

Article

# A Linear Iterative Controller for Software Defined Control Systems of Aero-Engines Based on LMI

Xiaoxiang Ji, Jiao Ren, Jianghong Li \* and Yafeng Wu

School of Power and Energy, Northwestern Polytechnical University, Xi'an 710072, China; jixiaoxiang@mail.nwpu.edu.cn (X.J.); renjiao@mail.nwpu.edu.cn (J.R.); yfwu@nwpu.edu.cn (Y.W.)  
\* Correspondence: jhli@nwpu.edu.cn

**Abstract:** Currently, most control systems of the aero-engines possess a central controller. The core tasks for the control system, such as control law calculations, are executed in this central controller, and its performance and reliability greatly impact the entire control system. This paper introduces a control system design named Software Defined Control Systems (SDCS), which features a controller-decentralized architecture. In SDCS, a network composed of a set of nodes serves as the controller, so there is no central controller in the system, and computations are distributed throughout the entire network. Since the controller is decentralized, there is a need for decentralized control tasks. To address this, this paper introduces a method for designing decentralized control tasks using periodic linear iteration. Each node in the network periodically broadcasts its own state and updates its next-step state as a weighted sum of its current state and the received current states of other nodes in the network. Each node in the network acts as a linear dynamic controller and maintains an internal state through information exchange with other nodes. We modeled the decentralized controller and obtained the model of the entire control system, and the workload of each obtained decentralized control task is balanced. Then, we obtained a parameter tuning method for each decentralized controller node based on Linear Matrix Inequalities (LMI) to stabilize the closed-loop system. Finally, the effectiveness of the proposed method was verified through digital simulation.

**Keywords:** decentralized controller; linear iterative; aero-engines; dynamic controller; Linear Matrix Inequalities (LMI)



**Citation:** Ji, X.; Ren, J.; Li, J.; Wu, Y. A Linear Iterative Controller for Software Defined Control Systems of Aero-Engines Based on LMI. *Actuators* **2023**, *12*, 259. <https://doi.org/10.3390/act12070259>

Academic Editors: Ti Chen, Junjie Kang, Shidong Xu and Shuo Zhang

Received: 28 May 2023  
Revised: 17 June 2023  
Accepted: 20 June 2023  
Published: 23 June 2023



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

## 1. Introduction

The traditional aero-engine control system uses a centralized control scheme, and the controller is connected to the analog sensors and actuator through cables and connectors. The controller executes most of the tasks of the control system, such as control law calculation, digital-to-analog conversion (D/A), analog-to-digital conversion (A/D), and so on, and the workload of the controller is large. Meanwhile, the centralized control scheme increases the difficulty and cost of upgrading and maintaining aero-engines throughout their entire life cycle. In addition, the presence of rotating components results in a large amount of cables within the system, which reduces the thrust-to-weight ratio of the aero-engines. To overcome the drawbacks of centralized control schemes, researchers have introduced the design concept of distributed control into the design of aero-engine control systems [1–5].

The Distributed Control System (DCS) first appeared in industrial process control [6,7], and it has been widely adopted in many fields [8–10]. The core design concept of the distributed control system for aero-engines is functional decentralization, which means that some functions of a centralized control system, such as signal conditioning and state monitoring, are executed by intelligent sensors and actuators, while the central controller focuses on the core tasks of the control system, such as control law calculation [11]. In a DCS, the controller, intelligent sensors, and intelligent actuators are connected through a common and standardized digital bus [12,13]. Compared to centralized control, DCS has

the following advantages: (1) digital intelligent sensors and actuators reduce the control tasks of the central controller. (2) The digital bus reduces the weight of the aero-engine control system. (3) Modular design solutions lower the difficulty and cost of system upgrades and maintenance processes.

In addition to the above advantages, DCS also has some disadvantages:

1. As the performance of aircraft and aero-engines gradually improve, the functionality and complexity of central control tasks also rapidly increase, which requires high-performance, multi-core microprocessors as controllers, and it places high demands on the thermal management system of the aviation engine.
2. Due to the increasing functionality and complexity of core control tasks, the amount of software code in the control system rapidly increases, reducing the software reliability of the control system.
3. Control tasks are centralized on the central controller. The central controller determines the performance of the aero-engine's control system, and its damage or failure have a significant impact on the control system.

The analysis of the DCS for aero-engines mentioned above shows that the drawbacks in the system are caused by the central controller present in the system. The drawbacks of DCS suggest that a feasible scheme is to make the controllers in the control system decentralized. This involves using a network of decentralized nodes as the system controller, where multiple microprocessors work together to execute the control tasks of the central controller of DCS. As a result, the workload of each decentralized node is significantly reduced. With the decrease in workload, the software code volume for each node also decreases, thereby improving the software reliability of the control system. Additionally, several low-performance, but highly reliable microprocessors, can be used as decentralized controllers, which enhances the reliability of individual nodes.

The origin of the decentralized control design scheme dates back to the 1970s, and its concept was mainly proposed to solve the control problem of large-scale systems. Large-scale systems are characterized by extensive spatial distribution, numerous external signals, or dynamic changes in control structure. The core problem is that the control system is too large, and the control problem is too complex [14]. It is generally difficult to achieve control objectives by using more powerful microprocessors and larger storage space for large-scale systems. Due to the characteristics of large-scale systems, it is usually necessary to analyze and process the controlled plants, divide the system control problems into independent sub-problems, and handle signal delays caused by large spaces [15]. Distributed control in the industrial process control field adopts this design concept. This includes the later network control systems (NCS) and wireless networked control systems (WNCS). In WNCS, wireless communication is used to transmit data between system components. Compared with wired communication, wireless communication has its advantages and disadvantages. Its advantages include flexibility in installation, easy configuration, and strong adaptability; disadvantages include communication delays, data packet loss, etc. Ref. [16] conducted a study on the decentralized control system for the autonomous guided vehicles (AGVs) path planning problem, proposing two decentralized control methods to solve the AGV control problem: task allocation for AGVs through consensus method and path planning coordination through decentralized control strategy. In this design scheme, the microprocessors of several vehicles form the controller for the AGV problem, specifically targeting path optimization in a large spatial range. However, for each vehicle, there still exists a central controller in its control system. Ref. [17] researched the decentralized control of automation systems, such as smart factories and smart cities. In addition to data related to control tasks, there is also data interaction, such as images and voices, resulting in high communication requirements for the system. Inspired by concepts, such as software-defined networking (SDN), a decentralized data interaction control method was designed for large space and large data flow systems through extending Lyapunov drift-plus-penalty (LDP) control to a new queuing system to adjust the data flow within the system. Ref. [18] studied the decentralized control problem of two coupled

power systems, including wind turbines and diesel generators. In this paper, the system was not decoupled and treated as a whole, and a decentralized controller was designed consisting of two PI-lead controllers. This scheme mainly focused on the research of PI controllers and parameter tuning, but the application of this scheme is limited for cases where the control effect of some PI controllers is poor. Ref. [19] proposed a decentralized control scheme for aero-engines, which replaced the central controller in traditional aero-engine control systems with multiple controllers to control different subsystems of the aero-engine. However, in this scheme, the controllers are fixedly mapped to the subsystems, and this means that there exists a central controller for each subsystem, and the control system structure is relatively inflexible. In the aforementioned studies, each decentralized controller can independently complete specific control tasks.

Ref. [20] proposed a controller decentralized design method, namely, the Software-Defined Control System (SDCS). In SDCS, the network composed of decentralized nodes serves as the controller to execute control tasks, such as control law execution. Therefore, the control tasks executed by the decentralized nodes are also decentralized, and each decentralized control task is a part of the core control tasks. The core control tasks are the sum of these decentralized control tasks in a specific way, no individual decentralized node can complete the system control task alone. The aero-engine is a safety-critical system that requires high reliability of the control system. When applying SDCS to the aero-engine control system, highly reliable but low-performance microprocessors should be used as decentralized nodes. This requires that the workload and software code volume of each decentralized control task are low, and complex control functions need to be executed.

The contributions of this paper are as follows. A linear iterative-based decentralized design scheme for control tasks was introduced. In this scheme, a network composed of several decentralized nodes act as the controller to undertake the system control task. Each decentralized node broadcasts its own state information to other decentralized nodes in each cycle period. Other nodes receive the broadcasted state information, and when all decentralized nodes have completed broadcasting their state information, all nodes update their own states as a weighted sum of their current state and the received states from other nodes. Therefore, each task node can act as a small dynamic controller. Through the linear iterative process, a model of the decentralized controller was constructed, which obtained the task executed on each decentralized node and the model of the entire control system. Due to the controller being located in the forward path and the linear iterative process introducing new internal state variables, conventional state feedback or output feedback design schemes cannot be used for decentralized controller node parameter tuning. Additionally, since the parameters with respect to the Lyapunov condition are nonlinear, conventional Lyapunov methods and LMI methods cannot be directly applied to parameter tuning of decentralized controller nodes. A parameter tuning method for decentralized controller nodes based on LMI was presented for the designed decentralized control task to ensure that the control system is Schur stable.

