# Functional Roots and Manufacturing Tasks

Ingo Schwab
Karlsruhe University of Applied Sciences
Karlsruhe, Germany
Ingo.Schwab@hs-karlsruhe.de

Norbert Link
Karlsruhe University of Applied Sciences
Karlsruhe, Germany
Norbert.Link@hs-karlsruhe.de

***Abstract*—Many production processes consist of repetitive, almost identical sub-processes. Process models are needed for state estimation and control purposes. Models are frequently formed from an analysis of input-output data relations of the overall process. For a repetitive process, the model of the repeated process is a functional root of the relation. Functional roots are introduced and symbolic approaches are presented. We propose to find functional roots via Symbolic Regression to model repetitive processes. As a first proof of principle we show the suitability of this approach with two basic and well-known problems in the scientific field of physics and nonlinear dynamics. The exact solutions of these problems are available from textbooks and can be used to assess the results of our approach. The first step in our project work therefore is to develop suitable concepts and technologies. The next steps will include analyzing real world data in cooperation with our project partners.**

***Keywords- Symbolic Regression; Manufacturing; Functional Roots; Machine Learning.***

## I. INTRODUCTION

Many manufacturing tasks and processes are composed of a repetition of some simple process steps, since the necessary power of the repeated process must only be a fraction compared to the power needed in a single-step process. In fact, repeating manufacturing tasks represent an important group of manufacturing tasks and are of high practical relevance.

One of the problems is that manufacturing conditions restrict the observation of the material properties during the process, which therefore can often not be quantified. In such cases only the initial and final state of the material or work piece is known. The knowledge of the intermediate material qualities is mandatory for optimal process control. It is represented by a process model, which has to be established for the process under consideration.

There are several methods used to model the dynamics of nonlinear complex systems [1]. Conceptually, they can be split into two classes. The first class includes prior domain knowledge from human experts. For example numerical simulations like finite elements or phase field methods simulate the behavior of systems with domain knowledge from human experts. The second approach is to use phenomenological or general base function models which try to fit the observed behavior of the systems as good as possible. The latter approach includes many machine learning, data mining and statistical methods.

The second class can be further refined in modeling via symbolic [2] (e.g., general formula expressions) and subsymbolic (e.g., dedicated base function class, support vector machines or neural networks) representations. Symbolic learning representations can be interpreted by human domain experts and they can help to understand the process in a more formal way. Therefore this class does not only aim to model the system behavior. Sometimes the human experts are able to identify previously unknown facts of the observed process.

In contrast subsymbolic representations are black boxes. In most cases it is very difficult or impossible to interpret the behavior of the learnt representation. In our approach, we interpret mathematical formulas as one form of symbolic representation which can be used to gain additional insight into the system behavior.

The remaining part of the paper is organized as follows: In Section 2, we introduce the relation between industrial processes and functional roots. Section 3 gives a summary of the background and of related work. Additionally the proposed method is further described. Section 4 introduces the sample experiments and Section 5 the results of the method application. A summary is drawn in Section 6 with an outlook to future work.

## II. INDUSTRIAL PROCESSES AND FUNCTIONAL ROOTS

One of our project tasks is to develop algorithms which are able to model the behavior of manufacturing processes. In the first steps we identified an important class of recurring problems which will be described in the following part of this subsection.

Technical processes like steel rolling or annealing are often recursive repetitions of some simple processes where the repeated application fulfills the original task. The main reason for such a recursive process is that the elementary process is much easier to handle. Figure 1 shows a schematic example of a steel mill. Stripes of metal are rolled in a sequence of up to seven identical stands where the task is to reduce their initial thickness of some centimeters down to some millimeters. That means that the resulting semi-

 ISBN: 978-1-61208-224-0 

manufactured products of a subprocess are the input of the next almost identical sub-process. This continues until the target properties are reached.

In Figure 1, a block of steel with known property $x_{in}$ is transformed by *n* stands to a stripe with the measurable property $x_{out}$.

The total process *F* can be modeled as a whole, but revealing a description of a single stand $f_i$ is equivalent to computing functional roots of *F*. Intermediate values $x_i$ are not accessible, but might be important to know for optimal process control.

