


Article

# Partial Singular Value Assignment for Large-Scale Systems

Yiting Huang, Qiong Tang and Bo Yu \* 

School of Science, Hunan University of Technology, Zhuzhou 412007, China; m22070100006@stu.hut.edu.cn (Y.H.); tangqiong@hut.edu.cn (Q.T.)

\* Correspondence: yubo@hut.edu.cn

**Abstract:** The partial singular value assignment problem stems from the development of observers for discrete-time descriptor systems and the resolution of ordinary differential equations. Conventional techniques mostly utilize singular value decomposition, which is unfeasible for large-scale systems owing to their relatively high complexity. By calculating the sparse basis of the null space associated with some orthogonal projections, the existence of the matrix in partial singular value assignment is proven and an algorithm is subsequently proposed for implementation, effectively avoiding the full singular value decomposition of the existing methods. Numerical examples exhibit the efficiency of the presented method.

**Keywords:** partial singular value assignment; large-scale system; sparse basis; orthogonal projection

**MSC:** 65F99



**Citation:** Huang, Y.; Tang, Q.; Yu, B. Partial Singular Value Assignment for Large-Scale Systems. *Axioms* **2023**, *12*, 1012. <https://doi.org/10.3390/axioms12111012>

Academic Editors: Artem Bykov, Svetlana Polukoshko and Valery Y. Glizer

Received: 15 September 2023

Revised: 18 October 2023

Accepted: 24 October 2023

Published: 27 October 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Partial singular value assignment (PSVA) is a powerful technique with application in a variety of fields, including control theory, optimization problems, signal processing and many more [1–10]. PSVA focuses on selectively modifying the singular values of a matrix while keeping others unchanged. This technique enables engineers, researchers, and practitioners to address various problems in a nuanced and efficient manner. Notably, PSVA plays a pivotal role in minimizing errors when designing observers for the following discrete-time descriptor systems

$$\begin{aligned} Ez_{k+1} &= Az_k + Bu_k + w_k \\ y_k &= Cz_k + r_k, \end{aligned}$$

where the coefficient matrices  $E, A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times l}$  and  $C \in \mathbb{R}^{m \times n}$ ,  $w_k \in \mathbb{R}^{n \times 1}$  and  $r_k \in \mathbb{R}^{m \times 1}$  are the system errors;  $z_k \in \mathbb{R}^{n \times 1}$ ,  $u_k \in \mathbb{R}^{l \times 1}$  and  $y_k \in \mathbb{R}^{m \times 1}$  are the state vector, input vector and output vector, respectively. This system, initially introduced by Luenberger in 1977 [11], has been extensively explored, particularly in cases involving the singular matrix  $E$ . Contemporary research has yielded a wealth of theoretical insights from control theorists and numerical analysts, covering the continuous-time descriptor systems. These investigations encompass a wide array of subjects, including controllability, observability [12,13], pole assignment [14,15], eigenstructure assignment [16], Kronecker's canonical form [17], solution algorithms [18] and the inequality method [19]. It is worth noting that many of the findings from these studies are applicable to discrete-time systems as well.

In the context of the discrete-time descriptor system, the observer can be described as follows

$$Ez_{k+1} + H(y_{k+1} - Cz_{k+1}) = Az_k + G(y_k - Cz_k) + Bu_k,$$

where  $H, G \in \mathbb{R}^{n \times m}$ . The resulting error vector,

$$e_k = z_k - x_k$$

satisfies the equality

$$(E - HC)e_{k+1} = (A - GC)e_k - Hr_{k+1} + Gr_k - w_k.$$

A key objective in PSVA is to find a matrix  $H$  such that the smallest singular values (which might be zeros) of  $(E - HC)$  are sufficiently large, or, equivalently, the norm of  $(E - HC)^{-1}$  is sufficiently small; it then greatly reduces the impact of unmodeled error terms [20,21]. For further insights into the error reduction achieved by PSVA in other optimal state-space controller designs, such as  $H_\infty$  and  $H_2$  systems, please refer to [22,23] for detailed discussions.

PSVA has another important application in numerical methods solving ordinary differential equations (ODEs) [24]. It is widely recognized that Euler’s method plays a crucial role in the context of ODEs. Despite its rapidly increasing error as calculations progress, Euler’s method stands out as one of the few options in generating initial values needed for higher-order methods. When we apply the explicit Euler method to a differential equation with initial values, the resulting numerical solution tends to underestimate the true solution. Conversely, when we employ the implicit Euler method, the numerical method tends to overestimate the true solution. This suggests that a combination of these two methods could lead to better results. However, straightforward averaging has its limitations. An intriguing alternative is to consider “randomly switching” between the two methods. If we average all possible outcomes generated by the “random switching”, the resulting method can be approximately tens or even hundreds of times more accurate than either the explicit or implicit Euler method. For example, consider a numerically linear method to solve the ODE given by

$$\dot{x} = -\lambda x \tag{1}$$

with  $\lambda$  being a constant; the numerical solution can be achieved by closing a feedback loop, following the principles of linear control theory [24].

In our discussion, we will leverage this control to modify the method so that singular values can be assigned. To illustrate this, we consider two linear methods

$$\begin{aligned} x_{k+1} &= Ax_k + bu_k, \\ y_k &= cx_k \end{aligned}$$

and

$$\begin{aligned} x_{k+1} &= Fx_k + gu_k, \\ y_k &= hx_k \end{aligned}$$

of solving the differential Equation (1), where  $b$ ,  $g$ ,  $c$  and  $h$  are vectors of compatible dimensions with the systems above. The matrices  $A$  and  $F$ , both belonging to  $\mathbb{R}^{n \times n}$ , possess properties determined by the chosen numerical method. However, they share the common attribute of having 1 as an eigenvalue with a multiplicity of 1. If there are other eigenvalues located on the unit circle, they also have a multiplicity of 1, while all remaining eigenvalues lie within the unit circle.

Similar to the improvements achieved with the Euler methods, a natural enhancement of the linear method can be obtained by randomly switching between these two linear methods. In practice, the feedback loop in control theory yields the following results:

$$x_{k+1} = (A - \lambda bc)x_k$$

and

$$x_{k+1} = (F - \lambda gh)x_k.$$

Let  $\delta_i$  be a random variable with values in  $\{0, 1\}$ . The “random switching” iteration

$$x_{k+1} = \prod_{i=0}^k (A - \lambda bc)^{\delta_i} (F - \lambda gh)^{1-\delta_i} x_0$$

depends heavily on the product

$$T_k = \prod_{i=0}^k (A - \lambda bc)^{\delta_i} (F - \lambda bc)^{1-\delta_i}.$$

It has been shown in [25] that the “random switching” iteration converges if and only if the 2-norm of  $T_k$  is less than 1. Since the 2-norm of  $T_k$  associates with the largest singular value, it raises the question of when the singular values of a linear system can be assigned through the state feedback.

