machines

MDPI

Article

Linear Algebra-Based Multivariable Controller Design for Gas Turbine Machines with State-Derivative Feedback

Belkacem Bekhiti [1], Kamel Hariche [2], Abderrezak Guessoum [3] and Abdel-Nasser Sharkawy [4,5,*]

[1] Institute of Aeronautics and Space Studies (IASS), University of Blida 1, BP 270, Blida 09000, Algeria; belkacem1988@hotmail.co.uk
[2] Institute of Electrical and Electronic Engineering (IGEE), University of Boumerdes, Boumerdes 35000, Algeria; khariche@yahoo.com
[3] Signal Processing and Image Laboratory (LATSI), Department of Electronics, Faculty of Technology, University of Blida, Blida 09000, Algeria; abderguessoum@gmail.com
[4] Mechanical Engineering Department, Faculty of Engineering, Qena University, Qena 83523, Egypt
[5] Mechanical Engineering Department, College of Engineering, Fahad Bin Sultan University, Tabuk 47721, Saudi Arabia
* Correspondence: abdelnassersharkawy@eng.svu.edu.eg or eng.abdelnassersharkawy@gmail.com; Tel.: +20-1001689823

Abstract

This paper presents a linear algebra-based control algorithm for multivariable gas turbine systems using matrix polynomial theory and the Kronecker product to assign block roots (i.e., block eigenvectors with prescribed latent structure). State and state-derivative feedback strategies are investigated and validated through simulations on an industrial gas turbine machine. The proposed method enables direct assignment of block roots governing closed-loop stability and transient response, while block eigenvectors shape the dynamic behavior of key turbine variables. Applicability of the approach requires block controllability and/or block observability, ensuring analytical transparency, design flexibility, and effectiveness for multivariable gas turbine control.

Keywords: state-derivative feedback; block-roots; block-eigenvectors; latent structure; matrix polynomial; Kronecker product

Check for updates

Academic Editor: Carmine Maria Pappalardo

Received: 31 December 2025
Revised: 24 January 2026
Accepted: 27 January 2026
Published: 2 February 2026

Copyright: © 2026 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.

1. Introduction

The systematic design of multivariable controllers has its theoretical roots in linear algebraic formulations of pole placement. Early seminal work established the conditions under which arbitrary pole assignment is achievable in multi-input controllable linear systems through state feedback, laying the groundwork for modern MIMO control synthesis [1]. Subsequent developments extended this notion by demonstrating that state feedback offers structural freedoms beyond mere eigenvalue placement, enabling deeper manipulation of closed-loop dynamics and performance characteristics [2]. These insights motivated a broad class of eigenstructure-based design techniques, where both eigenvalues and eigenvectors are explicitly shaped to meet stability and dynamic response requirements. The algebraic characterization of eigenstructure (ES) assignment rapidly evolved toward parametric formulations, enabling systematic controller synthesis with explicit degrees of freedom. Early parametric solutions introduced tractable formulations for shaping closed-loop eigenstructures in multivariable systems [3], while complete ES-assignment frameworks formalized the simultaneous control of modal directions and dynamic modes [4]. These approaches were further generalized to aerospace and large-scale

systems, where structural constraints necessitate refined algebraic tools [5]. Numerical algorithms addressing the computational challenges of eigenvalue assignment were subsequently proposed, emphasizing robustness and numerical efficiency [6]. Matrix polynomial theory emerged as a powerful framework for multivariable control, particularly for systems with complex interconnections and high dimensionality. The use of matrix polynomials enables controller synthesis through algebraic equations that directly encode system dynamics and desired closed-loop behavior [7]. Block pole placement strategies extended these ideas by introducing structured eigenvalue clustering, offering improved scalability for large MIMO systems [8]. Recent advances further integrate operator matrix polynomials into flight dynamics and aeroelastic control problems, highlighting their versatility in coupled physical phenomena [9].

While state feedback provides maximal flexibility, practical implementations often rely on output feedback due to sensing limitations. Reduced sensor-actuator configurations and output-feedback-based eigenvalue assignment techniques address this challenge by balancing performance and implementation complexity [10]. Robust ES-assignment methods explicitly incorporate uncertainties, ensuring stability and performance preservation under modeling errors [11]. Algebraic solution frameworks based on generalized Sylvester-type equations further unified eigenstructure synthesis for a wide class of linear systems [12]. Comprehensive theoretical treatments subsequently established rigorous conditions and classifications for ES-assignment problems [13]. Robust control formulations based on ES-assignment introduced multi-objective optimization into controller synthesis, enabling trade-offs between robustness, performance, and actuator constraints [14]. Parallel developments in intelligent and adaptive control demonstrated how algebraic controller structures can be embedded within learning-based frameworks [15]. These concepts were consolidated into authoritative monographs that formalized ES-assignment as a core paradigm in control theory [16]. Hybrid strategies combining genetic algorithms with gradient-based optimization further enhanced robustness and convergence properties in complex MIMO systems [17]. Efficient computation of matrix polynomial inverses and solvents is central to linear algebra-based control synthesis. Recursive schemes inspired by classical Leverrier–Faddeev algorithms have been proposed to address large-scale and high-order systems, with demonstrated effectiveness in structural dynamics applications [18]. Earlier foundational work on solvents and spectral factorization established the mathematical basis for such algorithms [19]. Non-iterative pole placement techniques subsequently refined these ideas, offering closed-form solutions with reduced computational burden [20].

Parametric static output feedback approaches provided additional flexibility for eigenvalue assignment without full state measurement [21]. These methods were extended to digital and adaptive control settings for large-scale MIMO systems, leveraging matrix Diophantine equations for systematic controller design [22]. Theoretical investigations into matrix polynomial solvents further deepened understanding of eigenvalue distributions in MIMO systems [23]. Recent studies revisited static output feedback gain design from an eigenspace characterization perspective, particularly for turbo-generator applications [24]. Block eigenvalue concepts introduced new solution paradigms for differential-matrix equations and descriptor systems [25]. State derivative feedback was shown to significantly enlarge the achievable eigenstructure set for descriptor systems, motivating renewed interest in derivative feedback strategies [26]. Matrix fraction descriptions further unified the treatment of large-scale descriptor systems within a polynomial framework [27], providing a natural foundation for state-derivative-based controller design.

Accurate gas turbine models are essential for high-performance controller synthesis. Data-driven identification techniques based on subspace methods have been successfully

applied to capture dominant turbine dynamics from input–output measurements [28]. Advanced control-oriented modeling approaches have also enabled real-time emulation of gas turbine engines using electric machines, facilitating controller validation under realistic operating conditions [29]. Robust and nonlinear control strategies have been extensively investigated for gas turbine applications. Mixed-sensitivity H∞ control provided early robust design solutions for turbine systems [30], while sliding mode control emerged as a powerful alternative due to its robustness against uncertainties and disturbances [31]. Model-predictive control frameworks incorporating economic objectives further enhanced efficiency and operational flexibility [32]. Experimental validation of sliding mode load controllers confirmed their effectiveness in real-time turbine operation [33]. Earlier studies also demonstrated the stabilizing capability of sliding mode control for linear multivariable gas turbine models [34], while optimal LQG/LTR designs offered performance improvements in industrial settings [35].

Despite the extensive literature on ES-assignment, matrix polynomial methods, and gas turbine control, three critical gaps remain. First, most algebraic multivariable control frameworks focus on state or output feedback, with limited exploitation of state-derivative feedback despite its proven capability to expand ES-assignability. Second, existing gas turbine control strategies are predominantly numerical or optimization-driven, offering limited analytical transparency and structural insight. Third, there is a lack of unified linear algebra-based controller design frameworks that directly integrate multivariable pole assignment, matrix polynomial formulations, and gas turbine dynamics within a single coherent methodology. These gaps motivate the present work. The main contributions of this paper are summarized as follows:

- Development of a linear algebra-based multivariable controller design framework incorporating vector spaces and state-derivative feedback.
- Formulation of the controller synthesis problem using matrix polynomials.
- Enhanced block-structure assignability compared to classical designs.
- Application to a MIMO gas turbine model with transparent and flexible design.
- Comparative study on a gas turbine showing the impact of the proposed method.

The remainder of this paper is organized as follows. Section 2 presents the mathematical preliminaries and problem formulation. Section 3 develops the proposed linear algebra-based controller synthesis with state-derivative feedback. Section 4 applies the methodology to a multivariable gas turbine model and discusses the results. The last section provides comparative analysis and performance evaluation. Finally, Section 5 concludes the paper and outlines directions for future research.

2. Mathematical Preliminaries

This paper is concerned with the control of the following high order dynamical linear system which is described by the following state and output equations [7]:

$$\mathbf{u}(t) = \sum_{k=0}^{\ell} \left\{ A_{\ell-k} \frac{d^k \mathbf{v}(t)}{dt^k} \right\}; \quad \text{and} \quad \mathbf{y}(t) = \sum_{k=0}^{\ell-1} \left\{ C_{\ell-k} \frac{d^k \mathbf{v}(t)}{dt^k} \right\} \tag{1}$$

Or more explicitly we can write

$$\left\{ \mathbf{u}(t) = A_0 \frac{d^{\ell} \mathbf{v}(t)}{dt^{\ell}} + \cdots + A_{\ell-1} \frac{d\mathbf{v}(t)}{dt} + A_{\ell} \mathbf{v}(t) \right\} \text{and} \left\{ \mathbf{y}(t) = C_1 \frac{d^{(\ell-1)} \mathbf{v}(t)}{dt^{(\ell-1)}} + \cdots + C_{\ell-1} \frac{d\mathbf{v}(t)}{dt} + C_{\ell} \mathbf{v}(t) \right\} \tag{2}$$

where $\mathbf{v}(t) \in \mathbb{R}^m$ $\mathbf{u}(t) \in \mathbb{R}^m$ $\mathbf{y}(t) \in \mathbb{R}^p$ are the state vector, the control vector and the output vector, respectively. $A_i, \in \mathbb{R}^{m \times m}$ with $i = 0, 1, 2, \ldots, \ell$ and $C_i \in \mathbb{R}^{p \times m}$ $i = 1, 2, \ldots, \ell$ are the system coefficient matrices where $\det(A_0) \neq 0$.

Now, by applying the Laplace transformation to Equations (1) and (2) we achieve the following dynamical equations:

$$\left\{\begin{array}{c}\mathbf{u}(\lambda)=\left[\boldsymbol{A}_0\lambda^{\ell}+\boldsymbol{A}_1\lambda^{\ell-1}+\cdots+\boldsymbol{A}_{\ell}\right]\mathbf{v}(\lambda)\\ \text{and}\\ \mathbf{y}(\lambda)=\left[\boldsymbol{C}_1\lambda^{\ell-1}+\boldsymbol{C}_1\lambda^{\ell-1}+\cdots+\boldsymbol{C}_{\ell}\right]\mathbf{v}(\lambda)\end{array}\right\}\quad\Leftrightarrow\quad\mathbf{y}(t)=\left[\boldsymbol{N}(\lambda)(\boldsymbol{D}(\lambda))^{-1}\right]\mathbf{u}(\lambda)\tag{3}$$

where $\boldsymbol{D}(\lambda)=\sum_{i=0}^{\ell}\boldsymbol{A}_i\lambda^{\ell-i}$ and $\boldsymbol{N}(\lambda)=\sum_{i=1}^{\ell}\boldsymbol{C}_i\lambda^{\ell-i}$. The matrices $\boldsymbol{D}(\lambda)$ and $\boldsymbol{N}(\lambda)$ are called matrix polynomials in complex variables λ with matrix coefficients which are defined on the field of real matrices $\mathbb{R}^{m\times m}$.

2.1. Latent Structure and Block Roots (Solvents)

Definition 1 ([7]). *Let $\{\boldsymbol{A}_k\}_{k=0}^{\ell}$ be a set of $m\times m$ complex matrices, then the matrix valued function $\boldsymbol{A}(\lambda)\in\mathbb{R}^{m\times m}[\lambda]$ of the complex variable λ is called a matrix polynomial of degree ℓ and order m where*

$$\boldsymbol{A}(\lambda)=\boldsymbol{A}_0\lambda^{\ell}+\boldsymbol{A}_1\lambda^{\ell-1}+\cdots+\boldsymbol{A}_{\ell-1}\lambda+\boldsymbol{A}_{\ell}=\sum_{i=0}^{\ell}\boldsymbol{A}_i\lambda^{\ell-i}\tag{4}$$

Definition 2 ([8]). *The matrix polynomial $\boldsymbol{A}(\lambda)\in\mathbb{R}^{m\times m}[\lambda]$ is called the following:*

- *Monic if $\boldsymbol{A}_0$ is the identity matrix.*
- *Comonic if $\boldsymbol{A}_{\ell}$ is the identity matrix.*
- *Regular if $\det(\boldsymbol{A}(\lambda))\neq 0$.*
- *Nonsingular if $\det(\boldsymbol{A}(\lambda))$ is not identically zero.*
- *Unimodular if $\det(\boldsymbol{A}(\lambda))$ is a nonzero constant.*

Definition 3 ([9,24]). *The complex number $\lambda_i\in\mathbb{C}$ is called a latent root of $\boldsymbol{A}(\lambda)$ if it is a solution of the scalar polynomial equation $\det(\boldsymbol{A}(\lambda_i))=\mathrm{O}_m$. The nontrivial vector $\boldsymbol{p}$ solution of $\boldsymbol{A}(\lambda_i)\boldsymbol{p}=\mathrm{O}_m$ is called a primary right latent vector associated with λ_i. Similarly, the nontrivial vector $\boldsymbol{q}$, solution of $\boldsymbol{q}^T\boldsymbol{A}(\lambda_i)=\mathrm{O}_m$ is called a primary left latent vector associated with λ_i. The term O_m is the $m\times 1$ or $1\times m$ zero matrix.*

Remark 1. *If the leading coefficient $\boldsymbol{A}_0$ is singular, then $\boldsymbol{A}(\lambda)$ has latent roots at infinity.*

From the definition, we can see that the latent problem of a matrix polynomial is a generalization of the concept of the eigen-problem for square matrices. Indeed, we can consider the classical eigenvalues/vector problem as finding the latent root/vector of a linear matrix polynomial $(\lambda\boldsymbol{I}-\boldsymbol{A})$. We can also define the spectrum of a matrix polynomial $\boldsymbol{A}(\lambda)$ as being the set of all its latent roots (notation $\sigma(\lambda)$). It is essentially the same definition as the one of the spectrum of a square matrix [22–25].

Definition 4 ([19]). *A right block root, also called the solvent of λ-matrix $\boldsymbol{A}(\lambda)$ is an $m\times m$ real matrix $\boldsymbol{R}$ in such a manner:*

$$\boldsymbol{R}^{\ell}+\boldsymbol{A}_1\boldsymbol{R}^{\ell-1}+\ \ldots\ \boldsymbol{A}_{\ell-1}\boldsymbol{R}+\boldsymbol{A}_{\ell}=\mathrm{O}_m\ \ \textit{or in other word}\ \ \boldsymbol{A}_R(\boldsymbol{R})=\sum_{i=0}^{\ell}\boldsymbol{A}_i\boldsymbol{R}^{\ell-i}=\mathrm{O}_m$$

while a left solvent is an $m\times m$ real matrix $\boldsymbol{L}$ in such a manner:

$$\boldsymbol{L}^{\ell}+\boldsymbol{L}^{\ell-1}\boldsymbol{A}_1+\ \ldots\ \boldsymbol{L}\boldsymbol{A}_{\ell-1}+\boldsymbol{A}_{\ell}=\mathrm{O}_m\ \ \textit{or in other word}\ \ \boldsymbol{A}_L(\boldsymbol{L})=\sum_{i=0}^{\ell}\boldsymbol{L}^{\ell-i}\boldsymbol{A}_i=\mathrm{O}_m$$

The following are important facts on solvents [8]:

- Solvents of a matrix polynomial do not always exist.

- Generalized right (left) eigenvectors of a right (left) solvent are the generalized latent vectors of the corresponding matrix polynomial.

2.2. Block Companion Forms

2.2.1. From Matrix Fraction Description to Block-State Space Realization

A linear time-invariant multivariable (MIMO) system expressed using a right matrix fraction description (RMFD) can be converted into a block state-space representation by applying a recursive procedure to the associated coupled differential equations [7].

Let now a dynamic system with the following RMFD:

$$\mathbf{y}(\lambda) = \boldsymbol{H}(\lambda)\mathbf{u}(\lambda) = \left[\boldsymbol{N}_R(\lambda)(\boldsymbol{D}_R(\lambda))^{-1}\right]\mathbf{u}(\lambda) \tag{5}$$

$\boldsymbol{D}_R(\lambda)$ and $\boldsymbol{N}_R(\lambda)$ are $m \times m$ and $p \times m$ matrix polynomials also called λ-matrices, the complex variable λ is often used instead of s, defined by the following:

$$\boldsymbol{D}_R(\lambda) = \boldsymbol{I}_m\lambda^{\ell} + \boldsymbol{A}_1\lambda^{\ell-1} + \ldots + \boldsymbol{A}_{\ell} \text{ and } \boldsymbol{N}_R(\lambda) = \boldsymbol{C}_1\lambda^{\ell-1} + \boldsymbol{C}_2\lambda^{\ell-2} + \ \ldots \ + \boldsymbol{C}_{\ell}$$

where $\boldsymbol{A}_i \in \mathbb{R}^{m\times m}$ constant matrices, $\boldsymbol{C}_i \in \mathbb{R}^{p\times m}$ are constant matrices, and $\boldsymbol{I}_m$ stands for the $m \times m$ identity matrix.

An alternative factorization of the matrix transfer function $\boldsymbol{H}(\lambda)$ is called the left matrix fraction description (LMFD) and is defined by [8] the following:

$$\boldsymbol{H}(\lambda) = \left[\boldsymbol{N}_R(\lambda)(\boldsymbol{D}_R(\lambda))^{-1}\right] = \left[(\boldsymbol{D}_L(\lambda))^{-1}\boldsymbol{N}_L(\lambda)\right] \tag{6}$$

Equation (6) represents a right and left matrix fraction description (MFD) of the matrix transfer function $\boldsymbol{H}(\lambda) \in \mathbb{R}^{p\times m}(\lambda)$. The terms $\boldsymbol{N}_R(\lambda) \in \mathbb{R}^{p\times m}[\lambda]$, $\boldsymbol{D}_R(\lambda) \in \mathbb{R}^{m\times m}[\lambda]$ are the right numerator and denominator matrix polynomials, while the terms $\boldsymbol{N}_L(\lambda) \in \mathbb{R}^{p\times m}[\lambda]$, and $\boldsymbol{D}_L(\lambda) \in \mathbb{R}^{p\times p}[\lambda]$ are the left numerator and denominator polynomials. For the equality to hold, the right and left MFDs must be equivalent, which requires $\boldsymbol{D}_L(\lambda)\boldsymbol{N}_R(\lambda) = \boldsymbol{N}_L(\lambda)\boldsymbol{D}_R(\lambda)$ This is a polynomial matrix Diophantine equation.

- The poles of $\boldsymbol{H}(\lambda)$ are the roots of $\boldsymbol{D}_R(\lambda)$ (right MFD) or $\boldsymbol{D}_L(\lambda)$ (left MFD).
- The zeros are related to the determinants of $[\boldsymbol{N}_R(\lambda), \boldsymbol{D}_R(\lambda)]$ or $[\boldsymbol{N}_L(\lambda), \boldsymbol{D}_L(\lambda)]$.

The general state space presentation of the given matrix fraction description, i.e., right or left, is obtained via the use of very well-known realization theory as follows [12,34]:

$$\left\{\dot{\mathbf{x}}(t) = \boldsymbol{A}\mathbf{x}(t) + \boldsymbol{B}\mathbf{u}(t) \qquad \text{and} \qquad \mathbf{y}(t) = \boldsymbol{C}\mathbf{x}(t) + \boldsymbol{D}\mathbf{u}(t)\right\} \tag{7}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ denotes the system matrix, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ represents the input matrix, $\boldsymbol{C} \in \mathbb{R}^{p\times n}$ is the output matrix, and $\boldsymbol{D} \in \mathbb{R}^{m\times p}$ is the direct feedthrough matrix. The state, input, and output vectors are given by the following: $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$, and $\mathbf{y}(t) \in \mathbb{R}^p$, respectively.