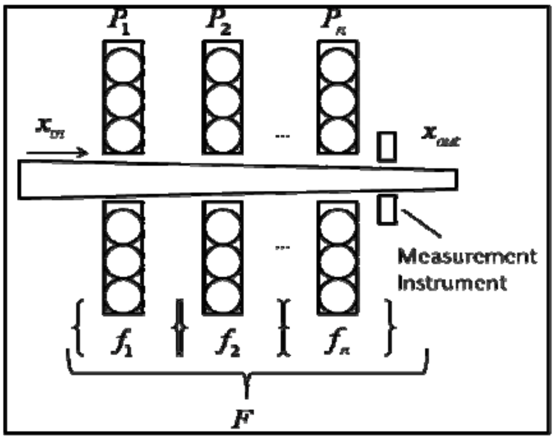


Figure 1. Model of a steel mill.

Due to technical reasons, it is impossible to measure some parameters like the profile of the stripes between the stands and the intermediate processing steps. However, this information is essential for optimal process control. Therefore, a model of a single stand can be generated from the measured values of the incoming and outgoing material and the fact that the transformation occurred in a number of identical steps. In [4], the whole process line is successfully modeled by a neural network. In [5], system identification of *F* and $f_i$ is done with neural networks. The disadvantage of this method is that the results are subsymbolic and cannot be interpreted by a human expert.

## III. BACKGROUND AND RELATED WORK

### A. Functional Roots

In one sense, the concept of functional roots (aka iterated functions) is the inverse problem to the well-known compositions of a function with itself. The function *f(x)* is not known, but its composition with itself is given. For example, what is *f(x)* such that *f(f(x))=F(x)*, where *F(x)* is a given function. This question is an important part of the theory of functional equations and the areas of application appear in various fields such as computer science (e.g., recursions), dynamic systems or chaos theory. Little mathematical theory is known to find functional roots. It can be shown that functional roots of all orders exist for at least all continuous and strictly increasing real-valued functions [7]. Theoretical solutions for the problem do only exist for specific cases, such as monotonic functions. There is no formal way to find solutions for the general case.

Nevertheless, they have practical significance and few tools can solve them. Symbolic Regression is one solution method [8] and in this paper we present our first results.

Definition: Given an arbitrary function $F(x): \Re \rightarrow \Re$, the function *f(x)* with f(f(x))≡F(x) is called a functional or iterative root of F.

Higher order roots can be defined as $f^k(x) = f(f(...f(x)...)) \equiv F(x)$ and the function $f = F^{1/k}$ is a *k*-th iterative root of F.

Some simple examples are shown in Table 1.

| Functional Root | Solution |
|---|---|
| $F(x) = x$ | $f(x) = x$ |
| $F(x) = x+1$ | $f(x) = x + \frac{1}{2}$ |
| $F(x) = x^2$ | $f(x) = \|x\|^{\sqrt{2}}$ |
| $F(x) = x^4$ | $f(x) = x^2$ |

Table 1. Functional roots.

To find a functional root to a problem seems on the first sight appealing because of its apparent simplicity and its natural idea. But, already the simple function $F(x) = x^2 - 2$ requires deep mathematical insight to be solved. In [8], it was shown that one analytical solution is

$f(x) = 2\cos(\sqrt{2}\cos^{-1}(\frac{x}{2}))$, which is not intuitive at first sight.

As a final remark, it should be mentioned that functional roots represent a universal concept and their use is not limited to the optimization of industrial processes. Applications range from data analysis to chaos theory.

### B. Classical Regression Analysis and Symbolic Regression

Regression analysis [9] is one of the basic tools of scientific investigation enabling identification of functional relationship between independent and dependent variables. The general task of regression analysis is defined as identification of a functional relationship between the independent variables **x** = [$x_1$, $x_2$, … , $x_n$] and dependent variables **y** = [$y_1$, $y_2$, … , $y_m$], where *n* is a number of independent variables in each observation and *m* is a number of dependent variables.

The task is often reduced from an identification of a functional relationship *f*() to an identification of the parameter values of a predefined (e.g., linear) function. That means that the structure of the function is predefined by a human expert and only the free parameters are adjusted. From this point of view Symbolic Regression goes much further.

Like other statistical and machine learning regression techniques Symbolic Regression also tries to fit observed and recorded experimental data. But unlike the well-known regression techniques in statistics and machine learning Symbolic Regression tries to identify an analytical mathematical description and it has more degrees of freedom in building it. A set of predefined (basic) operators is defined (e.g., add, multiply, sin, cos) and the algorithm is mostly free in concatenating them. Unlike the classical regression approaches which optimize the parameters of a predefined structure also the structure of the function is free and the algorithm both optimizes the parameters and the structure of the basis functions.

