Advances in Science and Technology Research Journal

Advances in Science and Technology Research Journal 2024, 18(4), 282–295
https://doi.org/10.12913/22998624/189497
ISSN 2299-8624, License CC-BY 4.0

Received: 2024.04.29
Accepted: 2024.06.05
Published: 2024.06.20

Evolutionary and Sparse Regression Approach for Data-Driven Modelling of an Overhead Crane Dynamics

Tom Kusznir[1], Jarosław Smoczek[1*], Bolesław Karwat[1]

[1] Faculty of Mechanical Engineering and Robotics, AGH University of Krakow, ul. Mickiewicza 30, 30-059 Kraków, Poland

* Corresponding author's e-mail: smoczek@agh.edu.pl

ABSTRACT

Identification of an accurate and simple model of a complex underactuated crane dynamics for varying operational conditions is a crucial step towards designing and implementation of real-time monitoring and control systems to enhance crane safety and operational efficiency. This paper considers a non-parametric data-driven identification of an overhead crane dynamics using symbolic regression techniques to find compromise between model complexity and predicted output accuracy. A grammar-guided genetic programming (G3P) combined with l_0 sparse regression is applied with two different variants of grammar to automatically construct a nonlinear autoregressive exogenous (NARX) model of different forms, termed extended and polynomial models. The proposed method is compared with a linear parameter-varying ARX (LPV-ARX) model. Identification is performed on experimental data obtained from a laboratory-scale overhead crane. The identified models are compared in terms of prediction accuracy, model's complexity measured using number of model terms, and execution time. The regularized G3P method outperformed the LPV-ARX model in terms of model predictive output accuracy. The G3P with the extended grammar resulted in more accurate crane velocity prediction models than the models with the polynomial grammar. The payload sway prediction model with the polynomial grammar was less complex in all measured metrics while there was no statistical significance in the accuracy when compared to the models with extended grammar.

Keywords: overhead crane, nonlinear dynamics, data-driven modeling, grammar-guided genetic programming, sparse regression.

INTRODUCTION

Material handling equipment, such as overhead cranes, are an indispensable and important part of the manufacturing and logistical processes, which require the transportation of heavy and oversized loads. An overhead crane system is a typical underactuated mechanical system, where the horizontal motion of the cable-suspended payload results in the transient and residual swing of a payload suspended on a wire rope that adversely affects the load positioning performances and may present a safety hazard for the surrounding environment and people. Many research works have addressed this problem by various anti-sway crane control approaches. However, to obtain effective crane control system, specifically model-based control system, an accurate crane dynamics model is required. An accurate prediction of crane's position, velocity, and sway, as well as computational efficiency of a model are crucial for model predictive controller performance and in real-time monitoring applications developed for automated crane systems to enhance safety and workflow efficiency. The risk of colliding with obstacles can be reduced by restricting crane motion, taking into account predicted payload position and residual sway, as well as a safety margin between the payload and an obstacle.

The state of the art of modelling and control approaches to underactuated crane systems is comprehensively discussed in recent papers [1, 2] that are focusing mostly on the physics-based modelling methods. Analytical models, for example derived from the Euler-Lagrange equations, are powerful but nevertheless can be lacking as there are many nonlinear and unmodelled effects that result in deviation of the physical system from the idealized mathematical model. Moreover, obtaining a reliable mathematical model of a complex underactuated nonlinear crane system and determining the correct parameters is a time consuming process. Therefore, a data-driven modelling approach is deemed as the more appropriate for model-based control design purposes.

A data-driven machine learning (ML) modelling of crane systems is mostly reported in the literature by using black-box modelling approaches. Artificial neural network (ANN) based models are developed for ship-mounted crane [3], a rubber-tired gantry crane [4], quay crane [5], and overhead crane [6]. A nonlinear autoregressive neural network with exogenous inputs (NARX-NN) is adapted in [7] to identify a data-driven model of an overhead crane based on input-output data examples collected on a small-scale laboratory crane driven by DC motors. The NARX-NN model with 30 neurons in single hidden layer is trained offline and online using extreme learning machines (ELM) method. After converting the data-driven model to a state-space form, an adaptive predictive anti-swing control law is developed using the empirical model. A radial basis function network is used to approximate dynamics of a shipboard container crane [8]. Linear parameter varying model identification for crane systems is also a popular method that has been often addressed by using Takagi-Sugeno fuzzy interpolation for switching linear models identified at local operating points [9, 10]. In [11], the adaptive neuro-fuzzy inference model is evolved using RNA genetic algorithm (GA) with hairpin genetic operators for the overhead crane dynamics identification.

However, the studies mostly present identification approach in which a specific black-box model structure is assumed (e.g. neural network, fuzzy or neuro-fuzzy models with fixed number of layers, nodes or rules) and parameters or weights are adjusted to reduce the model error. To address the issue of model structure selection some research works propose to use a flat output identification [12], Bayesian optimization method [13] or the Koopman operator theory merged with deep learning and regularization [14] for data-driven modelling of an underactuated crane dynamics. In [15], the multi gene genetic programming (MGGP) technique is used to establish a dynamical model of an overhead crane based on the experimental data measured on a laboratory stand. A hybrid least square and Levenberg-Marquardt algorithm is applied for parameter estimation in evolutionary evolved non-polynomial model. However, both evolutionary process of searching for the best model structure and parameters estimation are guided by the model performance criteria without taking into account the complexity of the model.

In this paper a regularized symbolic regression approach is proposed to evolve a crane dynamical model from experimental data and address a problem of structure selection and model dimensionality. The genetic programming is an effective symbolic regression method to automatically generate a system dynamics model based on predefined sets of input features and mathematical functions (terminal and nonterminal symbols) combined to obtain a grey-box model more interpretable than a black-box model. However, symbolic regression can be prone to generate complex expressions and overfitting when striving to decrease prediction error. This problem can be mitigated by providing a grammar to constrain the searching space and incorporate prior knowledge regarding the expected form and dimension of a target model. Another method of reducing a complexity of GP-based evolved model is to use regularization techniques [16]. Regularization is the popular method that is used in different forms in ML to overcame model complexity and prevent overfitting problem. The l_0 regularization provides better model sparsity, however l_0 norm is a nonconvex function that makes it difficult to optimize. The most popular regularization methods are based on easier to solve, convex l_1 and l_2 norms (convex relaxation of l_0 norm), e.g. Lasso and Ridge, or their hybrids introduced in elastic net [16]. In [17], the FFX deterministic algorithm has been stated as an alternative to GP, outperforming GP in symbolic regression problems. However, the recent works report that hybridization of these methods can be useful in reducing variance and complexity of a model [18, 19].

This paper follows the previous work [20], in which a G3P algorithm combined with sparse regression l_0, was compared with other ML techniques including conventional MGGP and ANN. This study is motivated by the need of further simplification of a crane data-driven dynamical model making it more feasible and computational efficient for implementation in real-time control and monitoring applications. This prerequisite is especially important when control platforms such as PLC (programmable logic controller) and PAC (programmable automation controller) are used. Thus, comparing to [20], this work studies the impact of a grammar on model's prediction performances and complexity. The different grammars are applied to obtain predictive NARX models with a polynomial basis and an extended basis with additional nonlinear functions. Moreover, the comparison is expanded by the LPV-ARX model where the sparse estimator is used to find the coefficient support and the least squares method used to re-estimate the nonzero coefficients. The models are derived from experimental data obtained from laboratory-scale overhead travelling crane for different working conditions (rope length and payload mass) and validated in terms of prediction performances (accuracy of model prediction output: crane velocity and payload sway), model's complexity (number of model terms), as well as computational efficiency (model's execution time).

