Open Access Library Journal
2026, Volume 13, e14254
ISSN Online: 2333-9721
ISSN Print: 2333-9705

Advanced Diagnostics and Prognostics through Linear-in-the-Harmonics System Optimization (Identification)

Jordan McBain

Pilot Systems International, Farmington Hills, USA
Email: jmcbain@pilotsi.com

How to cite this paper: McBain, J. (2026) Advanced Diagnostics and Prognostics through Linear-in-the-Harmonics System Optimization (Identification). *Open Access Library Journal*, **13**: e14254.
https://doi.org/10.4236/oalib.1114254

Received: September 8, 2025
Accepted: January 9, 2026
Published: January 12, 2026

Abstract

Many diagnostics-focused artificial intelligence techniques in the state-of-the-art rely on combinations of least-squares-based autoregressive models and neural-network-like approaches. Prognostics algorithms can be built on top of the magic numbers generated from their result, but these magic numbers have limited physics-based meaning and a sampling of the system's behavior in each of the multitude of failed states is likely necessary. In contrast, if a system identification algorithm could accurately generate measures of the parameters of the system's physics (e.g. stiffness, capacitance, inductance), then the challenge of both diagnostics and prognostics reduces to tracking these measures against thresholds specified by the system's engineer. In this work, the author proposes a least-squares technique paralleling the linear-in-the-parameters least-square formulation but with adaptations for the realities of the frequency domain which we expose by reviewing the intuition of Fourier's seminal approach (scarcely shared). This work puts the new technique at odds with the incumbent auto-regressive model and suggests that it can be extended beyond system identification to systems optimization possibly to solve such problems as computing optimal control-law parameters. It resonates with digital twin initiatives by providing one further factor to improve the economies of scale of the effort of systems physics modelling. Beyond diagnostics and prognostics, as a generalized approach to systems optimization, the new algorithm could provide a new formulation of model predictive control.

Subject Areas

Systems Optimization, System Identification, Control Systems, Non-Linear Theory

Keywords

Least-Squares, System Identification, Control Systems, Diagnostics, Prognostics, Model Predictive Control, Fault Detection

1. Introduction

Ground military vehicles have complex dynamics that have limited the application of online *in-situ* automated fault detection techniques for in-theater use. This class of machinery is generally subject to variable operating conditions including changing speed and load. The absence of generalized, sensitive and reliable methods for reliability analysis in this segment limits industry from leveraging the well-established benefits of condition monitoring including the avoidance of major stoppages in operations, the optimization of the employment of maintenance and reliability staff, just-in-time parts inventories, etc.

This paper exposes a novel algorithm where linear-in-the-parameter least-squares curve fitting is adapted for the frequency domain and differential equations. The technique works for systems governed by stable differential equations and uses a grey-box understanding of their structure along with measures of how the system is forced and measures of how the system responds. Parameters having physical meaning (e.g. stiffness, inductance, capacitance, etc.) are computed and can be tracked against a threshold for both diagnostic and prognostic purposes.

The underlying algorithm is a system's optimization technique not limited to system identification and so it may be extended to problems like finding the optimal parameters of a control law given the relevant system's structure and a specification of how the system should behave (*i.e.* it's outputs) with respect to a specification of its forcing inputs. The details of this extension are left to a future paper.

Since this methodology shares the same mathematical foundations in optimization techniques as machine-learning algorithms, like neural networks, it might be termed an artificial-intelligence technique.

The prior art is reviewed so the espoused novelty can be contrasted against the existing least-squares system-identification incumbent (auto regressive models) and other diagnostic and prognostic solutions found in artificial intelligence. A review of the intuition of Fourier's seminal work is undertaken to augment the reader's confidence that the simplicity of the novelty herein is not mistaken.

2. Prior Art

Most mature techniques in system identification [1] focus on linear systems under the assumption they can be extended when non-linear systems are linearized. This paper proposes a new formulation of a least-squares technique for systems optimization of which system identification is just one realization; as such, the prior art in auto-regressive models is reviewed to delineate the new technique's novelty. Its combined use with machine learning is then examined. Finally, the intuition of Fourier's technique is re-examined from the ground up to justify the simplicity of this paper's approach as practical.

2.1. Least-Squares Auto Regression

The foundational system-identification approach using least-squares regression is rooted in auto-regressive (AR) models which are used to represent a stochastic discrete time process. When a sampled discrete-time signal is represented by Equation (1).

$$x_t = b + \sum_{i=1}^{n} \varphi_i x_{t-i} + \epsilon_t \tag{1}$$

A reduced representation having p parameters can be calculated by using the pseudo inverse employed in the matrix-algebra formulation of Gauss's least-squares technique using the following matrix equation. (See Equation (2))

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_p \end{bmatrix} = \begin{bmatrix} x_0 & x_{-1} & x_{-2} & \cdots \\ x_1 & x_0 & x_{-1} & \cdots \\ x_2 & x_1 & x_0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ x_{p-1} & x_{p-2} & x_{p-3} & \cdots \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \vdots \\ \varphi_p \end{bmatrix} \tag{2}$$

wherein its instantiation of the regressed form is realized using a time-shifted variant of Equation (1). This is a strange reformulation of the curve-fitting exercise for a line, quadratic or cubic. It is applicable principally to stochastic signals like the one in Figure 1. While there are variants like Auto-Regressive Exogenous (ARX), Auto-Regressive Moving Average Exogenous (ARMX), to handle signals from systems driven by external/exogenous variables, the core of the technique

