputs a signal $\bar{u}_t$ that is *close* to u_t.

Under some assumptions, it is possible to show that minimizing $\frac{1}{N}\sum_{t=1}^{N}(\bar{u}_t - u_t)^2$, for $N \to \infty$, yields a law K that converges to the minimum of

$$\min_K \|M_r(z) - (1 - M_r)KG(z)\|_2.$$

Attack formulation

Attack Formulation

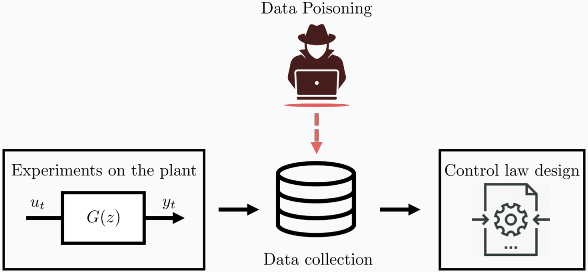


Figure 2: Data poisoning scheme.

- We denote by $u_t' = u_t + a_{u,t}$ the poisoned input, where $\mathbf{a_u}$ is the poisoning signal (similarly for y_t').
- We denote by $\mathcal{L}$ the learner's criterion (e.g., the MSE loss of VRFT).
- Similarly, $\mathcal{A}$ is the attacker's criterion.

Attack Formulation

We can cast the attacker's problem as a bi-level optimization problem.

$$\begin{aligned} \max_{\boldsymbol{u}',\boldsymbol{y}'} \quad & \mathcal{A}(\boldsymbol{u},\boldsymbol{y},K(\boldsymbol{u}',\boldsymbol{y}')) \\ \text{s.t.} \quad & K(\boldsymbol{u}',\boldsymbol{y}') \in \arg\min_K \mathcal{L}(\boldsymbol{u}',\boldsymbol{y}',K) \\ & \|\boldsymbol{u}'-\boldsymbol{u}\|_{q_u} \leq \delta_u, \quad \|\boldsymbol{y}'-\boldsymbol{y}\|_{q_y} \leq \delta_y, \end{aligned}$$

- We denote by $u_t' = u_t + a_{u,t}$ the poisoned input, where $\mathbf{a_u}$ is the poisoning signal (similarly for y_t').
- We denote by $\mathcal{L}$ the learner's criterion (e.g., the MSE loss of VRFT).
- Similarly, $\mathcal{A}$ is the attacker's criterion.
- q_u, q_y are convex norms; $\delta_u, \delta_y \in (0,1)$.

Attack Formulation

$$\max_{\boldsymbol{u}',\boldsymbol{y}'} \quad \mathcal{A}(\boldsymbol{u},\boldsymbol{y},K(\boldsymbol{u}',\boldsymbol{y}'))$$

$$\text{s.t.} \quad K(\boldsymbol{u}',\boldsymbol{y}') \in \arg\min_{K} \mathcal{L}(\boldsymbol{u}',\boldsymbol{y}',K)$$

$$\|\boldsymbol{u}'-\boldsymbol{u}\|_{q_u} \leq \delta_u, \quad \|\boldsymbol{y}'-\boldsymbol{y}\|_{q_y} \leq \delta_y,$$

1. Assume the inner problem $K(\boldsymbol{u}',\boldsymbol{y}') \in \arg\min_K \mathcal{L}(\boldsymbol{u}',\boldsymbol{y}',K)$ is convex and sufficiently regular.
    - We can perform single-level reduction [6] and replace the inner problem with its KKT conditions.
2. Then, assume K is parameterized by θ (we will write K_θ). We can conclude that

$$\nabla_\theta \mathcal{L}(\mathbf{u}',\mathbf{a}',K_\theta) = 0 \Rightarrow \nabla_{\mathbf{a}_u}\theta = -(\nabla_{\mathbf{a}_u}\nabla_\theta \mathcal{L})(\nabla_\theta^2 \mathcal{L})^{-1}$$

   (similarly also for $\mathbf{a}_y$).
3. **This allows us to find approximate attacks** by using gradient ascent methods.

