

Article

INVSID 1.0: An Inverse System Identification Toolbox for MATLAB

Runzhe Han ^{1,*} , Christian Bohn ² and Georg Bauer ²

¹ College of Optical, Mechanical and Electrical Engineering, Zhejiang A&F University, Hangzhou 311300, China

² Institut für Elektrische Informationstechnik, Technische Universität Clausthal, 38678 Clausthal-Zellerfeld, Germany; bohn@iei.tu-clausthal.de (C.B.); bauer@iei.tu-clausthal.de (G.B.)

* Correspondence: runzhe.han@alumni.tu-clausthal.de; Tel.: +86-19558180976

Abstract: The inverse system identification toolbox named INVSID 1.0 for MATLAB, which is used to identify the inversion of single-input single-output systems, is developed. The complete process from theoretical derivation to toolbox creation of developing the toolbox is demonstrated. Afterwards, numerical examples are illustrated to describe how the toolbox can be used to solve inverse identification problems. Simulation results demonstrate the effectiveness of the toolbox.

Keywords: system inversion; inverse system identification; MATLAB toolbox

1. Introduction

Nowadays, in many motion control systems, there are requirements of high performance, such as short motion times and small settling times. To fulfill these demands, combining feedback with feedforward control is normally implemented [1]. Figure 1 displays how a feedforward controller can be involved in feedback control systems. In total, there are two kinds of modes. The feedback controller guarantees stability and improves disturbance rejection, while the feedforward controller enhances tracking performance such that the feedforward controller should be designed as $F = G^\dagger$ (in the first mode) and $F = G_c^\dagger$ (in the second mode), where the symbol “ \dagger ” denotes the Moore–Penrose pseudo-inverse [2]. So system inversion is the key to the problem of feedforward control. Actually, in addition to applications in control systems, system inversion is frequently used in the areas of sensor calibration, loudspeaker linearization, digital predistortion for radio frequency communications, and so on [3]. So system inversion plays an important role in various research areas.

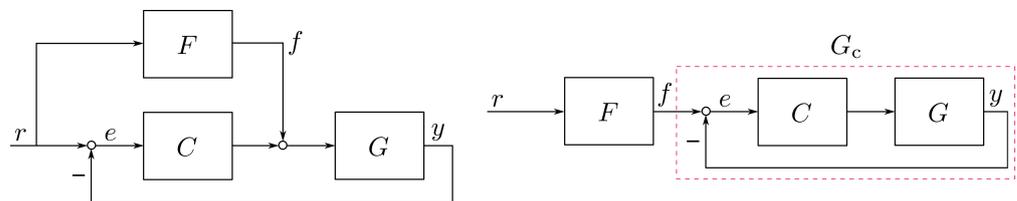


Figure 1. Inverse model-based feedforward–feedback control (r : reference signal; F : feedforward controller; C : feedback controller; G : plant model; G_c : closed-loop system model; e : error; f : feedforward controller output; y : control system output).

System inversion can be conducted by direct inversion and indirect inversion [4]. There exist several kinds of intrinsic limitations of direct inversion approaches; here, an example is used to illustrate this, denoting the transfer function of a finite-dimensional, discrete-time, single-input single-output, linear, constant dynamical system as



Citation: Han, R.; Bohn, C.; Bauer, G. INVSID 1.0: An Inverse System Identification Toolbox for MATLAB. *Appl. Sci.* **2024**, *14*, 6838. <https://doi.org/10.3390/app14156838>

Academic Editor: Alexander Zhibanov

Received: 21 May 2024

Revised: 10 July 2024

Accepted: 14 July 2024

Published: 5 August 2024



Copyright: © 2024 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/>).

$$G(z) = C(zI_n - A)^{-1}B + D := \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right], \quad (1)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^n$, $C \in \mathbb{R}^{1 \times n}$, $D \in \mathbb{R}$, I_n denotes an n -dimensional identity matrix, and (A, B, C, D) is a state-space realization of $G(z)$. Based on (1), the inversion of $G(z)$ can be represented as [5]

$$G^\dagger(z) := \left[\begin{array}{c|c} A - BD^\dagger C & -BD^\dagger \\ \hline D^\dagger C & D^\dagger \end{array} \right]. \quad (2)$$

There are at least two challenges in the direct inversion (2). (i) When D^\dagger does not exist, the direct inversion cannot be conducted; (ii) when there exist nonminimum-phase zeros (For discrete-time systems, nonminimum-phase zeros are zeros that lie outside the unit disk) in $G(z)$, the inversion $G^\dagger(z)$ will be unstable.

Due to the limitations of direct inversion approaches, various indirect inversion approaches have thus been proposed. A possible classification of existing indirect system inversion approaches consists in distinguishing between preview-based and non-preview-based approaches [4,6]:

- (a) Preview-based inversion approaches can be further categorized into infinite preview-based approaches and finite preview-based approaches. Infinite preview-based approaches admit an exact stable inversion solution; however, such a solution may require an infinite pre-actuation [7–10]. Because the length of the pre-actuation is proportional to the length of the desired output preview, infinite preview-based approaches are not applicable from a practical point of view. To handle the problem of applicability, finite preview-based approaches have been proposed [11–18].
- (b) Non-preview-based inversion approaches are preferred in practice. A family of approaches called pseudo-inversion, which can be conducted without preview, has been proposed [19,20]. However, such approaches will encounter other problems such as the difficulty of choosing a suitable basis function. Direct system identification-based inversion approaches use the input–output data from the system to be inverted to identify the inverse system directly; however, the system identification cannot be conducted when the system to be inverted is not stable [3]. For signal modeling-based inversion approaches, the input signal, which is to be reconstructed, must be a periodic signal under stationary operating conditions [21,22].

It should be noted that guaranteeing the stability of obtained solutions is a priority of both direct and indirect inversion approaches. So for some indirect inversion approaches, stability is ensured, but infinite or finite pre-actuation is needed, which cannot be applied well in practice because sometimes the desired output is unknown.

In this paper, an entirely different system inversion approach by combining time-domain observer design and frequency-domain subspace identification is proposed. The presented approach can guarantee the stability of obtained system inversion, and simultaneously the proposed approach does not need any pre-actuation. Moreover, the approach can be applied to stable or unstable systems/minimum-phase or nonminimum-phase systems to be inverted, and there is also no requirement for the type of input and output signals. Furthermore, it does not suffer from non-convex or input noise problems.

Even though the proposed approach has the above advantages by comparing with other approaches, the current version of the proposed approach can just be used for solving the system inversion problem of systems of which the inputs do not have overlapping frequency ranges. In this paper, only single-input single-output systems are considered.

To facilitate the use of the proposed system inversion approach by third parties, a MATLAB toolbox implementing the approach is created in this paper after theoretical derivation of the approach. The full name of the MATLAB toolbox is **IN**verse **S**ystem **ID**entification, with the first version abbreviated as **INVSID 1.0**.

The remainder of the paper is organized as follows. In Section 2, the inversion approach is first proposed, and then the corresponding MATLAB codes are generated,

based on which the toolbox NIVSID 1.0 is created, followed by Section 3, in which the usage of the toolbox NIVSID 1.0 is validated by both simulation and practical examples. Finally, conclusions and perspectives are given in Section 4.

2. Creation of INVSID 1.0

This section is illustrated by using three subsections with a progressive relationship. The proposed inverse identification approach is first presented, then on the basis of which the corresponding MATLAB codes and the toolbox INVSID 1.0 are finally created. It should be noted that the toolbox INVSID 1.0 is only used for the inverse identification of single-input single-output systems.

2.1. Inverse Identification Approach

A finite-dimensional, discrete-time, single-input single-output, linear, constant dynamical system G_d which is minimal-realized (The system is minimal-realized if and only if it is both controllable and observable) and proper (The system is proper when the degree of the numerator does not exceed the degree of the denominator of its transfer function; otherwise, the system is improper) is given, and the given system G_d can be either stable or unstable, and the sampling period of system G_d is T_s in seconds.