There are different ways to represent the solutions in Symbolic Regression. For example informal and formal grammars have been used in Genetic Programming to enhance the representation and the efficiency of a number of applications including Symbolic Regression [10].

Since Symbolic Regression operates on discrete representations of mathematical formulas non-standard optimization methods are needed to fit the data. The main idea of the algorithm is to focus the search on promising areas of the target space while abandoning unpromising solutions (see [3] for more details). In order to achieve this, the Symbolic Regression algorithm uses the main mechanisms of Genetic and Evolutionary Algorithms. In detail they are mutation, crossover and selection [6] and they are used to operate on an algebraic mathematical representation.

This representation is encoded in a tree [6] (see Figure 2). Both the parameters and the form of the equation are subject to search in the target space of all possible mathematical expressions of the tree.

In Symbolic Regression, many initially random symbolic equations compete to model experimental data in the most promising way. Promising are those solutions which are a good compromise between correct prediction quality of the experimental data and the length of the symbolic representation.

The operations are nodes in the tree (Figure 2 represents the formula 6x+2) and can be mathematical operations such as additions (add), multiplications (mul), abs, exp and others. The terminal values of the tree consist of the function's input variables and real numbers. The input variables are replaced by the values of the training data set.

Mutation in a symbolic expression can change the mathematical type of formula in different ways. For example a div is changed to add, the arguments of an operation changed (e.g., change 2*x to 3*x), delete an operation (e.g., change 2*$x$+1 to 2*x), or add an operation (e.g., change 2*x to 2*x+1).

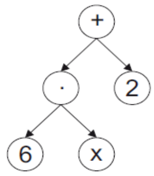


Figure 2. Tree representation of the equation 6x+2.

The fitness objective in Symbolic Regression, like in other machine learning and data mining mechanism, is to minimize the regression error on the training set. After an equation reaches a desired quality level of accuracy, the algorithm returns the best equation or a set of good solutions (the pareto front). In many cases the solution reflects the underlying principles of the observed system.

### C. *Proposed Method*

In this article, we introduce an approach which uses Symbolic Regression to model the intermediate processing steps of manufacturing tasks. Mathematically, this is equivalent to the problem of computing iterative or functional roots: Given the equation *F(x)=f(f(x))* and an arbitrary function *F(x)* we seek a solution for *f(x)*. The major advantage of this approach is the interpretability of the identified solutions.

## IV. SAMPLE EXPERIMENTS

In the following two subsections, we give a brief description of the two application scenarios of our first experiments. It should be noted that the next stage of our project is to evaluate the quality of the proposed methodologies on real-world data from industrial partners

### A. *Free Fall*

The Free Fall textbook problem belongs to the elementary problems in physics and every first-year student in physics will probably be familiar with it. Nevertheless, we used it as starting point to gain a better understanding of the developed methodologies and functional roots.

In a nutshell, the free fall describes a vertical motion of an object falling a small distance close to the surface of a planet. It is a good approximation in air as long as the force of gravity on the object is much greater than the force of aerodynamic resistance, or equivalently the object's velocity is always much smaller than the stationary velocity.

### B. *The Logistic Function*

A discrete map is the inverse to a functional root and is basically a sequence defined by the successive compositions of a function with itself. If, for example, we consider a function *f* from *R* to *R,* for each value in the domain we can define a sequence ( $x, f(x), f^2(x), ..., f^n(x)$ ), whereby

$f^k(x)$ describes the $k$ times concatenation $f \circ f \circ f ... \circ f$.

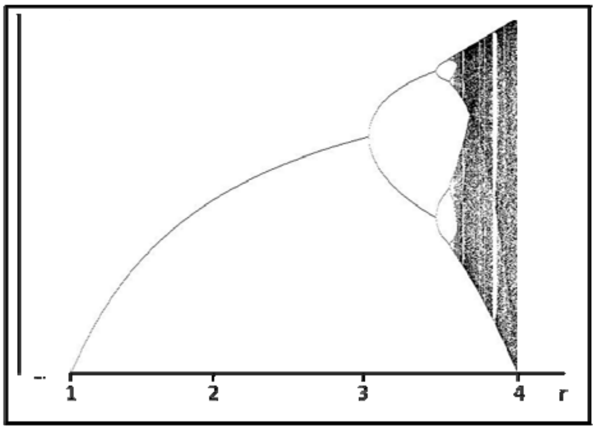


Figure 3. Bifurcation diagram.

