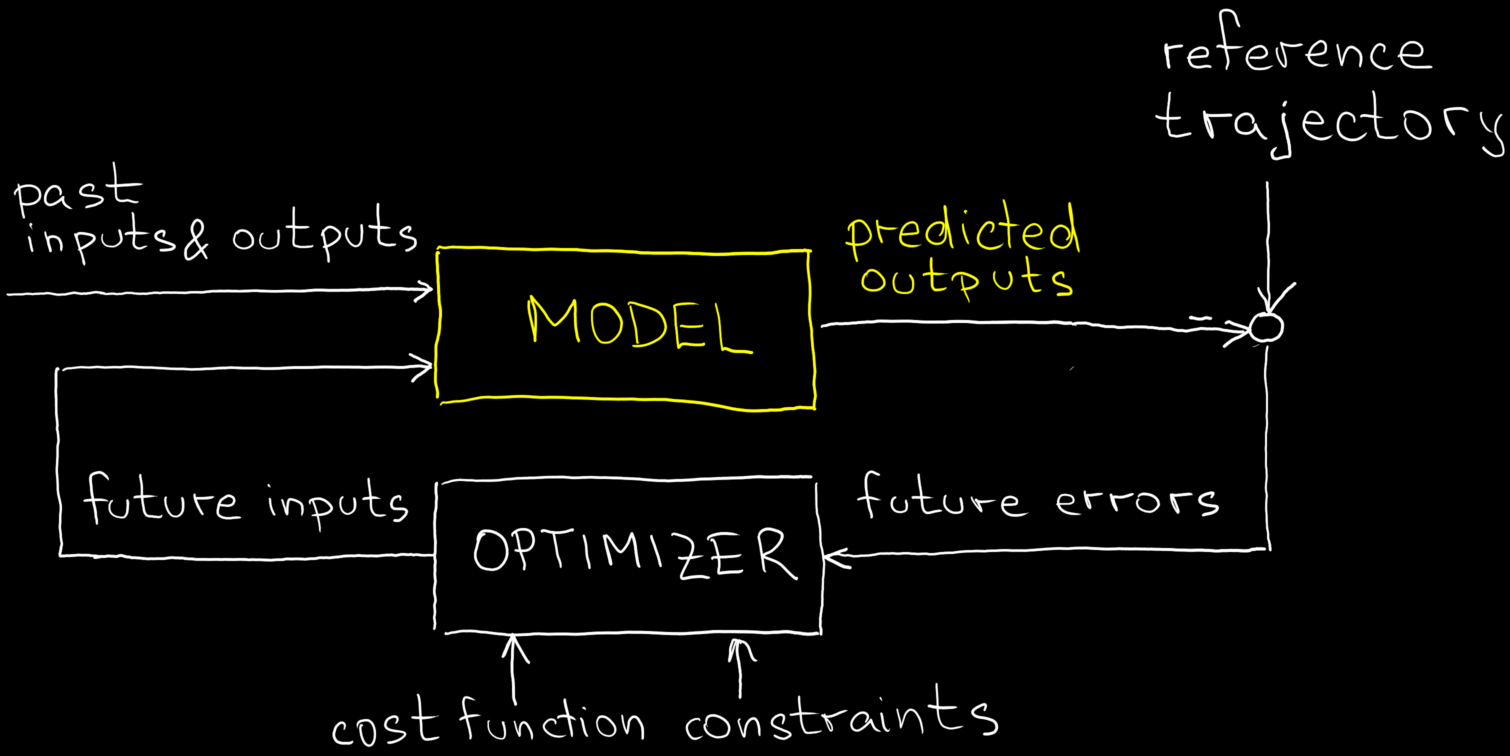
DATA-DRIVEN (M)PC

What is the role of the model in MPC?

