Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, No.3, June : 2024

MODELLING AND IDENTIFICATION OF SISO NONLINEAR SYSTEMS

Imran Ansari, M.Tech Student, Department of Electrical and Electronics Engineering, Al Falah University, Faridabad, Haryana, India.
Mohd Shahid, Associate Professor, Department of Electrical and Electronics Engineering, Al Falah University, Faridabad, Haryana, India.

ABSTRACT
The first step in the control design process is to create appropriate mathematical models of the system that has to be regulated. Physical laws or experimental data can be used to build these models. This section introduces the state-space and transfer function representations of dynamic systems. Next, we discuss some basic techniques for modeling electrical and mechanical systems and show how to build these models in MATLAB for further analysis. The observer is employed to watch the nonlinear function based on this. The stability of the closed-loop system is provided to guarantee the stability of the observer and the diagnosis system.

Keywords:
System Identification, Nonlinear Models, Linearization**.**

I. Introduction

Systems that adapt or change over time in accordance with a set of rules are called dynamic systems. This rule can be expressed as a system of first-order differential equations for a wide variety of physical systems:

$$\dot{x}=\frac{dy}{dx} = f(x(t), u(t), t) \quad (1)$$

Equation (1)'s state vector, x(t), is a set of variables that depicts the configuration of the system at a specific time. For instance, in a simple mechanical mass-spring-damper system, the two state variables may be the mass's location and velocity. The vector u(t) represents the external input vector to the system at time t[1-3]. The (possibly nonlinear) function f generates the time derivative (rate of change) of the state vector, dy/dx, at any given instant in time. Equation (1) can be integrated to precisely determine the state at any given future time, x(t_1), given knowledge of the starting state, x(t_0) and the temporal history of the inputs, u(t), between t_0 and t_1. There is a minimal number of state variables, n, necessary to capture the "state" of a given system and be able to forecast the system's future behavior (solve the state equations), even though the state variables themselves are not unique. The system order, or n, establishes the dimensionality of the state-space. The number of independent energy storage components in the system typically correlates with the system order[4-5]. Equation (1) provides a relatively generic relationship that may be applied to a wide range of systems; on the other hand, its analysis may prove to be highly challenging. To make the problem more manageable, there are two frequent simplifications. First, the system is considered to be time invariant if the function f, that is, $\dot{x}$ = f(x,u), does not depend directly on time. Considering that the fundamental physical principles themselves usually do not depend on time, this is generally a fairly plausible assumption. The parameters or coefficients of the function f are constant for time-invariant systems. It is possible for the state variables, x(t), and control inputs, u(t), to remain dependent on time. The system's linearity is the subject of the second widely held belief. Almost all physical systems are, in fact, nonlinear. Put otherwise, f is usually some complex function of the inputs and the state. Nonlinearities can occur in a variety of ways; in control systems, "saturation"—the occurrence of an element of the system reaching a physical limit—is one of the most prevalent. Fortunately, most systems have roughly linear dynamics over a limited enough working range (think of a tangent line close to a curve)[6-7]. The system of first-order differential equations in this instance can be expressed as a matrix equation,
$\dot{x}$=Ax+Bu.

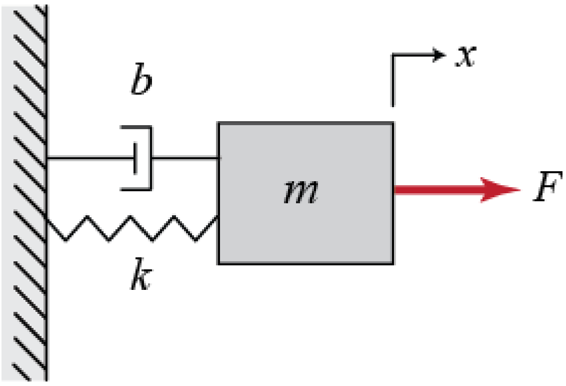

Fig-1 Mass spring damper system

Analysis of linear time-invariant (LTI) systems (Fig-1 Mass spring damper system) was the only feasible approach up until the introduction of digital computers, and even then, only to a significant extent. As a result, these presumptions form the basis of the majority of control theory's conclusions. Thankfully, as we will see, these outcomes have shown to be incredibly successful, and LTI techniques have been used to overcome numerous important engineering issues. The real strength of feedback control systems lies in their ability to function (be robust) in the face of inevitable modeling uncertainty[8-9].

