

---

Article

Not peer-reviewed version

---

# INVSID 1.0: An Inverse System Identification Toolbox for MATLAB

---

[Runzhe Han](#) <sup>\*</sup>, Georg Bauer , Christian Bohn

Posted Date: 11 June 2024

doi: [10.20944/preprints202406.0681.v1](https://doi.org/10.20944/preprints202406.0681.v1)

Keywords: System inversion; Inverse system identification; MATLAB toolbox



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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.

Article

# INVSID 1.0: An Inverse System Identification Toolbox for MATLAB

Runzhe Han <sup>1,\*</sup>, Christian Bohn <sup>2</sup>, Georg Bauer <sup>2</sup>

<sup>1</sup> College of Optical, Mechanical and Electrical Engineering, Zhejiang A&F University, 311300 Hangzhou, P.R. China

<sup>2</sup> Institut für Elektrische Informationstechnik, Technische Universität Clausthal, 38678 Clausthal-Zellerfeld, Germany; bohn@iei.tu-clausthal.de, bauer@iei.tu-clausthal.de

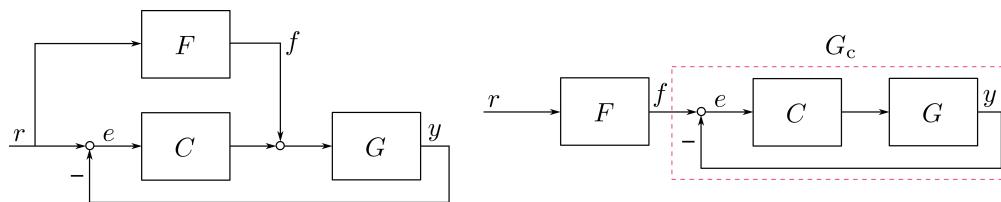
\* Correspondence: runzhe.han@alumni.tu-clausthal.de; Tel.: +86-19558180976

**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, the 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, totally 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 exists several kinds of intrinsic limitations of direct inversion approaches, here an example is used to illustrate this, denote the transfer function of a finite-dimensional, discrete-time, single-input single-output, linear, constant dynamical system as

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

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<sup>1</sup> 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 is 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 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 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.

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; Furthermore, the approach can be applied to stable or unstable, proper or improper systems<sup>2</sup> to be inverted, and there is also no requirement for the type of input and output signals. Furthermore, it does not suffer non-convex or input noise problems.

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 **INVerse System IDentification** with the first version which is abbreviated as INVSID 1.0.

The remainder of the paper is organized as follows. In Section 2, the inversion approach is proposed, and 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 a numerical example. Finally conclusions and future perspectives are given in Section 4.

<sup>1</sup> For discrete-time systems, nonminimum-phase zeros are zeros that lie outside the unit disk.

<sup>2</sup> 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.

## 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, 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-out systems.