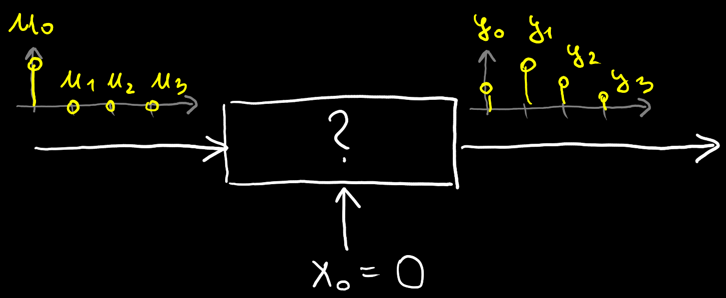


Models:
- state-space $\quad x_{k+1} = f_k(x_k, u_k), \quad y_k = g_k(x_k, u_k)$
- input-output $\quad Y(z) = G(z) \cdot U(z)$

But MPC actually started as DATA-DRIVEN

Late 1970's :
- Model Predictive Heuristic Control (MPHC)
- Model Algorithmic Control (MAC)
- Dynamic Matrix Control (DMC)
- ...

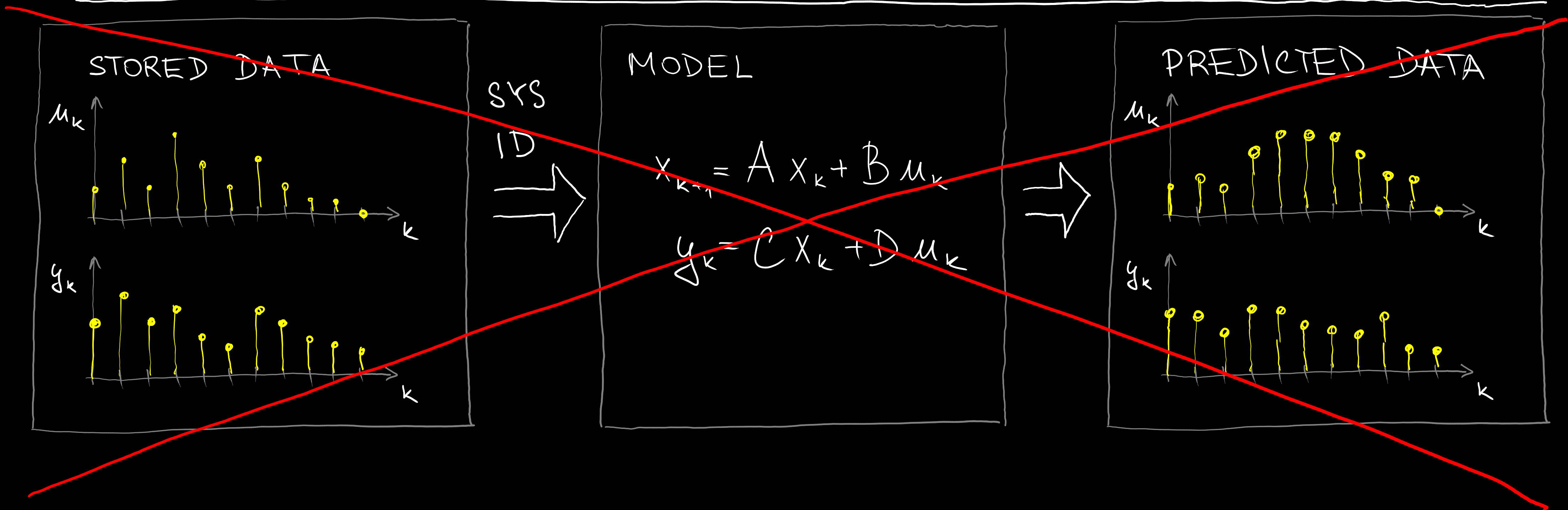
used IMPULSE RESPONSE & STEP RESPONSE as models

