Comparison of Direct and Indirect Data-Driven Predict Control

Peng Xie
Department of Electrical and Computer Engineering, Chair of Automatic Control Engineering (LSR)
Technical University of Munich
Munich, Germany
ge96tas@tum.de

***Abstract*—Learning from data is essential to every area of science. It is the core of statistics and artificial intelligence and is becoming ever more prevalent also in the engineering domain. Control engineering is one of the domains where learning from data is now considered as a prime issue.**

***Index Terms*—Data-Driven Control Design, Direct Data-Driven Control, Indirect Data-Driven Control, Model Based Control**

I. Introduction

Before explaining the data-driven method, I want to explain with model-based control. The research of model-based control (MBC) is relatively comprehensive, including design methods and stability analysis, and there are many applications. However, the bottleneck facing MBC is that many model control systems (robots, intelligent vehicles, and mechanical systems) have increasingly complex design structures, and the system models are highly nonlinear, making it difficult to establish an accurate dynamics model. In this situation, the control effectiveness of MBC is greatly restricted. In the era of big data, due to the development of sensors, accurate data resources are abundant and easily obtained. How to effectively use these data in the control system has become a research topic.

One approach of data-driven control (DDC) methods is to directly obtain the control model using data (Indirect Data-Driven Method), and another approach is based on Willems' Fundamental Lemma [1](direct Data-Driven Method). The entire process only requires system data, without relying on an accurate mathematical model. This method has a good practical application prospect. It is evident that DDC has many advantages over MBC, with the two primary methods having distinct strengths. As such, we aim to explore the superiority of both methods in varying situations. Here is an example. In large power plants, the dynamics of grid-tied power converters depend on the inherent control scheme and characteristics of the grid. However, on the one hand, the actual model of the grid is complex and variable. On the other hand, power converter suppliers typically do not share their proprietary models. Furthermore, in wind farms, there are complex aerodynamic interactions that are often difficult to model and considered interference. Therefore, model-based design and manual tuning can be cumbersome for commissioning engineers, since they may have neither grid nor converter models. Nonetheless, the dynamics of grid-connected wind turbines can be captured through input and output data in the power converter controller, which interacts with the grid in a closed-loop manner. These input and output data can be used for data-driven control design. An open question is whether the direct data-driven approach has advantages over the indirect data-driven approach.