2.2.2. Transformation Between State-Space Forms

Consider a system represented by the standard state-space Equation (7), assumed to be controllable and/or observable. It can be converted into different companion forms—such as the controller or observer forms—through appropriate similarity transformations. [8]. In our study, three specific transformations are required: the block controller form, the block observer form, and the block diagonal form. To transform a system represented by $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$, the following conditions must be satisfied:

- The number $n/m = \ell$ or $n/p = \ell$ is an integer.
- The matrix $\boldsymbol{\Omega} = \text{row}\left\{\boldsymbol{A}^i\boldsymbol{B}\right\}_{i=0}^{\ell-1} = \left[\boldsymbol{B}, \ldots, \boldsymbol{A}^{\ell-1}\boldsymbol{B}\right]$ or $\boldsymbol{\Omega} = \text{col}\left\{\boldsymbol{C}\boldsymbol{A}^i\right\}_{i=0}^{\ell-1}$ is of full rank.

Conversion to Block Controller Form

For a system defined by the general state-space model $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ that is block controllable, it can be converted into a block controller form [7,9]. If $n/m = \ell$ is an integer and the block controllability matrix $\boldsymbol{\Omega}_c = \left[\boldsymbol{B}, \boldsymbol{AB} \ldots, \boldsymbol{A}^{\ell-1}\boldsymbol{B}\right]$ is full rank, the state equation can be transformed into block controller form via the similarity transformation $\mathbf{x}_c = \boldsymbol{T}_c\mathbf{x}$, where

$$\left\{\dot{\mathbf{x}}_c(t) = \boldsymbol{A}_c\mathbf{x}_c(t) + \boldsymbol{B}_c\mathbf{u}(t) \qquad \text{and} \qquad \mathbf{y}(t) = \boldsymbol{C}_c\mathbf{x}_c(t) + \boldsymbol{D}_c\mathbf{u}(t)\right\} \tag{8}$$

$$\boldsymbol{T}_c = \begin{bmatrix} \boldsymbol{T}_{c\ell} \\ \boldsymbol{T}_{c\ell}\boldsymbol{A} \\ \vdots \\ \boldsymbol{T}_{c\ell}\boldsymbol{A}^{\ell-1} \end{bmatrix} \quad \text{and} \quad \boldsymbol{T}_{c\ell} = [\mathbf{O}_m, \cdots, \mathbf{O}_m, \boldsymbol{I}_m]\left[\boldsymbol{B}, \boldsymbol{AB} \cdots, \boldsymbol{A}^{\ell-1}\boldsymbol{B}\right]^{-1} \tag{9}$$

$$\boldsymbol{A}_c = \begin{bmatrix} \mathbf{O}_m & \boldsymbol{I}_m & & \mathbf{O}_m \\ \mathbf{O}_m & \mathbf{O}_m & & \mathbf{O}_m \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O}_m & \mathbf{O}_m & & \boldsymbol{I}_m \\ -\boldsymbol{A}_\ell & -\boldsymbol{A}_{\ell-1} & & -\boldsymbol{A}_1 \end{bmatrix}, \quad \boldsymbol{B}_c = \begin{bmatrix} \mathbf{O}_m \\ \mathbf{O}_m \\ \vdots \\ \mathbf{O}_m \\ \boldsymbol{I}_m \end{bmatrix}, \quad \boldsymbol{C}_c = [\boldsymbol{C}_\ell, \boldsymbol{C}_{\ell-1}, \cdots, \boldsymbol{C}_1] \tag{10}$$

with $\boldsymbol{A}_c = \boldsymbol{T}_c\boldsymbol{A}\boldsymbol{T}_c^{-1}$, $\boldsymbol{B}_c = \boldsymbol{T}_c\boldsymbol{B}$, $\boldsymbol{C}_c = \boldsymbol{C}\boldsymbol{T}_c^{-1}$.

Conversion to Block Observable Form

If the considered system is block observable, then it can be transformed into a block observer form [7,9]. So if $n/p = \ell$ is an integer number, and if the block observability matrix $\boldsymbol{\Omega}_o = \left[\boldsymbol{C}^\top, \boldsymbol{A}^\top\boldsymbol{C}^\top, \cdots, \left\{\boldsymbol{A}^{\ell-1}\right\}^\top\boldsymbol{C}^\top\right]^\top$ is full rank, then we can convert the state equation into block observer form using the following similarity transformation $\mathbf{x}_o = \boldsymbol{T}_o\mathbf{x}$

$$\left\{\dot{\mathbf{x}}_o(t) = \boldsymbol{A}_o\mathbf{x}_o(t) + \boldsymbol{B}_o\mathbf{u}(t) \qquad \text{and} \qquad \mathbf{y}(t) = \boldsymbol{C}_o\mathbf{x}_o(t) + \boldsymbol{D}_o\mathbf{u}(t)\right\} \tag{11}$$

$$\boldsymbol{T}_o = \left[\boldsymbol{T}_{o\ell}, \boldsymbol{A}\boldsymbol{T}_{o\ell}, \cdots, \boldsymbol{A}^{\ell-1}\boldsymbol{T}_{o\ell}\right] \quad \text{and} \quad \boldsymbol{T}_{o\ell} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}^{\ell-1} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{O}_p \\ \vdots \\ \mathbf{O}_p \\ \boldsymbol{I}_p \end{bmatrix} \tag{12}$$

$$\boldsymbol{A}_o = \begin{bmatrix} \mathbf{O}_p & & \mathbf{O}_p & -\boldsymbol{A}_\ell \\ \boldsymbol{I}_p & & \mathbf{O}_p & \vdots \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{O}_p & & \mathbf{O}_p & -\boldsymbol{A}_2 \\ \mathbf{O}_p & & \boldsymbol{I}_p & -\boldsymbol{A}_1 \end{bmatrix}, \quad \boldsymbol{B}_o = \begin{bmatrix} \boldsymbol{B}_\ell \\ \vdots \\ \vdots \\ \boldsymbol{B}_2 \\ \boldsymbol{B}_1 \end{bmatrix}, \quad \boldsymbol{C}_o = \left[\mathbf{O}_p, \ldots, \mathbf{O}_p, \boldsymbol{I}_p\right] \tag{13}$$

with $\boldsymbol{A}_o = \boldsymbol{T}_o^{-1}\boldsymbol{A}\boldsymbol{T}_o$, $\boldsymbol{B}_o = \boldsymbol{T}_o^{-1}\boldsymbol{B}$, $\boldsymbol{C}_o = \boldsymbol{C}\boldsymbol{T}_o$.

Conversion to Block Diagonal Form

If the matrix polynomial $\boldsymbol{A}(\lambda)$ possesses a complete set of solvents, then the companion matrices $\boldsymbol{A}_c$ or $\boldsymbol{A}_o$ can be block-diagonalized by right/left block Vandermonde matrices [25,27], defined as follows:

$$V_R(R_1,\cdots,R_\ell)=\begin{bmatrix} I_m & I_m & \cdots & I_m \\ R_1 & R_2 & \cdots & R_\ell \\ \vdots & \vdots & \ddots & \vdots \\ R_1^{\ell-1} & R_2^{\ell-1} & \cdots & R_\ell^{\ell-1} \end{bmatrix},\ V_L(L_1,\cdots,L_\ell)=\begin{bmatrix} I_m & L_1 & \cdots & L_1^{\ell-1} \\ I_m & L_2 & \cdots & L_2^{\ell-1} \\ \vdots & \vdots & \ddots & \vdots \\ I_m & L_\ell & \cdots & L_\ell^{\ell-1} \end{bmatrix} \tag{14}$$

$$[V_R(R_1,R_2,\cdots,R_\ell)]^{-1}A_c[V_R(R_1,R_2,\cdots,R_\ell)]=\text{blockdiag}(R_1,R_2,\cdots,R_\ell) \tag{15}$$

$$[V_L(L_1,L_2,\cdots,L_\ell)]\,A_o\,[V_L(L_1,L_2,\cdots,L_\ell)]^{-1}=\text{blockdiag}(L_1,L_2,\cdots,L_\ell) \tag{16}$$

These similarity transformations do a block decoupling of the spectrum of $A(\lambda)$ which is very useful in the analysis and design of large order control systems [15].

3. The Proposed Linear Algebra-Based Block Structure Assignment

3.1. Block Structure Assignment via State Feedback

This section introduces an alternative method for designing a linear state-feedback control law based on a desired left or right latent structure, using algebraic techniques and subspace theory. This approach serves as the MIMO counterpart to eigenstructure assignment, as discussed in prior studies [1–6].

Theorem 1 ([7]). *Let $A_c \in \mathbb{R}^{n\times n}$ denote a block companion controller form associated with the matrix polynomial $A(\lambda)$ and let $R_i \in \mathbb{R}^{m\times m}$ be a corresponding right block root (solvent). Then the following relation holds: $A_cX_{ci}=X_{ci}R_i$ or equivalently*

$$\left[I_m\otimes A_c-R_i^{\top}\otimes I_n\right]\text{vec}(X_{ci})=0 \ \ with \ \ X_{ci}=\left[I_m\ \vdots\ R_i^{\top}\ \vdots\ \cdots\ \vdots\ \left(R_i^{\ell-1}\right)^{\top}\right]^{\top} \tag{17}$$

*Here $X_{ci} \in \mathbb{R}^{n\times m}$ represents the block eigenvector matrix (**block vector**), $\otimes$ denotes the Kronecker product, and $\text{vec}(\cdot)$ is the column-stacking operator.*

Proof. This theorem can be readily verified by applying Definition 4 and performing direct algebraic manipulations. □

Theorem 2 ([8,9]). *Every eigenvalue of a right or left solvent of $A(\lambda)$ is also an eigenvalue of A_c.*

Proof. The block companion matrix A_c can be diagonalized using a block Vandermonde matrix through the similarity transformation: $V_R^{-1}A_cV_R=\Lambda=\text{blckdiag}(R_1,R_2,\cdots,R_\ell)$. Since $\det(\lambda I_n-A_c)=\det(\lambda I_n-\Lambda)=\det(\text{blckdiag}((\lambda I_m-R_1),\cdots,(\lambda I_m-R_\ell)))$, it follows that $\det(\lambda I_n-A_c)=\det(\lambda I_m-R_1)\cdots\det(\lambda I_m-R_\ell)$. Hence, any eigenvalue of a right or left solvent R_i or L_i of $A(\lambda)$ is also an eigenvalue of A_c. □

Remark 2. *Assigning a matrix R as a right solvent or a matrix L as a left solvent of $A(\lambda)$, is equivalent to specifying the eigenvalues and eigenvectors of R, or L, as the latent roots and right or left latent vectors of $A(\lambda)$. In other words, assigning eigenvalues to a solvent of $A(\lambda)$ directly corresponds to assigning these eigenvalues to the block companion matrix A_c [23,25].*

Theorem 3 ([22,27]). *Suppose $A(\lambda)$ has n linearly independent right latent vectors $p_1,\ldots,p_n$ (or left latent vectors $q_1,\ldots,q_n$) associated with latent roots $\lambda_1,\ldots,\lambda_n$, then $P\Lambda P^{-1}$, $\left(Q^{-1}\Lambda Q\right)$ forms a right (or left) solvent of $A(\lambda)$, where $P=[p_1,\cdots,p_n]$, $\left(Q=[q_1,\cdots,q_n]^{\top}\right)$ and $\Lambda=\text{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n)$.*

Consider the high-order dynamical system described by the state Equation (7). Applying the control law $\mathbf{u}(t)=K_1\mathbf{x}(t)+K_2\mathbf{r}(t)$, the closed-loop equation becomes the following:

$$\dot{\mathbf{x}}(t) = (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}_1)\mathbf{x}(t) + \boldsymbol{K}_2\mathbf{r}(t) \tag{18}$$

Bringing the system into a block controllable representation using: $\mathbf{x}_c = \boldsymbol{T}_c\mathbf{x}$.

$$\dot{\mathbf{x}}_c(t) = (\boldsymbol{A}_c + \boldsymbol{B}_c\boldsymbol{K}_{c1})\mathbf{x}_c(t) + \boldsymbol{K}_{c2}\mathbf{r}(t) \tag{19}$$

To perform block pole assignment, we consider relocating the solvents of the plant. Let $(\boldsymbol{R}_i, \boldsymbol{X}_{ci})$, denote the block eigen pairs, representing the complete set of desired solvents $\boldsymbol{R}_i$ with their corresponding block eigenvectors $\boldsymbol{X}_i$ where $i = 1, 2, \cdots, \ell$. By the use of Theorem 1, we have the following:

$$(\boldsymbol{A}_c + \boldsymbol{B}_c\boldsymbol{K}_{c1})\boldsymbol{X}_{ci} = \boldsymbol{X}_{ci}\boldsymbol{R}_i \qquad \text{with } i = 1, 2 \ldots, \ell \text{ or equivalently} \tag{20}$$

$$\boldsymbol{A}_c\boldsymbol{X}_{ci} - \boldsymbol{X}_{ci}\boldsymbol{R}_i + \boldsymbol{B}_c\boldsymbol{K}_{c1}\boldsymbol{X}_{ci} = \mathbf{O}_{n\times m} \quad \text{with } i = 1, 2 \ldots, \ell \tag{21}$$

Let the change in variable, $\boldsymbol{\mathcal{W}}_{ci} = \boldsymbol{K}_{1c}\boldsymbol{X}_{ci} \Rightarrow \boldsymbol{A}_c\boldsymbol{X}_{ci} - \boldsymbol{X}_{ci}\boldsymbol{R}_i + \boldsymbol{B}_c\boldsymbol{\mathcal{W}}_{ci} = \mathbf{O}_{n\times m}$, then

$$\left[\boldsymbol{I}_m \otimes \boldsymbol{A}_c - \boldsymbol{R}_i^\top \otimes \boldsymbol{I}_n\right]\mathrm{vec}(\boldsymbol{X}_{ci}) + (\boldsymbol{I}_m \otimes \boldsymbol{B}_c)\mathrm{vec}(\boldsymbol{\mathcal{W}}_{ci}) = \mathbf{O} \tag{22}$$

$$(22) \Leftrightarrow \left[\boldsymbol{I}_m \otimes \boldsymbol{A}_c - \boldsymbol{R}_i^\top \otimes \boldsymbol{I}_n \;\vdots\; (\boldsymbol{I}_m \otimes \boldsymbol{B}_c)\right]\begin{bmatrix}\mathrm{vec}(\boldsymbol{X}_{ci})\\ \mathrm{vec}(\boldsymbol{\mathcal{W}}_{ci})\end{bmatrix} = 0 \tag{23}$$

Let us define the following matrices $\boldsymbol{S}(\boldsymbol{R}_i) = \left[\boldsymbol{I}_m \otimes \boldsymbol{A}_c - \boldsymbol{R}_i^\top \otimes \boldsymbol{I}_n \;\vdots\; \boldsymbol{I}_m \otimes \boldsymbol{B}_c\right]$ and $\boldsymbol{\mathcal{X}}(\boldsymbol{X}_{ci}, \boldsymbol{K}_{c1}) = \left[\mathrm{vec}^\top(\boldsymbol{X}_{ci})\ \mathrm{vec}^\top(\boldsymbol{\mathcal{W}}_{ci})\right]^\top$. From standard linear algebra, it is straightforward to see that if $\boldsymbol{S}(\boldsymbol{R}_i)$ is singular, then Equation (23) admits a finite solution such that $\boldsymbol{\mathcal{X}}(\boldsymbol{X}_{ci}, \boldsymbol{K}_{c1}) \in \boldsymbol{\mathcal{N}}(\boldsymbol{S}(\boldsymbol{R}_i))$ where $\boldsymbol{\mathcal{N}}$ denotes the null space of $\boldsymbol{S}(\boldsymbol{R}_i)$. Thus, $\boldsymbol{\mathcal{X}}(\boldsymbol{X}_{ci}, \boldsymbol{K}_{c1})$ from the kernel space of $\boldsymbol{S}(\boldsymbol{R}_i)$. Consequently, the block feedback matrices satisfy

$$[\boldsymbol{\mathcal{W}}_c(\boldsymbol{R}_1),\ \boldsymbol{\mathcal{W}}_c(\boldsymbol{R}_2),\ \cdots,\ \boldsymbol{\mathcal{W}}_c(\boldsymbol{R}_\ell)] = \boldsymbol{K}_{1c}[\boldsymbol{X}_c(\boldsymbol{R}_1), \boldsymbol{X}_c(\boldsymbol{R}_2),\ \cdots, \boldsymbol{X}_c(\boldsymbol{R}_\ell)] \tag{24}$$

where $\boldsymbol{\mathcal{W}}_{ci} = \boldsymbol{\mathcal{W}}_c(\boldsymbol{R}_1)$ and $\boldsymbol{X}_{ci} = \boldsymbol{X}_c(\boldsymbol{R}_1)$. A real feedback gain $\boldsymbol{K}_{c1}$ exists if and only if the following conditions hold:

1. The assigned latent values are symmetric about the real axis.
2. The matrix $\boldsymbol{V} = [\boldsymbol{X}_c(\boldsymbol{R}_1), \boldsymbol{X}_c(\boldsymbol{R}_2),\ \cdots, \boldsymbol{X}_c(\boldsymbol{R}_\ell)]$ is nonsingular.
3. The complete set of desired solvents is constructed so that $\boldsymbol{V}_R$ has full rank.

The feedback gain is then given by

$$\boldsymbol{K}_{1c} = \boldsymbol{W}\boldsymbol{V}^{-1} \tag{25}$$

With $\boldsymbol{W} = [\boldsymbol{\mathcal{W}}_c(\boldsymbol{R}_1),\ \boldsymbol{\mathcal{W}}_c(\boldsymbol{R}_2),\ \cdots,\ \boldsymbol{\mathcal{W}}_c(\boldsymbol{R}_\ell)]$, $\boldsymbol{V} = [\boldsymbol{X}_c(\boldsymbol{R}_1), \boldsymbol{X}_c(\boldsymbol{R}_2),\ \cdots, \boldsymbol{X}_c(\boldsymbol{R}_\ell)]$

$$\boldsymbol{K}_1 = \boldsymbol{K}_{1c}\boldsymbol{T}_c \tag{26}$$

Some parameterization is necessary to obtain the feedback matrix that enforces the desired block structure. Consider the system $(\boldsymbol{A}, \boldsymbol{B})$ and define the following:

$$\boldsymbol{H}_1(\lambda) = \boldsymbol{N}(\lambda)\boldsymbol{D}^{-1}(\lambda) = (\lambda\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} \Leftrightarrow \lambda{\cdot}\boldsymbol{N}(\lambda) = \boldsymbol{A}{\cdot}\boldsymbol{N}(\lambda) + \boldsymbol{B}{\cdot}\boldsymbol{D}(\lambda) \tag{27}$$

where the operation $(\cdot)$ stands for the dot-product and

$$\left\{\boldsymbol{N}(\lambda) = \sum_{i=0}^{\ell}\boldsymbol{N}_i\lambda^{\ell-i}\ \ \boldsymbol{N}_i \in \mathbb{R}^{n\times m},\ \ \boldsymbol{D}(\lambda) = \sum_{i=0}^{\ell}\boldsymbol{D}_i\lambda^{\ell-i}\ \ \boldsymbol{D}_i \in \mathbb{R}^{m\times m} \ \text{ with } \boldsymbol{D}_0 = \boldsymbol{I}_{m\times m},\ \boldsymbol{N}_0 = \mathbf{O}_{n\times m}\right\}$$

The complete set of desired solvents satisfies Equation (27), giving the following:

$$\boldsymbol{N}(\boldsymbol{R}_i)\boldsymbol{R}_i = \boldsymbol{A}{\cdot}\boldsymbol{N}(\boldsymbol{R}_i) + \boldsymbol{B}{\cdot}\boldsymbol{D}(\boldsymbol{R}_i) \qquad i = 1, \cdots, \ell \tag{28}$$

$$\text{With } \boldsymbol{N}(\boldsymbol{R}_i) = [\boldsymbol{N}_\ell, \ \cdots \ , \boldsymbol{N}_0] \begin{bmatrix} \boldsymbol{I}_m \\ \boldsymbol{R}_i \\ \vdots \\ \boldsymbol{R}_i^\ell \end{bmatrix} \quad \text{and} \quad \boldsymbol{D}(\boldsymbol{R}_i) = [\boldsymbol{D}_\ell, \ \cdots \ , \boldsymbol{D}_0] \begin{bmatrix} \boldsymbol{I}_m \\ \boldsymbol{R}_i \\ \vdots \\ \boldsymbol{R}_i^\ell \end{bmatrix}$$

The closed-loop system with ($\mathbf{u}(t) = \boldsymbol{K}_1\mathbf{x}(t) + \mathbf{r}(t)$ where $\boldsymbol{K}_2$ is designed out) can be expressed parametrically as follows:

$$\boldsymbol{H}_2(\lambda) = \boldsymbol{N}(\lambda)\boldsymbol{D}_{new}^{-1}(\lambda) = (\lambda\boldsymbol{I} - \boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})^{-1}\boldsymbol{B} \Leftrightarrow \lambda\cdot\boldsymbol{N}(\lambda) = \boldsymbol{A}\cdot\boldsymbol{N}(\lambda) + \boldsymbol{B}\cdot\boldsymbol{D}_{new}(\lambda) + \boldsymbol{B}\cdot\boldsymbol{K}\cdot\boldsymbol{N}(\lambda) \quad (29)$$

Evaluating Equation (29) at a right solvent $\boldsymbol{R}_i$ yields

$$\boldsymbol{N}(\boldsymbol{R}_i)\boldsymbol{R}_i = \boldsymbol{A}\cdot\boldsymbol{N}(\boldsymbol{R}_i) + \boldsymbol{B}\cdot\boldsymbol{D}_{new}(\boldsymbol{R}_i) + \boldsymbol{B}\cdot\boldsymbol{K}\cdot\boldsymbol{N}(\boldsymbol{R}_i) \qquad i = 1, \cdots, \ell \quad (30)$$

Remark 3. *The assigned block eigenvalues are solvents of* $\boldsymbol{D}_{new}(\lambda)$*, hence* $\boldsymbol{D}_{new}(\boldsymbol{R}_i) = \mathbf{O}_{m\times m}$.