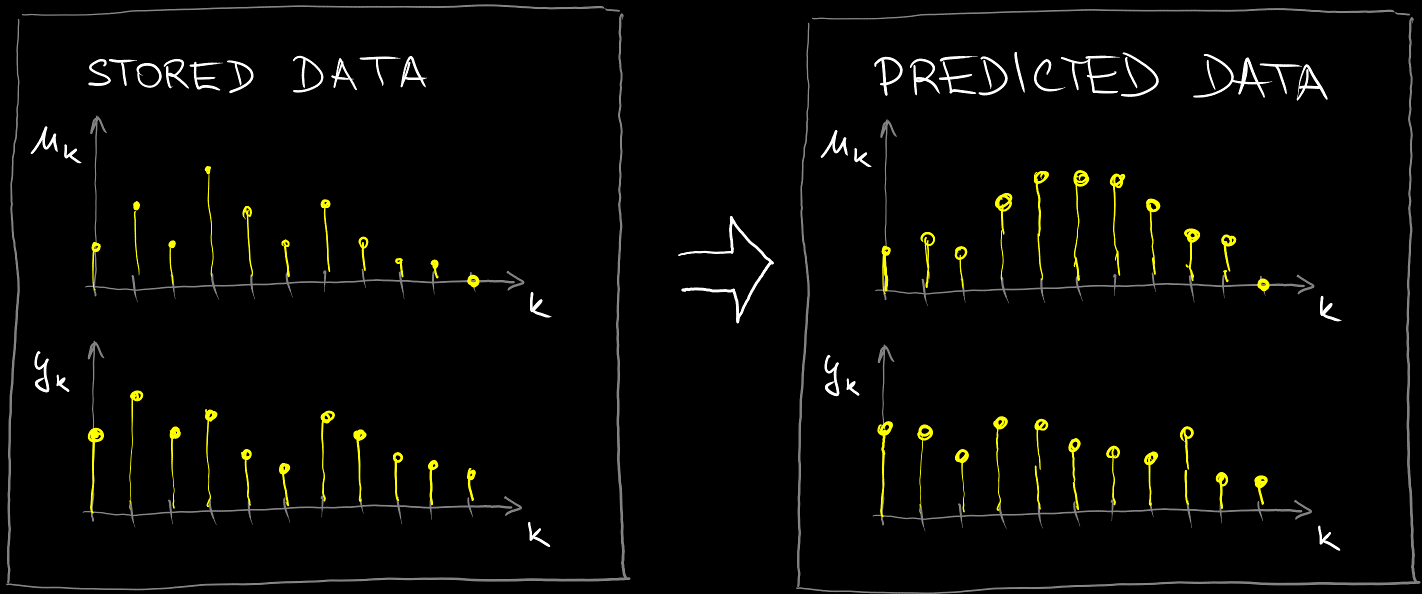


$$y_k^{future} = \sum_{i=0}^{k} y_i \, u_{k-i}^{future}$$

$$= \begin{bmatrix} y_0 & y_1 & y_2 & \dots \end{bmatrix} \cdot \begin{bmatrix} u_k^{future} \\ u_{k-1}^{future} \\ \vdots \end{bmatrix}$$