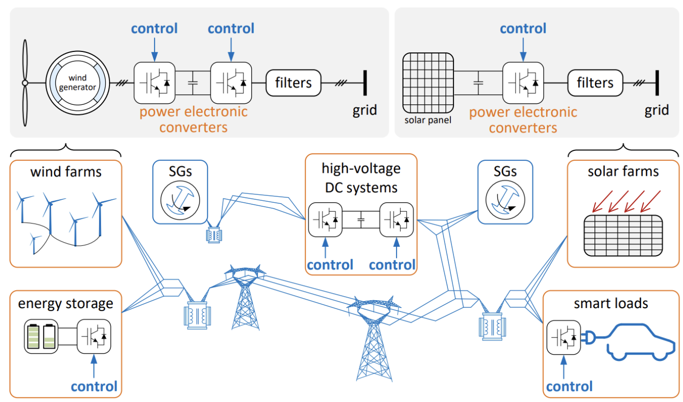


Fig. 1. 5 An illustration of a power electronics dominated power system

A. Example of Direct Data-Driven Control

One example of the direct data-driven method is the use of data-driven control for autonomous vehicles. In this scenario, data is collected from various sensors on the vehicle, such as cameras, lidars, and radars. The collected data is then used to directly generate a control signal for the vehicle. The control signal is generated based on the real-time sensory data, rather than a mathematical model of the vehicle dynamics. For instance, a camera on the vehicle can capture an image of the road ahead, and a deep neural network can be used to detect obstacles and predict the optimal path for the vehicle. The deep neural network can be trained on large datasets of images collected from various driving scenarios to learn how to detect obstacles and generate a control signal for the vehicle. The direct data-driven control system can then generate a control signal for the vehicle based on the sensory data from the camera, lidar, and radar, and the predictions from the deep neural network. The control signal can be used to control the speed and steer the vehicle to follow the predicted path. In this example, the direct data-driven method can be used to control the autonomous vehicle without the need for a mathematical model of the vehicle dynamics. This makes the control system more flexible and adaptive, as it can handle real-world uncertainties and changing environments. The direct data-driven method can be particularly useful in scenarios where the mathematical model of the vehicle dynamics is difficult to obtain or when the vehicle operates in environments with significant uncertainties.

B. Example of Indirect Data-Driven Control

One example of the application of the indirect data-driven method is in the control of a robotic arm. In a manufacturing environment, the robotic arm is used to perform repetitive tasks such as pick and place, assembly, and painting. To control the robotic arm using the Indirect Data Driven Method, data from various sensors such as encoders, force/torque sensors, and cameras are collected. This data is then processed using machine learning algorithms such as neural networks to obtain a model of the robotic arm's dynamics. This model is then used to design a control algorithm that can achieve the desired control objectives, such as tracking a reference trajectory or regulating the position and orientation of the end-effector. For instance, suppose that the goal is to track a desired end-effector trajectory. The Indirect Data Driven Method can be applied by collecting input-output data from the robotic arm while it follows a set of predefined reference trajectories. This data is then used to train a neural network that can predict the relationship between the reference inputs and the arm's outputs. The trained neural network model is then used as a substitute for the mathematical model of the arm's dynamics in the control design process. This leads to a data-driven control algorithm that can track the desired end-effector trajectory while considering the nonlinear dynamics of the robotic arm.

C. Direct Data-Driven Control vs Indirect Data-Driven Control

In conclusion, the Indirect Data Driven Method has proven to be a useful approach for controlling robotic arms in a manufacturing environment. By utilizing the data from various sensors, the method can overcome the limitations of model-based control methods, which rely on precise mathematical models that are difficult to obtain for complex systems. The main difference between direct and indirect data-driven methods lies in the way they use the data to control a system. Direct data-driven methods use the input-output data of a system to directly calculate control inputs, while indirect data-driven methods use the data to learn a system model and then use the model to generate control inputs. In direct data-driven methods, control inputs are calculated from the input-output data of the system using techniques such as system identification, adaptive control, or reinforcement learning. These methods do not require a mathematical model of the system, and the control inputs are directly calculated from the input-output data without the need for a system model. On the other hand, indirect data-driven methods use the input-output data of the system to learn a model of the system. The system model is then used to generate control inputs. The advantage of this approach is that the system model can capture the complex dynamics of the system and be used to generate control inputs that are more effective. However, this approach requires more computational resources and may be subject to errors in the model. In summary, both direct and indirect data-driven methods have their own advantages and disadvantages, and the choice of method depends on the specific requirements of the application. Direct data-driven methods are simpler, faster, and require less computational resources, while indirect data-driven methods can provide more accurate control inputs but are more complex and require more computational resources. [2]

II. Data-Driven Control

The key property of the underlying control task in the model-based setting is established at the beginning, which will be useful in setting up the data-driven formulation. The output $\mathbf{y} = (y_1, \ldots, y_T)$ of the system over the horizon $1, \ldots, T$ generated by the control input $\mathbf{u} = (u_0, \ldots, u_{T-1})$ is given by:

$$\mathbf{y} = \mathcal{G}\mathbf{u} + \mathcal{G}'\mathbf{w},$$

where $\mathbf{w} = (w_0, \ldots, w_{T-1})$ is the process noise, and:

$$\mathcal{G} = \begin{bmatrix} CB & 0 & \ldots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{T-1}B & CA^{T-2}B & \ldots & CB \end{bmatrix}.$$

