

# Data-driven simulation and control

Ivan Markovsky and Paolo Rapisarda

School of Electronics and Computer Science, University of Southampton

Southampton, SO17 1BJ, UK

{im,pr3}@ecs.soton.ac.uk

## Abstract

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. Similarly 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 function, matrix fraction, impulse response, etc. representations. We call this research paradigm ‘model-based’, and refer to problem formulations based on the input/state/output representation (perhaps the most often used representation) as classical problem formulations.

In this paper, we consider alternative problem formulations, which we call ‘data-driven’, that start from a given observed trajectory of the to-be-simulated or to-be-controlled system (the plant). From the applications point of view, data-driven formulations are closer than their classical counterparts to real-life problems: indeed in practise one rarely has a given input/state/output representation but often may observe a trajectory of the plant. From the theoretical point of view, data-driven formulations allow for a new class of solution methods and algorithms that are not based on any particular representation of the plant.

Of course, a data-driven problem could be solved by a combination of system identification and a model-based method. System identification derives from the given data a model representation of the plant, and a model-based

simulation or control design method could then be applied. We are looking for alternatives to this obvious two-stage solution 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 problems have data-driven solutions. Our purpose in this paper is to outline a generic approach to data-driven simulation and show its fundamental role for deriving data-driven control algorithms. We impose the strong assumption that the data is generated by a linear time-invariant system of an a priori bounded order and all the data is available off-line. Thus, the algorithms presented in this paper are non-recursive and apply to exact data. In a future work, we will address the approximation issue, the recursive and real-time operation of the algorithms, and data-driven feedback control.

### **Overview of the literature, outline of the paper, and summary of contributions**

Data-driven simulation of free response and impulse response is used in (Markovsky *et al.* 2005a) for deriving balanced input/state/output representations from data. In its full generality the data-driven simulation problem is first formulated and solved in (Markovsky *et al.* 2005b). In Section 4, we review the results of (Markovsky *et al.* 2005b), because they are fundamental for the solution of the data-driven control problems, considered in Sections 5 and 6.

Perhaps the first data-driven control method is the Ziegler-Nichols procedure for tuning PID controllers. That method is based, however, on the plant step response, which assumes zero initial conditions and step input. In addition, the method is graphical and does not generalise to other control problems. Procedures for deriving multivariable linear quadratic Gaussian controller, using the plant impulse response, are proposed in (Furuta and Wongsaisuwan 1995) and (Shi and Skelton 2000).

Data-driven synthesis methods using an arbitrary response are developed under different names: unfalsified control (Safonov 1996, Safonov and Tsao 1997), iterative feedback tuning (Hjalmarsson *et al.* 1998, Hildebrand *et al.* 2004), virtual reference feedback tuning (Campi *et al.* 2002), and model free control (Favoreel 1999, Woodley 2001, Ikeda *et al.* 2001, Fujisaki *et al.* 2004, Park and Ikeda 2006). The model free methods as well as the methods proposed in this paper can be classified as subspace-type methods, i.e., similarly to the subspace system identification methods (Van Overschee and De Moor 1996, Verhaegen and Dewilde 1992), they are based on linear algebra operations, such as projections and solution of linear systems of equations. The model less methods of (Favoreel 1999, Ikeda *et al.* 2001, Fujisaki *et al.* 2004) and the data-driven control method, presented in Section 6 of this paper, minimise a finite horizon 2-norm cost function. The solution to this problem is given in closed form by a projection and consequently the solution methods are related. The proposed algorithms for solving the problem, however, are different.

In Section 5, we consider a control problem, called output matching, which is in the following sense the opposite to simulation: the output matching controller aims to determine an input that achieves exactly a desired output, while a simulation algorithm aims to determine an output that corresponds to a given input. (In both problems the initial conditions are assumed to be given.) We present an algorithm for output matching, which is a trivial modification of the data-driven simulation algorithm. As will be shown, a shortcoming of the output matching problem formulation is that it does not impose restrictions on the control signal and therefore the input can grow unbounded.

In Section 6, we consider a data-driven finite horizon linear quadratic tracking problem, which implicitly constrains the input signal and thus avoids the main shortcoming of the output matching control. We present three solutions to the linear quadratic tracking problem. The first one is the classical model-based solution that first computes an input/state/output representation of the plant and then synthesises the controller by solving the corresponding Riccati equation. The second solution is to compute an impulse response representation of the plant and then find the optimal trajectory by solving a weighted least squares problem. The third solution computes the optimal trajectory directly from the given data by projecting the reference trajectory on the set  $\mathcal{B}_0$  of all zero initial conditions trajectories of the plant. Under certain conditions, which will be specified, a basis for  $\mathcal{B}_0$  can be computed from the given trajectory  $w_d$  of the plant, which makes the procedure implementable. We illustrate the equivalence of the three approaches by simulation examples. The third approach is similar to that used in (Favoreel 1999, Ikeda *et al.* 2001, Fujisaki *et al.* 2004). However, the approach presented here has the following novel features compared to the cited sources:

1. sufficient conditions are given under which the data-driven simulation and control problems are solvable,
2. a new algorithm is derived for data-driven control (see Algorithm 8), and
3. the solution of the data-driven control problem is related to the basic idea of computing special responses of the unknown plant (i.e., data-driven simulation).

The control criterion, considered in Section 6 of the paper, is over a *finite horizon*, so that the initial conditions play an important role. In iterative feedback tuning (Hjalmarsson *et al.* 1998, Hildebrand *et al.* 2004) and virtual reference feedback tuning (Campi *et al.* 2002) approaches, instead, the objective is to achieve a desired operating regime of the plant. In this case, an infinite horizon cost is used and therefore the initial conditions are not relevant. This difference is also reflected in the fact that in the approach of this paper one designs a control input while in iterative feedback tuning and virtual reference feedback tuning the aim is to tune the parameters of a feedback controller.

## 2 Preliminaries and notation

We use the behavioural language. A discrete-time dynamical system  $\mathcal{B}$  with  $w$  external variables (inputs and outputs) is a subset of the signal space  $(\mathbb{R}^w)^\mathbb{N}$ . The set of natural numbers  $\mathbb{N}$  is the time-axis, so that a trajectory  $w$  of  $\mathcal{B}$  is a vector time series  $w = (w(1), w(2), \dots)$ , where  $w(t) \in \mathbb{R}^w$ , for all  $t \in \mathbb{N}$ . The notation  $\mathcal{B}_t$  is used for the restriction of the behaviour on the interval  $[1, t]$ , i.e.,

$$\mathcal{B}_t := \{w_p \in (\mathbb{R}^w)^t \mid \text{there is } w_f, \text{ such that } w_p \wedge w_f \in \mathcal{B}\}, \quad (1)$$

where  $w_p \wedge w_f$  denotes the concatenation of the trajectories  $w_p$  and  $w_f$ . Thus,  $w \in \mathcal{B}_t$  is a finite trajectory  $(w(1), \dots, w(t))$  of the system  $\mathcal{B}$ .  $\sigma$  denotes the backwards shift operator  $\sigma w(t) := w(t+1)$ .

The number of inputs  $m$  and the number of outputs  $p$  of a system  $\mathcal{B} \in (\mathbb{R}^{m+p})^{\mathbb{N}}$  are invariant of the representation. Modulo a permutation of the variables, any trajectory  $w \in \mathcal{B}$  has an input/output partition

$$w = \text{col}(u, y) := \begin{bmatrix} u \\ y \end{bmatrix},$$

where  $u$  is an input, i.e., it is free, and  $y$  is an output, i.e., it is determined by the input, the system, and the initial condition.

Denote by  $\mathcal{L}^w$  the class of linear, time-invariant, and finite dimensional systems with  $w$  variables. There are a number of equivalent representations of a system  $\mathcal{B} \in \mathcal{L}^w$ . In this paper, we use the input/state/output representation

$$\mathcal{B}(A, B, C, D) := \{ \text{col}(u, y) \in (\mathbb{R}^{m+p})^{\mathbb{N}} \mid \exists x \in (\mathbb{R}^n)^{\mathbb{N}}, \text{ such that } \sigma x = Ax + Bu \text{ and } y = Cx + Du \},$$

parameterized by the matrices  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$ , and  $D \in \mathbb{R}^{p \times m}$ . The state dimension  $n$  is called the order of the representation. An input/state/output representation  $\mathcal{B}(A, B, C, D)$  is minimal if its order is as small as possible. This smallest possible order  $\mathbf{n}(\mathcal{B})$  is invariant and is called the order of the system. Another invariant that is used in the paper is the lag  $\mathbf{l}(\mathcal{B})$  of  $\mathcal{B}$ . It is defined as the observability index of an input/state/output representation  $\mathcal{B}(A, B, C, D)$  of  $\mathcal{B}$ , i.e., the smallest integer  $\ell$ , for which the observability matrix

$$\mathcal{O}_\ell(A, C) := \text{col}(C, CA, \dots, CA^{\ell-1}) \quad (2)$$

has rank  $\mathbf{n}(\mathcal{B})$ .