Goal: Use the past data directly for prediction

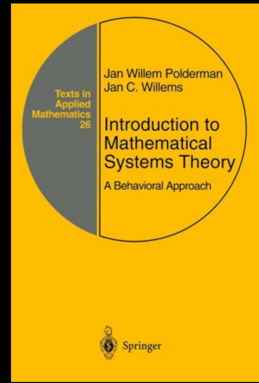


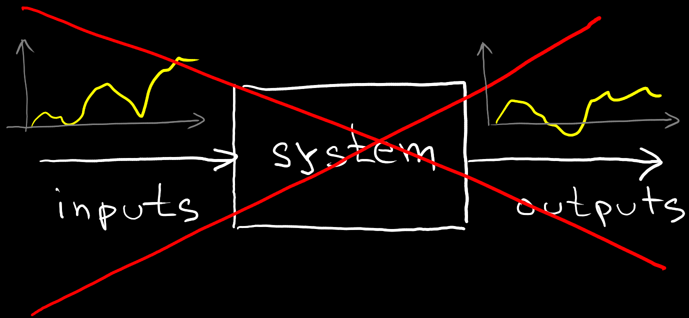


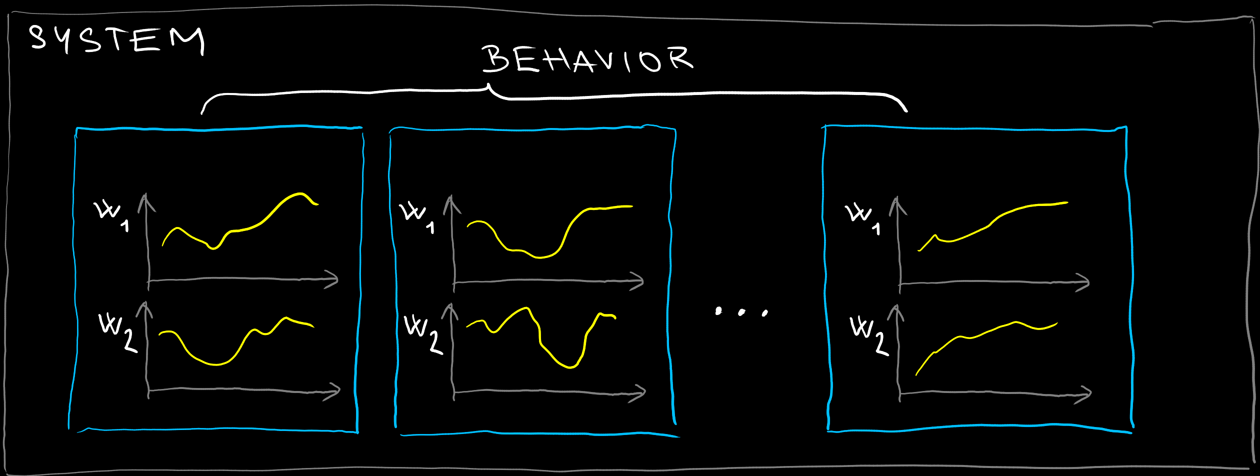
BEHAVIORAL APPROACH to systems & control

By Jan C. Willems since 1990s

Book with J.W. Polderman in 1998

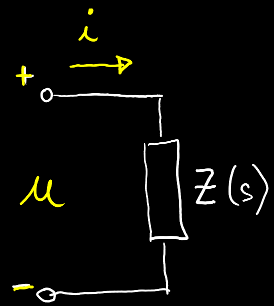






How to characterize?

MODEL-BASED way:

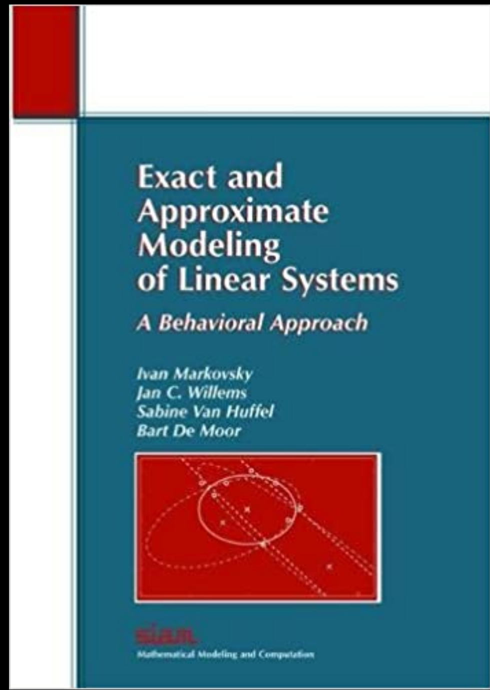


$\cancel{U(s) = Z(s) \cdot I(s)}$

$$\underbrace{\begin{bmatrix} 1 & -Z(s) \end{bmatrix}}_{M(s)} \cdot \underbrace{\begin{bmatrix} U(s) \\ I(s) \end{bmatrix}}_{W(s)} = 0 \iff \boxed{W(s) \in \mathrm{Kernel}(M(s))}$$

