ISBN: 978-1-948012-16-4

Contemporary Issues in Computing (CIC)

DOI: <http://doi.org/10.26480/cic.01.2020.14.19>

CrossMark

NON-LINEARITY SYSTEM RECTIFICATION USING MATHEMATICAL APPROXIMATION TECHNIQUE

Souvik Roy^a and Rajesh De^b^aDept. of ECE, Techno International, Batanagar, South 24 Paraganas, West Bengal, India^bDept. of ECE, Brainware Group of Institutions, Kolkata, West Bengal India

This is an open access article distributed under the Creative Commons Attribution License CC BY 4.0, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ARTICLE DETAILS

Article History:

Received 26 October 2020

Accepted 27 November 2020

Available online 03 December 2020

ABSTRACT

Nonlinear System are widely used in modern control system, communication and engineering applications viz. oscillating pendulum, load cell etc. However, rectification of output distorted by noise due to different inner and external variables, are big concern. Primary goal of this investigation is to determine the convergence of non-linear system and to fix it by using Adaptive and Approximation methods. Approximation Techniques used to generate the reference or training signal. Least Square Approximation (LSA), Group Average Method (GAM) and Particle Swarm Optimization (PSO) been used to generate training signal also optimized to the desired value. Later this training signal used in an adaptive scheme as a reference signal. Least Means Square Algorithms used to remove the noisy load cell output with the adaptive filter. Noises caused by the creeping and drifting mistake at the output. The Adaptive Filter utilizes the reference signal produced by approximation methods to eliminate both creeping and drifting errors and to produce a load cell's required reaction.

KEYWORDS

Optimization, Non-linear system, Load cell, Mass spring damper, Particle Swarm, Group Average.

1. INTRODUCTION

Load cells are widely used in various applications like Aerospace Engineering, automation, ticket or money vending machine. Information recorded has to be processed and control system does not work correctly if they get improper input data and compensation of optimizing the sensor is a modern important research aspect. Resting time for this kind of sensor is very large and highly damping in nature. Consequently, the weight measured by the load cell is a distorted response. Modern day load cells called smart cell where the effect of these influence of improper data or external data has been reduced drastically. Proper signal processing and dynamic computation technique can accomplish this (Jafaripناه et al., 2017; Rajesh et al., 2014; Shu, 1993).

It is the key factor to make correct decision as close enough as possible very quickly for dynamic system. But before resting time it is tedious to make correct declaration of final value for load cell while it is still oscillating. Consequently, we have to speed up speed up measurement process. Keeping in mind this fact several techniques been reviewed to solve this problem. Software technique for sensor compensation, Digital adaptive technique has been assessed (Brignell, 1991; Cleghorn).

Convergence of non-linear system have been studied using Phase Plane Analysis and Lyapunov Non-Linear Control System analysis with the help of compilation and simulation software. For some cases to rectify the output response, analog filtering been done (Brignell and White, 1994).

In this paper, we propose a complete new methodology for finding out

non-linearity of a second order system using compiling and simulation. After that, error in the transient dynamic response has been analyzed and three different algorithms has been taken in the context of examining percentage of error, linearity and time complexity for practical sensor voltages with reference of signal correction of load cell output of the previous investigator job (Shi et al., 1993). Section I yields mathematical modelling of second order nonlinear system whereas in section II we describe with linearization of the characteristic by different algorithm and their comparative study (Jafaripناه et al., 2003; Pavée et al., 2013; Shi et al., 1993).

2. MATHEMATICAL MODELLING OF NONLINEAR SYSTEM

A mass M is attached to a spring (stiffness k) and a dashpot (viscous friction coefficient f) on which the force F acts. Displacement y is positive in the shown direction. The zero position taken to be at the point where the spring and mass are in static equilibrium.