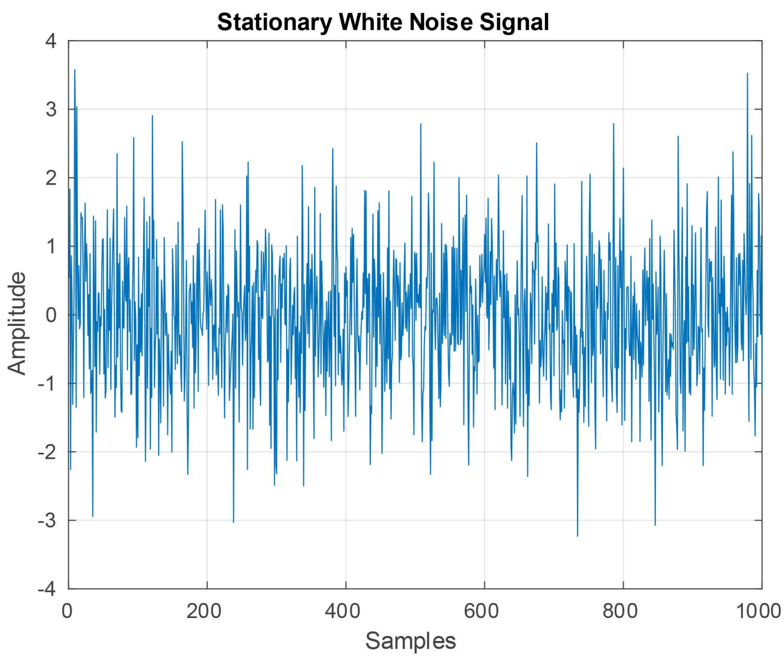

Figure 1. A stochastic signal in time.

will have utility limited to stochastic applications. Works purporting to introduce the field of system identification [2] expose almost exclusively stochastic AR approaches. Even the work in [3] entitled with the words "System Identification: A Frequency Domain Approach" does this and omits the novelty claimed in this thesis.

Because there are practical physical systems heavily governed by a stochastic element (such as gear and bearing vibration), AR models are nevertheless an important starting point in creating diagnostic and prognostic techniques applied to real *in-situ* machinery operation. Augmentations of the techniques to account for how such systems are forced (with time-varying speed and load) are essential to their successful use in practice.

Most practical machines will be governed by a heavy deterministic aspect representable using a system of differential equations – possibly with such stochastic added on.

2.2. Artificial Intelligence in Diagnostics

The field of artificial intelligence (AI) over the past 5 years has grown to include almost anything clever a computer programmer creates with software. Prior to that, AI research as it relates to the subject of diagnostics, was focused on machine learning techniques like neural networks (NN).

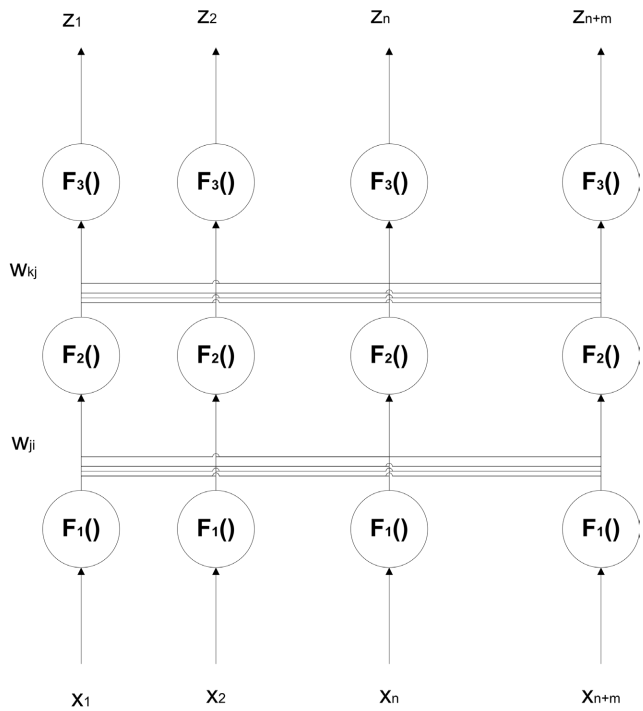

Figure 2. A neural network.

"There is nothing particularly magical about multi-layer neural networks; they implement linear discriminants, but in a space that has been mapped non-linearly." [4] The decision surface reflects this fact. (See Equation (3))

$$g_k(\vec{x}) = z_k = f\left(\sum_{j=1}^{n_H} w_{kj} f\left(\sum_{i=1}^{d} w_{ji}\vec{x}_i + w_{j_0}\right) + w_{k_0}\right) \tag{3}$$

and the "back-propagation learning algorithm" is applied to it to minimize the classification error by setting the weights in this function appropriately for the training set.

Neural networks can be considered as a network of nodes similar to the brain's neurons as in Figure 2. The transitions in the network are equivalent to the weights in the expression for the decision surface and the nodes are the non-linear mappings given by the function in Equation (3) which is selected by the user to provide good fitting of the data. This view on NNs allows for variances for forecasting through recursive NNs and other techniques like self-organizing maps and auto-encoders (See Figure 3).

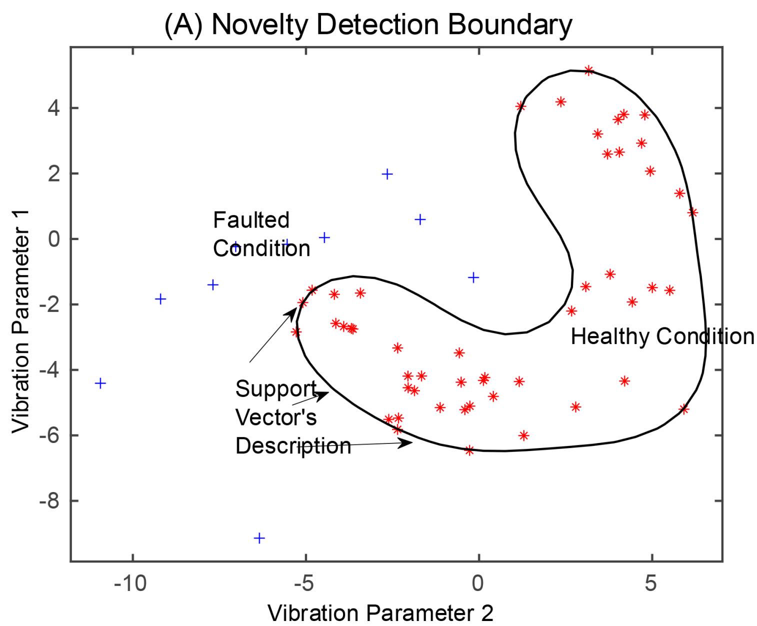

Figure 3. Novelty detection using support vectors instead of neural networks.