Characterization of BEHAVIOR using past DATA

Book by Markovsky, Willems, ... (2006)



Paper by Markovsky (2008)

International Journal of Control
Vol. 81, No. 12, December 2008, 1946–1959

Taylor & Francis
Taylor & Francis Group

Data-driven simulation and control

Ivan Markovsky* and Paolo Rapisarda

School of Electronics and Computer Science, University of Southampton, Southampton, UK

(*Received 21 June 2007; final version received 24 January 2008*)

Classical linear time-invariant system simulation methods are based on a transfer function, impulse response, or input/state/output representation. We present a method for computing the response of a system to a given input and initial conditions directly from a trajectory of the system, without explicitly identifying the system from the data. Similar to the classical approach for simulation, the classical approach for control is model-based: first a model representation is derived from given data of the plant and then a control law is synthesised using the model and the control specifications. We present an approach for computing a linear quadratic tracking control signal that circumvents the identification step. The results are derived assuming exact data and the simulated response or control input is constructed off-line.

Keywords: simulation; data-driven control; output matching; linear quadratic tracking; system identification

1. Introduction

The usual starting point of systems and control problems is a given representation of the plant. As a consequence, the developed solution methods and algorithms are based on input/state/output, transfer

that avoid the explicit derivation of a model representation.

Data-driven algorithms for systems and control problems are presently less developed than their model-based counterparts. Only a few control

HANKEL MATRIX

Given:

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{T-1} \\ w_T \end{bmatrix}$$

$$H_L(w) = \begin{bmatrix} w_1 & w_2 & \cdots & w_{T-L+1} \\ w_2 & w_3 & & \vdots \\ w_3 & w_4 & & \\ \vdots & \vdots & & \\ w_{L-1} & w_L & & \\ w_L & w_{L+1} & & w_T \end{bmatrix}$$

shifted windows of length L

$$w = \begin{bmatrix} u \\ y \end{bmatrix} \qquad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{T_f} \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{T_f} \end{bmatrix}$$

or:

$$w = \begin{bmatrix} \begin{bmatrix} u_1 \\ y_1 \end{bmatrix} \\ \begin{bmatrix} u_2 \\ y_2 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} u_{T_f} \\ y_{T_f} \end{bmatrix} \end{bmatrix}$$

Under some assumptions

FUNDAMENTAL LEMMA:

$$\text{colspan}\left(H_{T_f}\left(\begin{bmatrix} u \\ y \end{bmatrix} \right) \right) = \text{behavior}_{[1, T_f]}$$

Predicting future (input-output) responses based on past responses

Given: — past data $u^d = \begin{bmatrix} u_1^d \\ u_2^d \\ \vdots \\ u_T^d \end{bmatrix}$ $y^d = \begin{bmatrix} y_1^d \\ y_2^d \\ \vdots \\ y_T^d \end{bmatrix}$

input $u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{T_f} \end{bmatrix}$

Compute: future output $y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{T_f} \end{bmatrix}$

$$\begin{array}{c} U \\ \\ \\ Y \end{array} \left[\begin{array}{cccc} u_1^d & u_2^d & \cdots & u_{T-T_f+1}^d \\ u_2^d & u_3^d & \ddots & \vdots \\ \vdots & \vdots & & \vdots \\ u_{T_f}^d & u_{T_f+1}^d & \cdots & u_T^d \\ \hline y_1^d & y_2^d & \cdots & y_{T-T_f+1} \\ y_2^d & y_3^d & \ddots & \vdots \\ \vdots & \vdots & & \vdots \\ y_{T_f}^d & y_{T_f+1}^d & \cdots & y_T^d \end{array}\right] \underset{\text{vector}}{g} = \left[\begin{array}{c} u_1 \\ u_2 \\ \vdots \\ u_{T_f} \\ \hline y_1 \\ y_2 \\ \vdots \\ y_{T_f} \end{array}\right]$$