IDENTIFICATION OF CRANE DYNAMICS

This section details the identification experimental setup of the laboratory-scale overhead crane, the G3P with sparse regression algorithm for the identification of the input-output discrete-time dynamic model of an underactuated crane and the linear parameter varying ARX model used for a comparative study.

The identification experiments were carried out on a laboratory scale overhead crane driven by AC motors supplied by frequency inverters and controlled by a Mitsubishi FX2N series PLC. Figure 1 presents hardware experimental setup. The crane bridge is driven by two 0.18 kW AC gear motors supplied by LGiC5 0.4 kW frequency inverters, that are controlled by the voltage signal u within the range ± 10 V. The crane's position x, sway angle a and rope length l are measured using the incremental encoders with resolutions of 400, 2000 and 100 ppr (pulses per rotation), respectively. The encoders are installed on the wheels to measure the position x relative to the crane bridge, under the trolley connected to a pair of fork arms embracing the hoisting cable to measure the sway angle a, and on the hoisting drum to measure the rope length l. The data from the sensors was sampled at 10 Hz using a PC (16 GB RAM, Quad Core 4 GHz Intel Core i7-6700 K CPU) equipped with IO card PCI1710HG, MS Windows 10 operating system and Matlab software release R2020. During experiments, the PLC was used to transmit the signals from sensors to PC and control signal u to actuators. An IIR low-pass differentiator [21] was used to obtain an estimate of the velocity from the position measurement while the sway signal was filtered using a low-pass FIR filter. The filtered signals were down sampled by keeping every second sample. Due to the large mechanical impedance in the drive system, the underactuated crane dynamics was assumed to be decomposed into actuated and unactuated part, that can be modelled by two discrete-time models. The first model, denoted as the velocity model (1), presents the discrete-time relation between crane velocity v and control signal u, while the second model, denoted as the sway model (2), presents the discrete-time relation between crane velocity v and payload sway α.

$$v_k = f_1(v_{k-1}, \dots, v_{k-nv}, u_{k-1}, \dots, u_{k-nu}) \quad (1)$$

$$\alpha_k = f_2(\alpha_{k-1}, \dots, \alpha_{k-n\alpha}, v_{k-1}, \dots, v_{k-nv}) \quad (2)$$

Identification framework

Under the modelling assumptions given by (1) and (2), the NARX model class [22] is selected in order to identify the input-output nonlinear crane dynamics. The NARX model chosen to express discrete-time relations (1) and (2), is a nonlinear generalization of the ARX model and has the general form

$$y_k = F(y_{k-1}, \dots, y_{k-ny}, u_{k-1}, \dots, u_{k-nu}) + e_k \quad (3)$$

where: y is the output, u is the input, e is Gaussian white noise and the subscript k is the sample instant. Additionally, when the nonlinear function F is linear in the parameters then the model is described by

$$y_k = \sum_{i=1}^{M} \theta_i \phi_i + e_k \quad (4)$$

where: ϕ_i is the i-th model term that is evolved by the G3P algorithm, and θ_i is the i-th model term coefficient. The linear in the parameters model structure coefficients can be estimated deterministically by the method of least squares

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \|\Phi\theta - y\|_2^2 \quad (5)$$

where: Φ is a matrix whose i-th column comprises of the evaluation of the model terms ϕ_i. In the proposed algorithm the number of model terms ϕ is fixed for each individual in the population and a sparse solution to (5) is found by solving the l_0 regularized problem

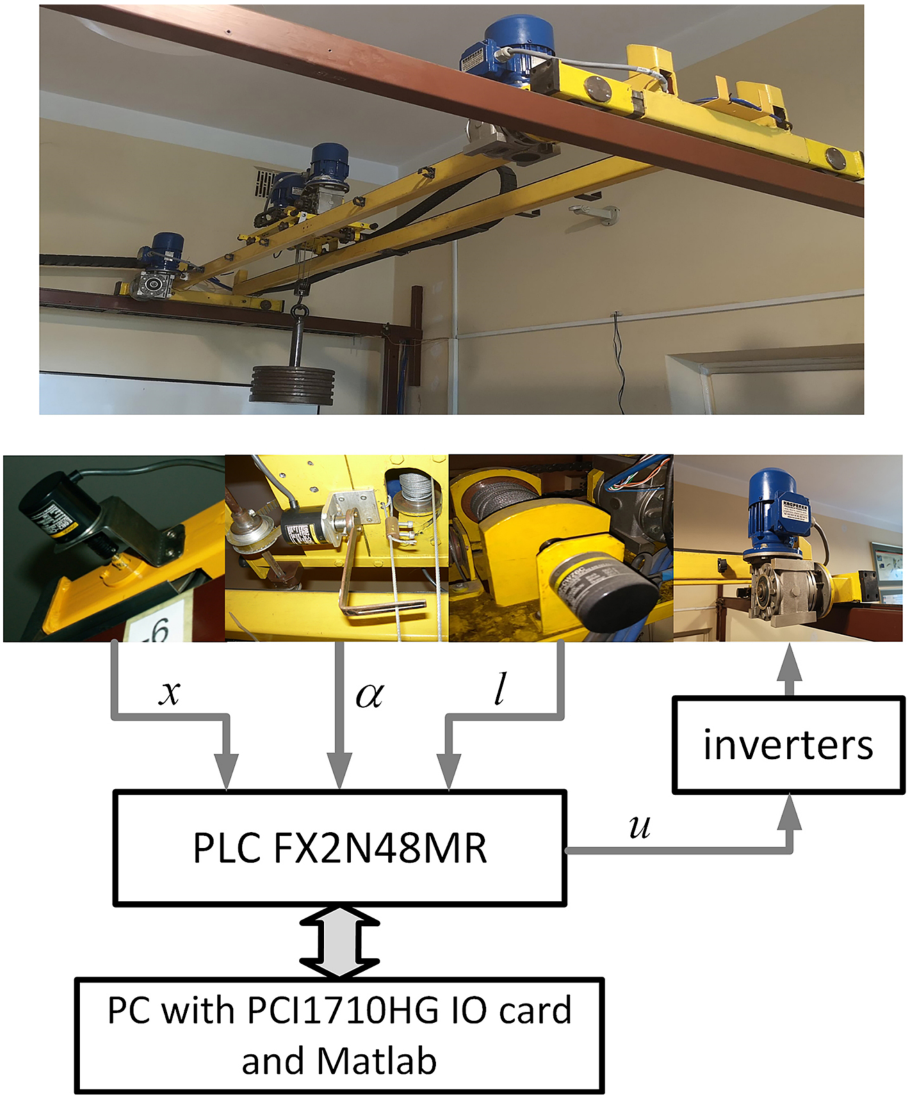

Figure 1. Schematic view of experimental setup

$$F(\theta) = g(\theta) + h(\theta) = \frac{1}{2}\|\Phi\theta - y\|_2^2 + \lambda\|\theta\|_0 (6)$$

where: $\|\theta\|_0$ is the l_0 pseudonorm, i.e. $\|\theta\|_0 = \sum_{i=1}^{N} \mathbb{1}_{\theta_i \neq 0}$. The problem (6) is nonconvex and has proven to be NP-hard [23] and therefore a suboptimal solution is found using a fast proximal gradient descent method [24] which converges to a critical point of $F(\theta)$. The proximal mapping of a function h is given by the proximal operator

$$prox_h(q) = \underset{v}{arg\,min}\; h(v) + \frac{1}{2}\|v - q\|_2^2 \quad (7)$$