In a more compact form, Equation (30) can be expressed as follows:

$$\begin{gathered} [\boldsymbol{N}_\ell, \ \cdots \ , \boldsymbol{N}_1]\boldsymbol{V}_R\boldsymbol{\Lambda}_R - (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})[\boldsymbol{N}_\ell, \ \cdots \ , \boldsymbol{N}_1]\boldsymbol{V}_R = \mathbf{O}_{n\times n} \\ \Leftrightarrow \boldsymbol{\Lambda}_R = ([\boldsymbol{N}_\ell, \ \cdots \ , \boldsymbol{N}_1]\boldsymbol{V}_R)^{-1}(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})([\boldsymbol{N}_\ell, \ \cdots \ , \boldsymbol{N}_1]\boldsymbol{V}_R) = (\boldsymbol{V}_R)^{-1}(\boldsymbol{A}_c)_{new}(\boldsymbol{V}_R) \end{gathered} \quad (31)$$

where

- $(\boldsymbol{A}_c)_{new} = [\boldsymbol{N}_\ell, \ \cdots \ , \boldsymbol{N}_1]^{-1}(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})[\boldsymbol{N}_\ell, \ \cdots \ , \boldsymbol{N}_1]$
- $\boldsymbol{V}_R = \text{Block} - \text{Vandermond Matrix}$
- $\boldsymbol{\Lambda}_R = \text{Block} - \text{diagonal Matrix}$
- $\boldsymbol{T}_c = [\boldsymbol{N}_\ell, \ \cdots \ , \boldsymbol{N}_1]^{-1}$

Remark 4. *If* $\boldsymbol{X}_{ci}$ *is a block eigenvector of the block companion matrix* $\boldsymbol{A}_c$*, then the corresponding eigenvector in the original coordinates is* $\boldsymbol{X}_i = {\boldsymbol{T}_c}^{-1}\boldsymbol{X}_{ci}$ *where* $\boldsymbol{X}_i, \boldsymbol{X}_{ci} \in \mathbb{R}^{n\times m}$ $i = 1, \cdots, \ell$.

$$\boldsymbol{X}_i = \boldsymbol{T}_c^{-1}\boldsymbol{X}_{ci} = \boldsymbol{T}_c^{-1}\left[\text{col}\left\{\boldsymbol{R}_i^k\right\}_{k=0}^{\ell-1}\right] = [\boldsymbol{N}_\ell, \ \cdots \ , \boldsymbol{N}_1]\left[\text{col}\left\{\boldsymbol{R}_i^k\right\}_{k=0}^{\ell-1}\right] = \sum_{k=0}^{\ell}\boldsymbol{N}_{l-k}\boldsymbol{R}_i^k = \boldsymbol{N}(\boldsymbol{R}_i)$$

$$\begin{gathered} \text{Equations (28) and (30)} \Rightarrow \boldsymbol{K}\cdot\boldsymbol{N}(\boldsymbol{R}_i) = \boldsymbol{D}(\boldsymbol{R}_i) \\ \Leftrightarrow \boldsymbol{K} = [\boldsymbol{D}(\boldsymbol{R}_1), \boldsymbol{D}(\boldsymbol{R}_2), \ \cdots \ , \boldsymbol{D}(\boldsymbol{R}_\ell)][\boldsymbol{N}(\boldsymbol{R}_1), \boldsymbol{N}(\boldsymbol{R}_2), \ \cdots \ , \boldsymbol{N}(\boldsymbol{R}_\ell)]^{-1} \end{gathered}$$

$$\Leftrightarrow \boldsymbol{K}_c = [\boldsymbol{D}(\boldsymbol{R}_1), \boldsymbol{D}(\boldsymbol{R}_2), \cdots, \boldsymbol{D}(\boldsymbol{R}_\ell)][\boldsymbol{N}(\boldsymbol{R}_1), \boldsymbol{N}(\boldsymbol{R}_2), \cdots, \boldsymbol{N}(\boldsymbol{R}_\ell)]^{-1}[\boldsymbol{N}_\ell, \cdots, \boldsymbol{N}_1] \quad (32)$$

Now we should develop an algorithmic procedure to reconstruct the matrix coefficient $\boldsymbol{N}_i$ and $\boldsymbol{D}_i$. Expanding Equation (27), gives the following:

$$\begin{cases} \boldsymbol{A}\boldsymbol{N}_\ell + B\boldsymbol{D}_\ell = \mathbf{O} \\ \boldsymbol{A}\boldsymbol{N}_i + B\boldsymbol{D}_i = \boldsymbol{N}_{i+1} \qquad i = 1, \cdots, l-1 \\ \boldsymbol{N}_0 = \mathbf{O}_{n\times m} \\ \boldsymbol{D}_0 = \boldsymbol{I}_{m\times m} \end{cases} \quad (33)$$

From Equation (33) solving from the above with back substitutions, we obtain:

$$\begin{bmatrix} \boldsymbol{D}_\ell \\ \vdots \\ \boldsymbol{D}_1 \end{bmatrix} = -\left[\boldsymbol{B}, \boldsymbol{A}\boldsymbol{B}, \cdots, \boldsymbol{A}^{\ell-1}\boldsymbol{B}\right]^{-1}\left(\boldsymbol{A}^\ell\boldsymbol{B}\right) = \boldsymbol{\Omega}_c^{-1}\left(\boldsymbol{A}^\ell\boldsymbol{B}\right) \quad (34)$$

$$[\boldsymbol{N}_\ell, \cdots, \boldsymbol{N}_1] = \left[\boldsymbol{B}, \cdots, \boldsymbol{A}^{\ell-1}\boldsymbol{B}\right] \begin{bmatrix} \boldsymbol{D}_{\ell-1} & \cdots & \boldsymbol{D}_1 & \boldsymbol{D}_0 \\ \vdots & & \iddots & \\ \boldsymbol{D}_1 & & & \\ \boldsymbol{D}_0 & & & \end{bmatrix} \tag{35}$$

The complete procedure for computing the block pole-placement gain using right block root assignment is summarized in Algorithm 1.

Algorithm 1 Block-Controllability-Based Pole Placement Using Right Block Roots

Step 1: Verify that the system is block controllable and determine the controllability index ℓ. Ensure that the matrix $\boldsymbol{\Omega}_c = \left[\boldsymbol{B}, \boldsymbol{AB}, \ldots, \boldsymbol{A}^{\ell-1}\boldsymbol{B}\right]$ is full rank and that $n = m\ell$. If n is not integer multiple of m apply the method proposed by Bekhiti et al. [7].

Step 2: Transform the system into block controller form with ℓ^2 sub-blocks using $\boldsymbol{T}_c$.

Step 3: Construct the desired right block roots based on the specified latent structure, ensuring that $\boldsymbol{V}_R(\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_\ell)$ is nonsingular.

Step 4: Compute the matrix coefficients $\boldsymbol{N}_i$ and $\boldsymbol{D}_i$.

Step 5: Compute $\boldsymbol{K}_{c1} = \boldsymbol{W}\boldsymbol{V}^{-1}\,[\boldsymbol{N}_\ell, \boldsymbol{N}_{\ell-1}, \cdots, \boldsymbol{N}_1]$ with $\boldsymbol{W} = [\boldsymbol{D}(\boldsymbol{R}_1), \boldsymbol{D}(\boldsymbol{R}_2), \cdots, \boldsymbol{D}(\boldsymbol{R}_\ell)]$, $\boldsymbol{V} = [\boldsymbol{N}(\boldsymbol{R}_1), \boldsymbol{N}(\boldsymbol{R}_2), \cdots, \boldsymbol{N}(\boldsymbol{R}_\ell)]$ and recover the original gain via, $\boldsymbol{K}_1 = \boldsymbol{K}_{c1}\boldsymbol{T}_c$.

Remark 5. *Also, we can assign left block poles using only the transposition and duality.*

Example 1. *Consider the 2-by-2-output system described by the following state space model:*

$$\boldsymbol{A} = \begin{bmatrix} 2.25 & -5.00 & -1.25 & -0.50 \\ 2.25 & -4.25 & -1.25 & -0.25 \\ 0.25 & -0.50 & -1.25 & -1.00 \\ 1.25 & -1.75 & -0.25 & -0.75 \end{bmatrix},\ \boldsymbol{B} = \begin{bmatrix} 4 & 6 \\ 2 & 4 \\ 2 & 2 \\ 0 & 2 \end{bmatrix},\ \boldsymbol{C} = \begin{bmatrix} 4.5 & -9 & 3.5 & 1.5 \\ 3.5 & -5 & -4.5 & 3.0 \end{bmatrix},\ \boldsymbol{D} = \mathrm{O}_{2\times 2}$$

Block controllability is ensured because the matrix $[\boldsymbol{B}, \boldsymbol{AB}]$ *is full rank with rank equal to 4, which gives a controllability index of* $\ell = n/m = 2$*. In the same manner, the system is block observable, as* $\operatorname{rank}\left(\left[\boldsymbol{C}^\top,\ \boldsymbol{C}^\top\boldsymbol{A}^\top\right]^\top\right) = 4$*, leading to an observability index of* $\ell = n/p = 2$*. Hence, the entire state vector is observable and can be accurately estimated.*

$$\boldsymbol{T}_{c\ell} = [\mathrm{O}_2, \boldsymbol{I}_2][\boldsymbol{B},\ \boldsymbol{AB}]^{-1},\quad \boldsymbol{T}_c = \begin{bmatrix} \boldsymbol{T}_{c\ell} \\ \boldsymbol{T}_{c\ell}A \end{bmatrix} = \begin{bmatrix} 0.0000 & -2.0000 & 2.0000 & 2.0000 \\ 0.6667 & 0.6667 & -2.0000 & -1.3333 \\ -1.5000 & 4.0000 & -0.5000 & -3.0000 \\ 0.8333 & -2.8333 & -1.1667 & 2.5000 \end{bmatrix}$$

The associated λ*-matrix can therefore be expressed as follows*:

$$\boldsymbol{A}(\lambda) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\lambda^2 + \begin{bmatrix} 4.5000 & 4.0000 \\ -2.3333 & -0.5000 \end{bmatrix}\lambda + \begin{bmatrix} 2.1667 & 2.5000 \\ -3.5000 & -3.0000 \end{bmatrix}$$

The goal is to place the system block poles according to a desired latent structure.

$$\left\{(-2, -2.5, -7, -14),\ \begin{bmatrix} 0.9705 & 0.2322 & 0.9705 & -0.1787 \\ 0.2409 & 0.9727 & -0.3102 & 0.9839 \end{bmatrix}\right\}$$

Accordingly, the desired block roots $\boldsymbol{R}_i$ *and their associated block eigenvectors* $\boldsymbol{X}_i$ *can be constructed according to* [7] *as follows*:

$$\left\{ \boldsymbol{R}_1 = \begin{bmatrix} -6.5590 & 1.3517 \\ -2.4279 & -14.441 \end{bmatrix},\ \boldsymbol{X}_1 = \begin{bmatrix} -40.3015 & -78.2392 \\ -25.1617 & -55.5606 \\ -16.1399 & -24.1786 \\ -8.5218 & -31.3819 \end{bmatrix} \right\};\ \left\{ \boldsymbol{R}_2 = \begin{bmatrix} -1.9685 & -0.1269 \\ 0.1319 & -2.5315 \end{bmatrix},\ \boldsymbol{X}_2 = \begin{bmatrix} -6.5825 & -12.6966 \\ -5.7427 & -10.8798 \\ -1.8398 & -3.3168 \\ -3.4028 & -7.5630 \end{bmatrix} \right\}$$

Applying the proposed method, the matrices $\boldsymbol{D}_i$ *and* $\boldsymbol{N}_i$ *can be derived from the system as follows*:

$$\left\{ \begin{array}{ccc} \boldsymbol{N}_2 = \begin{bmatrix} 0.5000 & 3.0000 \\ -2.3333 & -0.5000 \\ 1.8333 & 2.0000 \\ -3.6667 & -2.5000 \end{bmatrix}, & \boldsymbol{N}_1 = \begin{bmatrix} 4 & 6 \\ 2 & 4 \\ 2 & 2 \\ 0 & 2 \end{bmatrix}, & \boldsymbol{N}_0 = \mathrm{O}_{4\times 2} \\ \boldsymbol{D}_2 = \begin{bmatrix} 2.1667 & 2.5000 \\ -3.500 & -3.0000 \end{bmatrix}, & \boldsymbol{D}_1 = \begin{bmatrix} 4.5000 & 4.0000 \\ -2.3333 & -0.5000 \end{bmatrix}, & \boldsymbol{D}_0 = \boldsymbol{I}_{2\times 2} \end{array} \right\}$$

Combine the obtained results to parametrically compute the feedback gain matrix that assigns the following block eigen pairs $\{(\boldsymbol{X}_1, \boldsymbol{R}_1), (\boldsymbol{X}_2, \boldsymbol{R}_2)\}$ *to the given system*:

$$\boldsymbol{N}(\boldsymbol{R}_1) = \begin{bmatrix} -40.3015 & -78.2392 \\ -25.1617 & -55.5606 \\ -16.1399 & -24.1786 \\ -8.5218 & -31.3819 \end{bmatrix},\quad \boldsymbol{N}(\boldsymbol{R}_2) = \begin{bmatrix} -6.5825 & -12.6966 \\ -5.7427 & -10.8798 \\ -1.8398 & -3.3168 \\ -3.4028 & -7.5630 \end{bmatrix}$$

$$\boldsymbol{D}(\boldsymbol{R}_1) = \begin{bmatrix} 2.6803 & -77.5664 \\ 63.9972 & 206.3270 \end{bmatrix},\quad \boldsymbol{D}(\boldsymbol{R}_2) = \begin{bmatrix} -2.3056 & -7.6260 \\ 0.4336 & 4.9536 \end{bmatrix}$$

$$\boldsymbol{K}_1 = [\boldsymbol{D}(\boldsymbol{R}_1)\ \boldsymbol{D}(\boldsymbol{R}_2)][\boldsymbol{N}(\boldsymbol{R}_1)\ \boldsymbol{N}(\boldsymbol{R}_2)]^{-1} = \begin{bmatrix} 14.2169 & -5.8068 & -24.6660 & -3.6878 \\ -34.7681 & 17.1882 & 50.2814 & 10.9354 \end{bmatrix}$$

To achieve static tracking of a desired trajectory, the static gain matrix is designed as follows:

$$\boldsymbol{K}_2 = (\boldsymbol{H}(0))^{+} = \left(\boldsymbol{C}_{new}(\lambda \boldsymbol{I} - \boldsymbol{A}_{new})^{-1} \boldsymbol{B}_{new} \Big|_{\lambda=0} \right)^{+} = -\left(\boldsymbol{C}(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})^{-1} \boldsymbol{B} \right)^{+}$$

$$\boldsymbol{K}_2 = \begin{pmatrix} -1.6098 & -8.9335 \\ 5.2640 & 21.4814 \end{pmatrix}$$

where " $+$ " *denote the pseudo-inverse*.

3.2. Block Structure Assignment via State Derivative Feedback

This section addresses a feedback strategy that relies solely on state derivatives, rather than full-state measurements, and is therefore referred to as ***state-derivative feedback*** [26]. Naturally, this leads to the problem of ***block-structure*** assignment using such feedback. Consider a linear, time-invariant system that is completely controllable:

$$\dot{\mathbf{x}}(t) = \boldsymbol{A}\mathbf{x}(t) + \boldsymbol{B}\mathbf{u}(t);\ \ \mathbf{x}(t_0) = \mathbf{x}_0 \tag{36}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector, $\mathbf{u}(t) \in \mathbb{R}^m$ is the control input $(m < n)$, and $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ are the system and input matrices, respectively. It is assumed that the system is completely block controllable with a controllability index $(\ell = n/m)$ and that $\boldsymbol{B}$ has a full column rank m.

The goal is to stabilize the system via a linear feedback law that enforces a desired dynamic behavior. The block-structure assignment problem consists of finding a state-derivative feedback control law:

$$\mathbf{u}(t) = \boldsymbol{K}_1\dot{\mathbf{x}}(t) + \boldsymbol{K}_2\mathbf{r}(t) \tag{37}$$

which assigns the prescribed closed-loop latent values and associated latent vectors (or the target block roots), ensuring system stability and desired performance [7]. The resulting closed-loop dynamics are as follows:

$$\dot{\mathbf{x}}(t) = \Big((\boldsymbol{I}_n - \boldsymbol{B}\boldsymbol{K}_1)^{-1}\boldsymbol{A}\Big)\mathbf{x}(t) + (\boldsymbol{I}_n - \boldsymbol{B}\boldsymbol{K}_1)^{-1}\boldsymbol{B}\boldsymbol{K}_2\mathbf{r}(t) \tag{38}$$

Transforming the system into block controller coordinates via $\mathbf{x}_c(t) = \boldsymbol{T}_c\mathbf{x}(t)$ gives the following closed-loop state equation:

$$\dot{\mathbf{x}}_c(t) = \Big((\boldsymbol{I}_n - \boldsymbol{B}_c\boldsymbol{K}_{c1})^{-1}\boldsymbol{A}_c\Big)\mathbf{x}_c(t) + (\boldsymbol{I}_n - \boldsymbol{B}_c\boldsymbol{K}_{c1})^{-1}\boldsymbol{B}_c\boldsymbol{K}_{c2}\mathbf{r}(t) \tag{39}$$

where $\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$ is the identity matrix. It is assumed that $(\boldsymbol{I}_n + \boldsymbol{B}_c\boldsymbol{K}_{c1})$ is nonsingular, so the closed-loop system is well-defined. The corresponding characteristic polynomial satisfies the following:

$$\det\Big[\lambda\boldsymbol{I}_n - (\boldsymbol{I}_n - \boldsymbol{B}_c\boldsymbol{K}_{c1})^{-1}\boldsymbol{A}_c\Big] = 0 \tag{40}$$

Denote the right block-eigenvector matrix of $\boldsymbol{A}_{new} = (\boldsymbol{I}_n - \boldsymbol{B}_c\boldsymbol{K}_{c1})^{-1}\boldsymbol{A}_c$ by $\boldsymbol{X}_{ci}$, and we then have by definition: $\Big((\boldsymbol{I}_n - \boldsymbol{B}_c\boldsymbol{K}_{c1})^{-1}\boldsymbol{A}_c\Big)\boldsymbol{X}_{ci} = \boldsymbol{X}_{ci}\boldsymbol{R}_i$ with $i = 1, 2, \cdots, \ell$

$$\boldsymbol{A}_c\boldsymbol{X}_{ci} = (\boldsymbol{I}_n - \boldsymbol{B}_c\boldsymbol{K}_{c1})\boldsymbol{X}_{ci}\boldsymbol{R}_i \Leftrightarrow \boldsymbol{A}_c\boldsymbol{X}_{ci} - \boldsymbol{X}_{ci}\boldsymbol{R}_i = -\boldsymbol{B}_c\boldsymbol{K}_{c1}\boldsymbol{X}_{ci}\boldsymbol{R}_i \quad \text{with} \quad i = 1, 2, \cdots, \ell \tag{41}$$

Letting $\boldsymbol{\mathcal{W}}_{ci} = \boldsymbol{K}_{1c}\boldsymbol{X}_{ci}$ we will receive the following matrix equation:

$$\boldsymbol{A}_c\boldsymbol{X}_{ci} - \boldsymbol{X}_{ci}\boldsymbol{R}_i = -\boldsymbol{B}_c\boldsymbol{\mathcal{W}}_{ci}\boldsymbol{R}_i \quad \text{with} \quad i = 1, 2, \cdots, \ell \tag{42}$$

Using the Kronecker product, this can be compactly written as follows:

$$\begin{bmatrix} \boldsymbol{I}_m \otimes \boldsymbol{A}_c - \boldsymbol{R}_i^\top \otimes \boldsymbol{I}_n \ \vdots \ \Big(\boldsymbol{R}_i^\top \otimes \boldsymbol{B}_c\Big) \end{bmatrix} \begin{bmatrix} \mathrm{vec}(\boldsymbol{X}_{ci}) \\ \mathrm{vec}(\boldsymbol{\mathcal{W}}_{ci}) \end{bmatrix} = \mathbf{O} \tag{43}$$

Define the following structured matrices $\boldsymbol{S}(\boldsymbol{R}_i) = \begin{bmatrix} \boldsymbol{I}_m \otimes \boldsymbol{A}_c - \boldsymbol{R}_i^\top \otimes \boldsymbol{I}_n \ \vdots \ \Big(\boldsymbol{R}_i^\top \otimes \boldsymbol{B}_c\Big) \end{bmatrix}$ and $\boldsymbol{\mathcal{X}}(\boldsymbol{X}_{ci}, \boldsymbol{K}_{c1}) = \begin{bmatrix} \mathrm{vec}^\top(\boldsymbol{X}_{ci}) \ \mathrm{vec}^\top(\boldsymbol{\mathcal{W}}_{ci}) \end{bmatrix}^\top$. As shown previously, if $\boldsymbol{S}(\boldsymbol{R}_i)$ is singular, a finite solution to (43) exists such that $\boldsymbol{\mathcal{X}}(\boldsymbol{X}_{ci}, \boldsymbol{K}_{c1}) \in \boldsymbol{\mathcal{N}}(\boldsymbol{S}(\boldsymbol{R}_i))$ where $\boldsymbol{\mathcal{N}}$ denotes the null space of the matrix $\boldsymbol{S}(\boldsymbol{R}_i)$. Hence, $\boldsymbol{\mathcal{X}}(\boldsymbol{X}_{ci}, \boldsymbol{K}_{c1})$ can be constructed from the kernel space of the matrix $\boldsymbol{S}(\boldsymbol{R}_i)$.