The Hankel matrix with  $t_1$  block rows, composed of the signal  $w \in (\mathbb{R}^w)^T$  is denoted by

$$\mathcal{H}_{t_1, t_2}(w) := \begin{bmatrix} w(1) & w(2) & \cdots & w(t_2) \\ w(2) & w(3) & \cdots & w(t_2 + 1) \\ w(3) & w(4) & \cdots & w(t_2 + 2) \\ \vdots & \vdots & & \vdots \\ w(t_1) & w(t_1 + 1) & \cdots & w(t_1 + t_2 - 1) \end{bmatrix}. \quad (3)$$

If the index  $t_2$  is skipped, the matrix  $\mathcal{H}_{t_1}(w)$  is assumed to have the maximal possible number of columns  $t_2 = T - t_1 + 1$ . The signal  $u = (u(1), \dots, u(T))$  is called *persistently exciting* of order  $L$  if the Hankel matrix  $\mathcal{H}_L(u)$  is of full row rank.

The block lower-triangular Toeplitz matrix with  $t$  block rows, composed of the signal  $H = (H(0), H(1), \dots, H(t-1))$  is denoted by

$$\mathcal{T}_t(H) := \begin{bmatrix} H(0) & 0 & 0 & \cdots & 0 \\ H(1) & H(0) & 0 & \ddots & \vdots \\ H(2) & H(1) & H(0) & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ H(t-1) & \cdots & H(2) & H(1) & H(0) \end{bmatrix}. \quad (4)$$

Let  $H$  be the impulse response of  $\mathcal{B}$ , i.e., with  $\mathcal{B} = \mathcal{B}(A, B, C, D)$  of order  $\mathbf{n}$ , we have

$$H(0) = D, \quad H(\tau) = CA^{\tau-1}B, \quad \text{for } \tau = 1, 2, \dots, t-1. \quad (5)$$

Using the notation (2) for the observability matrix and the notation (4) for the lower triangular Toeplitz matrix, we can express the condition that  $w$  is a trajectory of  $\mathcal{B}$  as a linear system of equations

$$w = \text{col}(u, y) \in \mathcal{B}_t \iff \exists x_{\text{ini}} \in \mathbb{R}^{\mathbf{n}}, \text{ such that } y = \mathcal{O}_t(A, C)x_{\text{ini}} + \mathcal{T}_t(H)u. \quad (6)$$

We use the same letter  $y$  for the signal  $(y(1), \dots, y(t))$  as well as for the vector  $\text{col}(y(1), \dots, y(t))$ , i.e., we identify the spaces  $(\mathbb{R}^{\mathbf{p}})^t$  and  $\mathbb{R}^{\mathbf{p}t}$ . Corresponding to a given input  $u \in (\mathbb{R}^{\mathbf{m}})^t$ , the system  $\mathcal{B} \in \mathcal{L}^{\mathbf{m}+\mathbf{p}}$  has an  $\mathbf{n}(\mathcal{B})$ -dimensional space of outputs  $y$ , such that  $\text{col}(u, y) \in \mathcal{B}_t$ . This space is parameterized by the initial condition.

The Moore-Penrose pseudoinverse of the matrix  $A$  is denoted by  $A^+$ .  $\text{colspan}(A)$  denotes the span of the columns of  $A$ .  $I_m$  is the  $m \times m$  identity matrix and  $0_{m \times n}$  is the  $m \times n$  zero matrix.

### 3 Fundamental lemmas

#### 3.1 Specification of initial condition

In a representation free setting, we specify initial condition for a trajectory  $w_r \in \mathcal{B}_{T_r}$  by considering the extended trajectory  $w_{\text{ini}} \wedge w_r$ , where  $w_{\text{ini}} \in \mathcal{B}_{T_{\text{ini}}}$  is a suitably chosen initial trajectory. Assuming that the length  $T_{\text{ini}}$  of  $w_{\text{ini}}$  is greater than or equal to the lag  $\mathbf{l}(\mathcal{B})$  of the system, the state  $x(T_{\text{ini}} + 1)$  of a minimal input/state/output representation of  $\mathcal{B}$  is uniquely determined at time  $T_{\text{ini}} + 1$ . This state serves as an initial condition for the trajectory  $w_r$ .

**Lemma 1** (Initial condition). *Let  $\mathcal{B}(A, B, C, D)$  be a minimal input/state/output representation and let  $H$  be the impulse response of  $\mathcal{B} \in \mathcal{L}^{\mathbf{w}}$ . Then for all  $w_{\text{ini}} \in (\mathbb{R}^{\mathbf{w}})^{T_{\text{ini}}}$ , with  $T_{\text{ini}} \geq \mathbf{l}(\mathcal{B})$ ,*

$$w_{\text{ini}} \wedge \text{col}(u_r, y_r) \in \mathcal{B}_{T_{\text{ini}}+T_r} \implies \text{there is unique } x_{\text{ini}} \in \mathbb{R}^{\mathbf{n}(\mathcal{B})}, \text{ such that } y_r = \mathcal{O}_{T_r}(A, C)x_{\text{ini}} + \mathcal{T}_{T_r}(H)u_r. \quad (7)$$

*Proof.* Given an initial trajectory  $w_{\text{ini}}$ , we need to prove the existence of a unique initial vector  $x_{\text{ini}} \in \mathbb{R}^{\mathbf{n}(\mathcal{B})}$ , such that (7) holds, for all  $u_r \in (\mathbb{R}^{\mathbf{m}})^{T_r}$ . We do this constructively. Let  $\text{col}(u_{\text{ini}}, y_{\text{ini}})$  be an input/output partitioning of  $w_{\text{ini}}$ . Note that  $w_{\text{ini}}$  is not free:

$$w_{\text{ini}} \wedge \text{col}(u_r, y_r) \in \mathcal{B}_{T_{\text{ini}}+T_r} \implies w_{\text{ini}} \in \mathcal{B}_{T_{\text{ini}}}.$$

Since  $w_{\text{ini}}$  is a trajectory of  $\mathcal{B}_{T_{\text{ini}}}$ , it follows from (6), that there is an initial state  $x(1)$ , such that

$$y_{\text{ini}} = \mathcal{O}_{T_{\text{ini}}}(C, A)x(1) + \mathcal{T}_{T_{\text{ini}}}(H)u_{\text{ini}}. \quad (8)$$

Moreover, the assumptions that  $\mathcal{B}(A, B, C, D)$  is a minimal representation and  $T_{\text{ini}} \geq \mathbf{l}(\mathcal{B})$  imply that the extended observability matrix  $\mathcal{O}_{T_{\text{ini}}}(C, A)$  is full column rank. Therefore, the system of equations (8) has a unique solution  $x(1)$ . The initial condition  $x_{\text{ini}}$  is equal to the state  $x(T_{\text{ini}} + 1)$ , i.e.,

$$x_{\text{ini}} = x(T_{\text{ini}} + 1) = A^{T_{\text{ini}}}x(1) + \begin{bmatrix} A^{T_{\text{ini}}-1}B & A^{T_{\text{ini}}-2}B & \dots & AB & B \end{bmatrix} u_{\text{ini}}. \quad (9)$$

The uniqueness of  $x_{\text{ini}}$  follows from the uniqueness of  $x(1)$ .  $\square$

Lemma 1 shows that assuming  $w_{\text{ini}} \in \mathcal{B}_{T_{\text{ini}}}$  and  $T_{\text{ini}} \geq \mathbf{I}(\mathcal{B})$ , a trajectory  $w_{\text{r}} \in \mathcal{B}_{T_{\text{r}}}$ , such that  $w_{\text{ini}} \wedge w_{\text{r}} \in \mathcal{B}_{T_{\text{ini}}+T_{\text{r}}}$ , has a uniquely specified initial condition and is therefore unique, given its input component  $u_{\text{r}}$ . Moreover, the proof gives an explicit expression (8–9) for an initial state  $x_{\text{ini}}$  in terms of the initial trajectory  $w_{\text{ini}}$ , the input/state/output representation parameters  $A, B, C, D$ , and the impulse response  $H$ . This expression is used in Section 6.1 for the solution of the linear quadratic tracking problem by input/state/output representation.

### 3.2 Construction of trajectories

Next, we consider an important result for the solution of the data-driven simulation and control problems, related to the construction of new trajectories of the system from a known one. Assume that  $w_{\text{d}} \in (\mathbb{R}^w)^T$  is a trajectory of  $\mathcal{B} \in \mathcal{L}^w$ . The subscript d stands for ‘data’ and aims to distinguish the particular observed trajectory  $w_{\text{d}}$  from a general one. Due to the linearity and time-invariance of  $\mathcal{B}$ , the  $T - t + 1$  columns of the Hankel matrix  $\mathcal{H}_t(w_{\text{d}})$ , where  $t \leq T$ , are  $t$  samples long trajectories of  $\mathcal{B}$ . Therefore, any linear combination of the columns of  $\mathcal{H}_t(w_{\text{d}})$  is also a  $t$ -samples long trajectory of  $\mathcal{B}$ , i.e.,

$$\mathcal{H}_t(w_{\text{d}})g \in \mathcal{B}_t, \quad \text{for all } g \in \mathbb{R}^{T-t+1}.$$

The following lemma shows that under additional assumptions on  $w_{\text{d}}$  and  $\mathcal{B}$ , for every trajectory  $w \in \mathcal{B}_t$ , there is a vector  $g$ , such that  $w = \mathcal{H}_t(w_{\text{d}})g$ .

**Lemma 2** (Fundamental lemma (Willems *et al.* 2005)). *Let*

- $w_{\text{d}} = \text{col}(u_{\text{d}}, y_{\text{d}})$  be a  $T$  samples long trajectory of a linear time-invariant system  $\mathcal{B}$ , i.e.,  $w_{\text{d}} \in \mathcal{B}_T$ ;
- the system  $\mathcal{B}$  be controllable; and
- the input  $u_{\text{d}}$  be persistently exciting of order  $t + \mathbf{n}(\mathcal{B})$ .