Solve $\boxed{U g = u}$

$$\downarrow$$

$$\boxed{y = Y \cdot g}$$

Persistency of excitation (of order L)

$$H_L(u^d) = \begin{bmatrix} u_1^d & u_2^d & \cdots & u_{T-L+1}^d \\ u_2^d & u_3^d & & \\ \vdots & \vdots & \ddots & \\ u_L^d & u_{L+1}^d & \cdots & u_T^d \end{bmatrix}$$

$\boxed{L = T_f + n}$

must have full row rank

u^d must be long enough

u^d must be rich enough

Uniqueness of the computed response

State signals are not available $\Rightarrow$ initial state must be <u>ESTIMATED</u>

$$\underbrace{\begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix}}_{\text{initial trajectories of } u \,\&\, y.} \longrightarrow x_{ini}$$

<u>LENGTH</u> $\geq$ LAG ($\leq$ system order)

$\Downarrow$

UNIQUENESS

Given: – past data – $u^d = \begin{bmatrix} u^d_1 \\ u^d_2 \\ \vdots \\ u^d_T \end{bmatrix}$ $\quad y^d = \begin{bmatrix} y^d_1 \\ y^d_2 \\ \vdots \\ y^d_T \end{bmatrix}$

$u_{ini} = \begin{bmatrix} \vdots \end{bmatrix}$ $\quad y_{ini} = \begin{bmatrix} \vdots \end{bmatrix}$

input $u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{T_f} \end{bmatrix}$

Compute: future output $y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{T_f} \end{bmatrix}$

$$U = \begin{bmatrix} U_p \\ \hline U_f \end{bmatrix} = \begin{bmatrix} u^d_1 & \dots & u^d_{T-T_f-T_{ini}+1} \\ \vdots & & \vdots \\ u^d_{T_{ini}} & & \\ \hline u^d_{T_{ini}+1} & & \vdots \\ \vdots & & \\ u^d_{T_{ini}+T_f} & \dots & u^d_T \end{bmatrix}$$

$$Y = \begin{bmatrix} Y_p \\ \hline Y_f \end{bmatrix} = \begin{bmatrix} y^d_1 & \dots & y^d_{T-T_f-T_{ini}+1} \\ \vdots & & \vdots \\ y^d_{T_{ini}} & & \\ \hline y^d_{T_{ini}+1} & & \vdots \\ \vdots & & \\ y^d_{T_{ini}+T_f} & \dots & y^d_T \end{bmatrix}$$

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} \cdot g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{bmatrix} \Longrightarrow \boxed{\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} \cdot g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \end{bmatrix}} \quad \boxed{y = Y_f \cdot g}$$

Data-driven output ~~M~~PC - using behavioral approach

Here following F. Dörfler's DeePC

data-enabled predictive control

but there are some more (J. Berberich, ...)

2019 18th European Control Conference (ECC)
Napoli, Italy, June 25-28, 2019

Data-Enabled Predictive Control: In the Shallows of the DeePC

Jeremy Coulson John Lygeros Florian Dörfler

***Abstract*— We consider the problem of optimal trajectory tracking for unknown systems. A novel data-enabled predictive control (DeePC) algorithm is presented that computes optimal and safe control policies using real-time feedback driving the unknown system along a desired trajectory while satisfying system constraints. Using a finite number of data samples from the unknown system, our proposed algorithm uses a behavioural systems theory approach to learn a non-parametric system model used to predict future trajectories. The DeePC algorithm is shown to be equivalent to the classical and widely adopted Model Predictive Control (MPC) algorithm in the case of deterministic linear time-invariant systems. In the case of nonlinear stochastic systems, we propose regularizations to the DeePC algorithm. Simulations are provided to illustrate performance and compare the algorithm with other methods.**

I. Introduction

As systems are becoming more complex and data is becoming more readily available, scientists and practitioners are beginning to bypass classical model-based techniques in favour of data-driven methods [1]. Data-driven methods are suitable for applications where first-principle models are not conceivable, when models are too complex for control design, and when thorough modelling and parameter identification is too costly.