Figure 2 is used to demonstrate the basic idea behind the proposed inverse system identification approach, based on which the inverse model of the nominal model G_d can be derived. As can be seen in Figure 2, the proposed approach mainly consists of five steps:

- (a) Obtain a state-space representation of the nominal system G_d .
- (b) Generate a number of sine signals $u_m(k)$, $m = 1, 2, \dots, N$, corresponding to a number of N specified frequencies, model them in state-space representation such that N signal models G_m , $m = 1, 2, \dots, N$, can be obtained.
- (c) By combining the signal models G_m for $m = 1, 2, \dots, N$ with the model G_d , respectively, the augmented models $G_{a,m}$, $m = 1, 2, \dots, N$, can be obtained. Then, based on using the observers for the augmented models $G_{a,m}$, $m = 1, 2, \dots, N$, inverse models $G_{inv,m}$ with $m = 1, 2, \dots, N$, which can be used for reconstructing the input signals $u_m(k)$, $m = 1, 2, \dots, N$, can be obtained.
- (d) Use frequency-domain system identification approaches to identify the inverse model of G_d based on the frequency response function values of the models $G_{inv,m}$ with $m = 1, 2, \dots, N$ at specified frequencies.
- (e) Choose the best inverse model by validating the identified inverse models.

The above steps are discussed in detail as follows.

Step 1: If the transfer function of the system G_d is denoted as $G_d(z)$, and (A_d, B_d, C_d, D_d) represents a minimal realization of $G_d(z)$, the corresponding state-space model of the system G_d can be represented as

$$\begin{cases} \mathbf{x}_d(k+1) = \mathbf{A}_d \mathbf{x}_d(k) + \mathbf{B}_d u(k), \\ y(k) = \mathbf{C}_d \mathbf{x}_d(k) + D_d u(k), \end{cases} \quad (3)$$

where $\mathbf{x}_d(k) \in \mathbb{R}^{n_d}$ is the state vector of the model (3), and $u(k) \in \mathbb{R}$ and $y(k) \in \mathbb{R}$ are the input and output of the system G_d , respectively.

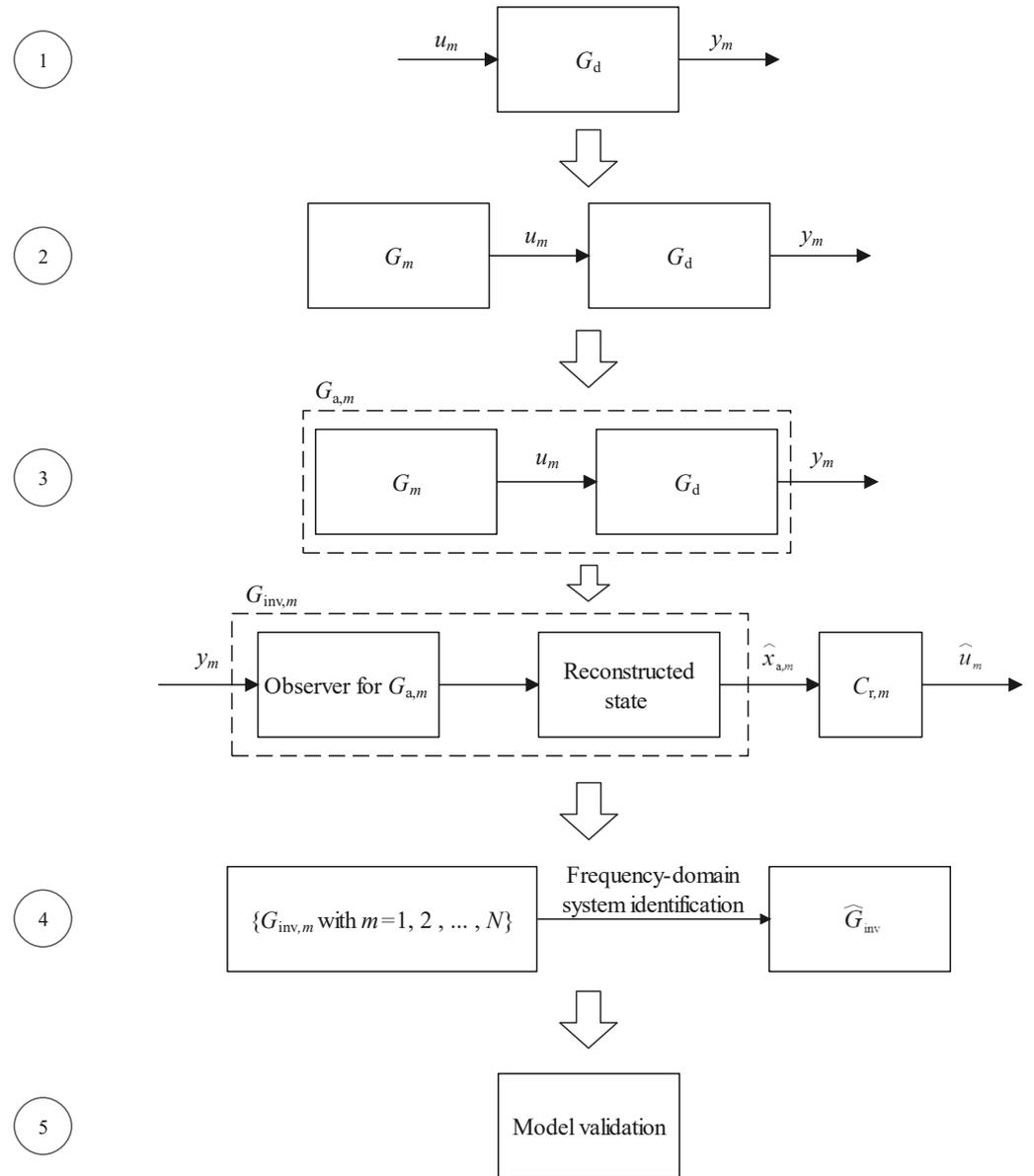


Figure 2. Flowchart of the proposed inverse system identification approach. (The symbol “^” denotes the reconstructed or the estimated value).

Step 2: A set S containing N discrete-time sine signals is given as:

$$S = \{u_m(k), m = 1, 2, \dots, N\}, \tag{4}$$

where

$$u_m(k) = \sin(2\pi f_m k T_s), \tag{5}$$

where f_m denotes the frequency in Hz, and the sequence $(f_m)_{m=1}^N$ is an arithmetic sequence, each sine wave in the set S can be represented as the output of a state-space model G_m , i.e.,

$$\begin{cases} \mathbf{x}_m(k+1) = \mathbf{A}_m \mathbf{x}_m(k), \\ u_m(k) = \mathbf{C}_m \mathbf{x}_m(k), \end{cases} \tag{6}$$

where $x_m(k) \in \mathbb{R}^2$ denotes the state vector of the model (6), and the matrices A_m and C_m can be denoted as

$$A_m = \begin{pmatrix} \cos(2\pi f_m T_s) & \sin(2\pi f_m T_s) \\ -\sin(2\pi f_m T_s) & \cos(2\pi f_m T_s) \end{pmatrix} \tag{7}$$

and

$$C_m = (1 \ 0), \tag{8}$$

respectively.

Remark 1. The frequencies of the signals $u_m(k)$ for $m = 1, 2, \dots, N$ in the set S can be specified by the following rule:

$$f_m = f_b + (m - 1)d, \tag{9}$$

where f_b is a non-negative value, and d is a positive value.

The rule in Equation (9) is not the only way to specify the frequencies.

The values of f_m for $m = 1, 2, \dots, N$ belong to the range $(0, \frac{f_s}{2})$ with $f_s = \frac{1}{T_s}$.

Step 3: If the signal $u_m(k)$ is used as the input signal of the model (3), we can obtain

$$\begin{cases} x_d(k+1) = A_d x_d(k) + B_d u_m(k), \\ y_m(k) = C_d x_d(k) + D_d u_m(k), \end{cases} \tag{10}$$

where $y_m(k)$ denotes the output of the model (3) when the input signal is $u_m(k)$.

