

Article

Soft-Computing-Based Estimation of a Static Load for an Overhead Crane

Tom Kuszniir  and Jaroslaw Smoczek * 

Department of Manufacturing Systems, Faculty of Mechanical Engineering and Robotics,
AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Kraków, Poland; tkuszniir@agh.edu.pl
* Correspondence: smoczek@agh.edu.pl

Abstract: Payload weight detection plays an important role in condition monitoring and automation of cranes. Crane cells and scales are commonly used in industrial practice; however, when their installation to the hoisting equipment is not possible or costly, an alternative solution is to derive information about the load weight indirectly from other sensors. In this paper, a static payload weight is estimated based on the local strain of a crane's girder and the current position of the trolley. Soft-computing-based techniques are used to derive a nonlinear input–output relationship between the measured signals and the estimated payload mass. Data-driven identification is performed using a novel variant of genetic programming named grammar-guided genetic programming with sparse regression, multi-gene genetic programming, and subtractive fuzzy clustering method combined with the least squares algorithm on experimental data obtained from a laboratory overhead crane. A comparative analysis of the methods showed that multi-gene genetic programming and grammar-guided genetic programming with sparse regression performed similarly in terms of accuracy and both performed better than subtractive fuzzy clustering. The novel approach was able to find a more parsimonious model with its direct implantation having a lower execution time.

Keywords: overhead crane; payload estimation; identification; genetic programming; sparse regression; subtractive fuzzy clustering



Citation: Kuszniir, T.; Smoczek, J. Soft-Computing-Based Estimation of a Static Load for an Overhead Crane. *Sensors* **2023**, *23*, 5842. <https://doi.org/10.3390/s23135842>

Academic Editors: Yu-Chen Hu and Debnath Bhattacharyya

Received: 21 May 2023
Revised: 14 June 2023
Accepted: 20 June 2023
Published: 23 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Cranes are material-handling machines widely used in the industrial, construction, and logistic sectors. Monitoring the static and dynamic load of material handling systems is important to satisfy safety, reliability, performance, and cost requirements [1–4]. The weight of the payload is crucial input information in crane applications, such as control, condition monitoring, maintenance planning, prediction of failures, and remaining useful life of crane equipment. Crane cells and scales are commonly used in industrial practice and implemented in different forms in load monitoring systems, such as Load Moment Indicator (LMI) and Rated Capacity Indicator (RCI) [1,2]. For hydraulic cranes, the payload weight is usually determined based on the hydraulic pressures in the lift cylinders [5]. When the load transportation is performed by cranes using ropes, cables, or chains, it requires an additional installation, usually of a strain-gauge-based sensor, onto the hoisting equipment.

The development of controllers for the safe and efficient operation of cranes in terms of crane positioning and payload sway suppression is an active area of research. There have been several control strategies [6] such as partial feedback linearization [7], sliding mode control [8], and nonlinear model predictive control [9] that require information about the operating point of the crane such as payload mass. Payload weight detection also plays an important role in condition monitoring [1,2]. However, in industrial practice there are a variety of crane configurations, thus, when the crane scale installation to the hoisting equipment is not possible, difficult, and costly, deteriorates crane maneuverability, or when the complementary measurement devices for hybrid payload monitoring are required, an

alternative solution is to derive information about the load weight indirectly from other sensors. The problem of load estimation for material handling lifting systems in several papers is studied through analytical modeling of their static and/or dynamics. A review of model-based methods of payload estimations in hydraulic excavators and a discussion of their applicability is provided in [10]. The model-based static and dynamic load estimation for hydraulically actuated material handling machines is commonly performed based on information obtained from the hydraulic pressure (pressure difference across the boom) and angular position sensors, while the model parameters are estimated offline and/or online usually using least square [11] or recursive least square method [12]. A variety of models addressing the dynamics of a hoisted load are developed in the literature mostly using the Lagrangian approach, such as a dynamic model of the hoisting motion of a hydraulic lattice mobile crane [13] and an overhead crane hoist with a crane's structure dynamic model [14–16]. An overhead crane hoist with an asynchronous electric motor model is developed and analyzed in numerical simulations to determine the relationship between the weight of a payload and the deviation in the amplitude of the electric motor current [17]. A digital-twin approach with a CAD model of a knuckle boom crane was proposed for real-time monitoring of a payload weight in [18]. A virtual strain gauge sensor was implemented with a nonlinear finite element model of a crane to find a linear relationship between the payload weight input and strain output signals, then the load weight is estimated through the inverse modeling approach applying the input signal from the physical strain gauge used in the experimental setup. The idea to estimate a payload weight using data-driven models identified using machine learning techniques based on data collected from available sensors is implemented in several recent papers. The ANN model trained using the Levenberg–Marquardt algorithm is developed in [19] for payload weight detection of the John Deere 644J hydraulic four-wheel-drive loader based on the differential pressure across the boom cylinder, boom and bucket strokes, and vibration acceleration of the frame. The ANN model is used to estimate a payload mass for a hydraulic forestry crane based on the hydraulic pressure of the inner boom cylinder and the grapple position [20,21]. The vision system and convolutional neural network are reported to recognize a wood log weight for a forestry crane [22]. The quay crane load identification method based on the signals from an accelerometer and strain gauge sensors installed on the crane's structure parts is reported in [23].

Strain gauge sensors are used in a variety of crane applications. Strain-gauge-based methods are applied to fatigue life prediction of existing crane runway girders and to evaluate the residual operating life of a bridge crane girder [24–26]. A combination of strain-gauge measurement and finite element analysis is used to identify crane wheel load for a quay ship-to-shore crane [27]. Strain sensing is used to identify the occurrence and intensity of the overhead crane skewing [28] and in the new technique of payload sway angle measurement [29].

In this paper, a static payload weight is estimated based on the local strain of a crane's girder and the current displacement of the trolley. The problem of the crane girder's deformation due to a moving trolley with a rope-suspended payload is studied in the literature mostly using analytical models developed based on the Euler-Bernoulli or Timoshenko beam deformation theories [30–34], or through applying finite elements methods [35–38]. The analytical models can be used to solve a nonlinear static inverse problem: knowing the deformation of the crane's girder and trolley displacement, determine the payload mass. However, obtaining an accurate mathematical model can be a challenging process and there are many factors, complexity characteristics, nonlinearities, parameters' uncertainty, etc., affecting the deviation between a mathematical model and the physical system. Thus, in this paper data-driven techniques are adapted to model-based estimation of a static payload weight. This work follows the previous paper [39], in which a payload weight is estimated for an overhead crane based on a local strain of a girder and the trolley position using a data-driven Takagi–Sugeno fuzzy (TSF) model identified through the subtractive clustering method and least square algorithm, thus, the nonlinear relation

between the payload weight, strain, and trolley's position is approximated by linear models expressed in fuzzy rule consequents. In comparison to the previous work, this paper addresses the issue of improving estimation accuracy and increasing interpretability using genetic-programming-based model structure optimization. Genetic programming is an evolutionary-based method whose outputs are computer programs. This makes genetic programming well suited to find a mathematical function that relates the input to the output, which is interpretable by the user and can be directly implemented reducing execution time. The reduction in execution time of the model is a significant advantage since the payload estimation is carried out online. Two variants of genetic programming, multi-gene genetic programming (MGGP) and grammar-guided genetic programming with sparse regression (G3PSR) are applied to obtain a static data-driven input–output model. The G3PSR algorithm uses sparse regression on a fixed set of evolved candidate model terms that are obtained from a biased search of the function space and the l_0 penalty is added for model selection, which results in reduced complexity and decreases the probability of overfitting. The subsequent sparse regression problem is solved by using the monotonically accelerated proximal gradient descent algorithm [40]. To the best of the authors' knowledge, the only other evolutionary sparse regression algorithm is based on an elastic net regularization [41]. The contribution of this paper can be summarized as follows:

- Develop a novel genetic programming variant called G3PSR that can be used for symbolic regression problems that can be expressed as a linear in the parameters model.
- Apply genetic programming variants, namely G3PSR and MGGP, to identify a mathematical relationship between the payload mass and the trolley position and girder strain.
- Compare the genetic programming models for mass estimation with a method proposed in the literature [39].