Figure 1: Mass Spring Damper

Quick Response Code	This paper was presented at	Access this article online	
	International Conference on Contemporary Issues in Computing (ICCIC-2020) - Virtual IETE Sector V, Salt Lake, Kolkata From 25th-26th July 2020	Website: www.intelcomp-design.com	DOI: 10.26480/cic.01.2020.14.19

Figure 2: Scientific representation of Mass Spring Damper

The systematic way of analyzing such a system is to draw a free diagram.

By applying Newton's Law of motion to the free body diagram. By applying Newton's law of motion to the free body diagram the force equation can be written as

$$F(t) - c \frac{dy}{dt} - ky = M \frac{d^2 y}{dt^2} \quad \text{Or} \quad F(t) = M \frac{d^2 y}{dt^2} + c \frac{dy}{dt} + ky$$

No spring is linear over an arbitrary range of extension. In fact, that is true of all physical as well as nonphysical systems.

The linear spring elemental law $F = ky$ is applicable within a limited range of extension y measured beyond the unstretched and of the spring large extension are encountered the spring law changes to $F(t) = ky^2$ which is graphically represented as F

Figure 3: Load cell with load

Figure 4: Non Linearity of the load cell

Under gravitational force $(M + m)g$, which when large pushes the spring nonlinear region of operation, the equation becomes $(M + m)g = (M + m)\ddot{y} + c\dot{y} + ky^2$

3. NON-LINEARITY MEASUREMENT OF SECOND ORDER SYSTEM

Different types of nonlinear systems are present in daily practice. A pendulum in oscillation, a mass spring damper or it could be a load cell are the examples for second order non-linear system. In this context we will analyze the convergence of different kinds of nonlinear system.

3.1 Undamped Pendulum

An undamped pendulum with no driving force represented mathematically as $\frac{d^2 y}{dt^2} + \sin(y) = 0$. We reduce this to standard Matlab form of a system of first order differential equation by letting $y_1 = y$ and $y_2 = \dot{y}_1$. This leads to

$$\dot{y}_1 = y_2 \quad \text{and} \quad \dot{y}_2 = -\sin(y_1)$$

In order to verify the convergence of pendulum, whose transient response is damped oscillatory we have proposed a program which gives convergence analysis when it is damping and after certain time elapsed tends to resting position. This program is written on MATLAB programming language. The flow chart for the program is given in Fig. 5 while the phase plane analysis is shown on Fig. 6. There are two methods to study nonlinearity of a second order system one is phase plane analysis and second one is Lyapunov Theory. This can also be verified using Nelinsys toolbox from MATLAB simulation. However, the result in Fig. 6 indicates that direction of the arrow head are towards outside means it is not at all converging in nature

3.2 Flow Chart

Figure 5: Flow chart for Matlab Program for damping pendulum

Figure 6: Flow chart for Matlab Program for load cell

3.3 Load Cell

Basically, load cell is a mass spring damper (MSD) shown in Fig. 3 where M is the load cell equivalent mass which is attached to the mass less spring with spring constant k . When a small mass m is applied over the load cell then it is subjected to displace downward having displacement y shown in fig. 3. At that time an offset force is generated due to spring displacement y is can be defined by Hook's law as $f_s = -ky$. Here the gravitational force also pushes the spring to downwards. This satisfy the equilibrium

characteristic of the load cell. For simplicity of our calculation we only consider the dynamic characteristic where the damping force $f_D = -c \frac{dy}{dt}$ where c is the damping coefficient. Now using the Newton's law of motion, the second order equation of dynamic equilibrium can be achieved:

$$(M + m) \frac{d^2 y}{dt^2} + c \frac{dy}{dt} + ky = F(t) \text{ where } F(t) \text{ is the force function}$$

After a certain point of time P_0 characteristic curve of the load cell is started to be nonlinear as a result of damping shown in fig. 4.

3.4 Simulated Result

Simulated result for the convergence of the load cell been shown. Arrowheads directed outwards proves the nonlinearity of the load cell.