The above applications indicate that the PSVA problem involves assigning several desired singular values and determining how the remaining singular values behave. In mathematics, PSVA can be expressed explicitly as follows.

**PSVA:** Given matrices  $A, B$  and the desired (non-zero) singular values  $\{\theta_1, \dots, \theta_p\}$ , find a matrix  $F$  such that

$$A + BF^T$$

has the  $p$  desired singular values.

This problem was initially addressed in [20] for small and medium-scale systems. Typically, one needs a full QR decomposition of matrix  $B$ , i.e.,

$$\begin{bmatrix} Q_B \\ \bar{Q}_B \end{bmatrix} B = \begin{bmatrix} R_B \\ 0 \end{bmatrix}$$

and a full singular value decomposition (SVD)  $\bar{Q}_B A$  to construct the matrix  $F$ . If all singular values need to be reassigned, the interlacing theory on singular values of  $A + BF$  and an SVD of the orthogonal projection of  $A$  onto the null space of  $B$  are required for the formation of  $F$  [26]. However, for large-scale systems, methods related to the full SVD of a large matrix are not feasible due to the computational complexity of  $O(n^3)$  and memory limitations. Similar issues were encountered in pole assignment [14,15,27,28]. Readers can find details in [29–32] and their references.

In this paper, we introduce an efficient algorithm to conduct PSVA on large-scale systems. The primary contributions and advantages of our algorithm are listed below.

- The core innovation of our devised approach is rooted in the construction of a sparse basis within the null space of the orthogonal projection of matrix  $A$ . This strategy offers a notable advantage by obviating the necessity for the full SVD of a large-scale matrix.
- Our algorithm brings to the forefront the possibility of computational savings, especially when dealing with matrices of significant scale and sparse structure (i.e., the number of non-zeros in each column of the matrix is bounded by a constant much smaller than  $n$ ). Then, the proposed algorithm strives to assign the desired singular values with exceptional computational efficiency, hopefully scaling with the computational complexity of  $O(n)$  floating-point operations.
- To validate the practical utility of our proposed algorithm, we conduct a series of numerical experiments. These experiments not only confirm the feasibility of our approach but also demonstrate its effectiveness in real-world scenarios.

Throughout this paper, we denote the  $i$ -th singular value, 2-norm and the condition number of  $A \in \mathbb{R}^{n \times n}$  by  $\sigma_i(A)$ ,  $\|A\|_2 = \sigma_1(A)$  (the maximal singular value) and  $\kappa_2(A) = \sigma_1(A)/\sigma_n(A)$  (the ratio of the maximal and the minimal singular value), respectively. The determinant of the matrix  $A$  is represented by  $\det(A)$ . The equivalence of two pairs of matrices is defined as follows.

**Definition 1** ([33]). *Two pairs of matrices  $(A, B)$  and  $(C, D)$  are equivalent if there are orthogonal matrices  $U$  and  $V$ , a matrix  $E$  and a nonsingular matrix  $R$  with compatible sizes such that*

$$\begin{aligned} C &= U(A + BE)V, \\ D &= UBR. \end{aligned}$$

The equivalence of matrix pairs is attractive as it preserves the same assignability of singular values. Thus, one can use the simplest representation to describe a class of equivalent pairs and study the existence of PSVA.

### 2. PSVA for Single Singular Value

We first delve into the single PSVA. This problem can be simplified to the following.

**Single PSVA:** Given a matrix  $A \in \mathbb{R}^{n \times n}$ , a vector  $b \in \mathbb{R}^{n \times 1}$  and the desired (non-zero) singular values  $\theta$ , find a vector  $f \in \mathbb{R}^{n \times 1}$  such that

$$A + bf^T$$

includes the desired singular value  $\theta$ .

To solve the single PSVA, we initiate the process by ensuring that the smallest singular value of a matrix associated with  $A$  is in proximity to zero, or even equal to zero. Once this condition is met, we can proceed with constructing the orthogonal projection of matrix  $A$  onto the null space of vector  $b$  by using

$$f_1 = -((b^T b)^{-1} b^T A)^T, \tag{2}$$

which effectively enforces a minimal singular value of zero. In fact, by performing the QR decomposition of  $b$  by

$$b = Qr = [q_1, Q_2] \begin{bmatrix} r \\ 0 \end{bmatrix} = q_1 r, \tag{3}$$

where  $q_1 \in \mathbb{R}^{n \times 1}$  and  $Q_2 \in \mathbb{R}^{n \times (n-1)}$ , we have the smallest singular value of

$$D = A + bf_1^T = (I - b(b^T b)^{-1} b^T)A = (I - q_1 q_1^T)A = (Q_2 Q_2^T)A$$

forced to be zero. Let the SVD of  $D$  be given by

$$U^T D V = \begin{bmatrix} \Sigma_1 & \\ & 0 \end{bmatrix}, \tag{4}$$

where  $U = [U_1, u_2]$  and  $V = [V_1, v_2]$  are orthogonal matrices with  $U_1; V_1 \in \mathbb{R}^{n \times (n-1)}$ ,  $u_2, v_2 \in \mathbb{R}^{n \times 1}$ ,  $\Sigma_1 = \text{diag}\{\sigma_n, \dots, \sigma_2\} \in \mathbb{R}^{(n-1) \times (n-1)}$  is a diagonal matrix with its  $n - 1$  non-zero diagonal elements being the singular values of  $D$ . We can demonstrate the existence of the single PSVA through the following theorem.

**Theorem 1.** For a given singular value  $\theta > 0$ , there exists a vector  $f \in \mathbb{R}^{n \times 1}$  such that the singular values of  $A + bf^T$  are  $\{\sigma_n, \dots, \sigma_2, \theta\}$ .

**Proof.** By using the SVD of  $D$  in (4), a constant  $c \neq 0$  can be found such that

$$U^T b c = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

We can then define vectors  $f_2 = V\omega^T$  with  $\omega = [\omega_1, \dots, \omega_n] \in \mathbb{R}^{1 \times n}$  and  $f = f_1 + f_2$ . As a result, we have

$$A + bf^T = U\Sigma V^T$$

with

$$\Sigma = \begin{bmatrix} \sigma_n & & & & \\ & \ddots & & & \\ & & \sigma_2 & & \\ z_n & \cdots & z_2 & z_1 & \end{bmatrix}$$

and  $z_i = \omega_i/c$  for  $i = 1, \dots, n$ . The assignment of the singular value  $\theta$  in  $A + bf^\top$  is now reduced to the assignment in  $\Sigma$  according to Definition 1.