The expression for $\mathcal{G}$ above is modified by replacing B with I_n, resulting in the formation of $\mathcal{G}'$. Subsequently, $\mathbf{v}$ is obtained as $\mathcal{G}'\mathbf{w}$ and the control task can be expressed using $F(\mathbf{u}, \mathbf{y}) = \sum_{t=0}^{T-1} (\alpha_t (u_t) + \beta_t (y_{t+1}))$. The control task is now:

$$\min_{\mathbf{u} \in \mathbb{R}^{mT}} \mathbb{E}_{\mathbf{v}} \{F(\mathbf{u}, \mathbf{y}) \mid \mathbf{y} = \mathcal{G}\mathbf{u} + \mathbf{v}\}.$$

The certainty equivalence property for model-based predictive control of stochastic LTI systems with a quadratic cost is established by the following lemma: (Certainty equivalent model-based predictive control). For $\mathbf{u}^* = \arg\min_{\mathbf{u} \in \mathbb{R}^{mT}} F(\mathbf{u}, \mathcal{G}\mathbf{u})$, we have $\mathbb{E}_{\mathbf{v}} [F (\mathbf{u}^*, \mathcal{G}\mathbf{u}^* + \mathbf{v})] = \min_{\mathbf{u} \in \mathbb{R}^{mT}} \mathbb{E}_{\mathbf{v}} [F(\mathbf{u}, \mathcal{G}\mathbf{u} + \mathbf{v})]$ [3]. It is established that the minimizer to the stochastic optimization problem can be equivalently obtained as the solution to the following deterministic optimization problem, given the input-output model $\mathcal{G}$.

$$\min_{\mathbf{u} \in \mathbb{R}^{mT}} F(\mathcal{P}\mathbf{u}), \quad \mathcal{P} = \begin{bmatrix} I \\ \mathcal{G} \end{bmatrix}.$$

The minimizer $\mathbf{u}^*$ is now computed. With $\mathbf{y}\text{ref} = I_T \otimes y_{\text{ref}}$,

$$F(\mathbf{u}, \mathbf{y}) = \mathbf{u}^\top \mathcal{Q} \mathbf{u} + (\mathbf{y} - \mathbf{y}_{\text{ref}})^\top \mathcal{R} (\mathbf{y} - \mathbf{y}_{\text{ref}}),$$

where $\mathcal{Q} = \operatorname{diag}(Q_0, \ldots, Q_{T-1})$ and $\mathcal{R} = \operatorname{diag}(R_0, \ldots, R_{T-1})$. It is further obtained:

$$\nabla F(\mathbf{u}, \mathbf{y}) = 2 \begin{bmatrix} \mathcal{Q} & 0 \\ 0 & \mathcal{R} \end{bmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{y} \end{pmatrix} - 2 \begin{pmatrix} 0 \\ \mathcal{R}\mathbf{y}_{\text{ref}} \end{pmatrix}.$$

The minimizer $\mathbf{u}^*$ satisfies $\mathcal{P}^\top \nabla F(\mathcal{P}\mathbf{u}^*) = 0$ (firstorder optimality condition). Substituting from the above:

$$\mathcal{P}^\top \left[\begin{array}{cc} \mathcal{Q} & 0 \\ 0 & \mathcal{R} \end{array} \right] \mathcal{P}\mathbf{u}^* = \mathcal{P}^\top \left(\begin{array}{c} 0 \\ \mathcal{R}\mathbf{y}_{\text{ref}} \end{array} \right),$$

Simplifying the above:

$$\mathbf{u}^* = \left(\mathcal{Q} + \mathcal{G}^\top \mathcal{R} \mathcal{G}\right)^{-1} \mathcal{G}^\top \mathcal{R} \mathbf{y}_{\text{ref}}.$$