There are many reasons why we may be interested in a sequence of this sort. For instance, the iteration of a suitable function can be successful in describing an event in the real world which is considered at discrete steps, such as the growth of a population of rabbits through its generations.
The Logistic Function is defined as

$$x_n = rx_{n-1}(1 - x_{n-1}), \ r > 0. \quad (1)$$

It is easy to check that this is the equation of an upside-down parabola, which goes through the origin and the intercepts the x-axis at x = 1. This function is a good model of growing populations, but it has also peculiar mathematical properties.

The function has three different defined ranges with different behavior.
$0 \le r \le 1$: the function converges to 0.
$1 < r \le 3$: the function converges to the attractor $1 - 1/r$.
$3 < r \le 4$: the function shows a periodic-doubling bifurcation. It starts with one attractor and approaches chaos via period doubling.

The logistic function is particularly interesting when $r > 2 + \sqrt{5}$. In this case, the dynamic system shows a deterministic chaotic behavior. That means that the system behavior is very sensitive to its initial conditions and infinitesimal variations for a dynamic system lead to large variations in behavior.

Figure 3 shows the Bifurcation or Feigenbaum diagram. The bifurcation parameter $r$ is shown on the horizontal axis of the plot and the vertical axis shows the possible long-term population values of the logistic function. Only the stable solutions are shown here, there are many other unstable solutions which are not shown in this diagram. The bifurcation diagram shows the forking of the possible periods of stable orbits from 1 to 2 to 4 to 8 etc. Each of these bifurcation points is a period-doubling bifurcation.

## V. EXPERIMENTS AND RESULTS

In our project, we have developed a Symbolic Regression framework. Additionally we adapted this algorithm to search for solutions for functional roots ($F(x)=f(f(x))$).
One of the main challenges posed in this paragraph is to modify algorithms to determine mathematical equations which are able to interpolate observed systems behavior. These data were measured at different points in time. In other words, we want to learn a function which is able to interpolate the dynamics of a system for nonlinear behavior.

### A. Free Fall

As a starting point of our project we analyzed the well-known physical free-fall problem. The experiment setup is easy: An object is falling from attitude $h_0$ to $h_1$. On level $h_0$ it has the velocity $v_0$ and on level $h_1$ $v_1$. The starting velocity is varied and the resulting speed is measured on level $h_1$.
With knowledge of the necessary physical laws it is easy to find the correct answer. E.g., with knowledge of the energy theorem, attitude m and gravitation a the function is

$$\frac{1}{2}mv_0^2 + ma(h_1 - h_0) = \frac{1}{2}mv_1^2 \quad (2)$$

The task was to determine a formula which satisfies the following conditions for time

$$t_m = t_0 + \frac{\Delta t}{2}$$

$$(v_1, h_1) = g(v_m, h_m) = g(g(v_0, h_0)) = f(v_0, h_0) \quad (3)$$

Replacing g with the function:

$$(v_m, h_m) = g(v_0, h_0) \text{ with } \begin{array}{l} v_m = v_0 + \frac{1}{2}a\Delta t \\ h_m = h_0 + \frac{1}{2}v_0\Delta t + \frac{1}{8}a\Delta t^2 \end{array} \quad (4)$$

the iterated function is $f$ and $g$ is the iterated function of $f$.

In a first step we generated a training set of 40 learning examples.

Then we used the Symbolic Regression algorithm to search for the solution. The operation set contained addition, subtraction, multiplication, division, sine, cosine, exponential, logarithm function. The terminal values consisted of the function's input variables and real numbers.

The main task was to learn a functional root for this function. Several experiments showed that the developed Symbolic Regression system had no problem in finding the iterated function for this first sample experiment.

It was good starting point, but a more complex problem was needed.

### B. The logistic function

The logistic function is defined as $x_n = rx_{n-1}(1 - x_{n-1})$.

At first, we generated a training set of 70 data sets. 4 times r was varied (see Figure 4). The vertical lines show the different r. In this first example each data set consists of a multitude of triples with $x_{n-1}$, $x_n$ and $r$.


Figure 4. Training Data.

In a first step of our experiments, we tried to learn r with given $x_{n-1}$ and $x_n$.