The remainder of this paper is organized as follows.

Section 2 describes a linear model for aero-engines. Section 3 presents the structure of the Software Defined Control System for aero-engines and a periodic linear iterative-based decentralized controller implementation scheme, along with control system modeling. Section 4 introduces a parameter tuning method for the controller based on LMI. Section 5 conducts simulation verification of the designed scheme. Section 6 is the final chapter of this paper, which summarizes and discusses the future outlook.

## 2. Aero-Engine Model

The mathematical model of aero-engines is crucial for the design of aero-engines control systems. As aero-engines are complex, time-varying, and strongly nonlinear systems, their mathematical models should be nonlinear [21–24]. The current controllers are designed based on linear systems for aero-engines [25,26]. The decentralized controller

proposed later in this paper is designed based on a periodic linear iterative scheme and is linear, hence requiring a linear model of the aero-engines.

The modeling methods for aero-engines generally include two types: identification method and analytical method. The identification method requires corresponding experimental conditions, has high costs, and some of its algorithms have large computational consumption. Moreover, the resulting model may lose certain characteristics of the engine. The analytical method is based on the principles of aero-engines to build models. Firstly, the components of the aero-engines are modeled; then, a series of nonlinear equations that describe the working process of the aero-engines by following the aerodynamic and thermodynamic principles during their operation are utilized to obtain the mathematical model of the aero-engines [27,28].

In this section, we analyze the case of a turbofan engine using the method described in Refs. [27,28]. This section first introduces the nonlinear model of turbofan engines and then describes the linearization method for the nonlinear model of turbofan engines.

### 2.1. Nonlinear Model of Turbofan Engines

In this sub-section, the models of various components of turbofan engines are presented, including the inlet, fan, compressor, combustor, turbine, bypass duct, mixer, and exhaust nozzle.

#### (1) Intake

When air enters the aero-engines, it first flows through the intake. The intake's inlet parameters are:

When  $H \leq 11$  km,

$$\begin{aligned} T_1 &= 288.15 - 0.0065H \\ P_1 &= 101,325 \times \left(1 - \frac{H}{44,331}\right)^{5.25588} \end{aligned} \quad (1)$$

When  $H > 11$  km,

$$\begin{aligned} T_1 &= 216.5 \\ P_1 &= 22,632 \times e^{\frac{11,000-H}{6342}} \end{aligned} \quad (2)$$

where;  $T_1$  and  $P_1$  are the total temperature and the total pressure at the inlet of the intake, respectively.

#### (2) Fan

The inlet parameters of aero-engines fan are:

$$\begin{aligned} T_2 &= T_1 \left(1 + \frac{k_1-1}{2} M_a^2\right) \\ P_2 &= \sigma_1 P_1 \left(1 + \frac{k_1-1}{2} M_a^2\right)^{\frac{k_1}{k_1-1}} \end{aligned} \quad (3)$$

where;  $T_2$  and  $P_2$  are the total temperature and the total pressure at the inlet of the fan, respectively,  $M_a$  is Mach number,  $\sigma_1$  is the total pressure recovery coefficient of the intake, and  $k_1$  is the air adiabatic index.

The outlet parameters of aero-engines fan are:

$$\begin{aligned} T_{21} &= T_2 \left(1 + \frac{\frac{k_F-1}{k_F} - 1}{\eta_F}\right) \\ P_{21} &= \pi_F P_2 \end{aligned} \quad (4)$$

where;  $T_{21}$  and  $P_{21}$  are the total temperature and the total pressure at the outlet of the fan, respectively,  $k_F$  is the air adiabatic index,  $\pi_F$  is the fan pressure ratio, and  $\eta_F$  is the efficiency of the fan.

## (3) Compressor

The inlet parameters of aero-engines compressor are:

$$\begin{aligned} T_{22} &= T_{21} \\ P_{22} &= \sigma_2 P_{21} \end{aligned} \quad (5)$$

where;  $T_{22}$  and  $P_{22}$  are the total temperature and the total pressure at the inlet of the compressor, respectively,  $\sigma_2$  is the total pressure recovery coefficient of the fan.

The outlet parameters of aero-engines compressor are:

$$\begin{aligned} T_3 &= T_{22} \left( 1 + \frac{\pi_C^{\frac{k_C-1}{k_C}} - 1}{\eta_C} \right) \\ P_3 &= \pi_C P_{22} \end{aligned} \quad (6)$$

where;  $T_3$  and  $P_3$  are the total temperature and the total pressure at the outlet of the compressor, respectively,  $\pi_C$  is the pressure ratio of the compressor,  $\eta_C$  is the efficiency of the compressor, and  $k_C$  is the air adiabatic index.

## (4) Combustion chamber

According to the energy conservation law, the simplified energy balance equation for the combustion chamber is:

$$W_{mf} H_u \eta_r + W_{ma} c_p T_3 = W_{ma} c_p T_4 \quad (7)$$

where;  $T_4$  is the temperature at the outlet of the combustion chamber,  $W_{mf}$  is the fuel flow,  $H_u$  is the calorific value of the fuel,  $\eta_r$  is the combustion efficiency,  $W_{ma}$  is the air flow at the inlet of the combustion chamber, and  $c_p$  is the specific heat capacity of air at constant pressure. The outlet pressure  $P_4$  of the combustion chamber can be calculated using the above equation.

$$P_4 = \sigma_3 P_3 \quad (8)$$

where;  $P_4$  is the pressure at the outlet of the combustion chamber, and  $\sigma_3$  is the total pressure recovery coefficient of the combustion chamber.

## (5) High-pressure turbine

The outlet parameters of aero-engines high-pressure turbine are:

$$\begin{aligned} T_{41} &= \frac{q_{mg,TH} T_4 + C_{HPTCool} q_{maC,total} T_{cool}}{q_{mg,TH,total}} \left( 1 - \left( 1 - \pi_{TH}^{\frac{1-k_{TH}}{k_{TH}}} \right) \eta_{TH} \right) \\ P_{41} &= \frac{P_4}{\pi_{TH}} \end{aligned} \quad (9)$$

where;  $T_{41}$  and  $P_{41}$  are the total temperature and the total pressure at the outlet of the high-pressure turbine, respectively,  $q_{mg,TH}$  is the gas flow of the high-pressure turbine,  $C_{HPTCool}$  is the proportion coefficient of the high-pressure compressor bleed air used to cool the high-pressure turbine,  $q_{maC,total} T_{cool}$  is the total flow of added air,  $q_{mg,TH,total}$  is the outlet air flow of the high-pressure turbine,  $\pi_{TH}$  is the high-pressure turbine pressure ratio,  $\eta_{TH}$  is the efficiency of the high-pressure turbine, and  $k_{TH}$  is the gas adiabatic index.

## (6) Low-pressure turbine

The outlet parameters of aero-engines low-pressure turbine are:

$$\begin{aligned} T_5 &= \frac{q_{mg,TL} T_{42} + C_{LPTCool} q_{maC,total} T_{cool}}{q_{mg,TL,total}} \left( 1 - \left( 1 - \pi_{TL}^{\frac{1-k_{TL}}{k_{TL}}} \right) \eta_{TL} \right) \\ P_5 &= \frac{P_{42}}{\pi_{TL}} \end{aligned} \quad (10)$$

where;  $T_5$  and  $P_5$  are the total temperature and the total pressure at the outlet of the low-pressure turbine, respectively,  $q_{mg,TL}$  is the gas flow converted by the high-pressure turbine, and  $T_{42}$  is the inlet temperature of the low-pressure turbine, which is approximately equals to the outlet temperature of the high-pressure turbine  $T_{41}$ .  $P_{42}$  is the inlet pressure of the low-pressure turbine, which is approximately equals to the outlet pressure of the high-pressure turbine  $P_{41}$ .  $C_{LPTCool}$  is the proportion coefficient of the high-pressure compressor bleed air used to cool the low-pressure turbine,  $q_{mg,TL,total}$  is the outlet air flow of the low-pressure turbine,  $\pi_{TL}$  is the low-pressure turbine pressure ratio,  $\eta_{TL}$  is the efficiency of the low-pressure turbine, and  $k_{TL}$  is the gas adiabatic index.

(7) Bypass duct

The outlet parameters of aero-engines bypass duct are:

$$\begin{aligned} T_6 &= T_{21} \\ P_6 &= \sigma_4 P_{21} \end{aligned} \quad (11)$$

where;  $T_6$  and  $P_6$  are the total temperature and the total pressure at the outlet of the bypass duct, respectively, and  $\sigma_4$  is the total pressure recovery coefficient of the bypass duct.