The notion of a neural network is an abstraction of a very concrete mathematical notion—fitting a curve to data. The network is the curve's equation represented in graphical form and its coefficients are learned through optimization techniques in an abstract parallel to how the brain's neurons strengthen over time.

Gauss's foundational least-squares curve fitting technique already discussed likely started the field. Whereas it seeks to fit a curve to represent a set of data, a neural network seeks to fit a curve to separate data or, similarly, to classify it. Figure 4 demonstrates how data from two classes (e.g. pike and walleye) might be separated with a decision boundary using the length and weight of the fish as classification parameters (in the figure: x,y). Neural networks will generally create much more sophisticated boundaries than just a line.

Using the length and weight as classification features for fish should seem quite dubious since the data spaces of each class will intersect heavily—making it difficult to find a curve that will classify/separate the two. The space of machine learning is challenged by this problem of feature selection with a great variety of techniques that range from computational mathematics to domain-specific analysis.

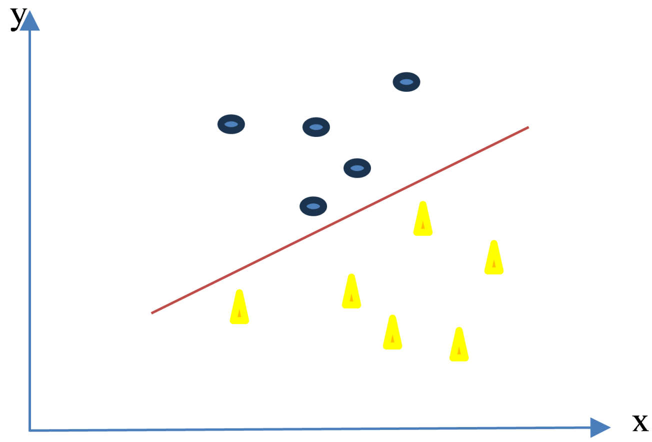

Figure 4. Neural Networks—Curve fits to separate 2D data from two classes for classification.

The biologist might add features like the number of spines on the fish's fins to its length and weight—increasing the dimensionality of the feature space which will require a greater amount of data to create good classification results (a problem known as the curse of dimensionality); while the computer scientist might create techniques like principle component analysis to reduce that data's dimensionality to a smaller one.

In the field of machine diagnostics, combinations of auto-regressive models and machine-learning techniques are often used to digest signals (e.g. vibration) measured from the physical machine into machine-learning algorithms for classification results. A special subset of machine learning, called one-class classification or novelty detection, is predominantly used to acknowledge that one cannot break production machinery in every way it might fail to gather data on how the machine "sounds" in those faulty states. Instead, behavior of the normal operating state is gathered and then fit to a decision boundary that circumscribes the data of the normal operating state. Figure 3 demonstrates the notion of this approach using support vectors as the machine-learning technique rather than neural networks.

Once the boundary is learned, it can be used to classify the data from machinery signals operating in situ to generate a novelty score. The decision boundary in Figure 3 can be seen in the plane of the graph in Figure 5 which also includes 3-dimensional exposure of the consequences of a novelty score; healthy areas of operation are shown in red while unhealthy (*i.e.* novel) ones are shown in blue with increasing intensity. If this seems ideal, the reader should be cautioned that the result is only a magic number and has limited physical meaning. For the purposes of prognostics, attempting to measure the time to failure, the novelty score has some but limited merit. Nevertheless, it does offer some possibilities for this domain.

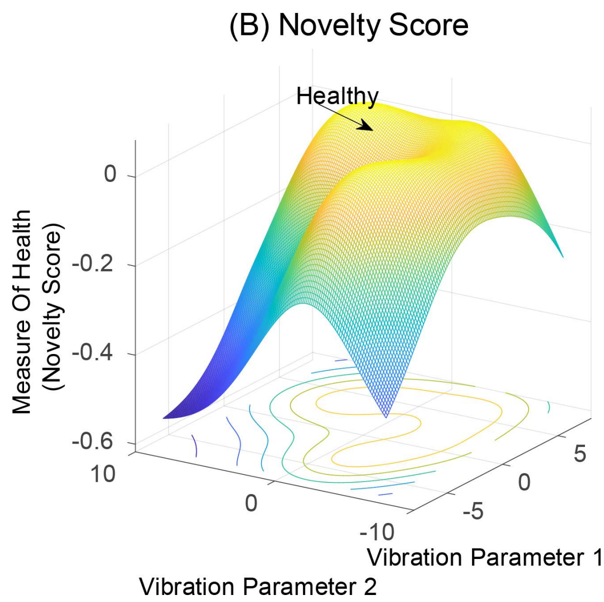

Figure 5. Novelty Score from classification boundary.

This type of application suffers from a double curse of dimensionality – since to properly characterize it one must gather both adequate data to represent the machine's operation but also over all its operating modes (e.g. for changes in load and speed for a gearbox). Creating a diagnostic rule that does not account for these parameters could miss significant classification opportunities—ones that would distinguish a machine operating under heavy duty cycles vs one that is faulted.

Many machine signals are periodic with frequency components that are phase-locked to the various shaft speeds and their harmonics, like gear-mesh frequencies. Others, like bearing signals, are tied to shaft speed with some randomness to their signals due to slippage.