The static model identification process was based on the experimental data obtained from a laboratory overhead crane. The comparative analysis of G3PSR, MGGP, and TSF models is presented in terms of accuracy and complexity. To the best of our knowledge, this is the first work on the crane's payload weight estimation using a data-driven model identified using multi-gene genetic programming and grammar-guided genetic programming with sparse regression. The other genetic programming-based crane modeling approaches used the MGGP for payload sway prediction [42,43]. Crane dynamics modeling and control were also studied by using a genetic algorithm (GA) in [44–47].

The rest of the paper is organized as follows. In Section 2, the methods used to identify the models for payload weight estimation are described. Section 3 presents the experimental setup with a laboratory overhead crane, discusses the results of identification experiments, and compares the model performances, while the final conclusions are delivered in Section 4. The lists of symbols and abbreviations used in the paper with their definitions are exposed in Appendix A.

2. Methodology

Genetic programming is an evolutionary algorithm that is able to synthesize programs and was developed as an offshoot of genetic algorithms in the 1990s by Koza [48]. One of the important applications of genetic programming has been symbolic regression in which the aim is to identify a mathematical expression that best fits an input–output dataset. In traditional genetic programming, an individual is represented by an abstract syntax tree (AST) whose offspring, obtained by undergoing a variation operation, passes on to the next generation. Different variants of genetic programming have developed, such as multi-dimensional genetic programming, where an individual is composed of different subprograms such as in multi-gene genetic programming [49], linear genetic programming [50], grammatical evolution [51], Keizen programming [52], etc.

2.1. Multi-Gene Genetic Programming

Multi-gene genetic programming is a robust variant of genetic programming that was developed in [49]. In MGGP, each individual in the population is composed of a linear combination of functions called genes

$$\hat{y} = \sum_{i=1}^N \theta_i G_i \tag{1}$$

where \hat{y} is the model output, θ_i is the coefficient of the i -th, gene and G_i is the result of evaluating the function represented by the i -th gene. The expression in (1) allows MGGP to combine traditional genetic programming’s ability to evolve a sequence of functions together with the linear least squares algorithm to improve the performance in terms of speed of finding a solution to the symbolic regression problem as well as accuracy.

Selection in MGGP occurs in two stages, at first a parent is selected based on the individual’s fitness, then a gene within the parent is selected at random. Once the gene is selected, it undergoes one of four variation operators: subtree crossover, high-level crossover [53], subtree mutation, and point mutation as shown in Figures 1–4. GPTIPS2 [53] an MGGP toolbox for MATLAB is used to obtain the model.

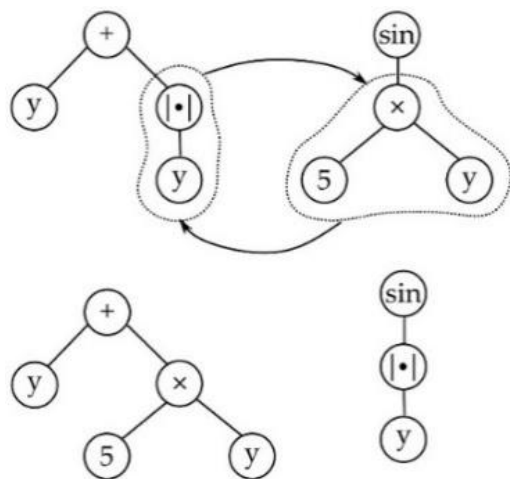


Figure 1. Subtree crossover.

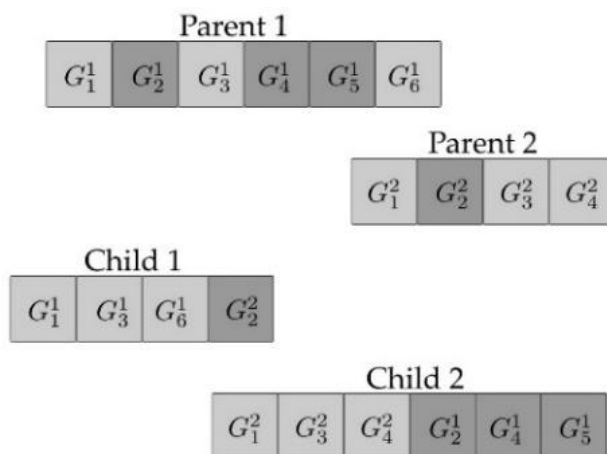


Figure 2. High level crossover.

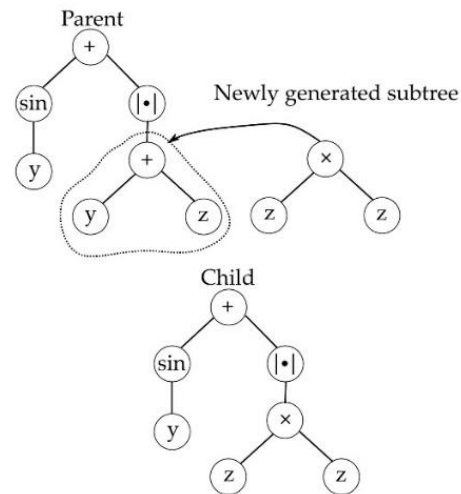


Figure 3. Subtree mutation.

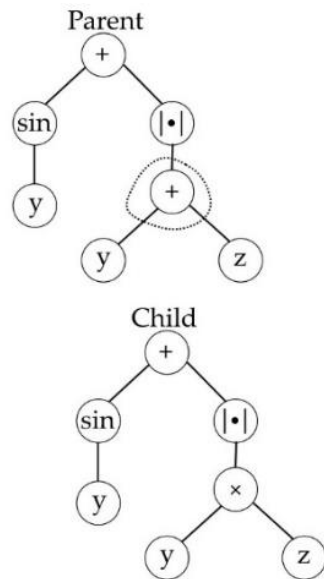


Figure 4. Point mutation.

The configuration of the multi-gene genetic programming hyperparameters used to obtain the mathematical expression of the payload mass, m , given the input data x (trolley position) and strain ϵ were obtained using trial and error based on previously published values [54], with the exception of the maximum number of genes, which was chosen to be 25 in order to have a fair comparison between MGGP and G3PSR, is given in Table 1.

Table 1. Configuration of the MGGP hyperparameters.

Parameters	Settings
Function set	\times, \sqrt, inv
Terminal set	x, ϵ
Population size	100
Number of generations	500
Initialization	Ramped Half-and-Half
Maximum number of genes	25
Maximum tree depth	5
Tournament size	2
Crossover probability	0.84
Mutation probability	0.14
Direct reproduction	0.02

2.2. Grammar-Guided Genetic Programming with Sparse Regression

Grammar-guided genetic programming with sparse regression is a novel algorithm in which an individual is composed of N_{max} functions, called model terms, and are the equivalent of genes in MGGP. The difference in representation between MGGP and G3PSR is that in MGGP an individual can have between 1— N_{max} genes, while in G3PSR an individual always has N_{max} candidate model terms. In order to select which model terms are included in the model an l_0 -penalty is added to the regression problem (2) which leads to a sparse solution vector

$$F(\theta) = \frac{1}{2} \|y - \phi\theta\|_2^2 + \lambda \|\theta\|_0 \quad (2)$$

where the $\|x\|_0$ counts the number of nonzero elements in a vector, i.e., $\|x\|_0 = \#\{i \text{ s.t. } x_i \neq 0\}$, y is the measured output data, ϕ is a matrix whose columns, j , are composed of evaluating the functions represented by the model terms and normalized by $\max|\phi_j|$, θ are the model term coefficients and λ is a constant that promotes the sparsification of the solution. The addition of the l_0 -penalty, which is nonconvex, makes solving (2) NP hard [55]. The solution to (2) is found using the monotone approximated gradient descent algorithm (mAPG) [40]. The pseudocode for mAPG is given in Algorithm 1, where the proximal operator for the l_0 -penalty is the hard thresholding operator (3). The algorithm runs until the maximum number of iterations has been reached or the objective function converges, i.e., $F(\theta_{k+1}) - F(\theta_k)$ is within a specified tolerance.