### 2.1. Inverse Identification Approach

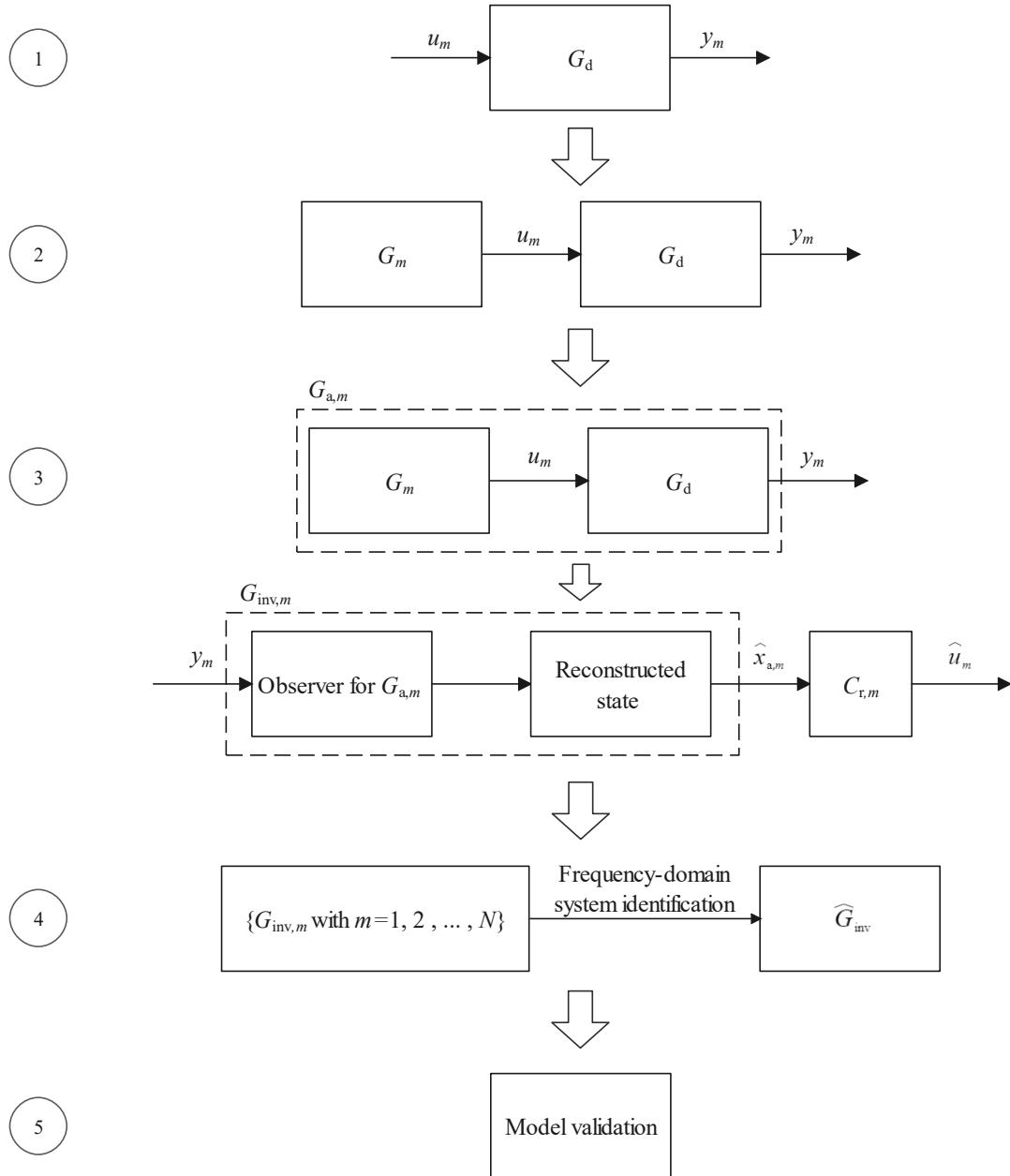
Given a finite-dimensional, discrete-time, single-input single-output, linear, constant dynamical system  $G_d$  which is minimal-realized<sup>3</sup> and proper, the given system  $G_d$  can be either stable or unstable, and the sampling period of system  $G_d$  is  $T_s$  in seconds.

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) Given a number of sine signals  $u_m(k)$ ,  $m = 1, 2, \dots, N$ , corresponding to a number of  $N$  specified frequencies, and 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.

---

<sup>3</sup> The dynamical system  $G_d$  is minimal-realized if and only if it is both controllable and observable.



**Figure 2.** Framework of proposed inverse system identification approach. (The symbol “ $\hat{\cdot}$ ” denotes reconstructed or estimated value.).

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

$$\begin{cases} x_d(k+1) = A_d x_d(k) + B_d u(k), \\ y(k) = C_d x_d(k) + D_d u(k), \end{cases} \quad (3)$$

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

**Remark 1.** Actually, the proposed inverse identification approach is not limited to proper systems, the approach can also be used for identifying inversion of improper systems by replacing the present input  $u(k)$  by future input in (3).

Step 2: Given a set  $S$  containing  $N$  discrete-time sine signals:

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

where

$$u_m(k) = \sin(2\pi f_m k T_s), \quad (5)$$

where  $f_m$  denotes the ordinary 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.,

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

where  $\mathbf{x}_m(k) \in \mathbb{R}^2$  denotes the state vector of the model (6), the matrices  $\mathbf{A}_m$  and  $\mathbf{C}_m$  can be denoted as

$$\mathbf{A}_m = \begin{pmatrix} \cos(2\pi f_m T_s) & \sin(2\pi f_m T_s) \\ -\sin(2\pi f_m T_s) & \cos(2\pi f_m T_s) \end{pmatrix} \quad (7)$$

and

$$\mathbf{C}_m = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad (8)$$

respectively.

