The minimal realization problem in physical coordinates

Fabio Marcuzzi

Università di Padova
Dipartimento di Matematica "Tullio Levi Civita"

February 14, 2022

joint work with Chiara Faccio (SNS)

Due Giorni di algebra Lineare Numerica e Applicazioni
Napoli, 14-15 Febbraio 2022

State-space systems

Let us consider a general, finite-dimensional, deterministic, Discrete-time, Linear Time-Invariant (DLTI) dynamical system, in the so-called *state-space* form:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{1}$$

where $x(k) \in \mathcal{R}^{n_x}$ is the *state vector*, $u(k) \in \mathcal{R}^{m}$ the *input vector*, $y(k) \in \mathcal{R}^{p}$ the *output vector* and $A \in \mathcal{R}^{n_x \times n_x}$, $B \in \mathcal{R}^{n_x \times m}$, $C \in \mathcal{R}^{p \times n_x}$ and $D \in \mathcal{R}^{p \times m}$ are the model matrices.

State-space systems

Let us consider a general, finite-dimensional, deterministic, Discrete-time, Linear Time-Invariant (DLTI) dynamical system, in the so-called *state-space* form:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{1}$$

where $x(k) \in \mathcal{R}^{n_x}$ is the *state vector*, $u(k) \in \mathcal{R}^m$ the *input vector*, $y(k) \in \mathcal{R}^p$ the *output vector* and $A \in \mathcal{R}^{n_x \times n_x}$, $B \in \mathcal{R}^{n_x \times m}$, $C \in \mathcal{R}^{p \times n_x}$ and $D \in \mathcal{R}^{p \times m}$ are the model matrices.
The **Markov coefficients**, G_k, of (1), are defined as:

$$G_0 = D \quad \text{e} \quad G_k = CA^{k-1}B \quad , \quad k = 1, 2, \ldots \tag{2}$$

and their sequence, for $k = 0, 1, 2, \ldots$, correspond to the **discrete impulse response** of the system, in the sense that if we apply a Kronecker delta to the i-th input, we obtain at the output:

$$h^{(i)}(0) = D[:, i] \quad \text{e} \quad h^{(i)}(k) = CA^{k-1}B[:, i] \quad , \quad k = 1, 2, \ldots \tag{3}$$

that is the i-th column of the (matrix) Markov coefficients (2).

Minimal realizations

Definition

(A, B, C, D) *is said* ***realization*** *of the sequence* $\{G_k\}_{k=0}^{\infty}$ *if it holds (2).*

Minimal realizations

Definition

(A, B, C, D) is said ***realization*** *of the sequence $\{G_k\}_{k=0}^{\infty}$ if it holds (2).*

Definition

The realization (A, B, C, D) is ***minimal*** *if the model order n_x is the minimum possible.*

Minimal realizations

Definition

*(A, B, C, D) is said **realization** of the sequence $\{G_k\}_{k=0}^{\infty}$ if it holds (2).*

Definition

*The realization (A, B, C, D) is **minimal** if the model order n_x is the minimum possible.*

Theorem (Kalman)

A realization is minimal iff it is reachable and observable.

The minimal realization problem

The *minimal (state-space) realization problem* can be formulated as follows[1]: “Given some input-output data $u(k), y(k)$, $k = 0, \ldots, N$, find a state-space description of minimal size n_x that is capable of reproducing the given data”.

[1] B. De Schutter. “Minimal state-space realization in linear system theory: An overview”. In: *Journal of Computational and Applied Mathematics,* Special Issue on Numerical Analysis in the 20th Century – Vol. I: Approximation Theory 121.1–2 (2000), pp. 331–354. DOI: 10.1016/S0377-0427(00)00341-1.

The Ho-Kalman algorithm

The first algorithm for this problem has been developed by Ho and Kalman in 1966, for single-input-single-output (SISO) state-space models and their discrete impulse response:

$$H = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & \dots \\ h_2 & h_3 & h_4 & \ddots & \dots \\ h_3 & h_4 & \ddots & \ddots & \dots \\ h_4 & \ddots & \ddots & \ddots & \\ \vdots & \vdots & \vdots & & \end{bmatrix} = \begin{bmatrix} c \\ cA \\ cA^2 \\ cA^3 \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} b & Ab & A^2b & A^3b & \dots \end{bmatrix} \tag{4}$$

The Ho-Kalman algorithm

The first algorithm for this problem has been developed by Ho and Kalman in 1966, for single-input-single-output (SISO) state-space models and their discrete impulse response:

$$H = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & \dots \\ h_2 & h_3 & h_4 & \ddots & \dots \\ h_3 & h_4 & \ddots & \ddots & \dots \\ h_4 & \ddots & \ddots & \ddots & \\ \vdots & \vdots & \vdots & & \end{bmatrix} = \begin{bmatrix} c \\ cA \\ cA^2 \\ cA^3 \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} b & Ab & A^2b & A^3b & \dots \end{bmatrix} \quad (4)$$

These matrices can be truncated at a number of blocks equal to the system dimension, thanks to the well-known Cayley-Hamilton theorem.