Now, we consider the characteristic polynomial of the matrix  $\delta I - \Sigma\Sigma^\top$ . It follows

$$\begin{aligned} p(\delta) &= \det(\delta I - \Sigma\Sigma^\top) \\ &= \det \left( \begin{bmatrix} \delta - \sigma_n^2 & & & -\sigma_n z_n \\ & \ddots & & \vdots \\ & & \delta - \sigma_2^2 & -\sigma_2 z_2 \\ -\sigma_n z_n & \cdots & -\sigma_2 z_2 & \delta - \sum_{i=1}^n z_i^2 \end{bmatrix} \right) \\ &= \left( \delta - \sum_{i=1}^n z_i^2 \right) \prod_{i=2}^n (\delta - \sigma_i^2) - z_2^2 \sigma_2^2 \prod_{j \neq 2} (\delta - \delta_j^2) - \dots \\ &\quad - z_n^2 \sigma_n^2 \prod_{j \neq n} (\delta - \delta_j^2) \\ &= \delta \prod_{i=2}^n (\delta - \delta_i^2) - \sum_{i=1}^n z_i^2 \prod_{j \neq i} (\delta - \delta_j^2) \\ &= \delta \prod_{i=2}^n (\delta - \delta_i^2) \left( 1 - \frac{z_1^2}{\delta} - \sum_{i=2}^n \frac{z_i^2}{\delta - \sigma_i^2} \right). \end{aligned}$$

By letting

$$p(\theta^2) = 0 \tag{5}$$

with  $0 < \theta < \sigma_2$ , we see that there are multiple solutions to Equation (5). One simple choice is to take  $z_1 = \theta$  and  $z_2 = z_3 = \dots = z_n = 0$ . By setting  $f_2 = V[0, \dots, 0, \theta]^\top$ , we have the desired vector of  $f = f_1 + f_2$  to complete the PSVA for a single singular value.  $\square$

**Remark 1.** Although there are multiple solutions to Equation (5), e.g., the vector  $z$  could be also taken as

$$[\theta/\sqrt{n}, \sqrt{(\theta^2 - \sigma_2^2)/n}, \dots, \sqrt{(\theta^2 - \sigma_n^2)/n},$$

the choice of  $z = [0, \dots, 0, \theta]$  in Theorem 1 is the most valid as it ensures that the other  $\sigma_i^2$  ( $i \geq 2$ ) is also a root of  $p(\delta) = 0$ .

The above theorem provides the motivation for constructing the vector  $f$  for the single PSVA. In fact, by equating the two sides of the SVD of  $D$  in (4), it follows that

$$DV_1 = U_1\Sigma_1, \quad Dv_2 = 0.$$

By taking  $u_2$  and  $f_2$  as  $q_1$  and  $\theta v_2$ , respectively, then, for a singular value of  $\theta > 0$  to be assigned, we have

$$\begin{aligned} U^\top(D + q_1f_2^\top)V &= \begin{bmatrix} \Sigma_1 + U_1^\top q_1g^\top V_1 & U_1^\top q_1g^\top v_2 \\ u_2^\top q_1g^\top V_1 & u_2^\top q_1g^\top v_2 \end{bmatrix} \\ &= \begin{bmatrix} \Sigma_1 \\ \theta \end{bmatrix} \end{aligned}$$

and the vector

$$f = f_1 + \frac{\theta v_2}{r}$$

completes the single PSVA. Algorithm 1 outlines the concrete steps involved in the single PSVA.

---

**Algorithm 1 PSVA for the Single Singular Value**

---

Input: A large-scale sparse matrix  $A \in \mathbb{R}^{n \times n}$ ,  $b \in \mathbb{R}^{n \times 1}$  and the desired singular value  $\theta$ .  
 Output: A vector  $f \in \mathbb{R}^{n \times 1}$  such that the minimal singular value of  $A + bf^\top$  is  $\theta$ .

1. Compute  $f_1 = -((b^\top b)^{-1}b^\top A)^\top$ ;
  2. Compute the economic QR decomposition of  $b$ , i.e.,  $b = q_1 r$ ;
  3. Compute the null space of the matrix  $D = A + bf_1^\top$ ;
  4. Form the vector  $f = f_1 + \theta v_2 / r$  such that the smallest singular value of  $A + bf^\top$  is  $\theta$ .
- 

**Remark 2.** (1) The QR decomposition of  $b$  has a complexity of  $O(2(n - 1/3))$  when the Householder transformation is used [34].

(2) The vector  $v_2$  in the null space of

$$D = (I - q_1 q_1^\top)A$$

should be as sparse as possible. Two methods have been tested in our experiments. One involves computing the complement of  $D^\top$  using the QR decomposition, while the other involves computing the LU factorization. If the elements in each column of  $A$  are sparse, with an upper bound of  $a$  (which is much less than  $n$ ), then the computational complexity may hopefully be  $O(an)$ .

**3. PSVA for  $p$  Singular Values**

The single PSVA can be readily generalized to the  $p$  singular value assignment with  $p > 1$ .

**$p$ -PSVA:** Given a large-scale sparse matrix  $A \in \mathbb{R}^{n \times n}$  with  $p$  smallest singular values  $\{\sigma_1, \dots, \sigma_p\}$  being close to (or equal to) zero, and a matrix  $B \in \mathbb{R}^{n \times p}$ , find a matrix  $F \in \mathbb{R}^{n \times p}$  such that

$$A + BF^\top$$

its  $p$  smallest singular values are the given  $\{\theta_p, \dots, \theta_1\}$ .

Let us construct the matrix

$$F_1 = -((B^\top B)B^\top A)^\top,$$

which is the orthogonal projection of  $A$  on the null space of  $B$ . By using the QR decomposition of  $B$

$$\text{(i.e., } B = QR = [Q_1, Q_2] = \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R)$$

with  $Q_1 \in \mathbb{R}^{n \times p}$ , the matrix

$$D = A + BF_1^\top = (I - Q_1 Q_1^\top)A$$

will have  $p$  zero singular values. Let the SVD of  $D$  be

$$U^\top D V = \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix}$$

with the diagonal matrix  $\Sigma_1 \in \mathbb{R}^{(n-p) \times (n-p)}$  having the diagonal elements  $\{\sigma_n, \dots, \sigma_{p+1}\}$ . The following theorem provides the existence of the PSVA.

**Theorem 2.** For singular values  $\{\theta_1, \dots, \theta_p\}$  to be assigned, there exists a matrix  $F \in \mathbb{R}^{n \times p}$  such that the singular values of  $A + BF^\top$  are  $\{\sigma_n, \dots, \sigma_{p+1}, \theta_p, \dots, \theta_1\}$ .