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

$$f_m = f_b + (m-1)d, \quad (9)$$

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

$$\begin{cases} \mathbf{x}_d(k+1) = \mathbf{A}_d \mathbf{x}_d(k) + \mathbf{B}_d u_m(k), \\ y_m(k) = \mathbf{C}_d \mathbf{x}_d(k) + \mathbf{D}_d u_m(k), \end{cases} \quad (10)$$

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

$$\begin{cases} \mathbf{x}_{a,m}(k+1) = \mathbf{A}_{a,m} \mathbf{x}_{a,m}(k), \\ y_m(k) = \mathbf{C}_{a,m} \mathbf{x}_{a,m}(k), \end{cases} \quad (11)$$

where the state vector  $\mathbf{x}_{a,m}(k)$  can be denoted as

$$\mathbf{x}_{a,m}(k) = \begin{pmatrix} \mathbf{x}_d(k) \\ \mathbf{x}_m(k) \end{pmatrix}, \quad (12)$$

and the matrices  $\mathbf{A}_{a,m}$  and  $\mathbf{C}_{a,m}$  can be denoted as

$$\mathbf{A}_{a,m} = \begin{pmatrix} \mathbf{A}_d & \mathbf{B}_d \mathbf{C}_m \\ \mathbf{0} & \mathbf{A}_m \end{pmatrix} \quad (13)$$

and

$$C_{a,m} = \begin{pmatrix} C_d & D_d C_m \end{pmatrix}, \quad (14)$$

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

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

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:

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

where

$$C_{r,m} = \begin{pmatrix} 0 \times C_d & C_m \end{pmatrix}. \quad (17)$$

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

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

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

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

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

**Remark 3.** The effective frequency range of the inverse model  $G_{\text{inv}}$  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  $\hat{G}_{\text{inv}}$  in series, and the resulted model can be represented as

$$G_s = \hat{G}_{\text{inv}} G_d \quad (19)$$

of which the frequency response function is written as

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

where  $\Omega = 2\pi f$  with  $f$  the ordinary frequency in Hz, so if the identified inverse model  $\hat{G}_{\text{inv}}$  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, the goodness of the identified inverse model  $\hat{G}_{\text{inv}}$  can be validated.

## 2.2. MATLAB Codes

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

Step 1: Given the transfer function of a nominal model  $G_d$ , make a state-space realization of the transfer function:

```
» systf=tf(numerator,denominator,Ts);
» sysss=ss(systf);
» Figure (1);
» opts=bodeoptions;
» opts.FreqUnits='Hz';
» bode(sysss,opts);
» set(title('Bode plot of G_d'),'interpreter','latex');
» grid;
```

Step 2: According to the bandwidth of the nominal model  $G_d$ , specify the frequencies  $f_m$ ,  $m = 1, 2, \dots, N$ , using the rule stated in Remark 2 and Remark 3, and 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 Equation (7) and Equation (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 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

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

so we can get the following MATLAB codes correspondingly:

```
» 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.,

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

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

and

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

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

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

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]

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

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

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

under the following conditions:

- (a)  $(A_{a,m}, C_{a,m})$  is detectable<sup>4</sup>.
- (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.,

$$L_N = \begin{pmatrix} L_{a,1} & L_{a,2} & \dots & L_{a,N} \end{pmatrix}. \quad (30)$$

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(sysss.a,1)+2;
» AaN=zeros(r1*dim,r1);
» for m=1:dim
    r2=1+(m-1)*r1;
    AaN(r2:r1*m,:)=[sysss.a sysss.b*CN(m,:);
                      zeros(2,size(sysss.a,1)) AN(((2*m-1):2*m),:)];
    end
» CaN=zeros(dim,r1);
» for m=1:dim
    CaN(m,:)=[sysss.c sysss.d*CN(m,:)];
    end
» CrN=zeros(dim,r1);
» for m=1:dim
    CrN(m,:)=[zeros(1,size(sysss.a,1)) CN(m,:)];
    end
» LN=zeros(r1,dim);
» for m=1:dim
    Q=qc*eye(r1);
    R=mc;
    sysa=ss(AaN(r2:r1*m,:),zeros(r1,r1),CaN(m,:),zeros(1,r1),Ts);