The Ho-Kalman algorithm

The first algorithm for this problem has been developed by Ho and Kalman in 1966, for single-input-single-output (SISO) state-space models and their discrete impulse response:

$$H = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & \dots \\ h_2 & h_3 & h_4 & \ddots & \dots \\ h_3 & h_4 & \ddots & \ddots & \dots \\ h_4 & \ddots & \ddots & \ddots & \\ \vdots & \vdots & \vdots & & \end{bmatrix} = \begin{bmatrix} c \\ cA \\ cA^2 \\ cA^3 \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} b & Ab & A^2b & A^3b & \dots \end{bmatrix} \qquad (4)$$

These matrices can be truncated at a number of blocks equal to the system dimension, thanks to the well-known Cayley-Hamilton theorem.
The algorithm associates a numerical factorization of the matrix H to this symbolic description. In this way, the realization (A, B, C) can be easily extracted from the numerical factors.

The Ho-Kalman algorithm

The first algorithm for this problem has been developed by Ho and Kalman in 1966, for single-input-single-output (SISO) state-space models and their discrete impulse response:

$$H = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & \dots \\ h_2 & h_3 & h_4 & \ddots & \dots \\ h_3 & h_4 & \ddots & \ddots & \dots \\ h_4 & \ddots & \ddots & \ddots & \\ \vdots & \vdots & \vdots & & \end{bmatrix} = \begin{bmatrix} c \\ cA \\ cA^2 \\ cA^3 \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} b & Ab & A^2b & A^3b & \dots \end{bmatrix} \tag{4}$$

These matrices can be truncated at a number of blocks equal to the system dimension, thanks to the well-known Cayley-Hamilton theorem.
The algorithm associates a numerical factorization of the matrix H to this symbolic description. In this way, the realization (A, B, C) can be easily extracted from the numerical factors.
For numerical stability issues, are commonly used the QR factorization or the SVD but the applicable numerical factorizations are infinite. The numerical values of the parameters depend therefore on the chosen factorization, i.e. they are not uniquely determined.

black-box vs grey-box modelling

What is the meaning of (A, B, C, D), i.e. the model parameters ?

black-box vs grey-box modelling

What is the meaning of (A, B, C, D), i.e. the model parameters ?

The model may be used as a statistical description of a phenomenon (see e.g. ”statistical learning”); in this case it is called a ”black-box” and its parameters and state-variables have no meaning by itself (like it happens in neural networks and most machine learning modeling techniques).

black-box vs grey-box modelling

What is the meaning of (A, B, C, D), i.e. the model parameters ?

The model may be used as a statistical description of a phenomenon (see e.g. ”statistical learning”); in this case it is called a ”black-box” and its parameters and state-variables have no meaning by itself (like it happens in neural networks and most machine learning modeling techniques).

The model parameters may have physical or socio-economic relevance; in this case it is called a ”gray-box” and its parameters and variables may be masses, friction/heat-transfer/etc coefficients and temperatures, velocities, etc. Usually such a model comes from the discretization of differential equations (ODEs or PDEs), i.e. the model is usually defined on a spatial and/or temporal continuum:

$$\begin{array}{rcl} \dot{x}(t) & = & A_c x(t) + B_c u(t) \\ y(t) & = & C_c x(t) + D_c u(t) \end{array} , \tag{5}$$

We will refer to this second framework, that has many applications, e.g. ”soft-sensors”.

Model discretization

Model discretization usually means that even for linear models there is a nonlinear map that relates the entries of (A, B, C, D) with their continuous counterpart (A_c, B_c, C_c, D_c).

Model discretization

Model discretization usually means that even for linear models there is a nonlinear map that relates the entries of (A, B, C, D) with their continuous counterpart (A_c, B_c, C_c, D_c).
For example, let T_{sc} be the time-discretization step, $x(k) \approx x_c(k\ T_{sc})$ be the state vector of the discrete-time model, and consider for simplicity the well-known θ-method:

$$\frac{x(k+1) - x(k)}{T_{sc}} = (1-\theta)f(x(k), u(k)) + \theta f(x(k+1), u(k+1)) \quad . \tag{6}$$

. Using (6) with e.g. $\theta = 1$ (the Implicit Euler method) we obtain from (5) a state-space discrete model in physical coordinates:

$$\begin{aligned} x(k+1) &= A_f x(k) + B_f u(k) \\ y(k) &= C_f x(k) \end{aligned} \tag{7}$$

with

$$A_f = (I - T_{sc}A_c)^{-1} \quad , \; B_f = (I - T_{sc}A_c)^{-1} T_{sc} B_c \quad , \; C_f = C_c \tag{8}$$