Algorithm 1: mAPG.

Input: ϕ, y, λ
Initialize: $\rho < 1, \delta, z_1 = \theta_1 = \theta_0, t_1 = 1, t_0 = 0, k = 1$
while not converged **do**
 $k \leftarrow k + 1$
 $w_k = \theta_k + \frac{t_{k-1}}{t_k}(z_k - \theta_k) + \frac{t_{k-1}-1}{t_k}(\theta_k - \theta_{k-1})$
 Initialize step size η_w and η_θ using Barzilai-Borwein method
 while $F(z_{k+1}) \geq F(w_k) - \delta \|z_{k+1} - w_k\|_2^2$ **do**
 $z_{k+1} = \text{prox}_{\eta_w \lambda}(z_k - \eta_w \phi^T(\phi z_k - y))$
 $\eta_w = \rho \eta_w$
 end while
 while $F(v_{k+1}) \geq F(\theta_k) - \delta \|v_{k+1} - \theta_k\|_2^2$ **do**
 $v_{k+1} = \text{prox}_{\eta_\theta \lambda}(v_k - \eta_\theta \phi^T(\phi v_k - y))$
 $\eta_\theta = \rho \eta_\theta$
 end while
 $t_{k+1} = \frac{\sqrt{4t_k^2 + 1} + 1}{2}$
 $\theta_{k+1} = \begin{cases} z_{k+1} & \text{if } F(z_{k+1}) \leq F(v_{k+1}) \\ v_{k+1} & \text{otherwise} \end{cases}$
end while
Output: θ

$$\text{prox}_{\gamma \|v\|_0} = \begin{cases} 0 & |v| \leq \sqrt{2\gamma v} \\ v & |v| > \sqrt{2\gamma v} \end{cases} \quad (3)$$

The variation operators used in G3PSR are similar to those used in MGGP with two main differences: the offspring should be admissible, as defined by the grammar, and in choosing the model terms that undergo crossover or mutation. Once an individual is selected from the population using tournament selection, the variation step is biased in the same manner as in [56] by setting the probability of choosing a model term from the individual proportional to its coefficient. The coefficients are first normalized using (4) and then the probability is calculated using (5), resulting in a higher chance of choosing model terms that have low or zero coefficients.

$$\bar{\theta}_i = \frac{\sum_{i=1}^q |\theta_i| - |\theta_i|}{\sum_{i=1}^q |\theta_i|} \quad (4)$$

$$P = \frac{\exp\bar{\theta}_i}{\sum_{i=1}^q \exp\bar{\theta}_i} \quad (5)$$

Grammars have been used in genetic programming to restrict the search space to increase interpretability and increase accuracy by incorporating prior knowledge about the problem. Several variants of grammar-based genetic programming, such as context-free grammar genetic programming [57] and grammatical evolution [51], have been applied successfully including the identification of nonparametric models of mechanical systems [58]. Context-free grammar is a four-tuple (N, Σ, P, S) used in computer science to generate syntactically correct sentences, where S is the start symbol, Σ is the set of terminal symbols, N is the set of all nonterminal symbols, and P is the set of production rules. The configuration of the G3PSR hyperparameters is given in Table 2, and the production rules are given in Table 3.

Table 2. Configuration of the G3PSR hyperparameters.

Parameters	Settings
Set of nonterminal symbols N	\times, \sqrt, inv
Set of terminal symbols Σ	x, ε
Population size	100
Number of generations	500
Initialization	Probability tree creation 2 (PTC2)
Number of candidate model terms	25
Maximum tree depth during initialization	8
Tournament size	2
Subtree crossover probability	0.75
High-level crossover probability	0.15
Mutation probability	0.1
Sparsification parameter λ	0.001

Table 3. Grammar used to obtain the G3PSR model.

$\langle S \rangle$::=	$\langle exp \rangle$
$\langle exp \rangle$::=	$\langle op_b \rangle \langle exp \rangle \langle exp \rangle \mid \langle op_u \rangle \langle exp \rangle \mid \langle T \rangle$
$\langle op_b \rangle$::=	\times
$\langle op_u \rangle$::=	$\sqrt \mid inv$
$\langle T \rangle$::=	$x \mid \varepsilon$

2.3. TS Fuzzy Model

TS fuzzy model [59,60] used in comparative study is defined as a set of R if-then rules with locally linear models in rule consequents:

$$\hat{m}_i = p_{i1}x + p_{i2}\varepsilon + p_{i3} \quad (6)$$

where $i = 1, 2, \dots, R$, p_{i1} , p_{i2} and p_{i3} are the rule-consequent parameters estimated using the least square method. The model output is calculated through interpolations of linear models between centroids of fuzzy clusters in the input space corresponding to rule antecedents:

$$\hat{m} = \frac{\sum_{i=1}^R w_i \hat{m}_i}{\sum_{i=1}^R w_i} \quad (7)$$

where w_i is a weight of i -th rule calculated as a product of Gaussian membership functions of input variables x (trolley position) and ε (strain)

$$w_i = \exp\left(-\frac{(x - x_i)^2}{2\sigma_{xi}^2}\right) \exp\left(-\frac{(\varepsilon - \varepsilon_i)^2}{2\sigma_{\varepsilon i}^2}\right) \quad (8)$$

where x_i , ε_i , σ_{x_i} , and σ_{ε_i} are the rule antecedent parameters corresponding to the expected values (x_i and ε_i coordinates of fuzzy clusters centroids in the input Cartesian space) and their standard deviations (σ_{x_i} and σ_{ε_i}).

The TSF model identification was performed using a combination of the fuzzy subtractive clustering algorithm [61] and the least square method. The subtractive algorithm was applied to derive the R rule's antecedents from measurement data through an iterative procedure that starts with the calculation of the potential of each k -th data point to be a cluster center based on a squared distance to the other data points:

$$P(d_k) = \sum_{j=1}^n \exp\left(-\frac{\|d_k - d_j\|^2}{\left(\frac{r_a}{2}\right)^2}\right) \quad (9)$$

where r_a is a positive constant called cluster radius. The data point with the highest potential is chosen as the first cluster center. In each next i -th step, the potential of all remaining data points is reduced according to their distance to the cluster center c_{i-1} selected in the previous step:

$$P(d_k) = P(d_k) - P(c_{i-1}) \exp\left(-\frac{\|d_k - c_{i-1}\|^2}{\left(\frac{r_b}{2}\right)^2}\right) \quad (10)$$

where $r_b = br_a$, and b is a positive parameter called the squash factor. The termination condition is defined twofold [61]. The upper and lower thresholds of data points' potential are determined using ζ_1 and ζ_2 called the accept and reject ratio, respectively,

$$\zeta_1 P_1 < P_i < \zeta_2 P_1 \quad (11)$$

where P_1 is the potential of the data point chosen as the first cluster center. Thus, the cluster centroid is either accepted if $P_i > \zeta_1 P_1$, or rejected if $P_i < \zeta_2 P_1$, and the algorithm is terminated. If condition (11) is satisfied, the shortest distance d_{min} between d_k and all previously found centroids is verified using condition (12). The data point is accepted as the centroid if condition (12) is satisfied, otherwise, the data point with the next highest potential is tested, and the algorithm is terminated if all data points violate condition (12).

$$\frac{d_{min}}{r_a} + \frac{P_i}{P_1} \geq 1 \quad (12)$$

3. Results of Identification Experiments

The data used in the identification procedure were obtained from experiments carried out on a double girder overhead crane installed in a laboratory (Figure 5). The trolley is driven by an AC gear motor with a gear ratio of 15.5, operating at 1400 rpm and output power of 0.12 kW. The girders' length, trolley wheelbase, and traveling range are 2.4 m, 0.3 m, and 1.8 m, respectively. The trolley position x along the girders is measured using the incremental encoder installed on the trolley's wheel with a resolution of 200 pulses per revolution (ppr). The strain gauge supplied by the ADAM 3016 input module is applied to measure the strain at the middle point of the girder. The data from sensors are sampled at 10 Hz using a PC with 16GB RAM and Quad Core 4GHz Intel Core i7-6700K CPU running Windows 10 and MATLAB R2020.