```

<sup>4</sup> A system is detectable if all the unobservable states are stable.

```

[~,LN(:,m),~]=kalman(sysa,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

```

**Remark 4.** 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`:

```

> 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);
> set(title('Bode plot of \hat{G}_{inv}'),'interpreter','latex');
> grid;

```

**Remark 5.** The values of  $nx$  denotes 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(sysss,Ginv);
```

Then the Bode plot of the combined model can be displayed using:

```

> figure;
> opts=bodeoptions;
> opts.FreqUnits='Hz';
> bode(Gs,opts);
> set(title('Bode plot of G_s'),'interpreter','latex');
> grid;

```

### 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 as follows.

```

1  function Ginv = INVSIDToolbox(numerator,denominator,Ts,fb,d,N,pc,
2      mc,nx)
3  % numerator and denominator: The numerator and denominator
4      % coefficients of the transfer function of the nominal model G_d.
5  % Ts: The sampling period of the nominal model G_d.
6  % fb: The smallest frequency among the frequency components for
7      % inverse system identification.
8  % d: The common difference.
9  % N: The number of the frequency components for inverse system
10     % identification.
11 % pc: The covariance of the process noise.
12 % mc: The covariance of the measurement noise.
13 % nx: Vector of model orders to scan.
14 % Ginv: The identified inverse model.
15
16 %% Step I
17 systf=tf(numerator,denominator,Ts);
18 sysss=ss(systf);
19 figure(1);
20 bode(sysss);
21 set(title('Bode plot of $G_{\rm d}$'),'interpreter','latex');
22 grid;
23
24 %% Step II
25 dim=N;
26 FN=zeros(dim,1);
27 for m=1:dim
28     FN(m,:)=fb+(m-1)*d;
29 end
30 AN=zeros(2*dim,2);
31 for m=1:dim
32     AN(((2*m-1):2*m),:)= [cos(2*pi*FN(m,:)*Ts)  sin(2*pi*FN(m,:)*Ts)
33                             -sin(2*pi*FN(m,:)*Ts)  cos(2*pi*FN(m,:)*
34                                         Ts)];
35 end
36 CN=zeros(dim,2);
37 for m=1:dim
38     CN(m,:)=[1 0];
39 end
40
41 %% Step III
42 r1=size(sysss.a,1)+2;
43 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     );
65     if isstable(sysr)==1
66         g{m,:}=sysr;
67         GN(m,:)=frd(g{m,:},FN(m,:),'Hz').ResponseData;
68     else
69         break
70     end
71 end
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(2)
77 bode(Ginv);
78 set(title('Bode plot of $\hat{G}_{\rm inv}$'),'interpreter','
79      latex');
80 grid;
81 %% Step V
82 Gs=series(sysss,Ginv);
83 figure(3);
84 bode(Gs);
85 set(title('Bode plot of $G_s$'),'interpreter','latex');
```

```

86 | grid;
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<sup>5</sup>.