Attack Formulation

$$\begin{aligned} \max_{\boldsymbol{u}', \boldsymbol{y}'} \quad & \mathcal{A}(\boldsymbol{u}, \boldsymbol{y}, K(\boldsymbol{u}', \boldsymbol{y}')) \\ \text{s.t.} \quad & K(\boldsymbol{u}', \boldsymbol{y}') \in \arg\min_{K} \mathcal{L}(\boldsymbol{u}', \boldsymbol{y}', K) \\ & \|\boldsymbol{u}' - \boldsymbol{u}\|_{q_u} \leq \delta_u, \quad \|\boldsymbol{y}' - \boldsymbol{y}\|_{q_y} \leq \delta_y, \end{aligned}$$

1. Assume the inner problem $K(\boldsymbol{u}', \boldsymbol{y}') \in \arg\min_K \mathcal{L}(\boldsymbol{u}', \boldsymbol{y}', K)$ is convex and sufficiently regular.
    - We can perform single-level reduction [6] and replace the inner problem with its KKT conditions.
2. Then, assume K is parameterized by θ (we will write K_θ). We can conclude that

$$\nabla_\theta \mathcal{L}(\mathbf{u}', \mathbf{a}', K_\theta) = 0 \Rightarrow \nabla_{\mathbf{a}_u} \theta = -(\nabla_{\mathbf{a}_u} \nabla_\theta \mathcal{L})(\nabla_\theta^2 \mathcal{L})^{-1}$$

   (similarly also for $\mathbf{a}_y$).
3. **This allows us to find approximate attacks** by using gradient ascent methods.

VRFT: Attack Formulation

VRFT: Attack Formulation

1. **Remember the VRFT criterion** $\frac{1}{N}\sum_{t=1}^{N}(u_t-\bar{u}_t)^2$, where $\bar{u}_t = K_\theta(z)(M_r^{-1}(z)-1)y_t$.
2. Assume $K_\theta = \beta^\top(z)\theta$ is linearly parametrized by $\theta \in \mathbb{R}^d$, with $\beta_i(z)$ being a rational transfer function. The learner's criterion under attack can be rewritten in matrix form as

$$\mathcal{L}(\mathbf{u}', \mathbf{y}', \theta) = \frac{1}{N}\|\mathbf{u}' - \Phi(\mathbf{y}')\theta\|_2^2$$

   where Φ is a matrix that includes the effect of $M_r(z)$ (ref. model) and $\beta(z)$.
3. **How do we choose the attacker's criterion? Simplest choice is to just maximize the original VRFT criterion!**

$$\begin{aligned} \max_{\boldsymbol{u}',\boldsymbol{y}'} \quad & \mathcal{A}(\boldsymbol{u}, \boldsymbol{y}, \hat{\theta}(\boldsymbol{u}', \boldsymbol{y}')) = \frac{1}{N}\left\|\boldsymbol{u} - \Phi(\boldsymbol{y})\hat{\theta}(\boldsymbol{u}', \boldsymbol{y}')\right\|_2^2 \\ \text{s.t.} \quad & \hat{\theta}(\boldsymbol{u}', \boldsymbol{y}') = \left(\Phi^\top(\boldsymbol{y}')\Phi(\boldsymbol{y}')\right)^{-1}\Phi^\top(\boldsymbol{y}')\boldsymbol{u}' \\ & \|\boldsymbol{u}' - \boldsymbol{u}\|_{q_u} \le \delta_u, \quad \|\boldsymbol{y}' - \boldsymbol{y}\|_{q_y} \le \delta_y. \end{aligned}$$

The problem is concave in the input signal $\mathbf{u}'$*, and non-convex in the output signal* $\mathbf{y}'$*.*

VRFT: Attack Formulation

1. **Remember the VRFT criterion** $\frac{1}{N}\sum_{t=1}^{N}(u_t - \bar{u}_t)^2$, where $\bar{u}_t = K_\theta(z)(M_r^{-1}(z) - 1)y_t$.
2. Assume $K_\theta = \beta^\top(z)\theta$ is linearly parametrized by $\theta \in \mathbb{R}^d$, with $\beta_i(z)$ being a rational transfer function. The learner's criterion under attack can be rewritten in matrix form as

$$\mathcal{L}(\mathbf{u}', \mathbf{y}', \theta) = \frac{1}{N}\|\mathbf{u}' - \Phi(\mathbf{y}')\theta\|_2^2$$

   where Φ is a matrix that includes the effect of $M_r(z)$ (ref. model) and $\beta(z)$.
3. **How do we choose the attacker's criterion? Simplest choice is to just maximize the original VRFT criterion!**

