

Article

# Nonlinear Model Predictive Control with Evolutionary Data-Driven Prediction Model and Particle Swarm Optimization Optimizer for an Overhead Crane

Tom Kuszniir  and Jarosław Smoczek \* 

Faculty of Mechanical Engineering and Robotics, AGH University of Krakow, 30-059 Kraków, Poland;  
tkuszniir@agh.edu.pl

\* Correspondence: smoczek@agh.edu.pl

**Abstract:** This paper presents a new approach to the nonlinear model predictive control (NMPC) of an underactuated overhead crane system developed using a data-driven prediction model obtained utilizing the regularized genetic programming-based symbolic regression method. Grammar-guided genetic programming combined with regularized least squares was applied to identify a nonlinear autoregressive model with an exogenous input (NARX) prediction model of the crane dynamics from input–output data. The resulting prediction model was implemented in the NMPC scheme, using a particle swarm optimization (PSO) algorithm as a solver to find an optimal sequence of the control actions satisfying multi-objective performance requirements and input constraints. The feasibility and performance of the controller were experimentally verified using a laboratory crane actuated by AC motors and compared with a discrete-time feedback controller developed using the pole placement technique. A series of experiments proved the effectiveness of the controller in terms of robustness against operating condition variation and external disturbances.

**Keywords:** overhead crane; oscillation suppression; nonlinear model predictive control; data-driven modeling; genetic programming; particle swarm optimization; sparse regression



**Citation:** Kuszniir, T.; Smoczek, J. Nonlinear Model Predictive Control with Evolutionary Data-Driven Prediction Model and Particle Swarm Optimization Optimizer for an Overhead Crane. *Appl. Sci.* **2024**, *14*, 5112. <https://doi.org/10.3390/app14125112>

Academic Editor: Liviija Cveticanin

Received: 15 May 2024

Revised: 3 June 2024

Accepted: 6 June 2024

Published: 12 June 2024



**Copyright:** © 2024 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

Crane systems have been attractive for decades in the modeling and control field, mainly due to their underactuated properties that present challenging modeling and control problems that are theoretically interesting and significant from a practical viewpoint. Since cranes are widely used in the industrial, construction, and logistic sectors for transporting heavy and oversized loads by using wire ropes, cables, or chains, an excessive transient and residual cable-suspended payload oscillation caused by actuated horizontal motion degrades positioning performance and may pose safety hazards [1]. This problem is extensively studied in the literature, and various open-loop and closed-loop control strategies are comprehensively reviewed in [2], while the more current state of the art focusing on the modeling and control of overhead cranes is provided in [3]. Recent works include input shaping [4,5], time-optimal [6], command shaping [7], and feedforward [8] anti-sway crane control strategies. New concepts in the field of closed-loop control have been developed using PID-like coupling control [9,10], sliding mode control [11] combined with a state observer [12] or PD controller [13], passivity-based control [14,15], H-infinity [16], and back-stepping-based [17] control schemes.

An increase in computing power has resulted in growing applications of linear MPC and nonlinear MPC (NMPC) in a variety of processes. MPC has received attention in relation to crane control systems because it allows the implementation of a set of constraints in the optimization algorithm that keep the crane in a feasible and safe operating region. A combination of feedforward control and NMPC was used in numerical simulations for the payload sway suppression of a planar nonlinear model of a shipboard crane [18]. A

real-time MPC method with a linearized model of a mobile harbor boom crane that is hydraulically actuated was developed in [19] and tested on a LIEBHERR harbor mobile crane with a load capacity of 104 tons and a rope length of up to 70 m. This work is further developed in [20], where the nonlinear behavior of the system is simplified by applying exact linearization and input–output linearization. Tysse et al. [21] proposed combining a Lyapunov-based controller to stabilize the pendulum dynamics and an NMPC scheme used to control the position of a small-scale knuckle boom crane. An NMPC-based scheme was designed in [22] to optimize the pendulum motion driven by a mechatronic cart–winch system. In [23], the dynamics of a 3D overhead crane system were decoupled into two second-order systems, and two separate NMPC schemes were developed for simultaneously positioning a payload along the two motion planes of a crane’s bridge and trolley, respectively. Experiments carried out on a small laboratory-scale overhead crane confirmed the effectiveness of this strategy. In [24], the MPC approach is applied to minimize an objective function that is formulated as the integration of energy consumption and swing angle. In [25], the authors developed a path-following controller for a tower crane using nonlinear model predictive control with a fast suboptimal solution and verified the algorithm in simulations and on a small experimental stand. In [26], the authors proposed a model predictive controller for an overhead crane. The model was linearized around the equilibrium point, and the sway constraints were converted to input constraints using the kinematic equation of the crane system. The MPC strategy is employed in [27] to optimize the crane velocity profile when the maximum position overshoot is given by the crane operator.

The aforementioned studies are mostly based on physics-based models of underactuated crane dynamics. However, the accurate analytical modeling of complex coupled crane dynamics can be difficult and time-consuming; thus, some recent works have proposed data-driven modeling approaches. An MPC strategy involving a discrete-time linear model, recursive least-squares parameter estimation, and a PSO solver of a multi-objective control optimization problem was proposed and tested on a laboratory overhead crane driven by DC motors in [28]. This linear MPC scheme is based on a straightforward approach that assumes a parametric model of crane dynamics, thereby reducing the identification task to parameter estimation using the RLS algorithm with parameter projection. The parametric identification procedure assumes there is an adequate model structure representation with uncertain parameters that are subject to optimization. Parametric identification requires fewer data to identify a model; however, some a priori knowledge and engineering experience with respect to the analyzed system are required to choose a suitable model structure [29]. Moreover, there is no unique model structure corresponding to different crane applications under different operating conditions. In this case, non-parametric methods and machine learning techniques are alternative approaches that can be useful for determining a model structure for a specific process and data set. Jakovlev et al. [30] analyzed several multi-layer perceptron neural network (NN) structures and learning approaches (e.g., variable learning rate backpropagation, scaled conjugate gradients, recursive prediction error, and the Levenberg–Marquardt algorithm) to solve a time-series prediction problem in an MPC scheme for a quay crane. In [31], an overhead crane’s dynamics were approximated by an adaptive neuro-fuzzy inference system (ANFIS) trained with operational data using a hairpin RNA genetic algorithm. Kim et al. [32] used the NARX-NN data-driven model trained offline and online using extreme learning machines (ELMs) in the predictive control of a small-scale overhead crane with a prediction and control horizon of  $N_p = N_c = 15$ . The Koopman operator theory and deep learning are merged in [33] to identify a model of a gantry crane subsequently used to develop a linear quadratic controller tested on a laboratory stand. A data-driven active learning control scheme based on the Koopman operator was studied in simulations on a bridge crane model [34]. Different structures of artificial neural network (ANN) models of an overhead crane’s inverse dynamics were trained and validated using the Levenberg–Marquardt algorithm, with data computed in simulations carried out using a model derived analytically in [35]. A

hybrid PID and MPC scheme with Bayesian optimization for the data-driven identification of linear-model and controller parameter tuning was tested on a laboratory-scale overhead crane in [36]. An interesting alternative to model-based control is data-driven model-free control, which has recently proven its effectiveness for complex dynamical problems [37,38]. Examples of such studies for underactuated crane systems are presented in [39,40].

The selection of an appropriate model structure is a crucial point of identification that requires a tradeoff between model fitting and parsimony. Although the machine learning data-driven discovery of crane dynamics has been successfully explored in recent works, especially using an ANN framework, the problem of nonlinear model structure selection is rarely studied. To address this issue, a non-parametric data-driven identification of overhead crane dynamics using multi-gene genetic programming (MGGP) is proposed in [41]. However, symbolic regression can be prone to generating complex expressions and overfitting when aiming to decrease prediction error [42]. Thus, in [43], grammar-guided genetic programming (G3P) combined with  $l_0$  sparse regression is applied to find a compromise between model complexity and prediction accuracy.

Following the work presented in [43] and in comparison to the linear parametric modeling-based MPC approach presented in [28], this paper develops an NMPC control strategy for an overhead crane based on a prediction model constructed using a regularized genetic programming (GP)-based approach. A polynomial NARX model structure is optimized using an evolutionary process and regularized least squares; the resulting prediction model is implemented in the NMPC scheme with a PSO algorithm used as the solver to find an optimal sequence of the control actions satisfying the multi-objective performance requirements and input constraints. The experimental verification of the proposed controller is carried out on a laboratory overhead crane by testing the NMPC performance at different operating points in the presence of external disturbances and comparing the results with those of a discrete-time feedback controller developed using the pole placement method. The main contributions of this paper are as follows:

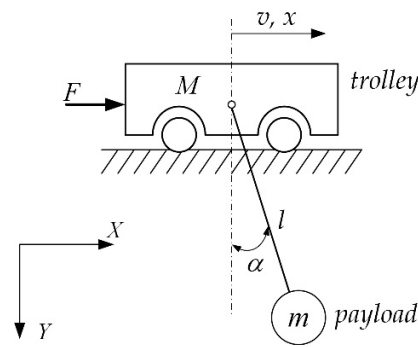
- (1) A regularized genetic programming data-driven modeling approach is used to develop an NMPC scheme for an underactuated overhead crane system.
- (2) The proposed data-driven modeling approach and model-based control scheme are experimentally verified under different operating conditions and in terms of robustness with respect to external disturbances.

The rest of this paper is organized as follows: Section 2 introduces modeling assumptions, provides the identification procedure developed using GP with sparse regression, and describes the NMPC control strategy with the PSO solver. The hardware experimental setup and the results are presented in Section 3, while the final conclusions are made in Section 4.