However, it should be noted that the certainty equivalence property was established under the assumption of availability of the true input-output behavior model $\mathcal{P}$. In the data-driven control setting, direct access to $\mathcal{P}$ for control design is not available. Instead, an estimate of $\mathcal{P}$ is either obtained from noisy input-output behavior data U, Y (indirect data-driven control), or the data matrix $\left[\begin{array}{cc} U^\top & Y^\top \end{array} \right]^\top$ is itself used in place of the behavior model (direct data-driven control). In other words, the use of an estimate $\widehat{\mathcal{P}}$ of the true behavior model $\mathcal{P}$ is involved in data-driven control design. In indirect data-driven control design, such an estimate is explicitly obtained from data, whereas direct data-driven control design involves the use of an implicit estimate. The data-driven control design problem is formulated by replacing $\mathcal{P}$ with $\widehat{\mathcal{P}}$:

$$\min_{\mathbf{u} \in \mathbb{R}^{mT}} F(\widehat{\mathcal{P}}\mathbf{u}), \quad \widehat{\mathcal{P}} = \left[\begin{array}{c} I \\ \widehat{\mathcal{G}} \end{array} \right].$$

Also, let $\widehat{\mathbf{u}} = \arg\min_{\mathbf{u} \in \mathbb{R}^{mT}} F(\widehat{\mathcal{P}}\mathbf{u})$. Following similar steps as in the computation of $\mathbf{u}^*$, and:

$$\widehat{\mathbf{u}} = \left(\mathcal{Q} + \hat{\mathcal{G}}^\top \mathcal{R} \widehat{\mathcal{G}}\right)^{-1} \hat{\mathcal{G}}^\top \mathcal{R} \mathbf{y}_{\text{ref}}.$$

The notation $F(\mathbf{u}, \mathbf{y})$ and $F(\mathcal{P}\mathbf{u})$ are used interchangeably. It is observed that $\mathcal{P}\mathbf{u} = (\mathbf{u}, \mathcal{G}\mathbf{u}) \in \mathbb{R}^{(m+p)T}$ has the same dimension as $(\mathbf{u}, \mathbf{y}) \in \mathbb{R}^{(m+p)T}$. The fact that the estimate $\widehat{\mathcal{G}}$ and the true model $\mathcal{G}$ do not match, implies that the control input $\widehat{\mathbf{u}}$ cannot be guaranteed to be optimal for the control task. The performance of $\widehat{\mathbf{u}}$ is measured by its suboptimality gap given by:

$$\operatorname{Gap}(\widehat{\mathbf{u}}) = F(\mathcal{P}\widehat{\mathbf{u}}) - F(\mathcal{P}\mathbf{u}^*).$$

This suboptimality gap serves as a metric for comparing the direct and indirect data-driven control design methodologies.It is noted that $\operatorname{Gap}(\widehat{\mathbf{u}}) \geqslant 0$ for any $\widehat{\mathbf{u}}$, and it can be concluded from the μ-strong convexity of F that $\operatorname{Gap}(\widehat{\mathbf{u}}) \geqslant \frac{\mu}{2} \|(\widehat{\mathbf{u}} - \mathbf{u}^*)\|^2$. Furthermore, the ν-Lipschitz continuity of the gradient of F implies that $\operatorname{Gap}(\widehat{\mathbf{u}}) \leqslant \frac{\nu}{2} \|(\widehat{\mathbf{u}} - \mathbf{u}^*)\|^2$. The combination of these results yields the following:

$$\frac{\mu}{2} \|(\widehat{\mathbf{u}} - \mathbf{u}^*)\|^2 \leqslant \operatorname{Gap}(\widehat{\mathbf{u}}) \leqslant \frac{\nu}{2} \|(\widehat{\mathbf{u}} - \mathbf{u}^*)\|^2.$$

It is observed that the error $\|(\widehat{\mathbf{u}} - \mathbf{u}^*)\|$ is caused by the mismatch Δ between $\widehat{\mathcal{G}}$ and $\mathcal{G}$. Additionally, the suboptimality gap $\operatorname{Gap}(\widehat{\mathbf{u}})$ is also controlled by this mismatch, as can be seen from the aforementioned. [4]

A. *Direct Data-Driven Method*

With the aim of obtaining a clearer and more intuitive appreciation of the physical significance of direct data-driven control, a new model has been proposed by C. De Persis and colleagues at the University of Zurich. This model facilitates enhanced research and simulation of the method in question. And the following figure can be used to illustrate the theorem.