**Proof.** Set

$$\Sigma_{(i)} = \begin{bmatrix} & \Sigma_1 & & 0 \\ z_{pn} & \cdots & z_{pp} & \cdots & z_{p1} \\ \vdots & & \vdots & & \vdots \\ z_{in} & \cdots & z_{ip} & \cdots & z_{i1} \\ 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix}_{n \times n}$$

for  $1 \leq i \leq p$ . According to Definition 1, the form of  $z_{ij}$  must be determined such that the singular values of  $A + BF^T$  are those of  $\Sigma_{(i)}$ . Firstly, consider  $i = p$ . By setting  $z_p = [z_{pn}, \dots, z_{pp}, \dots, z_{p1}]$  with  $z_{pp} = \theta_p$  and other elements being zeros, Theorem 1 indicates that there exists a vector  $f_p$  such that the non-zero singular values of  $A + B(F_1 + [f_p, 0, \dots, 0])^T$  are  $\{\sigma_n, \dots, \sigma_{p+1}, \theta_p\}$ . By using induction and Theorem 1 continuously, it can be shown that there is a matrix  $F_2 = [f_p, f_{p-1}, \dots, f_1] \in \mathbb{R}^{n \times p}$  such that the singular values of  $A + BF^T (F = F_1 + F_2)$  are  $\{\sigma_n, \dots, \sigma_{p+1}, \theta_p, \dots, \theta_1\}$ .  $\square$

We can further elaborate on the form of  $F$  in terms of Theorem 2. In fact, let the orthogonal matrices  $U$  and  $V$  in (4) be given as

$$U = [U_1, U_2] \text{ and } V = [V_1, V_2]$$

with  $U_2, V_2 \in \mathbb{R}^{n \times p}$ ,  $U_1, V_1 \in \mathbb{R}^{n \times (n-p)}$ , respectively. Then, by setting  $U_2 = Q_1$  and  $F_2 = -(R^{-1}\Sigma_2V_2^T)^T$ , we have

$$\begin{aligned} U^T(A + BF^T)V &= \begin{bmatrix} \Sigma_1 + U_1^T Q_1 \Sigma_2 V_2^T V_1 & U_1^T Q_1 \Sigma_2 V_2^T V_2 \\ U_2^T Q_1 \Sigma_2 V_2^T V_1 & U_2^T Q_1 \Sigma_2 V_2^T V_2 \end{bmatrix} \\ &= \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix}, \end{aligned}$$

where  $\Sigma_2$  represents a diagonal matrix, with its diagonal elements containing the desired non-zero singular values  $\theta_1, \dots, \theta_p$ . Furthermore, the matrix  $F$  can be expressed as

$$F = -\left(R^{-1}(Q_1^T A - \Sigma_2 V_2^T)\right)^T.$$

The steps involved in the PSVA for  $p$  singular values are outlined in Algorithm 2.

A crucial aspect of Algorithm 2 is to minimize the sparsity of the computation for the null sparse basis of matrix  $D$ . While the heuristic method suggested in [20] is an option, empirical testing has revealed that QR decomposition or LU factorization in MATLAB can produce adequately sparse vectors, particularly when the matrix  $A$  possesses a significant degree of sparsity.

**Remark 3.** As indicated by Remark 2, the computational complexity of Algorithm 2 is advantageous, with a time complexity of  $O(apn)$  when the matrix  $A$  exhibits a degree of sparsity (characterized by relatively small values of  $a$  and  $p$  in comparison to the matrix size  $n$ ). In this case, Algorithm 2 offers a substantial advantage over methods relying on the full SVD, which typically entail a much higher time complexity of  $O(n^3)$ . This stark contrast positions Algorithm 2 as a clear choice for the handling of large-scale systems, where the efficiency and computational speed are obviously superior. Moreover, when the matrix  $A$  reaches a considerable order (scaling up to the thousands), methods that rely on the full SVD approach tend to become prohibitively time-consuming, particularly when executed on standard personal computers. This temporal limitation restricts their applicability and practicality. This also the reason for exclusively showing the performance of our algorithm through numerical experiments on large-scale problems.

---

**Algorithm 2 PSVA for  $p$  Singular Values**

---

Input: A large-scale sparse matrix  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times p}$  and the desired singular values  $\theta_1, \dots, \theta_p$ .

Output: A matrix  $F \in \mathbb{R}^{n \times p}$  such that  $p$  smallest singular values of  $A + BF^\top$  are  $\{\theta_1, \dots, \theta_p\}$ .

1. Compute the economic QR decomposition of  $B$ , i.e.,  $B = Q_1 R_1$  with  $Q_1 = [Q_{11}, \dots, Q_{1p}]$  and

$$R = \begin{bmatrix} r_{11} & \cdots & \cdots & r_{1p} \\ & \ddots & & \vdots \\ & & r_{p-1,p-1} & \vdots \\ & & & r_{pp} \end{bmatrix}.$$

2. Compute the null space of  $DV_2 = (I - Q_1 Q_1^\top) A V_2 = 0$  with  $V_2 = [V_{21}, \dots, V_{2p}]$ .
3. Compute  $f_p$  by Algorithm 1 with the available  $A, Q_{1p}$  and  $\theta_p$ .
4. For  $i = p - 1 : 1$

$$f_i = \frac{1}{r_{ii}} \left[ \theta_i V_{2i} - A^\top Q_{1i} - \sum_{j=i+1}^p r_{ij} f_j \right]$$

End.

5. Form the matrix  $F = [f_1, \dots, f_p]$  such that the smallest  $p$  singular values of  $A + BF^\top$  are  $\{\theta_1, \dots, \theta_p\}$ .
- 

Another consideration in PSVA is to keep the condition number of the assigned matrix  $A + BF^\top$  relatively low. This can be achieved through optimization methods based on heuristic rules [29,30]. For instance, Shields [21] employed the optimization of some matrix function over a set of assigned matrices with a fixed condition number. However, this approach becomes unfeasible for large-scale problems due to the full SVD of a large matrix. On the other hand, if the singular values assigned are well separated from those in  $\Sigma_1$ , the following theorem shows that the condition number of  $A + BF^\top$  will be no worse than that of  $\Sigma_1$ .

**Theorem 3 ([21]).** For any matrix

$$P = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}$$

with  $P_2$  being nonsingular and

$$\sigma(A + BF^\top) \geq \sigma \left( \begin{bmatrix} \Sigma_1 & P_1 \\ & P_2 \end{bmatrix} \right),$$

the matrix  $F$  in the PSVA satisfies

$$\kappa_2(A + BF^\top) \geq \kappa_2(\Sigma_1).$$

Furthermore, there exists a non-unique matrix  $F$  to make the equality hold.