Model discretization

Model discretization usually means that even for linear models there is a nonlinear map that relates the entries of (A, B, C, D) with their continuous counterpart (A_c, B_c, C_c, D_c).
For example, let T_{sc} be the time-discretization step, $x(k) \approx x_c(k\ T_{sc})$ be the state vector of the discrete-time model, and consider for simplicity the well-known θ-method:

$$\frac{x(k+1) - x(k)}{T_{sc}} = (1-\theta)f(x(k), u(k)) + \theta f(x(k+1), u(k+1)) \quad . \tag{6}$$

. Using (6) with e.g. $\theta = 1$ (the Implicit Euler method) we obtain from (5) a state-space discrete model in physical coordinates:

$$\begin{aligned} x(k+1) &= A_f x(k) + B_f u(k) \\ y(k) &= C_f x(k) \end{aligned} \tag{7}$$

with

$$A_f = (I - T_{sc}A_c)^{-1} \quad , \; B_f = (I - T_{sc}A_c)^{-1} T_{sc} B_c \quad , \; C_f = C_c \tag{8}$$

Our aim is to obtain system (7) **by solving the minimal realization problem.**

Subspace identification methods

At present, the solution algorithms for general multi-input-multi-output (MIMO) state-space models are the so-called "subspace identification methods"[2], that we briefly recall. In the deterministic case, these methods usually derive the realization (A, B, C, D) from an *extended observability matrix* or from the estimated state vectors X. To give and idea, following this second approach, matrices are derived using least-squares on this equation:

$$\begin{bmatrix} X^d_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X^d_i \\ U_{i|i} \end{bmatrix} \tag{9}$$

[2] Peter Van Overschee and Bart De Moor. *Subspace Methods in System Identification*. Springer, 1996.

Subspace identification methods

At present, the solution algorithms for general multi-input-multi-output (MIMO) state-space models are the so-called "subspace identification methods"[2], that we briefly recall. In the deterministic case, these methods usually derive the realization (A, B, C, D) from an *extended observability matrix* or from the estimated state vectors X. To give and idea, following this second approach, matrices are derived using least-squares on this equation:

$$\begin{bmatrix} X_{i+1}^d \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_i^d \\ U_{i|i} \end{bmatrix} \tag{9}$$

Now the point is: which base have used the subspace methods to express the minimal realization found?

[2] Peter Van Overschee and Bart De Moor. *Subspace Methods in System Identification*. Springer, 1996.

Subspace identification methods

At present, the solution algorithms for general multi-input-multi-output (MIMO) state-space models are the so-called ”subspace identification methods”[2], that we briefly recall. In the deterministic case, these methods usually derive the realization (A, B, C, D) from an *extended observability matrix* or from the estimated state vectors X. To give and idea, following this second approach, matrices are derived using least-squares on this equation:

$$\begin{bmatrix} X^d_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X^d_i \\ U_{i|i} \end{bmatrix} \tag{9}$$

Now the point is: which base have used the subspace methods to express the minimal realization found?
Indeed, they compute the state estimates X from block-Hankel matrices built upon inputs and outputs measurements, and performing on these matrices oblique projections, SVD and QR factorizations. They are completely **data-driven**.

[2] Peter Van Overschee and Bart De Moor. *Subspace Methods in System Identification*. Springer, 1996.

Invariants to a basis change

Moreover, the minimal realization is not unique: given an arbitrary, invertible, basis-change matrix T, the system transformed in the new coordinates $\tilde{x} = T^{-1}x$ maintains the same input-output behavior. Therefore, there are **infinite possible data-driven bases** that may be used by subspace methods to build the minimal realization.

Invariants to a basis change

Moreover, the minimal realization is not unique: given an arbitrary, invertible, basis-change matrix T, the system transformed in the new coordinates $\tilde{x} = T^{-1}x$ maintains the same input-output behavior. Therefore, there are **infinite possible data-driven bases** that may be used by subspace methods to build the minimal realization.
In general, at each basis change the matrices $(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$ are different, and so the model parameters:

$$\tilde{A} = T^{-1}AT, \quad \tilde{B} = T^{-1}B, \quad \tilde{C} = CT, \quad \tilde{x} = T^{-1}x \tag{10}$$

Note that there are also other invariants, e.g. the eigenvalues of A, as can be easily noticed from (11).

Invariants to a basis change

Moreover, the minimal realization is not unique: given an arbitrary, invertible, basis-change matrix T, the system transformed in the new coordinates $\tilde{x} = T^{-1}x$ maintains the same input-output behavior. Therefore, there are **infinite possible data-driven bases** that may be used by subspace methods to build the minimal realization.
In general, at each basis change the matrices $(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$ are different, and so the model parameters:

$$\tilde{A} = T^{-1}AT, \quad \tilde{B} = T^{-1}B, \quad \tilde{C} = CT, \quad \tilde{x} = T^{-1}x \tag{10}$$

Note that there are also other invariants, e.g. the eigenvalues of A, as can be easily noticed from (11).