## 2. Materials and Methods

### 2.1. Data-Driven Modeling of Crane Dynamics

The proposed data-driven modeling and NMPC approach are studied for a planar lumped-mass model of an overhead crane. A simplified planar single-pendulum system, often used as a benchmark case to develop and evaluate different anti-sway control strategies [1–3], is illustrated in Figure 1, where  $M$ ,  $m$ ,  $l$ ,  $\alpha$ ,  $x$ ,  $v$ ,  $F$  are the masses of the trolley and pendulum, length of the pendulum, sway angle of the pendulum, trolley displacement and velocity, and force, respectively. The force consists of the actuating force (in real applications, usually produced by AC motors supplied by frequency inverters), disturbances, and friction forces. In this configuration, we assume that  $\alpha$ ,  $x$ , and  $v$  are measurable variables, the rope length  $l$  and the mass of the payload  $m$  are measurable or known parameters, and the actuating force is produced by an actuator controlled by a control signal  $u$ .



**Figure 1.** Schematic diagram of an overhead crane in 2D operating space.

The crane is assumed to have high mechanical impedance, which allows for a decomposition of the crane dynamics into the actuated and unactuated parts and, therefore, can be modeled by two discrete-time nonlinear models, which can be presented in the NARX (nonlinear autoregressive with exogenous inputs) form with operating parameters  $l$  and  $m$  as

$$v(k) = f_1(u(k-1), \dots, u(k-N_u), v(k-1), \dots, v(k-N_v), l, m) + e_1(k) \tag{1}$$

$$\alpha(k) = f_2(v(k-1), \dots, v(k-N_v), \alpha(k-1), \dots, \alpha(k-N_\alpha), l, m) + e_2(k) \tag{2}$$

where  $N_u, N_v, N_\alpha$  are the maximum lags for the system input and output,  $e_1(k)$  and  $e_2(k)$  are the errors at discrete time  $k$ , and the rope length  $l$  and the mass of the payload  $m$  are included in the nonlinear terms to take into account the variation in operating conditions. The nonlinear functions  $f_1$  and  $f_2$  can be reformulated in the linear-in-the-parameters regression functions as

$$v(k) = \phi_1^T(k)\theta_1 + e_1(k) \tag{3}$$

$$\alpha(k) = \phi_2^T(k)\theta_2 + e_2(k) \tag{4}$$

where  $\phi_1$  and  $\phi_2$  are the vectors of the nonlinear regressors composed of the lagged input and output and the operating parameters  $l$  and  $m$ , while  $\theta_1$  and  $\theta_2$  are vectors of the linear parameters that can be estimated using the least squares technique. The crane position at time step  $k$  is approximated by an explicit Euler integration of the velocity to obtain

$$x(k) = x(k-1) + t_s v(k) \tag{5}$$

where  $t_s$  is the sample time.

### 2.2. Genetic Programming-Based Identification

Under the modeling assumptions given in (1) and (2), the NARX model structure is selected with additional operating parameters in order to identify the input–output nonlinear crane dynamics, which can be presented in the general form

$$y(k) = f(\xi(k-1), \dots, \xi(k-N_\xi), y(k-1), \dots, y(k-N_y), p) + e(k) \tag{6}$$

where  $y$  is the output,  $\xi$  is the input excitation signal,  $e$  is Gaussian white noise, and  $p$  is the operating parameter.

The coefficients of the linear-in-the-parameters model can be estimated using the least squares method, as follows:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \|\Phi\theta - y\|_2^2 \tag{7}$$

where  $\Phi$  is a matrix, whose columns are the nonlinear regressors.

It is assumed that the dynamics of the system can be described by a subset of the regressors in  $\Phi$ ; therefore, the coefficients  $\theta$  are sparse. In order to promote the sparsity of the solution, the  $l_0$  penalty is added to (7) to obtain

$$F(\theta) = g(\theta) + h(\theta) = \frac{1}{2} \|\Phi\theta - y\|_2^2 + \lambda \|\theta\|_0 \tag{8}$$

where  $\|\theta\|_0$  is the  $l_0$  pseudonorm, i.e.,  $\|\theta\|_0 = \sum_{i=1}^N \mathbb{1}_{\theta_i \neq 0}$ . Problem (8) is nonconvex and NP-hard [44], therefore several methods have been proposed to obtain sparse solutions, such as replacing the  $l_0$  pseudonorm with the  $l_1$  norm, smoothly clipped absolute deviation (SCAD), minimax concave penalty (MCP), sequentially thresholded least squares (STLS), and proximal gradient methods. In this paper, a suboptimal solution is found by using monotone accelerated proximal gradient descent (mAPG) [45], which converges to a critical point of  $F(\theta)$ . The proximal mapping of a function  $h$  is given by the proximal operator

$$\text{prox}_h(q) = \underset{\mu}{\text{argmin}} h(\mu) + \frac{1}{2} \|\mu - q\|_2^2 \tag{9}$$

which for  $h(\cdot) = \lambda \|\cdot\|_0$  is separable and results in the hard thresholding operator with threshold  $\tau$ , as follows:

$$[T_\tau(q)]_s = \begin{cases} 0, & |q_s| \leq \tau \\ q_s, & \text{otherwise} \end{cases}, 1 \leq s \leq n \tag{10}$$

where  $s$  is the element index and  $n$  is the number of elements.

The monotone accelerated proximal gradient descent is initialized with  $z^{(1)} = \theta^{(1)} = \theta^{(0)}$ ,  $t_0 = 0$ ,  $t_1 = 1$ , and updates  $\theta^{(j+1)}$  according to

$$\theta^{(j+1)} = \begin{cases} z^{(j+1)}, & F(z^{(j+1)}) \leq F(v^{(j+1)}) \\ q^{(j+1)}, & \text{otherwise} \end{cases} \tag{11}$$

where

$$\begin{aligned} z^{(j+1)} &= \text{prox}_{\gamma_r \lambda \|\cdot\|_0} \left( r^{(j)} - \gamma_r \nabla g \left( r^{(j)} \right) \right) = T_{\sqrt{2\lambda\gamma_r}} \left( r^{(j)} - \gamma_r \nabla g \left( r^{(j)} \right) \right), \\ q^{(j+1)} &= \text{prox}_{\gamma_\theta \lambda \|\cdot\|_0} \left( \theta^{(j)} - \gamma_\theta \nabla g \left( \theta^{(j)} \right) \right) = T_{\sqrt{2\lambda\gamma_\theta}} \left( \theta^{(j)} - \gamma_\theta \nabla g \left( \theta^{(j)} \right) \right), \\ r^{(j)} &= \theta^{(j)} + \frac{t_{j-1}}{t_j} \left( z^{(j)} - \theta^{(j)} \right) + \frac{t_{j-1}-1}{t_j} \left( \theta^{(j)} - \theta^{(j-1)} \right), \\ t_{j+1} &= \frac{\sqrt{1+4t_j^2}+1}{2}, \end{aligned}$$

where  $j$  is the iteration number and  $\gamma_\theta$  and  $\gamma_r$  are the step sizes. The algorithm has been proven to converge to a critical point when  $\gamma_\theta, \gamma_r < \frac{1}{L}$  [45], where  $L$  is the Lipschitz constant of  $\nabla g$ . However, in order to speed up convergence, the Barzilai–Borwein [46] method is used to initialize the step size, and a line search is used to find the step sizes  $\gamma_\theta$  and  $\gamma_r$  that result in sufficient descent.

$$F(q_{j+1}) \leq F(\theta_j) + \delta \|q_{j+1} - \theta_j\|^2 \tag{12}$$

$$F(z_{j+1}) \leq F(r_j) + \delta \|z_{j+1} - r_j\|^2 \tag{13}$$

where  $\delta > 0$  is a small positive constant.

The regressors are evolved using grammar-guided genetic programming, which overcomes the need for the closure property required in traditional genetic programming to produce valid offspring. Additionally, user knowledge of the system can be incorporated into the context-free grammar to restrict the search space. A context-free grammar  $G$  is the 4-tuple  $(S, N, \Sigma, P)$ , where  $S \in N$  is the start symbol,  $N$  is a finite set of nonterminal

symbols,  $\Sigma$  is an alphabet of terminal symbols, and  $P$  is the set of production rules of the form  $X \rightarrow \beta$ , where  $X \in N$  and  $\beta \in (N \cup \Sigma)^*$ .

For identifying the crane system dynamics, the grammar was constructed in such a manner that the model regressors are monomials, thereby resulting in a polynomial NARX model; however, the grammar allows for additional operations on the parameters  $l$  and  $m$ , such as division and square root. The grammar, in Backus–Naur form (BNF), is given in Table 1.