These issues require care in online monitoring of steady-state machinery. When varying speed and load are introduced these concerns are greatly exacerbated. Vibration in most mechanical systems involves the periodic oscillation of energy from potential to kinetic which arises when under-damped mechanical systems are displaced from their equilibrium position and released. Oscillations will occur at the damped natural frequency of the system which can be estimated by an amalgamation of spring-like elements representing the equipment. The system will move about its frequency response curve according to changes in speed and load; vibrations will thus move through regions of damping or resonance. As Stack notes in [5], "when the machine and its bearings are healthy, these deviations in vibration are less pronounced and usually go unnoticed. However, as bearing health degrades, these deviations in machine vibration due to speed become quite significant." **Figure 6** shows a reproduction of his results—damping in undamaged machinery is largely insensitive to speed/load changes while damaged machinery is very sensitive to these parameters. A damaged bearing could be operated at maximum load and its vibratory levels could fall below the alarm threshold despite its damaged state while operation at a lower load could greatly exceed the same threshold.

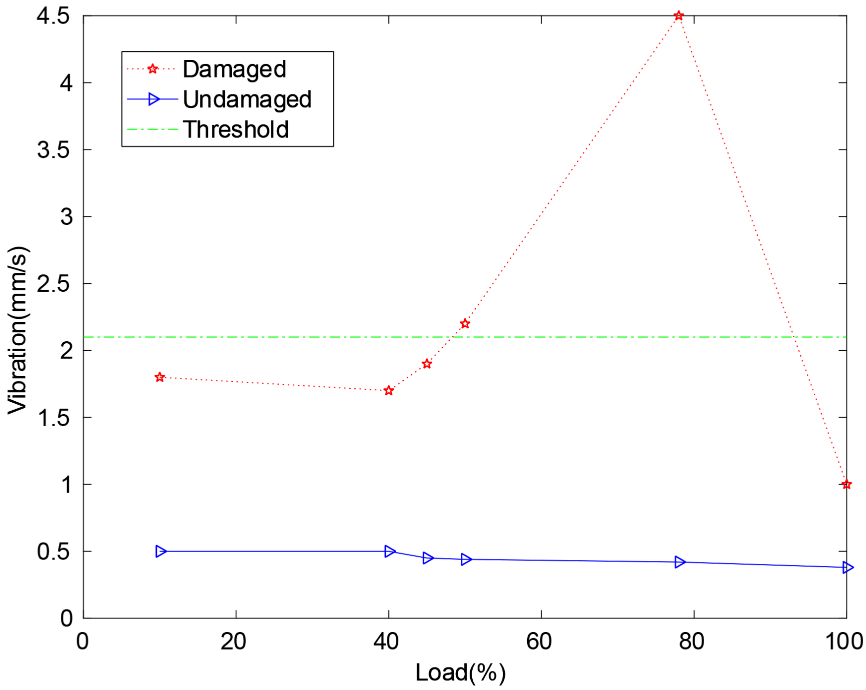

Figure 6. A health threshold in the context of healthy and undamaged machinery as a function of duty (sourced from Stack 2003).

Successful attempts to solve these problems can be found in the references but there is still much room for solutions that solve a broader range of diagnostic and prognostic challenges. The potential of system identification to take a system's inputs and outputs and infer its physical parameters (e.g. mass, spring stiffness, damping coefficient, capacitance, inductance, etc.) would be a silver bullet in this domain. A solution for prognostics follows almost immediately from the diagnostic work since the inferred parameters can be tracked to a threshold set by the system engineer.

2.3. This Work's Foundational Novelty: Parallels in Fourier's Method

This paper's thesis is mirrored in the foundational work in frequency-domain analysis by Joseph Fourier [6] from 1878. Fourier provided the foundational solution to the problem that many researchers [7] had been exploring – trying to find a way to fit a series of sinusoids to some generic curve. The problem was not identified by him but his solution is the predominant one used today – along with the almost identical one created by Laplace (exposed in rectangular coordinates rather than polar).

The modern exposition of the Fourier transform uses imaginary numbers and is explained, in part, through complex notions of Gram-Schmidt orthogonalization [8] both of which were not used by Fourier himself. In section II of [6] entitled "First example of the use of trigonometric series in the theory of heat" at page 137, Fourier states the problem of fitting some generic function with a series of cosines (where the generic function is simplified to the value of one for exemplar purposes). (See Equation (4))

$$1 = a\cos y + b\cos 3y + c\cos 5y + d\cos 7y + \&c \tag{4}$$

which is one equation with four unknowns (a, b, c and d) excluding a possible multitude of such sinusoids represented by &c. To create more equations with a commensurate number of unknowns, Fourier takes the derivative of the equation with respect to y three times. (See Equation (5))

$$\begin{aligned} 0 &= -a\sin y - 3b\sin 3y - 5c\sin 5y - 7d\sin 7y + \&c \\ 0 &= -a\cos y - 3^2 b\cos 3y - c5^2\cos 5y - 7^2 d\cos 7y + \&c \\ 0 &= a\sin y + 3^3 b\sin 3y + c5^3\sin 5y + 7^3 d\sin 7y + \&c \end{aligned} \tag{5}$$

Of course, the derivative of $\cos y$ is $-\sin y$ and successive applications of this approach will result in alternating negative signs which is analogous to the result when one multiples j (or the $\sqrt{-1}$) by itself successively. Fourier did not feel the need to use j in his exposition of the Fourier Series or Transform – although it appears in other parts of his work.

To continue the process started by Equation (5), Fourier then takes the further rudimentary step of reducing these equations by setting $y = 0$ which produces. (See Equation (6))

$$\begin{aligned} 1 &= a + b + c + d + \&c \\ 0 &= a + 3^2 b + 5^2 c + 7^2 d + \&c \\ 0 &= a + 3^4 b + 5^4 c + 7^4 d + \&c \\ 0 &= a + 3^6 b + 5^6 c + 7^6 d + \&c \end{aligned} \tag{6}$$

Some of the equations in Equation (5) disappeared when substituting in $y = 0$ so further equations need to be created by subsequent differentiation. Equation (6), having four equations and four unknowns, can now be solved.