Hence, we try to solve a harder problem: **to find the (unique) minimal realization whose state vector is expressed in the *physical base***, that is true when each of its state variables has a twin variable in the physical-mathematical model describing the real system. Only with this base the estimated model parameters have a physical meaning.

Physical parameters estimate

In system identification, the computation of physical parameter estimates classically adopts **nonlinear estimation procedures**[3]; these, anyway, suffer from convergence problems, depending much from the initialization of the estimates and from the ill-conditioning of the problem to be numerically solved.

[3] Lennart Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1999.

Physical parameters estimate

In system identification, the computation of physical parameter estimates classically adopts **nonlinear estimation procedures**[3]; these, anyway, suffer from convergence problems, depending much from the initialization of the estimates and from the ill-conditioning of the problem to be numerically solved.
The bibliography on the estimation of physical parameters with subspace methods imposes **constraints on the system's response** (impulse response, transfer function, weighting matrices constructed from the Markov parameters of the unknown observer, etc.)

[3] Lennart Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1999.

Physical parameters estimate

In system identification, the computation of physical parameter estimates classically adopts **nonlinear estimation procedures**[3]; these, anyway, suffer from convergence problems, depending much from the initialization of the estimates and from the ill-conditioning of the problem to be numerically solved.
The bibliography on the estimation of physical parameters with subspace methods imposes **constraints on the system's response** (impulse response, transfer function, weighting matrices constructed from the Markov parameters of the unknown observer, etc.)
There is also a huge literature on more physical approaches, tailored on a specific application, e.g. in vibration mechanics it it well known the so-called "inverse vibration problem"

[3] Lennart Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1999.

Physical parameters estimate

In system identification, the computation of physical parameter estimates classically adopts **nonlinear estimation procedures**[3]; these, anyway, suffer from convergence problems, depending much from the initialization of the estimates and from the ill-conditioning of the problem to be numerically solved.
The bibliography on the estimation of physical parameters with subspace methods imposes **constraints on the system's response** (impulse response, transfer function, weighting matrices constructed from the Markov parameters of the unknown observer, etc.)
There is also a huge literature on more physical approaches, tailored on a specific application, e.g. in vibration mechanics it it well known the so-called "inverse vibration problem"
In the literature there are also methods that exploit the structure of the matrices in the continuum-time model, because of their physical meaning, to compute directly a matrix T which should transform in physical coordinates the estimated model to obtain this it is necessary to reformulate the abstract model equations into a null-space-based problem, that brings to a quite involved solution and has in general an high computational cost or it is restricted to a small number of model structures.

[3] Lennart Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1999.

Physical parameters estimate

In system identification, the computation of physical parameter estimates classically adopts **nonlinear estimation procedures**[3]; these, anyway, suffer from convergence problems, depending much from the initialization of the estimates and from the ill-conditioning of the problem to be numerically solved.
The bibliography on the estimation of physical parameters with subspace methods imposes **constraints on the system's response** (impulse response, transfer function, weighting matrices constructed from the Markov parameters of the unknown observer, etc.)
There is also a huge literature on more physical approaches, tailored on a specific application, e.g. in vibration mechanics it it well known the so-called “inverse vibration problem”
In the literature there are also methods that exploit the structure of the matrices in the continuum-time model, because of their physical meaning, to compute directly a matrix T which should transform in physical coordinates the estimated model to obtain this it is necessary to reformulate the abstract model equations into a null-space-based problem, that brings to a quite involved solution and has in general an high computational cost or it is restricted to a small number of model structures.
The method here proposed is a linear estimation, eventually followed by a nonlinear map (matrix inversion).

[3] Lennart Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1999.

Our LA approach

Let us suppose that exists an (unknown) basis-change matrix T_f, that transforms the data-driven minimal realization obtained by subspace methods, into the minimal realization in the physical base:

$$A_f = T_f^{-1} A_s T_f, \quad B_f = T_f^{-1} B_s, \quad C_f = C_s T_f, \quad \tilde{x}_f = T_f^{-1} x_s \tag{11}$$

Our LA approach

Let us suppose that exists an (unknown) basis-change matrix T_f, that transforms the data-driven minimal realization obtained by subspace methods, into the minimal realization in the physical base:

$$A_f = T_f^{-1} A_s T_f, \quad B_f = T_f^{-1} B_s, \quad C_f = C_s T_f, \quad \tilde{x}_f = T_f^{-1} x_s \tag{11}$$

Our approach to find an approximation $\hat{T}_f$ starts by considering the eigen-decomposition of the (unknown) matrix A_f:

$$A_f = V_f \Lambda_{A_f} V_f^{-1} \tag{12}$$

Note that the **ordering** of the eigenvectors is not an invariant for the minimal realization in the physical base. In fact, a permutation of the eigenvectors is a basis-change which changes also the C_f matrix, which instead must remain fixed with the definition of the state variables and of the output variables.