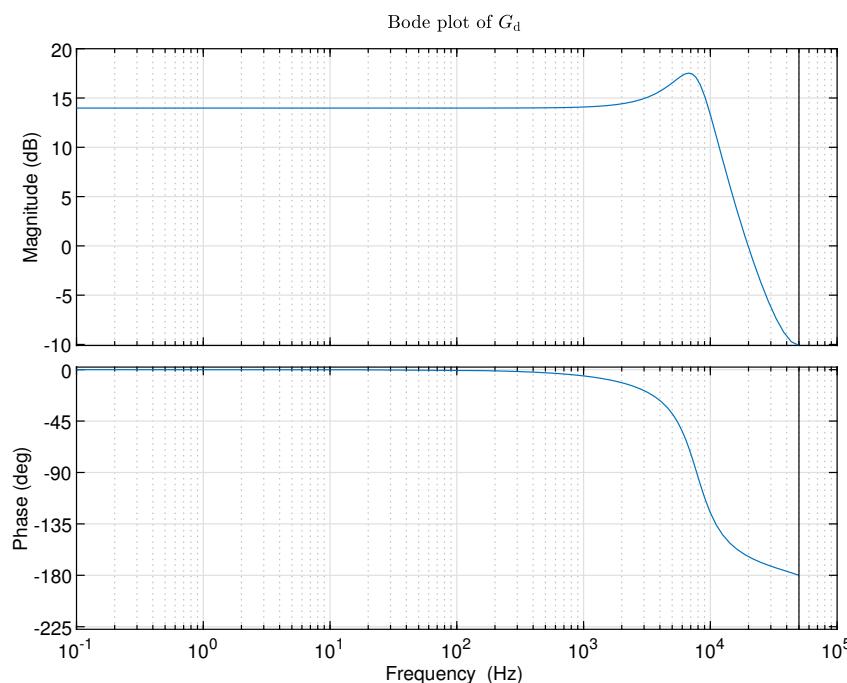
### 3. Numerical Studies

In this section, two numerical examples are used for validate the effectiveness of the toolbox INSID 1.0, i.e., check the effectiveness of the proposed inverse system identification approach.

Firstly, given 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, and 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 (34), 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 a nonminimum-phase zero.

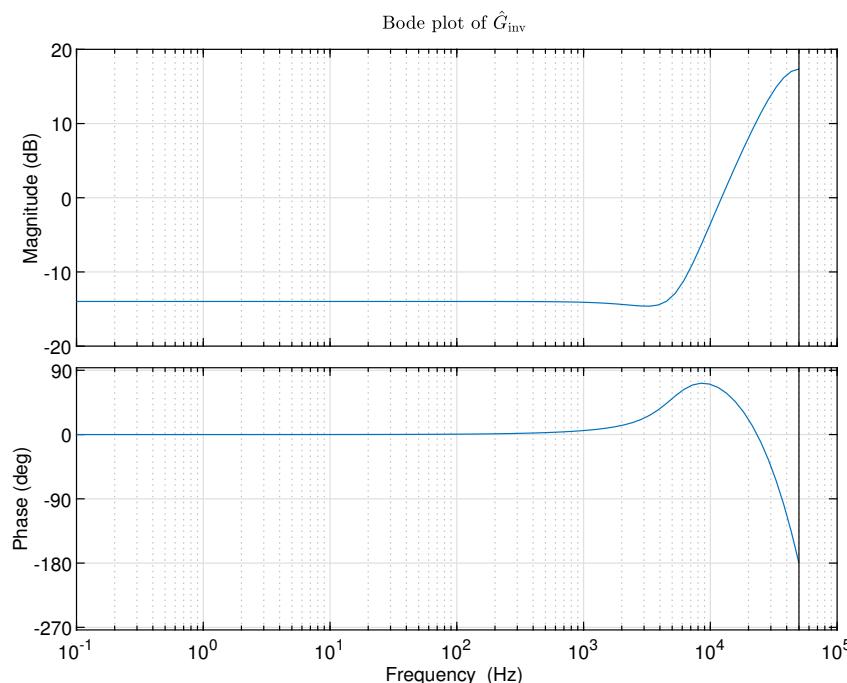
<sup>5</sup> Installation procedure of MATLAB toolboxes can be referred to [https://www.mathworks.com/help/matlab/matlab\\_prog/create-and-share-custom-matlab-toolboxes.html](https://www.mathworks.com/help/matlab/matlab_prog/create-and-share-custom-matlab-toolboxes.html).

According to the above observations, it can be known that the direct inversion of the system  $G$  is a challenging problem. So 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 (20), the frequency range to be identified can be determined. The parameters showed in Table 1 are used as the inputs of the inverse identification toolbox.

