1 Introduction

The LPV-PBSIDOPT algorithm as implemented in the LPVCORE function "lpvpbsidopt" differs in the originating paper [1] on a few features. In this document, these differences are explained, and the additional notation is introduced that is used in the script.

2 Separate basis functions

Separate basis functions for each of the open-loop system matrices (A, B, C, D, and K) are supported. Since the derivation in the paper does not consider this scenario, the model equations (1) and (2) are rewritten as follows:

$$\begin{aligned} x_{k+1} &= \sum_{i=1}^{m_A} \mu_{A,k}^{(i)} A^{(i)} x_k + \sum_{i=1}^{m_B} \mu_{B,k}^{(i)} B^{(i)} u_k + \sum_{i=1}^{m_K} \mu_{K,k}^{(i)} K^{(i)} e_k \\ y_k &= \sum_{i=1}^{m_C} \mu_{C,k}^{(i)} C^{(i)} x_k + \sum_{i=1}^{m_D} \mu_{D,k}^{(i)} D^{(i)} u_k + e_k \end{aligned} \tag{1}$$

Similarly, the basis functions for the closed-loop system matrices $\tilde{A}$ and $\tilde{B}$ are represented by $\mu_{\tilde{A},k}$ and $\mu_{\tilde{B},k}$, respectively, with the number of basis functions in each denoted by $m_{\tilde{A}}$ and $m_{\tilde{B}}$.

For simplicity, the closed-loop input matrix $\breve{B} = \begin{bmatrix} \tilde{B} & K \end{bmatrix}$ is expanded in terms of basis functions of $\tilde{B}$ only, i.e., the sparsity which arises from the fact that the $m_K \leq m_{\tilde{B}}$ is not exploited.

To keep the dimensions of the matrices used in the algorithm compatible, the definition of $\tilde{q}$ as introduced in **Definition 4** is generalized for the case $m_{\tilde{B}} \neq m_{\tilde{A}}$:

$$\tilde{q} = (r + \ell) \sum_{j=1}^{p} m_{\tilde{A}}^{j-1} m_{\tilde{B}} \tag{2}$$

Finally, **Definition 5** is also generalized:

$$P_{p|k} = \mu_{\tilde{A},k+p-1} \otimes \cdots \otimes \mu_{\tilde{A},k+1} \otimes \mu_{\tilde{B},k} \otimes I_{r+\ell} \tag{3}$$

3 Parameter-dependent output equation

A parameter-dependent output equation is supported (as was seen in the previous section). To this end, the *time-varying input-output transition matrix* $\bar{\mathcal{H}}_k^p$ is introduced. It maps past inputs from time steps k to $k+p-1$ to output at time step $k+p$ under the assumption that the past window p is sufficiently large so that the initial state k time steps ago has no influence:

$$\bar{\mathcal{H}}_k^p = \begin{bmatrix} C_{k+p}\phi_{p-1,k+1}\breve{B}_k, & \cdots, & C_{k+p}\phi_{1,k+p-1}\breve{B}_{k+p-2}, & C_{k+p}\breve{B}_{k+p-1} \end{bmatrix}. \tag{4}$$

Note $\bar{\mathcal{H}}_k^p = C_{k+p}\bar{\mathcal{K}}_k^p$.

In order to decompose $\bar{\mathcal{H}}_k^p$ into a constant and parameter-dependent part, the following matrices are introduced:

$$\mathcal{I}_j = \begin{bmatrix} C^{(1)}\mathcal{L}_j, & \cdots, & C^{(m_C)}\mathcal{L}_j \end{bmatrix}, \tag{5}$$

and

$$\mathcal{H}^p = \begin{bmatrix}\mathcal{I}_p, & \mathcal{I}_{p-1}, & \cdots, & \mathcal{I}_1\end{bmatrix} \in \mathbb{R}^{\ell \times m_C \tilde{q}}. \tag{6}$$

Note that $\mathcal{H}^p$ has m times as many columns as $\mathcal{K}^p$. The relation between $\mathcal{H}^p$ and $C^{(1)}\mathcal{K}^p$ is given as follows:

$$C^{(1)}\mathcal{K}^p = \mathcal{H}^p \underbrace{\operatorname{blkdiag}\left(J_{\mathcal{H}^p}^{(p)}, \ldots, J_{\mathcal{H}^p}^{(1)}\right)}_{J_{\mathcal{H}^p}} = \mathcal{H}^p J_{\mathcal{H}^p}, \tag{7}$$

with

$$J_{\mathcal{H}^p}^{(i)} = \begin{bmatrix} 1 \\ 0_{(m_C-1)\times 1}\end{bmatrix} \otimes I_{m_{\tilde{A}}^{i-1} m_{\tilde{B}}(r+\ell)} \tag{8}$$