**Table 1.** Grammar used for polynomial NARX models.

$\langle S \rangle ::=$	$\langle \text{exp} \rangle \mid \langle T \rangle$
$\langle \text{exp} \rangle ::=$	$\times \langle \text{exp} \rangle \langle \text{exp} \rangle \mid \times \langle \text{exp}_p \rangle \langle \text{exp} \rangle \mid \langle T \rangle$
$\langle \text{exp}_p \rangle ::=$	$\times \langle \text{exp}_p \rangle \langle \text{exp}_p \rangle \mid \div \langle \text{exp}_{p2} \rangle \langle \text{exp}_{p2} \rangle \mid \sqrt{\langle \text{exp}_p \rangle} \mid \langle T_p \rangle$
$\langle \text{exp}_{p2} \rangle ::=$	$+ \langle \text{exp}_{p2} \rangle \langle \text{exp}_{p2} \rangle \mid \times \langle \text{exp}_{p2} \rangle \langle \text{exp}_{p2} \rangle \mid \sqrt{\langle \text{exp}_{p2} \rangle} \mid \langle T_p \rangle$
$\langle T \rangle ::=$	$\xi(k-1) \mid \dots \mid \xi(k-N_\xi) \mid y(k-1) \mid \dots \mid y(k-N_u)$
$\langle T_p \rangle ::=$	$m \mid l$

In every generation, new individuals, called offspring, are produced by selecting individuals from the existing population, called parents, that pass through a stochastic variation operator. A total of four different variation operators were used: two binary variation operators, namely, subtree crossover and high-level crossover, and two unitary variation operators, namely, subtree mutation and point mutation.

Since every model term is represented by a separate tree, the selection of a parent occurs in two stages. In the first stage, the model is selected from the population by tournament selection, after which a model term is selected from the model to undergo variation. The model term is selected with the probability of selection given by

$$P = \frac{\exp \bar{\theta}_i}{\sum_{i=1}^M \exp \bar{\theta}_i} \tag{14}$$

where  $\bar{\theta}_i = \frac{\sum_{i=1}^M |\theta_i| - |\theta_i|}{\sum_{i=1}^M |\theta_i|}$  is the normalized term coefficient.

In both crossover methods, two parents are selected; in subtree crossover, two root nodes of the subtrees are selected, and the subtrees are switched in a manner so that the resulting offspring is syntactically correct. In high-level crossover [47], the selected model terms are switched between the two parents. In both mutational methods, only one parent is selected from the population. Subtree mutation is similar to subtree crossover, but instead of a subtree from a second parent, a new subtree is randomly generated. In point mutation, a single terminal node is changed in a manner so that the resulting offspring is syntactically correct. The genetic programming parameters used for the identification of the overhead crane dynamics are given in Table 2.

**Table 2.** GP parameters.

Parameters	Values
Population size	100
Number of generations	500
Initialization method	PTC2
Max tree depth during initialization	6
Max number of candidate model terms	50
Tournament size	2
Subtree crossover probability	0.75
High-level crossover probability	0.15
Mutation probability	0.1
Max number of proximal evaluation iterations	5000

The initialization method used was the probability tree creation 2 (PTC2) algorithm [48]. Due to the stochastic nature of genetic programming, each run could result in a possibly different solution; therefore, a total of 15 runs were carried out, and the solutions were compared in terms of accuracy, measured by using the mean square error, and complexity, measured by the number of model terms included in the model. One model from the Pareto frontier is then selected as the model used in the nonlinear model predictive control algorithm.

### 2.3. NMPC with PSO Solver

An NMPC controller with a PSO metaheuristic algorithm as the solver of the constrained optimal control problem was developed, implemented, and tested in real-time experiments on a laboratory stand. PSO is a stochastic optimization technique that was originally developed by Eberhart and Kennedy [49]. There have been several variations of the PSO algorithm, including utilizing different topologies, such as the star topology, which uses the swarm's global best position, or the ring topology, in which a particle uses the local best position in its neighborhood. While the original PSO formulation used the star topology, it has been proposed to use the ring topology and limit the star topology to unimodal problems; however, according to [50], the star topology is still acceptable for multi-objective optimization problems. An empirical study by Engelbrecht [51] on 60 benchmark functions showed that the two topologies perform similarly, with the star topology performing slightly better in terms of success rate and efficiency, while the ring topology performed slightly better in terms of consistency. Additionally, PSO variants with sub-swarm populations have been proposed, such as dynamic multi-swarm PSO [52] and symbiotic particle swarm optimization [53]. A review of the different PSO variants is presented in [54].

In the proposed control scheme, at each time step,  $k$ , the NMPC-GP receives the current state of the plant and solves the constrained optimization problem to find a sequence of control actions  $u(k+i-1)$  defined over the control horizon  $N_u$  to minimize the cost function, which depends on the deviation of the process variables  $x(k+i)$  and  $\alpha(k+i)$  from the set point  $x_r(k+i)$  as well as the control effort required to change the system states over the prediction horizon  $N_p$ . The PSO algorithm solves the optimization problem with the constraint  $u(k+i+1) \in \mathbb{U}$  directly as one large optimization problem, and, therefore, there is no requirement for breaking down the problem into smaller subproblems as is the case with dynamic programming. The future trajectories of the states are predicted using (3–5) based on measurements at the current time instant  $k$  and the candidate control sequence  $u(k+i+1)$  obtained iteratively using the PSO algorithm (Figure 2).

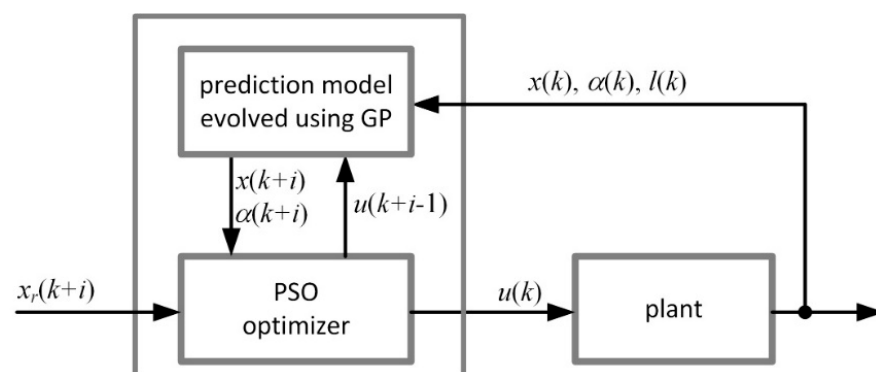


Figure 2. NMPC-GP scheme.

The proposed cost function includes the sum of the squared prediction errors between the desired and predicted crane positions on the prediction horizon, which corresponds to shifting the equilibrium to the desired position; the sum of the weighted squares of the

predicted payload angle on the prediction horizon; and the weighted squares of the control increments on the control horizon, as given in

$$J(k+i) = \sum_{i=1}^{N_p} \lambda_1 (e(k+i))^2 + \sum_{i=1}^{N_p} \lambda_2 (\alpha(k+i))^2 + \sum_{i=1}^{N_u} \lambda_3 (\Delta u(k+i-1))^2 \quad (15)$$

where

$$e(k+i) = x(k+i) - x_r(k+i),$$

$$\Delta u(k+i-1) = u(k+i-1) - u(k+i-2),$$

$$\lambda_1 = \begin{cases} 0 & \text{if } |e(k+i)| \leq e_a \\ 1 & \text{if } |e(k+i)| > e_a \end{cases},$$

$$\lambda_2 = \begin{cases} 0 & \text{if } |\alpha(k+i)| \leq \alpha_a \\ \lambda_{20} & \text{if } |\alpha(k+i)| > \alpha_a \end{cases}.$$

The position error,  $e(k+i)$ , is the difference between the predicted position  $x(k+i)$  and the reference signal  $x_r$  at time  $k+i$ ;  $\alpha(k+i)$  is the predicted sway angle of the payload;  $\Delta u(k+i-1)$  expresses the control effort, which is calculated as the predicted change in the control signal along the control horizon; and  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  are the weighting coefficients. The control goal is to reach the desired set point within a specified tolerance, and therefore the cost function coefficients  $\lambda_1$  and  $\lambda_2$  are set to  $\lambda_1 = 0$  and  $\lambda_2 = 0$  when the expected crane positioning accuracy  $e_a$  is satisfied and the payload sway oscillation is within the acceptable tolerance  $\alpha_a$ , respectively; otherwise, nonzero weights are applied ( $\lambda_1 = 1$ ,  $\lambda_2 = \lambda_{20}$ ). The tolerance  $\alpha_a$  of the pendulum oscillation corresponds to the range of the payload linear deviation  $d_p$  from the stable equilibrium point, where the linear deviation of a payload depends on the rope length  $d_p = l \sin \alpha$ , and for small values of the sway angle, it can be approximated as  $d_p \approx l\alpha$ . In the experiments carried out in this work, the crane and payload positioning accuracy was assumed to be  $e_a = d_p = 0.02$  m. Thus, during the experiments, the desired range of magnitude of the residual sway was set as  $\alpha_a = e_a/l = 0.02/l$ .

The PSO algorithm is adapted to determine the sequence of control signal values  $u(k+i-1)$  over the control horizon  $N_u$  to minimize the cost function (15). Consider a population of candidate solutions (a swarm of particles) in the constrained  $N_u$ -dimensional search space, where each particle's position is represented by the coordinates corresponding to the sequence of control signals over the control horizon  $N_u$ , as follows:

$$u_j = [u(k+i-1), u(k+1), \dots, u(k+N_u-1)] \quad (16)$$

satisfying the box constraints

$$u_{\min} \leq u_j \leq u_{\max} \quad (17)$$

where  $j = 1, 2, \dots, n_s$ ; and  $n_s$  is the size of the swarm. At each time step,  $t$ , called an iteration, the  $j^{\text{th}}$  particle's position and velocity are updated in the hyperspace according to

$$\tilde{u}_j^t = w\tilde{u}_j^{t-1} + c_1r_1(u_{j,p} - u_j^{t-1}) + c_2r_2(u_g - u_j^{t-1}) \quad (18)$$

$$u_j^t = u_j^{t-1} + \tilde{u}_j^t \quad (19)$$

where  $u_j^{t-1}$  and  $\tilde{u}_j^{t-1}$  are the particle's position and velocity in the previous iteration, respectively;  $u_{j,p}$  and  $u_g$  are the personal and global best solutions, respectively;  $w$  is called the inertia weight; positive constants  $c_1$  and  $c_2$  are called the acceleration coefficients and are responsible for the influence of the global and local attractors on the particle's velocity; and constants  $r_1$  and  $r_2$  are drawn uniformly from the interval  $[0, 1]$ . In this work, following [55], the inertia weight varies linearly between 0.9 and 0.4, together with the acceleration coefficients of 1.5, to ensure an efficient balance between exploration (the



ability of the particles to search wide regions of the solution space) and exploitation (the ability of the particles to refine the solution in the neighborhood of an attractor).