A challenging problem in systems control is optimal trajectory tracking, where a control policy is computed based on output feedback that drives a dynamical system along a desired output trajectory while minimizing a stage cost and respecting safety constraints. One of the most celebrated and widely used control techniques for trajectory tracking is receding horizon Model Predictive Control (MPC), precisely because it allows one to include safety considerations during control design [2]. The key ingredient for MPC is an accurate parametric state space model of the system, but obtaining such a model is often the most time-consuming and expensive part of control design [3].

In the context of unknown black-box systems, there is no approach which solves the optimal trajectory tracking problem subject to constraints and partial (output) observations. However, some more benign variations of the optimal trajectory tracking problem have been approached using data-driven and learning based methods. We single out some

approaches usually require a large number of data samples to perform well, and are often sensitive to hyper-parameters leading to non-reproducible and highly variable outcomes [4].

Other approaches propose performing sequential system identification (ID) and control. System ID can be used to produce an approximate model and provide finite sample guarantees quantifying model uncertainty, allowing for robust control design [5]. In this spirit, an end-to-end ID and control pipeline is given in [6] and arrives at a data-driven control solution with guarantees on the sample efficiency, stability, performance, and robustness. The system identification step in these approaches disregards one of the main advantages of a data-driven approach: independence from an underlying parametric representation. Additionally, they only consider regulation, rely on having full state information, and do not enforce constraint satisfaction.

Beyond reinforcement learning and sequential ID and control, there are many other safe learning approaches [7]. However, they rely on a-priori stabilizing controllers and safe regions, and thus apply only to a small class of problems.

MPC based on Dynamic Matrix Control has been historically used as a data-driven control technique, in which zero-initial condition step responses are used to predict future trajectories [8]. Although this technique has many limitations [9], it motivates the use of a non-parametric predictive control model. Other non-parametric predictive models have been proposed in [10], [11]. These methods do not solve the problem of optimal trajectory tracking with constraints, but serve as building blocks for our approach.

Here we present a **D**ata-**e**nabl**e**d **P**redictive **C**ontrol (DeePC) algorithm. Unlike classical MPC and the learning-based control techniques outlined above, the DeePC algorithm does not rely on a parametric system representation. Instead, similar to [10], we approach the problem from a behavioural systems theory perspective [12]. Rather than attempting to learn a parametric system model, we aim at learning the system's "behaviour" (see Section IV for the precise definition). Our novel predictive control strategy computes optimal controls for unknown systems using real-time output feedback via a receding horizon implementation,

Bridging direct & indirect data-driven control formulations via regularizations and relaxations

Florian Dörfler, Jeremy Coulson, and Ivan Markovsky

arXiv:2101.01273v1 [math.OC] 4 Jan 2021

***Abstract*—We discuss connections between sequential system identification and control for linear time-invariant systems, which we term indirect data-driven control, as well as a direct data-driven control approach seeking an optimal decision compatible with recorded data assembled in a Hankel matrix and robustified through suitable regularizations. We formulate these two problems in the language of behavioral systems theory and parametric mathematical programs, and we bridge them through a multi-criteria formulation trading off system identification and control objectives. We illustrate our results with two methods from subspace identification and control: namely, subspace predictive control and low-rank approximation which constrain trajectories to be consistent with a non-parametric predictor derived from (respectively, the column span of) a data Hankel matrix. In both cases we conclude that direct and regularized data-driven control can be derived as convex relaxation of the indirect approach, and the regularizations account for an implicit identification step. Our analysis further reveals a novel regularizer and sheds light on the remarkable performance of direct methods on nonlinear systems.**

I. Introduction

The vast realm of data-driven control methods can be classified into *indirect data-driven control* approaches consisting of sequential system identification and model-based control as well as *direct data-driven control* approaches seeking an optimal decision compatible with recorded data. Both approaches have a rich history, and they have received renewed interest cross-fertilized by novel methods and widespread interest in machine learning. Representative recent surveys are [1]–[6].