Finally, the following extensions to $P_{p|k}$ and N_k^p are needed:

$$Q_{p|k} = \mu_{C,k+p} \otimes \underbrace{\mu_{\tilde{A},k+p-1} \otimes \cdots \otimes \mu_{\tilde{A},k+1} \otimes \mu_{\tilde{B},k} \otimes I_{r+\ell}}_{P_{p|k}} \in \mathbb{R}^{m_C m_{\tilde{A}}^{(p-1)} m_{\tilde{B}}(r+\ell)\times(r+\ell)} \tag{9}$$

$$M_k^p = \begin{bmatrix} Q_{p|k} & & & 0 \\ & Q_{p-1|k+1} & & \\ & & \ddots & \\ 0 & & & Q_{1|k+p-1}\end{bmatrix} \in \mathbb{R}^{m_C \tilde{q} \times p(r+\ell)} \tag{10}$$

Based on the above, the factorization of $\bar{\mathcal{H}}_k^p$ is obtained as follows:

$$\bar{\mathcal{H}}_k^p = \mathcal{H}^p M_k^p \tag{11}$$

The matrix $Z_{\mathcal{H}}$ is introduced as the counter-part to Z in (12):

$$Z_{\mathcal{H}} = \begin{bmatrix} M_1^p \bar{z}_1^p, & \ldots, & M_{N-p+1}^p \bar{z}_{N-p+1}^p \end{bmatrix} \tag{12}$$

The next change is (13), where the optimization does not occur over $C\mathcal{K}^p$, but $\mathcal{H}^p$. Note that $C\mathcal{K}^p$ is a special case of $\mathcal{H}^p$ for constant C and that Z is replaced by $Z_{\mathcal{H}}$.

$$\min_{\mathcal{H}^p, D} \|Y - \mathcal{H}^p Z_{\mathcal{H}} - DU\|_F^2 \tag{13}$$

The matrix product $\Gamma^p \mathcal{K}^p$ (14) can be recovered by extracting $C\mathcal{K}^p$ (using the notation of the paper) from the columns of $\mathcal{H}^p$ that correspond to the parameter-independent part of C.

3.1 Kernel method

For the kernel method, the parameter-dependent output equation also requires additional definitions. First, $Z_{\mathcal{H}}^{i,j}$ as an extension to $Z^{i,j}$ in (23):

$$Z_{\mathcal{H}}^{i,j} = \begin{bmatrix} Q_{p-j+1|j-i+1} z_{j-i+1}, & \ldots, & Q_{p-j+1|\bar{N}+j-i} z_{\bar{N}+j-i}\end{bmatrix} \tag{14}$$

The matrix $\Gamma^p \mathcal{K}^p Z$ is constructed as follows:

$$\Gamma^p \mathcal{K}^p Z = \begin{bmatrix} \alpha \sum_{j=1}^{p} (Z_{\mathcal{H}}^{1,j})^{\mathrm{T}} J_{\mathcal{H}^p}^{(p-j+1)} Z^{1,j} \\ \alpha \sum_{j=2}^{p} (Z_{\mathcal{H}}^{2,j})^{\mathrm{T}} J_{\mathcal{H}^p}^{(p-j+1)} Z^{1,j} \\ \vdots \\ \alpha \sum_{j=p}^{p} (Z_{\mathcal{H}}^{p,j})^{\mathrm{T}} J_{\mathcal{H}^p}^{(p-j+1)} Z^{1,j} \end{bmatrix} \tag{15}$$

In order to compute $\hat{\alpha}$ in (19), the matrix $Z_{\mathcal{H}}^{\mathrm{T}} Z_{\mathcal{H}}$ is needed. With a small modification of **Theorem 10**, this matrix can be decomposed in terms of $Z_{\mathcal{H}}^{1,j}$, $j = 1, ..., p$:

$$Z_{\mathcal{H}}^{\mathrm{T}} Z_{\mathcal{H}} = \sum_{j=1}^{p} \left(Z_{\mathcal{H}}^{1,j} \right)^{\mathrm{T}} Z_{\mathcal{H}}^{1,j} \tag{16}$$

References

[1] Jan-Willem van Wingerden and Michel Verhaegen. Subspace identification of Bilinear and LPV systems for open- and closed-loop data. *Automatica*, 45(2):372–381, 2009.