Remark 6. *When the desired block poles are repeated, the corresponding block eigenvectors can be generated iteratively as:* $\boldsymbol{A}\boldsymbol{X}_k = \boldsymbol{X}_k\boldsymbol{R}_i + \boldsymbol{X}_{k-1}$ *or* $\mathrm{vec}(\boldsymbol{X}_k) = \Big(\boldsymbol{I}_m \otimes \boldsymbol{A} - \boldsymbol{R}_i^\top \otimes \boldsymbol{I}_n\Big)^{-1}\mathrm{vec}(\boldsymbol{X}_{k-1})$.

Similarly to the previous case, we need to parametrically determine the feedback gain matrix that assigns the full block structure using state-derivative feedback. The resulting closed-loop transfer function of the system $(\boldsymbol{A}, \boldsymbol{B})$ is given by

$$\boldsymbol{H}(\lambda) = \Big(\lambda\boldsymbol{I}_n - (\boldsymbol{I}_n - \boldsymbol{B}\boldsymbol{K}_1)^{-1}\boldsymbol{A}\Big)^{-1}(\boldsymbol{I}_n - \boldsymbol{B}\boldsymbol{K}_1)^{-1}\boldsymbol{B} = \boldsymbol{N}(\lambda)\boldsymbol{D}_{new}^{-1}(\lambda) = (\lambda(\boldsymbol{I}_n - \boldsymbol{B}\boldsymbol{K}_1) - \boldsymbol{A})^{-1}\boldsymbol{B} = \boldsymbol{N}(\lambda)\boldsymbol{D}_{new}^{-1}(\lambda)$$

This leads to

$$\boldsymbol{B}\cdot\boldsymbol{D}_{new}(\lambda) = (\lambda(\boldsymbol{I}_n - \boldsymbol{B}\boldsymbol{K}_1) - \boldsymbol{A})\cdot\boldsymbol{N}(\lambda) \tag{44}$$

After simplification of Equation (44) we have the following:

$$\lambda \cdot \boldsymbol{N}(\lambda) = \boldsymbol{A} \cdot \boldsymbol{N}(\lambda) + \lambda \cdot \boldsymbol{B} \cdot \boldsymbol{K}_1 \cdot \boldsymbol{N}(\lambda) + \boldsymbol{B} \cdot \boldsymbol{D}_{new}(\lambda) \tag{45}$$

The complete set of desired solvents satisfies Equation (45), giving

$$\boldsymbol{N}(\boldsymbol{R}_i)\boldsymbol{R}_i - \boldsymbol{B} \cdot \boldsymbol{K}_1 \cdot \boldsymbol{N}(\boldsymbol{R}_i)\boldsymbol{R}_i = \boldsymbol{A} \cdot \boldsymbol{N}(\boldsymbol{R}_i) + \boldsymbol{B} \cdot \boldsymbol{D}_{new}(\boldsymbol{R}_i) \quad i = 1, \cdots, \ell \tag{46}$$

Combining Equations (28) and (46) results in the matrix relation:

$$\boldsymbol{K}_1 \boldsymbol{N}(\boldsymbol{R}_i)\boldsymbol{R}_i = \boldsymbol{D}(\boldsymbol{R}_i) \qquad i = 1, \cdots, \ell \tag{47}$$

In a more compact form, this can be expressed as

$$\boldsymbol{K}_{c1} = \mathbb{W}\mathbb{V}^{-1}[\boldsymbol{N}_\ell, \boldsymbol{N}_{\ell-1}, \cdots, \boldsymbol{N}_1] \quad \text{where} \quad \boldsymbol{K}_1 = \boldsymbol{K}_{c1}\boldsymbol{T}_c \tag{48}$$

$$\text{With} \qquad \mathbb{W} = [\boldsymbol{D}(\boldsymbol{R}_1), \boldsymbol{D}(\boldsymbol{R}_2), \cdots, \boldsymbol{D}(\boldsymbol{R}_\ell)], \quad \mathbb{V} = [\boldsymbol{N}(\boldsymbol{R}_1), \boldsymbol{N}(\boldsymbol{R}_2), \cdots, \boldsymbol{N}(\boldsymbol{R}_\ell)]$$

Example 2. *Consider the 2-by-2 system represented by the following state space model*

$$\boldsymbol{A} = \begin{bmatrix} 10.1053 & -13.5732 & 7.0930 & -22.2085 \\ 5.0500 & -6.5831 & 2.0206 & -8.4311 \\ 8.4457 & -10.0912 & 5.1693 & -18.1104 \\ 5.3245 & -6.8585 & 4.2611 & -12.6915 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 7.1233 & 10.8439 \\ 3.7903 & 5.4963 \\ 4.4474 & 8.3113 \\ 2.5707 & 4.9250 \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} 0.2226 & 0.0658 & 0.7645 & 0.5436 \\ -0.3164 & 1.1572 & -0.3873 & 0.9099 \end{bmatrix}, \qquad \boldsymbol{D} = \mathbf{O}_{2\times 2}$$

The system is block controllable since the matrix $\left[\boldsymbol{B}, \boldsymbol{AB}, \boldsymbol{A}^2\boldsymbol{B}, \boldsymbol{A}^3\boldsymbol{B}\right]$ *has full rank, giving a controllability index* $\ell = n/m = 2$ *so that,* $\boldsymbol{\Omega}_c = [\boldsymbol{B}, \boldsymbol{AB}]$. *The desired closed-loop latent structures (latent roots and corresponding latent vectors) are chosen as follows,*

$$\left\{ (-10, -5, -1.5, -3), \begin{bmatrix} -0.8697 & 0.7632 & 0.8828 & -0.1950 \\ 0.4935 & -0.6461 & -0.4698 & 0.9808 \end{bmatrix} \right\}$$

This defines the next block roots and their corresponding block eigenvectors to be assigned:

$$\left\{ \boldsymbol{R}_1 = \begin{bmatrix} -20.1658 & -17.9154 \\ 8.6055 & 5.1658 \end{bmatrix}, \boldsymbol{X}_1 = \begin{bmatrix} -40.6166 & -60.0765 \\ -23.5804 & -32.7722 \\ -7.1815 & -21.7943 \\ -3.6278 & -12.5329 \end{bmatrix}; \boldsymbol{R}_2 = \begin{bmatrix} -1.3225 & 0.3335 \\ -0.8927 & -3.1775 \end{bmatrix}, \boldsymbol{X}_2 = \begin{bmatrix} -9.3877 & -20.5580 \\ -4.3636 & -9.4607 \\ -2.3202 & -9.9778 \\ -1.9660 & -6.7112 \end{bmatrix} \right\}$$

In this example, the goal is to design a state-derivative feedback control law of the form: $\mathbf{u}(t) = \boldsymbol{K}_1\dot{\mathbf{x}}(t) + \boldsymbol{K}_2\mathbf{r}(t)$ *, so that the desired latent structure (block roots) is assigned to achieve the specified performance. Applying the proposed state-derivative feedback procedure, the matrices* $\boldsymbol{D}_i$ *and* $\boldsymbol{N}_i$ *can be derived from the system matrices:*

$$\boldsymbol{N}_2 = \begin{bmatrix} 9.7132 & 11.5229 \\ 5.5556 & 6.7398 \\ 10.9810 & 14.9481 \\ 5.8303 & 8.0806 \end{bmatrix}, \quad \boldsymbol{N}_1 = \begin{bmatrix} 7.1233 & 10.8439 \\ 3.7903 & 5.4963 \\ 4.4474 & 8.3113 \\ 2.5707 & 4.9250 \end{bmatrix}, \quad \boldsymbol{N}_0 = \boldsymbol{O}_{4\times 2}$$

$$\boldsymbol{D}_2 = \begin{bmatrix} -0.7345 & -2.5844 \\ 3.1428 & 6.1675 \end{bmatrix}, \quad \boldsymbol{D}_1 = \begin{bmatrix} -1.3344 & -4.3346 \\ 2.2342 & 5.3344 \end{bmatrix}, \quad \boldsymbol{D}_0 = \boldsymbol{I}_{2\times 2}$$

The feedback gain matrix can then be constructed from the obtained results:

$$N(R_1)R_1 = \begin{bmatrix} 300.0791 & 417.3205 \\ 193.4962 & 253.1573 \\ -42.7302 & 16.0744 \\ -34.6945 & 0.2510 \end{bmatrix}, \quad N(R_2)R_2 = \begin{bmatrix} 30.7673 & 62.1922 \\ 14.2164 & 28.6060 \\ 11.9757 & 30.9307 \\ 8.5911 & 20.6693 \end{bmatrix}$$

$$D(R_1) = \begin{bmatrix} 241.3619 & 267.6612 \\ -125.0898 & -133.7889 \end{bmatrix}, \quad D(R_2) = \begin{bmatrix} 6.3509 & 9.2428 \\ -0.5569 & -0.2386 \end{bmatrix}$$

$$K_{c1} = \begin{bmatrix} D(R_1) \vdots D(R_2) \end{bmatrix} \begin{bmatrix} N(R_1)R_1 \vdots N(R_2)R_2 \end{bmatrix}^{-1} \begin{bmatrix} N_2 \vdots N_1 \end{bmatrix} = \begin{bmatrix} -0.6496 & -2.1549 & 1.2084 & 0.4517 \\ 0.4303 & 1.7686 & -0.2692 & 0.4933 \end{bmatrix}$$

$$K_1 = K_{c1} \begin{bmatrix} \begin{bmatrix} O_2 \vdots I_2 \end{bmatrix} \begin{bmatrix} B \vdots AB \end{bmatrix}^{-1} \\ \begin{bmatrix} O_2 \vdots I_2 \end{bmatrix} \begin{bmatrix} B \vdots AB \end{bmatrix}^{-1} A \end{bmatrix} = \begin{bmatrix} 0.4272 & 0.3291 & 0.4058 & -1.9012 \\ -0.8686 & 1.0059 & -1.6732 & 3.7137 \end{bmatrix}$$

The static pre-compensation gain is determined as follows:

$$K_2 = (H(0))^{+} = -\left[C(A + BK)^{-1}B\right]^{+} = \begin{bmatrix} -2.9525 & 9.2726 \\ 3.7843 & -11.3856 \end{bmatrix}$$

The latent-structure (block-root) assignment method has been demonstrated as a tool for designing both state and state-derivative feedback controllers, producing a closed-loop system with prescribed characteristics. This approach ensures stability when the necessary and sufficient conditions are satisfied. Compared to conventional state feedback, the state-derivative feedback controller can, in certain cases, achieve equivalent performance with smaller gain values. From a practical standpoint, minimizing feedback gains is desirable, as lower gains result in smaller control signals and reduced energy consumption [7,27].

4. Application to the Gas Turbine Machine

4.1. The Gas Turbine Machine Control via the Proposed Method

This section studies the GE MS5001P gas turbine at M'SILA power plant (Algeria), selected for its available real-time data. The plant has 22 single-shaft turbines driving the main generators. Each turbine includes an axial compressor, a combustion chamber, and a multi-stage turbine. Rare start-up/shutdown transients are excluded [29–32]. Key characteristics of this gas turbine machine are in Table 1.

Table 1. General performance of gas turbine GE MS5001P.

Quantity	Value	Quantity	Value
Compressor stages	16	Output	24,700 (kW)
Firing temperature	1730 (°F)	Heat rate	12,950 (kJ/kW-h)
Exhaust temperature	898 (°F)	Rotor Speed	5355 (rpm) (105%)
Air flow	928.5 (103 Lb/h)	Efficiency	27.8%

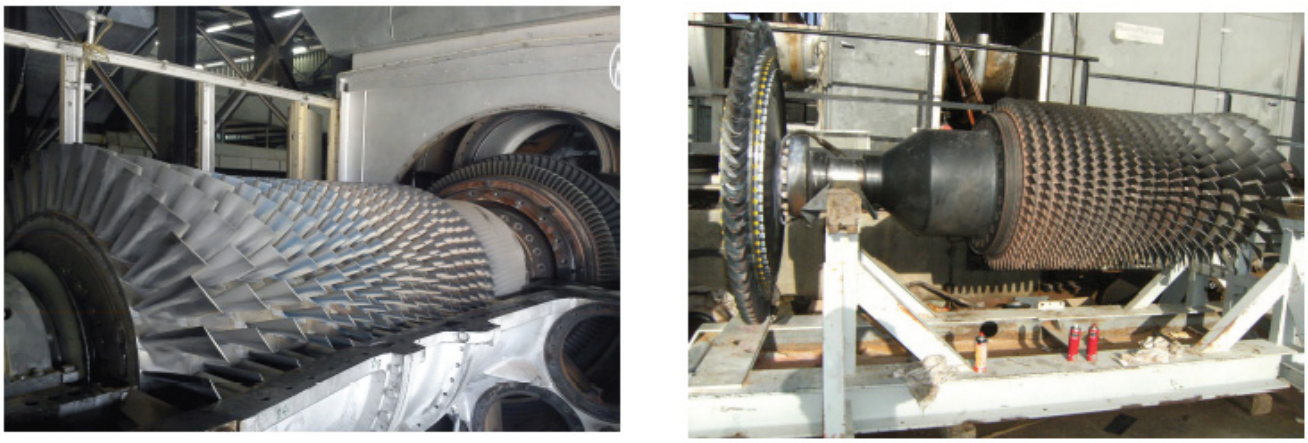
The single-shaft industrial gas turbine GE MS5001P is considered to operate in the vicinity of a nominal steady-state condition under ISO ambient conditions. The dynamic model is derived by linearizing the nonlinear thermodynamic equations about a selected equilibrium (trim) point [34].

Assumptions:

- Lumped-parameter model;
- Ideal gas behavior;

- Small perturbations around equilibrium;
- No variable geometry;
- Neglecting shaft torsional flexibility.

Numerous parameters influence gas turbine dynamics (Appendix A), with their impact varying according to their role within the main sections of the turbine, namely the compressor, the combustion chamber, and the turbine. In this study, the modeling effort is deliberately restricted to the two principal output variables of the GEMS5001P gas turbine under normal operating conditions: the rotor speed and the exhaust gas temperature. These outputs are directly affected by, and dynamically coupled with, three main variables: the gas control valve (GCV), the axial compressor discharge temperature (TCD), and the axial compressor discharge pressure (PCD) [28]. Figure 1. illustrates the axial compressor of the GE 5001P gas turbine and the turbine section of the same system.



(**a**) (**b**)

Figure 1. GE 5001P gas turbine components: (**a**) axial compressor; (**b**) turbine.

The overall dynamics of the gas turbine can be described by a nonlinear multivariable state–space model that captures the coupled mechanical, thermodynamic, and fluid-flow phenomena of the compressor–combustor–turbine assembly. The nonlinear model is expressed as $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$; $\mathbf{y} = \mathbf{h}(\mathbf{x}(t))$ where $\mathbf{x} \in \mathbb{R}^6$, $\mathbf{u} \in \mathbb{R}^2$, and $\mathbf{y} \in \mathbb{R}^2$. The state vector is defined as $\mathbf{x}(t) = [\omega_r, T_{CD}, P_{CD}, T_t, T_E, \tau_s]^\top$ with ω_r is the rotor speed, T_{CD} and P_{CD} are the compressor discharge temperature and pressure, T_t is the turbine inlet temperature, T_E is the exhaust gas temperature, and τ_s is the shaft torque. These states describe the dominant nonlinear interactions between the rotating shaft dynamics and the thermal processes occurring in the compressor, combustor, and turbine. The input vector is given by $\mathbf{u}(t) = \left[CGV, T_{CD}^{\text{ref}}\right]^\top$ where CGV denotes the gas control valve position governing the fuel mass flow rate into the combustor, and T_{CD}^{ref} is the input compressor discharge temperature, which indirectly affects combustion efficiency and downstream thermal dynamics [30–33]. The complex nonlinear model can be written as

$$\frac{d}{dt}\begin{bmatrix} \omega_r \\ T_{CD} \\ P_{CD} \\ T_t \\ T_E \\ \tau_s \end{bmatrix} = \begin{bmatrix} \left[\eta_t \dot{m}_f c_p (T_t - T_E) - \eta_c \dot{m}_a c_p R\left(T_{CD} - T_{CD}^{\text{ref}}\right) - J\omega_r \tau_s\right] / J\omega_r \\ \left[T_{CD}^{\text{ref}} - T_{CD} + k_1(\omega_r - \omega_{r0}) + k_2(P_{CD} - P_{CD0}) + k_3 \dot{m}_f\right] / \tau_{CD} \\ \left[\left(P_{CD}^{\text{ss}}(\omega_r, \dot{m}_f) - P_{CD}\right)\right] / \tau_P \\ \left[\alpha_f \dot{m}_f + \beta_a\left(T_{CD} - T_{CD}^{\text{ref}}\right) - T_t\right] / \tau_t \\ \left[T_t - T_E + \gamma(\omega_r - \omega_{r0})\right] / \tau_E \\ \left[k_t(\tau_t - \tau_c) - \tau_s\right] / \tau_\tau \end{bmatrix} \tag{49}$$

where J is the shaft inertia. η_t, η_c are the turbine and compressor efficiency factors. $\dot{m}_\mathrm{f}$ is the fuel mass flow rate (controlled via GCV $\dot{m}_\mathrm{f} = k_\mathrm{GCV} \cdot u_1$). $\dot{m}_\mathrm{a}$ is the air mass flow rate through the compressor. c_p, R are the specific heats. k_1, k_2, k_3 are gains capturing rotor-compressor coupling, and τ_{CD} is the thermal time constant. $P_{CD}^{\mathrm{ss}}(\omega_r, \dot{m}_\mathrm{f})$ is the steady-state compressor pressure as a function of rotor speed, τ_P is the pressure time constant. α_f, β_a representing the effect of fuel flow and compressed air on turbine temperature. γ captures the dynamic coupling between rotor speed and exhaust gas temperature, τ_E is exhaust thermal constant. k_t is a mechanical gain and τ_τ is the shaft torque time constant. The measured outputs are selected as $\mathbf{y}(t) = [\omega_r, T_E]^\top$ since rotor speed and exhaust gas temperature are the most relevant performance indicators during normal operating conditions.