A series of experiments with different payload masses within the range of 20 kg and 100 kg were carried out to collect training and testing data for identification. It was assumed, that payload weight estimation is performed based on the signal from the strain gauge located at the midpoint of the girder and the trolley position ranging in $x = [0.4 \text{ m}, 1.4 \text{ m}]$ (with the strain gauge sensor at the midpoint of this interval). Figure 6 presents the experimental data obtained in experiments in which the trolley transported payload along the girder. The experimental data used to identify G3PSR and MGGP models were partitioned into three sets, the training set, validation set, and testing set. The training set is used to minimize (2) for G3PSR and (1) for MGGP, and the root mean square error

(RMSE) (13) on the training set is used as the fitness function in G3PSR and MGGP. The validation set is used for model selection, i.e., the model with the lowest RMSE on the validation set is stored as the best model, while the testing set is used to evaluate the performance of the obtained model. The testing set comprises data from experiments with payload masses of 30, 50, 70, and 90 kg (Figure 6), while the rest of the experimental data (payload mass: 20, 40, 60, 80, and 100 kg) are combined and split randomly with a 70:30 ratio into the training and validation sets, respectively. The performances of the models were compared using relative error (RE), mean relative error (MRE), and RMSE (13), where m and \hat{m} are the real and model-estimated values of weight, respectively, and n is the number of sample data.

$$RE = \frac{|m - \hat{m}|}{m}, \quad MRE = \frac{1}{N} \sum_{i=1}^n \frac{|m - \hat{m}|}{m}, \quad RMSE = \sqrt{\frac{\sum_{i=1}^n (m_i - \hat{m}_i)^2}{n}} \quad (13)$$

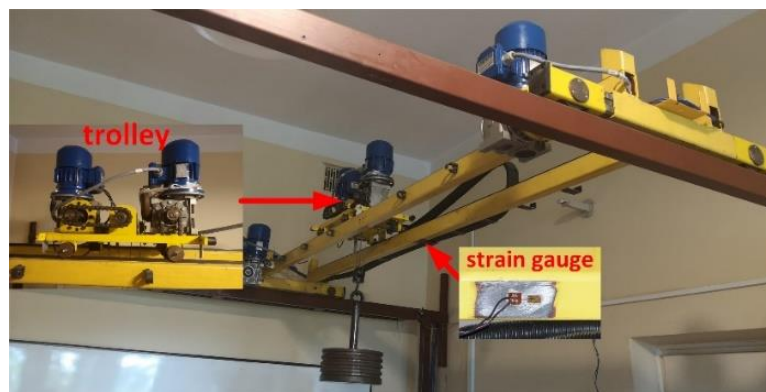


Figure 5. The laboratory stand.

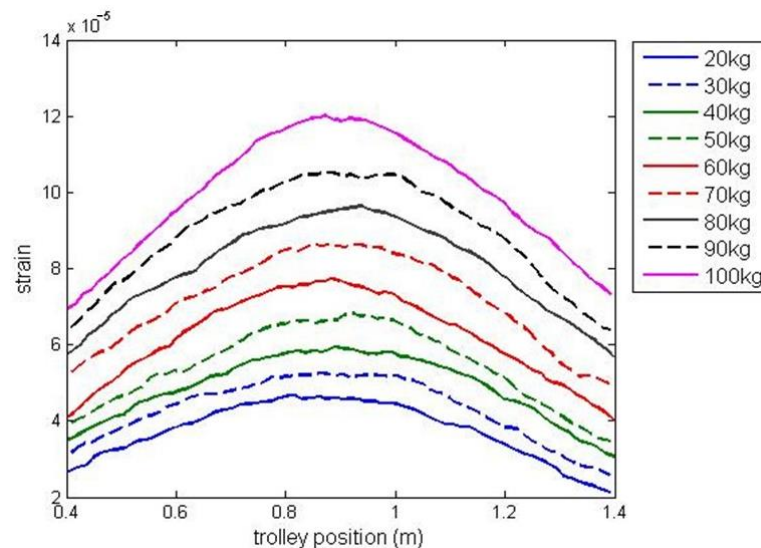


Figure 6. Experimental data partition into training and validation data (solid line), and testing data (dashed line).

The model terms and their corresponding coefficients for G3PSR and MGGP static models obtained in identification experiments are given in Table 4. The value of the fitness function (RMSE) for both the training and validation data sets are shown in Figures 7 and 8, for the G3PSR and MGGP models, respectively. Since both strategies use the performance on the validation set as the criteria for selection, the final models had RMSEs on the validation set of 1.2077 and 1.2565 by applying the G3PSR and MGGP techniques, respectively.

Table 4. The model terms and coefficients generated by G3PSR and MGGP.

G3PSR		MGGP	
Model Coefficients	Model Terms	Model Coefficients	Model Terms
-0.0058	$\frac{1}{x^3 \epsilon^6}$	-5.2140×10^5	1
4.5510×10^{22}	$\frac{1}{\sqrt{\epsilon}}$	879.6109	$x\epsilon^2$
15.6505	$\frac{1}{\sqrt[4]{\epsilon}}$	-1.2297×10^4	$x^4 \sqrt{\epsilon}$
1.1976×10^5	$x^3 \epsilon$	-5.0552×10^5	\sqrt{x}
1.6163×10^{-7}	$\frac{\sqrt[4]{\epsilon}}{\epsilon^2 x \sqrt{x}}$	8.3531×10^4	x
-161.6078	x	8.2332×10^5	$\frac{\epsilon}{x}$
4.4107×10^{-4}	$\frac{x\epsilon}{\sqrt{x}}$	2.0059×10^4	$\frac{\sqrt{\epsilon}}{x}$
18.2220	x^2	1.5669×10^4	$\sqrt{x} \sqrt[4]{\epsilon}$
-3.2326×10^{13}	$\epsilon^3 x$	-1.3150×10^5	$x \sqrt{\epsilon}$
2.3790×10^{-8}	$\frac{x}{\epsilon^2}$	3.3940×10^5	$\sqrt{x} \sqrt{\epsilon}$
14.1349	$\frac{x^4}{\sqrt[3]{\epsilon x}}$	-3.0590×10^5	$\sqrt{\epsilon}$
-3.2074	$x^8 \sqrt{x}$	1.2318×10^5	$x^2 \sqrt{\epsilon} \sqrt[4]{\epsilon}$
9.7761×10^7	$\sqrt{\epsilon^3}$	9.8117×10^3	$\frac{\sqrt[4]{\epsilon}}{\sqrt{x}}$
-5.7385×10^{-9}	$\frac{1}{\epsilon^2 x^8 \sqrt{x^5}}$	1.1880×10^5	$\frac{1}{\sqrt[4]{x}}$
		8.2782×10^5	$\sqrt[4]{x}$
		-3.7918×10^3	x^2
		-982.1465	$\frac{\sqrt[4]{\epsilon}}{x^2}$

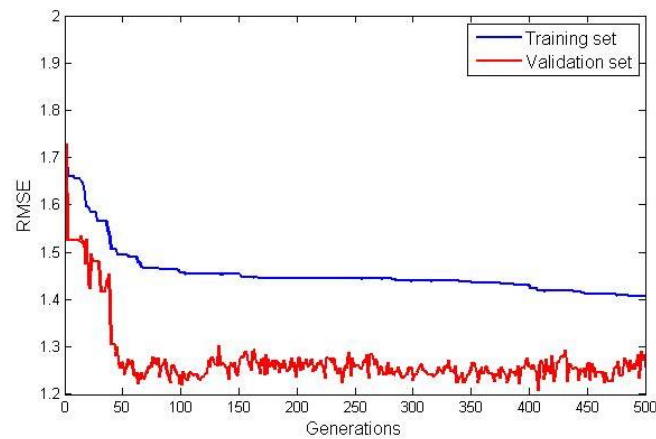


Figure 7. RMSE for training and validation data for the G3PSR.