The theory put forth by them suggests the utilization of prior trajectories for the estimation of future trajectories. The linear combination serves as a weighting factor, representing the effect of each trajectory at various times on the future trajectory. Consequently, the calculation of the linear. we use the Hankel matrix to represent the previous trajectories.

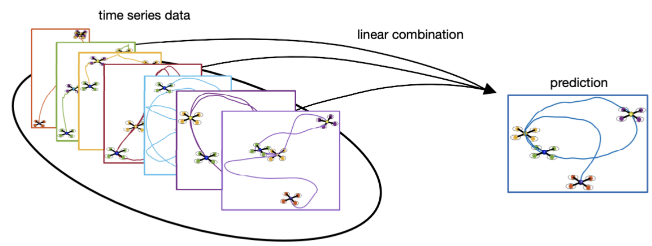


Fig. 2. The function of linear combination

Let now $u_{d,[0,T-1]}$ and $y_{d,[0,T-1]}$ be the input-output data of the system collected during an experiment, and let

$$[U:Y] := \left[\begin{array}{cccc} u_d(0) & u_d(1) & \cdots & u_d(T-t) \\ u_d(1) & u_d(2) & \cdots & u_d(T-t+1) \\ \vdots & \vdots & \ddots & \vdots \\ u_d(t-1) & u_d(t) & \cdots & u_d(T-1) \\ \hline y_d(0) & y_d(1) & \cdots & y_d(T-t) \\ y_d(1) & y_d(2) & \cdots & y_d(T-t+1) \\ \vdots & \vdots & \ddots & \vdots \\ y_d(t-1) & y_d(t) & \cdots & y_d(T-1) \end{array} \right]$$

be the corresponding Hankel matrix, here, $U = U_{0,t,T-t+1}$, $Y = Y_{0,t,T-t+1}$

Then any t-long input/output trajectory of system can be expressed as

$$\left[\begin{array}{c} u_{[0,t-1]} \\ y_{[0,t-1]} \end{array} \right] = \left[\begin{array}{c} U_{0,t,T-t+1} \\ \hline Y_{0,t,T-t+1} \end{array} \right] g$$

where $g \in \mathbb{R}^{T-t+1}$. [5]

Here each column represents a trajectory, we use the initial state to calculate the linear combination g, and to predict the next step.

B. *Indirect Data-Driven Method*

For the indirect method, we want to use the data collected to generate the model. we need to find the A, and B here.

$$x(k+1) = Ax(k) + Bu(k)$$

In fact, noting that

$$X_{1,T} = \begin{bmatrix} B & A \end{bmatrix} \begin{bmatrix} U_{0,1,T} \\ \hline X_{0,T} \end{bmatrix}$$

where

$$X_{1,T} = \begin{bmatrix} x_d(1) & x_d(2) & \dots & x_d(T) \end{bmatrix}$$

it follows immediately that

$$\begin{bmatrix} B & A \end{bmatrix} = X_{1,T} \begin{bmatrix} U_{0,1,T} \\ \hline X_{0,T} \end{bmatrix}^{\dagger}$$

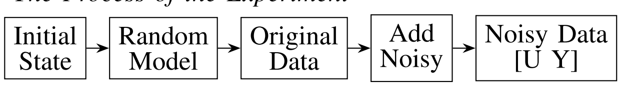
In particular, the right-hand side of the above identity is simply the minimizer of the least-square problem. [5]

III. Problem Setup

Therefore, we want to understand these two approaches, especially when there are different type of noises and the datasize. This is also the main research object of this paper.