Since the velocity of a particle refers to the change in a control signal, which is constrained by the lower and upper limits ( $u_{min}$ ,  $u_{max}$ ), the velocity in formula (18) is updated by taking into account the control signal limitation:

$$\tilde{u}_j^t = \max\left(\min\left(\tilde{u}_j^t, \tilde{u}_{j,max}^t\right), \tilde{u}_{j,min}^t\right) \tag{20}$$

where  $\tilde{u}_{j,min}^t = u_{min} - u_j^{t-1}$  and  $\tilde{u}_{j,max}^t = u_{max} - u_j^{t-1}$ .

The constraint described in (20) is the absorption constraint, i.e., if the particle's velocity would cause the particle to leave the feasible region, then the boundary "absorbs" the excess velocity, causing the particle to stay on the boundary. The personal best position  $u_{j,p}$  of the  $j$ -th particle and the global best position of swarm  $u_g$  are updated at each iteration. The iterative process is repeated at each time step  $k$ , starting with a randomly initialized swarm of particles and their velocities satisfying condition (20). The initial population size is  $n_s - 1$  and is completed by the global best solution from the previous time step  $k-1$  to enhance the convergence speed. The PSO searching process is terminated when the maximum number of iterations is reached and the first coordinate of the global best particle  $u(k)$  is applied as the control signal at the current instant  $k$ .

#### 2.4. Digital Feedback Controller

The NMPC-GP with the PSO solver is compared with a discrete-time feedback control system developed using the pole placement technique. The crane dynamics are decomposed in a similar manner as in Section 2.1 into the actuated and unactuated parts and identified using an output-error approach to obtain a discrete-time relation between the crane velocity  $v(k)$  and the control input  $u(k)$ , approximated by a first-order discrete-time model (21), while the relation between the payload sway angle  $\alpha(k)$  and the velocity  $v(k)$  is given by a second-order discrete-time model (22):

$$v(k) = \frac{D(z^{-1})}{C(z^{-1})}u(k) \tag{21}$$

$$\alpha(k) = \frac{B(z^{-1})}{A(z^{-1})}v(k) \tag{22}$$

where  $z^{-1}$  is the backward shift operator and  $A(z^{-1})$ ,  $B(z^{-1})$ ,  $C(z^{-1})$ , and  $D(z^{-1})$  are polynomials:

$$\begin{aligned} A(z^{-1}) &= 1 + a_1z^{-1} + a_2z^{-2}, \\ B(z^{-1}) &= b_1z^{-1} + b_2z^{-2}, \\ C(z^{-1}) &= 1 + c_1z^{-1}, \\ D(z^{-1}) &= d_1z^{-1}. \end{aligned}$$

A digital control law at step  $k$  for parallel feedback loops of the crane position and payload sway angle is proposed as follows:

$$u(k) = \frac{(k_1 + k_2z^{-1})e(k) + (k_3 + k_4z^{-1})\alpha(k)}{1 - k_5z^{-1}} \tag{23}$$

where  $e(k) = x(k) - x_r(k)$ ,  $x(k)$  is approximated by (5);  $x_r(k)$  is the reference signal at time  $k$ ; and  $k_1, k_2, k_3, k_4$ , and  $k_5$  are the controller parameters calculated using the pole placement method. In this case, the 5th-order closed-loop characteristic polynomial can be presented as

$$\delta(z^{-1}) = qSK \tag{24}$$

where

$$q = [1, z^{-1}, \dots, z^{-5}],$$

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ a_1 + c_1 & 0 & 0 & 0 & 0 & -1 \\ a_1c_1 - a_1 + a_2 - c_1 & t_s d_1 & 0 & -b_1 d_1 & 0 & -a_1 - c_1 + 1 \\ -a_1c_1 + a_2c_1 - a_2 & t_s a_1 d_1 & t_s d_1 & (b_1 - b_2)d_1 & -b_1 d_1 & -a_1c_1 + a_1 - a_2 + c_1 \\ -a_2c_1 & t_s a_2 d_1 & t_s a_1 d_1 & b_2 d_1 & (b_1 - b_2)d_1 & a_1c_1 - a_2c_1 + a_2 \\ 0 & 0 & t_s a_2 d_1 & 0 & b_2 d_1 & a_2c_1 \end{bmatrix},$$

$$K = [1, k_1, k_2, \dots, k_5]^T.$$

The desired polynomial  $\delta(z^{-1})$  can be obtained by assuming the desired location of the closed-loop poles  $\lambda_i$ :

$$\delta(z^{-1}) = \prod_{i=1}^{n=5} (1 - \lambda_i z^{-1}) \tag{25}$$

In [56], the desired closed-loop poles were assumed as  $\lambda_i = \exp(-\omega_n t_s)$ , where  $\omega_n = \sqrt{g/l}$  is the natural undamped frequency of the payload suspended on a rope of length  $l$ , while  $g = 9.81 \text{ ms}^{-2}$  is the acceleration gravity.

### 3. Results

The laboratory experimental setup hardware is illustrated in Figure 3. The double-girder crane bridge is driven by two 0.18 kW AC gear motors supplied by SV004iC5-1F 0.4kW frequency inverters (LS Industrial Systems Co., Ltd., Anyang-si, South Korea) that are controlled by the voltage signal  $u$  within the range  $\pm 10V$ . The incremental encoders with a resolution of 400, 2000, and 100 ppr (pulses per rotation) are used to measure position  $x$ , sway angle  $\alpha$ , and rope length  $l$ , respectively. The encoders are installed on the wheels to measure position  $x$  relative to the crane bridge, under the trolley connected to a pair of fork arms embracing the hoisting cable to measure sway angle  $\alpha$ , and on the hoisting drum to measure rope length  $l$ . The data from the sensors were sampled at 10 Hz using a PC (16 GB RAM, Quad Core 4 GHz Intel Core i7-6700 K CPU, Intel, Santa Clara, CA, USA) equipped with the IO card PCI1710HG, MS Windows 10 operating system, and Matlab software release R2020. During the experiments, the PLC was used to transmit the signals from the sensors to the PC and the control signal  $u$  to the actuators. A constant velocity Kalman filter was used to estimate the velocity of the crane  $v$  with the process noise and measurement noise that follow Gaussian distributions  $N(0, \sigma_p^2)$  and  $N(0, \sigma_m^2)$ , where  $\sigma_p = 8$  and  $\sigma_m = 0.2$ .

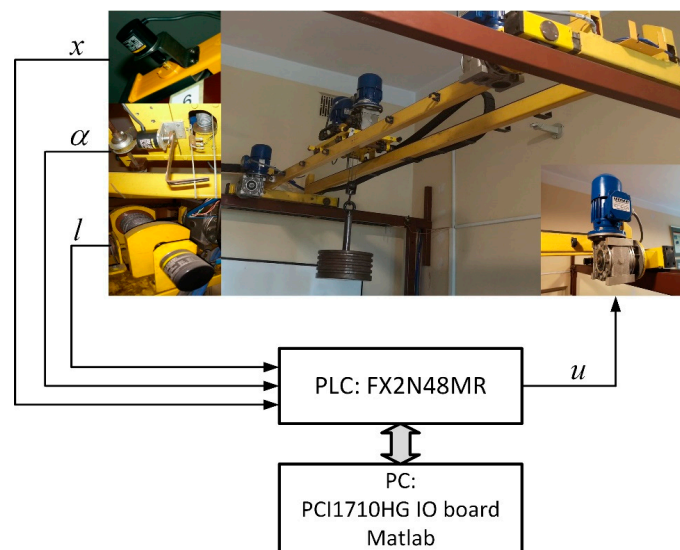
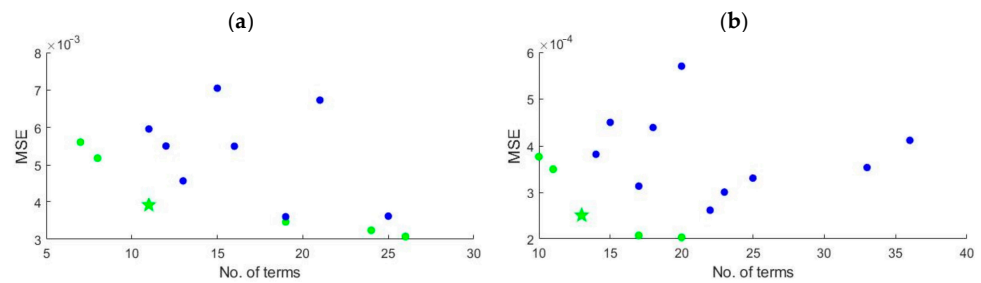


Figure 3. Experimental setup.