Our LA approach

Let us suppose that exists an (unknown) basis-change matrix T_f, that transforms the data-driven minimal realization obtained by subspace methods, into the minimal realization in the physical base:

$$A_f = T_f^{-1} A_s T_f, \quad B_f = T_f^{-1} B_s, \quad C_f = C_s T_f, \quad \tilde{x}_f = T_f^{-1} x_s \tag{11}$$

Our approach to find an approximation $\hat{T}_f$ starts by considering the eigen-decomposition of the (unknown) matrix A_f:

$$A_f = V_f \Lambda_{A_f} V_f^{-1} \tag{12}$$

Note that the **ordering** of the eigenvectors is not an invariant for the minimal realization in the physical base. In fact, a permutation of the eigenvectors is a basis-change which changes also the C_f matrix, which instead must remain fixed with the definition of the state variables and of the output variables.

→ From the other side, there is no way to guarantee that the subspace methods find a minimal realization in which the eigenvectors of A_s are in the same order as those of A_f, since here C_s is completely arbitrary and data-driven.

Imposing the physical C matrix

This suggests us that there is an unknown, optimal permutation that should be applied to the eigenvectors of A_s, and we insert the search for this optimal permutation in our algorithm, as follows.

Imposing the physical C matrix

This suggests us that there is an unknown, optimal permutation that should be applied to the eigenvectors of A_s, and we insert the search for this optimal permutation in our algorithm, as follows.
First of all, by using the eigenvectors V_s of A_s as a first basis change, we obtain a realization in modal coordinates

$$\{\Lambda_{A_s}, B_{V_s}, C_{V_s}\} \tag{13}$$

and get two advantages:

- Λ_{A_s} is a good estimate of A_f in modal coordinates, i.e. diagonalized. Indeed, only the diagonal elements of both matrices are different from zero and are equal to the eigenvalues, that are usually well estimated by subspace methods.
- the eigenspaces are now associated to single state variables and it is possible, with a row-column permutation, to associate them to specific measured variables, since the dynamics are decoupled, in this basis. To consider all the possible permutations for low-order models requires a modest effort and, as we will see, it turns out to be very effective on obtaining a good approximation of T_f among the infinite possible T.

Imposing the physical C_f matrix

Suppose we have decided a row-permutation M_r. If we apply the basis change $T^{-1} = M_r V_s^{-1}$ we get a realization $\{\Lambda_{A_s}^{perm}, B_{V_s}^{perm}, C_{V_s}^{perm}\}$ that can now be reconducted to the physical cohordinates by imposing a further basis change T_x such that $C_{V_s}^{perm} T_x = C_f$.

Imposing the physical C_f matrix

Suppose we have decided a row-permutation M_r. If we apply the basis change $T^{-1} = M_r V_s^{-1}$ we get a realization $\{\Lambda_{A_s}^{perm}, B_{V_s}^{perm}, C_{V_s}^{perm}\}$ that can now be reconducted to the physical cohordinates by imposing a further basis change T_x such that $C_{V_s}^{perm} T_x = C_f$.
This can be done by solving:

$$MT_x = G \; , \; M = \begin{bmatrix} C_{V_s}^{perm} \\ X \end{bmatrix} , \; G = \begin{bmatrix} C_f \\ Y \end{bmatrix} \tag{14}$$

where $M, T_x, G \in \mathcal{R}^{n_x \times n_x}$. The matrices X and Y can be chosen in different ways.

Imposing the physical C_f matrix

Suppose we have decided a row-permutation M_r. If we apply the basis change $T^{-1} = M_r V_s^{-1}$ we get a realization $\{\Lambda_{A_s}^{perm}, B_{V_s}^{perm}, C_{V_s}^{perm}\}$ that can now be reconducted to the physical cohordinates by imposing a further basis change T_x such that $C_{V_s}^{perm} T_x = C_f$.
This can be done by solving:

$$MT_x = G \; , \; M = \left[\begin{array}{c} C_{V_s}^{perm} \\ X \end{array} \right] , \; G = \left[\begin{array}{c} C_f \\ Y \end{array} \right] \tag{14}$$

where $M, T_x, G \in \mathcal{R}^{n_x \times n_x}$. The matrices X and Y can be chosen in different ways.
Let us define $H_r^{\perp}$ the matrix whose rows form a basis for the orthogonal complement of the row space of C_{V_s}, and $I[i_u, :]$ the matrix formed by the rows of the identity matrix of indexes corresponding to the unmeasured state variables. We found a few reasonable choices/methods to choose X and Y:

1. $X = Y = H_r^{\perp}$
2. $X = Y = I[i_u, :]$
3. $X = H_r^{\perp} \; , \; Y = I[i_u, :]$
4. $X = Y = H_r^{\perp} V_{As}$
5. $X = H_r^{\perp} V_{As} \; , \; Y = I[i_u, :]$

Permutations reduction

Different choices have different properties. We have found choice ”2” as the best one, since it reduces considerably the number of permutations that must be considered, as we have demonstrated in the following Lemma.