This is now a completely explicit nonlinear model, including all six states, two inputs, and the outputs. It is suitable for simulation, control design, and linearization. The linear model is obtained by *linearization around a steady operating point* $(\mathbf{x}_0, \mathbf{u}_0)$. The states' deviations considered are as follows: the rotor speed $x_1 = \Delta\omega_r$, the compressor discharge temperature $x_2 = \Delta T_{CD}$, the compressor discharge pressure $x_3 = \Delta P_{CD}$, the turbine inlet temperature $x_4 = \Delta T_t$, the exhaust gas temperature $x_5 = \Delta T_E$, and the shaft torque $x_6 = \Delta\tau_s$. These states capture the dominant interactions among the compressor, combustor, and turbine, including both thermal and mechanical dynamics. The system is influenced by two independent inputs: the gas control valve $u_1 = \Delta\mathrm{GCV}$, which modulates the fuel flow into the combustor, and the compressor discharge temperature reference $u_2 = \Delta T_{\mathrm{CD}}^{\mathrm{ref}}$, which affects the air–fuel mixture and consequently the downstream thermal states. The two measured outputs are the rotor speed and the exhaust gas temperature, denoted as $y_1 = \Delta\omega_r$ and $y_2 = \Delta T_e$, which are the primary indicators of turbine performance in normal operating conditions [28,30].

Finally, the linear model is obtained by linearizing the nonlinear dynamics around a nominal steady-state operating point $(\mathbf{x}_0, \mathbf{u}_0)$, corresponding to normal turbine operation, leading to the deviation model $\Delta\dot{\mathbf{x}} = \boldsymbol{A}\cdot\Delta\mathbf{x}(t) + \boldsymbol{B}\cdot\Delta\mathbf{u}(t)$; $\Delta\mathbf{y}(t) = \boldsymbol{C}\cdot\Delta\mathbf{x}(t)$. The operating point is computed by solving the trimming conditions $\mathbf{f}(x, u) = 0$ using optimization-based methods. The state-space matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ can, in principle, be obtained by evaluating the Jacobians of the nonlinear system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$; $\mathbf{y} = \mathbf{h}(\mathbf{x}(t))$ at the nominal operating point $(\mathbf{x}_0, \mathbf{u}_0)$. However, since many of the physical constants and parameters are not known a priori, an identification-based approach is employed to derive an approximate continuous linear model that accurately captures the system dynamics around the operating point [34].

4.2. Recursive Methods for Continuous Time System Identification

The multivariable continuous time recursive least square (MCTRLS) is an algorithm stated to minimize the criterion

$$\mathcal{J} = \int_0^t \left[\boldsymbol{\varphi}^\top(\tau)\boldsymbol{\theta}(\tau) - \mathbf{y}^\top(\tau)\right]^2 d\tau \implies \frac{\partial}{\partial\boldsymbol{\theta}}\left[\int_0^t \mathbf{e}^2(\tau)d\tau\right] = 2\int_0^t \boldsymbol{\varphi}(\tau)\left[\boldsymbol{\varphi}^\top(\tau)\boldsymbol{\theta}(\tau) - \mathbf{y}^\top(\tau)\right]d\tau = 0 \quad (50)$$

So that the least square estimate is

$$\boldsymbol{\theta}(t) = \left[\int_0^t \boldsymbol{\varphi}(\tau)\boldsymbol{\varphi}^\top(\tau)d\tau\right]^{-1}\left[\int_0^t \boldsymbol{\varphi}(\tau)\mathbf{y}^\top(\tau)d\tau\right] \quad (51)$$

For adaptive control applications, we are interested in recursive formulation where the parameters are updates continuously on the basis of input output data. Such update may be obtained by defining

$$\boldsymbol{P}(t) = \left[\int_0^t \boldsymbol{\varphi}(\tau)\boldsymbol{\varphi}^\top(\tau)d\tau\right]^{-1} \implies \frac{d}{dt}\left[\boldsymbol{P}^{-1}(t)\right] = \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^\top(t) \quad (52)$$

We use the fact that

$$\frac{d\boldsymbol{I}}{dt}=\frac{d}{dt}\left[\boldsymbol{P}(t)\boldsymbol{P}^{-1}(t)\right]=\left[\frac{d\boldsymbol{P}}{dt}\right]\cdot\boldsymbol{P}^{-1}(t)+\boldsymbol{P}(t)\cdot\left[\frac{d\boldsymbol{P}^{-1}}{dt}\right]=0\implies\frac{d\boldsymbol{P}}{dt}=-\boldsymbol{P}(t)\left[\frac{d\boldsymbol{P}^{-1}}{dt}\right]\boldsymbol{P}(t)=-\boldsymbol{P}(t)\left[\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\top}(t)\right]\boldsymbol{P}(t)\quad(53)$$

On the other hand

$$\boldsymbol{\theta}(t)=\boldsymbol{P}(t)\left[\int_0^t\boldsymbol{\varphi}(\tau)\mathbf{y}^{\top}(\tau)d\tau\right]\implies\frac{d\boldsymbol{\theta}}{dt}=\boldsymbol{P}(t)\boldsymbol{\varphi}(t)\mathbf{y}^{\top}(t)-\boldsymbol{P}(t)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\top}(t)\boldsymbol{\theta}(t)=\boldsymbol{P}(t)\boldsymbol{\varphi}(t)\mathbf{e}^{\top}(t)\quad(54)$$

Finally, we can write (with some forgetting Factor) this steps in Algorithm 2.

Algorithm 2 Multivariable Continuous Time Recursive Least Square

Step 1:	Input $\mathbf{u}(t)$, $\mathbf{y}(t)$ and form the data matrix $\boldsymbol{\phi}(t)$	If we define the matrix $\boldsymbol{R}(t)=\boldsymbol{P}^{-1}(t)$ then
Step 2:	$\frac{d\boldsymbol{\theta}}{dt}=\boldsymbol{P}(t)\boldsymbol{\phi}(t)\mathbf{e}^{\top}(t)$	$\frac{d\boldsymbol{\theta}}{dt}=\boldsymbol{R}^{-1}(t)\boldsymbol{\phi}(t)\mathbf{e}^{\top}(t)$
Step 3:	$\mathbf{e}^{\top}(t)=\mathbf{y}^{\top}(t)-\boldsymbol{\phi}^{\top}(t)\boldsymbol{\theta}(t)$	$\mathbf{e}^{\top}(t)=\mathbf{y}^{\top}(t)-\boldsymbol{\phi}^{\top}(t)\boldsymbol{\theta}(t)$
Step 4:	$\frac{d\boldsymbol{P}}{dt}=\alpha\boldsymbol{P}(t)-\boldsymbol{P}(t)\left[\boldsymbol{\phi}(t)\boldsymbol{\phi}^{\top}(t)\right]\boldsymbol{P}(t)$	$\frac{d\boldsymbol{R}}{dt}=\boldsymbol{\phi}(t)\boldsymbol{\phi}^{\top}(t)-\alpha\boldsymbol{R}(t)$

Linear continuous-time systems can be described in two main ways: the state-space (internal) representation, which captures the system's dynamics via state variables, and the input-output representation, expressed either as a time-domain differential equation or as an n^{th} order transfer function in the s-domain. Previously, we reviewed identification methods based on numerical integration, which require measurements of all states and inputs. In contrast, the method presented here relies only on the input sequences and a single output sequence to estimate the continuous-time system parameters.

The trapezoidal rule approximates the integrand $\mathbf{y}(t)$ by a linear function over each interval $(k-1)T\le t\le kT$. Within this interval, the linear approximation—known as a trapezoidal pulse function—can be expressed as follows:

$$\mathbf{f}(\mathbf{y})=\left\{\tfrac{(kT-t)}{T}\mathbf{y}_{k-1}-\tfrac{(t-(k-1)T)}{T}\mathbf{y}_k\right\};\qquad\text{for }\ (k-1)T\le t\le kT\quad(55)$$

Here, $\mathbf{y}_k$ denotes $\mathbf{y}(kT)$. The identification method requires successive integrals of $\mathbf{y}(t)$ to estimate the continuous-time system parameters. Accordingly, a set of approximate formulas is used to compute these integrals. Over the interval $(k-1)T\le t\le kT$, the area under the linear approximation of $\mathbf{y}(t)$ equals the standard trapezoidal integral. The recursive formula for the first and second integral of $\mathbf{y}(t)$ then can be expressed as follows:

$$\boldsymbol{I}_{1,k}=\boldsymbol{I}_{1,k-1}+\frac{T}{2}(\mathbf{y}_{k-1}+\mathbf{y}_k);\quad\text{and}\quad\boldsymbol{I}_{2,k}=\boldsymbol{I}_{2,k-1}+T\cdot\boldsymbol{I}_{1,k-1}+\frac{T^2}{3!}(2\mathbf{y}_{k-1}+\mathbf{y}_k)\quad(56)$$

In general, we can express the n^{th} integral of $\mathbf{y}(t)$ by the following equation

$$\boldsymbol{I}_{1,k}=\sum_{i=1}^{n}\left\{\tfrac{T^{n-i}}{(i-1)!}\boldsymbol{I}_{n-i+1,k-1}\right\}+\tfrac{T^n}{(n+1)!}\left[n\mathbf{y}_{k-1}+\mathbf{y}_k\right]\quad(57)$$

Now, the identification problem can be easily solved by simple applying this equation to the differential equation of continuous-time systems. Consider the MIMO continuous-time system described by $\left[\boldsymbol{I}\lambda^n+\sum_{i=1}^{n}\boldsymbol{D}_i\lambda^{n-i}\right]\mathbf{y}(\lambda)=\left[\sum_{i=0}^{m}\boldsymbol{N}_i\lambda^{m-i}\right]\mathbf{u}(\lambda)$ or by

$$\boldsymbol{D}(\lambda)\mathbf{y}(\lambda)=\boldsymbol{N}(\lambda)\mathbf{u}(\lambda)\Longleftrightarrow\left[\tfrac{d^n\mathbf{y}(t)}{dt^n}+\sum_{i=1}^{n}\boldsymbol{D}_i\tfrac{d^{n-i}}{dt^{n-i}}\right]\mathbf{y}(t)=\left[\sum_{i=0}^{m}\boldsymbol{N}_i\tfrac{d^{m-i}}{dt^{m-i}}\right]\mathbf{u}(t)\quad(58)$$

The identification problem now is to estimate the parameters $\boldsymbol{D}_i$ and $\boldsymbol{N}_i$. Integrating both sides of the above equation n times with respect to t over the subinterval $(k-1)T\le t\le kT$, we obtain

$$\Delta \mathbf{y}_k + \sum_{i=1}^{n} \boldsymbol{D}_i \Delta \boldsymbol{I}_{i,k}(\mathbf{y}) = \sum_{i=1}^{m} \boldsymbol{N}_i \Delta \boldsymbol{I}_{i,k}(\mathbf{u}) \quad \text{with} \quad \Delta \mathbf{y}_k = \mathbf{y}_k - \mathbf{y}_{k-1} \tag{59}$$

where the differences are $\Delta \mathbf{y}_k = \mathbf{y}_k - \mathbf{y}_{k-1}$, $\Delta \boldsymbol{I}_{i,k}(\mathbf{y}) = \boldsymbol{I}_{i,k}(\mathbf{y}) - \boldsymbol{I}_{i,k-1}(\mathbf{y})$ and $\Delta \boldsymbol{I}_{i,k}(\mathbf{u}) = \boldsymbol{I}_{i,k}(\mathbf{u}) - \boldsymbol{I}_{i,k-1}(\mathbf{u})$. The quantities $\boldsymbol{I}_{i,k}(\mathbf{y})$ and $\boldsymbol{I}_{i,k}(\mathbf{u})$ are the i^{th} integrals of $\mathbf{y}$ and $\mathbf{u}$ at the k^{th} sampling instant, respectively.

A multivariable continuous-time identification is performed via a trapezoidal integration–based recursive least squares (RLS) scheme as shown in Algorithm 3.

Algorithm 3 Multivariable Continuous Time Trapezoidal Integral RLS (For a 2nd order MDFD model)

Given:

- Sampling period $T_s = \Delta t$, final time t_F
- Input $\mathbf{u}_k \in \mathbb{R}^m$, output $\mathbf{y}_k \in \mathbb{R}^p$
- Number of samples $N = \text{round}(t_F/\Delta t)$,
- The overall time grid $t = 0 : \Delta t : t_F$

%-----Initial Values-----%

$\boldsymbol{\theta}_1 = \mathbf{O}$, with $\boldsymbol{\theta} \in \mathbb{R}^{8\times 2}$; % with θ = [D1′;D2′;N1′;N2′]

$\boldsymbol{P}_1 = \alpha \times \boldsymbol{I}_8 \in \mathbb{R}^8$; $\alpha = 10^{15}$; % Covariance matrix

$\mathbf{S}_1^y = [0\ 0]^\top$; $\mathbf{S}_1^u = [0\ 0]^\top$; $\Delta \boldsymbol{I}_2^y = [0\ 0]^\top$; $\Delta \boldsymbol{I}_2^u = [50\ 0]^\top$; % Initialize integral states

%-----Data Acquisition-----%

For k = 1:N, $\mathbf{y}_k = \boldsymbol{\mathcal{D}}(\mathbf{u}_k)$; **End** % where $\boldsymbol{\mathcal{D}}(\cdot)$ represents the real system dynamics.

For k = 2:N-1

%-------Integrations-------%

$\Delta \boldsymbol{I}_{k+1}^y = (\mathbf{y}_{k+1} + \mathbf{y}_k)\cdot(T_s/2)$; $\Delta \boldsymbol{I}_{k+1}^u = (\mathbf{u}_{k+1} + \mathbf{u}_k)\cdot(T_s/2)$; % Local trapezoidal integrals

$\mathbf{S}_k^y = \mathbf{S}_{k-1}^y + \Delta \boldsymbol{I}_k^y$; $\mathbf{S}_k^u = \mathbf{S}_{k-1}^u + \Delta \boldsymbol{I}_k^u$; % Cumulative 1st integral

$\Delta \mathbb{I}_{k+1}^y = \mathbf{S}_k^y \cdot T_s + [2\cdot \mathbf{y}_k + \mathbf{y}_{k+1}]\cdot(T_s^2/6)$; % Local 2nd order integrals

$\Delta \mathbb{I}_{k+1}^u = \mathbf{S}_k^u \cdot T_s + [2\cdot \mathbf{u}_k + \mathbf{u}_{k+1}]\cdot(T_s^2/6)$;

%-------RLS Algorithm-------%

$\Delta \mathbf{y}_k = \mathbf{y}_{k+1} - \mathbf{y}_k$; % Output increment

$\boldsymbol{\phi}_k = \left[-\Delta \boldsymbol{I}_{k+1}^y; -\Delta \mathbb{I}_{k+1}^y; \Delta \boldsymbol{I}_{k+1}^u; \Delta \mathbb{I}_{k+1}^u\right]$; % Regressor vector

$\boldsymbol{K}_k = \boldsymbol{P}_{k-1}\cdot\boldsymbol{\phi}_k / \left(1 + \boldsymbol{\phi}_k^\top \cdot \boldsymbol{P}_{k-1}\cdot\boldsymbol{\phi}_k\right)$; % RLS gain

$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \left[\boldsymbol{K}_k \cdot \left(\Delta \mathbf{y}_k^\top - \boldsymbol{\phi}_k^\top \cdot \boldsymbol{\theta}_{k-1}\right)\right]$; % Parameter update

$\boldsymbol{P}_k = \boldsymbol{P}_{k-1} - \left(\boldsymbol{K}_k \cdot \boldsymbol{\phi}_k^\top \cdot \boldsymbol{P}_{k-1}\right)$; % Covariance update

End

As discussed previously, it is often preferable to identify the parameters of a continuous-time system directly from discrete input-output measurements. The core concept is to integrate the system's differential equations over each sampling interval, yielding relationships between the sampled input-output data and the unknown model parameters. For instance, consider a state-space representation where the state $\mathbf{x}(t)$ is assumed measurable and $\mathbf{e}(t)$ is a white noise process. $\dot{\mathbf{x}}(t) = \boldsymbol{A}\mathbf{x}(t) + \boldsymbol{B}\mathbf{u}(t) + \mathbf{e}(t)$ where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{e} \in \mathbb{R}^n$, and $\mathbf{u} \in \mathbb{R}^m$. The terms $\boldsymbol{A}$, $\boldsymbol{B}$, are parameter matrices. The objective is to identify the system matrices from sampled input–state data $\mathbf{u}(t)$ and $\mathbf{x}(t)$ collected at $t = kT$, for $k = 0, \ldots, N$. The sampling period T is assumed sufficiently small to preserve the system dynamics. In practice, the identification is complicated by measurement noise.

If we integrate the differential equation in $\dot{\mathbf{x}}(t) = \boldsymbol{A}\mathbf{x}(t) + \boldsymbol{B}\mathbf{u}(t) + \mathbf{e}(t)$ between the limits $t_{k-1} = (k-1)T$ and $t_k = kT$, we obtain $dt = t_k - t_{k-1} = T$ and

$$\Delta \mathbf{x}_k = \mathbf{x}(t_k) - \mathbf{x}(t_{k-1}) = \boldsymbol{A}\int_{t_{k-1}}^{t_k} \mathbf{x}(t)dt + \boldsymbol{B}\int_{t_{k-1}}^{t_k} \mathbf{u}(t)dt + \int_{t_{k-1}}^{t_k} \mathbf{e}(t)dt = [\boldsymbol{A} \quad \boldsymbol{B}]\begin{bmatrix} \boldsymbol{I}_x(t_k) \\ \boldsymbol{I}_u(t_k) \end{bmatrix} + \boldsymbol{I}_e(t_k) \quad (60)$$

Therefore,

$$\Delta \mathbf{x}_k = \boldsymbol{\theta}^\top \boldsymbol{\varphi}_k(t_k) + \boldsymbol{I}_e(t_k) \quad \Longleftrightarrow \quad \Delta \mathbf{x}_k^\top = \begin{bmatrix} \boldsymbol{I}_x^\top(t_k) & \boldsymbol{I}_u^\top(t_k) \end{bmatrix}\begin{bmatrix} \boldsymbol{A}^\top \\ \boldsymbol{B}^\top \end{bmatrix} + \boldsymbol{I}_e^\top(t_k) = \boldsymbol{\varphi}_k^\top(t_k)\boldsymbol{\theta} + \boldsymbol{I}_e^\top(t_k) \quad (61)$$

With $\boldsymbol{I}_x(t_k) = \int_{t_{k-1}}^{t_k} \mathbf{x}(t)dt$; $\boldsymbol{I}_u(t_k) = \int_{t_{k-1}}^{t_k} \mathbf{u}(t)dt$; and $\boldsymbol{I}_e(t_k) = \int_{t_{k-1}}^{t_k} \mathbf{e}(t)dt$

Exact evaluation of the integrals is infeasible since only noisy samples of $\mathbf{x}(t_k)$, and $\mathbf{u}(t_k)$ are available. Therefore, the integrals are approximated using sampled data under an assumed model structure. Numerical integration–based identification methods are adopted, assuming a minimal (controllable and observable) system realization. If we choose a trapezoid to approximate the area of a typical panel, then we have the Trapezoidal rule, that is

$$\boldsymbol{I}_x(t_k) = \int_{t_{k-1}}^{t_k} \mathbf{x}(t)dt \approx \left\{\frac{\mathbf{x}_{k+1} + \mathbf{x}_k}{2}\right\}T; \quad \text{and} \quad \boldsymbol{I}_u(t_k) = \int_{t_{k-1}}^{t_k} \mathbf{u}(t)dt \approx \left\{\frac{\mathbf{u}_{k+1} + \mathbf{u}_k}{2}\right\}T \quad (62)$$

Means that

$$\mathbf{x}(t_{k+1}) - \mathbf{x}(t_k) = \left[\boldsymbol{A}\left(\frac{\mathbf{x}_{k+1} + \mathbf{x}_k}{2}\right) + \boldsymbol{B}\left(\frac{\mathbf{u}_{k+1} + \mathbf{u}_k}{2}\right)\right]T \quad (63)$$

$$\mathbf{x}(t_{k+1}) = \left(\boldsymbol{I} - \frac{\boldsymbol{A}T}{2}\right)^{-1}\left(\boldsymbol{I} + \frac{\boldsymbol{A}T}{2}\right)\mathbf{x}(t_k) + T\left(\boldsymbol{I} - \frac{\boldsymbol{A}T}{2}\right)^{-1}\boldsymbol{B}\left[\frac{\mathbf{u}_{k+1} + \mathbf{u}_k}{2}\right] = \boldsymbol{A}_d\mathbf{x}(t_k) + \boldsymbol{B}_d\mathbf{u}_0(t_k) \quad (64)$$