(8) Mixer

The gas flow  $q_{mg,7}$  at the outlet of the mixer is the sum of the air flow  $q_{ma,6}$  at the outlet of the bypass duct and the gas flow  $q_{mg,5}$  at the outlet of the low-pressure turbine, let  $\sigma_5$  denotes the total pressure recovery coefficient of the mixer, then the physical parameters at the outlet of the mixer are:

$$\begin{aligned} q_{mg,7} &= q_{mg,5} + q_{ma,6} \\ h_7 &= \frac{h_5 q_{mg,5} + h_6 q_{ma,6}}{q_{mg,7}} \\ P_7 &= \sigma_5 \cdot \frac{P_5 q_{mg,5} + P_6 q_{ma,6}}{q_{mg,7}} \end{aligned} \quad (12)$$

where:  $h_5$ ,  $h_6$ , and  $h_7$  are the specific enthalpy of the gas at the outlet of the low-pressure turbine, air at the outlet of the bypass duct, and gas at the outlet of the mixer, respectively. From  $h_7$ , it is easy to obtain the temperature  $T_7$  at the outlet of the mixer.

(9) Nozzle

The outlet parameters of aero-engines nozzle are:

$$\begin{aligned} \frac{P_8}{P_5} &= \pi_{NZ} \\ q_{mg,8} &= K_q \frac{P_8 A_8 q(\lambda_8)}{\sqrt{T_7}} \end{aligned} \quad (13)$$

where;  $P_8$  are the pressure at the outlet of nozzle,  $A_8$  is the sectional area of the nozzle,  $K_q$  is the state coefficient of the nozzle,  $q(\lambda_8)$  refers to the function related to the characteristics of the bypass duct and the core duct,  $\pi_{NZ}$  is the available pressure drop of the nozzle, and  $P_{S0}$  denotes the standard atmospheric pressure.

## 2.2. Common Working Equations

In this paper, the high-pressure turbine and high-pressure compressor form the high-pressure rotor in the turbofan engine. The high-pressure compressor is driven by the high-pressure turbine. The low-pressure turbine and fan form the low-pressure rotor. After passing through the inlet duct and fan, the gas flow is divided into two parts, with one entering the bypass duct and the other entering the core flow. Based on the flow rate, pressure and power balance between each engine component and the principle of constant rotational speed, the following common working equation can be obtained.

(1) Power balance equation of the high-pressure rotor:

$$P_H = P_{CH} + P_{ex,H} + D_H \left( \frac{dn_H}{dt} \right) \quad (14)$$

where;  $P_H$  is the power of high-pressure turbine,  $P_{CH}$  is the power of high-pressure compressor,  $P_{ex,H}$  is the power lost by transmission friction force,  $D_H \left( \frac{dn_H}{dt} \right)$  is the acceleration power of high-pressure rotor, and  $D_H = (\pi/30)^2 J_H n_H$ ,  $J_H$  is the moment of inertia of the high-pressure rotor, and  $n_H$  is the speed of the high-pressure rotor.

When the aero-engine is in stable state,  $dn_H/dt = 0$ , and ignore the power lost by transmission friction force  $P_{ex,H}$ , that Equation (14) is simplified into:

$$P_H = P_{CH} \quad (15)$$

(2) Power balance equation of the low-pressure rotor:

When the engine is in a stable state,  $dn_L/dt = 0$ . Ignoring the power loss  $P_{ex,L}$  caused by transmission friction, a simplified power balance equation for the low-pressure rotor can be obtained, similar to that of the high-pressure rotor.

$$P_L = P_{CL} \quad (16)$$

where;  $P_L$  is the power of low-pressure turbine, and  $P_{CL}$  is the power of low-pressure compressor.

(3) Flow balance equation of the fan:

After passing through the fan, the gas is divided into two parts: one enters the bypass duct and the other enters the high-pressure compressor. Ignoring loss of gas flow, we have:

$$q_{maF} = q_{maC} + q_{ma6} \quad (17)$$

where;  $q_{maF}$  denotes the air flow from the fan,  $q_{maC}$  denotes the air flow into the high-pressure compressor, and  $q_{ma6}$  denotes the air flow into bypass duct.

(4) Flow balance equation of the high-pressure turbine:

The gas flow of the high-pressure turbine satisfies the following equation.

$$q_{mg,A} = q_{maC} + q_{mf} \quad (18)$$

where;  $q_{mg,A}$  denotes the air flow into the high-pressure turbine,  $q_{maC}$  denotes the air flow from the high-pressure compressor, and  $q_{mf}$  denotes the fuel flow.

(5) Flow balance equation of the low-pressure turbine:

The air flow at the inlet of the low-pressure turbine is equal to the air flow at the outlet of the high-pressure gas turbine  $q_{mg,A2}$ , then:

$$q_{mg,A2} = q_{mg,A} + C_{HPTCool} q_{maC,totalTcool} \quad (19)$$

where;  $C_{HPTCool}$  is the proportion coefficient of the high-pressure compressor bleed air used to cool the high-pressure turbine, and  $q_{maC,totalTcool}$  is the total air flow into the high-pressure turbine.

(6) Flow balance equation of the nozzle:

The gas flow of the nozzle  $q_{mg,8}$  satisfies the following equation.

$$q_{mg,8} = q_{mg,5} + q_{ma,16} \quad (20)$$

where;  $q_{mg,5}$  is the gas flow at the low-pressure turbine outlet, and  $q_{ma,16}$  is the gas flow into the bypass duct.

### 2.3. Linearization of Nonlinear Models

The current design of control systems is mainly linear, so it is necessary to obtain a linear model of aero-engines. Aero-engines have strong non-linear characteristics. The

common method is to select several operating points within the flight envelope and establish a linear model at these points, thus obtaining a segmented linear model of the aero-engine [29].

The partial derivative method is a common method for obtaining the linear model of aero-engines [30]. This method first perturbs a given state variable with a small disturbance, while keeping all other control variables and state variables constant. The partial derivatives of the state variables with the perturbation are then calculated to obtain the state matrix and output matrix of the state space model of the aero-engine. Then, a small perturbation is given to a given control variable, while keeping all other control variables and all state variables constant. The input matrix of the state space model is obtained by taking the partial derivatives of the control variable with the perturbation. This method only has theoretical significance because most state variables and control variables are coupled, and it is difficult to change only a single state variable or control variable without affecting other state variables or control variables. This leads to large modeling errors in this method.

This section briefly introduces the fitting method of linearization for aero-engines [31]. Firstly, a small perturbation state variable model of the aero-engine is obtained based on the selected state variables, control variables, and output variables. The linear dynamic response is calculated by giving a small step input to each control variable. Similarly, the nonlinear dynamic response is obtained by giving the same step input to the control variables of the nonlinear model of the aero-engine. Using the nonlinear dynamic response data as a reference, the matrices in the linear model are fitted to make the linear dynamic response data as close as possible to the nonlinear dynamic response data, thus obtaining the linear model of the aero-engine.

Here, taking the bivariate state-space model of aero-engines as an example, the state-space model is given below:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \quad (21)$$

where; the state variables are selected as the low-pressure rotor speed  $n_L$  and high-pressure rotor speed  $n_H$ , the output variables are selected as the low-pressure rotor speed  $n_L$  and the turbine pressure ratio  $P_iT = \pi_{TL}\pi_{TH}$ , the control variables are selected as the fuel flow  $W_{mf}$ , and the area of the nozzle  $A_8$ . That is,  $x = [n_L \ n_H]^T$ ,  $u = [W_{mf} \ A_8]^T$ , and  $y = [n_L \ P_iT]^T$ .

Let,

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}, C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

From the selected output variables and state variables,  $c_{11} = 1, c_{12} = 0$ .

Giving a small step to the fuel flow while keeping the nozzle area constant, then:

$$\frac{\Delta n_L}{\Delta W_{mf}} \frac{(W_{mf})_0}{(n_L)_0} = \frac{a_{12}b_{21} - a_{22}b_{11}}{a_{11}a_{22} - a_{12}a_{21}} + \frac{(\lambda_1 - a_{22})b_{11} + a_{12}b_{21}}{\lambda_1(\lambda_1 - \lambda_2)} e^{\lambda_1 t} - \frac{(\lambda_2 - a_{22})b_{11} + a_{12}b_{21}}{\lambda_2(\lambda_1 - \lambda_2)} e^{\lambda_2 t}$$

$$\frac{\Delta n_H}{\Delta W_{mf}} \frac{(W_{mf})_0}{(n_H)_0} = \frac{a_{21}b_{11} - a_{11}b_{21}}{a_{11}a_{22} - a_{12}a_{21}} + \frac{(\lambda_1 - a_{11})b_{21} + a_{21}b_{11}}{\lambda_1(\lambda_1 - \lambda_2)} e^{\lambda_1 t} - \frac{(\lambda_2 - a_{11})b_{21} + a_{21}b_{11}}{\lambda_2(\lambda_1 - \lambda_2)} e^{\lambda_2 t}$$

$$c_{21} \frac{\Delta n_L}{\Delta W_{mf}} \frac{(W_{mf})_0}{(n_L)_0} = -c_{22} \frac{\Delta n_H}{\Delta W_{mf}} \frac{(W_{mf})_0}{(n_H)_0}$$

where;  $\lambda_1$  and  $\lambda_2$  are the eigenvalues of the state matrix  $A$ .