II. **State-Space Representation:**

The typical state-space representation for continuous linear time-invariant (LTI) systems is provided below:

$$\dot{x} = Ax + Bu \quad (2)$$

$$y = Cx + Du \quad (3)$$

Where A is the system matrix (nxn), B is the input matrix (nxp), C is the output matrix (qxn), D is the feedforward matrix (qxp), x is the vector of state variables (nx1), $\dot{x}$ is the time derivative of the state vector (nx1), u is the input or control vector (px1), and y is the output vector (qx1)[10-11]. Because there are frequently state variables that are not immediately seen or are otherwise not of interest, the output equation, Equation (3), is required. Which state variables (or combinations thereof) are accessible for use by the controller are indicated by the output matrix, C. Furthermore, it frequently happens that the state variables are the sole way that the outputs depend on the inputs; in this scenario, D is the zero matrix. Equation (1) makes it simple to handle nonlinear systems, systems with non-zero initial conditions, and multi-input/multi-output (MIMO) systems using the state-space model, also known as the time-domain representation. Consequently, "modern" control theory makes heavy use of the state-space representation[12-15].

III. **Transfer Function Representation:**

One of the most crucial characteristics of LTI systems is that, in the event that the system's input is sinusoidal, the output will likewise be sinusoidal, albeit potentially with varying magnitude and phase. The system's frequency response is represented by these magnitude and phase discrepancies, which are functions of frequency. A system's time-domain representation can be changed into a frequency-domain input/output representation, or transfer function, by applying the Laplace transform. By doing this, it also converts the differential equation that governs into an algebraic equation, which is frequently simpler to understand. The following defines the Laplace transform of a time domain function, f(t):

$$F(s) = L\big(f(t)\big) = \int_0^{\infty} e^{-st} f(t)dt \quad (4)$$

Where the complex frequency variable S=σ+jω is the parameter. In practical application, it is uncommon that you will need to assess a Laplace transform directly (but you should know how to). It is especially crucial to understand the Laplace transform of a function's nth derivative:

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, No.3, June : 2024

$$L\left\{\frac{d^n f}{dt^n}\right\} = s^n f(s) - s^{n-1} f(0) - s^{n-2} f(0) - \cdots . - f^{n-1}(0) \quad (5)$$

When evaluating LTI single-input/single-output (SISO) systems, such as those controlled by a constant coefficient differential equation, frequency-domain techniques are most frequently employed, as the following example illustrates:

$$a_n \frac{d^n y}{dt^n} + \ldots\ldots\ldots + a_1 \frac{dy}{dt} + a_0\, y(t) = b_m \frac{d^m u}{dt^m} + \ldots\ldots\ldots + b_1 \frac{du}{dt} + b_0\, u(t) \quad (6)$$

The Laplace transform of this equation is given below:

$$a_n s^n Y(s) + \cdots + a_1 s\, Y(0) + a_0 Y(s) = b_m s^m U(s) + \cdots + b_1 s\, U(s) + b_0 U(s) \quad (7)$$

Where the Laplace Transforms of y(t) and u(t), respectively, are denoted by Y(s) and U(s). Keep in mind that we always assume that y(0), $\dot{y}(0)$, u(0), and all other initial conditions are zero while finding transfer functions. The function of transfer from input $U(s)$ to output $Y(s)$ is, therefore:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} \ldots + b_1 s + b_0}{a_n s^n + b_{n-1} s^{n-1} \ldots + a_1 s + a_0} \quad (8)$$