The identification model was obtained from input–output experimental data collected from a series of 10 experiments carried out at different operating points, with rope lengths  $l = \{0.8, 1.1, 1.4, 1.7, 2.0\}$  m and payload masses  $m = \{10, 30, 50\}$  kg. The input excitation signal  $u$  was a sequence of step functions with varying amplitude. The data from the experiment carried out with  $l = 1.1$  m and  $m = 50$  kg and the experiment carried out with  $l = 1.7$  m and  $m = 10$  kg were used as the testing data, while the rest were used as the training data to evolve the GP-NARX model. Similarly to [43], the training data are split randomly with a 70:30 ratio into two sets, where the first data set is used in the GP algorithm to determine the MSE used as the fitness function, while the second data set is used for model selection, i.e., the model with the lowest MSE on the second training data set is stored as the best model. The GP algorithm was run 15 times with the grammar and parameters given in Tables 1 and 2, respectively, and the best model in each run was saved. The accuracy and complexity of the best models from each run are shown in Figure 4, with the green points corresponding to models on the Pareto frontier and the green pentagram representing the selected model. Nonlinear regressors  $\phi_1$  and  $\phi_2$  and coefficients  $\theta_1$  and  $\theta_2$  of the crane velocity (3) and pendulum sway (4) models, respectively, are presented in Table 3.



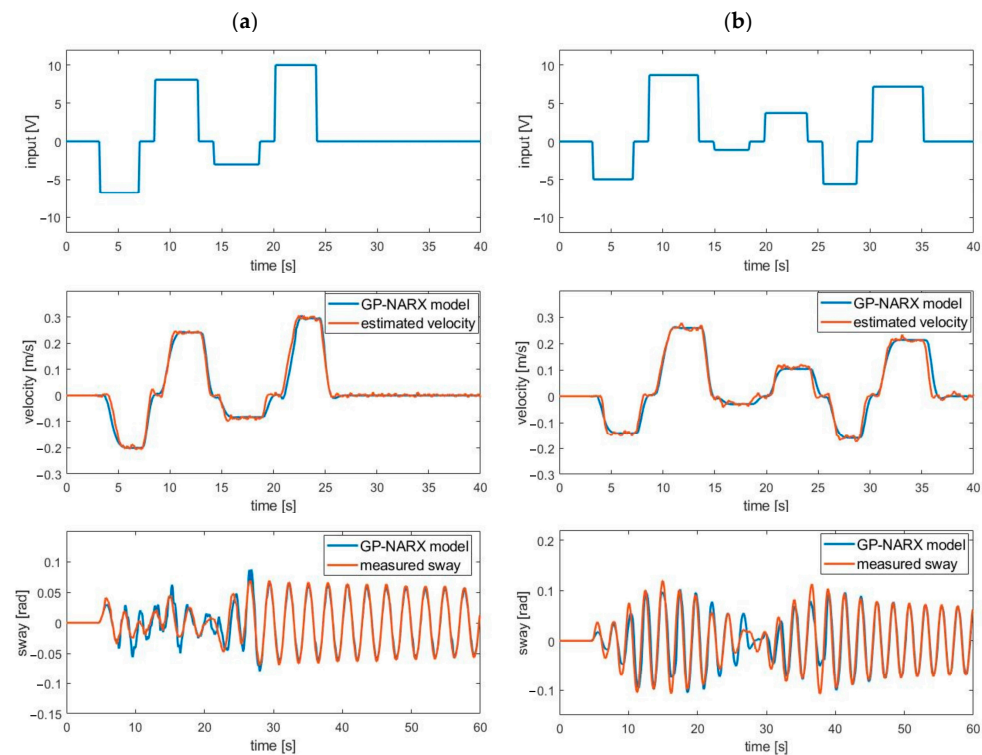
**Figure 4.** Comparison of GP-NARX models’ performance and complexity in 20-step-ahead prediction of (a) crane velocity and (b) payload sway (the green points correspond to models on the Pareto frontier, while the green pentagram point represents the selected model).

**Table 3.** Crane velocity and sway model terms and coefficients.

Velocity Model (3)		Sway Model (4)	
Coefficients $\theta_1$	Regressors $\phi_1$	Coefficients $\theta_1$	Regressors $\phi_2$
0.0464	$u(k - 7)$	-29.5814	$\alpha(k - 2)^3$
0.0508	$u(k - 3)u(k - 4)u(k - 9)$	0.0208	$v(k - 3)$
0.9514	$v(k - 1)$	22.8790	$\alpha(k - 1)\alpha(k - 2)\alpha(k - 3)$
0.1691	$v(k - 4)u(k - 1)u(k - 4)$	0.0364	$l^2\alpha(k - 2)$
0.1263	$v(k - 4)u(k - 1)u(k - 2)u(k - 3)u(k - 6)$	9.9522	$l\alpha(k - 1)\alpha(k - 3)\alpha(k - 4)$
-0.1294	$v(k - 4)$	-7.8066	$\alpha(k - 2)\alpha(k - 3)\alpha(k - 4)$
0.0608	$v(k - 4)u(k - 5)^2$	0.4542	$\alpha(k - 2)$
-0.1743	$v(k - 2)v(k - 4)^2u(k - 1)u(k - 4)$	-0.0205	$v(k - 1)$
-0.0904	$v(k - 3)u(k - 4)u(k - 5)$	0.8585	$\alpha(k - 1)$
-0.1309	$u(k - 1)u(k - 2)u(k - 4)$	-288.7701	$l^2\alpha(k - 1)^2\alpha(k - 2)\alpha(k - 3)\alpha(k - 4)$
0.1115	$u(k - 4)$	33.9964	$l\alpha(k - 1)^2\alpha(k - 4)$
		-17.9259	$l^2\alpha(k - 1)^2\alpha(k - 4)$
		-0.5518	$\alpha(k - 4)$

The GP-NARX model had 11 model terms selected for the velocity submodel (3), while the sway submodel (4) had 13 model terms selected. The mass of the payload did not appear in any of the terms, making the models independent of the payload mass. Figure 5 presents the input signal  $u$  applied to excite the underactuated part of the overhead crane in the two experiments carried out to obtain testing data at operating points  $\{l = 1.1$  m,  $m = 50$  kg $\}$  and  $\{l = 1.7$  m,  $m = 10$  kg $\}$ , and the 20-step-ahead prediction (crane velocity and payload sway angle) performance of the GP-NARX models. The models were then experimentally

verified with the NMPC-GP control scheme in terms of robustness against rope length and payload mass variation, as well as anti-disturbance performance.



**Figure 5.** GP-NARX models' performance on testing data: 20-step-ahead prediction of crane velocity and payload sway angle at operating points (a)  $\{l = 1.1 \text{ m}, m = 50 \text{ kg}\}$  and (b)  $\{l = 1.7 \text{ m}, m = 10 \text{ kg}\}$ .

The NMPC with the GP-NARX prediction model and the PSO solver, denoted NMPC-GP controller, was tested in experiments carried out with a sample time  $t_s = 0.1 \text{ s}$  and under the constraints on the control signal in the range  $-8 \text{ V} \leq u \leq 8 \text{ V}$ . The prediction and control horizon were established as  $N_p = 20$  and  $N_c = 5$ , respectively, while the swarm size and number of iterations of the PSO were set in the experiments as  $n_s = 40$  and  $n_t = 100$ , respectively. The error tolerance for the crane and payload positioning was assumed to be  $\pm 0.02 \text{ m}$ ; thus, the desired range of magnitude of the residual sway was approximated in the controller as  $\alpha_r \approx 0.02/l \text{ rad}$ . The weighting coefficients  $\lambda_{20} = 70$  and  $\lambda_3 = 0.05$  of the fitness function (15) were selected empirically during the experiments. The NMPC algorithm with the PSO solver was implemented using a C-MEX S-function in Matlab R2020. The mean execution time and standard deviation for 100 runs of the NMPC algorithm with the above-mentioned settings were 2.72 and 0.0424 milliseconds, respectively.

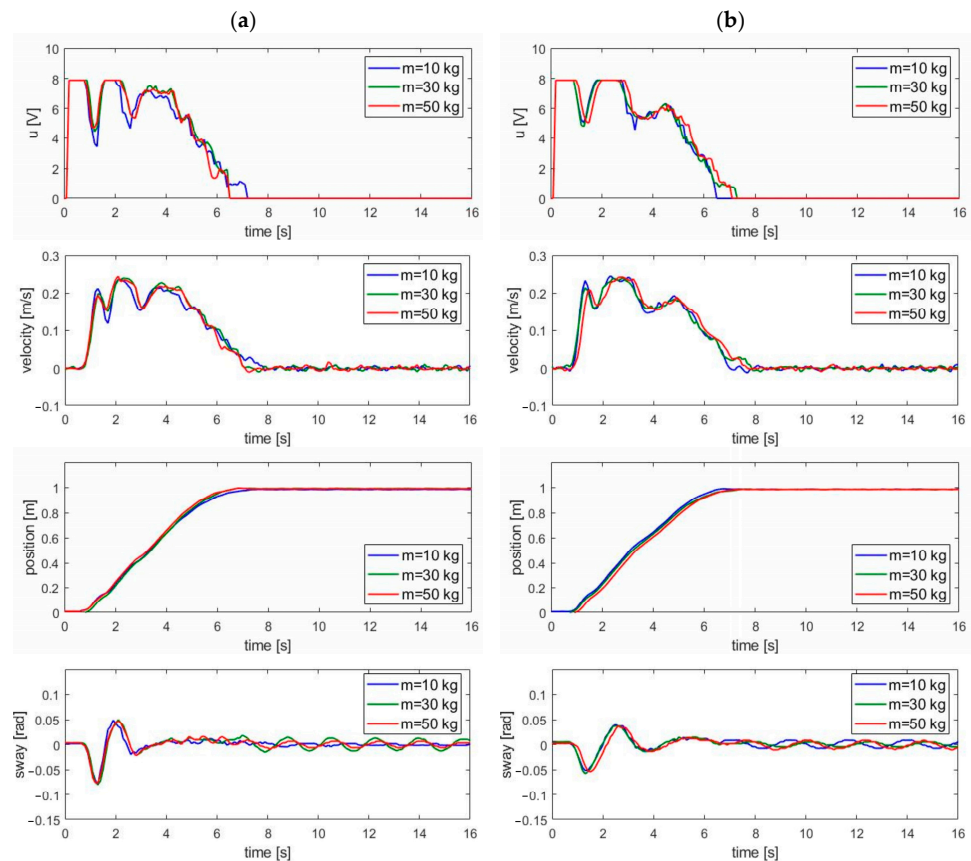
Three cases of experiments were carried out to investigate the performance and prove the feasibility and effectiveness of the developed control approach in terms of the controller's robustness against rope length and payload mass variation, in the case of an external disturbance affecting the pendulum, and by comparing the NMPC-GP with a linear discrete-time controller. The controller's performance was tested with both the operating points considered during the identification experiments, with rope lengths and payload masses not included in the training data.