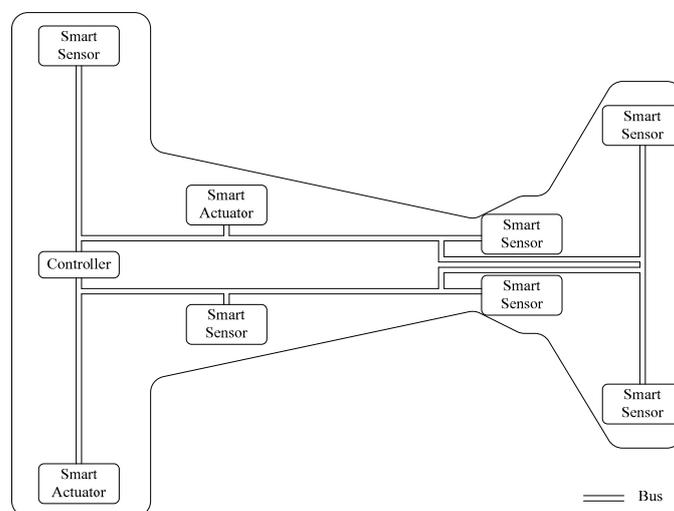
At the same steady-state point, by keeping the nozzle area and fuel flow constant, respectively, small steps are applied to the nozzle area and fuel flow to obtain the component-level nonlinear model's small step response data for aero-engines.

Linear and nonlinear models are given the same amplitude of control variable step at the same steady-state point; the target is to make the linear and nonlinear dynamic responses as close as possible. By establishing and fitting the equation group, the state-space model of aero-engines at that steady-state point can be obtained.

### 3. Aero-Engines' Linear Iterative Controller

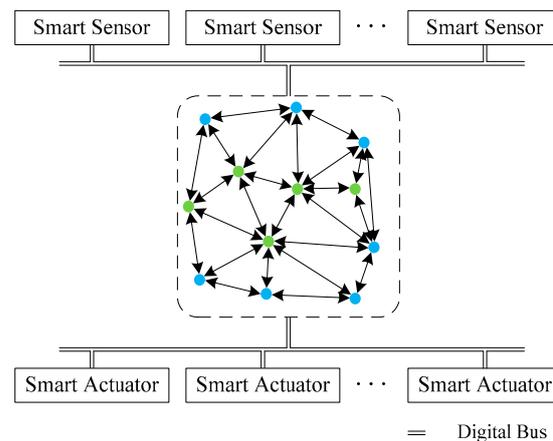
#### 3.1. Software Defined Control Systems of the Aero-Engines

Ref. [32] proposed the Software Defined Control Systems (SDCS) for aero-engines. The SDCS for aero-engines is based on the DCS for aero-engines, and it has been proposed alongside advancements in computer technology, communication technology, sensor technology, data storage technology, and other related support technologies. First, a brief introduction to the DCS for aero-engines will be provided. The DCS for aero-engines was developed based on a centralized control system. In the centralized control system for aero-engines, the controller is connected to the engine's sensors, actuators, and other devices via cables. These devices do not have signal processing or control capabilities and interact with the controller through cables. The controller undertakes all control functions of the control system, including signal acquisition and processing, control law calculation, and control signal output. The structure of the DCS for aero-engines is shown in Figure 1 [33]. In the aero-engine DCS, signal conditioning, A/D, D/A, and other functions are delegated to the sensors and actuators, while the controller only performs core control tasks, such as control law calculations [34,35]. The sensors and actuators integrate microprocessors internally, making them intelligent sensors and intelligent actuators. Communication between the controller, intelligent sensors, and intelligent actuators is performed through a digital bus. Intelligent sensors convert measurement signals into digital signals and provide them to the controller. The functions performed by intelligent sensors include A/D, redundancy management, and interface with the digital bus. Intelligent actuators receive control commands from the controller and control the actuator. The functions performed by intelligent actuators include D/A, closed-loop feedback, redundancy management, and interface with the digital bus. The controller sends control commands to intelligent actuators at a fixed rate through the digital bus. The bus structure of the aero-engine DCS is generally divided into circle structure and linear structure. The circle bus connects each node through a circular structure and has a simple structure. The linear bus uses fewer cables but has a more complex structure. Commonly used buses in aero-engine DCS research include MIL-STD-1553, Field Bus, CAN, and others. Currently, the tasks of controllers are becoming increasingly complex and require new, high-performance microprocessors as controllers. Therefore, this has a certain impact on the reliability of the control system.



**Figure 1.** Overview of a traditional aero-engine DCS.

The feature of SDCS is that the controllers are decentralized. In an SDCS, a decentralized network is formed by a set of low-cost, highly reliable, but low-performance, microprocessor nodes. As shown in Figure 2, each microprocessor node has computing, storage, and communication functions, and data can be transmitted between nodes. Aero-engines SDCS replaces the central controller in aero-engines' DCS with this decentralized network, and the entire decentralized network acts as the controller of the aero-engine control system. Some microprocessors in the network possess bus communication capabilities, allowing the entire decentralized network to communicate directly with intelligent sensors and actuators in the aero-engine control system. In Figure 2, green circles represent conventional nodes, and blue circles represent nodes with bus communication capabilities.



**Figure 2.** Overview of an aero-engine SDCS.

With the structure of aero-engines, the SDCS requires decentralized control tasks to be performed on each decentralized node. The process of obtaining decentralized control tasks from core control tasks is named control task virtualization, and each decentralized task is called a virtual control task (VCT). It can be seen that the core control tasks of the control system are the sum of these VCTs, and each VCT is mapped to a decentralized node in the network. Since each VCT is a subtask of the core control tasks of the control system, the task load on each decentralized node is effectively reduced, which allows for the use of low-cost, high-reliability, and low-performance microprocessors as decentralized nodes of the control system. At the same time, the amount of software code on each node is effectively reduced, which can improve the software reliability of each decentralized node.

### 3.2. Linear Iterative Controller Design Scheme

For the structure of a SDCS, nodes in the network act as controllers to execute system control tasks. Sensors, actuators, and nodes in the network can exchange information with each other, and each node maintains its own internal state. Let  $\gamma = \{v_1, v_2, \dots, v_n\}$  denote the set of all nodes,  $\gamma_D = \{vd_1, vd_2, \dots, vd_d\}$  denote the set of all error tracking nodes,  $\gamma_A = \{va_1, va_2, \dots, va_a\}$  denote the set of nodes that send data to the actuators,  $\gamma_C = \{vc_1, vc_2, \dots, vc_c\}$  denote the set of nodes that execute VCTs, and  $\gamma_{Ci}$  represent the neighboring node set of node  $vc_i$  in  $\gamma_C$ .

The linear iterative working mechanism of the nodes in the network is that each VCT node in the network broadcasts its own state information to other nodes in the network at each cycle. Other task nodes receive the broadcasted state information. After all, the VCT nodes have completed their state broadcasting, and each VCT node updates its own state as a weighted sum of its previous cycle state and the received state information of other VCT nodes. Thus, each task node can act as a small dynamic controller.

Consider the example of the linear iterative network shown in Figure 3, where each node maintains a scalar state. Nodes broadcast their own states to other nodes in the network in a certain order (it is known from the analysis process below that the broadcast

order has no effect on the iteration solution model). In each cycle, node  $vc_4$  broadcasts its state first, then node  $vc_5$  broadcasts its state, and so on, with node  $vc_2$  being the last node to broadcast its state in each cycle. Any node  $vc_i \in \gamma_C$  receives the states of other nodes in  $\gamma_C$ , and updates its own state as a weighted sum of its previous cycle state and the received states of all other nodes. Let  $z_i[k]$  denote the state of node  $vc_i \in \gamma_C$  in the  $k$ th cycle. Taking the example of placing a decentralized controller in the forward path to complete unity feedback control, where  $u_1$  is the tracking error between the output and input, and  $u_2$  is the output of the decentralized controller, based on the linear iterative process of the decentralized controller, we have:

$$z_i[k + 1] = w_{ii}z_i[k] + \sum_{v_j \in \gamma_{Ci}} (w_{ij}z_j[k]) + h_i u_1[k] \tag{22}$$