By augmenting the model (10) with the state vector of the model (6), an augmented model $G_{a,m}$ can be obtained, and it can be represented as

$$\begin{cases} x_{a,m}(k+1) = A_{a,m} x_{a,m}(k), \\ y_m(k) = C_{a,m} x_{a,m}(k), \end{cases} \tag{11}$$

where the state vector $x_{a,m}(k)$ can be denoted as

$$x_{a,m}(k) = \begin{pmatrix} x_d(k) \\ x_m(k) \end{pmatrix}, \tag{12}$$

and the matrices $A_{a,m}$ and $C_{a,m}$ can be denoted as

$$A_{a,m} = \begin{pmatrix} A_d & B_d C_m \\ \mathbf{0} & A_m \end{pmatrix} \tag{13}$$

and

$$C_{a,m} = (C_d \ D_d C_m), \tag{14}$$

respectively.

Based on the state-space model representation (11) for the augmented models $G_{a,m}$, $m = 1, 2, \dots, N$, we can totally build N full-order observers corresponding to the augmented models $G_{a,m}$, $m = 1, 2, \dots, N$, and we denote the m -th observer for the m -th augmented model $G_{a,m}$ as $E_{a,m}$, which can be described by

$$\hat{x}_{a,m}(k+1) = (A_{a,m} - L_{a,m} C_{a,m}) \hat{x}_{a,m}(k) + L_{a,m} y_m(k), \tag{15}$$

where $L_{a,m}$ denotes the observer gain, and $\hat{x}_{a,m}(k)$ denotes the reconstructed value of $x_{a,m}(k)$ using the observer.

With the reconstructed value $\hat{x}_{a,m}(k)$, we can obtain the reconstructed value of $u_m(k)$, which can be calculated by the following equation:

$$\hat{u}_m(k) = C_{r,m}\hat{x}_{a,m}(k), \tag{16}$$

where

$$C_{r,m} = (0 \times C_d \quad C_m). \tag{17}$$

By combining (15) with (16), we can obtain a state-space model

$$\begin{cases} \hat{x}_{a,m}(k+1) = (A_{a,m} - L_{a,m}C_{a,m})\hat{x}_{a,m}(k) + L_{a,m}y_m(k), \\ \hat{u}_m(k) = C_{r,m}\hat{x}_{a,m}(k), \end{cases} \tag{18}$$

which can be regarded as a reconstructor of the input $u_m(k)$ of the model (10).

Let the model (18) be denoted as $G_{inv,m}$, and the value of the frequency response function of the model $G_{inv,m}$ at the frequency f_m can then be represented as $G_{inv,m}(e^{-j\Omega_m T_s})$ where $\Omega_m = 2\pi f_m$.

Step 4: Let the inverse model of the model G_d be denoted as G_{inv} , and with the frequency response function values $G_{inv,m}(e^{-j\Omega_m T_s})$, $m = 1, 2, \dots, N$, the inverse model G_{inv} in state-space representation can be identified by using the subspace-based system identification method in the frequency domain [23]. After system identification, the identified inverse model is denoted as \hat{G}_{inv} .

Remark 2. The effective frequency range of the inverse model G_{inv} can be specified by selecting the range of the frequencies of the signals $u_m(k)$, $m = 1, 2, \dots, N$, in the set S .

Step 5: Connect the models G_d and \hat{G}_{inv} in series, and the resulting model can be represented as

$$G_s = \hat{G}_{inv}G_d \tag{19}$$

of which the frequency response function is written as

$$G_s(e^{-j\Omega T_s}) = |G_s(e^{-j\Omega T_s})|e^{j\angle G_s(e^{-j\Omega T_s})}, \tag{20}$$

where $\Omega = 2\pi f$ with f the ordinary frequency in Hz, so if the identified inverse model \hat{G}_{inv} is perfect, the frequency response of the model G_s should satisfy:

- (a) $|G_s(e^{-j\Omega T_s})| = 1$.
- (b) $\angle G_s(e^{-j\Omega T_s}) \in \{\theta \mid \theta = 2k\pi \text{ with } k = 0, 1, 2, \dots\}$.

By observing whether the frequency response function of the model G_s within the frequency range specified by inverse system identification satisfies the above conditions (a) and (b) and to what extent it satisfies them, the goodness of the identified inverse model \hat{G}_{inv} can be validated.

2.2. MATLAB Codes

The corresponding MATLAB (R2020b) commands to realize the above mentioned inverse identification approach are illustrated step by step.

Step 1: Firstly the transfer function of a nominal model G_d is given, then a state-space realization of the transfer function is made, afterwards the Bode plot is plotted. The process can be realized by following MATLAB codes.

```
> systf=tf(numerator,denominator,Ts);
> sysss=ss(systf);
> figure;
```

```

> opts=bodeoptions;
> opts.FreqUnits='Hz';
> bode(syss,opts);

```

Step 2: According to the bandwidth of the nominal model G_d , specify the frequencies f_m , $m = 1, 2, \dots, N$, by using the rule stated in Remarks 1 and 2, and then stack all the frequencies into the vector F_N , i.e.,

$$F_N = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}. \quad (21)$$

Based on Equations (7) and (8), create the matrices A_m and C_m for $m = 1, 2, \dots, N$ in the models of the N discrete-time sine signals in the set S , and then stack them into the matrices A_N and C_N , respectively, i.e.,

$$A_N = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_N \end{pmatrix}, \quad (22)$$

and

$$C_N = \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_N \end{pmatrix}. \quad (23)$$

We can obtain the following MATLAB codes for the above.

```

> dim=N;
> FN=zeros(dim,1);
> for m=1:dim
    FN(m,:)=fb+(m-1)*d;
end
> AN=zeros(2*dim,2);
> for m=1:dim
    AN(((2*m-1):2*m),:)= [cos(2*pi*FN(m,:)*Ts) sin(2*pi*FN(m,:)*Ts)
                        -sin(2*pi*FN(m,:)*Ts) cos(2*pi*FN(m,:)*Ts)];
end
> CN=zeros(dim,2);
> for m=1:dim
    CN(m,:)= [1 0];
end

```

Step 3: With the created matrices A_m and C_m for $m = 1, 2, \dots, N$, create the matrices $A_{a,m}$, $C_{a,m}$, and $C_{r,m}$ for $m = 1, 2, \dots, N$ by using Equation (13), Equation (14), and

Equation (17), respectively. Then stack the matrices $A_{a,m}$ and $C_{a,m}$, $m = 1, 2, \dots, N$, into the matrices A_N^a , C_N^a , and C_N^r , respectively, i.e.,

$$A_N^a = \begin{pmatrix} A_{a,1} \\ A_{a,2} \\ \vdots \\ A_{a,N} \end{pmatrix}, \tag{24}$$

$$C_N^a = \begin{pmatrix} C_{a,1} \\ C_{a,2} \\ \vdots \\ C_{a,N} \end{pmatrix}, \tag{25}$$

and

$$C_N^r = \begin{pmatrix} C_{r,1} \\ C_{r,2} \\ \vdots \\ C_{r,N} \end{pmatrix}. \tag{26}$$

Based on the expression of the model (18), calculate the transfer function of the reconstructors $G_{inv,m}$ by using

$$G_{inv,m}(z) = C_{r,m}[zI - (A_{a,m} - L_{a,m}C_{a,m})]^{-1}L_{a,m}, \tag{27}$$

where the observer gain $L_{a,m}$ can be chosen in a linear least-squares sense for stochastic systems, i.e., the observer gain $L_{a,m}$ can be obtained as the gain of the steady-state Kalman filter for the model (11) with process noise and measurement noise or can be obtained in a minimum mean-integral squared error sense [24]. Denote the process noise and measurement noise as $w(k) \in \mathbb{R}^{n_a}$ with $n_a = n_d + 2$ and $v(k) \in \mathbb{R}$, respectively, and assume that both $\{w(k), k = 1, 2, \dots\}$ and $\{v(k), k = 1, 2, \dots\}$ are white Gaussian sequences, $w(k) \sim N(0, Q)$ with $Q > 0$, $v(k) \sim N(0, R)$ with $R > 0$, and assume that the distribution of $x_{a,m}(0)$ is Gaussian, and assume that $\{w(k), k = 1, 2, \dots\}$ and $\{v(k), k = 1, 2, \dots\}$ are uncorrelated with $x_{a,m}(0)$ and with each other. Then, derive the gain of the steady-state Kalman filter for the model (11) by using the following equation [25]:

$$L_{a,m} = A_{a,m}P_m C_{a,m}^T (C_{a,m}P_m C_{a,m}^T + R)^{-1}, \tag{28}$$

where the value of P_m can be derived as the unique solution of the following algebraic Riccati equation:

$$P_m = A_{a,m}[P_m - P_m C_{a,m}^T (C_{a,m}P_m C_{a,m}^T + R)^{-1}C_{a,m}P_m]A_{a,m}^T + Q, \tag{29}$$

under the following conditions:

- (a) $(A_{a,m}, C_{a,m})$ is detectable. (A system is detectable if all the unobservable states are stable)
- (b) $(A_{a,m}, Q)$ is controllable.