The pros and cons of both paradigms have often been elaborated on: e.g., modeling and identification is cumbersome, its results are often not useful for control (due to, e.g., incompatible uncertainty quantifications), and practitioners generally prefer end-to-end approaches. While direct data-driven control promises to resolve these problems by learning control policies directly from data, the available methods often do not (yet) lend themselves to real-time and safety-critical control systems due to lack of certificates and overburdening computational and sample complexity, among others. Quite a few approaches tried to bridge the two paradigms. Of relevance to this article,

We take a similar perspective here: the sequential identification and control tasks can be abstracted as nested bi-level optimization problem: find the best con
where the model is the best fit to a da
pothesis class. This approach is modul
tractable formulations, but generally it i
is no separation principle – aside from
Section 4] – for these two nested opti
end-to-end direct algorithmic approach
indirect methods if a tractable formulat
the latter we resort to a paradigm square
system theory and subspace system ide

Behavioral system theory [11]–[13]
on dynamical systems as sets of trajec
require parametric representations whi
from a data-centric perspective. For
invariant (LTI) systems are character
subspaces within an ambient space of t
identification is to find such a low-dir
data. Subspace methods take a similar
mic) viewpoint [14]–[16] and extract p
the range and null spaces of a low-ran

Both lines of work come together
the Fundamental Lemma [17]; see als
extensions. It states that, under some
of all finite-length trajectories (the res
LTI system equals the range space of
This result serves as the theoretic unc
subspace identification [19]–[21] and
particular subspace predictive control b
models [22]–[24], explicit feedback pc
data matrices [25]–[27], and data-enal
seeking compatibility of predicted traj
space of a data Hankel matrix. The la
been established for deterministic LTI
and have recently been extended by su
optimal control problems. Closed-loop
in [30]. The regularizations were first r

Data-Enabled Predictive Control of Autonomous Energy Systems

Florian Dörfler

Automatic Control Laboratory, ETH Zürich

Data-enabled predictive control (DeePC)

$$\underset{g,\,u,\,y}{\text{minimize}} \sum_{k=0}^{T_f-1} \|y_k - r_k\|_Q^2 + \|u_k\|_R^2, \qquad R \succ 0, \quad Q \succeq 0$$

$$\text{subject to} \quad \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{bmatrix}$$

collected offline

updated online

$$u_k \in \mathcal{U} \qquad k = 0, 1, \ldots, T_f - 1$$

$$y_k \in \mathcal{Y}$$

Regularization against noisy past data

$$\underset{g,\, u,\, y}{\text{minimize}} \sum_{k=0}^{T_f-1} \|y_k - r_k\|_Q^2 + \|u_k\|_R^2 + \lambda_g \cdot \|g\|_1 \qquad R \geq 0, \quad Q \geq 0$$

$$\text{subject to} \quad \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} \cdot g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{bmatrix}$$

$$u_k \in \mathcal{U} \qquad k = 0, 1, \dots, T_f - 1$$

$$y_k \in \mathcal{Y}$$

Regularization against noisy measurements

$$\underset{g,\,u,\,y,\,\sigma_y}{\text{minimize}} \sum_{k=0}^{T_f-1} \|y_k - r_k\|_Q^2 + \|u_k\|_R^2 + \lambda_y \cdot \|\sigma_y\| \qquad R \succ 0, \quad Q \succeq 0$$

$$\text{subject to} \quad \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} \cdot g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma_y \\ 0 \\ 0 \end{bmatrix}$$

$$u_k \in \mathcal{U} \qquad k = 0, 1, \ldots, T_f - 1$$

$$y_k \in \mathcal{Y}$$

How about nonlinear systems?

Lifting finite-dimensional nonlinear

to ∞-dimensional linear

$\Downarrow$

KOOPMAN OPERATOR APPROACH, ...