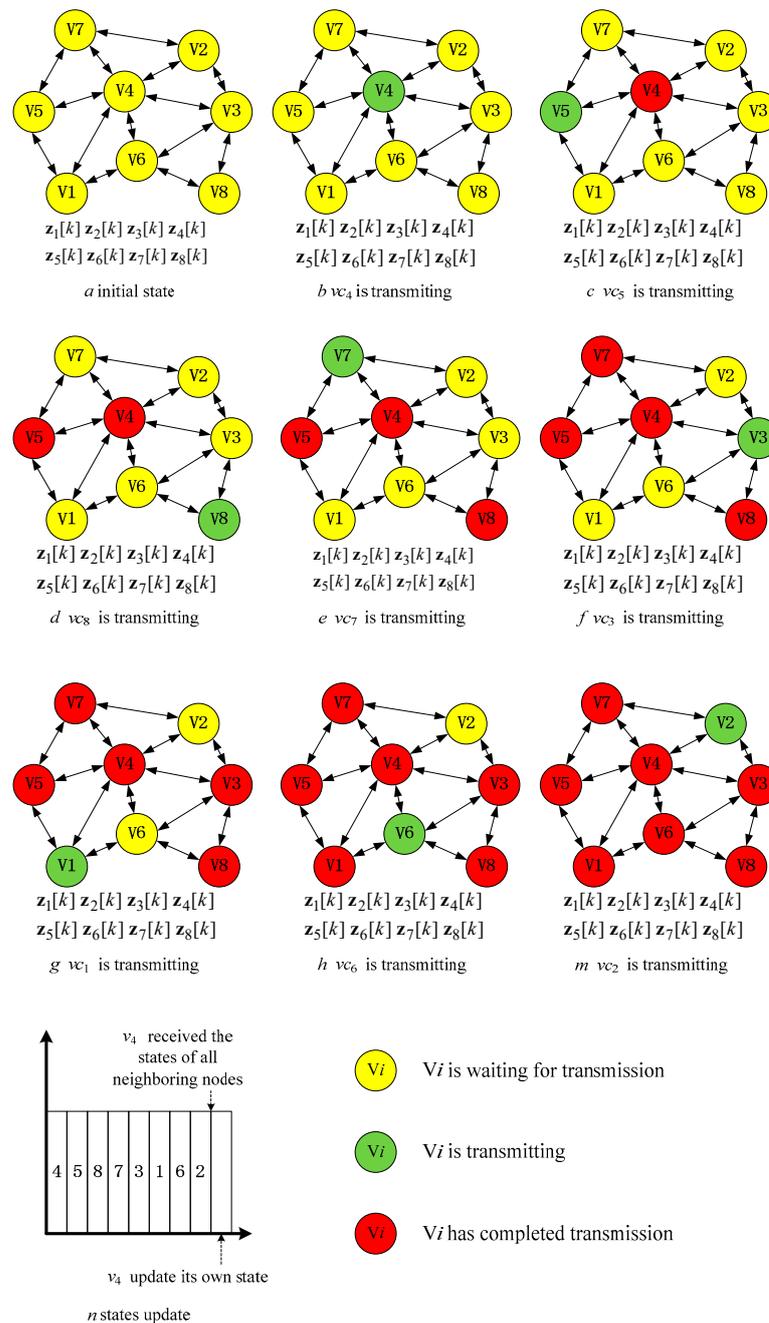


Figure 3. Example of a linear iteration scheme.

The actuator input  $u_2$  is obtained as a linear combination of the tracking error value  $u_1$  and the node states  $z_i$  in  $\gamma_C$ :

$$u_2[k] = \sum_{v_j \in \gamma_C} (g_j z_j[k]) \quad (23)$$

The vector  $z[k] = [z_1[k] \ z_2[k] \ \cdots \ z_N[k]]^T$ , which is a concatenation of all node states in  $\gamma_C$  (where  $N$  is the number of internal state variables maintained by the node in  $\gamma_C$ ), we obtain the iteration process of the entire network as:

$$z[k+1] = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & \cdots & w_{NN} \end{bmatrix} z[k] + \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_N \end{bmatrix} u_1[k] \quad (24)$$

$$u_2[k] = [g_1 \ g_2 \ \cdots \ g_N] z[k] \quad (25)$$

Let,

$$\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & \cdots & w_{NN} \end{bmatrix} = W$$

$$\begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_N \end{bmatrix} = H$$

$$[g_1 \ g_2 \ \cdots \ g_N] = G$$

Then,

$$\begin{aligned} z[k+1] &= Wz[k] + Hu_1[k] \\ u_2[k] &= Gz[k] \end{aligned} \quad (26)$$

This sub-section describes the working process of the linear iteration strategy for nodes in the network and obtains the system model of the linear iteration strategy. From the above analysis, it can be concluded that the broadcasting order of nodes in Figure 3 has no impact on modeling the linear iteration strategy of the network, and the tasks of each node are balanced. The key to designing the linear iteration strategy is to determine the link weights (i.e.,  $W$ ,  $H$ , and  $G$  in Equation (26)). A simple method for obtaining link weight parameters will be introduced in the following sub-sections of this paper.

### 3.3. Control System Modeling

As shown in Figure 4, this is a schematic diagram of the control system structure for aero-engines, with the controller placed in the forward path. In Figure 4,  $u$  represents the system input,  $y$  represents the system output,  $u_1$  represents the tracking error between the output and input, and  $u_2$  represents the output of the controller. The control system adopts unity negative feedback design. The controller calculates the control signal using the difference between the system reference input and output and sends the control signal to the actuator to complete the system control task. The system control task is centralized at a specific controller node.

For SDCS, the system feature is that the controller is decentralized, that is, a controller is composed of several decentralized nodes in the network with computing, storage, and communication capabilities. The corresponding SDCS control system structure diagram is shown in Figure 5. In the figure,  $u$  is the system input,  $y$  is the system output,  $u_1$  is the

tracking error between the output and input,  $u_2$  is the output of the decentralized node based controller, the dashed box represents the decentralized nodes, the blue lines represent communication between nodes, and for simplicity, not all data transmission relationships between decentralized nodes are shown. Each decentralized node has computing, storage, and communication capabilities. The node used to calculate the tracking error between the output and input is called the error tracking node, which is isomorphic to other nodes in the network.

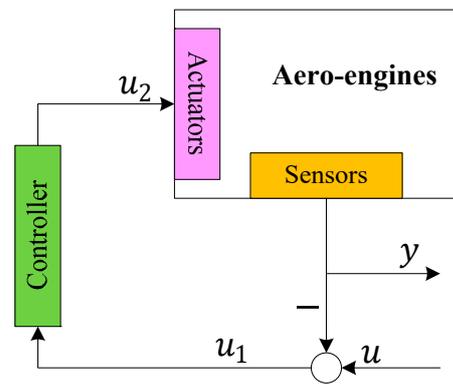


Figure 4. Structure of a unit feedback control system.

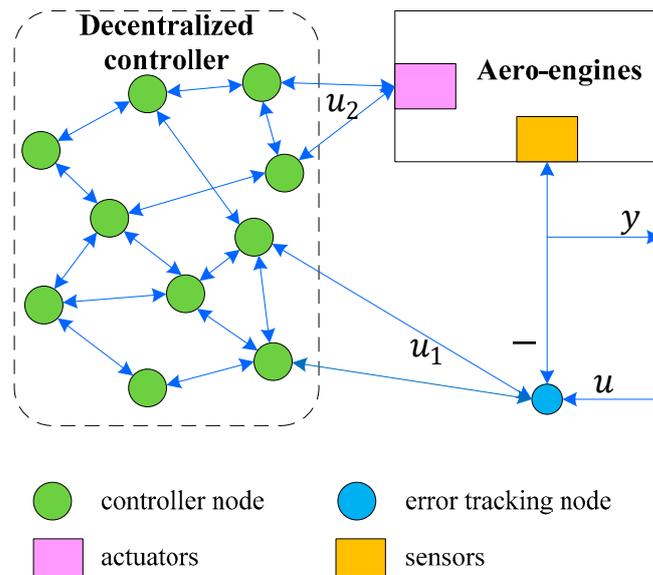


Figure 5. Decentralized controller structure with unit negative feedback.

The general time-discrete linear time-invariant model of the aero-engine in Figure 5 is:

$$\begin{aligned} x[k + 1] &= Ax[k] + Bu_2[k] \\ y[k] &= Cx[k] \end{aligned} \tag{27}$$

From the control system structure in Figure 5,

$$u_1[k] = u[k] - y[k] \tag{28}$$

Based on the periodic linear iterative strategy derived from Equations (24) and (25), the state–space model of the control system can be represented as  $\tilde{x}[k] = [x[k]^T \ z[k]^T]^T$ .

In Figure 5, when the nodes in the network adopt the linear iterative strategy shown in Figure 3, the state equation of the closed-loop system is:

$$\tilde{x}[k+1] = \begin{bmatrix} A & BG \\ -HC & W \end{bmatrix} \tilde{x}[k] - \begin{bmatrix} 0 \\ H \end{bmatrix} u[k] \quad (29)$$

Let,

$$\begin{bmatrix} A & BG \\ -HC & W \end{bmatrix} = \tilde{A}$$

$$\begin{bmatrix} 0 \\ H \end{bmatrix} = \tilde{B}$$

Then,

$$\tilde{x}[k+1] = \tilde{A}\tilde{x}[k] - \tilde{B}u[k] \quad (30)$$

From the above analysis, it can be concluded that the control-related computing tasks performed by the decentralized controller nodes responsible for system control are described by Equation (22), which is a linear weighted sum process. The linear weighted sum of a single node is equivalent to a VCT. It can be seen that, for each VCT, only simple addition and multiplication operations are required, with minimal computational requirements and low utilization of node computing resources. Each VCT actually uses its own state and the state of other task nodes to perform linear iterations in order to maintain its own state.