*Then any  $t$  samples long trajectory  $w = \text{col}(u, y)$  of  $\mathcal{B}$  can be written as a linear combination of the columns of  $\mathcal{H}_t(w_{\text{d}})$  and any linear combination  $\mathcal{H}_t(w_{\text{d}})g$ , where  $g \in \mathbb{R}^{T-t+1}$ , is a trajectory of  $\mathcal{B}$ , i.e.,*

$$\text{colspan}(\mathcal{H}_t(w_{\text{d}})) = \mathcal{B}_t.$$

Under the assumptions of the fundamental lemma, computing a trajectory  $w$  of  $\mathcal{B}_t$  with certain specified properties, e.g.,  $w$  corresponds to a given input  $u$  and initial trajectory  $w_{\text{ini}}$ , amounts to choosing a vector  $g$  that imposes those properties through  $w = \mathcal{H}_t(w_{\text{d}})g$ . For example, choosing  $g$ , such that  $w_{\text{ini}} \wedge \text{col}(u, y) \in \mathcal{B}_{T_{\text{ini}}+t}$  requires solving a linear system of equations for  $g$  with right-hand-side  $w_{\text{ini}}$  and  $u$ .

## 4 Data-driven simulation

In this section, we consider the simulation problem: given a system  $\mathcal{B}$  with an input/output partition  $w = \text{col}(u, y)$  of the variables, an input signal  $u_r$ , and an initial condition; find the response  $y_r$  of  $\mathcal{B}_t$  to  $u_r$  and the given initial condition.

### 4.1 Problem formulation

Classically, the system  $\mathcal{B}$  and implicitly the input/output partition are given by an input/state/output, transfer function, or impulse response representation. The input/state/output representation is used most often, because it allows a convenient specification of the initial condition by an initial state vector  $x_{\text{ini}}$  and because of the good numerical linear algebra methods developed in the state space setting. Consequently, in what follows, we refer to the simulation problem with a given input/state/output representation of the system as the ‘classical simulation problem’.

**Problem 3** (Classical simulation). Given

- an input/state/output representation  $\mathcal{B}(A, B, C, D)$  of a linear time-invariant system  $\mathcal{B}$ ,
- an input signal  $u_r = (u_r(1), \dots, u_r(T_r)) \in (\mathbb{R}^m)^{T_r}$ , and
- an initial condition  $x_{\text{ini}} \in \mathbb{R}^n$ ,

find the response  $y_r$  of  $\mathcal{B}(A, B, C, D)$  to the input  $u_r$  and the initial condition  $x_{\text{ini}}$ .

The data-driven simulation problem is defined similarly to the classical simulation problem with the differences that the system  $\mathcal{B}$  and the initial conditions are specified implicitly by given trajectories.

**Problem 4** (Data-driven simulation). Given

- a trajectory  $w_d = (w_d(1), \dots, w_d(T)) \in (\mathbb{R}^w)^T$  of a linear time-invariant system  $\mathcal{B}$ ,
- an input signal  $u_r = (u_r(1), \dots, u_r(T_r)) \in (\mathbb{R}^m)^{T_r}$ , and
- an initial trajectory  $w_{\text{ini}} = (w_{\text{ini}}(1), \dots, w_{\text{ini}}(T_{\text{ini}})) \in (\mathbb{R}^w)^{T_{\text{ini}}}$ , where  $w_{\text{ini}} \in \mathcal{B}_{T_{\text{ini}}}$ ,

find a response  $y$  of  $\mathcal{B}$  to the input  $u_r$ , such that

$$w_{\text{ini}} \wedge \text{col}(u_r, y) \in \mathcal{B}_{T_{\text{ini}}+T_r}. \quad (10)$$

*Note 5* (Uniqueness of the response  $y$ ). Equation (10) does not necessarily define a unique  $y$ , i.e., in Problem 4 we do not impose that the response  $y$  is unique. In most cases, however, uniqueness is desirable. As proven in Lemma 1, the additional assumption  $T_{\text{ini}} \geq \mathbf{I}(\mathcal{B})$  imposes uniqueness.

It is assumed, as part of the problem formulation, that  $w_d$  and  $w_{\text{ini}}$  are trajectories of a linear time-invariant system  $\mathcal{B}$ . Since  $w_d$  and  $w_{\text{ini}}$  are of finite length and the order of  $\mathcal{B}$  is not a priori restricted, the assumptions  $w_d \in \mathcal{B}_T$

and  $w_{\text{ini}} \in \mathcal{B}_{T_{\text{ini}}}$  can be made without loss of generality. We do need restrictive assumptions, however, in order to guarantee that  $w_{\text{d}}$  uniquely specifies  $\mathcal{B}$  (identifiability). If, in addition, we insist on a unique solution to Problem 4,  $w_{\text{ini}}$  should specify a unique initial condition. Sufficient conditions for identifiability can be derived from Lemma 1, see (Markovsky *et al.* 2006a, Theorem 8.16), and a uniqueness condition is given in Lemma 2.

Assuming that  $\mathcal{B}$  is identifiable from  $w_{\text{d}}$ , standard identification algorithms can be used to find an explicit representation of  $\mathcal{B}$ . This suggests the following solution of the data-driven simulation problem that reduces it to the classical simulation problem:

1. identification  $w_{\text{d}} \mapsto (A, B, C, D)$ ,
2. observer design  $(w_{\text{ini}}, (A, B, C, D)) \mapsto x_{\text{ini}}$ ,
3. classical simulation  $(u_{\text{r}}, x_{\text{ini}}, (A, B, C, D)) \mapsto y$ .

The question occurs:

Can we find a response  $y$  such that (10) holds without deriving an explicit representation of  $\mathcal{B}$ ?

As shown in the following section, the answer is positive. Moreover, the main idea for doing data-driven simulation can be used as well for data-driven control. This is explained in Sections 5 and 6.

## 4.2 Data-driven simulation algorithm

Consider given  $w_{\text{d}}$ ,  $u_{\text{r}}$ , and  $w_{\text{ini}}$ . Under the assumptions of the fundamental lemma,

$$w_{\text{ini}} \wedge \text{col}(u_{\text{r}}, y) \in \mathcal{B}_{T_{\text{ini}}+T_{\text{r}}} \iff \exists g, \text{ such that } \mathcal{H}_{T_{\text{ini}}+T_{\text{r}}}(w_{\text{d}})g = \text{col}(w_{\text{ini}}, \text{col}(u_{\text{r}}, y)).$$

The equation

$$\mathcal{H}_{T_{\text{ini}}+T_{\text{r}}}(w_{\text{d}})g = \text{col}(w_{\text{ini}}, \text{col}(u_{\text{r}}, y)) \tag{11}$$

is a linear system for the unknowns  $g$  and  $y$ . The equations with right-hand-side  $y$  can be used to define  $y$  for given  $g$ . The remaining equations impose restrictions on  $g$ , determined by  $w_{\text{ini}}$  and  $u_{\text{r}}$ . This suggests the following generic data-driven simulation algorithm:

1. compute any solution  $g$  of the equations in (11) with right-hand-side  $w_{\text{ini}}$  and  $u_{\text{r}}$ ,
2. define  $y$  from the remaining equations in (11) by substituting the computed  $g$  on step 1.

In order to make the algorithm more concrete, define

$$U := \mathcal{H}_{T_{\text{ini}}+T_{\text{r}}}(u_{\text{d}}) \quad \text{and} \quad Y := \mathcal{H}_{T_{\text{ini}}+T_{\text{r}}}(y_{\text{d}})$$

and the partitionings

$$U =: \begin{bmatrix} U_{\text{p}} \\ U_{\text{f}} \end{bmatrix} \begin{matrix} mT_{\text{ini}} \\ mT_{\text{r}} \end{matrix} \quad \text{and} \quad Y =: \begin{bmatrix} Y_{\text{p}} \\ Y_{\text{f}} \end{bmatrix} \begin{matrix} pT_{\text{ini}} \\ pT_{\text{r}} \end{matrix}.$$

The blocks  $U_p$  and  $Y_p$  are referred to as the ‘past’ (of the computed response) and are used to set up the initial conditions. The blocks  $U_f$  and  $Y_f$  are referred to as the ‘future’ and are used for computing the desired response  $y$ . With this notation steps 1 and 2 above can be formulated as the following Algorithm 1.

---

**Algorithm 1** Data-driven simulation.

---

**Input:**  $w_d \in (\mathbb{R}^w)^T$ ,  $u_r \in (\mathbb{R}^m)^{T_r}$ , and  $w_{ini} \in (\mathbb{R}^w)^{T_{ini}}$ .

1: Compute a solution of

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u_r \end{bmatrix}. \quad (12)$$

2: Compute

$$Y_f g =: y. \quad (13)$$

**Output:**  $y$ .

---

**Proposition 6.** *Assume that*

- *the data generating system  $\mathcal{B}$  is controllable, and*
- *the input component  $u_d$  of the trajectory  $w_d$  is persistently exciting of order  $T_{ini} + T_r + \mathbf{n}(\mathcal{B})$ .*

*Then for any  $u_r \in (\mathbb{R}^m)^{T_r}$ , the system of equations (12) is compatible and for any particular solution  $g$ , the output  $y := Y_f g$  of Algorithm 1 is such that (10) holds. If in addition,  $T_{ini} \geq \mathbf{l}(\mathcal{B})$ , then  $y = y_r$  is the unique signal that satisfies (10).*

*Proof.* Under the assumptions of the proposition, we can apply Lemma 2. It follows that  $\text{colspan}(\mathcal{H}_{T_{ini}+T_r}(w_d)) = \mathcal{B}_{T_{ini}+T_r}$ . Since by assumption  $w_{ini} \in \mathcal{B}_{T_{ini}}$ , for any  $u_r \in (\mathbb{R}^m)^{T_r}$ , there is  $y \in (\mathbb{R}^p)^{T_r}$ , such that (10) holds. Then there is in general a nonunique vector  $g$ , such that (11) holds. The system of equations (12) on step 1 of Algorithm 1 consists of a subset of the equations of (11), which proves that (12) is compatible. Assume now that  $\bar{g}$  is a particular solution of (12) and  $y$  is defined as in (13). Then by Lemma 2,

$$\mathcal{H}_{T_{ini}+T_r}(w_d)\bar{g} = \text{col}(w_{ini}, \text{col}(u_r, y)) \in \mathcal{B}_{T_{ini}+T_r},$$

so that (10) holds.