Fourier's exposition follows the pattern of using simple examples and then merging many complex steps to achieve his final result. Fourier's collegial sharing of this simple intuition then later turns into sophisticated analysis to extend the technique from the trivial example in Equation (4) to a transform that can be applied to some generic function. In section VI of his work, entitled "Development of an arbitrary function in trigonometric series," he uses Taylor series to expand this intuition to a generic function to create the Fourier Series of a generic function. Later in the same work, he creates an early version of the well-known Fourier Transform. These derivations are not repeated here because the intuition of the basic technique was already exposed with enough detail hopefully to convince the reader that the overly simplistic nature of this work's thesis/novelty is plausible.

The intuition of Fourier's groundbreaking work resonates in the Hartley transform [9] which is an equivalent to the Fourier and Laplace Transforms except without imaginary numbers using purely sinusoids. Hartley's transform is scarcely touched in the research but it is well established theory with an equivalent Fast Hartley Transform for computational efficiency.

3. Linear-in-the-Harmonics Systems Optimization

For all the complexities that arise when trying to solve a differential equation in the manner done in most undergraduate science-technology-engineering-and-mathematics (STEM) programs, this algorithm for system identification is contrastingly quite simple.

Examining Figure 7, the first observation is trivial: notwithstanding the fact the equation is a non-linear multivariate differential equation (seemingly quite difficult), the left side equals the right side; if the left side equals the right side in the time domain, then it does so in the frequency domain and it also does so at any given frequency. By taking the squared error of one or more harmonics for a given frequency, a series of equations can be developed having cardinality at least equal to the number of unknowns.

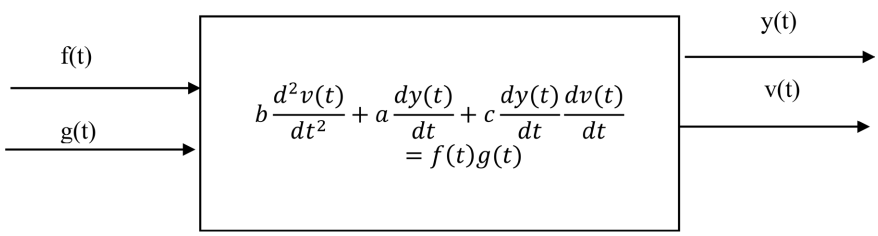

Figure 7. Multi-ınput multi-output non-linear system.

In Figure 7, the forcing functions ($f(t)$ and $g(t)$) are either measured or known and the outputs ($y(t)$ and $v(t)$) are measured. The discrete time samples of each function can be represented in the frequency domain using the Fast Fourier Transform (FFT) and the remaining operations carried out using frequency transform pairs. For instance, the frequency domain pair of the derivative of a function is the frequency domain representation of the function multiplied by $j\omega$. This is true for successive derivative operations on a function; therefore, the nth derivative's frequency domain representation of a function is the frequency domain representation of the function multiplied by $(j\omega)n$.

The non-linearity in Figure 7 stems from the multiplication of the derivatives of $y(t)$ and $v(t)$. In the frequency domain, the multiplication of two signals is the convolution of the signal's frequency domain representations. The utility of Fourier's result is that a function is broken down into a sum of sinusoids. Multiplying two series of sinusoids together will consist of the sum of the plethora of sinusoids. The multiplication of one sinusoid against another produces the harmonics which are the sum and difference of the sinusoid's frequency—the foundational mathematics of amplitude modulation. The underlying elementary trig identity is shown in Equation (7); it may suffice to say that, notwithstanding the complex name, the convolution operation can be reduced to elementary trig identities.

$$\cos\alpha\cos\beta = \frac{1}{2}\left(\cos(\alpha-\beta)+\cos(\alpha+\beta)\right) \tag{7}$$

When the frequency transform of sampled signals is known, the convolution operation on them is simple to compute (shown in Equation (8)).

$$(f*g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m] \tag{8}$$

The left side of frequency domain representation of the equation in **Figure 7** can therefore be reduced into numerical coefficients and the unknown system parameters *b*, *a* and *c*. If a machine were represented by this equation, these parameters would be things like capacitance, inductance, stiffness, etc.

The right side of the frequency domain representation of the equation in **Figure 7**, can be similarly reduced but to harmonics with no unknowns.

If the right side equals the left side at every frequency, then we should be able to pick frequencies (harmonics) and put the left side at odds with the right side in the least-squares sense at those frequencies. We can select as many frequencies as there are unknowns and create a system of simple algebraic equations in (a,b,c) with numerical parameters; we might take more frequencies than there are unknowns and apply this algorithm to produce an over-determined system which can account for noisy signals further than simply using a set of equations having cardinality equal to the unknowns.

The system under examination in **Figure 7** has only one governing differential equation. There might be a plurality of such equations with a plurality of unknown parameters within them. The process will be the same but repeated for each such equation. The result is the same—a system of algebraic equations with unknowns that can be solved for with rudimentary matrix algebra.

Only one non-linearity is present in **Figure 7** in the form of the product of the derivative of two signals. Other non-linearities can be handled much in the same way using other frequency-domain transform pairs.

The technique works so long as the system is stable. No production machinery should be governed by an unstable equation; while the machine's eventual deterioration to such a state should be indicated by tracking the inferred parameters long before that happens.

The prognostics and diagnostics rules can therefore be expressed in thresholds with numbers having physical meaning (stiffness, capacitance, etc.). Prognostics become especially simple since one need but extrapolate how long the parameter appears to track to a faulted threshold defined by the system engineer and likely requires minimal further machine-learning algorithms.