Permutations reduction

Different choices have different properties. We have found choice ”2” as the best one, since it reduces considerably the number of permutations that must be considered, as we have demonstrated in the following Lemma.

Lemma

Consider problem (14), with the choice $X = Y = I[i_u, :]$. If we change basis to the realization $\{\Lambda_{A_s}, B_{V_s}, C_{V_s}\}$ with $T^{-1} = M_r$ a row-permutation matrix, we determine a specific selection of eigenvectors to form the matrix $\hat{T}_f^{-1}$. Moreover, the number of permutations to be considered is restricted to $\binom{n_x}{p} = \frac{n_x!}{(n_x-p)!\ p!}$.

Permutations reduction

Different choices have different properties. We have found choice ”2” as the best one, since it reduces considerably the number of permutations that must be considered, as we have demonstrated in the following Lemma.

Lemma

Consider problem (14), with the choice $X = Y = I[i_u, :]$. If we change basis to the realization $\{\Lambda_{A_s}, B_{V_s}, C_{V_s}\}$ with $T^{-1} = M_r$ a row-permutation matrix, we determine a specific selection of eigenvectors to form the matrix $\hat{T}_f^{-1}$. Moreover, the number of permutations to be considered is restricted to $\binom{n_x}{p} = \frac{n_x!}{(n_x-p)!\ p!}$.

In practice, in the test examples this means to consider e.g. with $n_x = 6$, 15 permutations instead of 720.

Looking for the optimal permutation

Here a fundamental problem is that the algorithm cannot directly measure the efficacy of a given permutation in the estimation of parameters, since they are unknown and there is no evident algebraic characterization of the physical basis in the discrete model.

Looking for the optimal permutation

Here a fundamental problem is that the algorithm cannot directly measure the efficacy of a given permutation in the estimation of parameters, since they are unknown and there is no evident algebraic characterization of the physical basis in the discrete model.

Let us suppose to know a coarse estimate of (at least) a few parameters of the continuous model, that we want to estimate more precisely with the algorithm proposed. With this novel piece of information we apply then an heuristic method, which we will validate with the numerical experiments. An example of a reasonable heuristic method may be e.g. the following:

1. from the coarse initial estimate of the parameters of the continuous model, compute the matrix $\tilde{A}_c$;
2. from $\hat{T}_f$ obtain $\hat{A}_f$ and compute $\hat{A}_c$ from (8); choose the permutation where the submatrix $\hat{A}_c[i_m, i_m]$ is closer, in a chosen norm, to $\tilde{A}_c[i_m, i_m]$.

Looking for the optimal permutation

Here a fundamental problem is that the algorithm cannot directly measure the efficacy of a given permutation in the estimation of parameters, since they are unknown and there is no evident algebraic characterization of the physical basis in the discrete model.

Let us suppose to know a coarse estimate of (at least) a few parameters of the continuous model, that we want to estimate more precisely with the algorithm proposed. With this novel piece of information we apply then an heuristic method, which we will validate with the numerical experiments. An example of a reasonable heuristic method may be e.g. the following:

1. from the coarse initial estimate of the parameters of the continuous model, compute the matrix $\tilde{A}_c$;
2. from $\hat{T}_f$ obtain $\hat{A}_f$ and compute $\hat{A}_c$ from (8); choose the permutation where the submatrix $\hat{A}_c[i_m, i_m]$ is closer, in a chosen norm, to $\tilde{A}_c[i_m, i_m]$.

The numerical experiments confirm that this method is a good choice to distinguish the optimal permutation, even starting from a matrix discretized with a coarse estimate of the physical parameters.

Algorithm

Now we can formulate a solution algorithm[4]:

Minimal realization in physical base

1: given a set of I/O data, find a minimal realization $\{A_s, B_s, C_s\}$ through a subspace algorithm;

2: diagonalize A_s and get the decomposition (12);

3: for each convenient permutation of the eigenvalues/eigenvectors, use the permuted eigenvectors V_s to change the basis of the state vector in modal coordinates (13) and compute the basis-change matrix $\hat{T}_f = T_x^{-1} M_r V_s^{-1}$;

4: find the optimal permutation using the apriori, coarse, estimate of the parameters and compute the resulting minimal realization in partial physical coordinates.

[4] C. Faccio and F. Marcuzzi. "A linear algorithm for the minimal realization problem in physical coordinates with a non-invertible output matrix". In: *Linear Algebra and its Applications* (accepted).