Under the additional assumption that  $T_{ini} \geq \mathbf{l}(\mathcal{B})$ , we can apply Lemma 1. It follows that  $w_{ini}$  and  $u_r$  determine a unique output  $y_r$ , such that (10) holds. Then the signal  $y$ , computed by Algorithm 1 is unique and is equal to  $y_r$ .  $\square$

*Note 7* (‘Noisy’ data). The choice of the particular solution  $g$  on step 1 of Algorithm 1 is irrelevant as long as the assumptions of Proposition 6 are satisfied. However, if one or more of the assumptions are not satisfied, then (12) may have no solution. In this case, one can still execute Algorithm 1 by computing an approximate solution for  $g$ , aiming at an approximate response of the system. The question of which notion of approximate solution one should use depends on additional assumptions (or prior knowledge) about the data generating mechanism. Results for the case when  $u_d$  is exact and  $y_d$  is noisy (output error setup) are given in (Markovsky *et al.* 2006b).

*Note 8 (Recursive implementation).* The persistency of excitation for  $u_d$ , required by Algorithm 1 is of order  $T_r + T_{\text{ini}} + \mathbf{n}(\mathcal{B})$ . Typically  $T_{\text{ini}}$  and  $\mathbf{n}(\mathcal{B})$  are small compared to the length  $T$  of the given trajectory, however,  $T_r$  might be big. In (Markovsky *et al.* 2005a) a refinement of the data-driven simulation algorithm is presented (in the special cases of impulse response and free response computation) that requires persistency of excitation of order  $1 + T_{\text{ini}} + \mathbf{n}(\mathcal{B})$ . The idea is to construct the response  $y$  iteratively by weaving together consecutive patches of  $y$ . Due to lack of space, we do not present iterative versions of the algorithms in this paper.

Next, we describe three special cases of the generic data-driven simulation algorithm that are of independent interest and are used in the solution of data-driven control problems. The observations made in Notes 7 and 8 can be repeated verbatim for the special cases of Algorithms 1, presented below.

### 4.3 Special case $u = 0$ : zero input response

The columns of an extended observability matrix  $\mathcal{O}_{T_r}(A, C)$  of  $\mathcal{B}$  are  $T_r$  samples long free responses of  $\mathcal{B}$ . Therefore, by using a data-driven simulation algorithm we can construct an observability matrix directly from data. In Algorithm 2 we compute  $j := T - T_r - T_{\text{ini}} + 1$  free responses  $f_1, \dots, f_j$  and do a rank revealing factorisation of the matrix  $F := \begin{bmatrix} f_1 & \dots & f_j \end{bmatrix}$ , i.e., factor  $F$  as  $\mathcal{O}X_{\text{ini}}$ , where  $\mathcal{O}$  is full column rank and  $X_{\text{ini}}$  is full row rank. Due to the persistency of excitation assumption of Proposition 6, the number of linearly independent free responses computed in this way is  $\mathbf{n}(\mathcal{B})$ .

---

**Algorithm 2** Compute an observability matrix.

---

**Input:**  $w_d \in (\mathbb{R}^w)^T$ ,  $T_{\text{ini}} \in \mathbb{N}$ , and  $T_r \in \mathbb{N}$ .

1: Compute a solution of

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} G = \begin{bmatrix} U_p \\ Y_p \\ 0_{mT_r \times j} \end{bmatrix}.$$

2: Compute

$$Y_f G =: F.$$

3: Compute a rank revealing factorisation  $F = \mathcal{O}X_{\text{ini}}$ .

**Output:**  $\mathcal{O}$ .

---

**Proposition 9.** *If the assumptions of Proposition 6 are satisfied and  $T_r \geq \mathbf{l}(\mathcal{B})$ , the matrix  $\mathcal{O}$ , computed by Algorithm 2, is an observability matrix of  $\mathcal{B}$  with  $T_r$  block rows, corresponding to a minimal input/state/output representation  $\mathcal{B}(A, B, C, D)$  of  $\mathcal{B}$ , i.e.,  $\mathcal{O} = \mathcal{O}_{T_r}(A, C)$ .*

*Proof.* From Proposition 6 it follows that every column of the matrix  $F$ , computed on step 2 of Algorithm 2, is a free response of  $\mathcal{B}$ . Moreover, from (Willems *et al.* 2005, Corollary 2) and the assumption  $T_r \geq \mathbf{l}(\mathcal{B})$  it follows that  $\text{rank}(F) = \mathbf{n}(\mathcal{B})$ . The statement of the proposition follows from  $\text{rank}(\mathcal{O}) = \text{rank}(F) = \mathbf{n}(\mathcal{B})$ .  $\square$

#### 4.4 Special case $w_{\text{ini}} = 0$ : zero initial condition response

Define the zero initial conditions subbehaviour  $\mathcal{B}_0$  of  $\mathcal{B}$  as follows

$$\mathcal{B}_{0,T_r} := \left\{ w_f \in (\mathbb{R}^w)^{T_r} \mid \underbrace{(0, \dots, 0)}_{T_{\text{ini}}} \wedge w_f \in \mathcal{B}_{T_{\text{ini}}+T_r}, \text{ for all } T_{\text{ini}} \in \mathbb{N} \right\}. \quad (14)$$

The set  $\mathcal{B}_0$  consists of all trajectories of  $\mathcal{B}$  that when preceded with an arbitrary number of zero samples are still trajectories of  $\mathcal{B}$ .  $\mathbf{I}(\mathcal{B})$  or more leading zero samples specify zero initial conditions, so  $\mathcal{B}_0$  is indeed the subspace of  $\mathcal{B}$  consisting of all zero initial conditions trajectories. Note that  $\dim(\mathcal{B}_{0,T_r}) = mT_r$ , so we can compute a basis for  $\mathcal{B}_{0,T_r}$  directly from data, by computing  $mT_r$  linearly independent zero initial condition responses. In Algorithm 3, we compute  $j := T - T_r - T_{\text{ini}} + 1$  zero initial condition responses  $w_{0,1}, \dots, w_{0,j}$ , because the persistency of excitation of  $u_d$  guarantees that among them there are exactly  $mT_r$  linearly independent one. Then a basis for  $\mathcal{B}_{0,T_r}$  can be computed from the matrix  $W_0 := \begin{bmatrix} w_{0,1} & \cdots & w_{0,j} \end{bmatrix}$  by doing a rank revealing factorisation.

---

**Algorithm 3** Compute a basis for the zero initial conditions subbehaviour.

---

**Input:**  $w_d \in (\mathbb{R}^w)^T$ ,  $T_{\text{ini}} \in \mathbb{N}$ , and  $T_r \in \mathbb{N}$ .

1: Compute a solution of

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} G = \begin{bmatrix} 0_{mT_{\text{ini}} \times j} \\ 0_{pT_{\text{ini}} \times j} \\ \mathcal{H}_{T_r}(u_d) \end{bmatrix}.$$

2: Compute

$$Y_f G =: Y_0.$$

3: Define  $W_0 := \begin{bmatrix} \text{col}(u_{0,1}, y_{0,1}) & \cdots & \text{col}(u_{0,j}, y_{0,j}) \end{bmatrix}$ , where  $u_{0,i}$  is the  $i$ th column of  $\mathcal{H}_{T_r}(u_d)$  and  $y_{0,i}$  is the  $i$ th column of  $Y_0$ .

4: Compute a rank revealing factorisation  $W_0 = B_0 L$

**Output:**  $B_0$ .

---

**Proposition 10.** *If the assumptions of Proposition 6 are satisfied, the columns of the matrix  $B_0$ , computed by Algorithm 3, form a basis for the zero initial conditions subbehaviour  $\mathcal{B}_{0,T_r}$  of  $\mathcal{B}_{T_r}$ .*

*Proof.* From Proposition 6, it follows that the columns of  $Y_0$  are zero initial condition responses of  $\mathcal{B}$ . By construction the corresponding inputs are the columns of  $\mathcal{H}_{T_r}(u_d)$ , so that  $\text{colspan}(W_0) \subset \mathcal{B}_{0,T_r}$ . The persistency of excitation assumption implies  $\text{rank}(\mathcal{H}_{T_r}(u_d)) = mT_r$ . Therefore  $\text{rank}(W_0) = mT_r$ . But  $\dim(\mathcal{B}_{0,T_r}) = mT_r$ , and consequently

$$\text{colspan}(B_0) = \text{colspan}(W_0) = \mathcal{B}_{0,T_r}.$$

□

#### 4.5 Special case $w_{\text{ini}} = 0$ , $u = I\delta$ : impulse response

Let  $e_i$  be the  $i$ th column of the  $m \times m$  identity matrix  $I_m$ . Choosing zero initial conditions  $w_{\text{ini}} = 0$  and pulse input  $u_i(1) = e_i$ ,  $u_i(t) = 0$ , for  $t > 1$ , we find the impulse response of  $\mathcal{B}$ . Note that this procedure constitutes a system identification method, because a representation of  $\mathcal{B}$  is derived from data.