Based on the running process and mathematical description of the linear iterative SDCS closed-loop control system presented in this section, the following advantages of this method can be identified:

#### 1. Low resource consumption

In the linear iteration scheme proposed in this section, the task of each VCT is a linear weighted sum of the node's own state and the states of other task nodes. The computation is simple and requires minimal resources, resulting in low computational overhead for the nodes. As aero-engines are safety-critical systems that often use mature and stable but low-performance electronic devices, the SDCS control system design based on the linear iteration scheme is suitable for aero-engine control systems. It can use limited computing resources to perform simple periodic tasks and complete system control tasks.

#### 2. Balanced resource consumption

As mentioned earlier, the task of each VCT is a linear weighted sum of multiple node states, resulting in equal task loads for each task node and balanced resource consumption across the nodes.

#### 3. Good software performance

Since the tasks of each task node are the same, with only different link weights, there is no need to design the software system for each VCT separately. In addition, the simple weighted summation task of each VCT results in less code and simpler tasks, making it less likely to affect the control system due to software system failures during operation.

#### 4. Easy to develop

This solution does not require homogeneous hardware nodes, as long as the nodes meet the basic requirements of computing, communication, and storage performance, they can be easily added to the control system for system expansion.

In this section, a SDCS model based on the linear iterative strategy was obtained, and a fully distributed decentralized controller was implemented, where VCT represents the linear weighted combination process executed on a single task node. Not every node needs to directly exchange data with sensors and actuators, as data exchange mainly occurs

between decentralized nodes in the network. The tuning of link weights in the linear iteration process of this scheme will be described in detail in the next section.

#### 4. Controller Parameter Tuning Method Based on LMI

To ensure the normal running of the control system, it is essential to ensure that the control system is stable. For the design scheme in this paper, since the linear iterative process of the decentralized controller is discrete, it is required that the system be Schur stable.

The most intuitive method is to use the Lyapunov method [36], which involves finding a matrix  $P$  that satisfies the following equation.

$$\begin{aligned} P &> 0 \\ \tilde{A}^T P \tilde{A} - P &< 0 \end{aligned} \quad (31)$$

However, since the linear iterative method of the decentralized controller in this paper does not impose any restrictions on the network topology of the nodes, and the parameters  $P$  and  $\tilde{A}(W, H, G)$  in Equation (31) are nonlinear with respect to the Lyapunov condition, it is difficult to solve using general Linear Matrix Inequality (LMI) methods. Here, the following lemma is introduced,

**Lemma 1.** *System  $\tilde{A}$  is Schur stable if and only if there exist matrices  $P$  and  $Q$ , such that the following equation holds:*

$$\begin{aligned} P &> 0 \\ Q &> 0 \\ Q &= P^{-1} \\ \tilde{A}^T Q^{-1} \tilde{A} - P &< 0 \end{aligned} \quad (32)$$

**Proof.** From Equation (31), according to the Schur complement theorem, we can conclude that,

$$\begin{bmatrix} P & \tilde{A}^T \\ \tilde{A} & P^{-1} \end{bmatrix} > 0$$

Let,  $Q = P^{-1}$ , then,

$$\begin{bmatrix} P & \tilde{A}^T \\ \tilde{A} & Q \end{bmatrix} > 0$$

According to the Schur complement theorem, we can obtain that,

$$\tilde{A}^T Q^{-1} \tilde{A} - P < 0$$

Additionally,  $P$  and  $Q$  are positive definite matrices.  $\square$

From Refs. [37,38], we can obtain the following lemma.

**Lemma 2.** *For positive definite matrices  $P$  and  $Q$ ,  $P$  and  $Q$  are the optimal solutions to the following optimization problem, if and only if  $Q = P^{-1}$ :*

$$\begin{aligned} &\min \text{trace}(PQ) \\ &s.t. \quad \begin{cases} P > 0 \\ Q > 0 \\ P > Q^{-1} \end{cases} \end{aligned}$$

By Lemma 1 and Lemma 2, we can easily derive the following conclusion:

System  $\tilde{A}$  is Schur stable, if and only if the following optimization problem has a solution.

$$\begin{aligned} & \min \text{trace}(PQ) \\ & \text{s.t.} \quad \begin{cases} P > 0 \\ Q > 0 \\ P > Q^{-1} \\ \tilde{A}^T Q^{-1} \tilde{A} - P < 0 \end{cases} \end{aligned} \tag{33}$$

In the above optimization condition, the four constraint conditions are linear, but the objective function  $\min \text{trace}(PQ)$  is nonlinear. In order to facilitate solving the problem using LMI method, the objective function can be linearized at any  $P_0$  and  $Q_0$  [38], that is:

$$\text{trace}(PQ)|_{P_0, Q_0} = \text{trace}(P_0Q + PQ_0) + c \tag{34}$$

In Equation (34),  $c$  is a constant. Therefore, the optimization objective can take  $\text{trace}(P_0Q + PQ_0)$  as the objective function. Since  $\text{trace}(P_0Q + PQ_0)$  is linear, the LMI method can be used to solve the above optimization problem. Based on this conclusion, the following  $\tilde{A}(W, H, G)$  parameter tuning algorithm is given. First, the LMI problem is presented:

$$\begin{aligned} & \min \text{trace}(P_k Q_{k+1} + P_{k+1} Q_k) \\ & \text{s.t.} \quad \begin{cases} P_{k+1} > 0 \\ Q_{k+1} > 0 \\ P_{k+1} > Q_{k+1}^{-1} \\ \tilde{A}_{k+1}^T Q_{k+1}^{-1} \tilde{A}_{k+1} - P_{k+1} < 0 \end{cases} \end{aligned} \tag{35}$$

The flowchart for the parameter tuning algorithm is shown in Figure 6.

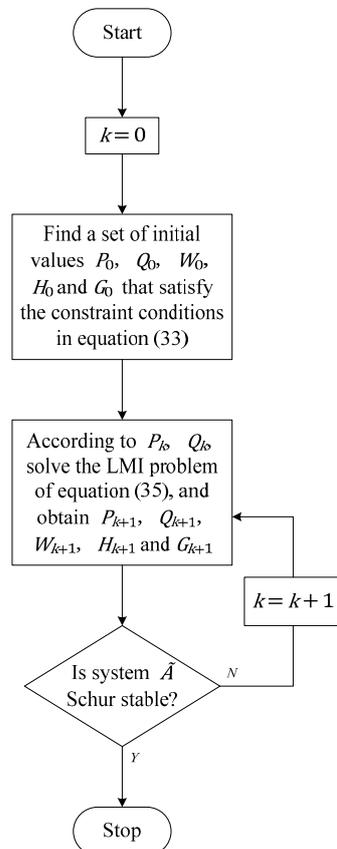


Figure 6. The flowchart for the parameter tuning algorithm.

In the above algorithm, during each iteration,  $P_k$  and  $Q_k$  are equivalent to  $P_0$  and  $Q_0$  in Equation (34), and the matrix variables  $P_{k+1}$  and  $Q_{k+1}$  to be solved are equivalent to  $P$  and  $Q$  in Equation (34). Therefore, the optimization objective function in Equation (35),  $\min \text{trace}(P_k Q_{k+1} + P_{k+1} Q_k)$ , is linear. The parameters  $W_{k+1}$ ,  $H_{k+1}$ , and  $G_{k+1}$  obtained from the above algorithm are the required parameters  $W$ ,  $H$ , and  $G$  for the decentralized controller. Under this control parameterization, the control system  $\tilde{A}$  is Schur stable.

## 5. Simulation Analysis

The previous sections introduced a decentralized controller design based on linear iteration, provided the mathematical model of the control system, and presented a controller parameter tuning method based on LMI. The purpose of this simulation is to verify the feasibility of the proposed decentralized controller design of the aero-engines SDCS and parameter tuning algorithm in this paper.