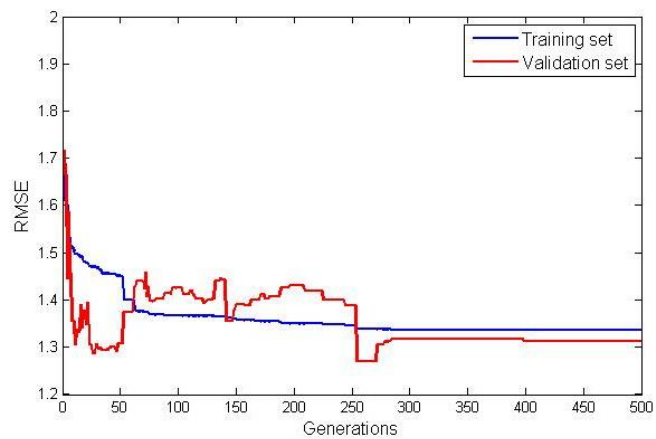


Figure 8. RMSE for training and validation data for the MGGP.

In the case of the TSF model identified using the combination of subtractive clustering and least square methods, the best model was obtained by setting by trial and error the cluster radius and squash factor to $r_a = 0.7$ and $b = 1.1$, respectively. For this setting, a clustering technique partitioned input space into eight clusters generating rules with premise and consequent parameters given in Table 5.

Table 5. TSF model parameters.

Rule Number	Antecedent (Gaussian) Parameters		Consequent (Linear Function) Parameters
i	$[\sigma_{xi}, x_i]$	$[\sigma_{ei}, \varepsilon_i] \times 10^{-5}$	$[p_{1i}, p_{2i}, p_{3i}]$
1	[0.250, 0.8652]	[2.447, 5.8133]	$[-60.65, 3.97 \times 10^3, 1.288 \times 10^2]$
2	[0.250, 0.8523]	[2.447, 9.4161]	$[-286.01, 5.605 \times 10^5, 286.19]$
3	[0.250, 1.2646]	[2.447, 5.2689]	$[32.11, -1.4716 \times 10^6, 228.17]$
4	[0.250, 0.4850]	[2.447, 5.0614]	$[5.81 \times 10^3, -2.467 \times 10^8, 1.824 \times 10^4]$
5	[0.250, 1.2048]	[2.447, 3.3429]	$[119.45, -1.365 \times 10^5, -270.93]$
6	[0.250, 1.2379]	[2.447, 9.1502]	$[2.16, 1.211 \times 10^6, 34.67]$
7	[0.250, 0.5311]	[2.447, 3.4812]	$[3.99 \times 10^3, -1.309 \times 10^8, -8.82 \times 10^3]$
8	[0.250, 0.4473]	[2.447, 7.4674]	$[-230.99, 3.9165 \times 10^7, 5.935 \times 10^3]$

The comparison of model performances for testing data in terms of accuracy and complexity is presented in Table 6. The G3PSR and MGGP models show similar performances for testing data: RMSE and MRE are 1.7813 and 0.0285 for G3PSR, while for MGGP they are 1.8069 and 0.0283, respectively. The TSF model performs slightly worse, as the RMSE and MRE are 1.8875 and 0.0294, respectively. In terms of the number of parameters to be estimated, the G3PSR and MGGP models have 14 and 17 parameters, respectively, while the number of parameters of the TSF model (56 parameters) is significantly greater. The G3PSR model notably outperforms the other models in terms of computational complexity, since the execution time (mean value from 10^3 runs plus/minus standard deviation) is $5.2 \times 10^{-3} \pm 5.8 \times 10^{-7}$ ms, while the MGGP and TSF models' execution times are $80.4 \times 10^{-3} \pm 5.5 \times 10^{-3}$ ms and $367.5 \times 10^{-3} \pm 7.6 \times 10^{-3}$ ms, respectively.

Table 6. Comparison of models complexity and performances for testing data.

	G3PSR	MGGP	TSF
RMSE	1.7813	1.8069	1.8875
MRE	0.0285	0.0283	0.0294
No. of parameters	14	17	56
Mean execution time (ms) ± standard deviation	5.2×10^{-3} $\pm 5.8 \times 10^{-7}$	$80.4 \times 10^{-3} \pm 5.5 \times 10^{-3}$	$367.5 \times 10^{-3} \pm 7.6 \times 10^{-3}$

Payload estimated mass along the test trajectories is compared in Figure 9, while the relative errors between the real and model-estimated values of weight for G3PSR, MGGP, and TSF models estimating mass 30 kg, 50 kg, 70 kg, and 90 kg are presented in Figures 10–13, as well as in Table 7 where the models' performances are compared using the MRE and maximum value of RE. Obviously, the accuracy in weight estimation increases with the increase in the payload mass. The models' performances are similar for a payload mass of 90 kg. The maximum relative error is 0.0395, 0.0388, and 0.0402 for G3PSR, MGGP, and TSF models, respectively. The second obvious conclusion, which comes from Figures 9–13 is that dividing a girder length into more sections with strain gauges at midpoints enhances estimation accuracy at the cost of more sensors and models to be used. It should be also noticed that the models were developed for the payload weight range of 20–100 kg, and we can expect that model complexity can rise with the increase in the range of the payload mass to be estimated. Thus, model optimization is necessary to save computational consumption.

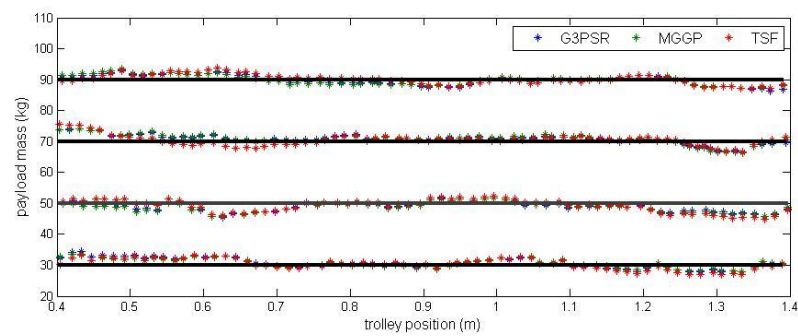


Figure 9. Payload estimated mass along the test trajectories.

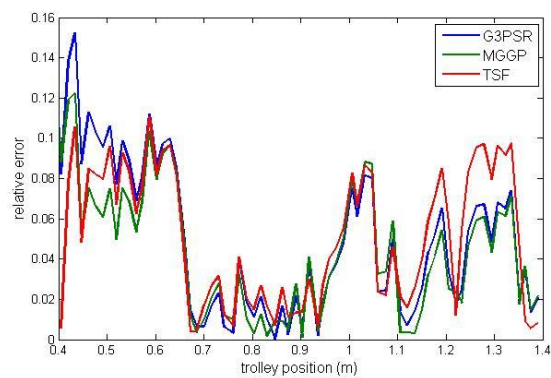


Figure 10. Relative error for estimated mass 30 kg.

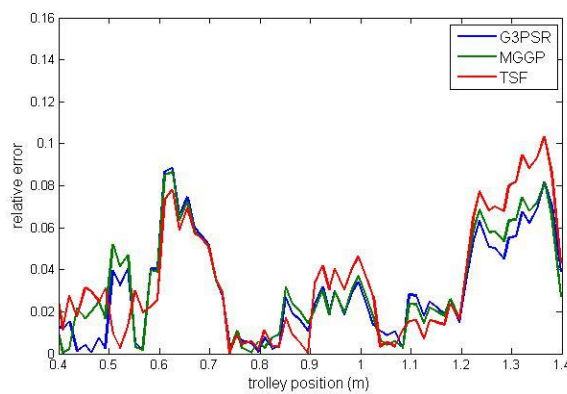


Figure 11. Relative error for estimated mass 50 kg.