---

**Algorithm 4** Compute the impulse response.

---

**Input:**  $w_d \in (\mathbb{R}^w)^T$ ,  $T_{\text{ini}} \in \mathbb{N}$ , and  $T_r \in \mathbb{N}$ .

1: Compute a solution of

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} G = \begin{bmatrix} 0_{m T_{\text{ini}} \times m} \\ 0_{p T_{\text{ini}} \times m} \\ \text{col}(I_m, 0_{m(T_r-1) \times m}) \end{bmatrix}. \quad (15)$$

2: Compute

$$Y_f G =: H.$$

**Output:**  $H$ .

---

**Proposition 11.** *If the assumptions of Proposition 6 are satisfied, the matrix  $H$ , computed by Algorithm 4, contains the first  $T_r$  samples of the impulse response of  $\mathcal{B}$ .*

*Proof.* Follows trivially from Proposition 6 and the structure of the right-hand-side of (15).  $\square$

## 5 Output matching control

Simulation can be viewed as ‘input matching’: determine the output component of a system trajectory, corresponding to a given input component and initial condition. By reversing the roles of the input and the output, we have an ‘output matching’ problem: determine the input component of a system trajectory, given an output component and initial condition. This is a control problem, which we call ‘output matching control’.

The output matching problem is a special (singular) case of the output tracking problem, when the input does not enter into the cost function and the desired output can be achieved exactly. Alternatively, the output matching problem can be viewed as a special dead-beat control problem, when the initial condition of the system is given. In Section 5.2, we present a model-based solution and in Section 5.3 we present a data-driven solution to this problem.

### 5.1 Problem formulation

We call the output matching problem ‘classical’, if the to-be-controlled system is given by an input/state/output representation.

**Problem 12** (Classical output matching). Given

- an input/state/output representation  $\mathcal{B}(A, B, C, D)$  of a linear time-invariant system  $\mathcal{B}$ ,
- a reference response  $y_r = (y_r(1), \dots, y_r(T_r)) \in (\mathbb{R}^p)^{T_r}$ , and
- an initial condition  $x_{\text{ini}} \in \mathbb{R}^n$ ,

find an input  $u \in (\mathbb{R}^m)^{T_r}$ , such that the response of  $\mathcal{B}$  to  $u$  and initial condition  $x_{\text{ini}}$  is  $y_r$ .

In the data-driven output matching control problem, the to-be-controlled system and the initial condition are implicitly specified by given trajectories of the system.

**Problem 13** (Data-driven output matching). Given

- a trajectory  $w_d = (w_d(1), \dots, w_d(T)) \in (\mathbb{R}^w)^T$  of  $\mathcal{B}$ ,
- a reference response  $y_r = (y_r(1), \dots, y_r(T_r)) \in (\mathbb{R}^p)^{T_r}$ , and
- an initial trajectory  $w_{\text{ini}} = (w_{\text{ini}}(1), \dots, w_{\text{ini}}(T_{\text{ini}})) \in (\mathbb{R}^w)^{T_{\text{ini}}}$ , where  $w_{\text{ini}} \in \mathcal{B}_{T_{\text{ini}}}$ ,

find an input  $u \in (\mathbb{R}^m)^{T_r}$ , such that

$$w_{\text{ini}} \wedge (u, y_r) \in \mathcal{B}_{T_{\text{ini}}+T_r}. \quad (16)$$

## 5.2 Model-based output matching

Figure 1 shows the structure of a feedforward output matching controller.  $\mathcal{B}^{-1}$  denotes a right inverse of  $\mathcal{B}$ , i.e., a linear time-invariant system with  $p$  inputs and  $m$  outputs, such that the series connection of  $\mathcal{B}^{-1}$  and  $\mathcal{B}$  is  $I_p$ . Note that the existence of a right inverse requires that the given system  $\mathcal{B}$  has more inputs than outputs.

**Proposition 14** (Model-based output matching control). *If a causal right inverse system  $\mathcal{B}^{-1}$  of  $\mathcal{B}$  exists, then for any reference signal  $y_r \in (\mathbb{R}^p)^{T_r}$  and initial condition  $x_{\text{ini}} \in \mathbb{R}^n$ , there is a control signal  $u$ , such that (16) holds.*

*Proof.* If a causal right inverse  $\mathcal{B}^{-1}$  exists, the feedforward controller, given on Figure 1 solves the output matching problem for any  $y_r \in (\mathbb{R}^p)^{T_r}$  and  $x_{\text{ini}} \in \mathbb{R}^n$ .  $\square$

We do not consider further conditions for existence of a causal right inverse system  $\mathcal{B}^{-1}$  nor algorithms for computing its representation from a given representation of  $\mathcal{B}$ . The reader is referred to (Sain and Massey 1969) for related results. Our concern next is an algorithm for data-driven output matching control.

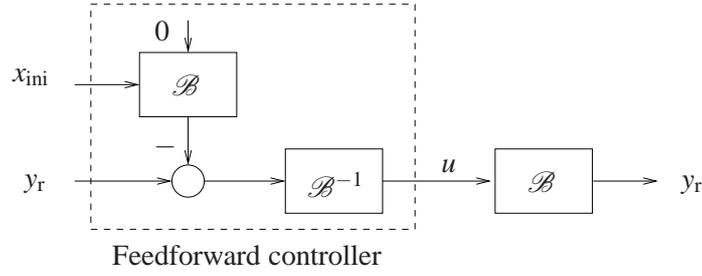


Figure 1: Structure of the output matching controller.

### 5.3 Data-driven output matching algorithm

Consider given  $w_d$ ,  $y_r$ , and  $w_{ini}$ . Under the assumptions of the fundamental lemma,

$$w_{ini} \wedge \text{col}(u, y_r) \in \mathcal{B}_{T_{ini}+T_r} \iff \exists g, \text{ such that } \mathcal{H}_{T_{ini}+T_r}(w_d)g = \text{col}(w_{ini}, \text{col}(u, y_r)).$$

The equation

$$\mathcal{H}_{T_{ini}+T_r}(w_d)g = \text{col}(w_{ini}, \text{col}(u, y_r)) \quad (17)$$

now has as unknowns  $g$  and  $u$ . Therefore, we determine  $g$  from the equations in (17) with right-hand-side  $w_{ini}$  and  $y_r$ , and define  $u$  from the remaining equations. Using the notation  $U_p$ ,  $U_f$ ,  $Y_p$ ,  $Y_f$ , defined in Section 4.2, we obtain Algorithm 5.

---

**Algorithm 5** Data-driven output matching.

---

**Input:**  $w_d \in (\mathbb{R}^w)^T$ ,  $y_r \in (\mathbb{R}^p)^{T_r}$ , and  $w_{ini} \in (\mathbb{R}^w)^{T_{ini}}$ .

1: Compute a solution of

$$\begin{bmatrix} U_p \\ Y_p \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ y_r \end{bmatrix}. \quad (18)$$

2: Compute

$$U_f g =: u.$$

**Output:**  $u$ .

---

**Proposition 15.** *If the assumptions of Proposition 6 are satisfied and the zero initial conditions subbehaviour  $\mathcal{B}_0$  of  $\mathcal{B}$ , see (14), satisfies the property  $\mathcal{B}_{0,1} = \mathbb{R}^m$ , the vector  $u$  computed by Algorithm 5 is such that (16) holds.*

*Proof.* Consider an input/state/output representation  $\mathcal{B}(A, B, C, D)$  of  $\mathcal{B}$  and let  $H = (H(0), H(1), \dots)$  be the impulse response of  $\mathcal{B}$ . Under the assumptions of Proposition 6,  $w_{ini}$  defines a unique initial state  $x_{ini}$ . Then (18) is compatible if and only if

$$y_r - \mathcal{O}_{T_r}(A, C)x_{ini} = \mathcal{T}_{T_r}(H)u \quad (19)$$

is compatible. A necessary and sufficient condition for compatibility of (19) is that the matrix  $\mathcal{T}_{T_r}(H)$  is full row rank. Due to the lower triangular Toeplitz structure of  $\mathcal{T}_{T_r}(H)$ , it is easy to see that the full row rank condition for  $\mathcal{T}_{T_r}(H)$

is equivalent to a full row rank condition for  $H(0) = D$ . The statement of the proposition follows from

$$D \text{ is full row rank} \iff \mathcal{B}_{0,1} = \mathbb{R}^m.$$

□

*Example 16 (Unbounded input).* Consider the linear time-invariant system  $\mathcal{B}$  of order  $n = 2$ , with  $m = 1$  input and  $p = 1$  output, induced by the transfer function

$$H(z) = \frac{(z - 0.7847)(z + 1.17)}{z^2 - 1.615z + 0.6972}.$$

The trajectory  $w_d$  is a random one with  $T = 200$  samples, the reference signal is  $y_r = 0$  with horizon  $T_r = 30$ , and the initial trajectory is  $w_{\text{ini}} = (1, 1)$ . Figure 2 shows the input  $u$ , computed by Algorithm 5. The assumptions of Proposition 15 are satisfied and we experimentally verify that the input  $u$  achieves the reference output  $y_r$ . In the particular experiment, however,  $u$  grows unbounded as  $t$  increases. The reason for this is that the plant  $\mathcal{B}$  is non-minimum phase (has zero at  $-1.17$ ), and the output matching controller internally computes the response of the inverse system  $\mathcal{B}^{-1}$ .