Then, stack all the calculated observer gains $L_{a,m}$ for $m = 1, 2, \dots, N$ into the matrix L_N , i.e.,

$$L_N = (L_{a,1} \quad L_{a,2} \quad \dots \quad L_{a,N}). \tag{30}$$

After calculating the reconstructor transfer functions $G_{\text{inv},m}(z)$ for $m = 1, 2, \dots, N$ based on (27), stack all the obtained transfer functions into the vector function E_N , which can be denoted as

$$E_N = \begin{pmatrix} G_{\text{inv},1}(z) \\ G_{\text{inv},2}(z) \\ \vdots \\ G_{\text{inv},N}(z) \end{pmatrix}. \quad (31)$$

Obtain the frequency response function values $G_{\text{inv},m}(e^{-j\Omega_m T_s})$ for $m = 1, 2, \dots, N$ by replacing z in Equation (27) with $e^{-j\Omega_m T_s}$ for $m = 1, 2, \dots, N$, then stack all the frequency response function values into the vector G_N , i.e.,

$$G_N = \begin{pmatrix} G_{\text{inv},1}(e^{-j\Omega_1 T_s}) \\ G_{\text{inv},2}(e^{-j\Omega_2 T_s}) \\ \vdots \\ G_{\text{inv},N}(e^{-j\Omega_N T_s}) \end{pmatrix}. \quad (32)$$

The above process can be realized by the following MATLAB codes.

```

> r1=size(syssss.a,1)+2;
> AaN=zeros(r1*dim,r1);
> for m=1:dim
    r2=1+(m-1)*r1;
    AaN(r2:r1*m,:)= [syssss.a syssss.b*CN(m,:);
                    zeros(2,size(syssss.a,1)) AN(((2*m-1):2*m),:)]];
    end
> CaN=zeros(dim,r1);
> for m=1:dim
    CaN(m,:)= [syssss.c syssss.d*CN(m,:)]];
    end
> CrN=zeros(dim,r1);
> for m=1:dim
    CrN(m,:)= [zeros(1,size(syssss.a,1)) CN(m,:)]];
    end
> LN=zeros(r1,dim);
> for m=1:dim
    Q=pc*eye(r1);
    R=mc;
    r2=1+(m-1)*r1;
    LN(:,m)=dlqr(AaN(r2:r1*m,:)',CaN(m,:)',Q,R)';
    end
> g=cell(dim,1);
> GN=zeros(dim,1);
> for m=1:dim
    r2=1+(m-1)*r1;
    sysr=ss(AaN(r2:r1*m,:)-LN(:,m)*CaN(m,:),LN(:,m),CrN(m,:),0,Ts);
    if isstable(sysr)==1
        g{m,:}=sysr;
        GN(m,:)=frd(g{m,:},FN(m,:), 'Hz').ResponseData;
    else
        break
    end
end

```

Remark 3. The values of Q and R can be tuned by changing the values of pc and mc .

Step 4: With the calculated frequency response function values $G_{inv,m}(e^{-j\Omega_m T_s})$, $m = 1, 2, \dots, N$, the inverse model G_{inv} in state-space representation can be identified by using the MATLAB function `n4sid`, then the Bode plot of the identified inverse model can be obtained.

```

> fdata=idfrd(GN,2*pi*FN,Ts);
> opt=n4sidOptions("EnforceStability",1);
> Ginv=n4sid(fdata,nx,'Ts',Ts,opt);
> figure;
> opts=bodeoptions;
> opts.FreqUnits='Hz';
> bode(Ginv,opts);

```

Remark 4. The values of nx denote the inverse model order which can be specified.

Step 5: The following MATLAB command can be used for the series connection of the models G_d and \hat{G}_{inv} .

```

> Gs=series(syssss,Ginv);

```

Then, the Bode plot of the combined model can be displayed using the following codes.

```

> figure;
> opts=bodeoptions;
> opts.FreqUnits='Hz';
> bode(Gs,opts);

```

2.3. Inverse System Identification Toolbox Creation

Building the inverse system identification toolbox consists of two parts.

Part 1: Based on the MATLAB codes of inverse identification obtained in Section 2.2, a MATLAB function file, which is an m-file, can be created. The specific content of the m-file is given in Listing 1.

Listing 1. MATLAB function of inverse system identification.

```

1 function Ginv = INVSIDToolbox(numerator,denominator,Ts,fb,
2   d,N,pc,mc,nx)
3 % numerator and denominator: The numerator and denominator
4   coefficients of the transfer function of the nominal
5   model G_d.
6 % Ts: The sampling period of the nominal model G_d.
7 % fb: The smallest frequency among the frequency
8   components for inverse system identification.
9 % d: The common difference.
10 % N: The number of the frequency components for inverse
11   system identification.
12 % pc: The covariance of the process noise.
13 % mc: The covariance of the measurement noise.
14 % nx: Vector of model orders to scan.
15 % Ginv: The identified inverse model.

```

```

12 %% Step I
13 systf=tf(numerator,denominator,Ts);
14 sysss=ss(systf);
15 figure;
16 opts=bodeoptions;
17 opts.FreqUnits='Hz';
18 bode(sysss,opts);
19
20 %% Step II
21 dim=N;
22 FN=zeros(dim,1);
23 for m=1:dim
24 FN(m,:)=fb+(m-1)*d;
25 end
26 AN=zeros(2*dim,2);
27 for m=1:dim
28 AN(((2*m-1):2*m),:)= [cos(2*pi*FN(m,:)*Ts) sin(2*pi*FN(m,:)*Ts)
    *Ts)
    -sin(2*pi*FN(m,:)*Ts) cos(2*pi*FN(m,:)*Ts)];
29 end
30 CN=zeros(dim,2);
31 for m=1:dim
32 CN(m,:)=[1 0];
33 end
34
35
36 %% Step III
37 r1=size(sysss.a,1)+2;
38 AaN=zeros(r1*dim,r1);
39 for m=1:dim
40 r2=1+(m-1)*r1;
41 AaN(r2:r1*m,:)= [sysss.a sysss.b*CN(m,:);
42 zeros(2,size(sysss.a,1)) AN(((2*m-1):2*m),:)]];
43 end
44 CaN=zeros(dim,r1);
45 for m=1:dim
46 CaN(m,:)= [sysss.c sysss.d*CN(m,:)]];
47 end
48 CrN=zeros(dim,r1);
49 for m=1:dim
50 CrN(m,:)= [zeros(1,size(sysss.a,1)) CN(m,:)]];
51 end
52 LN=zeros(r1,dim);
53 for m=1:dim
54 Q=pc*eye(r1);
55 R=mc;
56 r2=1+(m-1)*r1;
57 LN(:,m)=dlqr(AaN(r2:r1*m,:),CaN(m,:),Q,R)';
58 end
59 g=cell(dim,1);
60 GN=zeros(dim,1);
61 for m=1:dim
62 r2=1+(m-1)*r1;
63 sysr=ss(AaN(r2:r1*m,:)-LN(:,m)*CaN(m,:),LN(:,m),CrN(m,:),
    ,0,Ts);

```

```

64 if isstable(sysr)==1
65 g{m,:}=sysr;
66 GN(m,:)=frd(g{m,:},FN(m,:), 'Hz').ResponseData;
67 else
68 break
69 end
70 end
71
72 %% Step IV
73 fdata=idfrd(GN,2*pi*FN,Ts);
74 opt=ssestOptions("EnforceStability",1);
75 Ginv=ssest(fdata,nx,'Ts',Ts,opt);
76 figure
77 opts=bodeoptions;
78 opts.FreqUnits='Hz';
79 bode(Ginv,opts);
80
81 %% Step V
82 Gs=series(sysss,Ginv);
83 figure;
84 opts=bodeoptions;
85 opts.FreqUnits='Hz';
86 bode(Gs,opts);
87 end

```

Part 2: With the MATLAB m-file created in the first part, the inverse system identification toolbox INVSID 1.0 can be installed. The complete installation procedure contains five steps, from the first step about the selection of the item Package Toolbox from the Add-Ons menu to the end step about saving the created toolbox (Installation procedure of MATLAB toolboxes can be referred to the following link: https://www.mathworks.com/help/matlab/matlab_prog/create-and-share-custom-matlab-toolboxes.html).