IV. EXPERIMENT DESIGN

A. The Process of the Experiment



Step 1: Original Data   Step 2: Noisy Data

We proceed by utilizing an available model to generate data, which is subsequently subjected to the introduction of noise. Subsequently, we apply two methods to the resultant noisy data and compare the outcomes with the original data. The criterion for judgment is the error gap $\mathrm{Gap}(\widehat{\mathbf{u}})$ mentioned above.
According to the error gap, we can determine which method is better. The biggest difference between indirect and direct data-driven control is whether we have to create a model to describe the system by using the data collected from the experience. For the Indirect data-driven method, we have to build a model to predict the next step.

B. Direct Data-Driven Method

The algorithm will be like this :

Given the initial state $y_{[0,0]}, u_{[0,N]}, \bar{u}_{[0,N]}, \bar{y}_{[0,N]}$

- **step 1: given** $y_{[0,0]}, u_{[0,T-1]}$ **and use it to generate the** $\mathcal{H}_L(u_{[0,T-1]}, y_{[0,0]})$
- **step 2: calculate the** g **via the pseudoinverse of** $\mathcal{H}_L(u_{[0,T-1]}, y_{[0,0]})$
- **step 3: calculate the** $y_{[1,T]}$**, and put** $y_{[T,T]}$ **as the initial value to replace** $y_{[0,0]}$**, back to the step1, untill get all** $y_{[0,N]}$

end

C. Indirect Data-Driven Method

As we mentioned before, the algorithm we used is based on least-square algorithm

V. Experimental Results

A. The Role g Played in Direct Data Driven ethod

Here are some experiment data generated in Matlab.
And the linear combination g calculated in this example in the first iteration has such value:

- **first iteration**: $0.9984, 0.0142, 0.0329, -0.0126, 0.0055, -0.0008, 0.0024, -0.0004, 0.0018, 0.0007, -0.0002, 0.0005, -0.0002, 0.0012, -0.0001, -0.0008, 0.0022, 0.0027, 0.0019, 0.0004...$

we can conclude: g plays as the temporarily weighting value which changes every iteration so we can judge the corresponding influence the previous trajectories on the predicted one according to the current g value.

B. The Matrix A and B in Indirect Data Driven Method

Here are some experiments data generated by MatLab. The model I used to generate the experiment data is:

$$A = \begin{bmatrix} -0.3304 & 0.3083 \\ 0.3083 & -0.0379 \end{bmatrix} B = \begin{bmatrix} -0.7342 & 0.2323 \\ -0.0308 & 0.4264 \end{bmatrix}$$

We use the indirect data-driven method to process the noise-added experiment data. And we get the system model:

$$A_{\text{id}} = \begin{bmatrix} -0.3301 & 0.3093 \\ 0.3094 & -0.0405 \end{bmatrix} B_{\text{id}} = \begin{bmatrix} -0.7344 & 0.2319 \\ -0.0300 & 0.4270 \end{bmatrix}$$

As we can see these two model is very close.

C. The Results Simulated by MatLab

In the following picture, both methods predict the signal pretty well. They all track the original signal with tolerable errors which is under 1% on the situation that the noise power is 5% of the signal power.

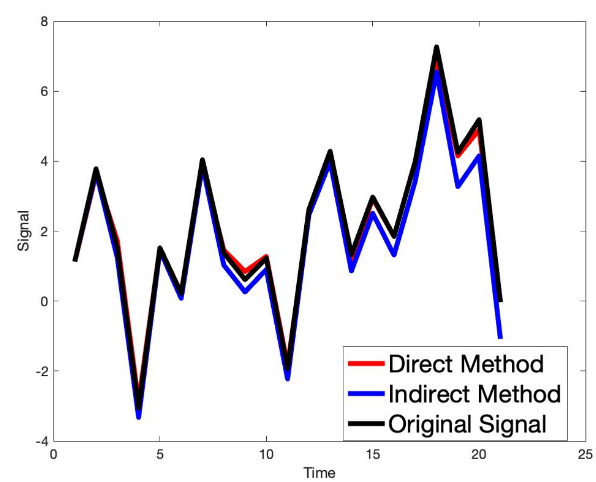


Fig. 3. Original signal and calculated signals through two methods

And here is the comparison of those two methods' performance under the addictive x noise and addictive input u