$$\begin{aligned}
\max_{\boldsymbol{u}', \boldsymbol{y}'} \quad & \mathcal{A}(\boldsymbol{u}, \boldsymbol{y}, \hat{\theta}(\boldsymbol{u}', \boldsymbol{y}')) = \frac{1}{N}\left\|\boldsymbol{u} - \Phi(\boldsymbol{y})\hat{\theta}(\boldsymbol{u}', \boldsymbol{y}')\right\|_2^2 \\
\text{s.t.} \quad & \hat{\theta}(\boldsymbol{u}', \boldsymbol{y}') = \left(\Phi^\top(\boldsymbol{y}')\Phi(\boldsymbol{y}')\right)^{-1}\Phi^\top(\boldsymbol{y}')\boldsymbol{u}' \\
& \|\boldsymbol{u}' - \boldsymbol{u}\|_{q_u} \leq \delta_u, \quad \|\boldsymbol{y}' - \boldsymbol{y}\|_{q_y} \leq \delta_y.
\end{aligned}$$

The problem is concave in the input signal $\mathbf{u}'$, and non-convex in the output signal $\mathbf{y}'$.

VRFT: Attack Formulation

Input: Data-set $(\boldsymbol{u}, \boldsymbol{y})$; objective function $\mathcal{A}$;
parameters $\boldsymbol{\delta}$, η

Output: Attack vectors $\boldsymbol{a}_u, \boldsymbol{a}_y$

$i \leftarrow 0, (\boldsymbol{a}_u^{(i)}, \boldsymbol{a}_y^{(i)}) \leftarrow (\mathbf{0}, \mathbf{0})$

$\hat{\theta}^{(i)} \leftarrow \hat{\theta}(\boldsymbol{u} + \boldsymbol{a}_u^{(i)}, \boldsymbol{y} + \boldsymbol{a}_y^{(i)})$

$J^{(i)} \leftarrow \mathcal{A}(\boldsymbol{u}, \boldsymbol{y}, \hat{\theta}^{(i)})$

do

- $\boldsymbol{a}_u^{(i+1)} \leftarrow$ solve attacker's problem in $\boldsymbol{a}_u$ using CCP [7]
- $\boldsymbol{a}_y^{(i+1)} \leftarrow \mathrm{PGA}(\delta_y, \gamma_i, \hat{\theta}(\boldsymbol{u}+\boldsymbol{a}_u^{(i+1)}, \boldsymbol{y}+\boldsymbol{a}_y^{(i)}))$
- $\hat{\theta}^{(i+1)} \leftarrow \hat{\theta}(\boldsymbol{u} + \boldsymbol{a}_u^{(i+1)}, \boldsymbol{y} + \boldsymbol{a}_y^{(i+1)})$
- $J^{(i+1)} \leftarrow \mathcal{A}(\boldsymbol{u}, \boldsymbol{y}, \hat{\theta}^{(i+1)})$
- $i \leftarrow i + 1$

while $|J^{(i+1)} - J^{(i)}| > \eta$

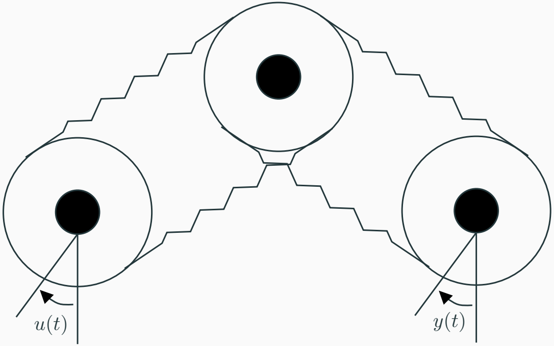
-Remember that $\mathbf{u}' = \mathbf{u} + \mathbf{a_y}$ (resp. $\mathbf{y}'$).
-The attacker wants to solve

$$\begin{aligned} \max_{\boldsymbol{u}', \boldsymbol{y}'} \quad & \frac{1}{N} \left\| \boldsymbol{u} - \Phi(\boldsymbol{y}) \hat{\theta}(\boldsymbol{u}', \boldsymbol{y}') \right\|_2^2 \\ \text{s.t.} \quad & \hat{\theta}(\boldsymbol{u}', \boldsymbol{y}') = \left(\Phi^\top(\boldsymbol{y}') \Phi(\boldsymbol{y}') \right)^{-1} \Phi^\top(\boldsymbol{y}') \boldsymbol{u}' \\ & \|\boldsymbol{u}' - \boldsymbol{u}\|_{q_u} \le \delta_u, \quad \|\boldsymbol{y}' - \boldsymbol{y}\|_{q_y} \le \delta_y. \end{aligned}$$