3. Numerical Studies

3.1. Pure Simulation Examples

In this section, two simulation examples are used to validate the effectiveness of the toolbox INVSID 1.0, i.e., check the effectiveness of the proposed inverse system identification approach.

Firstly, consider a discrete-time, single-input single-output, linear, constant dynamical system G_d , of which the transfer function is described by

$$G_d(z) = \frac{z}{z^2 - 1.5z + 0.7} \quad (33)$$

with the sampling period $T_s = 1 \times 10^{-5}$ seconds.

The Bode plot of the system G_d is displayed in Figure 3.

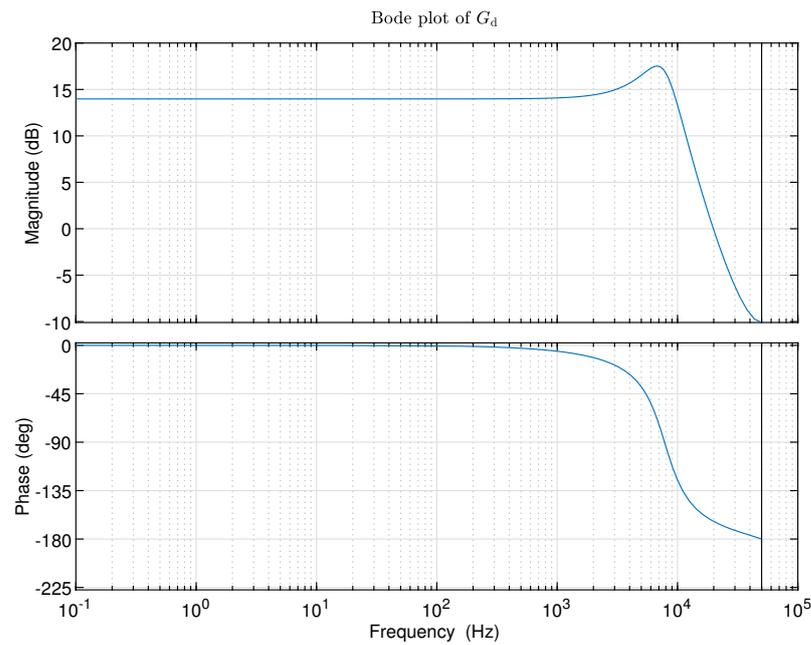


Figure 3. Bode plot of G_d .

According to the transfer function (33), the following observations can be made:

- (a) G_d is stable.
- (b) G_d is proper.
- (c) G_d is minimal-realized.
- (d) G_d has one nonminimum-phase zero.

According to the above observations, it can be known that the direct inversion of the system G is a challenging problem. So we now turn to using the proposed toolbox INVSID 1.0 to identify the inverse model of the system G . The proposed inverse identification approach can specify the frequency range of interest, i.e., by selecting the values of f_b , m , and d in Equation (9), the frequency range to be identified can be determined. The parameters shown in Table 1 are used as the inputs of the inverse identification toolbox.

Table 1. Parameters for inverse identification.

Parameter	Value in MATLAB
numerator	[0, 1, 0]
denominator	[1, -1.5, 0.7]
T_s	1×10^{-5}
f_b	10
d	10
N	50
pc	1×10^{-3}
mc	1×10^{-3}
nx	2:10

With the above inputs, the final output of the inverse identification toolbox is the identified inverse model which is the best model corresponding to the recommended singular value. The model order of the identified inverse model \hat{G}_{inv} is recommended to be 4. The frequency response properties of the model \hat{G}_{inv} with fourth order are demonstrated in Figure 4.

In addition, the inversion \hat{G}_{inv} is identified using the MATLAB function `n4sid` with stability enforcement, so the identified model \hat{G}_{inv} is stable and causal.

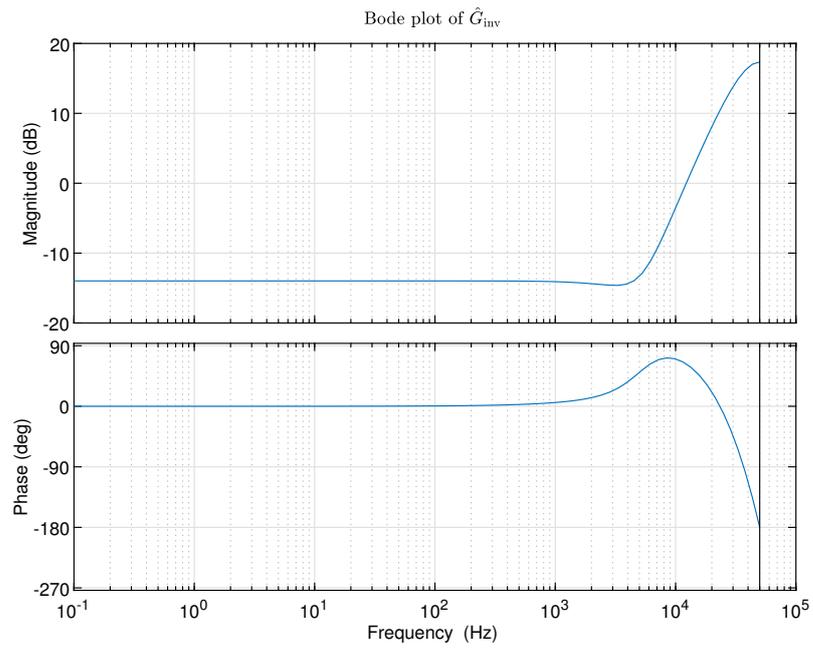


Figure 4. Bode plot of \hat{G}_{inv} .

By connecting the model G_d and the model \hat{G}_{inv} in series using Equation (19), the model G_s can be obtained. The obtained model G_s can then be used for validating the goodness of the identified inverse model \hat{G}_{inv} in the frequency range of interest. The frequency response of the obtained model \hat{G}_{inv} is shown in Figure 5; in the specified frequency range from 10 Hz to 500 Hz, the magnitude is nearly a constant near 0 dB, and the phase is nearly a constant around 0 degrees. The values of magnitude and phase can indicate the effectiveness of the proposed inverse identification toolbox for stable systems to be inverted.