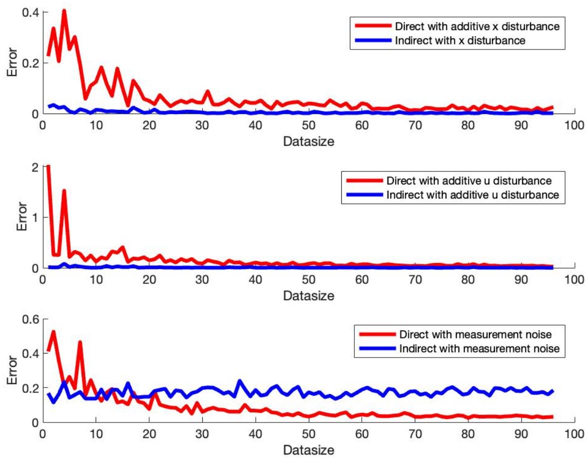


Fig. 4. Gaussien random noise 5% of the signal power

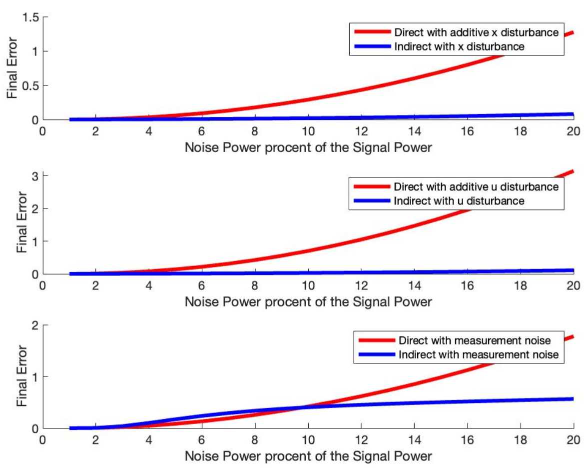


Fig. 5. The influence of changing gaussien random noise intensity from 1% to 20%

noise and the measurement noise. So we want to ask how the different type of noises influence the final result. After introducing the data-driven method, here we can get some results from the experiment data. And the essential parameters which may lead to great different gap errors are:

- database size (N)
- noise intensity
- noise type - Gaussien Random noise and Constant Offset

VI. Discussion

From the above figures generated by MatLab, we can observe: when it comes to the parameter - size of dataset. The result can be conclued as:

- Datasize is small, $\mathbf{error}_{\text{direct}} > \mathbf{error}_{\text{indirect}}$
- $N \to \infty$, $\mathbf{error}_{\text{direct}} \to 0$, $\mathbf{error}_{\text{indirect}} \to \neq 0$

When it comes to the noise type, we observe from experiments in figure.4 that the indirect approach outperformers the direct one with addictive states gaussien random noise and addictive input gaussien random noise. However when it comes to the measurement noise the direct approach performers better in large datasize. When the noise type is constant offset in figure.6, the result is totally different. Both approaches cannot deal such noise very well and the indirect way performers worse and worse with the datasize increasing.