**Remark 4.** From Theorem 3, it follows that  $A + BF^\top$  will have a poor condition number if the gap between the maximal and the minimal singular values of  $\Sigma_1$  is large. Conversely, if suitable singular values  $\theta_1, \dots, \theta_p$  and a proper  $F$  are chosen to implement the PSVA, the obtained condition number of  $A + BF^\top$  may be as close to that of  $\Sigma_1$  as possible. This fact can be used to assess the effectiveness of Algorithm 2 in the next section.

**4. Numerical Examples**

In this section, we examine the efficacy of Algorithm 2 in addressing large-scale PSVA problems. The algorithm was coded by MATLAB 2014 and all examples were executed



on a desktop equipped with an Intel i5 3.4 GHz processor and 16 GB RAM. We denote the authentically assigned singular value by  $\theta$  and the computed singular value by  $\hat{\theta}$ . The relative error is computed as

$$ERR = \frac{|\theta - \hat{\theta}|}{\theta}$$

when the matrix  $F$  is at hand.

We opted not to carry out a direct comparison between Algorithm 2 and the methods based on full SVD. The rationale behind this choice is grounded in the fact, as stated in Remark 3, these conventional approaches can become excessively time-consuming, particularly when executed on typical personal desktop setups.

**Example 1.** For  $i = 1, \dots, n$ , we set  $a_i = 10 + 90\zeta_i$  and  $b_i = \eta_i$ , where  $\zeta_i$  and  $\eta_i$  are random numbers drawn from a uniformed distribution over the interval  $[0, 1]$  (generated by the MATLAB command “rand(n, 1)”). We define the matrices

$$A = \begin{bmatrix} a_1 & b_1 & & & & & & & & \\ & \ddots & \ddots & & & & & & & \\ & & \ddots & \ddots & & & & & & \\ & & & \ddots & \ddots & & & & & \\ & & & & \ddots & b_{n-1} & & & & \\ & & & & & a_n & & & & \end{bmatrix}$$

and

$$B^T = \begin{bmatrix} 1 & -1 & 0 & 0 & \dots & 0 & 0 & 1 & -1 \\ 0 & -1 & 0 & 1 & \dots & 1 & 0 & -1 & 1 \\ -1 & 0 & 1 & 0 & \dots & 0 & -1 & 1 & 0 \\ 1 & 0 & -1 & 1 & \dots & -1 & 0 & 1 & 1 \\ 0 & 1 & -1 & 0 & \dots & 1 & -1 & 0 & -1 \end{bmatrix}.$$

The singular values to be assigned are  $[\theta_1, \dots, \theta_5] = [2.2, 4.3, 5.5, 7.6, 8.2]$ .

Algorithm 2 was utilized to carry out the PSVA, and 50 random experiments were conducted for  $n = 10,000$ . The resulting errors are depicted in Figure 1, with the vertical axis representing the error level and the horizontal axis reflecting the number of experiments. The assigned singular values are  $[Del(1), \dots, Del(5)]$ , corresponding to  $[\theta_1, \dots, \theta_5]$ . It is evident from Figure 1 that all errors are within a range of  $3 \times 10^{-14}$ . Furthermore, it is noteworthy that among 50 experiments, the errors of the assigned singular values  $Del(1)$  (represented by the star line) and  $Del(2)$  (represented by the diamond line) fluctuate more dramatically compared to the other errors, while the errors of  $Del(5)$  (represented by the cross line) fluctuate the least.

In Figure 2, we also list the computed CPU time and condition numbers of matrices  $D$  (CondS) and  $A + BF^T$  (CondPSVA), with all data on the vertical axis being recorded in log 10 to minimize the gaps between the various lines. It can be seen that the CPU time of all tests is approximately 0.7 s, and the range of CondPSVA lies between 48.1 and 51.5, compared to that of CondS, which lies between 8.81 and 13.43. This highlights the efficiency of Algorithm 2 in carrying out PSVA. It is worth mentioning that the condition number of  $A + BF^T$  is closely linked to the assigned singular values. In our experiments, we observed that CondPSVA approached CondS more closely when the minimal singular values to be assigned were larger.



that all of the errors fall well below the threshold of  $2 \times 10^{-13}$ , suggesting a high degree of accuracy in the assignment of singular values. Furthermore, the range of fluctuation among these errors is relatively small. In other words, the variations between the assigned singular values and the desired values are quite minor, reinforcing the precision and consistency of the assignments.

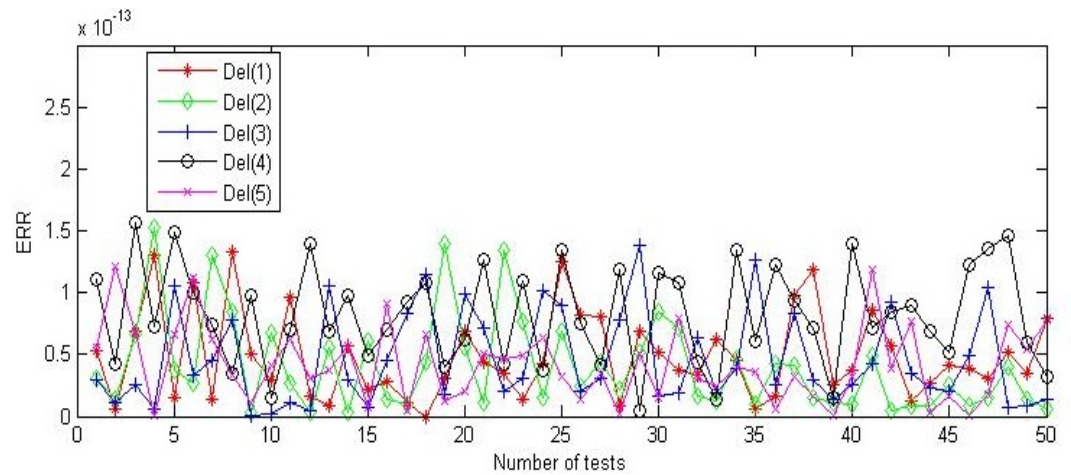


Figure 3. Error in 50 random tests in Example 2.

We also present the consumed CPU time in Figure 4, which shows that the range varies from 0.40 to 0.46 s. The obtained CondS and CondPSVA of 50 random experiments fall within the interval [4.47, 9.11] and [366.53, 435.61], respectively (with the exception of the value 160.74 in the 35th experiment). Analogously, the value of CondPSVA can be reduced further by assigning relatively larger singular values in  $\Sigma_1$ . In fact, if the assigned singular values are increased to 160, 162, 164, 166 and 168, the condition number of  $A + BF^T$  in 50 random experiments will fall within the interval [8.05, 12.30].