The algorithm started with the following operation set: addition, subtraction, multiplication, division, sine, cosine, exponential, logarithm functions. The terminal values consisted of the function's input variables and real numbers. As expected the system was able to detect the correct formula:

$$r = \frac{x_n}{x_{n-1} - (x_{n-1})^2} \quad (5)$$

The next experiment was to find a formula which is able to predict $x_n$ without a given r. To solve the problem, it is not sufficient to make $x_{n-1}$ available to the learning algorithm. Therefore, we added $x_{n-2}$ (predecessor of $x_{n-1}$) to the data set und detected the formula which describes each point $x_n$ of the Feigenbaum-Diagram with only two given points $x_{n-1}$, $x_{n-2}$:

$$x_n = \frac{x_{n-1}^3 - x_{n-1}^2}{x_{n-2}^2 - x_{n-2}} \quad (6)$$

Remarkably, this formula is able to describe every $x_n$ with only two given data points $x_{n-1}$, $x_{n-2}$ and without given *r*.

**Functional root**

The final experiment for the logistic function experiment was to determine the functional root of the logistic function with given *r*. Unlike in the former experiments, our algorithm was not able to find an exact analytical solution to this problem. But, experiments with a separated validation data set showed that they are good approximation to this problem.

Again, our Symbolic Regression algorithm was searching for the solution with the operation set of addition, subtraction, multiplication, division, sine, cosine, exponential, logarithm. The terminal values consisted of the function's input variables and real numbers.

Two runs of the Symbolic Regression algorithm found the following solutions:

$$\begin{aligned} x_n &= f(f(x_{n-1}, r), r) = \\ &1.0578035 * \sin(x_{n-1}) * \sqrt{r} * \cos(3.7038517 + x_{n-1}) \end{aligned} \quad (7)$$

$$\begin{aligned} x_n &= f(f(x_{n-1}, r), r) = \\ &0.30775887 + 0.42397907 * \sqrt{r} * \\ &\cos(1.9426149 + 2.3924117 * x_{n-1}) \end{aligned} \quad (8)$$

## VI. CONCLUSIONS

In this paper, we address the task to find mathematical formulas to functional roots with Symbolic Regression. A practical real-world application is the interpolation of recursive repetitions of manufacturing tasks. This problem arises in many scientific fields but few existing tools can be used to find the functional root analytically or to analyze them. Our approach is applicable to arbitrary problems, and does not require deep mathematical insight into this research field. It is especially favorable for analyzing systems in which little expert knowledge is available.

In a first step of our project, we demonstrated the feasibility of this approach by two well-known problems. Based on the results from our Symbolic Regression analyses we found a solution for the logistic function which is able to predict the next time step with arbitrary and unknown *r* and only with two previous data measurements.

Our results show that Symbolic Regression is a suitable tool for modeling the dynamics of systems and to find functional roots for iterated processes of arbitrary behavior and dynamics.

## REFERENCES


[1] R. O. Duda, P. E. Hart, D. G. Stork, "Pattern Classification", 2nd ed., Wiley Interscience, 2000.

[2] I. Schwab, N. Link, "Reusable Knowledge from Symbolic Regression Classification", Genetic and Evolutionary Computing (ICGEC 2011), 2011.

[3] S. Choenni ,"Design and Implementation of a Genetic-Based Algorithm for Data Mining," in VLDB 2000, pp. 33-42, 2000.

[4] L. Kindermann, A. Lewandowski, and P. Protzel, "A framework for solving functional equations with neural networks," in Proc. Eighth Int'l Conf. on Neural Information

Processing - ICONIP'2001, Fudan University Press, Shanghai (2001), pp. 1075-1078.

[5] L. Kindermann, and P. Protzel, “Computing iterative roots with second order training methods,” Proc. of the International Joint Conference on Neural Networks (IJCNN'2001), Washington DC 2001, pp. 424-427.

[6] J. R. Koza, “Genetic Programming: On the Programming of Computers by Means of Natural Selection”. Cambridge, MA, USA: MIT Press, 1992.

[7] M. Kuczama, and B. Choczewski , R. Ger, “Iterative Functional Equations”. Cambridge University Press, Cambridge, 1990.

[8] M. Schmidt, and H. Lipson, “Solving Iterated Functions Using Genetic Programming”, Proc. of Genetic and Evolutionary Computation Conference (GECCO'09), Montreal, Canada.

[9] D. A. Freedman, “Statistical Models: Theory and Practice,” Cambridge University Press, 2005.

[10] M. O'Neill, C. Ryan, "Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language," Kluwer Academic Publishers, Dordrecht Netherlands, 2003.