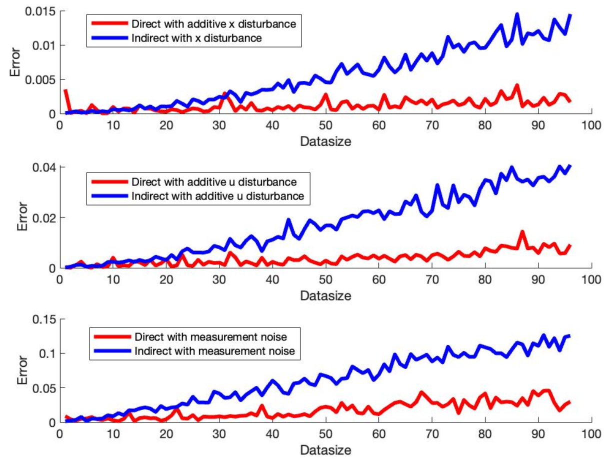


Fig. 6. Constant offset 5% of the signal power

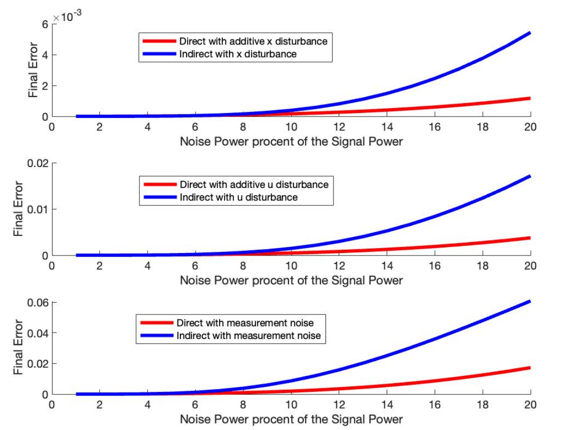


Fig. 7. The influence of changing constant offset from 1% to 20%

When it comes to the noise intensity, we we observe from experiments in figure.5 that with the noise intensity increasing, direct approach increases much faster than indirect one. However when the noise is constant offset, in figuer.7, direct approach perfomers better than indirect one.

The result can be conclued as:

When the noise follows a Gaussian distribution, the performance of the direct approach improves with increasing data size. Conversely, the indirect approach is less affected by data size. Initially, the direct approach is inferior to the indirect approach, but as data size increases, the difference between them approaches zero, particularly when there is measurement noise.

When the noise is a constant offset, the performance of both approaches is poor, but the indirect approach performs even worse. Additionally, there is a linear relationship between the gap and data size in both approaches.

When data size is fixed, the performance of the direct approach deteriorates with increasing strength of random noise, while the indirect approach can maintain relatively good performance. However, when the measurement noise exceeds a certain intensity, the indirect approach also exhibits poor performance.

When data size is fixed and the noise is a constant offset, the opposite is observed. The performance of the indirect approach deteriorates with increasing strength of the offset, while the direct approach can maintain relatively good performance. However, when the offset exceeds a certain intensity, the direct approach also exhibits poor performance.

Both approaches do not perform well when dealing with constant offsets, but the direct approach outperforms the indirect approach when dealing with measurement noise. When the data size is small, the indirect approach performs better.

VII. Conclusion

The study reveals the performance of both direct and indirect data-driven predictive control designs, which precludes the conclusion that either approach is always superior [6].

In such cases, the direct approach is observed to outperform the indirect approach, especially when the dataset size (N) is larger. And when datasize is small, indirect approach is a better option.

In conclusion, the choice between the direct and indirect methods should be based on the specific requirements of the control problem and the characteristics of the available data. The direct approach may be the better option when dealing with large control horizons and noisy data, while the indirect approach may be more suitable when working with longer trajectories and smaller datasets.

References

[1] H. J. van Waarde, C. De Persis, M. K. Camlibel, and P. Tesi, "Willems' fundamental lemma for state-space systems and its extension to multiple datasets," *IEEE Control Systems Letters*, vol. 4, no. 3, pp. 602–607, 2020.

[2] B. Nortmann and T. Mylvaganam, "Data-driven control of linear time-varying systems," pp. 3939–3944, 12 2020.

[3] H. J. van Waarde, C. De Persis, M. K. Camlibel, and P. Tesi, "Willems' fundamental lemma for state-space systems and its extension to multiple datasets," *IEEE Control Systems Letters*, vol. 4, no. 3, pp. 602–607, 2020.

[4] T. Maupong and P. Rapisarda, "Data-driven control: A behavioral approach," *Systems Control Letters*, vol. 101, 05 2016.

[5] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2020.

[6] V. Krishnan and F. Pasqualetti, "On direct vs indirect data-driven predictive control," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 736–741, 2021.