By using the modeling method in Section 2, we obtain the discrete-time state space model of a turbofan engine under the conditions of 0 altitude and 0 Mach number, which is given by:

$$\begin{cases} \begin{bmatrix} n_L[k+1] \\ n_H[k+1] \end{bmatrix} = A \begin{bmatrix} n_L[k] \\ n_H[k] \end{bmatrix} + B \begin{bmatrix} W_{mf}[k] \\ A_8[k] \end{bmatrix} \\ \begin{bmatrix} n_L[k] \\ P_i T[k] \end{bmatrix} = C \begin{bmatrix} n_L[k] \\ n_H[k] \end{bmatrix} \end{cases}$$

where,

$$A = \begin{bmatrix} 0.9561 & -0.0139 \\ -0.0168 & 0.9721 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0152 & 0.0117 \\ 0.0104 & 0.0081 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ -3.7430 & 6.8260 \end{bmatrix}$$

$n_L$  and  $n_H$  are the low-pressure and high-pressure rotor speeds of the turbofan engine, respectively,  $P_i T = \pi_{TL} \pi_{TH}$  is the turbine pressure ratio,  $W_{mf}$  is the fuel flow, and  $A_8$  is the area of the nozzle.

Using the LMI based algorithm designed in Section 4 and the MATLAB YALMIP toolbox, with “sdpt3” solver, the parameters  $W$ ,  $H$ , and  $G$  of the decentralized controller are calculated as follows:

$$W = \begin{bmatrix} 2.0579 & 1.6296 \\ 1.6139 & 1.2523 \end{bmatrix} \times 10^{-5}$$

$$H = \begin{bmatrix} 4.0930 \times 10^{-4} & -5.9683 \times 10^{-5} \\ -5.9683 \times 10^{-5} & -7.7824 \times 10^{-5} \end{bmatrix}$$

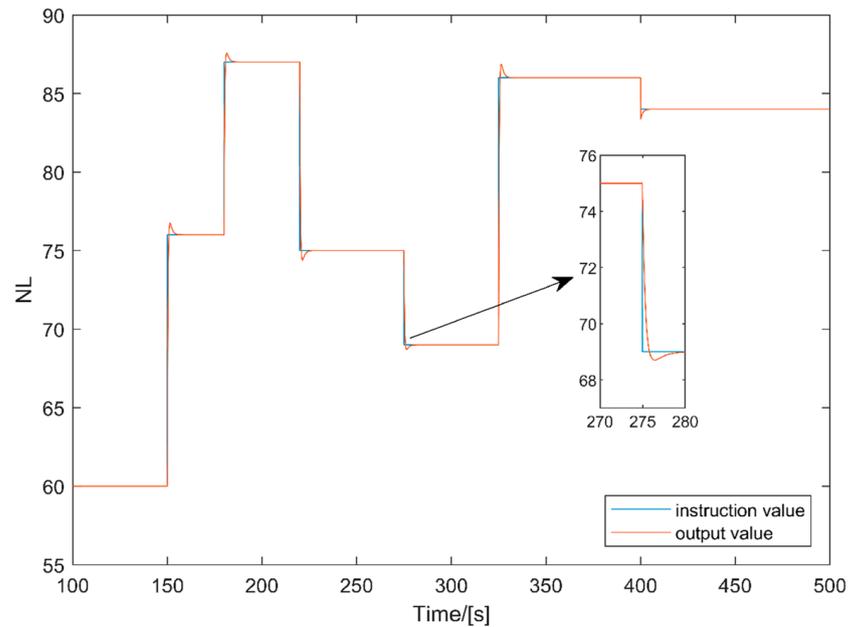
$$G = \begin{bmatrix} -0.1846 & 0.2545 \\ 0.2545 & -0.3239 \end{bmatrix}$$

It can be verified that  $\tilde{A}(W, H, G)$  is Schur stable. Therefore, it can be seen that only two nodes are required to form a decentralized controller of the aero-engine’s control system.

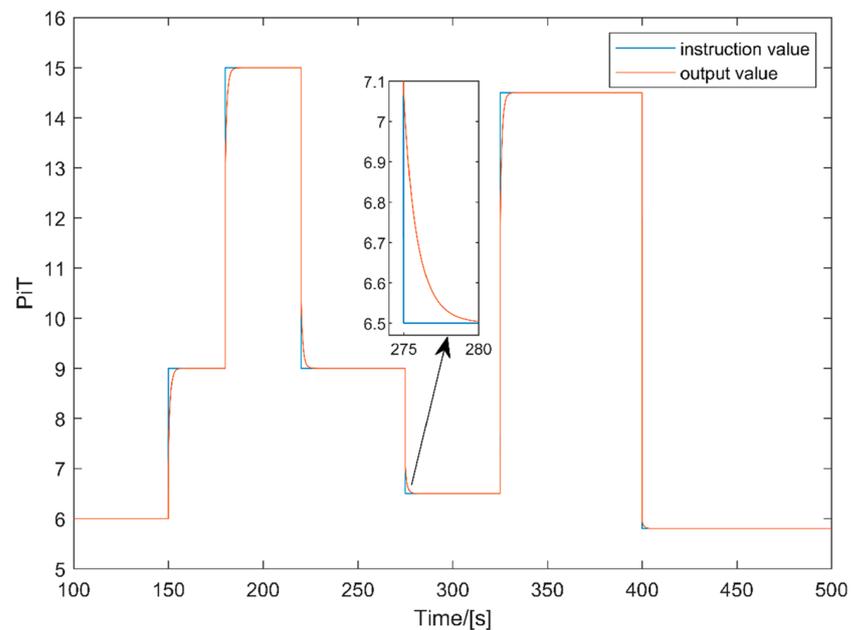
Figure 7 shows the schematic diagram of the low-pressure rotor speed response of an aero-engine under corresponding inputs. Figure 8 shows the schematic diagram of the turbine pressure ratio response of an aero-engine under corresponding inputs.

From the simulation results of Figures 7 and 8, it can be seen that the response of the low-pressure rotor speed and turbine pressure ratio under the corresponding input is stable. In the design scheme proposed in this paper, a decentralized controller design scheme is first presented to obtain the VCT of the SDCS. The model of the control system is obtained

through the model of the decentralized controller. A control parameter tuning scheme is given using LMI, which obtains the parameters of each VCT. For the parameter tuning scheme, the objective is to ensure the stability of the control system. By analyzing the result of the parameter tuning scheme, the system  $\tilde{A}(W, H, G)$  is Schur stable, meanwhile, the simulation results show that the system is stable, demonstrating the effectiveness of the proposed design.



**Figure 7.** Schematic diagram of the low-pressure rotor speed response.



**Figure 8.** Schematic diagram of the turbine pressure ratio response.

In the design process of this paper, a decentralized controller can be composed using only a small number of nodes, making the design of the control system simple. Furthermore, because the internal nodes of the controller use periodic broadcasting for linear iteration, the communication resources consumed by the controller increase as the number of nodes executing the VCT increases. From the simulation results, it can be seen that using a

decentralized node with the same number as the system order can complete the control task well.

The proposed design scheme makes the application of SDCS in aero-engine control systems simple. From the perspective of control system design, existing aero-engine models can be used to obtain the control tasks of decentralized controller nodes through the algorithm introduced in this paper, simplifying the system design. From the perspective of software design, the control tasks executed by each decentralized node in the controller are simply linearly weighted sums of the node's own state and the states of other nodes. Different VCTs executed on different nodes only have different weights on the state, and the software structure is the same, making software code writing and development easy and short. From the perspective of hardware development, since the consumption of computing, storage, and communication resources between different VCTs is indistinguishable, and homogeneous nodes can be used for hardware system design, making hardware design easy without requiring specific designs for different VCTs, effectively reducing the difficulty and cycle of the hardware design process.

## 6. Conclusions

This paper briefly introduces the architecture of SDCS for aero-engines. The SDCS is composed of a network of low-cost, low-performance, and highly reliable nodes that serve as the controller for the aero-engine control system. The characteristic of SDCS is that the controller is decentralized, and there is no central controller node in the control system. In order to meet the decentralized characteristic of the SDCS controller, this paper introduces a virtualization scheme for control tasks based on node linear iteration, obtaining the virtual control tasks executed on each node. The linear iterative scheme for decentralized controllers is modeled, and the mathematical model of the aero-engine control system is obtained. In the proposed scheme, the control tasks executed on each node have balanced workloads, and the workloads of the decentralized nodes are small, making it easy to use low-cost, low-performance, and highly reliable microprocessors as decentralized controllers of the control system. To stabilize the control system with the designed controller, a controller parameter tuning scheme based on the LMI method is introduced. Finally, the feasibility of the proposed design scheme is verified through digital simulation.

**Author Contributions:** Conceptualization, X.J. and J.L.; data curation, J.R.; investigation, X.J. and J.R.; methodology, X.J., J.L. and Y.W.; software, X.J. and J.R.; supervision, J.L. and Y.W.; validation, X.J. and J.R.; writing—original draft, X.J.; writing—review and editing, X.J., J.L., J.R. and Y.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data used to support the findings of this study are included within the article.