The technique is termed linear-in-the-harmonics optimization since it parallels linear-in-the-parameter least-squares curve fitting but adapted for the realities of the Frequency domain. A least-squares curve fit can be achieved for a non-linear curve by first linearizing it. For instance the curve

$$y = x * e^{x}$$

Can be linearized by taking its natural logarithm

$$\ln(y) = \ln x + x$$

and curve fitting for a set of data for ordered pairs <*x*,*y*> can then be calculated in least-squares fashion.

By analogy, curve-fitting for a differential equation can be reduced to a linear optimization problem by evaluating the system response at various harmonics and setting the model's structure ("left side" in the discussion above) at odds with the system response ("right side" in the discussion above) in the least-square sense. An example will help.

4. Numerical Example

In Table 1, sample MATLAB code is depicted to ease the reader's burden in reproducing the author's results. A theoretical second-ordered system (mass = 0.05, viscous damping = 2, stiffness = 2.3) is numerically "forced" by a multi-sine and the technique's ability to successfully reproduce some of the parameters is demonstrated. The following system is used

$$\begin{aligned} & m\frac{\mathrm{d}^2 x(t)}{\mathrm{d}t^2} + b\frac{\mathrm{d}x(t)}{\mathrm{d}t} + kx(t) \\ & = 10.25\sin(2\pi * 100t) + 20\sin(2\pi * 198t) + 30\sin(2\pi * 256t) \end{aligned}$$

The code evaluates the system's response at the three forcing frequencies (100 Hz, 198 Hz, 256 Hz). First we select the harmonics at the forcing frequencies from the spectra calculated using an FFT. As an alternative, to an FFT, we also calculate the Fourier integrals for each of the frequencies whose parameters will also be used in an alternative formulation of the proposed system identification technique. Each harmonic, whether calculated by FFT or Fourier integral, produces two pieces of information (phase/amplitude) which makes the system identification technique overdetermined when using three forcing frequencies.