Where $\boldsymbol{A}_d = \left(\boldsymbol{I} - \frac{\boldsymbol{A}T}{2}\right)^{-1}\left(\boldsymbol{I} + \frac{\boldsymbol{A}T}{2}\right)$; $\boldsymbol{B}_d = T\left(\boldsymbol{I} - \frac{\boldsymbol{A}T}{2}\right)^{-1}\boldsymbol{B}$; $\mathbf{u}_0(t_k) = \left[\frac{\mathbf{u}_{k+1} + \mathbf{u}_k}{2}\right]$

By using this identification method we see that the discrete-time system obtained is identical to the one calculated by the bilinear z transformation. Clearly, if we have all the state measurements and the input samples, the parameters $\boldsymbol{A}$ and $\boldsymbol{B}$ can be easily estimated by applying the least squares approach

$$\Delta \mathbf{x}_k^\top = \boldsymbol{\varphi}_k^\top(t_k)\boldsymbol{\theta} + \boldsymbol{I}_e^\top(t_k) \quad \Longleftrightarrow \quad \begin{bmatrix} \Delta \mathbf{x}_1^\top \\ \Delta \mathbf{x}_2^\top \\ \vdots \\ \Delta \mathbf{x}_n^\top \end{bmatrix} = \begin{bmatrix} \int_{t_0}^{t_1} \mathbf{x}^\top(t)dt & \int_{t_0}^{t_1} \mathbf{u}^\top(t)dt \\ & \\ \vdots & \vdots \\ \int_{t_{n-1}}^{t_n} \mathbf{x}^\top(t)dt & \int_{t_{n-1}}^{t_n} \mathbf{u}^\top(t)dt \end{bmatrix}\begin{bmatrix} \boldsymbol{A}^\top \\ \\ \boldsymbol{B}^\top \end{bmatrix} + \begin{bmatrix} \int_{t_0}^{t_1} \mathbf{e}^\top(t)dt \\ \\ \vdots \\ \int_{t_{n-1}}^{t_n} \mathbf{e}^\top(t)dt \end{bmatrix} \quad (65)$$

Or in compact form we write

$$\Delta \mathbf{x}_k^\top = \boldsymbol{\varphi}_k^\top(t_k)\boldsymbol{\theta} + \boldsymbol{I}_e^\top(t_k) \quad \Longleftrightarrow \quad \begin{bmatrix} \Delta \mathbf{x}_1^\top \\ \vdots \\ \Delta \mathbf{x}_n^\top \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_x^\top(t_1) & \boldsymbol{I}_u^\top(t_1) \\ \vdots & \vdots \\ \boldsymbol{I}_x^\top(t_n) & \boldsymbol{I}_u^\top(t_n) \end{bmatrix}\begin{bmatrix} \boldsymbol{A}^\top \\ \\ \boldsymbol{B}^\top \end{bmatrix} + \begin{bmatrix} \boldsymbol{I}_e^\top(t_1) \\ \vdots \\ \boldsymbol{I}_e^\top(t_n) \end{bmatrix} \quad \Longleftrightarrow \quad \Delta \boldsymbol{X} = \boldsymbol{\phi}\boldsymbol{\theta} + \boldsymbol{\varepsilon} \quad (66)$$

By using the ordinary least square we obtain $\boldsymbol{\theta} = \left(\boldsymbol{\phi}^\top\boldsymbol{\phi}\right)^{-1}\boldsymbol{\phi}^\top\Delta\boldsymbol{X} + \left(\boldsymbol{\phi}^\top\boldsymbol{\phi}\right)^{-1}\boldsymbol{\phi}^\top\boldsymbol{\varepsilon}$ whose optimal values is given by

$$\boldsymbol{\theta}^\star = \left(\boldsymbol{\phi}^\top\boldsymbol{\phi}\right)^{-1}\boldsymbol{\phi}^\top\Delta\boldsymbol{X} \quad (67)$$

where the matrix $\boldsymbol{\phi}$ can be approximated by

$$\boldsymbol{\phi} = \begin{bmatrix} \boldsymbol{I}_x^\top(t_1)\, \boldsymbol{I}_u^\top(t_1) \\ \vdots\, \vdots \\ \boldsymbol{I}_x^\top(t_n)\, \boldsymbol{I}_u^\top(t_n) \end{bmatrix} \approx \frac{1}{2}\begin{bmatrix} (\mathbf{x}_1 + \mathbf{x}_0)^\top\, (\mathbf{u}_1 + \mathbf{u}_0)^\top \\ \vdots\, \vdots \\ (\mathbf{x}_n + \mathbf{x}_{n-1})^\top\, (\mathbf{u}_n + \mathbf{u}_{n-1})^\top \end{bmatrix} \approx \frac{1}{2}(\boldsymbol{\phi}_x + \boldsymbol{\phi}_u)$$

Now let us implement this identification method using Algorithm 4.

Algorithm 4 Multivariable Continuous Time Identification (For a Gas Turbine Model)

Given:

- $\Delta t = 0.0001$, % Sampling period
- $t_1 = 0$, $t_F = 5.0$ % Initial and final time
- $\mathbf{u}_k \in \mathbb{R}^m$, $\mathbf{x}_k \in \mathbb{R}^n$, % Input and state vectors
- $\boldsymbol{\theta} \in \mathbb{R}^{8\times 6}$ % Parameter matrix $\boldsymbol{\theta} = \left[\boldsymbol{A}^\top; \boldsymbol{B}^\top\right]$
- $N = \text{round}(t_F/\Delta t)$, % Number of samples
- $t = 0 : \Delta t : t_F$ % The overall time grid
- $\boldsymbol{\phi}_1 = [\,]$; $\Delta_x = [\,]$; $\boldsymbol{I}_1^x = [0\,0\,0\,0\,0\,0]^\top$; $\boldsymbol{I}_1^u = [0\,0]^\top$; % Initial values

%----- Data Acquisition -----%

For k = 1:N, $\mathbf{x}_k = \boldsymbol{\mathcal{D}}(\mathbf{u}_k)$; **End** % where $\boldsymbol{\mathcal{D}}(\cdot)$ represents the real system dynamics.

%-------State Space Model by Ordinary Least Squares (OLS) Algorithm-------%

For k = 1:N−1

$t_{k+1} = t_k$; % Update time index

$\boldsymbol{I}_k^x = \Delta t\cdot(\mathbf{x}_{k+1} + \mathbf{x}_k)/2$; $\boldsymbol{I}_k^u = \Delta t\cdot(\mathbf{u}_{k+1} + \mathbf{u}_k)/2$; % Trapezoidal rule for state and input

$\Delta\mathbf{x}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$; $\boldsymbol{I}_k = \left[(\boldsymbol{I}_k^x)^\top \;\; (\boldsymbol{I}_k^u)^\top\right]$; % Increment over Δt and regression row

$\boldsymbol{\phi}_{k+1} = [\boldsymbol{\phi}_k;\; \boldsymbol{I}_k]$; $\Delta_x = \left[\Delta_x;\; \Delta\mathbf{x}_k^\top\right]$; % Stack regressor matrix and state vector

$t_k = t_k + \Delta t$; % Advance time

End

$\boldsymbol{\theta} = \left(\boldsymbol{\phi}_N^\top\cdot\boldsymbol{\phi}_N\right)\cdot\boldsymbol{\phi}_N^\top\cdot\Delta_x$; % Batch OLS (offline estimation)

In previous code we have seen the identification by the OLS algorithm which requires of providing the data instantaneously, due to such reason we must change the identification scheme to the recursive version.

%-------State Space Model by RLS Algorithm-------%

$\boldsymbol{\theta} = 10\cdot\text{rand}(8,6)$; $\boldsymbol{P} = 10^5\cdot\text{eye}(8,8)$; % Initial parameter and covariance matrix

For k = 1:N

$t_{k+1} = t_k$; $t_k = t_k + \Delta t$; % Time update

$\boldsymbol{I}_k^x = \Delta t\cdot(\mathbf{x}_{k+1} + \mathbf{x}_k)/2$; $\boldsymbol{I}_k^u = \Delta t\cdot(\mathbf{u}_{k+1} + \mathbf{u}_k)/2$; % Trapezoidal integration

$\Delta\mathbf{x}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$; $\boldsymbol{\varphi}_k = [\boldsymbol{I}_k^x;\; \boldsymbol{I}_k^u]$; % Increment and instantaneous regressor

$\boldsymbol{K}_k = \boldsymbol{P}_k\cdot\boldsymbol{\varphi}_k/\left(1 + \boldsymbol{\varphi}_k^\top\cdot\boldsymbol{P}_k\cdot\boldsymbol{\varphi}_k\right)$; % RLS gain computation

$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \left(\boldsymbol{K}_k\cdot\left(\Delta\mathbf{x}_k^\top - \boldsymbol{\varphi}_k^\top\cdot\boldsymbol{\theta}_k\right)\right)$; % Parameter update

$\boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \left(\boldsymbol{K}_k\cdot\boldsymbol{\varphi}_k^\top\cdot\boldsymbol{P}_k\right)$; % Covariance update

$\boldsymbol{A}_k^{\text{est}} = \boldsymbol{\theta}_k^\top(1:6,:)$; $\boldsymbol{B}_k^{\text{est}} = \boldsymbol{\theta}_k^\top(7:8,:)$; % Extract estimated system matrices

End

The identified model is validated using statistical and performance metrics. The "Akaike Information Criterion" (AIC) and the "Final Prediction Error" (FPE) assess the accuracy–complexity trade-off, the "Root Mean Square Error" (RMSE) measures average prediction error, and "Variance Accounted For" (VAF) indicates how well the model reproduces output dynamics. Together, these indices ensure both accuracy and robustness. The AIC for an estimated model: AIC $=$ Ln(V) $+$ 2d/N where V is the loss function, d is the number of estimated parameters and N is the number of samples in the estimation data set. The Akaike's FPE is defined by FPE $= V[(1 + d/N)/(1 - d/N)]$. The normalized RMSE, is given by the expression: RMSE $= \sqrt{\left(\sum_{i=1}^{N}(\mathbf{y} - \mathbf{y}^{\text{est}})^2\right)/N}$. The "variance accounting

for" is defined as follows: $\mathrm{VAF} = 100\% \cdot \left[1 - \mathrm{var}\left(\mathbf{y} - \mathbf{y}^{\mathrm{est}}\right)/\mathrm{var}(\mathbf{y})\right]$. The optimal system order corresponds to the minima of AIC, FPE, RMSE and the maximum of VAF.

A schematic diagram for turbine identification is shown in Figure 2, illustrating the system inputs and outputs, the main components (axial compressor, combustion chamber, and turbine), and the electrical alternator as the load [31].

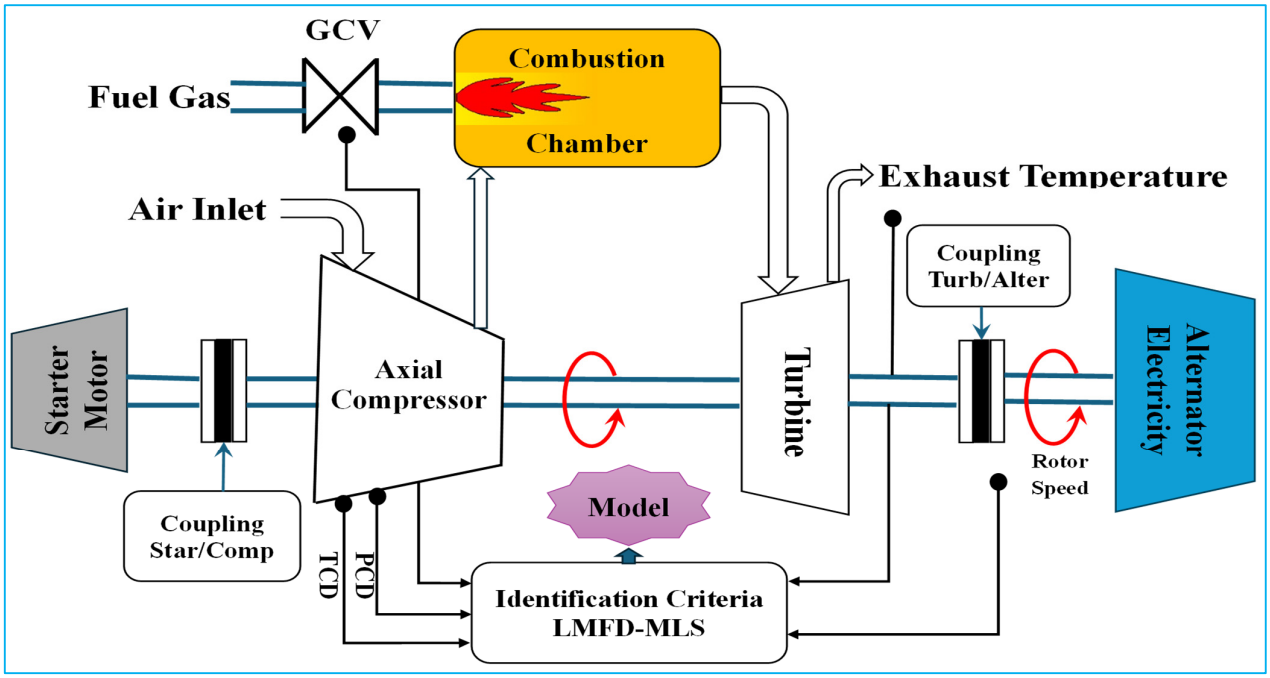


Figure 2. Schematic overview of the GE MS5001P gas turbine illustrating its parts.

The following state-space representation models the gas turbine dynamics

$$\left\{ \begin{array}{l} \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} -0.32 & 0.15 & 0.10 & 0 & 0.18 & 0.25 \\ 0.10 & -0.45 & 0.30 & 0.20 & 0 & 0 \\ 0.18 & 0.25 & -0.60 & 0.15 & 0 & 0 \\ 0 & 0.30 & 0.20 & -0.55 & 0.25 & 0 \\ 0 & 0.15 & 0 & 0.30 & -0.50 & 0 \\ 0.35 & 0 & 0 & 0 & 0.20 & -0.65 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0.60 & 0.10 \\ 0.50 & 0.30 \\ 0.20 & 0.40 \\ 0.80 & 0.15 \\ 0.70 & 0.20 \\ 0.90 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \end{array} \right\}$$

$u_1 = \Delta$GCV Gas control valve (fuel flow command);
$u_2 = \Delta T_{\mathrm{CD}}^{\mathrm{ref}}$ Compressor discharge temperature reference;
$y_1 = \Delta\omega_r$ Rotor speed;
$y_2 = \Delta T_e$ Exhaust gas temperature.

Physical Interpretation: Rotor speed is governed by shaft torque and turbine temperature, while compressor discharge temperature and pressure are strongly coupled, and exhaust temperature reflects combustion and turbine expansion dynamics. Shaft torque provides the link between thermal and mechanical subsystems. The GCV influences fuel flow, turbine temperature, torque, and speed, whereas the TCD reference regulates compressor and exhaust thermal states. Within this context, block controllability represents the ability to regulate these coupled mechanical–thermal subsystems through the available

actuators. The identification results are evaluated by comparing the model-predicted outputs with the measured data. Figure 3 shows the temporal evolution of the input signals, the resulting state trajectories, and the corresponding outputs.

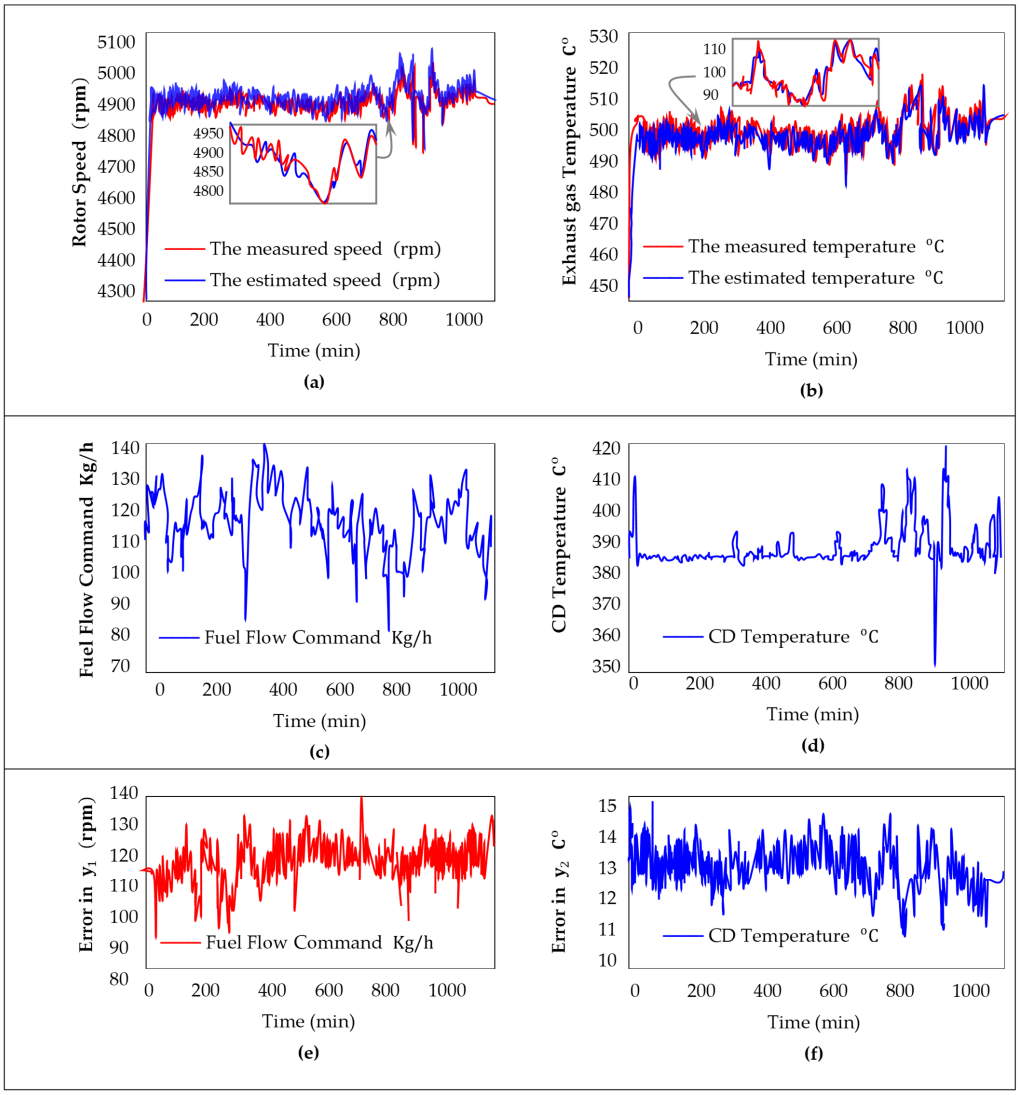


Figure 3. Measured outputs against model outputs and states with small prediction errors.

Figure 3 illustrates the temporal evolution of the system inputs, states, and outputs, together with the associated modeling errors. Specifically, Figure 3a compares the measured and estimated rotor speed $y_1 = \Delta\omega_r$, while Figure 3b presents the measured and estimated exhaust gas temperature $y_2 = \Delta T_e$, showing a close agreement in both cases. The applied fuel flow command is depicted in Figure 3c, and the corresponding combustion chamber (TCD) temperature response is shown in Figure 3d. The modeling accuracy is further assessed through the error signals, defined as the difference between measured and estimated outputs, where Figure 3e shows the error in rotor speed y_1 and Figure 3f shows the error in exhaust gas temperature y_2. As observed, the errors remain small over the entire operating range, indicating that the identified model accurately captures the system dynamics and provides a reliable basis for control design.

Table 2 shown below clarifies the change in the orders n until the best one is reached. From the results of this table, it can be concluded that the best model, which fits the minima of AICs, FPEs, RMSEs, the maximum of VAF validation criteria, and the minimum loss

function without pole/zero cancelation (left coprime), is the model of order $n = 6$ with number of blocks $\ell = 3$.

Table 2. Gas turbine system model orders with validations criteria, respectively.