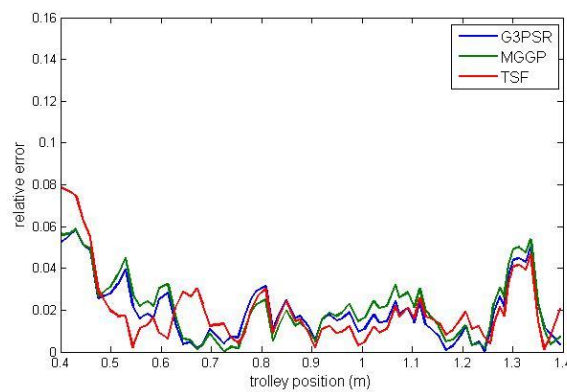


Figure 12. Relative error for estimated mass 70 kg.

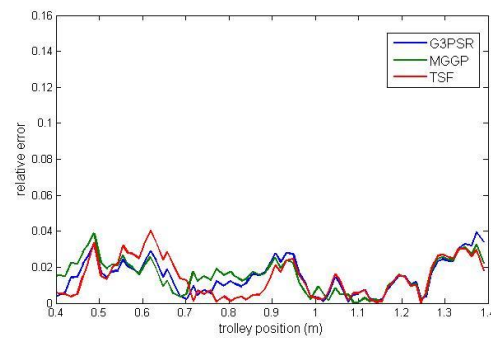


Figure 13. Relative error for estimated mass 90 kg.

Table 7. Models performances at testing operating points.

Payload Mass (kg)	G3PSR		MGGP		TSF	
	MRE	max RE	MRE	max RE	MRE	max RE
30	0.0502	0.1523	0.0449	0.1570	0.0499	0.1108
50	0.0302	0.0883	0.0320	0.0862	0.0344	0.1033
70	0.0200	0.0585	0.0219	0.0606	0.0207	0.0802
90	0.0148	0.0395	0.0156	0.0388	0.0142	0.0402

Uncertainty analysis is performed taking into account the effect of the strain signal variation. Another set of testing data obtained from eight experiments carried out on the laboratory stand was used to determine the standard deviation of strain signal with respect to the data used to identify the models assumed as the nominal data for comparison. The nominal strain input data used for models identification were then perturbed with a standard deviation of $\sigma = 0.8463 \times 10^{-6}$. The comparison of model performances at testing operating points for nominal input data (ϵ) and strain signal deviated by $\epsilon \pm \sigma$ is presented in Table 8 using RMSE and MRE. All models exhibit a similar increase in RMSE (2.2115, 2.2104, and 2.2889 for G3PSR, MGGP, and TSF models, respectively) when the nominal trajectory of strain signal is deviated by $\epsilon + \sigma$, while the G3PSR model is slightly less affected by strain signal perturbation $\epsilon - \sigma$, since the RMSE is 2.0835, 2.2169, and 2.2169 for G3PSR, MGGP, and TSF models, respectively. Taking into account the nominal and deviated strain signal, the uncertainty of the models' output (estimated payload weight) is expressed by standard error calculated according to (14), and the results are presented in Table 9 and Figure 14 (where the dotted line is a linear interpolation between testing points). The general tendency observed in Figure 14 and Table 9 is that the standard error decreases with the increase in the payload weight from 30 to 90 kg, from 2.2297 to 1.7460 for the G3PSR model, and from 2.0670 to 1.7786 and from 2.2066 to 1.7740 for MGGP and TSF models, with the exception of 50 kg.

$$s = \sqrt{\frac{\sum_{i=1}^N (m_i - \hat{m}_i)^2}{N - 1}} \quad (14)$$

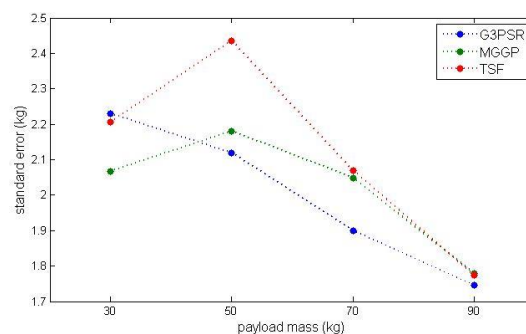


Figure 14. Standard error at testing points.

Table 8. Models performances at testing operating points for deviated strain input signal.

	G3PSR			MGGP			TSF		
	ϵ	$\epsilon + \sigma$	$\epsilon - \sigma$	ϵ	$\epsilon + \sigma$	$\epsilon - \sigma$	ϵ	$\epsilon + \sigma$	$\epsilon - \sigma$
RMSE	1.7813	2.2115	2.0835	1.8069	2.2104	2.2169	1.8875	2.2889	2.2169
Payload mass (kg)	MRE								
	ϵ	$\epsilon + \sigma$	$\epsilon - \sigma$	ϵ	$\epsilon + \sigma$	$\epsilon - \sigma$	ϵ	$\epsilon + \sigma$	$\epsilon - \sigma$
30	0.0502	0.0793	0.0504	0.0449	0.0748	0.0469	0.0499	0.0771	0.0532
50	0.0302	0.0255	0.0487	0.0320	0.0246	0.0518	0.0344	0.0348	0.0474
70	0.0200	0.0296	0.0155	0.0219	0.0318	0.0172	0.0207	0.260	0.0196
90	0.0148	0.0161	0.0169	0.0156	0.0172	0.0166	0.0142	0.0177	0.0153

Table 9. Standard error at testing operating points.

Payload Mass (kg)	G3PSR	MGGP	TSF
	Standard Error (kg)		
30	2.2297	2.0670	2.2066
50	2.1200	2.1813	2.4352
70	1.9005	2.0492	2.0698
90	1.7460	1.7786	1.7740

4. Conclusions

The paper compares the data-driven soft-computing models for payload weight estimation for an overhead crane. The payload mass is estimated based on a local strain of a crane's girder and the trolley position. Two genetic programming variants are used for model optimization, multi-gene genetic programming, a novel technique and grammar-guided genetic programming with sparse regression. The genetic programming methods are compared with the TSF model evolved using the fuzzy subtractive clustering and least square methods. The comparative study is presented in terms of accuracy and complexity. The G3PSR and MGGP models show similar estimation accuracy with an RMSE of 1.7813 and 1.8069 respectively, and slightly better than in the case of the TSF model with an RMSE of 1.8875. The relative error decreased for all models identified as the payload mass increased, this is due to the fact that the absolute error does not differ by a significant amount depending on the payload mass while the relative error is normalized by the payload mass, which, when increased, results in a decrease in the relative error. The model complexity expressed as the number of model parameters to be estimated is lowest for the G3PSR model (14 parameters) in comparison to the MGG and TSF models at 17 and 56 parameters, respectively. The significant difference we observe in the model execution time is that the G3PSR outperforms the other models with a mean execution time of 5.2×10^{-3} ms, while the MGGP model and the TSF model had mean execution times of 80.4×10^{-3} ms and 367.5×10^{-3} ms, respectively. The direct implementation of the input-output function decreased the computational time by approximately 98.6% and 78.1% for the G3PSR and MGGP models, respectively, when compared to the TSF model. Future works will be focused on testing the proposed approach on a crane system with a larger girder length and payload weight range. The G3PSR algorithm would also be applied to the inverse modeling approach to estimate the strain/stress of a girder [18] for a given payload mass, as well as in other static and dynamic model optimization problems [62,63].

Author Contributions: T.K. and J.S. conceived, designed, and performed the experiments; analyzed the data; and wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: The work has been financially supported by the Polish Ministry of Education and Science.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Symbols and Definitions

This appendix consists of a list of all the symbols used in this paper along with their definitions (Table A1) in addition to abbreviations (Table A2).

Table A1. List of symbols and their definitions.