It is useful to factor the numerator and denominator of the transfer function into what is termed zero-pole-gain form:

$$G(s) = \frac{N(s)}{D(s)} = K \frac{(s-z_1)(s-z_2) \ldots\ldots (s-z_{m-1})(s-z_m)}{(s-p_1)(s-p_2) \ldots\ldots (s-p_{n-1})(s-p_n)} \quad (9)$$

The **zeros** of the transfer function, $z_1 \ldots \ldots z_m$ are the roots of the numerator polynomial, i.e. the values of s such that $N(s) = 0$. The poles of the transfer function, $p_1 \ldots \ldots p_n$, are the roots of the denominator polynomial, i.e. the values of s such that D(s)=0. Both the zeros and poles may be complex valued (have both real and imaginary parts). The system gain is

$$K = \frac{b_m}{a_n}.$$

Keep in mind that the state-space representation also allows us to directly determine the transfer function in the following ways:

$$G(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1} + D \quad (10)$$

IV. Mechanical Systems:

Mechanical system analysis is based on Newton's equations of motion. Equation (11), which represents Newton's second law, indicates that the product of a body's mass and acceleration is the total force acting on it. For our purposes, Newton's third law says that two bodies in contact feel the same amount of contact force, but it acts in different directions.

$$\sum F = ma = m \frac{d^2 x}{dt^2} \quad (11)$$

It is best to create a free-body diagram (FBD) of the system that displays all of the applied forces when using this equation. The free-body diagram for this system is shown below. The spring force is proportional to the displacement of the mass, x, and the viscous damping force is proportional to the velocity of the mass, $v = \dot{x}$. Both forces oppose the motion of the mass and are, therefore, shown in the negative x-direction. Note also that x=0 corresponds to the position of the mass when the spring is unstretched (shown in fig-2).

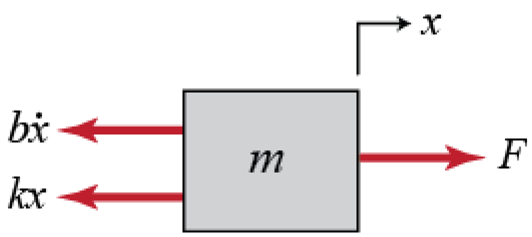

Fig-2 Free body diagram of Mass spring damper system

We now calculate the total force and apply Equation (11), Newton's second law, in each direction. There aren't any forces at work in the y direction in this instance, but there are some in the x direction.

$$\sum F_x = F(t) - b\dot{x} - kx = m\ddot{x} \quad (12)$$

The dynamic state of the system is fully described by this equation, which is sometimes referred to as the governing equation. We will later see how to utilize this to examine system attributes like stability and performance, as well as to compute the system's reaction to any external input, F(t). The mass-spring-damper system's state-space representation can be found by breaking down the second-order governing equation into a pair of first-order differential equations. We select the position and velocity as our state variables in order to achieve this.

$$X = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \quad (13)$$

Whereas the velocity variable records the kinetic energy held by the mass, the position variable records the potential energy contained in the spring. The damper doesn't hold energy; it just releases it. It is frequently useful to think about which variables best capture the energy stored in the system when selecting state variables.

The state equation in this case is:

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} F(t) \quad (14)$$

For example, the output equation is as follows if we want to be able to control the mass's position:

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \quad (15)$$

Table-1 Parameters of mass spring damper system[16-20]

m	Mass	10.0 kg
k	Spring constant	1.0 N/m
b	Damping constant	0.5 Ns/m
F	Input force	2.0 N

For this system, assuming zero initial conditions, the Laplace transform is

$$ms^2X(s) + bsX(s) + kX(s) = F(s) \quad (16)$$

Consequently, the force input to displacement output transfer function is

$$\frac{X(s)}{F(s)} = \frac{1}{ms^2+bs+k} \quad (17)$$

$$\frac{X(s)}{F(s)} = \frac{1}{10s^2+0.5s+1}$$

Response of deseeded system can be checked with different input signal with help of bode and nyquist plot with matlab.