LMFD ARX Model						
ℓ	n	**Loss Fun**	**AIC**	**FPE**	**RMSE**	**VAF (%)**
1	2	5.4385	1.7200	5.5826	5.2354	82.5000
2	4	1.9967	0.7445	2.1024	5.4573	85.9050
3	**6**	**1.0263**	**0.1319**	**1.1350**	**1.8217**	**87.7750**
4	8	1.1045	0.1789	1.1923	4.7200	87.6550

4.3. Supervisory Control of the Gas Turbine Machine

In order to fulfill the main objective of this paper, the proposed algorithm is implemented on the studied system in accordance with the previously described steps.

- ✓ The ratio $\ell = n/m$ equals to $6/2 = 3$, it is an integer
- ✓ The block controllability matrix is satisfied as $\text{rank}(\mathbf{\Omega}_c) = 6$ (full rank).
- ✓ The transformation matrix $\boldsymbol{T}_c \in \mathbb{R}^{6\times 6}$ is calculated by Equations (31)–(33).
- ✓ Construct the desired right block roots such that $\boldsymbol{V}_R$ is nonsingular.
- ✓ Find the matrix coefficient $\boldsymbol{N}_i$ and $\boldsymbol{D}_i$ by Equations (34) and (35).
- ✓ The feedback gain matrix $\boldsymbol{K}_c$ is calculated based on (32) or (48).

The overall supervisory control architecture, including the reference adjustment mechanism, is illustrated in Figure 4.

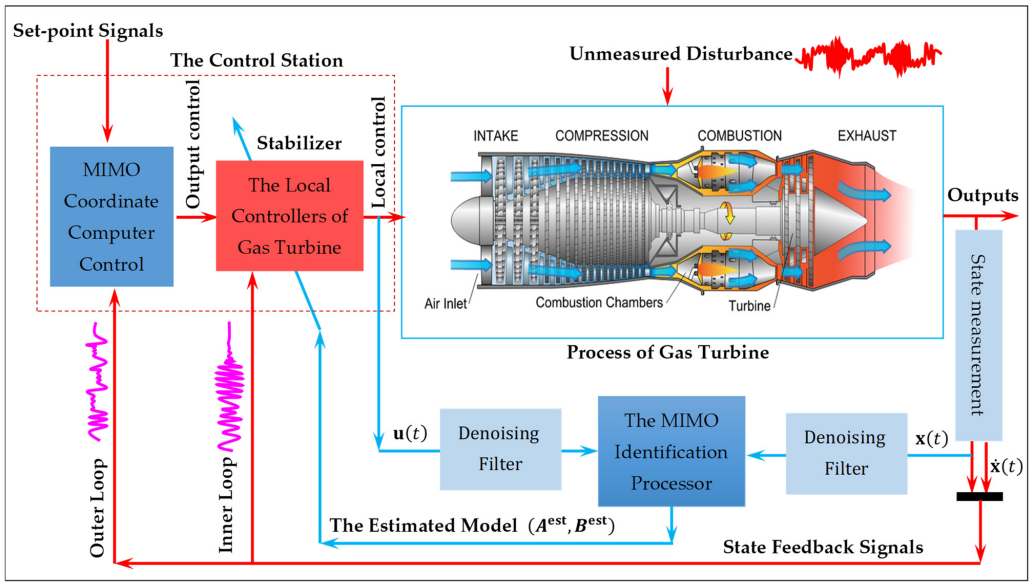


Figure 4. Supervisory control scheme incorporating reference adjustment.

In recent years, various control strategies have been proposed for multivariable gas turbine systems, involving trade-offs between performance, robustness, and computational complexity. Robust and adaptive methods such as H∞ and sliding mode control effectively handle uncertainties and disturbances but often increase controller complexity and tuning sensitivity [29,30]. Model Predictive Control (MPC) provides systematic constraint handling and multi-objective optimization, yet its computational burden can limit real-time implementation for fast dynamics [32]. Matrix-based and linear algebraic approaches, including generalized eigenstructure assignment and block polynomial methods [27,36], are closely related to the present work through closed-loop pole assignment

and structured state feedback, but typically assume full state availability or suffer from higher algebraic complexity in MIMO extensions. Observer-based schemes address state unavailability at the cost of added dynamics and noise sensitivity [37,38].

Figure 5 illustrates the dynamic response of the system variables and control inputs. Subplots (a) and (b) show the rotor speed and exhaust gas temperature, respectively, comparing the reference values (setpoints), actual values (process variables), and measured values obtained from sensors. Subplots (c) and (d) present the filtered control inputs alongside the actual control signals, demonstrating the effectiveness of the control strategy in tracking the references and maintaining system stability.

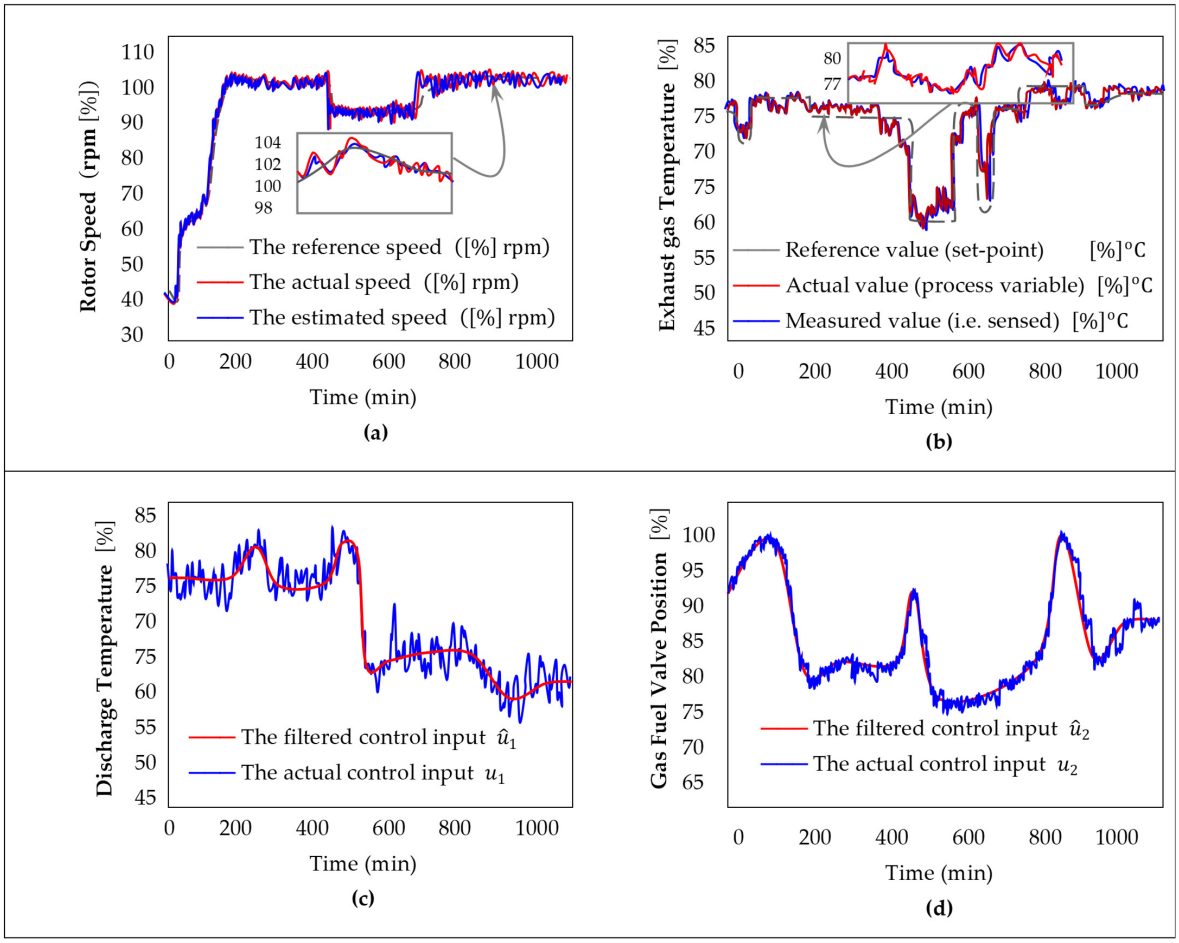


Figure 5. Responses and control inputs, showing reference, actual, and measured values.

The proposed state-derivative feedback leverages the time derivatives of key mechanical and thermal states to enhance closed-loop performance. By incorporating derivative information, the controller improves damping, reduces overshoot, and accelerates transient response, particularly for rotor speed and shaft torque, which exhibit faster dynamics. In practical implementations, since direct measurement of state derivatives is often infeasible due to sensor limitations and noise sensitivity, the proposed approach relies on estimated derivatives obtained through adaptive filtering techniques that ensure accurate estimation while attenuating measurement noise. Compared to conventional state-feedback approaches, this method achieves superior stability and robustness while maintaining a low computational burden, making it well-suited for real-time control of industrial gas turbines. To demonstrate its effectiveness and advantages, a comparative study (Table 3) is conducted with classical and recent control strategies, including classical pole placement with stable eigenvalue assignment [39], discrete time linear quadratic regulator

(DLQR) based on minimizing energy criteria [35], eigenstructure assignment with separate placement of eigenvalues and eigenvectors [36], stabilizing linear multivariable gas turbine model via sliding mode control [34], and the proposed linear algebraic state and state-derivative feedback method based on matrix polynomial theory, which simplifies implementation by directly leveraging measurable derivatives and ensures competitive pole placement capabilities.

Table 3. The norms (1, 2, inf) of the feedback gain matrix K associated with each method.

Method	$\|K\|_1$	$\|K\|_2$	$\|K\|_\infty$
Pole placement [39]	35.120	43.500	80.900
DLQR [35]	21.850	30.4674	77.2140
Eigen-S Assignment [36]	37.1325	45.7253	94.2423
Linear Sliding Mode [34]	65.0156	77.6217	114.1543
Proposed Method	22.4150	31.4213	75.8065

The table compares the feedback gain norms ($\|\boldsymbol{K}\|_1$,$\|\boldsymbol{K}\|_2$,$\|\boldsymbol{K}\|_\infty$) for different control methods applied to the gas turbine system. The proposed method and DLQR exhibit relatively low gain magnitudes, indicating moderate control effort and less aggressive feedback, while linear sliding mode shows the highest norms, reflecting stronger control action. Classical pole placement and Eigen-Structure Assignment lie in between, with Eigen-S having slightly higher norms due to the complexity of eigenvector shaping. Overall, the proposed state-derivative feedback achieves a good balance between performance and control effort.

Let λ_i, $\mathbf{v}_i$ and $\boldsymbol{t}_i$ be the i^{th} eigenvalue, right and left eigenvectors of $\boldsymbol{A}$, respectively, and let $\lambda_i + \Delta\lambda_i$ be the i^{th} eigenvalue of $\boldsymbol{A} + \Delta\boldsymbol{A}$ (i = 1, 2, …, n). Then, the eigenvalue sensitivity is defined as $s(\lambda_i) = \|\boldsymbol{t}_i\| \cdot \|\mathbf{v}_i\| \geq \Delta\lambda_i / \|\Delta\boldsymbol{A}\|$. The individual relative change in eigenvalues is $r_i = r(\lambda_i) = |\Delta\lambda_i| / |\lambda_i|$ for i = 1, 2, …, n. Let $\{\lambda_i\}_{i=1}^{n}$ be the set of eigenvalues of an $n \times n$ matrix denoted by $\boldsymbol{A}$ and assuming that all the eigenvalues are stable ($\mathfrak{Re}(\lambda_i) < 0$, $\forall i$) and have been arbitrarily assigned to guarantee performance. The three robust stability measures are defined by the following: $M_1 = \min_{\omega \in [0\ \infty[} \{\sigma(\boldsymbol{A} - j\omega\boldsymbol{I})\}$ where σ denotes the smallest singular value, and $M_2 = \min_{\omega \in [0\ \infty[} \left\{\kappa^{-1}(\boldsymbol{\Lambda}) \cdot \mathfrak{Re}(\lambda_n)\right\}$ in such a manner: $\mathfrak{Re}(\lambda_n) < \cdots < \mathfrak{Re}(\lambda_1)$, the matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times n}$ is the diagonal form of $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, and $M_3 = \min_{i \in [1\ n]} \left\{s^{-1}(\lambda_i) \cdot |\mathfrak{Re}(\lambda_i)|\right\}$ [7].

The sensitivity of the closed-loop eigenvalues with respect to perturbations in the system matrix is presented in Table 4, where the proposed method consistently exhibits lower sensitivity values compared to the other methods.

Table 4. Robust stability (eigenvalues sensitivity $s(\lambda_i)$).

Method	$s(\lambda_1)$	$s(\lambda_2)$	$s(\lambda_3)$	$s(\lambda_4)$	$s(\lambda_5)$	$s(\lambda_6)$
Pole Placement [39]	213.0689	213.0689	129.6670	95.5432	524.5128	524.5128
Discrete-LQR [35]	31.9535	262.7456	262.7456	1.8277×10^3	1.835×10^3	54.6149
Eigen-Structure Assignment [36]	28.1816	201.5546	118.1315	92.1314	403.6781	70.1324
Linear Sliding Mode [34]	27.3356	197.5478	115.0456	90.4378	450.6071	70.8843
Proposed Method	12.7286	91.0052	91.0052	28.4540	81.4513	83.3787

As shown in Table 5, the proposed method exhibits the smallest relative eigenvalue variations under perturbations while achieving the largest stability margins M_1, M_2, and M_3, confirming its superior robustness performance.

Table 5. Robust performance (relative change r in the eigenvalues λ_i under perturbation).

Method	Relative Change in λ_i Under Perturbation						Stability Measures		
	$r(\lambda_1)$	$r(\lambda_2)$	$r(\lambda_3)$	$r(\lambda_4)$	$r(\lambda_5)$	$r(\lambda_6)$	M_1	M_2	M_3
P-placement [39]	1.303	1.201	1.657	0.850	1.317	1.606	0.2725×10^{-5}	0.1337×10^{-5}	0.1438×10^{-5}
D-LQR [35]	8.12×10^{-4}	0.018	0.019	0.115	0.116	0.091	0.8220×10^{-4}	0.3989×10^{-3}	0.4289×10^{-4}
ES Method [36]	8.01×10^{-4}	0.011	0.017	0.100	0.124	0.082	0.2316×10^{-3}	0.1124×10^{-3}	0.1208×10^{-3}
LS Mode [34]	7.953×10^{-4}	0.011	0.098	0.059	0.069	0.069	0.2659×10^{-3}	0.1290×10^{-3}	0.1387×10^{-3}
Proposed	7.526×10^{-4}	0.003	0.003	0.002	0.010	0.002	0.230×10^{-2}	0.1116×10^{-2}	0.1200×10^{-2}

From Tables 4 and 5, it is observed that the proposed method yields significantly lower eigenvalue sensitivities $s(\lambda_i)$ and smaller relative variations $r(\lambda_i)$ under perturbations compared with the classical and recent control strategies. Moreover, the associated stability measures M_1–M_3 attain their highest values for the proposed approach, indicating improved robustness margins and enhanced tolerance to modeling uncertainties. Table 6 compares the settling time (t_s) and maximum peak (M_p) of each state for different control methods.

Table 6. Time domain specifications (settling time t_s and the maximum peak M_p).

Method		x_1	x_2	x_3	x_4	x_5	x_6
Pole Placement [39]	t_s	60	60	62	62	62	62
	M_p	73.4124	143.6067	275.8774	480.0302	63.8776	245.0559
Discrete-LQR [35]	t_s	40	37	51	50	46	51
	M_p	32.3220	61.5960	35.9795	82.4200	36. 4478	72.6120
Eigen-Structure Assignment [36]	t_s	39	38	45	42	40	39
	M_p	28.4440	56.3750	30.9560	75.5260	32.654	65.4120
Linear Sliding Mode [34]	t_s	40	39	30	38	42	40
	M_p	27.4520	50.8980	28.7795	68.3250	31.4576	64.8120
Proposed Method	t_s	37	37	28	37	36	35
	M_p	15.7367	30.4302	8.6442	25.3969	11.3107	26.7737

The proposed method achieves the fastest settling times and the lowest overshoots across all states, indicating superior transient response and damping. While DLQR and ES-assignment also show moderate improvements over pole placement, linear sliding mode reduces overshoot for some states but does not consistently outperform the proposed approach. Overall, the results highlight the effectiveness of the proposed state-derivative feedback in improving both speed of response and peak suppression.

Table 7 summarizes a qualitative comparison of the considered control methods in terms of gain norm, time-domain specifications, robust stability, and robust performance.

Table 7. Qualitative evaluation of different control strategies for the gas turbine system.

		Pole Placement [39]	Discrete-LQR [35]	Eigen-Structure Assignment [36]	Linear Sliding Mode [34]	Proposed Method
Norm		Acceptable	Excellent	Poor	Poor	**Good**
Time specifications		Poor	Poor	Acceptable	Good	**Excellent**
Robust stability	$s(\lambda_i)$	Poor	Poor	Good	Acceptable	**Excellent**
	M_1	Poore	Poor	Acceptable	Acceptable	**Excellent**
	M_2	Poor	Acceptable	Acceptable	Acceptable	**Excellent**
	M_3	Poor	Poor	Acceptable	Acceptable	**Excellent**
Robust performance $r(\lambda_i)$		Poor	Good	Good	Good	**Excellent**

The results of the feedback gain matrix $\boldsymbol{K}$, obtained by applying all the steps of the proposed state-derivative feedback algorithm, are compared with other control methods, including pole placement, DLQR, Eigen-Structure Assignment, and linear sliding mode.

The comparison, summarized in Table 7, is performed under the same conditions, including identical initial state values, desired pole locations, and a consistent level of injected disturbances. The qualitative evaluation considers key criteria such as norms, time specifications, robust stability measures ($s(\lambda_i)$, M_1, M_2, M_3), and robust performance $r(\lambda_i)$, highlighting the relative advantages of each method.

4.4. Interpretations of the Obtained Results

- Table 3 presents the amplitudes of the feedback gain matrix $\boldsymbol{K}$ computed for the first, second, and ∞-norms using the four reference methods and the proposed state-derivative feedback method. It is evident that the proposed method and DLQR yield the lowest gain norms, indicating moderate control effort. However, DLQR shows inferior performance in terms of other characteristics. Therefore, the $\boldsymbol{K}$ matrix obtained via the proposed method provides high performance with minimal control energy.
- The eigenvalue sensitivity results, following perturbations $\Delta\boldsymbol{A}$ in the dynamic matrix $\boldsymbol{A}$, are reported in the tables above (Table 4). The proposed method demonstrates greater robustness to internal dynamic changes compared to the other methods.
- The proposed method exhibits minimal relative change in eigenvalues, whereas other algorithms show significant variations that could potentially compromise system stability. Based on the defined stability measures and the obtained results (Table 5), the proposed algorithm consistently achieves the highest values for M_1, M_2 and M_3, particularly for M_1, and M_3, indicating superior stability performance.
- Table 6 summarizes the time-domain specifications, including settling time (t_s) and maximum peak (M_p) for all states. The proposed method achieves the fastest settling times and lowest overshoots, which can be attributed to the diagonal assignment of robust block roots derived from the speed eigenvalues and eigenvectors.
- Overall, the comprehensive comparison presented in Table 7 confirms that the proposed state-derivative feedback algorithm ensures the best combination of robust stability, transient performance, and eigenvalue sensitivity among all methods considered in this study.

4.5. Comments and Perspectives

- A necessary condition for the proposed method to have block-root and block-vector assignment is the block controllability; if it is satisfied, then we may use the artificial control proposed by Malika Yaici [8].
- In this work, the latent structure was arbitrarily chosen so that the block roots to be placed are arbitrary; hence, to meet specified objectives, some criterion should be explored to give the best desired block roots in order to improve the performance of the closed-loop system behavior.
- In some systems, the full state is not available, and then one may think to explore and develop the problem of the block-structure assignment via state feedback to block-structure assignment via output feedback or feedback compensator design.
- Latent structure assignment, which is the (block-roots, block-vectors) placement, provides a large degree of freedom in the design of the feedback gain matrix because the latent structure is more general and alters both the stability and the transient responses.
- Many research works have been conducted on the specification of eigenvalues and eigenvectors for multivariable systems to achieve certain desired behavior or performance of the system, but there is no known latent structure specifications research using state and state derivative feedback via linear algebra theory and matrix polynomials.