Table 1. MATLAB code for simple linear system.

```
% For the standard differential equation m*x''+b*x'+k*x=F(t)
m =0.05; b = 2; k = 2.3;

%With forcing function:
forcFreq= [100 198 256];
u = @(t) 10.25*sin(2*pi*forcFreq(1)*t)+20*sin(2*pi*(forcFreq(2))*t)-30*sin(2*pi*(forcFreq(3))*t);

dt=0.001;
endTime=8;%8 seconds worth of data

numSamples=2.^((nextpow2(endTime/dt)));%Fourier transform likes powers of 2
endTime=numSamples*dt;
t = 0:dt:(endTime-dt);
```

```
u_t=u(t);

%% Solve with ODE45
% {
% Define system matrices for a second-order system
A = [0 1; -k/m -b/m];
B = [0; 1/m];
C = [1 0];
D = 0;

% Function for state derivatives
sys_ode_func = @(t, x) A*x + B*u(t)';

% Simulate using ode45
x0 = [0; 0];
[t_ode, x_ode] = ode45(sys_ode_func, t, x0);
x_ode = C*x_ode';
% }

%% Alternatively: Solve with TF (Control Systems Toolbox)
% {
% Define transfer function
num = [1];
den = [m b k];
sys_tf = tf(num, den);

% Simulate using lsim
[x_cst, t_lsim_out] = lsim(sys_tf, u(t), t);

%%
%Get a frequency vector

% If you have the Control Systems Toolbox;
x=x_cst';
%If you don't have the Control Systems Toolbox
%x=x_ode;

freqs =2*pi/dt*((-numSamples/2 : numSamples/2 - 1) / numSamples )';

%% Calculate FFT Parameters
%Above we evaluated the leftside by taking the fft of the left side fed
%with the signal x. Here we compute the fft of the signal x and then compute the
%left side by multiplying by jw (the derivative operator).
leftSide = fft(filter(m*[1,-2,1]/dt^2 + b*[1,0,-1]/2/dt +k*[0,1,0],1,x));
leftSide = fftshift(leftSide);
rightSide = fftshift(fft(u(t)));

x_fft=fftshift(fft(x));
dxdt=1i*freqs.*x_fft;
d2xdt2=1i*freqs.*dxdt;

[ignore,freq1_idx]=min(abs(freqs-2*pi*forcFreq(1))); %Not very efficient... good enough
```

```
for a script
[ignore,freq2_idx]=min(abs(freqs-2*pi*forcFreq(2)));
[ignore,freq3_idx]=min(abs(freqs-2*pi*forcFreq(3)));

rightSide_freq1_imag=rightSide(freq1_idx);
rightSide_freq2_imag=rightSide(freq2_idx);
rightSide_freq3_imag=rightSide(freq3_idx);

xfft_freq1_imag=x_fft(freq1_idx);
xfft_freq2_imag=x_fft(freq2_idx);
xfft_freq3_imag=x_fft(freq3_idx);

a_fft = real( xfft_freq1_imag );
c_fft = imag( xfft_freq1_imag );
d_fft = real( rightSide_freq1_imag );
e_fft = imag( rightSide_freq1_imag );

f_fft = real( xfft_freq2_imag );
g_fft = imag( xfft_freq2_imag );
h_fft = real( rightSide_freq2_imag );
l_fft = imag( rightSide_freq2_imag );

n_fft = real( xfft_freq3_imag );
o_fft = imag( xfft_freq3_imag );
p_fft = real( rightSide_freq3_imag );
q_fft = imag( rightSide_freq3_imag );

%% Compute parameters via the Fourier Series

%We use the Fourier Series here
a = (2/endTime) * trapz( t, x .* sin( forcFreq(1) * 2 * pi * t ) );
c = (2/endTime) * trapz( t, x .* cos( forcFreq(1) * 2 * pi * t ) );
d = (2/endTime) * trapz( t, u_t .* sin( forcFreq(1) * 2 * pi * t ) );
e = (2/endTime) * trapz( t, u_t .* cos( forcFreq(1) * 2 * pi * t ) );

f = (2/endTime) * trapz( t, x .* sin( forcFreq(2) * 2 * pi * t ) );
g = (2/endTime) * trapz( t, x .* cos( forcFreq(2) * 2 * pi * t ) );
h = (2/endTime) * trapz( t, u_t .* sin( forcFreq(2) * 2 * pi * t ) );
l = (2/endTime) * trapz( t, u_t .* cos( forcFreq(2) * 2 * pi * t ) );

n = (2/endTime) * trapz( t, x .* sin( forcFreq(3) * 2 * pi * t ) );
o = (2/endTime) * trapz( t, x .* cos( forcFreq(3) * 2 * pi * t ) );
p = (2/endTime) * trapz( t, u_t .* sin( forcFreq(3) * 2 * pi * t ) );
q = (2/endTime) * trapz( t, u_t .* cos( forcFreq(3) * 2 * pi * t ) );


w1 = freqs(freq1_idx);
w2 = freqs(freq2_idx);
w3 = freqs(freq3_idx);


diffedNorm=@(sysParams,a,c,d,e,w,x0,xd)((-sysParams(1)*w.^2*c+sys-
```

```
Params(2)*w*a+sysParams(3)*c-e).^2+(-sysParams(1)*w.^2*a-sysParams(2)*c*w+sys-
Params(3)*a-d).^2);
fForMin=@(sysParams)(diffedNorm(sys-
Params,a,c,d,e,w1,x(1),dxdt(1))+diffedNorm(sys-
Params,f,g,h,l,w2,x(1),dxdt(1))+diffedNorm(sysParams,n,o,p,q,w3,x(1),dxdt(1)));

%m =0.05; b = 2; k = 2.3;
firstGuess=[1,4,6];
sol=fminsearch(fForMin,firstGuess);

sys_tf = tf(num, sol);


fprintf('Simulated Parameters: mass( %f ), damping( %f ), stiffness( %f )\n',m,b,k);
fprintf('Inferred Parameters (fminsearch with fourier series): mass( %f ), damping( %f ),
stiffness( %f )\n',sol(1),sol(2),sol(3));

X = [-c*w1^2,w1*a,c;-a*w1^2,-c*w1,a;-g*w2^2,w2*f,g;-f*w2^2,-g*w2,f;-o*w3^2,w3*n,o;-
n*w3^2,-o*w3,n];
y = [e;d;l;h;q;p];

Sol_mls=inv(X'*X)*X'*y;
fprintf('Inferred Parameters (matrix least square with fourier series): mass( %f ), damp-
ing( %f ), stiffness( %f )\n',Sol_mls(1),Sol_mls(2),Sol_mls(3));

X = [-c_fft*w1^2,w1*a_fft,c_fft;-a_fft*w1^2,-c_fft*w1,a_fft;-g_fft*w2^2,w2*f_fft,g_fft;-
f_fft*w2^2,-g_fft*w2,f_fft;-o_fft*w3^2,w3*n_fft,o_fft;-n_fft*w3^2,-o_fft*w3,n_fft];
y = [e_fft;d_fft;l_fft;h_fft;q_fft;p_fft];

Sol_mls_fft=inv(X'*X)*X'*y;
fprintf('Inferred Parameters (mls with fft): mass( %f ), damping( %f ), stiff-
ness( %f )\n',Sol_mls_fft(1),Sol_mls_fft(2),Sol_mls_fft(3));

% Simulate using lsim
[x_cst_inferredParams, t_lsim_out_inferreed] = lsim(sys_tf, u(t), t);


%% Auto-regressive model
%{
 arData=iddata(x',u_t',dt);
 sol_armax=armax(arData,[2 2 1 1]);
%}


%% Compare ODE solution vs. Control System Toolbox Output vs. System Forced with
Inferred Parameters
%for a lark
% {
figure;

plot(t_ode, x_ode, 'b-', t_lsim_out, x_cst, 'r--',t_lsim_out_inferreed,x_cst_inferred-
Params,'g-.')
```

```
legend('ODE45', 'Control System Toolbox','From Inferred Params')'
title('Comparison of ode45 and lsim')
% }
```

The inferences of the system's parameters (m,b, k) are then calculated by least-squares matrix formulation and alternatively using MATLAB's non-linear optimization function fminsearch. The results are demonstrated in Table 2.

The technique is not limited by stationarity but is heavily limited by the assumption of periodicity inherent in an FFT and even Fourier-integral analysis. Mass and viscous damping are adequately reproduced but stiffness is heavily misrepresented. The system's forcing is demonstrated in Figure 8. Red and blue demonstrate the system's response using MATLAB's control systems toolbox and ODE45 solver results respectively; while the green plot demonstrates the system forced with the inferred parameters rather than idealized ones. The system's response in this analysis does not exclude the transient response which limits the

Table 2. MATLAB estimated parameter results.

	Mass	Damping	Stiffness
Simulated Parameters	0.05	2.00	2.3
Inferred (fminsearch with Fourier Integrals)	0.058970	2.234951	12.09881
Inferred (matrix least squares with Fourier Integrals)	0.061936	2.237835	4402.68
Inferred (matrix least squares with FFT)	0.061881	2.241650	4426.039