**Table 1.** Parameters for inverse identification.

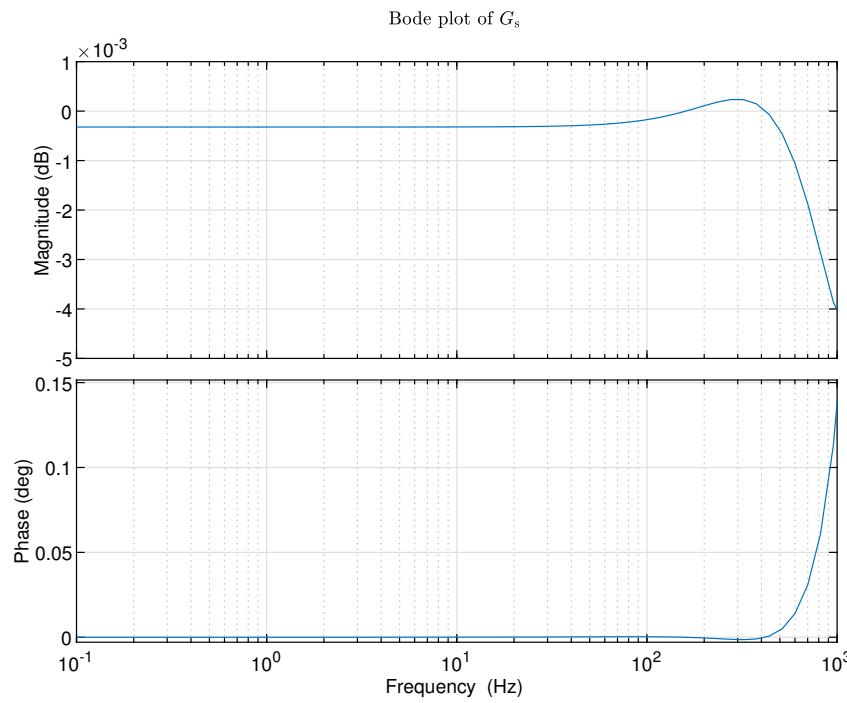
Parameter	Value in MATLAB
numerator	[0,1,0]
denominator	[1,-1.5,0.7]
Ts	1e-5
fb	10
d	10
N	50
pc	1e-3
mc	1e-3
nx	2:10

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  $\hat{G}_{\text{inv}}$  is recommended to be 4. The frequency response properties of the model  $\hat{G}_{\text{inv}}$  with fourth order is demonstrated in Figure 4.



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

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



**Figure 5.** Bode plot of  $G_s$ .

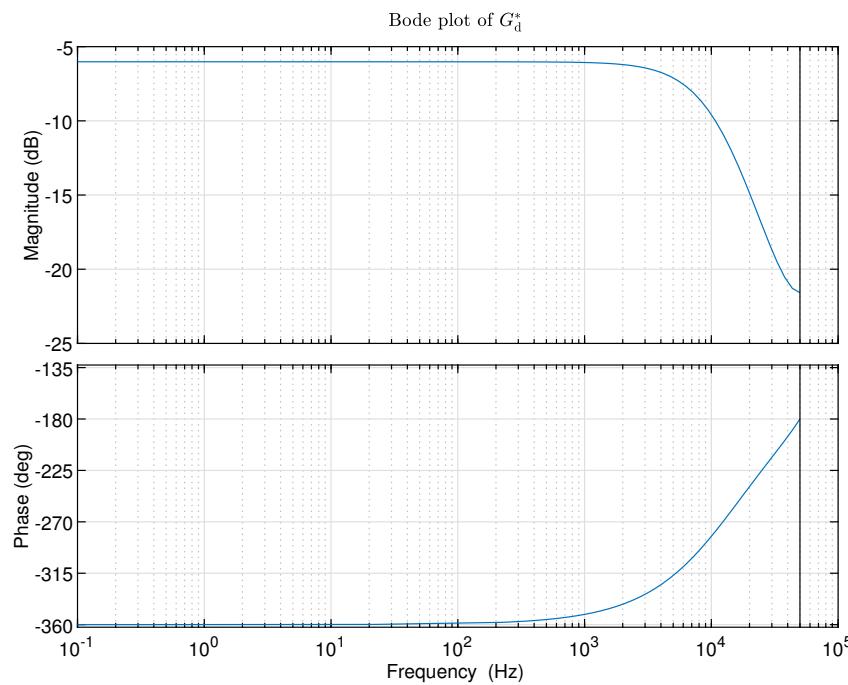
By connecting the model  $G_d$  and the model  $\hat{G}_{inv}$  in series using Equation (19), the model  $G_s$  can be obtained. The obtained model  $G_s$  can then be used for validating the goodness of the identified inverse model  $\hat{G}_{inv}$  in the frequency range of interest. The frequency 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 to 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.