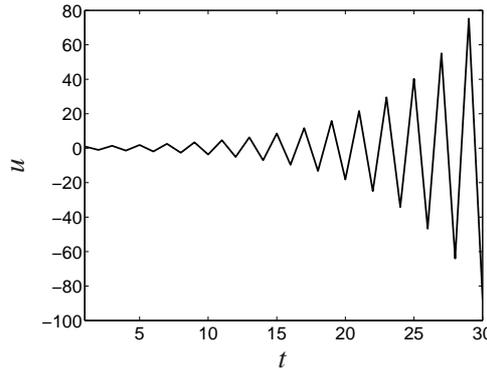


Figure 2: An input  $u$  that achieves the reference output  $y_r$  in the output matching example.

In the next section, we consider a more general tracking problem—follow a reference trajectory  $w_r$  by trading-off errors in both the input and the output. In the infinite horizon case, the latter problem avoids the instability issue, pointed out in Example 16.

## 6 Linear quadratic tracking

In a linear quadratic tracking problem the objective is to choose a trajectory  $w = (u, y)$  of a linear time-invariant system  $\mathcal{B}$  that is as close as possible to a given reference trajectory  $w_r \in (\mathbb{R}^w)^{T_r}$ . More precisely, in a finite-horizon case, the objective is to minimise over  $w \in \mathcal{B}_{T_r}$  the cost function

$$\|w_r - w\|_{\Phi}^2 := \sum_{t=1}^{T_r} (w_r(t) - w(t))^{\top} \Phi (w_r(t) - w(t)), \quad (20)$$

where  $\Phi \in \mathbb{R}^{w \times w}$  is a given positive definite weight matrix. In the special case when the reference trajectory is the zero trajectory, the tracking problem is referred to as the *regulation problem*. In this case, the optimal tracking, aiming solely at minimising the criterion (20) over all trajectories  $w$  of  $\mathcal{B}$ , has a trivial solution—the zero trajectory. The regulation problem is meaningful, when a nonzero initial condition is specified. Therefore, we introduce initial conditions specification in the general tracking problem.

The classical formulation of the tracking problem assumes that an input/state/output representation of the system  $\mathcal{B}$  is given. In the context of data-driven tracking, we start instead from a given trajectory  $w_d \in (\mathbb{R}^w)^T$  of  $\mathcal{B}$ , and aim at finding an optimal trajectory without explicitly computing a representation (in particular an input/state/output representation) of  $\mathcal{B}$ .

**Problem 17** (Linear quadratic tracking). Given

- a trajectory  $w_d = (w_d(1), \dots, w_d(T)) \in (\mathbb{R}^w)^T$  of a linear time-invariant system  $\mathcal{B}$ ,
- a reference trajectory  $w_r = (w_r(1), \dots, w_r(T_r)) \in (\mathbb{R}^w)^{T_r}$ ,
- an initial trajectory  $w_{\text{ini}} = (w_{\text{ini}}(1), \dots, w_{\text{ini}}(T_{\text{ini}})) \in (\mathbb{R}^w)^{T_{\text{ini}}}$ , where  $w_{\text{ini}} \in \mathcal{B}_{T_{\text{ini}}}$ , and
- a positive definite matrix  $\Phi \in \mathbb{R}^{w \times w}$ ,

find a trajectory  $w^* \in \mathcal{B}$  that minimises the cost function  $\|w_r - w\|_{\Phi}^2$  and has as a prefix the initial trajectory  $w_{\text{ini}}$ , i.e., solve the problem

$$\min_w \|w_r - w\|_{\Phi}^2 \quad \text{subject to} \quad w_{\text{ini}} \wedge w \in \mathcal{B}_{T_{\text{ini}}+T_r}.$$

In the following we propose three solutions to Problem 17. The first one computes parameters  $A, B, C, D$  of an input/state/output representation of the data generating system and then proceeds by solving a classical tracking problem, using standard linear quadratic control techniques. The second solution computes the impulse response of the system and then solves the linear quadratic tracking problem as a weighted least squares problem. The third solution for solving Problem 17 is completely data-driven. It is not based on any particular representation of the plant, however, as in the solution using the impulse response, one of the steps is a solution of a large weighted least squares problem.

## 6.1 Algorithm using input/state/output representation

The classical but indirect solution of Problem 17 is to compute first an input/state/output representation  $\mathcal{B}(A, B, C, D)$  of  $\mathcal{B}$  from the data  $w_d$  and then using the parameters  $(A, B, C, D)$  to compute the optimal trajectory  $w^*$ . This is the well known model-based approach that we summarise for completeness.

By assumption the data  $w_d$  is an exact trajectory of the unknown system  $\mathcal{B}$ . Therefore, we are dealing with an exact (deterministic) identification problem  $w_d \mapsto \mathcal{B}$ . Under the assumptions of (Markovsky *et al.* 2006a, Theorem 8.16), the system is identifiable and an input/state/output representation of  $\mathcal{B}$  can be computed. Algorithms for computing an

input/state/output representation from exact data are developed in the (deterministic) subspace identification setting, see (Van Overschee and De Moor 1996, Chapter 2) and (Markovsky *et al.* 2006a, Chapter 8).

A procedure for deriving the initial state from the initial trajectory is outlined in the proof of Lemma 1. Once the state space representation and the initial state are available, Problem 17 becomes

$$\min_{x,u,y} \|w_r - \text{col}(u,y)\|_{\Phi}^2 \quad \text{subject to} \quad \begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}, \quad x(1) = x_{\text{ini}}, \quad \text{for } t = 1, \dots, T_r. \quad (21)$$

The solution leads to a difference Riccati equation that depends only on the  $A, B, C, D$  matrices and a backward in time recursion that depends on the reference signal  $w_r$  and the initial condition  $x_{\text{ini}}$ . The formulae for the continuous-time case are given in (Markovsky *et al.* 2002, Theorem 1). For simplicity, next we give the solution for the finite-horizon linear quadratic regulation problem, i.e., for the special case of  $w_r = 0$ .

We can eliminate the variable  $y$  in (21) by substitution. This gives the following linear quadratic regulation problem with complete state information

$$\min_{x,u} \sum_{t=1}^{T_r} \begin{bmatrix} u(t) \\ x(t) \end{bmatrix}^{\top} \begin{bmatrix} I & 0 \\ D & C \end{bmatrix}^{\top} \Phi \begin{bmatrix} I & 0 \\ D & C \end{bmatrix} \begin{bmatrix} u(t) \\ x(t) \end{bmatrix} \quad (22)$$

subject to  $x(t+1) = Ax(t) + Bu(t), \quad x(1) = x_{\text{ini}}, \quad \text{for } t = 1, \dots, T_r.$

Define the partitioning

$$\Phi =: \begin{bmatrix} \Phi_u & \Phi_{uy} \\ \Phi_{yu} & \Phi_y \end{bmatrix}.$$

The solution of (22) is (see, e.g., (Åström and Wittenmark 1997, Theorem 11.1))

$$\begin{aligned} x^*(t+1) &= (A - BL_t)x^*(t), \quad x(1) = x_{\text{ini}}, \\ w^*(t) &= \begin{bmatrix} -L_t \\ C - DL_t \end{bmatrix} x^*(t), \end{aligned} \quad (23)$$

where

$$L_t := (B^{\top} S_{t+1} B + \Phi_u + \Phi_{uy} D + D^{\top} \Phi_{uy}^{\top} + D^{\top} \Phi_y D)^{-1} (B^{\top} S_{t+1} A + \Phi_{uy} C + D^{\top} \Phi_y C) \quad (24)$$

and

$$\begin{aligned} S_t &= A^{\top} S_{t+1} A + C^{\top} \Phi_y C - (B^{\top} S_{t+1} A + \Phi_{uy} C + D^{\top} \Phi_y C)^{\top} \\ &\quad \times (B^{\top} S_{t+1} B + \Phi_u + \Phi_{uy} D + D^{\top} \Phi_{uy}^{\top} + D^{\top} \Phi_y D)^{-1} (B^{\top} S_{t+1} A + \Phi_{uy} C + D^{\top} \Phi_y C), \quad S_{T_r+1} = 0. \end{aligned} \quad (25)$$

The resulting algorithm for solving Problem 17, in the special case  $w_r = 0$ , using an input/state/output representation of the plant is Algorithm 6.

---

**Algorithm 6** Data-driven linear quadratic regulation, using an input/state/output representation.

---

**Input:**  $w_d \in (\mathbb{R}^w)^T$ ,  $w_r \in (\mathbb{R}^p)^{T_r}$ ,  $w_{ini} \in (\mathbb{R}^w)^{T_{ini}}$ , and  $\Phi \in \mathbb{R}^{w \times w}$ ,  $\Phi > 0$ .