Since the minimization problem of (7) is separable the *prox* operator is the element wise hard-thresholding operator with threshold τ:

$$[T_\tau(q)]_j = \begin{cases} 0, |q_j| \le \tau \\ q_j, \text{otherwise} \end{cases}, 1 \le j \le n \quad (8)$$

The fast monotone accelerated proximal gradient descent used in this paper, starts with the initial conditions $z^{(1)} = \theta^{(1)} = \theta^{(0)}$, $t_0 = 0$, and update $\theta^{(j+1)}$ according to

$$\theta^{(j+1)} = \begin{cases} z^{(j+1)}, F\left(z^{(j+1)}\right) \le F\left(\theta^{(j)}\right) \\ \theta^{(j)}, \text{otherwise} \end{cases} \quad (9)$$

where:

$$z^{(j+1)} = \text{prox}_{s\lambda\|\cdot\|_0}\left(w^{(j)} - s\nabla g\left(w^{(j)}\right)\right) =$$
$$= T_{\sqrt{2\lambda s}}\left(w^{(j)} - s\nabla g\left(w^{(j)}\right)\right)$$
$$w^{(j)} = P_{supp\left(z^{(j)}\right)}\left(r^{(j)}\right),$$
$$r^{(j)} = \theta^{(j)} + \frac{t_{j-1}}{t_j}\left(z^{(j)} - \theta^{(j)}\right) +$$
$$+ \frac{t_{j-1} - 1}{t_j}\left(\theta^{(j)} - \theta^{(j-1)}\right),$$
$$t_{j+1} = \frac{\sqrt{1 + 4t_j^2} + 1}{2},$$

and n is the number of elements, j is the iteration number, $P_{supp(z^{(j)})}(r^{(j)})$ is a support projection operator that returns a vector whose elements with indices in $z^{(j)}$ are the same as those in $r^{(j)}$, and s is the step size.

The algorithm converges when the step size $s \leq \min\left\{\frac{2\lambda}{G^2}, \frac{1}{L}\right\}$ where L is the Lipschitz constant of the gradient ∇g and is given by the largest singular value of Φ, and G is the bound on the gradient, i.e. $\|\nabla g(\theta)\|_\infty \leq G$. In order to find the appropriate hyperparameter λ the fast monotone accelerated proximal gradient descent algorithm is run with 10 different values of λ logarithmically spaced between $\lambda_{\min}$ and $\lambda_{\max}$:

$$\lambda_{min} = \frac{\lambda_{max}}{50000}, \lambda_{max} = = \frac{\max\left(\left(\theta^{(0)} - \eta\Phi^T\left(\Phi\theta^{(0)} - y\right)\right)^2\right)}{2\eta} \tag{10}$$

where: the step size $\eta = min\left\{\frac{1}{2|S|}, \frac{1}{L}\right\}$ and $|S|$ is the number of nonzero elements in $\theta^{(0)}$.

Grammar guided genetic programming

Traditional genetic programming relies on the closure property [25], i.e. that each non-terminal can accept any value from either the terminal set of as the return value of every other non-terminal, in order to produce admissible offsprings after undergoing a variation operator such as crossover or mutation. The closure requirement is restrictive as it does not allow to bias the search space other than selecting the terminal and non-terminal set; to overcome this, grammars, such as context-free grammars (CFG), can be used to bias the search space while making sure that the produced offspring are syntactically correct. A context-free grammar G is the four-tuple $G = \{S,N,\Sigma,P\}$ where S is the start symbol, N is the set of all non-terminal symbols, Σ is the set of all terminal symbols and N and Σ are disjoint, and P is the set of all production rules. All expressions have the start symbol as the root node, the non-terminal symbols are the internal nodes while the terminal symbols are the leaves of the tree.

In this study two different grammars were used to obtain the dynamic model of the overhead crane in terms of varying conditions: the rope length l and the mass of the payload m. The grammar used for representing individuals consists of a terminal set which includes lagged input and output variables, the rope length l and the mass of the payload m, which are assumed as the measurable parameters. The two grammars, presented in the Backus-Naur form (BNF) in Table 1, were tested to compare performances of the G3P algorithm used to construct the NARX model given in the linear in the parameters form in (4). The simpler grammar, which we call polynomial grammar, allows the model terms to be monomials in the lagged outputs and inputs, but also allows for the multiplication, protected division, and square root operators for the variables and . The second grammar, which we call the extended grammar, contains an extended function set and the production rules. The number of input and output delays for both extended and polynomial models were set to 5.

Variation operators

New individuals are produced through the variation operators which modify an existing individual in a stochastic manner. An individual needs to be selected before the variation operators are applied to it to produce offspring in the next generation. In the proposed method, selection occurs in two stages: in the first stage an individual is selected from the population using tournament selection, in the second stage a model term is selected from the individual. The probability of selecting a model term is given by

$$P = \frac{\exp\bar{\theta}_i}{\sum_{i=1}^{M}\exp\bar{\theta}_i} \tag{11}$$

where: $\bar{\theta}_i = \frac{\sum_{i=1}^{M}|\theta_i| - |\theta_i|}{\sum_{i=1}^{M}|\theta_i|}$ is the normalized coefficients of the model terms.

In the proposed method, three different variation operators are used, such as subtree crossover,

Table 1. The grammar used for the G3P extended and polynomial model

Grammar for extended model		Grammar for polynomial model	
$\langle E\rangle$	$::= \{\times\langle E\rangle\langle E\rangle \mid \sqrt{}\langle E\rangle \mid \mathit{tanh}\langle E_p\rangle \mid \mathit{pdiv}\langle E_p\rangle\langle E_p\rangle \mid \times\langle E\rangle\langle P\rangle \mid \langle T_{uy}\rangle\}$	$\langle S\rangle$	$::= \{\langle M\rangle\}$
$\langle E_p\rangle$	$::= \{+\langle E_p\rangle\langle E_p\rangle \mid +\langle E_p\rangle\langle \varepsilon\rangle \mid \times\langle E_p\rangle\langle E_p\rangle \mid \times\langle E_p\rangle\langle \varepsilon\rangle \mid \langle T_{uy}\rangle\}$	$\langle M\rangle$	$::= \{\times\langle M\rangle\langle M\rangle \mid \times\langle M\rangle\langle P\rangle \mid \langle T_{uy}\rangle\}$
$\langle P\rangle$	$::= \{\times\langle P\rangle\langle P\rangle \mid \mathit{pdiv}\langle P\rangle\langle P\rangle \mid \sqrt{}\langle P\rangle \mid \langle T_P\rangle\}$	$\langle P\rangle$	$::= \{\times\langle P\rangle\langle P\rangle \mid \mathit{pdiv}\langle P\rangle\langle P\rangle \mid \sqrt{}\langle P\rangle \mid \langle T_P\rangle\}$
$\langle T_{uy}\rangle$	$::= \{u_{k-1} \mid \ldots \mid u_{k-5} \mid y_{k-1} \mid \ldots \mid y_{k-5}\}$	$\langle T_{uy}\rangle$	$::= \{u_{k-1} \mid \ldots \mid u_{k-5} \mid y_{k-1} \mid \ldots \mid y_{k-5}\}$
$\langle T_P\rangle$	$::= \{m \mid l\}$	$\langle T_P\rangle$	$::= \{m \mid l\}$