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

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

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

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

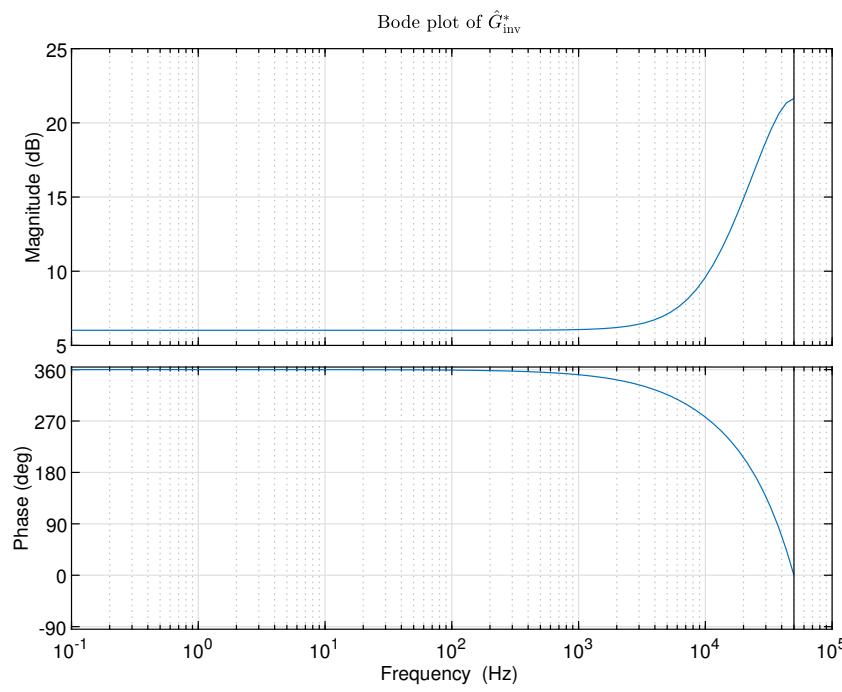
- (a)  $G_d^*$  is unstable.
- (b)  $G_d^*$  is proper.
- (c)  $G_d^*$  is minimal-realized.
- (d)  $G_d^*$  has a nonminimum-phase zero.

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

**Table 2.** Parameters for inverse identification.

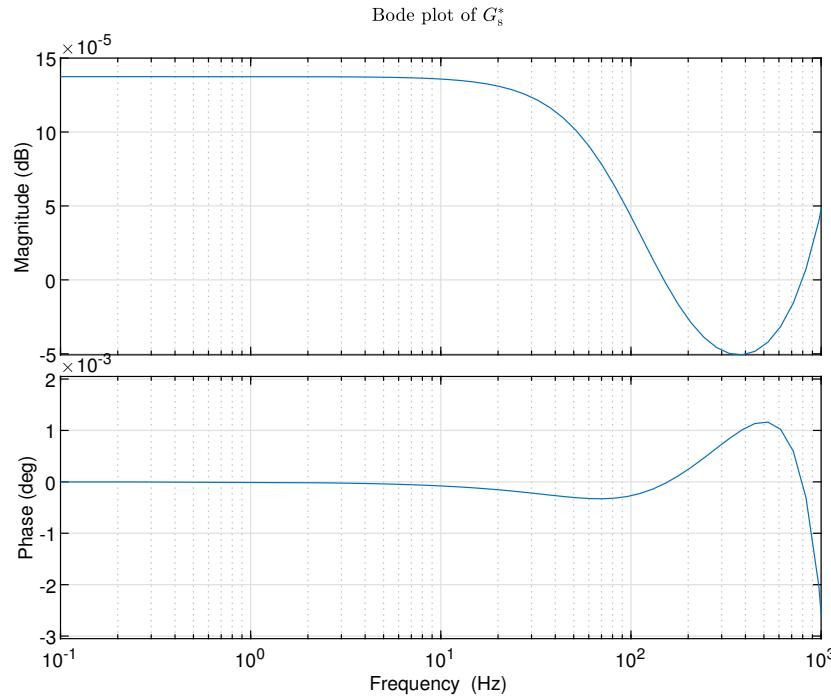
Parameter	Value in MATLAB
numerator	[0, 1, 0]
denominator	[1, -5, 6]
Ts	1e-5
fb	10
d	10
N	50
pc	1e-3
mc	1e-3
nx	2:10

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  $\hat{G}_{\text{inv}}^*$  is recommended to be 4. The frequency response properties of the model  $\hat{G}_{\text{inv}}^*$  with fourth order is demonstrated in Figure 7.