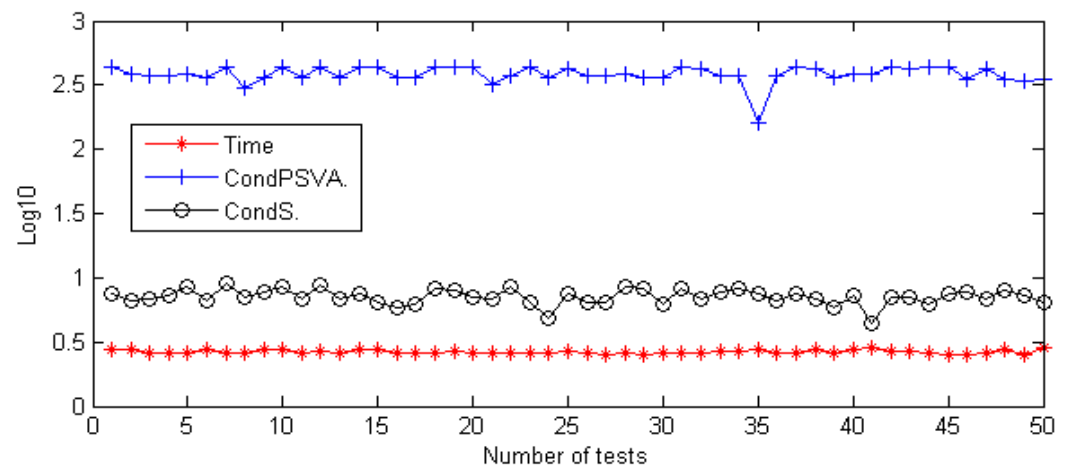


Figure 4. Condition number and time of 50 random tests in Example 2.

**Example 3.** In this example, we seek to affirm the robustness and efficiency of Algorithm 2 by subjecting it to real-world testing. The examination involves the application of this algorithm to nine large-scale sparse matrices, which have been drawn from practical applications. These specific matrices are conveniently available in the SuiteSparse Matrix Collection, which was previously known as the University of Florida Sparse Matrix Collection, accessed via the following link: <https://sparse.tamu.edu/>, accessed on 10 September 2023.

The matrices under scrutiny bear the following names: ‘dubcova1’, ‘gridgena’, ‘onetone2’, ‘wathen100’, ‘poli\_large’, ‘msc10848’, ‘bodyy4’, ‘bodyy5’ and ‘bodyy6’. Each of these matrices varies in size, catering to the demands of diverse applications. The dimensions of these matrices are as

follows:  $16,129 \times 16,129$ ,  $48,962 \times 48,962$ ,  $36,057 \times 36,057$ ,  $30,401 \times 30,401$ ,  $15,575 \times 15,575$ ,  $10,848 \times 10,848$ ,  $17,546 \times 17,546$ ,  $18,589 \times 18,589$  and  $19,366 \times 19,366$ . Their sparse structures are plotted in Figure 5, where ‘nz’ indicates the non-zero elements.

In our experiments, the matrix

$$B^T = \begin{bmatrix} 1 & -1 & 0 & 0 & \dots & 0 & 0 & 1 & -1 \\ 0 & 1 & -1 & 1 & \dots & 1 & 0 & -1 & 1 \end{bmatrix}$$

remains invariant for all different problems, but the two assigned singular values vary for each matrix.

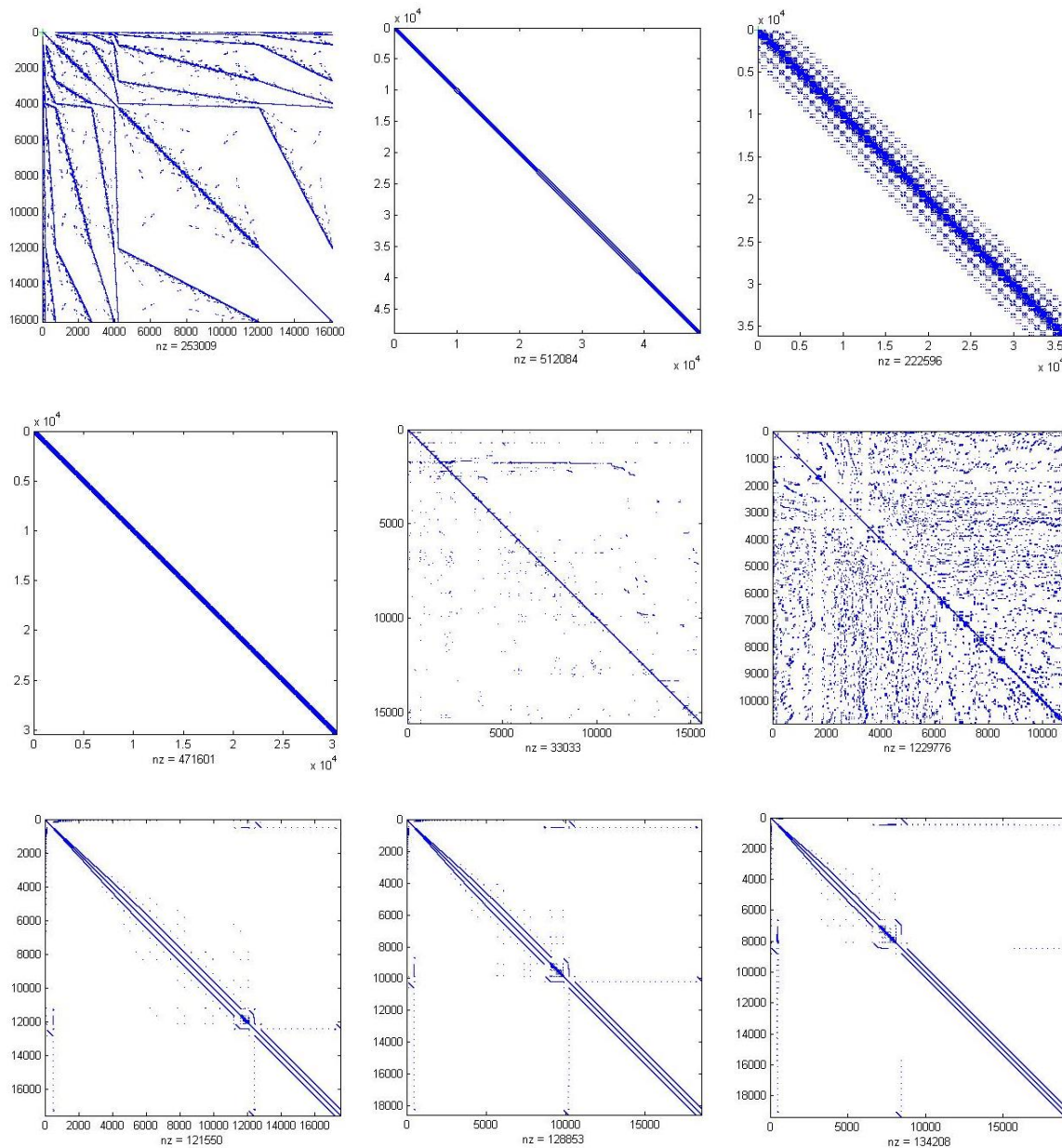


Figure 5. Sparse structures of nine matrices in Example 3.