Figure 7: Convergence of Load cell

Figure 8: Convergence of pendulum

4. NONLINEARITY RECTIFICATION

Researchers have applying various methods to fix a suitable target according to which adaptive system is adjusted or modified. Here in this research three types of mathematical approximation techniques are applied to search the desired value of adaptive signal correction techniques.

4.1 Least Square Approximation Method (LSA)

One of the first applications of the method of least squares was to settle a controversy involving Earth's shape. The least square approximation method is a standard way in regression analysis, which used to solve the systems having more unknown equations. "Least squares" means the solution, which minimizes the squared sum of the residuals of every single equation.

LSA is a statistical method for estimating the true value of some quantity based on a consideration of errors in observation for measurement. In particular, the line (the function $y_i = mx_i + c$ where x_i are the values at which y_i is measured and i denotes the specific observation) minimize the sum of the squared distance (deviation) from the line to each observation is used to approximate a relationship that is assured to be liner i.e. the sum overall i of $(y_i - c - mx_i)^2$ is minimized by setting the practical derivatives of the sum with respect to m and c equal with zero.

4.1.1 Mathematical Model

Two-mathematical model can analyzed in LSA.

A straight-line can expressed as $y = mx + c$. Which has a number of points (say) = an n^{th} term that means n no of inputs taken and correspondingly outputs are observed. Where ' m ' is taken as gradient & ' c ' is taken as offset. Now, the equation derived as

$$\sum_{i=1}^n y_i = \sum_{i=1}^n x_i + n * c$$

(1) Where $\sum_{i=1}^n (y_i)$ gives the n no of voltages observed and $\sum_{i=1}^n (x_i)$ gives the temperature readings. In this equation multiply n with " c ", due to " n "- the number of readings. Here " c " is a constant & we have to calculate the value of " m " and " c " through Matlab Programming

Now from a set of predefined data inputs and corresponding output readings shows that Error =

$$(y_1 - f(x_1))^2 + (y_2 - f(x_2))^2 + (y_3 - f(x_3))^2 + \dots (y_n - f(x_n))^2 \quad (2)$$

Equation (2) can written by

$$\text{Error} = \sum_{i=1}^n (y_i - f(x_i))^2 \quad (3)$$

Where $f(x) = mx + c$ (4)

Generalizing for any random value $f(x_i) = mx_i + c$ (5)

After putting the value of $f(x_i)$ in eqn. (3), we find Error=

$$\sum_{i=1}^n (y_i - (mx_i + c))^2 \quad (6)$$

Which can solved as,

$$\frac{\partial \mathcal{E}}{\partial n} = 0 \text{ Means } \sum_{i=1}^n y_i = m \sum_{i=1}^n x_i + n * c \quad (7)$$

$$\frac{\partial \mathcal{E}}{\partial c} = 0 \text{ Means } \sum_{i=1}^n x_i y_i = m \sum_{i=1}^n x_i^2 + n * c \sum_{i=1}^n x_i \quad (8)$$

Where $\mathcal{E}$ indicates the "Error" in equation (7) and (8)

Value of C and m can formulated for programming by the following way

$$\text{Type I: } c = \frac{\left(\sum_{i=1}^n x_i y_i * \sum_{i=1}^n x_i \right) - \left(\sum_{i=1}^n y_i * \sum_{i=1}^n x_i \right)}{\left(n \sum_{i=1}^n x_i \sum_{i=1}^n x_i \right) - \left(n * \sum_{i=1}^n x_i x_i \right)} \text{ and } m = \frac{\sum_{i=1}^n y_i - n * c}{\sum_{i=1}^n x_i} \quad (9)$$

After solving eqn. (7) and (8), we can find the mathematical value of m and c .