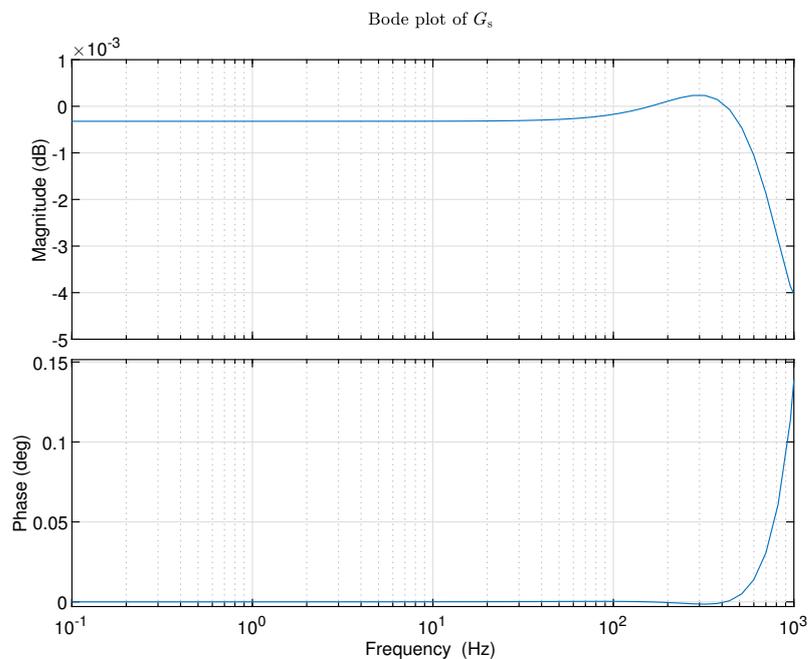


Figure 5. Bode plot of G_s .

In practice, unstable systems are also frequently encountered. So the second numerical example is about using the toolbox INVSID 1.0 to solve the system inversion problem of an unstable system.

Consider a discrete-time, single-input single-output, linear, constant dynamical system G_d^* , of which the transfer function is described by

$$G_d^*(z) = \frac{z}{z^2 - 5z + 6} \tag{34}$$

with the sampling period $T_s = 1 \times 10^{-5}$ s.

The Bode plot of the system G_d^* is displayed in Figure 6.

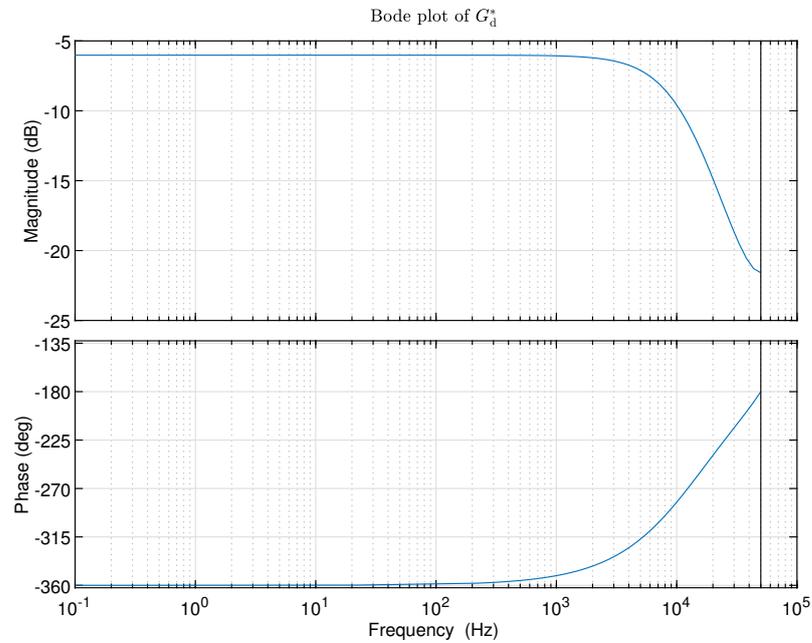


Figure 6. Bode plot of G_d^* .

According to the transfer function (34), the following observations can be made:

- (a) G_d^* is unstable.
- (b) G_d^* is proper.
- (c) G_d^* is minimal-realized.
- (d) G_d^* has one nonminimum-phase zero.

The parameters shown in Table 2 are used as the inputs of the inverse identification toolbox.

Table 2. Parameters for inverse identification.

Parameter	Value in MATLAB
numerator	[0, 1, 0]
denominator	[1, -5, 6]
T_s	1×10^{-5}
f_b	10
d	10
N	50
pc	1×10^{-3}
mc	1×10^{-3}
nx	2:10

With the above inputs, the final output of the inverse identification toolbox is the identified inverse model which is the best model corresponding to the recommended singular value. The model order of the identified inverse model \hat{G}_{inv}^* is recommended to be 4. The frequency response properties of the model \hat{G}_{inv}^* with fourth order are demonstrated in Figure 7.

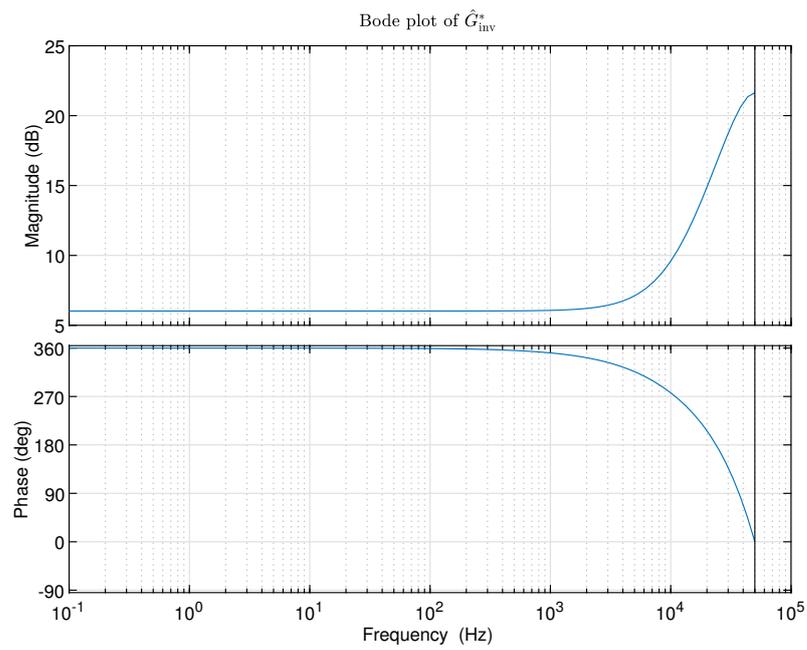


Figure 7. Bode plot of \hat{G}_{inv}^* .

In addition, the inversion \hat{G}_{inv}^* is identified using the MATLAB function `n4sid` with stability enforcement, so the identified model \hat{G}_{inv}^* is stable and causal.

By connecting the model G_d^* and the model \hat{G}_{inv}^* in series using Equation (19), the model G_s^* can be obtained. The frequency response of the obtained model \hat{G}_{inv}^* is shown in Figure 8; in the specified frequency range from 10 Hz to 500 Hz, the magnitude is nearly a constant near 0 dB, and the phase is nearly a constant around 0 degrees. The values of magnitude and phase can indicate the effectiveness of the proposed inverse identification toolbox for unstable systems to be inverted.

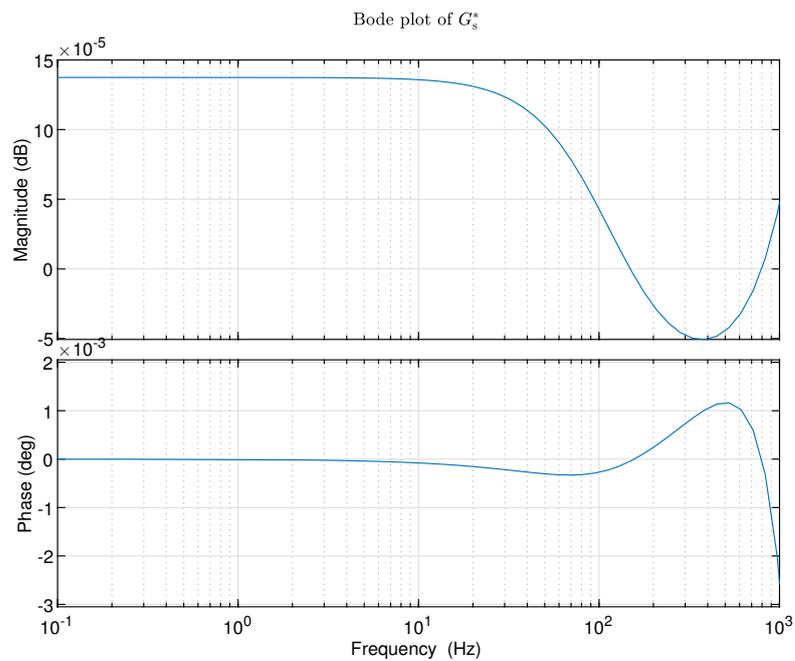


Figure 8. Bode plot of G_s^* .