-The problem is concave in the input signal $\mathbf{u}'$: **we use convex-concave programming techniques.**
-The problem is non-convex in the output signal $\mathbf{y}'$: **we use projected gradient ascent.**

Simulations

Simulations[2]



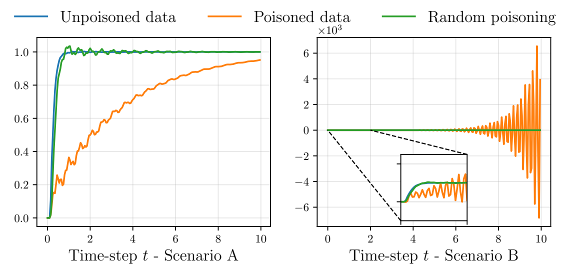
Flexible transmission system, from [4].

$$\max_{\boldsymbol{u}',\boldsymbol{y}'} \quad \frac{1}{N}\left\|\boldsymbol{u}-\Phi(\boldsymbol{y})\hat{\theta}(\boldsymbol{u}',\boldsymbol{y}')\right\|_2^2$$
$$\text{s.t.} \quad \hat{\theta}(\boldsymbol{u}',\boldsymbol{y}') = \left(\Phi^\top(\boldsymbol{y}')\Phi(\boldsymbol{y}')\right)^{-1}\Phi^\top(\boldsymbol{y}')\boldsymbol{u}'$$
$$\|\boldsymbol{u}'-\boldsymbol{u}\|_{q_u} \le \delta_u, \quad \|\boldsymbol{y}'-\boldsymbol{y}\|_{q_y} \le \delta_y.$$

- **Linearly parametrized controller** $K_\theta(z) = \beta^\top(z)\theta$, where $\theta \in \mathbb{R}^6$ and $\beta_i(z) = \frac{z^{-i+2}}{z-1}$, $i = 1, \dots, 6$.
- $\delta_y = \varepsilon_y\|\mathbf{y}\|_2$ and $\delta_u = \varepsilon_u\|\mathbf{u}\|_2$.
- **Scenario (A)**: u_t is a step signal
- **Scenario (B)**: u_t is a white noise signal

[2]All the code can be found at `https://github.com/rssalessio/PoisoningDataDrivenControl`.

Simulations[2]



Step response: comparison between unpoisoned and poisoned data. On the left scenario (A), on the right scenario (B). We used $\varepsilon_u = 0.1$ and $\varepsilon_y = 0.07$.

$$\max_{\boldsymbol{u}', \boldsymbol{y}'} \quad \frac{1}{N} \left\| \boldsymbol{u} - \Phi(\boldsymbol{y}) \hat{\theta}(\boldsymbol{u}', \boldsymbol{y}') \right\|_2^2$$

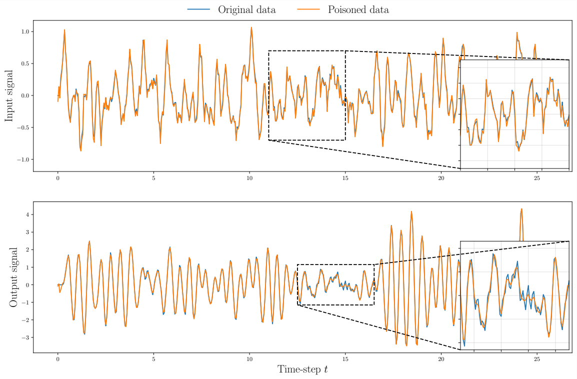
$$\text{s.t.} \quad \hat{\theta}(\boldsymbol{u}', \boldsymbol{y}') = \left(\Phi^\top(\boldsymbol{y}') \Phi(\boldsymbol{y}') \right)^{-1} \Phi^\top(\boldsymbol{y}') \boldsymbol{u}'$$

$$\|\boldsymbol{u}' - \boldsymbol{u}\|_{q_u} \leq \delta_u, \quad \|\boldsymbol{y}' - \boldsymbol{y}\|_{q_y} \leq \delta_y.$$