- 1:  $(w_d, T_{ini}) \xrightarrow{\text{Identification, e.g., (Markovsky et al. 2006a, Algorithm 8.5)}} (A, B, C, D)$
- 2:  $(w_{ini}, A, B, C, D) \xrightarrow{\text{Observer (5,8,9)}} x_{ini}$
- 3:  $(\Phi, w_r, x_{ini}, A, B, C, D) \xrightarrow{\text{Synthesis (23,24,25)}} w^*$

**Output:**  $w^*$ .

---

## 6.2 Algorithm using impulse response representation

Another approach for solving Problem 17 is to compute the impulse response  $H$  of  $\mathcal{B}$  from the data  $w_d$ , using Algorithm 4, and then using  $H$  to compute  $w^*$ . Let  $\mathcal{B}(A, B, C, D)$  be an input/state/output representation of  $\mathcal{B}$  and let

$$\tilde{\mathcal{O}} := \begin{bmatrix} \text{col}(0_{m \times n}, C) \\ \text{col}(0_{m \times n}, CA) \\ \vdots \\ \text{col}(0_{m \times n}, CA^{T_{ini}+T_r-1}) \end{bmatrix}$$

be an extended observability matrix of  $\mathcal{B}$  with  $T_{ini} + T_r$  block rows. Any zero-input trajectory  $w_{free} = \text{col}(0, f)$  of  $\mathcal{B}$  can be written as  $w_{free} = \tilde{\mathcal{O}}x_{ini}$ , for some  $x_{ini} \in \mathbb{R}^n$ . Define also the augmented Markov parameters

$$\begin{aligned} \tilde{H}(0) &= \text{col}(I_m, H(0)), \quad \text{and} \\ \tilde{H}(t) &= \text{col}(0_m, H(t)), \quad \text{for } t = 1, 2, \dots \end{aligned}$$

Then for any trajectory  $w_{ini} \wedge w \in \mathcal{B}_{T_{ini}+T_r}$  there is a corresponding initial condition  $x_{ini}$  and an input signal  $u_{ini} \wedge u$ , such that

$$\begin{bmatrix} w_{ini} \\ w \end{bmatrix} = \tilde{\mathcal{O}}x(1) + \mathcal{F}_{T_{ini}+T_r}(\tilde{H}) \begin{bmatrix} u_{ini} \\ u \end{bmatrix}. \quad (26)$$

Define the partitionings

$$\tilde{\mathcal{O}} =: \begin{bmatrix} \tilde{\mathcal{O}}_1 \\ \tilde{\mathcal{O}}_2 \end{bmatrix} \quad \text{and} \quad \mathcal{F}_{T_{ini}+T_r}(\tilde{H}) =: \begin{bmatrix} \tilde{T}_{11} & 0 \\ \tilde{T}_{21} & \mathcal{F}_{T_r}(\tilde{H}) \end{bmatrix},$$

that are conformable with the partitionings of  $\text{col}(w_{ini}, w)$  and  $\text{col}(u_{ini}, u)$ . The equations in (26) with left-hand-side  $w_{ini}$

$$w_{ini} = \tilde{\mathcal{O}}_1 x(1) + \tilde{T}_{11} u_{ini} \quad (27)$$

are decoupled from the other equations and can be solved independently in terms of the initial state  $x(1)$  and initial input signal  $u_{ini}$ . The remaining equations

$$w = \tilde{\mathcal{O}}_2 x(1) + \tilde{T}_{21} u_{ini} + \mathcal{F}_{T_r}(\tilde{H}) u$$

specify  $w$  in terms of the input  $u$ . Define

$$w_{free} := \tilde{\mathcal{O}}_2 x(1) + \tilde{T}_{21} u_{ini}. \quad (28)$$

Note that  $w_{\text{free}}$  is the free trajectory of  $\mathcal{B}$ , with prefix  $w_{\text{ini}}$  (or equivalently by the initial condition  $x_{\text{ini}}$ ).

The optimal tracking problem becomes

$$\min_u \|w_r - w_{\text{free}} - \mathcal{T}_T(\tilde{H})u\|_{\tilde{\Phi}}^2,$$

which is a standard weighted least squares problem. Its solution is

$$u^* = (\mathcal{T}_T^\top(\tilde{H})\tilde{\Phi}\mathcal{T}_T(\tilde{H}))^{-1}\mathcal{T}_T^\top(\tilde{H})\tilde{\Phi}(w_r - w_{\text{free}}), \quad (29)$$

where

$$\tilde{\Phi} = \text{diag}(\Phi, \dots, \Phi) \in \mathbb{R}^{T_r w \times T_r w},$$

so that

$$w^* = \mathcal{T}_T(\tilde{H})u^* + w_{\text{free}}. \quad (30)$$

The resulting algorithm for solving Problem 17, using an impulse response representation of the plant, is Algorithm 7.

---

**Algorithm 7** Data-driven linear quadratic tracking, using an impulse response representation.

---

**Input:**  $w_d \in (\mathbb{R}^w)^T$ ,  $w_r \in (\mathbb{R}^p)^{T_r}$ ,  $w_{\text{ini}} \in (\mathbb{R}^w)^{T_{\text{ini}}}$ , and  $\Phi \in \mathbb{R}^{w \times w}$ ,  $\Phi > 0$ .

- 1:  $(w_d, T_{\text{ini}}, T_r) \xrightarrow{\text{Algorithm 4}} H$
- 2:  $(w_d, T_{\text{ini}}, T_{\text{ini}} + T_r) \xrightarrow{\text{Algorithm 2}} \mathcal{O}$
- 3:  $(w_{\text{ini}}, H, \mathcal{O}) \xrightarrow{(27,28)} w_{\text{free}}$
- 4:  $(\Phi, w_r, w_{\text{free}}, H) \xrightarrow{(29,30)} w^*$

**Output:**  $w^*$ .

---

### 6.3 Algorithm for data-driven linear quadratic tracking

In this section, we present a complete data-driven solution to Problem 17, i.e., one that does not rely on any specific system representation. The basic idea underlying the data-driven linear quadratic tracking algorithm is to project the trajectory  $w_r - w_{\text{free}}$  on the zero initial conditions subbehaviour  $\mathcal{B}_{0,T_r}$  (see (14)) of  $\mathcal{B}$ . The data-driven simulation algorithms, presented in Section 4 can be used to determine the free trajectory  $w_{\text{free}}$  and a basis for  $\mathcal{B}_{0,T_r}$ .

**Theorem 18.** *Let  $W_0 \in \mathbb{R}^{T_r w \times \bullet}$  be a matrix, such that  $\text{colspan}(W_0) = \mathcal{B}_{0,T_r}$ . Then the solution of Problem 17 is given by*

$$w^* = W_0(W_0^\top \tilde{\Phi} W_0)^+ W_0^\top \tilde{\Phi}(w_r - w_{\text{free}}) + w_{\text{free}}, \quad (31)$$

where  $w_{\text{free}}$  is the free trajectory of  $\mathcal{B}$ , caused by the initial trajectory  $w_{\text{ini}}$ . If  $W_0$  defines a basis for  $\mathcal{B}_{0,T_r}$ , then the pseudoinverse in (31) can be replaced by inverse.

*Proof.* Any zero initial conditions trajectory  $w_0 = \text{col}(u_0, y_0) \in (\mathbb{R}^w)^{T_r}$  is of the form  $w_0 = \mathcal{T}_{T_r}(\tilde{H})u_0$ . Therefore,

$$\mathcal{B}_{0,T_r} = \text{colspan}(\mathcal{T}_{T_r}(\tilde{H})) = \text{colspan}(W_0).$$

Consider the space  $\mathcal{W} = (\mathbb{R}^w)^{T_r}$  with inner product defined by  $\langle w_1, w_2 \rangle = w_1^\top \tilde{\Phi} w_2$ . The projector on  $\mathcal{B}_{0,T_r}$  in  $\mathcal{W}$  is

$$\mathcal{T}_{T_r}(\tilde{H})(\mathcal{T}_{T_r}^\top(\tilde{H})\tilde{\Phi}\mathcal{T}_{T_r}(\tilde{H}))^{-1}\mathcal{T}_{T_r}^\top(\tilde{H})\tilde{\Phi} = W_0(W_0^\top\tilde{\Phi}W_0)^+W_0^\top\tilde{\Phi}.$$

Then (31) follows from (29,30).  $\square$

Theorem 18 is based on the fact that the optimal solution  $w^*$  depends only on the subspace  $\mathcal{B}_{0,T_r}$ , the metric, given by the weight matrix  $\Phi$ , and the free trajectory  $w_{\text{free}}$ , initiated by  $w_{\text{ini}}$ , and not on the particular basis of  $\mathcal{B}_{0,T_r}$ . In (29,30), we used as a basis for  $\mathcal{B}_{0,T_r}$  the columns of the Toeplitz matrix  $\mathcal{T}_{T_r}(\tilde{H})$  constructed from the impulse response of  $\mathcal{B}$ . However, using Algorithm 3, we can find another basis for  $\mathcal{B}_{0,T_r}$ . In addition, the free trajectory  $w_{\text{free}}$  can be computed directly from the data  $w_d$ , using Algorithm 1. Then by Theorem 18, the optimal trajectory is (31), and we completely circumvent the need to compute the impulse response  $H$ .

The resulting algorithm for solving Problem 17 is Algorithm 8

---

**Algorithm 8** Data-driven linear quadratic tracking.

---

**Input:**  $w_d \in (\mathbb{R}^w)^T$ ,  $w_r \in (\mathbb{R}^p)^{T_r}$ ,  $w_{\text{ini}} \in (\mathbb{R}^w)^{T_{\text{ini}}}$ , and  $\Phi \in \mathbb{R}^{w \times w}$ ,  $\Phi > 0$ .