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

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



**Figure 8.** Bode plot of  $G_s^*$ .

By connecting the model  $G_d^*$  and the model  $\hat{G}_{\text{inv}}^*$  in series using Equation (19), the model  $G_s^*$  can be obtained. The frequency response of the obtained model  $\hat{G}_{\text{inv}}^*$  is shown in Figure 8, in the specified

frequency range from 10 Hz to 500 Hz, the magnitude is nearly a constant near to 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.

#### 4. Conclusion and Outlook

In this paper, an alternative system inversion approach is proposed, and 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.
- (b) Preview is not needed.
- (c) The frequency range of interest can be specified.
- (d) Stability of the identified inverse model can be guaranteed.
- (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.

**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).

#### References

1. Goodwin, G.C.; Graebe, S.F.; Salgado, M.E. *Control System Design*; Upper Saddle River, NJ: Prentice Hall, 2001.
2. Golub, G.H.; Van Loan, C.F. *Matrix Computations*; John Hopkins University Press, MD: Baltimore, 2012.
3. Jung, Y. *Inverse System Identification with Applications in Predistortion*; Linköping University, Sweden: Linköping, 2018.
4. van Zundert, J.; Oomen, T. On inversion-based approaches for feedforward and ILC. *Mechatronics* **2018**, *50*, 282–291.
5. Zhou, K.; Doyle, J.C.; Glover, K. *Robust and Optimal Control*; Prentice Hall, NJ: Englewood Cliffs, 1996.
6. B. P. Rigney, L.Y.P.; Lawrence, D.A. Nonminimum phase dynamic inversion for settle time applications. *IEEE Trans. Control Syst. Technol.* **2009**, *17*, 989–1005.
7. S. Devasia, D.C.; Paden, B. Nonlinear inversion-based output tracking. *IEEE Trans. Autom. Control* **1996**, *41*, 930–942.
8. Hunt, L.; Meyer, G.; Su, R. Noncausal inverses for linear systems. *IEEE Trans. Autom. Control* **1996**, *41*, 608–611.
9. Widrow, B.; Walach, E. *Adaptive Inverse Control: A Signal Processing Approach*; John Wiley & Sons, NJ: Hoboken, 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.
11. Tomizuka, M. Zero phase error tracking algorithm for digital control. *J. Dyn. Syst. Meas. Control* **1987**, *109*, 65–68.
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.

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.
14. Mirkin, L. On the  $H_\infty$  fixed-lag smoothing: How to exploit the information preview. *Automatica* **2003**, *39*, 1495–1504.
15. Hazell, A.; Limebeer, D.J.N. An efficient algorithm for discrete-time  $H_\infty$  preview control. *Automatica* **2008**, *44*, 2441–2448.
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*, 1–11.
17. Zhu, Q.; Zhang, Y.; Xiong, R. Stable inversion based precise tracking for periodic systems. *Asian J. Control* **2020**, *22*, 217–227.
18. Zou, Q. Optimal preview-based stable-inversion for output tracking of nonminimum-phase linear systems. *Automatica* **2020**, *45*, 230–237.
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.
20. Romagnoli, R.; Garone, E. A general framework for approximated model stable inversion. *Automatica* **2019**, *101*, 182–189.
21. Bohn, C.; Cortabarria, 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.
22. R. Han, C.B.; Bauer, G. Comparison between calculated speed-based and sensor-fused speed-based engine in-cylinder pressure estimation method. *IFAC-PapersOnLine* **2022**, *55*, 223–228.
23. McKelvey, T.; Akçay, H.; Ljung, L. Subspace-based multivariable system identification from frequency response data. *IEEE Trans. Autom. Control* **1996**, *41*, 960–979.
24. O'Reilly, J. *Observers for Linear Systems*; Academic Press, UK: London, 1983.
25. Lewis, F.L.; Xie, L.; Popa, D. *Optimal and Robust Estimation: With an introduction to stochastic control theory*; CRC Press, FL: Boca Raton, 2008.

**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.