In Case 1, the experiments were performed for different operating points corresponding to varying rope lengths ( $l = \{0.8, 1.4, 2.0\}$  m) and payload masses ( $m = \{10, 30, 50\}$  kg), while the desired set point was  $x_r = 1 \text{ m}$ . The control performance of the NMPC-GP controller is presented in Table 4 in terms of rise time  $t_r$ , settling time  $t_s$ , maximum magnitude of the pendulum oscillation  $\alpha_{max}$ , and residual sway  $\alpha_r$ , which is also converted to linear deviation from an equilibrium point. In addition, Table 4 includes data on the maximum deviation and residual deviation of the rope-suspended payload ( $d_{max}$  and  $d_r$ , respectively).

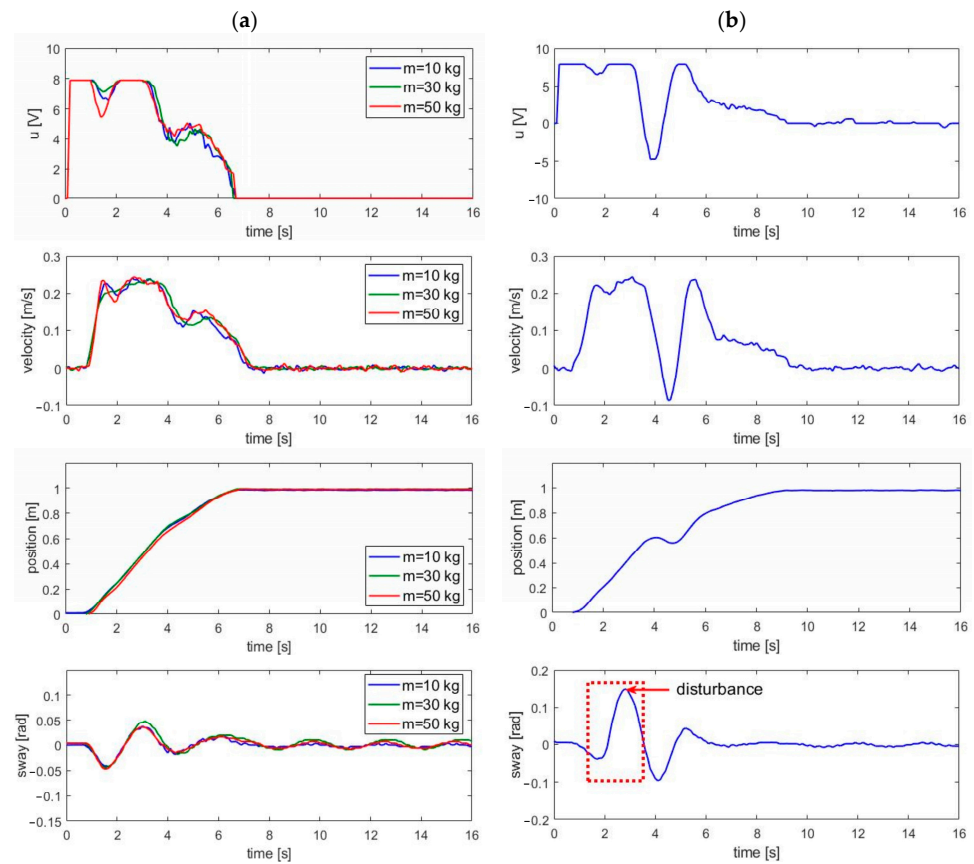
The control input  $u$ , determined by the NMPC-GP controller, as well as the crane velocity, position, and payload sway angle are depicted in Figures 6 and 7a. The crane position rise time is within 5.4–5.8 s, while the settling time range is within 6.4–7.2 s; after this time, the residual oscillation remains within the expected range (the linear payload deviation from equilibrium point is less than 0.02 m). The maximum value of the payload sway angle varies with rope length, while the influence of the payload mass is minor within the range tested during the experiments.

**Table 4.** Performance of NMPC-GP in terms of rope length and payload mass variation.

Rope Length $l$ [m]	Payload Mass $m$ [kg]	Rise Time $t_r$ [s]	Settling Time $t_s$ [s]	Maximum Sway Angle $\alpha_{max}$ [rad]	Maximum Deviation $d_{max}$ [m]	Residual Sway $\alpha_r$ [rad]	Residual Deviation $d_r$ [m]
0.8	10	5.7	7.2	0.080	0.064	0.004	0.0032
	30	5.5	6.5	0.079	0.063	0.012	0.0096
	50	5.4	6.4	0.076	0.061	0.006	0.0048
1.4	10	5.5	6.5	0.052	0.073	0.001	0.0014
	30	5.7	7.1	0.058	0.081	0.001	0.0014
	50	5.8	7.0	0.055	0.077	0.001	0.0014
2.0	10	5.7	6.6	0.044	0.088	0.003	0.006
	30	5.7	6.5	0.044	0.088	0.008	0.016
	50	5.8	6.6	0.046	0.092	0.008	0.016



**Figure 6.** NMPC-GP performance for  $m = \{10, 30, 50\}$  kg and (a)  $l = 0.8$  m and (b)  $l = 1.4$  m.



**Figure 7.** NMPC-GP performance for (a)  $m = \{10, 30, 50\}$  kg and  $l = 2.0$  m, and (b) in the presence of a disturbance (between 1.8 and 2.8 s, the pendulum is displaced from  $-0.04$  rad to  $0.15$  rad).

In Case 2, the performance of the NMPC-GP was verified in the presence of an external disturbance affecting the pendulum. It is presented in Figure 7b. The experiment was carried out for a rope length  $l = 2.0$  m and a payload mass  $m = 10$  kg. In the time interval between 1.8 and 2.8 s, the pendulum is displaced from  $-0.04$  rad to  $0.15$  rad, the control signal varies between  $-4.8$  V and  $8$  V (corresponding to a velocity change between  $-0.09$  m/s and  $0.24$  m/s), and the crane and payload positions are settled at the desired value with a settling time of 9 s with the suppression of the suspended payload oscillation within the allowable tolerance range of below 0.01 rad.

In Case 3, the proposed NMPC-GP is compared with the DFC developed using the pole placement method in Section 2.3. The comparison was carried out for two operating points not included in the training of the GP-NARX model: rope length  $l = 1.1$  m and  $l = 1.7$  m, and payload mass  $m = 10$  kg. Using input–output data obtained from the experiment carried out with a rope length  $l = 1.1$  m, a discrete-time output-error model identification was performed to parametrize models (13) and (14), as follows:

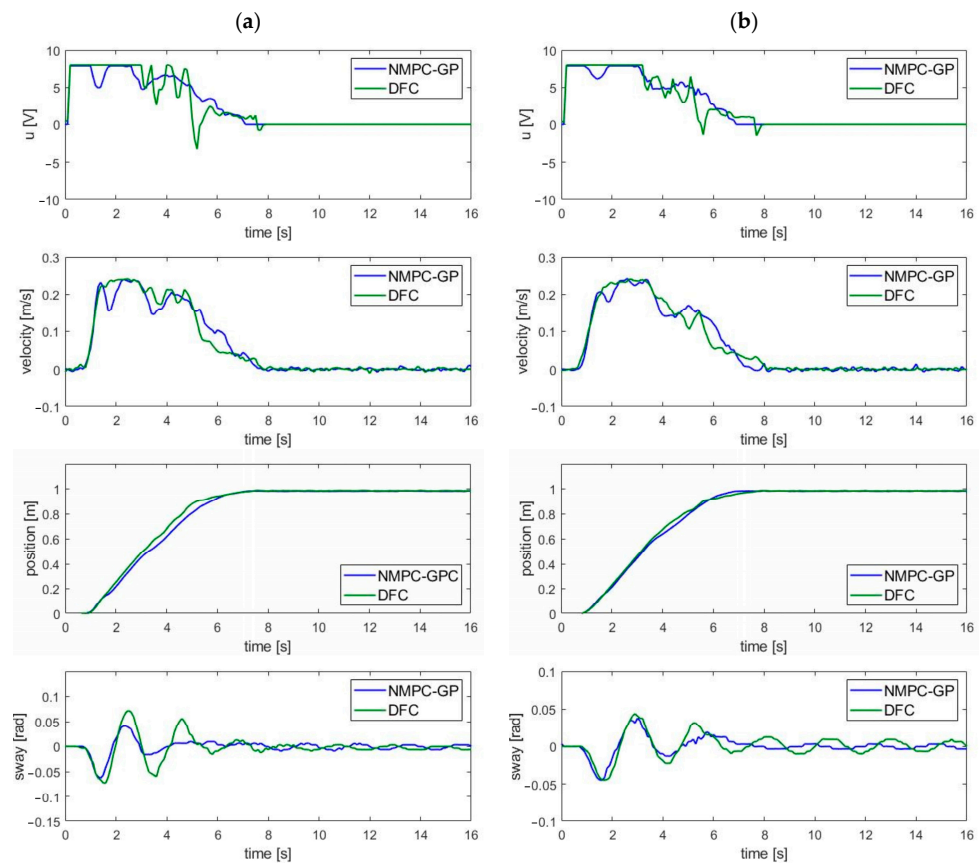
$$v(k) = \frac{0.0031z^{-1}}{1 - 0.8942z^{-1}}u(k), \alpha(k) = \frac{-0.1409z^{-1} + 0.1423z^{-2}}{1 - 1.9302z^{-1} + 0.9999z^{-2}}u(k) \quad (26)$$