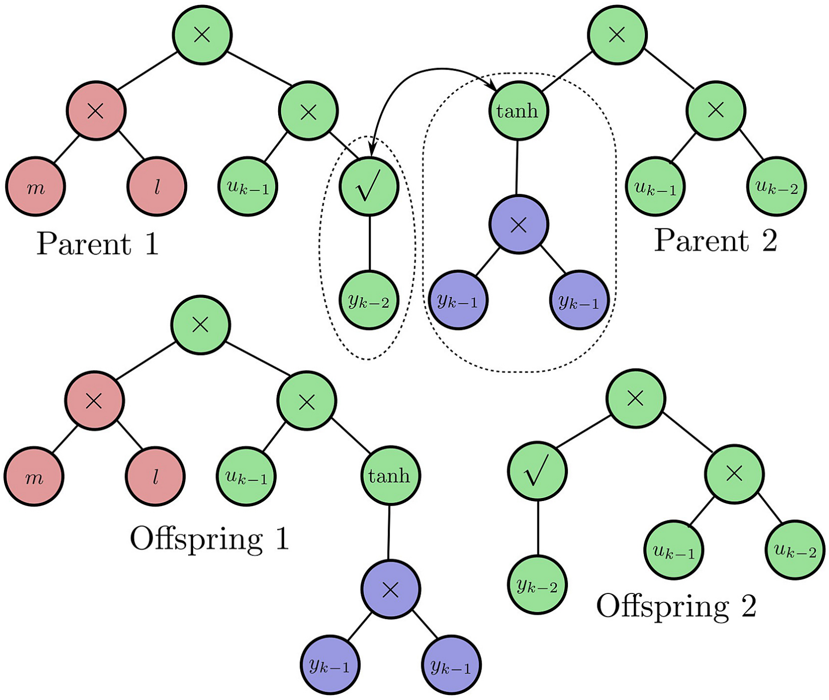

Figure 2. Illustration of subtree crossover

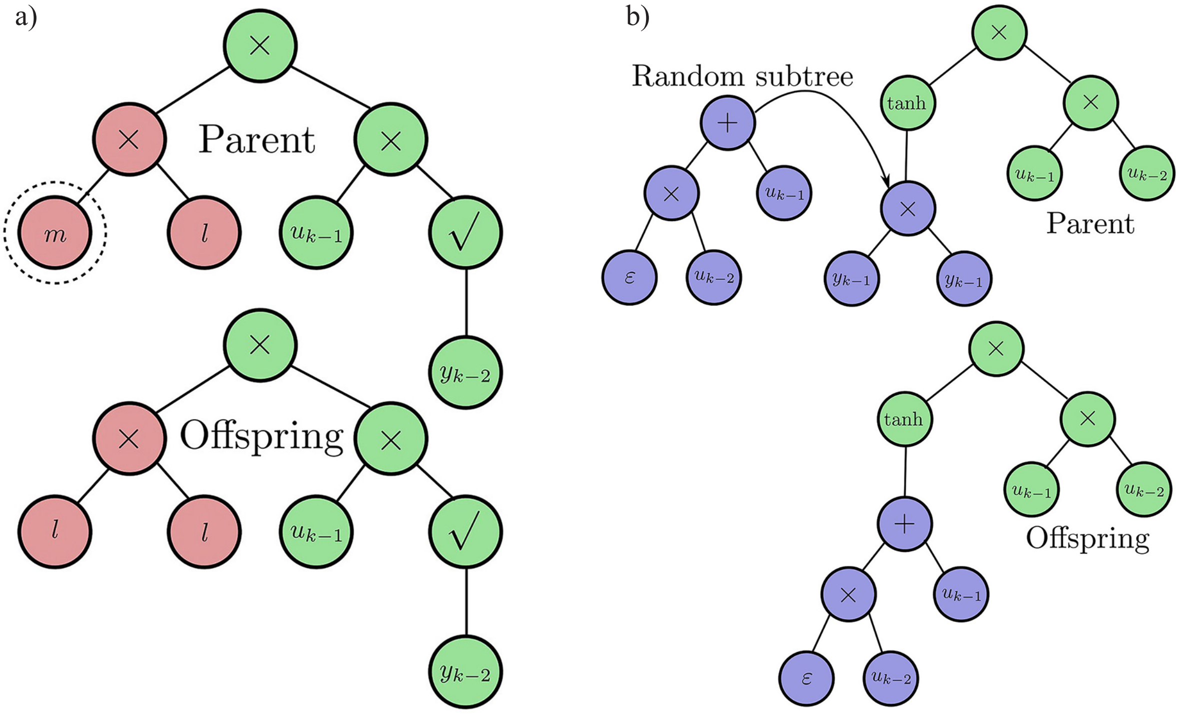

Figure 3. Illustration of mutation operators: a) point mutation, b) subtree mutation

subtree mutation and point mutation, which are illustrated in Figure 2 and 3. In subtree crossover, once two model terms are selected from two different individuals in the population, a random subtree is selected from one model term and a subtree is searched in the second model term so that once the subtrees are swapped the offspring are syntactically correct according to the grammar. In subtree mutation, once a model term is selected form an individual in the population, a random subtree in that model term is selected and replaced by a newly generated subtree in a manner so that the offspring is syntactically correct. In point mutation a terminal node is selected and is replaced by another terminal node.

Linear parameter varying models

Linear parameter varying input-output models are capable of modeling nonlinear systems by having the model parameters being a function of a measurable time-varying signal called the scheduling variable, while there exists only a linear relationship between the lagged inputs and outputs. Therefore, an LPV-ARX model, with the rope length *l* and payload mass *m* as the scheduling variables, was chosen to perform a comparative study with the G3P models. A single input single output (SISO) discrete-time linear parameter varying system can be written as

$$y_k + \sum_{i=1}^{n_a} a_i(p_k) y_{k-i} = \\ = \sum_{j=1}^{n_b} b_j(p_k) u_{k-j} + e_k \tag{12}$$

where: *p* is the scheduling variable and it is assumed that the functions $a_i(p_k)$ and $b_j(p_k)$ can be linearly parameterized as follows

$$a_i(p_k) = a_{i,0} + \sum_{q=1}^{n_q} a_{i,q}\psi_q \tag{13}$$

$$b_j(p_k) = b_{j,0} + \sum_{q=1}^{n_q} b_{j,q}\psi_q \tag{14}$$

where: ψ is a set of basis functions, which in the comparative study will be a set of monomials of the scheduling variables *l* and *m*, and $a_{i,0}, \dots, a_{i,q}$ and $b_{j,0}, \dots, b_{j,q}$ are unknown parameters. Under those assumptions, the model (12) can be rewritten as

$$y_k = \phi^T\theta + e_k \tag{15}$$

where:

$$\phi = \begin{bmatrix} -y_{k-1}, -y_{k-1}\psi, \dots, \\ -y_{k-n_a}\psi, u_{k-1}, u_{k-1}\psi, \dots, u_{k-n_b}\psi \end{bmatrix}^T,$$

$$\psi = \left[\psi_1, \dots, \psi_{n_q}\right],$$

$$\theta = \left[a_{1,0}, a_{1,1}, \dots, a_{n_a,n_q}, b_{0,0}, b_{0,1}, \dots, b_{n_b,n_q}\right]^T.$$

Including all possible model terms can lead to overparameterization and result in a more complex with reduced accuracy due to overfitting. Thus, the SPARSEVA method [26] is used to find an LPV-ARX model [27] with a reduced number of model terms by solving the following minimization problem

$$\min_{\theta} \|\theta\|_1 \\ s.t.: V_N(\theta) \leq V_N\left(\theta_N^{LS}\right)(1 + \varepsilon_N) \tag{16}$$

where: $V_N(\theta) = \frac{1}{N}\|\Phi\theta - y\|_2^2$, θ_N^{LS} is the unpenalized least squares solution of , and is a predefined quantity.