Type II:

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n} \quad \bar{Y} = \frac{\sum_{i=1}^n y_i}{n} \quad (10)$$

Then m and c can written by

$$m = \frac{\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y})}{\sum_{i=1}^n (x_i - \bar{X})^2} \text{ and } c = \bar{Y} - m\bar{X} \quad (11)$$

1.1. 4.1.2 Software Analysis of LSA method

1.2. The Least Square Method is one type of mathematical procedure for finding the best- fitting curve to a provided set of points by minimizing the sum of the squares of the offsets of the points from the curve. The sum of squares of the offsets value been used instead of the offset absolute values because this allows the residuals to be treated as a continuously differentiable quantity. This theoretical concept along with the graphical concept from the previous section been implemented in the MATLAB program.

1.3. 4.1.3 Algorithm of LSM

Step 1 : Start

Step 2 : Declare and Initialize Variables - sumx=0, sumy=0, sumxx=0, sumxy=0, X[30], Y[30].

Step 3 : Take number of input sets n

Step 4 : Take n number of input sets as array in the form of X [], Y [];

Step 5 : CALCULATE the $\sum X, \sum Y, \sum XY, \sum X^2, \sum Y^2, \sum X \sum Y$

Step 6 : CALCULATE "c" and "m" as per equation (9)

Step 7 : Put the value of m and c in the equation (1)

Step 8 : Display m and c

Step 9 : Stop

Figure 9: Curve fitting using GAM

1.4. 4.2 Group Average Method (GAM)

1.5. The method of Group Average is another approximation technique of curve fitting and linearization of system equation. GAM is based on the fact sum of the residual or sum of the error is always zero. First declassify the data in two groups and find the individual average of that two group. Hence compare the changes in both the groups. After plotting linear graph we observe the error and this error can be positive or negative but the sum of the error is zero as in LSM. Here the graphical model is discussed as below.

1.6. In the fig. 9, a linear plot is shown which is observed by a number of discrete lines L1, L2, L3.... L5.... The x-axis is denoted as X and the y-axis is denoted as Y. Consider a straight line:

$$y = mx + c$$

There are n observations $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$

If $x = x_1$, the value of y will be y_1 . Now, $y_1 = mx_1 + c$. Then Error

$$\epsilon_1 = y_1 - (mx_1 + c)$$

Similarly $x = x_2$ the value of y will be y_2 . For this case also error

$$\epsilon_2 = y_2 - (mx_2 + c) \text{ and so on}$$

$$\text{For } n\text{th value } \epsilon_n = y_n - (mx_n + c)$$

The error can be positive or negative, but the sum of the error will be zero. $\sum_{k=1}^n \epsilon_k = 0$

To find out the constant of m and c, we require two equation and for that purpose we divided into two groups. The first expression contains k observations $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$. The second expression

contains n-k observations $(x_{k+1}, y_{k+1}), (x_{k+2}, y_{k+2}), \dots, (x_n, y_n)$.

Assuming that sum of the error in each group is zero we get

$$[y_1 - (mx_1 + c)] + [y_2 - (mx_2 + c)] + \dots + [y_k - (mx_k + c)] = 0 \quad (12)$$

$$[y_{k+1} - (mx_{k+1} + c)] + [y_{k+2} - (mx_{k+2} + c)] + \dots + [y_n - (mx_n + c)] = 0 \quad (13)$$

On simplifying this we obtain

$$\frac{y_1 + y_2 + y_3 + \dots + y_k}{k} = m \frac{x_1 + x_2 + x_3 + \dots + x_k}{k} + c \quad (14)$$

$$\frac{y_{k+1} + y_{k+2} + y_{k+3} + \dots + y_n}{n-k} = m \frac{x_{k+1} + x_{k+2} + x_{k+3} + \dots + x_n}{n-k} + c \quad (15)$$

From the equation (14) and (15)

$\frac{1}{k}(x_1 + x_2 + x_3 + \dots + x_k)$ and $\frac{1}{k}(y_1 + y_2 + y_3 + \dots + y_k)$ indicates the mean values of x's and y's of the first group. Similarly

$\frac{1}{n-k}(x_{k+1} + x_{k+2} + x_{k+3} + \dots + x_n)$ and

$\frac{1}{n-k}(y_{k+1} + y_{k+2} + y_{k+3} + \dots + y_n)$ indicates the mean values of x's and y's of the second group. By solving these two equations, we get the value of m and c.