The parameters of control law (15) were determined using the pole placement method as  $k_1 = 162.471$ ,  $k_2 = -141.906$ ,  $k_3 = 394.426$ ,  $k_4 = -364.362$ , and  $k_5 = 0.1078$ , by choosing the desired closed-loop poles as  $\lambda_i = 0.7865$  ( $i = 1, 2, \dots, 5$ ) to obtain a similar settling time for the crane position as in the case of the NMPC-GP. The control performances are compared in Table 5 and Figure 8. The DFC performs similarly to the NMPC-GP in the experiment with  $l = 1.1$  m in terms of settling time of crane position and payload residual oscillation. However, the NMPC-GP outperforms the DFC in the reduction in payload oscillations in the transient state. The first two amplitudes of oscillation are reduced by the

NMPC-GP to 0.063 rad and 0.041 rad, respectively, compared to 0.073 and 0.072 in the case of the DFC scheme. The NMPC-GP provides faster suppression of the payload sway below 0.02 rad in less than 3 s, while the DFC controller performs it in 5 s. In the experiment with rope length  $l = 1.7$  m, there is performance deterioration of the DFC with respect to the settling time ( $t_s = 7.7$  s) when compared to the NMPC ( $t_s = 6.8$  s). The predictive controller performs better and is faster in reducing the pendulum oscillation with a residual sway of 0.003 rad compared to 0.012 rad in the case of the DFC.

**Table 5.** The performances of NMPC-GP and DFC.

Rope Length $l$ [m]	Controller	Rise Time $t_r$ [s]	Settling Time $t_s$ [s]	Maximum Sway Angle $\alpha_{max}$ [rad]	Residual Sway $\alpha_r$ [rad]
1.1	NMPC-GP	5.7	7.1	0.063	0.005
	DFC	5.2	7.3	0.073	0.005
1.7	NMPC-GP	5.8	6.8	0.045	0.003
	DFC	5.7	7.7	0.045	0.012



**Figure 8.** Comparison of the NMPC-GP and DFC at operating points (a)  $l = 1.1$  m,  $m = 10$  kg; and (b)  $l = 1.7$  m,  $m = 10$  kg.

#### 4. Conclusions

This study conducted an experimental verification of a nonlinear model predictive control with particle swarm optimization as the solver for an overhead crane with a data-driven overhead crane dynamic model. Grammar-guided genetic programming with  $l_0$  regularization was used to construct an input–output dynamic model of the overhead crane. The experimental data used for identification were obtained at 10 different operating points by varying the rope length and the payload mass. A total of 15 genetic programming

runs were carried out and compared in terms of accuracy and complexity, and a model for the NMPC was selected from the Pareto frontier.

The NMPC cost function was selected to be a quadratic function of the payload positioning error, the sway angle, and the control input, which was solved with the PSO algorithm. Due to the physical limitations of the overhead crane, constraints on the control input were implemented as box constraints in the PSO algorithm. To prove the efficacy of the proposed method, three different experimental cases were carried out.

In the first case, the proposed nonlinear model predictive controller was tested at different operating points with the rope length ranging from 0.8 to 2.0 m and the payload mass ranging from 10 to 50 kg. The crane position rise time was within 5.4–5.8 s, while the settling time range was within 6.4–7.2 s. The residual oscillation was within the acceptable tolerance level of the linear payload deviation of  $\pm 0.02$  m for all operating points. In the second case, the payload was subjected to an external disturbance during positioning in the time interval 1.8–2.8 s. The proposed controller positioned the payload at the desired location with a settling time of 9 s and had residual oscillations within the acceptable tolerance level. In the third case, the nonlinear model predictive controller was compared with a digital feedback controller developed using the pole placement method with a data-driven linear output-error model. The NMPC had better oscillation suppression than the DFC in the transient state, as well as lower residual oscillations.

This study shows that the NMPC with PSO used to minimize the cost function of a data-driven dynamic model of an overhead crane is a viable alternative for the control of overhead cranes. The proposed method was able to position the payload with varying rope lengths and payload masses, as well as in the presence of external disturbances. A comparative study with a digital feedback controller was also performed, in which the proposed method performed better in the transient state and was able to achieve a residual oscillation within the desired tolerance.

The proposed identification method was only used to identify a NARX model; however, a NARMAX model would be able to model a larger class of systems by including lagged error terms. Additionally, the crane model was assumed to have high mechanical impedance, which in some industrial settings may no longer be valid. Therefore, in future work, we aim to expand the model structure from the NARX model to the more general NARMAX model structure and incorporate the coupling between the crane velocity and the payload sway into the model. We will also consider investigating the different prediction and control horizon lengths more comprehensively.

**Author Contributions:** Conceptualization, T.K. and J.S.; Methodology, T.K. and J.S.; Software, T.K. and J.S.; Validation, T.K. and J.S.; Formal analysis, T.K. and J.S.; Investigation, T.K. and J.S.; Resources, T.K. and J.S.; Data curation, T.K. and J.S.; Writing—original draft, T.K. and J.S.; Writing—review and editing, T.K. and J.S.; Visualization, T.K. and J.S.; Supervision, J.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Abdel-Rahman, E.M.; Nayfeh, A.H.; Masoud, Z.N. Dynamics and Control of Cranes: A Review. *J. Vib. Control* **2003**, *9*, 863–908. [[CrossRef](#)]
2. 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](#)]