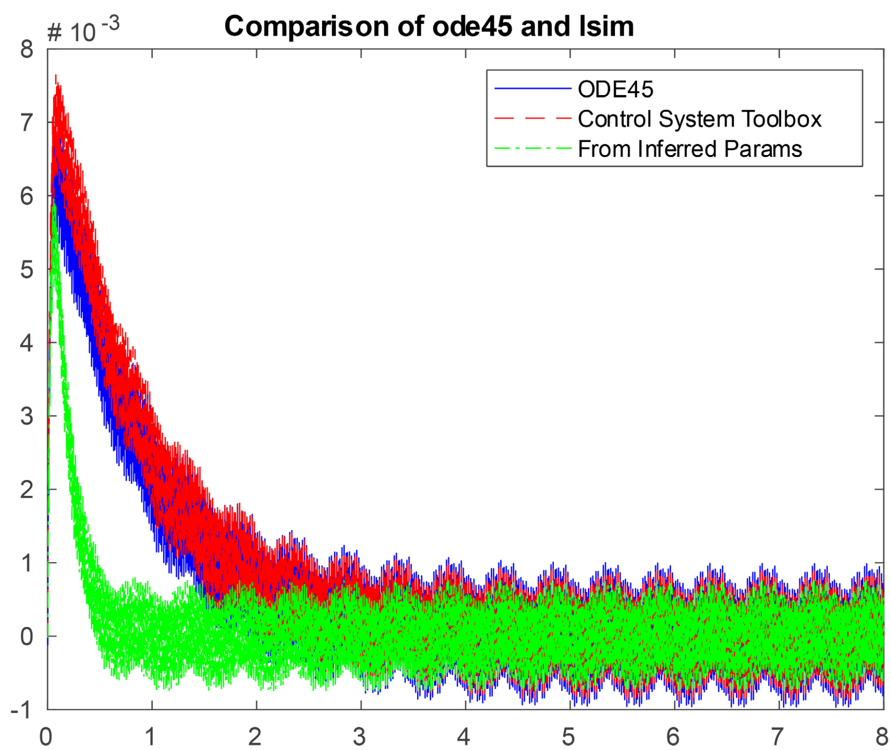

Figure 8. Simulation of system (green shows the system forced with inferred parameters).

accuracy of the measured frequencies and ultimately the quality of the inferred parameters.

Finally the sample code demonstrates the results of an auto-regressive fit.

5. Conclusions

The intuition of Fourier's foundational transform and Hartley's modern representation of it provides confidence in a similarly developed technique for least-squares system identification. The parameters of a system governed by a differential equation can be inferred from measures of how the signal is forced if the system's response is also measured.

The technique can be abstracted beyond system identification to system optimization; for instance, it might be used to find the ideal look-up table or control-law parameters given a specification of how the system should respond in the context of how it is forced or it might be used to reformulate the model-predictive control solution.

It produces numerical measures having physical meaning like capacitance and stiffness. As such, the diagnostic and prognostic technique then is simplified to a systems engineer specifying a desired threshold in those terms. It should be superior to many established techniques in the space of fault diagnosis [10]-[17].

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] Tangirala, A. (2018) Principles of System Identification: Theory and Practice. CRC Press.

[2] Ljung, L. (1999) System Identification: Theory for the User. Prentice Hall.

[3] Pintelon, R. (2012) System Identification: A Frequency Domain Approach. Wiley.

[4] Duda, R., Hart, P. and Stork, D. (2001) Pattern Classification. 2nd Edition, Wiley.

[5] Stack, J.R., Habetler, T.G. and Harley, R.G. (2003) Effects of Machine Speed on the Development and Detection of Rolling Element Bearing Faults. *IEEE Power Electronics Letters*, **1**, 19-21. https://doi.org/10.1109/lpel.2003.814607

[6] Fourier, J. (1878) The Analytical Theory of Heat. Cambridge University Press.

[7] Valentinuzzi, M. (2016) Highlights in the History of the Fourier Transform. IEEE Pulse.

[8] Sjoden, G. (2015) Gram-Schmidt Orthogonalization and Fourier Series in Foundations in Applied Nuclear Engineering Analysis. 2nd Edition.

[9] Bracewell, R. (1986) The Hartley Transform. Oxford University Press, Inc.

[10] McBain, J. (2024) Linear-in-the-Harmonics Systems Optimization: Realizing A Technique for System Identification for Systems Linear and Non-Linear. MIMO and SISO, US Patent Office Provisional Application # 63/661,717.

[11] McBain, J. and Timusk, M. (2009) Fault Detection in Variable Speed Machinery: Statistical Parameterization. *Journal of Sound and Vibration*, **327**, 623-646. https://doi.org/10.1016/j.jsv.2009.07.025

[12] McBain, J. and Timusk, M. (2011) Feature Extraction for Novelty Detection as Applied to Fault Detection in Machinery. *Pattern Recognition Letters*, **32**, 1054-1061. https://doi.org/10.1016/j.patrec.2011.01.019

[13] McBain, J. and Timusk, M. (2012) Software Architecture for Condition Monitoring of Mobile Underground Mining Machinery: A Framework Extensible to Intelligent Signal Processing and Analysis. 2012 *IEEE Conference on Prognostics and Health Management*, Denver, 18-21 June 2012, 1-12. https://doi.org/10.1109/icphm.2012.6299543

[14] McBain, J. and Timusk, M. (2014) Cross Correlation for Condition Monitoring of Variable Load and Speed Gearboxes. *Journal of Industrial Mathematics*, **2014**, Article ID: 543056. https://doi.org/10.1155/2014/543056

[15] McBain, J., Lakanen, G. and Timusk, M. (2013) Vibration- and Acoustic-Emissions Based Novelty Detection of Fretted Bearings. *Journal of Quality in Maintenance Engineering*, **19**, 181-198. https://doi.org/10.1108/13552511311315977

[16] McBain, J. and Timusk, M. (2012) System Identification for Fault Detection in Variable Speed and Load Machinery. *International Journal of Condition Monitoring*, **2**, 32-39. https://doi.org/10.1784/204764212804729723

[17] Timusk, M.A., Lipsett, M.G., McBain, J. and Mechefske, C.K. (2009) Automated Operating Mode Classification for Online Monitoring Systems. *Journal of Vibration and Acoustics*, **131**, Article ID: 041003. https://doi.org/10.1115/1.3142871