3.2. Practical Hard Disk Drive System

Figure 9 shows a block diagram to illustrate the goal of the system inversion design, where $r(k) \in \mathbb{R}$, $u(k) \in \mathbb{R}$, and $y(k) \in \mathbb{R}$ represent the discrete-time reference, the input, and the output signals, respectively. Note that F can be implemented as a block in the feedforward controller design. In Figure 9, the overall transfer function from the reference signal $r(k)$ to the output signal $y(k)$ is

$$\frac{Y(z)}{R(z)} = F(z)G(z) = \hat{G}_{\text{inv}}(z)G(z) = G_s(z), \tag{35}$$

where $F(z) = \hat{G}_{\text{inv}}$ denotes the inversion of the system $G(z)$. Equation (35) can reflect the accuracy of the inversion $F(z)$.

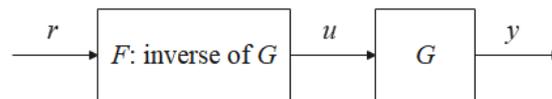


Figure 9. Block diagram to illustrate the goal of the system inversion design. Note that F can be implemented as a feedforward controller.

We take the hard disk drive system as an illustrative example [26,27]. The transfer function of the system with a sampling frequency of 26.4 kHz is

$$G(z) = z^{-3} \frac{1.447663(z + 0.050852)(z + 2.494311)}{z^2 - 1.978354z + 0.978808}. \tag{36}$$

The Bode plot of the system G is shown in Figure 10.

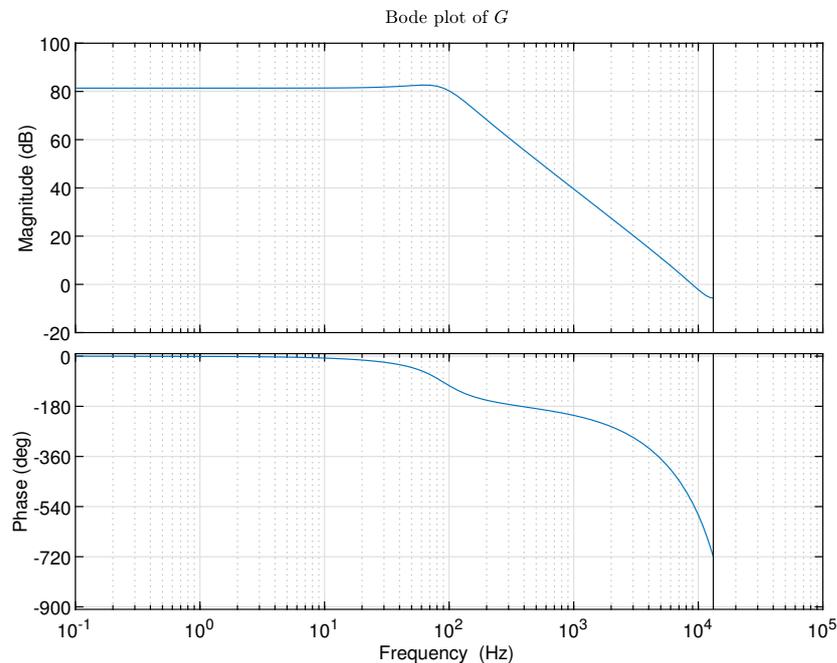


Figure 10. Bode plot of G .

Based on the transfer function (36), the following observations can be made for the system G :

- (a) G is stable.
- (b) G is proper.
- (c) G is minimal-realized.

(d) G has one nonminimum-phase zero at around -2.5 .

The parameters shown in Table 3 are used as the inputs of the inverse identification toolbox.

Table 3. Parameters for inverse identification.

Parameter	Value in MATLAB
numerator	[0, 0, 0, 1.4477, 3.6845, 0.1836]
denominator	[1, -1.9784, 0.9788, 0, 0, 0]
T_s	1/26,400
f_b	0.1
d	0.1
N	7000
pc	1×10^{-3}
mc	1×10^{-3}
nx	2:10

With the above inputs, the final output of the inverse identification toolbox is the identified inverse model, of which the model order is chosen as 10. The frequency response properties of the model \hat{G}_{inv} with tenth order are demonstrated in Figure 11.

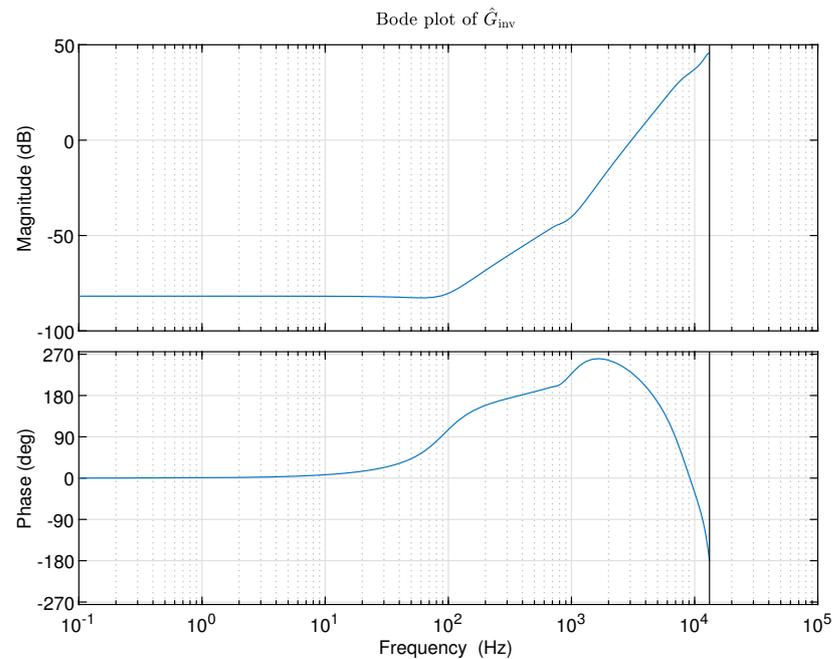


Figure 11. Bode plot of \hat{G}_{inv} .

The inversion \hat{G}_{inv} is identified using the MATLAB function `n4sid` with stability enforcement, so the identified model \hat{G}_{inv} is stable and causal.

As shown in Figure 9, by connecting the model G and the model \hat{G}_{inv} in series using Equation (19), the model G_s can be obtained. The frequency response of the obtained model \hat{G}_{inv}^* is shown in Figure 12; in the specified frequency range from 0.1 Hz to 700 Hz, the magnitude is nearly a constant near 0 dB, and the phase is nearly a constant around 0 degrees. The above results can indicate that the proposed system inversion approach is effective in the practical application in the feedforward controller design of the hard disk drive system.

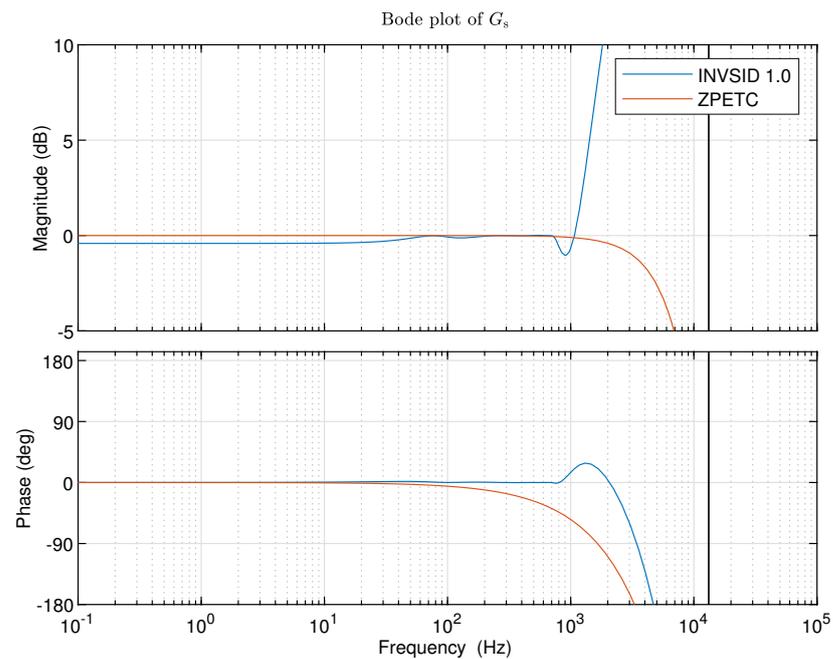


Figure 12. Bode plot of G_s . ZPETC: zero phase error tracking algorithm.