We ran Algorithm 2 in dealing with the nine large-scale matrices and the obtained results are exhibited in Table 1. The CPU column in the table provides information regarding the elapsed time, measured in seconds, required to perform the PSVA. The DesSV and AsiSV columns, respectively, denote the two sets of singular values, one representing the desired values (DesSV) and the other representing the values that were actually assigned

(AsiSV). These columns allow us to assess how closely the assigned singular values align with the desired values. They serve as a crucial indicator of the accuracy and precision of the PSVA process. The next column, labeled RelErr, offers insights into the relative errors encountered during the PSVA computation. A smaller relative error indicates the more accurate assignment of singular values. The final two columns, CondS and CondPSVA, provide information about the condition numbers of two distinct entities. CondS represents the condition number of  $\Sigma_1$ , while CondPSVA represents the condition number of the matrix  $A + BF^T$ . These condition numbers help to assess the numerical stability and quality of the PSVA solution. Smaller condition numbers are indicative of more favorable and robust solutions.

**Table 1.** Assigned results for nine large-scale matrices in Example 3.

| Problem    | CPU  | DesSV        | AsiSV                                      | RelErr                                       | CondS              | CondPSVA           |
|------------|------|--------------|--|--|--------------------|--------------------|
| dubcova1   | 1.29 | 0.016; 0.015 | $1.59 \times 10^{-2}; 1.49 \times 10^{-2}$ | $1.21 \times 10^{-13}; 3.34 \times 10^{-13}$ | $2.74 \times 10^2$ | $3.19 \times 10^2$ |
| gridgena   | 9.77 | 0.16; 0.15   | $1.59 \times 10^{-1}; 1.49 \times 10^{-1}$ | $1.08 \times 10^{-6}; 2.52 \times 10^{-7}$   | $1.55 \times 10^5$ | $1.84 \times 10^5$ |
| onetone2   | 2.48 | 0.001; 0.005 | $1.00 \times 10^{-3}; 5.00 \times 10^{-3}$ | $9.39 \times 10^{-15}; 8.76 \times 10^{-15}$ | $3.08 \times 10^6$ | $3.92 \times 10^6$ |
| wathen100  | 4.29 | 0.10; 0.05   | $1.00 \times 10^{-1}; 5.00 \times 10^{-2}$ | $9.99 \times 10^{-15}; 3.99 \times 10^{-14}$ | $5.81 \times 10^3$ | $7.39 \times 10^3$ |
| poli_large | 0.06 | 0.010; 0.005 | $1.00 \times 10^{-2}; 5.00 \times 10^{-3}$ | $5.63 \times 10^{-15}; 6.99 \times 10^{-15}$ | $2.51 \times 10^3$ | $3.75 \times 10^3$ |
| msc10848   | 4.57 | 80.0; 81.0   | $7.99 \times 10^1; 8.09 \times 10^1$       | $8.06 \times 10^{-9}; 7.34 \times 10^{-9}$   | $7.60 \times 10^9$ | $7.86 \times 10^9$ |
| bodyy4     | 0.95 | 1.0; 0.9     | $9.99 \times 10^{-1}; 8.99 \times 10^{-1}$ | $3.99 \times 10^{-15}; 3.99 \times 10^{-1}$  | $8.06 \times 10^2$ | $9.24 \times 10^2$ |
| bodyy5     | 1.21 | 1.0; 0.9     | $1.00 \times 10^0; 8.99 \times 10^{-1}$    | $2.66 \times 10^{-14}; 8.99 \times 10^{-14}$ | $7.87 \times 10^3$ | $8.93 \times 10^3$ |
| bodyy6     | 1.11 | 1.0; 0.9     | $9.99 \times 10^{-1}; 8.99 \times 10^{-1}$ | $1.29 \times 10^{-14}; 9.99 \times 10^{-15}$ | $7.69 \times 10^4$ | $8.68 \times 10^4$ |

It can be seen that Algorithm 2 is efficient in assigning the required two singular values in a relatively short CPU time for all matrices. The RelErr column indicates that the relative error is very small, except for two matrices, ‘gridgena’ and ‘msc10848’, with respective levels  $10^{-7}$  and  $10^{-9}$ . The CondS column lists the condition numbers of  $\Sigma_1$  for different matrices, where a larger order means a wider gap between the maximal and the minimal singular values. The CondPSVA column records the condition number of  $A + BF^T$  after the assignment, showing a relatively modest increase in the condition number from CondS. This also reflects that Algorithm 2 is efficient in implementing PSVA, as stated in Remark 4, and provides valuable insights into the algorithm’s capacity to handle complex, large-scale, real-world systems.

### 5. Conclusions

In the context of large-scale discrete-time systems, a novel algorithm has been introduced to facilitate partial singular value assignment (PSVA), particularly when the matrices  $A$  and  $B$  exhibit a significant degree of sparsity. This method leverages the capabilities of a sparse solver and incorporates the orthogonal projection of matrix  $A$  onto the null space of matrix  $B$  as a fundamental component. Crucially, we have established the existence of a matrix  $F$  that plays a pivotal role in both single and multiple PSVA problems. This implies that it can be applied effectively to address a range of singular value assignment scenarios, offering flexibility and versatility in practical applications.

The proposed approach is validated through a series of numerical experiments. The derived results affirm the algorithm’s efficacy in achieving the partial assignment of desired singular values, thereby underscoring its real-world applicability for large-scale discrete-time systems. For future research, we will explore the possibility to tackle more complex and practical systems in various fields, from control theory to areas of signal processing.

**Author Contributions:** Conceptualization, B.Y.; methodology, B.Y.; software, Y.H.; validation, Y.H.; and formal analysis, Q.T. All authors have read and agreed to the final version of this manuscript.

**Funding:** This work was supported partly by the NSF of Hunan Province (2021JJ50032, 2023JJ50164, 2023JJ50165), the NSF of China (12305016), Degree and Postgraduate Education Reform Project of Hunan University of Technology and Hunan Province (JG2315, 2023JGYB210).