Experiments

Let us consider a well-known class of models:

$$M\ddot{d}(t) + G\dot{d}(t) + Kd(t) = f(t) \tag{15}$$

Here $x(t) = \begin{bmatrix} \dot{d}(t) \\ d(t) \end{bmatrix}$

and $A_c = \begin{bmatrix} -M^{-1}G & -M^{-1}K \\ I & 0 \end{bmatrix}$, $B_c = \begin{bmatrix} M^{-1} \\ 0 \end{bmatrix}$

Experiments

Let us consider a well-known class of models:

$$M\ddot{d}(t) + G\dot{d}(t) + Kd(t) = f(t) \tag{15}$$

Here $x(t) = \begin{bmatrix} \dot{d}(t) \\ d(t) \end{bmatrix}$

and $A_c = \begin{bmatrix} -M^{-1}G & -M^{-1}K \\ I & 0 \end{bmatrix} \quad , \quad B_c = \begin{bmatrix} M^{-1} \\ 0 \end{bmatrix}$

Let us consider the following index E_i to quantify the physical parameters estimation error:

$$E_i = |diag(\hat{A}_c[i_m, i_m] - A_c[i_m, i_m]/A_c[i_m, i_m])| \tag{16}$$

Experiments

$p=2$	$median\{min(E_i)\}_{i=0\dots N-1}, median\{max(E_i)\}_{i=0\dots N-1}$	
n_x	standard ss	$\hat{T}_f$ opt perm
4	0.86, 1.12	0.01, 0.03
6	0.94, 1.21	0.06, 0.13
10	0.87, 1.52	0.03, 0.09
20	0.97, 1.17	0.17, 0.45
30	1.00, 1.38	0.52, 0.80

Table: The Table shows the results on the estimation error E_i (16), for various n_x and various estimation methods, from left to right: a standard subspace method ("standard ss") and $\hat{T}_f$ with the best permutation obtainable by Algorithm 17 ("opt_perm"). Each cell contains two results: the median minimum relative error $median\{min(E_i)\}_{i=0\dots N-1}$ and the median maximum relative error $median\{min(E_i)\}_{i=0\dots N-1}$ through N experiments.

Experiments

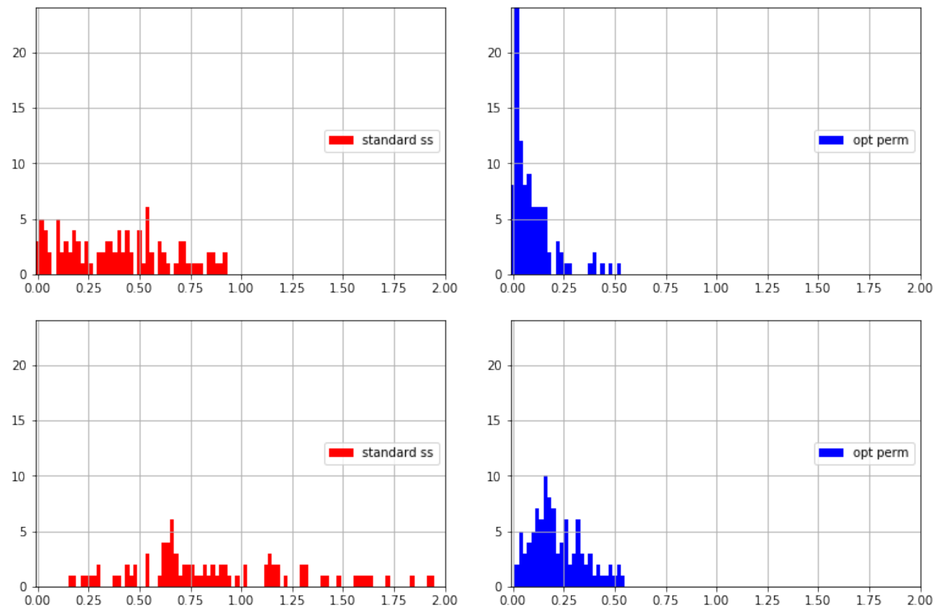


Figure: Above left: histogram of $min(E_i),\ i = 0 \dots N-1$, with "standard ss"; above right: histogram of $min(E_i),\ i = 0 \dots N-1$ with "opt perm"; below left: histogram of $max(E_i),\ i = 0 \dots N-1$ with "standard ss"; below right: histogram of $max(E_i),\ i = 0 \dots N-1$ with "opt perm".

We have put on evidence that eigenvectors approximation is a key issue in the minimal realization problem.