Furthermore, the well-known zero phase error tracking algorithm [11] is also implemented into the feedforward controller design, and according to the comparison between INVSID 1.0 and zero phase error tracking algorithm. For this example, it can be concluded that within the specified frequency range, INVSID 1.0 performs better than the zero phase error tracking algorithm in terms of phase, and in addition, INVSID 1.0 can be implemented without any preview.

4. Conclusions and Perspectives

In this paper, an alternative system inversion approach is proposed, based on which the toolbox named INVSID 1.0 is developed. The advantages of the toolbox INVSID 1.0 can be concluded as follows:

- (a) The proposed inverse identification toolbox can be used for stable or unstable systems, minimum-phase or nonminimum-phase systems.
- (b) Preview is not needed, i.e., the causality of the identified inverse model can be guaranteed.
- (c) Stability of the identified inverse model can be guaranteed.
- (d) The frequency range of interest can be specified.
- (e) Subspace identification is used such that there is no non-convex problem.

Furthermore, according to the theoretical derivation of the proposed system inversion approach, it can be indicated that the proposed approach can be used for systems with noise, because an observer is involved in the approach.

Currently, the inverse identification toolbox INVSID 1.0 is used for single-input single-output systems, while in the future, the proposed inverse system identification approach will be extended to identify the inverse models of general multiple-input multiple-output systems such that more advanced versions of the INVSID toolbox can be created, and squaring down approaches [28] have a potential to solve the extension problem.

Author Contributions: Conceptualization, R.H.; methodology, R.H.; software, R.H.; validation, R.H.; formal analysis, R.H.; investigation, R.H.; resources, R.H.; data curation, R.H.; writing—original draft preparation, R.H.; writing—review and editing, R.H., C.B. and G.B.; visualization, R.H.; supervision, R.H.; project administration, R.H.; funding acquisition, R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Zhejiang province, P.R. China (Grant No. LQ23F030005) and the University Research Development Foundation in Zhejiang A&F University, P.R. China (Grant No. 203402000601).

Data Availability Statement: The original contributions presented in the 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.

References

1. Goodwin, G.C.; Graebe, S.F.; Salgado, M.E. *Control System Design*; Prentice Hall: Upper Saddle River, NJ, USA, 2001.
2. Golub, G.H.; Van Loan, C.F. *Matrix Computations*; John Hopkins University Press: Baltimore, MD, USA, 2012.
3. Jung, Y. *Inverse System Identification with Applications in Predistortion*; Linköping University: Linköping, Sweden, 2018.
4. van Zundert, J.; Oomen, T. On inversion-based approaches for feedforward and ILC. *Mechatronics* **2018**, *50*, 282–291. [[CrossRef](#)]
5. Zhou, K.; Doyle, J.C.; Glover, K. *Robust and Optimal Control*; Prentice Hall: Englewood Cliffs, NJ, USA, 1996.
6. Rigney, B.P.; Pao, L.Y.; Lawrence, D.A. Nonminimum phase dynamic inversion for settle time applications. *IEEE Trans. Control Syst. Technol.* **2009**, *17*, 989–1005. [[CrossRef](#)]
7. Devasia, S.; Chen, D.; Paden, B. Nonlinear inversion-based output tracking. *IEEE Trans. Autom. Control* **1996**, *41*, 930–942. [[CrossRef](#)]
8. Hunt, L.; Meyer, G.; Su, R. Noncausal inverses for linear systems. *IEEE Trans. Autom. Control* **1996**, *41*, 608–611. [[CrossRef](#)]
9. Widrow, B.; Walach, E. *Adaptive Inverse Control: A Signal Processing Approach*; John Wiley & Sons: Hoboken, NJ, USA, 2008.
10. Sogo, T. On the equivalence between stable inversion for nonminimum phase systems and reciprocal transfer functions defined by the two-sided Laplace transform. *Automatica* **2010**, *46*, 122–126. [[CrossRef](#)]
11. Tomizuka, M. Zero phase error tracking algorithm for digital control. *J. Dyn. Syst. Meas. Control* **1987**, *109*, 65–68. [[CrossRef](#)]
12. Gross, E.; Tomizuka, M.; Messner, W. Cancellation of discrete time unstable zeros by feedforward control. *J. Dyn. Syst. Meas. Control* **1994**, *116*, 33–38. [[CrossRef](#)]
13. Roover, D.D.; Bosgra, O.H. Synthesis of robust multivariable iterative learning controllers with application to a wafer state motion system. *Int. J. Control* **2000**, *73*, 968–979. [[CrossRef](#)]
14. Mirkin, L. On the H_∞ fixed-lag smoothing: How to exploit the information preview. *Automatica* **2003**, *39*, 1495–1504. [[CrossRef](#)]
15. Hazell, A.; Limebeer, D.J.N. An efficient algorithm for discrete-time H_∞ preview control. *Automatica* **2008**, *44*, 2441–2448. [[CrossRef](#)]
16. Zhang, Y.; Liu, S. Pre-actuation and optimal state to state transition based precise tracking for maximum phase system. *Asian J. Control* **2016**, *18*, 1728–1738. [[CrossRef](#)]
17. Zhu, Q.; Zhang, Y.; Xiong, R. Stable inversion based precise tracking for periodic systems. *Asian J. Control* **2020**, *22*, 217–227. [[CrossRef](#)]
18. Zou, Q. Optimal preview-based stable-inversion for output tracking of nonminimum-phase linear systems. *Automatica* **2020**, *45*, 230–237. [[CrossRef](#)]
19. Jetto, L.; Orsini, V.; Romagnoli, R. Accurate output tracking for nonminimum phase nonhyperbolic and near nonhyperbolic systems. *Eur. J. Control* **2014**, *20*, 292–300. [[CrossRef](#)]
20. Romagnoli, R.; Garone, E. A general framework for approximated model stable inversion. *Automatica* **2019**, *101*, 182–189. [[CrossRef](#)]
21. Bohn, C.; Cortabarría, A.; Härtel, V.; Kowalczyk, K. Active control of engine-induced vibrations in automotive vehicles using disturbance observer gain scheduling. *Control Eng. Pract.* **2004**, *12*, 1029–1039. [[CrossRef](#)]
22. Han, R.; Bohn, C.; Bauer, G. Comparison between calculated speed-based and sensor-fused speed-based engine in-cylinder pressure estimation method. *IFAC-PapersOnLine* **2022**, *55*, 223–228. [[CrossRef](#)]
23. McKelvey, T.; Akçay, H.; Ljung, L. Subspace-based multivariable system identification from frequency response data. *IEEE Trans. Autom. Control* **1996**, *41*, 960–979. [[CrossRef](#)]
24. O'Reilly, J. *Observers for Linear Systems*; Academic Press: London, UK, 1983.
25. Lewis, F.L.; Xie, L.; Popa, D. *Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory*; CRC Press: Boca Raton, FL, USA, 2008.
26. Chen, X.; Tomizuka, M. A minimum parameter adaptive approach for rejecting multiple narrow-band disturbances with application to hard disk drives. *IEEE Trans. Control Syst. Technol.* **2011**, *20*, 408–415. [[CrossRef](#)]
27. Wang, D.; Chen, X. H_∞ -based selective inversion of nonminimum-phase systems for feedback controls. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 702–710. [[CrossRef](#)]
28. Saberi, A.; Sannuti, P. Squaring down by static and dynamic compensators. *IEEE Trans. Autom. Control* **1988**, *33*, 358–365. [[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.