**Acknowledgments:** We are grateful to the anonymous referees for their useful comments and suggestions, which significantly enhanced the quality the original paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Bangle, H.E. Feature extraction of cable partial discharge signal based on DT\_CWT\_Hankel\_SVD. In Proceedings of the 2019 6th International Conference on Information Science and Control Engineering (ICISCE), Shanghai, China, 20–22 December 2019; pp. 1063–1067.
- Cooke, S.A.; Minei, A.J. The pure rotational spectrum of 1, 1, 2, 2, 3-pentafluorocyclobutane and applications of singular value decomposition signal processing. *J. Mol. Spectrosc.* **2014**, *306*, 37–41. [[CrossRef](#)]
- Datta, B.N. Linear and numerical linear algebra in control theory: Some research problems. *Linear Algebra Its Appl.* **1994**, *197*, 755–790. [[CrossRef](#)]
- Hou, Z.; Jin, S. *Model Free Adaptive Control: Theory and Applications*; CRC Press: Boca Raton, FL, USA, 2013.
- Sontag, E.D. *Mathematical Control Theory: Deterministic Finite Dimensional Systems*; Springer Science & Business Media: New York, NY, USA, 2013; Volume 6.
- Sogaard-Andersen, P.; Trostmann, E.; Conrad, F. A singular value sensitivity approach to robust eigenstructure assignment. In Proceedings of the 1986 25th IEEE Conference on Decision and Control, Athens, Greece, 10–12 December 1986; pp. 121–126.
- Van Dooren, P. Some numerical challenges in control theory. In *Linear Algebra for Control Theory*; Springer: New York, NY, USA, 1994; pp. 177–189.
- Weinmann, A. *Uncertain Models and Robust Control*; Springer Science & Business Media: Vienna, Austria, 2012.
- Zimmerman, D.C.; Kaouk, M. Eigenstructure assignment approach for structural damage detection. *AIAA J.* **1992**, *30*, 1848–1855. [[CrossRef](#)]
- Saraf, P.; Balasubramaniam, K.; Hadidi, R.; Makram, E. Partial right eigenstructure assignment based design of robust damping controller. In Proceedings of the 2016 IEEE Power and Energy Society General Meeting (PESGM), Boston, MA, USA, 17–21 July 2016; pp. 1–5.
- Luenberger, D. Dynamic equations in descriptor form. *IEEE Trans. Autom. Control* **1977**, *22*, 312–321. [[CrossRef](#)]
- Ishihara, J.Y.; Terra, M.H. Impulse controllability and observability of rectangular descriptor systems. *IEEE Trans. Autom. Control* **2001**, *46*, 991–994. [[CrossRef](#)]
- Feng, Y.; Yagoubi, M. *Robust Control of Linear Descriptor Systems*; Springer: Singapore, 2017.
- Chu, E.K. A pole-assignment algorithm for linear state feedback. *Syst. Control. Lett.* **1986**, *7*, 289–299. [[CrossRef](#)]
- Chu, E.K. A pole-assignment problem for 2-dimensional linear discrete systems. *Int. J. Control* **1986**, *43*, 957–964. [[CrossRef](#)]
- Duan, G.R. Eigenstructure assignment in descriptor systems via output feedback: A new complete parametric approach. *Int. J. Control* **1999**, *72*, 345–364. [[CrossRef](#)]
- Kumar, A.; Pait, i M.; Gupta, M.K. Analysis of impulsive modes in Kronecker Canonical form for rectangular descriptor systems. In Proceedings of the IEEE 8th Indian Control Conference (ICC), Chennai, India, 14–16 December 2022; pp. 349–354.
- Shahzad, A.; Jones, B.L.; Kerrigan, E.C.; Constantinides, G.A. An efficient algorithm for the solution of a coupled Sylvester equation appearing in descriptor systems. *Automatica* **2011**, *47*, 244–248. [[CrossRef](#)]
- Masubuchi, I.; Kamitane, Y.; Ohara, A.; Suda, N.  $H_\infty$  control for descriptor systems: A matrix inequalities approach. *Automatica* **1997**, *33*, 669–673. [[CrossRef](#)]
- Pearson, D.W.; Chapman, M.J.; Shields, D.N. Partial singular-value assignment in the design of robust observer for discrete-time descriptor systems. *IMA J. Math. Control Inform.* **1988**, *5*, 203–213. [[CrossRef](#)]
- Shields, D.N. Singular-value assignment in the design of observers for discrete-time systems. *IMA J. Math. Control Inform.* **1991**, *8*, 151–164. [[CrossRef](#)]
- Hague, T.N. An Application of Robust  $H_2/H_\infty$  Control Synthesis to Launch Vehicle Ascent. Ph.D. Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA, 2000.
- Sun, X.D.; Clarke, T. Application of hybrid  $\mu H^\infty$  control to modern helicopters. In Proceedings of the International Conference on Control (Control'94 IET), Coventry, UK, 21–24 March 1994; Volume 2, pp. 1532–1537.
- Holder, D.; Luo, S.; Martin, C. The control of error in numerical methods, in modeling, estimation and control. In *Lecture Notes in Control and Information Sciences*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 183–192.
- Elsner, L.; Friedland, S. Norm conditions for the convergence of infinite products. *Linear Algebra Appl.* **1997**, *250*, 133–142. [[CrossRef](#)]
- Martin, C.F.; Wang, X.A. Singular value assignment. *SIAM J. Control Optim.* **2009**, *48*, 2388–2406. [[CrossRef](#)]
- Kautsky, J.; Nichols, N.K.; Van Dooren P. Robust pole assignment in linear state feedback. *Int. J. Control.* **1985**, *41*, 1129–1155. [[CrossRef](#)]
- Nichols, N.K. Robustness in partial pole placement. *IEEE Trans. Auto. Control* **1987**, *32*, 728–732. [[CrossRef](#)]
- Chu, E.K. Optimisation and pole assignment in control system design. *Int. J. Appl. Math. Comput. Sci.* **2001**, *11*, 1035–1053.
- Datta, S.; Chakraborty, D.; Chaudhuri, B. Partial pole placement with controller optimization. *IEEE Tans. Auto. Control* **2012**, *57*, 1051–1056. [[CrossRef](#)]

31. Datta, S.; Chaudhuri, B.; Chakraborty, D. Partial pole placement with minimum norm controller. In Proceedings of the 49th IEEE Conference on Decision and Control, Atlanta, GA, USA, 15–17 December 2010; pp. 5001–5006.
32. Saad, Y. Projection and deflation methods for partial pole assignment in linear state feedback. *IEEE Trans. Automat. Control* **1988**, *33*, 290–297. [[CrossRef](#)]
33. Horn, R.A.; Johnson, C.R. *Matrix Analysis*; Cambridge University Press: Cambridge, UK, 1985.
34. Golub, G.H.; Van Loan, C.F. *Matrix Computations*; Johns Hopkins University Press: Baltimore, MD, USA, 1996.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.