- 1:  $(w_d, T_{\text{ini}}, T_r) \xrightarrow{\text{Algorithm 3}} W_0$
- 2:  $(w_d, 0_{mT_r \times 1}, w_{\text{ini}}) \xrightarrow{\text{Algorithm 1}} w_{\text{free}}$
- 3:  $(\Phi, w_r, w_{\text{free}}, W_0) \xrightarrow{(31)} w^*$

**Output:**  $w^*$ .

---

## 6.4 Simulation examples

We show simulation examples that illustrate numerically the equivalence of the three methods for data-driven control linear quadratic tracking. The to-be-controlled plant  $\mathcal{B}$  is the same as that in Example 16. The data  $w_d$ , used by the algorithms, is a random trajectory of  $\mathcal{B}$  with  $T = 200$  samples. It is the same in both experiments. A reference trajectory  $w_r$  with  $T_r$  samples and an initial trajectory  $w_{\text{ini}}$  with  $T_{\text{ini}} = \mathbf{1}(\mathcal{B}) = 2$  samples are chosen as follows:

- Experiment 1: *data-driven regulation*  $T_r = 30$ ,

$$w_r = 0, \quad \text{and} \quad w_{\text{ini}} = \left( \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right),$$

- Experiment 2: *data-driven step tracking*  $T_r = 60$ ,

$$u_r = 0, \quad y_r(t) = \begin{cases} 0, & \text{for } t = 1, \dots, 30 \\ 1, & \text{for } t = 31, \dots, 60 \end{cases}, \quad \text{and } w_{\text{ini}} = \left( \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right).$$

The weight matrix  $\Phi$  is chosen as the  $2 \times 2$  identity matrix.

In both experiments the compared methods compute the same optimal trajectory, see Figures 3 and 4. The optimal values of the cost function in experiment 1 and 2 are  $\|w^*\|_{\Phi}^2 = 1.1139$  and  $\|w_r - w^*\|_{\Phi}^2 = 2.1034$ , respectively.

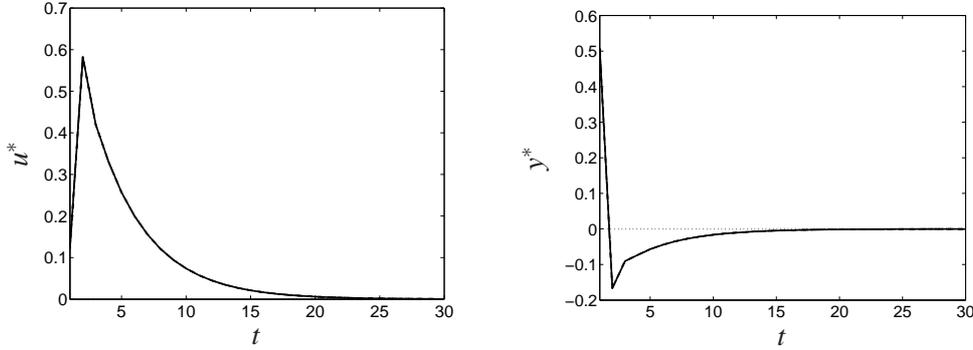


Figure 3: First experiment:  $w_r$  dotted line,  $w^*$  solid line.

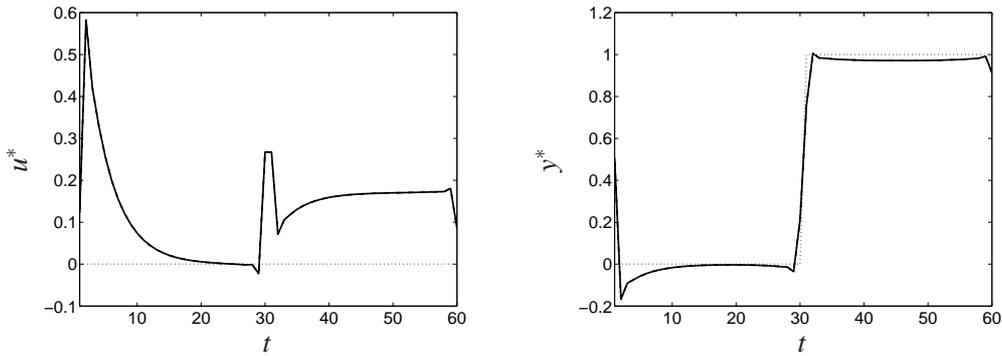


Figure 4: Second experiment:  $w_r$  dotted line,  $w^*$  solid line.

## 7 Conclusions

We presented a generic approach for construction of linear time-invariant system responses directly from data and showed its fundamental role for solving output matching and finite horizon linear quadratic tracking data-driven control problems. Three solutions to the linear quadratic tracking problem are: 1) identify an input/state/output representation of the plant and use classical state space synthesis, 2) identify an impulse response representation of the plant and solve a weighted least squares problem, and 3) do a projection on the subbehaviour  $\mathcal{B}_0$  of zero-initial conditions responses of the plant. The third solution is based on simulation of zero input and zero initial conditions responses of the plant. Using the data-driven simulation algorithm such responses can be computed without deriving a representation of the plant. All solutions need the same basic assumptions: the plant  $\mathcal{B}$  is controllable, and an input component of the given trajectory is persistently exciting of order  $\mathbf{n}(\mathcal{B}) + \mathbf{l}(\mathcal{B}) + 1$ , which are the identifiability conditions of (Markovsky *et al.* 2006a, Chapter 8). The solution given by the input/state/output approach, however, is

in the form of a feedback, while the other solutions compute off-line the optimal trajectory. This poses the question of finding an optimal feedback control law directly from data.

## References

- Åström, K. and B. Wittenmark (1997). *Computer Control Systems: Theory and Design*. Prentice Hall.
- Campi, M., A. Lecchini and S. Savaresi (2002). Virtual reference feedback tuning: a direct method for the design of feedback controllers. *Automatica* **38**, 1337–1346.
- Favoreel, W. (1999). Subspace methods for identification and control of linear and bilinear systems. PhD thesis. Faculty of Engineering, K.U.Leuven.
- Fujisaki, Y., Y. Duan and M. Ikeda (2004). System representation and optimal control in input-output data space. In: *Proc. 10th IFAC Symposium on Large Scale Systems, Osaka, Japan*. pp. 197–202.
- Furuta, K. and M. Wongsaisuwan (1995). Discrete-time LQG dynamic controller design using plant Markov parameters. *Automatica* **31**(9), 1317–1324.
- Hildebrand, R., A. Lecchini, G. Solari and M. Gevers (2004). Prefiltering in iterative feedback tuning: Optimization of the prefilter for accuracy. *IEEE Trans. Automat. Control* **49**, 1801–1806.
- Hjalmarsson, H., M. Gevers, S. Gunnarsson and O. Lequin (1998). Iterative feedback tuning: theory and applications. *IEEE Control Systems Magazine* **18**, 26–41.
- Ikeda, M., Y. Fujisaki and N. Hayashi (2001). A model-less algorithm for tracking control based on input-output data. *Nonlinear Analysis* **47**, 1953–1960.
- Markovsky, I., J. C. Willems and B. De Moor (2002). Continuous-time errors-in-variables filtering. In: *Proceedings of the 41st Conference on Decision and Control*. Las Vegas, NV. pp. 2576–2581.
- Markovsky, I., J. C. Willems, P. Rapisarda and B. De Moor (2005a). Algorithms for deterministic balanced subspace identification. *Automatica* **41**(5), 755–766.
- Markovsky, I., J. C. Willems, P. Rapisarda and B. De Moor (2005b). Data driven simulation with applications to system identification. In: *Proceedings of the 16th IFAC World Congress*. Prague, Czech Republic.
- Markovsky, I., J. C. Willems, S. Van Huffel and B. De Moor (2006a). *Exact and Approximate Modeling of Linear Systems: A Behavioral Approach*. number 11 In: *Monographs on Mathematical Modeling and Computation*. SIAM.
- Markovsky, I., R. J. Vaccaro and S. Van Huffel (2006b). System identification by optimal subspace estimation. Technical Report 06–162. Dept. EE, K.U.Leuven.

- Park, U. and M. Ikeda (2006). Control design of linear time-invariant discrete-time systems with direct use of time series data. In: *Proc. of the Conf. on Decision and Control*. pp. 4454–4459.
- Safonov, M. (1996). Focusing on the knowable: Controller invalidation and learning. In: *Control using logic-based switching* (A. Morse, Ed.). pp. 224–233. Springer-Verlag, Berlin.
- Safonov, M. and T. Tsao (1997). The unfalsified control concept and learning. *IEEE Trans. Automat. Control* **42**(6), 843–847.
- Sain, M. and J. Massey (1969). Invertibility of linear time-invariant dynamical systems. *IEEE Trans. Automat. Control* **14**, 141–149.
- Shi, G. and R. Skelton (2000). Markov data-based LQG control. *Journal of Dynamic Systems, Measurement, and Control* **122**, 551–559.
- Van Overschee, P. and B. De Moor (1996). *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Kluwer. Boston.
- Verhaegen, M. and P. Dewilde (1992). Subspace model identification, Part 1: The output-error state-space model identification class of algorithms. *Int. J. Control* **56**, 1187–1210.
- Willems, J. C., P. Rapisarda, I. Markovsky and B. De Moor (2005). A note on persistency of excitation. *Control Lett.* **54**(4), 325–329.
- Woodley, B. (2001). Model free subspace based  $H_\infty$  control. PhD thesis. Stanford University.