- Future research will focus on incorporating recent advances in gas turbine component design and manufacturing into control-oriented dynamic models. In particular, developments in integral blade rotor manufacturing, additive-manufactured burners, and topology-optimized centrifugal impellers demonstrate how modern fabrication techniques can significantly affect structural, thermal, and dynamic characteristics of gas turbines [40–46]. Accounting for these effects will enable the extension of the proposed linear algebra-based state-derivative feedback control framework to next-generation gas turbines, improving robustness and performance.
- Future work should focus on developing and evaluating alternative system identification methods [47–51] to enhance the fidelity of control-oriented models. Particular attention is needed to address structural constraints within the modeling framework and to ensure robustness under limited excitation and measurement uncertainty. Improvements in model accuracy under practical operating conditions, including nonlinearities and parameter variations, are essential for reliable controller design and predictive performance.
- Extending the experimental validation of state-derivative feedback remains a key area for further investigation [52–55]. This includes accurate derivative estimation under noisy measurements, assessment of actuator limitations and delays, and comprehensive evaluation of robustness beyond classical eigenvalue-based criteria. Developing methodologies that ensure performance in the presence of sensor noise, disturbances, and real-world operational constraints will be critical for transitioning these control strategies from laboratory to industrial applications.
- The scalability of the proposed control framework to large-scale gas turbine systems warrants further exploration [56–62]. Future studies should assess computational efficiency and feasibility for high-dimensional models and complex real-time applications. In addition, extending the framework to modern gas turbines requires consideration of advanced manufacturing techniques, evolving structural dynamics, and digital-twin-based monitoring. Such efforts will facilitate the practical deployment of the methodology in industrial settings, enabling robust, efficient, and scalable control for next-generation gas turbine systems.

5. Conclusions

In this paper, a new state-derivative feedback control algorithm for a class of MIMO linear systems has been proposed, based on the assignment of block roots extracted from the dominant eigenvalues. The block roots are assigned diagonally and placed in the right block controllable form (right block Vandermonde matrix) of the transformed decomposed system using the feedback gain $\boldsymbol{K}$. To validate the proposed method, the GE MS5001P gas turbine was selected due to its industrial importance, particularly in electrical power generation. The algorithm was applied to a real turbine model obtained from on-site experimental data, using parametric identification based on LMFD MIMO least squares. The robustness and stability of the proposed method were verified through stability measures, which demonstrated its effectiveness and validity. A comparative study with classical and recent control strategies clearly showed that the proposed algorithm provides superior performance, enhanced robustness, and improved transient response, highlighting the direct impact of block-root assignment on turbine operation. Consequently, this approach can be considered a promising solution for stability control in various industrial applications. The main limitations of the proposed algorithm are as follows:

- The ratio n/m must be an integer; otherwise, other techniques can be employed.
- The block controllability matrix must have full rank; otherwise, the algorithm is not applicable. Model order reduction can be used.

Future research can focus on optimal block root selection to reduce the norm of the feedback matrix $\boldsymbol{K}$ and minimize control energy. Additionally, the development of multivariable optimal static output-feedback controllers based solely on available outputs represents a promising direction for extending this work.

Author Contributions: B.B.: Conceptualization, methodology, software, formal analysis, investigation, data curation, original draft writing, visualization, and project administration; K.H. and A.G.: Supervision, validation, methodology, and review and editing of the manuscript; A.-N.S.: Visualization, formal analysis, methodology, data curation, validation, overall supervision, funding acquisition, project coordination, draft writing and critical manuscript revision. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DLQR	Discrete time linear quadratic regulator
ESA	Eigen-Structure Assignment
GCV	Gas control valve
LMFD	Left matrix fraction description
LQR	Linear quadratic regulator
LTR	Loop Transfer Recovery
MFD	Matrix fraction description
MIMO	Multi-Input Multi-Output
MPC	Model Predictive Control
RMFD	Right matrix fraction description
TCD	Temperature of the Compressor Discharge
PCD	Pressure of the Compressor Discharge

Appendix A

This appendix provides supplementary data supporting the numerical results obtained from the application of the state-derivative feedback controller, with particular emphasis on the computational platform used for optimization and parallel simulations. All simulations were performed on a high-performance server equipped with an Intel® Xeon® E5-2630 v2 processor operating at 2.60 GHz, featuring 12 physical cores and 24 logical processors for efficient parallel execution. The system includes 16 GB of RAM and a 15 MB shared CPU cache, ensuring adequate memory bandwidth and low-latency data access for large-scale numerical optimization. The 22 nm architecture and high-speed 7.2 GT/s QPI interconnect provide stable performance and scalability, with support for up to 768 GB of DDR3 memory for future expansion. Parallel computing tools such as *parpool*, *parfor*, *spmd*, and distributed arrays were employed to accelerate optimization routines and simulation workloads, enabling reliable evaluation of the proposed control strategy.



Processor Intel(R) Xeon(R) E5-2630 v2 @ 2.60 GHz	
Essential information and performance	
Vertical Segment	Server
Number of Cores	12
Number of CPUs	24
Memory	16 GB
Memory CPU Cache	15 MB
Lithography	22 nm
Max Memory Size	768 GB
Memory Types	DDR3
Bus Speed	7.2 GT/s QPI
Fundamental parallel computing functions	
parpool:	Create the parallel pool
parfor:	Parallel for loop
spmd:	Execute code in parallel on workers
distributed:	Create distributed array from data

References

1. Wonham, W.M. On pole assignment in multi-input controllable linear systems. *IEEE Trans. Autom. Control* **1967**, *12*, 660–665. [CrossRef]
2. Moore, B.C. On the flexibility offered by state feedback in multivariable systems beyond closed loop eigenvalue assignment. *IEEE Trans. Autom. Control* **1976**, *21*, 689–692. [CrossRef]
3. Roppenecker, G. Minimum norm output feedback design under specified eigenvalue areas. *Syst. Control. Lett.* **1983**, *3*, 101–103. [CrossRef]
4. Fahmy, M.M.; O'Reilly, J. Eigenstructure Assignment in Linear Multivariable Systems-A Parametric Solution. *IEEE Trans. Autom. Control.* **1983**, *28*, 290–294. [CrossRef]
5. Andry, A.N.; Shapiro, E.Y.; Chung, J.C. Eigenstructure assignment for linear systems. *IEEE Trans. Aerosp. Electron. Syst.* **1983**, *19*, 711–729. [CrossRef]
6. Patel, R.V.; Misra, P. Numerical algorithms for eigenvalue assignment by state feedback. *Proc. IEEE* **1984**, *72*, 1755–1764. [CrossRef]
7. Bekhiti, B. Multivariable Control System Design Using the Theory of Matrix Polynomials. Ph.D. Thesis, Boumerdes University Algeria, Boumerdès, Algeria, 2018.
8. Yaici, M.; Hariche, K. On eigen-structure assignment using block poles placement. *Eur. J. Control.* **2014**, *20*, 217–226. [CrossRef]
9. Bekhiti, B.; Hariche, K.; Zaitsev, V.; Duan, G.R.; Sharkawy, A.-N. The Algebraic Theory of Operator Matrix Polynomials with Applications to Aero-elasticity in Flight Dynamics and Control. *Math. Comput. Appl.* **2025**, *30*, 131. [CrossRef]
10. Becher, D. Eigenvalue assignment by output-feedback with reduced sensor-actuator design. *Mech. Syst. Signal Process.* **2025**, *241*, 113372. [CrossRef]
11. Siahlooei, E.; Fazeli, S.A.S.; Karbassi, S.M. Eigenvalue assignment via uncertain state feedback controllers. *Asian J. Control* **2019**, *22*, 1860–1869. [CrossRef]
12. Duan, G.R. Solutions of the equation AV + BW = V F and their application to eigenstructure assignment in linear systems. *IEEE Trans. Aut. Control* **1993**, *38*, 276–280. [CrossRef]
13. Sobel, K.M.; Shapiro, E.Y.; Andry, A.N. Eigenstructure Assignment. *Int. J. Control* **1994**, *59*, 13–37. [CrossRef]
14. Liu, G.P.; Patton, R.J. Robust control design using eigenstructure assignment and multi-objective optimization. *Int. J. Syst. Sci.* **1996**, *27*, 871–879. [CrossRef]
15. Bekhiti, B.; Al-Sabur, R.; Roudane, M.; Younis, J.A.; Sharkawy, A.-N. Intelligent neuro-fuzzy adaptive MIMO control for a self-balancing two wheeled autonomous robot via recursive resolution of the matrix Diophantine equation. *Discov. Robot.* **2025**, *1*, 11. [CrossRef]
16. Liu, G.P.; Patton, R.J. *Eigenstructure Assignment for Control System Theory*; John Wiley & Sons: Hoboken, NJ, USA, 1998; pp. 55–122.

17. Patton, R.J.; Liu, G.P. Robust control design via eigenstructure assignment, genetic algorithms and gradient-based optimization. *IEEE Proc. Control. Theory Appl.* **1994**, *141*, 202–208. [CrossRef]
18. Bekhiti, B.; Fragulis, G.F.; Maraslidis, G.S.; Hariche, K.; Cherifi, K. A Novel Recursive Algorithm for Inverting Matrix Polynomials via a Generalized Leverrier–Faddeev Scheme: Application to FEM Modeling of Wing Vibrations in a 4th-Generation Fighter Aircraft. *Mathematics* **2025**, *13*, 2101. [CrossRef]
19. Shieh, L.S.; Tsay, Y.T.; Coleman, N.I. Algorithms for solvents and spectral factors of matrix polynomials. *Int. J. Control* **1981**, *12*, 1303–1316. [CrossRef]
20. Bachelier, O.; Mehdi, D. Non-iterative pole placement technique: A step further. *J. Frankl. Inst.* **2008**, *345*, 267–281. [CrossRef]
21. Karbassi, S.M.; Saadatjou, F. A parametric approach for eigenvalue assignment by static output feedback. *J. Frankl. Inst.* **2009**, *346*, 289–300. [CrossRef]
22. Bekhiti, B.; Hariche, K.; Kouzou, A.; Younis, J.A.; Sharkawy, A.-N. On Matrix Linear Diophantine Equation-Based Digital-Adaptive Block Pole Placement Control for Multivariable Large-Scale Linear Process. *AppliedMath* **2025**, *5*, 139. [CrossRef]
23. Pereira, E. On solvents of matrix polynomials. *Appl. Numer. Math.* **2003**, *47*, 197–208. [CrossRef]
24. Bekhiti, B.; Hariche, K.; Iqbal, J.; Sharkawy, A.N. Parametric Eigen-space Characterization via Static Output Feedback in MIMO Control Systems: A New Look to Output Feedback Gain Matrix Design for Turbo-Generator System. *Int. J. Robot. Control. Syst.* **2025**, *5*, 2546–2567. [CrossRef]
25. Periera, E. Block eigenvalues and solution of diff-matrix equation. *Math. Notes* **2003**, *4*, 45–51. [CrossRef]
26. Zhang, B. Parametric Eigenstructure Assignment for Descriptor Systems via Proportional plus Derivative State Feedback. *Int. J. Control. Autom. Syst.* **2011**, *9*, 15–22. [CrossRef]
27. Bekhiti, B.; Hariche, K.; Fragulis, G.F.; Sharkawy, A.-N. Matrix Fraction Description in Large Scale MIMO Descriptor Systems: Matrix Polynomials Approaches. *Symmetry* **2025**, *17*, 1413. [CrossRef]
28. Bagua, H.; Hafaifa, A.; Iratni, A.; Guemana, M. Model variables identification of a gas turbine using a subspace approach based on input/output data measurements. *Control. Theory Technol.* **2021**, *19*, 183–196. [CrossRef]
29. Bianco, S.J. Sliding Mode Transient Scaling Controller for Gas Turbine Engine Emulation on an Electric Machine. In Proceedings of the AIAA 2020-3677. Session: Software Systems and Modeling for Electrified Propulsion, Virtual, 24–28 August 2020. [CrossRef]
30. Biss, D.; Woodgate, K.G. Gas Turbine Control Using Mixed Sensitivity H∞-Optimisation. In *Robust Control of Linear Systems and Nonlinear Control: Progress in Systems and Control Theory*; Kaashoek, M.A., van Schuppen, J.H., Ran, A.C.M., Eds.; Birkhäuser: Boston, MA, USA, 1990; Volume 4. [CrossRef]
31. Bonfiglio, A.; Cacciacarne, S.; Invernizzi, M.; Lanzarotto, D.; Palmieri, A.; Procopio, R. A Sliding Mode Control Approach for Gas Turbine Power Generators. *IEEE Trans. Energy Convers.* **2019**, *34*, 921–932. [CrossRef]
32. Hou, G.; Gong, L.; Yang, Z.; Zhang, J. Multi-objective economic model predictive control for gas turbine system based on quantum simultaneous whale optimization algorithm. *Energy Convers. Manag.* **2020**, *207*, 112498. [CrossRef]
33. Palmieri, A.; Lanzarotto, D.; Cacciacarne, S.; Torre, I.; Bonfiglio, A. An Innovative Sliding Mode Load Controller for Gas Turbine Power Generators: Design and Experimental Validation Via Real-Time Simulation. *Energy* **2021**, *217*, 119363. [CrossRef]
34. Nail, B.; Kouzou, A.; Hafaifa, A.; Bekhiti, B. Stabilizing linear multivariable gas turbine model via sliding mode control. *Fundam. Sci. Appl.* **2016**, *22*, 25–30.
35. Shabaninia, F.; Jafari, K. Using LQG/LTR Optimal Control Method to Improve Stability and Performance of Industrial Gas Turbine System. *Int. Sch. Res. Netw. ISRN Electron.* **2012**, *2012*, 134580. [CrossRef]
36. Abdellah, K. Block pole assignment stabilization in large scale systems: Application to gas turbine power plant. In Proceedings of the International Conference on Technological Advances in Electrical Engineering ICTAEE'16, Skikda, Algeria, 24–26 October 2016.
37. Simani, S.; Spina, P.R. Kalman Filtering to Enhance the Gas Turbine Control Sensor Fault Detection. *Theory Pract. Control. Syst.* **1999**, 443–450. [CrossRef]
38. Ahmed, J.M. Optimal tuning linear quadratic regulator for gas turbine by genetic algorithm using integral time absolute error. *Int. J. Electr. Comput. Eng.* **2020**, *10*, 1367. [CrossRef]
39. Sarjaš, A.; Chowdhury, A.; Svečko, R. Robust Optimal Regional Closed-loop Pole Assignment over Positivity Conditions and Differential Evolution. *IFAC PapersOnLine* **2015**, *48*, 141–146. [CrossRef]
40. González-Barrio, H.; Calleja-Ochoa, A.; Lamikiz, A.; de Lacalle, L.N.L. Manufacturing Processes of Integral Blade Rotors for Turbomachinery, Processes and New Approaches. *Appl. Sci.* **2020**, *10*, 3063. [CrossRef]
41. Cascino, A.; Meli, E.; Rindi, A.; Pucci, E.; Matoni, E. Experimental Validation and Dynamic Analysis of Additive Manufacturing Burner for Gas Turbine Applications. *Machines* **2025**, *13*, 1111. [CrossRef]
42. Liu, S.; Mi, D.; Wen, C.; Zheng, Y. Topology Optimization Design With Addictive Manufacturing Constraints for Centrifugal Impeller. *J. Phys. Conf. Ser.* **2022**, *2252*, 012026. [CrossRef]

43. Adamou, A.; Turner, J.; Costall, A.; Jones, A.; Copeland, C. Design, simulation, and validation of additively manufactured high-temperature combustion chambers for micro gas turbines. *Energy Convers. Manag.* **2021**, *248*, 114805. [CrossRef]
44. Zajadatz, M.; Güthe, F.; Freitag, E.; Ferreira-Providakis, T.; Wind, T.; Magni, F.; Goldmeer, J. Extended Range of Fuel Capability for GT13E2 AEV Burner With Liquid and Gaseous Fuels. *J. Eng. Gas Turbines Power* **2018**, *141*, 051017. [CrossRef]
45. Remache, A.; Pérez-Sánchez, M.; Hidalgo, V.H.; Ramos, H.M. Hybrid Optimization Approaches for Impeller Design in Turbomachinery: Methods, Metrics, and Design Strategies. *Water* **2025**, *17*, 1976. [CrossRef]
46. Meli, E.; Furferi, R.; Rind, A.; Ridolfi, A.; Volpe, Y.; Buonamici, F. A General Framework for Designing 3D Impellers Using Topology Optimization and Additive Manufacturing. *IEEE Access* **2020**, *8*, 60259–60269. [CrossRef]
47. Yang, R.; Liu, Y.; He, X.; Liu, Z. Gas Turbine Model Identification Based on Online Sequential Regularization Extreme Learning Machine with a Forgetting Factor. *Energies* **2023**, *16*, 304. [CrossRef]
48. Villarreal-Valderrama, F.; Liceaga-Castro, E.; Hernandez-Alcantara, D.; Santana-Delgado, C.; Ekici, S.; Amezquita-Brooks, L. Control-oriented system identification of turbojet dynamics. *Aerospace* **2024**, *11*, 630. [CrossRef]
49. Sadjadi, E.N.; Garcia, J.; Lopez, J.M.M.; Borzabadi, A.H.; Abchouyeh, M.A. Fuzzy Model Identification and Self Learning with Smooth Compositions. *Int. J. Fuzzy Syst.* **2019**, *21*, 2679–2693. [CrossRef]
50. Wang, Q.-G.; Zhang, L. System Identification in the Network Era: A Survey of Data Issues and Innovative Approaches. *IEEE CAA J. Autom. Sin.* **2025**, *12*, 1305–1319. [CrossRef]
51. Ma, S.; Wu, Y.; Hua, Z.; Gou, L. Application of Fuzzy Inference System in Gas Turbine Engine Fault Diagnosis Against Measurement Uncertainties. *Chin. J. Mech. Eng.* **2025**, *38*, 2. [CrossRef]
52. Wen, D.; Sun, C.; Huang, S.; Yi, S. Robust fault estimation and proportional derivative fault tolerant control for a class of singular systems with interval time-varying delay and disturbance. *Optim. Control. Appl. Methods* **2024**, *45*, 928–953. [CrossRef]
53. Sariyildiz, E.; Yu, H.; Ohnishi, K. A Practical Tuning Method for the Robust PID Controller with Velocity Feed-Back. *Machines* **2015**, *3*, 208–222. [CrossRef]
54. Yu, P.; Wan, H.; Zhang, B.; Wu, Q.; Zhao, B.; Xu, C.; Yang, S. Review on System Identification, Control, and Optimization Based on Artificial Intelligence. *Mathematics* **2025**, *13*, 952. [CrossRef]
55. Yessef, M.; Benbouhenni, H.; Lagrioui, A.; Bizon, N.; Bossoufi, B.; Al-Gahtani, S.F.; Elbarbary, Z.M.S. Processor-in-the-Loop validation of direct power control based on fractional-order modified super-twisting algorithm for doubly-fed induction generators. *Sci. Rep.* **2025**, *15*, 22846. [CrossRef]
56. Aquize, R.; Cajahuaringa, A.; Machuca, J.; Mauricio, D.; Villanueva, J.M.M. System Identification Methodology of a Gas Turbine Based on Artificial Recurrent Neural Networks. *Sensors* **2023**, *23*, 2231. [CrossRef]
57. Wang, R.; Hu, J.; Xin, D.; Liu, S.; Zhao, K. Robust subspace tracking in intelligent fault diagnosis of digital twin gas turbines base on the adaptive Markov transfer. *Appl. Energy* **2025**, *401*, 126747. [CrossRef]
58. Pedroso, L.; Batista, P.; Heemels, W. Distributed design of ultra-large-scale control systems: Progress, Challenges, and Prospects. *Annu. Rev. Control.* **2025**, *59*, 100987. [CrossRef]
59. Farhat, H.; Altarawneh, A. Physics-Informed Machine Learning for Intelligent Gas Turbine Digital Twins: A Review. *Energies* **2025**, *18*, 5523. [CrossRef]
60. Bruni, G.; Maleki, S.; Krishnababu, S.K. Deep learning modeling of manufacturing and build variations on multistage axial compressors aerodynamics. *Data Centric Eng.* **2025**, *6*, e9. [CrossRef]
61. Zhang, J.; Wang, Z.; Li, S.; Wei, P. A digital twin approach for gas turbine performance based on deep multi-model fusion. *Appl. Therm. Eng.* **2024**, *246*, 122954. [CrossRef]
62. Lim, J.T.; Habibullah, A.; Ng, E.Y.K. Towards a Digital Twin for Gas Turbines: Thermodynamic Modeling, Parameter Estimation, and Performance Optimization using PINN and PSO. *Energies* **2025**, *18*, 3721. [CrossRef]

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.