**Conflicts of Interest:** The authors declare that they have no competing financial and non-financial interests.

## References

1. Wang, X.; Du, X.; Wang, X.; Sun, X.; Li, Y. Controller Design of Aero-engines under the Distributed Architecture with Time Delays. In Proceedings of the 2020 39th Chinese Control Conference (CCC), Shenyang, China, 27–29 July 2020.
2. De Giorgi, M.G.; Strafella, L.; Ficarella, A. Neural Nonlinear Autoregressive Model with Exogenous Input (NARX) for Turbohaft Aeroengine Fuel Control Unit Model. *Aerospace* **2021**, *8*, 206. [\[CrossRef\]](#)
3. Zhang, L.; Xie, S.; Zhang, Y.; Ren, L.; Zhou, B.; Wang, H.; Peng, J.; Wang, L.; Li, Y. Aero-Engine DCS Fault-Tolerant Control with Markov Time Delay Based on Augmented Adaptive Sliding Mode Observer. *Asian J. Control.* **2020**, *22*, 788–802. [\[CrossRef\]](#)
4. Lv, C.; Chang, J.; Bao, W.; Yu, D. Recent Research Progress on Airbreathing Aero-Engine Control Algorithm. *Propuls. Power Res.* **2022**, *11*, 1–57. [\[CrossRef\]](#)
5. Culley, D.; Thomas, R.; Saus, J. Concepts for Distributed Engine Control. In Proceedings of the 43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, Cincinnati, OH, USA, 8–11 July 2007.

6. Gianluca, A. Interconnected Dynamic Systems: An Overview on Distributed Control. *IEEE Contr. Syst. Mag.* **2013**, *33*, 76–88.
7. Xu, S.; Bao, J. Distributed Control of Plantwide Chemical Processes. *J. Process. Control.* **2009**, *19*, 1671–1687. [[CrossRef](#)]
8. Chen, T.; Shan, J. Distributed Tracking of Multiple Under-actuated Lagrangian Systems with Uncertain Parameters and Actuator Faults. In Proceedings of the 2019 American Control Conference (ACC), Philadelphia, PA, USA, 10–12 July 2019.
9. Chen, T.; Shan, J. Distributed Adaptive Attitude Control for Multiple Underactuated Flexible Spacecraft. In Proceedings of the 2018 Annual American Control Conference (ACC), Milwaukee, WI, USA, 27–29 June 2018.
10. Chen, T.; Shan, J. Distributed Tracking of a Class of Underactuated Lagrangian Systems with Uncertain Parameters and Actuator Faults. *IEEE Trans. Ind. Electron.* **2020**, *67*, 4244–4253. [[CrossRef](#)]
11. Zhao, Y.; Qiu, H.; Song, H. Distributed Measurement and Control System of the Test-rig of Ram-compressed Rotor Aero-engine Based on PLC and PXI bus. In Proceedings of the 2012 IEEE International Conference on Control Applications, Dubrovnik, Croatia, 3–5 October 2012.
12. Li, G.; Wang, X.; Ren, X. Multi-package Transmission Aero-engine DCS Neural Network Sliding Mode Control Based on Multi-Kernel LS-SVM Packet Dropout Online Compensation. *PLoS ONE* **2020**, *15*, e0234356.
13. Zhang, S.; Zhang, D.; Zhang, Z.; Zhang, Q.; Lu, B. Dynamic Control Allocation Fault-Tolerant Method for a Class of Distributed Control Systems. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018.
14. Lubomir, B. Decentralized Control: An Overview. *Annu. Rev. Control* **2008**, *32*, 87–98.
15. Lubomir, B. Decentralized Control: Status and Outlook. *Annu. Rev. Control* **2014**, *38*, 1367–5788.
16. Maria, F.; Agostino, M.; Giovanni, P.; Walter, U. A Decentralized Control Strategy for the Coordination of AGV Systems. *Control Eng. Pract.* **2018**, *70*, 86–97.
17. Yang, C.; Jaime, L.; Antonia, T.; Andreas, M. Decentralized Control of Distributed Cloud Networks with Generalized Network Flows. *IEEE. Trans. Commun.* **2023**, *71*, 256–268.
18. Shojaee, M.; Azizi, M. Optimal Decentralized Control of a Wind Turbine and Diesel Generator System. *Optim. Control Appl. Meth.* **2023**, *44*, 677–698. [[CrossRef](#)]
19. Pan, M.; Cao, L.; Zhou, W.; Huang, J.; Chen, Y. Robust Decentralized Control Design for Aircraft Engines: A Fractional Type. *Chin. J. Aeronaut.* **2019**, *32*, 347–360. [[CrossRef](#)]
20. Ji, X.; Li, J.; Ren, J.; Wu, Y.; Wang, K. Fully Connected Clustering Based Software Defined Control System and Node Failure Analysis. *J. Northwest. Polytech. Univ.* **2019**, *37*, 1238–1247. [[CrossRef](#)]
21. Stöhr, M.; Geigle, K.; Hadeff, R.; Boxx, I.; Carter, C.; Grader, M.; Gerlinger, P. Time-Resolved Study of Transient Soot Formation in an Aero-engine Model Combustor at Elevated Pressure. *Proc. Combust. Inst.* **2019**, *37*, 5421–5428. [[CrossRef](#)]
22. Zheng, Q.; Fang, J.; Hu, Z.; Zhang, H. Aero-Engine On-Board Model Based on Batch Normalize Deep Neural Network. *IEEE Access* **2019**, *7*, 54855–54862. [[CrossRef](#)]
23. Pang, S.; Li, Q.; Feng, H. A Hybrid Onboard Adaptive Model for Aero-engine Parameter Prediction. *Aerosp. Sci. Technol.* **2020**, *105*, 105951. [[CrossRef](#)]
24. Ren, L.; Ye, Z.; Zhao, Y. A Modeling Method for Aero-engine by Combining Stochastic Gradient Descent with Support Vector Regression. *Aerosp. Sci. Technol.* **2020**, *99*, 105775. [[CrossRef](#)]
25. Zhao, Y.; Zhao, J.; Fu, Y.; Shi, Y.; Chen, C. Rate Bumpless Transfer Control for Switched Linear Systems with Stability and Its Application to Aero-Engine Control Design. *IEEE Trans. Ind. Electron.* **2020**, *67*, 4900–4910. [[CrossRef](#)]
26. Shi, Y.; Sun, X. Bumpless Transfer Control for Switched Linear Systems and Its Application to Aero-Engines. *IEEE Trans. Circuits I.* **2021**, *68*, 2171–2182. [[CrossRef](#)]
27. Morteza, M.; Ali, R.; Ali, J.; Milad, E. Design and Implementation of MPC for Turbofan Engine Control System. *Aerosp. Sci. Technol.* **2019**, *92*, 99–113.
28. Morteza, M.; Ali, R. Analyzing Different Numerical Linearization Methods for the Dynamic Model of a Turbofan Engine. *Mech. Ind.* **2019**, *20*, 303.
29. Gou, L.; Zeng, X.; Wang, Z.; Han, G.; Lin, C.; Cheng, X. A Linearization Model of Turbofan Engine for Intelligent Analysis Towards Industrial Internet of Things. *IEEE Access.* **2019**, *7*, 145313–145323. [[CrossRef](#)]
30. Chen, Q.; Huang, J.; Pan, M.; Lu, F. A Novel Real-Time Mechanism Modeling Approach for Turbofan Engine. *Energies* **2019**, *12*, 3791. [[CrossRef](#)]
31. Ling, Y.; Zhou, W.; Zhu, P.; Zeng, J. Modeling of a Large Envelope System for Turbofan Engine. *J. Nanjing Univ. Aeronaut. Astronaut.* **2021**, *53*, 529–536.
32. Ji, X.; Li, J.; Ren, J.; Wu, Y. A Decentralized LQR Output Feedback Control for Aero-Engines. *Actuators* **2023**, *12*, 164. [[CrossRef](#)]
33. Belapurkar, R. Stability and Performance of Proplulsion Control Systems with Distributed Control Architectures and Failure. Ph.D. Thesis, The Ohio State University, Columbus, OH, USA, 2012.
34. Thompson, H.; Fleming, P. Distributed Aero-Engine Control Systems Architecture Selection Using Multi-Objective Optimisation. In Proceedings of the 5th IFAC Workshop on Algorithm & Architecture for Real Time Control (AARTC' 98), Cancun, Mexico, 15–17 April 1998.
35. Skira, C.; Agnello, M. Control Systems for the Next Century's Fighter Engines. *J. Eng. Gas Turbines Power.* **1992**, *114*, 749–754. [[CrossRef](#)]
36. Ricardo, C.; Pedro, L. LMI Conditions for Robust Stability Analysis Based on Polynomially Parameter-dependent Lyapunov Functions. *Syst. Control Lett.* **2006**, *55*, 52–61.

37. Bruno, M.; Joao, Y.; Eduardo, S. LMI-based Consensus of Linear Multi-agent Systems by reduced-order Dynamic Output Feedback. *ISA Trans.* **2020**, *129*, 121–129.
38. Ghaoui, L.; Oustry, F.; AitRami, M. A cone complementarity linearization algorithm for static output-feedback and related problems. *IEEE Trans. Automat. Contr.* **1997**, *42*, 1171–1176. [[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.