Following [26], the value of $\varepsilon_N = \frac{2n}{N}$, which corresponds to Akaike's information criterion, is used. Additionally, since the l_1 penalty biases the solution, (16) is used to find the support of and the non-zero values are re-estimated using least-squares, called SPARSEVA-RE. The solution of (16) was solved using the SPGL1 algorithm [28].

EXPERIMENTAL RESULTS

The experimental data was obtained in series of 10 experiments carried out on a laboratory-scale overhead crane (Fig. 1) for different operating points, rope lengths *l* = {0.8, 1.1, 1.4, 1.7, 2.0} m and payload masses *m* = {10, 30, 50} kg. The input signal *u* was a sequence of step functions with varying amplitude in order to excite the underactuated part of the overhead crane. The data from experiment with *l* = 1.1 m and *m* = 50 kg, and experiment with *l* = 1.7 m and *m* = 10 kg were used as the testing data, while the rest was used as the training data to evolve the G3P-NARX and LPV-ARX models.

In every generation of G3P evolution the training data is resampled with 60% used by the fast monotone accelerated proximal gradient descent algorithm to obtain the model coefficients while the remaining 40% is used to select the appropriate hyperparameter . The fitness of the individual is the root mean square error (RMSE) of the model predictive output (MPO) on the entire training data set:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} y_i^2}} \tag{17}$$

where: y and $\hat{y}$ are observed and predicted values, respectively.

The LPV model was run with the same number of input and output lags available to the G3P algorithm and the basis functions were chosen to be ψ = [*m*, *l*, *ml*, m^2, l^2, m^2 *l*, ml^2] which results in a maximum number of 80 parameters in the model. The G3P algorithm with hyperparameters given in Table 2 was run 20 times for each grammar (Table 1) to obtain both the velocity and sway NARX polynomial and extended models.The models are

Table 2. G3P hyperparameters

Number of generations	250
Maximum number of prox iterations	3500
Initialization	Probability tree creation 2 [29]
Maximum number of model terms	35
Population size	100
Maximum tree size	13
Maximum tree depth	8
Ephemeral random constant	[-1.1]
Tournament size	4

compared in terms of accuracy and complexity, with the accuracy evaluated in a one step ahead (OSA) prediction and for MPO using the RMSE, and the complexity being measured by the number of model terms and the total number of nodes. Additionally, the G3P-NARX velocity and sway models with the highest accuracy are combined and the MPO is evaluated 1000 times to obtain the median and median absolute deviation (MAD) value of the execution time. The medians and minimum values for the accuracy and complexity measures of the runs are given in Table 3 and Table 4 for the velocity and sway models respectively, while the execution time is given in Table 5. In order to determine if there exists a difference between the medians of the multiple runs using the polynomial and extended grammar the Wilcoxon rank sum test is conducted and the results given in Table 6. The boxplots of the G3P models OSA, MPO, and complexity for the velocity and sway models are shown in Figures 4–6, where G3P-E and G3P-P denote respectively extended and polynomial NARX models identified using the G3P algorithm.

The LPV-ARX velocity model had a MPO RMSE approximately 2.5 and 1.8 higher than the median G3P-NARX model on with polynomial grammar on the training and testing sets respectively, and approximately 2.9 and 2 times higher than the median G3P-NARX model with the extended grammar on the training and testing sets respectively. For the sway model LPV-ARX had a RMSE MPO of approximately 1.3 times higher than the median G3P-NARX model with polynomial grammar on both the training and validation sets while being

Table 3. Models performances for crane velocity prediction

Specification		OSA (RMSE)		MPO (RMSE)		Complexity	
		Training	Testing	Training	Testing	No. model terms	No. tree nodes
G3P (polynomial)	Median	0.5677	0.6163	1.0049	1.3985	27	159.5
	Minimum	0.5397	0.5625	0.9417	1.3378	20	112
G3P (extended)	Median	0.5176	0.5650	0.8788	1.2774	27	166.5
	Minimum	0.4881	0.5247	0.8451	1.1815	22	131
LPV-ARX		0.7041	0.6606	2.5708	2.5571	80	–

Table 4. Models performances for payload sway prediction

Specification		OSA (RMSE)		MPO (RMSE)		Complexity	
		Training	Testing	Training	Testing	No. model terms	No. tree nodes
G3P (polynomial)	Median	0.0047	0.0033	0.0290	0.0314	18	95
	Minimum	0.0034	0.0026	0.0148	0.0177	10	44
G3P (extended)	Median	0.0039	0.0028	0.0145	0.0251	23	125.5
	Minimum	0.0037	0.0022	0.0122	0.0134	17	65
LPV-ARX		0.0036	0.0028	0.0367	0.0419	80	–

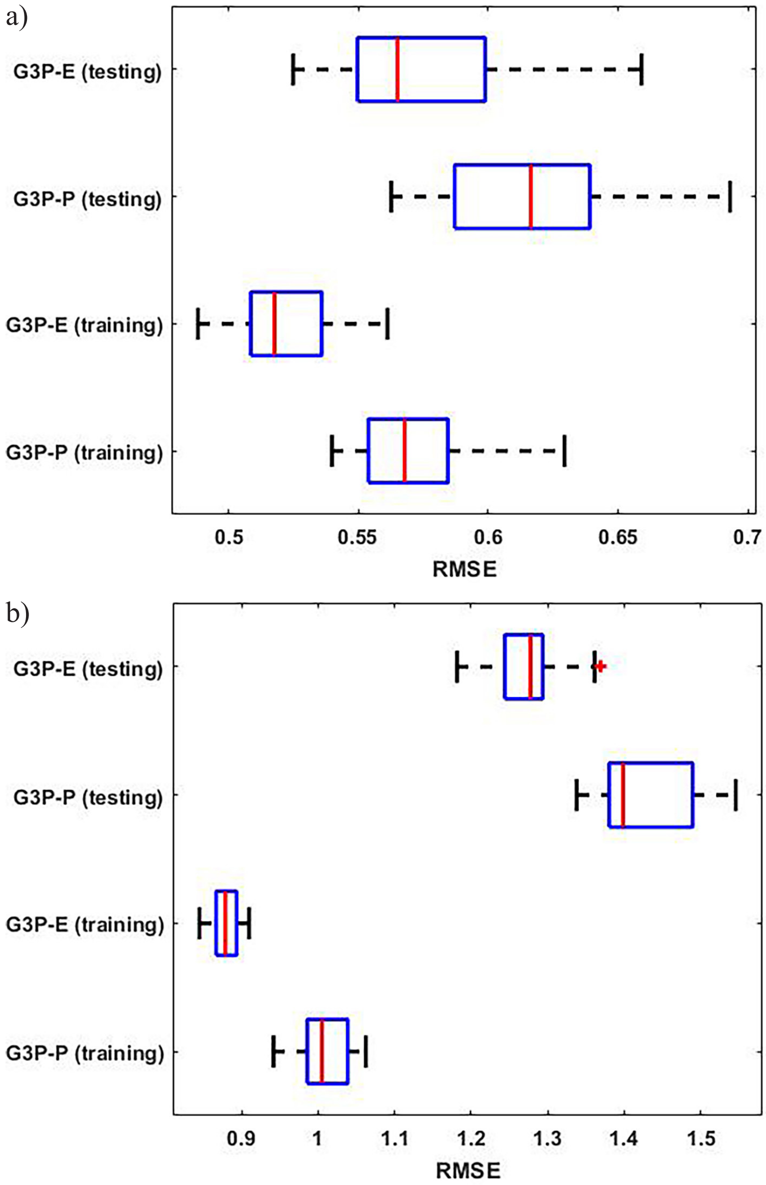

Figure 4. Performances of crane velocity models: a) OSA, b) MPO

Table 5. Execution time of the combined models

Specification	Median [s]	MAD [s]
G3P (polynomial)	1.4294×10^{-3}	2.9849×10^{-4}
G3P (extended)	4.8121×10^{-3}	7.0053×10^{-4}
LPV-ARX	9.6065×10^{-4}	4.0878×10^{-4}

approximately 2.5 and 1.7 times higher than the median G3P-NARX models with the extended grammar on the training and testing sets respectively.