V. Results:

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, No.3, June : 2024

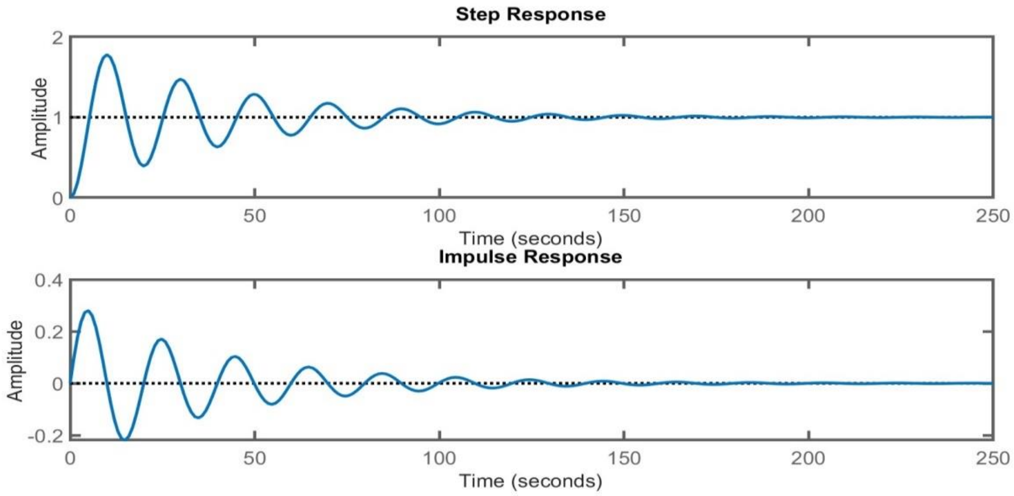

Fig-3 Step Response and Impulse response of designed system

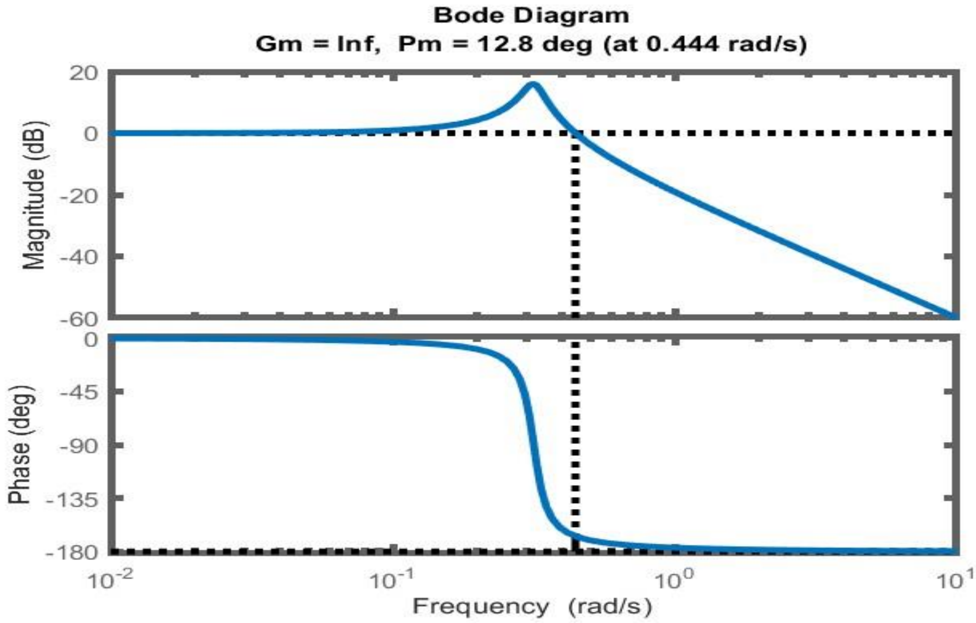

Fig-4 Gain margin and phase margin of designed system

Gm = Inf, Pm = 12.8371, Wcg = Inf, Wcp = 0.4444

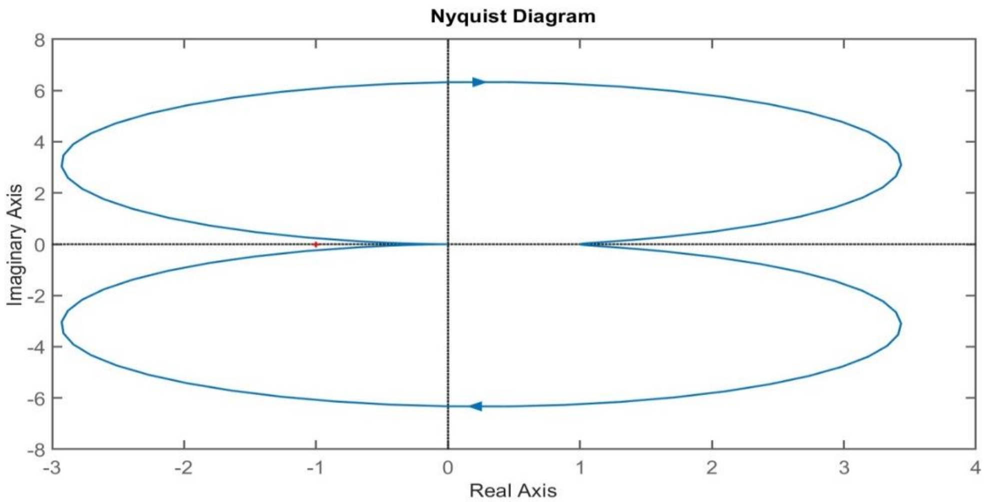

Fig-5 Nyquist plot of designed system

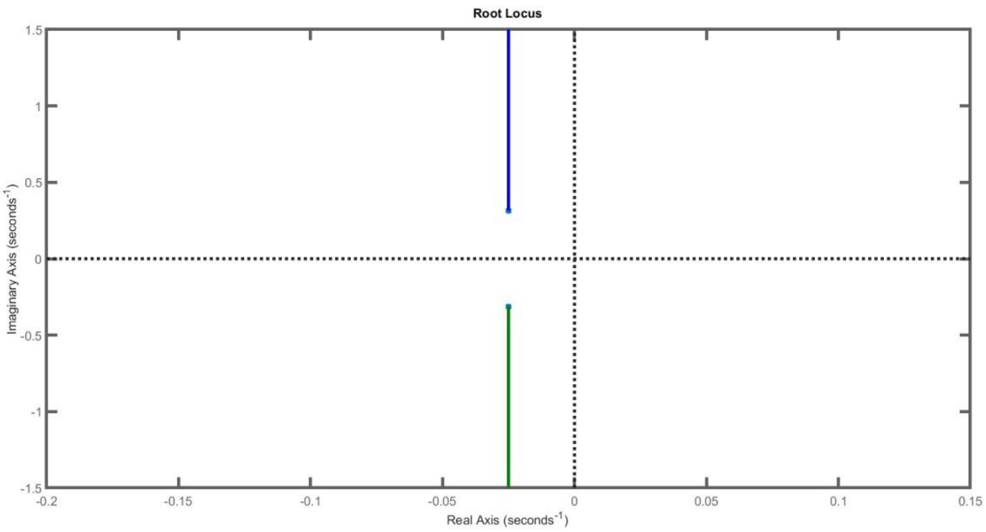

Fig-6 root locus plot of designed system

VI. Conclusion:

This paper considers the identification of nonlinear time-invariant single input-single output (SISO) systems, consisting of a multivariable linear dynamic system and one static SISO nonlinear system. The LEM model approximates the original nonlinear system around the equilibrium manifold and is built on the basis of the local models, which can be derived, for example, by identification. The LEM model can make a decent approximation model for the kinds of systems under consideration, as demonstrated by a numerical example.