3. Mojallizadeh, M.R.; Brogliato, B.; Prieur, C. Modeling and Control of Overhead Cranes: A Tutorial Overview and Perspectives. *Annu. Rev. Control* **2023**, *56*, 100877. [[CrossRef](#)]
4. Kim, C.-L.; Sung, Y.-G. Robust Input Shapers for Acceleration-Limit Actuators. *Appl. Sci.* **2023**, *13*, 12499. [[CrossRef](#)]
5. Tang, W.; Ma, R.; Wang, W.; Gao, H. Optimization-Based Input-Shaping Swing Control of Overhead Cranes. *Appl. Sci.* **2023**, *13*, 9637. [[CrossRef](#)]
6. Stein, A.; Singh, T. Minimum Time Control of a Gantry Crane System with Rate Constraints. *Mech. Syst. Signal Process.* **2023**, *190*, 110120. [[CrossRef](#)]
7. Miao, X.; Yang, L.; Ouyang, H. Artificial-Neural-Network Optimal Smoother Design for Oscillation Suppression Control of Underactuated Overhead Cranes with Distributed Mass Beams. *Mech. Syst. Signal Process.* **2023**, *200*, 110497. [[CrossRef](#)]
8. Matsui, K.; Kajiwara, H. Feedforward Control Input Generation Method for a Crane System with Restrictions on Drive System. *Mech. Syst. Signal Process.* **2022**, *170*, 108865. [[CrossRef](#)]
9. Sun, N.; Yang, T.; Fang, Y.; Wu, Y.; Chen, H. Transportation Control of Double-Pendulum Cranes With a Nonlinear Quasi-PID Scheme: Design and Experiments. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1408–1418. [[CrossRef](#)]
10. Yang, Y.; Zhang, T.; Zhou, X.; Zhang, H.; Li, J. Neural Network Adaptive PID-like Coupling Control for Double Pendulum Cranes with Time-Varying Input Constraints. *Mech. Syst. Signal Process.* **2024**, *208*, 110997. [[CrossRef](#)]
11. Wang, T.; Tan, N.; Zhang, X.; Li, G.; Su, S.; Zhou, J.; Qiu, J.; Wu, Z.; Zhai, Y.; Donida Labati, R.; et al. A Time-Varying Sliding Mode Control Method for Distributed-Mass Double Pendulum Bridge Crane with Variable Parameters. *IEEE Access* **2021**, *9*, 75981–75992. [[CrossRef](#)]
12. 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](#)]
13. Zhang, M.; Zhang, Y.; Chen, H.; Cheng, X. Model-Independent PD-SMC Method With Payload Swing Suppression for 3D Overhead Crane Systems. *Mech. Syst. Signal Process.* **2019**, *129*, 381–393. [[CrossRef](#)]
14. Shen, P.-Y.; Schatz, J.; Caverly, R.J. Passivity-Based Adaptive Trajectory Control of an Underactuated 3-DOF Overhead Crane. *Control Eng. Pract.* **2021**, *112*, 104834. [[CrossRef](#)]
15. Zhang, S.; Zhu, H.; He, X.; Feng, Y.; Pang, C.K. Passivity-Based Coupling Control for Underactuated Three-Dimensional Overhead Cranes. *ISA Trans.* **2022**, *126*, 352–360. [[CrossRef](#)] [[PubMed](#)]
16. Rigatos, G.; Siano, P.; Abbaszadeh, M. Nonlinear H-Infinity Control for 4-DOF Underactuated Overhead Cranes. *Trans. Inst. Meas. Control.* **2018**, *40*, 2364–2377. [[CrossRef](#)]
17. Jiang, J.; Astolfi, A. Stabilization of a Class of Underactuated Nonlinear Systems via Under-actuated Back-Stepping. *IEEE Trans. Automat. Control* **2021**, *66*, 5429–5435. [[CrossRef](#)]
18. Kimiaghalam, B.; Homaiifar, A.; Sayarrodsari, B. An Application of Model Predictive Control for a Shipboard Crane. In Proceedings of the American Control Conference, Arlington, VA, USA, 25–27 June 2001; pp. 929–934. [[CrossRef](#)]
19. Arnold, E.; Sawodny, O.; Neupert, J.; Schneider, K. Anti-Sway System for Boom Cranes Based on a Model Predictive Control Approach. In Proceedings of the IEEE International Conference Mechatronics and Automation, Niagara Falls, ON, Canada, 29 July–1 August 2005; pp. 1533–1538. [[CrossRef](#)]
20. Neupert, J.; Arnold, E.; Schneider, K.; Sawodny, O. Tracking and Anti-Sway Control for Boom Cranes. *Control Eng. Pract.* **2010**, *18*, 31–44. [[CrossRef](#)]
21. Tysse, G.O.; Cibicik, A.; Tingelstad, L.; Egeland, O. Lyapunov-Based Damping Controller with Nonlinear MPC Control of Payload Position for a Knuckle Boom Crane. *Automatica* **2022**, *140*, 110219. [[CrossRef](#)]
22. Vukov, M.; Van Loock, W.; Houska, B.; Ferreau, H.J.; Swevers, J.; Diehl, M. Experimental Validation of Nonlinear MPC on an Overhead Crane Using Automatic Code Generation. In Proceedings of the American Control Conference, Montreal, QC, Canada, 27–29 June 2012; pp. 6264–6269. [[CrossRef](#)]
23. Schindele, D.; Aschemann, H. Fast Nonlinear MPC for an Overhead Travelling Crane. *IFAC Proc.* **2011**, *44*, 7963–7968. [[CrossRef](#)]
24. Wu, Z.; Xia, X.; Zhu, B. Model Predictive Control for Improving Operational Efficiency of Overhead Cranes. *Nonlinear Dyn.* **2015**, *79*, 2639–2657. [[CrossRef](#)]
25. Bock, M.; Kugi, A. Real-Time Nonlinear Model Predictive Path-Following Control of a Laboratory Tower Crane. *IEEE Trans. Control Syst. Technol.* **2014**, *22*, 1461–1473. [[CrossRef](#)]
26. Chen, H.; Fang, Y.; Sun, N. A Swing Constraint Guaranteed MPC Algorithm for Underactuated Overhead Cranes. *IEEE/ASME Trans. Mechatron.* **2016**, *21*, 2543–2555. [[CrossRef](#)]
27. Giacomelli, M.; Faroni, M.; Gorni, D.; Marini, A.; Simoni, L.; Visioli, A. Model Predictive Control for Operator-in-the-loop Overhead Cranes. In Proceedings of the 23rd International Conference on Emerging Technologies and Factory Automation, Turin, Italy, 4–7 September 2018; pp. 589–596. [[CrossRef](#)]
28. Smoczek, J.; Szytka, J. Particle Swarm Optimization-Based Multivariable Generalized Predictive Control for an Overhead Crane. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 258–268. [[CrossRef](#)]
29. Ljung, L. *System Identification—Theory for the User*, 2nd ed.; Prentice-Hall: Upper Saddle River, NJ, USA, 1999.
30. 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. [[CrossRef](#)]
31. Zhu, X.; Wang, N. Hairpin RNA Genetic Algorithm Based ANFIS for Modeling Overhead Cranes. *Mech. Syst. Signal Process.* **2022**, *165*, 108326. [[CrossRef](#)]

32. Kim, G.-H.; Yoon, M.; Jeon, J.Y.; Hong, K.-S. Data-Driven Modeling and Adaptive Predictive Anti-swing Control of Overhead Cranes. *Int. J. Control Autom. Syst.* **2022**, *20*, 2712–2723. [[CrossRef](#)]
33. 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. [[CrossRef](#)]
34. Lin, H.; Lou, X. Data-Driven Active Learning Control for Bridge Cranes. *Math. Comput. Appl.* **2023**, *28*, 101. [[CrossRef](#)]
35. 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. [[CrossRef](#)]
36. Bao, H.; Kang, Q.; An, J.; Ma, X.; Zhou, M. A Performance-Driven MPC Algorithm for Underactuated Bridge Cranes. *Machines* **2021**, *9*, 177. [[CrossRef](#)]
37. Wang, X.; Karimi, H.R.; Shen, M.; Liu, D.; Li, L.-W.; Shi, J. Neural network-based event-triggered data-driven control of disturbed nonlinear systems with quantized input. *Neural Netw.* **2022**, *156*, 152–159. [[CrossRef](#)] [[PubMed](#)]
38. Shen, M.; Wang, X.; Park, J.H.; Yi, Y.; Che, W.-W. Extended Disturbance-Observer-Based Data-Driven Control of Networked Nonlinear Systems With Event-Triggered Output. *IEEE Trans. Syst. Man. Cybern.* **2023**, *53*, 3129–3140. [[CrossRef](#)]
39. Yang, T.; Sun, N.; Chen, H.; Fang, Y. Adaptive Optimal Motion Control of Uncertain Underactuated Mechatronic Systems with Actuator Constraints. *IEEE/ASME Trans. Mechatron.* **2023**, *28*, 210–222. [[CrossRef](#)]
40. Zhang, H.; Zhao, C.; Ding, J. Online Reinforcement Learning with Passivity-Based Stabilizing Term for Real Time Overhead Crane Control Without Knowledge of the System Model. *Control Eng. Pract.* **2022**, *127*, 105302. [[CrossRef](#)]
41. Kuszniar, 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](#)]
42. Muñoz, L.; Trujillo, L.; Silva, S. Transfer Learning in Constructive Induction with Genetic Programming. *Genet. Program. Evolvable Mach.* **2020**, *21*, 529–569. [[CrossRef](#)]
43. Kuszniar, T.; Smoczek, J.; Karwat, B. Data-Driven Identification of Crane Dynamics Using Regularized Genetic Programming. *Appl. Sci.* **2024**, *14*, 3492. [[CrossRef](#)]
44. Natarajan, B.K. Sparse Approximate Solutions to Linear Systems. *SIAM J. Comput.* **1995**, *24*, 227–234. [[CrossRef](#)]
45. Li, H.; Lin, Z. Accelerated Proximal Gradient Methods for Nonconvex Programming. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 379–387.
46. Barzilai, J.; Borwein, J.M. Two-Point Step Size Gradient Methods. *IMA J. Numer. Anal.* **1988**, *8*, 141–148. [[CrossRef](#)]
47. Searson, D.P. 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 International Publishing: Cham, Switzerland, 2015; pp. 551–573. [[CrossRef](#)]
48. Luke, S. Two Fast Tree-Creation Algorithms for Genetic Programming. *IEEE Trans. Evol. Comput.* **2000**, *4*, 274–283. [[CrossRef](#)]
49. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948. [[CrossRef](#)]
50. Clerc, M. From Theory to Practice in Particle Swarm Optimization. In *Handbook of Swarm Intelligence*; Panigrahi, B.K., Shi, Y., Lim, M.-H., Eds.; Adaptation, Learning, and Optimization; Springer: Berlin/Heidelberg, Germany, 2011; Volume 8, pp. 3–36. [[CrossRef](#)]
51. Engelbrecht, A.P. Particle Swarm Optimization: Global Best or Local Best? In Proceedings of the IEEE BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence, Ipojuca, Brazil, 8–11 September 2013; pp. 124–135. [[CrossRef](#)]
52. Liang, J.J.; Suganthan, P.N. Dynamic Multi-Swarm Particle Swarm Optimizer. In Proceedings of the 2005 IEEE Swarm Intelligence Symposium, SIS 2005, Pasadena, CA, USA, 8–10 June 2005; pp. 124–129. [[CrossRef](#)]
53. Peng, C.-C.; Chen, C.-H. Compensatory Neural Fuzzy Network with Symbiotic Particle Swarm Optimization for Temperature. *Control. Appl. Math. Model.* **2015**, *39*, 383–395. [[CrossRef](#)]
54. Wang, D.; Tan, D.; Liu, L. Particle Swarm Optimization Algorithm: An Overview. *Soft Comput.* **2018**, *22*, 387–408. [[CrossRef](#)]
55. Eberhart, R.C.; Shi, Y. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. In Proceedings of the Congress on Evolutionary Computation, La Jolla, CA, USA, 16–19 July 2000; pp. 84–88. [[CrossRef](#)]
56. Smoczek, J. Interval Arithmetic-Based Fuzzy Discrete-Time Crane Control Scheme Design. *Bull. Pol. Acad. Sci.* **2013**, *61*, 863–870. [[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.