The G3P velocity prediction model runs with the extended grammar had a median OSA RMSE of 0.5176 and 0.5650 on the training and testing set respectively and a median MPO RMSE of 0.8788 and 1.2774 on the training and testing set respectively. The runs with polynomial grammar had a median OSA RMSE of 0.5677 and 0.6163 on the training and testing set respectively and a median MPO RMSE of 1.0049 and 1.3985 on the training and testing set respectively. The median complexity of the models with the extended grammar was 27 and 166.5 when measured by the number of model terms and the number of tree nodes respectively while for the polynomial grammar were 27 and 159.5 when measured by the number of model terms and number of tree nodes respectively. There was a statistically significant difference in the medians of the OSA and MPO RMSE between the velocity models with the extended grammar and the polynomial grammar, however there was no statistical significance between the complexity both in terms of the number of model terms and the number of tree nodes. The G3P sway

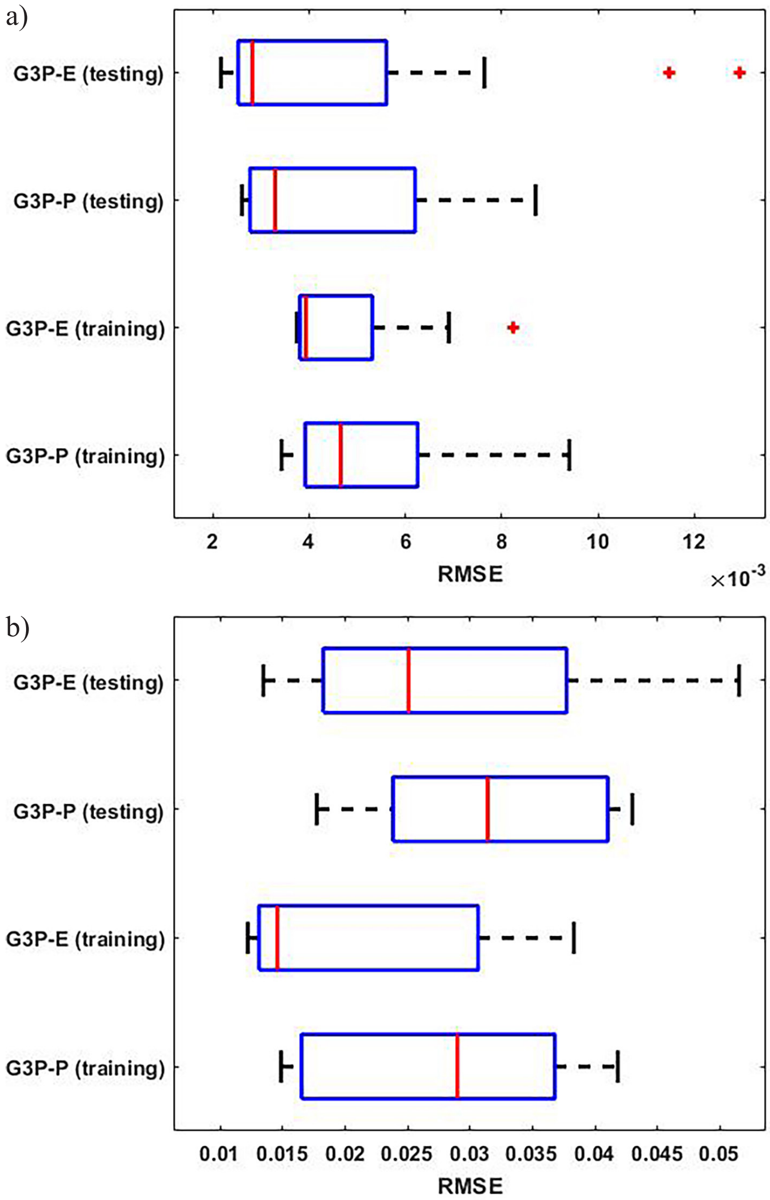

Figure 5. Performances of payload sway models: a) OSA, b) MPO

Table 6. Wilcoxon rank sum test p-values

Specification		Velocity	Sway
OSA	Training	$\mathbf{7.9479\times10^{-7}}$	0.2184
	Testing	$\mathbf{3.7499\times10^{-4}}$	0.2503
MPO	Training	$\mathbf{6.7956\times10^{-8}}$	**0.0060**
	Testing	$\mathbf{1.6571\times10^{-7}}$	0.0764
No. of model terms		0.9024	**0.0028**
No. of nodes		0.7149	**0.0074**

model runs with the extended grammar had a median OSA RMSE of 0.0039 and 0.0028 on the training and testing set respectively as while the OSA RMSE for the polynomial grammar was 0.0047 and 0.0033 on the training and testing set respectively. The median of the sway model runs MPO RMSE was 0.0145 and 0.0251 on the training and testing set respectively for the extended grammar and 0.0290 and 0.0314 on the training and testing set respectively for the polynomial grammar. The median complexity