VII. Reference:

[1] Hu Zhenggao, Zhao Guorong, Li Fei and Zhou Dawang, "Fault Diagnosis of Nonlinear Dynamic Systems Based on Adaptive Unknown Input Observer[J]", *Control and Decision*, vol. 31, no. 05, pp. 901-906, 2016.

[2] Wu LN, Model-based robust fault detection and estimation method for uncertain systems [D], Harbin Institute of Technology, 2013.

[3] Yan Junying, Design of the VAV BOX controller for VAV air conditioning system [D], Xi'an University of Architecture and Technology, 2010.

[4] Liu Yanjun, Recursive Least Squares Identification of Non-uniform Sampling Data Systems [D], Jiangnan University, 2009.

[5] Sun Rong, Liu Sheng and Zhang Yufang, "Fault diagnosis observer design for a class of nonlinear systems [J]", *Control theory and application*, vol. 30, no. 11, pp. 1462-1466, 2013.

[6] Jordan, M. I. and Rumelhart, D. E., “Forward models: supervised learning with a distal teacher,” *Cognitive Science*, vol. 16, pp. 307–354, 1992.

[7] Narendra, K. S. and Parthasarathy, K., “Identification and control of dynamical systems using neural networks,” *IEEE Trans. Neural Networks*, vol. 1, pp. 4–27, 1990.

[8] Chen, F. C. and Khalil, H. K., “Adaptive control of a class of nonlinear discrete-time system using neural networks,” *IEEE Trans. Automat. Control*, vol. 40, no. 5, pp. 791–801, 1995.

[9] Goh, C. J. and Lee, T. H., “Direct adaptive control of nonlinear systems via implicit function emulation,” *Control Theory Adv. Tech.*, vol. 10, no. 3, pp. 539–551, 1994.

[10] Goh, C. J., “Model reference control of nonlinear systems via implicit function emulation,” *Internat. J. Control*, vol. 60, no. 1, pp. 91–115, 1994.

[11] Hou, Z. and Huang, W., “The model-free learning adaptive control of a class of SISO nonlinear systems,” in *Proceedings of the American Control Conference*, Albuquerque, New Mexico, pp. 343–344, 1997.

[12] Clarke, D. W., Mohtadi, C. and Tuffs, P. S., “Generalized predictive control - part I. The basic algorithm,” *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.
[13] Clarke, D. W. and Mohtadi, C., “Properties of generalized predictive control,” *Automatica*, vol. 25, no.6, pp. 859–875, 1989.
[14] Soeterboek, R., *Predictive Control: A Unified Approach*, Prentice-Hall: Englewood Cliffs, NJ, 1992.
[15] Ydstie, B. E., “Extended horizon adaptive control,” in *Proceedings of 9th IFAC World Congress*, Budapest, Hungary, 1984.
[16] De Keyser, R. and Van Cauwenberghe, A., “A self-tuning multistep predictor application,” *Automatica*, vol. 17, no. 1, pp. 167–174, 1979.
[17] Clarke, D. W., Mosca, E. and Scattolini, R., “Robustness of an adaptive predictive controller,” *IEEE Trans. Automat. Control*, vol. 39, no. 5, pp. 1052–1056, 1994.
[18] Kroyszig, E., *Advanced Engineering Mathematics*, Wiley, New York, 7th edition, 1993.
[19] Inonnou, P. A. and Sun, J., *Robust Adaptive Control*, Prentice-Hall: Englewood Cliffs, NJ, 1996.
[20] Haykin, S., *Adaptive Filter Theory*, Prentice-Hall: Englewood Cliffs, NJ 3rd edition, 1996, Chapter 17.