1.7. 4.2.1 Software Analysis of GAM

This theoretical concept along with the graphical concept from the previous section can be implemented in the MATLAB program.

1.8. 4.2.2 Algorithm of GAM

Step 1: Start

Step 2: Declare variables count, temp, voltage, a, b, i

Step 3: Read the values of count, temp, voltage

Step 4: Display the values of temp and voltage

Step 5: Declare variables halfsumtemp, halfsumvolt, restsumtemp, restsumvolt

Step 6: Initialize variables halfsumtemp =0, halfsumvolt =0, restsumtemp=0, restsumvolt =0

Step 7: Repeat the steps until $i < (\text{count}/2)$

i) halfsumtemp = halfsumtemp + temp(i);

ii) halfsumvolt = halfsumvolt + voltage(i);

Step 8: Repeat the steps from $i = (\text{count}/2) + 1$ to $i < n$

i) restsumtemp = restsumtemp + temp(i);

ii) restsumvolt = restsumvolt + voltage(i);

Step 9: Calculate the equations-

first_avg_temp = halfsumtemp / (count/2); rest_avg_temp =

restsumtemp / (count/2); first_avg_volt = halfsumvolt / (count/2);

rest_avg_volt = restsumvolt / (count/2);

$m = (\text{rest_avg_volt} - \text{first_avg_volt}) / (\text{rest_avg_temp} - \text{first_avg_temp})$;

$c = \text{first_avg_volt} - (m * \text{first_avg_temp})$;

Step 10: Display the value of m and c

Step 11: Stop

4.3 Simulated output LSA vs GAM vs PSO

Figure 10: LSA Type I and type II

Figure 11: GAM

Figure 12: Comparison of linearization using LSA and GAM

Figure 13: Linearization using PSO

4.4 Flow Chart LSA vs GAM

Figure 14: Flowchart LSA for type I

Figure 15: Flowchart for GAM

4.5 Experimental Result of LSM Type I, LSM Type II and GAM

If load cell voltage is varied, take different reading and for those cases, if we take fitted values from fig. 10 and fig. 11 and calculate percentage of error then we see that average percentage of error for GAM is less of its predecessor.

4.6 Particle Swarm Optimization (PSO)

Particle Swarm Optimization is a population based stochastic optimization technique invented by Dr. Eberhart and Dr. Kennedy in 1995 inspired by social behavior of bird flocking or fish schooling. PSO tells that a large number of birds can search their food according to a particular way where the food is limited and the most important thing is birds do not know that where the food is actually located but they know how far they can travel. It has a wide area of application like complex computational method or fixing a piecewise linear model from a scattered two-dimensional data set. It is an arbitrarily approximation technique where linearization of the curve can be done by computing objective function and finding it out lower of it continuously. It is same like regression technique of LSA, GAM but performance been much improved. The advantage of PSO is it is easy to implement in different field of engineering by little bit of parameter adjustment. Here we have applied PSO in sensor output curve fitting and used the weight age factor of 0.9. Here solution considered as the velocity of the set i.e. in case sensor output characteristic it is dataset. Rate of the change of voltage and overall change of voltage has been calculated and considered as the current solution and the best solution.

During movement the data set can be updated its own position and velocity based on the best experience of its own and entire data set. The following steps are involved

1. The velocity and the position of each dataset are randomly set within predefined range.

2. In each iteration, the velocities of the entire particle updated according to the following equation.

$$v(t+1) = w * v(t) + c_1 * R_1 * (p_{best} - p(t)) \quad (16)$$

Where $p(t)$ and $v(t)$ are the current position and current velocity of data set and p_{best} is the position having best objective function (here slope is considered as objective function) found so far.