- **Linearly parametrized controller** $K_\theta(z) = \beta^\top(z)\theta$, where $\theta \in \mathbb{R}^6$ and $\beta_i(z) = \frac{z^{-i+2}}{z-1}$, $i = 1, \dots, 6$.
- $\delta_y = \varepsilon_y \|\mathbf{y}\|_2$ and $\delta_u = \varepsilon_u \|\mathbf{u}\|_2$.
- **Scenario (A)**: u_t is a step signal
- **Scenario (B)**: u_t is a white noise signal

[2] All the code can be found at `https://github.com/rssalessio/PoisoningDataDrivenControl`.

Simulations[2]



Difference between original/poisoned data for scenario (B). We used $\varepsilon_u = 0.1$ and $\varepsilon_y = 0.07$.

$$\max_{\boldsymbol{u}',\boldsymbol{y}'} \quad \frac{1}{N}\left\|\boldsymbol{u} - \Phi(\boldsymbol{y})\hat{\theta}(\boldsymbol{u}',\boldsymbol{y}')\right\|_2^2$$

$$\text{s.t.} \quad \hat{\theta}(\boldsymbol{u}',\boldsymbol{y}') = \left(\Phi^\top(\boldsymbol{y}')\Phi(\boldsymbol{y}')\right)^{-1}\Phi^\top(\boldsymbol{y}')\boldsymbol{u}'$$

$$\|\boldsymbol{u}' - \boldsymbol{u}\|_{q_u} \le \delta_u, \quad \|\boldsymbol{y}' - \boldsymbol{y}\|_{q_y} \le \delta_y.$$

- **Linearly parametrized controller** $K_\theta(z) = \beta^\top(z)\theta$, where $\theta \in \mathbb{R}^6$ and $\beta_i(z) = \frac{z^{-i+2}}{z-1}$, $i = 1, \dots, 6$.
- $\delta_y = \varepsilon_y\|\mathbf{y}\|_2$ and $\delta_u = \varepsilon_u\|\mathbf{u}\|_2$.
- **Scenario (A)**: u_t is a step signal
- **Scenario (B)**: u_t is a white noise signal

[2] All the code can be found at `https://github.com/rssalessio/PoisoningDataDrivenControl`.

Conclusions

Conclusions

- Data Poisoning is not a new concept in Machine Learning (see Biggio et al. [8]).
- Several methods developed by the ML community could be adapted to the data-driven control case.
- Multiple venues of research:
    1. Improve the data-poisoning algorithm. The solution heavily depends on the PGA step.
    2. Better investigate the theoretical properties of the attack.
    3. Make the data-driven algorithm more robust: either by (1) adding constraints, or (2) use some kind of adversarial training.
    4. Attack detection using statistical testing.
    5. Further tests on real control systems.

Thank you for listening!

References

1. Willems, Jan C., et al. "A note on persistency of excitation." Systems & Control Letters 54.4 (2005): 325-329.
2. Hjalmarsson, Hakan, et al. "Iterative feedback tuning: theory and applications." IEEE control systems magazine 18.4 (1998): 26-41.
3. Karimi, A., L. Mikovi, and D. Bonvin. "Iterative correlationbased controller tuning." International journal of adaptive control and signal processing 18.8 (2004): 645-664.
4. Campi, Marco C., Andrea Lecchini, and Sergio M. Savaresi. "Virtual reference feedback tuning: a direct method for the design of feedback controllers." Automatica 38.8 (2002): 1337-1346.
5. De Persis, Claudio, and Pietro Tesi. "Formulas for data-driven control: Stabilization, optimality, and robustness." IEEE Transactions on Automatic Control 65.3 (2019): 909-924.
6. Sinha, Ankur, Pekka Malo, and Kalyanmoy Deb. "A review on bilevel optimization: from classical to evolutionary approaches and applications." IEEE Transactions on Evolutionary Computation 22.2 (2017): 276-295.
7. Shen, Xinyue, et al. "Disciplined convex-concave programming." 2016 IEEE 55th Conference on Decision and Control (CDC). IEEE, 2016.
8. Biggio, Battista, Blaine Nelson, and Pavel Laskov. "Poisoning attacks against support vector machines." arXiv preprint arXiv:1206.6389 (2012).