Variable	Definition
b, c, d	Squash factor, cluster center, distance between cluster centers
m	Suspended payload mass
p_{i1}, p_{i2}, p_{i3}	Rule consequent parameters
r_a	Cluster radius
w_i	Weights of i -th rule
x	Trolley position
N	Set of all nonterminal symbols
P	Set of production rules (also potential of chosen datapoint as cluster center and probability of selection)
S	Start symbol
ε	Strain
ζ_1, ζ_2	Accept ratio, reject ratio
η	Step size
θ	Model term coefficients
$\bar{\theta}$	Normalized coefficients
λ	Sparsification parameter
ϕ	Regressor matrix
Σ	Set of all terminal symbols

Table A2. List of abbreviations and their definitions.

Abbreviation	Definition
GP	Genetic programming
G3PSR	Grammar guided genetic programming with sparse regression
mAPG	Monotone accelerated proximal gradient descent
MGGP	Multi-gene genetic programming
PTC2	Probability tree creation 2
TSF	Takagi–Sugeno fuzzy

References

- Neitzel, R.L.; Seixas, N.S.; Ren, K.K. A review of crane safety in the construction industry. *Appl. Occup. Environ. Hyg.* **2001**, *16*, 1106–1117. [[CrossRef](#)] [[PubMed](#)]
- Fang, Y.; Cho, Y.K.; Chen, J. A framework for real-time pro-active safety assistance for mobile crane lifting operations. *Autom. Constr.* **2016**, *72*, 367–379. [[CrossRef](#)]
- Świder, J.; Szewerda, K.; Herbuś, K.; Jura, J. Testing the impact of braking algorithm parameters on acceleration and braking distance for a suspended monorail with regard to acceptable travel speed in hard coal mines. *Energies* **2021**, *14*, 7275. [[CrossRef](#)]
- Świder, J.; Szewerda, K.; Tokarczyk, J.; Plewa, F.; Grodzicka, A.; Kędzia, K. An overview of possibilities of increasing the permissible speed of underground suspended monorails for transporting people in the conditions of polish underground mining. *Energies* **2023**, *16*, 3703. [[CrossRef](#)]
- Kalairassan, G.; Boopathi, M.; Mohan, R.M. Analysis of load monitoring system in hydraulic mobile cranes. *IOP Conf. Ser. Mater. Sci. Eng.* **2017**, *263*, 062045. [[CrossRef](#)]
- Ramli, L.; Mohamed, Z.; Abdullahi, A.M.; Jaafar, H.I.; Lazim, I.M. Control strategies for crane systems: A comprehensive review. *Mech. Syst. Signal Process.* **2017**, *95*, 1–23. [[CrossRef](#)]
- Hamdy, M.; Shalaby, R.; Sallam, M. A hybrid partial feedback linearization and deadbeat control scheme for a nonlinear gantry crane. *J. Frankl. Inst.* **2018**, *355*, 6286–6299. [[CrossRef](#)]
- Guo, Q.; Chai, L.; Liu, H. Anti-swing sliding mode control of three-dimensional double pendulum overhead cranes based on extended state observer. *Nonlinear Dyn.* **2023**, *111*, 391–410. [[CrossRef](#)]
- Krupa, F.; Nemcik, J.; Ozana, S.; Slanina, Z. NMPC design and embedded application for overhead crane: Case study. *IFAC-PapersOnLine* **2022**, *55*, 356–361. [[CrossRef](#)]

10. Bennett, N.; Walawalkar, A.; Schindler, C. Payload estimation in excavators: Model-based evaluation of current payload estimation systems. In Proceedings of the 3rd Commercial Vehicle Technology Symposium (CTV 2014), University of Kaiserslautern, Kaiserslautern, Germany, 11–13 March 2014; pp. 333–346.
11. Ferlibas, M.; Ghabcheloo, R. Load weight estimation on an excavator in static and dynamic motions. In Proceedings of the 17th Scandinavian International Conference on Fluid Power (SICFP'21), Linköping, Sweden, 1–2 June 2021.
12. Renner, A.; Wind, H.; Sawodny, O. Online payload estimation for hydraulically actuated manipulators. *Mechatronics* **2020**, *66*, 102322. [[CrossRef](#)]
13. Sun, G.; Kleeberger, M.; Liu, J. Complete dynamic calculation of lattice mobile crane during hoisting motion. *Mech. Mach. Theory* **2005**, *40*, 447–466. [[CrossRef](#)]
14. Bogdevičius, M.; Vik, A. Investigation of the dynamics of an overhead crane lifting process in a vertical plane. *Transport* **2005**, *20*, 176–180. [[CrossRef](#)]
15. Gaška, D.; Margielewicz, J.; Haniszewski, T.; Matyja, T.; Konieczny, Ł.; Chróst, P. Numerical identification of the overhead traveling crane's dynamic factor caused by lifting the load off the ground. *J. Meas. Eng.* **2015**, *3*, 34–35.
16. Haniszewski, T. Modeling the dynamics of cargo lifting process by overhead crane for dynamic overload factor estimation. *J. Vibroengineering* **2017**, *19*, 75–86. [[CrossRef](#)]
17. Semykina, I.Y.; Kipervasser, M.V.; Gerasimuk, A.V. Study of drive currents for lifting bridge cranes of metallurgical enterprises or early diagnosis of load excess weight. *J. Min. Inst.* **2021**, *247*, 122–131. [[CrossRef](#)]
18. Moi, T.; Cibicik, A.; Rolvag, T. Digital twin based condition monitoring of a knuckle boom crane: An experimental study. *Eng. Fail. Anal.* **2020**, *112*, 104517. [[CrossRef](#)]
19. Hindman, J.J. Dynamic Payload Estimation in Four Wheel Drive Loaders. Ph.D. Dissertation, University of Saskatchewan, Saskatoon, SK, Canada, 2008.
20. Geiger, C.; Greff, D.; Geimer, M.; Starke, M.; Ziesak, M. The potential of a weight detection system for forwarders using an artificial neural network. In Proceedings of the 51st International Symposium on Forestry Mechanization: Improved Forest Mechanisation: Mobilizing Natural Resources and Preventing Wildfires, Madrid, Spain, 25–27 September 2018; pp. 157–164.
21. Starke, M.; Geiger, C. Field setup and assessment of a cloud-data based crane scale (CCS) considering weight and local green wood density-related volume references. *Croat. J. For. Eng.* **2022**, *43*, 29–45. [[CrossRef](#)]
22. Geiger, C.; Maier, N.; Kalinke, F.; Geimer, M. Assistance system for an automated log-quality and assortment estimation based on data-driven approaches using hydraulic signals of forestry machines. In Proceedings of the 12th International Fluid Power Conference (IFK 2020), Dresden, Germany, 12–14 October 2020; pp. 83–92.
23. Zhou, X.; Hu, X. Load identification method of crane based on vibration acceleration signal and stress signal. *Int. Core J. Eng.* **2022**, *8*, 741–747.
24. Pástor, M.; Lengvarsky, P.; Hagara, M.; Kulka, J. Experimental investigation of the fatigue life of a bridge crane girder using S-N method. *Appl. Sci.* **2022**, *12*, 10319. [[CrossRef](#)]
25. Frankovský, P.; Delyová, I.; Sivák, P.; Kurylo, P.; Pivarčiová, E.; Neumann, V. Experimental assessment of time-limited operation and rectification of a bridge crane. *Materials* **2020**, *13*, 2708. [[CrossRef](#)]
26. Rettenmeier, P.; Roos, E.; Weihe, S.; Schuler, X. Assessment of mixed mode crack propagation of crane runway girders subjected to cyclic loading. *Eng. Fract. Mech.* **2016**, *153*, 11–24. [[CrossRef](#)]
27. Starykov, M.; Kokoshko, E. Approach for measuring a ship to shore crane actual wheel load. *J. Marit.* **2022**, *1*, 25–28. [[CrossRef](#)]
28. Mantič, M.; Kul'ka, J.; Grega, R.; Virostko, M.; Kopas, M. Application of sensors for incorrect behavior identification in a transport system. *Sensors* **2023**, *23*, 1635. [[CrossRef](#)] [[PubMed](#)]
29. Grosinger, P.; Rybář, J.; Dunaj, Š.; Ďuriš, S.; Hučko, B. A new payload swing angle sensing device and its accuracy. *Sensors* **2021**, *21*, 6612. [[CrossRef](#)] [[PubMed](#)]
30. Michaltsos, G.T.; Sophianopoulos, D.; Kounadis, A.N. The effect of a moving mass and other parameters on the dynamic response of a simply supported beam. *J. Sound Vib.* **1996**, *191*, 357–362. [[CrossRef](#)]
31. Oguamanam, D.C.D.; Hansen, J.S.; Heppler, G.R. Dynamic response of an overhead crane system. *J. Sound Vib.* **1998**, *213*, 889–906. [[CrossRef](#)]
32. Dyniewicz, B.; Bajer, C.I. New feature of the solution of a Timoshenko beam carrying the moving mass particle. *Arch. Mech.* **2010**, *62*, 327–341.
33. Xin, Y.; Xu, G.; Su, N.; Dong, Q. Nonlinear vibration of ladle crane due to a moving trolley. *Math. Probl. Eng.* **2018**, *2018*, 5756180. [[CrossRef](#)]
34. Chen, Q.; Cheng, W.; Gao, L.; Du, R. Dynamic response of a gantry crane's beam subjected to a two-axle moving trolley. *Math. Probl. Eng.* **2020**, *2020*, 3096213. [[CrossRef](#)]
35. Wu, J.J. Transverse and longitudinal vibrations of a frame structure due to a moving trolley and the hoisted object using moving finite element. *Int. J. Mech. Sci.* **2008**, *50*, 613–625. [[CrossRef](#)]
36. Zrnica, N.D.; Hoffmann, K.; Bosnjak, S.M. Modelling of dynamic interaction between structure and trolley for mega container cranes. *Math. Comput. Model. Dyn. Syst.* **2009**, *15*, 295–311. [[CrossRef](#)]
37. Jaafar, H.I.; Mohamed, Z.; Jamian, J.J.; Abidin, A.F.Z.; Kassim, A.M.; Ghani, Z.A. Dynamic behavior of a nonlinear gantry crane system. *Proc. Technol.* **2013**, *11*, 419–425. [[CrossRef](#)]