3. The position of the entire data set has been updated using the following rule

$$p(t+1) = p(t) + v(t+1) \quad (17)$$

This could applied in any simulation software. We have implemented this concept of PSO in MATLAB.

4.6.1 Algorithm of PSO

- Step 1: Start
 Step 2: Declare variables count, temp, p and v to hold current position and velocity
 Step 3: Declare the number of knot locations for the entire search and weight age.
 Step 4: Start the data set from the origin and map accordingly using the formula temp (index)=temp(index)-temp(1);
 volt (index)=volt(index)-volt(1);
 Step 5: Calculate total slope and store it into a variable named p_{best}
 Step 6: Calculate rate of change voltage every time in accordance with rate of change of temperature and assign it into a variable $p(index)$ dynamically.
 Step 7: Declare velocity and assign the same value into it.
 Step 8: Update the velocity according to the equation (16)
 Step 9: Update next particle or data according to equation (17)
 Step 10: Plot the new data set
 Step 11: Stop

4.6.2 Flowchart PSO

Figure 16: PSO flow direction

Table 2: Analysis of time complexity and linearity			
	Time Complexity(second)	Slope	Constant
LSA Type I	0.028475	0.0518	-0.0311
LSA Type II	0.002139	0.0068	2.9974
GAM	0.002730	0.0070	2.9830
PSO	0.003062	2.2785e-04	-6.7919e-04

5. CONCLUSION

Therefore, non-linearity correction of second order system we have presented four optimization technique in this research paper out of which we can conclude Particle Swarm Optimization is the best one in terms of linearity. However, time complexity been little bit diminished. Slope of GAM little bit lesser rather than LSM Type I and Type II as shown. We have taken fourteen no of observations in Table 1 and it could have been done for PSO also, but it shows higher errors as we know that PSO is a high gain process. If increase no of observation to 50 or 100 it shows much better result than LSM and GAM. Moreover, slope of PSO much lower than LSM and GAM. Finally, we say that using PSO the voltage is much linear than LSM and GAM.

REFERENCES

- Brignell, J.E. 1991. Software Technique for sensor compensation. Sensors and Actuator A, 25-27, 29-35.
- Brignell, J.E. and N.M. White. 1994. Intelligent Sensor Systems, London UK: Institute of Physics.
- Cleghorn, C.W., A.P. Engelbrecht. Piecewise Linear Approximation of n-Dimensional Parametric Curves using Particle Swarms(draft), Department of Computer Science, University of Pretoria.
- Jafaripناه, M., B.M. Al-Hashimi and N.M. White. 2003. Load Cell Response Correction using Analog Adaptive Technique, in Proc IEEE Int. Symp. Circuits Systems (ISCAS) Bangkok, Thailand, 755.
- Jafaripناه, M., B.M. Al-Hashimi and N.M. White. 2017. Load Cell Response Correction using Analog Adaptive Technique, ESD Group, Department of Electronics and Computer Science, University of Southampton, 1BJ, UK.
- Pavee, S. 2013. Fitting Piecewise Linear Functions using Particle Swarm Optimization, Received: February 28, 2013; Revised: March 12, 2013; Accepted: June 03, 2013.
- Rajesh, D., Atrreyye, B., Suman, K.L., Amlan, P. and Achintya, Das. 2014. Signal Correction of Load Cell Output Using Adaptive Method, International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, 2(1).
- Shi, W.J., N.M. White and J.E. Brignell. 1993. Adaptive filters in Load Cell response correction, Sensors Actuators, 37-38, 280-285,
- Shu, W.Q. 1993. Dynamic Weighing under nonzero initial condition, IEEE Transaction on Instrumentation and measurement, 42(4), 806-811