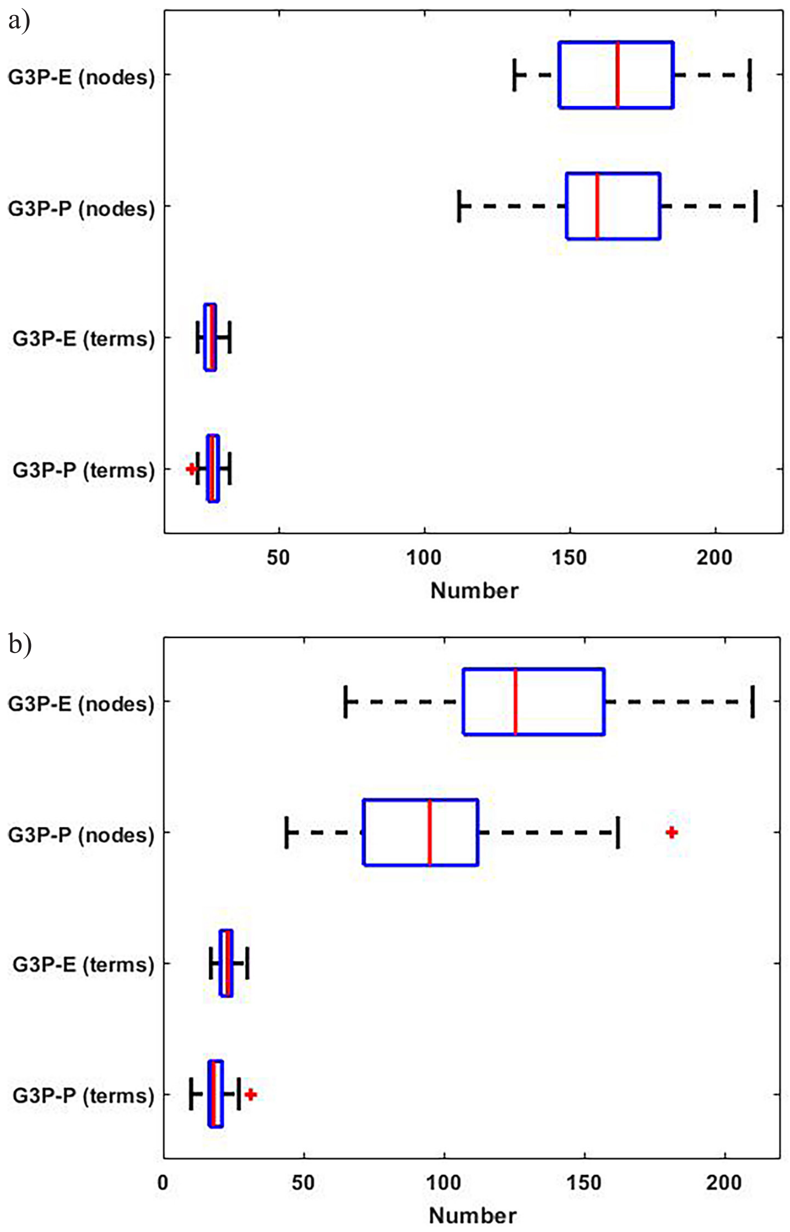

Figure 6. Complexity of G3P models: a) velocity model, b) sway model

for the sway model with the extended grammar was 23 and 125.5 when measured by the number of model terms and number of tree nodes respectively while for the polynomial grammar it was 18 and 95 respectively. The G3P sway model runs with the extended grammar did not result in statistically significant lower OSA and MPO RMSE on the testing set compared to the runs with the polynomial grammar, however it resulted in statistically significant lower MPO RMSE on the training set. The median difference in the complexity measures for the sway model were statistically significant meaning that the G3P with polynomial grammar resulted in less complex models.Although the Wilcoxon rank sum tests the null hypothesis that the medians are equal, when taking the best model out of all the runs, the G3P with the extended grammar had higher accuracy than the G3P with the polynomial grammar. The MPO of the G3P-NARX models with the lowest RMSE are compared with the LPV-ARX in Figure 7 and Figure 8 for the velocity and sway respectively. The LPV-ARX models, even though having the highest complexity measure in terms of the number of parameters, had the lowest execution time. The G3P-NARX models with the extended and polynomial grammar took approximately 5 and 1.5 times longer respectively to compute than the LPV-ARX model.

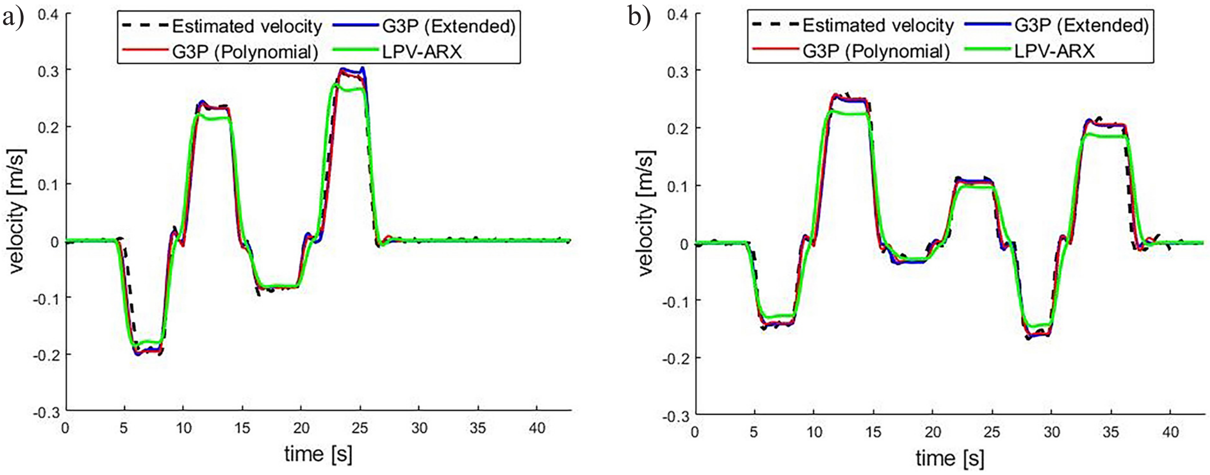

Figure 7. MPO of crane velocity model at an operating point: a) l = 1.7 m, m = 10 kg, b) l = 1.1 m, m = 50 kg

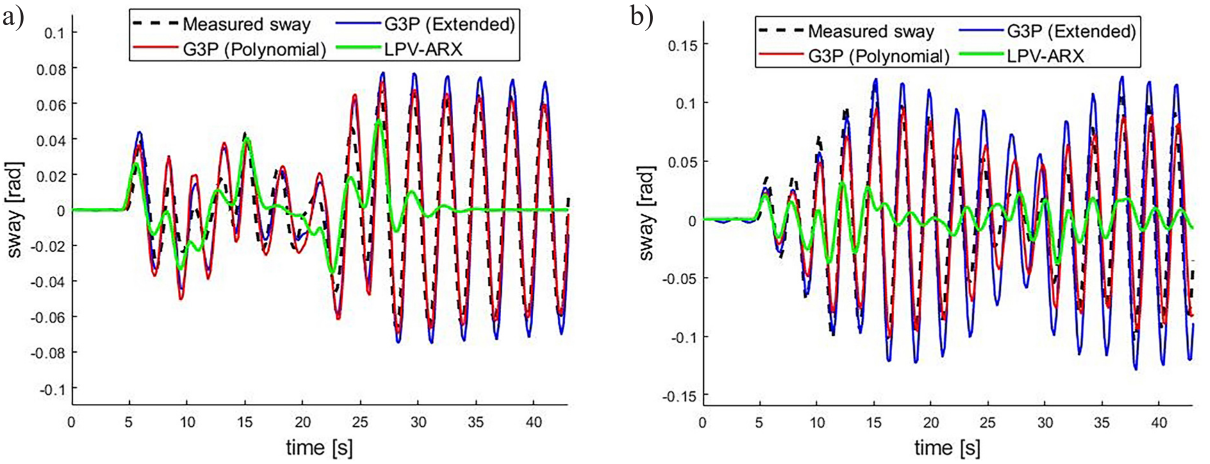

Figure 8. MPO of sway model at an operating point: a) l = 1.7 m, m = 10 kg, b) l = 1.1 m, m = 50 kg

CONCLUSIONS

In this paper input-output data was collected from a laboratory-scale overhead crane which was used to identify the system dynamics by decomposing the crane dynamics into an actuated and unactuated part. G3P used the input-output data to evolve NARX models thereby automating the model selection procedure, while a local search with a deterministic regularized least squares was used to obtain the model term coefficients and promote sparsity thereby reducing overfitting and resulting in less complex models. Due to the non-convex and non-smooth penalty, the fast monotone proximal gradient descent was used to find a sub-optimal solution to the regularized least squares problem. The G3P algorithm performance was tested for two different sets of grammar and the models were compared with an LPV-ARX model in terms of accuracy and complexity.