38. Milana, G.; Banisoleiman, K.; González, A. An investigation into the moving load problem for the lifting boom of a ship unloader. *Eng. Struct.* **2021**, *234*, 111899. [[CrossRef](#)]
39. Smoczek, J.; Hyla, P.; Kuszniir, T. Machine learning based approach to a crane load estimation. *J. KONBiN* **2021**, *51*, 1–10. [[CrossRef](#)]
40. Huan, L.; Zhouchen, L. Accelerated proximal gradient methods for nonconvex programming. In Proceedings of the NIPS15 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 379–387.
41. Goharoodi, S.K.; Dekemele, K.; Loccufier, M.; Dupre, L.; Crevecoeur, G. Evolutionary-based sparse regression for the experimental identification of Duffing oscillator. *Math. Probl. Eng.* **2020**, *2020*, 7286575.
42. Kuszniir, T.; Smoczek, J. Genetic programming based identification of an overhead crane. *J. KONBiN* **2021**, *51*, 123–133. [[CrossRef](#)]
43. Kuszniir, T.; Smoczek, J. Multi-gene genetic programming-based identification of a dynamic prediction model of an overhead traveling crane. *Sensors* **2022**, *22*, 339. [[CrossRef](#)] [[PubMed](#)]
44. Kawada, K.; Yamamoto, T. Evolutionary identification using closed-loop data for a mechanical system. In Proceedings of the SICE Annual Conference, IEEE, Taipei, Taiwan, 18–21 August 2010; pp. 1065–1070.
45. Duong, S.C.; Uezato, E.; Kinjo, H.; Yamamoto, T. A hybrid evolutionary algorithm for recurrent neural network control of a three-dimensional tower crane. *Autom. Constr.* **2012**, *23*, 55–63. [[CrossRef](#)]
46. Smoczek, J.; Szpytko, J. Evolutionary algorithm-based design of a fuzzy TBF predictive model and TSK fuzzy anti-sway crane control system. *Eng. Appl. Artif. Intell.* **2014**, *28*, 190–200. [[CrossRef](#)]
47. Çakan, A.; Önen, U. Anti-swing control of an overhead crane by using genetic algorithm based LQR. *Int. J. Eng. Comput. Sci.* **2017**, *6*, 21612–21616.
48. Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press: Cambridge, MA, USA, 1992.
49. Willis, M.; Hiden, H.; Hinchliffe, M.; McKay, B.; Barton, G.W. Systems modelling using genetic programming. *Comput. Chem. Eng.* **1997**, *21*, 1161–1166. [[CrossRef](#)]
50. Banzhaf, W. Genetic programming for pedestrians. In Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA'93), San Mateo, CA, USA, 17–22 July 1993; Forrest, S., Ed.; Morgan Kaufmann: San Francisco, CA, USA, 1993.
51. Ryan, C.; Collins, J.; Neill, M.O. Grammatical evolution: Evolving programs for an arbitrary language. In *Genetic Programming, Proceedings of the First European Workshop, EuroGP'98, Paris, France, 14–15 April 1998*; Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C., Eds.; Springer: Berlin/Heidelberg, Germany, 1998; pp. 83–96.
52. De Melo, V.V. Kaizen programming. In Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, ACM, Vancouver, BC, Canada, 12–16 July 2014; pp. 895–902.
53. Searson, D.P. Chapter 22—GPTIPS 2: An open-source software platform for symbolic data mining. In *Handbook of Genetic Programming Applications*; Gandomi, A.H., Alavi, A.H., Ryan, C., Eds.; Springer: New York, NY, USA, 2015.
54. Riahi-Madvar, H.; Dehghani, M.; Seifi, A.; Singh, V.P. Pareto optimal multigene genetic programming for prediction of longitudinal dispersion coefficient. *Water Resour. Manag.* **2019**, *33*, 905–921. [[CrossRef](#)]
55. Natarajan, B.K. Sparse approximate solutions to linear systems. *SIAM J. Comput.* **1995**, *24*, 227–234. [[CrossRef](#)]
56. La Cava, W.; Moore, J.H. Learning feature spaces for regression with genetic programming. *Genet. Program. Evolvable Mach.* **2020**, *21*, 433–467. [[CrossRef](#)]
57. Whigham, P.A. Grammatically-based genetic programming. In Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications, Tahoe City, CA, USA, 9 July 1995; Rosca, J., Ed.; University of Rochester: Rochester, NY, USA, 1995; pp. 33–41.
58. Ratle, A.; Sebag, M. Grammar-guided genetic programming and dimensional consistency: Application to non-parametric identification in mechanics. *Appl. Soft Comput.* **2001**, *1*, 105–118. [[CrossRef](#)]
59. Takagi, T.; Sugeno, M. Fuzzy identification of systems and its application to modeling and control. *IEEE Trans. Syst. Man Cybern.* **1985**, *15*, 116–132. [[CrossRef](#)]
60. Sugeno, M.; Kang, T. Structure identification of fuzzy model. *Fuzzy Sets Syst.* **1988**, *28*, 15–33. [[CrossRef](#)]
61. Chiu, S.L. Fuzzy model identification based in cluster estimation. *J. Intell. Fuzzy Syst.* **1994**, *2*, 267–278. [[CrossRef](#)]
62. Efkolidis, N.; Markopoulos, A.; Karkalos, N.; Hernández, C.G.; Talón, J.L.H.; Kyratsis, P. Optimizing models for sustainable drilling operations using genetic algorithm for the optimum ANN. *Appl. Artif. Intell.* **2019**, *33*, 881–901. [[CrossRef](#)]
63. Karkalos, N.E.; Efkolidis, N.; Kyratsis, P.; Markopoulos, A.P. A comparative study between regression and neural networks for modeling Al6082-T6 alloy drilling. *Machines* **2019**, *7*, 13. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.