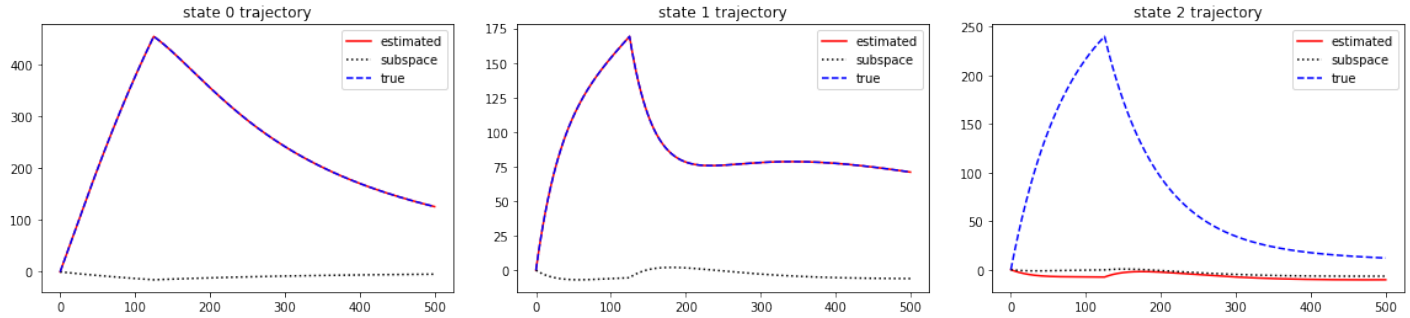
We have put on evidence that eigenvectors approximation is a key issue in the minimal realization problem.

Open source codes

All codes discussed available online at the URL:
`https://github.com/NLALDlab/subspace-methods-in-physical-base.`

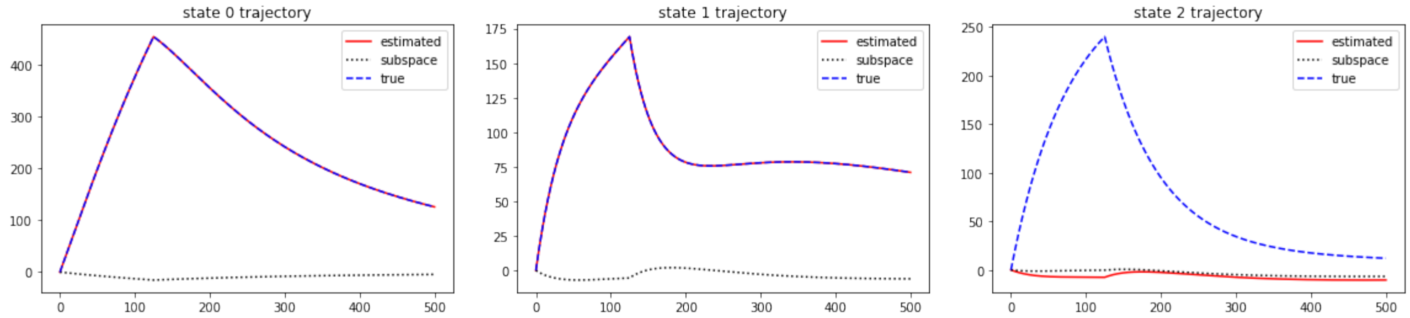
Constraining eigenvectors: overdetermined case

The basis change T_x obtained in (14) gives an exact matrix $C = C_f$:



Constraining eigenvectors: overdetermined case

The basis change T_x obtained in (14) gives an exact matrix $C = C_f$:



As a future work, we are trying to relax this statement, with an obvious precision loss in the state variables, adopting a regularization term that should improve the eigenvectors to follow a prescribed behaviour:

$$\begin{bmatrix} C_s V_s M_r^T \\ \lambda V_s M_r^T \end{bmatrix} T_x = \begin{bmatrix} C_f \\ \lambda Z \end{bmatrix} \quad (17)$$

where Z is a matrix of known eigenvectors, reasonably close to what should be the eigenvectors of A_f. They could be e.g. known from the theory related to the specific class of models, or the eigenvectors of $\tilde{A}_f$.

Constraining eigenvectors: underdetermined case

Actually, the matrix Z could be itself learned from data, i.e. a dictionary of eigenvectors from which to recover T_x with a sparse recovery from this underdetermined system:

$$\begin{bmatrix} C_s V_s M_r^T & 0 \\ \lambda V_s M_r^T & -\lambda Z \end{bmatrix} \begin{bmatrix} T_x \\ S \end{bmatrix} = \begin{bmatrix} C_f \\ 0 \end{bmatrix} \tag{18}$$

Thank you for your attention!

References I

- B. De Schutter. “Minimal state-space realization in linear system theory: An overview”. In: *Journal of Computational and Applied Mathematics,* Special Issue on Numerical Analysis in the 20th Century – Vol. I: Approximation Theory 121.1–2 (2000), pp. 331–354. DOI: 10.1016/S0377-0427(00)00341-1.
- C. Faccio and F. Marcuzzi. “A linear algorithm for the minimal realization problem in physical coordinates with a non-invertible output matrix”. In: *Linear Algebra and its Applications* (accepted).
- Lennart Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1999.
- Peter Van Overschee and Bart De Moor. *Subspace Methods in System Identification*. Springer, 1996.