The G3P algorithm with both the polynomial grammar and the extended grammar obtained NARX models with a minimum MPO RMSE on the testing set of 1.3378 and 1.1815 respectively for the velocity model and a minimum MPO RMSE on the testing set of 0.0177 and 0.0134 respectively for the sway model, while the LPV-ARX NARX model with monomial basis functions had an MPO RMSE of 2.5571 and 0.0419 on the testing set for the velocity and sway models respectively. The LPV-ARX had a median execution time for evaluating the MPO of 9.6065×10^{-4}, as opposed to the median execution time for evaluating the MPO of 1.4294×10^{-3} and 4.8121×10^{-3} for the G3P models with polynomial and extended grammars respectively. The G3P algorithm with the extended grammar obtained more accurate velocity models than with the polynomial grammar while being equally complex, however, although the G3P algorithm with the extended grammar obtained the models with the smallest MPO RMSE, there was no statistical significance between the median MPO RMSE for the sway model when compared to the models obtained with the polynomial grammar while the polynomial models tended to be less complex in all measured metrics including the execution time. The aforementioned advantages of the model obtained with the polynomial grammar make it better suited for practical implementation of the model, as well as real time applications such as model predictive control where optimization is limited due to expensive function evaluations in a limited time frame.

REFERENCES

1. 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. https://doi.org/10.1016/j.ymssp.2017.03.015
2. Mojallizadeh M.R., Brogliato B., Prieur C. Modeling and control of overhead cranes: A tutorial overview and perspectives. Annu. Rev. Control. 2023; 56: 100877. https://doi.org/10.1016/j.arcontrol.2023.03.002
3. Yang T., Sun N., Chen H., Fang Y. Neural network-based adaptive antiswing control of an underactuated ship-mounted crane with roll motions and input dead zones. IEEE Trans. Neural Netw. Learn. Syst. 2020; 31: 901–914. https://doi.org/10.1109/TNNLS.2019.2910580
4. Tuan L.A. Neural observer and adaptive fractional-order backstepping fast-terminal sliding-mode control of RTG cranes. IEEE Trans. Ind. Electron. 2021; 68: 434–442. https://doi.org/10.1109/TIE.2019.2962450
5. Jakovlev S., Eglynas T., Voznak M. Application of neural network predictive control methods to solve the shipping container sway control problem in quay cranes. IEEE Access. 2021; 9: 78253–78265. https://doi.org/10.1109/ACCESS.2021.3083928
6. Rincon L., Kubota Y., Venture G., Tagawa Y. Inverse dynamic control via “simulation of feedback control” by artificial neural networks for a crane system. Control Eng. Pract. 2020; 94: 104203. https://doi.org/10.1016/j.conengprac.2019.104203
7. Kim G.H., Yoon M., Jeon J.Y., Hong K.S. Data-driven modelling and adaptive predictive anti-swing control of overhead cranes. Int. J. Control Autom. Syst. 2022; 20: 2712–2723. https://doi.org/10.1007/s12555-022-0025-8
8. Tuan L.A., Cuong H.M., Trieu P.V., Nho L.C., Thuan V.D., Anh L.V. Adaptive neural network sliding mode control of shipboard container cranes considering actuator backlash. Mech. Syst. Signal Process. 2018; 112: 233–250. https://doi.org/10.1016/j.ymssp.2018.04.030
9. Smoczek J. Experimental verification of a GPC-LPV method with RLS and P1-TS fuzzy-based estimation for limiting the transient and residual vibration of a crane system. Mech. Syst. Signal Process. 2015; 62–63: 324–340. https://doi.org/10.1016/j.ymssp.2015.02.019
10. Shao X., Zhang J., Zhang X. Takagi-Sugeno fuzzy modeling and PSO-based robust LQR anti-swing control for overhead crane. Math. Probl. Eng. 2019; 4596782. https://doi.org/10.1155/2019/4596782
11. Zhu X., Wang N. Hairpin RNA genetic algorithm based ANFIS for modeling overhead cranes. Mech. Syst. Signal Process. 2022; 165: 108326. https://doi.org/10.1016/j.ymssp.2021.108326
12. Ma S.F., Leylaz G., Sun J.-Q. Identification of differentially flat output of underactuated dynamic systems. Int. J. Control. 2022; 95: 114–125. https://doi.org/10.1080/00207179.2020.1779960
13. Bao H., Kang Q., An J., Ma X., Zhou M. A performance-driven MPC algorithm for underactuated bridge cranes. Machines 2021; 9: 177. https://doi.org/10.3390/machines9080177
14. Maksakov A., Golovin I., Shysh M., Palis S. Data-driven modeling for damping and positioning control of gantry crane. Mech. Syst. Signal Process. 2023; 197: 110368. https://doi.org/10.1016/j.ymssp.2023.110368
15. Kusznir T., Smoczek J. Multi-gene genetic programming-based identification of a dynamic prediction model of an overhead traveling crane. Sensors 2022; 22: 339. https://doi.org/10.3390/s22010339
16. Zou H., Hastie T. Regularization and variable selection via the Elastic Net. J. R. Stat. Soc. B Stat. 2005; 67: 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x
17. McConaghy T. FFX: Fast, scalable, deterministic symbolic regression technology. In: Genetic Programming Theory and Practice IX: Genetic and Evolutionary Computation. Riolo R., Vladislavleva E., Moore J. (eds.), Springer, New York, NY, USA 2011; 235–260. https://doi.org/10.1007/978-1-4614-1770-5_13
18. Icke I., Bongard, J.C. Improving genetic programming based symbolic regression using deterministic machine learning. In IEEE Congress on Evolutionary Computation, Cancun, Mexico, 2013; 1763–1770. https://doi.org/10.1109/CEC.2013.6557774

19. Vanneschi L., Castelli M. Soft target and functional complexity reduction: A hybrid regularization method for genetic programming. Expert Syst. Appl. 2021; 177: 114929. https://doi.org/10.1016/j.eswa.2021.114929
20. Kusznir T., Smoczek J., Karwat B. Data-driven identification of crane dynamics using regularized genetic programming. *Appl. Sci.* 2024; *14*: 3492. https://doi.org/10.3390/app14083492
21. Al-Alaoui M.A. Linear phase low-pass IIR digital differentiators. IEEE Trans. Signal Process. 2007; 55: 697–706. https://doi.org/10.1109/TSP.2006.885741
22. Billings S.A. Nonlinear system identification: NARMAX methods in the time, frequency, and spatio–temporal domains. John Wiley & Sons, Ltd. 2013.
23. Natarajan B.K. Sparse approximate solutions to linear systems. SIAM J. Comput. 1995; 24: 227–234. https://doi.org/10.1137/S0097539792240406
24. Yang Y., Yu, J. Fast proximal gradient descent for a class of non-convex and non-smooth sparse learning problems. In: Proceedings of the 35th Uncertainty in Artificial Intelligence Conference 2019; 115: 1253–1262.
25. Koza J.R. Genetic programming: on the programming of computers by means of natural selection: complex adaptive systems. MIT Press, Cambridge, MA, USA 1992.
26. Rojas C.R., Hjalmarsson H. Sparse estimation based on a validation criterion. In Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA 2011; 2825–2830.
27. Toth R., Hjalmarsson H., Rojas C.R. Order and structural dependence selection of LPV-ARX models revisited. In Proceedings of the 51st IEEE Conference on Decision and Control, Maui, HI, USA 2012; 6271–6276.
28. Van Den Berg E., Friedlander M.P. Probing the Pareto Frontier for basis pursuit solutions. SIAM J. Sci. Comput. 2009; 31: 890–912. https://doi.org/10.1137/080714488
29. Luke S. Two fast tree-creation algorithms for genetic programming. IEEE Trans. Evol. Comput. 2000; 4: 274–283. https://doi.org/10.1109/4235.873237